obtaining at least one additional feature element of each of the second components ascertained to not be structures of interest, wherein the at least one additional feature elements are radiometric attributes;

obtaining at least one additional feature element of each of the second components ascertained to be structures of interest;

analyzing the at least one additional feature elements of the second components ascertained to be structures of interest in view of the at least one additional feature elements of the second components ascertained to not be structures of interest; and

removing second components ascertained to not be of interest based on the analyzing.

11. The method of claim 10, further comprising:

generating, with the processor, a rule for removing second components that are not associated with structures of interest in the at least one input overhead image using at least one additional feature element of the detected first components and the at least one additional feature element of the second components ascertained to not be of interest, wherein each of the at least one additional feature elements are radiometric attributes.

12. The method of claim 11, further comprising:

obtaining at least one additional attribute of each of the second components ascertained to be associated with structures of interest in the at least one input overhead image;

applying, using the processor, the rule against the at least one additional attribute of each of the second components ascertained to be associated with structures of interest in the at least one input overhead image to remove second components that are not associated with structures of interest in the at least one input overhead image.

13. The method of claim 1, wherein the dissimilarity measure is a maximum intensity difference.

14. The method of claim 13, wherein the at least one input overhead image comprises a multispectral image including a plurality of spectral bands, and wherein the dissimilarity measure is a maximum intensity difference among all of the plurality of spectral bands.

15. A system for extracting structures of interest from overhead imagery, comprising:

a construction engine, executable by a processor, that decomposes a plurality of pixels of an input overhead image associated with a geographic area into a plurality of components of a hierarchical data structure, wherein each pixel in each group is adjacent to at least one other pixel in the group, wherein adjacent pixels in each group do not differ by more than a predetermined level of a dissimilarity measure between the adjacent pixels, wherein a plurality of such groups of pixels at each of a plurality of successive predetermined levels of the dissimilarity measure are identified, and wherein each group is a first component;

a differential area profile (DAP) engine, executable by the processor, that:

tags each of the first components to identify one of a plurality of area zones within which the first component resides based on a number of pixels present in the first component, wherein each successive area zone includes first components having a greater number of pixels than do the first components of a previous area zone;

ZESTY.AI EX1002 (Part 2 of 3)
U.S. Patent No. 10,529,029

generates, for each pixel of the input overhead image, a vector including a number of entries corresponding to each of the area zones, wherein each entry comprises a difference between a lowest dissimilarity measure level of a first component within which the pixel is disposed in the corresponding area zone and a lowest dissimilarity measure level of a first component within which the pixel is disposed in a subsequent area zone; and

derives, for each pixel of the input overhead image, at least one attribute of the pixel based on the vector of the pixel.

16. The system of claim 15, wherein the DAP engine derives at least one attribute of each pixel by determining a largest entry in the vector corresponding to the pixel, wherein the largest entry is a lifetime attribute.

17. The system of claim 16, wherein the DAP engine derives at least one attribute of each pixel by determining the area zone of the largest entry in the vector corresponding to the pixel, wherein the area zone of the largest entry is a scale attribute.

18. The system of claim 17, wherein the DAP engine derives at least one attribute of each pixel by determining the lowest predetermined level of the dissimilarity measure in the area zone of the largest entry in the vector corresponding to the pixel, wherein the lowest predetermined level of the dissimilarity measure in the area zone of the largest entry in the vector corresponding to the pixel is a dissimilarity attribute.

19. The system of claim 16, further comprising:

a non-target clustering engine, executable by the processor, that detects first components in the hierarchical data structure that each include pixels collectively comprising a geometric attribute level greater than a threshold geometric attribute level; obtains at least one attribute of each of the detected first components; ascertains whether the at least one attribute of each of the detected first components is associated with a structure of interest in the at least one input overhead image; and generates a mask corresponding to the image layer based on the detected first component ascertaining step that includes first regions that are ascertained to be associated with structures of interest and second regions that are ascertained to not be associated with structures of interest.

20. The system of claim 19, wherein the detected first components comprise lower predetermined levels of the dissimilarity measure than other of the predetermined levels of the dissimilarity measure.

21. The system of claim 20, wherein the geometric attribute comprises geographical area.

22. The system of claim 20, wherein the geometric attribute comprises non-compactness.

23. The system of claim 20, further comprising a mapping engine that:

generates an image layer based on the at least one attributes of the pixels;

applies the mask over the image layer to remove regions of the image layer that are not associated with structures of interest; and

maps resulting portions of the image layer into a resultant image that is associated with the geographical region.

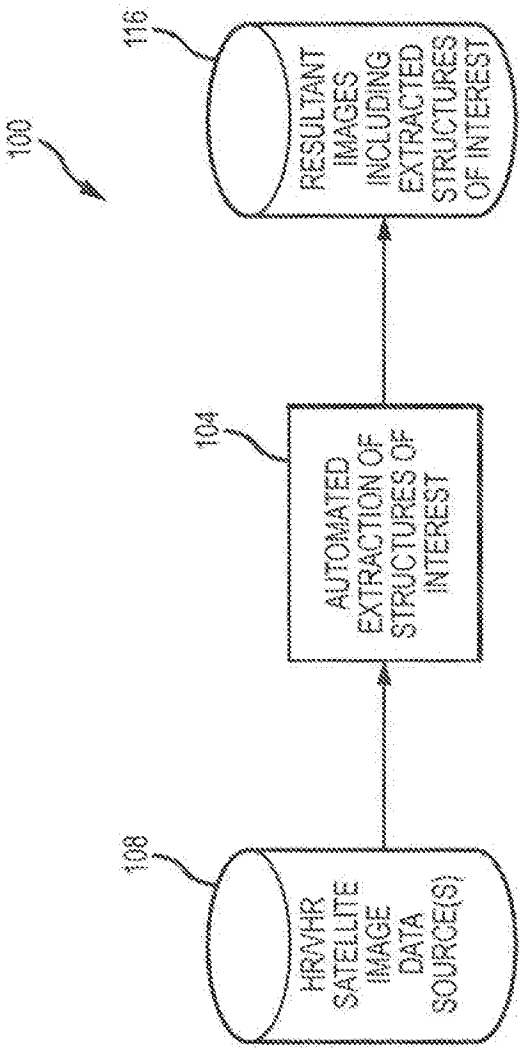

100
108
HR/VHR SATELLITE IMAGE DATA SOURCE(S)
104
AUTOMATED EXTRACTION OF STRUCTURES OF INTEREST
116
RESULTANT IMAGES INCLUDING EXTRACTED STRUCTURES OF INTEREST

FIG.1

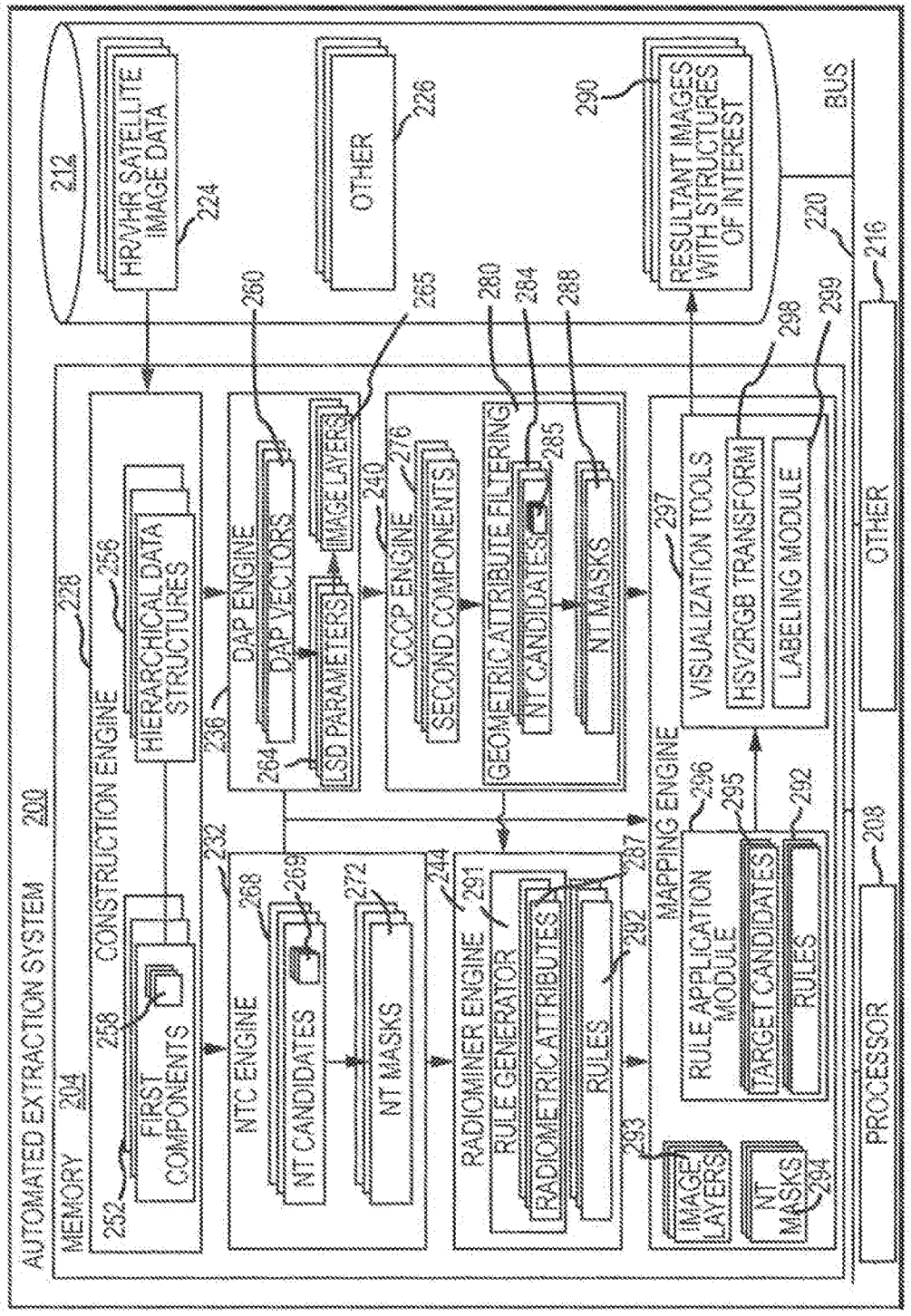

AUTOMATED EXTRACTION SYSTEM 200
MEMORY 204
232
FIRST COMPONENTS
CONSTRUCTION ENGINE
HIERARCHICAL DATA STRUCTURES
228
236
NTC ENGINE
NT CANDIDATES
NT MASKS
RADIOMINER ENGINE
RULE GENERATOR
RADIOMETRIC ATTRIBUTES
RULES
MAPPING ENGINE
IMAGE LAYERS
NT MASKS
RULE APPLICATION MODULE
TARGET CANDIDATES
RULES
DAP ENGINE
DAP VECTORS
LSD PARAMETERS
IMAGE LAYERS
CCCP ENGINE
SECOND COMPONENTS
GEOMETRIC ATTRIBUTE FILTERING
NT CANDIDATES
NT MASKS
VISUALIZATION TOOLS
HSV2RGB TRANSFORM
LABELING MODULE
OTHER
PROCESSOR
BUS
212
HR/VHR SATELLITE IMAGE DATA
224
OTHER
226
RESULTANT IMAGES WITH STRUCTURES OF INTEREST
290
220
216
208
252
268
269
272
244
291
287
282
233
236
295
292
264
260
265
240
276
280
285
284
288
297
298
299

FIG. 2

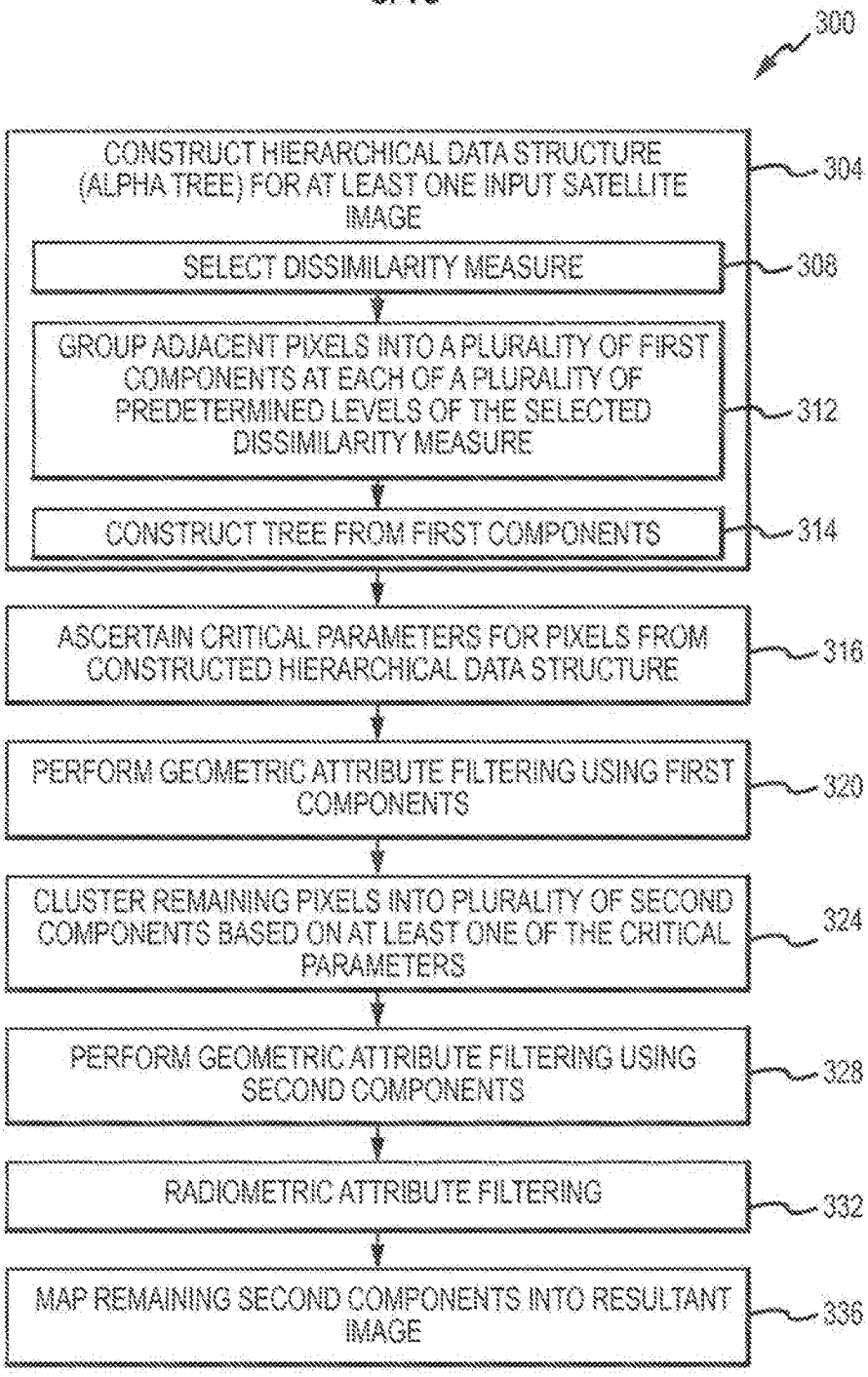

300
CONSTRUCT HIERARCHICAL DATA STRUCTURE (ALPHA TREE) FOR AT LEAST ONE INPUT SATELLITE IMAGE
304
SELECT DISSIMILARITY MEASURE
308
GROUP ADJACENT PIXELS INTO A PLURALITY OF FIRST COMPONENTS AT EACH OF A PLURALITY OF PREDETERMINED LEVELS OF THE SELECTED DISSIMILARITY MEASURE
312
CONSTRUCT TREE FROM FIRST COMPONENTS
314
ASCERTAIN CRITICAL PARAMETERS FOR PIXELS FROM CONSTRUCTED HIERARCHICAL DATA STRUCTURE
316
PERFORM GEOMETRIC ATTRIBUTE FILTERING USING FIRST COMPONENTS
320
CLUSTER REMAINING PIXELS INTO PLURALITY OF SECOND COMPONENTS BASED ON AT LEAST ONE OF THE CRITICAL PARAMETERS
324
PERFORM GEOMETRIC ATTRIBUTE FILTERING USING SECOND COMPONENTS
328
RADIOMETRIC ATTRIBUTE FILTERING
332
MAP REMAINING SECOND COMPONENTS INTO RESULTANT IMAGE
336

FIG.3

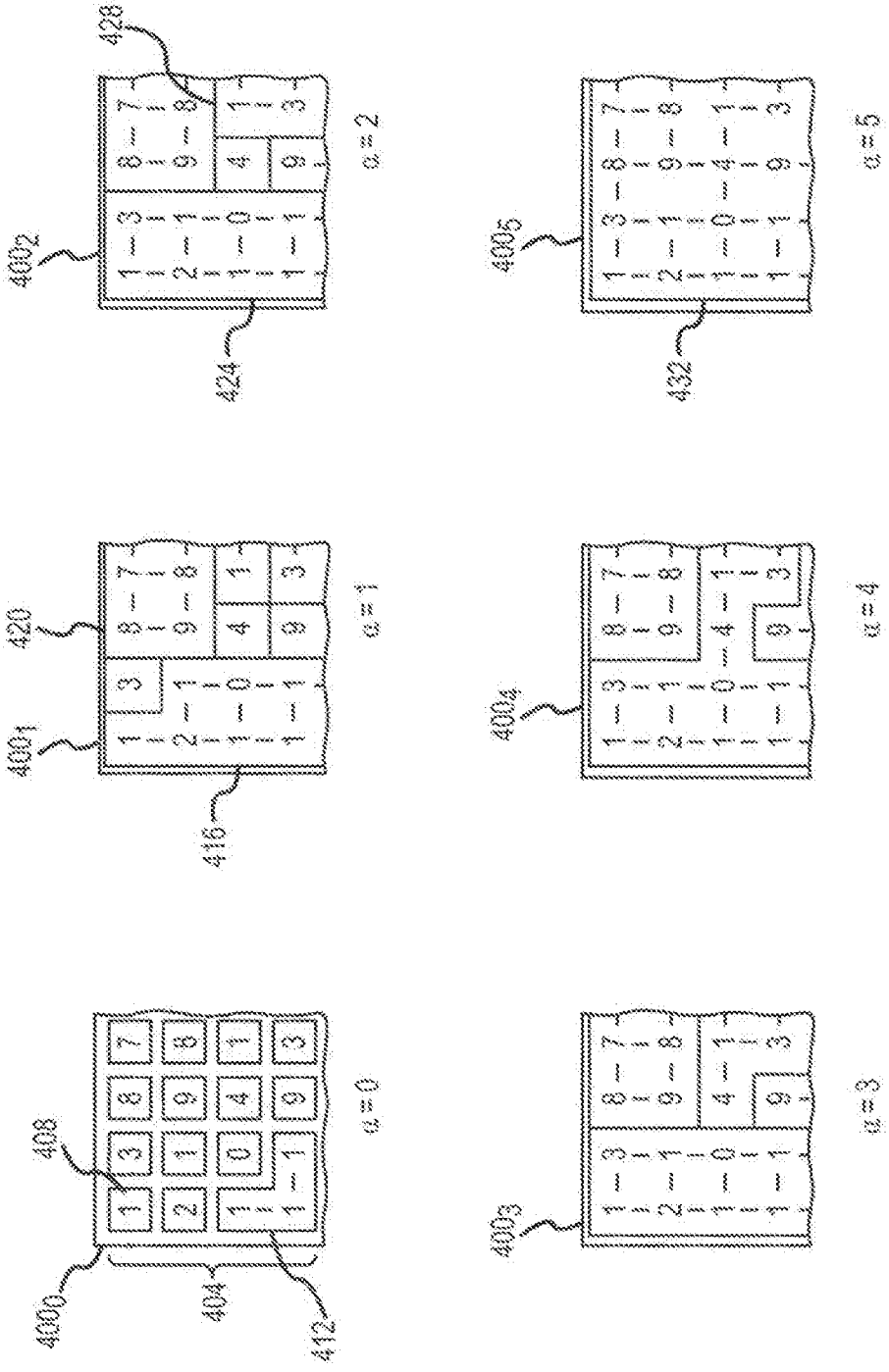

400₀
404
408
412
α = 0
400₁
416
420
α = 1
400₂
424
428
α = 2
400₃
α = 3
400₄
α = 4
400₅
432
α = 5

FIG.4

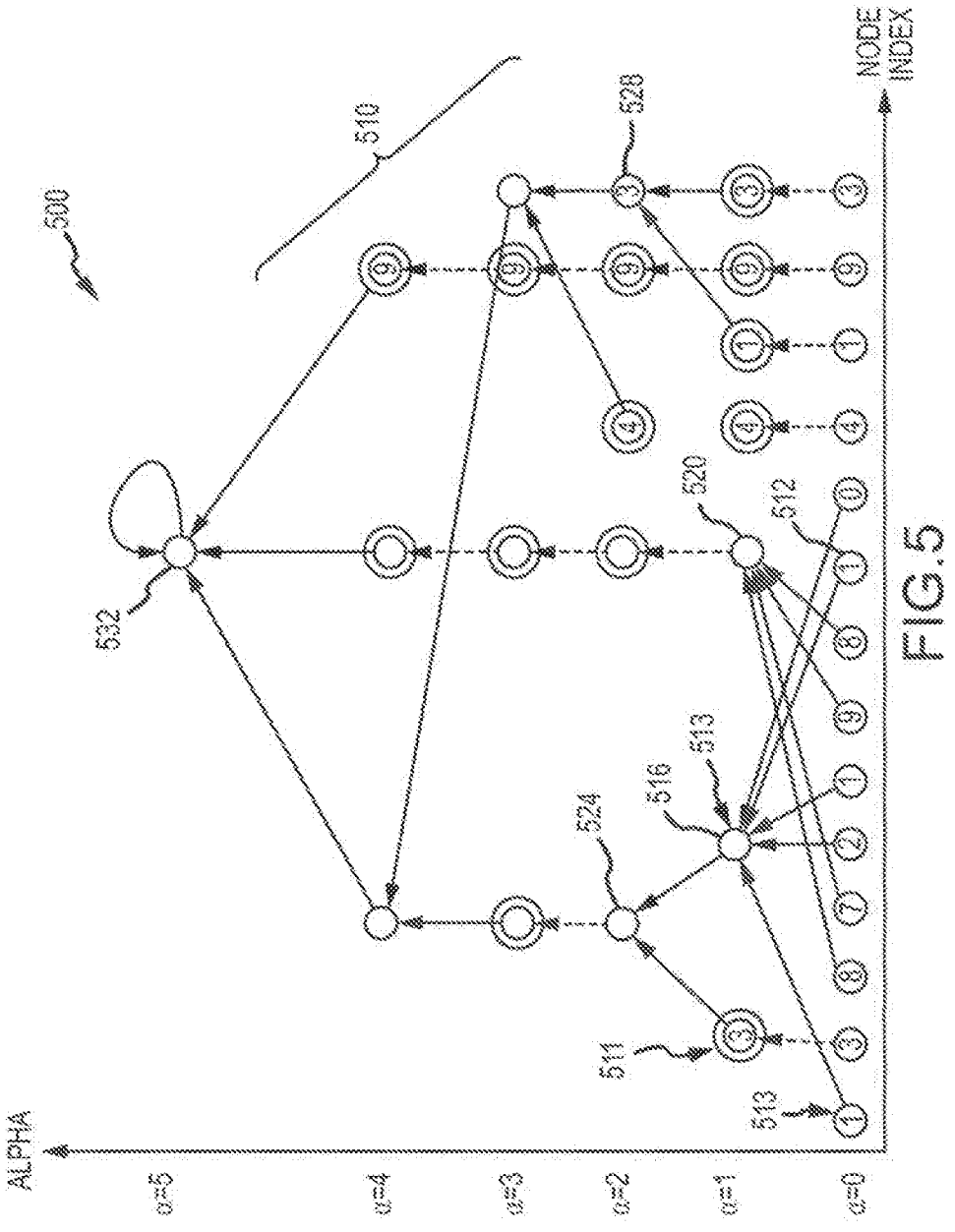

500
510
528
532
520
512
513
516
524
511
513
NODE INDEX
ALPHA
α=5
α=4
α=3
α=2
α=1
α=0

FIG.5

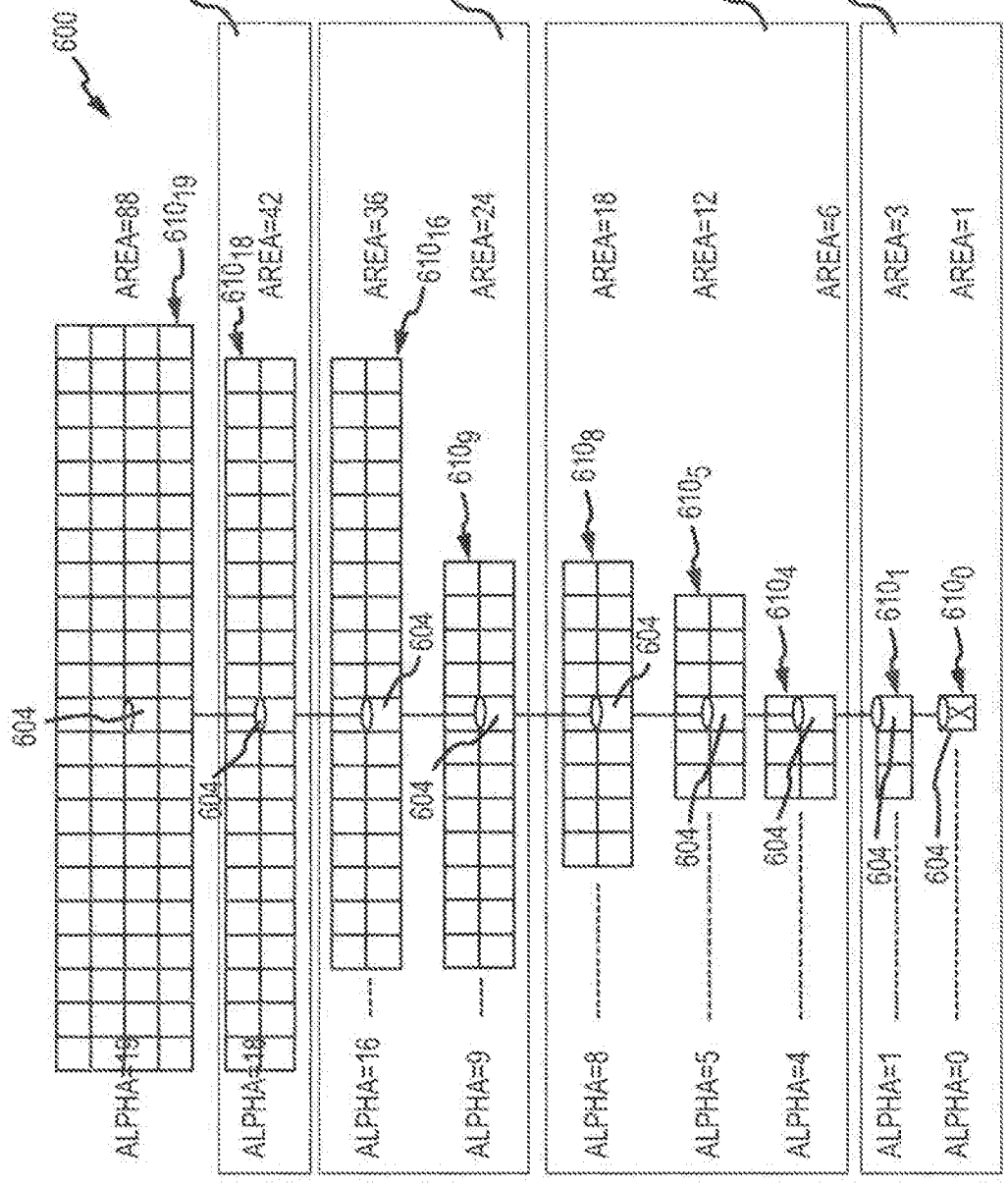

600
682
AREA ZONE 4
41-84 PIXELS
658
AREA ZONE 3
21-40 PIXELS
654
AREA ZONE 2
5-20 PIXELS
650
AREA ZONE 1
1-4 PIXELS
604
ALPHA=16
ALPHA=9
ALPHA=8
ALPHA=5
ALPHA=4
ALPHA=1
ALPHA=0
AREA=84
AREA=42
AREA=36
AREA=24
AREA=18
AREA=12
AREA=6
AREA=3
AREA=1
610_{19}
610_{18}
610_{16}
610_9
610_8
610_5
610_4
610_1
610_0

FIG.6

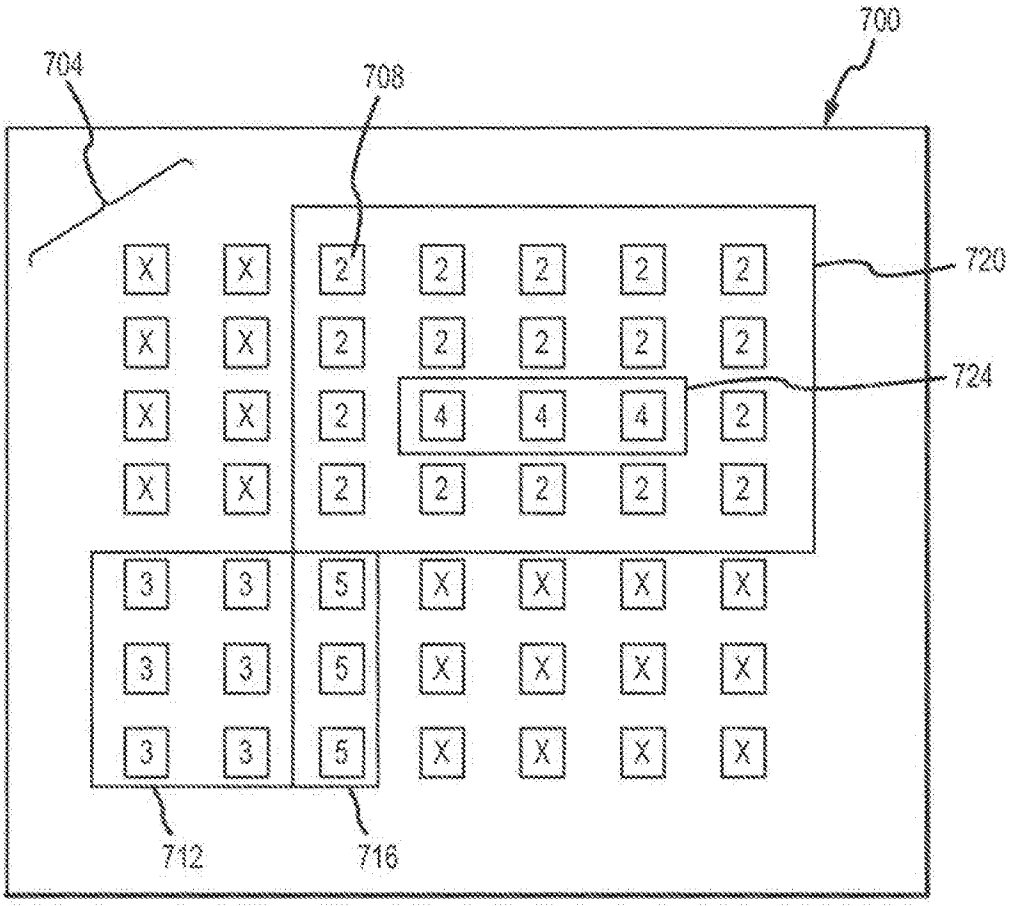


FIG.7

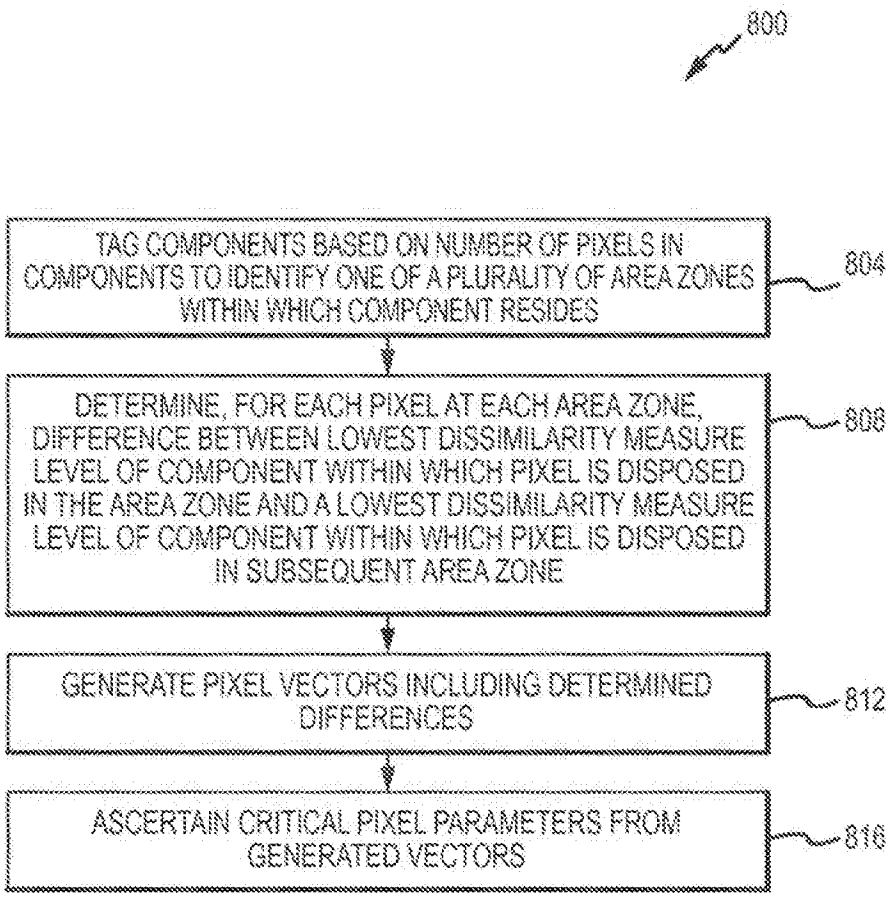

800
TAG COMPONENTS BASED ON NUMBER OF PIXELS IN COMPONENTS TO IDENTIFY ONE OF A PLURALITY OF AREA ZONES WITHIN WHICH COMPONENT RESIDES
804
DETERMINE, FOR EACH PIXEL AT EACH AREA ZONE, DIFFERENCE BETWEEN LOWEST DISSIMILARITY MEASURE LEVEL OF COMPONENT WITHIN WHICH PIXEL IS DISPOSED IN THE AREA ZONE AND A LOWEST DISSIMILARITY MEASURE LEVEL OF COMPONENT WITHIN WHICH PIXEL IS DISPOSED IN SUBSEQUENT AREA ZONE
808
GENERATE PIXEL VECTORS INCLUDING DETERMINED DIFFERENCES
812
ASCERTAIN CRITICAL PIXEL PARAMETERS FROM GENERATED VECTORS
816

FIG.8

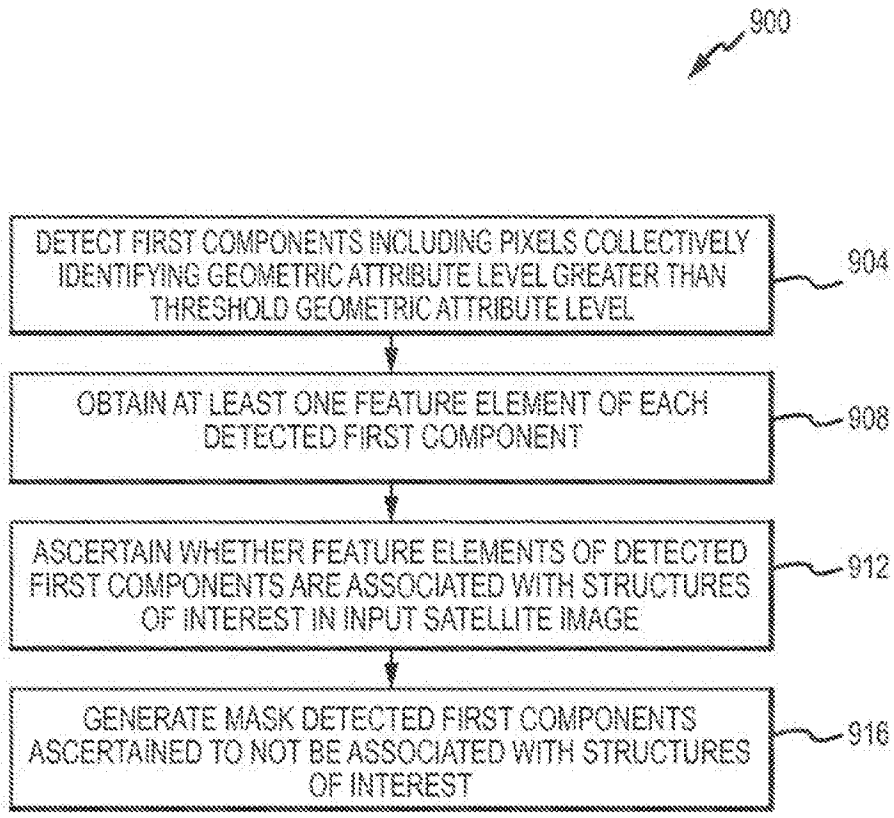

900
DETECT FIRST COMPONENTS INCLUDING PIXELS COLLECTIVELY IDENTIFYING GEOMETRIC ATTRIBUTE LEVEL GREATER THAN THRESHOLD GEOMETRIC ATTRIBUTE LEVEL
904
OBTAIN AT LEAST ONE FEATURE ELEMENT OF EACH DETECTED FIRST COMPONENT
908
ASCERTAIN WHETHER FEATURE ELEMENTS OF DETECTED FIRST COMPONENTS ARE ASSOCIATED WITH STRUCTURES OF INTEREST IN INPUT SATELLITE IMAGE
912
GENERATE MASK DETECTED FIRST COMPONENTS ASCERTAINED TO NOT BE ASSOCIATED WITH STRUCTURES OF INTEREST
916

FIG.9

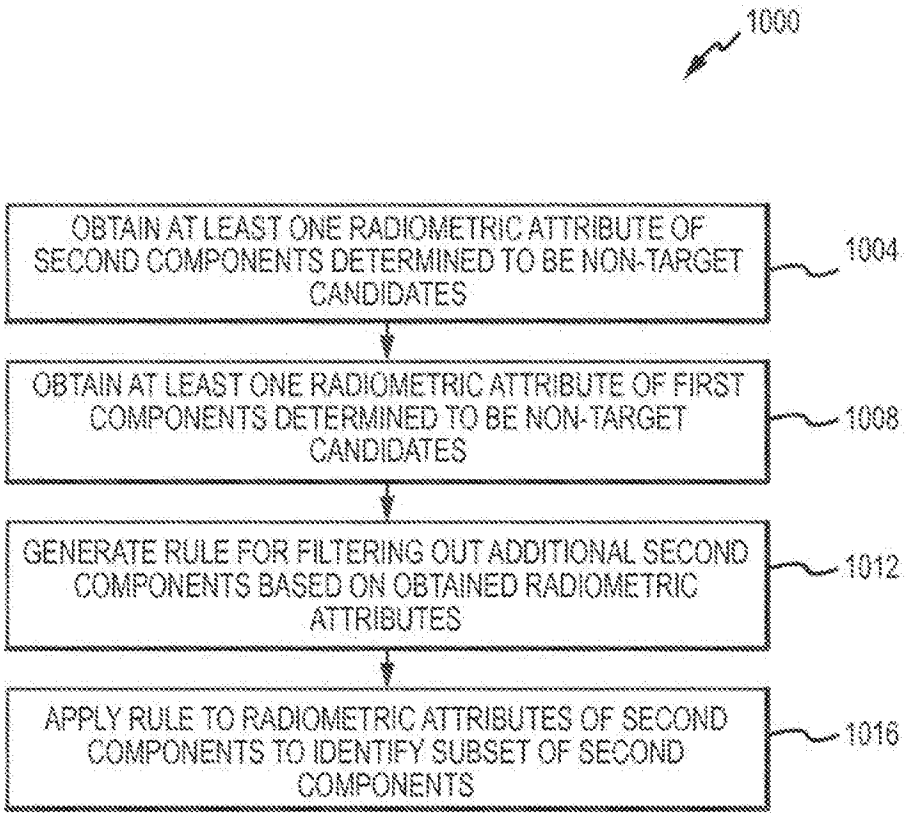

1000
OBTAIN AT LEAST ONE RADIOMETRIC ATTRIBUTE OF SECOND COMPONENTS DETERMINED TO BE NON-TARGET CANDIDATES
1004
OBTAIN AT LEAST ONE RADIOMETRIC ATTRIBUTE OF FIRST COMPONENTS DETERMINED TO BE NON-TARGET CANDIDATES
1008
GENERATE RULE FOR FILTERING OUT ADDITIONAL SECOND COMPONENTS BASED ON OBTAINED RADIOMETRIC ATTRIBUTES
1012
APPLY RULE TO RADIOMETRIC ATTRIBUTES OF SECOND COMPONENTS TO IDENTIFY SUBSET OF SECOND COMPONENTS
1016

FIG.10

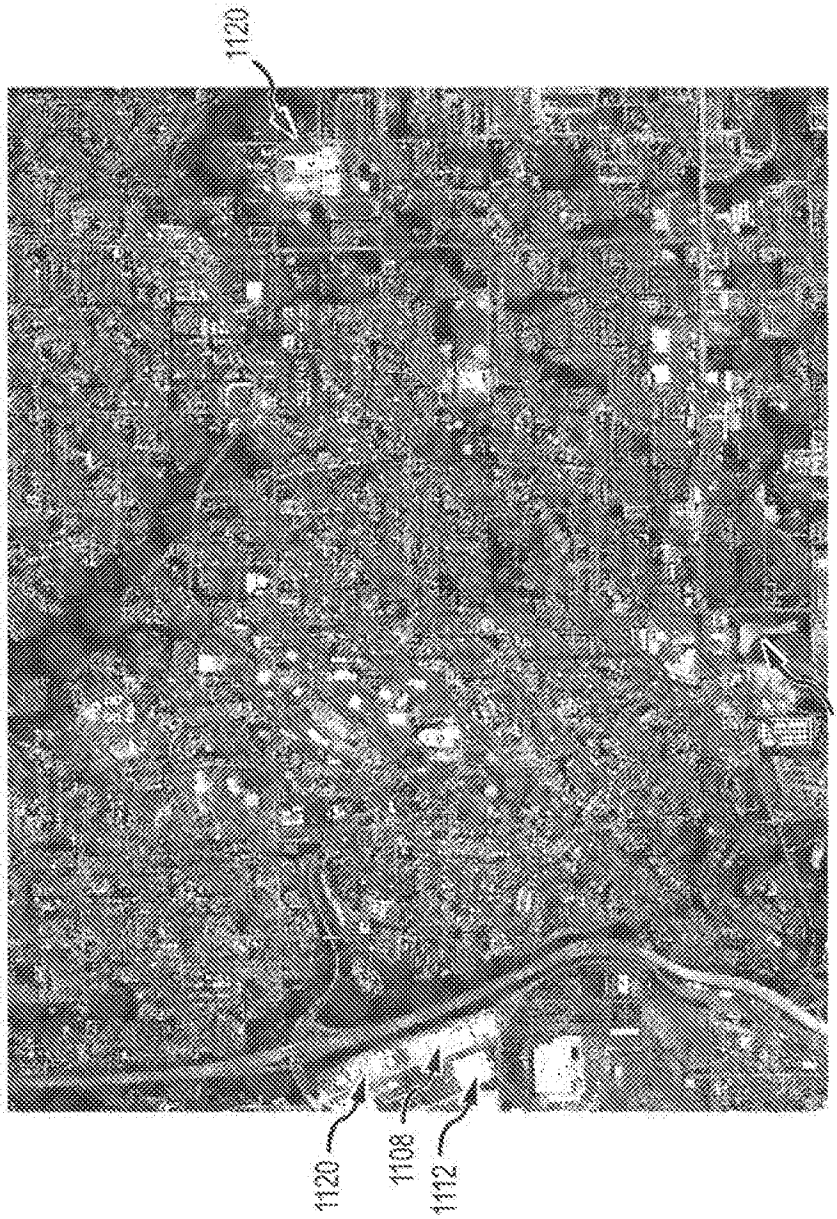


FIG. 11

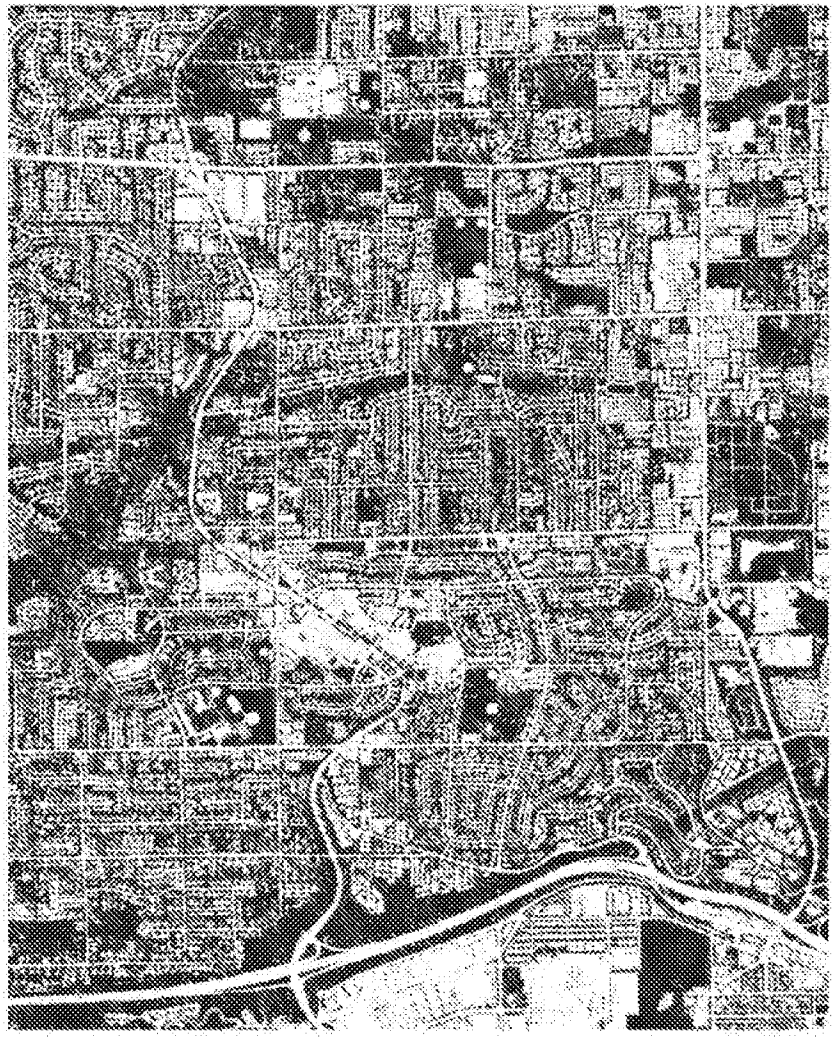


FIG. 12

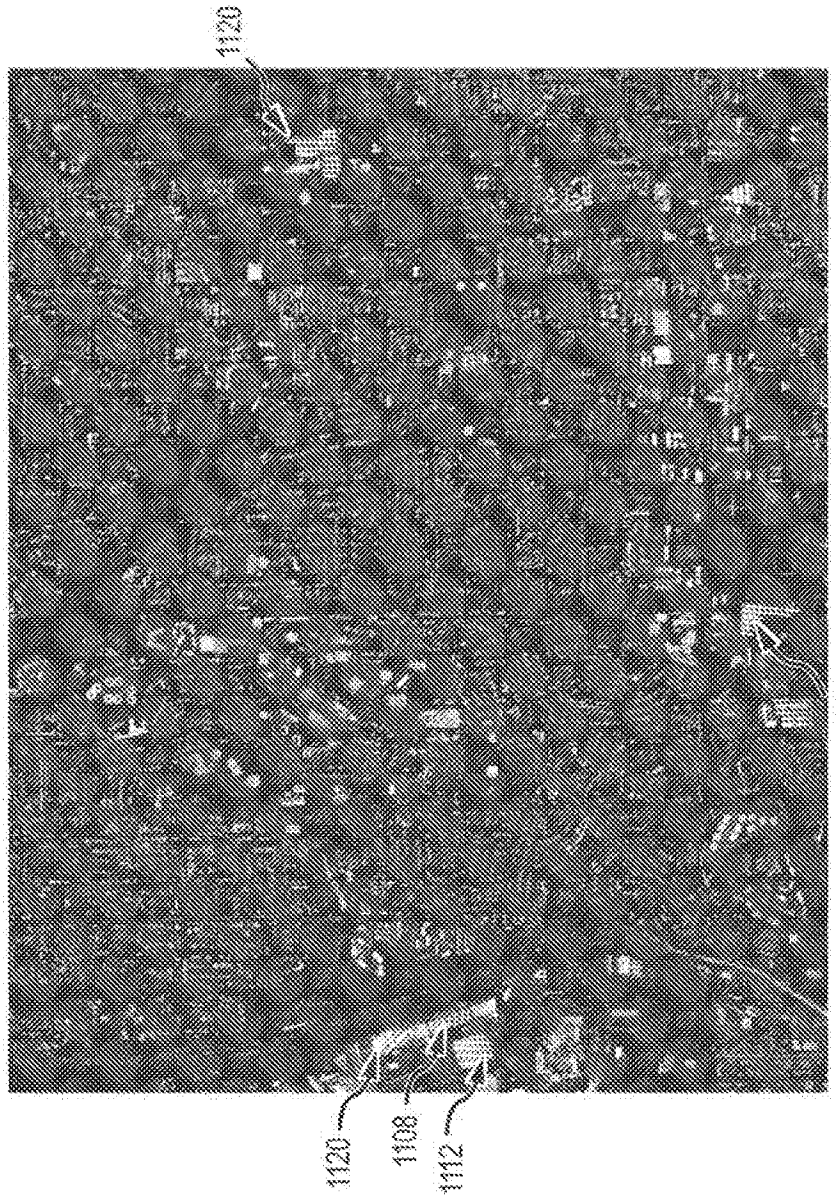


FIG. 13

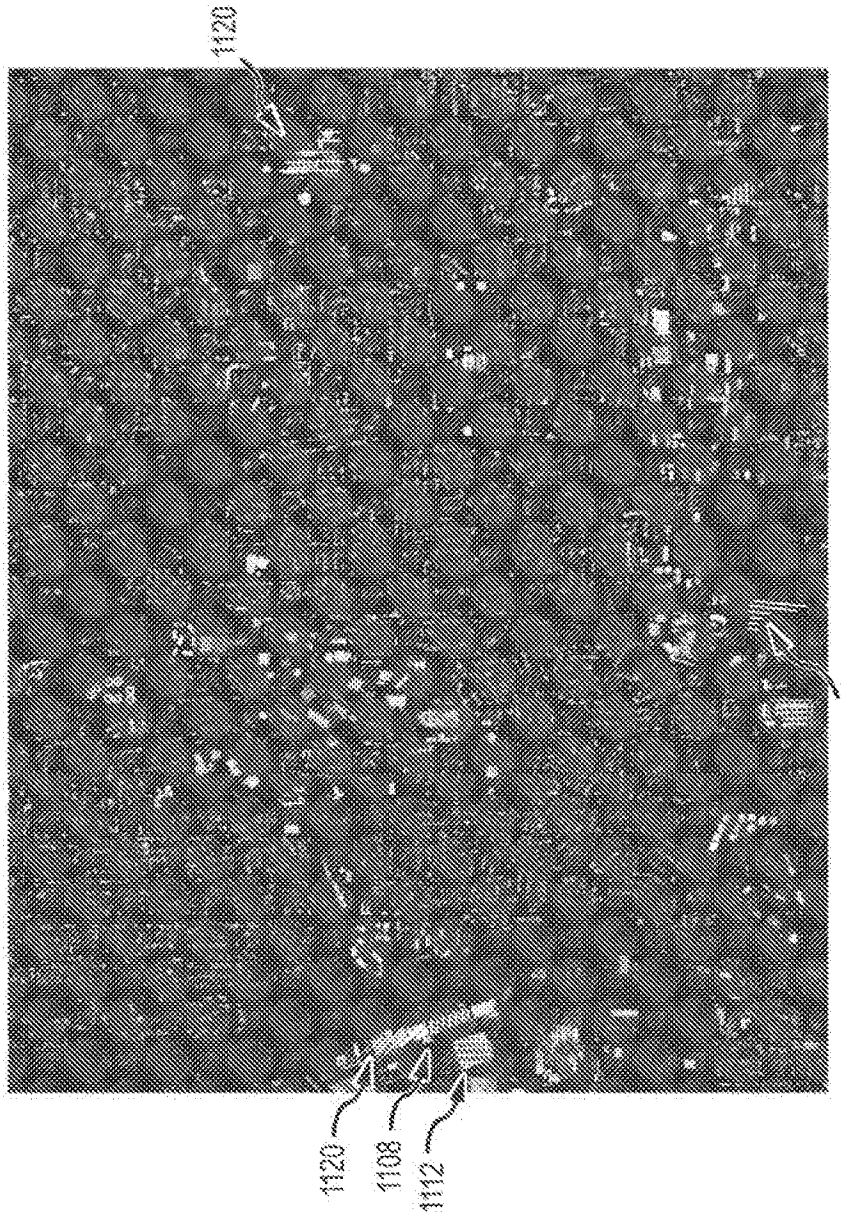


FIG.14

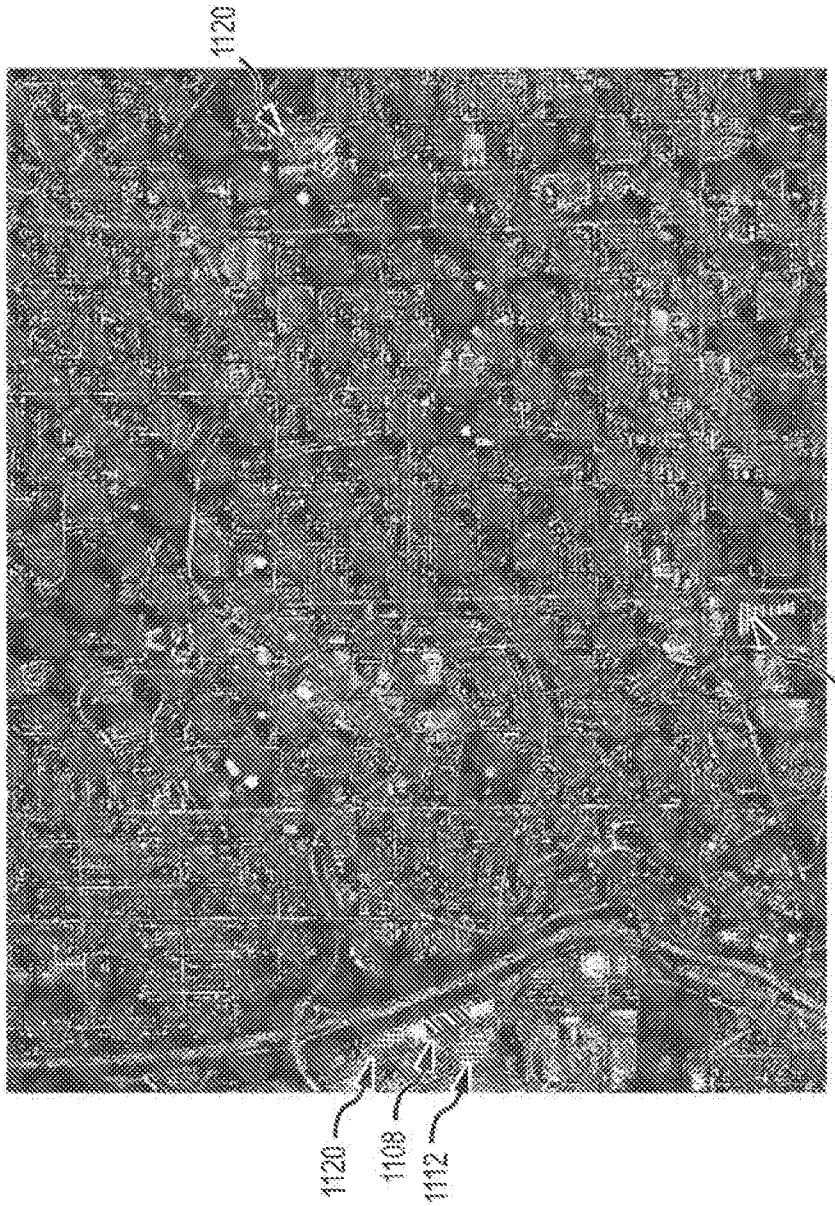


FIG. 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/046811

A. CLASSIFICATION OF SUBJECT MATTER

G06K 9/34(2006.01)i, G06K 9/46(2006.01)i, G06K 9/66(2006.01)i, G06K 9/62(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06K 9/34; H04N 7/18; G06F 15/00; H04N 5/225; G06T 7/60; G06K 9/00; G06K 9/62; G09G 5/00; G06K 9/46; G06K 9/66

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: overhead, pixel, geographical, region, identify, tag, threshold

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011-0075882 A1 (TAO GUO et al.) 31 March 2011 See paragraphs [0003]-[0004]; claims 1-2; and figure 3A.	1-23
A	US 2011-0102461 A1 (STEPHEN SCHULTZ et al.) 05 May 2011 See paragraphs [0013]-[0014]; and figure 3B.	1-23
A	US 2012-0007982 A1 (FRANK D. GIUFFRIDA et al.) 12 January 2012 See paragraph [0015]; and figure 1.	1-23
A	US 2013-0077819 A1 (WEI BU et al.) 28 March 2013 See paragraphs [0045], [0066]; and figure 5.	1-22
A	WO 2013-082280 A1 (PICTOMETRY INTERNATIONAL CORP.) 06 June 2013 See paragraph [0010]; and figure 4.	1-23

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 November 2014 (18.11.2014)	**18 November 2014 (18.11.2014)**

Name and mailing address of the ISA/KR	Authorized officer
International Application Division Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea	KANG, Sung Chul
Facsimile No. +82-42-472-7140	Telephone No. +82-42-481-8405

Form PCT/ISA/210 (second sheet) (July 2009)

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0075682 A1	31/03/2011	JP 2011-076178 A	14/04/2011
		JP 5369989 B2	04/12/2013
		US 8483425 B2	09/07/2013
US 2011-0102461 A1	05/05/2011	AU 2006-304414 A1	26/04/2007
		AU 2006-304414 B2	20/01/2011
		AU 2007-289119 A1	06/03/2008
		AU 2007-289120 A1	06/03/2008
		AU 2007-289120 B2	12/04/2012
		AU 2007-289120 C1	06/09/2012
		AU 2007-289124 A1	06/03/2008
		CA 2639689 A1	26/04/2007
		CA 2661905 A1	06/03/2008
		CA 2662312 A1	06/03/2008
		CA 2662355 A1	06/03/2008
		CA 2662355 C	06/05/2014
		CN 101506827 A	12/08/2009
		CN 101506827 B	28/11/2012
		CN 101511645 A	19/08/2009
		CN 101511645 B	19/01/2011
		EP 1956271 A2	13/08/2008
		EP 2057039 A2	13/05/2009
		EP 2057585 A2	13/05/2009
		EP 2057585 A4	13/03/2013
		EP 2057598 A2	13/05/2009
		EP 2068638 A2	17/06/2009
		EP 2070005 A2	17/06/2009
		EP 2070005 B1	02/10/2013
		JP 2009-512094 A	19/03/2009
		JP 2010-502496 A	28/01/2010
		JP 2010-503077 A	28/01/2010
		JP 2010-503078 A	28/01/2010
		JP 2010-503079 A	28/01/2010
		JP 4981135 B2	18/07/2012
		JP 5035721 B2	26/09/2012
		KR 10-1433715 B1	25/08/2014
		MX 2009002339 A	23/03/2009
		US 2007-0226015 A1	27/09/2007
		US 2007-0226123 A1	27/09/2007
		US 2008-0054656 A1	06/03/2008
		US 2008-0059656 A1	06/03/2008
		US 2008-0080298 A1	03/04/2008
		US 2008-0123994 A1	29/05/2008
		US 2008-0183507 A1	31/07/2008
		US 2009-0054252 A1	26/02/2009
		US 2009-0251987 A1	08/10/2009
		US 2010-0211405 A1	19/08/2010
		US 2011-0045131 A1	24/02/2011
		US 2012-201478 A1	09/08/2012

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2013-208996 A1	15/08/2013
		US 2014-126836 A1	08/05/2014
		US 7461874 B2	09/12/2008
		US 7564737 B2	21/07/2009
		US 7724484 B2	08/06/2010
		US 7873238 B2	18/01/2011
		US 7943310 B2	17/05/2011
		US 8081841 B2	20/12/2011
		US 8396724 B2	12/03/2013
		US 8452125 B2	28/05/2013
		US 8660382 B2	25/02/2014
		US 8788294 B2	22/07/2014
		WO 2007-047605 A2	26/04/2007
		WO 2007-047605 A3	20/09/2007
		WO 2008-028038 A2	06/03/2008
		WO 2008-028038 A3	14/08/2008
		WO 2008-028039 A2	06/03/2008
		WO 2008-028039 A3	14/08/2008
		WO 2008-028040 A2	06/03/2008
		WO 2008-028040 A3	13/11/2008
		WO 2008-028041 A2	06/03/2008
		WO 2008-028041 A3	30/10/2008
		WO 2008-028042 A2	06/03/2008
		WO 2008-028042 A3	29/05/2008
		WO 2008-028043 A2	06/03/2008
		WO 2008-028043 A3	13/11/2008
		WO 2008-028044 A2	06/03/2008
		WO 2008-028044 A9	22/04/2009
		WO 2008-028045 A2	06/03/2008
		WO 2008-028045 A3	15/05/2008
		WO 2008-028046 A2	06/03/2008
		WO 2008-028046 A3	27/11/2008
US 2012-0007982 A1	12/01/2012	AU 2011-314338 A1	24/01/2013
		AU 2011-314338 B2	22/08/2013
		AU 2011-314338 C1	27/03/2014
		CA 2804587 A1	19/04/2012
		EP 2591313 A2	15/05/2013
		MX 2013000158 A	18/03/2013
		US 2013-113831 A1	09/05/2013
		US 2013-135471 A1	30/05/2013
		US 2014-063243 A1	06/03/2014
		US 8477190 B2	02/07/2013
		WO 2012-050648 A2	19/04/2012
		WO 2012-050648 A3	21/06/2012
US 2013-0077819 A1	28/03/2013	AU 2012-227223 A1	11/04/2013
		CA 2790491 A1	23/03/2013
		NZ 602580 A	26/07/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/046811

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2013-082280 A1	06/06/2013	AU 2012-345876 A1	20/02/2014
		CA 2842944 A1	06/06/2013
		US 2013-153109 A1	20/06/2013

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 February 2015 (05.02.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/017366 A1

(51) International Patent Classification:
G06Q 50/10 (2012.01) *G06Q 50/26* (2012.01)

(21) International Application Number:
PCT/US2014/048537

(22) International Filing Date:
29 July 2014 (29.07.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/955,268 31 July 2013 (31.07.2013) US

(71) Applicant: **DIGITALGLOBE, INC.** [US/US]; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(72) Inventor: **GUEGUEN, Lionel**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(74) Agent: **MARSH FISCHMANN & BREYFOGLE LLP**; Robert J. Crouch, 8055 E. Tufts Ave., Suite 450, Denver, Colorado 80237 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

---- *with international search report (Art. 21(3))*

(54) Title: AUTOMATIC GENERATION OF BUILT-UP LAYERS FROM HIGH RESOLUTION SATELLITE IMAGE DATA

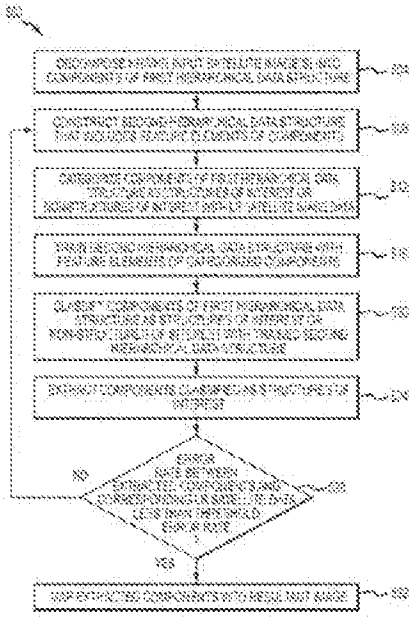


FIG.5

(57) Abstract: A system for automatically extracting interesting structures or areas (e.g., built-up structures such as buildings, tents, etc.) from HR/VHR satellite imagery data using corresponding LR satellite imagery data. The system breaks down HR/VHR input satellite images into a plurality of components (e.g., groups of pixels), organizes the components into a first hierarchical data structure (e.g., a Max-Tree), generates a second hierarchical data structure (e.g., a KD-Tree) from feature elements (e.g., spectral and shape characteristics) of the components, uses LR satellite imagery data to categorize components as being of interest or not, uses the feature elements of the categorized components to train the second data structure to be able to classify all components of the first data structure as being of interest or not, classifies the components of the first data structure with the trained second data structure, and then maps components classified as being of interest into a resultant image.

WO 2015/017366 A1

AUTOMATIC GENERATION OF BUILT-UP LAYERS FROM HIGH RESOLUTION SATELLITE IMAGE DATA

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority of U.S. Patent App. No. 13/955,268, entitled " AUTOMATIC GENERATION OF BUILT-UP LAYERS FROM HIGH RESOLUTION SATELLITE IMAGE DATA," and filed on July 31, 2013, the entire contents of which are incorporated herein by reference as if set forth in full.

BACKGROUND

[0002] Information on human settlements is crucial for a wide range of applications including emergency response, disaster risk reduction, population estimation/analysis, and urban/regional planning. Urbanization pressure generates environmental impacts, indicates population growth, and relates to risk and disaster vulnerability. For instance, the global population passed the mark of 7.0 billion in 2011 with more than half of the population living in urban areas. Between 2011 and 2050, the urban population is expected to increase by about 2.7 billion, passing from 3.6 billion in 2011 to 6.3 billion in 2050. The population growth in urban areas is projected to be concentrated in the cities and towns of the less developed countries and continents. Asia, in particular, is projected to see its urban population increase by 1.4 billion, Africa by 0.9 billion, and Latin America and the Caribbean by 0.2 billion.

[0003] Population growth is therefore becoming largely an urban phenomenon concentrated in the developing world resulting in major challenges to manage the urban development in a sustainable manner. A central issue in this respect is the availability of up-to-date information on the extent and quality of the urban settlement (e.g., the urban "build-up" or "built-up") which is largely unavailable in developing countries. For instance, cities are often growing at a pace that cannot be fully controlled by the local or regional mapping agencies. As demographic pressure increases exponentially at a global level, the ability to monitor, quantify and characterize urbanization processes around the world is becoming paramount. The information about the quality of urban development can provide precious input for understanding the vulnerability of the population living on our planet.

[0004] While satellite imagery could provide information about the world-wide built-up environment, there are few global data sets available that could be used to map the human settlements. Examples include the night-time lights of the world based on the Defense Meteorological Satellite Program - Operational Linescan System (DMSP-OLS) sensor, Moderate Resolution Imaging Spectroradiometer (MODIS) based land use/land cover classifications, and global population data sets like LandScan™ or the gridded population of the world (GPW). While the aforementioned data sets are useful for global analysis, the data sets have the tendency to under-represent small, scattered rural settlements due to the low spatial resolution of the data sets between, for instance, 500 and 2,000m. Furthermore, the data sets represent single snap-shots in time that do not allow for regular monitoring. Still further, if the data sets are updated (e.g., the LandScan data set), they are not directly comparable due to changing input sources.

SUMMARY

[0005] Although high resolution (HR, 1--10m spatial resolution) and even very high resolution (VHR, ≤ 1 m) data with an almost global coverage is or will be available with different sensors (e.g., System for Earth Observation (SPOT), China–Brazil Earth Resources Satellite program (CBERS), RapidEye, IKONOS® 2, QuickBird, WorldView-1, WorldView-2, WorldView-3), no consistent global coverage of built-up settlements derived from these datasets exists. Mapping and monitoring of urban areas at HR and VHR scales is mostly limited in terms of temporal and spatial coverage. The lack of a consistent global layer with HR/VHR spatial resolution can be attributed to a number of reasons. In one regard, global coverage is costly due to, for instance, the fact that most HR/VHR satellite missions are operated on a commercial basis. In another regard, no systems have yet been able to demonstrate the capacity to automatically extract global information layers about human settlement (built-up structures such as cities, refugee camps, etc.) from HR/VHR satellite data with relatively low levels of time, effort (e.g., low number of processor clock cycles), and other computational costs.

[0006] It has thus been determined that globally and regionally consistent information from HR and VHR input image data (e.g., multispectral, panchromatic, etc.) is needed that can be obtained inside time constraints and data constraints (e.g., in relation to resolution, volume, quality, etc.) typical of crisis management scenarios. Broadly, disclosed herein are utilities (e.g., systems, processes, etc.) for automatically extracting or isolating structures or areas of interest (e.g., built-up structures such as buildings, houses, shelters, tents, etc.) from HR/VHR satellite imagery data using corresponding LR satellite imagery data. More particularly, the disclosed utilities employ a first hierarchical data structure (e.g., a "Max-Tree") for organizing HR/VHR input images (e.g., of a particular geographic area) into a plurality of meaningful, hierarchically arranged, connected components in addition to determining one or more features or feature elements of each of the components (e.g., contrast, area, compactness, linearity, average spectral response, eccentricity or inertia of the component). The disclosed utilities also employ use of a second hierarchical data structure (e.g., a "KD-Tree") for managing organization of the plurality of feature elements (i.e., of the "feature-space" of the input image). Generally, the second hierarchical data structure offers a structured representation of the feature-space from which a classification (e.g. built-up or non built-up) of the components of the first hierarchical data structure can be directly determined. For instance, those components classified as built-up may be projected or otherwise mapped into a spatial domain of a resultant image having a high resolution (e.g., HR, VHR) with relatively low levels of computational effort.

[0007] As opposed to the classical paradigm of interactive learning followed by a time consuming model application on the spatial domain, the disclosed utilities shift the operational complexity to the feature space structuring. As a result, interactive classification of massive image data can be launched in near real-time. For instance, an experiment utilizing the disclosed utilities on an eight-band multi-spectral image (each band obtaining an image that is 10070x58734 pixels) may be concluded in 577 seconds using a 2.4GHz Intel® Core™ CPU and 8GB RAM (thus representing 14,400km²/hCPU). An interactive query of the information content may then be conducted on the classification results. When the hierarchical image and feature space data representation structures (i.e., the first and

second hierarchical data structures) are stored in memory (e.g., volatile memory, such as RAM), scene classification (subject to different criteria) can be re-iterated rapidly to offer dynamic views of what may be massive image information content.

[0008] In one aspect, a method of extracting built-up structures (e.g., and/or other structures of interest) from satellite imagery data includes decomposing, using a processor, at least one input satellite image into a plurality of components (e.g., pixels or groups of pixels) of a first hierarchical data structure (e.g., a space-partitioning data structure for organizing data points, such as a Max-Tree). The input image is associated with a geographic area (e.g., town, city, region, etc.) and has a first resolution (e.g., HR, such as 1-10m spatial resolution; VHR, such as less than 1m spatial resolution; etc.). Each of the components is characterized by at least one feature element (e.g., a k-dimensional shape and/or spectral descriptor such as contrast, area, compactness, linearity, average spectral response, standard deviation of spectral response, eccentricity or inertia of the component).

[0009] This method also includes constructing, using the processor, a second hierarchical data structure (e.g., another space-partitioning data structure for organizing data points, such as a KD-Tree) that includes a plurality of hierarchically-arranged nodes, where each of the feature elements depends from at least one of the nodes; and deriving or otherwise obtaining training components (e.g., training examples, such as 1000 components, 2000 components, etc.) from the plurality of components of the first hierarchical data structure that indicate built-up (e.g., and/or other structures of interest) and non-built-up structures (e.g., or other areas/land not of interest) in the input image using a first reference image data set (e.g., Landscan population density reference layer, MODIS-500m Global Urban Extent (GUE) information layer, National Land Cover Database (NLCD) 2006, Corine Land Cover 2006, etc.) that is associated with the geographic area and has a second resolution lower than the first resolution. The first reference image data set may be one or more relatively low-resolution satellite imagery data sets (e.g., maps, layers) of any appropriate resolution (e.g., greater than 10m spatial resolution, such as greater than 30m spatial resolution, or even greater than 100m spatial resolution) for which relatively consistent global coverage is available. For instance, one of the training components may identify a built-up structure or area when the training component falls within a built-up area as indicated by the first reference data set (e.g., when the training component includes pixels that fall within an area of increased population and/or built-up structures in the first reference data set).

[0010] In one arrangement, the method may include training, with the processor, the second hierarchical data structure with the feature elements of the training components for detecting built-up structures. As an example, various feature elements in the second hierarchical data structure that are similar and/or substantially the same as feature elements of the training components may be identified. Thereafter, the various nodes of the second hierarchical data structure that are disposed on the same leaf-path (e.g., branch(es)) as the identified feature elements may be appropriately labeled (e.g., flagged, marked, etc.) as being in the same class (e.g., built-up or non-built-up) as the corresponding feature elements of the training components. For instance, each node of the second hierarchical data structure may include a "positive" (e.g., identifying built-up or other structure(s) of interest) class counter and/or a "negative" (e.g., identifying non-built-up or structure(s) not of interest) class counter. In this regard,

the positive and/or negative counters of each node may be appropriately updated (e.g., incremented upwardly or downwardly) during the training process. In the event that subsets of feature elements in the second hierarchical data structure are clustered or grouped (e.g., are close enough with respect to some dissimilarity), any updates to class counters of the nodes along the same leaf-path as one of the feature elements may, in one embodiment, be automatically propagated to the class counters of the nodes along the same leaf-paths as one or more of the other feature elements in the clusters.

[0011] Thereafter, the method may include classifying, with the processor and using the trained second hierarchical data structure, the plurality of components of the first hierarchical data structure as identifying built-up structures or non-built-up structures. Stated differently, the method may include extracting, by the processor, those components in the first hierarchical data structure that depict built-up structures as identified by the trained second hierarchical data structure. As an example, the processor may systematically analyze and/or compare the collective set of feature elements for each component of the first hierarchical data structure in relation to the feature elements of the trained second hierarchical data structure to determine whether the component identifies a built-up structure or a non-built-up structure. For instance, a component (e.g., pixel or group of pixels) in the first hierarchical data structure may be labeled, tagged or classified as "built-up" when one or more (e.g., a majority, most, all, etc.) of its feature elements match or are similar to feature elements of the trained second hierarchical data structure associated with built-up structures (e.g., have a value within some percentage of the value of a feature element from the trained second hierarchical data structure).

[0012] In one arrangement, the method may include mapping or projecting components of the plurality of components that identify built-up structures as classified during the classifying step into a resultant image that is associated with the geographic area and that has a third resolution greater than the second resolution. For instance, the resultant image may be presented on any appropriate display (e.g., of a laptop, tablet, smartphone, etc.) and may be similar or even substantially the same as (e.g., in relation to spatial resolution, depicted geographic area, etc.) the input image but free of non-built-up structures (e.g., or free of areas that are not of interest). Advantageously, crisis management teams may be able to quickly and accurately assess possible high population areas to determine appropriate response strategies.

[0013] Various measures may be taken to improve the accuracy of the aforementioned built-up detection and extraction process. In one arrangement, the method may continue to perform the deriving, training and classifying steps with respect to additional portions of the plurality of components to further refine the accuracy of the detected and extracted built-up structures that are mapped into the resultant image. For instance, the method may include deriving additional training components (e.g., at least partially or fully non-overlapping with the previous set of training components, such as from a different portion of the input satellite image(s)) from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the input image using the first reference image data set; training, with the processor, the second hierarchical data structure with the feature elements of the additional training components for detecting built-up structures; and classifying, with the processor

and using the second hierarchical data structure as trained with the additional training components, the plurality of components of the first hierarchical data structure as identifying built-up structures or not identifying built-up structures.

[0014] In another arrangement, the method may include ascertaining error rates (e.g., equal error rates (EER), etc.) between components that identify built-up structures from the classifying step and corresponding portions of a second reference data set (e.g., a low spatial resolution data set the same as or different than the first reference image data set, and that is assumed to have consistent, non-obscured global coverage) that identify built-up structures; and mapping components that identify built-up structures that are associated with error rates below a threshold error rate into a resultant image that is associated with the geographic areas and that has a third resolution that is greater than the second resolution. In one variation, built-up structure components may be extracted from successive input images of the geographic area (e.g., obtained via heterogeneous sources, different satellites, different locations, different times such as multitemporal images, different frequencies or wavelengths such as multispectral images, etc.) and mapped into the resultant image to allow for an iterative refinement of the resultant image over a plurality of cycles.

[0015] For instance, particular components of a first input image that have been classified as identifying built-up structures may be associated with error rates over a particular threshold when compared against corresponding portions of the second reference image data set (e.g., such as due to cloud cover or the like when the first input image was obtained). In this regard, the particular components of the first input image may be not mapped into the resultant image or at least mapped into the resultant image to a lesser degree (e.g., assigned a lower weight based how far over the threshold were the corresponding error rates). However, particular components of a second input image that have been classified as identifying built-up structures and over the same geographic vicinity as the particular components of the first input image may be associated with error rates below a particular threshold when compared against corresponding portions of the second reference image data set (e.g., such as due to the image being obtained from a high azimuth and during a partially or fully cloudless, sunny day).

[0016] In this regard, the particular components of the second input image may be mapped into the resultant image or even mapped into the resultant image to a greater degree (e.g., assigned a higher weight based how far under the threshold were the corresponding error rates). Advantageously, those built-up structure components that have been obscured for one reason or another from each successive input image (e.g., as determined by the ascertained error rates) may be at least partially excluded from the resultant image to allow the method to converge on extensive, high resolution coverage of built-up structures in the geographic area (e.g., where the resultant image includes at least some components of each of the successive input images). Stated differently, only those pixels of each of the successive images that are determined to be the "best" (e.g., as determined by the ascertained error rates) may be included in the resultant image.

[0017] In another aspect disclosed herein, a system for extracting structures of interest from optical images includes a construction engine, executable by a processor, that decomposes an input image associated with a geographic area and having a first resolution into a plurality of components, generates a first hierarchical data structure from the plurality of components that includes a plurality of k-dimensional feature elements of each of the components, and constructs a second hierarchical data structure that include a plurality of hierarchically-arranged nodes, where each of the plurality of k-dimensional feature elements depends from at least one of the nodes. The disclosed system also includes a training engine that uses feature elements of a first portion of the plurality of components of the first hierarchical data structure to train the second hierarchical data structure to detect components of the plurality of components in the first hierarchical data structure that correspond to structures of interest in the input image.

[0018] In one arrangement, the system includes a classification engine, executable by the processor, that uses the trained second hierarchical data structure to extract a second portion of the plurality of components of the first hierarchical data structure that corresponds to the structures of interest in the input image. For instance, the system may also include a mapping engine that maps the second portion into a resultant image that is associated with the geographic area and that has a third resolution that is greater than the second resolution.

[0019] Any of the embodiments, arrangements, or the like discussed herein may be used (either alone or in combination with other embodiments, arrangement, or the like) with any of the disclosed aspects. Merely introducing a feature in accordance with commonly accepted antecedent basis practice does not limit the corresponding feature to the singular. Any failure to use phrases such as "at least one" does not limit the corresponding feature to the singular. Use of the phrase "at least generally," "at least partially," "substantially" or the like in relation to a particular feature encompasses the corresponding characteristic and insubstantial variations thereof. Furthermore, a reference of a feature in conjunction with the phrase "in one embodiment" does not limit the use of the feature to a single embodiment.

[0020] In addition to the exemplary aspects and embodiments described above, further aspects and embodiments will become apparent by reference to the drawings and by study of the following descriptions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] Figure 1 is a simplified block diagram illustrating a process of extracting structures of interest from satellite imagery data.

[0022] Figure 2 is a more detailed block diagram of an automated extraction system for obtaining structures of interest from satellite imagery data.

[0023] Figure 3 is a block diagram of a Max-Tree for hierarchically arranging components of an input satellite image.

[0024] Figure 4a is a schematic diagram of a KD-Tree based spaced partitioning procedure for use in hierarchically arranging feature elements of the components of Figure 3.

[0025] Figure 4b is a KD-Tree built from the partitioning procedure illustrated in Figure 4a.

[0026] Figure 5 is a flow diagram of a method for extracting structures of interest from satellite imagery data.

[0027] Figure 6 illustrates overlapping portions of a National Land Cover Database 2006 information layer and a plurality of multispectral images acquired by the WorldView 2 satellite over a portion of Texas near Dallas, TX.

[0028] Figure 7a is a close-up of a resultant image including built-up extracted from the multispectral images of Figure 6 using the information layer of Figure 6, where the close-up is at a first level of zoom.

[0029] Figure 7b is a close-up of the information layer of Figure 6 corresponding to the resultant image close-up of Figure 7a.

[0030] Figure 8a is a resultant image close-up similar to the close-up of Figure 7a, but at a closer, second level of zoom.

[0031] Figure 8b is a close-up of the multispectral images of Figure 6 corresponding to the resultant image close-up of Figure 8a.

DETAILED DESCRIPTION

[0032] Disclosed herein are utilities (e.g., systems, processes, etc.) for automatically extracting or isolating structures or areas of interest (e.g., built-up structures such as buildings, houses, shelters, tents; agricultural areas; etc.) from HR/VHR satellite imagery data using corresponding LR satellite imagery data. The disclosed utilities employ a unique use of first and second hierarchical data structures (e.g., space-partitioning data structures for organizing data points, such as Max and KD-Trees) to break down HR and/or VHR input satellite images into a plurality of hierarchically arranged connected portions or components (e.g., groups of pixels), organize corresponding feature elements (e.g., spectral and shape characteristics) of each of the components into a manageable structure, train the manageable structure of feature elements to recognize structures of interest in the components, classify the various components of the input image with the trained feature elements, extract components classified as structures of interest from the input image, and map the extracted components into a resultant image that includes the various structures of interest. The disclosed utilities can be executed at high rates of speed and with reduced use of computational resources. The resultant images can be used in numerous contexts such as in assessing population densities, qualities of life, vulnerability factors, disaster risks, sufficiency of civil infrastructures, economic growth, poverty levels, event monitoring and evolution, and the like.

[0033] At the outset, it is noted that, when referring to the earth herein, reference is made to any celestial body of which it may be desirable to acquire images or other remote sensing information. Furthermore, when referring to a

satellite herein, reference is made to any spacecraft, satellite, and/or aircraft capable of acquiring images or other remote sensing information. Furthermore, the utilities described herein may also be applied to other imaging systems, including imaging systems located on the earth or in space that acquire images of other celestial bodies. It is also noted that the drawing figures contained herein are not necessarily drawn to scale and that such figures have been provided for the purposes of discussion and illustration only.

[0034] Generally, high resolution images of selected portions of a celestial body's surface have become a product desired and used by government agencies, corporations, and individuals. For instance, many consumer products in common use today include images of the Earth's surface, such as Google® Earth. Various types of remote sensing image collection platforms may be employed, including aircraft, earth-orbiting satellites, and the like. In the case of a consumer digital camera, as one non-limiting example, an image sensor is generally arranged in an area array (e.g., 3,000 rows of 3,000 pixels each, or 9,000,000 total pixels) which collects the image area in a single "snapshot." In the case of satellite-based imaging, as another non-limiting example, the "push-broom scanning" principle is sometimes employed whereby each image sensor includes a relatively small number of rows of a great number of pixels (e.g., 50,000 or more) in each row. Each row of pixels may be scanned across the earth to build an image line by line, and the width of the image is the product of the number of pixels in the row times the pixel size or resolution (e.g., 50,000 pixels at 0.5 meter ground resolution produces an image that is 25,000 meters wide). The length of the image is controlled by the scan duration (i.e. number of lines), which is typically settable for each image collected. The resolution of satellite images varies depending on factors such as the particular instrumentation utilized, the altitude of the satellite's orbit, and the like.

[0035] Image collection platforms (e.g., aircraft, earth-orbiting satellites, etc.) may collect or acquire various types of imagery in one or more manners. As one non-limiting example, image collection platforms may perform panchromatic collection of scenes of a celestial body which generally refers to the collection of image data across a single broad range of wavelengths (e.g., all visible light, from near infrared (NIR) to near ultraviolet (NUV), etc.). As another non-limiting example, image collection platforms may additionally or alternatively capture image data within the visible light band and include respective filters to separate the incoming light into red, green and blue portions. As a further non-limiting example, image collections platforms may additionally or alternatively perform multispectral collection of scenes of a celestial body which generally refers to the collection of image data at multiple specific spectral bands across the electromagnetic spectrum (e.g., within bands both inside and outside of the visible light range such as NIR, short wave infrared (SWIR), far infrared (FIR), etc.). For instance, a satellite may have one image sensor that is sensitive to electromagnetic radiation across only a first spectral band (e.g., the visible light band, such as a wavelength range of about 380-750 nm) in addition to one or more additional image sensors that are sensitive to electromagnetic radiation only across other spectral bands (e.g., NIR, 750-1400 nm; SWIR, 1400-3000 nm; etc.). Multi-spectral imaging may allow for the extraction of additional information from the radiance received at a satellite after being reflected from the Earth's surface (which may include atmospheric effects such as from aerosols, clouds, etc.).

[0036] As discussed previously, there are generally few global data sets available that could be used to map the human settlements, much less high resolution satellite image data sets (e.g., HR, VHR) that could be used to do so. For instance, current global data sets (e.g., MODIS 500m, Landscan, NLCD 2006, Corine Land Cover 2006, etc.) have the tendency to under-represent small, scattered rural settlements due to their low spatial resolution (e.g., between 50 and 2,000m). Furthermore, the data sets represent single snap-shots in time that do not allow for regular monitoring. Still further, if the data sets are updated, they are typically not directly comparable due to changing input sources.

[0037] In this regard, Figure 1 presents a simplified block diagram of a system 100 that may be used to generate and map regionally and globally consistent structures of interest such as human settlements (e.g., including built-up structures) within the time constraints and data constraints (e.g., in relation to resolution, volume, quality, etc.) typical of crisis management scenarios and the like. At the heart of the system 100 may be the automated extraction 104 of structures of interest from HR/VHR satellite image data source(s) 108 (e.g., <1 - 10m spatial resolution satellite image data obtained by a number of heterogeneous platforms such as SPOT 2 and 5, CBERS 2B, RapidEye 2 and 4, IKONOS® 2, QuickBird 2, WorldView 1 and 2) and generation of resultant images 116 that include the extracted structures of interest therein. The automated extraction 104 may use data from LR satellite image data source(s) 112 (e.g., >10m spatial resolution, such as MODIS 500m, Landscan, etc.) for use in obtaining samples of the HR/VHR satellite image data 108 that will be used to train a hierarchical data structure for classifying and thus extracting structures of interest from the HR/VHR satellite image data 108.

[0038] Turning now to Figure 2, a more detailed block diagram of an automated extraction system 200 that may be used to implement the automated extraction 104 of structures of interest shown in Figure 1 is presented. Although depicted as a single device (e.g., server, workstation, laptop, desktop, mobile device, and/or other computing device), one or more functionalities, processes or modules of the system 200 may be allocated or divided among a plurality of machines, devices and/or processes which may or may not be embodied in a single housing. In one arrangement, functionalities of the server 200 may be embodied in any appropriate cloud or distributed computing environment.

[0039] Broadly, the system 200 may include memory 204 (e.g., one or more RAM or other volatile memory modules, etc.), a processing engine or unit 208 (e.g., one or more CPUs, processors, processor cores, or other similar pieces of hardware) for executing computer readable instructions from the memory 204, storage 212 (e.g., one or more magnetic disks or other non-volatile memory modules or non-transitory computer-readable mediums), and/or a number of other components 216 (e.g., input devices such as a keyboard and mouse, output devices such as a display and speakers, and the like), all of which may be appropriately interconnected by one or more buses 220. While not shown, the system 200 may include any appropriate number and arrangement of interfaces that facilitate interconnection between the one or more buses 220 and the various components of the system 200 as well as with other devices (e.g., network interfaces to allow for communication between the system 200 and other devices over one or more networks, such as LANs, WANs, the Internet, etc.).

[0040] The system 200 may retrieve any appropriate HR/VHR satellite image data 224 (e.g., from one or more HR/VHR satellite image data sources 108 of Figure 1) as well as any appropriate LR satellite image data 228 (e.g., from one or more LR satellite image data sources 112 of Figure 1) and store the same in any appropriate form in storage 212 (e.g., such as in one or more databases and manageable by any appropriate database management system (DBMS) to allow the definition, creation, querying, update, and administration of the databases). The processing engine 208 may execute a DBMS or the like to retrieve and load the HR/VHR satellite image data 224 and/or LR satellite image data 228 into the memory 204 for manipulation by a number of engines or modules of the system 200 as will be discussed in more detail below.

[0041] As shown, the system 200 may include a "construction" engine 232 that is broadly configured to construct first and second hierarchical data structures from input satellite images, a "training" engine 236 that is broadly configured to train the second hierarchical data structure to detect the components of structures of interest in the first hierarchical data structure, a "classification" engine 240 that is broadly configured to employ the trained second hierarchical data structure to detect structures of interest in the first hierarchical data structure, and a "mapping" engine 244 that is broadly configured to transfer the components classified as being of interest into a resultant image. Each of the engines (and/or other engines, modules, logic, etc. disclosed and/or encompassed herein) may be in the form of one or more sets of computer-readable instructions for execution by the processing unit 208 and that may be manipulated by users in any appropriate manner to perform automated extraction and presentation of structures of interest (e.g., built-up layers or the like) on a display (not shown). In this regard, the combination of the processor 208, memory 204, and/or storage 212 (i.e., machine/hardware components) on the one hand and the various engines/modules disclosed herein in one embodiment create a new machine that becomes a special purpose computer once it is programmed to perform particular functions of the extraction utilities disclosed herein (e.g., pursuant to instructions from program software).

[0042] In one arrangement, any appropriate portal in communication with the various engines may run on the system 200 and be accessible by users (e.g., via any appropriate browser) to access the functionalities of the system 200. While the various engines have been depicted in Figure 2 as being separate or distinct modules, it is to be understood that the functionalities or instructions of two or more of the engines may actually be integrated as part of the same computer-readable instruction set and that the engines have been depicted in the manner shown in Figure 2 merely to highlight various functionalities of the system 200. Furthermore, while the engines have been illustrated as being resident within the (e.g., volatile) memory 204 (e.g., for execution by the processing engine 208), it is to be understood that the engines may be stored in (e.g., non-volatile) storage 212 (and/or other non-volatile storage in communication with the system 200) and loaded into the memory 204 as appropriate.

[0043] To facilitate the reader's understanding of the various engines of the system 200, additional reference is now made to Figure 5 which illustrates a method 500 for use in performing the automated extraction processes disclosed herein as well as to Figures 3, 4a and 4b which respectively illustrate a Max-Tree and a KD-Tree for use in organizing data as part of the method 500 of Figure 5. While specific steps (and orders of steps) of the method 500

have been illustrated and will be discussed, other methods (including more, fewer or different steps than those illustrated) consistent with the teachings presented herein are also envisioned and encompassed within the present disclosure.

[0044] The method 500 may begin by decomposing 504 one or more HR/VHR input satellite images of a particular geographic area into a plurality of connected components (e.g., groups of pixels that collectively define the input images as whole) of a first hierarchical data structure. With reference to Figures 2 and 3, for instance, the construction engine 232 of the automated extraction system 200 may receive one or more input satellite images 304 (e.g., HR/VHR satellite image data 224) of a particular geographic area (e.g., WorldView 2 multispectral images acquired over a portion of Texas near Dallas, TX as shown in Figure 6) and break the input images 304 down into a plurality of components 308. The construction engine 232 then appropriately organizes or arranges the various components 308 of the input images 304 into a first hierarchical data structure 248 such as a Max-Tree 300 made up of a plurality of hierarchically interconnected nodes 312, where each node 312 includes one or more of the components. Part of organizing the components 308 into the Max-Tree 300 also includes appropriately determining or otherwise obtaining feature elements 314 (e.g., k-dimensional descriptors) for each component 308 such as contrast, area, compactness, linearity, average spectral response, eccentricity and/or inertia for the component. The various feature elements 314 of each component 308 may be appropriately organized into a feature element or attribute vector for the component 308.

[0045] In the case of a multi-spectral image, for instance, the various spectral bands may be fused into a single band in any appropriate manner before the image is decomposed into the first hierarchical data structure. For example, the spectral bands of an 8-band multi-spectral image may be fused into a single band by way of the following built-up (BU) index which may be computed for each pixel of the input image:

$$BU = \frac{RE - NIR2}{RE + NIR2}$$

where "RE" is the intensity of electromagnetic radiation received in the red edge band of the electromagnetic spectrum, and "NIR2" is the intensity of electromagnetic radiation received in the NIR2 band of the electromagnetic spectrum. In this regard, the pixels of the 8-band multispectral image may be broken and arranged into a plurality of hierarchical components based on the respective BU values of the pixels.

[0046] As another example, the spectral bands of a 4-band multi-spectral image may be fused into a single band by way of the following built-up (BU) index which may be computed for each pixel of the input image:

$$BU = \frac{R - NIR}{R + NIR}$$

where "R" is the intensity of electromagnetic radiation received in the red band of the electromagnetic spectrum, and "NIR" is the intensity of electromagnetic radiation received in the NIR band of the electromagnetic

spectrum. In this regard, the pixels of the 4-band multispectral image may be broken and arranged into a plurality of hierarchical components based on the respective BU values of the pixels.

[0047] In one arrangement, the Max-Tree 300 may be a rooted, uni-directed tree with its leaves (e.g., leaves 316, 320, 324) corresponding to a regional maxima of the input images 304 and its root (e.g., node 328) corresponding to a single connected component defining the background of the input images 304. For instance, the hierarchical ordering of the nodes 312 may encode the nesting of peak components (e.g., pixels with intensities greater than a level "h") with respect to the gray-scale range of the input images 304. The image may be thresholded at each grey level to provide as many binary images as the number of grey levels. Each binary image may then be analyzed to derive its connected components. Each node 312 may generally point to its parent (i.e., the first ancestor node 312 below the given level) while the root node 328 points to itself. In one embodiment, each node 312 may include a pointer to a data structure that stores auxiliary data for each node 312. The construction engine 232 may compute or otherwise determine the feature elements 314 from such auxiliary data. The Max-Tree 300 may thus allow for compact storage of the connected components from all grey levels while having a limited computational complexity. In any case, the first hierarchical data structure(s) 248 (e.g., Max-Tree 300) may be appropriately stored in memory 204 for quick retrieval during subsequent steps of the method 500.

[0048] As shown in Figure 5, the method 500 may then include constructing 508 a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, where each of the feature elements 314 depends from one of the nodes. With reference to Figures 2 and 3, the construction engine 232 may obtain the various feature elements 314 of the nodes 312 (e.g., of the components 308), where each feature element lies in a two-dimensional (e.g., x, y) space, and construct a second hierarchical data structure 252 such as a KD-Tree. Constructed second hierarchical data structure(s) 252 may be appropriately stored in memory 204 for quick retrieval during subsequent steps of the method 500.

[0049] Figure 4a illustrates a simplified KD-Tree-based space partitioning procedure 400 that may be used to build a (hierarchically-arranged) KD-Tree. In one arrangement, the feature elements 314 may be initially disposed at appropriate locations with respect to an x, y coordinate system as shown in a first step 404 of the procedure 400. As just one example, the x-axis may correspond to the area of each component (e.g., a ground area collectively represented by the image pixels making of the component) and the y-axis may correspond to the red color response of each component. The construction engine 232 may then create a root node f_1 as shown in a second step 408 by splitting the feature elements 314 into two groups with a vertical line through the median x-coordinate of the feature elements 314. A similar procedure may then be performed to create child nodes f_2, f_3, as shown in a third step 412, only with respective horizontal lines through the respective median y-coordinates of the feature elements 314 on either side of the root node f_1.

[0050] The splitting may then continue recursively to create leaf nodes f_4, f_5 f_6, f_7 as shown in a fourth step 416, where each leaf node f_4, f_5 f_6, f_7 contains a single feature element 314 or up to a maximum of "*m*" feature elements

314, where *m* may be appropriately designated in advance. Figure 4b illustrates a simplified KD-Tree built from the nodes f_1 - f_7 and feature elements 314 illustrated in the fourth step 416 of Figure 4a. As shown, each of the feature elements 314 depends from at least one of the nodes 314. While a two-dimension space partitioning structure is illustrated in each of the steps of Figure 4a, it is to be understood that more complicated space partitioning structures may be constructed to accommodate more than two dimensions of feature elements. For instance, all of the feature elements 314 may be fused together in the partitioning structure and then splitting may occur against the most varying dimension. The feature elements 314 may then be clustered with respect to their full description which is the concatenation of the shape descriptors plus the spectral averages.

[0051] Returning to Figure 5, the method 500 may include categorizing 512 at least some components of the first hierarchical data structure as either structures of interest (e.g., built-up) or non-structures of interest (e.g., non-built-up) using LR satellite image data. With reference to Figure 2, the training engine 236 of the automated extraction system 200 may receive one or more reference satellite image data sets (e.g., LR satellite image data 228) over a geographic area that overlaps the geographic area over which the one or more input satellite images 304 were obtained (e.g., Lebanon). For instance, Figure 6 illustrates a graphical representation of an NLCD 2006 information layer over a portion of Texas near Dallas, TX (e.g., where more highly populated areas (e.g., around Mesquite, Ennis and Corsicana, TX) are represented by different colors than less populated areas (e.g., along highway 287 between Ennis and Corsicana, TX).

[0052] In one arrangement, the training engine 236 may derive training components from the plurality of components 308 of the first hierarchical data structure 248/300 to be subsequently used to train the second hierarchical data structure 252/450 as will be described below. As just one example, the training engine 236 may consider a particular component 308 (e.g., group of pixels of the one or more input satellite images 304) of the first hierarchical data structure 248/300 as being a "positive" example in the event that the particular component 308 falls mostly or completely within an urban area as indicated in a corresponding portion of the one or more reference satellite image data sets (e.g., the NLCD 2006 information layer of Figure 6). In contrast, the training engine 236 may consider a particular component 308 of the first hierarchical data structure 248/300 as being a "negative" example in the event that a corresponding portion of the one or more reference satellite image data sets is mostly or completely devoid of urban areas. In one embodiment, components 308 of the first hierarchical data structure 248/300 may be appropriately weighted to indicate the relative degree to which corresponding portions of the one or more reference satellite image data sets do or do not indicate urban or built-up areas.

[0053] In any event, and once a group of training components has been obtained (e.g., a subset of the plurality of components 308 of the first hierarchical data structure 248/300, such as one or more small, limited, and/or random portions of Figure 6 so as to limit computational costs), the method 500 includes training 516 the second hierarchical data structure 252/450 with the particular feature elements 314 of the trained or categorized components obtained in step 512 to obtained a trained second hierarchical data structure 256 (see Figure 2). With reference to Figure 3, assume for purposes of discussion that component(s) 308 of leaf node 320 was/were identified as positive (e.g., built-

up) examples in the categorizing step 512. In this regard, the training engine 236 may obtain the particular feature element(s) 314 (e.g., particular contrast level, compactness, average spectral response, etc.) associated with the component(s) 308 of the leaf node 320 (e.g., as well as feature element(s) 314 of parent, grandparent, root nodes, etc. along a common leaf-path), identify any same or similar feature elements 314 in the second hierarchical data structure 252/450, and label (e.g., tag, flag, mark, note, etc.) the nodes along the respective leaf-paths of the identified feature elements 314 in the second hierarchical data structure 252/450 as being positive (e.g., built-up) feature elements and/or negative (e.g., non-built-up) feature elements in any appropriate manner.

[0054] For instance, each of the nodes f_1 - f_7 of the second hierarchical data structure 252/450 may include respective positive and/or negative class counters 478 configured to convey whether or not (or a degree to which) the nodes f_1 - f_7 are within the positive or negative class. In this regard, the training engine 236 may update (e.g., increment, add to, etc.) the positive class counters 478 of all nodes f_1 - f_7 along the leaf-path (e.g., branch) within which is located one or more feature elements 314 that are the same as or similar to those of the training components 308. In contrast, the positive class counters 478 of those of the nodes f_1 - f_7 not resident within leaf-paths of one or more feature elements 314 that are the same as or similar to those of the training components 308 may be left untouched, the positive class counters 478 may be appropriately updated (e.g., reduced), the negative class counters 478 may be updated (e.g., incremented, added to), and/or the like. A similar procedure may be performed for each feature element 314 of each of the training components 308.

[0055] In one arrangement, two or more of the feature elements 314 of the second hierarchical data structure 252/450 that are close enough with respect to some dissimilarity (e.g., that are in the same neighborhood or cluster) may be assigned the same class label (e.g., and the class counters 478 of the nodes of respective leaf-paths appropriately updated). Stated differently, the class label of a single feature element 314 can be propagated to all other elements in the cluster (e.g., and their respective nodes) at little to no further computational cost thus reducing the overall computational overhead.

[0056] With reference to Figure 4b, for instance, assume that feature element(s) 454, 458 are disposed within a first cluster 462 while feature element(s) 466, 470 are disposed with a second cluster 474. Furthermore, assume the training engine 236 initially labeled feature element(s) 454 as being in the "positive" class due to a similarity to feature element(s) 314 of the training components 308 but did not initially label any of feature elements 458, 466, 470 as being in the positive class. In this regard, the training engine 236 may also label the feature element(s) 458 as being in the positive class due to the feature elements 454, 458 being within the first cluster 462. In one variation, the more feature elements 314 of a particular cluster that are initially labeled as being within the positive class, the greater degree to which feature elements 314 in the cluster that are not initially labeled as being within the positive class are subsequently labeled as being within the positive class. However, the training engine 236 may not label the feature element(s) 466, 470 as being in the positive class as the feature element(s) 466, 470 are not within the first cluster 462. Of course, the feature elements 314 of other training components 308 may still cause the feature element(s) 466, 470 to be labeled in the positive class.

[0057] Again with reference back to Figure 5, the method 500 may include classifying 520 components of the first hierarchical data structure 248/300 as structures of interest (e.g., built-up) or non-structures of interest (e.g., non-built-up) with the trained second hierarchical data structure 256. For instance, the classification engine 240 of the automated extraction system 200 (see Figure 2) may identify those of the nodes f_1 - f_7 of the trained second hierarchical data structure 256 whose class counters 478 indicate the node as being within the positive class (e.g., as being associated with structures of interest, such as built-up). In one arrangement, any appropriate thresholds or the like may be employed to allow the classification engine 240 to quickly identify those of the nodes f_1 - f_7 in the positive class (e.g., class counters above or below one or more particular thresholds). In any case, the classification engine 240 may then identify the feature elements 314 depending from the leaves of each of the positively identified nodes f_1 - f_7 and use the identified feature elements 314 to classify components 304 of the first hierarchical data structure 300 as being structures of interest (e.g., positive class, built-up) or non-structures of interest (e.g., negative class, non-built-up).

[0058] With reference to Figure 3, for instance, assume that feature element(s) 332 of node 324 of the first hierarchical data structure 300 is or are the same or similar to a feature element 314 from the second hierarchical data structure 450 that depends from a leaf node of a positively identified node. In this regard, the classification engine 240 may classify component 336 of node 324 as being or identifying a structure of interest (e.g., an urban area). A similar process may be performed for other feature elements 314 associated with positively identified nodes from the second hierarchical data structure 450. Furthermore, other nodes 312 disposed along the same path or branch as the initially classified node 324 may also be classified as being structures of interest. While not discussed in more detail, a similar process may be performed to classify certain components 308 as being non-structures of interest.

[0059] It is to be understood that this discussion has been greatly simplified and has merely been provided to assist the reader in understanding the functionalities disclosed herein. Furthermore, numerous variations and more complicated arrangements are envisioned. For instance, one arrangement envisions that more than a particular threshold number of feature elements 314 of a particular node 312 of the first hierarchical data structure 300 must be labeled as structures of interest (e.g., two or more, three or more, etc.) before the corresponding component(s) 336 of the node 312 can be classified as being a structure of interest. In another arrangement, certain of the nodes 312 and/or components 308 may be appropriately clustered, grouped or the like (e.g., as discussed in relation to the second hierarchical data structure 252/450). In this regard, classification of one component 308 in a certain regard may result in similar classification of other clustered components 308.

[0060] In one variation, the categorizing 512 may be recursively performed with respect to additional portions of the plurality of components 308 of the first hierarchical data structure 248/300 to further refine the accuracy of the training 516 and classifying 520. For instance, a second subset of the components 308 of the first hierarchical data structure 248/300 (e.g., corresponding to one or more different, partially or fully non-overlapping portions of Texas in Figure 6) may be categorized as structures of interest or non-structures of interest using corresponding portions of the

reference satellite image data set (e.g., the NLCD 2006 of Figure 6) to derive additional "training components" from the first hierarchical data structure 248/300. Thereafter, the second hierarchical data structure 252/450 may be trained 516 as discussed previously with the feature elements 314 of the second subset (e.g., with the additional training components) and then (all) of the components 308 of the first hierarchical data structure 248/300 may be classified/reclassified as being structures of interest or non-structures of interest.

[0061] As shown in Figure 5, the method 500 may then include extracting 524 (e.g., isolating, obtaining, etc.) components (e.g., see extracted components 260 in Figure 2) that are classified as identifying structures of interest from the classifying step 520 and querying 528, for each respective extracted component 308, whether an error rate (e.g., EERs) between the extracted component and a corresponding portion of a reference satellite image data set (e.g., LR satellite image data associated with the subject geographic area and having a spatial resolution lower than that of the input satellite images 304) is less than a particular threshold error rate. The reference satellite image data set used to determine the error rates may be the same as or different than the reference satellite image data set used in the categorizing step 512. Those extracted components 308 associated with error rates below the threshold error rate may be mapped 532 (e.g., via mapping engine 244) into a resultant image (e.g., resultant image 264) that includes the structures of interest (e.g., at a spatial resolution greater than that of the reference satellite image data set(s), such as at least substantially the same as that of the input image(s) 304). For instance, see Figures 7a and 8a which present a result of the mapping step 532 over a portion of Texas at respective first and second levels of zoom. Resultant images 264 may be appropriately stored in both (volatile) memory 204 and (non-volatile) storage 212.

[0062] As an example, cloud cover or the like when the input satellite image(s) 304 was/were obtained may result in error rates above the threshold for extracted components classified as identifying built-up structures. In this regard, such extracted components may be not mapped into the resultant image or at least mapped into the resultant image to a lesser degree (e.g., assigned a lower weight based how far over the threshold were the corresponding error rates). However, the decomposing 504, constructing 508, categorizing 512, training 516, classifying 520, extracting 524, querying 528 and mapping 532 may be performed with respect to subsequent input satellite images (e.g., second, third, etc.) over the substantially same geographic area that are obtained via heterogeneous sources, different satellites, different locations, different times such as multi-temporal images, different frequencies or wavelengths such as multispectral images, and/or the like. Incorporation of additional input satellite images into the method 500 may increase the likelihood that those portions of the resultant image that would otherwise be depicting built-up (but for the corresponding previously extracted components not being mapped into the resultant image due to error rates over the threshold) do eventually depict corresponding built-up portions (e.g., due to the subsequent/additional input satellite images being obtained from a high azimuth, during a partially or fully cloudless, sunny day, etc.) to allow for an iterative refinement of the resultant image (e.g., the images presented in Figures 7a and 8a) over a plurality of cycles. While the method 500 has been discussed as including the querying step 528, some arrangements envision proceeding from the extracting step 524 directly to the mapping step 532 without performing the querying step 528 (e.g., to reduce computational costs, such as time, resources, etc.).

Non-Limiting Example:

[0063] Four WorldView-2 8-band multispectral images acquired over a portion of Texas near Dallas, TX as shown in Figure 6 are considered. Max and KD-Trees (e.g., first and second hierarchical data structures) are obtained as discussed previously with respect to steps 504 and 508 of Figure 5. Samples from the globally-consistent NLCD 2006 information layer shown in Figure 6 are systematically collected (where the data was collected from orbiting Landsat satellites at a spatial resolution of 30 meters). In the case of the multispectral scenes of Figure 6 containing 10070 x 58734 pixels, components can arranged into a Max-Tree (e.g., as in Figure 3) and then a corresponding KD-Tree (e.g., as in Figure 4b) can be generated in about 387s. Subsequently, training components are derived considering the NLCD 2006 information layer (e.g., step 512 of Figure 5) in about 23s before being used for training the KD-Tree in under 1s (step 516 of Figure 5). The classification (e.g., step 520 of Figure 5) is then obtained in well under 1s (e.g., much closer to 0s than to 1s) before components identifying built-up are extracted and reprojected in (e.g., mapped into at step 532 of Figure 5) the image space. The result of this process on the WorldView-2 multispectral scenes is displayed in Figures 7a and 8a. Figures 7b and 8b respectively illustrate close-up portions of the NLCD 2006 and WV2 multispectral images corresponding to the resultant images of Figures 7a and 8a.

[0064] At low spatial resolutions (e.g., satellite image data obtained from Landsat 30m sensor, MODIS 500m sensor, etc.), spectral measurements can calibrated such that they are not affected by the sun, satellite angles, atmospheric conditions, and the like. In this regard, LR satellite image data can be used to accurately classify structures of interest in HR/VHR satellite image data as disclosed herein. In one arrangement, the utilities disclosed herein may be used to generate a globally-consistent HR/VHR satellite image layer that includes structures of interest (e.g., built-up) and that is devoid of non-structures of interest. The globally-consistent layer can be used in numerous contexts such as in assessing population densities, qualities of life, vulnerability factors, disaster risks, sufficiency of civil infrastructures, economic growth, poverty levels, event monitoring and evolution, and the like.

[0065] It will be readily appreciated that many deviations and/or additions may be made from or to the specific embodiments disclosed in the specification without departing from the spirit and scope of the invention. In one arrangement, it is envisioned that different structures of interest (e.g., different types of built-up) may be differently shaded, colored, and/or the like (e.g., such as based on one or more geometric and/or radiometric attributes of pixels or components of the resulting image) in the resulting image to allow analysts to be able to quickly analyze the resulting image. As an example, the mapping engine 264 may analyze the feature elements 314 of the extracted input image components to determine a type of built-up structure represented by the particular component(s). For instance, buildings may exhibit one general type of feature element "signature" while tents may exhibit another type of feature element signature. The mapping engine 264 may have access to a database that correlates feature element signatures with respective colors, shades, and/or the like. During the mapping of the extracted components into the resulting image, the mapping engine 264 may access the database and implement the colors, shades, etc. as appropriate.

[0066] In another arrangement, it is envisioned that manual (e.g., analyst) input may replace or at least supplement categorizing step 512 of Figure 5. For instance, it is envisioned that a user may be able to manually select (e.g., on a display with any appropriate user manipulable device) one or more positive training regions (e.g., representing structures of interest) of the input satellite image and/or one or more negative training regions (e.g., not representing structures of interest) of the input satellite image. The feature elements of the components of the first hierarchical data structure representing the selected regions may then be used to train 516 the second hierarchical data structure before components are appropriately classified in step 520. For example, the user may continue to select different training regions/examples after each classification step 520. In one variation, it is envisioned that a user may manually selected at least some positive and negative training regions/components and then a reference data set (e.g., NLCD 2006) may be used to identify other training regions/components.

[0067] In a further arrangement, a plurality (e.g., a "forest") of second hierarchical data structures (e.g., KD-Trees) may be constructed and trained for use in classifying and thus extracting different types of structures of interest from input satellite images. As one example, a first KD-Tree may be generated with a first set of feature elements of the first hierarchical data structure and a second KD-Tree may be generated with a second set of feature elements of the first hierarchical data structure that is at least partially non-overlapping with the first set of feature elements (e.g., the first set includes contrast, compactness and linearity values of each of the components of the first hierarchical data structure and the second set includes compactness, linearity and average spectral response of each of the components of the first hierarchical data structure). Thereafter, training components may be selected in any appropriate manner from the first hierarchical data structure (e.g., with one or more low resolution information layers as discussed herein, via manual input, and/or the like) and their respective feature elements used to train each of the first and second KD-Trees to be able to detect particular types of structures (e.g., particular types of built-up) in at least one input satellite image. The first and second (e.g., and/or additional) KD-Trees may then be used to classify particular objects/areas/regions of the at least one input satellite image which may then be extracted and incorporated into a resultant image.

[0068] As another example, it is envisioned that a single second hierarchical data structure (e.g., a single KD-Tree) that includes all of the feature elements from the first hierarchical data structure may be initially constructed as disclosed herein (e.g., via fusing all of the feature elements together in the partitioning structure and then performing the splitting against the most varying dimension). Thereafter, however, a plurality of copies of the KD-Tree may be made to create a "forest" of KD-Trees, where each copy may be trained to detect different objects/areas/etc. in the at least one input satellite image. For instance, one or more training components representing positive and/or negative examples of a first type of object/structure/area of the first hierarchical data structure may be selected from the first hierarchical data structure in any appropriate manner (e.g., with one or more low resolution information layers as discussed herein, via manual input, and/or the like) and their feature elements used to train a first of the KD-Tree copies to detect the first type of object/structure/area. Additionally, one or more different training components representing positive and/or negative examples of a additional types of objects/structures/areas of the first

hierarchical data structure may be selected from the first hierarchical data structure in any appropriate manner (e.g., with one or more low resolution information layers as discussed herein, via manual input, and/or the like) and their feature elements used to train additional ones of the KD-Tree copies to detect the additional types of objects/structures/areas. In one variation, different low resolution information layers may be used to identify different types of training components from the first hierarchical data structure. The various differently-trained KD-Trees may then be used to classify particular objects/areas/regions of the at least one input satellite image which may then be extracted and incorporated into one or more resultant images.

[0069] Furthermore, while the first hierarchical data structure has been discussed primarily in form of a Max-Tree, is it envisioned that other forms of the first hierarchical data structure may be used with the utilities disclosed herein. In one arrangement, the first hierarchical data structure may be in the form of an "Alpha-Tree" whereby pixels of the input satellite image (e.g., intensity values, radiance values, BU values, etc.) may be hierarchically grouped into components (e.g. nodes) based on any appropriate measure of dissimilarity between adjacent nodes. Thereafter, the second hierarchical data structure may be constructed from feature elements (e.g., attribute vectors) of the components/nodes of the Alpha-Tree, training components/nodes of the Alpha-Tree may be selected, the second hierarchical data structure may be trained, the components/nodes of the Alpha-Tree may be classified, and structures of interest may be extracted from the input image and displayed in a resultant image as disclosed herein.

[0070] Embodiments disclosed herein can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer-readable medium for execution by, or to control the operation of, data processing apparatus. For example, the logic or software of the construction engine 232, training engine 236, classification engine 240 and mapping engine 244 responsible for the various functionalities disclosed herein may be provided in such computer-readable medium of the automated extraction system 200 and executed by the processor 208 as appropriate. The computer-readable medium can be a machine-readable storage device, a machine-readable storage substrate, a non-volatile memory device, a composition of matter affecting a machine-readable propagated signal, or a combination of one or more of them. In this regard, the system 200 may encompass one or more apparatuses, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. In addition to hardware, the system 200 may include code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[0071] A computer program (also known as a program, software, software application, script, or code) used to provide any of the functionalities described herein (e.g., construction of the first and second hierarchical data structures and the like) can be written in any appropriate form of programming language including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other

programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0072] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). Processors suitable for the execution of a computer program may include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. Generally, the elements of a computer are one or more processors for performing instructions and one or more memory devices for storing instructions and data. The techniques described herein may be implemented by a computer system configured to provide the functionality described.

[0073] While this specification contains many specifics, these should not be construed as limitations on the scope of the disclosure or of what may be claimed, but rather as descriptions of features specific to particular embodiments of the disclosure. Furthermore, certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0074] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and/or parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software and/or hardware product or packaged into multiple software and/or hardware products.

[0075] The above described embodiments including the preferred embodiment and the best mode of the invention known to the inventor at the time of filing are given by illustrative examples only.

WHAT IS CLAIMED IS:

1. A method of extracting built-up structures from satellite imagery data, comprising:

decomposing, using a processor, an input satellite image into a plurality of components of a first hierarchical data structure, wherein the input image is associated with a geographic area, and wherein each of the components is characterized by at least one feature element;

constructing, using the processor, a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, wherein each of the feature elements depends from at least one of the plurality of nodes;

deriving training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the input image using a first reference image data set that is associated with the geographic area;

training, with the processor, the second hierarchical data structure with the feature elements of the training components for detecting built-up structures;

classifying, with the processor and using the trained second hierarchical data structure, the plurality of components of the first hierarchical data structure as identifying built-up structures or non-built-up structures; and

mapping components of the plurality of components that identify built-up structures as classified during the classifying step into a resultant image that is associated with the geographic area.

2. The method of claim 1, further comprising:

deriving additional training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the input image using the first reference image data set;

training, with the processor, the second hierarchical data structure with the feature elements of the additional training components for detecting built-up structures; and

classifying, with the processor and using the second hierarchical data structure as trained with the additional training components, the plurality of components of the first hierarchical data structure as identifying built-up structures or not identifying built-up structures.

3. The method of any of claims 1-2, further comprising:

ascertaining error rates between components that identify built-up structures from the classifying step and corresponding portions of a second reference image data set that identify built-up structures; and

mapping components that identify built-up structures that are associated with error rates below a threshold error rate into the resultant image that is associated with the geographic area.

4. The method of claim 3, wherein the first and second reference data sets are different.

5. The method of claim 3, wherein the first and second reference data sets are the same.

6. The method of any of claims 3-5, wherein the input image comprises a first input image, wherein the method further comprises:

decomposing, using the processor, a second input image into a plurality of components of a first hierarchical data structure, wherein the second input image is associated with the geographic area, wherein each of the components is characterized by at least one feature element, and wherein the first and second input images are at least one of multi-temporal or captured from different locations;

constructing, using the processor, a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, wherein each of the feature elements of the second input image depends from at least one of the plurality of nodes; and

deriving training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the second input image using the first reference image data set;

training, with the processor, the second hierarchical data structure with the feature elements of the training components of the second input image for detecting built-up structures;

classifying, with the processor and using the second hierarchical data structure as trained with the training components of the second input image, the plurality of components of the second input image of the first hierarchical data structure as identifying built-up structures or non-built-up structures;

ascertaining error rates between components of the second input image that identify built-up structures from the classifying step and corresponding portions of the second reference data set that identify built-up structures; and

mapping components that identify built-up structures in the second input image that are associated with error rates below a threshold error rate into the resultant image, wherein the resultant image comprises components from the first and second input images.

7. The method of claim 6, further comprising:

continuing to perform the decomposing, constructing, deriving, training, classifying, ascertaining, and mapping steps with respect to additional input images that are at least one of multi-temporal relative to the first and second input images or captured from locations different than those from which the first and second input images were captured.

8. The method of any of claims 1-7, wherein one of the training components identifies built-up structures when the training component falls within a built-up area as indicated by the first reference data set.

9. The method of any of claims 1-8, wherein the at least one feature element comprises contrast, area, compactness, linearity, average spectral response, eccentricity or inertia of a respective one of the plurality of components.

10. A method of extracting built-up structures from satellite imagery data, comprising:

decomposing, using a processor, an input satellite image into a plurality of components of a first hierarchical data structure, wherein the input image is associated with a geographic area and has a first resolution, and wherein each of the components is characterized by at least one feature element;

constructing, using the processor, a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, wherein each of the feature elements depends from at least one of the plurality of nodes; and

deriving training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the input image using a first reference image data set that is associated with the geographic area and has a second resolution lower than the first resolution.

11. The method of claim 10, wherein one of the training components identifies built-up structures when the training component falls within a built-up area as indicated by the first reference data set.

12. The method of any of claims 10-11, further comprising:

training, with the processor, the second hierarchical data structure with the feature elements of the training components for detecting built-up structures; and

classifying, with the processor and using the trained second hierarchical data structure, the plurality of components of the first hierarchical data structure as identifying built-up structures or non-built-up structures.

13. The method of claim 12, wherein the training step comprises:

updating class counters of nodes of the second hierarchical data structure in response to feature elements of the training components corresponding to built-up or non built-up structures.

14. The method of claim 13, wherein each node of the second hierarchical data structure comprises a built-up structure counter and a non built-up structure counter, and wherein the updating steps comprises:

increasing the count of one of the built-up structure counter and the non built-up structure counter.

15. The method of any of claims 12-14, further comprising:

mapping components of the plurality of components that identify built-up structures as classified during the classifying step into a resultant image that is associated with the geographic area and that has a third resolution that is greater than the second resolution.

16. The method of claim 15, further comprising:

deriving additional training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the input image using the first reference image data set;

training, with the processor, the second hierarchical data structure with the feature elements of the additional training components for detecting built-up structures; and

classifying, with the processor and using the second hierarchical data structure as trained with the additional training components, the plurality of components of the first hierarchical data structure as identifying built-up structures or not identifying built-up structures.

17. The method of claim 16, wherein the training components are non-overlapping with the additional training components.

18. The method of any of claims 16-17, further comprising before the mapping:

continuing to perform the deriving, training and classifying steps with respect to additional portions of the plurality of components.

19. The method of any of claims 16-18, further comprising:

not mapping components of the plurality of components that identify non built-up structures as classified during the classifying step into the resultant image.

20. The method of any of claims 12-19, further comprising:

ascertaining error rates between components that identify built-up structures from the classifying step and corresponding portions of a second reference image data set that identify built-up structures; and

mapping components that identify built-up structures that are associated with error rates below a threshold error rate into a resultant image that is associated with the geographic areas and that has a third resolution that is greater than the second resolution.

21. The method of claim 20, wherein the error rates comprise equal error rates.

22. The method of any of claims 20-21, wherein the second reference image set is associated with the geographic area and has a fourth resolution lower than the first resolution.

23. The method of any of claims 20-22, wherein the first and second reference data sets are different.

24. The method of any of claims 20-22, wherein the first and second reference data sets are the same.

25. The method of any of claims 20-24, wherein the input image comprises a first input image, wherein the method further comprises:

decomposing, using the processor, a second input image into a plurality of components of a first hierarchical data structure, wherein the second input image is associated with the geographic area and has a first resolution, wherein each of the components is characterized by at least one feature element, and wherein the first and second input images are at least one of multi-temporal or captured from different locations;

constructing, using the processor, a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, wherein each of the feature elements of the second input image depends from at least one of the plurality of nodes; and

deriving training components from the plurality of components of the first hierarchical data structure that indicate built-up and non-built-up structures in the second input image using the first reference image data set;

training, with the processor, the second hierarchical data structure with the feature elements of the training components of the second input image for detecting built-up structures;

classifying, with the processor and using the second hierarchical data structure as trained with the training components of the second input image, the plurality of components of the second input image of the first hierarchical data structure as identifying built-up structures or non-built-up structures;

ascertaining error rates between components of the second input image that identify built-up structures from the classifying step and corresponding portions of the second reference data set that identify built-up structures; and

mapping components that identify built-up structures in the second input image that are associated with error rates below a threshold error rate into the resultant image, wherein the resultant image comprises components from the first and second input images.

26. The method of claim 25, further comprising:

continuing to perform the decomposing, constructing, deriving, training, classifying, ascertaining, and mapping steps with respect to additional input images that are at least one of multi-temporal relative to the first and second input images or captured from locations different than those from which the first and second input images were captured.

27. The method of any of claims 10-26, further comprising:

storing the first and second hierarchical data structures in volatile memory, wherein the training step, classify step and mapping step utilize the first and second hierarchical data structures as stored in the memory.

28. The method of any of claims 10-27, wherein a total quantity of the training components is not greater than about 2000.

29. The method of claim 28, wherein a total quantity of the training components is not greater than about 1000.

30. The method of any of claims 10-29, wherein the at least one feature element comprises contrast, area, compactness, linearity, average spectral response, eccentricity or inertia of a respective one of the plurality of components.

31. The method of any of claims 10-30, wherein the at least one feature element is k-dimensional.

32. The method of any of claims 10-31, wherein the first hierarchical data structure comprises a Max-Tree.

33. The method of any of claims 10-32, wherein the second hierarchical data structure comprises a KD-Tree.

34. A system for extracting structures of interest from optical images, comprising:

a construction engine, executable by a processor, that decomposes an input image associated with a geographic area and having a first resolution into a plurality of components, generates a first hierarchical data structure from the plurality of components that includes a plurality of k-dimensional feature elements of each of the components, and creates a second hierarchical data structure that includes a plurality of hierarchically-arranged nodes, wherein each of the feature elements depends from at least one of the plurality of nodes; and

a training engine that uses feature elements of a first portion of the plurality of components of the first hierarchical data structure to train the second hierarchical data structure to detect components of the plurality of components in the first hierarchical data structure that correspond to structures of interest in the input image.

35. The system of claim 34, further comprising:

a classification engine, executable by the processor, that uses the trained second hierarchical data structure to extract a second portion of the plurality of components of the first hierarchical data structure that corresponds to the structures of interest in the input image.

36. The system of claim 35, wherein the second portion encompasses the first portion.

37. The system of any of claims 35-36, further comprising:

a mapping engine that maps the second portion into a resultant image associated with the geographic area.

38. The system of any of claims 35-37, further comprising:

a volatile memory that stores the first and second hierarchical data structures, wherein the construction engine, the training engine and the classification engine manipulate the first and second hierarchical data structures as stored in the volatile memory.

39. The system of any of claims 37-38, further comprising:

a non-volatile memory that stores the resultant image.

40. The system of any of claims 34-39, wherein the training engine utilizes a portion of a reference information layer associated with the geographic area to identify the first portion.

41. The system of claim 40, wherein the input image has a first resolution and the reference information layer has a second resolution lower than the first resolution.

42. The system of claim 34, wherein the training engine utilizes a portion of a reference information layer associated with the geographic area to identify the first portion, wherein the input image has a first resolution and the reference image has a second resolution lower than the first resolution, and wherein the system further comprises:

a mapping engine that maps the second portion into a resultant image that is associated with the geographic area and that has a third resolution that is greater than the second resolution.

43. The system of claim 42, further comprising:

a volatile memory that stores the first and second hierarchical data structures, wherein the construction engine, the training engine and the classification engine manipulate the first and second hierarchical data structures as stored in the volatile memory.

44. The system of claim 43, further comprising:

a non-volatile memory that stores the resultant image.

45. The system of any of claims 34-44, wherein a total quantity of the first portion of the plurality of components is not greater than about 2000.

46. The system of claim 45, wherein a total quantity of the first portion of the plurality of components is not greater than about 1000.

47. The system of any of claims 34-46, wherein each of the plurality of feature elements comprises contrast, area, compactness, linearity, average spectral response, eccentricity or inertia of a respective one of the plurality of components.

48. The system of any of claims 34-47, wherein each of the plurality of feature elements is k-dimensional.

49. The system of any of claims 34-48, wherein the first hierarchical data structure comprises a Max-Tree.

50. The system of any of claims 34-49, wherein the second hierarchical data structure comprises a KD-Tree.

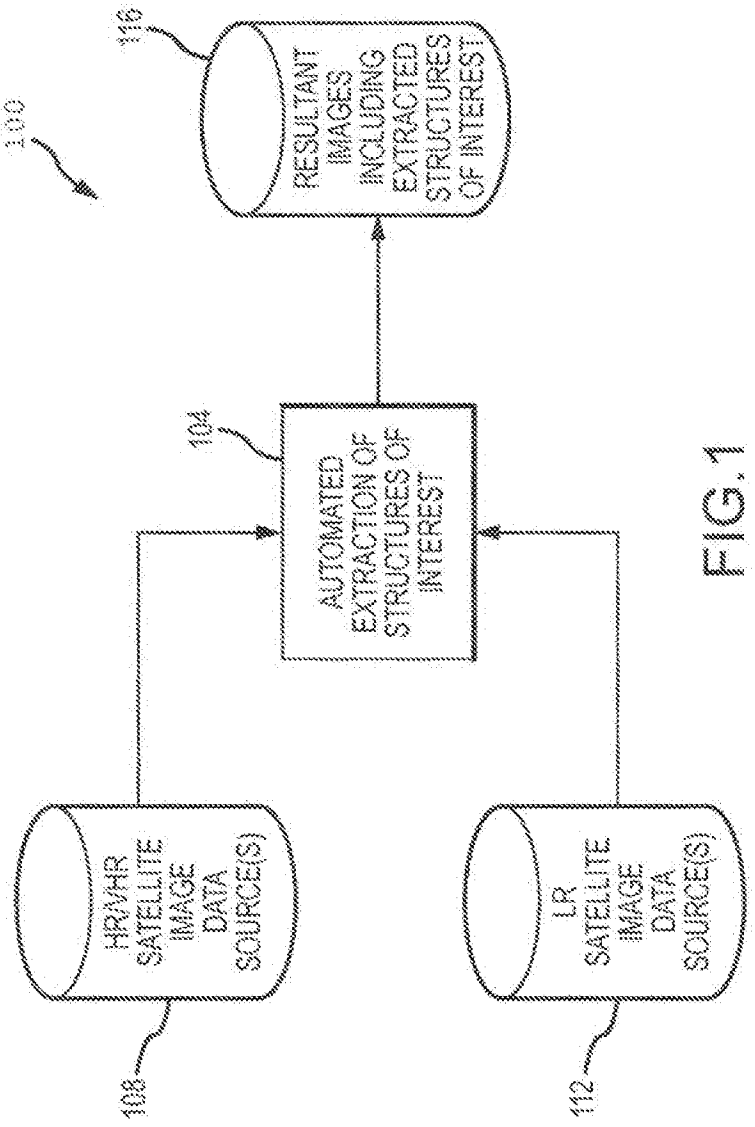

100
116
RESULTANT IMAGES INCLUDING EXTRACTED STRUCTURES OF INTEREST
104
AUTOMATED EXTRACTION OF STRUCTURES OF INTEREST
HR/VHR SATELLITE IMAGE DATA SOURCE(S)
108
LR SATELLITE IMAGE DATA SOURCE(S)
112

FIG. 1

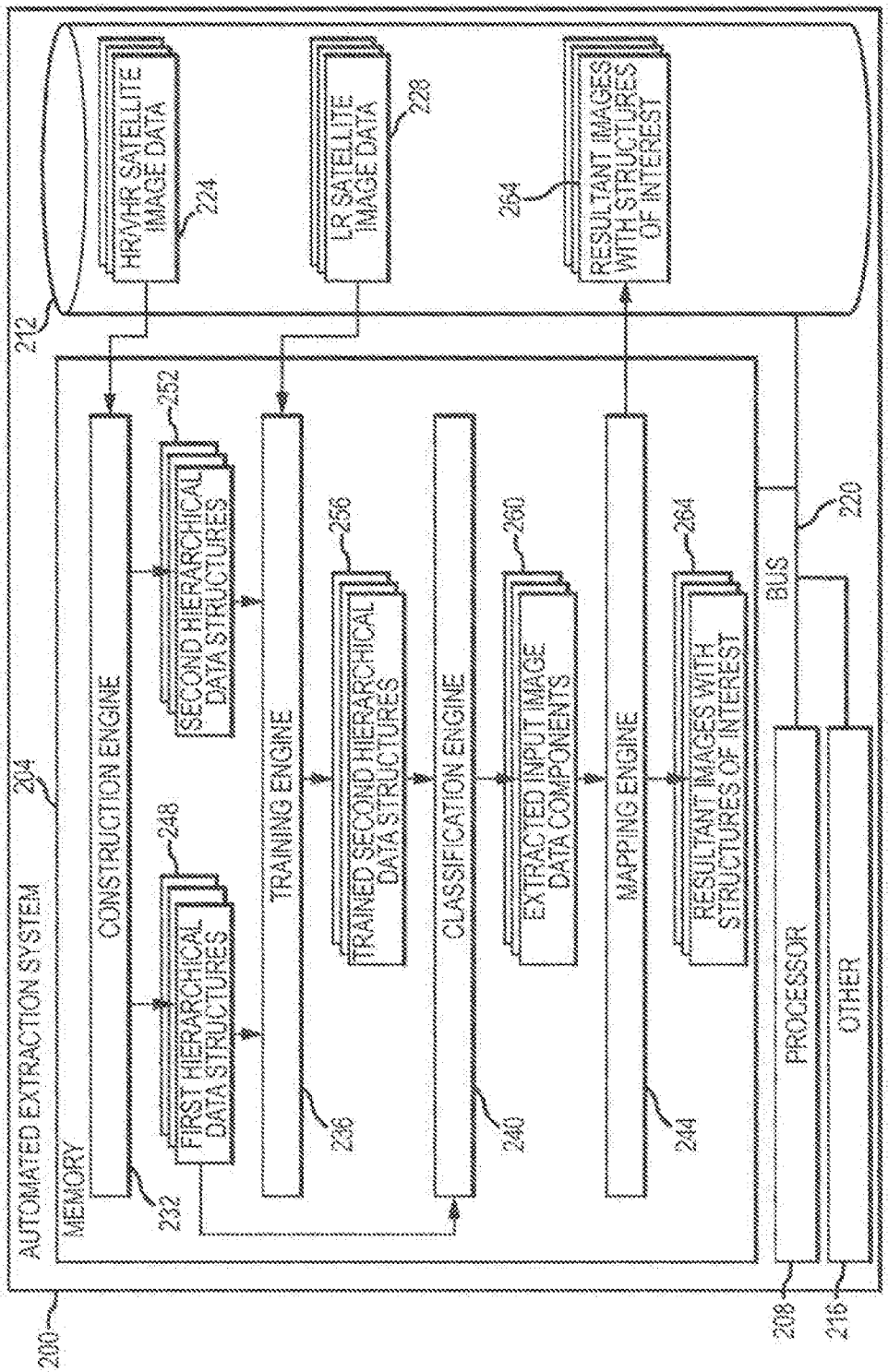

AUTOMATED EXTRACTION SYSTEM
200
204
MEMORY
232
CONSTRUCTION ENGINE
FIRST HIERARCHICAL DATA STRUCTURES
248
SECOND HIERARCHICAL DATA STRUCTURES
252
TRAINING ENGINE
236
TRAINED SECOND HIERARCHICAL DATA STRUCTURES
256
CLASSIFICATION ENGINE
240
EXTRACTED INPUT IMAGE DATA COMPONENTS
260
MAPPING ENGINE
244
RESULTANT IMAGES WITH STRUCTURES OF INTEREST
264
212
HR/VHR SATELLITE IMAGE DATA
224
LR SATELLITE IMAGE DATA
228
RESULTANT IMAGES WITH STRUCTURES OF INTEREST
264
BUS
220
PROCESSOR
208
OTHER
216

FIG. 2

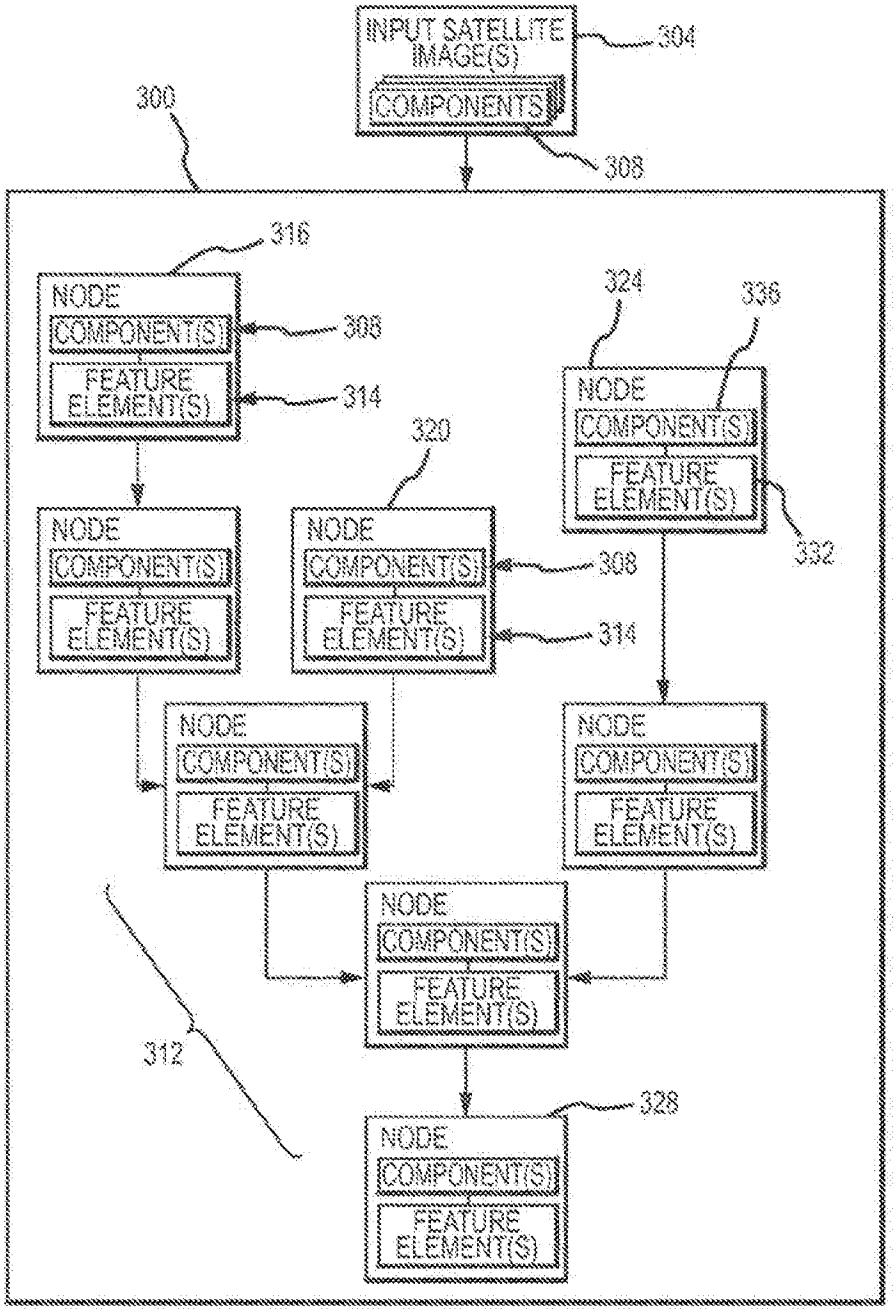

INPUT SATELLITE IMAGE(S)
304
COMPONENTS
308
300
316
NODE
COMPONENT(S)
308
FEATURE ELEMENT(S)
314
324
336
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
332
320
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
NODE
COMPONENT(S)
FEATURE ELEMENT(S)
312
328
NODE
COMPONENT(S)
FEATURE ELEMENT(S)

FIG.3

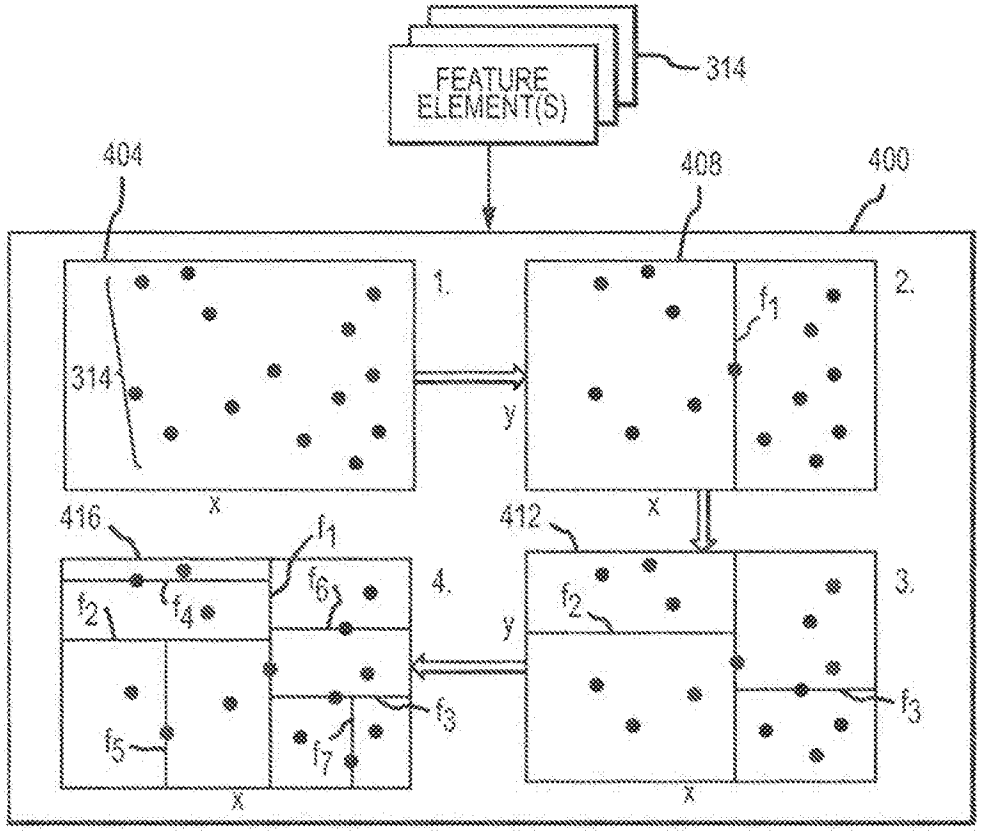

FEATURE ELEMENT(S)
314
404
408
400
1.
2.
3.
4.
314
x
y
x
x
y
x
416
412
f_1
f_2
f_4
f_6
f_2
f_3
f_5
f_7
f_3

FIG.4a

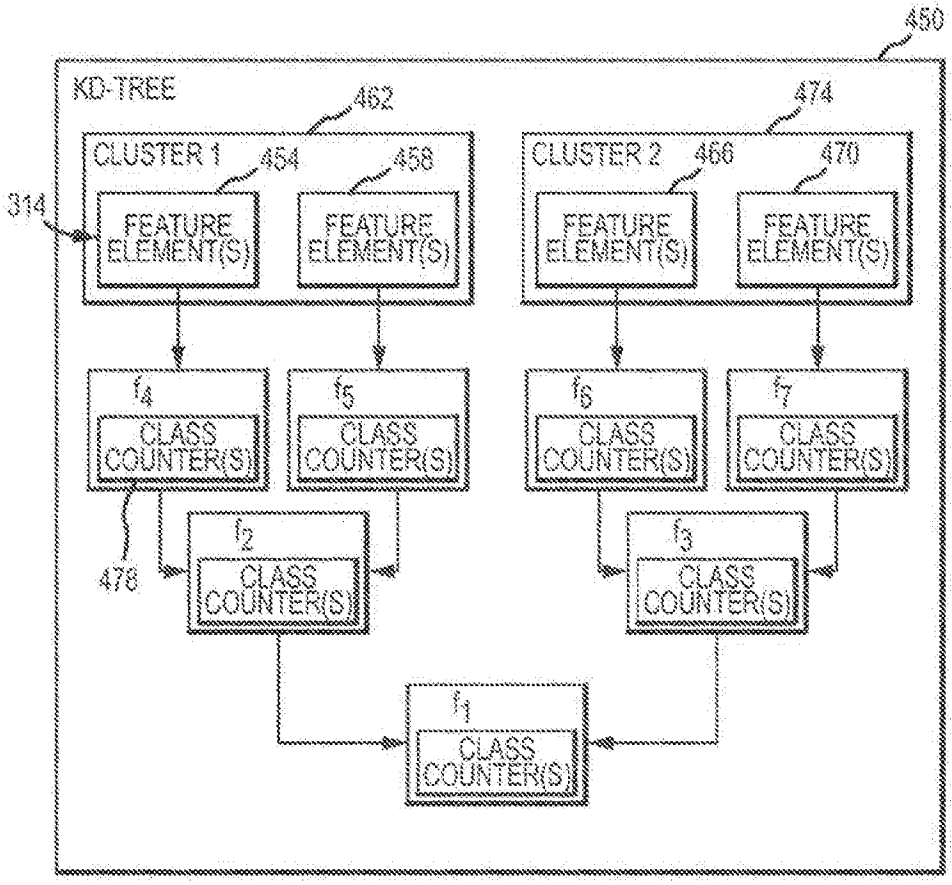

450
KD-TREE
462
474
CLUSTER 1
454
458
CLUSTER 2
466
470
314
FEATURE ELEMENT(S)
FEATURE ELEMENT(S)
FEATURE ELEMENT(S)
FEATURE ELEMENT(S)
f_4
f_5
f_6
f_7
CLASS COUNTER(S)
CLASS COUNTER(S)
CLASS COUNTER(S)
CLASS COUNTER(S)
478
f_2
f_3
CLASS COUNTER(S)
CLASS COUNTER(S)
f_1
CLASS COUNTER(S)

FIG.4b

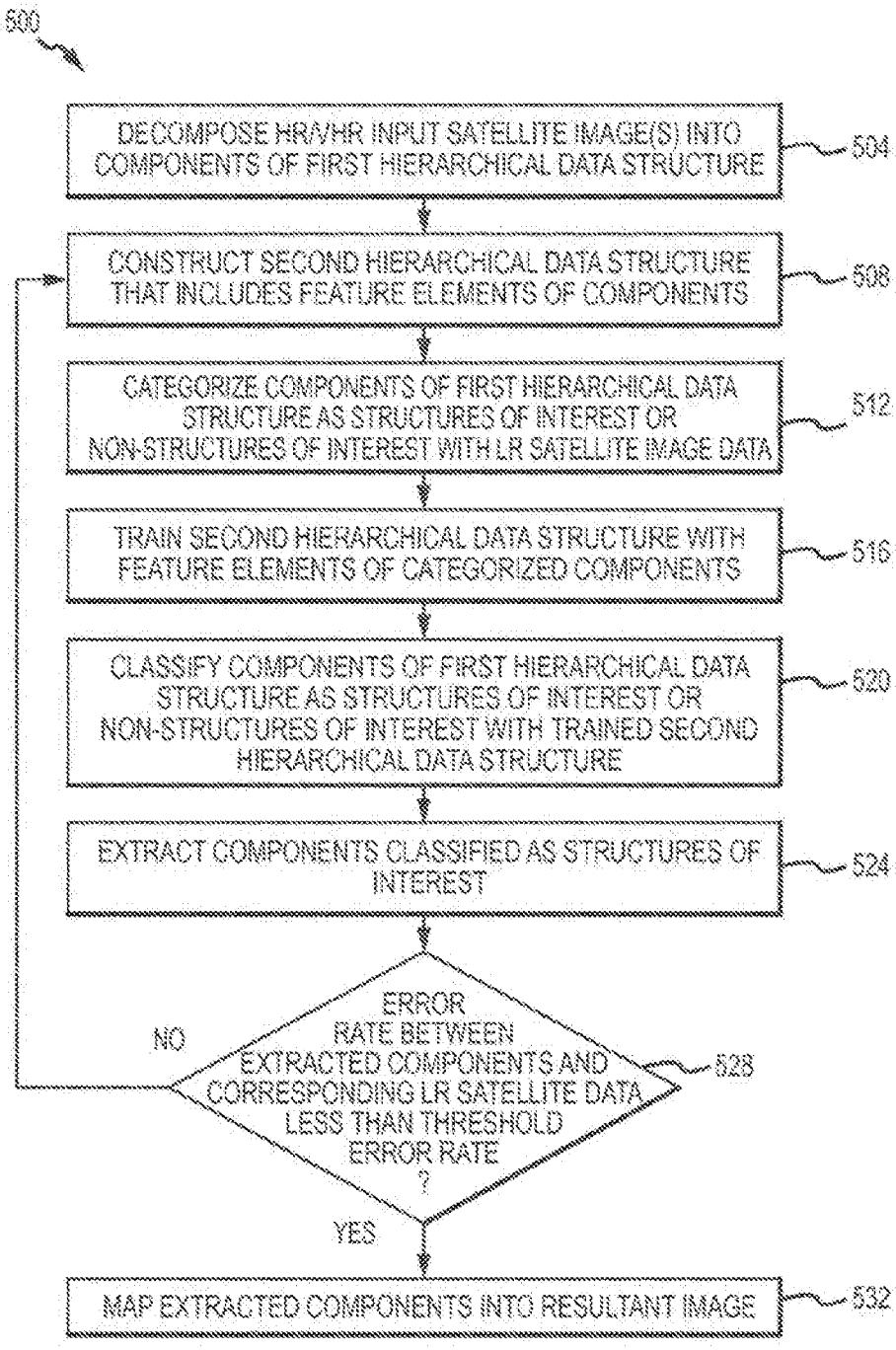

500
DECOMPOSE HR/VHR INPUT SATELLITE IMAGE(S) INTO COMPONENTS OF FIRST HIERARCHICAL DATA STRUCTURE
504
CONSTRUCT SECOND HIERARCHICAL DATA STRUCTURE THAT INCLUDES FEATURE ELEMENTS OF COMPONENTS
508
CATEGORIZE COMPONENTS OF FIRST HIERARCHICAL DATA STRUCTURE AS STRUCTURES OF INTEREST OR NON-STRUCTURES OF INTEREST WITH LR SATELLITE IMAGE DATA
512
TRAIN SECOND HIERARCHICAL DATA STRUCTURE WITH FEATURE ELEMENTS OF CATEGORIZED COMPONENTS
516
CLASSIFY COMPONENTS OF FIRST HIERARCHICAL DATA STRUCTURE AS STRUCTURES OF INTEREST OR NON-STRUCTURES OF INTEREST WITH TRAINED SECOND HIERARCHICAL DATA STRUCTURE
520
EXTRACT COMPONENTS CLASSIFIED AS STRUCTURES OF INTEREST
524
ERROR RATE BETWEEN EXTRACTED COMPONENTS AND CORRESPONDING LR SATELLITE DATA LESS THAN THRESHOLD ERROR RATE ?
528
NO
YES
MAP EXTRACTED COMPONENTS INTO RESULTANT IMAGE
532

FIG.5

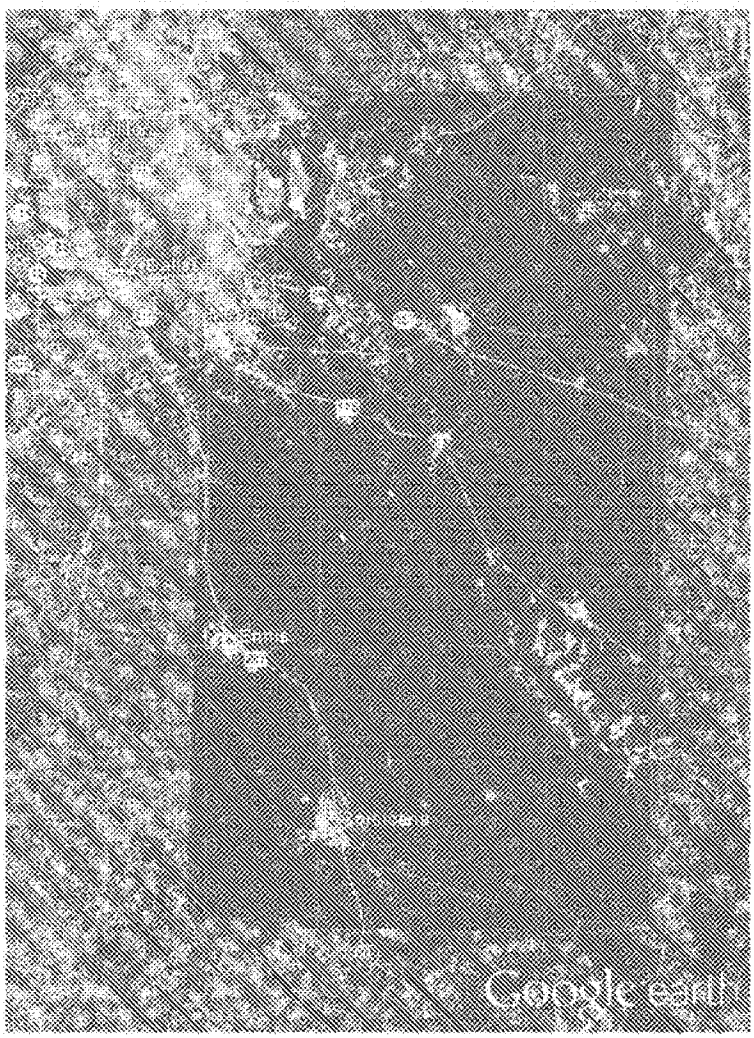


FIG.6

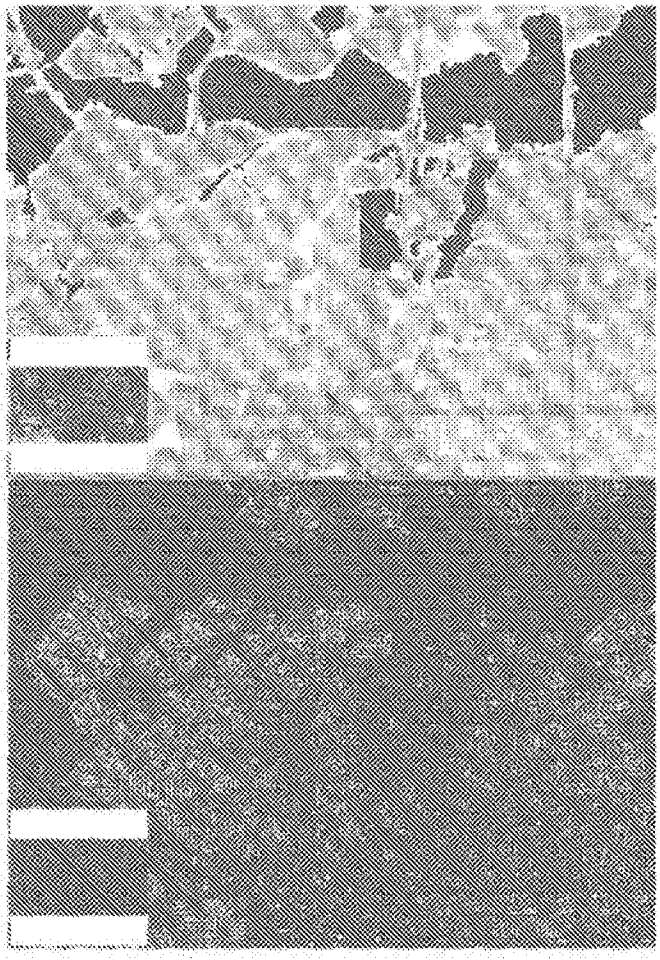


FIG.7b

FIG.7a



FIG.8b

FIG.8a

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/048537

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 7,16-18,21,26,41,46
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
The claims refer to a multiple dependent claim that refers to other multiple dependent claims. Therefore, the meaning of the technical feature to which they refer is vague and unclear.

3. ☒ Claims Nos.: 6,8-9,15,19-20,22-25,27-33,38-40,45,47-50
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of any additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/048537

A. CLASSIFICATION OF SUBJECT MATTER

G06Q 50/10(2012.01)i, G06Q 50/26(2012.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06Q 50/10; G06K 9/36; G06F 17/30; G06K 9/62; G09G 5/00; G06Q 50/26

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: satellite image, high resolution, extract, taining, feature, map, geographic area

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011-0055290 A1 (QING-HU LI et al.) 03 March 2011 See abstract, claims 1, 15 and figure 7.	1-5,10-14,34-37 ,42-44
A	US 2013-0077819 A1 (WEI BU et al.) 28 March 2013 See abstract, claims 1-3, 19-20 and figure 12.	1-5,10-14,34-37 ,42-44
A	US 2009-0303251 A1 (ANDRAS BALOGH et al.) 10 December 2009 See abstract, claims 1-5, 12, 36 and figures 6-7.	1-5,10-14,34-37 ,42-44
A	US 2009-0226114 A1 (MYUNG-JIN CHUNG et al.) 10 September 2009 See abstract, claims 1, 9 and figure 2.	1-5,10-14,34-37 ,42-44
A	US 2012-0027250 A1 (HO OWER et al.) 02 February 2012 See abstract, claim 1 and figure 3.	1-5,10-14,34-37 ,42-44

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 November 2014 (12.11.2014)	**12 November 2014 (12.11.2014)**

Name and mailing address of the ISA/KR	Authorized officer
International Application Division Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea	OH, Eung Gie
Facsimile No. +82-42-472-7140	Telephone No. +82-42-481-8744

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/048537

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0050290 A1	03/03/2011	CN 102027468 A	20/04/2011
		CN 102027468 B	23/04/2014
		WO 2009-137967 A1	19/11/2009
US 2013-0077819 A1	28/03/2013	AU 2012-227223 A1	11/04/2013
		CA 2790491 A1	23/03/2013
		NZ 602590 A	26/07/2013
US 2009-0303251 A1	10/12/2009	None	
US 2009-0226114 A1	10/09/2009	KR 10-0944462 B1	03/03/2010
		KR 10-2009-0096142 A	10/09/2009
		US 8320712 B2	27/11/2012
US 2012-0027250 A1	02/02/2012	CN 102270304 A	07/12/2011
		CN 102270304 B	18/06/2014
		US 2013-287307 A1	31/10/2013
		US 8508794 B2	06/08/2013

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 February 2015 (05.02.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/017855 A1

(51) International Patent Classification:
H04N 13/00 (2006.01)

(21) International Application Number:
PCT/US2014/049605

(22) International Filing Date:
4 August 2014 (04.08.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/861,610 2 August 2013 (02.08.2013) US

(71) Applicant: **XACTWARE SOLUTIONS, INC.** [US/US]; One Xactware Plaza, Orem, UT 84097 (US).

(72) Inventors: **TAYLOR, Jeffrey**; 1327 E. 900 S., Pleasant Grove, UT (US). **WERECKE, Edmund**; 6907 W. 10300 N., Highland, UT (US).

(74) Agents: **NIKOLSKY, Mark, E.** et al.; McCarter & English, LLP, Four Gateway Center, 100 Mulberry Street, Newark, NJ 07102 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

---- *with international search report (Art. 21(3))*

(54) Title: SYSTEM AND METHOD FOR DETECTING FEATURES IN AERIAL IMAGES USING DISPARITY MAPPING AND SEGMENTATION TECHNIQUES

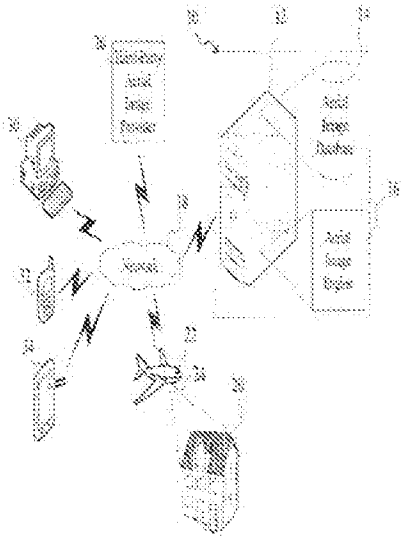


FIG. 1

(57) Abstract: A system for aerial image detection and classification is provided herein. The system comprising an aerial image database storing one or more aerial images electronically received from one or more image providers, and an object detection pre- processing engine in electronic communication with the aerial image database, the object detection pre-processing engine detecting and classifying objects using a disparity mapping generation sub-process to automatically process the one or more aerial images to generate a disparity map providing elevation information, a segmentation sub-process to automatically apply a pre-defined elevation threshold to the disparity map, the pre-defined elevation threshold adjustable by a user, and a classification sub-process to automatically detect and classify objects in the one or more stereoscopic pairs of aerial images by applying one or more automated detectors based on classification parameters and the pre-defined elevation threshold.

WO 2015/017855 A1

SYSTEM AND METHOD FOR DETECTING FEATURES IN AERIAL IMAGES USING DISPARITY MAPPING AND SEGMENTATION TECHNIQUES

SPECIFICATION

BACKGROUND

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Patent Application No. 61/861,610 filed on August 2, 2013, the entire disclosure of which is expressly incorporated herein by reference

FIELD OF THE DISCLOSURE

The present disclosure relates generally to the field of aerial image detection and classification. More specifically, the present disclosure relates to a system and method for detecting features in aerial images using disparity mapping and segmentation techniques.

RELATED ART

Accurate and rapid identification and estimation of objects in aerial images is increasingly important for a variety of applications. For example, roofing information is often used by construction professionals to specify materials and associated costs for both newly-constructed buildings, as well as for replacing and upgrading existing structures. Further, in the insurance industry, accurate information about construction materials and costs is critical to determining the proper costs for insuring buildings/structures.

Various software systems have been implemented to process aerial images to identify building structures and associated features thereof. However, such systems are often time-consuming and difficult to use, and require a great deal of manual input by a user. Further, such systems may not have the ability to improve results through continued usage over time.

In view of existing technology in this field, what would be desirable is a system that automatically and efficiently processes aerial images to automatically identify various types of objects in the images. Moreover, what would be desirable is a system that self-improves over time to become more accurate and efficient. Accordingly, what would be desirable, but has not yet been provided, is a system and method for detecting features in

aerial images using disparity mapping and segmentation techniques which addresses these needs.

SUMMARY

The present system of the current disclosure detects features in aerial images using disparity mapping and segmentation techniques. More specifically, the system includes an object detection pre-processing engine for object detection and classification using one or more aerial images. The object detection pre-processing engine includes disparity map generation, segmentation, and classification to identify various objects and types of objects in an aerial image. Detection algorithms, including region growing algorithms and split-and-merge segmentation algorithms, are applied to an image to identify structures. These component-based algorithms can evolve and become more efficient over time. The information derived from these pre-processed images can then be used by the mass production engine for the manual and/or automated production of drawings, sketches, and models. A quality control engine could also be used for ensuring the accuracy of any drawings, sketches, or models generated by the system.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing features will be apparent from the following Detailed Description, taken in connection with the accompanying drawings, in which:

FIG. 1 is a diagram showing the system of the present disclosure for detecting features in aerial images;

FIG. 2 is a diagram showing software components of the aerial image engine;

FIG. 3 is a flowchart showing the steps for processing one or more images by the object detection preprocessing engine;

FIG. 4 is an image depicting object classification carried out by the system using a pre-defined elevation threshold;

FIG. 5 is a diagram of various classification parameters utilized by the system;

FIG. 6 illustrates processing of an aerial image by the system using a region growing algorithm to identify structures, as well as footprints of such structures;

FIG. 7 illustrates processing of an aerial image by the system using a split-and-merge segmentation algorithm to identify structures;

FIG. 8 is a flowchart showing steps for evaluating and improving models generated by the system;

FIGS. 9-11 are screenshots showing various user interface screens generated by the system;

FIG. 12 is a diagram showing hardware and software components of the system;

FIG. 13 is a diagram showing processing steps carried out by the aerial image engine of the present invention for generating a point cloud from a pair of aerial images;

FIG. 14 is a disparity map generated by the engine using a depth from disparity method;

FIG. 15 is a point cloud generated by the engine using the depth from disparity method;

FIG. 16 is a disparity map generated by the engine using the eye-ray method;

FIG. 17 is a point cloud generated by the engine using the eye-ray method;

FIG. 18 is a diagram showing processing steps for generating a composite point cloud from multiple image pairs;

FIG. 19 is a nadir view of a point cloud based on a nadir pair of images;

FIG. 20 is an oblique view of a point cloud based on a nadir pair of images;

FIG. 21 is a slice of the point cloud that crosses the front windows of the building shown based on a nadir pair of images;

FIG. 22 is a nadir view of a point cloud based on an oblique pair of images;

FIG. 23 is an oblique view of a point cloud based on an oblique pair of images;

FIG. 24 is a slice of the point cloud that crosses the front windows of the building shown based on an oblique pair of images;

FIG. 25 is a nadir view of a point cloud based on a combination of nadir and oblique pairs of images;

FIG. 26 is an oblique view of a point cloud based on a combination of nadir and oblique pairs of images;

FIG. 27 is a slice of the point cloud that crosses the front windows the building shown based on a combination of nadir and oblique pairs of images; and

FIG. 28 is a diagram illustrating modules and associated processing steps for automated building recognition carried out by the aerial image engine.

DETAILED DESCRIPTION

The present disclosure relates to a system and method for detecting features in aerial images using disparity mapping and segmentation techniques, as discussed in detail below in connection with **FIGS. 1-28**.

FIG. 1 is a diagram showing the system of the present disclosure for detecting features in aerial images, indicated generally at **10**. The system **10** comprises a computer system **12** (e.g., a server) having an aerial image database **14** stored therein and a software aerial image engine (module) **16**. The database **14** could be stored on the computer system **12**, or located externally (e.g., in a separate database server in communication with the system **10**). As will be discussed in greater detail below, the aerial image engine **16** allows users to detect features in aerial images and generate three-dimensional models therefrom.

The system **10** can communicate through a network **18** with one or more of a variety of image providers to obtain aerial images or photographs of a building structure **20** and can store them in the aerial image database **14** in any suitable format, such as JPEG, TIFF, GIF, etc. Network communication could be over the Internet using standard TCP/IP communications protocols (e.g., hypertext transfer protocol (HTTP), secure HTTP (HTTPS), file transfer protocol (FTP), electronic data interchange (EDI), etc.), through a private network connection (e.g., wide-area network (WAN) connection, e-mails, electronic data interchange (EDI) messages, extensible markup language (XML) messages, Javascript Object Notation messages (JSON) file transfer protocol (FTP) file transfers, etc.), or any other suitable wired or wireless electronic communications format. Image providers that the computer system **12** could communicate with include, but are not limited to, an airplane **22** (or unmanned autonomous vehicle (UAV)) having a camera **24** capable of capturing images of the structure **20**, and/or a third-party aerial image provider **26**, such as Pictometry, Google, or Bing.

The computer system **12** could be any suitable computer server (e.g., a server with an INTEL microprocessor, multiple processors, multiple processing cores) running any suitable operating system (e.g., Windows by Microsoft, Linux, etc.). The computer system **12** includes non-volatile storage, which could include disk (e.g., hard disk), flash memory, read-only memory (ROM), erasable, programmable ROM (EPROM), electrically-erasable, programmable ROM (EEPROM), or any other type of non-volatile memory. The aerial image engine **16**, discussed in greater detail below, could be embodied as computer-readable instructions stored in computer-readable media (e.g., the non-volatile memory

mentioned above), and programmed in any suitable programming language (e.g., C, C++, Java, etc.).

The system **10** could be web-based and could allow for remote access to the system **10** over a network **18** (e.g., Internet, WAN, LAN, etc.) by one or more devices, such as a personal computer system **30**, a smart cellular telephone **32**, a tablet computer **34**, or other devices. It is also contemplated that at least some of the functionality of the system **10** could run locally on devices (e.g., personal computer **30**, smart cellular telephone **32**, tablet computer **34**, etc.) programmed with software in accordance with the present disclosure. It is conceivable that, in such circumstances, the device could communicate with a remote aerial image database over a network **18**.

FIG. 2 is a diagram showing software components **50** of the aerial image engine **16**. The components **50** include a project and task management system **52**, object detection preprocessing engine **60**, mixes production engine **62**, and quality control (QC) engine **64** (discussed in more detail below). The project and task management system **52** allows for the creation, processing, and monitoring of projects relating to the processing of aerial images. The projects could be based on high resolution aerial imagery of areas, where the images have been captured, copied, and pre-processed. Each project is tracked in detail (e.g., steps toward completion, time spent on each task, etc.) to help pinpoint target areas for optimization, simplification, and/or improvement.

The project and task management system **52** includes several distinct modules. More specifically, the system includes a management server **54**, work manager **56**, and web manager **58**. The management server **54** is a set of web services that store and serve geo-referenced data, including raw data (e.g., data generated by computer vision (CV)) and elaborated data (e.g., new and previous sketches, ITV's (insurance-to-value), insurance claims, and other related data). The management server **54** also provides a feedback mechanism that lets users quickly and efficiently return new and improved training data to the object detection preprocessing engine **60**.

The work manager **56** is a set of web services that dispatches tasks to low-cost, highly-trained operators, and then processes and stores the results of the work that they accomplish. The work manager **56** ensures that projects and tasks are assigned orderly based on priority and urgency levels. For example, customer requests could be assigned the highest priority, followed by customer PIF (policy in force) addresses, and then AOI's (areas of interest) with the most historical significance. The web manager **58** is a full web

application user interface that allows managers to handle creating projects, managing property contractors and operators, work monitoring, and tracking of historic data, productivity statistics (e.g., per operator, team, and/or project block, etc.), and other statistics.

The object detection preprocessing engine **60**, discussed in more detail below, detects structures in images, and then processes the images to identify different types of objects. More specifically, the object preprocessing engine **60** processes imagery to analyze stereoscopic pairs of images and detect various objects of interest (e.g., buildings, trees, pools, noise (elements with a significant level of entropy), etc.). For example, the object detection preprocessing engine **60** could take preprocessed building structure perimeter information, add automatic line finding capabilities, and provide the ability to gather height information from stereoscopic pairs.

The mass production engine **62** (e.g., mass production client application) is an automatically updated smart client (e.g., desktop, mobile, or web application) for quickly creating aerial models (e.g., 3D models) of one or more structures and accompanying prefill and metadata for an aerial image library (which could be address-based). The mass production engine **62** includes software tools to support the manual and automated process of creating a roof and/or property report. The mass production engine **62** could be a closed system which works in conjunction with designated web services and is programmed to protect any personally identifiable information (PII) data (e.g., the system could withhold from operations of the system actual address information or geocode information of a structure, remove imagery that is no longer needed from the local cache, etc.).

The quality control engine **64** ensures the accuracy of the model and related data generated from the images. The quality control engine **64** could be automated and/or could guide a technician in review and verification.

FIG. 3 is a flowchart showing the steps for processing one or more images by the object detection preprocessing engine **60**. Broadly, the process includes disparity map and/or point cloud generation sub-process **74**, segmentation sub-process **84**, and classification sub-process **90**. In step **72**, the system electronically receives one or more aerial images (e.g., imagery intake, imagery input data, etc.). High-level imagery specifications (e.g., proprietary satellite images) could be used for maximum automation and efficiency in relation to production, but the system could use lower-quality data inputs

(e.g., third-party data) as well. The clearer the aerial images that are used as input, the better the results of the automated detection algorithms.

In sub-process **74**, the system generates a disparity map and/or point cloud, which provides information about the elevation of the structures (e.g., objects, elements, etc.) present in the stereoscopic pair of images. To generate a disparity map and/or point cloud, in step **76**, the system uses world file information to process the overlapped region between stereoscopic images. One or more image pairs can be used in this process, and the resulting disparity maps and/or point clouds can be combined to gain additional information. In step **78**, the orientation of each image (e.g., left and right images) is processed, such as by using the camera position. In step **80**, if needed (e.g., particularly if the overlapping images are from different flight lines), the brightness of the images is normalized. A disparity map and/or point cloud is then generated in step **82**. The parameters used to generate the disparity map and/or point cloud are fine-tuned to account for differences between imagery data (e.g., differences produced by different camera systems, differences in sun angles, etc.) and other factors. The system could use other in-flight or post-flight processing systems capable of producing accurate disparity maps and/or point clouds.

In sub-process **84**, segmentation is applied to the image, which allows the system to detect changes in different parts of the image that are later grouped together into areas based on similarities. These areas are subsequently classified (e.g., as structures, trees, pools, etc.), as discussed below in more detail. To apply segmentation, in step **86**, a height threshold is applied to the disparity map and/or point cloud. This threshold is adjustable, but (for reasons relating to classification) should be taller than the height of a standard house or the tallest tree in a given area. In step **88**, one or more automated detectors (e.g., algorithms) are applied to objects in the image that are below the threshold to initially detect other objects (e.g., buildings). Automated detectors become more accurate and efficient over time and can be tuned and continually added. When one or more new detectors are added, the database could be reprocessed to run just the new detectors. Algorithms that could be used include region growing algorithms and/or split-and-merge segmentation algorithms (which could be used to find blobs that may be subsequently identified as structures, trees, noise, etc.), as well as object/feature detection algorithms. These algorithms are discussed in more detail in **FIGS. 6-7**.

In step **98**, classification is applied to detect and classify objects (e.g., buildings, trees, pools, noise, etc.). To apply classification, in step **92**, objects higher and/or taller than the certain predefined threshold (based on the height information derived by the disparity map) are automatically added as structures (e.g., automatically classified as buildings). In step **94**, areas are classified based on classification parameters using (and training) machine learning algorithms, such as neural networks. Machine learning algorithms and neural networks are powerful mechanisms which provide the system with the ability to learn and acquire experience from existing data and processes. For example, the network could be trained using an image database containing any number of stereoscopic image pairs, where the images are taken from different locations (including residential, industrial and commercial areas) and from datasets that have been captured using different types of sensor technology. The trained network could be tested using a test image database and an automated test tool. After the images have been pre-processed, a data package containing all information derived from the aerial images could be stored in a property database for future use by users or software applications.

FIG. 4 is an image **170** depicting object classification carried out by the system using a pre-defined elevation threshold. As shown, the height threshold **174** is adjustable to be a certain distance from the ground **172**. The threshold **174** is set to be larger than trees **176** and houses **178** in the area. Anything above that threshold **174**, such as a building tower **180**, is automatically classified as a building, because it is too tall to be anything else.

FIG. 5 is a diagram of various classification parameters **96** that can be utilized by the system. Classification is based on a set of specific parameters **96** that can be modified and evolved to refine and improve the classification or detection of new elements (e.g., cars, vegetation, etc.). Classification parameters **96** could include aspect **98**, disparity **114**, color **126**, and/or texture **132** parameters. More specifically, the aspect parameter **98** could include area **100** (e.g., blob area in pixels, where small blob areas are automatically discarded), aspect ratio **102** (e.g., best blob fitting oriented rectangle width and height ratio), rectangle fitting **104** (e.g., best blob fitting oriented rectangle area and blob area ratio, which tends to separate buildings from trees and noise), contour defects **106** (e.g., sum of defects of the blob convex polygon, which tends to separate buildings from trees and noise, where the noise generates more defects), HOG (histogram of oriented gradient) maximum **108** (e.g., distance between two main peaks in the HOG, and is ideally used for

buildings having a distance of 90°), HOG property **110** (e.g., contour pixel proportion that belongs to the two main peaks in the HOG, which tends to distinguish buildings from trees), and/or HOG uniform **112** (e.g., Kolmogorov test of similarity between the HOG and the uniform distribution, which shows a low similarity in buildings and a high value in trees due to spread orientations).

The disparity parameter **114** could include height mean **116** (e.g., mean of the blob disparity values, because noise has lower values than buildings or trees), height deviation **118** (e.g., standard deviation of the blob disparity values), distance to height **120** (e.g., sum of contour pixel distance to the edges of the disparity map, because noise usually presents a high distance value), contour correspondence **122** (e.g., sum of contour correspondences with contrasted disparity, because buildings and trees present a high contour correspondence value), ground prop **124** (e.g., analysis of the disparity between a reference point (ground point) and a studied point of a given blob, because noise usually belongs to ground). The color parameter **126** could include RGB (red green blue) **128** (e.g., mean value of color channels, such as to separate buildings from trees, which are usually green) and HSV (hue, saturation value) **130** (e.g., mean value of HSV channels). The texture parameter **132** could include deviation mean **134** (e.g., mean of the deviation of a window ceiling of the blob, which could separate trees from buildings due to contrasted lighting in leaves) and/or Sobel mean **136** (e.g., mean of the deviation of a window ceiling of the blob with a high pass filter applied to increase contrast).

FIG. 6 illustrates processing of an aerial image **150** by the system using a region growing algorithm to identify structures, as well as footprints **152** of such structures. The algorithm uses the highest points found in the disparity map as seed values and begins the region growing process from there outward. The region growing algorithm uses not only color information, but also disparity map information, which prevents growing over low-elevation areas. This works particularly well on high-resolution imagery (though the processing time per image is relatively large). However, if the image resize parameter is too large, it can cause inaccuracies in the structure footprint detected. Parameters of the region growing algorithm could include:

Parameter	Default Value
Image resize	30%
Ground proportion	45%
Similar RGB & HSV	Fuzzy set {0, 5, 10}
Building detection threshold	Disparity = 200
Noise threshold	Area = 700 pixels
Thresholding analysis step	Disparity = 10
Minimum seed area	100 pixels

Table 1

FIG. 7 illustrates processing of an aerial image **162** by the system using a split-and-merge segmentation algorithm to identify structures. The split-and-merge segmentation algorithm (which is faster than the region growing algorithm) looks at the whole image **162** and determines if it is non-uniform or uniform. If it is non-uniform, the image is split into four squares. Then, it looks at each of those four squares and repeats the process on the squares until a complete segmentation map is created, where all squares are 100% uniform within them, as shown in image **164**. This algorithm is useful for finding differences between different areas of an image and detecting the location of the changes. This, as shown in image **166**, can be used to define the roof lines that separate the different sections of a roof. The split-and-merge algorithm uses not only color information but also the disparity map information (e.g., height map information) as input data. In this way, the interior lines are detected based on both the imagery and the disparity map information.

FIG. 8 is a flowchart showing steps **200** for evaluating and improving models generated by the system. This process is optional and not required for object detection. The basic workflow includes determining the dimensions of the structure and creating a sketch or drawing. For example, the system provides the ability to determine and define a geocode at a front door (or one of the front doors) of a structure. Further, the system provides the ability to determine the footprint of the structure, roof dimensions (e.g., number of faces and other roof attributes), default area living space and non-living space (e.g., square feet), number of stories, number of car garage attributes, number and size of housing attributes (e.g., windows and doors), type of roofing material (e.g., shingles, metal, wood shake, etc.), exterior wall materials (e.g., stucco, brick, stone, etc.), and the number,

size, and material of exterior property attributes (e.g., swimming pools, decks, etc.), among other things.

In step **204**, the project and task management system (e.g., web manager application) guides a manager **202** in creating and publishing one or more projects. Publishing the project (automatically or manually) assigns it to a specific team (or operator) and makes the tasks in the project available in a queue. The manager can prioritize tasks within a project and across projects, thereby controlling the priority (on a per project basis) of how the models and metadata are processed.

In step **206**, an operator of the assigned team, once available, is automatically assigned the highest priority task from the queue. The necessary pre-processed data, including data defaults and imagery, is then retrieved from one or more databases for the operator. These secondary methods are provided for operators to derive information where required and where automated detectors yield inaccurate or undetected results. Generally, the mass production engine guides an operator through the following steps: define **210**, perimeter **212**, interior lines **214**, metadata **216**, and submit **218**, as discussed below. In step **210**, the mass production engine allows the operator **208** to define the property by displaying for his/her review one or more images from the specified location and the default data thereof. When required, the operator **208** marks which buildings and related structures belong to that particular property. This provides the operator **208** with the ability to separate and combine structures and/or to identify new structures, which is useful if the object preprocessing engine did not accurately find a structure or merged together separate structures. If a new structure is created, a new task will be added to the appropriate queue and subsequently assigned to another operator. Where the parcel boundary geographic accuracy and/or detector default data is acceptable, this step would only require a quick review and verification by the operator **208**.

In step **212**, the mass production engine allows/guides the operator **208** to verify and/or edit (e.g., creates, adjusts, etc.) the roof perimeter. Although preprocessing would most likely have accurately identified the perimeter, it may be necessary to adjust the perimeter (e.g., moving the points defining the perimeter) to match the exact contour of the building. In step **214**, the mass production engine allows/guides the operator **208** to verify and/or edit (e.g., correct, add, remove, etc.) the interior lines of the structure.

In step **216**, the mass production engine allows/guides the operator **208** in creating the metadata associated with the property. The operator could examine the imagery and answer a specific set of questions about the property. The user interface would guide the operator **208** through desired attribute specifications or to verify automated pre-fill results. Answering the question could require clicking on a point on the model, such as marking the front door geo-code or verifying roof features. The metadata could include roof material (e.g., shingle, shake, metal, tile/slate or membrane), number and placement of roof features (e.g., chimneys, roof vents, turtle vents, skylights, etc.), front door geocode location, number of levels, walls, exterior material(s)/percentages, default area living compared with nonliving space, number, size, and placement of doors, windows, garage stalls, rain gutters, air conditioner units, trees, swimming pools, etc. After all phases of work have been successfully completed, in step **218**, the operator **208** submits the model and metadata.

In step **220**, automated QC checks (automated algorithms and/or operator input prompts) are implemented by the mass production engine to verify the validity and accuracy of the model and related data (e.g., metadata). This ensures that the images and related data will successfully import into other products (e.g., programs, engines, etc.). If the checks fail, the operator is notified of the error and the submission process is canceled. Otherwise, depending on the parameters of the project, the operator is given a new task, and the model is selected and added to the QC queue (or alternatively published for use).

In step **224**, when the QC technician **222** is ready, the system (e.g., web application) pulls the highest priority structure from the QC queue and displays it on top of the appropriate nadir and oblique imagery. The system also allows any other imagery of the particular location to be displayed with the model. In step **226**, the QC engine prompts the QC technician **222** to review both the model and related data (e.g., metadata). In step **228**, the engine prompts the QC technician **222** to mark the structure as either verified (to be published to the library) or rejected and returned to the operator for review (its priority increased to push it higher in the operator's queue). When rejecting the model, the QC technician **222** can specify the reason from a canned list of possible issues provided by the system and/or draft a custom message. Multiple levels of quality assurance (e.g., teams) could be configured with varying responsibilities.

FIGS. 9-11 are showing various user interface screens generated by the system. More specifically, **FIGS. 8-9** are screenshots **250**, **260** showing a split pane interface allowing multiple views (e.g., left view **252**, **262** and right view **254**, **264**) of the structure to help better evaluate and analyze details. The system provides an operator with basic tools including pan and zoom, the ability to display all images covering the structure, line and vertices adjustments (e.g., adding, editing and deleting vertices and lines), snapping appropriate angles between lines (90°, 45°, etc.), ensuring lines are parallel and/or horizontal, snapping to existing lines/vertices, etc. The interface could also provide upper toolbar **256**, **266** and lower toolbar **258**, **268**, providing further functionality (e.g., define property, create contour, create break lines, submit model, new vertex, new rectangular boundary, find geometries by color, autodetect model, force coplanarity, display filling color, show detected polygon, new block, remove block, etc.). This allows the operator to view and edit the model frame with the model frame superimposed on one or more aerial different aerial images of the roof (e.g., orthographic, etc.).

FIG. 11 is another screenshot of a user interface of the system. The screenshot **270** shows the split pane interface, with an aerial view of a housing structure in the left pane **272**, a three-dimensional model of the housing structure in the right pane **274**, and a submission window **276** overlaying both. The submission window **276** includes a comment area **282** for a user to manually write a report, a submit button **278**, and a list of possible issues or submission alternatives **280** to select (e.g., accept with warning, bad images, tree was found, pool was found, unrecognized area, etc.). The system could also present the operator with questions to record any useful feedback (e.g., for quality assurance).

FIG. 12 is a diagram showing hardware and software components of the system **300**. The system **300** comprises a processing server **302** which could include a storage device **304**, a network interface **308**, a communications bus **310**, a central processing unit (CPU) (microprocessor) **312**, a random access memory (RAM) **314**, and one or more input devices **316**, such as a keyboard, mouse, etc. The server **302** could also include a display (e.g., liquid crystal display (LCD), cathode ray tube (CRT), etc.). The storage device **304** could comprise any suitable, computer-readable storage medium such as disk, non-volatile memory (e.g., read-only memory (ROM), eraseable programmable ROM (EPROM), electrically-eraseable programmable ROM (EEPROM), flash memory, field-programmable gate array (FPGA), etc.). The server **302** could be a networked computer system, a

personal computer, a smart phone, tablet computer etc. It is noted that the server **302** need not be a networked server, and indeed, could be a stand-alone computer system.

The functionality provided by the present disclosure could be provided by an aerial image engine **306**, which could be embodied as computer-readable program code stored on the storage device **304** and executed by the CPU **312** using any suitable, high or low level computing language, such as Python, Java, C, C++, C#, .NET, etc. The network interface **308** could include an Ethernet network interface device, a wireless network interface device, or any other suitable device which permits the server **302** to communicate via the network. The CPU **312** could include any suitable single- or multiple-core microprocessor of any suitable architecture that is capable of implementing and running the program **306** (e.g., Intel processor). The random access memory **314** could include any suitable, high-speed, random access memory typical of most modern computers, such as dynamic RAM (DRAM), etc.

FIG. 13 is a diagram showing processing steps **400** for generating a point cloud from a pair of aerial images, performed by an aerial image engine, and more specifically the object detection pre-processing engine. It is noted that point cloud data could be acquired during data capture and/or during image processing.

At step **402**, a pair of aerial images (e.g., image pair) is selected/identified (automatically by the engine or manually by the user). The pair of aerial images could be electronically received from a computer system, electronically transmitted from a database, etc. The engine could utilize a number of constraints in selecting the pair of aerial images. For example, the engine could require the pair of aerial images to have the same basic orientation (e.g., both are vertical images, both are oblique west images, etc.), the engine could require that the images have a large overlap area, and/or the engine could require that there is a small difference in capture time between both images (e.g., to avoid the effect of illumination changes).

At step **404**, the engine projects an overlapping area on ground plane data (using the pair of aerial images of step **402**). More specifically, the engine calculates the overlapping area of the images, and projects the relevant area from both images onto a horizontal ground plane of an approximate height (e.g., where the height could be extracted from the image parameters). This corrects small scale and tilt differences between the two images.

At step **406**, the engine aligns the images to the direction of flight of an aircraft or other flying vehicle from which aerial images are being taken (e.g., the line joining the observation points of both images) to produce an aligned stereoscopic pair of aerial images at step **408**. This allows the engine to find horizontal correspondences between the two images. The engine rotates the images to align them to the flight direction to assimilate them to a canonical stereoscopic pair. The engine could also apply template matching to finely adjust the overlapping images.

At step **410**, the engine computes dense correspondence mapping (e.g., disparity map, using the aligned stereoscopic pair of images of step **408**). More specifically, the engine applies a multi-scale disparity map module to the stereoscopic pair of images. This provides a measurement of the discrepancy distance between corresponding features on both images. The engine assigns a disparity value to each pixel on at least one of the stereoscopic images (e.g., the left image).

Using a depth from disparity method, the engine calculates depth from disparity at step **412**, which then generates a point cloud at step **414** (and electronically transmits point cloud data to another computer system). More specifically, the engine calculates a height map by applying an algorithm to compute depth (e.g., distance from an observation point) as a linear function of the disparity value at each pixel (e.g., the focal distance of the camera and the distance between observation points). To generate the point cloud, the engine transforms coordinates of the height map from the aligned stereoscopic pair back to the original image pair.

Alternatively (or additionally), the engine computes point heights using an eye-ray method at step **416** (based on the pair of images of step **402**), and the produces the point cloud **414**. More specifically, the engine transforms disparity values of the disparity map from the aligned stereoscopic pair back to the original image pair. Then the engine applies the eye-ray method, which triangulates each point using the vision rays from both observation points. This provides a more precise height map than the depth from disparity method.

FIGS. 14-15 are views of a disparity map and point cloud generated from the depth from disparity method of **FIG. 13**. More specifically, **FIG. 14** is a disparity map **418** generated by the engine using the depth from disparity method. **FIG. 15** is a point cloud **420** generated by the engine using the depth from disparity method.

FIGS. 16-17 are views of a disparity map and point cloud generated from the depth from disparity method of **FIG. 13**. More specifically, **FIG. 16** is a disparity map **422** generated by the engine using the eye-ray method. **FIG. 17** is a point cloud **424** generated by the engine using the eye-ray method.

FIG. 18 is a diagram showing processing steps **500** for generating a composite point cloud from multiple image pairs, such as those used in **FIGS. 13-17**. At step **502**, multiple pairs of aerial images are selected/identified (automatically by the engine or manually by the user). The multiple pairs of aerial images could be electronically received from a computer system, electronically transmitted from a database, etc. The engine could utilize a number of constraints in selecting the pairs of aerial images (as similarly described above). For example, the engine could require that each pair of aerial images have the same basic orientation (e.g., both are vertical images, both are oblique west images, etc.), the engine could require that the images have a large overlap area, and/or the engine could require that there is a small difference in capture time between both images (e.g., to avoid the effect of illumination changes).

Once multiple image pairs have been selected, the engine applies a Levenberg-Marquadt optimization module **504** to the multiple image pairs. More specifically, at step **506**, the module **504** generates point clouds for each image pair (using the process described in **FIG. 13**). There are overlapping zones between different point clouds, although the overlap may not be a perfect fit. In step **508**, multiple point clouds are generated.

In step **510**, the engine calculates the error resulting from the multiple point clouds (e.g., discrepancy between overlapping zones). More specifically, the engine calculates 3D features for each point cloud. The engine evaluates the discrepancy between point clouds as an error metric that uses distances between corresponding 3D features. The engine accumulates the error metric to include overlaps between all point clouds.

At step **512** the engine determines whether the error is low (e.g., a pre-defined threshold). If no, the process proceeds to step **514**, and the engine calculates an error gradient according to image parameters. More specifically, the engine adjusts the camera parameters to each image covering a large area and containing many different buildings. The discrepancies between point clouds are expected to be produced by minor camera parameter errors (e.g., as the camera parameters may not be the best for each single

building on the image). The engine checks the change of error gradient against minor changes in camera parameters (e.g., using a Jacobain matrix and determinant).

Then, in step **516**, the engine modifies projection parameters toward a lower error value. More specifically, the engine makes small changes to the camera parameters so that the error is reduced in a new computation of the point clouds. The process then reverts back to step **506**, and new point clouds are generated. The process is repeated until the generated point clouds are calculated by the engine to have a low error. In this way, this process is an iterative gradient-descent optimization.

If, in step **512**, the engine makes a positive determination that the error is low (thereby concluding the Levenberg-Marquadt optimization), then the process proceeds to step **518** and the engine removes redundant points. More specifically, the engine removes redundant points by using the ones with higher confidence according to the orientation of each point cloud region. Then the engine generates a composite point cloud at step **520**. Redundant points are removed because a composite point cloud (including all points from each individual point cloud) contains a large amount of information, and discrepancies in overlapping areas (although low) may be seen as noise by other engines (e.g., modules, algorithms, etc.), such as by a plane detection module.

FIGS. 19-21 are views related to a point cloud based on a nadir pair of images. More specifically, **FIG. 19** is a nadir (vertical) view **522** of a point cloud based on a nadir pair of images. **FIG. 20** is an oblique view **524** of a point cloud based on a nadir pair of images. **FIG. 21** is a slice **526** of the point cloud that cross the front windows of the building shown based on a nadir pair of images.

FIGS. 22-24 are views related to a point cloud based on an oblique pair of images. More specifically, **FIG. 22** is a nadir (vertical) view **528** of a point cloud based on an oblique pair of images. **FIG. 23** is an oblique view **530** of a point cloud based on an oblique pair of images. **FIG. 24** is a slice **532** of the point cloud that cross the front windows of the building shown based on an oblique pair of images.

FIGS. 25-27 are views related to a point cloud based on a combination of nadir and oblique pairs of images. More specifically, **FIG. 25** is a nadir (vertical) view **534** of a point cloud based on a combination of nadir and oblique pairs of images. **FIG. 26** is an oblique view **536** of a point cloud based on a combination of nadir and oblique pairs of images. **FIG. 27** is a slice **538** of the point cloud that cross the front windows the building shown based on a combination of nadir and oblique pairs of images.

FIG. 28 is a diagram of modules and associated processing steps for automated building recognition (and/or recognition of any other object) carried out by an aerial image engine **600**, and more specifically the object detection pre-processing engine. The aerial image engine **600** could include one or more modules, such as a stereo processing module **602**, a segment-based induction module **604**, a contour detection module **605**, a roof model induction model **606**, an optimization module **608**, and an evaluation module **610**.

In step **612**, the stereo processing module **602** automatically selects/identifies/receives (or a user manually selects/identifies) a set of calibrated aerial images (e.g., as input). The calibrated aerial images could be received electronically from another computer system, a database, etc. In step **614**, the segment-based induction module **604** uses the set of calibrated aerial images to detect 2D line segments on each image. The segment-based induction module **604** matches lines and generates candidate 3D lines at step **616**, and detects and discards ground lines at step **618**. Then, the segment-based induction module **604** detects horizontal lines by finding parallel clusters at step **620**, and could concurrently, detect oblique lines by finding clusters of line intersections at step **622**. In step **624**, the segment-based induction module **604** induces a set of roof model primitives, which are subsequently used at step **678** by the optimization module **608**, discussed in more detail below.

Returning to step **612**, once the set of calibrated aerial images are selected/identified, the process (concurrently) proceeds to step **626**, where the stereo processing module **602** selects image pairs in any orientation, and then the image pairs are rectified in step **628**. The stereo processing module **602** computes a multiscale disparity map at step **630**, then computes and merges pairwise point clouds at step **632**, and then generates a global point cloud at step **634**. The global point cloud generated is used at step **656** by the roof model induction module **606**, discussed in more detail below.

Returning to step **612**, once the set of calibrated aerial images are selected/identified, the process (concurrently) proceeds such that the stereo processing module **602** selects a pair of nadir images in step **636**, and then generates a stereo pair of images in step **638**. The stereo processing module **602** rectifies the stereo pair of images at step **640**, and then (concurrently) projects and aligns the stereo images at step **642**. The stereo processing module **602** then computes a multiscale disparity map at step **644**, and computes and filters a point cloud at step **646**.

The process then proceeds to the contour detection module **605**. The contour detection module **605** includes one or more algorithms to detect contours. More specifically, the contour detection module **605** could include a grabcut approach algorithm **648**, an MSER (maximally stable extremal regions) approach algorithm **650**, and/or a point cloud approach algorithm **652**. The grabcut approach **648** and the MSER approach **650** each receive the selected pair of nadir images of step **636** and the computed multiscale disparity map of step **644** as inputs. The point cloud approach **652** receives the selected pair of nadir images of step **636** and the computed and filtered point cloud of step **646** as inputs. Each of the approaches then generates an output to be used by the roof model induction module **606**.

Processing proceeds to the roof model induction module **606** which builds contours at step **654** (based on the output of the contour detection module **605**), and detects planes at step **656** (based on the global point cloud generated at step **634**). Then the roof model induction module **606** finds intersecting lines at step **660** and generates an intersecting line adjacency graph at step **662**. Concurrently with steps **660**, **662**, the roof model induction module **606** generates a plane image adjacency graph at step **658**. The roof model induction module **606** then generates a set of roof model primitives at step **664**.

The process then proceeds to the optimization module **608**, which extracts segments from images in all views at step **666** (based on the set of calibrated aerial images of step **612** and based on the set of roof model primitives of step **664**). The optimization module **608** then applies a distance transform per image at step **668** and (concurrently) applies a distance to the nearest segment at step **670**. The results/outputs of steps **668** and **6670** are then used as inputs in one or more optimization algorithms of the optimization module **608**. More specifically, the optimization algorithms could include a Lavenberg-Marquadt optimization algorithm **672**, a differential evolution optimization algorithm **674**, and/or a variable neighborhood search optimization algorithm **676**. Then at step **678** a set of adjusted primitives is generated by the optimization module **608** (based on the set of roof model primitives induced at step **624** and based on the output of the one or more optimization algorithms **672**, **674**, **676**).

The optimization module **608** then calculates overlapping with 2D lines at **680** (using the set of adjusted primitives **678**), and then applies one or more high overlapping transformation options at step **682**. Additionally, the optimization module **608** generates a model containing a roof and extensions at step **684**. The optimization module **608** applies

VNS (variable neighborhood search) optimization at step **686** and generates an adjusted model at step **688**. The adjusted model and VNS optimization are then outputted to the evaluation module **610**.

The process then proceeds to the evaluation module **610**, which measures error by comparing roof segments at step **690** (based on the adjusted model of step **688**, and based on a collection of blueprints with validated sketches of step **692**). The evaluation module **610** then generates an error metric at step **694**. Additionally, the evaluation module **610** generates confidence estimation at step **696** (based on the VNS optimization of steps **676** and **686**). The evaluation module **610** then generates a confidence metric at step **698**.

Having thus described the system and method in detail, it is to be understood that the foregoing description is not intended to limit the spirit or scope thereof. It will be understood that the embodiments of the present disclosure described herein are merely exemplary and that a person skilled in the art may make any variations and modification without departing from the spirit and scope of the disclosure. All such variations and modifications, including those discussed above, are intended to be included within the scope of the disclosure.

CLAIMS

1. A system for aerial image detection and classification, comprising:

an aerial image database storing one or more aerial images electronically received from one or more image providers; and

a computer system in communication with the aerial image database, the computer system including an object detection pre-processing engine in electronic communication with the aerial image database, the object detection pre-processing engine detecting and classifying objects using:

a disparity mapping generation module for automatically processing the one or more aerial images to generate a disparity map providing elevation information;

a segmentation module for automatically applying a pre-defined elevation threshold to the disparity map, the pre-defined elevation threshold adjustable by a user; and

a classification module for automatically detecting and classifying objects in the one or more stereoscopic pairs of aerial images by applying one or more automated detectors based on classification parameters and the pre-defined elevation threshold.

2. The system of Claim 1, wherein the object detection pre-processing engine automatically classifies as an object as a building if the object is above the pre-defined elevation threshold.

3. The system of Claim 1, wherein the classification parameters include at least one of aspect parameters, disparity parameters, color parameters, or texture parameters.

4. The system of Claim 1, further comprising a mass production engine in electronic communication with the object detection pre-processing engine, the mass production engine automatically producing drawings, sketches, and models using the elevation information generated by the object detection pre-processing engine.

5. The system of Claim 1, further comprising a project and task management system in electronic communication with the object detection pre-processing engine, the project and task management system guiding a user in creating and publishing one or more projects using the elevation information generated by the object detection pre-processing engine.

6. The system of Claim 1, wherein the one or more automated detects includes at least one of a region growing algorithm or a split-and-merge segmentation algorithm.

7. The system of Claim 1, wherein the disparity mapping generation module of the object detection pre-processing engine generates a point cloud.

8. The system of Claim 7, wherein the object detection pre-processing engine generates the point cloud by computing point heights using an eye-ray method.

9. A method for aerial image detection and classification, comprising:

electronically receiving at a computer system including an object detection pre-processing engine one or more aerial images from an aerial image database, the one or more aerial images electronically received at the aerial image database from one or more image providers;

automatically processing, by the object detection pre-processing engine, the one or more aerial images to generate a disparity map providing elevation information;

automatically applying, by the object detection pre-processing engine, a pre-defined elevation threshold to the disparity map, the pre-defined elevation threshold adjustable by a user; and

automatically detecting and classifying, by the object detection pre-processing engine, objects in the one or more stereoscopic pairs of aerial images by applying one or more automated detectors based on classification parameters and the pre-defined elevation threshold.

10. The method of Claim 9, wherein the object detection pre-processing engine automatically classifies an object as a building if the object is above the pre-defined elevation threshold.

11. The method of Claim 9, wherein the classification parameters include at least one of aspect parameters, disparity parameters, color parameters, or texture parameters.

12. The method of Claim 9, further comprising automatically producing, by a mass production engine, drawings, sketches, and models by using the elevation information generated by the object detection pre-processing engine, the mass production engine in electronic communication with the object detection pre-processing engine.

13. The method of Claim 9, further comprising guiding, by a project and task management system, a user in creating and publishing one or more projects using the elevation information generated by the object detection pre-processing engine, the project and task management system in electronic communication with the object detection pre-processing engine.

14. The method of Claim 9, wherein the engine executes at least one of a region growing algorithm or a split-and-merge segmentation algorithm.

15. The method of Claim 9, further comprising generating a point cloud by the object detection pre-processing engine.

16. The method of Claim 15, wherein the object detection pre-processing engine generates the point cloud by computing point heights using an eye-ray method.

17. A non-transitory computer-readable medium having computer-readable instructions stored thereon which, when executed by a computer system having an object detection pre-processing engine, cause the computer system to perform the steps of:

electronically receiving one or more aerial images from an aerial image database, the one or more aerial images electronically received at the aerial image database from one or more image providers;

automatically processing, by the object detection pre-processing engine, the one or more aerial images to generate a disparity map providing elevation information;

automatically applying, by the object detection pre-processing engine, a pre-defined elevation threshold to the disparity map, the pre-defined elevation threshold adjustable by a user; and

automatically detecting and classifying, by the object detection pre-processing engine, objects in the one or more stereoscopic pairs of aerial images by applying one or more automated detectors based on classification parameters and the pre-defined elevation threshold.

18. The non-transitory computer-readable medium of Claim 17, wherein the object detection pre-processing engine automatically classifies an object as a building if the object is above the pre-defined elevation threshold.

19. The non-transitory computer-readable medium of Claim 17, wherein the classification parameters include at least one of aspect parameters, disparity parameters, color parameters, or texture parameters.

20. The non-transitory computer-readable medium of Claim 17, further comprising automatically producing, by a mass production engine, drawings, sketches, and models by using the elevation information generated by the object detection pre-processing engine, the mass production engine in electronic communication with the object detection pre-processing engine.

21. The non-transitory computer-readable medium of Claim 17, further comprising guiding, by a project and task management system, a user in creating and publishing one or more projects using the elevation information generated by the object detection pre-processing engine, the project and task management system in electronic communication with the object detection pre-processing engine.

22. The non-transitory computer-readable medium of Claim 17, wherein the engine executes at least one of a region growing algorithm or a split-and-merge segmentation algorithm.

23. The non-transitory computer-readable medium of Claim 17, further comprising generating a point cloud by the object detection pre-processing engine.

24. The non-transitory computer-readable medium of Claim 23, wherein the object detection pre-processing engine generates the point cloud by computing point heights using an eye-ray method.

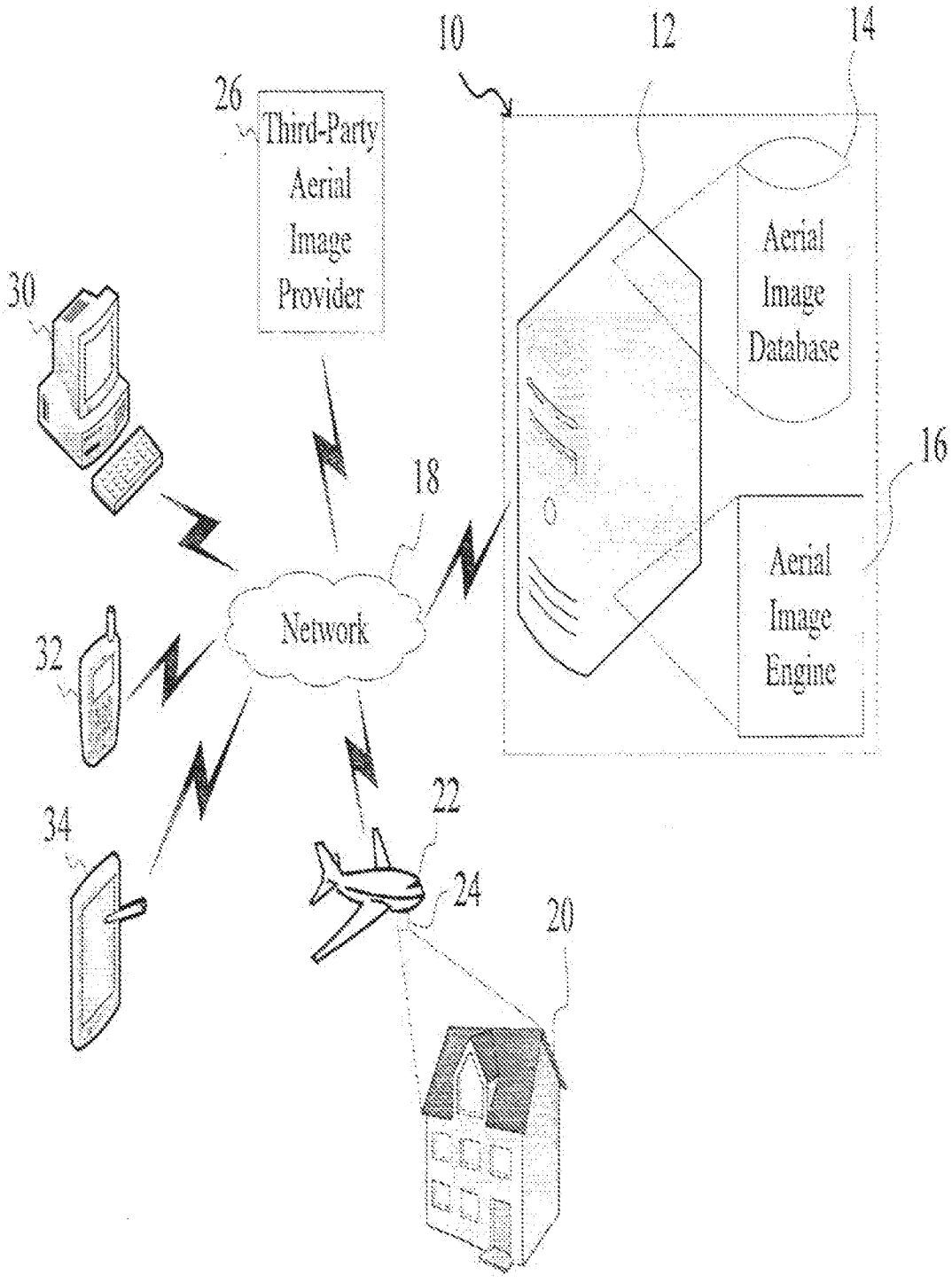

10
12
14
26
Third-Party Aerial Image Provider
Aerial Image Database
30
16
18
Aerial Image Engine
Network
32
22
34
24
20

FIG. 1

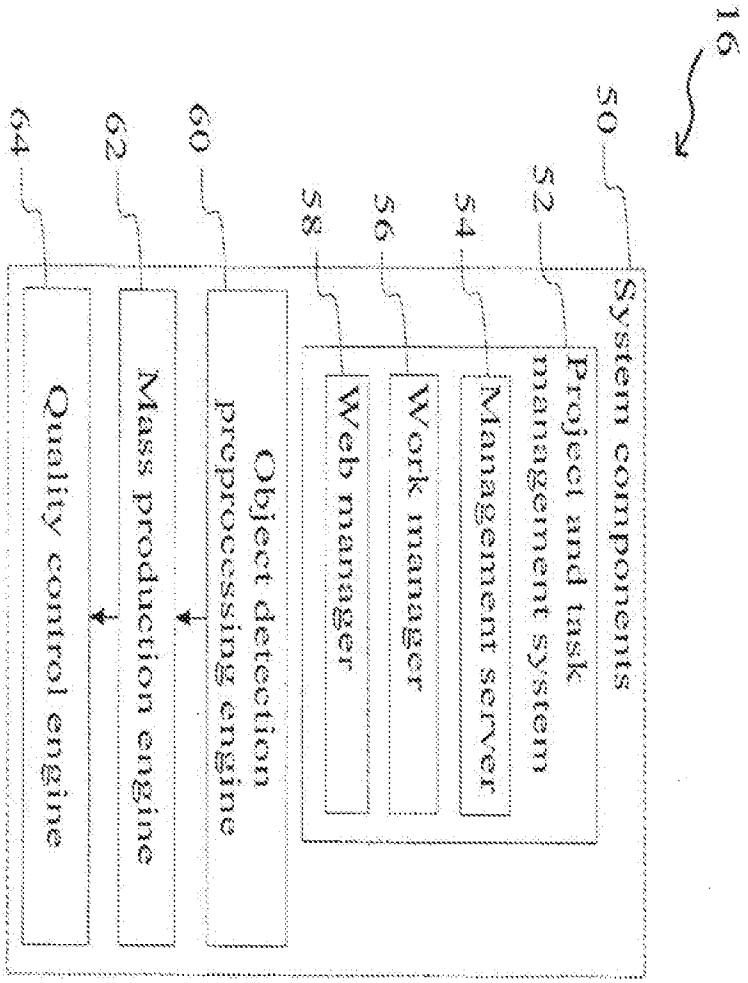

16
50
System components
52
Project and task management system
54
Management server
56
Work manager
58
Web manager
60
Object detection preprocessing engine
62
Mass production engine
64
Quality control engine

FIG. 2

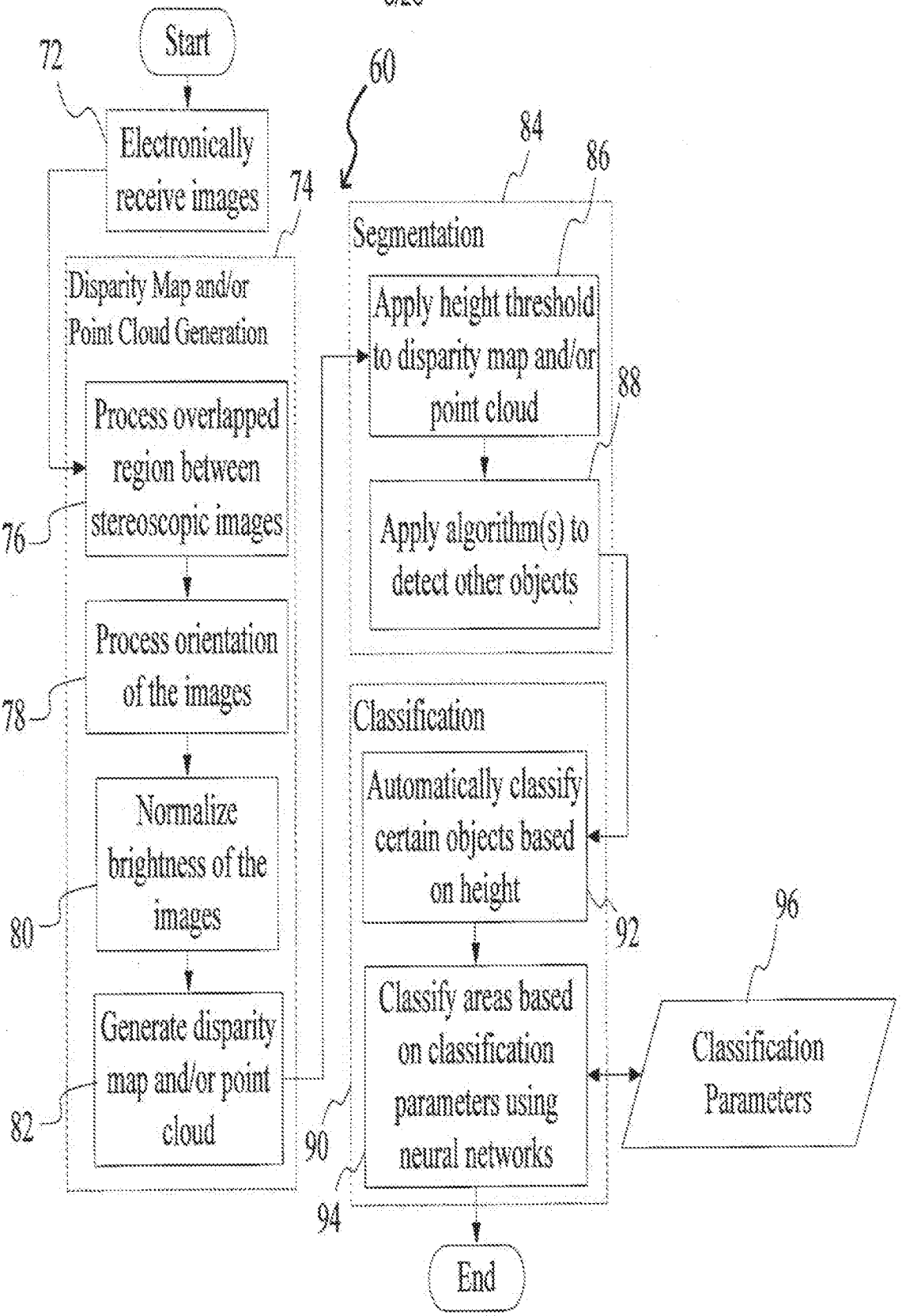

Start
72
Electronically receive images
60
84
86
74
Segmentation
Disparity Map and/or Point Cloud Generation
Apply height threshold to disparity map and/or point cloud
88
Process overlapped region between stereoscopic images
Apply algorithm(s) to detect other objects
76
Process orientation of the images
78
Classification
Automatically classify certain objects based on height
Normalize brightness of the images
80
92
96
Generate disparity map and/or point cloud
Classify areas based on classification parameters using neural networks
Classification Parameters
82
90
94
End

FIG. 3

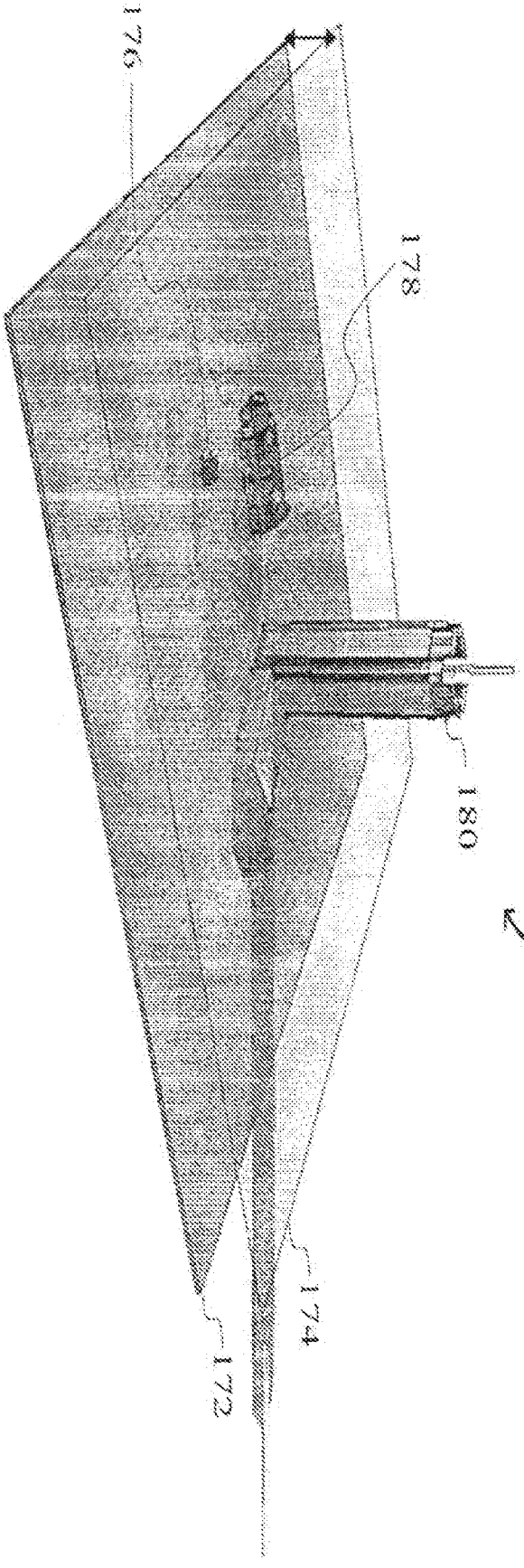


FIG. 4

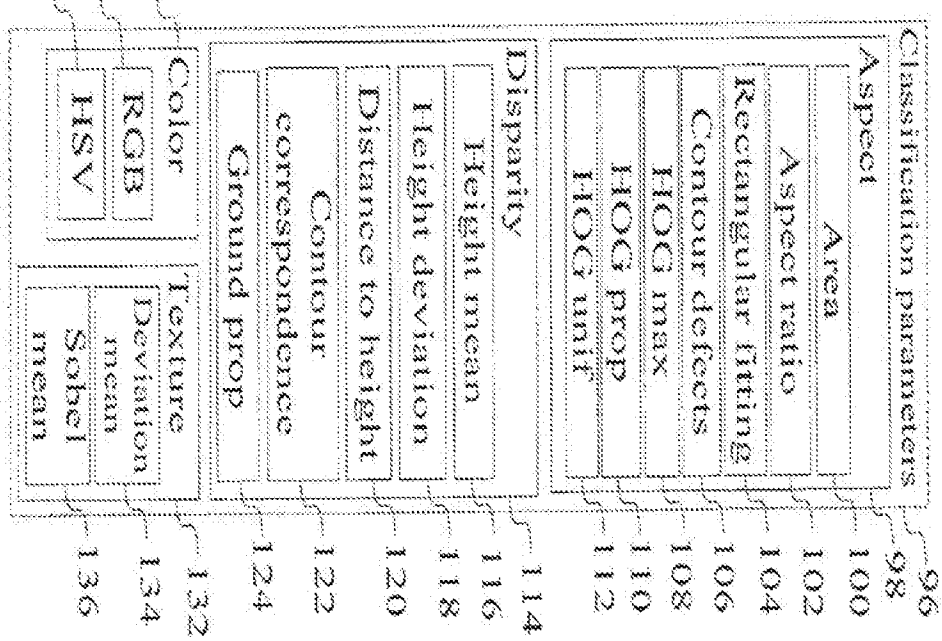

Classification parameters
96
Aspect
98
Area
100
Aspect ratio
102
Rectangular fitting
104
Contour defects
106
HOG max
108
HOG prop
110
HOG unif
112
Disparity
114
Height mean
116
Height deviation
118
Distance to height
120
Contour correspondence
122
Ground prop
124
Color
126
RGB
128
HSV
130
Texture
132
Deviation
134
Sobel mean
136

FIG. 5

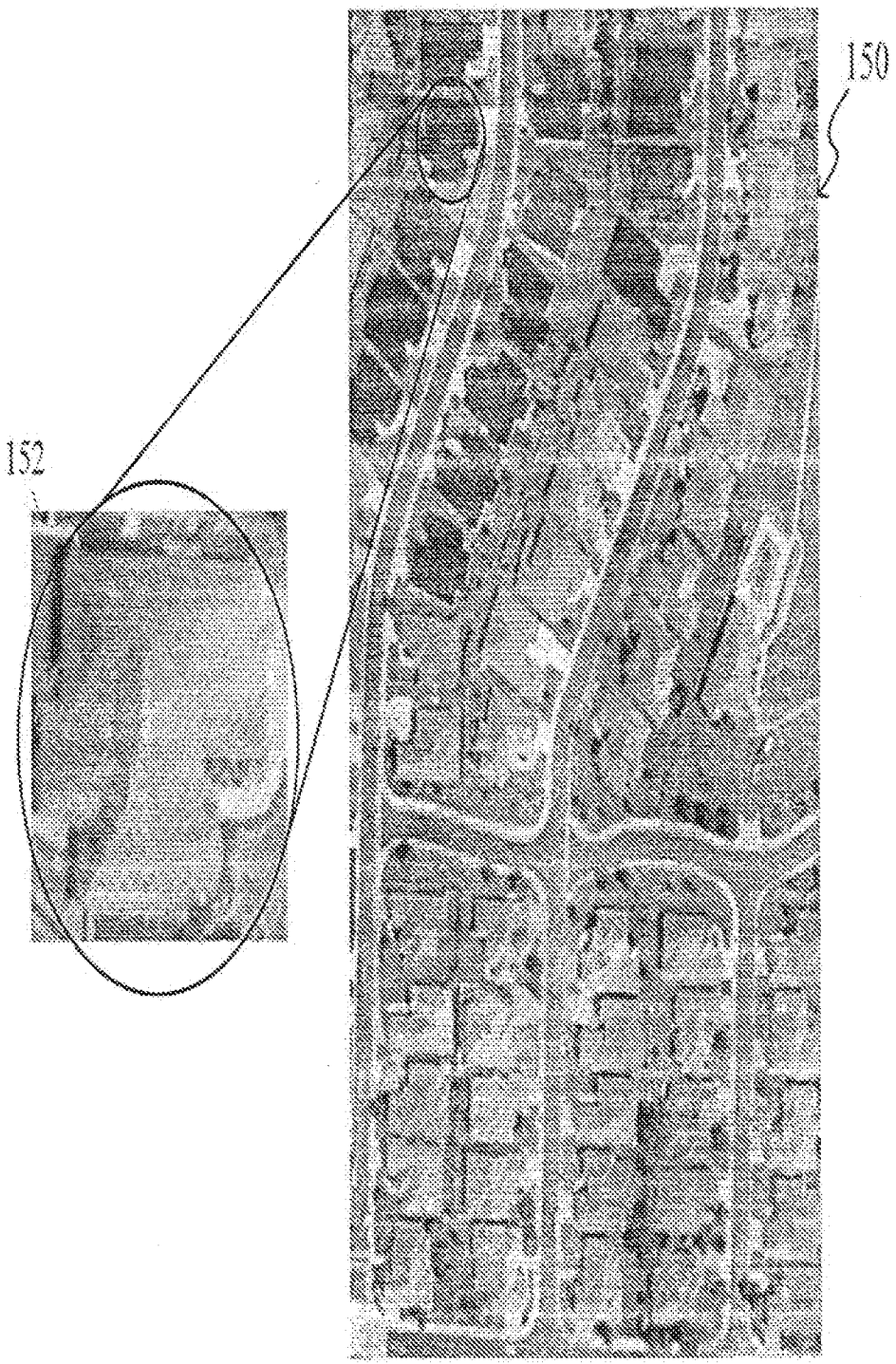


FIG. 6

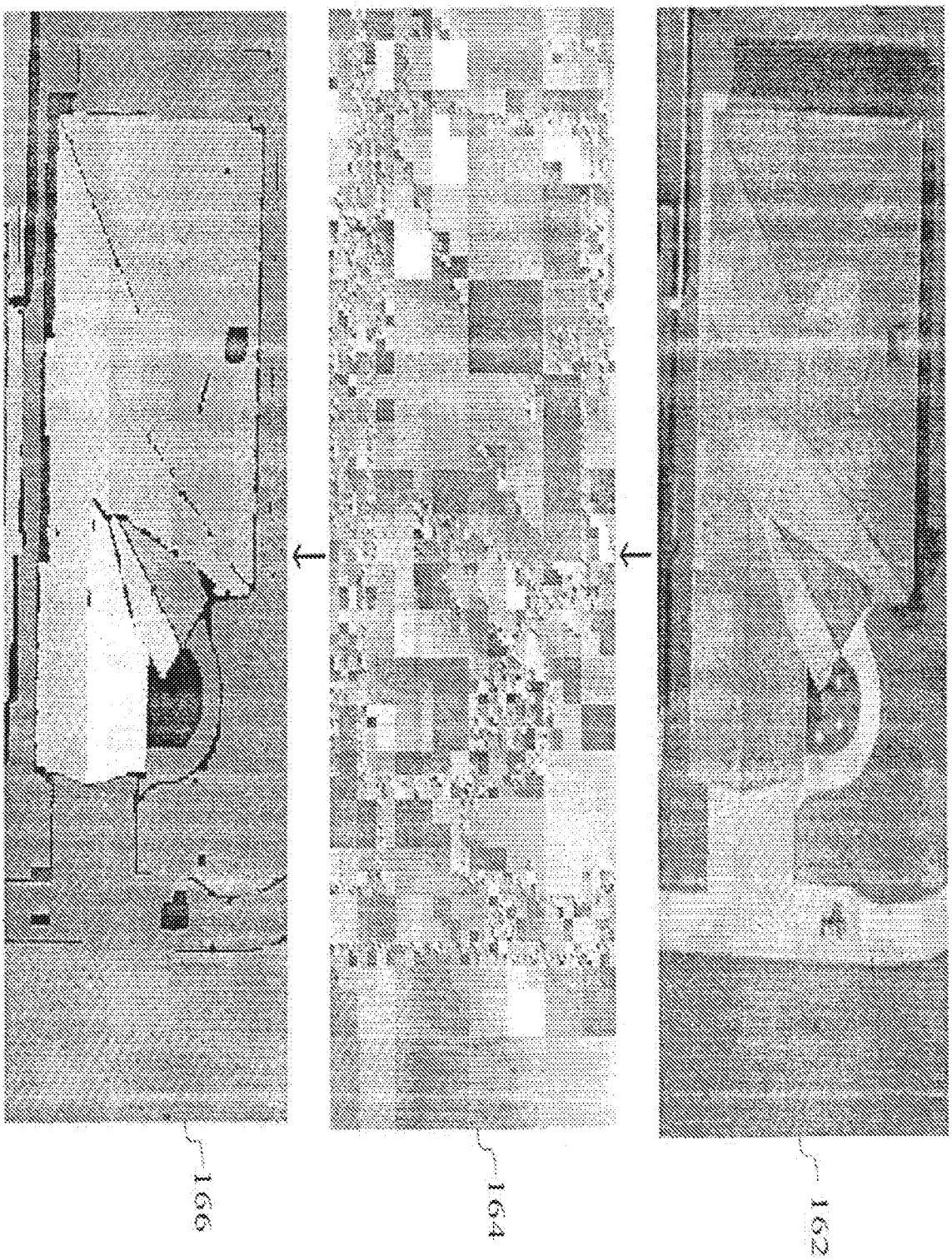


FIG. 7

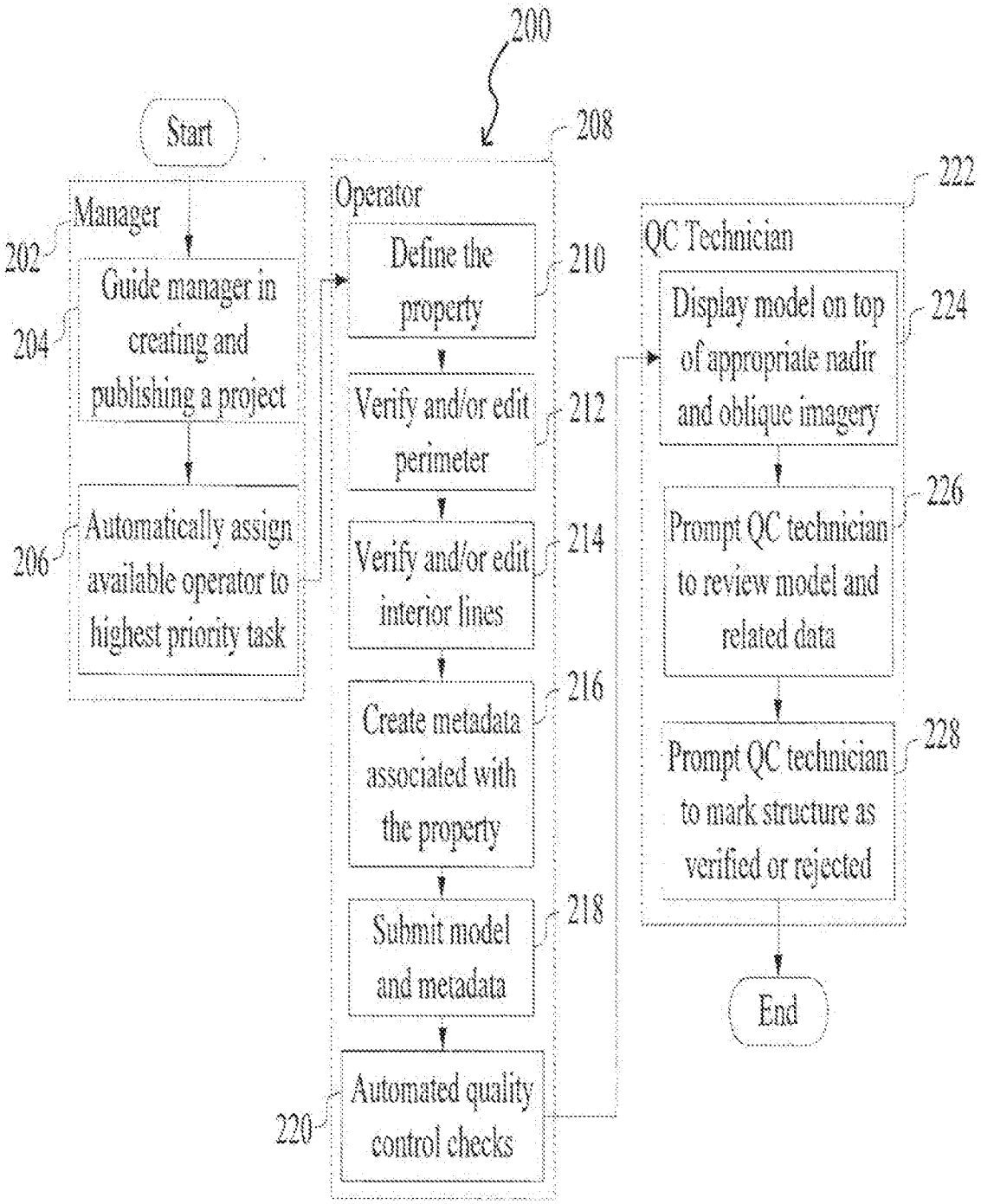

200
Start
Manager
202
Guide manager in creating and publishing a project
204
Automatically assign available operator to highest priority task
206
208
Operator
Define the property
210
Verify and/or edit perimeter
212
Verify and/or edit interior lines
214
Create metadata associated with the property
216
Submit model and metadata
218
Automated quality control checks
220
222
QC Technician
Display model on top of appropriate nadir and oblique imagery
224
Prompt QC technician to review model and related data
226
Prompt QC technician to mark structure as verified or rejected
228
End

FIG. 8

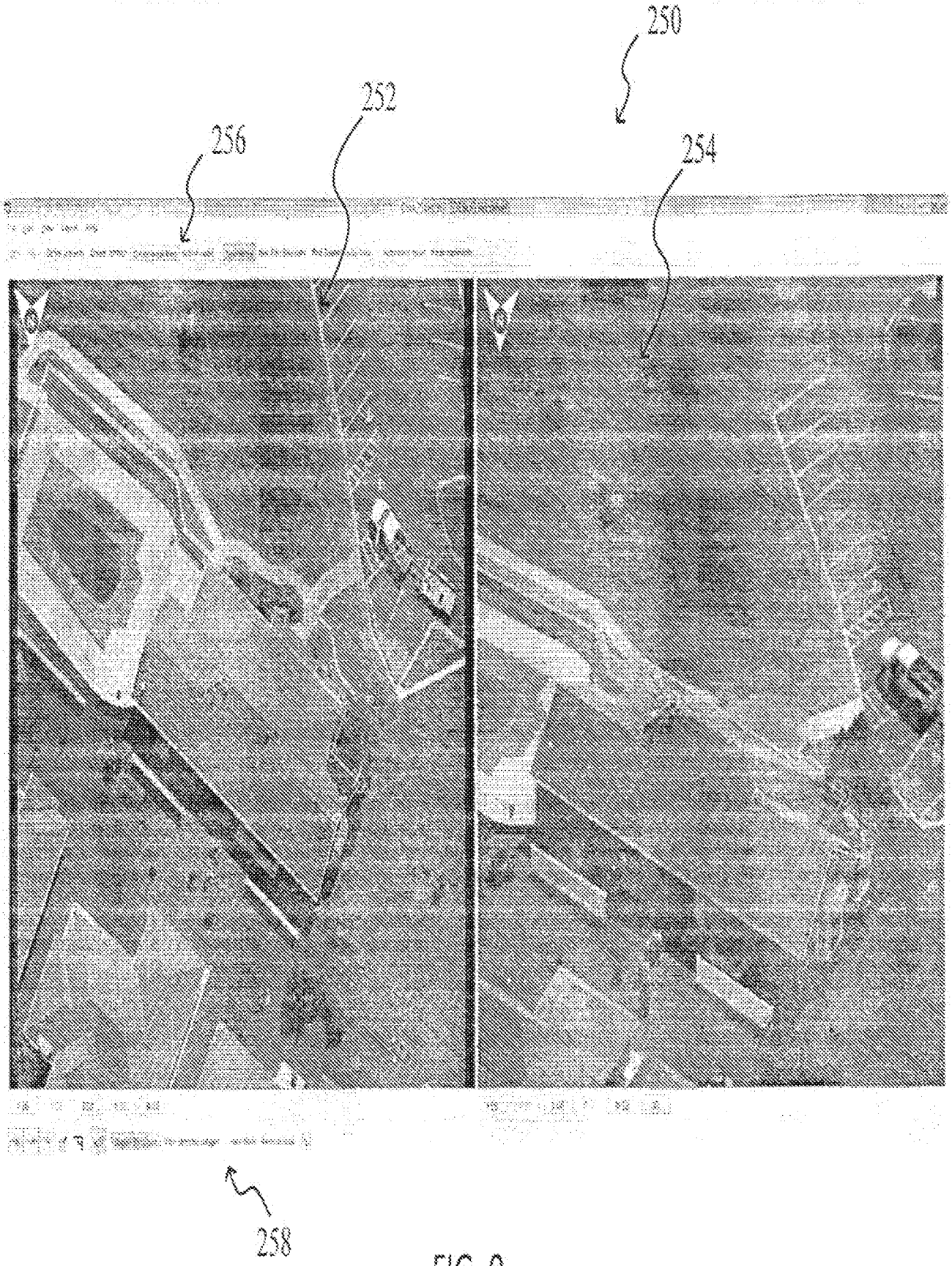


FIG. 9

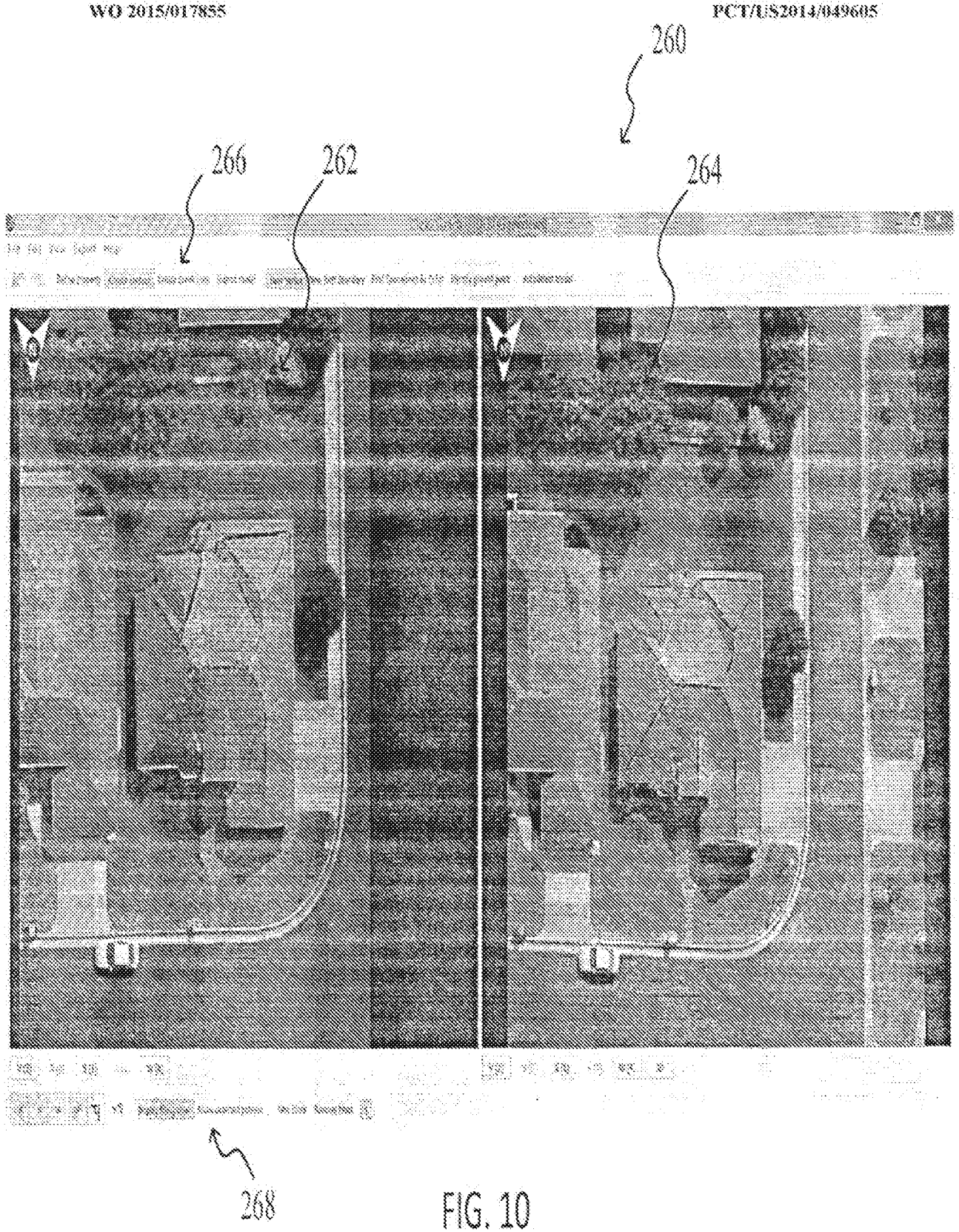


FIG. 10

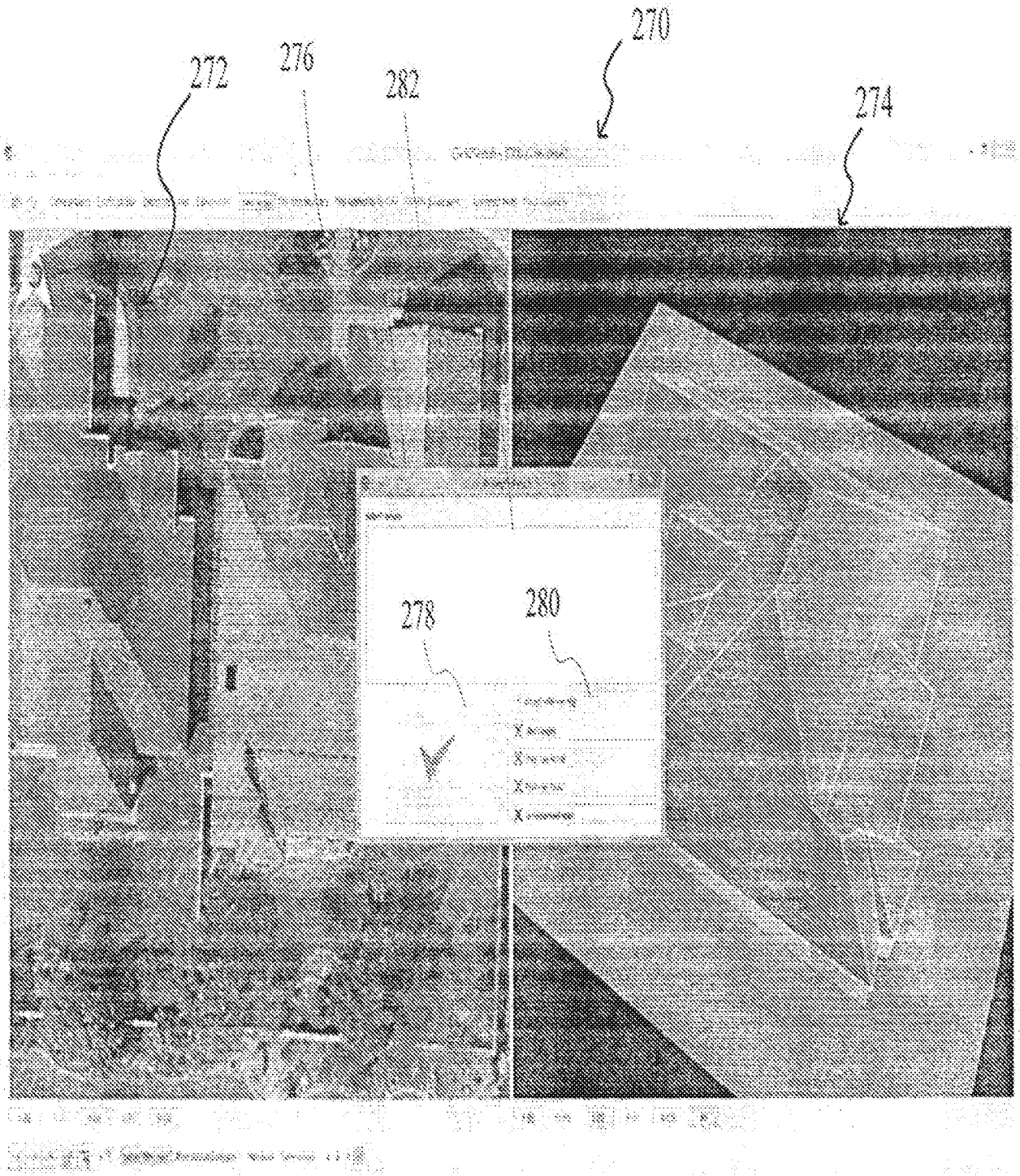


FIG. 11

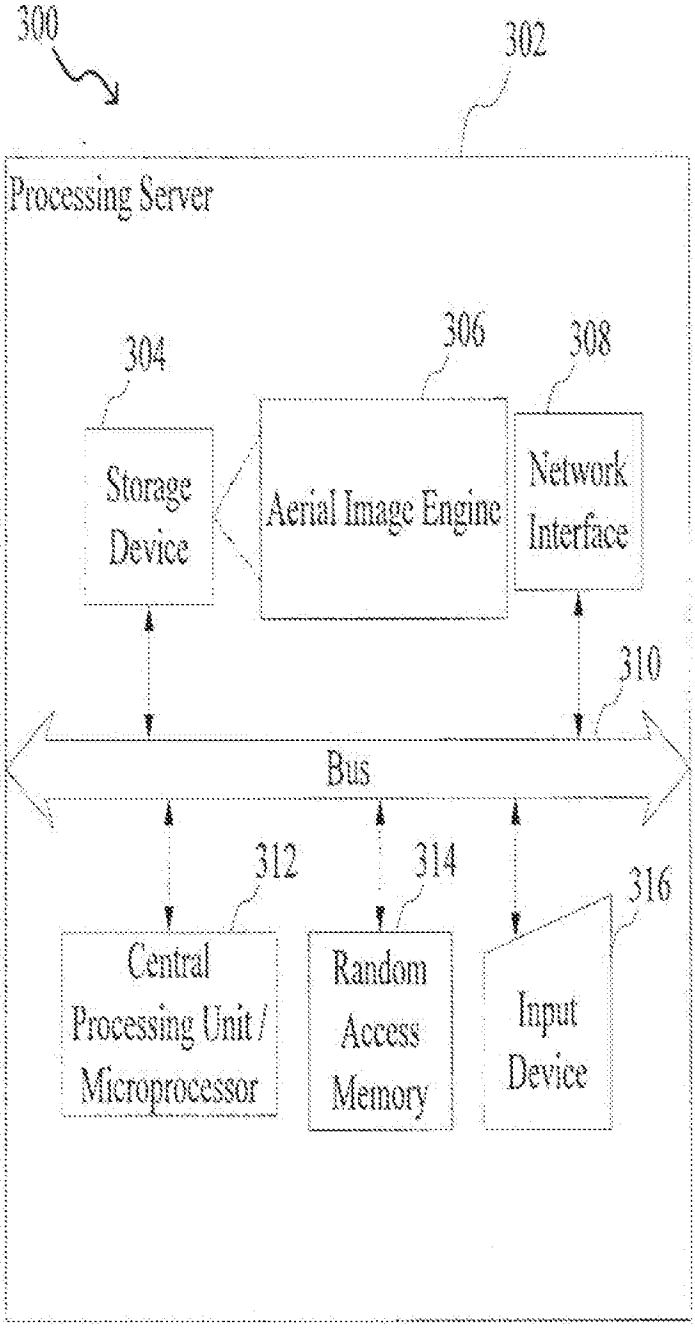

300
302
Processing Server
304
306
308
Storage Device
Aerial Image Engine
Network Interface
310
Bus
312
314
316
Central Processing Unit / Microprocessor
Random Access Memory
Input Device

FIG. 12

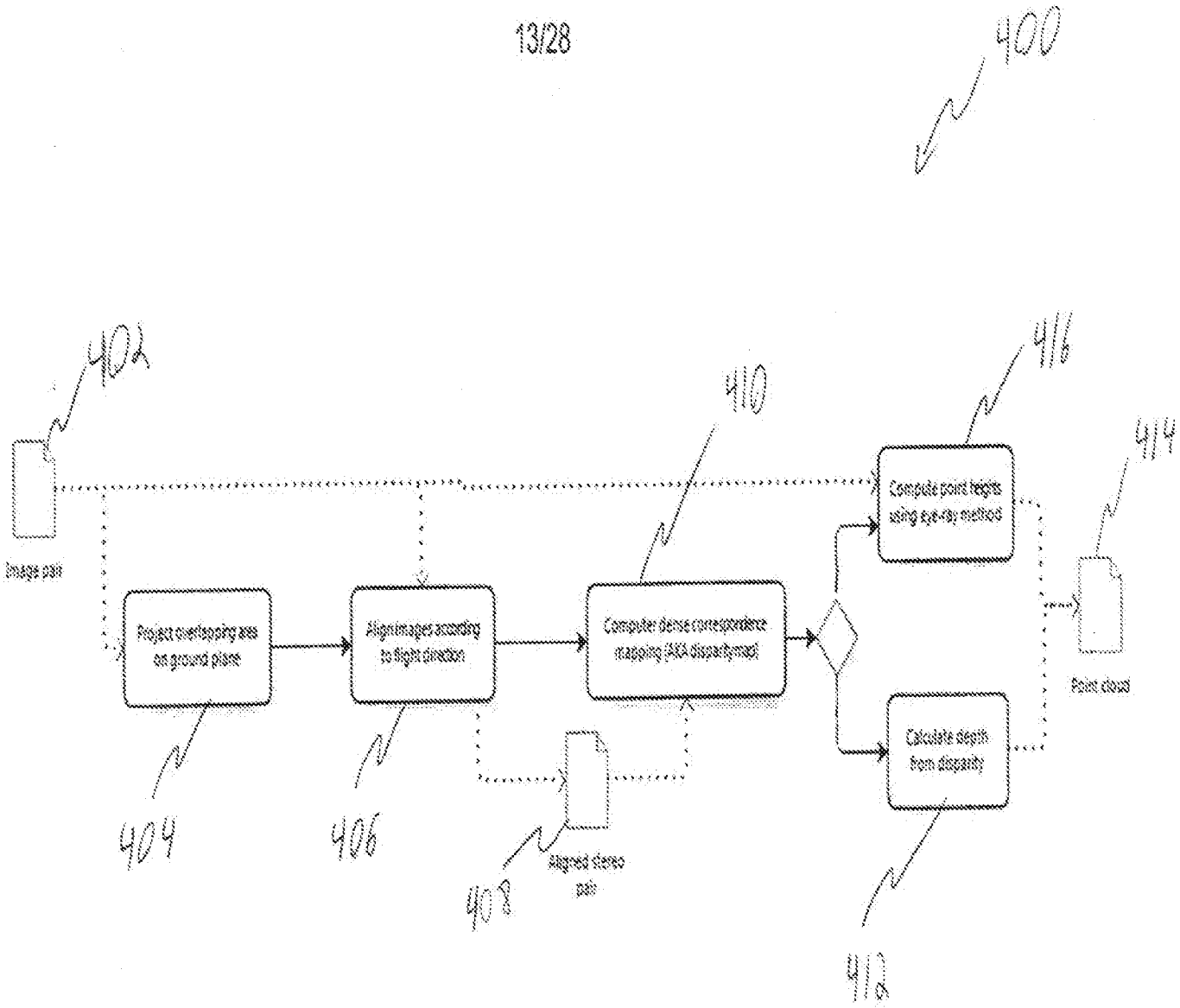

400
402
Image pair
Project overlapping area on ground plane
404
Align images according to flight direction
406
Aligned stereo pair
408
Compute dense correspondence mapping (XX disparity map)
410
Compute point heights using eye-sky method
416
Calculate depth from disparity
418
Point cloud
414

FIG. 13

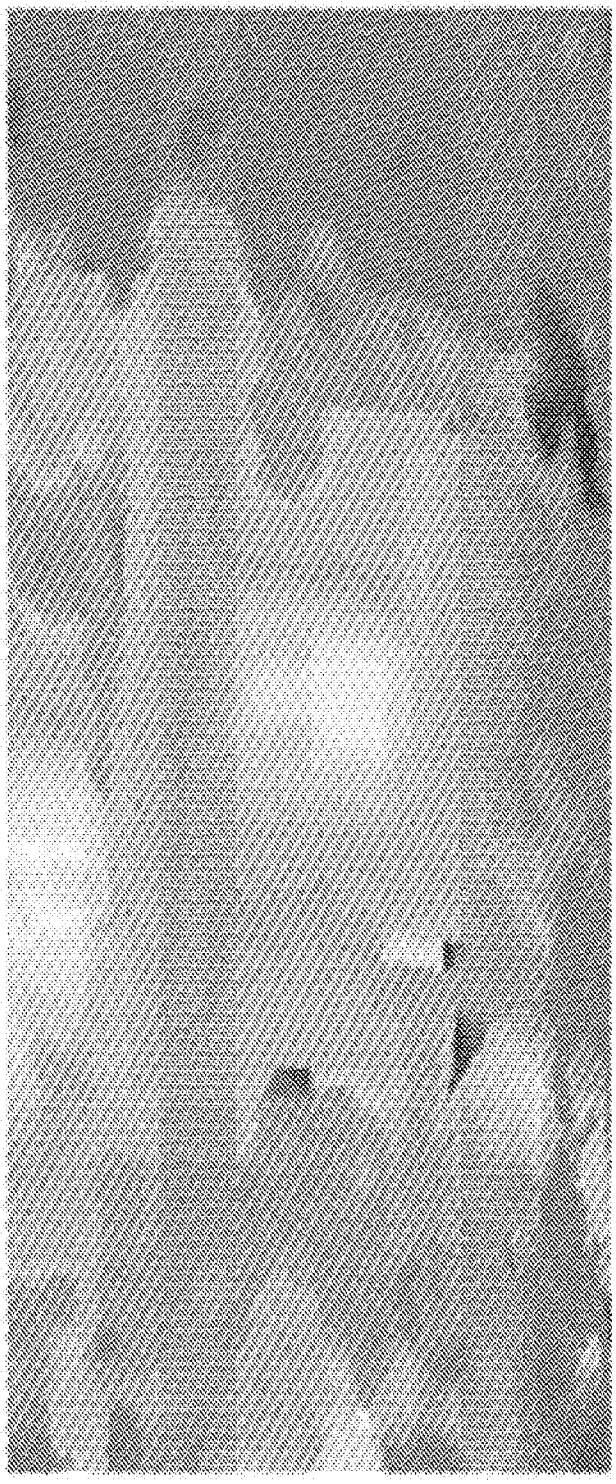


FIG. 14

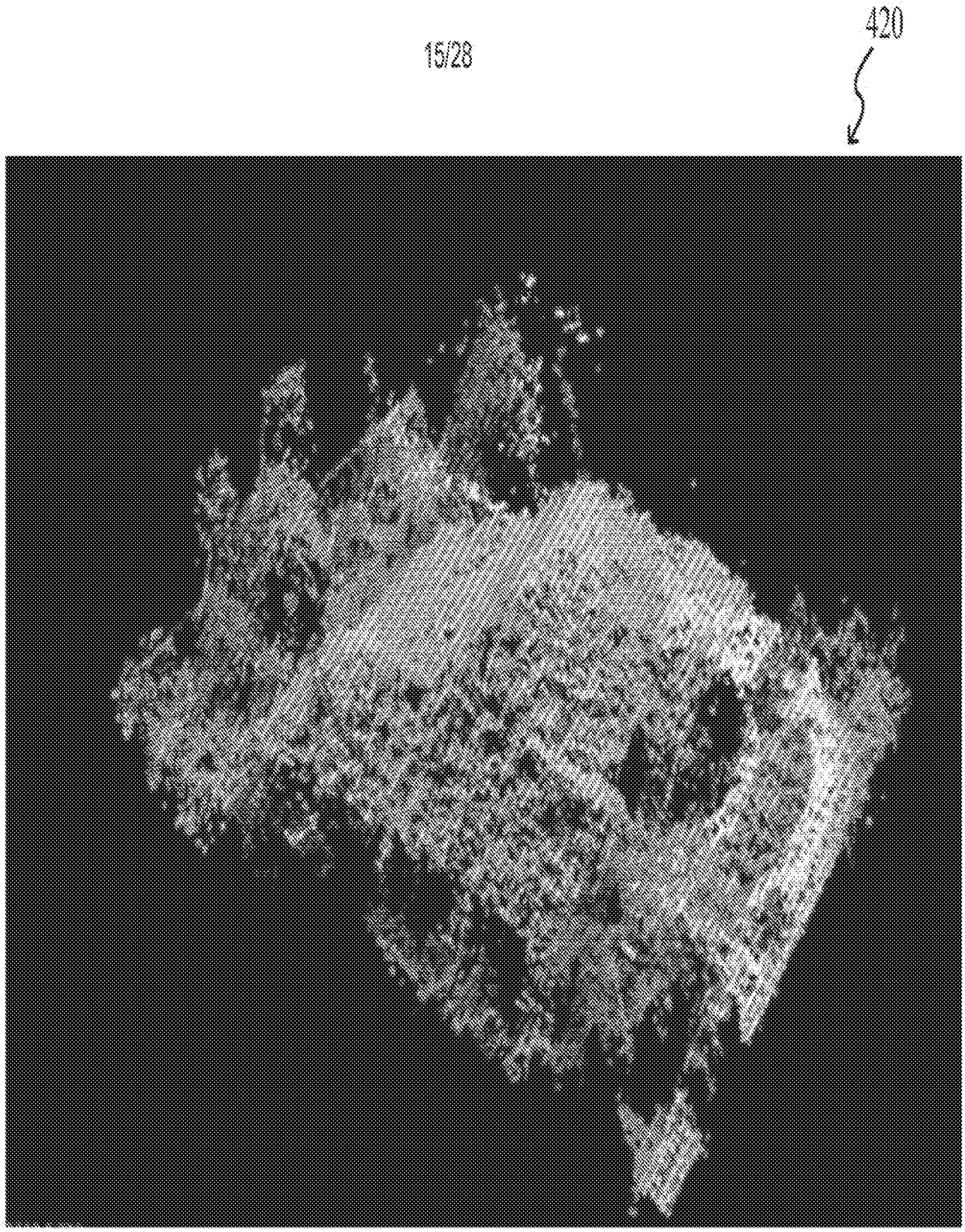


FIG. 15

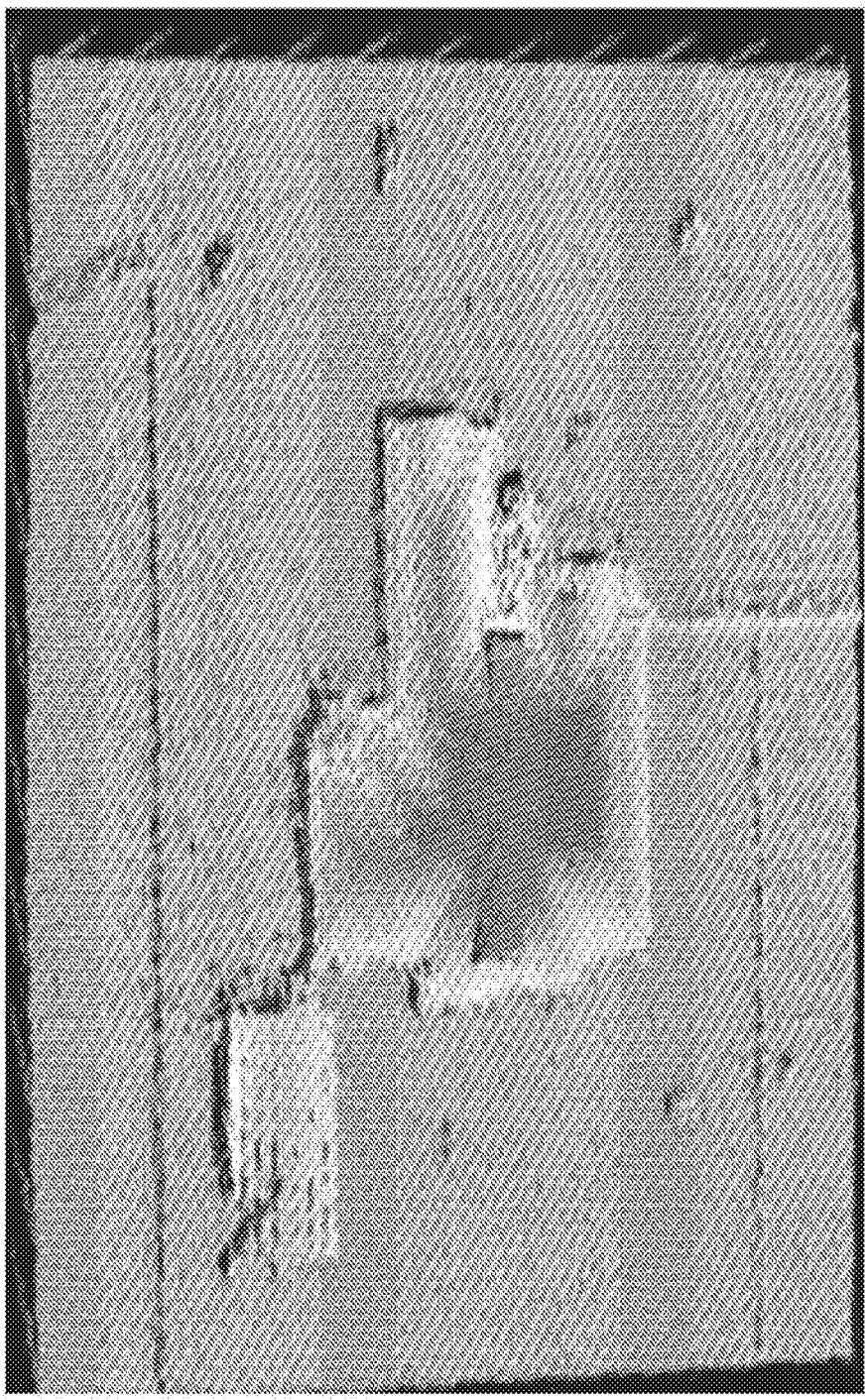


FIG. 16

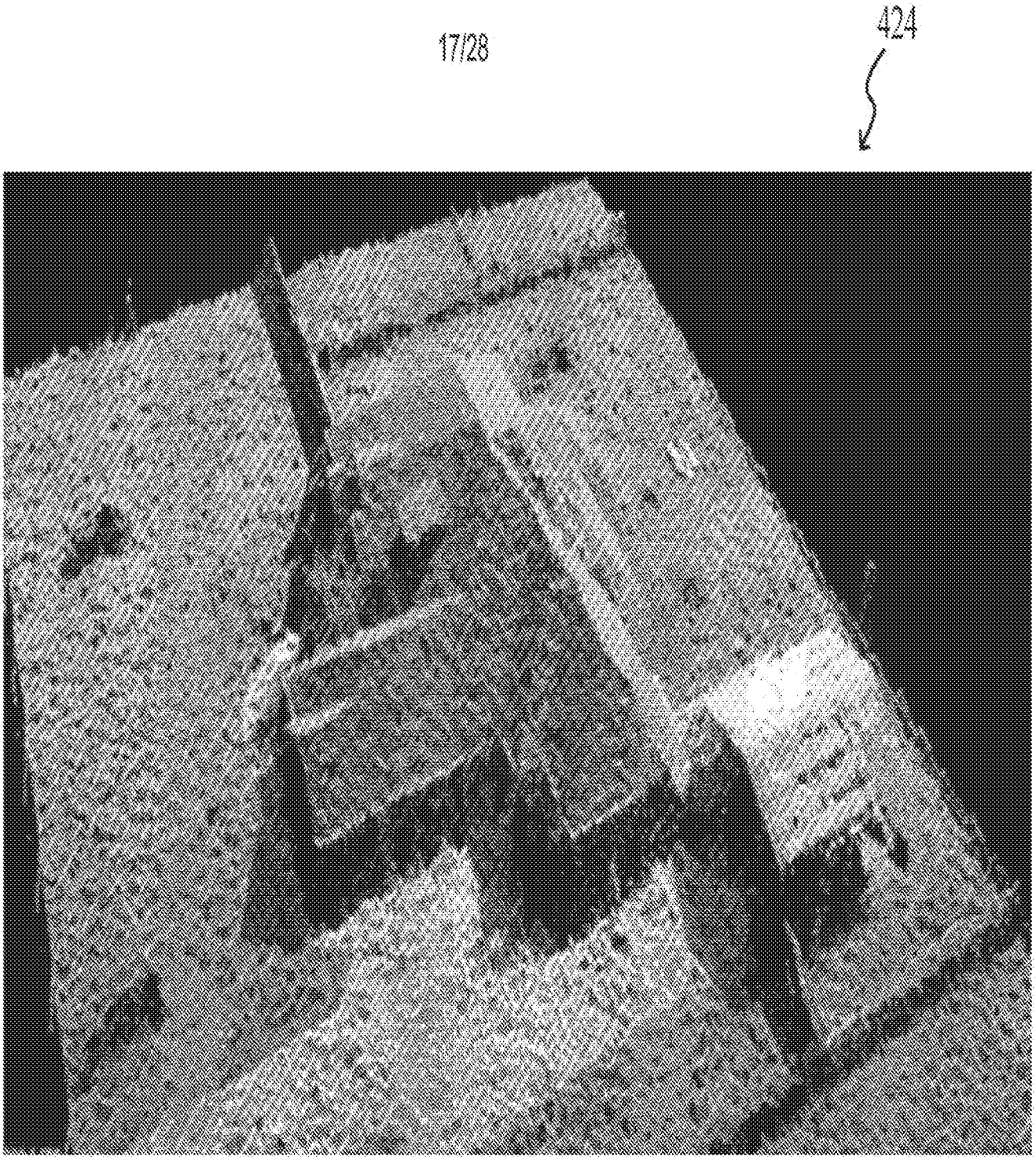


FIG. 17

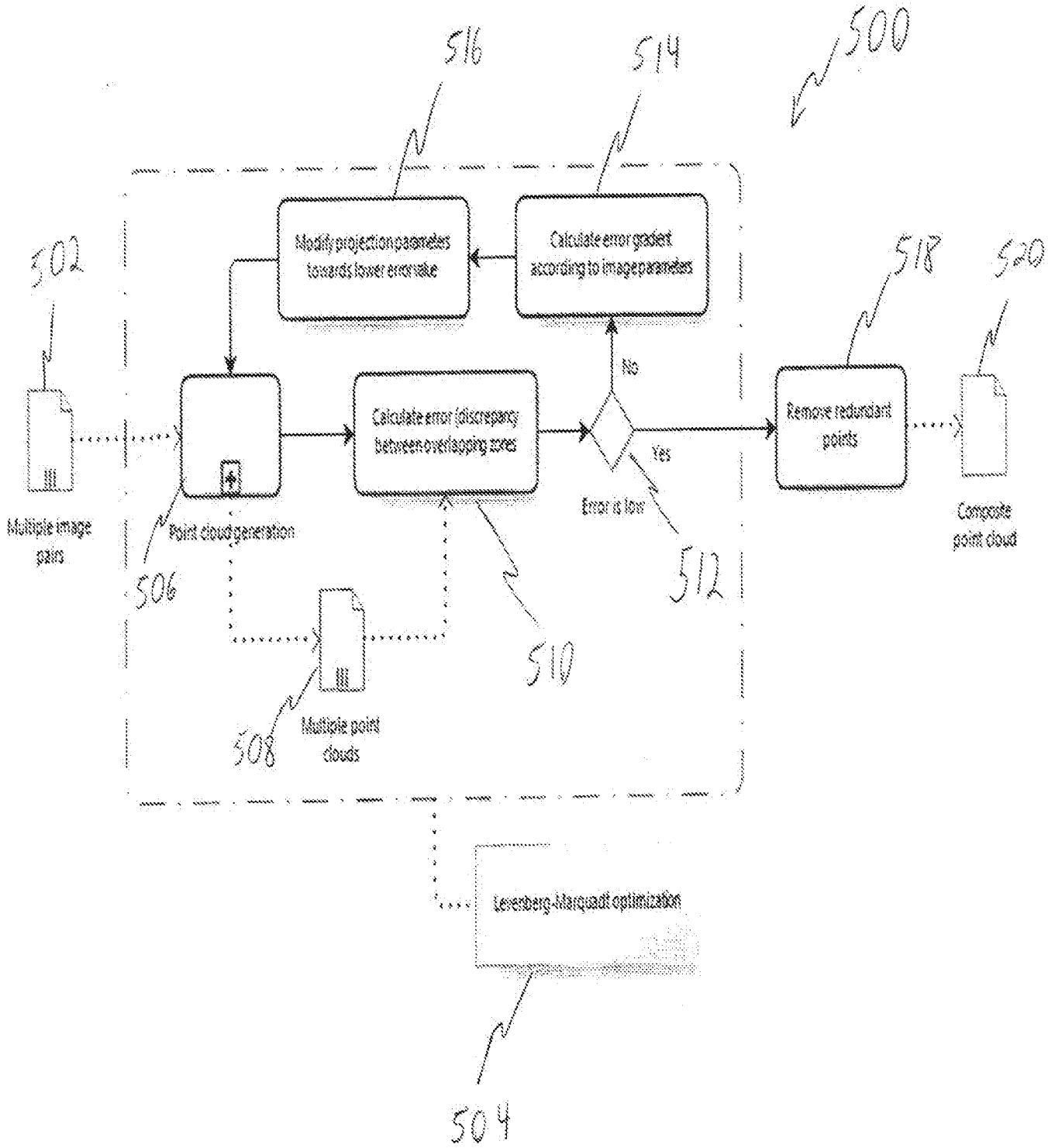

500
516
514
Modify projection parameters towards lower error rate
Calculate error gradient according to image parameters
502
518
520
No
Calculate error (discrepancy) between overlapping sizes
Remove redundant points
Yes
Error is low
Multiple image pairs
Point cloud generation
Composite point cloud
506
512
510
508
Multiple point clouds
Levenberg-Marquardt optimization
504

FIG. 18

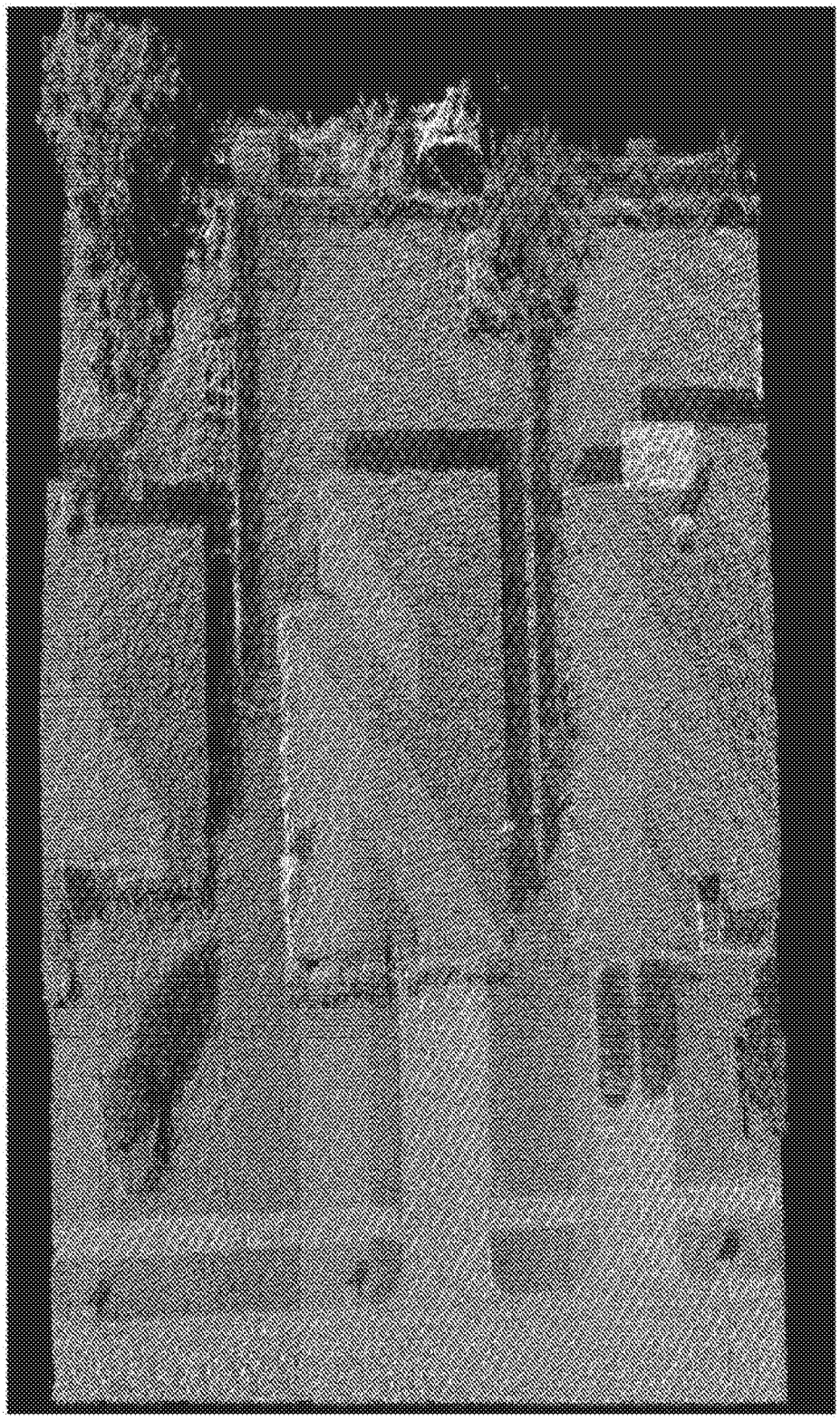


FIG. 19

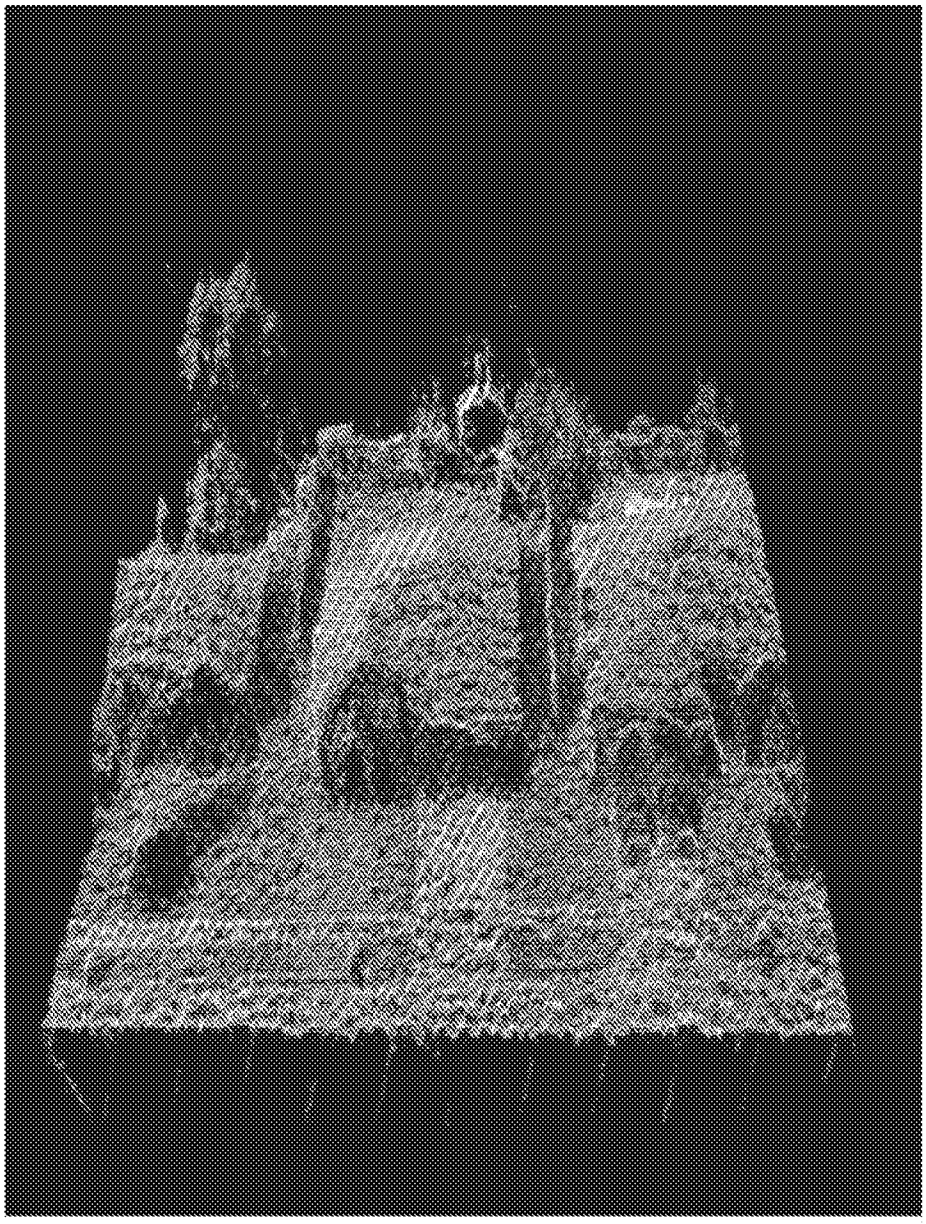


FIG. 20

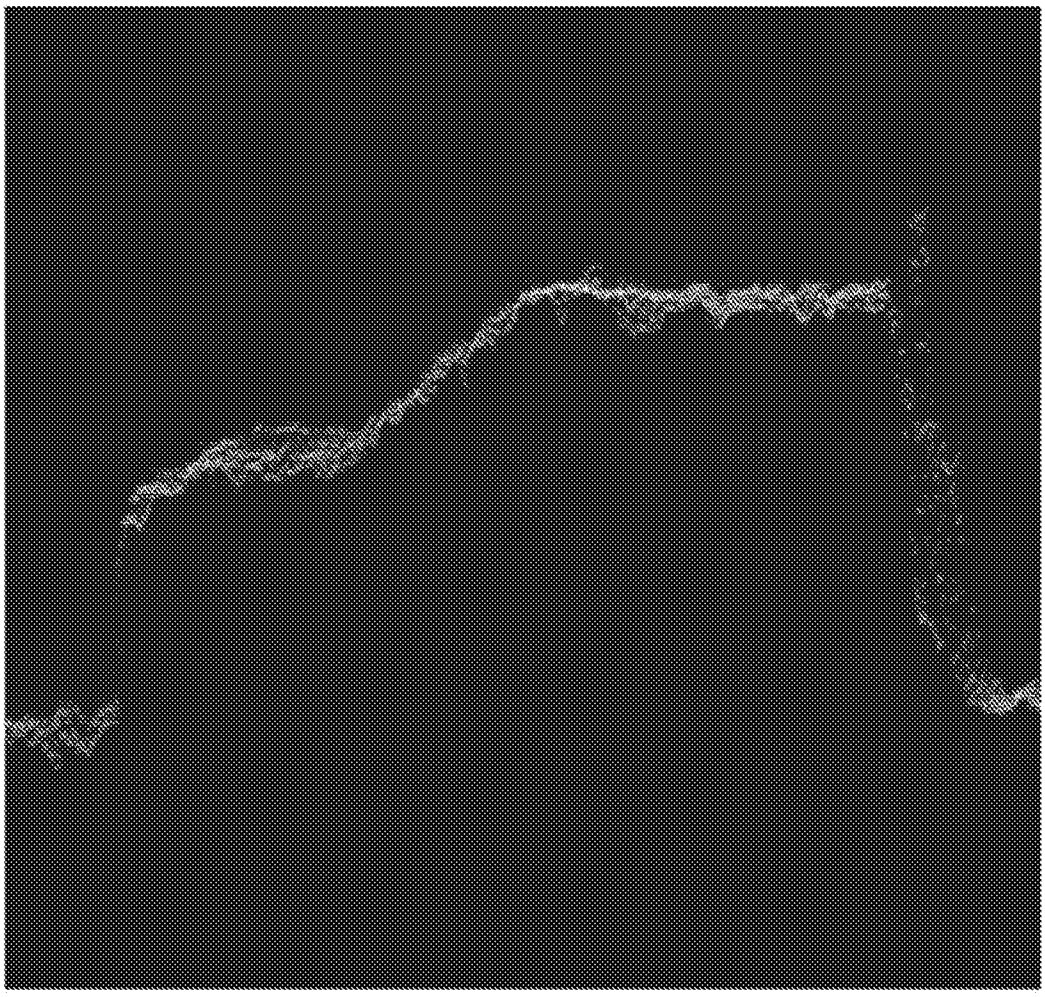


FIG. 21

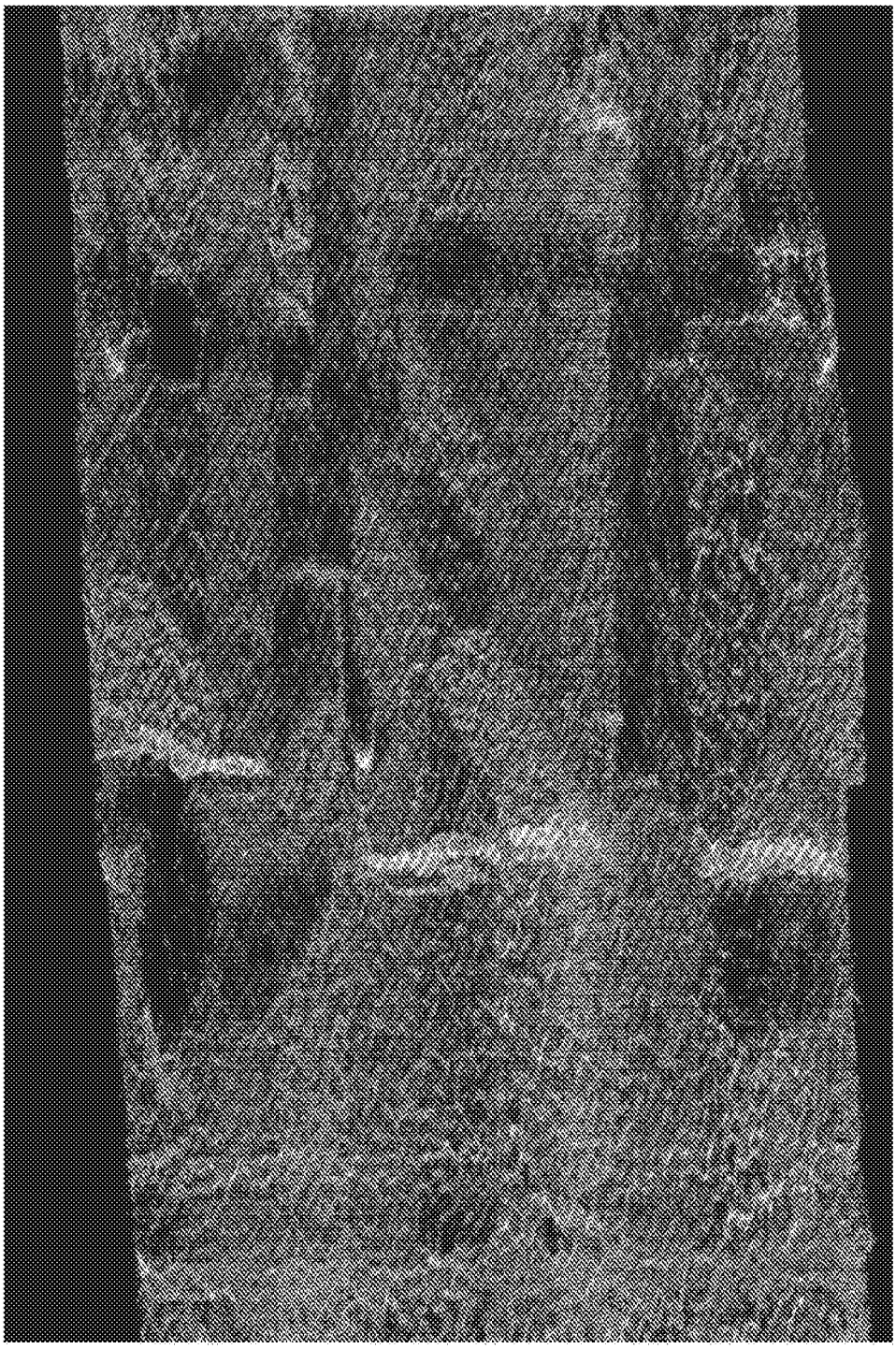


FIG. 22

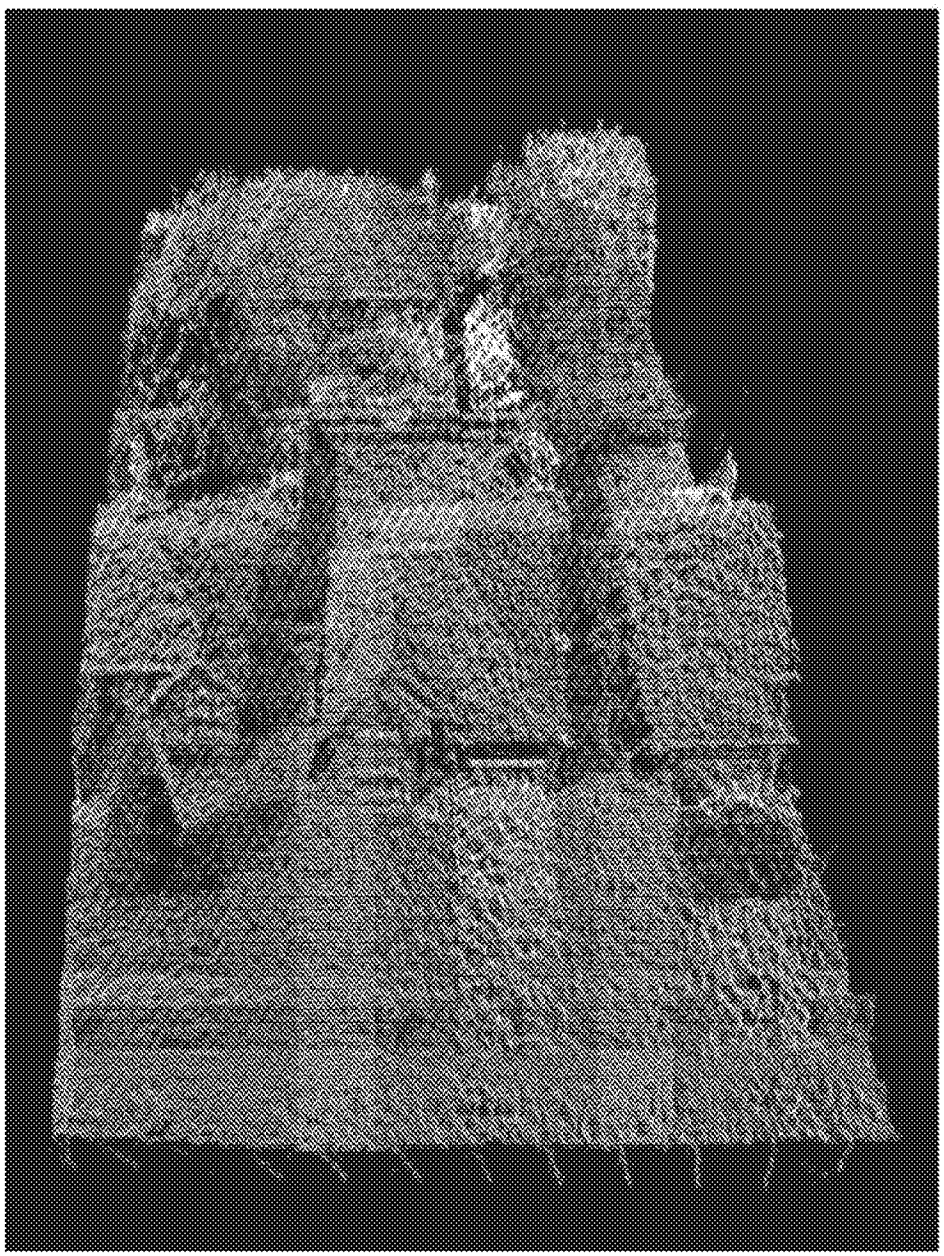


FIG. 23

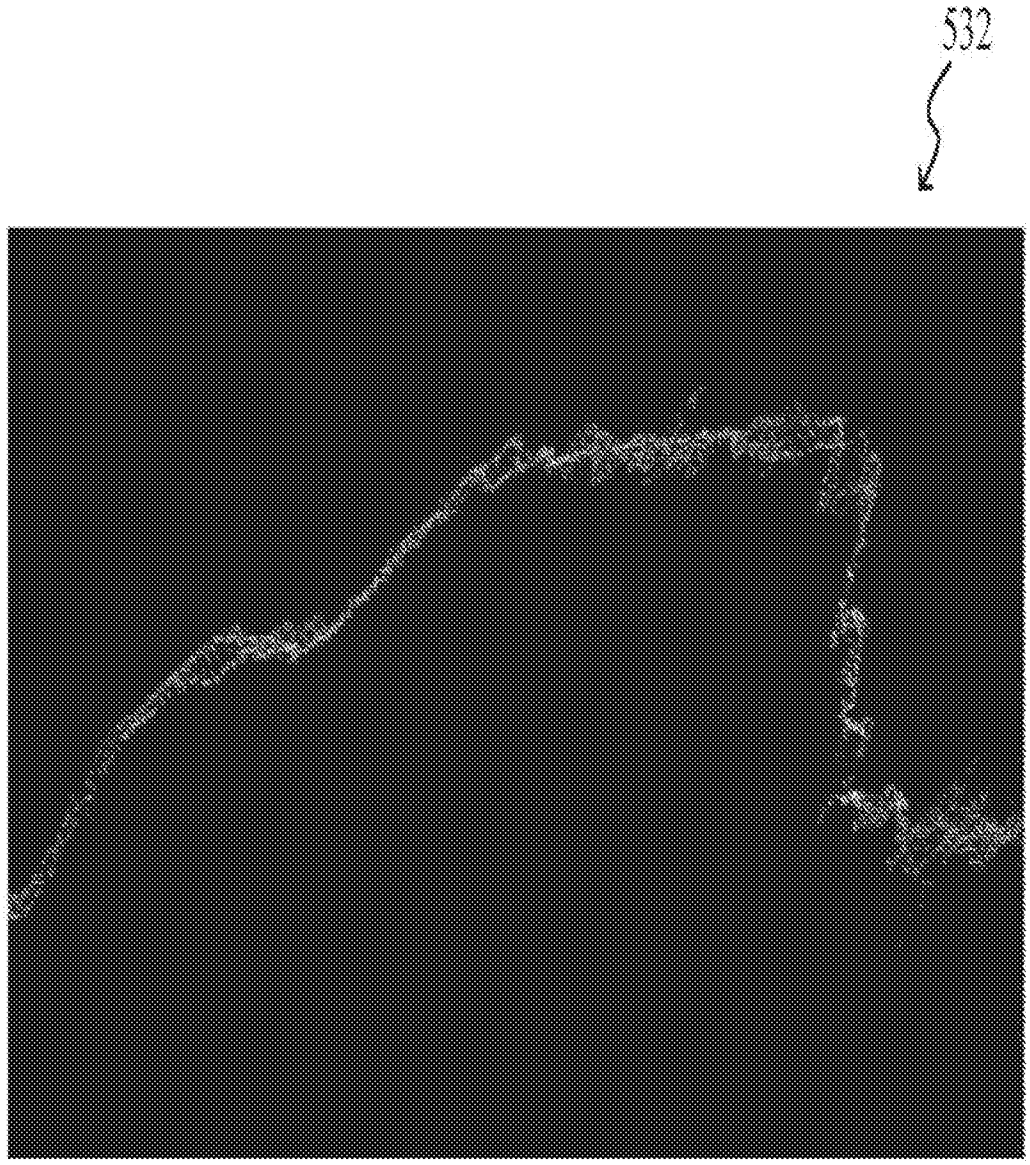


FIG. 24

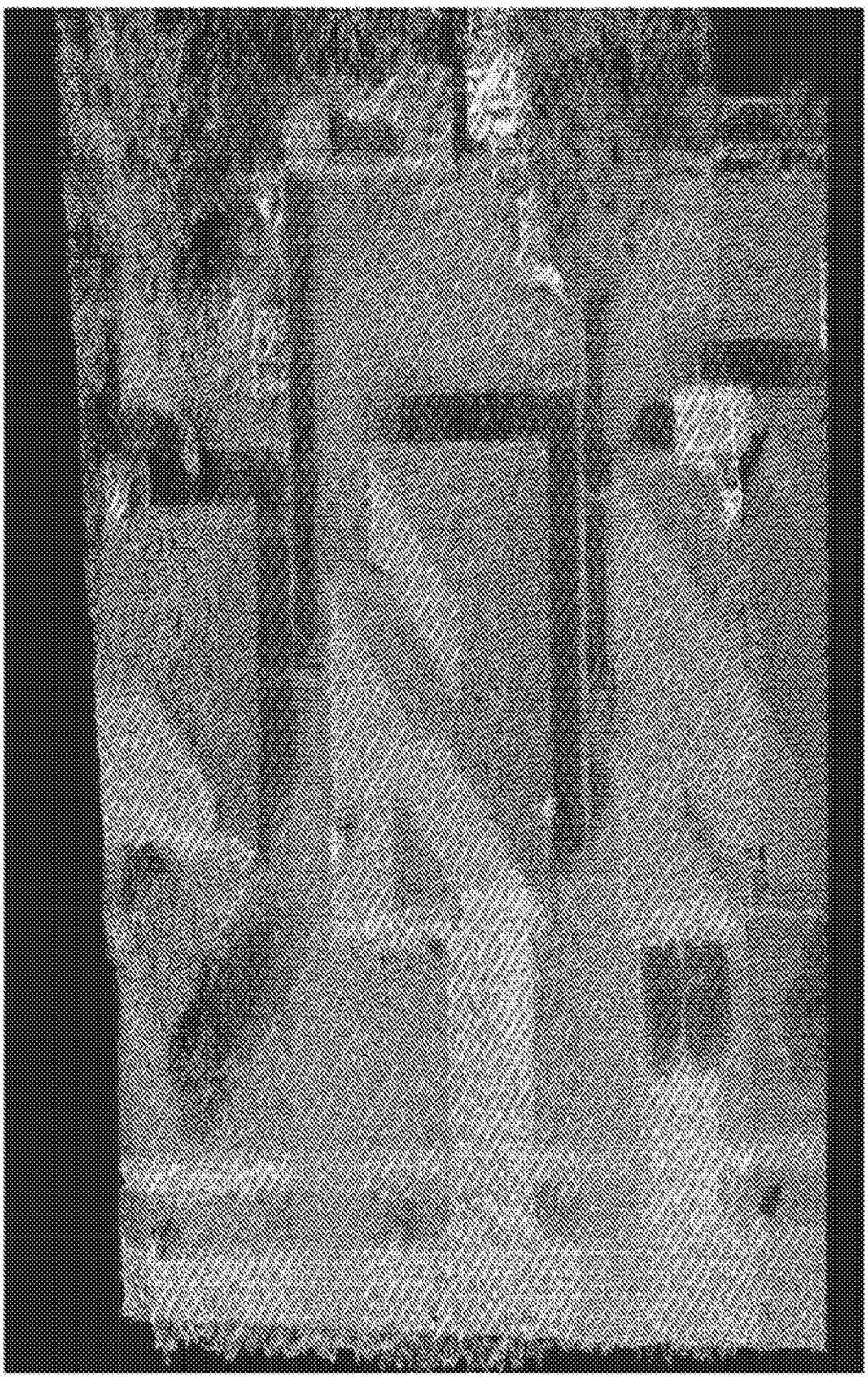


FIG. 25

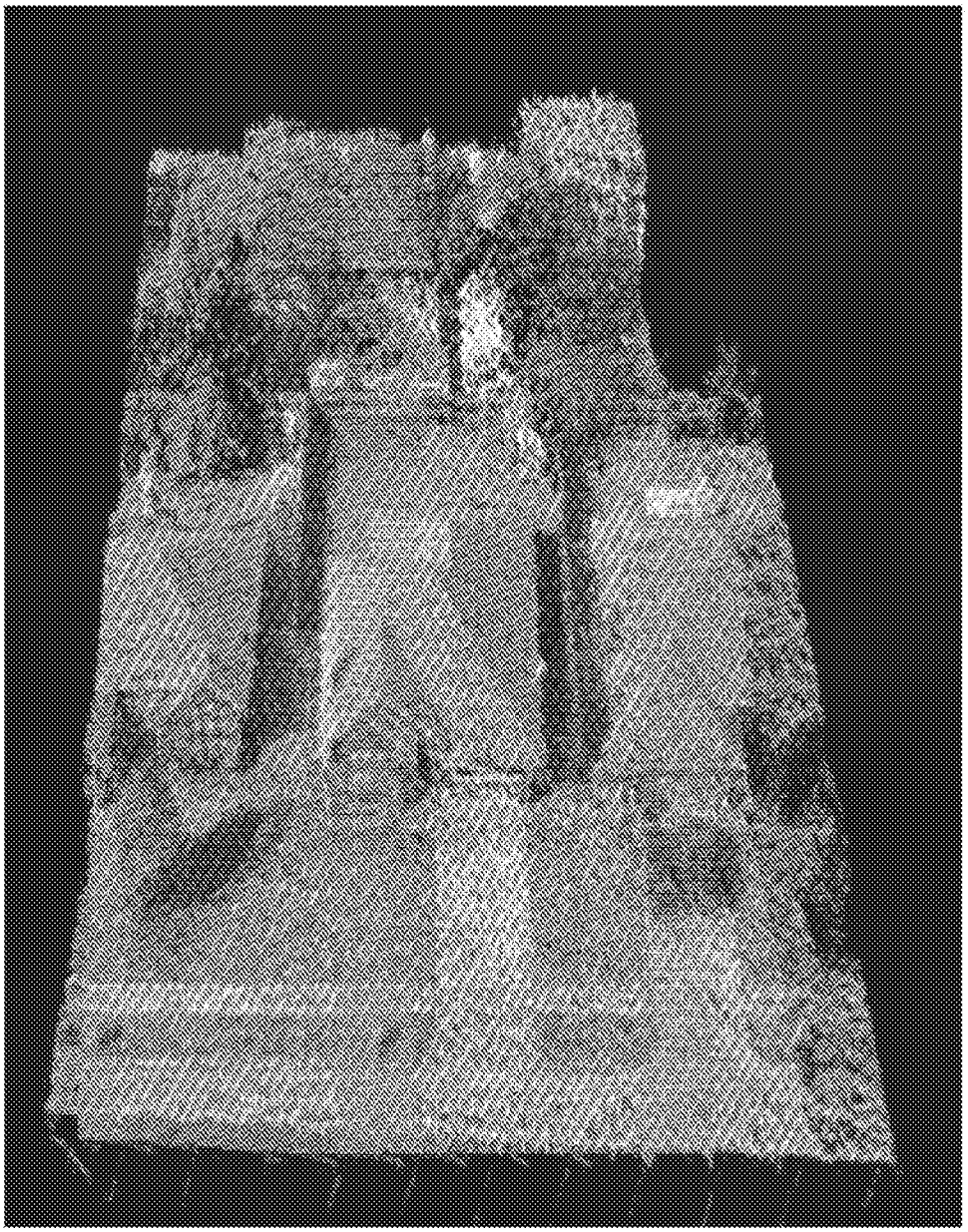


FIG. 26

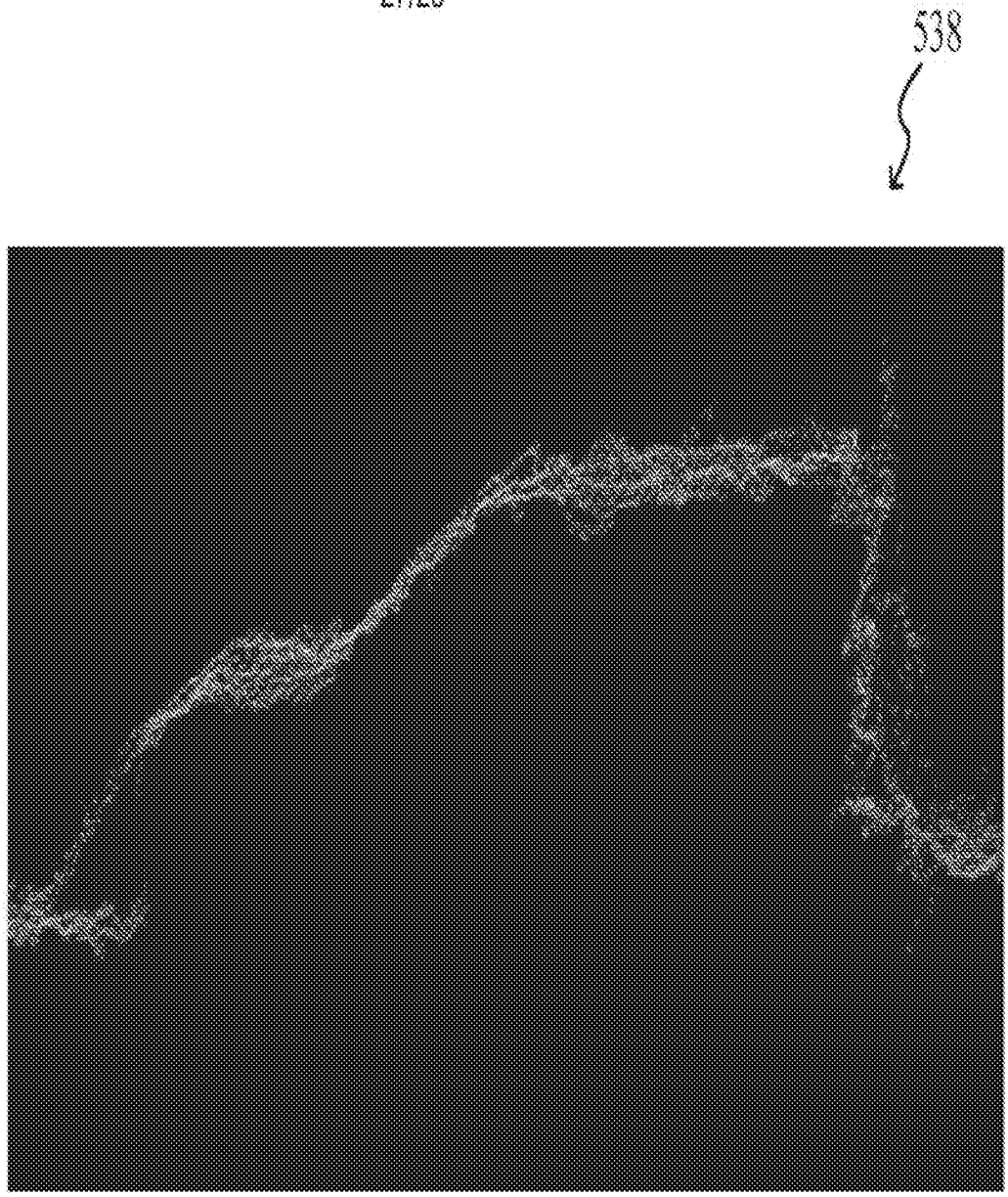


FIG. 27

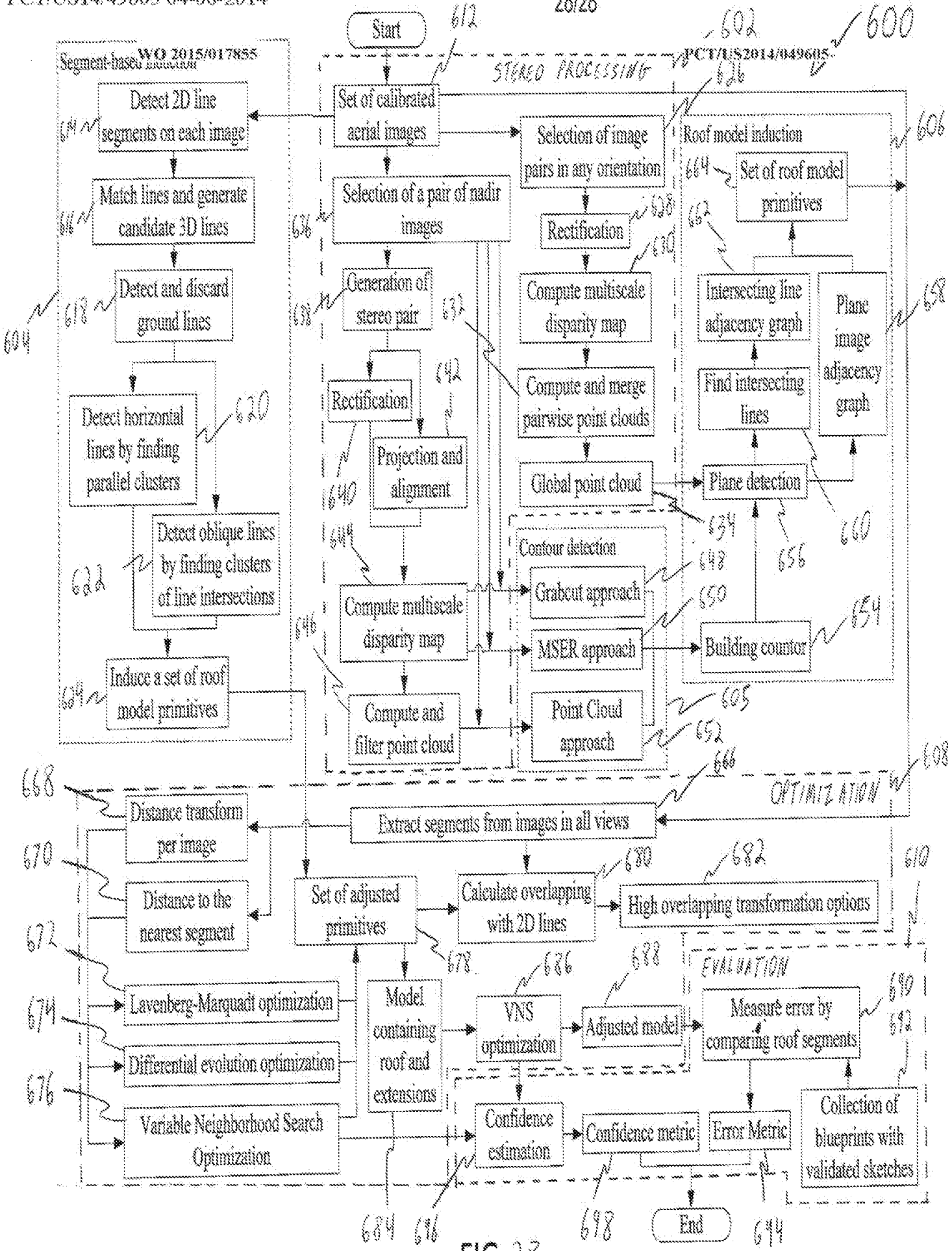

Start
612
602
600
WO 2015/017855
PCT/US2014/049605
STEREO PROCESSING
Segment-based location
Detect 2D line segments on each image
Set of calibrated aerial images
Selection of image pairs in any orientation
Roof model induction
606
Match lines and generate candidate 3D lines
Selection of a pair of nadir images
Rectification
Set of roof model primitives
Detect and discard ground lines
Generation of stereo pair
Compute multiscale disparity map
Intersecting line adjacency graph
Plane image adjacency graph
Detect horizontal lines by finding parallel clusters
Rectification
Compute and merge pairwise point clouds
Find intersecting lines
Projection and alignment
Global point cloud
Plane detection
Detect oblique lines by finding clusters of line intersections
Contour detection
Grabcut approach
Compute multiscale disparity map
MSER approach
Building counter
Induce a set of roof model primitives
Compute and filter point cloud
Point Cloud approach
OPTIMIZATION
Distance transform per image
Extract segments from images in all views
Distance to the nearest segment
Set of adjusted primitives
Calculate overlapping with 2D lines
High overlapping transformation options
EVALUATION
Lavenberg-Marquadt optimization
Model containing roof and extensions
VNS optimization
Adjusted model
Measure error by comparing roof segments
Differential evolution optimization
Variable Neighborhood Search Optimization
Confidence estimation
Confidence metric
Error Metric
Collection of blueprints with validated sketches
End

FIG. 28

INTERNATIONAL SEARCH REPORT

~~WO/US2014/049603 17.11.2014~~

International application No.
PCT/US2014/049603

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04N 13/00 (2014.01)
CPC - G06T 2207/10032 (2014.09)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - H04N 13/00, G06T 5/50, G06T 7/20, G02B 27/22, G06K 9/36, G06K 9/00, G06T 7/00, G01C 23/00 (2014.01)
USPC - 382/154, 382/285, 359/462, 348/42

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - G06T 2207/10032, G06T 5/50, G06T 7/2066, G06T 2207/10021, G06K 9/0063, G06T 7/2053, G01C 23/00, G06K 9/00664, G06T 2207/10012, G06T 7/0075, G06K 9/209, G06T 2207/30128, G06T 2207/30181 (2014.09)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Orbit, Google Patents, Google Scholar, Google.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	BAILLARD et al. "3-D reconstruction of urban scenes from aerial stereo imagery: a focusing strategy." Computer Vision and Image Understanding 76.3. December 1999 (12.1999). Retrieved from <http://www2.ic.uff.br/~aconci/papers/baillard.pdf>, entire document	1-24
Y	US 2010/0303340 A1 (ABRAHAM et al) 02 December 2010 (02.12.2010), entire document	1-24
Y	PRECIOZZI. "Dense urban elevation models from stereo images by an affine region merging approach." Master's Thesis, Universidad de la República, Montevideo, Uruguay (2006). Retrieved from <http://iie.fing.edu.uy/publicaciones/2006/Pre06/Pre06.pdf>, entire document	1-24
Y	US 2011/0222757 A1 (YEATMAN JR et al) 15 September 2011 (15.09.2011), entire document	7, 8, 15, 16, 23, 24
A	US 2006/0239537 A1 (SHRAGAI et al) 26 October 2006 (26.10.2006), entire document	1-24
A	US 2004/0264763 A1 (MAS et al) 30 December 2004 (30.12.2004), entire document	1-24
A	LU et al. "STEREO IMAGE MATCHING USING ROBUST ESTIMATION AND IMAGE ANALYSIS TECHNIQUES FOR DEM GENERATION." 2000 (2000). Retrieved from <http://www.isprs.org/proceedings/XXXIII/congress/part3/520_XXXIII-part3.pdf>, entire document	1-24
A	SYED et al. "Semi-automatic 3D building model generation from lidar and high resolution imagery." Proceedings of SSC Spatial Intelligence September 2005 (09.2005). Retrieved from <http://www.ecognition.com/sites/default/files/265_0187.pdf>, entire document	1-24

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 October 2014	17 NOV 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2015 (19.03.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/038358 A1

(51) International Patent Classification:
G06F 19/00 (2011.01) *G06T 7/00* (2006.01)

(21) International Application Number:
PCT/US2014/053387

(22) International Filing Date:
29 August 2014 (29.08.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/024,418 11 September 2013 (11.09.2013) US

(71) Applicant: **DIGITALGLOBE, INC.** [US/US]; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(72) Inventors: **MARCHISIO, Giovanni B.**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US). **TUSK, Carsten**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US). **KOPERSKI, Krzysztof**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US). **TABB, Mark D.**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US). **SHAFER, Jeffrey D.**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(74) Agent: **CROUCH, Robert, G.**; Marsh Fischmann & Breyfogle LLP, Robert G. Crouch, 8055 E. Tufts Ave., Suite 450, Denver, Colorado 80237 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

[Continued on next page]

(54) Title: CLASSIFICATION OF LAND BASED ON ANALYSIS OF REMOTELY-SENSED EARTH IMAGES

(57) Abstract: Land classification based on analysis of image data. Feature extraction techniques may be used to generate a feature stack corresponding to the image data to be classified. A user may identify training data from the image data from which a classification model may be generated using one or more machine learning techniques applied to one or more features of the image. In this regard, the classification module may in turn be used to classify pixels from the image data other than the training data. Additionally, quantifiable metrics regarding the accuracy and/or precision of the models may be provided for model evaluation and/or comparison. Additionally, the generation of models may be performed in a distributed system such that model creation and/or application may be distributed in a multi-user environment for collaborative and/or iterative approaches.

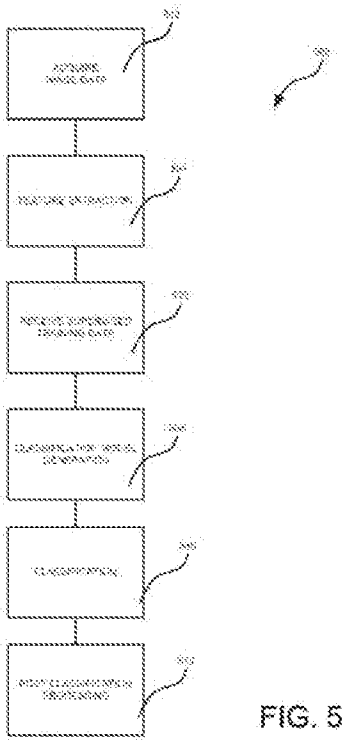


FIG. 5

WO 2015/038358 A1

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

CLASSIFICATION OF LAND BASED ON ANALYSIS OF REMOTELY-SENSED EARTH IMAGES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority of U.S. Patent App. No. 14/024,418, entitled " CLASSIFICATION OF LAND BASED ON ANALYSIS OF REMOTELY-SENSED EARTH IMAGES," and filed on September 11, 2013, the entire contents of which are incorporated herein by reference as if set forth in full.

BACKGROUND

The use of geospatial imagery has increased in recent years and represents a significant tool that may be utilized in a number of contexts. As such, high quality geospatial imagery has become increasingly valuable. For example, a variety of different entities (e.g., individuals, governments, corporations, or others) may utilize geospatial imagery (e.g., satellite imagery) for a multitude of applications in a variety of contexts.

With increasingly capable satellites being commissioned and launched, very high resolution (VHR) remotely-sensed multispectral Earth imagery has become increasingly available and useable. For example, as the number of satellite image acquisition systems in operation grows, acquisition ability and flexibility improves. In an example, DigitalGlobe, Inc. of Longmont, CO currently operates a number of satellites including, IKONOS, GeoEye-1, QuickBird, WorldView 1, and WorldView 2, with an anticipated launch of WorldView 3. Accordingly, around the clock global coverage may be achieved though the satellite constellation currently in operation. As such, the DigitalGlobe constellation of satellites can image the entire Earth's landmass every 75 days and may capture over six times the Earth's landmass every year with a capacity to collect at least 2.6 million square kilometers of imagery a day. With selective tasking, DigitalGlobe's satellite constellation may quickly and continually collect imagery from targeted locations to provide near real time feedback on global events or the like.

Furthermore, the resolution of image acquisition satellites also continues to increase. For instance, currently operated satellites may have a maximum spatial resolution of 50 cm (wherein each pixel in the resulting images acquired corresponds with the distance measure of the spatial resolution). Additionally, planned satellite launches may provide even greater resolution capabilities with spatial resolutions as high as about 30 cm.

In this light, the amount and quality of VHR remotely-sensed multispectral Earth imagery continues to increase as does the amount and types of image data collected. Accordingly, the nature of the VHR remotely-sensed multispectral Earth imagery may facilitate uses beyond simply providing pixels as image data. For instance, higher level data processing may be applied to the images to, for example, identify objects, identify textures, or otherwise extract useful data from the raw image data. One such application has been in land use identification and land classification, whereby remotely-sensed images are analyzed to categorize pixels of an image into a category of land use or land class. However, as the amount of image data that is available grows and the nature of the image data acquired changes and is improved, advanced image data processing and image analytics are needed to keep pace with the advances in image acquisition technology to facilitate new techniques applied to acquired images to expand the number of applications for which such technology may be used.

SUMMARY

In this regard, the present disclosure generally relates to land classification using very high resolution (VHR) remotely-sensed multispectral Earth images. VHR imagery may comprise remotely sensed imagery with a resolution not less that about 20 m, not less than about 10 m, not less than about 5 m, or even not less than about 1 m. In this regard and in some embodiments, VHR imagery may make a resolution as high as 0.50 m or greater. Specifically, the present disclosure provides approaches that may leverage the capability of increasingly powerful and versatile satellite constellations and other image acquisition techniques to provide for improved land classification in an increasing number of contexts. In this regard, at least some of the aspects disclosed herein may be particularly suited for application to images acquired using multispectral images including, for example, eight or more multispectral bands that are leveraged to provide robust land classification.

Additionally, the present disclosure may provide a multi-user environment that allows for collaboration to develop land classification information and/or classification models for use in determining land classifications. As one goal for land classification is to generate a global land classification layer, the present disclosure may provide architecture that facilitates collaborative, incremental generation of classification models to provide worldwide land classification in a manageable, efficient manner using a modular, scalable, and/or collaborative architecture. As such, the present disclosure, in at least some embodiments, includes a distributed multiuser

environment whereby users may collaborate and/or iteratively generate and/or apply classification models.

Further still, the present disclosure includes a land classification approach that may include feature extraction from images to generate a feature stack having one or more data layers related to image data that may facilitate improved model generation and pixel classification for the image data. For instance, the present disclosure includes feature generation modules for creation of morphological, textural, and/or spectral feature layers for a given image that may be utilized in model generation and/or pixel classification to provide for improved land use classification.

Additionally, the present disclosure provides metrics for evaluation of land classification. For instance, the present disclosure provides techniques that may be used to judge the ability of different classification models to accurately classify image data into land use classifications. As such, it may be appreciated that for different geographies and/or different time periods, the specific techniques used in classification of image data may not be constant. That is, development of a one-size-fits-all type model is impractical and less accurate than specific models for specific geographies and/or specific time periods that may be generated. As such, determination of evaluation metrics of modules may allow for creation of robust classification modules specifically tailored for an application. In this regard, the distributed, multi-user collaborative environment facilitated by the architecture of land classification systems herein may allow for collaborative development of tailored classification models based on a specific context to be addressed by the model.

A first aspect includes a land classification system operable for analysis of very high resolution (VHR) remotely-sensed multispectral Earth imagery. The system includes an image store containing image data corresponding to VHR remotely-sensed multispectral Earth images. The system also includes at least one feature extraction module in operative communication with the image store. The feature extraction module is operable to produce feature data regarding at least a portion of image data. In turn, the system includes a feature stack that includes the image data and the feature data. The system also includes a client interface operable to receive training data regarding at least one or more pixels of the image data from a user regarding at least one class to which the one or more pixels belong. The system also includes a classification compute module operable to generate a classification model at least in part based on a portion of the feature stack corresponding to the one or more pixels and the training data. The classification model relates to classification of pixels of image data into one or more classes.

A number of feature refinements and additional features are applicable to the first aspect. These feature refinements and additional features may be used individually or in any combination. As such, each of the following features that will be discussed may be, but are not required to be, used with any other feature or combination of features of the first aspect.

For instance, in an embodiment the classification compute module may be operable to apply the classification model to the image data to classify pixels of the image data into the one or more classes. Accordingly, the classification compute module may classify pixels of the image data based on an analysis of the feature stack in relation to the classification model.

In an embodiment, the system may include a post classification analysis module that is operable to reclassify at least one pixel of the image data based on at least one post classification rule. For instance, the post classification rule comprises at least one of a minimum mapping unit rule, a surrounded by rule, a topological relation rule, a majority analysis rule, or a smoothing rule.

It may be appreciated that the land classification system may be particularly adept at classification of image data that is remotely-sensed, multispectral Earth image data (e.g., satellite image data). In this regard, the VHR remotely-sensed multispectral Earth image data may include spectral band data corresponding to at least 8 multispectral bands. The multispectral bands may collectively range from at least about 300 nanometers in wavelength to at least about 1100 nanometers in wavelength. In another embodiment, the multispectral bands may collectively range from at least about 300 nanometers in wavelength to at least about 2400 nanometers in wavelength. Furthermore, the multispectral bands may include a plurality of short wave infrared (SWIR) bands. The plurality of SWIR bands range from at least about 1100 nanometers in wavelength to at least about 2400 nanometers in wavelength.

Additionally, the feature stack may include a plurality of data layers, wherein each data layer includes different feature data for each pixel of the image data. The feature stack includes at least one spectral feature layer, at least one morphological feature layer, or at least one textural feature layer. The at least one spectral feature layer may include data values for each pixel of the image data based on spectral band data collected by a remote image acquisition platform. For example, the remote image acquisition platform may include a satellite imaging system. In any regard, at least one spectral feature layer may include a relative measure between at least two spectral band data layers.

Furthermore, the at least one morphological feature layer may include, for each given pixel of the image data, information based on the arrangement of adjacent pixels relative to the given pixel. The textural feature layer may include information for a given pixel based on a spatial distribution of tonal variations within one or more spectral band data layers relative to a given pixel.

In an embodiment, the client interface may include an internet browser-based client operable to provide a user interface for receiving the training data from a user. In this regard, the client interface may include a plurality of distributed client interfaces operable to receive training data from a plurality of users. As such, the classification model is at least partially based on training data received from a plurality of users.

In an embodiment, the system may include a plurality of feature extraction modules each operable to generate a feature data layer of the feature stack. One or more of the feature extraction modules may be selectively applicable to image data by a user to generate feature data layers. For example, a number of feature extraction modules may be, but are not required to be used in the land classification system. These feature extraction modules may include a fractal dimension feature extraction module, a total variational feature extraction module, a rotationally invariant histogram of gradients (HOG) feature extraction module, a Gabor wavelet feature extraction module, a clustering histogram feature extraction module, or any other appropriate feature extraction module.

Accordingly, the fractal dimension feature extraction module may be operative to generate feature data by identifying an image of a predetermined size of pixels, dividing the image into a plurality of abutting square windows having a side dimension less than the predetermined size of the image, and constructing for each plurality of abutting square windows a column having a height as represented in increments of the side dimension of the abutting window. The height of the column may be determined such that the ratio of the predetermined size of the image to the side dimension of the abutting windows equals the ratio of a maximum pixel value for a given multispectral band of the image to the height. The squared side dimension and the height may represent a plurality of boxes whose volume is defined by increments of the side dimension. In this regard, the fractal dimension feature extraction module may determine, for each abutting window a box from the plurality of boxes in which a maximum pixel value and a minimum pixel value occur and may calculate, for each of the plurality of abutting windows, the number of boxes separating the box in which the maximum pixel value is disposed and the box in which the minimum pixel value is disposed. The module may also sum the number of

boxes separating maximum and minimum pixel levels over all of the plurality of abutting windows to generate a summed value. In turn, the module may repeat each of the constructing, determining, calculating, and summing operations for all possible values of the side dimension of the plurality of abutting windows such that each side dimension is less than or equal to half of the predetermined size of the image and at least greater than three pixels in length. Additionally, the module may plot each summed value generated in the repeating step to determine a slope of the summed values, wherein the slope comprises the fractal dimension of the image.

In an embodiment, the image of the predetermined size is a subset portion of a larger image. In turn, the fractal dimension may be calculated for each subset portion of the larger image is used to generate a fractal map comprising an aggregated mapping of the fractal values of each subset portion over the larger image. As such, a fractal map may be generated for each spectral band of a multispectral image. The fractal dimension may be at least partially dependent upon the spectral band, wherein each fractal map is independently generated of other fractal maps of the other spectral bands of the plurality of spectral bands.

In another embodiment, at least one of the feature extraction modules may include a total variational (TV) feature extraction module. The TV feature extraction module may include a multispectral filtering module using a plurality of spectral bands of an image to generate a filtered image corresponding to the image. The number of spectral bands used in the filtering operation may be at least 8 multispectral bands. The filtering operation may include minimizing a function including at least a first term representing the difference between the original image and the filtered image and at least a second term representing the spatial homogeneity of the filtered image. In this regard, the first term and the second term may include a multivalued vector for each pixel representative of the image or filtered image , respectively, wherein the multivalued vector values include a component for each row, column, and multispectral band in the image. At least one of the first term or the second term include an L1 norm. Additionally, the second term is weighted by a smoothing factor. The minimization of the function may be solved using a Split Bregman technique. The Split Bregman technique may include Gauss-Seidel updates. In this regard, the minimization is solved by execution on a graphics processing unit (GPU).

In an embodiment, at least one of the feature extraction modules may include a rotational invariant histogram of gradients (HOG) feature extraction module. The rotational invariant HOG feature extraction module may be operative to generate feature data by establishing a plurality of angular bins for an image in a gradient

domain, histogramming gradients from the image with respect to the plurality of angular bins, and selecting based on the histogramming the angular bin with the largest histogram value to define a primary direction. Additionally, the module may set a plurality of pooling windows with relative offsets from a pixel of interest and define a gradient orientation window encompassing the plurality of pooling windows. The module may configure the gradient orientation window to align the configuration of the gradient orientation window relative to the primary direction. Also, the module may shift the positions of the pooling windows relative to the gradient orientation window based on the configuring. The module may also rotate the internal orientation of the pooling windows based on the configuring.

In an embodiment, the shifting may include shifting the pooling windows relative to the gradient orientation window by a number of bins determined in the selecting step. The rotating may also include shifting the internal orientation of the pooling windows by a number of bins determined by the rotating step. Accordingly, a HOG is calculated relative to each of the shifted and rotated pooling windows. The calculating may include using an integral image to calculate the HOG feature for each pooling window.

In an embodiment, at least one of the feature extraction modules may include a Gabor wavelet feature extraction module. The Gabor wavelet feature extraction modules may be operative to generate feature data by applying a plurality of Gabor wavelet functions to an image, wherein each of the plurality of Gabor wavelet function includes a different orientation and computing at least one feature based on at least two of the plurality of Gabor wavelet functions. For example, the at least one feature may include at least one of a sum of the absolute values of a first Gabor wavelet function of the plurality of Gabor wavelet functions and a second Gabor wavelet function of the plurality of Gabor wavelet functions, a sum of the magnitude of values of the product of each perpendicularly oriented Gabor wavelet function of the plurality of Gabor wavelet functions, or the difference of the maximum and minimum values of magnitude of all of the plurality of Gabor wavelet functions for all image data. As such, for each orientation of the plurality of Gabor wavelet functions, a plurality of different Gabor wavelet functions with a different scale are calculated. The different scales may include varied values for at least one of a wavelength of the Gabor wavelet function or an envelope of the Gabor wavelet function. Each of the plurality of Gabor wavelet functions may be applied to each pixel of each multispectral band of an image comprising a plurality of multispectral bands. At least one feature may be determined independently for each of

the plurality of bands. Additionally or alternatively, the at least one feature may be determined with respect to all of the plurality of bands.

In an embodiment, at least one of the feature extraction modules may include a clustering histogram feature extraction module for extraction of features based on clustered and classified spectra values. The clustering histogram feature extraction module may be operative to generate feature data by assigning each pixel of an image into one of a plurality of a predefined classes and identifying a kernel surrounding each pixel containing a plurality of others pixels from the image, wherein each of the plurality of other pixels have been assigned into one of the plurality of predefined classes. Furthermore, the module may build a histogram for each pixel based on the number of other pixels in each of the plurality of predefined classes relative to the pixel for which the histogram is built. The plurality of predefined classes may correspond to identified land classifications based on at least one of land use or land class. The pixels may be assigned into a predefined class based on the radiometric properties of a pixel used in a clustering technique. The clustering technique may include at least one of k-means clustering, a Euclidean distance approach, or a Mahalanobis distance approach.

In an embodiment, the classification compute module may be operable to utilize at least one machine learning algorithm to generate the classification model. The classification compute module may also be operable to utilize a plurality of machine learning algorithms to generate the classification model. For instance, a user may be operable to specify at least one of a plurality of machine learning algorithms for use in generation of the classification model.

Also, in an embodiment, the classification model may include at least one model parameter corresponding to at least one of a geographic region or a temporal period. The classification computation module model may be operable to determine the classification model for classification of image data based on the at least one model parameter and image data metadata. In an embodiment, the one or more classes correspond to at least one of a land cover, a land use, or temperature.

The system may also include a model evaluation module operable to provide quantifiable evaluation data regarding the classification module based on performance of the classification module with respect to a portion of training data received by a user. The model evaluation module may be operable to generate a classifier accuracy value for the classification module. For instance, the model evaluation module may be operable to generate at least one precision measurement for the classification module. The model evaluation

module may also be operable to compare a first model to a second model to determine if the performance of the first model is statistically significant over the performance of the second model.

In an embodiment, the feature extraction module may be operable to first produce feature information for pixels belonging to the training data prior to generation of the classification model by the classification compute model. The module may further be operable to second produce feature information for pixels other than the training data prior to application of the classification model to classify pixels other than the training data.

A second aspect includes a method for land classification based on analysis of very high resolution (VHR) remotely-sensed multispectral Earth imagery. The method includes storing VHR remotely-sensed multispectral Earth image data in an image store, generating feature data based on the image data using at least one feature extraction module, and compiling a feature stack comprising the image data and corresponding feature data. The method further includes receiving training data regarding at least one or more pixels of the image data from a user regarding at least one class to which the one or more pixels belong and generating a classification model at least in part based on a portion of the feature stack corresponding to the one or more pixels and the training data, wherein the classification model related to classification of pixels of image data into one or more classes.

A number of feature refinements and additional features are applicable to the second aspect. These feature refinements and additional features may be used individually or in any combination. As such, each of the following features that were discussed above in relation to the first aspect may be, but are not required to be, used with any other feature or combination of features of the second aspect.

BRIEF DESCRIPTION OF THE FIGURES

The accompanying drawings illustrate several embodiments of the present disclosure and, together with the description, serve to explain the principles according to the embodiments. One skilled in the art will recognize that the particular embodiments illustrated in the drawings are merely exemplary, and are not intended to limit the scope of the present disclosure.

Fig. 1 is a block diagram illustrating an exemplary hardware architecture of a computing device used in an embodiment.

Fig. 2 is a block diagram illustrating an exemplary logical architecture for a client device, according to an embodiment.

Fig. 3 is a block diagram showing an exemplary architectural arrangement of clients, servers, and external services, according to an embodiment.

Fig. 4 is a block diagram illustrating an exemplary system for generation and application of classification models for land classification.

Fig. 5 is a flow chart of an embodiment of a method for land classification.

Fig. 6 depicts an embodiment of a feature stack of feature data including a plurality of data layers that may be used in land classification.

Figs. 7-10B depict embodiments of screen shots of a user interface for receipt of training data from a user and presentation of land classification results to a user.

Fig. 11 depicts an embodiment of a graphical representation of a classification model that may be generated that is operable to classify image data.

Fig. 12 depicts an embodiment of a graphical representation of a method for determining a fractal dimension of an image.

Fig. 13 depicts an embodiment of a plurality of bins for use in a histogram of gradients feature extraction technique.

Figs. 14A and 14B depict an embodiment of a graphical representation of a rotationally invariant histogram of gradients technique applied to an image.

DETAILED DESCRIPTION

The following description is not intended to limit the invention to the forms disclosed herein. Consequently, variations and modifications commensurate with the following teachings, skill and knowledge of the relevant art, are within the scope of the present invention. The embodiments described herein are further intended to explain modes known of practicing the invention and to enable others skilled in the art to utilize the invention in such, or other embodiments and with various modifications required by the particular application(s) or use(s) of the present invention.

One or more different inventions may be described in the present application. Further, for one or more

of the inventions described herein, numerous alternative embodiments may be described; it should be understood that these are presented for illustrative purposes only. The described embodiments are not intended to be limiting in any sense. One or more of the inventions may be widely applicable to numerous embodiments, as is readily apparent from the disclosure. In general, embodiments are described in sufficient detail to enable those skilled in the art to practice one or more of the inventions, and it is to be understood that other embodiments may be utilized and that structural, logical, software, electrical and other changes may be made without departing from the scope of the particular inventions. Accordingly, those skilled in the art will recognize that one or more of the inventions may be practiced with various modifications and alterations. Particular features of one or more of the inventions may be described with reference to one or more particular embodiments or figures that form a part of the present disclosure, and in which are shown, by way of illustration, specific embodiments of one or more of the inventions. It should be understood, however, that such features are not limited to usage in the one or more particular embodiments or figures with reference to which they are described. The present disclosure is neither a literal description of all embodiments of one or more of the inventions nor a listing of features of one or more of the inventions that must be present in all embodiments.

Devices that are described as being in communication with each other need not be in continuous communication with each other, unless expressly specified otherwise. In addition, devices that are described as being in communication with each other may communicate directly or indirectly through one or more intermediaries, logical or physical. A description of an embodiment with several components in communication with each other does not imply that all such components are required. To the contrary, a variety of optional components may be described to illustrate a wide variety of possible embodiments of one or more of the inventions and in order to more fully illustrate one or more aspects of the inventions.

Similarly, although process steps, method steps, algorithms or the like may be described in a sequential order, such processes, methods and algorithms may generally be configured to work in alternate orders, unless specifically stated to the contrary. In other words, any sequence or order of steps that may be described in this patent application does not, in and of itself, indicate a requirement that the steps be performed in that order. The steps of described processes may be performed in any order practical. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step). Moreover, the illustration of a process by its depiction in a drawing does not

imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to one or more of the invention(s), and does not imply that the illustrated process is preferred. Also, steps are generally described once per embodiment, but this does not mean they must occur once, or that they may only occur once each time a process, method, or algorithm is carried out or executed. Some steps may be omitted in some embodiments or some occurrences, or some steps may be executed more than once in a given embodiment or occurrence.

When a single device or article is described, it will be readily apparent that more than one device or article may be used in place of a single device or article. Similarly, where more than one device or article is described, it will be readily apparent that a single device or article may be used in place of the more than one device or article. The functionality or the features of a device may be alternatively embodied by one or more other devices that are not explicitly described as having such functionality or features. Thus, other embodiments of one or more of the inventions need not include the device itself.

Techniques and mechanisms described or referenced herein will sometimes be described in singular form for clarity. However, it should be noted that particular embodiments include multiple iterations of a technique or multiple instantiations of a mechanism unless noted otherwise. Process descriptions or blocks in figures should be understood as representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of embodiments of the present invention in which, for example, functions may be executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those having ordinary skill in the art.

As used herein, a "feature" may correspond to an individual measurable heuristic property of a phenomenon being observed (e.g., a value related to image data). In this regard, features or feature data may be extracted from image data (e.g., by performing calculations and/or transformations of image data). As used herein, a "pixel-level feature" is a feature at a base level of classification. For example, pixel-level features may include textural properties, mathematical transforms applied to pixel data, a land-cover classification, or some other appropriate property description specific to individual pixels. As used herein, a "region-level feature" is a feature at a higher level of classification. For example, region-level feature observations may be related to

morphology, i.e., may have shape properties (such as, area, perimeter, compactness, elongation, eccentricity, etc.), spatial relationships (such as, arrangement, distance, etc.), object classifications (for example, a school, paring lot, swimming pool, plane, shopping mall, etc.), and the like. As used herein, a "scene-level feature" is a feature that may aggregate statistics on lower level features (e.g., region-level features or pixel-level features), such as, percentage land cover (for example, 25% deciduous forest, 10% water, etc.), aggregate object counts (for example, 10 schools, 35 parking lots, etc.), other descriptive classifications (for example, desert, city, rainforest, etc.)

As used herein, a "graphics processing unit (GPU)" is a specialized electronic circuit known in the art that is designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display or for other computational purposes. As used herein, a "Compute Unified Device Architecture (CUDA)" is a parallel computing platform and programming model known in the art created by NVIDIA™ and implemented by the graphics processing units (GPUs) that they produce that gives developers access to a virtual instruction set and memory of the parallel computational elements in CUDA GPUs.

Generally, the techniques disclosed herein may be implemented on hardware or a combination of software and hardware. For example, they may be implemented in an operating system kernel, in a separate user process, in a library package bound into network applications, on a specially constructed machine, on an application-specific integrated circuit (ASIC), or on a network interface card.

Software/hardware hybrid implementations of at least some of the embodiments disclosed herein may be implemented on a programmable network-resident machine (which should be understood to include intermittently connected network-aware machines) selectively activated or reconfigured by a computer program stored in memory. Such network devices may have multiple network interfaces that may be configured or designed to utilize different types of network communication protocols. A general architecture for some of these machines may be disclosed herein in order to illustrate one or more exemplary means by which a given unit of functionality may be implemented. According to specific embodiments, at least some of the features or functionalities of the various embodiments disclosed herein may be implemented on one or more general-purpose computers associated with one or more networks, such as for example an end-user computer system, a client computer, a network server or other server system, a mobile computing device (e.g., tablet computing device, mobile phone, smartphone, laptop, and the like), a consumer electronic device, a music player, or any

other suitable electronic device, router, switch, or the like, or any combination thereof. In at least some embodiments, at least some of the features or functionalities of the various embodiments disclosed herein may be implemented in one or more virtualized computing environments (e.g., network computing clouds, virtual machines hosted on one or more physical computing machines, or the like).

Referring now to Fig. 1, there is shown a block diagram depicting an exemplary computing device 100 suitable for implementing at least a portion of the features or functionalities disclosed herein. Computing device 100 may be, for example, any one of the computing machines listed in the previous paragraph, or indeed any other electronic device capable of executing software- or hardware-based instructions according to one or more programs stored in memory. Computing device 100 may be adapted to communicate with a plurality of other computing devices, such as clients or servers, over communications networks such as a wide area network, a metropolitan area network, a local area network, a wireless network, the Internet, or any other network, using known protocols for such communication, whether wireless or wired.

In one embodiment, computing device 100 includes one or more central processing units (CPU) 102, one or more interfaces 110, and one or more busses 106 (such as a peripheral component interconnect (PCI) bus). When acting under the control of appropriate software or firmware, CPU 102 may be responsible for implementing specific functions associated with the functions of a specifically configured computing device or machine. For example, in at least one embodiment, a computing device 100 may be configured or designed to function as a server system utilizing CPU 102, local memory 101 and/or remote memory 120, and interface(s) 110. In at least one embodiment, CPU 102 may be caused to perform one or more of the different types of functions and/or operations under the control of software modules or components, which for example, may include an operating system and any appropriate applications software, drivers, and the like.

CPU 102 may include one or more processors 103 such as, for example, a processor from one of the Intel, ARM, Qualcomm, and AMD families of microprocessors. In some embodiments, processors 103 may include specially designed hardware such as application-specific integrated circuits (ASICs), electrically erasable programmable read-only memories (EEPROMs), field-programmable gate arrays (FPGAs), and so forth, for controlling operations of computing device 100. In a specific embodiment, a local memory 101 (such as non-volatile random access memory (RAM) and/or read-only memory (ROM), including for example one or more levels of cached memory) may also form part of CPU 102. However, there are many different ways in which

memory may be coupled to system 100. Memory 101 may be used for a variety of purposes such as, for example, caching and/or storing data, programming instructions, and the like. As used herein, the term "processor" is not limited merely to those integrated circuits referred to in the art as a processor, a mobile processor, or a microprocessor, but broadly refers to a microcontroller, a microcomputer, a programmable logic controller, an application-specific integrated circuit, and any other programmable circuit.

In one embodiment, interfaces 110 are provided as network interface cards (NICs). Generally, NICs control the sending and receiving of data packets over a computer network; other types of interfaces 110 may for example support other peripherals used with computing device 100. Among the interfaces that may be provided are Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, graphics interfaces, and the like. In addition, various types of interfaces may be provided such as, for example, universal serial bus (USB), Serial, Ethernet, Firewire™, PCI, parallel, radio frequency (RF), Bluetooth™, near-field communications (e.g., using near-field magnetics), 802.11 (WiFi), frame relay, TCP/IP, ISDN, fast Ethernet interfaces, Gigabit Ethernet interfaces, asynchronous transfer mode (ATM) interfaces, high-speed serial interface (HSSI) interfaces, Point of Sale (POS) interfaces, fiber data distributed interfaces (FDDIs), and the like. Generally, such interfaces 110 may include ports appropriate for communication with appropriate media. In some cases, they may also include an independent processor and, in some in stances, volatile and/or non-volatile memory (e.g., RAM).

Although the system shown in Fig. 1 illustrates one specific architecture for a computing device 100 for implementing one or more of the inventions described herein, it is by no means the only device architecture on which at least a portion of the features and techniques described herein may be implemented. For example, architectures having one or any number of processors 103 may be used, and such processors 103 may be present in a single device or distributed among any number of devices. In one embodiment, a single processor 103 handles communications as well as routing computations, while in other embodiments a separate dedicated communications processor may be provided. In various embodiments, different types of features or functionalities may be implemented in a system according to the invention that includes a client device (such as a tablet device or smartphone running client software) and server systems (such as a server system described in more detail below).

Regardless of network device configuration, the system of the present invention may employ one or

more memories or memory modules (such as, for example, remote memory block 120 and local memory 101) configured to store data, program instructions for the general-purpose network operations, or other information relating to the functionality of the embodiments described herein (or any combinations of the above). Program instructions may control execution of or comprise an operating system and/or one or more applications, for example. Memory 120 or memories 101, 120 may also be configured to store data structures, configuration data, encryption data, historical system operations information, or any other specific or generic non-program information described herein.

Because such information and program instructions may be employed to implement one or more systems or methods described herein, at least some network device embodiments may include nontransitory machine-readable storage media, which, for example, may be configured or designed to store program instructions, state information, and the like for performing various operations described herein. Examples of such nontransitory machine- readable storage media include, but are not limited to, magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD- ROM disks; magneto-optical media such as optical disks, and hardware devices that are specially configured to store and perform program instructions, such as read-only memory devices (ROM), flash memory, solid state drives, memristor memory, random access memory (RAM), and the like. Examples of program instructions include both object code, such as may be produced by a compiler, machine code, such as may be produced by an assembler or a linker, byte code, such as may be generated by for example a Java™ compiler and may be executed using a Java virtual machine (JVM) or equivalent, or files containing higher level code that may be executed by the computer using an interpreter (for example, scripts written in Python, Perl, Ruby, Groovy, or any other scripting language).

In some embodiments, systems according to the present invention may be implemented on a standalone computing system. Referring now to Fig. 2, there is shown a block diagram depicting a typical exemplary architecture of one or more embodiments or components thereof on a standalone computing system. Computing device 200 includes processors 210 that may run software that carry out one or more functions or applications of embodiments of the invention, such as for example a client application 230. Processors 210 may carry out computing instructions under control of an operating system 220 such as, for example, a version of Microsoft's Windows™ operating system, Apple's Mac OS/X or iOS operating systems, some variety of the Linux operating system, Google's Android™ operating system, or the like. In many cases, one or more shared

services 225 may be operable in system 200, and may be useful for providing common services to client applications 230. Services 225 may for example be Windows™ services, user-space common services in a Linux environment, or any other type of common service architecture used with operating system 210. Input devices 270 may be of any type suitable for receiving user input, including for example a keyboard, touchscreen, microphone (for example, for voice input), mouse, touchpad, trackball, or any combination thereof. Output devices 260 may be of any type suitable for providing output to one or more users, whether remote or local to system 200, and may include for example one or more screens for visual output, speakers, printers, or any combination thereof. Memory 240 may be random-access memory having any structure and architecture known in the art, for use by processors 210, for example to run software. Storage devices 250 may be any magnetic, optical, mechanical, memristor, or electrical storage device for storage of data in digital form. Examples of storage devices 250 include flash memory, magnetic hard drive, CD-ROM, and/or the like.

In some embodiments, systems of the present invention may be implemented on a distributed computing network, such as one having any number of clients and/or servers. Referring now to Fig. 3, there is shown a block diagram depicting an exemplary architecture for implementing at least a portion of a system according to an embodiment of the invention on a distributed computing network. According to the embodiment, any number of clients 330 may be provided. Each client 330 may run software for implementing client-side portions of the present invention; clients may comprise a system 200 such as that illustrated in Fig. 2. In addition, any number of servers 320 may be provided for handling requests received from one or more clients 330. Clients 330 and servers 320 may communicate with one another via one or more electronic networks 310, which may be in various embodiments any of the Internet, a wide area network, a mobile telephony network, a wireless network (such as WiFi, Wimax, and so forth), or a local area network (or indeed any network topology known in the art; the invention does not prefer any one network topology over any other). Networks 310 may be implemented using any known network protocols, including for example wired and/or wireless protocols.

In addition, in some embodiments, servers 320 may call external services 370 when needed to obtain additional information, or to refer to additional data concerning a particular call. Communications with external services 370 may take place, for example, via one or more networks 310. In various embodiments, external services 370 may comprise web-enabled services or functionality related to or installed on the hardware device itself. For example, in an embodiment where client applications 230 are implemented on a smartphone or other

electronic device, client applications 230 may obtain information stored in a server system 320 in the cloud or on an external service 370 deployed on one or more of a particular enterprise's or user's premises.

In some embodiments of the invention, clients 330 or servers 320 (or both) may make use of one or more specialized services or appliances that may be deployed locally or remotely across one or more networks 310. For example, one or more databases 340 may be used or referred to by one or more embodiments of the invention. It should be understood by one having ordinary skill in the art that databases 340 may be arranged in a wide variety of architectures and using a wide variety of data access and manipulation means. For example, in various embodiments one or more databases 340 may comprise a relational database system using a structured query language (SQL), while others may comprise an alternative data storage technology such as those referred to in the art as "NoSQL" (for example, Hadoop Cassandra, Google BigTable, and so forth). In some embodiments, variant database architectures such as column-oriented databases, in-memory databases, clustered databases, distributed databases, or even flat file data repositories may be used according to the invention. It will be appreciated by one having ordinary skill in the art that any combination of known or future database technologies may be used as appropriate, unless a specific database technology or a specific arrangement of components is specified for a particular embodiment herein. Moreover, it should be appreciated that the term "database" as used herein may refer to a physical database machine, a cluster of machines acting as a single database system, or a logical database within an overall database management system. Unless a specific meaning is specified for a given use of the term "database", it should be construed to mean any of these senses of the word, all of which are understood as a plain meaning of the term "database" by those having ordinary skill in the art.

Similarly, most embodiments of the invention may make use of one or more security systems 360 and configuration systems 350. Security and configuration management are common information technology (IT) and web functions, and some amount of each are generally associated with any IT or web systems. It should be understood by one having ordinary skill in the art that any configuration or security subsystems known in the art now or in the future may be used in conjunction with embodiments of the invention without limitation, unless a specific security 360 or configuration system 350 or approach is specifically required by the description of any specific embodiment.

In various embodiments, functionality for implementing systems or methods of the present invention

may be distributed among any number of client and/or server components. For example, various software modules may be implemented for performing various functions in connection with the present invention, and such modules may be variously implemented to run on server and/or client components.

Fig. 4 is a block diagram illustrating an exemplary system 400 that may be used for land classification. As illustrated, system 400 may comprise a plurality of software or hardware elements that may be variably interconnected, and it should be appreciated that such components may be variable in implementation; that is, components may be software-based and stored and operating on separate or joined hardware systems interchangeably (for example by running multiple software systems on a single computer server), or may be physically-distinct hardware such as dedicated computing hardware for specific elements as illustrated. It should be further appreciated that connections between components as illustrated may be direct physical connections (such as via cables or internal software connectivity within a hardware system operating multiple components as described above) or they may be remote data connections such as communication over the Internet or another communications network. In this manner, it should be appreciated that the constructions and arrangement illustrated is exemplary, and a variety of arrangements may be possible, and it should be further appreciated that the quantity and arrangement of components is exemplary and alternate or additional components may be utilized. In this manner it can be appreciated that the nature of the system 400 is flexible and scalable, i.e. it may be arranged as suitable for varying particular implementations as may be appropriate (such as a small-scale implementation utilizing software elements operating within a personal computer for individual use, or a large-scale implementation utilizing separate computer servers for operation utilizing greater computing power or storage space, as may be desirable for an enterprise or cloud-based service).

As illustrated, image data 411 may be made available to an image data server 420, and may be stored in a data store 410 or similar data storage. Image data 411 may be of variable nature such as simple stored images or more detailed information or "metadata" stored along with images such as image tags or other identifying data (such as, as is common in the art, data pertaining to when, where, or how an image was captured, such as might be recorded by an image capture device 412 when taking a snapshot). Furthermore, image data 411 need not be an existing cache of stored data, and could be a quantity of "live" data being accessed by system 400 for operation such as capturing images via any appropriate image-capture devices 412

connected to system 400 such as cameras 413, image capture-capable electronic devices such as tablet computing devices or smartphones 414, or aerial or satellite imaging systems 415.

It should be appreciated that the data store 410 may be accomplished via a variety of means common in the art, such as a single physical storage medium (such as magnetic, optical, solid-state, or other storage devices common in the art), or a distributed storage system (such as a distributed file storage system as may be utilized in computing clusters), and further that such storage may be local (i.e., stored and operating alongside other components of system 400 such as a server's hard disk drive or a connected storage system) or remote (such as a remotely-connected and accessible storage device or a cloud-based storage service), and that connection may be accomplished via either physical or remote means, as described previously. In this manner it should be appreciated that image storage is highly variable, further enabling a flexible and scalable design that may be readily adapted to a variety of storage means as may be appropriate, without necessitating the manipulation of existing data.

The system 400 may include a land classification module 405 that includes the image data server 420. As such, image data 411 may be received at the land classification module 405 and in turn stored in the data store 410. As described above, the image data 411 may include raw image data received from an image capture device 412 or the image data 411 may include image data 411 that has been preprocessed. For instance, in the case of VHR remotely-sensed multispectral Earth imagery, the image data 411 received at the land classification module 406 may be orthorectified and/or atmospherically compensated satellite imagery. Other image processing common to remotely-sensed imagery may also be provided without limitation.

In this regard, the image data 411 may include a digital number representing a gray level value for each pixel of an image in at least one spectral band. The gray level value may be representative of a surface reflectance of the Earth for a given pixel in a given spectral band and/or may include an atmospherically compensated value. In one exemplary application, the image data 411 received at the land classification module 405 may be multispectral satellite imagery including a plurality of bands, a panchromatic band, or a combination thereof. For instance, in VHR remotely-sensed multispectral Earth imagery, a plurality of spectral bands may be collected. That is, the image sensor (e.g., provided on a satellite imaging system 415) used to collect the image data 411 may have a plurality of specifically designed sensor portions capable of detected light at a predetermined wavelength. For instance, WorldView 2 operated by DigitalGlobe, Inc. of Longmont, CO collects

data in 8 spectral bands including, for example, a coastal band (400-450 nm), a blue band (450-510 nm), a green band (510-580 nm), a yellow band (585-625 nm), a red band (630-690 nm), a red edge band (705-745 nm), a near-infrared 1 band (770-895 nm), and a near-infrared 2 band (860-1040 nm). There may also be a panchromatic sensor capable of detecting black and white imagery (also referred to as a panchromatic band) in the wavelength band of 450-800 nm. Further still, in at least some embodiments, one or more bands in the short-wavelength infrared range (SWIR) may be provided. For example, one or more SWIR bands may be provided including, for example, SWIR 1 (1195-1225 nm), SWIR 2 (1550-1590 nm), SWIR 3 (1640-1680 nm), SWIR 4 (1710-1750 nm), SWIR 5 (2145-2185 nm), SWIR 6 (2185-2225 nm), SWIR 7 (2235-2285 nm), and/or SWIR 8 (2295-2365 nm). Other combinations and/or ranges of SWIR bands generally from about 1195 nm to about 2400 nm may be provided in any combination. For example, in some embodiments, band definitions broader and/or narrower than those described above may be provided without limitation. In any regard, there may be a plurality of band values corresponding to gray level values for each band for each given pixel in a portion of multispectral image data.

Additionally, the image data 411 may include metadata that may include supplementary data regarding the acquisition of the image. For instance, image metadata that may accompany and/or form a portion of the image data 411 may include satellite parameters (e.g., off nadir satellite angles, satellite attitudes, solar elevation angles, etc.), time/date of acquisition, and/or other appropriate parameters may be attributed to the metadata of an image 411.

The land classification module 405 may also include a client communication module 440 that is operable to communicate with a client interface 430. In view of the description above, it may be appreciated that the land classification module 405 may be executed with respect to a distributed system. In this regard, the client interface 430 may include a user interface that is presented to a remote user such that the client interface 430 may communicate with the land classification module 405 over an appropriate network. As such, it may be appreciated that there may be provided a plurality of client interfaces 430 to facilitate multiple users accessing the land classification module 405. Accordingly, the client communication module 440 may be operable to communicate (e.g., receipt and/or transmission of data) to and/or from the plurality of client interfaces 430.

In an embodiment, the client interface 430 may be provided as a web browser based interface that may communicate with an application programming interface at the client communication module 440 to facilitate

communication between the land classification system 405 and the client interface 430. In this regard, the communication between the client interface 430 and the client communication module 440 may be provided in an appropriate protocol associated with internet communication such as, for example, hypertext transfer protocol (http), although other protocols may be used. Accordingly, the client interface 430 may be provided as a web based interface that facilitates user interaction with the land classification system 405 as will be described in greater detail below.

In another embodiment, the land classification system 405 and client interface 430 may be provided locally at a single computing device for execution. As such, the client interface 430 may compriee one or more user interface devices of a workstation that may also execute the land classification module 405 locally.

The land classification system 405 may also include one or more feature extraction modules 450. The feature extraction module(s) 450 may be operable to access the data store 410 and image data 411. Each feature extraction module 450 may be operable to generate feature data 452 regarding one or more predetermined features. The feature data 452 may be generated based on the image data 411. In turn, the feature data 452 may be stored in the data store 410. For instance, the feature data 452 may be stored in corresponding relation to the image data 411 such that for a given pixel, a plurality of feature layers of data may be provided that is stored in correspondence with the image.

With further reference to Fig. 6, it may be appreciated that for each portion of image data 411, a plurality of layers of information or data related to or comprising subcomponents of the image data 411 may be provided in the form of a feature stack 600. In this regard, the feature stack 600 may include image data 411 (e.g., spectral band data 610 for each pixel) and one or more layers generated by way of a feature extraction module 450. For instance, the spectral band data 610 may include a gray value for each pixel for each spectral band provided. In this regard, as shown in Fig. 6, eight spectral bands are provided corresponding to, for example, spectral data collected from the eight spectral detectors of the WorldView 2 satellite imaging system 415 operated by DigitalGlobe, Inc. However, additional or fewer numbers of spectral bands may be provided in the spectral band data 610.

In any regard, a visible image may be generated that may include components from one or more of the layers of the feature stack 600. For instance, in one embodiment, data from the spectral band data 610 corresponding to wavelengths visible to the human eye, such as for example, the red, green, and blue bands

may be combined to produce a natural color image representative of the image as visible to the human eye. This layer may be stored as a spectral feature layer 620 as it may comprise a function (i.e., an extracted feature) based on one or more of the spectral band data 610. Furthermore, visible images may be generated based on gray level values from spectral band data 610 not visible to the human eye to generate visible representations of such bands (e.g., alone or in combination with other human visible and/or nonhuman visible bands to generate false color images). In this regard, the spectral feature layers 620 of the feature stack 600 may include, for example, layers that include spectral band data 610 of the image data 411 and/or combinations thereof.

Accordingly, the extraction of a limited number of bands (e.g., the red, green and blue bands or "RGB" bands to construct an RGB natural color image) may allow for reduced computational overhead when presenting the image data to the user. That is, rather than loading and/or attempting to display all values of all bands available for an image, the RGB bands may be used to present an image that represents the human-visible portion of the spectrum. Accordingly, where the user views or manipulates the image (e.g., for purposes of providing training data or the like), the use of the RGB image may be more efficiently loaded and/or manipulated with less computational overhead.

The spectral feature layers 620 may also include ratios of gray values for one or more bands for each pixel based on the spectral band data 610. For example, in one ratio of spectral bands that may be generated as a layer includes:

$$\frac{(Band\ A_i - Band\ B_i)}{(Band\ A_i + Band\ B_i)}$$

Equation 1

where *Band A*$_i$ represents a pixel value from a first spectral band in the spectral band data 610 for a given pixel *i*, and *Band B*$_i$ represents a pixel value from a second spectral band in the spectral band data 610 for the given pixel *i*.

The feature extraction modules 450 may also be operable to generate data layers that include feature values for a given portion of image data 411. For example, textural feature layers 630 and/or morphological feature layers 640 may also be generated for a portion of image data 411 that form a portion of the feature stack 600.

In any regard, it may be appreciated that the layers in the feature stack 600 may each include corresponding pixel feature data for a pixel from the image data 411. For example, in Fig. 6, the image data 411 may include a nine pixel by nine pixel portion of image data 411. A highlighted pixel 650 has been denoted with an "x." Values for pixels 650a-650f are also provided in various feature data layers that correspond to pixel 650 and also denoted in the feature data layers of the feature stack 600. These additional corresponding pixels 650a-650f are each provided in a corresponding position in a respective feature data layer to pixel 650. That is, for each feature data layer, a corresponding nine by nine pixel map may be provided with values corresponding to the feature data for each one of the pixels of the image data 411. It may be appreciated that the resulting feature data layer generated by a feature extraction module 450 may, but need not, correspond to the size of the portion of image data 411 used to generate the layer. That is, the resulting data layer may be a larger or smaller array than the image data 411. Furthermore, each pixel in the feature data layer may be a vector value or other type of output corresponding to a portion of the image data 411. However, in the example depicted in Fig. 6, for example, pixels 650a and 650b may include values corresponding to spectral features extracted from pixel 650 (e.g., from one or more of the spectral band data 610 corresponding to pixel value 650). Pixels 650c and 650d may include values corresponding to textural features extracted based on a feature extraction operation carried out relative to pixel 650. Pixels 650e and 650f may include values corresponding to morphological features extracted based on a feature extraction operation carried out relative to pixel 650. In any regard, it may be appreciated that for a given pixel 650, corresponding feature values (i.e., pixels 650a-650f) may be provided that correspond to the pixel 650. As such, the values for pixels 650a-650f may be associated with pixel 650 such that the values for each of the pixels may be associated. This may be particularly valuable in a supervised machine learning process as will be appreciated in greater detail below.

It may further be appreciated that in the case of VHR remotely-sensed multispectral Earth imagery, each pixel 650 may be correlated to a corresponding geographic position. As such, for each pixel 650, a geographic identifier may be provided that is indicative of the geographic location corresponding to a given pixel 650.

Textural feature layers 630 may be generated using feature extraction modules 450 including, for example, Gabor filter feature data, histogram of oriented gradients (HOG) features, or any appropriate other textural features. Morphological feature layers 640 may be generated using feature extraction modules 450

including, for example, identification of human buildup or the like (e.g., as discussed in U.S. Pat. App. No. 14/013,304 entitled AUTOMATIC EXTRACTION OF BUILT-UP FOOTPRINTS FROM HIGH RESOLUTION OVERHEAD IMAGERY THROUGH MANIPULATION OF ALPHATREE DATA STRUCTURES filed on August 29, 2013, the entirety of which is incorporated by reference). As may be appreciated from Fig. 6, there may be any number of spectral feature layers 620, textural feature layers 630, and/or morphological feature layers 640 provided in the feature stack 600. In this regard, it may be appreciated that certain feature modules 450 capable of generating feature layers may provide particularly relevant features for use in land use classification.

For example, specific examples of feature extraction modules 450 that may be utilized in the land classification system 405 are discussed below. However, it may further be appreciated that any other feature extraction module 450 may be utilized that may generate feature data 452 to generate a data layer that may be useful in land classification.

One example of a feature extraction module 450 that may execute in the feature extraction system 400 described above is a fractal dimension feature extraction module. The term "fractal dimension" effectively defines the extent to which a lower dimensional function (e.g., a one dimensional line, two dimensional plane, etc.) effectively occupies a space of a higher dimension (e.g., two dimensional space in the case of the one dimension line or three dimensional space in the case of the two dimensional plane). That is, the fractal dimension may provide a measure of the complexity of a function of a given dimension. As an example, two one dimensional lines defined as $y = \sin\left(\frac{10}{x}\right)$ and $y = x$, respectively, may have differing fractal dimensions for an interval of $-0.2 \leq x \leq 0.2$. That is, the function $y = \sin\left(\frac{10}{x}\right)$ may have a higher fractal dimension over the interval than the function $y = x$ as the sine function may occupy much more of the space in the interval although both functions have a topological dimension of one.

Accordingly, for an image, the gray level values for a given spectral band may be considered as describing a convoluted, two-dimensional surface, the fractal dimension of which may provide information about the "roughness" of the two-dimensional surface defined by the gray level values. That is, the image may be conceptualized as a three dimensional surface whose height from the normal at each pixel is represented by a gray value of the pixel. In this regard, the fractal dimension feature extraction module may be operable to extract feature information in relation to the fractal dimension of one or more portions of an image that may provide

useful information regarding the nature of the image. The fractal dimension for a given portion of an image may be determined using any appropriate method for calculation of a fractal dimension.

In one particular embodiment of a fractal dimension feature extraction module conceptually illustrated in Fig. 12, an image 900 may be divided into a plurality of subdivisions 910 of a predetermined size of pixels. This predetermined size of the subdivision 910 of the image 900 may be defined as $N \times N$ pixels. As may be appreciated, the subdivision of the image may be provided in accordance with a tiling scheme applied to geospatial images such as tiling schemes commonly used in mapping applications such as, for example, the Google Maps Tiling Scheme, described in the Google Maps API document available from Google, Inc., Mountain View, CA, the Bing Maps Tiling Scheme, described in the Bing Maps Tile System available from Microsoft Corporation, Redmond, WA, or a custom tiling scheme that may, for example, be a geographic based projection. For each predetermined image of size $N \times N$, the image may be divided into a plurality of abutting windows 912 within the $N \times N$ image of size $W \times W$ pixels, where W is an integer that complies with the inequalities $W \leq \frac{N}{2}$ and $W \geq 3$ and represents a side dimension of the abutting window 612.

As such, for each window 912 of size $W \times W$, a column of boxes 914 having a volume of $W \times W \times W'$ may be constructed. In this regard, W' may be considered the height of the column 914. The height may be represented in increments of the side dimension of the abutting window (W). In this regard, the height may be determined such that the ratio of the predetermined size of the image (N) to the side dimension of the abutting windows (W) equals the ratio of the maximum pixel gray value for a given multispectral band of the image to the height of the column 614. That is, where D is the gray level range for the image, the height of the column 914 may be determined with the equation $\frac{N}{W} = \frac{D}{W'}$. As such, a column of $W \times W \times W'$ is created for each abutting window in the image.

In this regard, for each abutting window 912 of size $W \times W$, the minimum and maximum gray value for each pixel is located within a specific one of the boxes defined in the $W \times W \times W'$ sized column 914. The boxes of size $W \times W \times W'$ may be numbered with successive integers in ascending order extending away from the base of the column (e.g., as shown in Fig. 12 as Box 1, Box 2, Box 3...Box n). Once the minimum and maximum gray value has been located, the distance separating the minimum and maximum values may be computed as $n = h - l + 1$ where h and l are the numbers of boxes with the highest and lowest values,

respectively, over the $W \times W \times W'$ sized box. This process is repeated for each abutting window 912 of size $W \times W$ to determine a column height W' and a n value for each. The total number of boxes to cover the whole image subdivision 910 of size $N \times N$ may be calculated as:

$$N_W = \sum_{i,j} n$$

Equation 2

This process is further repeated for all values of W that satisfy the above noted inequalities related to the size of W. In turn, a log-log plot of values of N_W versus W may be generated and the negative slope of the least squares line of the plot may define the fractal dimension. In turn, each subdivision 910 of the image 900 may undergo this process to determine the fractal dimension of each subdivision 910. In turn, a fractal map may be generated that is an aggregation of information from each unique $N \times N$ subdivision 910 of the original image 900. Such information may be suited for application to decision tree techniques for classification as the textural information per pixel may be relatively dense.

Another feature extraction module 450 that may be provided may include a total variational feature extraction module. In VHR multispectral remotely-sensed Earth imagery, a plurality of spectral bands may be collected. That is, the image sensor used to collect image data may have a plurality of specifically designed sensor portions capable of detecting light at a predetermined wavelength. For instance, WorldView 2 operated by DigitalGlobe, Inc. collects data in 8 spectral bands as described above. There may also be a panchromatic sensor capable of detecting black and white imagery in a broad wavelength band (e.g., the wavelength band of 450-800 nm). In any regard, there may be a plurality of band values (e.g., spectral band data 610 shown in Fig. 6) corresponding to gray level values for each band for each given pixel in the multispectral image data 411. Traditionally, feature sets have utilized combinations of gray level values for each spectral band and/or normalized difference ratios as a feature set for analysis, such as the approach described above relating to spectral band ratios. However, it may be appreciated that these values may be corrupted by, for example, sensor noise or local texture on a given surface type. Accordingly, prior approaches have been proposed that filter gray level values over a small spatial window by, for example, taking the mean or median gray level value for a group of pixels in the spatial window. However, while this type of filtering may reduce noise, the filtering may blur information across discontinuities, which, for example, may obscure edge detection or other techniques

to be applied to the image. Furthermore, nonlinear median and morphological filters typically operate on a per band basis, where the filter is independently applied to each of the plurality of spectral bands of an image. This becomes less effective for images having a plurality of bands.

In turn, the total variational feature extraction module may operate on a plurality of spectral bands of an image (e.g., all 8 spectral bands of 8 band image data 411) simultaneously to perform a variational simplification across the multiple bands of the image. Accordingly, the output of the total variational feature extraction module is a data layer comprising a filtered set of image bands (i.e., a filtered image) that may be utilized in further image processing (e.g., utilizing the filtered gray level values or normalized ratios therefrom).

The total variational feature extraction module may be a global optimization that all image pixels from a plurality of spectral bands are optimized jointly using the formula:

$$f(u) = \sum_i |u_i - x_i| + \lambda |grad(u_i)|$$

Equation 3

where *i* represents a multivalued vector at a pixel, *u* is the filtered image stack (row, column, band), and *x* is the input image. The L1 vector norm of a given function *g(x)* is represented by the value $|g(x)|$ in the above equation, where the L1 vector norm is defined by the equation:

$$|g(x)| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Equation 4

The function *grad()* represents a vector-valued spatial gradient defined as:

$$|grad(u_i)| = \sqrt{|gradx(u_i)|^2 + |grady(u_i)|^2}$$

Equation 5

where *gradx()* and *grady()* represent the spatial gradient in the image x and y directions, respectively. In this regard, Equation 3 may be optimized to determine a global minimum for the image over a plurality of bands. As such, the best filtered image *u* is found when Equation 3 is globally minimized over the entire image including all spectral bands. In this regard, it may be appreciated that the first term of Equation 3 may penalize the filtered image as the filtered image differs from the original image. The second term of Equation 3 attempts to regularize

the filtered image u such that the resulting image values of u are spatially homogenous. To this end, it may be appreciated that as the L1 norm is utilized (e.g., in contrast to the L2 norm), in the first term, an occasional large discrepancy or outlier may be tolerated in the minimization. Similarly, the use of the L1 norm in the second term allows for abrupt edge discontinuities in the second term. Additionally, in the second term λ represents a smoothing factor that allows for tuning of the desired smoothness in the resulting filtered image u.

In this regard, Equation 3 may be optimized to determine the resulting global minimum for the function $f(u)$. The solution may be obtained using any optimization approach, however, it has been found that in a preferred embodiment, a Split Bregman technique may be employed. Additionally, Gauss-Seidel updates may be performed. A discussion of the Split Bregman technique is disclosed in Goldstein, Tom and Osher, Stanley, The Split Bregman Method for L1 Regularized Problems, available at ftp://ftp.math.ucla.edu/pub/camreport/cam08-29.pdf, the entirety of which is incorporated by reference herein. It has been found that this approach to the optimization is quite fast and maps well to execution on a GPU (e.g., a CUPA process) due to the highly parallel nature of the algorithm.

Another feature extraction module 450 that may be provided is a rotational invariant histogram of gradients module. In this regard, one skilled in the art will recognize that histograms of oriented gradients (HOGs) is an established technique for computing and histogramming gradients in images based on the orientation and magnitudes of the gradients over some window. For example, Dalal, Navneet and Triggs, Bill, Histograms of Oriented Gradients for Human Detection, International Conference on Computer Vision and Pattern Recognition, (2005), discusses one such approach and is incorporated by reference in its entirety.

However, previous approaches to HOGs relied on underlying assumptions regarding the orientation of objects to be identified using HOGs. For instance, prior work may have assumed that human figures would generally appear vertically in a standing position. However, in the case of VHR multispectral remotely-sensed Earth imagery, such assumptions based on suspected orientations may be of very little to no value as the orientation of objects in such imagery may be provided over a plurality of orientations or at random. As such, a rotationally invariant approach to HOGs is proposed and discussed herein.

Specifically, as may be appreciated, a gradient for a specific pixel may include an x (i.e., horizontal) and y (i.e., vertical) component that may be orthogonal to each other. In the rotationally invariant approach to HOG, an image 900 may be represented in gradient domain 950. In the gradient domain 950, N annular bins 952 may

be established in addition to a low gradient region disposed in the center 954 of the *N* annular bins 952. In this regard, each annular bin 952 may correspond with a portion of a ring surrounding the center 954. Fig. 13 depicts an example of one arrangement of the annular bins 952 and center 954 as arranged in a plurality of cells 950. In another embodiment, a second gradient magnitude threshold may be defined to yield *2N + 1* bins. Each pixel from an image may be categorized into a bin 952 based on the direction and/or value of a gradient at the pixel as represented in the gradient domain 950.

Additionally, attached to a pixel may be a set of *P* pooling windows 960 with set relative offsets from the pixel as shown in Fig. 14A. Also, a larger window 962, G, may be defined over which the gradients are to be histogrammed. The angular bin *N* from the gradient domain 950 with the largest histogram value (from Fig. 13) is selected as the primary direction. That is, for the example shown in Fig. 13 one of the annular bins 952 is chosen based on the chosen angular bin having the largest histogram value. For each of the pooling windows 960, 2N angular bins may be stored to provide proper oversampling.

All *P* pooling windows may be are tied to the primary direction of the window 962 as determined from the selected angular bin 952 with the largest histogram value. Accordingly, shifting the pooling windows 960 to correspond to the primary direction results in only one degree of freedom being lost over all windows when achieving rotational invariance. Once the primary direction is determined from selecting the angular bin 952 with the largest magnitude of histogram, the window 962 is also used to rotate the configuration compared to the primary direction. Accordingly, the pooling windows 960 shift position and the internal orientation of the pooling windows 960 may be rotated, as shown in Fig. 14B.

In an embodiment, integral images may be used to allow all pooling windows 960 to be calculated to generate a HOG for each pooling window 960. The internal orientation of the pooling windows 960 is rotated by shifting the resulting histogram by a fixed number of angular bins 952 determined for the window *G* 962.

Accordingly, this approach to HOG may provide for rotational invariance. As such, regardless of the orientation of the object to be identified, the HOG produced may be capable of producing useful features associated with the identification of such features. As may be appreciated, for VHR multispectral remotely-sensed Earth imagery, such rotational invariance may be particularly useful given the lack of uniform orientation of features to be identified.

Yet another feature extraction module that may be provided is a Gabor wavelet feature extraction module with rotational invariance. In this regard, while Gabor filters have been utilized in edge detection of images in image processing, the Gabor wavelet feature extraction module described below may provide for rotational invariance, which as described above may be particularly beneficial in the context of VHR multispectral remotely-sensed Earth imagery.

In this regard, the image may be convolved with a Gabor wavelet functions as follows. For the real portion of the function, the following equation may be provided:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Equation 6

Furthermore, the imaginary component of the function may be represented by the function:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Equation 7

In both the functions provided above, the values of x' and y' may be provided as follows:

$$x' = x\cos\theta + y\sin\theta$$

Equation 8

$$y' = -x\sin\theta + y\cos\theta$$

Equation 9

In the foregoing equations, λ represents the wavelength of the sinusoidal factor of the filter. The value of θ represents the orientation of the normal to the parallel stripes of the Gabor function. The value of ψ is the phase offset, σ is the sigma of the Gaussian envelope, and γ is the spatial aspect ratio, which specifies the ellipticity of the support of the Gabor function.

Using the foregoing definitions of the Gabor wavelet functions, a calculation using the module is calculated for each pixel using different values of θ. For instance, a plurality of orientations may be provided for the calculating including, for example, varying values at 22.5 degree increments (e.g., 0 degrees, 22.5 degrees, 45 degrees, 67.5 degrees, 90 degrees, 112.5 degrees, 135 degrees, 157.5 degrees, and 180 degrees). Other intervals may be utilized such as 11.25 degree increments, 45 degree increments, or other appropriate angular

increments. In an embodiment, the calculations made for the various values of θ are made keeping the other variables for the Gabor filter constant. In other embodiments, the calculations may be made at each angular increment using different values for the other variables in the equation to establish a plurality of scales each calculated over the various intervals of θ as described above.

In any regard, once the values for each angular increment for each pixel have been calculated, the results of the calculations for the various intervals may be further processed to achieve rotational invariance. For example, in an embodiment, the sum of the absolute values of the results for all orientations at a given pixel may be calculated. That is, the sum of all calculations for each value of θ may be provided. In another embodiment, the sum of the magnitude for the values for the multiplication of convolution values for the Gabor functions that are perpendicular may be provided. That is, for example, the results of the calculation for the orientation (i.e., a θ value) of 0 degrees may be summed with the result of the calculation for the orientation of 90 degrees, the results of the calculation for the orientation of 22.5 degrees may be summed with the result of the calculation for the orientation of 112.5 degrees, and so on. Further still, the difference between the maximum and minimum values of the magnitudes of the calculations at the various orientations may be calculated for each pixel

In this regard, the Gabor wavelet feature extraction module may output any of the foregoing values based on the calculations made at each orientation. In turn, the output may be rotationally invariant as the orientations calculated may account for rotations of features in the VHR remotely-sensed multispectral Earth images examined. In this regard, the various values described above calculated based on the results of the Gabor filter over the various orientation values may result in a rotationally invariant output for each pixel. As described above, such rotational invariance may be particularly useful in the context of remotely-sensed Earth images.

Another feature extraction module 450 that may be provided is a clustering histogram feature extraction module. The clustering histogram feature extraction module may include computing a histogram of cluster values in a predefined area surrounding a pixel. For example, each pixel of an image may be classified into a predefined set of classes. The classes may be established and the pixels may be categorized according to any appropriate technique. For example, in an embodiment a clustering technique such as k-means clustering using Euclidean or Mahalanobis distance may be used. In one particular embodiment for VHR remotely-sensed

multispectral Earth images, the classes into which pixels are categorized may correspond to land use and/or land cover classifications defined for the image.

In any regard, once the pixels of the image are classified, for each pixel a histogram of cluster values for all pixels within a predefined distance of the pixel is created. The histogram may be built by counting the number of pixels within the predefined distance from the subject pixel that belong to each particular class. In this regard, the histogram may provide details regarding the nature of the surrounding pixels to any given pixel. In this regard, the histograms for the pixels of an image may provide useful information that may be particularly suited to the analysis of VHR remotely-sensed multispectral Earth imagery.

For instance, in the context of classification of land use or in object recognition, a feature that reflects the surrounding neighborhood of pixels may provide useful insight. For instance, it may be recognized that certain conditions may be recognized from the surrounding pixels when classifying the pixel or identifying an object. As an example, a dock may be surrounded by water such that a pixel belonging to a dock may reflect in the histogram created by the clustering histogram feature extraction module that neighboring pixels include those classified as water. As such, when identifying the dock or classifying the pixel, it may be recognized that a dock is generally surrounded by water. As such, the surrounding pixels may provide insight into properties of a given pixel, and such surrounding pixel information may be captured in the histogram created. Other examples may also be provided such as in an application of land use classification where, for example, the histogram for a number of surrounding pixels may provide insight into the type of land use (e.g., a mix of roads, roof and vegetation may indicate a residential area whereas a lack of vegetation may indicate a commercial area). In this regard, the clustering histogram feature extraction module may provide useful information that may be leveraged in further image analysis.

Another feature extraction module 450 that may be provided is a gray-level co-occurrence matrix (GLCM) module. The GLCM is a tabulation of how often different combinations of pixel gray level values occur in an image. As such, the GLCM may provide textural information regarding an image that may be useful in classification techniques. A full description of an embodiment of a GLCM module may be found in Hall-Beyer, The GLCM Tutorial Home Page, Version 2.10 available at http://www.fp.ucalgary.ca/mhallbey/tutorial.htm, the entirety of which is incorporated herein by reference.

In turn, once a GLCM has been generated by the GLCM module, statistics may be generated based on the GLCM. For example, mean, variance, and correlation calculations may be performed. Furthermore, contrast groups may be calculated that are related to contrast use weights related to the distance from the GLCM diagonal. Examples include contrast calculations or "sum of squares variance" for the GLCM, dissimilarity calculations, or homogeneity calculations or "inverse difference moments". Additionally, orderliness groups may be calculated such as angular second moment calculations, maximum probability calculations, or entropy equations.

A number of parameters may be established for creation and use of GLCMs in the GLCM module. For instance, the size of the window over which a GLCM is calculated may be provided as a variable definable by a user. In this regard, the window is preferably large enough to cover features to be identified, but small enough to be localized. For example, calculation of the GLCM in a forest must at least have a window size large enough to identify a tree or calculation of the GLCM in an agricultural field may have a window size roughly correlated to the row spacing of crops planted in the field. Furthermore, the direction of the offset of the GLCM may be definable as may be the bands for which the GLCM is calculated and the measures (e.g., those described above) calculated for the GLCM.

Yet further feature extraction modules 450 may be provided without limitation. An example of a further feature extraction technique that may be utilized include a mean feature approach. In this approach, the mean values for all bands based on a defined kernel are extracted. The kernel may be sized, for example, to be a 3 x 3 pixel kernel although other values may be used. In this regard, the mean feature module may provide smoothing over the kernel.

Another feature extraction module 450 that may be used includes a digital elevation model (DEM) module that may analyze image data in relation to a DEM. In this regard, some land classes may be tied to relatively flat areas (e.g., crops, wetlands, etc.). A such, a module may generate features based on a DEM corresponding to the geography covered by the image. In addition to providing useful information directly related to land classes, reference to DEM data for a portion of the image may influence illumination of various portions of the image that may be accounted for in processing (e.g., identification of shady areas based on elevation profiles).

As may be appreciated, the foregoing feature extraction models 450 may generate data layers for the feature stack 600 that may each be particularly suited for various classification tasks. In this regard, a user developing a model may be capable of selecting one or more of the feature extraction modules 450 for execution during development of a classification model 462. Further still, at least one parameter related to the execution of a feature extraction module 450 may be adjustable by a user to tailor the feature extraction module 450 to a particular application. In this regard, the selection of certain ones of the feature extraction modules 450 and/or customization of parameters related thereto may facilitate improved processing efficiency as unused and/or unhelpful feature data layers may not be generated based on a user's selection of a limited set of the feature extraction modules 450 for use in generation of the classification model 462.

Returning to Fig. 4, the land classification system 405 may also include a classification compute module 460. The classification computer module 460 may generally be operable to receive supervised training data that is analyzed with respect to one or more feature layers from the feature stack 600 including, for example, image data 411 and/or feature data 452 such as in the form of one or more feature layers from the feature stack 600 described above and shown in Fig. 6 that are retrieved from the data store 410. The analysis of the training data in relation to all or a portion of the feature stack 600 may be used to generate a classification model 462 based on the feature stack 600. As may be appreciated from Fig. 4, the classification model 462 may be stored in the data store 410.

The classification model 462 may include a classification structure based on an analysis of training data provided by a user in relation to the image data 411. As the pixels 650 of the image data 411 may be correspondingly provided with feature layers in the feature stack 600, corresponding pixels in the feature stacks 600 may also be used by the classification compute module 460 in connection with the training data. That is, the feature stack 600 may define one or more attributes of a pixel that may be used in classifying the pixel based on the attributes provided to the classification compute cluster 460. Training data may be received from a user (e.g., from the client interface 430 using a user interface as will be described in greater detail below) that is indicative that certain pixels of the image data 411 belong to a particular class. In this regard, training pixels for each of a plurality of predefined classes may be identified. The classification compute module 460 may analyze the training data to generate the model (e.g., using a supervised learning technique) that includes classification rules generated based on the analysis of the training data in view of the feature stack 600. In turn, once

generated the classification model 462 may be used to classify other pixels of the image data 411 for which training data was not provided to classify those pixels using the classification rules of the classification compute module 460 as applied to the feature stack 600 related to those pixels.

The classes into which the classification model 462 classifies pixels may correspond to any appropriate physical property associated with a pixel. For example, as primarily described herein, the classes may correspond to land classification including classes that relate to land classes and/or land use. Examples of land classes may include classes related to the material imaged by a pixel, for example, water, exposed dirt, rock, cropland, grass, forest, tundra, road, roof, asphalt, concrete, etc. Furthermore, land use may correspond to a the manner in which the land is utilized and may include classes corresponding to, for example, human urban growth (e.g., including residential use, industrial use, etc.), agricultural use, etc.

As may be appreciated, classification models 462 may have limited portability or accuracy when classifying images from different geographies and/or images that were acquired at different times (e.g., certain periods of the year corresponding to seasons or the like). In this regard, the classification compute module 460 may be used to generate a plurality of classification models 462 that may each be generated with respect to a particular geographic and/or temporal coverage. For instance, the classification model 462 may be provided with certain classification parameters that may relate to the geography and/or acquisition time/date of the image data 411 used to generate the model 462. In this regard, when the classification compute module 460 applies the classification model 462 to classify image data 411, the metadata for the image data 411 may be used to determine for which image data 411 a given classification model 462 is to be used. For instance, for a given geographic area, a plurality of models 462 may be available corresponding to different seasons. Therefore, metadata of the image data 411 that indicates when the image was acquired may be used to determine the model 462 to be applied to classify image data 411 based on a correlation between a seasonal model 462 and the image acquisition time. Additionally, geographic descriptors for image data 411 may be used to determine an appropriate model 462 for application to the image data 411 to classify pixels of the image data 411 based on the corresponding geographic area related to the pixels.

Furthermore, it may be appreciated that the generation of classification models 462 by users may be done in a collaborative manner. That is, as described above, the client communication module 440 may be operable to communicate with a plurality of instances of a client interface 430 that may, for example, correspond

to disinbuted user terminals over a network. In this regard, different ones of the users at different client interfaces 430 may be operable to access the classification models 462 in the data store 462 to iteratively develop the model 462. That is, a user may build on an existing model 462 generated by another user or use another model 462 as a basis for generation of a new model 462. Accordingly, a different user may modify and/or supplement training data, modify and/or add data layers utilized in the feature stack 600, or other modify parameters to modify an existing model 462. Additionally, as the models 462 may be particular to a geographic area, different users utilizing different client interfaces 430 may select different geographies for which to build a classification model 462. In this regard, the potential collaborative nature of the land classification module 405 may allow for a iterative, piecemeal approach to a global land classificaiion layer.

One potential approach to creation of the classification model 462 may include the use of a decision tree approach to supervised machine learning. The basic algorithm for such a decision tree may be a greedy algorithm optimization problem. The approach may include constructing a top-down recursive tree that is optimized in an divide-and-conquer manner. That is, at the initiation of the algorithm, all training examples (i.e., each pixel having attributes derived from the feature stack 600) may be provided at the root such that each example constitutes a node of the decision tree. Attributes may be categorical. If attributes are continuous-valued, they may be discretized in advance or all thresholds of the attributes may be considered. Examples from the training data (i.e., positively identified pixels belonging to a class or negatively identified pixels that do not belong to one or more classes) may be partitioned recursively based on selected attributes. That is, examples with common attributes may be partitioned to develop a decision or branch of the decision tree for classification based on the attribute subject to the partition. Test attributes may be selected on the basis of a heuristic or statistical measure such as, for example, information gain as will be described in greater detail below. In turn, conditions for ceasing the partitioning may be provided such as, for example, when all samples for a given node belong to the same class or when there are no remaining attributes for further partitioning. In this latter regard, majority voting may be provided for classifying the resulting examples remaining in a partition. One example of an output of a decision tree classification model 462 that may be provided as an output 800 of a decision tree approach by the classification compute module 460 is shown in Fig. 11.

In Fig. 11, the output 800 includes a plurality of decision levels 810a-810f. Each line 820 of the output 800 may include a decision based on one or more features or attributes of the pixel being classified. For

instance, at level 810a, a near infrared 1 band (N1) of the pixel to be classified is scrutinized to determine if the gray value for the pixel is greater than or less than or equal to a value of 191. In the case where the gray value for the N1 band is greater than a value of 191, the classification compute module 460 moves to the next level, level 810b, to continue the analysis. For instance, at the next level 810b, for pixels with a gray level in the N1 band of greater than 191, it may be determined if a ratio of the N1 band to the yellow band (Y) is less than or equal to the value of 0.078947. Depending on the value of the pixel being evaluated, the classification compute module 460 may move to the next level 810c at a row corresponding to the value of the ratio of the N1 and Y band. In turn, features of the pixel may be analyzed in this branched analysis method until a determination of a class for the pixel is determined.

As described above, when deciding when to split examples or partition examples based on an attribute in the tree based on a decision may be determined using one of a number of approaches for determining how and when to split an attribute as a decision in the tree structure 800. For example, an information gain approach, a gain ratio approach, or a Gini index approach may be provided. For the information gain approach, a probability value p_i may be defined as the probability that an arbitrary example in D (i.e., the set of all pixels of the training data) belongs to class C_i (i.e., a given class). In this regard, the value for p_i may be estimated as:

$$p_i = \frac{|C_{i,D}|}{|D|}$$

Equation 10

where $C_{i,D}$ is the total number of examples in the class C_i. Accordingly, the expected information needed to classify an example in D prior to a split may be provided as:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

Equation 11

In turn, the information needed to classify D after a split may be calculated as:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Equation 12

In turn, the information gained by partitioning examples based on an attribute A may be calculated as:

$$Gain(A) = Info(D) - Info_A(D)$$

Equation 13

In turn, the information gain when choosing splits may be optimized over the classes or a minimum threshold for additional information gain may be established such that further partitioning of examples in the tree for an attribute that add information below the threshold may not occur.

Another approach to determining when to split examples in the tree based on an attribute *A* (e.g., from the feature stack 600) may be the ratio gain approach. In this approach, the information gained by making a split may be defined as:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times log_2\left(\frac{|D_j|}{|D|}\right)$$

Equation 14

In turn, the gain ratio for splitting based on attribute *A* may be defined as:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

Equation 15

This ratio may be maximized at each step such that the greatest contribution to the information may determine which attribute is used for each subsequent split in the tree to partition examples based on a attribute *A*.

In the context of continuous-valued attributes, the best split point for an attribute *A* may also be determined. In this regard, the values of *A* may be sorted in increasing order. Typically, the midpoint between each pair of adjacent values is considered as a possible split point of the attribute or:

$$SP_A = \frac{(a_i + a_{i+1})}{2}$$

Equation 16

The point with the minimum expected information requirement for *A* is selected as the split point for *A* where *D1* is the set of examples in *D* satisfying the requirement of *A* less than or equal to the split point and *D2* is the set of examples in *D* satisfying the requirement of *A* greater than the split point.

Additionally a Gini index approach to a split may be taken. In this approach, a Gini index (*gini(D)*) may be defined as:

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

Equation 16

where p_j is the relative frequency of class j in D. A reduction in impurity may be calculated as:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

Equation 17

In this regard, it may be appreciated that the foregoing approaches to selection measures for determining how to partition examples based on an attribute or, in other words, where to insert a decision into the tree, may have certain compromises. For example, the information gain approach may be biased towards multivalued attributes. The gain ratio approach tends to prefer unbalanced spits in which one partition is much smaller than the others. The Gini index is also biased to multivalued attributes and may have difficulty in determining attribute selection when the number of classes is large. As such, the Gini index tends to favor tests that result in equal-sized partitions and purity in both partitions.

In other embodiments, other attribute selection methods may be employed in the classification compute module 460. Examples include, but are not limited to, a CHAID algorithm, a C-SEP algorithm, a G-statistic algorithm, a minimal description length (MDL) algorithm, a multivariate split algorithm (e.g., a CART algorithm) or other appropriate approach.

Additionally, "pruning" approaches to the decision tree may be provided to improve the resulting classification model 462 based on creation of a decision tree. For instance, a scenario where a generated tree overfits the training data may be generated where too many branches are provided that reflect anomalies due to noise or outlying data values. In this regard, prepruning or postpruning techniques may also be used. In prepruning, tree construction is halted early by not spitting a node if the result would provide a measure below a threshold (e.g., providing a minimum information gain or the like). In postpruning, branches of the decision tree may be removed once the decision tree is generated. Additionally, in the decision tree processing, other enhancements may be provided such as attribute construction or attributing attributes for missing attribute values.

Another approach to generation of a classification model 462 by a classification compute module 460 includes use of an artificial neural network. In a neural network, the inputs to the network may correspond to the

attributes measured from each training value. These attributes may be weighted and fed together into a hidden layer. In this regard, during the learning process, the network may adjust the weights to predict the correct class label of the training examples. In turn, the network may perform nonlinear regression that approximates a function that classifies the training examples.

Still another approach to machine learning that may be used by the classification compute module 460 to generate a classification model 462 is the use of a support vector machine (SVM). The SVM uses a nonlinear mapping to transform original training data into a higher dimension. In the new dimension, the SVM determines a linear optimal separating hyperplane or "decision boundary." With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can be separated by a hyperplane. The SVM finds the hyperplane using support vectors and margins.

Furthermore, other appropriate machine learning approaches may be additionally or alternatively used to generate classification models 462. Examples include, but are not limited to, instance-based learning, maximum likelihood approaches, genetic algorithms, logistic regression, active learning, transfer learning, classifier ensembles, bootstrap aggregation, boosting (e.g., using the Adaboost algorithm), Random Forest algorithms, or other appropriate machine learning techniques without limitation.

In this regard, it may be appreciated that for different models 462 or for different portions of a given model 462, different machine learning approaches may be utilized. As such, the use of machine learning approaches may be parameterized such that a user may select the approach and/or variable values related to the approach to tailor the machine learning process to the model or model portion to be constructed. In this regard, approaches to model evaluation may be provided to assist in selecting and/or tailoring a model 462 or the approach used to generate the model 462. As such, a model evaluation module 490 may be provided.

Accordingly, the model evaluation module 490 may be operable to evaluate classification modules 462 using a test set of examples that may be generated based on training data received by a user. For example, a test set of class-labeled examples instead of training data may be provided to assess the accuracy of the model 462 and/or evaluate the approach used to generate the model 462. Thus, methods may be executed to evaluate a classifier's accuracy or compare classifiers. Accordingly, a classifier accuracy value may be defined that relates to the recognition rate of the model 462. The classifier accuracy value may be the percentage of test set examples that are correctly classified. The classifier accuracy value may be defined as:

$$Accuracy = \frac{(TruePositives + TrueNegatives)}{AllTestSet}$$

Equation 18

Furthermore, an error rates may be provided as:

$$Error\ Rate = 1 - Accuracy$$

Equation 19A

or

$$Error\ Rate = \frac{(FalsePositives + FalseNegatives)}{AllTestSet}$$

Equation 19B

Additionally, one or more precision measurements may be provided. These measurements may be defined as:

$$Precision = \frac{TruePositive}{(TruePositive - FalsePositive)} - Consumer'sAccuracy$$

Equation 20

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} - Producer'sAccuracy$$

Equation 21

$$FMeasure = 2 \times Precision \times \frac{Recall}{Precision + Recall}$$

Equation 22

Additionally, a receiver operative characteristic ROC may be calculated as:

$$TruePositiveRate = Recall$$

Equation 23

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

Equation 24

Also, a kappa coefficient may be calculated that takes into account the probability of a chance or random agreement. The kappa coefficient may be provided as:

$$\kappa = \frac{(\Pr(a) - \Pr(e))}{(1 - \Pr(e))}$$

Equation 25

where Pr(a) is the observed agreement (or accuracy) and Pr(e) is the probability of a chance agreement.

The generation of the test set of data may be provided in a number of ways. For example, in a holdout method, training data is randomly partitioned into two sets for model construction and accuracy estimation, respectively. For instance, the training data may be portioned randomly such that, for example, two thirds of the training data belongs to the training set used for model construction and one third of the training data belongs to the test set for accuracy estimation. Another approach may be cross-validation where the training data is randomly partitioned into *k* mutually exclusive subsets each of approximately the same size. For example, ten is a common value used for *k*. At an iteration *i*, one of the subsets D_i is used as a test set and all others are used as the training set.

One variation on the cross validation approach is for sets of small size, where for each iteration, one of the samples is left out of the training set and used for testing, which is known as the "leave-one-out" approach. Another approach to cross validation is the stratified cross validation approach where each class distribution in each set D_i is approximately same as that in the full data set *D*. Additionally, a bootstrap approach may be provided that samples the training examples uniformly with replacement.

Accordingly, confidence intervals may be calculated for a model 462. For instance, an estimation of the confidence interval may be provided using a t-test. In this regard, if only one test set is available the comparison may be pairwise. Accordingly, for the *i*-th round of cross validation with *k=10*, the same cross partitioning is used to obtain the error rate for a first model (M_1) and a second model (M_2) to obtain $\overline{err}(M_1)$ and $\overline{err}(M_2)$, which are the average of the error rate over ten rounds for each model respectively. A formula to compute a t-test with t-statistic with k1 degrees of freedom may be provided as:

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{\frac{var(M_1 - M_2)}{k}}}$$

Equation 26

where

$$var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))]^2$$

Equation 27

In the case where two sets of test data are available, a non-paired t-test may be used where:

$$var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}}$$

Equation 28

and k_1 and k_2 are the number of cross-validation samples used for M_1 and M_2 respectively.

In turn, a t-distribution table for critical values may be used to determine significance levels between M_1 and M_2. For instance, a significance level of 0.05 or 5% means M_1 and M_2 are significantly different for 95% of the population. A confidence limit may be defined as $z = sig/2$. Accordingly, to determine if M_1 and M_2 are significantly different, t may be computed and a significance level established (e.g., sig = 5%). In this regard, a t-distribution table may be consulted to find a t value corresponding to k-1 degrees of freedom (e.g., 9 in the case where $k = 10$. For a symmetric t-distribution, the upper value is typically used. In this regard, the value for the confidence limit may be looked up. If $t > z$ or $t < -z$, then the mean error rates of M_1 and M_2 are the same and there is a statistically significant difference between M_1 and M_2. Otherwise, it may be concluded that any difference between the models is due to chance or randomness.

The land classification module 405 may also include a post classification processing module 480. In this regard, once image data 411 has been classified into one of the plurality of land classes by the classification compute module 460, the classified image data 411 may be further analyzed to improve classification accuracy. In this regard, one or more post classification rules may be generated (e.g., collaboratively via a plurality of users) to modify classifications of the image data 411 after the data has been classified by the classification compute module 460.

In this regard, post classification rules may include, for example, a minimum mapping unit rule, a "surrounded by" rule, a topological relation rule, a majority analysis rule, and/or a smoothing rule. However, other post classification processing techniques may be applied without limitation. The minimum mapping unit rule may dictate that a cluster of classified pixels (e.g., where at least one pixel of a common classification borders at least one other pixel of common classification to form a group of adjacent similarly classified pixels) have a minimum unit size. For instance, certain classes (crops, tidal flats, wetlands, etc.) may have a larger minimum unit size required than other classes (e.g., roads, roofs, shadows, trees, etc.). This may reflect an understanding that certain classes are likely to be of a minimum threshold size. In this regard, if the minimum

unit size is not attained, the pixels may be reclassified. For instance, a group of pixels classified as tidal marsh that does not exceed the minimum unit size may be reclassified based on, for example, surrounding pixels classifications or a revision to the classification model 462 used to classify the pixels originally.

In another example, a topological relations rule may be provided. In this regard, pixel classifications of a certain size (e.g., less than the minimum unit size) that are completely or partially surrounded by another class of pixels may be dissolved to become a part of the surrounding pixel classification. As an example, in an image with a group of pixels grouped as tidal flat that is completely surrounded by pixels classified as road may be dissolved to become classified as road in accord with the surrounding pixels. Various heuristics may be provided for determining how the pixels may be changed in relation to the topological relation rule including, for example, dissolving pixels into the class of pixels that includes the largest border with the pixels to be dissolved, dissolving pixels into the largest group of similarly classified pixels adjacent to the pixels to be dissolved, reclassification with the model 462 originally used to classify the pixels while omitting the original class assigned as an option for classification, etc. In this regard, the topological relations rule may include parameters related to the types of classes subject to the rule, the size of the group of classified pixels to which the rule is applied, the identity of the pixels into which pixels are to be dissolved, etc.

Another post classification processing module 480 may include a majority processing rule. In the majority processing rule, a pixel for a given class is reclassified if surrounding pixels in a kernel surrounding a given pixel belonging to another class. For instance, a pixel may be reclassified if all other pixels in the kernel belong to another class or if a majority of pixels in the kernel belong to the other class. The majority processing rule may include majority analysis processing with kernel sizes determined per class. For instance, larger kernel sizes may be provided for certain classes (e.g., crops, tidal flats, grass, wetlands, etc.) and smaller kernel sizes may be provided for other classes (e.g., roofs, roads, shadows, trees, etc.). In this regard, the majority processing rule may smooth boundaries and dissolve long regions. Additionally or alternatively, other smoothing techniques may be applied without limitation to further smooth the classification boundaries of pixels using the post classification processing module 480. Again, the post classification processing module 480 may be selectively applied by a user such that various processing may be iteratively conducted by different users selecting a different one or more post classification processing rules for application after classification of image data 411 using a classification model 462.

As may further be appreciated from Fig. 4, the land classification module 405 may include a database management module 470. The database management module 470 may act to direct interaction between the land classification module 405 (e.g., various components thereof including the image data server 420, the feature extraction module 450, and/or the classification compute module 460) and the data store 410. This may be particularly relevant in the case where a plurality of client interfaces 430 are operable to access and/or utilize the land classification module 405 simultaneously.

With further reference to Fig. 5, an embodiment of a method 500 for land classification (e.g., utilizing the land classification module 405) is depicted. The method 500 may begin with image acquisition 502. As described above, the image acquisition 502 may include any appropriate means for acquiring image data 411 for analysis. In one embodiment, the image acquisition 502 may include acquiring a plurality of bands of spectral data for an image using a satellite imaging system 415. The data acquisition 502 may also include preprocessing of image data 411 such as, for example, orthorectification, atmospheric correction, normalization, etc.

In any regard, the image data 411 acquired from the image acquisition 502 may undergo feature extraction 504. As described above, feature extraction 504 may include any processing of the image data to generate data layers for the image data 411 including, for example, extracting individual spectral bands, performing comparative calculations using the spectral bands, generating textural or morphological data, and/or any one or more of the above described feature extraction modules without limitation. In this regard, the feature extraction 504 may include selection of one or more feature extraction modules 450 to be applied to the image data 411 and/or selection of one or more parameters related to a feature extraction module 450 as discussed above. In any regard, the feature extraction 504 may result in generation of a plurality of data layers for the image data 411 (e.g., in the form of feature data 452).

The method 500 may also include receiving supervised training data 506. As stated above, the training data may be received from a client interface 430 by receiving data from a user that includes an indication of various pixels that belong to one or more class. As described above, the classes may be generated by the user such that the user may generate a class and provides training pixels belonging to the class or may comprise predetermined classes for which the user provides training pixels. The user may provide the training data by way of a client interface 430, which may include a user interface for receiving the training data.

It may be appreciated that the feature extraction 504 may be relatively taxing on computational resources. In this regard, the feature extraction modules 450 used to generate a particular classification model 462 may first be applied to produce a feature stack 600 only for those portions of the image data 411 for which training data has been received. That is, the feature extraction 504 may be carried out only to pixels for which training data has been provided so that the feature stack 600 provided to the classification compute module 460 corresponds to only the trained pixels. In turn, the model 462 may still be generated. Thus, the computational resources needed to construct the model 462 may be minimized as feature data relating only to those portions of the image data 411 used to generate the module may be generated. In turn, upon application of the model 462 to a larger portion of the image data 411 (e.g., the full scene including pixels for which training data has not been provided), the feature extraction module 450 may execute to produce the feature stack 600 for the full imagery. Thus, the feature data may be computed on an "as needed" basis to conserve processing resources. Furthermore, it may be appreciated that many feature extraction modules 450 may include highly parallel computations that may be particularly suited for execution on a GPU.

With further reference to Figs. 7-10B, embodiments of screen shots of screens of a user interface 700 are shown. With respect to Fig. 7, image data 411 may be displayed to a user. For example, a human visible image comprising red, blue, and green band information may be displayed. In turn, tools may be provided to the user by way of the user interface 700 to select pixels that belong to a class. For instance, as shown in Fig. 7, a user may have provided a number of regions 710a-710c corresponding to different classes to be identified from the image data 411. For example, regions 710a may correspond to a region identified by the user corresponding to a class "water." Region 710b may correspond to a region identified by the user corresponding to a class "concrete." Region 710c may correspond to a region identified by the user corresponding to a class "roof." In this regard, it may be appreciated that predefined classes may be provided for which users are tasked to identify positive and/or negative examples of pixels belonging or not belonging to a class. Additionally or alternatively, a user may identify regions of pixels to define a new class, which may or may not be defined by a user.

For instance, a region of pixels for which the user is not sure of the class may identify the pixels belonging to the undefined class such that similar pixels may still be classified even if the class definition is to be determined later. Furthermore, while not shown in Fig. 7, a user may also provide negative examples in the form of a region identified as not belonging to one or more categories. Further still, for a given class, multiple

instances of training data for the class in different conditions may be provided. For example, for the class "roof," training data may be provided for "shaded roof" corresponding to pixels of a roof that are in shade and "unshaded roof" for corresponding to pixels of a roof that are in direct sun. In this regard, it may be that pixels identified as "shaded roof" and "unshaded roof" may each be attributed to a parent class of "roof" as the user may not be interested in divisions between portions of roof that are shaded or unshaded. That is, while individual pixels corresponding to shaded roofs and unshaded roofs may be separately identified using the separately provided training data, the separately identified pixels may be aggregated into a single class, i.e., roof, without regard to whether shaded or unshaded. This may be particularly useful in the context of VHR remotely-sensed multispectral imagery where cloud cover or the like is present to identify training data for shaded and unshaded pixels from a single class. Accordingly, even in identifying pixels from data where cloud cover may be present, an accurate classification of pixels may be made despite potential variations in what pixels are in shade and which are not.

Fig. 8 depicts another screen of a user interface 700. As may be appreciated, the user interface 700 may include a plurality of windows. For instance, human visible image data 411 may appear in a first window 720 and a classification map overlay 730 may appear in a second window 722. In this regard, the user may be capable of viewing image data 411 and the classification overlay 730 side by side. Accordingly, in an embodiment, the method 500 of land classification may be performed for a portion of image data 411 for verification of the sufficiency of training data, a classification module 462, or other parameters. In this regard, as may be appreciated from Fig. 8 depicting the user interface 700 the image data 411 presented to the user may comprise a limited portion of a larger scene to which a completed classification regime (e.g., including a selection of feature extraction module, classification model and/or post classification rules to be applied) is to be used. In this regard, when applied to a large scene, the processing time required to complete the processing may be significant. As such, the method 500 may be performed on a more limited portion of the image data 411 so that the user may perform real time verification of a classification regime prior to executing the classification regime on an entire scene to be classified by viewing the image data 411 and classification overlay 730 side by side.

The user interface 700 may also include navigational tools 740 that may allow a user to manipulate the portion of the image data 411 and/or a corresponding portion of the classification overlay 730. For instance, panning, zooming, or other tools (e.g., a selector for which spectral bands, classifications, etc. should be

displayed) may be provided. The zooming functionality may be especially important when selecting regions 710 of training data. This is especially evident comparing the images of Figs. 7 and 8 where it is clear that the regions 710 may correspond to relatively small areas of the image that are more easily defined at high zoom levels.

The user interface 700 may also include a region selector tool set 732. The region selector tool set 732 may allow a user to change the manner and/or shape of a region to be added to the image data 411 to define training data. For instance, a polygon, rectangle, or point tool may be provided that allow for differently shaped regions to be created. A "stamp" tool may be provided that allows a region of a fixed size to be created. Once the "stamp" region is added to the image data 411, the region may be resized. A "magic wand" tool may also be provided that selects a region based on similarities of adjacent pixels (e.g., based on standard deviation, max area parameters, or the like) as is known in the art. Additionally, the region selector tool set 732 may include histogram-based thresholding to subselect pixels from a polygon or rectangle.

The user interface 700 may also have project management functions. These functions may allow users to load, save, and create new classification models 462. The project management functions may also provide current job status and allow users to set preferences for a model or for global model creation. The project management functions may also allow a user to upload and/or select image data 411 for which to create a model 462.

With further reference to Fig. 9, it may be appreciated that the side by side viewing of image data 411 in a first window 720 and a classification overlay 730 in a second window 722 may include regions 710 of identified training data that are identified in the first window 720. In this regard, regions 710g may be shown in a first color indicative of an association with a first class. In turn, a first class designator color 720g' may be presented in the class overlay 730 that corresponds with the first color of the region 710g. In a similar regard, regions 710h may be shown in a second color indicative of an association with a second class. In turn, a second class designator color 720g' may be presented in the class overlay 730 that corresponds with the second color of the region 710g. In this regard, a user may review the class overlay 730 generated based on the regions 710g, 710h of training data to evaluate the accuracy of the class overlay 730. As such, the user may delete, add, or modify the regions 710g, 710h to improve the accuracy of the class overlay 730.

Fig. 10A and 10B depict various region metrics 750 that may provide information to the user about the regions that have been created and properties related thereto. For instance, in Fig. 10B, a set of region metrics 750 is includes that provides each class name, the color designated to the class, the number of pixels of training data received for the class, and the number of polygons created to define the total number of pixels of training data. In Fig. 10B, a histogram 752 is depicted that displays the relative number of pixels of training data that has been provided for each particular class available. In this regard, the region metric 750 and histogram 752 may assist a user in determining which of the classes would benefit from additional training data.

In any regard, once the training data has been received, the method 500 may include generation 508 of a classification model 462 based on the training data. As described above, the generation of the classification model 462 may include performing a supervised machine learning technique with respect to the training data to generate a classification model 462 for application to image data 411 to classify pixels thereof. Examples of supervised machine learning algorithms that may be applied to generate the classification model 462 may include any of the foregoing approaches described. In any regard, the resulting classification model 462 may include a classification process that may be based on one or more portions of image data 411 and/or feature data 452 to determine for a given pixel of image data 411 a class into which the pixel should be classified. As such, the method 500 may further include classification 510 of image data 411 to classify pixels thereof. The image data 411 may include a complete scene of satellite image data or a more limited portion (e.g., as defined by a user). The method 500 may also include post classification processing 512 that may include application of one or more of the post classification rules described above.

In an embodiment, the method 500 may be performed for a portion of image data 411 for verification of the sufficiency of training data, a classification module 462, or other parameters. In this regard, as may be appreciated from the foregoing discussion of the user interface, a portion of the image data 411 may be presented to a user. In this regard, the image data 411 presented to the user may comprise a limited portion of a larger scene to which a completed classification regime (e.g., including a selection of feature extraction module, classification model and/or post classification rules to be applied) is to be used. In this regard, it may be appreciated that when applied to a large scene, the processing time required to complete the processing may be significant. As such, the method 500 may be performed on a more limited portion of the image data 411 so that

the user may perform real time verification of a classification regime prior to executing the classification regime on an entire scene to be classified.

The method 500 in turn may include classifying 510 pixels of the image into a class based on a generated or selected classification model 462. That is, attributes of a given pixels that, for example, may be taken from the data stack 600 may be analyzed with respect to the classification model 462 to determine which class the pixel is to be classified. Further still, the method 500 may include post classification processing 512 including application of one or more post classification processing rules as described above.

While the invention has been illustrated and described in detail in the drawings and foregoing description, such illustration and description is to be considered as exemplary and not restrictive in character. For example, certain embodiments described hereinabove may be combinable with other described embodiments and/or arranged in other ways (e.g., process elements may be performed in other sequences). Accordingly, it should be understood that only the preferred embodiment and variants thereof have been shown and described and that all changes and modifications that come within the spirit of the invention are desired to be protected.

What is claimed is:

1. A land classification system operable for analysis of very high resolution (VHR) remotely-sensed multispectral Earth imagery, comprising:

an image store containing image data corresponding to VHR remotely-sensed multispectral Earth images;

at least one feature extraction module in operative communication with the image store, the feature extraction module being operable to produce feature data regarding at least a portion of image data;

a feature stack comprising the image data and the feature data;

a client interface operable to receive training data regarding at least one or more pixels of the image data from a user regarding at least one class to which the one or more pixels belong; and

a classification compute module operable to generate a classification model at least in part based on a portion of the feature stack corresponding to the one or more pixels and the training data, wherein the classification model relates to classification of pixels of image data into one or more classes.

2. The system of claim 1, wherein the classification compute module is operable to apply the classification model to the image data to classify pixels of the image data into the one or more classes.

3. The system of claim 2, wherein the classification compute module classifies pixels of the image data based on an analysis of the feature stack in relation to the classification model.

4. The system of claim 3, further comprising:

a post classification analysis module operable to reclassify at least one pixel of the image data based on at least one post classification rule.

5. The system of claim 4, wherein the post classification rule comprises at least one of a minimum mapping unit rule, a surrounded by rule, a topological relation rule, a majority analysis rule, or a smoothing rule.

6. The system of claim 1, wherein the VHR remotely-sensed multispectral Earth image data comprises spectral band data corresponding to at least 8 multispectral bands.

7. The system of claim 6, wherein the multispectral bands collectively range from at least about 300 nanometers in wavelength to at least about 1100 nanometers in wavelength.

8. The system of claim 7, wherein the multispectral bands collectively range from at least about 300 nanometers in wavelength to at least about 2400 nanometers in wavelength.

9. The system of claim 8, wherein the multispectral bands comprise a plurality of short wave infrared (SWIR) bands.

10. The system of claim 9, wherein the plurality of SWIR bands range from at least about 1100 nanometers in wavelength to at least about 2400 nanometers in wavelength.

11. The system of claim 1, wherein the feature stack comprises a plurality of data layers, wherein each data layer includes different feature data for each pixel of the image data.

12. The system of claim 11, wherein the feature stack includes at least one spectral feature layer, at least one morphological feature layer, or at least one textural feature layer.

13. The system of claim 12, wherein the at least one spectral feature layer comprises data values for each pixel of the image data based on spectral band data collected by a remote image acquisition platform.

14. The system of claim 13, wherein the remote image acquisition platform comprises a satellite imaging system.

15. The system of claim 13, wherein at least one spectral feature layer comprises a relative measure between at least two spectral band data layers.

16. The system of claim 12, wherein the at least one morphological feature layer comprises for each given pixel of the image data information based on the arrangement of adjacent pixels relative to the given pixel.

17. The system of claim 12, wherein the textural feature layer comprises information for a given pixel based on a spatial distribution of tonal variations within one or more spectral band data layers relative to a given pixel.

18. The system of claim 1, wherein the client interface comprises an internet browser-based client operable to provide a user interface for receiving the training data from a user.

19. The system of claim 18, wherein the client interface comprises a plurality of distributed client interfaces operable to receive training data from a plurality of users, wherein the classification model is at least partially based on training data received from a plurality of users.

20. The system of claim 1, further comprising a plurality of feature extraction modules each operable to generate a feature data layer of the feature stack.

21. The system of claim 20, wherein one or more of the feature extraction modules are selectively applicable to image data by a user to generate feature data layers.

22. The system of claim 21, wherein at least one of the feature extraction modules comprises a fractal dimension feature extraction module.

23. The system of claim 22, wherein the fractal dimension feature extraction module is operative to generate feature data by:

identifying an image of a predetermined size of pixels;

dividing the image into a plurality of abutting square windows having a side dimension less than the predetermined size of the image;

constructing for each plurality of abutting square windows a column having a height as represented in increments of the side dimension of the abutting window, wherein the height is determined such that the ratio of the predetermined size of the image to the side dimension of the abutting windows equals the ratio of a maximum pixel value for a given multispectral band of the image to the height, wherein the squared side dimension and the height represent a plurality of boxes whose volume is defined by increments of the side dimension;

determining, for each abutting window a box from the plurality of boxes in which a maximum pixel value and a minimum pixel value occur;

calculating, for each of the plurality of abutting windows, the number of boxes separating the box in which the maximum pixel value is disposed and the box in which the minimum pixel value is disposed; and

summing the number of boxes separating maximum and minimum pixel levels over all of the plurality of abutting windows to generate a summed value;

repeating each of the constructing, determining, calculating, and summing operations for all possible values of the side dimension of the plurality of abutting windows such that each side dimension is less than or equal to half of the predetermined size of the image and at least greater than three pixels in length;

plotting each summed value generated in the repeating step to determine a slope of the summed values, wherein the slope comprises the fractal dimension of the image.

24. The system of 23, wherein the image of the predetermined size is a subset portion of a larger image.

25. The system of claim 24, wherein the fractal dimension calculated for each subset portion of the larger image is used to generate a fractal map comprising an aggregated mapping of the fractal values of each subset portion over the larger image.

26. The system of claim 25, wherein a fractal map is generated for each spectral band of a multispectral image.

27. The system of claim 26, wherein the fractal dimension is at least partially dependant upon the spectral band, wherein each fractal map is independently generated of other fractal maps of the other spectral bands of the plurality of spectral bands.

28. The system of claim 21, wherein at least one of the feature extraction modules comprises a total variational (TV) feature extraction module.

29. The system of claim 28, wherein the TV feature extraction module comprises a multispectral filtering module using a plurality of spectral bands of an image to generate a filtered image corresponding to the image.

30. The system of claim 29, wherein the number of spectral bands used in the filtering operation comprises at least 8 multispectral bands.

31. The system of claim 29, wherein the filtering operation comprises minimizing a function including at least a first term representing the difference between the original image and the filtered image and at least a second term representing the spatial homogeneity of the filtered image.

32. The system of claim 31, wherein the first term and the second term include a multivalued vector for each pixel representative of the image or filtered image , respectively, wherein the multivalued vector values include a component for each row, column, and multispectral band in the image.

33. The system of claim 32, wherein at least one of the first term or the second term include an L1 norm.

34. The system of claim 33, wherein the second term is weighted by a smoothing factor.

35. The system of claim 31, wherein the minimization of the function is solved using a Split Bregman technique.

36. The system of claim 35, wherein the Split Bregman technique includes Gauss-Seidel updates.

37. The system of claim 36, wherein the minimization is solved by execution on a graphics processing unit (GPU).

38. The system of claim 21, wherein at least one of the feature extraction modules comprises a rotational invariant histogram of gradients (HOG) feature extraction module.

39. The system of claim 38, wherein the rotational invariant HOG feature extraction module is operative to generate feature data by:

establishing a plurality of angular bins for an image in a gradient domain;

histogramming gradients from the image with respect to the plurality of angular bins;

selecting based on the histogramming the angular bin with the largest histogram value to define a primary direction

setting a plurality of pooling windows with relative offsets from a pixel of interest;

defining a gradient orientation window encompassing the plurality of pooling windows;

configuring the gradient orientation window to align the configuration of the gradient orientation window relative to the primary direction;

shifting the positions of the pooling windows relative to the gradient orientation window based on the configuring; and

rotating the internal orientation of the pooling windows based on the configuring.

40. The system of claim 39, wherein the shifting includes shifting the pooling windows relative to the gradient orientation window by a number of bins determined in the selecting step, and wherein the rotating includes shifting the internal orientation of the pooling windows by a number of bins determined by the rotating step.

41. The system of claim 40, wherein a HOG is calculated relative to each of the shifted and rotated pooling windows.

42. The system of claim 41, wherein the calculating includes using an integral image to calculate the HOG feature for each pooling window.

43. The system of claim 21, wherein at least one of the feature extraction modules comprises a Gabor wavelet feature extraction module.

44. The system of claim 43, wherein the Gabor wavelet feature extraction modules is operative to generate feature data by:

applying a plurality of Gabor wavelet functions to an image, wherein each of the plurality of Gabor wavelet function includes a different orientation; and

computing at least one feature based on at least two of the plurality of Gabor wavelet functions.

45. The system of claim 44, wherein the at least one feature comprises at least one of a sum of the absolute values of a first Gabor wavelet function of the plurality of Gabor wavelet functions and a second Gabor wavelet function of the plurality of Gabor wavelet functions, a sum of the magnitude of values of the product of each perpendicularly oriented Gabor wavelet function of the plurality of Gabor wavelet functions, and the difference of the maximum and minimum values of magnitude of all of the plurality of Gabor wavelet functions for all image data.

46. The system of claim 45, wherein for each orientation of the plurality of Gabor wavelet functions, a plurality of different Gabor wavelet functions with a different scale are calculated.

47. The system of claim 46, wherein the different scales comprise varied values for at least one of a wavelength of the Gabor wavelet function or an envelope of the Gabor wavelet function.

48. The system of claim 47, wherein each of the plurality of Gabor wavelet functions are applied to each pixel of each multispectral band of an image comprising a plurality of multispectral bands.

49. The system of claim 48, wherein the at least one feature is determined independently for each of the plurality of bands.

50. The system of claim 49, wherein the at least one feature is determined with respect to all of the plurality of bands.

51. The system of claim 21, wherein at least one of the feature extraction modules comprises a clustering histogram feature extraction module for extraction of features based on clustered and classified spectra values.

52. The system of claim 51, wherein the clustering histogram feature extraction module is operative to generate feature data by:

assigning each pixel of an image into one of a plurality of a predefined classes;

identifying a kernel surrounding each pixel containing a plurality of others pixels from the image, wherein each of the plurality of other pixels have been assigned into one of the plurality of predefined classes;

building a histogram for each pixel based on the number of other pixels in each of the plurality of predefined classes relative to the pixel for which the histogram is built.

53. The system of claim 52, wherein the plurality of predefined classes correspond to identified land classifications based on at least one of land use or land class.

54. The system of claim 53, wherein the pixels are assigned into a predefined class based on the radiometric properties of a pixel used in a clustering technique.

55. The system of claim 54, wherein the clustering technique comprises at least one of k-means clustering, a Euclidean distance approach, or a Mahalanobis distance approach.

56. The system of claim 1, wherein the classification compute module is operable to utilize at least one machine learning algorithm to generate the classification model.

57. The system of claim 56, wherein the classification compute module is operable to utilize a plurality of machine learning algorithms to generate the classification model.

58. The system of claim 56, wherein a user is operable to specify at least one of a plurality of machine learning algorithms for use in generation of the classification model.

59. The system of claim 1, wherein the classification model comprises at least one model parameter corresponding to at least one of a geographic region or a temporal period.

60. The system of claim 59, wherein the classification computation module model is operable to determine the classification model for classification of image data based on the at least one model parameter and image data metadata.

61. The system of claim 1, wherein the one or more classes correspond to at least one of a land cover, a land use, or temperature.

62. The system of claim 1, further comprising:

a model evaluation module operable to provide quantifiable evaluation data regarding the classification module based on performance of the classification module with respect to a portion of training data received by a user.

63. The system of claim 62, wherein the model evaluation module is operable to generate a classifier accuracy value for the classification module.

64. The system of claim 63, wherein the model evaluation module is operable to generate at least one precision measurement for the classification module.

65. The system of claim 62, wherein the model evaluation module is operable to compare a first model to a second model to determine if the performance of the first model is statistically significant over the performance of the second model.

66. The system of claim 1, wherein the feature extraction module is operable to first produce feature information for pixels belonging to the training data prior to generation of the classification model by the classification compute model and is operable to second produce feature information for pixels other than the training data prior to application of the classification model to classify pixels other than the training data.

67. A method for land classification based on analysis of very high resolution (VHR) remotely-sensed multispectral Earth imagery, comprising:

storing VHR remotely-sensed multispectral Earth image data in an image store;

generating feature data based on the image data using at least one feature extraction module;

compiling a feature stack comprising the image data and corresponding feature data;

receiving training data regarding at least one or more pixels of the image data from a user regarding at least one class to which the one or more pixels belong; and

generating a classification model at least in part based on a portion of the feature stack corresponding to the one or more pixels and the training data, wherein the classification model related to classification of pixels of image data into one or more classes.

68. The method of claim 67, further comprising:

applying the classification model to the image data to classify pixels of the image data into the one or more classes.

69. The method of claim 68, wherein the applying comprises classifying pixels of the image data based on an analysis of the feature stack in relation to the classification model.

70. The method of claim 69, further comprising:

executing a post classification analysis module to reclassify at least one pixel of the image data based on at least one post classification rule.

71. The method of claim 70, wherein the post classification rule comprises at least one of a minimum mapping unit rule, a surrounded by rule, a topological relation rule, a majority analysis rule, or a smoothing rule.

72. The method of claim 67, wherein the VHR remotely-sensed multispectral Earth image data comprises

spectral band data corresponding to at least 8 multispectral bands.

73. The method of claim 72, wherein the multispectral bands collectively range from at least about 300 nanometers in wavelength to at least about 1100 nanometers in wavelength.

74. The method of claim 73, wherein the multispectral bands collectively range from at least about 300 nanometers in wavelength to at least about 2400 nanometers in wavelength.

75. The method of claim 74, wherein the multispectral bands comprise a plurality of short wave infrared (SWIR) bands.

76. The method of claim 75, wherein the plurality of SWIR bands range from at least about 1100 nanometers in wavelength to at least about 2400 nanometers in wavelength.

77. The method of claim 67, wherein the feature stack comprises a plurality of data layers, wherein each data layer includes different feature data for each pixel of the image data.

78. The method of claim 77, wherein the feature stack includes at least one spectral feature layer, at least one morphological feature layer, or at least one textural feature layer.

79. The method of claim 78, wherein the at least one spectral feature layer comprises data values for each pixel of the image data based on spectral band data collected by a remote image acquisition platform.

80. The method of claim 79, wherein the remote image acquisition platform comprises a satellite imaging system.

81. The method of claim 80, wherein at least one spectral feature layer comprises a relative measure between at least two spectral band data layers.

82. The method of claim 78, wherein the at least one morphological feature layer comprises for each given pixel of the image data information based on the arrangement of adjacent pixels relative to the given pixel.

83. The method of claim 78, wherein the textural feature layer comprises information for a given pixel based on a spatial distribution of tonal variations within one or more spectral band data layers relative to a given pixel.

84. The method of claim 67, further comprising:

providing the client interface as an internet browser-based client operable to provide a user interface for receiving the training data from a user.

85. The method of claim 84, wherein the client interface comprises a plurality of distributed client interfaces operable to receive training data from a plurality of users, wherein the classification model is at least partially based on training data received from a plurality of users.

86. The method of claim 67, further comprising generating a plurality of feature data layers using a plurality of feature extraction modules each operable to generate a feature data layer of the feature stack.

87. The system of claim 86, wherein one or more of the feature extraction modules are selectively applicable to image data by a user to generate feature data layers.

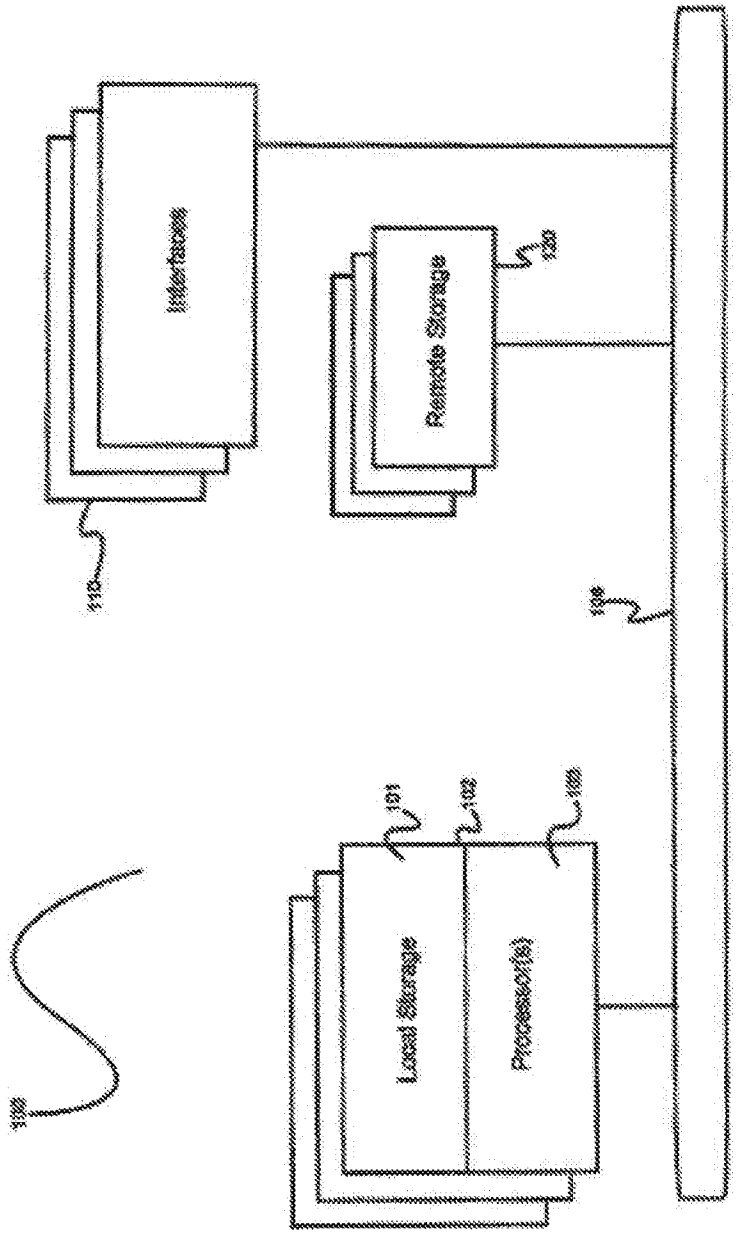

Interfaces
Remote Storage
120
110
130
Local Storage
Processor(s)
101
102
105
100

Fig.1

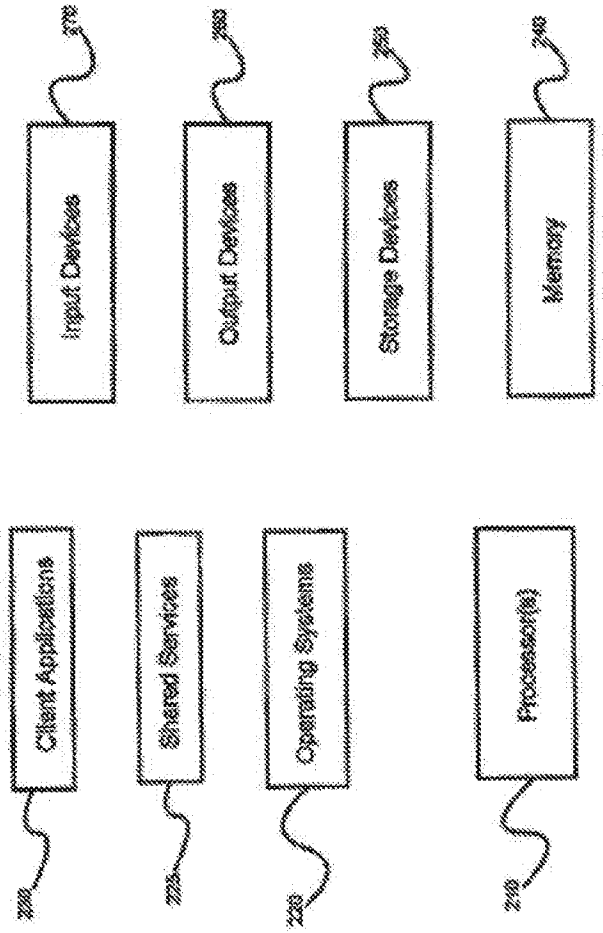

Input Devices
Output Devices
Storage Devices
Memory
Client Applications
Shared Services
Operating Systems
Processor(s)

Fig. 2

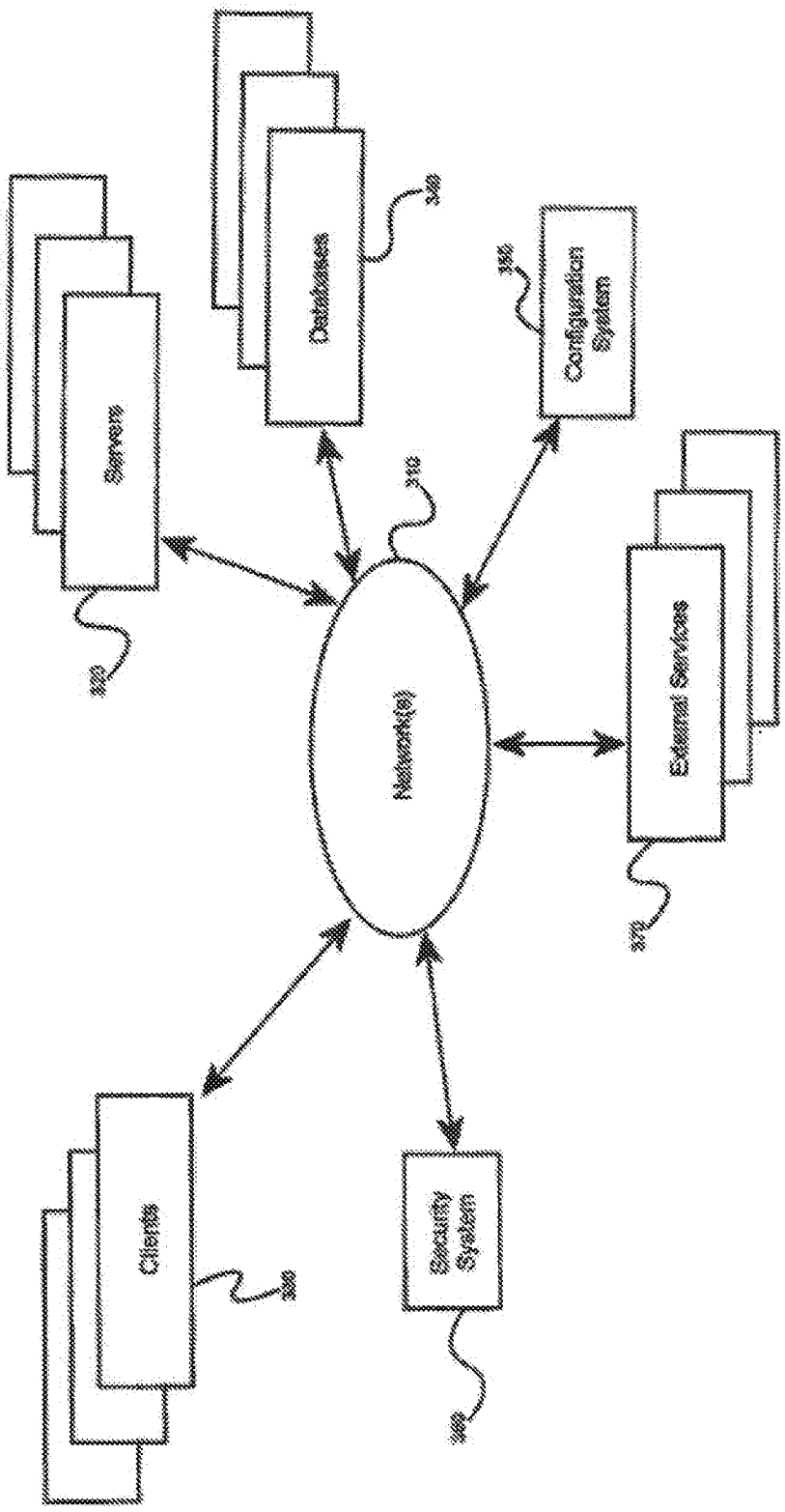

Servers
Databases
Configuration System
Network(s)
External Services
Clients
Security System

Fig. 3

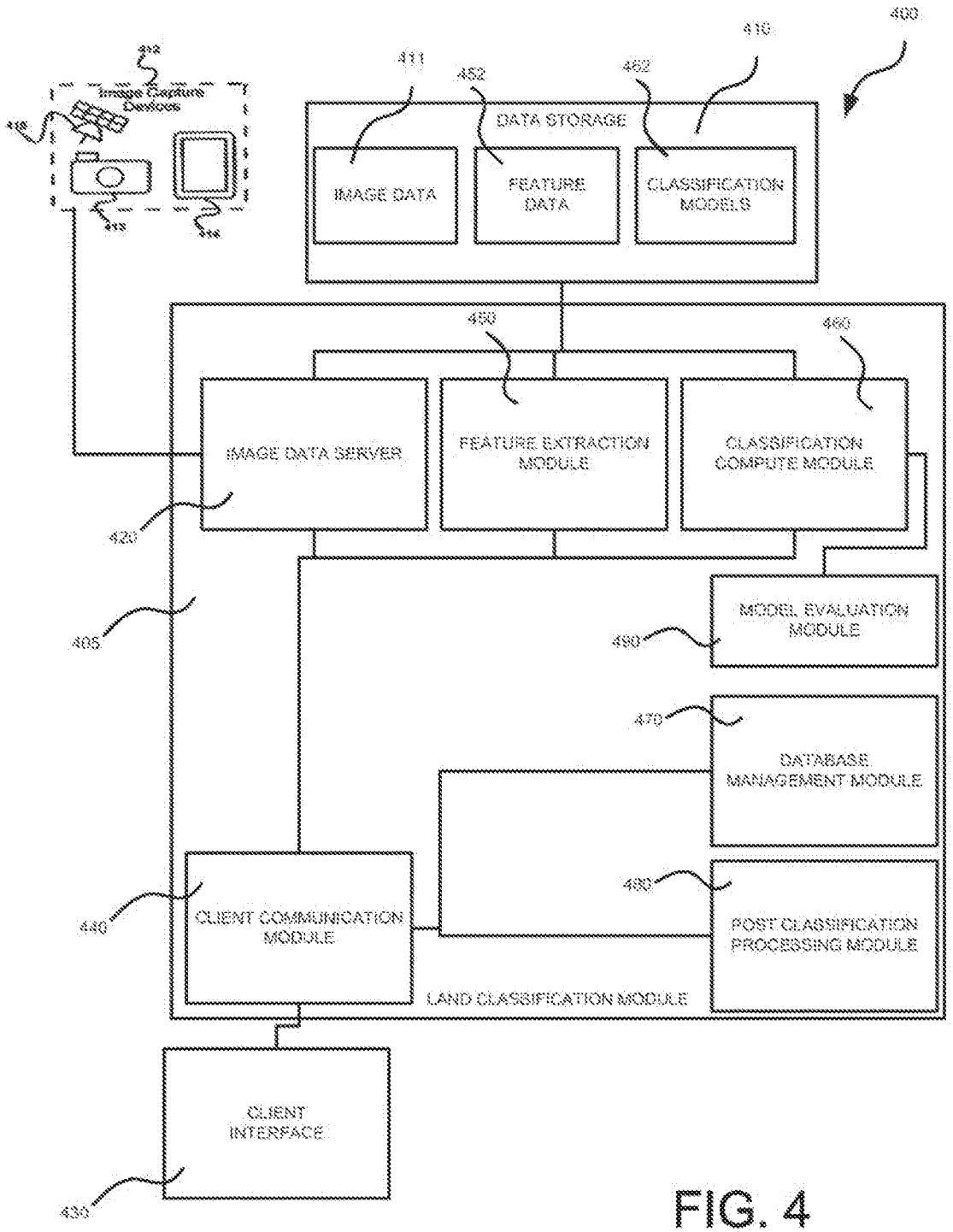

400
410
411
412
413
DATA STORAGE
IMAGE DATA
FEATURE DATA
CLASSIFICATION MODELS
450
460
IMAGE DATA SERVER
FEATURE EXTRACTION MODULE
CLASSIFICATION COMPUTE MODULE
420
405
MODEL EVALUATION MODULE
490
470
DATABASE MANAGEMENT MODULE
480
440
CLIENT COMMUNICATION MODULE
POST CLASSIFICATION PROCESSING MODULE
LAND CLASSIFICATION MODULE
CLIENT INTERFACE
430

FIG. 4

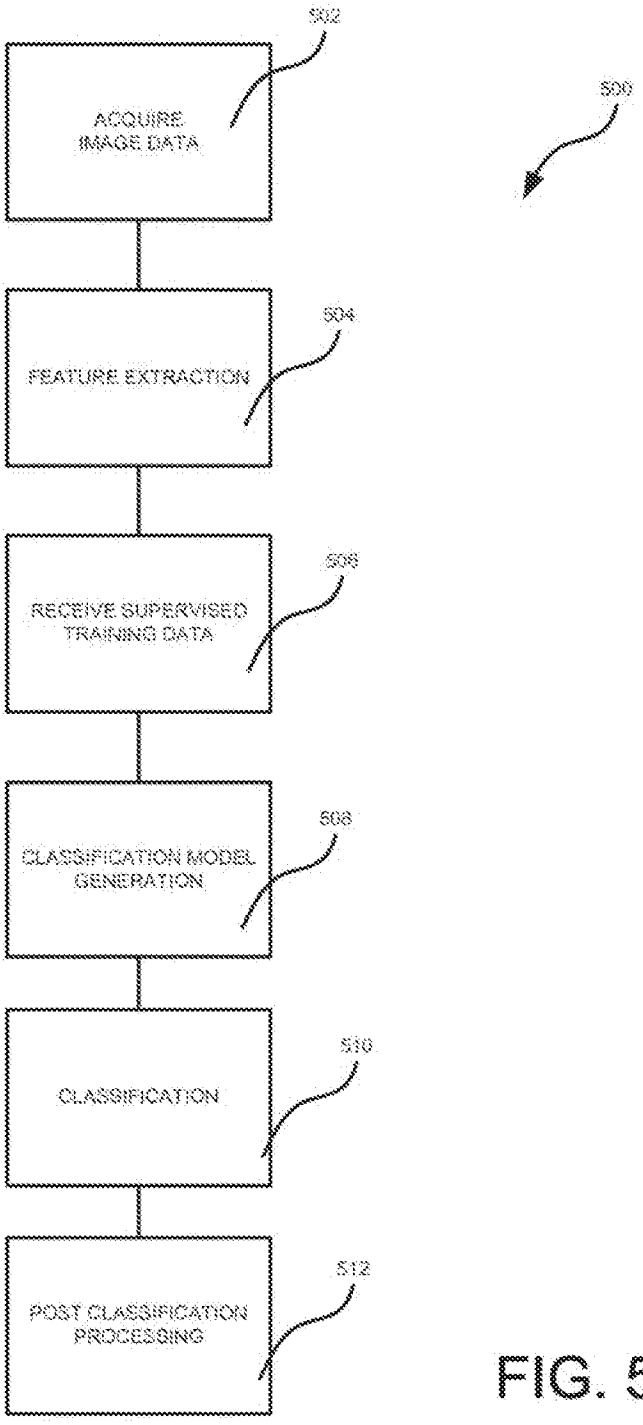

500
502
ACQUIRE IMAGE DATA
504
FEATURE EXTRACTION
506
RECEIVE SUPERVISED TRAINING DATA
508
CLASSIFICATION MODEL GENERATION
510
CLASSIFICATION
512
POST CLASSIFICATION PROCESSING

FIG. 5

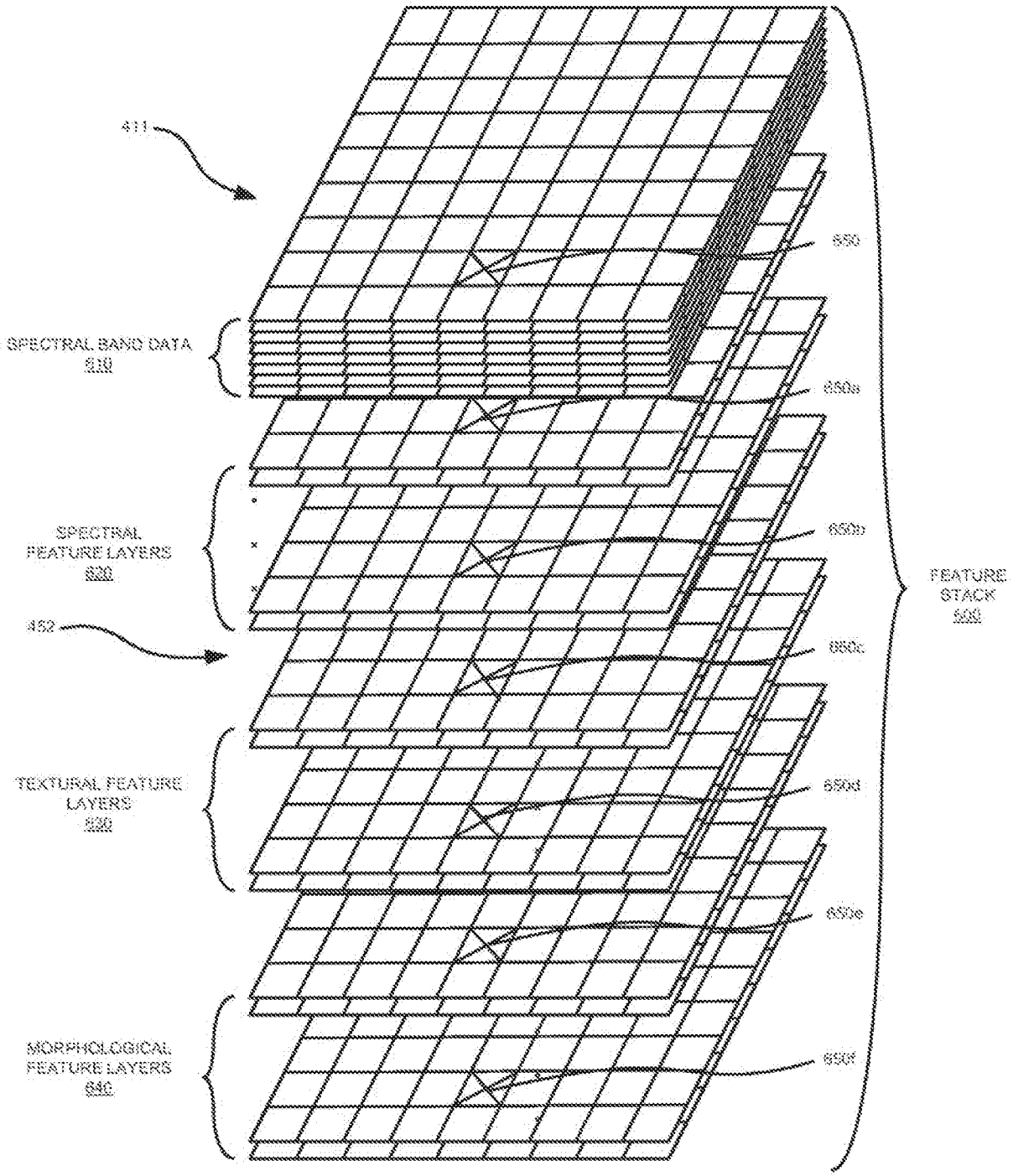

411
650
650a
650b
650c
650d
650e
650f
SPECTRAL BAND DATA
610
SPECTRAL
FEATURE LAYERS
620
452
TEXTURAL FEATURE
LAYERS
630
MORPHOLOGICAL
FEATURE LAYERS
640
FEATURE
STACK
600

FIG. 6

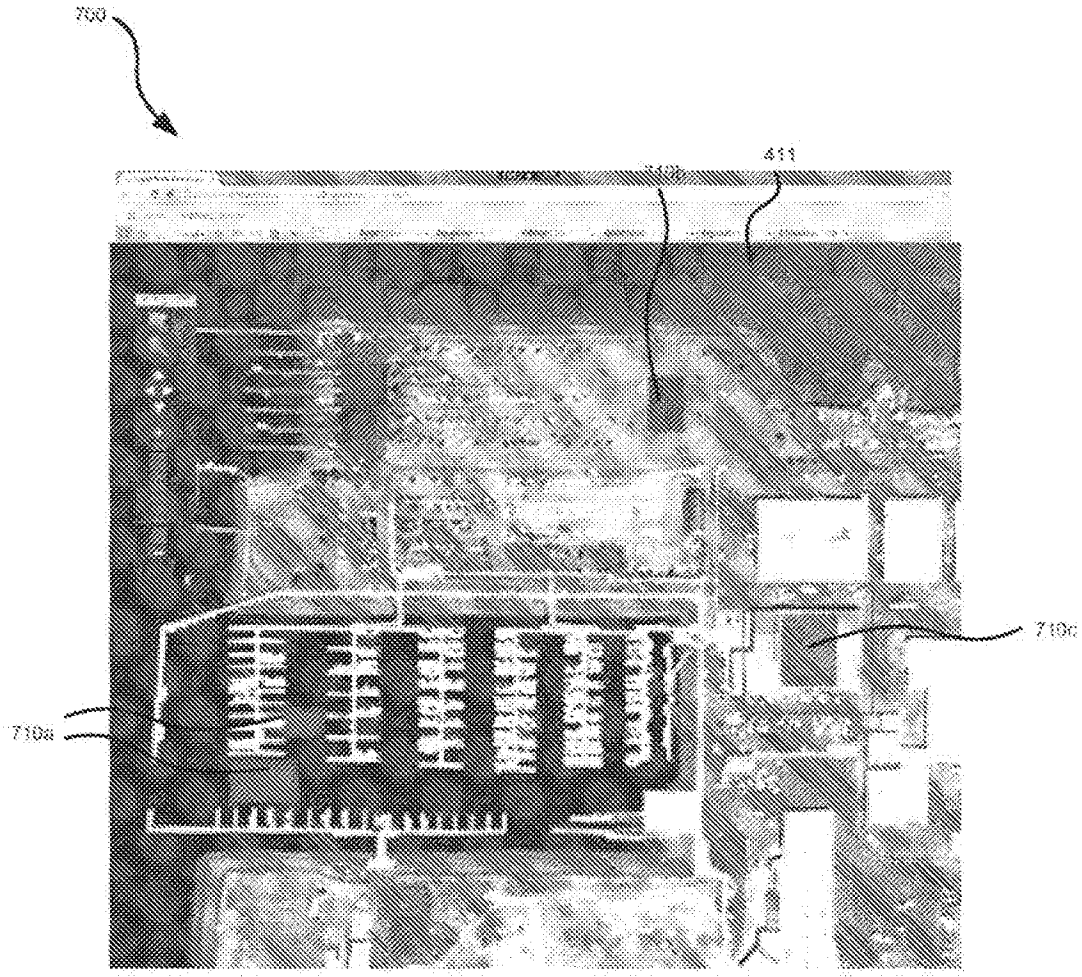


FIG. 7



FIG. 8

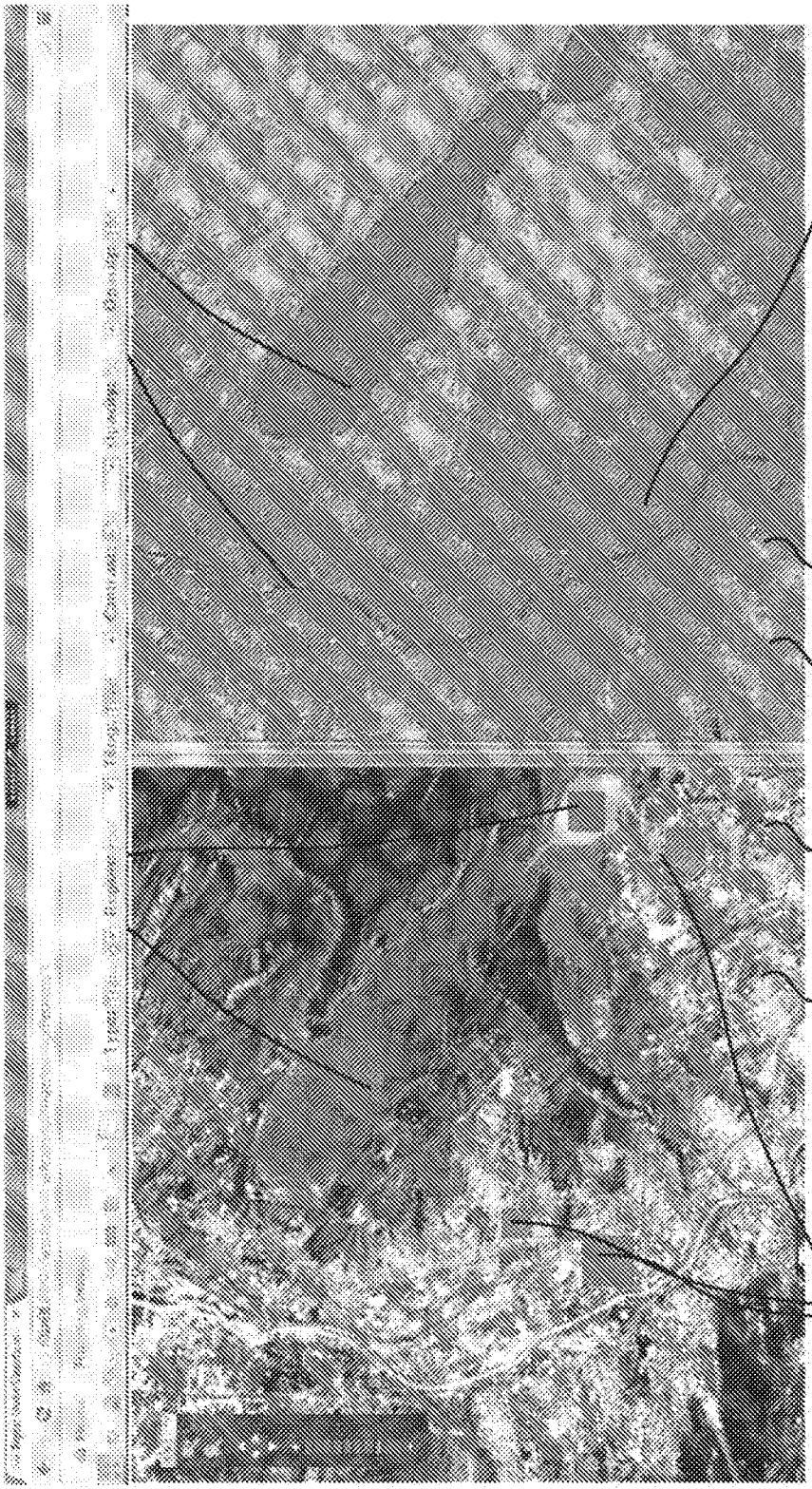


FIG. 9

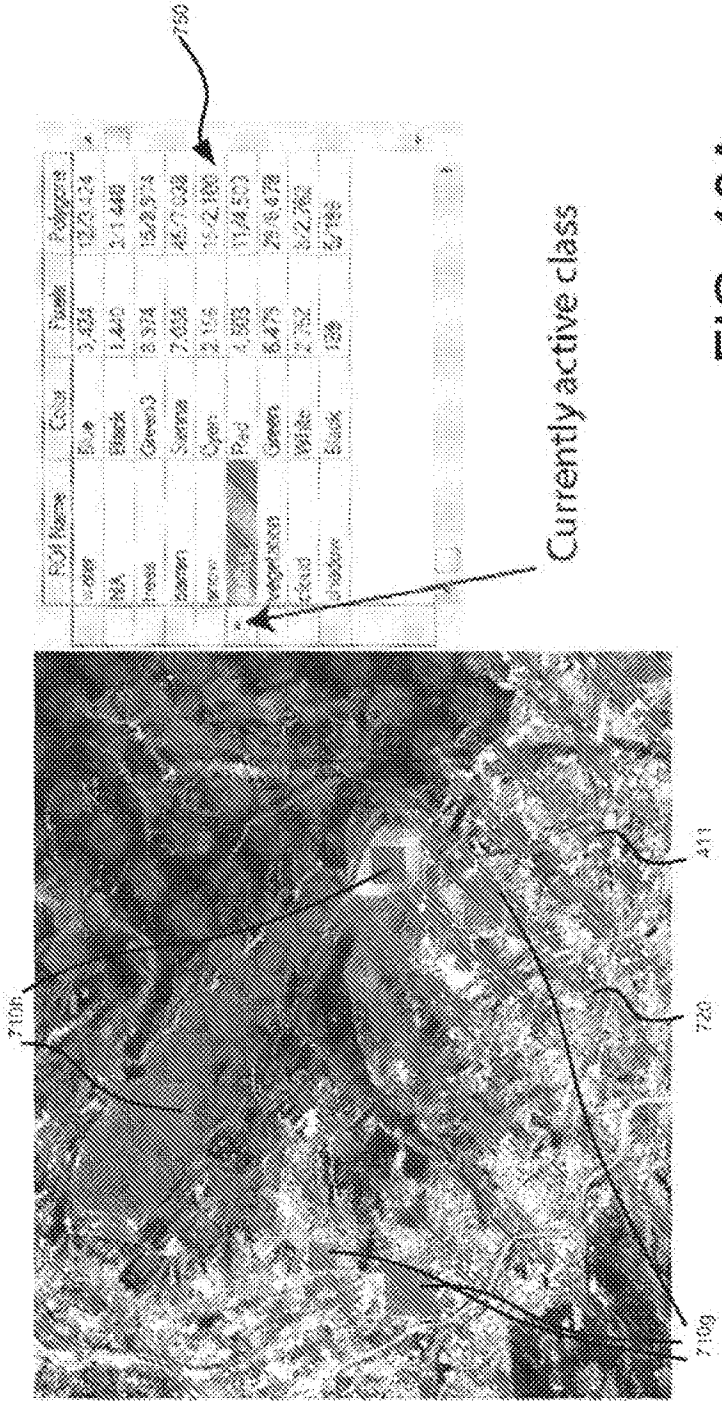


FIG. 10A

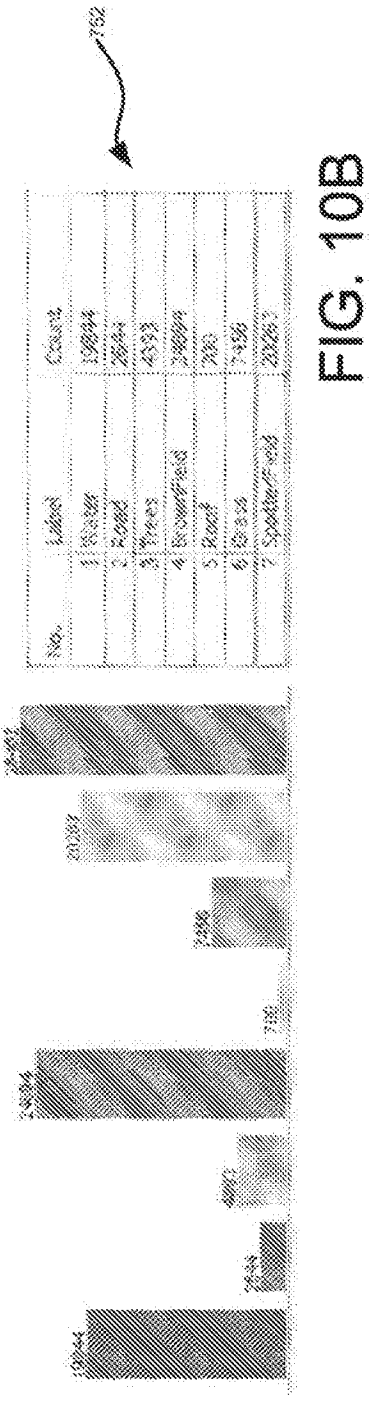


FIG. 10B

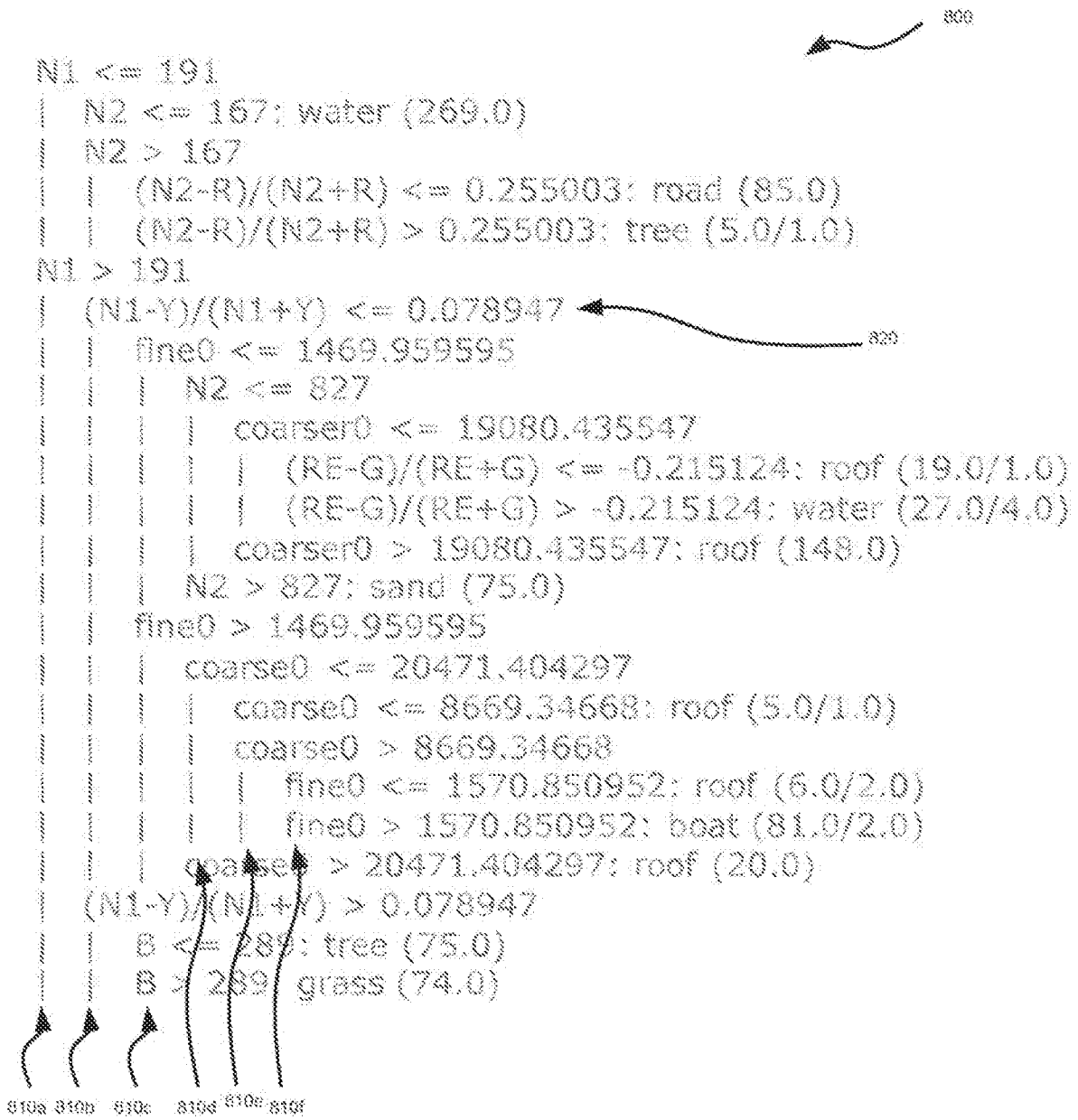

800
N1 <= 191
| N2 <= 167: water (269.0)
| N2 > 167
| | (N2-R)/(N2+R) <= 0.255003: road (85.0)
| | (N2-R)/(N2+R) > 0.255003: tree (5.0/1.0)
N1 > 191
| (N1-Y)/(N1+Y) <= 0.078947
820
| | fine0 <= 1469.959595
| | | N2 <= 827
| | | | coarser0 <= 19080.435547
| | | | | (RE-G)/(RE+G) <= -0.215124: roof (19.0/1.0)
| | | | | (RE-G)/(RE+G) > -0.215124: water (27.0/4.0)
| | | | coarser0 > 19080.435547: roof (148.0)
| | | N2 > 827: sand (75.0)
| | fine0 > 1469.959595
| | | coarse0 <= 20471.404297
| | | | coarse0 <= 8669.34668: roof (5.0/1.0)
| | | | coarse0 > 8669.34668
| | | | | fine0 <= 1570.850952: roof (6.0/2.0)
| | | | | fine0 > 1570.850952: boat (81.0/2.0)
| | | coarse0 > 20471.404297: roof (20.0)
| (N1-Y)/(N1+Y) > 0.078947
| | B <= 289: tree (75.0)
| | B > 289: grass (74.0)
810a
810b
810c
810d
810e
810f

FIG. 11

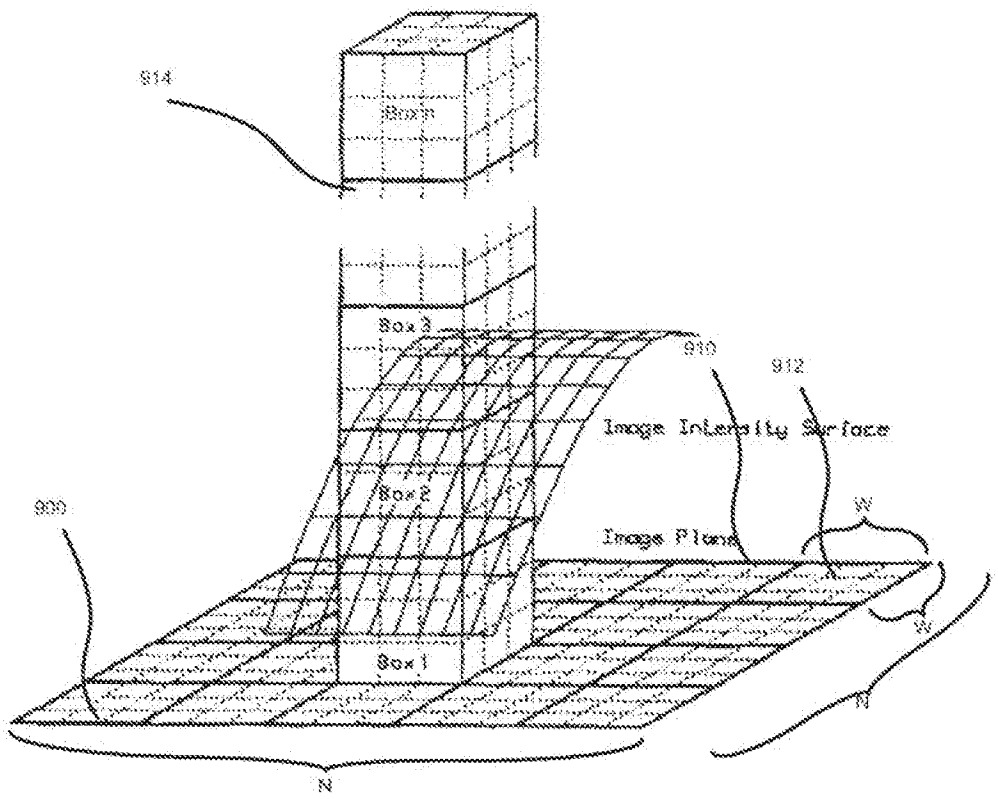

914
Box 3
Box 2
Box 1
910
912
Image Intensity Surface
Image Plane
900
W
W
N
N

FIG. 12

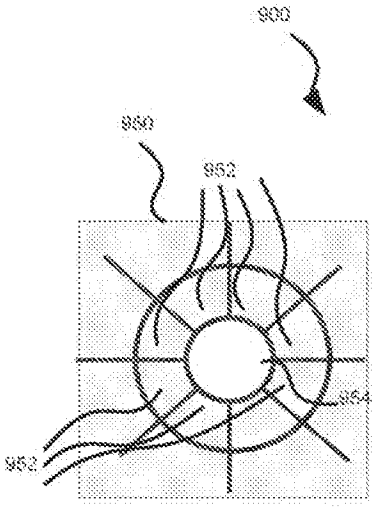


FIG. 13

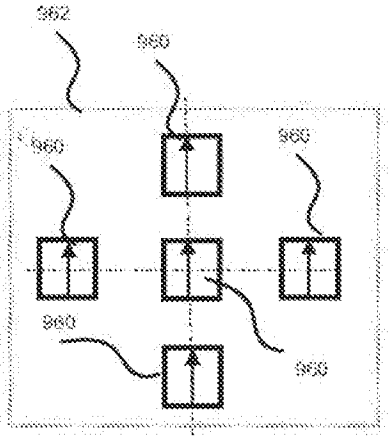


FIG. 14A

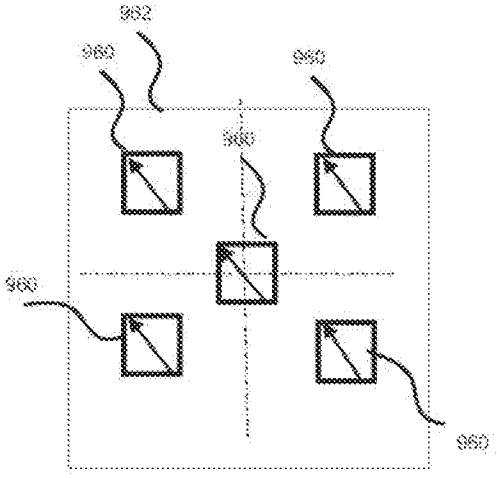


FIG. 14B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/053387

A. CLASSIFICATION OF SUBJECT MATTER

G06F 19/00(2011.01)i, G06T 7/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 19/00; G06F 15/16; G06K 9/62; G06K 9/00; G06K 9/66; G06T 7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: land, classification, imagery, analysis, feature, extraction, feature stack, training, classification model, and similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0183225 A1 (PAL, SANKAR KUMAR et al.) 19 July 2012 See paragraphs [0002], [0006], [0021]-[0024], [0027]-[0033], [0037], [0040]-[0042], and [0056]-[0058]; claim 1; and figures 3-4 and 7.	1-87
Y	EP 1846889 B1 (CAMBRIDGE RESEARCH & INSTRUMENTATION, INC.) 05 October 2011 See paragraphs [0006]-[0007], [0010], [0044], [0049], [0053], and [0097]; and figures 1 and 3.	1-87
Y	US 2009-0220153 A1 (HALL, STEWART E. et al.) 03 September 2009 See paragraphs [0029]-[0032] and figures 1-2.	10-55,57-58,85-87
A	US 2009-0251472 A1 (ANTOINE, LENNOX BERTRAND) 08 October 2009 See paragraphs [0007]-[0010] and [0050]-[0060].	1-87
A	US 2008-0089577 A1 (BANG, YOUNIAN) 17 April 2008 See paragraphs [0028]-[0036] and figure 5.	1-87

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 December 2014 (12.12.2014)	**12 December 2014 (12.12.2014)**

Name and mailing address of the ISA/KR	Authorized officer
International Application Division Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. +82-42-472-7140	NHO, Il Myong Telephone No. +82-42-481-8528

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/053387

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0183225 A1	19/07/2012	CN 103052962 A	17/04/2013
		KR 10-2013-0028978 A	20/03/2013
		WO 2012-069891 A1	31/05/2012
		WO 2012-069891 A8	09/08/2012
EP 1846869 B1	05/10/2011	AT 527619 T	15/10/2011
		CN 101151623 A	26/03/2008
		CN 101151623 B	23/06/2010
		CN 101916359 A	15/12/2010
		CN 101916359 B	20/06/2012
		EP 1846869 A1	24/10/2007
		EP 2237189 A2	06/10/2010
		EP 2237189 A3	16/07/2014
		US 2006-0245631 A1	02/11/2006
		US 2009-0297016 A1	03/12/2009
		US 2011-0255753 A1	20/10/2011
		US 2013-0077875 A1	28/03/2013
		US 7555155 B2	30/06/2009
		US 7953264 B2	31/05/2011
		US 8280140 B2	02/10/2012
		US 8639043 B2	28/01/2014
		WO 2006-081547 A1	03/08/2006
US 2009-0220132 A1	03/09/2009	AU 2009-217782 A1	03/09/2009
		CA 2715971 A1	03/09/2009
		CN 101965578 A	02/02/2011
		CN 101965578 B	24/09/2014
		EP 2257912 A2	08/12/2010
		JP 2011-514594 A	12/05/2011
		JP 5530726 B2	17/09/2014
		US 8428310 B2	23/04/2013
		WO 2009-108256 A2	03/09/2009
		WO 2009-108256 A3	22/10/2009
US 2009-0251472 A1	08/10/2009	EP 1764745 A2	21/03/2007
		EP 1764745 A3	25/05/2011
		US 2007-0064005 A1	22/03/2007
		US 7538473 B2	19/05/2009
		US 8139156 B2	12/06/2012
US 2008-0089577 A1	17/04/2008	None	

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 April 2016 (07.04.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/053791 A1

(51) International Patent Classification:
G06T 7/00 (2006.01)

(21) International Application Number:
PCT/US2015/052291

(22) International Filing Date:
25 September 2015 (25.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/499,440 29 September 2014 (29.09.2014) US

(71) Applicant: **DIGITALGLOBE, INC.** [—/US]; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(72) Inventor: **OUZOUNIS, Georgios**; 1601 Dry Creek Drive, Suite 260, Longmont, Colorado 80503 (US).

(74) Agent: **CROUCH, Robert G.**; Marsh Fischmann & Breyfogle LLP, 8055 E. Tufts Ave., Suite 450, Denver, Colorado 80237 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: MULTI-SPECTRAL IMAGE LABELING WITH RADIOMETRIC ATTRIBUTE VECTORS OF IMAGE SPACE REPRESENTATION COMPONENTS

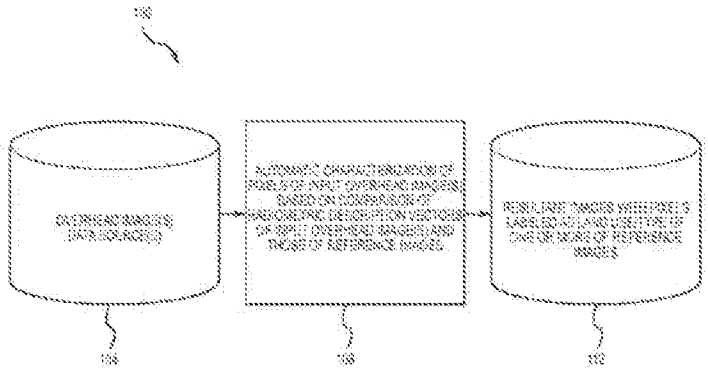


FIG. 1

(57) Abstract: Automatic characterization or categorization of portions of an input multispectral image based on a selected reference multispectral image. Sets (e.g., vectors) of radiometric descriptors of pixels of each component of a hierarchical representation of the input multispectral image can be collectively manipulated to obtain a set of radiometric descriptors for the component. Each component can be labeled as a (e.g., relatively) positive or negative instance of at least one reference multispectral image (e.g., mining materials, crops, etc.) through a comparison of the set of radiometric descriptors of the component and a set of radiometric descriptors for the reference multispectral image. Pixels may be labeled (e.g., via color, pattern, etc.) as positive or negative instances of the land use or type of the reference multispectral image in a resultant image based on components within which the pixels are found.

WO 2016/053791 A1

MULTI-SPECTRAL IMAGE LABELING WITH RADIOMETRIC ATTRIBUTE VECTORS OF IMAGE SPACE REPRESENTATION COMPONENTS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application No. 14/499,440, entitled "MULTI-SPECTRAL IMAGE LABELING WITH RADIOMETRIC ATTRIBUTE VECTORS OF IMAGE SPACE REPRESENTATION COMPONENTS, filed on September 29, 2014, the entire contents of which are incorporated herein by reference as if set forth in full.

BACKGROUND

[0002] The use of geospatial or overhead imagery has increased in recent years and represents a significant tool that may be utilized in a number of contexts. As such, high quality geospatial imagery has become increasingly valuable. For example, a variety of different entities (e.g., individuals, governments, corporations, or others) utilize geospatial imagery (e.g., satellite imagery) for a multitude of applications in a variety of contexts.

[0003] With increasingly capable satellites being commissioned and launched, very high resolution (VHR) remotely-sensed multispectral Earth imagery has become increasingly available and useable. For example, as the number of satellite image acquisition systems in operation grows, acquisition ability and flexibility improves. In an example, DigitalGlobe, Inc. of Longmont, CO currently operates a number of satellites including, IKONOS, GeoEye-1, QuickBird, WorldView 1, WorldView 2, and WorldView 3. Accordingly, around the clock global coverage may be achieved through the satellite constellation currently in operation. For instance, the DigitalGlobe constellation of satellites can image the entire Earth's landmass every 75 days and may capture over six times the Earth's landmass every year with a capacity to collect at least 2.6 million square kilometers of imagery a day. With selective tasking, DigitalGlobe's satellite constellation may quickly and continually collect imagery from targeted locations to provide near real time feedback on global events or the like.

[0004] Furthermore, the resolution of image acquisition satellites also continues to increase. For instance, currently operated satellites may acquire images with a maximum spatial resolution of 50

cm (wherein each pixel in the images corresponds with the distance measure of the spatial resolution). Additionally, planned satellite launches may provide even greater resolution capabilities with spatial resolutions as high as about 30 cm or greater (i.e., less than 30 cm, such as 25 cm, 15 cm, or lower).

[0005] In this light, the amount and quality of VHR remotely-sensed multispectral Earth imagery continues to increase as does the amount and types of image data collected. Accordingly, the nature of the VHR remotely-sensed multispectral Earth imagery may facilitate uses beyond simply providing pixels as image data. For instance, higher level data processing may be applied to the images to, for example, identify objects, identify textures, or otherwise extract useful data from the raw image data. One such application has been in land use identification and land classification, where remotely-sensed images are analyzed to categorize pixels of an image into a category of land use or land class. As just one example, image pixels can be categorized or analyzed to identify and/or characterize areas of the urban settlement (e.g., the urban "build-up" or "built-up," such as three-dimensional man-made structures or the like).

[0006] As the amount of image data that is available grows and the nature of the image data acquired changes and is improved, advanced image data processing and image analytics are needed to keep pace with the advances in image acquisition technology to facilitate new techniques applied to acquired images to expand the number of applications for which such technology may be used.

SUMMARY

[0007] Broadly, disclosed herein are utilities (e.g., systems, processes, etc.) for automatically characterizing or categorizing portions of an input multispectral image based on a selected reference multispectral image. The input multispectral images to be characterized may be substantially raw images (e.g., as received at the image sensors of a satellite) or may be pre-processed in any appropriate manner (e.g., to appropriately account for atmospheric effects and/or the like). Sets (e.g., vectors) of radiometric descriptors of pixels of each component of a hierarchical representation of the input multispectral image can be collectively manipulated to obtain a set of radiometric descriptors for the component. Each component can be labeled as a (e.g., relatively) positive or negative instance of at least one reference multispectral image (e.g., mining materials, crops, etc.) through a comparison of the set of radiometric descriptors of the component and a set of

radiometric descriptors for the reference multispectral image. Pixels may be labeled (e.g., via color, pattern, etc.) as positive or negative instances of the land use or type of the reference multispectral image in a resultant image based on components within which the pixels are found. Resultant images can be used in numerous contexts such as in analyzing mining materials, performing agricultural monitoring, identifying the presence of built-up, mapping constructed facilities, and the like.

[0008] In one arrangement, any appropriate fusion image or the like may be appropriately generated from the one or more input multispectral images to be used in the process of characterizing portions of the input multispectral image(s). For instance, a "similarity map" may be initially constructed from the input multispectral image by way of reducing a plurality of radiometric descriptors (e.g., intensity level values) at a plurality of spectral bands for each pixel of a plurality of pixels of the input multispectral image to a single radiometric descriptor for each pixel of the plurality of pixels of the input multispectral image. Each single radiometric descriptor thus encodes multispectral information for a respective pixel. The various single radiometric descriptors can be used as input for segmenting the image into a plurality of connected components of a hierarchical representation of the image (e.g., Max-Tree). Rather than generating multiple Max-Trees, for instance, from a plurality of respective grayscale values of the pixels (one Max-Tree per grayscale image), a single Max-Tree can now be constructed from multi-spectral information. As another example, a corresponding "dissimilarity map" may be constructed from the input multispectral image from which an Alpha-Tree may be generated.

[0009] In one aspect, a method for use in classifying areas of interest in overhead imagery includes organizing, using a processor, a plurality of pixels of at least one input multispectral image of a geographic area into a plurality of components of a hierarchical image representation structure; deriving, using the processor, at least one set of radiometric descriptors for each component of the plurality of components; obtaining at least one set of radiometric descriptors for a reference multispectral image; determining, using the processor, for each component of the hierarchical image representation structure, a similarity between the set of radiometric image descriptors for the component and the set of radiometric image descriptors of the reference multispectral image; and using the determined similarity metrics to determine a degree to which the pixels of each connected component identify the at least one land use or land type of the reference multispectral image.

[0010] In one arrangement, the deriving includes, for each pixel of the component, obtaining a set of radiometric descriptors for the pixel, wherein each set includes a plurality of entries, and using the sets of radiometric descriptors for the pixels of the component to derive the set of radiometric descriptors for the component. For instance, the set of radiometric descriptors for each pixel may be a vector of intensities of the pixel at each spectral band of a plurality of spectral bands of the input multispectral image or a vector of intensity gradients of the pixel over the plurality of spectral bands. More specifically, the sets of radiometric descriptors for the pixels of the component may be used to derive the set of radiometric descriptors for the component by way of obtaining means of each respective entry of the sets of radiometric descriptors of the pixels of the component to obtain radiometric descriptor mean values for corresponding respective entries of the set of radiometric descriptors for the component. For instance, the first entry in the vector for the component may be the mean of the first entries of the vectors for the pixels, the second entry in the vector for component may be the mean of the second entries of the vectors for the pixels, and so on.

[0011] Any of the embodiments, arrangements, or the like discussed herein may be used (either alone or in combination with other embodiments, arrangement, or the like) with any of the disclosed aspects. Merely introducing a feature in accordance with commonly accepted antecedent basis practice does not limit the corresponding feature to the singular. Any failure to use phrases such as "at least one" does not limit the corresponding feature to the singular. Use of the phrases "at least generally," "at least partially," "substantially" or the like in relation to a particular feature encompass the corresponding characteristic and insubstantial variations thereof. Furthermore, a reference of a feature in conjunction with the phrase "in one embodiment" does not limit the use of the feature to a single embodiment.

[0012] In addition to the exemplary aspects and embodiments described above, further aspects and embodiments will become apparent by reference to the drawings and by study of the following descriptions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Figure 1 is a simplified block diagram illustrating a process of detecting or characterizing pixels in multispectral overhead imagery of a geographic area.

[0014] Figure 2 is a more detailed block diagram of an automated system for characterizing pixels in multispectral overhead imagery of a geographic area.

[0015] Figure 3 is a schematic diagram illustrating the pixels of a portion of an input multispectral image of a geographic area at each of a plurality of spectral bands and a similarity map of the pixels that may be generated from radiometric descriptors of the pixels at each of the spectral bands.

[0016] Figure 4 illustrates a simplified hierarchical image representation of the input multispectral image of Figure 3, where one of a plurality of connected components of the hierarchical image representation is shown to include four of the pixels of Figure 3.

[0017] Figure 5 illustrates the hierarchical image representation of Figure 4, where the connected components include normalized scores as a result of performing a proximity analysis between radiometric descriptor sets of the connected components and a radiometric descriptor set of a reference multispectral image.

[0018] Figure 6 is a flow diagram of a method for characterizing pixels in multispectral overhead imagery of a geographic area.

[0019] Figure 7 is a flow diagram of a method of generating a similarity map from radiometric descriptors of pixels of an input multispectral image at each of the spectral bands.

DETAILED DESCRIPTION

[0020] At the outset, it is noted that, when referring to the earth herein, reference is made to any celestial body of which it may be desirable to acquire images or other remote sensing information. Furthermore, when referring to "overhead" imagery herein, such imagery may be obtained by any spacecraft, satellite, aircraft, and/or the like capable of acquiring images or other remote sensing information. Furthermore, the utilities described herein may also be applied to other imaging systems, including imaging systems located on the earth or in space that acquire images of other celestial bodies. It is also noted that the figures contained herein are not necessarily drawn to scale and that such figures have been provided for the purposes of discussion and illustration.

[0021] Generally, high resolution images of selected portions of a celestial body's surface have become a product desired and used by government agencies, corporations, and individuals. For

instance, many consumer products in common use today include images of the Earth's surface, such as Google® Earth. Various types of remote sensing image collection platforms may be employed, including aircraft, earth-orbiting satellites, and the like. In the case of a consumer digital camera, as one non-limiting example, an image sensor is generally arranged in an area array (e.g., 3,000 rows of 3,000 pixels each, or 9,000,000 total pixels) which collects the image area in a single "snapshot." In the case of satellite-based imaging, as another non-limiting example, the "push-broom scanning" principle is sometimes employed whereby each image sensor includes a relatively small number of rows (e.g., a couple) of a great number of pixels (e.g., 50,000 or more) in each row. Each row of pixels may be scanned across the earth to build an image line by line, and the width of the image is the product of the number of pixels in the row times the pixel size or resolution (e.g., 50,000 pixels at 0.5 meter ground resolution produces an image that is 25,000 meters wide). The length of the image is controlled by the scan duration (i.e. number of lines), which is typically settable for each image collected. The resolution of overhead images varies depending on factors such as the particular instrumentation utilized, the altitude of the satellite's (or other aircraft's) orbit, and the like.

[0022] Image collection platforms (e.g., aircraft, earth-orbiting satellites, etc.) may collect or acquire various types of imagery in one or more manners. As one non-limiting example, image collection platforms may perform panchromatic collection of scenes of a celestial body which generally refers to the collection of image data across a single broad range of wavelengths (e.g., all visible light, from near infrared (NIR) to near ultraviolet (NUV), etc.). As another non-limiting example, image collection platforms may additionally or alternatively capture image data within the visible light band and include respective filters to separate the incoming light into red, green and blue portions. As a further non-limiting example, image collections platforms may additionally or alternatively perform multispectral collection of scenes of a celestial body which generally refers to the collection of image data at multiple specific spectral bands across the electromagnetic spectrum (e.g., within bands both inside and outside of the visible light range such as NIR, short wave infrared (SWIR), far infrared (FIR), etc.). For instance, a satellite may have one image sensor that is sensitive to electromagnetic radiation across only a first spectral band (e.g., the visible light band, such as a wavelength range of about 380-750 nm) in addition to one or more additional image sensors that are sensitive to electromagnetic radiation only across other spectral bands (e.g., NIR, 750-1400 nm; SWIR, 1400-3000 nm; etc.). Multi-spectral imaging may allow for the extraction of

additional information from the radiance received at a satellite after being reflected from the Earth's surface (which may include atmospheric effects such as from aerosols, clouds, etc.).

[0023] Turning now to Figure 1, a simplified block diagram of a system 100 is illustrated that may be used to automatically characterize portions of input multispectral images (e.g., <1 - 10m spatial resolution satellite image data) based on an analysis of radiometric descriptors (e.g., intensity values) in the images. Initially, the system 100 obtains at least one input multispectral image of a geographic areas from any appropriate overhead image data sources 104 (e.g., SPOT 2 and 5, CBERS 2B, RapidEye 2 and 4, IKONOS® 2, QuickBird 2, WorldView-1, WorldView-2, WorldView-3, etc.) and performs the automated characterization 108 of portions (e.g., connected components, groups of pixels, etc.) of the input multispectral image based on a comparison of radiometric descriptor sets (e.g., vectors) of the portions and radiometric descriptor sets of one or more reference images corresponding to region types to be identified. The system 100 then generates resultant images 116 that depict or convey one or more detected or characterized region types in the input multispectral image such as water bodies, vegetation, metals, bare soil, etc.

[0024] Figure 2 presents a more detailed block diagram of an automated system 200 that may be used to implement the automated characterization 108 of Figure 1. Although depicted as a single device (e.g., server, workstation, laptop, desktop, mobile device, and/or other computing device), one or more functionalities, processes or modules of the system 200 may be allocated or divided among a plurality of machines, devices and/or processes which may or may not be embodied in a single housing. In one arrangement, functionalities of the system 200 may be embodied in any appropriate cloud or distributed computing environment.

[0025] Broadly, the system 200 may include memory 204 (e.g., one or more RAM or other volatile memory modules, etc.), a processing engine or unit 208 (e.g., one or more CPUs, processors, processor cores, or other similar pieces of hardware) for executing computer readable instructions from the memory 204, storage 212 (e.g., one or more magnetic disks or other non-volatile memory modules or non-transitory computer-readable mediums such as solid-state drives (SSDs)), and/or a number of other components 216 (e.g., input devices such as a keyboard and mouse, output devices such as a display and speakers, and the like), all of which may be appropriately interconnected by one or more buses 220. While not shown, the system 200 may include any appropriate number and

arrangement of interfaces that facilitate interconnection between the one or more buses 220 and the various components of the system 200 as well as with other devices (e.g., network interfaces to allow for communication between the system 200 and other devices over one or more networks, such as LANs, WANs, the Internet, etc.).

[0026] The system 200 may appropriately retrieve one or more multispectral overhead images 224 of a geographic area (e.g., from one or more overhead image data sources 104 of Figure 1) and store the same in any appropriate form in storage 212 (e.g., such as in one or more databases and manageable by any appropriate database management system (DBMS) to allow the definition, creation, querying, update, and administration of the databases). The processing engine 208 may execute a DBMS or the like to retrieve and load the one or more overhead images 224 into the memory 204 for manipulation by a number of engines or modules of the system 200 as will be discussed in more detail below.

[0027] As shown, the system 200 may include a construction engine 228 that is broadly configured to generate similarity maps from the radiometric descriptors of pixels at each of a plurality of spectral bands (e.g., three or more) of the input multispectral image, use the similarity map to generate hierarchical image representations, and generate sets (e.g., vectors) of radiometric descriptors for each connected component of the hierarchical image representation (e.g., where each entry in the vector represents a radiometric descriptor for the connected component at a respective one of the plurality of spectral bands). The system 200 may also include an analysis engine 232 that is broadly configured to analyze the sets of radiometric descriptors of the connected component in view of sets of radiometric descriptors of reference multispectral images to generate similarity metrics that are indicative of a degree to which the components identify one or more land uses or types (e.g., water bodies, agricultural, crops, etc.) of the reference multispectral images.

[0028] A mapping engine 236 may also be included to map categorized portions of the input multispectral image into a resultant image. For instance, the resultant image may be in the form of a heat map whereby color corresponds to land type and intensity level corresponds to confidence that the portion of the image actually corresponds to the particular land type. Each of the engines (and/or other engines, modules, logic, etc. disclosed and/or encompassed herein) may be in the form of one or more sets of computer-readable instructions for execution by the processing unit 208 and

that may be manipulated by users in any appropriate manner to perform automated characterization of portions of an input multispectral image 225 for presentation on a display (not shown). In this regard, the combination of the processing unit 208, memory 204, and/or storage 212 (i.e., machine/hardware components) on the one hand and the various engines/modules disclosed herein in one embodiment create a new machine that becomes a special purpose computer once it is programmed to perform particular functions of the characterization utilities disclosed herein (e.g., pursuant to instructions from program software).

[0029] In one arrangement, any appropriate portal in communication with the various engines may run on the system 200 and be accessible by users (e.g., via any appropriate browser) to access the functionalities of the system 200. While the various engines have been depicted in Figure 2 as being separate or distinct modules, it is to be understood that the functionalities or instructions of two or more of the engines may actually be integrated as part of the same computer-readable instruction set and that the engines have been depicted in the manner shown in Figure 2 merely to highlight various functionalities of the system 200. Furthermore, while the engines have been illustrated as being resident within the (e.g., volatile) memory 204 (e.g., for execution by the processing engine 208), it is to be understood that the engines may be stored in (e.g., non-volatile) storage 212 (and/or other non-volatile storage in communication with the system 200) and loaded into the memory 204 as appropriate.

[0030] To facilitate the reader's understanding of the various engines and functionalities of the system 200, additional reference is now made to Figure 6 which illustrates a method 600 for use in performing the automated characterization processes disclosed herein. While specific steps (and orders of steps) of the method 600 (as well as other methods disclosed herein) have been illustrated and will be discussed, other methods (including more, fewer or different steps than those illustrated) consistent with the teachings presented herein are also envisioned and encompassed within the present disclosure.

[0031] The method 600 may begin by organizing 604 the pixels of at least one input multispectral image (e.g., at least three frequency or spectral bands such as RGB bands, 8-bands, etc.) of a particular geographic area into a plurality of hierarchically connected components or nodes (e.g., groups of pixels that collectively define the input image(s) as whole) of a hierarchical image

representation (e.g., hierarchical data structure, partition pyramid, etc.) that generally maps image partitions from fine to course. With reference to Figures 2 and 4, for instance, the construction engine 228 of the automated extraction system 200 may receive at least one input multispectral image 225 (including a plurality of spectral bands 226) of a particular geographic area and organize the pixels of the input multispectral image 225 into a plurality of connected components 244/404 (e.g., nodes) of a hierarchical image representation 240/400 (e.g., Max-Tree, Alpha Tree, etc.).

[0032] In the case of the hierarchical image representation 400 being in the form of an Alpha-Tree, for example, it may be a rooted, uni-directed tree or partition pyramid with its leaf components 408 corresponding to one or more pixels that are identical in relation intensity and/or the like and its root component 412 corresponding to the tip of the hierarchy (e.g., the single component whose extent defines the image definition domain), where pixels are grouped into connected components based on any appropriate definition or measure of dissimilarity between adjacent pixels of the input image (e.g., maximum pixel intensity, native reflectance, etc.). For instance, one manner of constructing an Alpha-Tree is disclosed in U.S. Patent No. 8,682,079, entitled "Automatic Extraction of Built-Up Footprints from High Resolution Overhead Imagery through Manipulation of Alpha-Tree Data Structures," which is incorporated herein by reference in its entirety.

[0033] In the case of the hierarchical image representation 400 being in the form of a Max-Tree, for example, it may be a rooted, uni-directed tree with its leaf components 408 (e.g., leaf nodes) corresponding to regional maxima of any appropriate fusion image (e.g., the similarity map constructed from the input multispectral image) and its root component 412 (e.g., root node) corresponding to a single connected component defining the background of the input multispectral image. For instance, the input multispectral image may be thresholded at each of a number of intensity or grayscale levels to provide as many binary images as the number of grey levels, where each binary image may then be analyzed to derive its connected components 404. At least one morphological attribute filter (e.g., an edge preserving operator) may progressively accept (or reject) connected components 404 of the tree based on some attribute criterion. For instance, one manner of extracting built-up from high resolution overhead imagery is disclosed in U.S. Patent App. No. 13/955,268 and entitled "Automatic Generation of Built-up Layers from High Resolution Satellite Image Data," which is incorporated herein by reference in its entirety.

[0034] In one arrangement, the various spectral bands 226 of the input multispectral image 225 may be fused into a single band in any appropriate manner before the input multispectral image 225 is organized into the hierarchical image representation 240/400. Stated differently, the various radiometric descriptors 227 (e.g., intensity values) among the plurality of spectral bands 226 for each pixel may be fused into a single value for the pixel, where the single band includes the fused single values of the plurality of pixels of the input multispectral image 225. In one arrangement, and turning now to Figure 7, a method 700 of generating a fusion image in the form of a similarity map from radiometric descriptors of pixels of an input multispectral image at each of a plurality of spectral band is illustrated, where the generated similarity map includes a plurality of single radiometric descriptors that each encodes multispectral radiometric descriptor information for a respective pixel of the image.

[0035] Reference will also now be made to Figure 3 which presents a schematic diagram of the pixels 306 of a portion of an input multispectral image 300 of a geographic area at each of a plurality of spectral bands 308, 312, 316 and a similarity map 320 of the pixels (e.g., similarity map 238 of Figure 2) that may be generated from radiometric descriptors of the pixels at each of the spectral bands. Specifically, the numbers 1-16 in Figure 3 represent pixel numbers (e.g., location IDs) of the pixels 306 in each of the spectral bands 308, 312, 316 (e.g., red, green, blue, and/or any other appropriate combination of bands) as well as in the similarity map 320. While generation of the similarity map 320 for pixels 1-16 of a portion of the input multispectral image 300 will be discussed, it is to be understood that in reality a similarity map 320 of all or substantially all of the pixels 306 of the entire input multispectral image 300 may be generated.

[0036] At 704 of the method 700 of Figure 7, a set (e.g., vector) of differences (absolute differences) for each pixel between a radiometric descriptor (e.g., pixel intensity) of the pixel and the radiometric descriptors (e.g., pixel intensities) of adjacent pixels (e.g., 4 neighbors, 8 neighbors, etc.) may be determined at each spectral band of the plurality of spectral bands of the input multispectral image. In the case of pixel "6" and its eight neighboring adjacent pixels in Figure 3, for instance, the three sets of differences for the three spectral bands 308, 312, 316 may be respectively represented by:

$$\begin{bmatrix} |f(6)_R - f(1)_R| \\ |f(6)_R - f(2)_R| \\ |f(6)_R - f(3)_R| \\ |f(6)_R - f(5)_R| \\ |f(6)_R - f(7)_R| \\ |f(6)_R - f(9)_R| \\ |f(6)_R - f(10)_R| \\ |f(6)_R - f(11)_R| \end{bmatrix} \quad \begin{bmatrix} |f(6)_G - f(1)_G| \\ |f(6)_G - f(2)_G| \\ |f(6)_G - f(3)_G| \\ |f(6)_G - f(5)_G| \\ |f(6)_G - f(7)_G| \\ |f(6)_G - f(9)_G| \\ |f(6)_G - f(10)_G| \\ |f(6)_G - f(11)_G| \end{bmatrix} \quad \begin{bmatrix} |f(6)_B - f(1)_B| \\ |f(6)_B - f(2)_B| \\ |f(6)_B - f(3)_B| \\ |f(6)_B - f(5)_B| \\ |f(6)_B - f(7)_B| \\ |f(6)_B - f(9)_B| \\ |f(6)_B - f(10)_B| \\ |f(6)_B - f(11)_B| \end{bmatrix}$$

Radiometric Descriptor Difference Set (RDDS) - Red Band ate pi Radiometric Descriptor Difference Set (RDDS) - Green Band ere Radiometric Descriptor Difference Set (RDDS) - Blue Band

descriptor (e.g., intensity) of a particular pixel at a particular spectral band.

[0037] A similar process may be performed for each of the other pixels in the input multispectral image. In the case of pixel "11," for instance the differences between the intensity of pixel 11 ($f(11)$) and those of each of pixels 6, 7, 8, 10, 12, 14, 15 and 16 may be determined at each of the three spectral bands to obtain three corresponding RDDSs for pixel 11. In any case, the method 700 may, in one arrangement, include obtaining 708 a minimum one of the differences in each RDDS and then equating 712 the largest of the minimum differences to a single radiometric descriptor for the pixel that broadly represents the radiometric descriptors of the pixel at all of the spectral bands of the input multispectral image. Again in the case of pixel 6, the single radiometric descriptor of pixel 6 ($s(6)$) may be represented as:

$$s(6) = MAX\{MIN\{RDDS\_RED_6\}, MIN\{RDDS\_GREEN_6\}, MIN\{RDDS\_BLUE_6\}\}$$

[0038] A similar process may be performed for each of the other pixels 306 in the input multispectral image to obtain single radiometric descriptors for each of the other pixels 306 which may be loaded 724 into a similarity map 320 (e.g., similarity map 238), where the similarity map 320 may be organized 728 into a plurality of connected components of a hierarchical image representation (e.g., step 604 of Figure 6). For instance, steps 708 and 712 of Figure 7 may be performed for each pixel of the input multispectral image to obtain a similarity map 320 of single radiometric descriptors from which a hierarchical image representation in the form of a Max-Tree may be generated.

[0039] In another arrangement, the method 700 may include obtaining 716 a maximum one of the differences in each RDDS and then equating 720 the smallest of the maximum differences to a single radiometric descriptor for the pixel that broadly represents the radiometric descriptors of the pixel at all of the spectral bands of the input multispectral image. Again in the case of pixel 6, the single radiometric descriptor of pixel 6 ($s(6)$) may be represented as:

$$s(6) = MIN\{MAX\{RDDS\_RED_6\}, MAX\{RDDS\_GREEN_6\}, MAX\{RDDS\_BLUE_6\}\}$$

[0040] A similar process may be performed for each of the other pixels 306 in the input multispectral image to obtain single radiometric descriptors for each of the other pixels 306 which may be loaded 724 into a dissimilarity map (e.g., dissimilarity map 238, similar to similarity map 320, but instead with dissimilarity values for the pixels such as D(1), D(2), etc.), where the dissimilarity map may be organized 728 into a plurality of connected components of a hierarchical image representation (e.g., step 604 of Figure 6). For instance, steps 716 and 720 of Figure 7 may be performed for each pixel of the input multispectral image to obtain a dissimilarity map of single radiometric descriptors from which a hierarchical image representation in the form of an Alpha Tree may be generated.

[0041] Returning to Figure 6, and after the pixels of the input multispectral image have been organized (e.g., based on their single radiometric descriptors of the similarity map 320 according to the method 700 of Figure 7) into connected components of a hierarchical image representation (e.g., connected components 404 of hierarchical image representation 400 of Figure 4), the method 600 may include deriving 608 (e.g., with construction engine 228) a set (e.g., vector) of radiometric descriptors for each connected component 404 (e.g., connected component radiometric descriptor vectors 252 of Figure 2). Stated differently, the method 600 may include deriving 608 a radiometric descriptor for each spectral band for each connected component, where the radiometric descriptor of each spectral band collectively represents the radiometric descriptors of all pixels of the connected component at the respective spectral band.

[0042] To derive a set of radiometric descriptors for a connected component 404, respective sets or vectors 250 of radiometric descriptors or attributes for each of the pixels of the connected component 404 may first be derived or otherwise obtained. In one arrangement, each set of radiometric descriptors for each pixel of the connected component 404 may be a vector of the

pixel's intensities over the plurality of spectral bands 226 of the input multispectral image 225. For instance, component 416 in Figure 4 includes pixels 11, 12, 15 and 16 from input multispectral image 300 of Figure 3. In this regard, the four vectors of pixel intensities over the three spectral bands 308, 312, 316 may be expressed as follows:

$$\begin{bmatrix} f(11)_R \\ f(11)_G \\ f(11)_B \end{bmatrix} \begin{bmatrix} f(12)_R \\ f(12)_G \\ f(12)_B \end{bmatrix} \begin{bmatrix} f(15)_R \\ f(15)_G \\ f(15)_B \end{bmatrix} \begin{bmatrix} f(16)_R \\ f(16)_G \\ f(16)_B \end{bmatrix}$$

[0043] In another arrangement, each set of radiometric descriptors for each pixel of the connected component 404 may be a vector of the pixel's intensity gradients over the plurality of spectral bands, wherein each intensity gradient is the difference between the pixel's intensity at one spectral band and the pixel's intensity at an adjacent spectral band. Again in relation to component 416 of Figure 4, the four vectors of pixel intensity gradients over the three spectral bands 308, 312, 316 may be expressed as follows:

$$\begin{bmatrix} f(11)_R - f(11)_G \\ f(11)_G - f(11)_B \end{bmatrix}$$

$$\begin{bmatrix} f(12)_R - f(12)_G \\ f(12)_G - f(12)_B \end{bmatrix}$$

$$\begin{bmatrix} f(15)_R - f(15)_G \\ f(15)_G - f(15)_B \end{bmatrix}$$

$$\begin{bmatrix} f(16)_R - f(16)_G \\ f(16)_G - f(16)_B \end{bmatrix}$$

[0044] In a further arrangement, each set of radiometric descriptors for each pixel of the connected component 404 may be a vector of "critical" or "characteristic" parameters for the pixel including a maximum gradient of the pixel (e.g., the response R), the gradient ID (the ID of the entry in the vector corresponding to the maximum gradient, or the scale S), and the value of the largest intensity among the two intensities in the maximum gradient (e.g., the level L). In the case of pixel "11" of component 416, and assuming the intensity gradient of the first entry was larger than the intensity gradient of the second entry and that $f(11)_G$ is larger than $f(11)_B$, such a vector may be expressed as follows:

$$\begin{bmatrix} f(11)_R - f(11)_G \\ 0 \\ f(11)_G \end{bmatrix}$$

where the ID of the first entry is 0, the ID of the second entry is 1, and so on. A similar process may be performed to obtain sets of radiometric descriptors for pixels 12, 15 and 16 of component 416.

[0045] Once respective vectors 250 of radiometric descriptors or attributes for each of the pixels of the connected component 416 have been derived or otherwise obtained, such sets of radiometric descriptors or attributes for the pixels may be used to generate a single set of radiometric descriptors for the component 416. Broadly, any appropriate method that produces an average value for an entry of the component vector from all respective vector entries from the pixels making up the component may be used. In one arrangement, each entry in a vector 252 of radiometric descriptors for component 416 may be a mean value (e.g., average) of the corresponding entries in the vectors of radiometric descriptors of the pixels making up the component. For instance, the first entry in the vector 252 for component 416 may be the mean of the first entries of the vectors 250 for the pixels, the second entry in the vector 252 for component 416 may be the mean of the second entries of the vectors 250 for the pixels, and so on. In the case of the vectors of pixel intensities of the pixels 11, 12, 15 and 16 over the spectral bands 308, 312, 316, for instance, such a vector for component 404 may be expressed as:

$$\begin{bmatrix} MEAN\{f(11)_R, f(12)_R, f(15)_R, f(16)_R\} \\ MEAN\{f(11)_G, f(12)_G, f(15)_G, f(16)_G\} \\ MEAN\{f(11)_B, f(12)_B, f(15)_B, f(16)_B\} \end{bmatrix}$$

[0046] Additional metrics may be used to evaluate the variability of the averaging function. In one arrangement, the value might be refined by revisiting each pixel making up the respective component if the average value for any entry of the component vector shows high variability (e.g., where the contribution of a particular pixel's vector entry may be ignored in the event the entry has a value above a certain threshold on the variability metric). One manner in which the variabilities of the various entries may be assessed is by obtaining the standard deviation of the corresponding entries in the vectors of radiometric descriptors of the pixels making up the component. Again in the case of the vectors of pixel intensities of the pixels 11, 12, 15 and 16 over the spectral bands 308, 312, 316, a vector of standard deviations for component 404 may be expressed as:

$$\begin{bmatrix} STD\_DEV(f(11)_R, f(12)_R, f(15)_R, f(16)_R) \\ STD\_DEV(f(11)_G, f(12)_G, f(15)_G, f(16)_G) \\ STD\_DEV(f(11)_B, f(12)_B, f(15)_B, f(16)_B) \end{bmatrix}$$

[0047] For instance, the entry of a particular pixel may be removed or ignored if the entry is more than a particular number of standard deviations above the average. Upon revisiting all pixels making up the component and refining the selection of contributors, the average may then be recomputed. Additional examples of variability metric are the range between extreme values around the average, the principal components of the values making up the average, and/or the like.

[0048] Similar processes may be performed or executed to obtain respective sets (e.g., vectors) of radiometric descriptors for each of the other connected components 404 of the hierarchical image representation 400. In any case, the method 600 of Figure 6 may also include ascertaining or otherwise obtaining 612 a set or vector 260 (see Figure 2) of radiometric descriptors for a reference multispectral image (e.g., reference image "chip") that includes one or more attributes sought to be identified in one or more portions of the input multispectral image 225 (e.g., in one or more of the connected components 404 of the hierarchical image representation 400 of Figure 4 of the input multispectral image 300 of Figure 3). Various types of reference chips corresponding to various different types of land types, land uses, etc. (e.g., bare soil, parking lots, etc.) for identifying such land or land uses in the input multispectral image 225 are envisioned and encompassed herein. The reference multispectral image may be an image from the same platform or sensor and of the same number of spectral bands as the input multispectral image.

[0049] In one arrangement, the reference multispectral image may be a small and/or defined portion of a multispectral image with contents (e.g., pixels) coinciding explicitly to the radiometric pattern of interest (e.g., explicitly to forest, explicitly to pavement, etc.), such as a portion of the input multispectral image 225 (e.g., defined by rectangular coordinates or the like), a defined portion of another multispectral image that is non-overlapping with the input multispectral image, etc. For instance, a set of radiometric descriptors may be derived for each pixel of the defined portion (e.g., pixel radiometric descriptor vectors 250 of Figure 2) and then the set of radiometric descriptors for the reference multispectral image may be derived from those of the pixels making up the defined portion. As discussed previously in relation to deriving the set of radiometric descriptors for each connected component of the hierarchical image representation using the sets of radiometric

descriptors of the pixels making up the connected component, the set of radiometric descriptors of the reference multispectral image may be a vector including the same number and type of entries as that of a particular connected component 404 of the input multispectral image to be analyzed (various analyses discussed in more detail below).

[0050] In another arrangement, the set of radiometric descriptors of the reference multispectral image may be equated to the set of radiometric descriptors of at least one of the components of the hierarchical image representation 400. As an overly simplistic example, and in the case where the reference multispectral image overlaps with the input multispectral image, assume connected component 420 includes pixels corresponding to the radiometric pattern of interest (i.e., corresponding to the land use/type/etc. sought to be identified in connected component 416 and the other connected components 404). In this regard, the reference multispectral image radiometric descriptor vector 260 (Figure 2) may be equated to the connected component radiometric descriptor vector 252 of connected component 420 (e.g., determined in step 608 of Figure 6). Of course, more complicated arrangements are also envisioned. For instance, assume connected components 420 and 424 collectively include pixels corresponding to the radiometric pattern of interest. In this regard, the connected component radiometric descriptor vectors 252 of connected components 420, 424 may be appropriately combined or fused into a single reference multispectral image radiometric descriptor vector 260 (e.g., via taking a mean or standard deviation of each of the first entries, each of the second entries, etc.).

[0051] In another arrangement, such as in the case where the reference multispectral image is non-overlapping with the 225 input multispectral image and thus where a hierarchical image representation 400 (e.g., and corresponding connected component radiometric descriptor vectors) may not have been constructed for the area encompassing the reference multispectral image, the pixels of the area may be organized 604 into another hierarchical image representation and then radiometric descriptor vectors or sets may be derived 608 for the connected components of the hierarchical image representation. The radiometric descriptor vector(s) of the connected components corresponding to the reference multispectral image may then be equated to or appropriately fused into a reference multispectral image radiometric descriptor vector 260 for the reference multispectral image.

[0052] With reference again to Figure 6, the method 600 may include determining 616 (e.g., via the analysis engine 232 of Figure 2) a similarity or proximity metric 264 (Figure 2) between the radiometric descriptor set of each connected component 404 of the input multispectral image and the radiometric descriptor set of the reference multispectral image. The determined similarity metrics 264 may be stored in memory 228 and/or in storage 212 of the system 200 in any appropriate manner. Broadly, increasing similarity or proximity corresponds to an increasing likelihood that the pixels of the particular connected component 404 under consideration associate to or otherwise identify the same land use or type as that of the pixels of the reference multispectral image. Various manners of determining such similarity metrics 264 are envisioned and included in the present disclosure. For instance, assume that x represents the radiometric descriptor vector 252 for the connected component 404 to be analyzed (e.g., for connected component 416 of Figure 5), y represents the radiometric descriptor vector 260 for the reference multispectral image, and n represents the total number of entries in each of the vectors. In one arrangement, the similarity metric may be the Euclidean distance (d) between the two vectors which may be expressed as:

$$d = \sqrt{\sum_{1}^{n} (x_i - y_i)^2}$$

[0053] In another arrangement, the similarity metric may be the Manhattan distance (d) between the two vectors which may be expressed as:

$$d = \sum_{1}^{n} |x_i - y_i|$$

[0054] In another arrangement, the similarity metric may be the Pearson correlation (d) between the two vectors which may be expressed as:

$$d = 1 - \frac{Z(x) \times Z(y)}{n}$$

where: $Z_i(x) = \frac{x_i - \bar{x}}{std(x)}$

$\bar{x}$: $mean$

$std(x)$: $standard\ deviation\ of\ x$

[0055] In another arrangement, the similarity metric may be the Cosine similarity ($cos(\Theta)$) between the two vectors which may be expressed as:

$$cos(\Theta) = \frac{x \times y}{\|x\| \times \|y\|}$$

[0056] In one arrangement, the determining 616 may include assigning components 404 a first normalized similarity metric score (e.g., 1) when the determined similarity metric 264 is above or below a threshold metric indicating that pixels of the component 404 identify the at least one land use or land type of the reference multispectral image (e.g., are a positive instance of the land use/type of the reference multispectral image) or a second normalized similarity metric score (e.g., 0) when the determined similarity metric is the other of above or below the threshold metric indicating that pixels of the component 404 do not identify the at least one land use or land type of the reference multispectral image (e.g., are a negative instance of the land use/type of the reference multispectral image). As an example, Figure 5 illustrates the hierarchical image representation 400 of Figure 4, where the connected components 404 include normalized scores as a result of performing a proximity analysis between radiometric descriptor sets of the connected components 404 and a radiometric descriptor set of a reference multispectral image (e.g., reference image radiometric descriptor vector 260). In one arrangement, users may empirically (or otherwise) determine threshold metric levels above which indicate (e.g., with 80%, 90%, etc. certainty or the like) that the pixels of the particular connected component 404 identify the same land use or land type as the reference multispectral image. In any case, the use of "similarity metric" or variations thereof herein may connote both raw similarity metrics and normalized similarity metrics (e.g., normalized scores).

[0057] Returning to Figure 6, the method 600 may then map 624 (e.g., via mapping engine 236 of Figure 2) the determined similarity metrics into a resultant image 268 of the geographic area of the input multispectral image. Broadly, each pixel in the input multispectral image 225 may acquire the similarity metric(s) 264 of particular connected component(s) 244/404 within which the pixel is found or contained. For instance, a particular land use or land type of the reference multispectral image (e.g., built-up, vegetation, etc.) may be associated with a particular color while an increasing intensity of the color may convey an increasing degree to which a particular pixel identifies the particular land use or land type of the reference multispectral image (e.g., as part of a "heat map").

When the similarity metrics 264 of the components 244/404 are normalized to one of two scores (e.g., to positive or negative instances of the land use/type of the reference multispectral image), a first marking (e.g., color, pattern, etc.) may indicate a positive instance of the particular land use/type and a different second marking (e.g., color, pattern, etc.) or no marking may indicate a negative instance of the particular land use/type.

[0058] In the case of the hierarchical image representation 400 being in the form of a Max-Tree, each component 404 may include pixels that are different (e.g., unique) from those of every other component 404 in the representation 400. In the case of an Alpha-Tree, the pixels of each leaf component 408 may be different from those of every other leaf component 408. In any case, the analysis engine 232 may recursively determine 616 the similarity metric (e.g., raw, normalized, etc.) for each component 404 of the hierarchical image representation 400 (e.g., starting with the leaf components 408 and moving towards the root component 412) and store the same while the mapping engine 236 may map the determined metrics into a resultant image 268.

[0059] In one arrangement, the input multispectral image 225 of the geographic area may be partitioned or segmented into a series of partitions, where each partition includes a plurality of adjacent pixels of the input multispectral image 225. That is, rather step 604 of Figure 6 including organizing the pixels of the input multispectral image into a plurality of components of a hierarchical image representation, step 604 may include pre-organizing the input multispectral image 225 into a partition of the image space, such as by mapping a grid of cells (e.g., squares, rectangles, etc.) over the image space of the geographic area. In this regard, step 608 of Figure 6 may include deriving a set of radiometric descriptors for each partition/cell of the grid such as by deriving respective sets/vectors of radiometric descriptors for each of the pixels of the partition/cell and using the vectors of radiometric descriptors for the various pixels to derive the set of radiometric descriptors for the partition/cell (e.g., as discussed previously in relation to deriving the sets of radiometric descriptors for the components of the hierarchical image representations).

[0060] As also discussed previously, a set of radiometric descriptors for a reference multispectral image (chip) may be acquired or obtained 612, similarity metrics between the sets of radiometric descriptors of each of the partitions and the set of radiometric descriptors of the reference multispectral image may be determined 616, and the similarity metrics may be mapped 620 into a

resultant image of the geographic areas indicating pixels of the input multispectral image that have the same land use or land type as that of the reference multispectral image (and/or indicating a degree to which pixels associate or convey the same land use or land type as that of the reference multispectral image). The automated characterization system 200 of Figure 2 would thus be largely similar to that as discussed previously. For instance, reference numeral 240 would be the above-discussed partition structure (rather than the previously-discussed hierarchical image representation) while the connected components 244 would be the partitions/cells of the partition structure rather than the nodes of the hierarchical image representation. Furthermore, the similiarity/dissimilarity maps 238 need not be computed and/or otherwise used in the case of the partition structure.

[0061] In some arrangements, similarity metrics 264 for two or more reference multispectral images may be determined for each component or partition of an input multispectral image 225 and combined into a single resultant image 268 in any appropriate manner. As an example, a first pass through the components 244/404 of a hierarchical image representation 240/400 or image space partition of an input multispectral image 225 may be made with a set 260 of radiometric descriptors of a first reference multispectral image for identifying pixels of the input multispectral image 225 that have the same land use or land type as that of the first reference multispectral image (and/or a relative degree to which pixels associate or convey the same land use or land type as that of the first reference multispectral image).

[0062] Another pass through the components 244/404 of the hierarchical image representation 240/400 or image space partition of the input multispectral image 225 may also be made with a set 260 of radiometric descriptors of a second reference multispectral image (e.g., including a different land use/type than that of the first reference multispectral image) for identifying pixels of the input multispectral image 225 that have the same land use or land type as that of the second reference multispectral image (and/or a relative degree to which pixels associate or convey the same land use or land type as that of the second reference multispectral image). For instance, pixels identifying the land use/type of the first reference multispectral image may be marked with a first color and pixels identifying the land use/type of the first reference multispectral image may be marked with a second color different than the first color. In one arrangement, an intensity of the color of a pixel may convey a relative degree to which the pixel convey or associates with the land use/type of the

corresponding multispectral reference image (e.g., where increasing intensity conveys higher degree of confidences that the pixel identifies the particular land use/type and vice versa).

[0063] The utilities and functionalities disclosed herein may be used to identify various types of land uses or land types in various contexts. For instance, assume the system 200 generates or otherwise acquires respective vectors 260 of radiometric descriptors for reference chips (reference multispectral images) identifying a respective plurality of types of mining materials (e.g., coal, metals, limestone, etc.). In this regard, a user may be able to select (e.g., on any appropriate user interface in communication with the system 200) one of the reference chips and the system may be configured to automatically generate a corresponding resultant image 268 (e.g., heat map) of a geographic area of an input multispectral image 225 indicating how similar various portions (e.g., pixels) of the input multispectral image 225 are to the selected reference chip. In one embodiment, a user may be able to select a particular portion of the input multispectral image 225 to serve as the reference chip (e.g., via clicking and dragging a cursor on a user interface to create a box about the particular portion), where the system may then automatically generate a vector 260 of radiometric descriptors for the selected areas to be used in the determination of similarity metrics 264 of the remainder of the input multispectral image 225.

[0064] In another arrangement, the system 200 may be used for agricultural monitoring. For instance, assume a government subsidizes some of the operations of a particular farm in exchange for the farmer growing a field (e.g., ½ mile x ½ mile) of soy beans. As part of the agreement, the government may want actual confirmation or at least reassurance that the entirety of the particular field is growing soy beans as opposed to only a portion of the field. In this regard, the input multispectral image may correspond to the entirety of the field while the reference multispectral image (reference chip) may correspond to a small portion of the field known to only include soybeans. For instance, a user may appropriately select a portion of an input multispectral image 225 of the field to serve as the reference chip.

[0065] After generating the respective sets 252 of radiometric descriptors of the components (or sets of radiometric descriptors for image partitions) of the input multispectral image 225 and the set of radiometric descriptors 260 of the reference chip, similarity metrics 264 may be generated and a heat map 268 of the field displayed conveying a degree to which the field is growing soy beans. As

another example, a user may selected a desired reference chip (e.g., soy beans, corn, wheat, etc.) receive a heat map of how different other crops in the vicinity are to the selected crop, how healthy and/or unhealthy are crops of the same type, how mixed with other vegetation is the selected crop type, and/or the like. As a further example, a user may appropriate select a reference chip corresponding to a desired type of building material and receive a heat map of portions of an input multispectral image similar to the selected reference chip as part of classification of buildings, detection of buildings and constructed facilities (e.g., parking lots, road networks, etc.), and/or the like.

[0066] It will be readily appreciated that many deviations and/or additions may be made from or to the specific embodiments disclosed in the specification without departing from the spirit and scope of the invention. In one arrangement, any of the resultant images or heat maps may incorporate any appropriate textural attributes of the pixels of the input multispectral image. For instance, one or more geometric descriptors (e.g., contour, smoothness, linearity, area, etc.) may be derived for each of the plurality of components or partitions in any appropriate manner, where the one or more geometric descriptors may be conveyed in the resultant image 268 or heat map in any appropriate manner.

[0067] Embodiments disclosed herein can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer-readable medium for execution by, or to control the operation of, data processing apparatus. For example, the logic or software of the construction engine 228, analysis engine 232 and mapping engine 236 responsible for the various functionalities disclosed herein may be provided in such computer-readable medium of the automated categorization system 200 and executed by the processor 208 as appropriate. The computer-readable medium can be a machine-readable storage device, a machine-readable storage substrate, a non-volatile memory device, a composition of matter affecting a machine-readable propagated signal, or a combination of one or more of them. In this regard, the system 200 may encompass one or more apparatuses, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. In addition to hardware, the system 200 may include code that creates an execution environment for the computer program in question, e.g., code that constitutes processor

firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[0068] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). Processors suitable for the execution of a computer program may include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. Generally, the elements of a computer are one or more processors for performing instructions and one or more memory devices for storing instructions and data. The techniques described herein may be implemented by a computer system configured to provide the functionality described.

[0069] While this specification contains many specifics, these should not be construed as limitations on the scope of the disclosure or of what may be claimed, but rather as descriptions of features specific to particular embodiments of the disclosure. Furthermore, certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0070] The above described embodiments including the preferred embodiment and the best mode of the invention known to the inventor at the time of filing are given by illustrative examples only.

WHAT IS CLAIMED IS:

1. A method for use in classifying areas of interest in overhead imagery, comprising:

organizing, using a processor, a plurality of pixels of at least one input multispectral image of a geographic area into a plurality of components of a hierarchical image representation;

deriving, using the processor, at least one set of radiometric descriptors for each component of the plurality of components;

obtaining at least one set of radiometric descriptors for a reference multispectral image, wherein pixels of the reference multispectral image identify at least one land use or land type;

determining, using the processor, for each component of the hierarchical image representation structure, a similarity metric between the set of radiometric image descriptors for the component and the set of radiometric image descriptors of the reference multispectral image, wherein the determined similarity metrics indicate a degree to which the pixels of each component identify the at least one land use or land type of the reference multispectral image.

2. The method of claim 1, further including:

obtaining a single radiometric descriptor for each pixel that is representative of a plurality of radiometric descriptors of the pixel at a respectively plurality of spectral bands of the input multispectral image, wherein the organizing utilizes the single radiometric descriptors of the plurality of pixels to generate the hierarchical image representation.

3. The method of claim 2, wherein the obtaining the single radiometric descriptor for each pixel includes:

determining a minimum difference between the radiometric descriptor of the pixel and the radiometric descriptors of adjacent pixels at each spectral band of the spectral bands, wherein the single radiometric descriptor is the largest difference of the minimum differences at each spectral band of the plurality of spectral bands.

4. The method of claim 2, wherein the obtaining the single radiometric descriptor for each pixel includes:

determining a maximum difference between the radiometric descriptor of the pixel and the radiometric descriptors of adjacent pixels at each spectral band of the spectral bands, wherein

the single radiometric descriptor is the smallest difference of the maximum differences at each spectral band of the plurality of spectral bands.

5. The method of any of claims 2-4, wherein the deriving includes, for each pixel of the component:

obtaining a set of radiometric descriptors for the pixel, wherein each set includes a plurality of entries; and

using the sets of radiometric descriptors for the pixels of the component to derive the set of radiometric descriptors for the component.

6. The method of claim 5, wherein the set of radiometric descriptors for each pixel is a vector of intensities of the pixel at each spectral band of a plurality of spectral bands of the input multispectral image.

7. The method of claim 5, wherein the set of radiometric descriptors for each pixel is a vector of intensity gradients of the pixel over the plurality of spectral bands, wherein each intensity gradient is the difference between the intensity of the pixel at one spectral band and the intensity of the pixel at an adjacent spectral band.

8. The method of claim 5, wherein the using includes:

obtaining means of each respective entry of the sets of radiometric descriptors of the pixels of the component to obtain radiometric descriptor mean values for corresponding respective entries of the set of radiometric descriptors for the component.

9. The method of claim 8, further including:

obtaining variability metrics of each respective entry of the sets of radiometric descriptors of the pixels of the component to obtain radiometric descriptor variability metrics for the corresponding respective entries of the set of radiometric descriptors for the component;

analyzing each entry of each set of radiometric descriptors of the pixels in view of the respective radiometric descriptor variability metric of the corresponding entry of the set of radiometric descriptors for the component; and

making a decision as to whether or not to remove each entry of each set of radiometric descriptors of the pixels from the obtaining means step based on a result of the analyzing step.

10. The method of claim 9, wherein the analyzing and making steps respectively include:

determining whether or not each entry of each set of radiometric descriptors of the pixels is above a threshold value of the respective radiometric descriptor variability metric of the corresponding entry of the set of radiometric descriptors for the component; and

removing an entry of a set of radiometric descriptors of the pixels when the entry is above the threshold value.

11. The method of claim 9, wherein the variability metrics including standard deviations, a range between extreme entries around the mean, or principal entries making up the mean.

12. The method of claim 1, wherein the obtaining at least one set of radiometric descriptors for the reference multispectral image includes:

equating the set of radiometric image descriptors of at least one of the components of the hierarchical image representation of the input multispectral image to the set of radiometric descriptors of the reference multispectral image.

13. The method of claim 1, wherein the obtaining at least one set of radiometric descriptors for the reference multispectral image includes:

obtaining a set of radiometric descriptors for each pixel of a defined portion of a multispectral image, wherein each set includes a plurality of entries; and

deriving the set of radiometric descriptors for the reference multispectral image using the sets of radiometric descriptors for the pixels of the defined portion.

14. The method of claim 13, wherein the multispectral image is a portion of the input multispectral image.

15. The method of claim 14, wherein the multispectral image is non-overlapping with the input multispectral image.

16. The method of claim 13, further including:

determining, for each pixel of the defined portion, whether its set of radiometric descriptors deviates from the set of radiometric descriptors of the reference multispectral image by more than a threshold; and

re-deriving the set of radiometric descriptors for the reference multispectral image free of any sets of radiometric descriptors of pixels determined to deviate from the derived set of radiometric descriptors of the reference multispectral image by more than a threshold.

17. The method of claim 1, wherein the determining the similarity metric includes:

ascertaining at least one of a Euclidean Distance, a Manhattan Distance, a Pearson Correlation, or a Cosine Similarity between the set of radiometric image descriptors for the component and the set of radiometric image descriptors of the multispectral reference image, wherein the at least one of the Euclidean Distance, the Manhattan Distance, the Pearson Correlation, or the Cosine Similarity is the similarity metric.

18. The method of claim 1, further including:

mapping the similarity metrics into a resultant image of the geographic area.

19. The method of claim 18, wherein the mapping includes:

applying a marking to pixels in the input multispectral image that is indicative of the similarity metric of a component within which the pixel is found.

20. The method of claim 19, wherein the marking includes at least one of a color, pattern, or shade.

21. The method of claim 18, wherein the determining the similarity metric further includes:

assigning the component of the hierarchical image representation structure a first normalized similarity metric score when the similarity metric is above a threshold similarity metric indicating that pixels of the component identify the at least one land use or land type of the reference multispectral image; and

assigning the component of the hierarchical image representation structure a second normalized similarity metric score when the similarity metric is below a threshold similarity metric indicating that pixels of the component do not identify the at least one land use or land type of the reference multispectral image.

22. The method of claim 1, wherein the reference multispectral image is a first reference multispectral image, wherein the pixels of the reference multispectral image identify at least one first land use or land type, wherein the similarity metrics are first similarity metrics, and wherein the method further includes:

obtaining at least one set of radiometric descriptors for a second reference multispectral image, wherein pixels of the reference multispectral image identify at least one second land use or land type, and wherein the first land use or land type is different than the second land use or land type;

determining, using the processor, for each component of the hierarchical image representation structure, a second similarity metric between the set of radiometric image descriptors for the component and the set of radiometric image descriptors of the second reference multispectral image, wherein the determined second similarity metrics indicate a degree to which the pixels of each component identify the at least one second land use or land type; and

mapping the first and second similarity metrics into a resultant image of the geographic area.

23. The method of claim 22, wherein the mapping includes:

applying a first marking to at least some of the pixels in the input multispectral image that is indicative of the first similarity metric of a component within which the pixel is found; and

applying a second marking to at least some of the pixels in the input multispectral image that is indicative of the second similarity metric of a component within which the pixel is found, wherein the first and second markings are different.

24. The method of claim 23, wherein the first and second markings include first and second different colors, and wherein a relative intensity of each of the first and second markings indicates a degree to which a respective pixels indicates the first or second land use or land type.

25. The method of claim 1, wherein the hierarchical image representation is a Max-Tree, a Min-Tree, or an Alpha Tree.

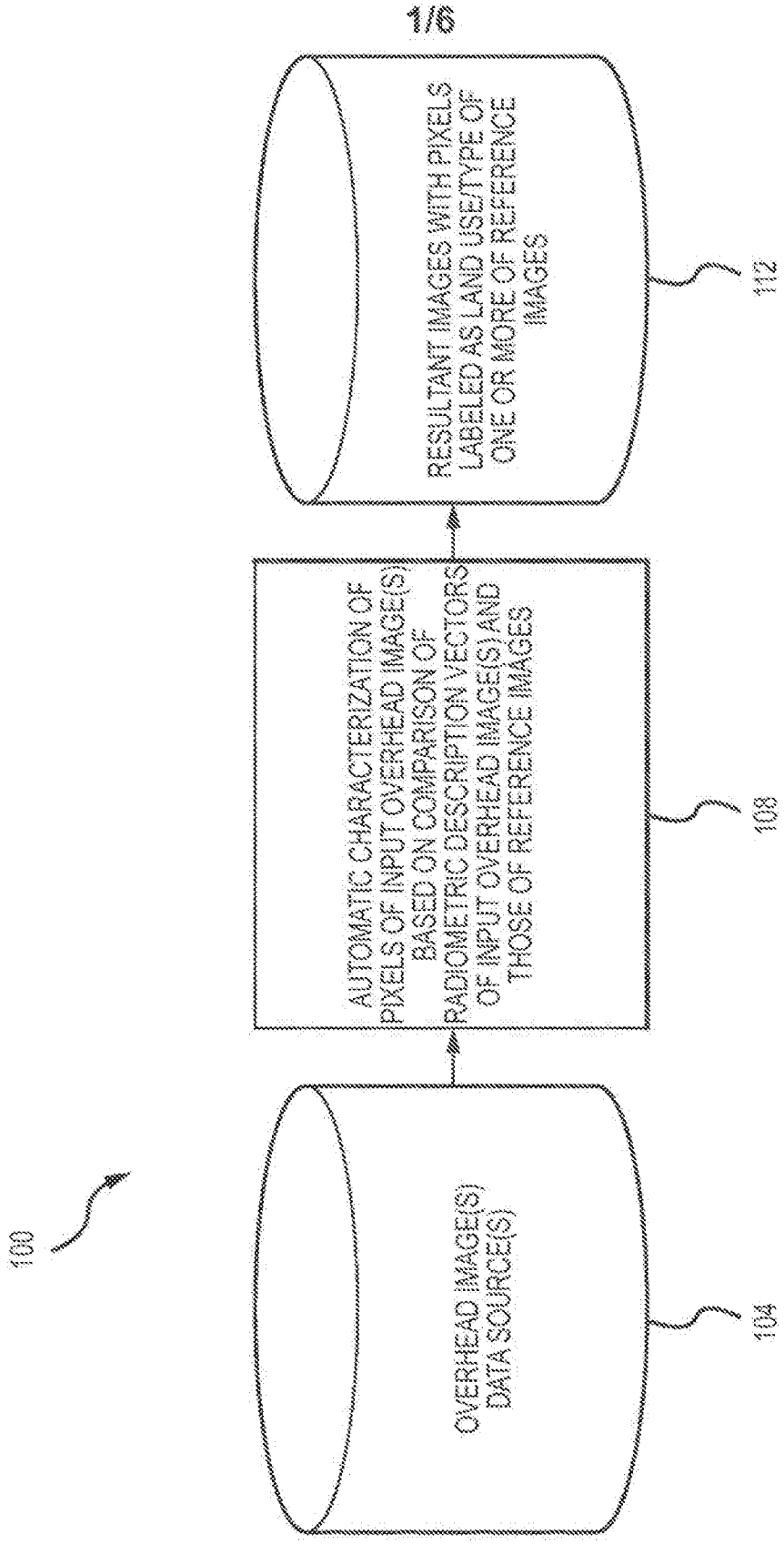

100
OVERHEAD IMAGE(S) DATA SOURCE(S)
104
AUTOMATIC CHARACTERIZATION OF PIXELS OF INPUT OVERHEAD IMAGE(S) BASED ON COMPARISON OF RADIOMETRIC DESCRIPTION VECTORS OF INPUT OVERHEAD IMAGE(S) AND THOSE OF REFERENCE IMAGES
108
RESULTANT IMAGES WITH PIXELS LABELED AS LAND USE/TYPE OF ONE OR MORE OF REFERENCE IMAGES
112

FIG. 1

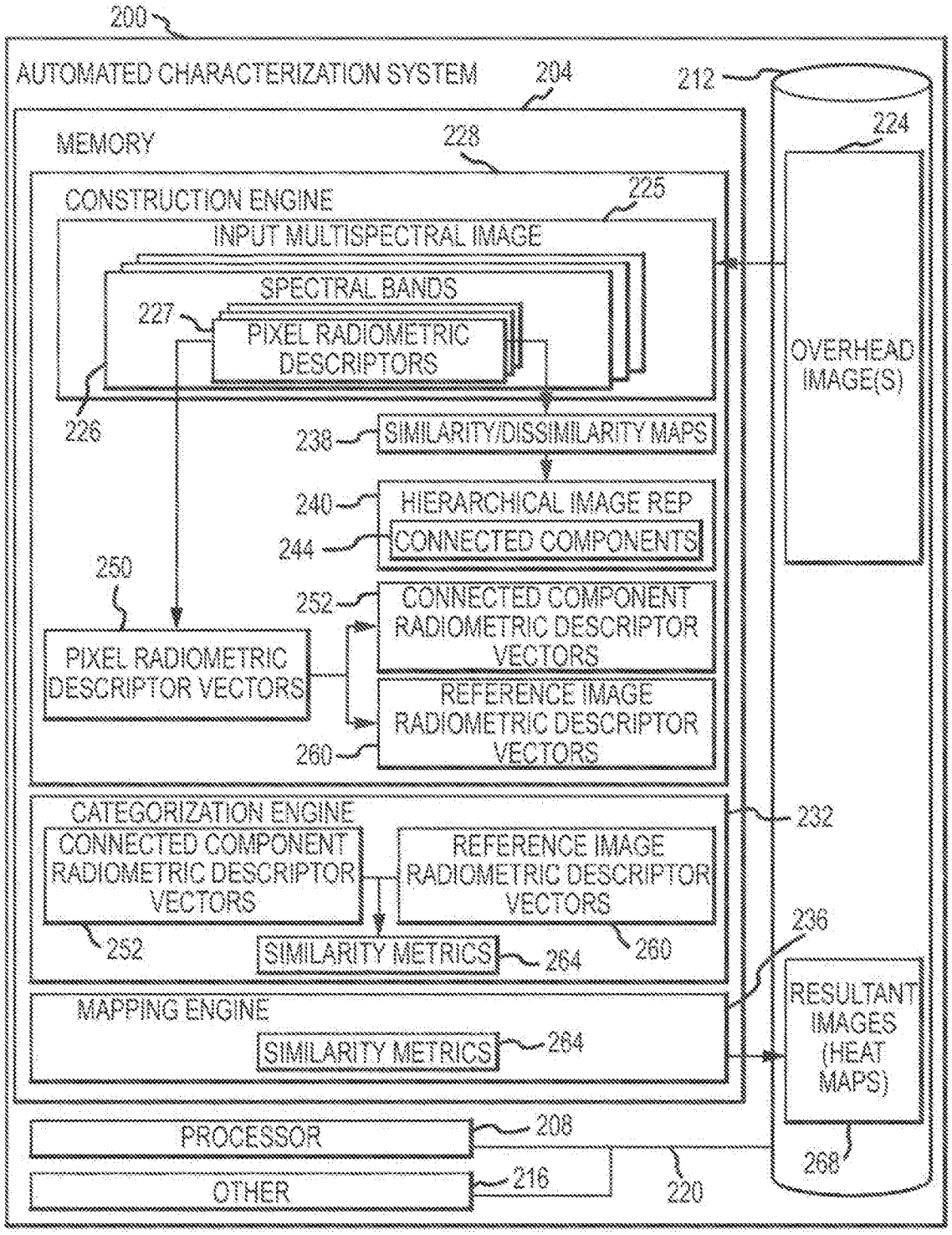

200
AUTOMATED CHARACTERIZATION SYSTEM
204
212
224
MEMORY
228
CONSTRUCTION ENGINE
225
INPUT MULTISPECTRAL IMAGE
SPECTRAL BANDS
227
PIXEL RADIOMETRIC DESCRIPTORS
OVERHEAD IMAGE(S)
226
238
SIMILARITY/DISSIMILARITY MAPS
240
HIERARCHICAL IMAGE REP
244
CONNECTED COMPONENTS
250
252
CONNECTED COMPONENT RADIOMETRIC DESCRIPTOR VECTORS
PIXEL RADIOMETRIC DESCRIPTOR VECTORS
REFERENCE IMAGE RADIOMETRIC DESCRIPTOR VECTORS
260
CATEGORIZATION ENGINE
232
CONNECTED COMPONENT RADIOMETRIC DESCRIPTOR VECTORS
REFERENCE IMAGE RADIOMETRIC DESCRIPTOR VECTORS
236
252
SIMILARITY METRICS
264
260
RESULTANT IMAGES (HEAT MAPS)
MAPPING ENGINE
SIMILARITY METRICS
264
PROCESSOR
208
OTHER
216
220
268

FIG.2

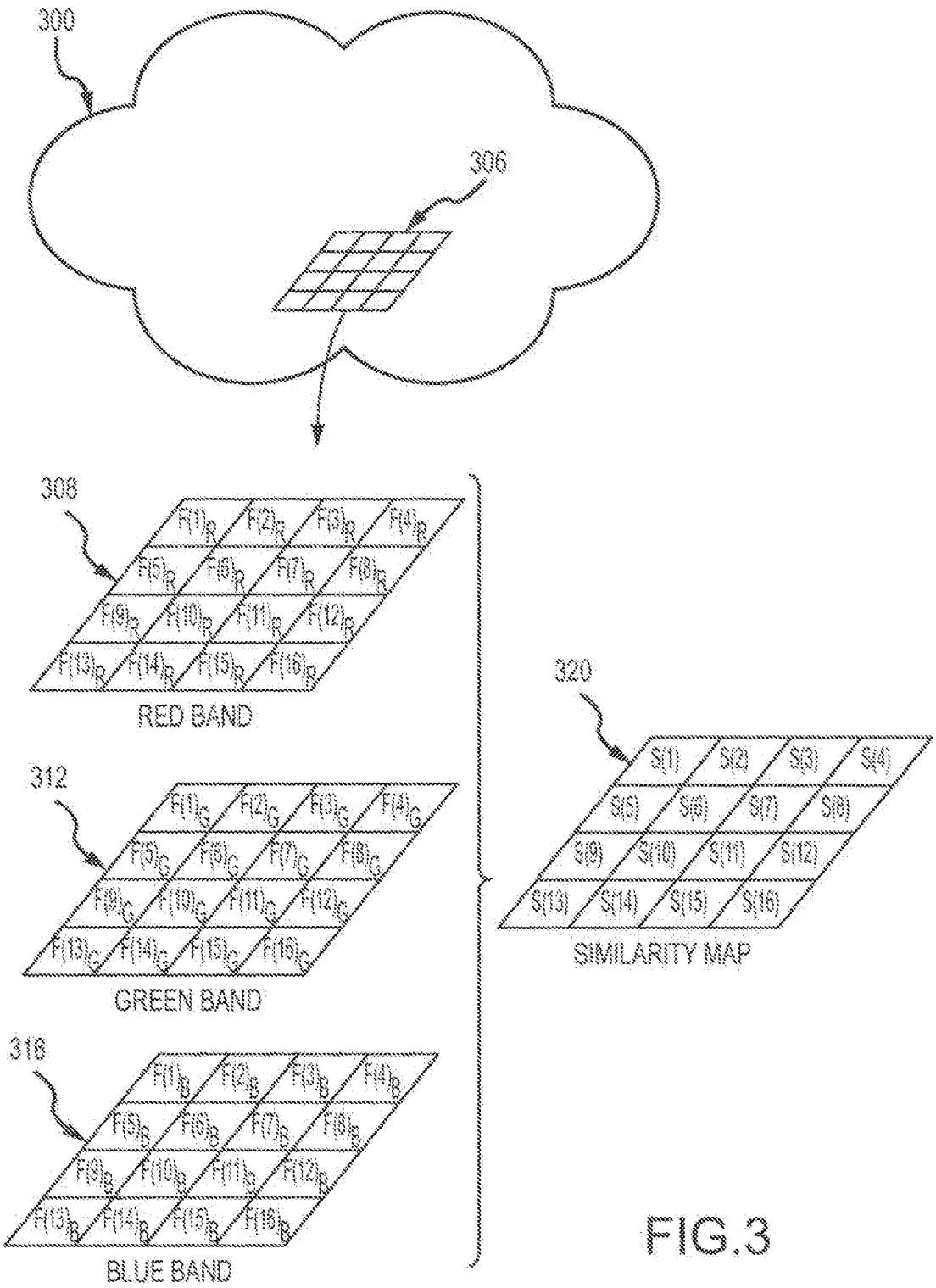

300
306
308
RED BAND
312
GREEN BAND
316
BLUE BAND
320
SIMILARITY MAP

FIG.3

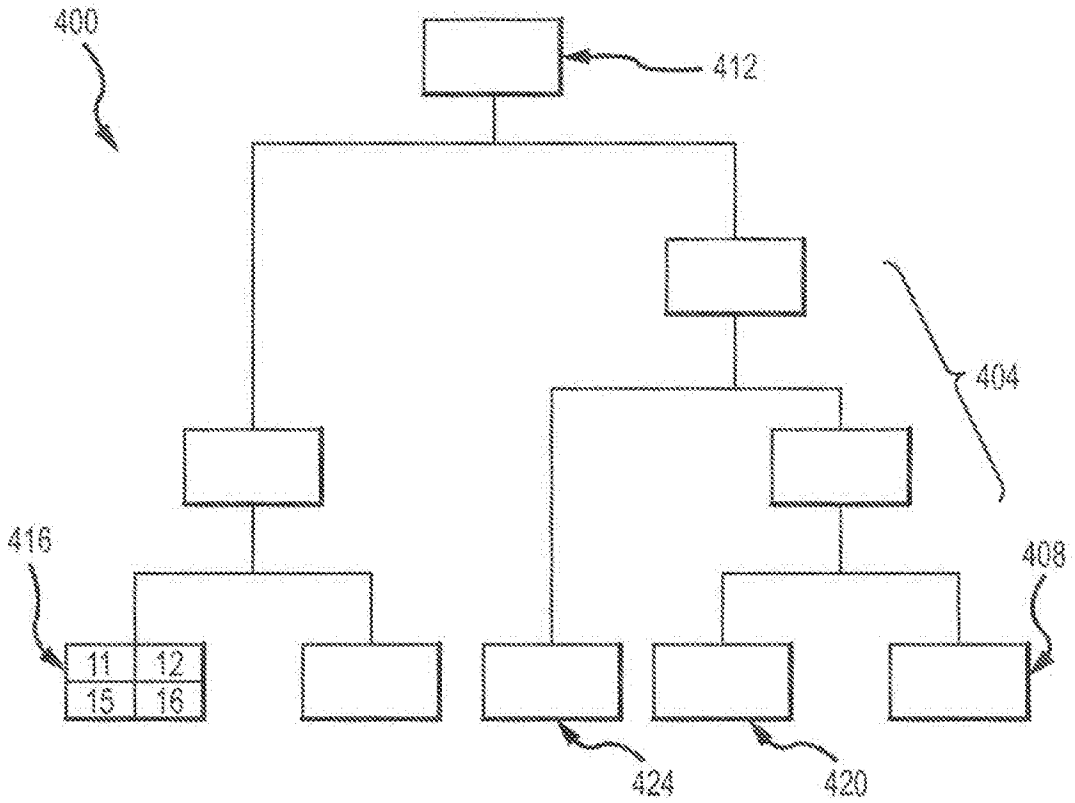


FIG.4

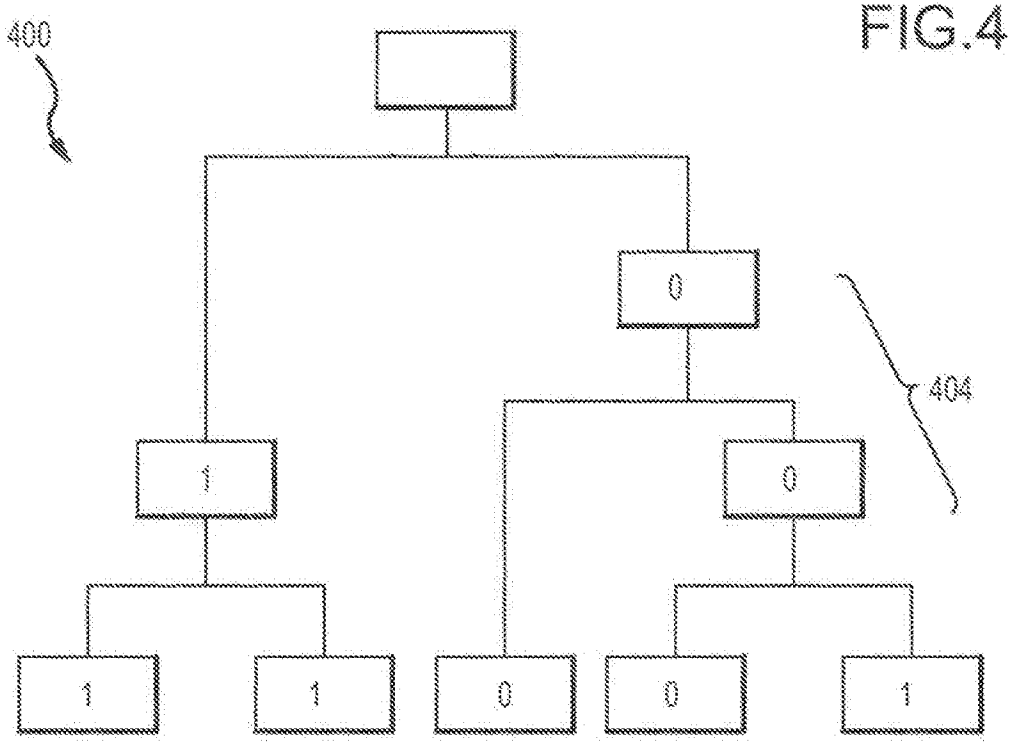


FIG.5

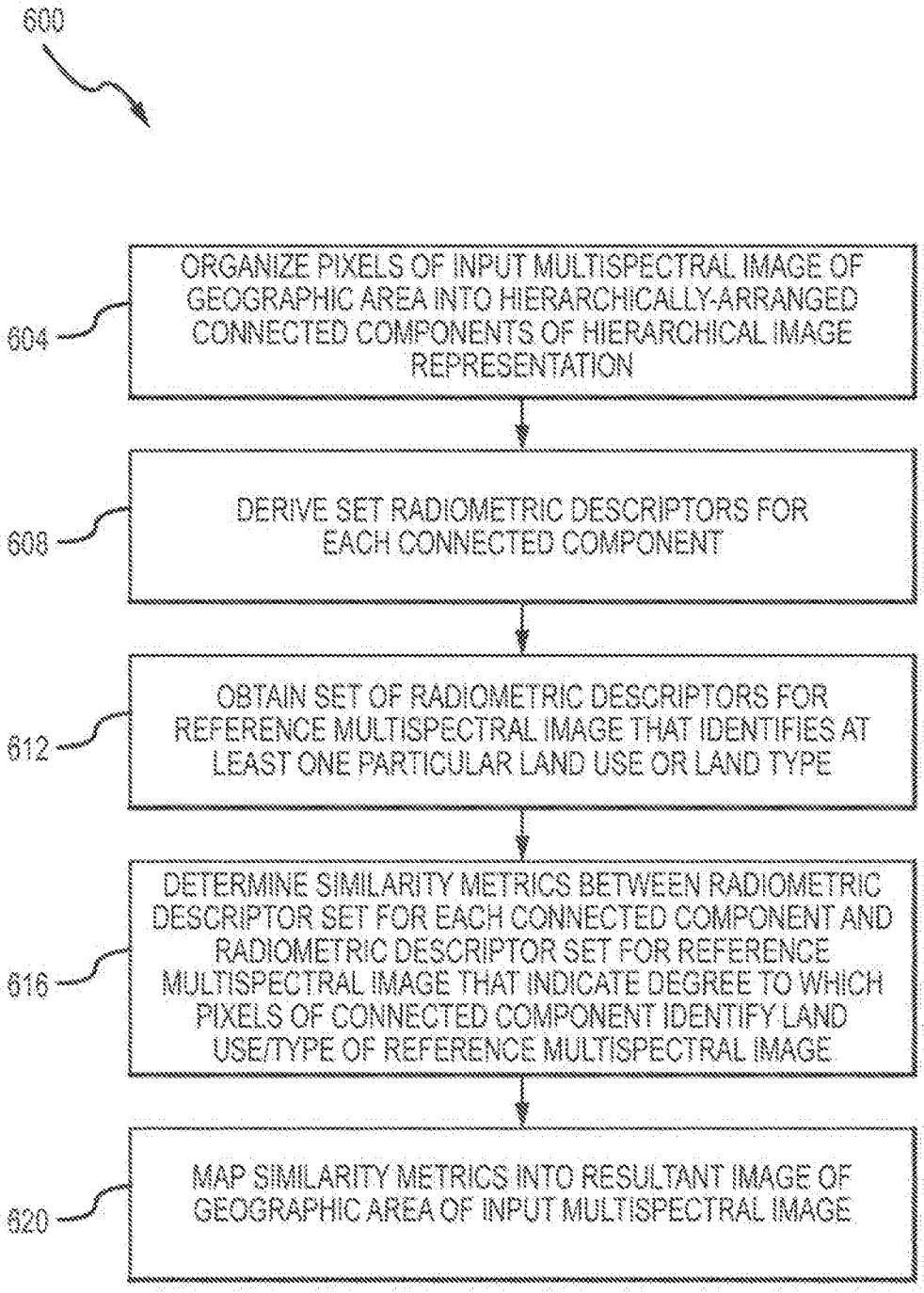

600
604
ORGANIZE PIXELS OF INPUT MULTISPECTRAL IMAGE OF GEOGRAPHIC AREA INTO HIERARCHICALLY-ARRANGED CONNECTED COMPONENTS OF HIERARCHICAL IMAGE REPRESENTATION
608
DERIVE SET RADIOMETRIC DESCRIPTORS FOR EACH CONNECTED COMPONENT
612
OBTAIN SET OF RADIOMETRIC DESCRIPTORS FOR REFERENCE MULTISPECTRAL IMAGE THAT IDENTIFIES AT LEAST ONE PARTICULAR LAND USE OR LAND TYPE
616
DETERMINE SIMILARITY METRICS BETWEEN RADIOMETRIC DESCRIPTOR SET FOR EACH CONNECTED COMPONENT AND RADIOMETRIC DESCRIPTOR SET FOR REFERENCE MULTISPECTRAL IMAGE THAT INDICATE DEGREE TO WHICH PIXELS OF CONNECTED COMPONENT IDENTIFY LAND USE/TYPE OF REFERENCE MULTISPECTRAL IMAGE
620
MAP SIMILARITY METRICS INTO RESULTANT IMAGE OF GEOGRAPHIC AREA OF INPUT MULTISPECTRAL IMAGE

FIG.6

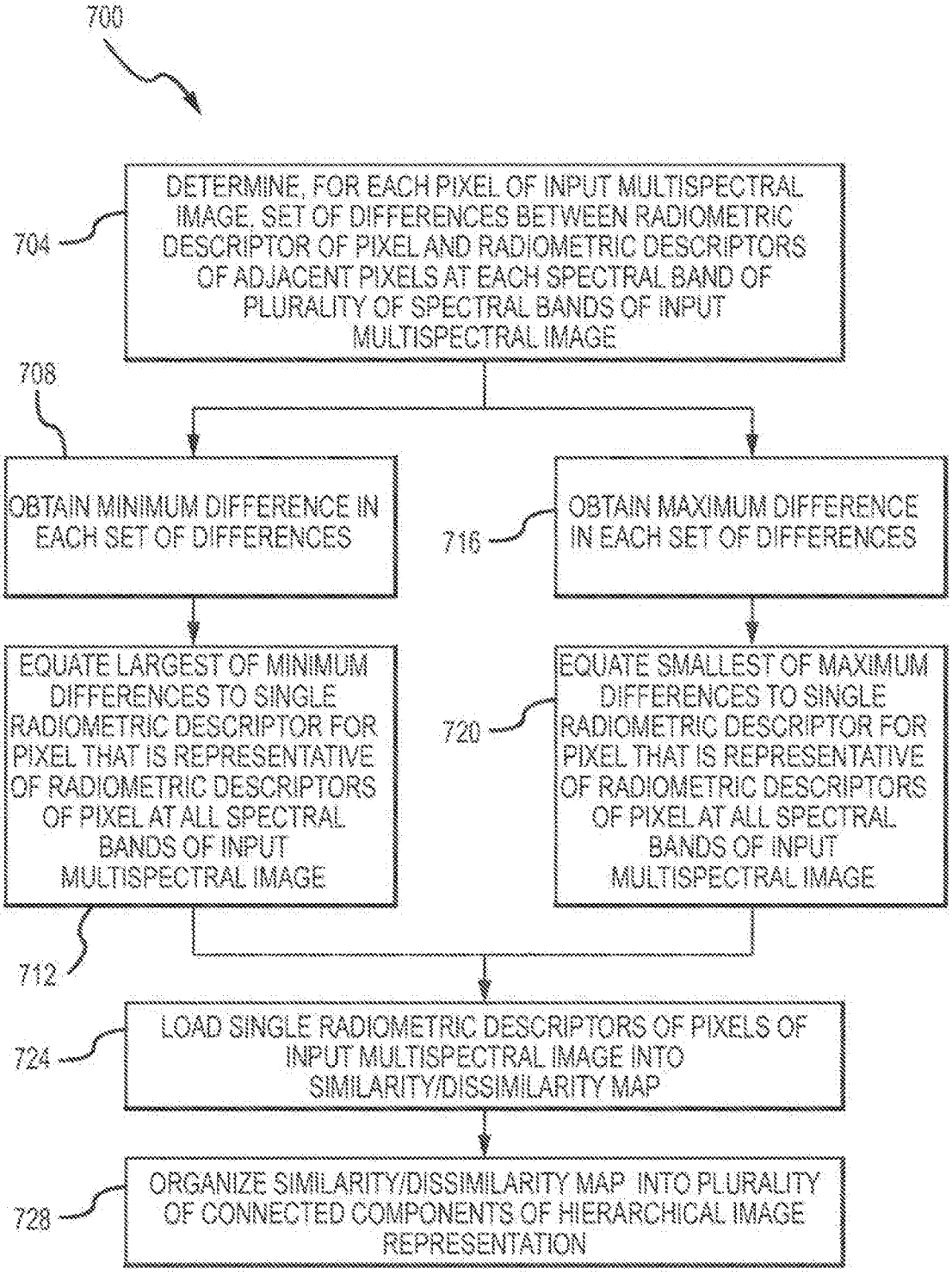

700
704
DETERMINE, FOR EACH PIXEL OF INPUT MULTISPECTRAL IMAGE, SET OF DIFFERENCES BETWEEN RADIOMETRIC DESCRIPTOR OF PIXEL AND RADIOMETRIC DESCRIPTORS OF ADJACENT PIXELS AT EACH SPECTRAL BAND OF PLURALITY OF SPECTRAL BANDS OF INPUT MULTISPECTRAL IMAGE
708
OBTAIN MINIMUM DIFFERENCE IN EACH SET OF DIFFERENCES
716
OBTAIN MAXIMUM DIFFERENCE IN EACH SET OF DIFFERENCES
EQUATE LARGEST OF MINIMUM DIFFERENCES TO SINGLE RADIOMETRIC DESCRIPTOR FOR PIXEL THAT IS REPRESENTATIVE OF RADIOMETRIC DESCRIPTORS OF PIXEL AT ALL SPECTRAL BANDS OF INPUT MULTISPECTRAL IMAGE
720
EQUATE SMALLEST OF MAXIMUM DIFFERENCES TO SINGLE RADIOMETRIC DESCRIPTOR FOR PIXEL THAT IS REPRESENTATIVE OF RADIOMETRIC DESCRIPTORS OF PIXEL AT ALL SPECTRAL BANDS OF INPUT MULTISPECTRAL IMAGE
712
724
LOAD SINGLE RADIOMETRIC DESCRIPTORS OF PIXELS OF INPUT MULTISPECTRAL IMAGE INTO SIMILARITY/DISSIMILARITY MAP
728
ORGANIZE SIMILARITY/DISSIMILARITY MAP INTO PLURALITY OF CONNECTED COMPONENTS OF HIERARCHICAL IMAGE REPRESENTATION

FIG.7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/052291

A. CLASSIFICATION OF SUBJECT MATTER

G06T 7/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06T 7/00; G06K 9/46; G06T 3/40; G06K 9/00; G06K 9/34; G09G 5/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & keywords: geospatial, overhead, imagery, radiometric descriptor, similarity metric, hierarchical image, reference multispectral image, classify, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 8682079 B1 (DIGITALGLOBE, INC.) 25 March 2014 See column 9, line 38 - column 10, line 7; claims 1, 10; and figure 3.	1-25
A	US 2011-0075882 A1 (TAO GAO et al.) 31 March 2011 See paragraphs [0048]-[0073]; claim 1; and figures 1-4.	1-25
A	RAKESH KUMAR MISHRA, "automatic moving vehicles information extraction from single pass worldview 2 imagery", International Archives of the Photogrammetry, Remote Sensing and spatial Information Sciences, 01 September 2012 (http://www.isprs-archives-photogramm-remote-sens-spatial-inf-sci.net) See pages 1-6.	1-25
A	US 2011-0102461 A1 (STEPHEN SCHULTZ et al.) 05 May 2011 See paragraphs [0047]-[0088]; claim 1; and figures 1-2.	1-25
A	US 2014-0029867 A1 (PICTOMETRY INTERNATIONAL CORP.) 30 January 2014 See paragraphs [0053]-[0084]; claims 1, 8; and figures 2A-4.	1-25

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
05 January 2016 (05.01.2016)	**06 January 2016 (06.01.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

Form PCT/ISA/210 (second sheet) (January 2015)

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8682079 B1	25/03/2014	US 2015-0023550 A1	22/01/2015
		US 9031325 B2	12/05/2015
		WO 2015-009798 A1	22/01/2015
US 2011-0075882 A1	31/03/2011	JP 05360989 B2	04/12/2013
		JP 2011-076178 A	14/04/2011
		US 8483425 B2	09/07/2013
US 2011-0102461 A1	05/05/2011	AU 2007-289120 A1	06/03/2008
		AU 2007-289120 B2	12/04/2012
		AU 2007-289120 C1	06/09/2012
		CA 2662355 A1	06/03/2008
		CA 2662355 C	06/05/2014
		EP 2057585 A2	13/05/2009
		EP 2057585 A4	13/03/2013
		JP 04981135 B2	18/07/2012
		JP 2010-503078 A	28/01/2010
		MX 2009002339 A	23/03/2009
		US 2008-0122994 A1	29/05/2008
		US 2012-0201478 A1	09/08/2012
		US 2013-0208096 A1	15/08/2013
		US 2014-0126838 A1	08/05/2014
		US 7873238 B2	18/01/2011
		US 8081841 B2	20/12/2011
		US 8452125 B2	28/05/2013
		US 8660382 B2	25/02/2014
		WO 2008-028040 A2	06/03/2008
US 2014-0029867 A1	30/01/2014	EP 2221792 A2	18/05/2011
		US 2010-034483 A1	11/02/2010
		US 2013-315503 A1	26/11/2013
		US 8588547 B2	19/11/2013
		US 8630510 B2	14/01/2014
		US 9147276 B2	29/09/2015
		WO 2010-017255 A2	11/02/2010

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 March 2017 (09.03.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/040691 A1

(51) International Patent Classification:
G06K 9/62 (2006.01) *G06N 3/02* (2006.01)
G06K 9/66 (2006.01) *H01Q 5/22* (2015.01)
G06T 1/40 (2006.01)

(21) International Application Number:
PCT/US2016/049739

(22) International Filing Date:
31 August 2016 (31.08.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/212,424 31 August 2015 (31.08.2015) US
62/315,180 30 March 2016 (30.03.2016) US

(71) Applicant: **CAPE ANALYTICS, INC.** [US/US]; Town & Country Village, 855 El Camino Real, Building 3, Suite 405, Palo Alto, CA 94301 (US).

(72) Inventors: **KOTTENSTETTE, Ryan**; 1635 Marymeade Lane, Los Altos, CA 94024 (US). **LORENZEN, Peter**; 465 Gabilan Street, Apt. 8, Los Altos, CA 94022 (US). **GEDIKLI, Suat**; Prinzregentenstrasse 154, 81677 Munich (DE).

(74) Agents: **MAK, Evelyn, C.** et al.; Wilmer Cutler Pickering Hale And Dorr LLP, 60 State Street, Boston, MA 02109 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) Title: SYSTEMS AND METHODS FOR ANALYZING REMOTE SENSING IMAGERY

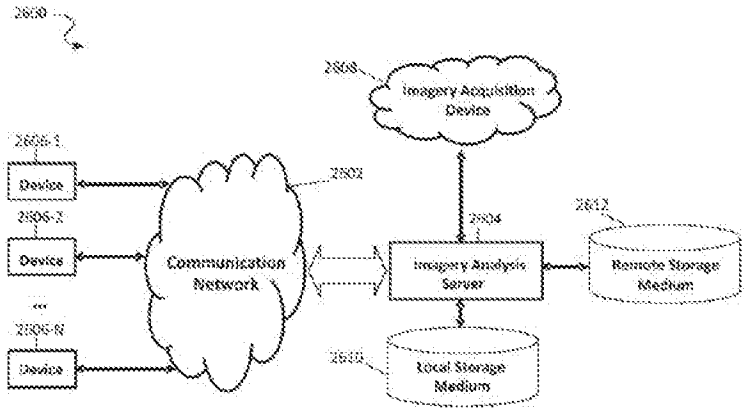


FIG. 26

(57) Abstract: Disclosed systems and methods relate to remote sensing, deep learning, and object detection. Some embodiments relate to machine learning for object detection, which includes, for example, identifying a class of pixel in a target image and generating a label image based on a parameter set. Other embodiments relate to machine learning for geometry extraction, which includes, for example, determining heights of one or more regions in a target image and determining a geometric object property in a target image. Yet other embodiments relate to machine learning for alignment, which includes, for example, aligning images via direct or indirect estimation of transformation parameters.

WO 2017/040691 A1

SYSTEMS AND METHODS FOR ANALYZING REMOTE SENSING IMAGERY

RELATED APPLICATION

[0001] This application claims priority to the following applications, the entire contents of which are incorporated herein by reference: U.S. Provisional Patent Application No. 62/212,424, titled "Systems And Methods For Analyzing Remote Sensing Imagery," which was filed on August 31, 2015; and U.S. Provisional Patent Application No. 62/315,180, titled "Systems And Methods For Analyzing Remote Sensing Imagery," which was filed on March 30, 2016.

BACKGROUND OF THE INVENTION

Technical Field

[0002] Embodiments of the present disclosure relate to the field of remote sensing, deep learning, and object detection.

Description of the Related Art

[0003] Structural and geospatial data are valuable in areas such as, for example, real estate transaction, planning, and/or insurance. Non-limiting examples of the structural and geospatial data include the following: the area of real property including land and/or buildings; the square footage of a building; the roof size and/or type; the presence of a pool and its size and/or location; and the presence of trees and its type, size, and/or location.

[0004] Traditionally, the structural and geospatial information can be obtained by (1) manually checking real estate records from relevant agencies; or (2) manually surveying the underlying real properties. These traditional methods suffer from a number of drawbacks and deficiencies. First, the records can be out of date, missing, or destroyed. Second, the manual checking and surveying are labor intensive and costly. Third, surveying area such as the roof of a building or the crown of a tree can be dangerous.

[0005] Therefore, there is a need in the art to provide systems and methods for obtaining structural and geospatial data that overcome these drawbacks and deficiencies.

SUMMARY

[0006] In accordance with the disclosed subject matter, systems, methods, and computer readable media are provided for remote sensing, deep learning, and detecting object.

[0007] A method of identifying a class of a pixel in a target image according to one embodiment of the present disclosure can include receiving, at an object detector, training images, wherein each training image includes pixels; receiving, at the object detector, labels, wherein each label associates a class to each of the pixels of the training images; creating, at the object detector, a classifier configured to identify the class of each pixel based on the training images and the labels; receiving, at the object detector, a target image; dividing, at the object detector, the target image into a plurality of divisions, wherein each division includes a plurality of pixels; selecting, at the object detector, a target pixel from the plurality of pixels; and determining, at the object detector using the classifier, a target class for the target pixel of the target image by comparing the target pixel to neighboring pixels from the plurality of pixels.

[0008] According to some embodiments, the method can further include generating a segmented image based on the target image and the determined target class for the target pixel.

[0009] According to some embodiments, the segmented image can be generated using one of a graph-cut, GrabCut, conditional random fields, or manual technique.

[0010] According to some embodiments, each of the training images and the target image can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0011] According to some embodiments, the classifier can be created using a convolutional neural network.

[0012] According to some embodiments, the neighboring pixels can be immediately adjacent to the target pixel.

[0013] According to some embodiments, each of the class and the target class can be at least one of a building, roof, pool, road, trampoline, automobile, truck, boat, plane, communication tower, power transmission tower, hedge, porch, patio, deck, or a tree trunk.

[0014] A method of generating a label image based on a parameter set according to one embodiment of the present disclosure can include receiving, at an object detector, training images; receiving, at the object detector, training parameter sets, wherein each of the

training parameter sets corresponds to a transformed representation of one or more labels, and wherein the one or more labels correspond to a different training image; creating, at the object detector, a classifier configured to determine a parameter set for an image based on the training images and the training parameter sets; receiving, at the object detector, a target image; determining, at the object detector using the classifier, a target parameter set that corresponds to one or more target labels for the target image, wherein the target parameter set corresponds to a target transformed representation of the one or more target labels; and generating, at the object detector, a label image by applying an inverse target parameter set to the target image, wherein the inverse target parameter set corresponds to an inverse transformation of a transformation represented by the target parameter set, wherein one or more pixels of the label image are associated with a class.

[0015] According to some embodiments, each of the one or more classes can be either two-class or multi-class.

[0016] According to some embodiments, the method can further include receiving second training parameter sets for the training images, wherein the second training parameter sets comprise at least one of time, date, sun direction, sun position, latitude, or longitude.

[0017] According to some embodiments, each of the training parameters sets and the target parameter set can represent a transformation created using one of discrete cosine transform, wavelets, discrete Fourier transform, principal component analysis, non-negative Matrix Factorization, or Hadamard transform.

[0018] According to some embodiments, each of the training images and the target image can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0019] A method of determining heights of one or more regions in a target image according to one embodiment of the present disclosure can include receiving, at an extractor, training images, wherein each of the training images includes one or more regions, wherein each region includes one or more pixels; receiving, at the extractor, labels that indicate a height for each pixel of the one or more regions of each training image; creating, at the extractor, a regression model configured to determine a height of a region based on the training images and the labels; receiving, at the extractor, a target image comprising a target region; and determining, at the extractor using the regression model, a height of the target region in the target image.

[0020] According to some embodiments, each of the training images and the target image can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0021] According to some embodiments, receiving the training images can include receiving the training images from one or more sensors at fixed positions.

[0022] According to some embodiments, receiving the training images can further include receiving orientations of the sensors relative to one another if the training images are from two or more sensors.

[0023] According to some embodiments, receiving the training images can include receiving the training images from one or more sensors and parameters related to at least one of one or more positions of the one or more sensors, orientation of illumination source that associates formation of the training images, time, date, latitude, or longitude.

[0024] According to some embodiments, the training images can be projected onto a same geographic surface that defines the one or more regions.

[0025] According to some embodiments, the method can further include receiving, at the extractor, parameters related to positions of one or more sensors configured to capture the training images.

[0026] A method of determining a geometric object property in a target image according to one embodiment of the present disclosure can include receiving, at an extractor, training images; receiving, at the extractor, a geometric object corresponding to a portion of one or more of the training images; receiving, at the extractor, training geometric object properties, wherein each of the training geometric object properties identifies a corresponding geometric object; receiving, at the extractor, first one or more parameters related to orientation of illumination source that associates formation of the training images for the training images; creating, at the extractor, at least one of a classifier or a regression model configured to determine a geometric object property for an image based on the training images, the geometric object, the training geometric object properties, and the first one or more parameters; receiving, at the extractor, a target image; receiving, at the extractor, a target geometric object corresponding to a portion of the target image; receiving, at the extractor, second one or more parameters related to orientation of illumination source that associates formation of the target image; and determining, at the extractor using the at least one of a classifier or a regression model, a target geometric object property associated with the target geometric object.

[0027] According to some embodiments, each of the training images and the target image can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0028] According to some embodiments, the training geometric object property can be at least one of slope, pitch, dominant pitch, material, area, height, or volume, and the target geometric object property can be at least one of slope, pitch, dominant pitch, material, area, height, or volume.

[0029] According to some embodiments, the geometric object can be at least one of a point, a contour, an area, or a binary mask, and the target geometric object can be at least one of a point, a contour, an area, or a binary mask.

[0030] According to some embodiments, the first one or more parameters can include at least one of time, date, sun direction, sun position, latitude, longitude, or object material, and the second one or more parameters can include at least one of time, date, sun direction, sun position, latitude, longitude, or object material.

[0031] A method of aligning images according to one embodiment of the present disclosure can include receiving, at an aligning device, N first type of image sets, wherein N>1, wherein each of the N first type of image sets includes one or more first type of images; receiving, at the aligning device, N second type of image sets, wherein each of the N second type of image sets includes different one or more second type of images, wherein each of the N second type of image sets is aligned with a different one of the N first type of image sets; generating, at the aligning device, M transform parameters, wherein M>1; generating, at the aligning device, M transformed second type of image sets for each of the N first type of image sets so that there are N * M total transformed second type of image sets, wherein each of the M transformed second type of image sets for each of the N first type of image sets is generated by transforming a respective one of the N second type of images sets using a different one of the M transform parameters; creating, at the aligning device, a regressor configured to identify parameters of a transformation that maps a second type of image set to a first type of image set, wherein the regressor is created based on the N first type of image sets, the M transform parameters, and the N * M total transformed second type of image sets; receiving, at the aligning device, a target first type of image set and a target second type of image set; generating, at the aligning device using the regressor, a target transform parameter based on the target first type of image set and the target second type of image set; and generating, at the aligning device, a transformed target second type of image set by

transforming the target second type of image set using the target transform parameter so that the transformed target second type of image set is aligned with the target first type of image set.

[0032] According to some embodiments, the first type of images can be red-green-blue images, and the second type of images can be parcel maps.

[0033] According to some embodiments, each image in the first type of image sets, the second type of image sets, the target first type of image set, and the target second type of image set can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0034] According to some embodiments, the method can further include determining, at the aligning device, whether the target transform parameter results in convergence.

[0035] According to some embodiments, the method can further include the following: if it is determined that the target transform parameter does not result in convergence, performing the following steps until convergence: (1) generating, at the aligning device, a new target transform parameter; and (2) determining, at the aligning device, whether the new target transform parameter results in convergence.

[0036] According to some embodiments, one or more of the N first type of image sets can include a plurality of co-registered first type of images.

[0037] According to some embodiments, the transform parameters and the target transform parameter can be for at least one of translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, or high-order bivariate polynomials.

[0038] A method of aligning images according to one embodiment of the present disclosure can include receiving, at an aligning device, N first type of image sets, wherein N>1, wherein each of the N first type of image sets includes one or more first type of images; receiving, at the aligning device, N second type of image sets, wherein each of the N second type of image sets includes different one or more second type of images, wherein each of the N second type of image sets is aligned with a different one of the N first type of image sets; creating, at the aligning device, a regressor configured to identify a second type of image set given a first type of image set, wherein the regressor is created based on the N first type of image sets and the N second type of image sets; receiving, at the aligning device, a target first type of image set and a plurality of first key points of the target first type of image set; receiving, at the aligning device, a target second type of image set; dividing, at the aligning device, the target first type of image set into a first plurality of divisions, wherein each of the

first plurality of divisions is centered around a different one of the plurality of first key points; generating, at the aligning device using the regressor, a second plurality of divisions of the target second type of image set based on the first plurality of division of the target first type of image set; identifying, at the aligning device, a plurality of second key points, wherein each second key point is from a different division of the second plurality of division; identifying, at the aligning device, transform parameters that maps the plurality of second key points to the plurality of the first key points; and generating, at the aligning device, a transformed target second type of image set by transforming the target second type of image set using the transform parameters so that the transformed target second type of image set is aligned with the target first type of image set.

[0039] According to some embodiments, each image in the first type of image sets, the second type of image sets, the target first type of image set, and the target second type of image set can be one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

[0040] According to some embodiments, the transform parameters can be for at least one of translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, or high-order bivariate polynomials.

[0041] According to some embodiments, one or more of the N first type of image sets can include a plurality of co-registered first type of images.

[0042] The present disclosure also discloses systems that are operative to perform the method steps described above. Disclosed systems can include but not limited to a processor, memory, storage medium (e.g., local storage medium, remote storage medium), communication network, server (e.g., imagery analysis server), device (e.g., imagery acquisition device, client device), module (e.g., analytical module), web service, and pipeline.

[0043] The present disclosure also discloses computer readable media that include executable instructions (e.g., computer program of instructions) operable to cause a system to perform the method steps described above.

[0044] There has thus been outlined, rather broadly, the features of the disclosed subject matter in order that the detailed description thereof that follows may be better understood, and in order that the present contribution to the art may be better appreciated. There are, of course, additional features of the disclosed subject matter that will be described hereinafter and which will form the subject matter of the claims appended hereto.

[0045] In this respect, before explaining at least one embodiment of the disclosed subject matter in detail, it is to be understood that the disclosed subject matter is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The disclosed subject matter is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

[0046] As such, those skilled in the art will appreciate that the conception, upon which this disclosure is based, may readily be utilized as a basis for the designing of other structures, methods and systems for carrying out the several purposes of the disclosed subject matter. It is important, therefore, that the claims be regarded as including such equivalent constructions insofar as they do not depart from the spirit and scope of the disclosed subject matter.

[0047] These together with the other objects of the disclosed subject matter, along with the various features of novelty which characterize the disclosed subject matter, are pointed out with particularity in the claims annexed to and forming a part of this disclosure. For a better understanding of the disclosed subject matter, its operating advantages and the specific objects attained by its uses, reference should be made to the accompanying drawings and descriptive matter in which there are illustrated preferred embodiments of the disclosed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0048] Various objects, features, and advantages of the disclosed subject matter can be more fully appreciated with reference to the following detailed description of the disclosed subject matter when considered in connection with the following drawings, in which like reference numerals identify like elements.

[0049] FIG. 1A illustrates an image of a residential neighborhood including a pool.

[0050] FIG. 1B illustrates a heat map representing the likelihood of the presence of a pool or some other structure.

[0051] FIG. 2 illustrates a method for producing a heat map for identifying roofs in an image.

[0052] FIG. 3 illustrates method for automatically identifying a class of a pixel according to some embodiments of the present disclosure.

[0053] FIG. 4 illustrates an example of automatically identifying a class of a pixel according to some embodiments of the present disclosure.

[0054] FIG. 5A illustrates an example of detecting objects using manually identifying labels.

[0055] FIG. 5B illustrates an example of detecting objects according to some embodiments of the present disclosure.

[0056] FIG. 6 illustrates an example of generating labels for a subset of divisions for an image according to some embodiments of the present disclosure.

[0057] FIG. 7 illustrates an application of segmentation encoding for efficient pixel classification according to some embodiments of the present disclosure.

[0058] FIG. 8 illustrates examples of input and output for a classifier according to some embodiments of the present disclosure.

[0059] FIG. 9 illustrates an example portion of a label space in multi-label classification according to some embodiments of the present disclosure.

[0060] FIGS. 10A, 10B, and 10C respectively illustrate an example set of a DSM image, a DEM image, and an AGH image according to some embodiments of the present disclosure.

[0061] FIGS. 11A, 11B, and 11C respectively illustrate an example set of a DSM image, a DEM image, and an AGH image according to some embodiments of the present disclosure.

[0062] FIG. 12 illustrates a method of determining heights of one or more regions in an image according to some embodiments of the present disclosure.

[0063] FIG. 13 illustrates example images in a feature space and a label space according to some embodiments of the present disclosure.

[0064] FIG. 14 illustrates a method for determining a geometric object property according to some embodiments of the present disclosure.

[0065] FIG. 15 illustrates a method for training a classifier model (e.g., a CNN) for learning the parameters of a transformation that maps the space of geospatial imagery to the space of parcel boundaries according to some embodiments of the present disclosure.

[0066] FIG. 16 illustrates a method for identifying parameters of a transformation that maps a first type of image set to a second type of image set according to some embodiments of the present disclosure.

[0067] FIG. 17 illustrates a training example that uses sample input, transform parameters, and pairs of feature space and transform parameters according to some embodiments of the present disclosure.

[0068] FIG. 18 illustrates a method of aligning images based on identifying a second type of image set given a first type of image set according to some embodiments of the present disclosures.

[0069] FIGS. 19A and 19B illustrate an example of an RGB image and corresponding images of different multi-spectral bands.

[0070] FIG. 20 illustrates examples involving the use of images representing the expectation taken over the predicted distribution of local binary super pixels according to some embodiments of the present disclosures.

[0071] FIG. 21 illustrates a method for generating a label image based on a parameter set according to some embodiments of the present disclosure.

[0072] FIG. 22 illustrates an example input and output of the classifier according to some embodiments of the present disclosure.

[0073] FIG. 23 illustrates an example input to a pseudo multi-label classifier according to some embodiments of the present disclosure.

[0074] FIG. 24 illustrates an example parameter set and a label map formed from the parameter set according to some embodiments of the present disclosure.

[0075] FIG. 25 illustrates an example RGB image and an example AGHM image according to some embodiments of the present disclosure.

[0076] FIG. 26 illustrates a system for analyzing remote sensing imagery according to some embodiments of the present disclosure.

[0077] FIG. 27 illustrates a block diagram of an imagery analysis server according to some embodiments of the present disclosure.

[0078] FIG. 28 illustrates a top-level software architecture of a system for analyzing remote sensing imagery according to some embodiments of the present disclosure.

[0079] FIG. 29 illustrates a process for the whole model setting, where an input raster image is fed to both a heat map generation module and a geometry extraction module according to some embodiments of the present disclosure.

[0080] FIG. 30 illustrates a process for the parts-based model setting, where an input raster image is fed to both a heat map generation module and a geometry extraction module according to some embodiments of the present disclosure.

[0081] FIG. 31 illustrates an example of a library of template shapes according to some embodiments of the present disclosure.

[0082] FIGS. 32A and 32B illustrate an application of autoencoders to multiple machine learning tasks according to some embodiments of the present disclosure.

[0083] FIGS. 33A-33D illustrate example parameter values related to a sensor according to some embodiments of the present disclosure.

[0084] FIG. 34 illustrates an example of determining a geometric object property according to some embodiments of the present disclosure.

[0085] FIG. 35 illustrates an example of using an object detector according to some embodiments of the present disclosure.

[0086] FIG. 36 illustrates an example of ground truth data generated from an RGB input according to some embodiments of the present disclosure.

[0087] FIG. 37 illustrates a binary output generated from using a hybrid segmentation method according to some embodiments of the present disclosure.

[0088] FIG. 38 illustrates a binary output generated from using a pseudo multi-label classification method according to some embodiments of the present disclosure.

[0089] FIG. 39 illustrates a binary output generated from using a fully convolutional network according to some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0090] In the following description, numerous specific details are set forth regarding the systems and methods of the disclosed subject matter and the environment in which such systems and methods may operate, etc., in order to provide a thorough understanding of the disclosed subject matter. It will be apparent to one skilled in the art, however, that the disclosed subject matter may be practiced without such specific details, and that certain features, which are well known in the art, are not described in detail in order to avoid complication of the disclosed subject matter. In addition, it will be understood that the examples provided below are exemplary, and that it is contemplated that there are other systems and methods that are within the scope of the disclosed subject matter.

[0091] Disclosed systems and methods relate to the field of remote sensing, deep learning, and object detection.

Remote Sensing

[0092] Remote sensing is a technology field that includes the remote acquisition of sensory information. This can include, for example, satellite imagery, aerial imagery, radar, sonar, Light Detection and Ranging (LIDAR), seismography, and/or any other suitable mode or combination of modes of sensory information gathering that does not require direct physical contact.

Deep Learning

[0093] Deep learning is a type of machine learning where a multi-level neural network is developed and based on progressively higher levels of abstraction. In a deep learning framework, features are typically learned from raw input data. This is in contrast to a shallow learning framework, which can be viewed as having one input-output layer where handcrafted features (*e.g.*, Scale Invariant Feature Transform (SIFT)) are used for a specific task.

[0094] A deep learning framework can be decomposed into generative and discriminative architectures. Generative architectures attempt to characterize high-order correlation properties of input data (*e.g.*, autoencoders) for pattern analysis or synthesis. These approaches estimate joint distributions of input data and associated classes (*e.g.*, as in Bayes nets). In discriminative architectures, networks are designed to provide discriminative power to classification by directly estimating posterior distributions of class conditioned on the input data (*e.g.*, as in convolutional neural networks). The architecture of convolutional neural networks (CNN) can be viewed as a cascade of modules consisting of a convolutional layer (which share many weights spatially) and a pooling layer (which subsamples the output of the convolutional layer).

Object Detection

[0095] Object detection is a process of detecting regions in images that have semantic meaning (*e.g.*, locating pools, gabled roofs, roads, trees in aerial imagery). Object detection methods can combine both a bottom-up and a top-down approach.

[0096] An example of bottom-up object detection is image segmentation. Image segmentation is a process where pixels in an image are labeled in such a way that pixels sharing the same characteristics (*e.g.*, color, texture, proximity) have the same label. In this way, an image is parceled into distinct segments, typically in a bottom-up approach, where low-level local image properties are used to detect coherent regions. Image segmentation is often difficult in situations where noise, occlusions, shadows, etc. exist. Methods that can be

used to overcome these situations include thresholding, clustering, edge detection, region growing, and graph partitioning.

[0097] Top-down approaches for object detection often involve a training stage to obtain class-specific model features (*e.g.*, parameters for a classifier) that are used to generate a hypothesis for the existence of objects.

[0098] Embodiments of the present disclosure utilize a combined top-down and bottom-up approach.

Remote Sensing Imagery System

[0099] FIG. 26 illustrates a system 2600 for analyzing remote sensing imagery in accordance with embodiments of the present disclosure. The system 2600 can include an imagery analysis server 2604, at least one client device 2606 (*e.g.*, client devices 2606-1, 2606-2, … 2606-N), an imagery acquisition device 2608, a local storage medium 2610, and a remote storage medium 2612. All components in the system 2600 can be coupled directly or indirectly to a communication network 2602. The components described in the system 2600 can be further broken down into more than one component and/or combined together in any suitable arrangement. Further, one or more components can be rearranged, changed, added, and/or removed. For example, in some embodiments, the system 2600 can obtain the data from third-party vendors. In other embodiments, the system 2600 can directly acquire data through the imagery acquisition device 2608.

[00100] Each client device 2606 can communicate with the imagery analysis server 2604 to send data to, and receive data from, the imagery analysis server 2604 via the communication network 2602. Each client device 2606 can be directly or indirectly coupled to the imagery analysis server 2604. Additionally, each client device 2606 can be connected to the imagery analysis server 2604 via any other suitable device(s), communication network, or combination thereof. A client device 2606 can include, for example, a desktop computer, a mobile computer, a tablet computer, a cellular device, a smartphone, a television, or any computing system that is capable of performing computation.

[00101] The imagery analysis server 2604 can receive imagery data from the imagery acquisition device 2608. The imagery analysis server 2604 can extract, analyze, and/or label structural and/or geospatial information of the received imagery data based on the techniques disclosed in this present disclosure. In some embodiments, a classifier can be trained and/or maintained in the imagery analysis server 2604. The imagery analysis server 2604 is shown as a single server; however, the imagery analysis server 2604 can include more than one

server. For example, in some embodiments, the imagery analysis server 2604 can include multiple modular and scalable servers and/or other suitable computing resources. The imagery analysis server 2604 can support elastic computing, which can dynamically adapt computing resources based on demand. The imagery analysis server 2604 can be deployed locally and/or remotely in a third-party cloud-based computing environment. In some embodiments, within the imagery analysis server 2604 or any other suitable component of system 2600, a device or a tool— including a classifier, a regressor, an object detector, an extractor, or an aligning device that are described in the present disclosure—can be implemented as software and/or hardware.

[00102] The imagery acquisition device 2608 can provide the imagery analysis server 2604 with imagery data. In some embodiments, the imagery acquisition device 2608 can acquire satellite imagery, aerial imagery, radar, sonar, LIDAR, seismography, or any other suitable mode or combination of modes of sensory information. In some embodiments, the system 2600 does not include the imagery acquisition device 2608 and can obtain imagery data from third-party vendors.

[00103] The system 2600 includes two storage media: the local storage medium 2610 and the remote storage medium 2612. The local storage medium 2610 can be located in the same physical location as the imagery analysis server 2604, and the remote storage medium 2612 can be located at a remote location or any other suitable location or combination of locations. In some embodiments, the system 2600 can include more than one local storage medium, more than one remote storage medium, and/or any suitable combination thereof. In some embodiments, the system 2600 may only include the local storage medium 2610 or only include the remote storage medium 2612.

[00104] The system 2600 can also include one or more relational databases, which include scalable read replicas to support dynamic usage. The one or more relational databases can be located in the local storage medium 2610, the remote storage medium 2612, the imagery analysis server 2604, and/or any other suitable components, combinations of components, or locations of the system 2600.

[00105] The communication network 2602 can include the Internet, a cellular network, a telephone network, a computer network, a packet switching network, a line switching network, a local area network (LAN), a wide area network (WAN), a global area network, or any number of private networks currently referred to as an Intranet, and/or any other network or combination of networks that can accommodate data communication. Such networks may

be implemented with any number of hardware and software components, transmission media and/or network protocols. In some embodiments, the communication network 2602 can be an encrypted network. While the system 2600 shows the communication network 2602 as a single network, the communication network 2602 can also include multiple interconnected networks described above.

[00106] FIG. 27 illustrates a block diagram of the imagery analysis server 2604 in accordance with some embodiments of the present disclosure. The imagery analysis server 2604 includes a processor 2702, a memory 2704, and an analytical module 2706. The imagery analysis server 2604 may include additional modules, fewer modules, or any other suitable combination of modules that perform any suitable operation or combination of operations.

[00107] The processor 2702 is configured to implement the functionality described herein using computer executable instructions stored in temporary and/or permanent non-transitory memory. The processor can be a general purpose processor and/or can also be implemented using an application specific integrated circuit (ASIC), programmable logic array (PLA), field programmable gate array (FPGA), and/or any other integrated circuit.

[00108] The processor 2702 can execute an operating system that can be any suitable operating system (OS), including a typical operating system such as Windows, Windows XP, Windows 7, Windows 8, Windows Mobile, Windows Phone, Windows RT, Mac OS X, Linux, VXWorks, Android, Blackberry OS, iOS, Symbian, or other OS.

[00109] The analytical module 2706 can be configured to cause the processor 2702 or the imagery analysis server 2604 to extract, analyze, and/or label structural and/or geospatial information of any imagery data based on the techniques disclosed in the present disclosure. In some embodiments, the analytical module can implement and/or execute to detect objects, extract objects, align images, and/or any other suitable task or combination of tasks described in the present disclosure. In some embodiments, within the analytical module 2706 or any other suitable component of the imagery analysis server 2604, a device or a tool—including a classifier, a regressor, an object detector, an extractor, or an aligning device that are described in the present disclosure—can be implemented as software and/or hardware.

[00110] In some embodiments, the analytical module 2706 can be implemented in software using the memory 2704. The memory 2704 can be a non-transitory computer readable medium, flash memory, a magnetic disk drive, an optical drive, a programmable

read-only memory (PROM), a read-only memory (ROM), or any other memory or combination of memories.

[00111] FIG. 27 shows the imagery analysis server 2604 having the analytical module 2706 that causes the processor 2702 or the imagery analysis server 2604 to perform the operations in accordance with certain embodiments of the disclosed subject matter. The processor 2702 and/or the imagery analysis server 2604 may include additional modules, less modules, or any other suitable combination of modules that perform any suitable operation or combination of operations.

[00112] FIG. 28 illustrates a top-level software architecture of a system 2800 for analyzing remote sensing imagery in accordance with some embodiments of the present disclosure. System 2800 can include one or more of the following four components: web services 2802, a Mechanical Turk (M-Turk) pipeline 2804, a training pipeline 2806, or a prediction pipeline 2808.

[00113] Web services 2802 can include one or more of the following components: front end 2810, flow control back end 2812, attribute uService 2814, attribute DB 2816, parcel service 2818, tile map service 2820, parcel DB 2822, or image DB 2824. Front end 2810 can provide services to a user. For example, the user can use a browser on a workstation to connect to, and communicate with front end 2810 to execute certain web services functions, such as detecting and extracting objects from remote images. Flow control back end 2812 can communicate with front end 2810. Front control back end 2812 can implement a control flow logic and support a public Representational State Transfer (RESTful) API to client applications and web-based front-end user interface. RESTful APIs can support lightweight, maintainable, and scalable architectures. In some embodiments, flow control back end 2812 can receive and control multiple requests from front end 2810. Flow control back end 2812 can update the status of requests to front end 2810. In some embodiments, flow control back end 2812 can receive data, such as parcel images, from parcel service 2818, which can retrieve these parcel images from parcel DB 2822. Flow control back end 2812 can also receive data, such as images, from tile map service 2820, which can retrieve these images from image DB 2824. Flow control back end 2812 can communicate with attribute uService 2814, which can also receive images from tile map service 2820. In some embodiments, attribute uService 2814 can provide information about detected objects within a context. For example, Attribute uService 2814 can provide information about detected roofs within RGB imagery provided by tile map service 2820. In

some embodiments, the information can be related to attributes that can include, for example, whether the detected roof has solar panels, the roof footprint area, etc. In some embodiments, there can be multiple Attribute uServices, each of which can handle a different attribute type. Attribute uService 2814 can also receive attributes from attribute DB 2816. In some embodiments, these attributes can be related to objects in an image. For example, a slope of a roof can be an attribute of the roof. In some embodiments, web services 2802, using attribute uService 2814, can provide information on an attribute of an object in an image based on the image and a related parcel image.

[00114] M-Turk pipeline 2804 can support ground truth acquisition by, for example, correctly labeling structures. M-Turk pipeline 2804 can include one or more of the following components: M-Turk labeling 2826, ground truth Candidate DB 2830, M-Turk validation 2832, or ground truth DB 2828. M-Turk labeling 2826 can label images from tile map service 2820 and store the labeled images in ground truth candidate DB 2830. M-Turk validation 2832 can determine whether the ground truth candidate(s) in ground truth candidate DB 2830 is valid. In some embodiments, a ground truth candidate can be valid if its accuracy level can be determined to exceed a threshold value. If the ground truth candidate(s) is valid, it can be stored in ground truth DB 2828.

[00115] Training pipeline 2806 can include one or more of the following components: training 2834 and classifier 2836. Training 2834 can receive training images from tile map service 2820 for training. Training 2834 can also receive ground truth data from ground truth DB 2828. Based on the received training images and the received ground truth data, training 2834 can train a classifier 2836. Non-limiting examples of specific instances of training 2834 and classifier 2836 are described in detail below.

[00116] Prediction pipeline 2808 can include a prediction 2838 component. Prediction 2838 can use classifier 2836 to predict or classify attributes in an image from tile map service 2820. In some embodiments, the output of classifier 2836 can provide data to prediction 2838. In some embodiments, prediction 2838 can be used to test classifier 2836 to determine whether classifier has been properly tested. In other embodiments, prediction 2838 can be used during the production stage. Prediction 2838 can provide data, including results, from its operation to attribute DB 2816 for storage. In some embodiments, prediction 2838 can optionally provide its data from operation to ground truth candidate DB 2830 for storage. M-Turk validation 2832 can determine whether the data provided from prediction 2838 is valid.

[00117] In some embodiments, system 2800 can include both on-line components and/or off-line components. The on-line components can run in real-time and can include one or more of the following components: (1) front-end 2810 (*e.g.*, web-based user interface); (2) flow control back end 2812; (3) tile map service 2820; (4) parcel service 2818; and (5) attribute uService 2814. In some embodiments, parcel service 2818 and/or attribute uService 2814 can be internal modules that can communicate via JavaScript Object Notation Remote Procedure Call (JSON-RPC) protocol. In some embodiments, internal modules are not exposed to top layer applications. For example, when parcel service 2818 and attribute uService 2814 are internal modules, parcel service 2818 and attribute uService 2814 can logically reside behind flow control back end 2812 such that they are not directly visible to front end 2810.

[00118] The off-line components can run as separate executables and populate the systems databases, and can include one or more of the following components: (1) M-Turk Pipeline 2804; and (2) machine learning/computer vision (*e.g.*, training pipeline 2806 and prediction pipeline 2808). In some embodiments, training pipeline 2806 and prediction pipeline 2808 are implemented in modules that have their own processing pipelines. In some embodiments, the machine learning/computer vision components can also run on-line. In some embodiments, the on-line components can be implemented as off-line components, the off-line components can be implemented as on-line components, and/or any other suitable combination thereof.

[00119] System 2800 may include additional components, less components, or any other suitable combination of components that perform any suitable operation or combination of operations. In some embodiments, these components can reside in a single machine, such as a server. In other embodiments, these components can be distributed across multiple machines.

[00120] The hardware, system, and architecture described in connection with FIGS. 26-28 can be deployed, entirely or partially, in a cloud-based environment. Further, the present disclosure is not limited to the exemplary hardware and/or software arrangements described in connection with FIGS 26-28; other suitable hardware and/or software implementations are also within the spirit of the present disclosure.

Machine Learning for Object Detection

[00121] In computer vision, an image can be divided into segments, each of which can represent one or more objects. Each segment can include one or more pixels. Each object type can be associated with a class. A label can be used to associate a class to each pixel. If there are two or more object types from which the class can be associated, the class is considered multi-class. For example, a red-green-blue (RGB) image taken above the ground can include objects, such as, for example, roads, roofs, gardens, and pools. The RGB image can be segmented such that each pixel of the image can correspond to, or labeled as one of the identified objects. A pixel that is part of a road can be labeled as the road class, while another pixel that is part of a roof can be labeled as the roof class.

[00122] In some cases, only a specific type of an object may be of interest and thus, needs to be identified. For example, one may be interested in identifying only roofs in an image. In such cases, each pixel of an image can be labeled in a binary way (*e.g.*, represented as a binary "1" or "0"). For example, each pixel can be labeled as either part of a roof (*e.g.*, represented as binary "1") or not part of a roof (*e.g.*, represented as binary "0"). In these cases, the class is considered two-class – that is, each class corresponds to either being that object or not being that object.

[00123] In some cases, a pixel can be labeled to indicate a probability of being part of a specific object. In some embodiments, the probability can be a real-valued scalar on the interval [0,1]. In some embodiments, the probability can be expressed as a percentage. For example, each pixel of an image can be labeled with the probability of being part of a roof, such as 0%, 20%, 50%, 93%, or 100% of being part of a roof, or any other suitable percentage.

Model Building

[00124] In some embodiments, spatial resolution is measured by a sensor's capability to resolve the smallest objects with distinct boundaries. In remote sensing images, the pixel-center to pixel-center distance is typically called ground sample distance (GSD). For example, a GSD can be10 cm. In this example, sub-pixel accuracy can imply error of less than 10 cm; pixel accuracy can imply error of about 10 cm, and super-pixel accuracy can imply error of more than 10 cm. According to some embodiments, building models can be fitted with remote sensing imagery in a sub-pixel, pixel, and/or super-pixel accurate manner. These models can be either a whole model or a parts-based model. In both instances, 2D and 3D models can be considered. In the 2D model, the height variable can be held fixed. Per

the parts-based model, an object's measures can be taken to be more granular. For example, using the parts-based model, roof summary measures (*e.g.*, dominant pitch, dominant gable configuration, etc.) can be taken to be more granular (*e.g.*, roof is composed of, by area, 25% gable and 75% hip components). In the 2D model setting, the original 3D problem (e.g., measuring a dominant pitch is originally a 3D problem) can become ill-posed. To address this issue, one or more variables can be fixed. For example, the roof height can be fixed, for example, with a small set of common heights.

[00125] FIG. 29 illustrates a process for the whole model setting. At step 2901, an input raster image can be used for heat map generation. At step 2904, the input raster image can be used for geometry extraction, which can assume several embodiments including, but not limited to, edge detection and corner detection. At step 2902, the resulting heat map from step 2901 can be used to produce semantically relevant objects by a process that can assume several embodiments including, but not limited to, thresholding, clustering, active contour segmentation, and/or any of the object detection methods discussed in this disclosure (*e.g.*, steps or operations described in connection with methods 300 and 2100). In some embodiments, candidate regions can be generated to identify semantically relevant objects. At step 2903, whole models can be fitted to the regions. In some embodiments, polygons are fitted. At step 2905, these models can be further refined using the extracted geometry from 2904 by using, for example, the estimation of a joint probability of heat map and edges.

[00126] FIG. 30 illustrates a process for the parts-based model setting. At step 3001, an input raster image can be used for heat map generation. At step 3004, the input raster image can be used for geometry extraction, which can assume several embodiments including, but not limited to, edge detection and corner detection. At step 3002, the resulting heat map from step 3001 can be used to produce semantically relevant objects by a process that can assume several embodiments including, but not limited to, thresholding, clustering, and/or active contour segmentation. In some embodiments, candidate regions can be generated to identify semantically relevant objects. At step 3003, parts-based models can be fitted to the regions. In some embodiments, polygons are fitted. At step 3005, these models can be further refined using the extracted geometry from 3004 by using, for example, the estimation of a joint probability of heat map and edges. The distinction between this approach and the whole model approach as shown in FIG. 29 is that extraction geometry 3004 is also used to inform model selection 3003. As an example, in the parts-based model, individual facets of a roof can be modeled.

Hybrid Heat Maps

[00127] A heat map can be used to graphically represent an image to show the mapping between the pixels of the image and the class conditional probability that corresponds to each of these pixels. In some embodiments, a segmentation can be derived from the class conditional probability. To distinguish one class from another, different color schemes can be used. Non-limiting examples of color schemes include multi-color, grayscale, and black and white.

[00128] In a multi-class label, each color can correspond to an object. For example, white can be used to identify a house, blue can be used to identify a pool, and green can be used to identify a lawn in an image. In some embodiments, any other suitable color or combination of colors can be used to identify an object. Examples of using a heat map for object detection include semantic segmentation of large images into different types of objects, such as pools, buildings, trees, and property parcels.

[00129] In a two-class label, each color can correspond to whether a specific pixel corresponds to the specified object. Any suitable color scheme can be used. For example, in identifying a roof in an image, if a black and white scheme is used, white can be used to identify a roof, and black can be used to identify any other object that is not a roof, or vice versa. As another example, if a grayscale scheme is used, white can be used to identify what is certain or almost certain to be a roof (*e.g.*, based on a predetermined probability or percentage), and black can be used to identify what is certain or almost certain to not be a roof (*e.g.*, based on a predetermined probability or percentage), or vice versa. In addition, shades of gray color can be used to represent different probabilities of an object being a roof. For example, the lighter the gray, the more likely the corresponding pixel is a roof, and the darker the gray, the more likely the corresponding pixel is not a roof, or vice versa.

[00130] The notion of a heat map H, associated with a raster image I, can be a scalar field, where for each pixel location x, a class probability, conditioned on spatially local features over a neighborhood N(x), is defined as $H(x) = p(c(x)|I(N(x))$. For example, the class c may be the structural category or class "pool," and the local features may be the raw image signal values (*e.g.*, a red-green-blue (RGB) triple) in a spatial neighborhood of a given pixel location. In this setting, the heat map can represent the probability that a given pixel is part of a pool. Raw image signal values may be multispectral (multiple discrete narrow bands) and hyperspectral (continuous block of narrow bands).

[00131] FIG. 1A illustrates an image of a residential neighborhood 100a including a pool 102a. FIG. 1B illustrates a corresponding heat map 100b representing the likelihood of the presence of a pool or some other structure. The pool 102a in the residential neighborhood 100a is now represented as a light region 102b in the heat map 100b. Bar 104a shows grayscale colors that correspond to the likelihood of presence of a pool or some other structure. For example, a value of "1" indicates that it is certain that the identified object is a pool, while a value of "0" indicates that it is certain that the identified object is not a pool but another structure. In heat map100b, the light regions, corresponding to a value near "1," indicates a high likelihood of the presence of a pool (and a low likelihood of the presence of some other structure). And the dark regions, corresponding to a value near "0," indicate a low likelihood of the presence of a pool (and a high likelihood of the presence of some other structure).

[00132] Traditionally, an image could be segmented using manual hints, where some of the pixels of the image were manually labeled. A computer vision method could use these manual hints to assign labels to the remaining pixels by local similarity (*e.g.*, color) or distance to the labeled pixels. This could be done using the so-called graph-cut approach. Using this approach, the computer vision method could then produce a heat map that segments the image.

[00133] FIG. 2 illustrates a prior art method for producing a heat map for identifying roofs in an image. Computer vision method 206 takes in image 202 and manual hints 204. Manual hints 204 are provided by a human who draws lines that correspond to what the person sees as roofs and optionally, what the person sees as not roofs (some other structure). Computer vision method 206 takes in these inputs, assigns a label to each pixel of the image, and produces heat map 208. In heat map 208, the white areas represent the roofs in the image while the black areas represent non-roof areas. This conventional method has a number of problems. First, this method is not efficient and not scalable. This requires a human to provide manual hints for each image, which is a time consuming process. Second, this method may not be accurate. Because this depends on a human to provide accurate hints, it is vulnerable to human errors. In addition, because the hints cover only a small portion of the image, a lot of estimation by the computer needs to be done, which increases the risk of producing an inaccurate heat map.

[00134] The problems identified in the prior art can be solved by using an object detector in accordance with some embodiments of the present disclosure. For example, a top-

down object detection approach can be used, where heat maps are generated from trained classifiers. In one instance, a heat map(s) can be generated through the use of trained convolutional neural networks (CNNs), where either a sparse or a dense application of the heat map(s) can be applied to an input image.

[00135] Heat maps can be used to generate a semantic pixel labeling of the image space in a number of ways including via the simple threshold of a single heat map or by maximum likelihood selection across multiple heat maps. In some embodiments, the maximum likelihood selection can be taken at the pixel-level. As shown in heat map 100b (FIG. 1B), heat maps may not be fully consistent with object edges so a refinement step is often required to produce a final segmentation. In another instance, computer vision methods are used to produce the pixel labeling by using the CNN generated heat maps as hints or weights. Examples of these computer vision methods include GrabCut and conditional random fields (CRFs). In GrabCut image segmentation, an iterative optimization process can be applied to estimating foreground and background. This approach alternates between enforcing intra-region pixel label homogeneity and applying graph-cut optimization (max-flow min-cut) to infer pixel labels with color distribution of foreground objects and background modeled using a Gaussian mixture model. In the case of CRFs, a framework for constructing probabilistic models for image segmentation can be used, where the weights of different features at different states can be balanced against one another through a single joint exponential model for the joint probability.

[00136] FIG. 3 illustrates a method 300 for automatically identifying a class of a pixel according to some embodiments of the present disclosure. In some embodiments, method 300 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 300 can be performed at an object detector. In some embodiments, the object detector can be located in the imagery analysis server 2604 or any other suitable location of the system 2600.

[00137] At step 302, training images can be received. The training images can include one or more pixels. In some embodiments, each of the training images includes more than one pixel. In some embodiments, each of the training images is one of red-green-blue (RGB), panchromatic, infrared (IF), ultraviolet (UV), multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, these training images can be the same image type. For example, these training images are all of one of the following image types: RGB, IR, and UV. In other embodiments, these training images can be a different suitable

image type or combination of image types. For example, of the training images, 30% can be RGB, 25% can be IR, and 45% can be UV.

[00138] At step 304, labels for the training images can be received. In some embodiments, a label can associate a class to a pixel of one of the training images. For example, a training image can include a roof and a pool. A label can associate one of the pixels making up the roof to the roof class, while another label can be used to associate one of the pixels making up the pool to the pool class. As another example, a label can associate a pixel to one of the two classes—foreground and background. The foreground class can refer to a pixel that is part of a specified object (*e.g.*, object of interest), while the background class can refer to a pixel that is not part of the specified object. A label can also associate a pixel to one of many classes, each of which relates to the same object. For example, a pixel can be one of "definitely foreground," "likely foreground," "definitely background" (*e.g.*, not foreground), and "likely background" (*e.g.*, not likely foreground). In some embodiments, any other suitable label or combination of labels can be used. In some embodiments, the class can be one of a building, roof, a pool, a road, a trampoline, an automobile, a truck, a boat, a plane, a communication tower, a power transmission tower, a hedge, a porch, a patio, a deck, a tree trunk, and/or any other suitable object or combination of objects. In some embodiments, a label can associate a class to each of the pixels of the training images.

[00139] In some embodiments, a label can associate a class to multiple pixels of one of the training images. For example, a label can associate all the pixels making up the roof to the roof class. In some embodiments, all the pixels of all the training images can be associated to a class by the received labels.

[00140] At step 306, a classifier can be created based on the received training images and the received labels. The classifier can be configured to identify the class of each pixel of an image. In some embodiments, the classifier can be created using a machine learning system, which includes one or more classifiers such as an artificial neural network (ANN) including but not limited to a convolutional neural network (CNN), as would be appreciated by one of ordinary skill in the art.

[00141] At step 308, a target image can be received. In some embodiments, the target image is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, the target image can be the same image type as the image type of the training images. For example, the image type of the target image and the training images can all be RGB.

[00142] At step 310, the target image can be divided into divisions. Each division can include one or more pixels. For example, a division can consist of one pixel. In another example, a division can consist of 1500 pixels or any other suitable number of pixels. In some embodiments, the target image is divided equally, such that the area size is the same for all the divisions. For example, a 256 pixel-by-256 pixel image, consisting of 65,536 pixels, can be divided into 4,096 divisions, where each division consists of 16 pixels. In some embodiments, the shape of each division can be the same. For example, each of the 4,096 divisions can be 4 pixel-by-4 pixel. In other embodiments, the shapes of different divisions can be different. For example, some of the 4,096 divisions can be 4 pixel-by-4pixel, while some other divisions can be 8 pixel-by-2pixel, or any other suitable size. In some embodiments, the target image is not divided equally, such that area sizes can be different between different divisions. For example, one division can consist of 1 pixel, another division can consist of 16 pixels, and another division can consist of 2500 pixels. In some embodiments, the division mechanism can be configured either automatically or manually. In some embodiments, the target image can be divided into divisions such that one or more divisions overlap. In other embodiments, the target image can be divided into divisions such that none of the divisions overlaps.

[00143] At step 312, a target pixel can be selected. In some embodiments, the target pixel can be from one of the divisions of the target image. In some embodiments, the center or near-center pixel from each of the divisions is selected as a target pixel. For example, in a 3 pixel-by-3 pixel division, the center pixel that is selected as the target pixel is the fifth pixel —counting from left to right, top to bottom (the 2nd pixel from the left and 2nd pixel from the top). In other embodiments, the target pixel is selected from only some of the divisions. For example, if a target image is divided into 1,000 divisions, a target pixel from each of 300 of these 1,000 divisions can be selected. In some embodiments, more than one target pixel can be selected from a division.

[00144] At step 314, a target class for a target pixel or pixels of the target image can be determined using a classifier. In some embodiments, this classifier is the classifier created at step 306. In some embodiments, a CNN can be used as the classifier. In some embodiments, the target class can be determined by comparing the target pixel(s) to neighboring pixels of the target image. For example, the color of a target pixel can be compared to the color of a neighboring pixel. If the two colors are the same or similar, the class of the target pixel can be determined to be the same as the class of the neighboring pixel. In some embodiments,

other suitable attributes can be compared among pixels. In some embodiments, one or more of the neighboring pixels can be immediately adjacent to the target pixel. In some embodiments, the target class can be one of a building, roof, a pool, a road, a trampoline, an automobile, a truck, a boat, a plane, a communication tower, a power transmission tower, a hedge, a porch, a patio, a deck, a tree trunk, and/or any other suitable object or combination of objects.

[00145] At step 316, a segmented image can be generated based on the target image and the target class determined for one or more target pixels. In some embodiments, the segmented image can be a heat map. Various techniques can be used to generate the segmented image. Non-limiting examples of techniques include graph-cut, GrabCut, CRFs, and manual. A graph cut can be used to achieve inference (*e.g.*, label assignment). GrabCut is a graph-based method that can be used to estimate the color distribution of foreground and background objects using a Gaussian Mixed Model (GMM). A GMM can be used to form a CRF over pixel labels, where the CRF is associated with discriminative models. The CRF can use an energy function that includes a pairwise term that favors connected regions having the same label.

[00146] In GrabCut, some or all pixels can be assigned to the foreground and background with weights that can indicate how likely the pixel(s) belong to the assigned class. This construct can be represented with a graph. A graph can include nodes and edges, where the edges can describe a relationship between exactly two nodes. Additionally, an edge can have real-valued weights assigned to it to capture a relationship between the two nodes. For example, a graph can capture the flight connections between airports, where each flight can be represented as an edge from one node (airport) to another node, and the edge's weight can be the distance, price etc. Within the context of GrabCut, the set of nodes can include a node for each pixel and two label nodes (often referred to as the "source" and "sink," each corresponding to one of background and foreground). In this setting, edges can be formed between the pixel-nodes and the label-nodes. Additionally, edges can be formed between pixel-nodes and their neighboring pixel-nodes, where the weights indicate pixel similarity (*e.g.*, closeness in color). Graph-cut can be used in GrabCut. Graph-cut can attempt to separate the sink from the source, so that there is no path from sink to source. Graph-cut can search for the min-cut, which can involve looking for the set of edges to be cut (in order to separate sink from source) with the smallest sum of their weights. A dual problem is max-flow, wherein the edge weights are interpreted as capacities of "pipes," and

maximizing the "flow" from source to sink is desired. If the flow is maximized, the saturated pipes are cut.

[00147] FIG. 4 illustrates an example of using method 300 (FIG. 3) according to some embodiments of the present disclosure. In some embodiments, object detector 404 can be used to produce binary-class heat map 410 to show roof objects. Object detector 404 can implement and execute a classifier that can take image 402 as input. After object detector 404 divides 402 into multiple divisions, object detector 404 can select a center pixel from each of these divisions. Object detector 404 can then determine a class for each of these center pixels and generate label map 406 based on the determined class. Label map 406 can include automatically generated labels. In some embodiments, each white dot, such as 420, can represent the roof class, each black dot, such as 422, can represent the non-roof class, and each gray dot or gray area, such as 424, can respectively represent an unclassified dot or area. In some embodiments, the number or percentage of pixels that are given labels can be customized manually or automatically.

[00148] Computer vision method 408 can take image 402 and label map 406 as input. In some embodiments, computer vision method 408 can be a segmentation mechanism that takes hints or labels. Example segmentation mechanisms include a graph-cut, GrabCut, and conditional random fields. Computer vision method 408 can generate heat map 410 based on image 402 and label map 406. Heat map 410 can show all the roof objects in image 402 by indicating the roof objects as white areas and the non-roof objects as black areas.

[00149] FIGS. 5A and 5B illustrate the comparison of object detection between using method 500A, which is the prior art method of manually identifying labels, and using method 500B, which is in accordance with some embodiments of the present disclosure. Prior art method 500A takes in image 502A as an input. 504A shows hints, which have been manually entered by a human. In FIG. 5A, two types of hints can be provided: foreground and background. Foreground hints 506A and 512A are used to indicate an object type of interest, while background hints 508A are used to indicate any other areas that are not part of the object type of interest. The segmentation method, such as GrabCut, takes image 502A and the hints in 504A to generate a final binary foreground mask 510A. Binary foreground mask 510A shows foreground 514A and 518A in white, and background 516A in black.

[00150] Method 500B shows a method according to some embodiments of the present disclosure. Method 500B can take the same image (image 502A) as prior art method 500A. However, method 500B automatically generates labels by classifying divisions of image

502A by using a classifier. In method 500B, foreground labels 506B and 512B indicate an object type of interest, while background labels 508B indicate any other areas that are not the object type of interest. The segmentation method, such as GrabCut, takes in image 502A and labels 504B to generate a final binary foreground mask 510B. Binary foreground mask 510B shows foreground 514B and 518B in white, and background 516B in black. More pixels can be classified in method 500B than in prior art method 500A because with the prior art method, it is impractical for a human to provide hints for a large number of pixels. Thus, method 500B can result in a more accurate binary foreground mask 510B than binary foreground mask 510A produced by prior art method 500A. Moreover, unlike prior art method 500A, method 500B can be scalable to process a large number of pixels.

[00151] FIG. 6 illustrates an example of generating labels for a subset of divisions for an image according to some embodiments of the present disclosure. In some embodiments, a subset of total divisions, such as division 602 and division 604, can be selected. These selected divisions can be used as inputs to classifier 606, which can generate labels for these divisions. The generated labels for these divisions are shown at 608 and 610, respectively. By generating labels for only a subset of divisions, processing time and storage can be saved. These subsets can be chosen such that they cover objects or areas of interest, such as local windows, part roof, sub areas, or any other suitable object or areas of interest. In some embodiments, only the center pixel of a division can be analyzed. In this case, the division size (*e.g.*, the number of pixels per division) can be selected to provide sufficient context for the classifier to identify the center pixel's class.

Segmentation Encoding for Efficient Classification

[00152] Dimensionality reduction (compression) techniques for representing image segmentation can be used for efficient pixel foreground classification. A non-limiting example of this process includes, given an RGB image, finding a set of discrete cosine transform (DCT) coefficients that best describes an implicit binary foreground segmentation. Applying the inverse DCT transformation to the coefficients produces the classification (*e.g.*, at each pixel, the likelihood of being a foreground object).

[00153] In some embodiments, a classifier can be trained, where inputs are geospatial imagery (*e.g.*, RGB and infrared images) and outputs (label space) are the parameters that characterize a compressed segmentation (*e.g.*, coefficients of an orthogonal transformation).

[00154] FIG. 7 illustrates an application 700 of segmentation encoding for efficient pixel classification in accordance with some embodiments of the present disclosure. In some embodiments, the application 700 can include the following two stages: a training stage 710 and a testing stage 712. The training stage 710 can be used to train a classifier to produce a set of parameters that can be inversely transformed to produce the resulting classification for an image. The testing stage 712 can be used to test the trained classifier from the training stage 710.

[00155] The training stage 710 can include a segmentation step 702 and a forward transformation step 704. At step 702, for a set of input imagery, a binary segmentation can be applied to produce corresponding foreground masks. Examples of segmentation include those based on techniques such as GrabCut and manual. At step 704, a forward transformation can be applied to the binary segmentations. In some embodiments, the most significant parameters of these segmentations can characterize the compressed version of the same. Examples of such compression transformation include DCT, wavelets, discrete Fourier transform, principal component analysis (PCA), non-negative Matrix Factorization (NMF), Hadamard transform, and/or any other suitable transformation. The input imagery associated with the binary segmentations and their associated compression parameters can be used to train a classifier. Examples of the classifier model include ANNs including CNNs.

[00156] After the classifier has been trained, the classifier can be tested at the testing stage 712, which can include an "apply classifier" step 706 and an inverse transformation step 708. At step 706, a test image can be supplied to the trained classifier that can produce a set of parameters (*e.g.*, coefficients of one of the aforementioned orthogonal transform). At step 708, the inverse transformation can be applied to these parameters to produce the resulting classification (*e.g.*, likelihood of foreground).

[00157] FIG. 8 illustrates examples of three RGB input images (802, 804, 806) and three corresponding output images (808, 810, 812) for the trained classifier. These RGB input images can be used for a classifier during a testing or production stage. For example, RGB input images 802, 804, and 806 can be the test images in the "apply classifier" step 706 (Fig. 7). The classifier can produce three sets of parameters based on these three images. At step 708 (Fig. 7), the classifier can apply the inverse transformation to these parameter sets such that the three corresponding output images 808, 810, and 812 are produced.

Pseudo Multi-Label Classification

[00158] In some embodiments, multi-label classification describes situations where multiple target labels are assigned to each input instance. This generalizes the concept of multi-class classification where each instance is restricted to have one class label. A common multi-label classification is semantic scene understanding (*e.g.*, "this image patch contains a roof and pool") or segmentation (*e.g.*, one label per pixel).

[00159] In classification, a distribution over object classes can be received when, for example, using an ANN. Thus, for a given input, a classifier that can distinguish between a set of classes can produce a normalized array of real-valued numbers that reflect the likelihoods of the classes for the given input. In some embodiments, this normalized array can include two real-valued numbers, where one number represents the likelihood of a pixel being a specific object (*e.g.*, a roof) and the other number represents the likelihood of the pixel not being the specific object (*e.g.*, not a roof). When a single class is desired, the class associated with the largest real-valued number can be selected. For example, if, for a pixel, the likelihood of roof object is 0.9 and the likelihood of not roof object is 0.1, the pixel can be classified as belonging to a roof object.

[00160] A classifier can also be configured to produce a non-normalized array of real-valued scalars. In this setting, the output is not considered to be a distribution over classes and, thus, the classifier can be used to solve a regression problem. In the context of training under pseudo multi-label classification, the set of labels are not object categories, such as "roof" and "trampoline," but rather a set of normalized or non-normalized array of floating point values that can be used to reconstruct the final segmentation output. In the case of regression, the non-normalized output parameters can be produced by any kind of invertible transformation that produces a more compact and generalized description of shape. PCA and DCT are two non-limiting examples that can compress and generalize shape (*e.g.*, by considering most likely shapes or lower frequencies). In the case of classification, the classifier can indicate how similar the segmented version of an input image is to a set of classes (*e.g.*, corresponding to discrete patterns). In this case, the classifier output can be interpreted as being the likelihood that the segmentation output looks like the pattern the class is associated with. These patterns can be shapes as "corners" and "rectangles" that are present in various locations within the image. The final output is then generated as the weighted average (or best match, etc.) of these patterns by using their likelihood as the weights. Fig. 9 illustrates examples of patterns used (described below).

[00161] Instead of using multiple labels to train (which is more difficult to train, design networks, and is slower), a single label can be used to describe the class of a shape of an object (or pattern) to be segmented within a small image chip region. In this setting, the output of the classifier can be a probability distribution over the space of classes (patterns). A chip region can be a portion of the image. In some embodiments, a chip region can represent an area of interest. In some embodiments, an image can be processed only at chip regions such that the whole image does not need to be processed. In some embodiments, a heat map (*e.g.*, where each pixel is labeled with the likelihood of being a specific object such as a roof) can be segmented to produce a binary image of two classes (*e.g.*, roof and not roof).

[00162] FIG. 9 illustrates an example portion 900 of a label space in multi-label classification in accordance with some embodiments of the present disclosure. The example portion 900 can include learned patterns 902, 904, 906, 908, 910, 912, 914, 916, 918, 920, 922, and 924. This example illustrates the cluster centers of a k-mean process against the discrete binary super-pixel pattern case. Each of the learned patterns represents a different cluster center. In some embodiments, different cluster centers are generated to produce a representative sampling for training. Although generating different cluster centers is not required for producing a representative sampling for training, it can be useful in creating a more equal distribution over the feature space. In some embodiments, different cluster centers can be generated using principal component analysis. The weighted average (expectation) of these learned patterns can be taken as the final heat map. In some embodiments, the family of patterns can include binary superpixels (either 0 or 1); affinity propagation, mean-shift, db-scan, patterns generated from k-means clustering learned from a training set; patterns generated from PCA over a training set; and low frequency DCT patterns.

[00163] FIG. 20 illustrates examples involving the use of images representing the expectation taken over the predicted distribution of local binary super pixels in 2000. These examples represent non-limiting examples of using pseudo multi-label classification. RGB images 2002, 2006, and 2010 can be inputs to a classifier using pseudo multi-label classification. The classifier can generate the corresponding output heat map images 2004, 2008, and 2012, respectively. In some embodiments, by using pseudo multi-label classification, the output can be reduced to a relatively small number of coefficients (*e.g.*, several hundreds) that can represent image segmentations. This can increase processing speed and save storage space.

[00164] FIG. 21 illustrates a method 2100 of generating a label image based on a parameter set according to some embodiments of the present disclosure. In some embodiments, method 2100 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 2100 can be performed at an object detector. In some embodiments, the object detector can be located in the imagery analysis server 2604 or any other suitable location of the system 2600.

[00165] At step 2102, training images can be received. The training images can include one or more pixels. In some embodiments, these training images can be the same image type. For example, these training images are all of one of the following image types: RGB, IR, and UV. In some embodiments, each of the training images is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, an image can include two or more co-registered channels or bands of electromagnetic spectrum. For example, an image can include RGB and IR.

[00166] At step 2104, training parameter sets can be received. In some embodiments, each training parameter set can correspond to one or more labels that have been transformed using a transformation technique, such as DCT, wavelets, discrete Fourier transform, PCA, NMF, Hadamard transform, and/ or any other suitable transformation. In some embodiments, the one or more labels can correspond to a different training image. In some embodiments, every training image corresponds to a training parameter set that represents labels for the training image.

[00167] At step 2106, a classifier can be created based on the training images and the parameter sets. The classifier can be configured to determine a parameter set for an image. In some embodiments, the classifier can be created using a machine learning system, such as an ANN or a CNN as would be appreciated by one of ordinary skill in the art.

[00168] At step 2108, a target image is received. At step 2110, a target parameter set that corresponds to one or more labels for the target image can be determined using the trained classifier. In some embodiments, this classifier is the classifier created at step 2106. In some embodiments, the target parameter set can correspond to one or more labels that have been transformed using a transformation technique, such as DCT, wavelets, discrete Fourier transform, PCA, NMF, and Hadamard transform. In some embodiments, the target image is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00169] At step 2112, a label image can be generated by applying an inverse transformation to the target parameter set. For example, if the target parameter set generated at step 2110 corresponds to DCT, then an inverse transformation of DCT can be applied to this target parameter set to produce a label image or label map. In some embodiments, one or more pixels of the segmented image can be associated with a class. For example, a particular pixel of the label image can be associated with a roof. In some embodiments, the class can be two-class. In other embodiments, the class can be multi-class. An example of step 2112 is described in connection with Fig. 24 below.

[00170] In some embodiments, method 2100 can also include receiving other training parameter sets for the training images. For example, a training parameter set can include at least one of time, date, sun direction, sun position, latitude, longitude, and/or any other suitable parameter or combination of parameters. In some embodiments, this additional information from the training parameter sets can be related to corresponding training images. For example, the time value can be related to the time when the training image was captured. As another example, the latitude value can be related to the latitude of the location where the training image was captured. In some embodiments, the additional information from the training parameter sets can increase the accuracy of the classifier.

[00171] FIG. 22 illustrates an example input and output of the classifier as described in connection with FIG. 21. Multi-label classifier 2204 is an example of a classifier created from step 2106 in FIG. 21. Multi-label classifier 2204 can receive an input, such as RGB image 2202 (*e.g.*, from step 2108 in FIG. 21), and produce a segmented image 2206 (*e.g.*, a heat map generated from step 2112 in FIG. 21). In some embodiments, segmented image 2206 can be represented as a heat map.

[00172] FIG. 23 illustrates an example input to a pseudo multi-label classifier 2304 that can take an image 2302 (*e.g.*, RGB image) as input and generate a parameter set 2306 as output. The parameter set 2306 can characterize an associated implicit label map under an orthogonal transformation (*e.g.*, DCT, PCA, etc.). Non-limiting examples include the use of a CNN as the pseudo multi-label classifier 2304.

[00173] FIG. 24 illustrates an example parameter set 2402 and a label map 2406 that is formed from the parameter set 2402. The parameter set 2402 is based on an orthogonal transformation using DCT. An image 2404 illustrates the 2D DCT frequencies that form the basis functions of the DCT transformation. A symbol 2408 represents applying the inverse DCT. The inverse DCT can be applied with correct normalization and summing over all the

basis functions from the image 2404, where the product of each basis function and its corresponding coefficient in the parameter set 2402 is calculated. The resulting image can be a label map 2406, which can be a heat map that can show a class associated with each pixel of the segmented image. In some embodiments, a non-DCT transformation can be used instead of the DCT transformation.

[00174] FIG. 35 illustrates an example of using an object detector according to some embodiments of the present disclosure. Object detector 3504 can implement a method of generating a label image using a parameter set. For example, object detector 3504 can implement one or more steps of method 2100 (FIG. 21). Objector detector 3504 can take and process an input image, such as RGB image 3502. The output can be a label image. For example, the resulting label image can be a heat map 3506, where the light regions (e.g., light region 3508) show the roofs in the image and the dark regions (e.g., dark region 3510) show other structures.

Experiment Data

[00175] The accuracy and precision of the output from an object detector using machine learning can vary depending on many factors. These factors include, for example, the object detection method used, the type of input, the complexity of the input, the amount of data used in training the object detector, the quality of the data used in training the object detector, and the similarity between the training data and the input. For example, the more similar the input is to the training data, the more accurate the output is. As another example, using a large amount of training data can generally result in a more accurate output compared to using a small amount of training data.

[00176] FIG. 36 illustrates ground truth data 3604 generated from an RGB image 3602 as input. Ground truth data 3604 can be considered accurate, and thus, it can represent the reference data, to which experimental data can be compared. In some embodiments, the ground truth data 3604 can be generated manually using, for example, an interactive polygon editing tool. This ground truth data 3604 can be used to compare with the outputs from an object detector. FIGS. 37-39 illustrate using the same input as FIG. 36 (e.g., RGB image 3602). The outputs shown in these figures are, however, different because they use different object detection methods. The experimental data described in FIGS. 37-39 is for illustrative purposes.

[00177] FIG. 37 illustrates a binary output 3702 generated from using a hybrid segmentation method, such as method 300 (FIG. 3). The binary output 3702 can be evaluated for accuracy using ground truth data 3604. Another form of heat map 3704 can be used to show the level of accuracy for the binary output 3702. For example, an area where the color is the same or similar to segment 3706 (*e.g.*, light gray) indicates that the area has been accurately segmented. A pixel in this area can be called a "true pixel." On the other hand, an area where the color is the same or similar to segment 3708 (*e.g.*, dark gray) indicates that the area has not been accurately segmented. A pixel in this area can be called a "false pixel." In some embodiments, segmentation of an area can be considered accurate if the level of accuracy exceeds a threshold value. For example, given an area, if the percentage of an object in the area is correctly identified by the binary output 3702 as compared to ground truth data 3604, and the percentage exceeds a threshold value, segmentation of that area can be considered accurate. Moreover, a true positive occurs at a pixel when the observation (*e.g.*, results of a disclosed system such as the binary output 3702) and the ground truth data both indicate the foreground at the pixel. A false positive occurs at a pixel when the observation indicates the foreground and the ground truth data indicates the background at the pixel. A false negative occurs at a pixel when the observation indicates the background and the ground truth data indicates the foreground at the pixel.

[00178] FIG. 38 illustrates a binary output 3802 generated from using a pseudo multi-label classification method, such as method 2100 (FIG. 21). Similar to heat map 3704 (Fig. 37), heat map 3804 can be used to show the level of accuracy for the binary output 3802. Heat map 3804 shows the resulting evaluation of this output, which is compared against ground truth data 3604.

[00179] FIG. 39 illustrates binary output 3902 generated from using a fully convolutional network(s) (FCN) with a pseudo multi-label classification method. The FCN(s) can be used to convert all fully-connected layers of a CNN into equivalent convolutional layers. This can allow input images of arbitrary size to be processed in one forward pass through the network instead of using separate image chips in their own forward passes. Consequently, the convolutional kernels in FCN(s) are required to be evaluated only once, independent of the actual number of chips that take them into account for classification. Similar to heat map 3704 (Fig. 37), heat map 3904 can be used to show the level of accuracy for the binary output 3902.

Machine Learning for Geometry Extraction

Digital Elevation and Height Maps from Multiple Images

[00180] Elevation is taken to be relative to a global reference (*e.g.*, mean sea level). Height is taken to be relative to local environment (*e.g.*, the height of buildings above the surface of neighbors). With one or more images of the same area, an above ground height model (AGHM) image can be generated using a classifier, where every pixel of the AGHM image can be assigned to a height above ground. The classifier can estimate elevations or heights of geographic regions and/or objects in images by direct or relative prediction without performing feature search or extraction. The classifier can generate an AGHM based on the estimation. In some embodiments, the elevations and heights can be represented as a heat map. In some embodiments, the specified geographic regions or objects can be determined relative to a baseline elevation. For example, the baseline elevation can be the ground elevation. Using the ground elevation, each object height, such as a building height, can be relative to the ground on which it stands. In other embodiments, the baseline elevation can be sea level.

[00181] FIG. 25 illustrates an example RGB image 2502 and an example AGHM image 2504. In some embodiments, a classifier can be created so that it takes in an RGB image (*e.g.*, as shown in 2502) as an input and generates an AGHM image (*e.g.*, as shown in 2504) as output.

[00182] In some embodiments, two types of images—a digital surface model (DSM) and a digital elevation model (DEM)—can be used to generate AGHM images. A DSM can represent the elevation of the tops of all reflected surfaces, including both natural and built environments. A DEM image can show the baseline ground elevation without any structure (*e.g.*, buildings) or vegetation (*e.g.*, trees). In some embodiments, the DEM is a 3D surface model. In some embodiments, a model (*e.g.*, DSM, DEM) can be a 2D surface that resides in 3D. In some embodiments, a model can be a 2D surface that projects into a 2D map. In some embodiments, a model can refer to both the 3D and 2D variants. The height above ground can be derived by taking the difference between the DSM and the DEM. Using this difference, the object height that is relative to the ground on which it stands can be determined. Both DSM and DEM can be derived, for example, from LIDAR or stereo reconstruction. Stereo reconstruction can involve taking two or more images of the same object from different locations.

[00183] A machine learning network can be trained to learn methods—such as photogrammetric techniques (*e.g.*, stereo rectification) and variants of block-matching techniques—to determine the disparity or shift between matching points across multiple images. These methods can further rely on explicit regularization and smoothing frameworks (*e.g.*, Markov Random Fields) to ensure that nearby or neighboring pixels have similar depth values and that depth values for each pixel are not far from their original estimates. In some embodiments, a machine learning network can be trained to learn these methods within the context of a deep neural network (DNN).

[00184] In some embodiments, common key points in different images can be identified. The relative distance between these common key points on each respective image can be used to estimate the location of the camera/aperture and/or the location of each key point.

[00185] In some embodiments, a machine learning network can use one or more of the following data items for training to create a classifier: two or more raw 2D images; ground sampling distance (GSD); relative estimated camera/aperture locations for each image; camera location; and an AGHM as ground-truth. In some embodiments, machine learning uses an ANN or a CNN. In some embodiments, a trained classifier can receive 2D images and optionally the camera location(s), and output an AGHM.

[00186] Machine learning has many advantages. For example, a machine-learned device does not require handcrafted feature design and extraction, classical stereo-rectification, epipolar geometry, and block-matching. The device can also accommodate regions of homogeneous intensity or color. The device does not need explicit regularization. Moreover, it is edge-preserving, which is desired for subsequent processing, such as segmentation.

[00187] During the training stage, the label space for the classifier can be a heat map, whose ground truth can be derived from LIDAR measurements or classical stereo vision. For example, LIDAR measurements can be in the form of point clouds or derived data such AGHMs. In some embodiments, the training can be performed with single pixels, such as sparse 3D LIDAR points. During the test or execution stage of the classifier, the output can be a heat map.

[00188] Different feature spaces can be used to train a network. Below is a description of four non-limiting examples of feature spaces.

[00189] First, a feature space can include a set of images defining the same geographic area taken from sensors, whose relative pose (translation and rotation) is fixed. For example, images can be taken from one nadir sensor and four oblique sensors configured in a circular pattern. "Nadir" and "oblique" refer to the orientation of the sensor relative to the ground target. A nadir sensor can point directly below at its target (*e.g.*, camera axis is vertical relative to its target). An oblique sensor can point at angle to its target. The network can be trained just for this particular "stereo" or "n-focal rig." Both the intrinsic camera parameters (*e.g.*, pinhole camera matrix and those related to lens distortion) and the extrinsic camera parameters (*e.g.*, relative position and orientation of camera to a world coordinate system) can be implicit to the model. In other words, these intrinsic and extrinsic camera parameters may not need to be directly estimated.

[00190] Second, a feature space can include a set of images defining the same geographic area taken from sensors whose relative pose is not fixed. The feature space can further include a set of relative poses and intrinsic parameters of these sensors. For example, the pose can be represented as a 6 real-valued number, and each sensor can have real-valued numbers as its parameters. The network can take in these additional parameters in addition to the images.

[00191] Third, a feature space can include a set of images, all of which are projected onto the same geographic surface (*e.g.*, WGS84 with web-mercator), defining the same geographic area. An example includes image tiles which lie in the same coordinate space. The set of relative poses and camera parameters—both intrinsic and extrinsic—is implicit to the model and can be learned by the network. The network can be trained just for a specific "stereo-rig" or "n-focal-rig."

[00192] Fourth, a feature space can include a set of images, all of which are projected onto the same geographic surface (*e.g.*, WGS84 with web-mercator), defining the same geographic area. In contrast to the third example, the set of relative poses and camera parameters—both intrinsic and extrinsic—are part of the feature space. In other words, this set is explicitly given as input to the network. Consequently, the network is not only trained for a specific "stereo-rig" or "n-focal-rig," but also can be used for multiple configurations.

[00193] For all types of features spaces, including the example feature spaces described above, dense reconstruction can be achieved, for example, by applying the training method to scaled images followed by, in post-processing, interpolation/regularization techniques, such as triangulation and Markov Random Fields.

[00194] FIGS. 33A-D illustrate example parameter values related to a sensor in accordance with certain embodiments of the present disclosure. FIG. 33A illustrates extrinsic camera parameters. The R_x, R_y, and R_z refer to rotation about three cardinal axes in a reference frame. The T_x, T_y, and T_z refer to translation in a reference frame. FIG. 33B illustrates intrinsic camera parameters. The $focal_x$ and $focal_y$ refer to focal length in cardinal dimension in an image. The skew refers to shear distortion. The ppt_x and ppt_y refer to the principal point in an image. FIG. 33C illustrates date, time, latitude, and longitude. FIG. 33D illustrates lens distortion parameters. The kappa and tau parameters refer to radial and tangential distortion coefficients often associated with lens.

[00195] FIGS. 10A-C illustrate an example set of a DSM image, a DEM image, and an AGH image, respectively, in accordance with certain embodiments of the present disclosure. In FIG. 10A, DSM image 1002A is a grayscale image, where various heights of the objects in DSM image 1002A are indicated by varying shades of grayscale. For example, the brighter the color (*e.g.*, closer to white), the higher up the indicated object is from a reference level (*e.g.*, higher elevation), compared to the darker color (*e.g.*, closer to black). In some embodiments, the brighter the color indicates, the closer the object is to the sensor or camera that captured the image from above the object. Legend 1004A provides mapping between a spectrum of brightness to an elevation of a given point (measured on top surface). The elevation can, for example, be measured in reference to above the sea level. Legend 1004A provides a range of elevations of about 1310 meters to 1380 meters.

[00196] In FIG. 10B, DEM image 1002B is a grayscale image, where the height of each pixel that represents the area in the image is indicated by varying shades of grayscale. For example, the brighter the color (*e.g.*, closer to white), the higher up the indicated area is from a reference level (*e.g.*, higher elevation), compared to the darker color (*e.g.*, closer to black). In some embodiments, the brighter the color, the closer the area is to the sensor or camera that captured the image from above the area. Legend 1004B provides mapping between a spectrum of brightness to an elevation of a given point (measured from a top surface). The elevation can, for example, be measured in reference to above the sea level. Legend 1004B provides a range of elevations of about 1310 meters to just over 1360 meters. In some embodiments, these elevations are the baseline ground elevations without any built structure or vegetation.

[00197] In FIG. 10C, AGH image 1002C is a grayscale image that is generated based on DSM image 1002A and DEM image 1002B. Legend 1004C provides a mapping between

a spectrum of brightness to an AGH of a given point. In some embodiments, AGH image 1002C can be generated by subtracting the elevation given in DEM image 1002B from the elevation given in DSM image 1002A for every pixel.

[00198] FIGS. 11A-C illustrate another example set of a DSM image, a DEM image, and an AGH image, respectively, in accordance with certain embodiments of the present disclosure. In FIG. 11A, DSM image 1102A is a grayscale image, where various heights of the objects in DSM image 1102A are indicated by varying shades of grayscale. For example, the brighter the color (*e.g.*, closer to white), the higher up the indicated object is from a reference level (*e.g.*, higher elevation), compared to the darker color (*e.g.*, closer to black). In some embodiments, the brighter the color indicates, the closer the object is to the sensor or camera that captured the image from above the object. Legend 1104A provides mapping between a spectrum of brightness to an elevation of a given point (measured on top surface). The elevation can, for example, be measured in reference to above the sea level. Legend 1104A provides a range of elevations of about 1342 meters to 1350 meters.

[00199] In FIG. 11B, DEM image 1102B is a grayscale image, where the height of each pixel that represents the area in the image is indicated by varying shades of grayscale. For example, the brighter the color (*e.g.*, closer to white), the higher up the indicated area is from a reference level (*e.g.*, higher elevation), compared to the darker color (*e.g.*, closer to black). In some embodiments, the brighter the color, the closer the area is to the sensor or camera that captured the image from above the area. Legend 1104B provides mapping between a spectrum of brightness to an elevation of a given point (measured from a top surface). The elevation can, for example, be measured in reference to above the sea level. Legend 1104B provides a range of elevations of about 1342 meters to about 1350 meters. In some embodiments, these elevations are the baseline ground elevations without any built structure or vegetation.

[00200] In FIG. 11C, AGH image 1102C is a grayscale image that is generated based on DSM image 1102A and DEM image 1102B. Legend 1104C provides a mapping between a spectrum of brightness to an AGH of a given point. In some embodiments, AGH image 1102C can be generated by subtracting the elevation given in DEM image 1102B from the elevation given in DSM image 1102A for every pixel.

[00201] Elevations and elevation ranges are not limited to those described in FIGS. 10A-C and 11A-C. Elevations can be any numbers, and elevation ranges can be any range of numbers.

[00202] FIG. 12 illustrates a method 1200 of determining heights of one or more regions in an image according to some embodiments of the present disclosure. In some embodiments, method 1200 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 1200 can be performed at an extractor. In some embodiments, method 1200 can use one of the four feature spaces described above.

[00203] At step 1202, training images can be received. In some embodiments, each of the training images is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, each training image can include one or more regions, where each region includes one or more pixels. The training images can include one or more pixels. In some embodiments, these training images can be the same image type. For example, these training images are all of one of the following image types: RGB, IR, and UV. In some embodiments, the training images can be from one or more sensors. In some embodiments, the one or more sensors can be at fixed positions. In some embodiments, in addition to the training images, parameters related to the position(s) of the sensor(s), orientations of the sensors relative to one another, and/or orientation of illumination source that associates formation of the training images can be received. Examples of these parameters can include, for example, time, date, sun direction, sun position, latitude, longitude, and/or any other suitable parameter or combination of parameters. In some embodiments, a latitude and/or a longitude are coordinate(s) for the position of a sensor. In other embodiments, a latitude and/or a longitude are coordinate(s) for the position of an object or an image. In some embodiments, parameters related to positions of the sensor(s) that are configured to capture the training images can be received. In some embodiments, the training images can be projected onto a same geographic surface that defines the region(s).

[00204] At step 1204, labels can be received. In some embodiments, each of these labels can indicate a height for one or more pixels of a region. In some embodiments, a label can indicate a height for more than one pixel that represents one or more regions.

[00205] At step 1206, a regression model can be created based on the received training images and the received training labels. The regression model can be configured to determine a height of one or more regions of an image. In some embodiments, the regression model can be created using a machine learning system, such as an ANN or a CNN as would be appreciated by one of ordinary skill in the art.

[00206] At step 1208, a target image can be received as an input to the regression model. In some embodiments, the target image is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. The target image can have one or more target regions. In some embodiments, the target image is the same type as the training images. For example, if the training images are RGB, the target image can also be RGB. In other embodiments, the target image is a different type from the training images. In some embodiments, when the target image is a different type from the training images, the target image can be transformed to the same type as the training images.

[00207] At step1210, one or more heights of the one or more target regions can be determined using the regression model. In some embodiments, this regression model is the regression model created at step 1206.

[00208] In some embodiments, during both the training stage and execution stage, additional parameters can be received. These additional parameters can include parameters related to (1) positions of the one or more sensors and (2) orientation of illumination source that associates formation of the training images. Examples of these parameters include time, date, sun direction, sun position, latitude, longitude, and/or any other suitable parameter or combination of parameters.

[00209] FIG. 13 illustrates example images in a feature space 1302 and a label space 1304. Feature space 1302 includes two input images—$Input_1$ and $Input_2$. Label space 1304 includes one input image, which is an AGHM image. An extractor 1306 can take these three images as input, in addition to other images, for training and creating a regression model that is configured to determine the heights of different regions in an image.

Surface Roughness (Object Height) from Shadows

[00210] Given a nadir image, traditional computer vision methods use shadows, date, time, and geographic position (latitude and longitude) information to triangulate height. These methods, however, require a significant amount of human interaction, and thus are slow and expensive. Moreover, these methods are not robust in the presence of vegetation because it is difficult to automatically detect shadow lines on uneven and cluttered surfaces, although in remote sensing images, there is typically enough information for humans to infer shadow lines, if not the height of structures.

[00211] Disclosed methods and systems can directly estimate the elevation or height of a specified region or object from various types of images, such as those from monocular

imagery. The elevation or height can be relative to a baseline elevation and can be represented as a heat map. For example, the baseline elevation can be the ground elevation. Using the ground elevation, each object height, such as a building height, can be relative to the ground on which it stands. In other embodiments, the baseline elevation can be the sea level.

[00212] A machine learning network can be trained on what humans can directly see such that there is no need to segment shadows or extract shadow boundary features which may be noisy. In some embodiments, a machine learning network can be trained to learn these methods within the context of an ANN, a DNN or a CNN as would be appreciated by one of ordinary skill in the art.

[00213] The machine learning network can be trained with various types of data, such as data regarding building height and location (*e.g.*, AGHM and table of values), images of buildings (*e.g.*, nadir or near-nadir imagery), and information regarding the location of the camera and the sun.

[00214] During the training stage, the label space for the classifier can be a heat map, whose ground truth can be derived from LIDAR measurements or classical stereo vision. For example, LIDAR measurements can be in the form of point clouds or derived data such AGHMs. In some embodiments, the training can be performed with single pixels, such as sparse 3D LIDAR points.

[00215] The feature space can include an image representing a geographic area; a date and time; and a geographic point (latitude and longitude). In some embodiments, a set of date and time and a set of latitude and longitude are encoded as real-valued numbers. These two sets can infer the direction of the sun-light when and where the image is acquired.

[00216] For all types of features spaces, including the example feature space described above, dense reconstruction can be achieved, for example, by applying the training method to scaled images followed by, in post-processing, interpolation/regularization techniques, such as triangulation and Markov Random Fields.

[00217] After the machine learning network has been trained, a classifier can be created such that it can receive a single image (*e.g.*, nadir image) and parameters such as sun and camera locations. Using these input values, the classifier can output height estimates, a DEM, or a "roughness map." In some embodiments, the roughness map is analogous to an AGHM but only offers relative height changes or roughness. The output can be in the form of a heat map or a set of values. In some embodiments, a geometry-based model can be built,

where shadows from a 3D model are generated by the known light position and compared to the input data.

Roof Pitch from Shading Using Deep Learning

[00218] Traditional computer vision methods cannot automatically distinguish the pitch of a rooftop from a monocular nadir image. Manual methods require humans to make certain assumptions and include additional data.

[00219] Disclosed methods and systems can automatically distinguish roof pitch from images. For a given roof, disclosed methods and systems can also estimate the roof's dominant pitch. For example, given an image that includes a roof, a disclosed method or system can take the image as an input and determine the dominant pitch of the roof (*e.g.*, as 30 degrees).

[00220] A machine learning network, such as a CNN, can be trained with images such as monocular images; location and heat map of structures of interest; and/or ground truth regarding roof pitch for each building. The images may contain visible and/or non-visible bands (*e.g.*, IR). An advantage lies in having many bands that provide complimentary information. Explicit modeling of reflectance can involve material constants with respect to Lambertian Law and other techniques.

[00221] In some embodiments, the machine learning network can be trained further with information regarding camera and sun location, hyperspectral imagery, multispectral imagery, and/or differential intensity gradients in various spectrums.

[00222] Given multiple faces of the same roof of the same material, the machine learning network can estimate the pitch by using computer graphics models, such as lighting and shading models.

[00223] During the training stage, the label space for the classifier can be a single real-valued number that represents the dominant pitch. For example, the dominant pitch can be expressed in degrees or percentages.

[00224] Different feature spaces can be used to train a network. Here, two non-limiting example feature spaces are provided. First, a feature space can include an image or set of images representing a roof. A network can be trained for a given combination of date, time, and geographic position (latitude and longitude). Second, a feature space can include an image or a set of images representing a roof along with a date, a time, and a geographic position (latitude and longitude) as real-valued numbers. The non-imagery portion of this

feature space can be used to infer the direction of the sun-light at the location and/or time of image acquisition. The non-imagery portion can also be given through a normalization pre-processing of the input image. For example, the input image can be rotated so that the sun direction is from the South and the shadows are cast Northward. The input image can further be scaled so that the shadow cast by a roof of fixed height is always the same size in pixels.

[00225] After the machine learning network has been trained, the classifier can take the appearances of each face in multiple spectra (*e.g.*, RGB, IR) and a model that describes the reflectance of the sun light from a surface with given material properties. Additionally, the material constants can be estimated whenever a sufficient number of facets are visible. The output of the production classifier can include a map of the buildings and roof pitch.

[00226] FIG. 14 illustrates method 1400 for determining a geometric object property according to some embodiments of the present disclosure. In some embodiments, method 1400 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 1400 can be performed at an extractor.

[00227] At step 1402, training images can be received. In some embodiments, each of the training images is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, these training images can be the same image type. For example, these training images are all of one of the following image types: RGB, IR, and UV.

[00228] At step 1404, a geometric object that corresponds to a portion of the training image(s) can be received. In some embodiments, a geometric object can be at least one of a point, a contour, an area, or a binary mask on a training image. In some embodiments, a geometric object can be other suitable portions of the training image. The geometric objects can be used to point to objects or regions of interest in the training images. For example, if a geometric object is an area on a training image, the geometric object can be used to point to the area that covers the boundary of a building of interest in the training image.

[00229] At step 1406, training geometric object properties can be received. Each geometric object property can identify a corresponding geometric object by, for example, identifying a property or attribute of the corresponding geometric object. In some embodiments, a geometric object property can be one of slope, pitch, dominant pitch, material, area, height, or volume. For example, if a corresponding geometric object is a roof, the geometric object property can identify the pitch of the roof.

[00230] At step 1408, parameters related to the orientation of an illumination source for the training images can be received. These parameters can include at least one of time, date, sun direction, sun position, latitude, longitude, object material, and/or any other suitable parameter or combination of parameters. In some embodiments, one or more of these parameters can provide information from the formation of the training images. For example, if a training image of a location is taken at 3:05 pm on July 1, 2000, where the location's latitude is 64.2008 degrees North and the longitude is 149.4937 degrees West, one or more of the following can be used for training: 3:05 pm, July 1, 2000, 64.2008 degrees North, and 149.4937 degrees West.

[00231] At step 1410, a classifier and/or a regression model can be created based on the training images, the geometric object, the training geometric object properties, and the parameters for the training images. The classifier and/or the regression model can be configured to determine a geometric object property for an image. In some embodiments, the classifier and/or the regression model can be created using a machine learning system, such as an ANN or a CNN as would be appreciated by one of ordinary skill in the art.

[00232] At step 1412, a target image and a target geometric object that corresponds to a portion of the target image can be received. In some embodiments, the target image is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image. In some embodiments, the target image can be the same image type as the image type of the training images. For example, the image type of the target image and the training images are all RGB. In some embodiments, a target geometric object can be at least one of a point, a contour, an area, or a binary mask on a training image. In some embodiments, a target geometric object can be other suitable portions of the target image. The target geometric objects can be used to point to objects or regions of interest in the target image. For example, if a target geometric object is an area on the target image, the target geometric object can be used to point to the area that covers the boundary of a building of interest in the target image.

[00233] At step 1414, parameters related to the orientation of an illumination source for the target image can be received. These parameters can include at least one of time, date, sun direction, sun position, latitude, longitude, object material, and/or any other suitable parameter or combination of parameters. In some embodiments, these parameters for the target image can be the same type of parameters as for the training images. In some

embodiments, one or more of these parameters can provide information from the formation of the target image.

[00234] At step 1416, a target geometric object property can be determined. The target geometric object property can identify a corresponding geometric object by, for example, identifying a property or attribute of the corresponding geometric object. In some embodiments, the target geometric object property can be one of slope, pitch, dominant pitch, material, area, height, or volume. For example, if the corresponding geometric object is a roof, the target geometric object property can identify the dominant pitch of the roof.

[00235] FIG. 34 illustrates an example of determining a geometric object property according to some embodiments of the present disclosure. RGB image 3402 can be an input that is provided to an extractor. In some embodiments, the extractor can implement one or more steps of method 1400 (FIG. 14). The extractor can detect that there is an object of interest 3406. In this example, object of interest 3406 is a pool. In some embodiments, object of interest 3406 can be outlined on a different background as shown in image 3404 than the original image (RGB image 3402). In some embodiments, the object of interest 3406 can be identified automatically by using a method, such as method 300 (FIG. 3). In other embodiments, the object of interest can be identified manually. The extractor can identify a target geometric object property for the object of interest 3406. For example, the extractor can determine that the area of the pool is 40 m^2 as shown in 3410. In some embodiments, the target geometric object property can be displayed on top of RGB image 3402 as shown in image 3408.

Deep Learning for Alignment

Machine Learning Multi-Modal Image Stack Registration

[00236] In the domain of recovering structure from remote or aerial imaging, it may be useful to align images of the same scene that are acquired under disparate illumination conditions, viewing geometries, and sensor types and configurations. Traditional computer vision methods use discrete feature matching, feature distance functions, and regularization components in an optimization that are explicitly and separately defined. Traditional computer vision methods also use information theoretic distances or rely on jointly estimating the transformation and an underlying structural model.

[00237] These traditional methods, however, are not scalable, can be inefficient, and can be inaccurate. For example, the traditional methods face difficulty in matching features across disparate bands as the feature appearance can be radically different (*e.g.*, IR band UV band). As another example, the information theoretic distances described above do not scale well with a number of bands.

[00238] To solve these problems, systems and methods according to some embodiments of the present disclosure can apply machine learning, such as DNNs, to multi-modality image registration by learning the parameters of a geometric transformation that maps the space of one image into that of another. These systems and methods can extend to mapping one image set (of co-registered images) to another image set (of co-registered images). In some embodiments, the terms "channels," "bands," and "electronic magnetic spectrums" can be used interchangeably in the context of multi-modality.

[00239] Parcel alignment is one instance of multi-modal image stack registration. Parcel alignment techniques, for example, can align an RGB image with a binary parcel mask.

[00240] In some embodiments, the machine learning based parcel alignment method can be generalized to the problem of multi-modal image registration, where the binary parcel mask can be replaced with a set of co-registered geo-spatial images. In other embodiments, three or more multimodal image stacks can be simultaneously aligned to a common space (*e.g.*, as in bundle adjustment or unbiased atlas formation). Co-registered complementary images can provide more information to support robust registration than single channel (mono-modal) images alone.

Vision Based Parcel Alignment

[00241] There are various databases of (1) geospatial imagery and sensory information, and (2) parcel boundary information, which can provide polygons describing property lines. It is generally useful to have good alignment when overlaying parcel boundary information over the top of imagery or other data layers. This is a known industry challenge for multiple reasons. For example, parcel data can come from disparate databases. As another example, boundaries can be defined and encoded in different ways or projected on different surfaces.

[00242] Imagery, computer vision, and machine learning can be used to derive hints of where parcel boundaries should line up, and then use these hints for alignment to find a best

fit. For example, a method or system can identify probable roadways and seek a uniform setback. As another example, a method or system can identify fences, sidewalks or other typical real property boundaries, and use these for best fit alignment.

[00243] The final parcel boundaries can be estimated by using a robust estimator that minimizes joint distances from all hints, using a distortion model. Example techniques for transformation include similarity, perspective, thin-plate-splines, piece-wise affine, B-Splines, and higher order bivariate polynomials.

Machine Learning Based Parcel Alignment

[00244] Another approach to parcel based alignment involves learning the parameters of a transformation that maps the space of geospatial imagery (and optionally, co-registered sensory information such as IR) to the space of parcel boundaries (encoded as binary mask images) using machine learning methods including, but not limited to, CNNs. Once this transformation, which is represented by a set of parameters, is learned, the parcel boundaries can be mapped to the space of geospatial imagery. The learned transformation can be global or local, and can be recovered in a multi-scale fashion. A transformation can be global if the transformation is applied uniformly to every pixel within an image. A transformation can be local if different parameters of the transformation are used to define the transformation in different regions of the image. Multi-scale in this setting can refer to: (1) (in resolution) the use of a learned transformation at a course level of spatial resolution to initialize the estimation of a transformation at a finer level of spatial resolution; and/or (2) (in transformation parameter complexity) the use of learned transformation of a low order (*e.g.*, translation) to initialize the estimation of a transformation of higher order (*e.g.*,thin-plate-splines). Example techniques for transformation include similarity, perspective, thin-plate-splines, piece-wise affine, B-Splines, and higher order bivariate polynomials. These transformations can be characterized as sets of real-valued numbers.

[00245] FIG. 15 illustrates a method of training a classifier model (*e.g.*, a CNN) for learning the parameters of a transformation that maps the space of geospatial imagery to the space of parcel boundaries in accordance with certain embodiments of the present disclosure. Steps 1502, 1504, and 1506 can be executed during the training stage; steps 1508 and 1510 can be executed during the test stage. At step 1502, a pseudo-random set of transformation can be generated from a selected transformation group. At step 1504, for a given geospatial image and associated parcel data pair, each pseudo-random transformation can be applied to

the given parcel data, thereby producing a family of deformed parcels all based off the original images. In some embodiments, this process can be repeated for many geospatial images and associated parcel data pairs.

[00246] In some embodiments, the input feature space to the classifier can be defined to be the set of geospatial images (and optionally, co-registered sensory information such as infrared) and deformed parcel data. The output label space can be defined to be the associated pseudo-random transformations. At step 1506, the classifier model can be trained using these input-output pairs from step 1504.

[00247] At the test stage, the trained CNN model can be first applied to geospatial imagery (and, optionally, co-registered sensory information) and an associated novel instance of parcel mask to produce the transformation (step 1508). At step 1510, using this transformation, the input parcel mask can be deformed or resampled to the space of the geospatial imagery. In some embodiments, the parcel mask alignment can be refined iteratively by deforming the parcel mask with the output parameters from the trained classifier and repeating process until convergence is achieved. Example tests of convergence can include classifier model cost function minimization, preset maximum number of iterations, and normed change in transformation parameters falls below a set threshold.

Machine Learning for Alignment via Direct Estimation of Transformation Parameters

[00248] In some embodiments, techniques described above related to parcel alignment can be generalized to alignment via direct estimation. Transformation parameters can be used for this purpose.

[00249] FIG. 16 illustrates a method 1600 of identifying parameters of a transformation that maps a first type of image set to a second type of image set in accordance with some embodiments of the present disclosure. In some embodiments, method 1600 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 1600 can be performed at an aligning device.

[00250] At step 1602, a first type of image sets can be received. In some embodiments, each first type of image set can include one or more first type of images. In some embodiments, one or more of the first type of image sets can include two or more co-registered images. Co-registered images can refer to aligned images, where specific pixel coordinates of those images represent the identical physical location (e.g., a location on the

surface of the Earth, often expressed as a latitude and longitude). Consequently, these co-registered images can share, at least in part, the same scene. In some embodiments, one or more of the first type of image sets can include a fixed image stack of co-registered images of disparate bands in the electromagnetic spectrum. In some embodiments, the first type of image sets is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00251] At step 1604, a second type of image sets can be received. In some embodiments, the number of the received first type of image sets can be the same as the number of the received second type of image sets. In some embodiments, each of the second type of image sets can be aligned with a different one of the first type of image sets. In some embodiments, one or more of the second type of image sets can include moving image stack of co-registered images of disparate bands in the electromagnetic spectrum. In some embodiments, the moving image stack can be co-registered with the fixed image stack from step 1602. In some embodiments, the second type can be a parcel map. In some embodiments, the second type of image sets is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00252] At step 1606, transform parameters can be generated. In some embodiments, one or more of these transform parameters are generated using a computer randomizer that can produce pseudo random data. In some embodiments, the number and/or type of generated transform parameters can be customized. In some embodiments, the number and/or type of generated transform parameters can be either statically or dynamically set.

[00253] At step 1608, the transformed second type of image sets can be generated. In some embodiments, these transformed second type of image sets can be generated for each of the first type of image sets, where each of the transformed second type of image sets for each of the first type of image sets can be generated by transforming a respective one of the second type of image sets using a different one of the transform parameters. For example, the moving image stack from step 1604 can be deformed based on the generated transform parameters from step 1606. In some embodiments, this transformation results in the total number of transformed second type of image sets that equals the number of the second type of image sets multiplied by the number of transform parameters. In some embodiments, the operations in step 1608 can produce the ground truth.

[00254] At step 1610, a regressor can be created based on the first type of image sets, the transform parameters, and the transformed second type of image sets. In some

embodiments, the regressor can be trained to learn a transformation as would be appreciated by one of ordinary skill in the art. The regressor can be configured, given a first type of image set and a second type of image set, to identify parameters of a transformation that maps the second type of image set to the first type of image set.

[00255] At step 1612, a target first type of image set can be received. In some embodiments, the target first type of image set is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00256] At step 1614, a target second type of image set can be received. In some embodiments, the target second type of image set is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00257] At step 1616, a target transform parameter can be generated using the regressor based on the received target first type of image set and the received target second type of image set.

[00258] At step 1618, a transformed target second type of images can be generated by transforming the target second type of image set using the target transform parameters so that the transformed target second type of image set can be aligned with the target first type of image set.

[00259] Convergence in transformation between multiple images or image sets can depend on the amount of alignment between them. A condition for meeting convergence in transformation can be set statically or dynamically. For example, a condition can be set such that convergence in transformation is met if two images align for more than 80% (or any other suitable percentage) of their pixels. In some embodiments, a method or system can determine whether convergence in transformation is met. If convergence is not met, steps 1614, 1616, and/or 1618 can be repeated. In some embodiments, these steps can be repeated until convergence in transformation is met.

[00260] The transform parameters and the target transform parameter can be generated and/or used for various transformation techniques, such as translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, and high-order bivariate polynomials. In some embodiments, all the transform parameters and the target transform parameter can use the same technique (*e.g.*, one of the above listed technique).

[00261] FIG. 17 illustrates a training example that uses a sample input 1702, transform parameters 1704, and pairs of feature space and transform parameters 1706. Sample input 1702 can include an RGB image (a first type of image set) and a parcel ground truth image (a

second type of image set). Sample input 1702 can represent example types of data that can be used at steps 1602 and 1604 (FIG. 16). In some embodiments, an aligning device can generate transform parameters 1704. In some embodiments, the aligning device can generate one or more of these transform parameters by using a computer randomizer that can produce pseudo random data. In some embodiments, the number and/or type of generated transform parameters can be customized. In some embodiments, the number and/or type of generated transform parameters can be either statically or dynamically set. This set of parameters can represent an example type of transform parameters generated at step 1606 (FIG. 16). In 1706, a feature space can have a number of sample pairs of images. The first image in a pair (*e.g.*, image 1708) can represent an RGB image, while the second image (*e.g.*, image 1710) in the pair can represent a parcel ground truth image that has been transformed using one of the transform parameters (*e.g.*, transform parameters 1712) . These sample pairs can be examples of transformed second type of image sets from steps 1608 (FIG. 16).

Weighting of Low Resolution Channels with High Resolution Asynchronous Overlay

[00262] There are certain use cases for which one may have multiple sets of geospatial data, none of which are sufficient to answer a given question. But once overlaid, aligned, and used to inform one another, accurate inferences can be drawn.

[00263] For example, one may want to determine roof material, and in particular, distinguish wood shingles from brown asphalt shingles or other materials. One may have access to accurate, high resolution (*e.g.*, 30 cm GSD), low cost imagery in RGB. This imagery can be used to locate and/or identify clean unobscured patches of roof. However, at this resolution brown asphalt shingles may be indistinguishable from brown wood shake shingles in RGB. Further, one may have access to low resolution (*e.g.*, 1.5m GSD) hyperspectral imagery. This imagery can differentiate between asphalt and wood based on hyperspectral reflectance, but this low resolution may be unable to determine whether a patch of roof is obscured (*e.g.*, with a solar panel, skylight, shadow, etc.) and likely to give an inaccurate reading.

[00264] By overlaying the RGB (for accurate masking in the x-y dimensions) and hyperspectral (to obtain signal in the relevant wavelength), one can make an accurate estimate in cases such as the example above. These two image sets may be obtained from different platforms at different times. For example, hyperspectral data from a satellite can be received on a given day and RGB data can be received from an airplane at a later date. The

imagery can be aligned based on coordinates, where the images can be assumed to be of the same object.

[00265] Another example can be overlaying a DEM, which is accurate in z-dimension (height), but must be accurately aligned in x-y using higher resolution RGB imagery.

[00266] FIG. 19A illustrates an example RGB image, which is within visible bands. FIG. 19B illustrates eight example images of different multi-spectral bands that are associated with the RGB image in FIG. 19A. Each of these example images corresponds to light response in discrete bands in the electromagnetic spectrum. Analogously, the red, green, blue bands are sensitive to light centered near 645nm, 555n, and 470nm respectively. Two or more images of different multi-spectral bands can be combined to classify ground material (*e.g.*, vegetation and roof types).

Mixed Image Resolution/Modality Alignment

[00267] Systems and methods in accordance with the present disclosure can align two or more raster images with disparate modalities, such as resolutions, bands, and/or electronic magnetic spectrums. In some embodiments, machine learning is used for estimating corresponding key points across images of disparate modalities and identify transform parameters using these key points. Using these transform parameters, two or more images with disparate modalities can be aligned. Once they are aligned, they can be combined to draw inferences about the scene they describe. These systems and methods solve problems in traditional methods, where hand crafting matching key-point feature extraction across multiple modalities (*e.g.*, IR band vs. UV band) is a difficult task. These traditional methods often rely on sparsely populated joint distribution of image intensities or estimation of underlying structure, which can be inaccurate and are not scalable.

[00268] FIG. 18 illustrates a method 1800 of aligning images based on identifying a second type of image set given a first type of image set in accordance with some embodiments of the present disclosure. In some embodiments, method 1800 can be modified by, for example, having steps combined, divided, rearranged, changed, added, and/or removed. In some embodiments, method 1800 can be performed at an aligning device.

[00269] At step 1802, a first type of image sets can be received. In some embodiments, each first type of image set can include one or more first type of images. In some embodiments, one or more of the first type of image sets can include two or more co-registered images. In some embodiments, one or more of the first type of image sets can

include a fixed image stack of co-registered images of disparate bands in the electromagnetic spectrum. In some embodiments, the first type of image sets is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00270] At step 1804, a second type of image sets can be received. In some embodiments, the number of the received second type of image sets can be the same as the number of the received first type of image sets. In some embodiments, each of the second type of image sets can be aligned with a different one of the first type of image sets. In some embodiments, one or more of the second type of image sets can include moving image stack of co-registered images of disparate bands in the electromagnetic spectrum. In some embodiments, the moving image stack can be co-registered with the fixed image stack from step 1802. In some embodiments, the second type of image sets is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00271] At step 1806, a regressor can be created based on the first type of image sets and the second type of image sets. In some embodiments, the regressor can be created based on learning an association between image stacks of the first type and the image stacks of the second type. The regressor can be configured, given a first type of image set, to identify a second type of image set.

[00272] At step 1808, a target first type of image set and first key points of the first type of image set can be received. In some embodiments, one or more of the first key points can be objects that have been identified manually, semi-manually, or automatically. In some embodiments, an object detector, such as the one described in FIG. 3, can be used to detect one or more of the first key points. In some embodiments, the target first type of image set is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00273] At step 1810, a target second type of image set can be received. In some embodiments, the target second type of image set is one of RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, and/or any other suitable image.

[00274] At step 1812, the target first type of image set can be divided into first divisions. In some embodiments, each of the first divisions can be centered around a different one of the first key points.

[00275] At step 1814, second divisions of the target second type of image set can be generated based on the first divisions. At step 1816, second key points can be identified, where each second key point is from a different division of the second divisions. In some

embodiments, an object detector, such as the one described in FIG. 3, can be used to detect one or more of the second key points.

[00276] At step 1818, transform parameters can be identified. These transform parameters can map the second key points to the first key points. At step 1820, a transformed target second type of image set can be generated by transforming the target second type of image set using the transform parameter so that the transformed target second type of image set is aligned with the target first type of image set. For example, the first key points may be associated with roofs in the target first type of images, and the second key points may be associated with roofs in the target second type of images. To map and align each roof from the target first type of image set to the corresponding roof from the target second type of image set, the target second type of images can be transformed using the transform parameters. In some embodiments, these transform parameters are similar in the format as example target parameters 1704 shown in FIG. 17.

[00277] Convergence in transformation between multiple images or image sets can depend on the amount of alignment between them. A condition for meeting convergence in transformation can be set statically or dynamically. For example, a condition can be set such that convergence in transformation is met if two images align for more than 80% (or any other suitable percentage) of their pixels. In some embodiments, a method or system can determine whether convergence in transformation is met. If convergence is not met, a different transform parameter can be selected and used for transforming the target second type of image set. In some embodiments, the process of using a different transform parameters to transform the target second type of image set can be repeated until convergence is met.

[00278] The transform parameters can be generated and/or used for various transformation techniques, such as translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, and high-order bivariate polynomials. In some embodiments, all the transform parameters can use the same technique (*e.g.*, one of the above-listed techniques).

Geospatial Change Detection

[00279] It can be useful and valuable to flag some geospatial changes as of interest or not of interest. This can be a laborious process when executed manually. Examples of geospatial changes that may be of interest include moss on roof, shingle loss, hurricane

ripping off roof, additions to existing structures, and introduction of new structures. Examples of geospatial changes that may not be of interest include snow on roof and seasonal foliage present on roof.

[00280] In some embodiments, a classifier can be trained to take an image and determine whether there has been a change. If there has been a change, the classifier can further determine whether the change is significant or not significant. In some embodiments, object-oriented deep learning can be used for change detection. For example, the presence of a pool can be determined. As another example, areas of material change can be identified. In some embodiments, parameter changes can be reported.

[00281] In some embodiments, a list of changes that are of interest can be created. For example, such a list can include information on whether someone can build an addition, can cut a tree, the presence of a pool, and the presence of a patio.

[00282] Humans can look at images to determine whether a change has occurred. A neural network in accordance with some embodiments is a structural form used in deep learning that can know, for example, whether the leaves on a tree is a change of consequence. For example, if information related to whether the leaves have fallen off the tree is of interest, the leaves on the tree can be a change of consequence. In some embodiments, crowd source can be utilized to generate ground-truth labeling of objects.

Tree Size Analyzer/Estimator

[00283] In some embodiments, a tree size analyzer or estimator can use geospatial imagery combined with other data sets to create information of interest. For example, given satellite imagery (*e.g.*, 12" GSD RGB, as well as hyperspectral), a tree's diameter and the tree type can be estimated by using the geometry and texture to guess the tree type. Once an estimate of diameter and type are known, a full tree model can be developed based on a separate database of tree growth models. For example, an evergreen tree has a certain shape, meaning that a given diameter can be predictive of a certain height, trunk diameter, bough diameter, and/or mass.

[00284] Information from the tree analyzer can be useful, for example, in estimating the potential damage related to a tree falling on a nearby structure.

Best Guess of Occluded Polygons

[00285] It can be useful and valuable to estimate the underlying shape of an occluded polygon. For example, in a nadir image, the shape of the corner of a house can be estimated even if occluded by an overhanging tree. In this context, a dictionary or library of template shapes to be matched to partially-defined polygons can be maintained. FIG. 31 illustrates an example of a library of template shapes. The different shapes within the dictionary or library can capture the expected modes of variation in the occluded polygons to be estimated. This can be solved with several approaches. One approach can involve the following five steps: (1) segment regions (*e.g.*, houses, trees); (2) fit polygons to segmented regions; (3) where house regions are adjacent to tree regions, assume occluded house regions may underlay the tree regions; (4) use intersection of rectilinear borders of house regions to predict hidden vertices; and (5) construct new house polygon using rectilinear edges and vertices including hidden vertices. Another approach can involve the following three steps: (1) learn or define a set of likely polygons or polygon rules (*e.g.*, rectilinear, 90 degree vertices, etc.); (2) segment house and tree regions; and (3) fit polygon from defined library based on segmentation, and regarding tree segments as uncertain.

Autoencoders for Satellite and Aerial Imagery

[00286] An autoencoder in an artificial neural network can be used to learn a representation (or an encoding) for a set of data (often digital images). It can be viewed as a generative model in that images can be reconstructed from this representation. FIGS. 32A-B illustrate an application of autoencoders to multiple machine learning tasks in accordance with some embodiments of the present disclosure. Block 3200A illustrates that the structure of an autoencoder is like an hourglass with input images (*e.g.*, image 3202A) on one end fed through the first half of the network (*e.g.*, the encoder stage 3204A) to the neck (*e.g.*, code 3206A) and then through the second half of the network (*e.g.*, the decoder stage 3208A) producing the same image (*e.g.*, image 3210A). In some embodiments, autoencoders can be trained to extract the essence of aerial and satellite images that serves to remove redundant information and noise (compression) as well as capture low-level features that characterize this type imagery.

[00287] In some embodiments, an autoencoder (*e.g.*, the encoder 3202B) can be first trained. Subsequently, the resulting encoder half can be used to train a family of networks each for a different purpose (*e.g.*, roof pixel classification) as illustrated in block 3200B. Decoder 3204B can be used to decode any code encoded by encoder 3202B to generate

labels. This approach has additional advantages including (1) training the autoencoder can be fast since ground truth is, by construction, readily available (*e.g.*, any geospatial imagery will) and (2) training newly designed networks are easier since the encoder stage is performed only once.

Geospatial Image Label Transfer

[00288] When trying to automatically identify objects from images using learning algorithms, it can be advantageous to have labeled images and/or labeled attributes. Separately, acquired images require tradeoffs between resolution, coverage, cost and frequency.

[00289] In the case of geospatial images, multiple sets of images can share common, known locations. The images of a common location can be identified via their GIS metadata, and the labels can be shared or transferred. This can be valuable in the following example case: (1) costly high resolution images are gathered and labeled by hand; and (2) these labels are transferred to satellite imagery which is of poorer pixel resolution and could not be labeled by hand for certain features (*e.g.*, roof material). But where there remains a sufficient signal in the data, a useful classifier can still be trained to recognize the attribute in satellite data

[00290] Alternatively, this can also be called geospatial image registration and masking. This can be relevant when there are multiple geospatial data sets that cover the same physical area (*e.g.*, 2 sets of images, both of which can be geo-registered and overlap the same area on the surface of the earth). The core concept is to use one of the data sets to define areas of intersect (*e.g.*, using high resolution RGB imagery to find a clean spot on a rooftop), and a second dataset to extract additional information about the region of interest (*e.g.*, information from a hyperspectral image in the region of interest in order to determine roof material type, which could not be known from the RGB data alone). This can also be used, for example, by getting a digital elevation map from one image source, and a shadow map from a second source, then using the digital elevation map to extract height information from the shadows mapped onto the digital elevation map surface.

Distinguishing Roof Types

[00291] It can be useful and valuable to distinguish roof material attributes (*e.g.*, wood vs tile shingles), as well as roof age at scale. Non-optimal data, such as low resolution satellite images, can be used. A given material can behave in a certain way with respect to reflectance, given known parameters (*e.g.*, sun spectrum/angle/intensity, camera angles), and desired parameters (*e.g.*, age, material). The desired parameters can be extracted from the image based on the hyperspectral data (outside visible range). Calibration can be performed based on (1) ground truth, and (2) an image classifier.

[00292] Additionally, the image data can be made more accurate by properly selecting and sampling uniform aspects of the roof (*e.g.*, on a common plane, avoiding edges, devoid of shadows, roof debris, and structures such as A/C units or skylights).

[00293] From nadir imagery, relative geometry of roof joints (where facets of the roof meet along a ridge) can provide cues for roof geometry. Using a texture-based edge detector, a skeleton of the roof projection can be constructed (*e.g.*, rejecting solar panel roof boundaries). More information can be gathered from a medial axis representation estimated from the roof boundary using traditional techniques (or vice versa). The relative geometry of roof joints (*e.g.*, encoded using histograms of number of angles at an intersection of ridges, and perhaps lengths of incident edges) can characterize roof type and complexity.

Roof Pitch from Drone Aerial Imagery

[00294] Classical computer vision methods can be applied to recovering roof pitch from aerial images (*e.g.*, from airplanes and drones), global positioning systems (GPS), inertial measurement units (IMUs), and gyroscopes. These methods include a range of techniques from recovering depth from stereo to the more sophisticated structure from motion (SFM) where bundle adjustment and graph optimization methods are used with images acquired around and above roofs to reconstruct shape. In all methods, the optimization process is constrained to planar surfaces, suitable for structures of the built environment.

[00295] Roof footprints localized on the acquired imagery, or mapped to the same from another geospatial imagery source, can be used to identify facet (planar surfaces) to roof associations. Estimating the dihedral angles made by these facets and a horizontal ground plane produce the roof pitch.

Roof Pitch from Shading

[00296] From a 3D roof model, the calculation of pitch/slope of a roof facet immediately follows from geometry. In this method, a 3D roof model is fitted using both spatial and appearance information. Shading information can be used as a cue for form, position, and depth (in particular, roof facet orientation) and depends on material properties. Non-limiting examples of shading model components include: ambient reflection, diffuse (Lambertian) reflection, specular reflection, etc. Given a sufficient number of input image channels (*e.g.*, RGB, panchromatic, IR, UV, multi-spectral, hyperspectral, etc.), roof facet orientation and intrinsic color/texture (proxy for material) can be estimated using a shading model.

[00297] In another embodiment, the parameter space of this model (*e.g.*, intrinsic color, facet orientation, etc.) can be reduced using any combination of (but not limited to) the following simplifying assumptions: (1) roof ridge lines are parallel to horizontal ground plane; (2) neighboring roof facets incident to the same ridge line are symmetric in the sense that they have the same pitch/slope; and/or (3) the space of allowable roof pitch/slope is limited to a discrete set of common values. Non-limiting examples of parameters include values of run/rise that are 12/1, 12/2, 12/3, etc., where the run value can be defined to be the horizontal length, and the rise value can be defined to be the vertical length of a general right triangle, whose hypotenuse is the roof surface. The roof surface can be used to calculate the roof pitch or slope.

Condition Estimation

[00298] In some embodiments, systems and methods can be utilized to estimate conditions of a road, parcel, and/or roof (*e.g.*, effective age).

[00299] It is to be understood that the disclosed subject matter is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The disclosed subject matter is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting. Unless otherwise noted, examples provided in this disclosure are not limiting.

[00300] As such, those skilled in the art will appreciate that the conception, upon which this disclosure is based, may readily be utilized as a basis for the designing of other structures, systems and methods for carrying out the several purposes of the disclosed subject

matter. It is important, therefore, that the claims be regarded as including such equivalent constructions insofar as they do not depart from the spirit and scope of the disclosed subject matter.

[00301] Although the disclosed subject matter has been described and illustrated in the foregoing exemplary embodiments, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of implementation of the disclosed subject matter may be made without departing from the spirit and scope of the disclosed subject matter, which is limited only by the claims which follow.

[00302] For example, a device, such as an object detector or an aligning device, can cover implementations in a single hardware/software device or in multiple hardware/software devices. In some embodiments, each of the multiple hardware/software devices can process different parts of the device functionality.

[00303] As another example, machine learning can cover various deep learning architectures, including a deep neural network (DNN), a convolutional neural network (CNN), a deep belief network (DBN), and a recurrent neural network (RNN). Moreover, a reference to a particular architecture does not limit the given application to that particular architecture. Systems and methods in accordance with embodiments of this disclosure can be applied to any type of machine learning or deep learning architecture.

[00304] Yet in another example, unless otherwise noted, the term image can cover any kind of image from any detectable band of the electromagnetic spectrum, including IR, UV, panchromatic, multi-spectral, or hyperspectral images. An image can be RGB, grayscale, or black and white. An image can be any resolution, including high resolution and low resolution. An image may be taken by various methods and/or devices, including satellites, drones, robots, stationary or moving camera or sensor devices, and humans using a camera or sensor device. An image may be taken as a digital image or as an analog image. An image may have been originated as an analog image but later converted to a digital format. An image can indicate a 2D array of pixels with one or more mutually co-registered channels or bands per pixel, summarizing the same area. Non-limiting examples include (1) a 1-channel image that can be a panchromatic image, a grayscale image, a label map, a binary mask, or any other suitable image with one channel; (2) a 3-channel image that can be an RGB image, a blue-green-red (BGR) image, or any other suitable image with three channels; (3) a 4-channel image that can be an RGB and IR image or any other suitable image with four channels; and (4) a 8-channel image that can be an 8-band multi-spectral image. Generally,

the channels can represent different bands of the electromagnetic spectrum, to which a detector is sensitive.

CLAIMS

What is claimed is:

1. A method of identifying a class of a pixel in a target image, comprising:

receiving, at an object detector, training images, wherein each training image includes pixels;

receiving, at the object detector, labels, wherein each label associates a class to each of the pixels of the training images;

creating, at the object detector, a classifier configured to identify the class of each pixel based on the training images and the labels;

receiving, at the object detector, a target image;

dividing, at the object detector, the target image into a plurality of divisions, wherein each division includes a plurality of pixels;

selecting, at the object detector, a target pixel from the plurality of pixels; and

determining, at the object detector using the classifier, a target class for the target pixel of the target image by comparing the target pixel to neighboring pixels from the plurality of pixels.

2. The method of claim 1 further comprising generating a segmented image based on the target image and the determined target class for the target pixel.

3. The method of claim 2, wherein the segmented image is generated using one of a graph-cut, GrabCut, conditional random fields, or manual technique.

4. The method of claim 1, wherein each of the training images and the target image is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

5. The method of claim 1, wherein the classifier is created using a convolutional neural network.

6. The method of claim 1, wherein the neighboring pixels are immediately adjacent to the target pixel.

7. The method of claim 1, wherein each of the class and the target class is at least one of a building, roof, pool, road, trampoline, automobile, truck, boat, plane, communication tower, power transmission tower, hedge, porch, patio, deck, or a tree trunk.

8. A method of generating a label image based on a parameter set, comprising:

receiving, at an object detector, training images;

receiving, at the object detector, training parameter sets, wherein each of the training parameter sets corresponds to a transformed representation of one or more labels, and wherein the one or more labels correspond to a different training image;

creating, at the object detector, a classifier configured to determine a parameter set for an image based on the training images and the training parameter sets;

receiving, at the object detector, a target image;

determining, at the object detector using the classifier, a target parameter set that corresponds to one or more target labels for the target image, wherein the target parameter set corresponds to a target transformed representation of the one or more target labels; and

generating, at the object detector, a label image by applying an inverse target parameter set to the target image, wherein the inverse target parameter set corresponds to an inverse transformation of a transformation represented by the target parameter set, wherein one or more pixels of the label image are associated with a class.

9. The method of claim 8, wherein each of the one or more classes is either two-class or multi-class.

10. The method of claim 8 further comprising receiving second training parameter sets for the training images, wherein the second training parameter sets comprise at least one of time, date, sun direction, sun position, latitude, or longitude.

11. The method of claim 8, wherein each of the training parameters sets and the target parameter set represents a transformation created using one of discrete cosine transform, wavelets, discrete Fourier transform, principal component analysis, non-negative Matrix Factorization, or Hadamard transform.

12. The method of claim 8, wherein each of the training images and the target image is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

13. A method of determining heights of one or more regions in a target image, comprising:

receiving, at an extractor, training images, wherein each of the training images includes one or more regions, wherein each region includes one or more pixels;

receiving, at the extractor, labels that indicate a height for each pixel of the one or more regions of each training image;

creating, at the extractor, a regression model configured to determine a height of a region based on the training images and the labels;

receiving, at the extractor, a target image comprising a target region; and

determining, at the extractor using the regression model, a height of the target region in the target image.

14. The method of claim 13, wherein each of the training images and the target image is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

15. The method of claim 13, wherein receiving the training images comprises receiving the training images from one or more sensors at fixed positions.

16. The method of claim 15, wherein receiving the training images further comprises receiving orientations of the sensors relative to one another if the training images are from two or more sensors.

17. The method of claim 13, wherein receiving the training images comprises receiving the training images from one or more sensors and parameters related to at least one of one or more positions of the one or more sensors, orientation of illumination source that associates formation of the training images, time, date, latitude, or longitude.

18. The method of claim 13, wherein the training images are projected onto a same geographic surface that defines the one or more regions.

19. The method of claim 18 further comprising receiving, at the extractor, parameters related to positions of one or more sensors configured to capture the training images.

20. A method of determining a geometric object property in a target image, comprising:

receiving, at an extractor, training images;

receiving, at the extractor, a geometric object corresponding to a portion of one or more of the training images;

receiving, at the extractor, training geometric object properties, wherein each of the training geometric object properties identifies a corresponding geometric object;

receiving, at the extractor, first one or more parameters related to orientation of illumination source that associates formation of the training images for the training images;

creating, at the extractor, at least one of a classifier or a regression model configured to determine a geometric object property for an image based on the training images, the geometric object, the training geometric object properties, and the first one or more parameters;

receiving, at the extractor, a target image;

receiving, at the extractor, a target geometric object corresponding to a portion of the target image;

receiving, at the extractor, second one or more parameters related to orientation of illumination source that associates formation of the target image; and

determining, at the extractor using the at least one of a classifier or a regression model, a target geometric object property associated with the target geometric object.

21. The method of claim 20, wherein each of the training images and the target image is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

22. The method of claim 20, wherein the training geometric object property is at least one of slope, pitch, dominant pitch, material, area, height, or volume, and wherein the target geometric object property is at least one of slope, pitch, dominant pitch, material, area, height, or volume.

23. The method of claim 20, wherein the geometric object is at least one of a point, a contour, an area, or a binary mask, and wherein the target geometric object is at least one of a point, a contour, an area, or a binary mask.

24. The method of claim 20, wherein the first one or more parameters comprises at least one of time, date, sun direction, sun position, latitude, longitude, or object material, and wherein the second one or more parameters comprise at least one of time, date, sun direction, sun position, latitude, longitude, or object material.

25. A method of aligning images, comprising:

receiving, at an aligning device, N first type of image sets, wherein N>1, wherein each of the N first type of image sets includes one or more first type of images;

receiving, at the aligning device, N second type of image sets, wherein each of the N second type of image sets includes different one or more second type of images, wherein each of the N second type of image sets is aligned with a different one of the N first type of image sets;

generating, at the aligning device, M transform parameters, wherein M>1;

generating, at the aligning device, M transformed second type of image sets for each of the N first type of image sets so that there are N * M total transformed second type of image sets, wherein each of the M transformed second type of image sets for each of the N first type of image sets is generated by transforming a respective one of the N second type of images sets using a different one of the M transform parameters;

creating, at the aligning device, a regressor configured to identify parameters of a transformation that maps a second type of image set to a first type of image set, wherein the regressor is created based on the N first type of image sets, the M transform parameters, and the N * M total transformed second type of image sets;

receiving, at the aligning device, a target first type of image set and a target second type of image set;

generating, at the aligning device using the regressor, a target transform parameter based on the target first type of image set and the target second type of image set; and

generating, at the aligning device, a transformed target second type of image set by transforming the target second type of image set using the target transform parameter so that the transformed target second type of image set is aligned with the target first type of image set.

26. The method of claim 25, wherein the first type of images are red-green-blue images, and the second type of images are parcel maps.

27. The method of claim 25, wherein each image in the first type of image sets, the second type of image sets, the target first type of image set, and the target second type of image set is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

28. The method of claim 25 further comprising determining, at the aligning device, whether the target transform parameter results in convergence.

29. The method of claim 28 further comprising, if it is determined that the target transform parameter does not result in convergence, performing the following steps until convergence: (1) generating, at the aligning device, a new target transform parameter; and (2) determining, at the aligning device, whether the new target transform parameter results in convergence.

30. The method of claim 25, wherein one or more of the N first type of image sets includes a plurality of co-registered first type of images.

31. The method of claim 25, wherein the transform parameters and the target transform parameter are for at least one of translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, or high-order bivariate polynomials.

32. A method of aligning images, comprising:

receiving, at an aligning device, N first type of image sets, wherein N>1, wherein each of the N first type of image sets includes one or more first type of images;

receiving, at the aligning device, N second type of image sets, wherein each of the N second type of image sets includes different one or more second type of images, wherein each of the N second type of image sets is aligned with a different one of the N first type of image sets;

creating, at the aligning device, a regressor configured to identify a second type of image set given a first type of image set, wherein the regressor is created based on the N first type of image sets and the N second type of image sets;

receiving, at the aligning device, a target first type of image set and a plurality of first key points of the target first type of image set;

receiving, at the aligning device, a target second type of image set;

dividing, at the aligning device, the target first type of image set into a first plurality of divisions, wherein each of the first plurality of divisions is centered around a different one of the plurality of first key points;

generating, at the aligning device using the regressor, a second plurality of divisions of the target second type of image set based on the first plurality of division of the target first type of image set;

identifying, at the aligning device, a plurality of second key points, wherein each second key point is from a different division of the second plurality of division;

identifying, at the aligning device, transform parameters that maps the plurality of second key points to the plurality of the first key points; and

generating, at the aligning device, a transformed target second type of image set by transforming the target second type of image set using the transform parameters so that the transformed target second type of image set is aligned with the target first type of image set.

33. The method of claim 32, wherein each image in the first type of image sets, the second type of image sets, the target first type of image set, and the target second type of image set is one of a red-green-blue, panchromatic, infrared, ultraviolet, multi-spectral, or hyperspectral image.

34. The method of claim 32, wherein the transform parameters are for at least one of translation, similarity, perspective, thin-plate-splines, piece-wise affine, B-spline, or high-order bivariate polynomials.

35. The method of claim 32, wherein one or more of the N first type of image sets includes a plurality of co-registered first type of images.

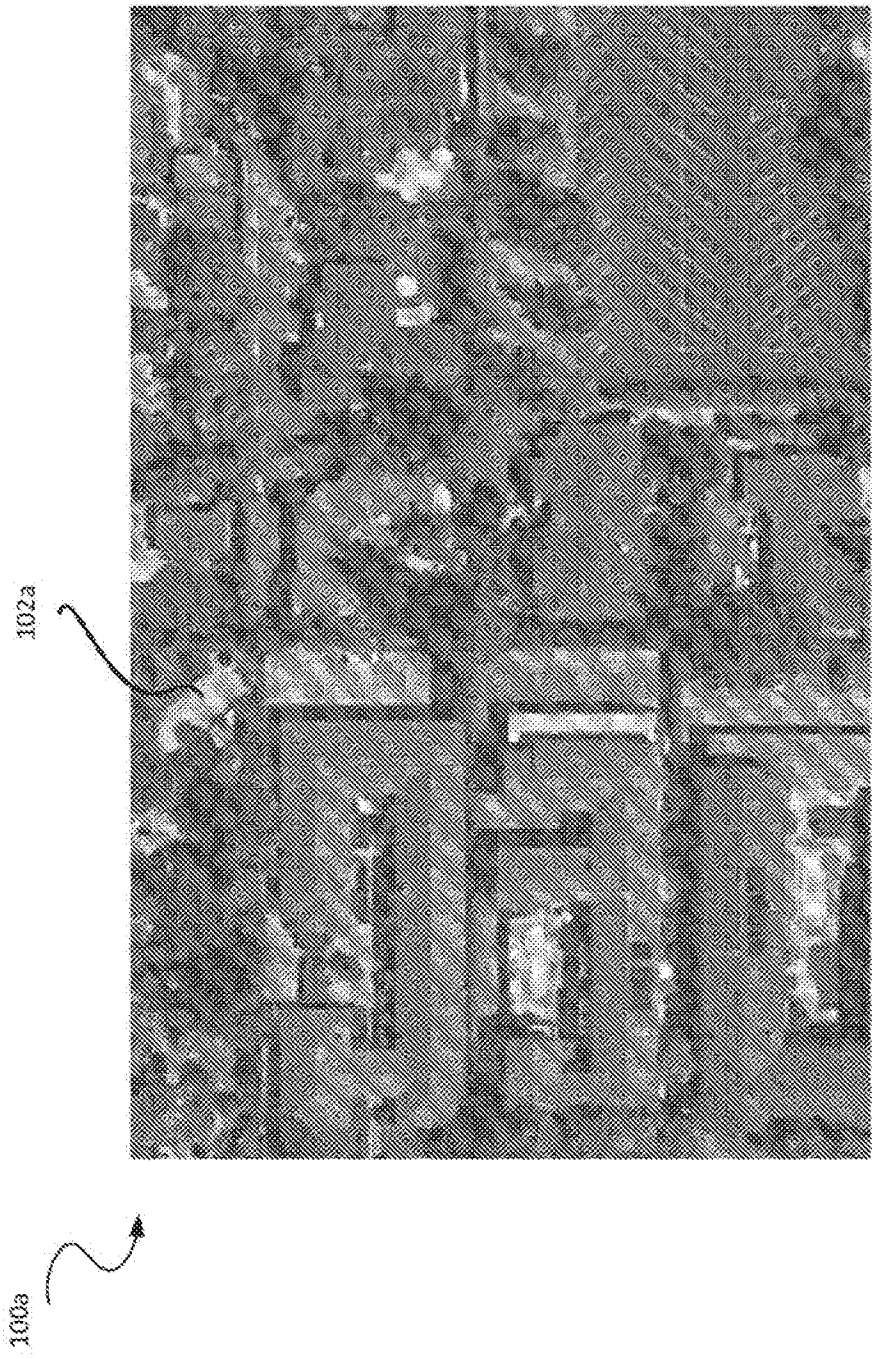


FIG. 1A

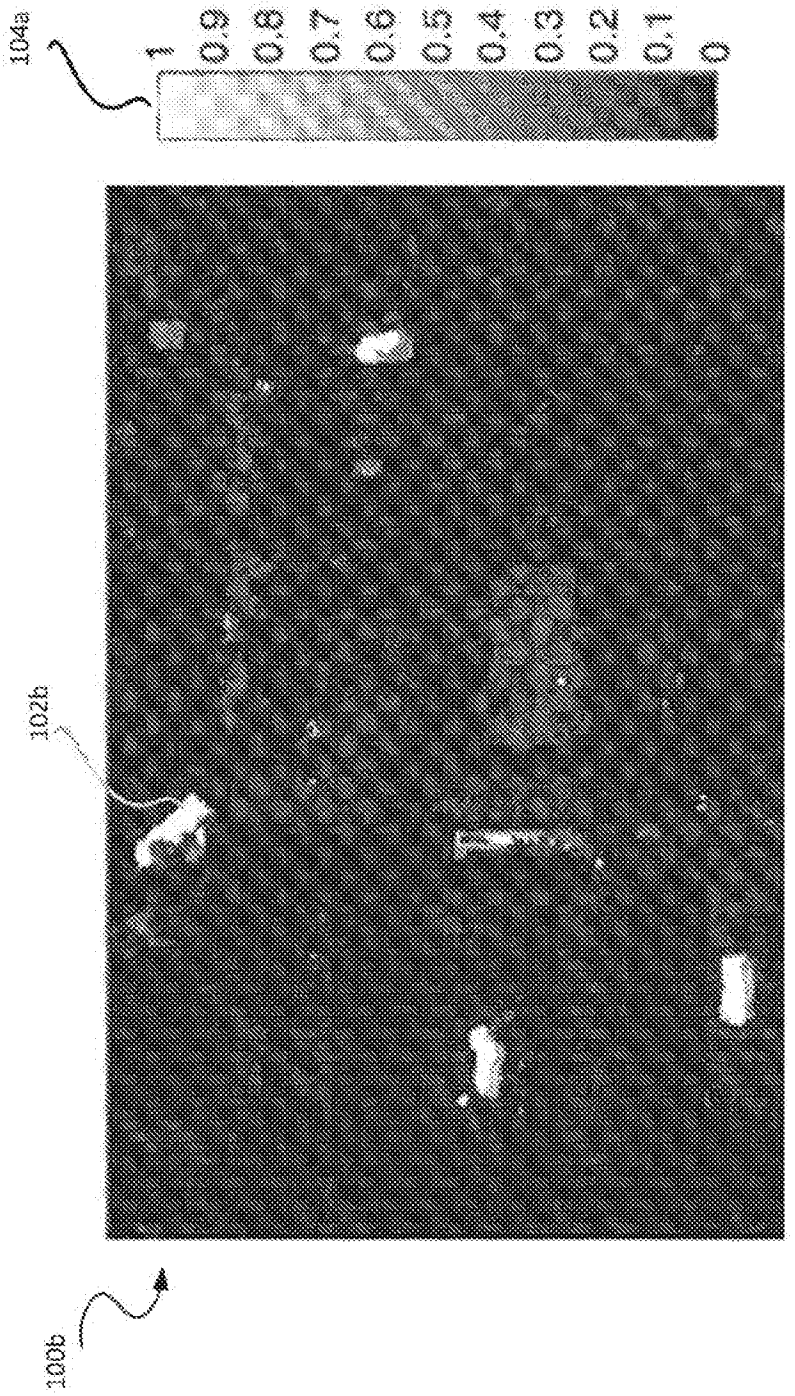


FIG. 1B

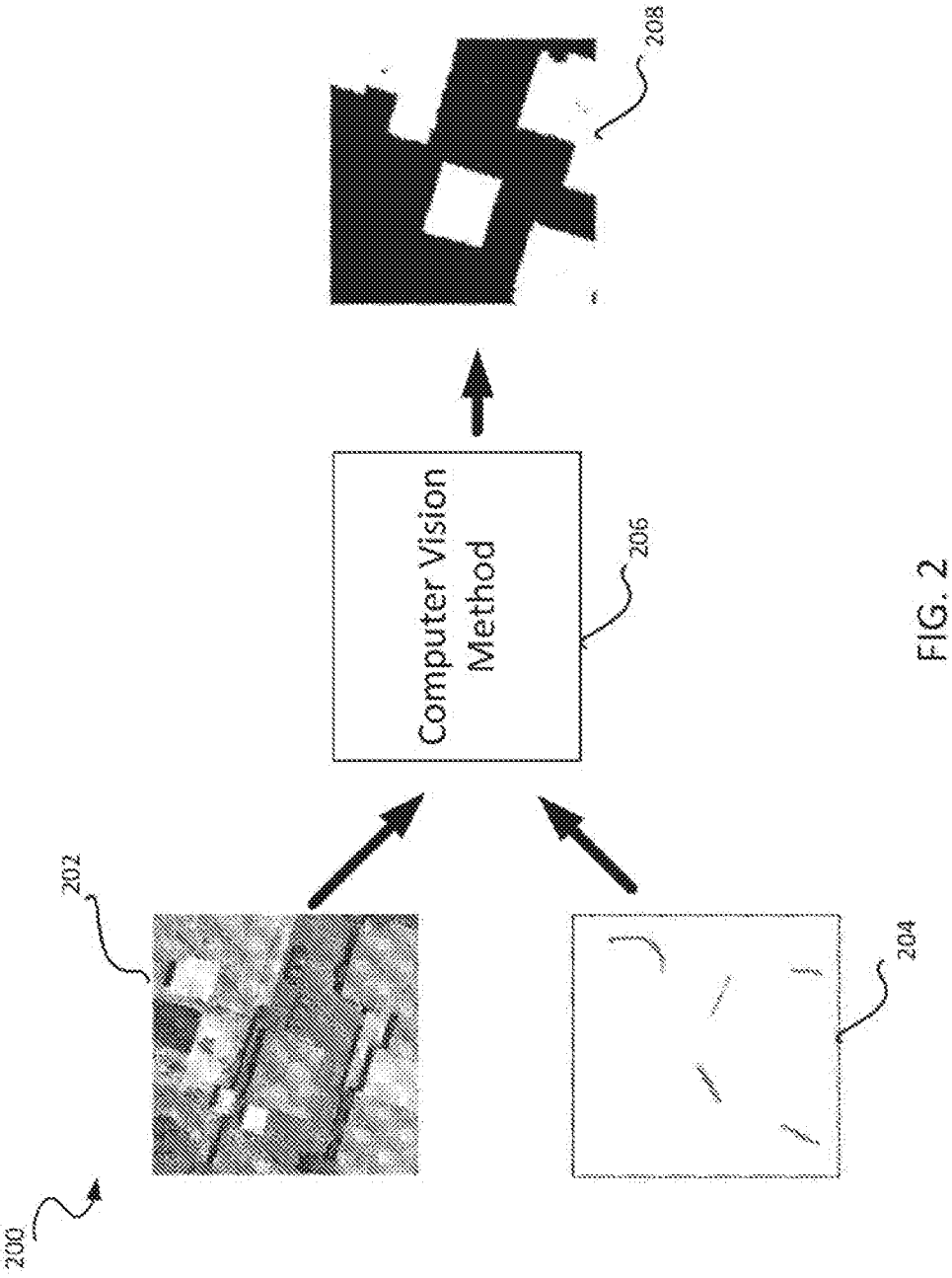


FIG. 2

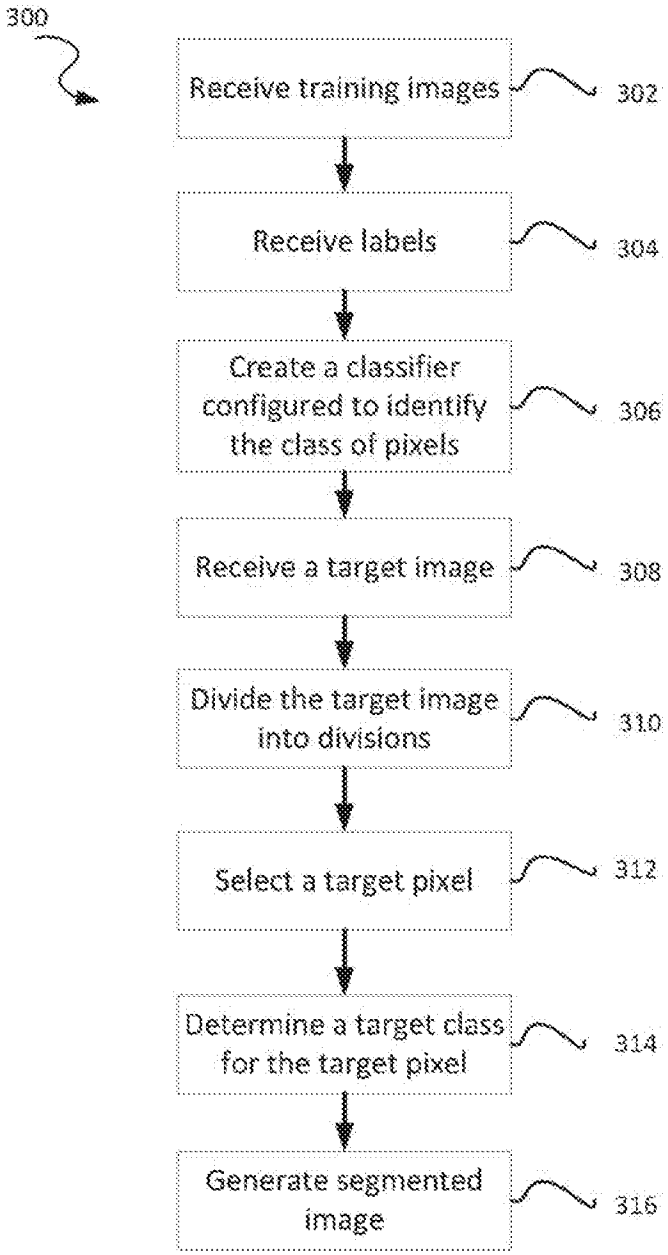

300
Receive training images
302
Receive labels
304
Create a classifier configured to identify the class of pixels
306
Receive a target image
308
Divide the target image into divisions
310
Select a target pixel
312
Determine a target class for the target pixel
314
Generate segmented image
316

FIG. 3

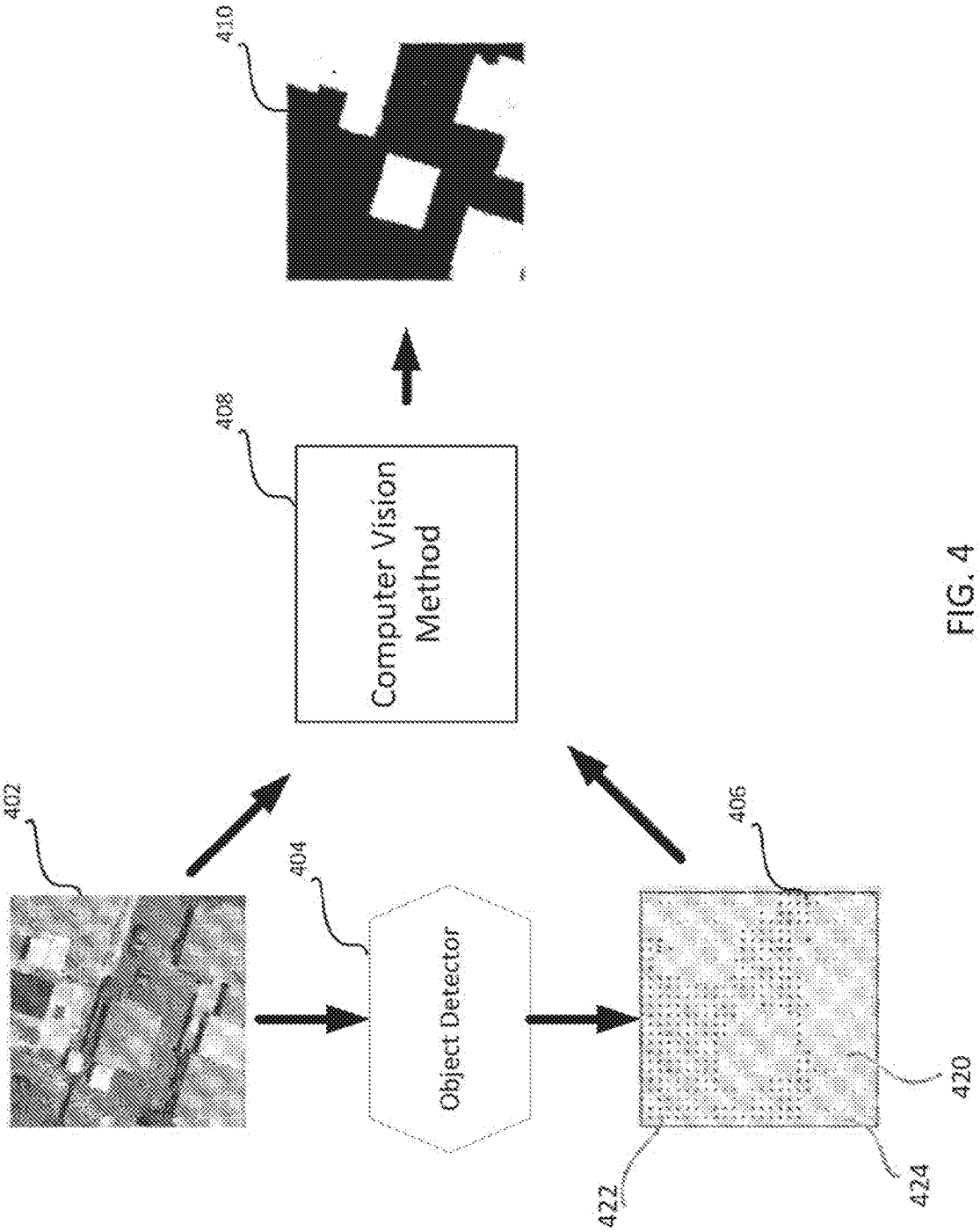

410
408
Computer Vision Method
402
404
Object Detector
406
420
422
424

FIG. 4

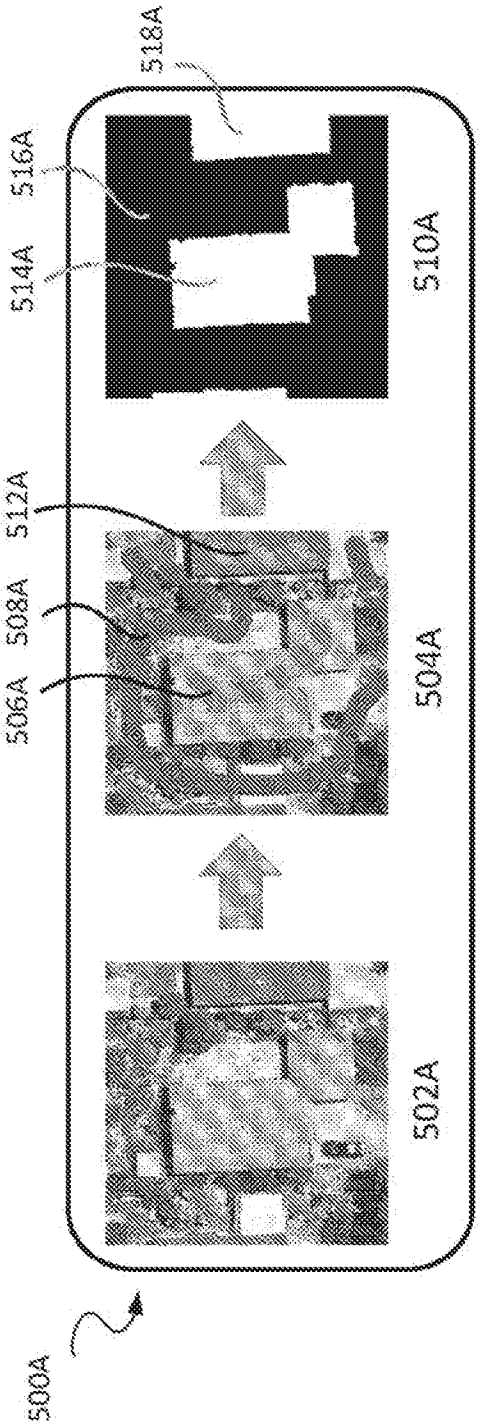


FIG. 5A

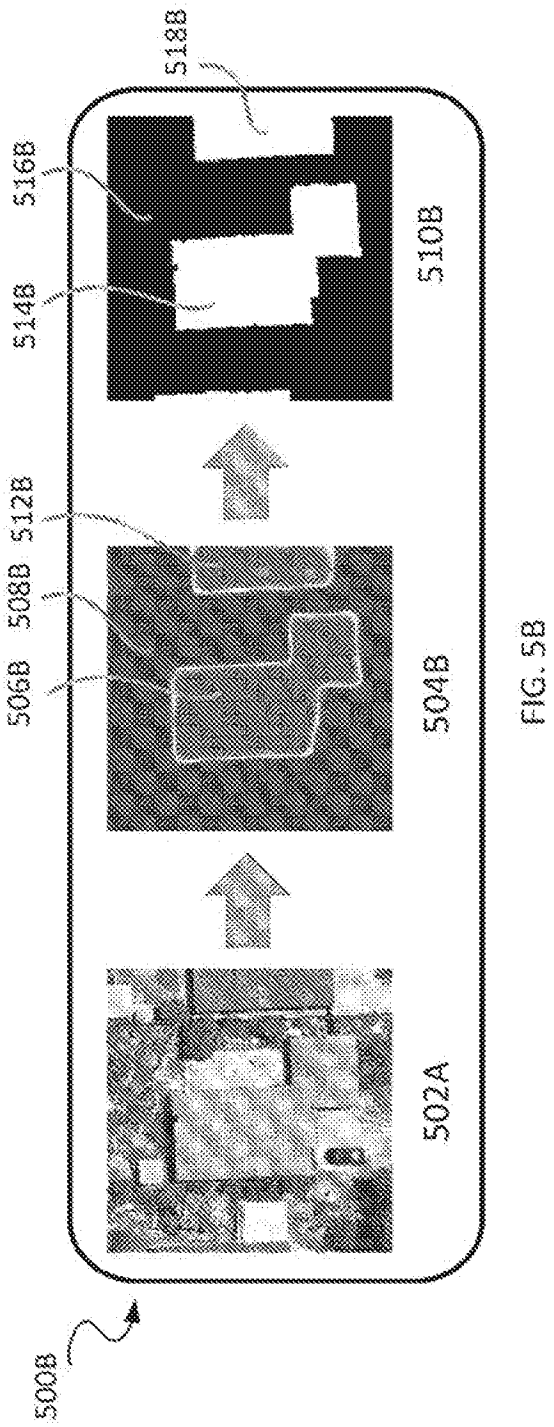


FIG. 5B

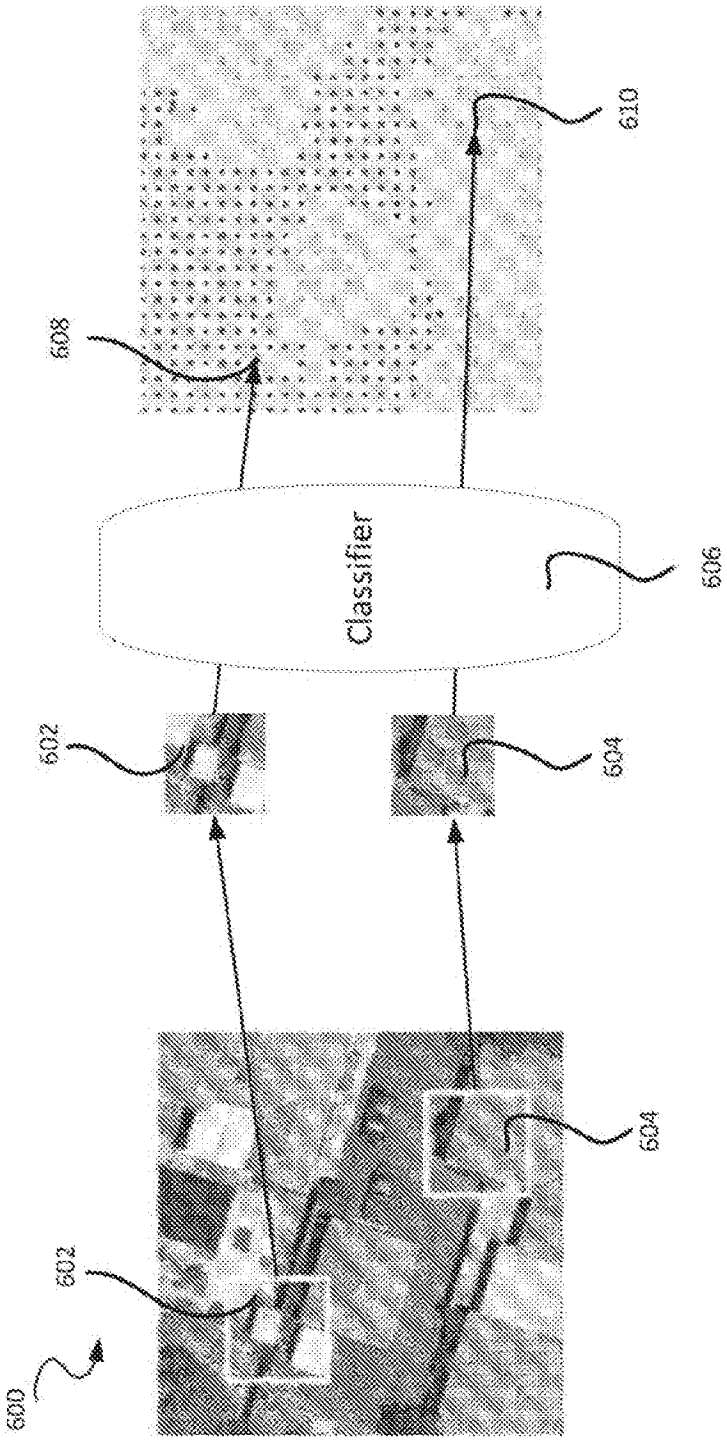


FIG. 6

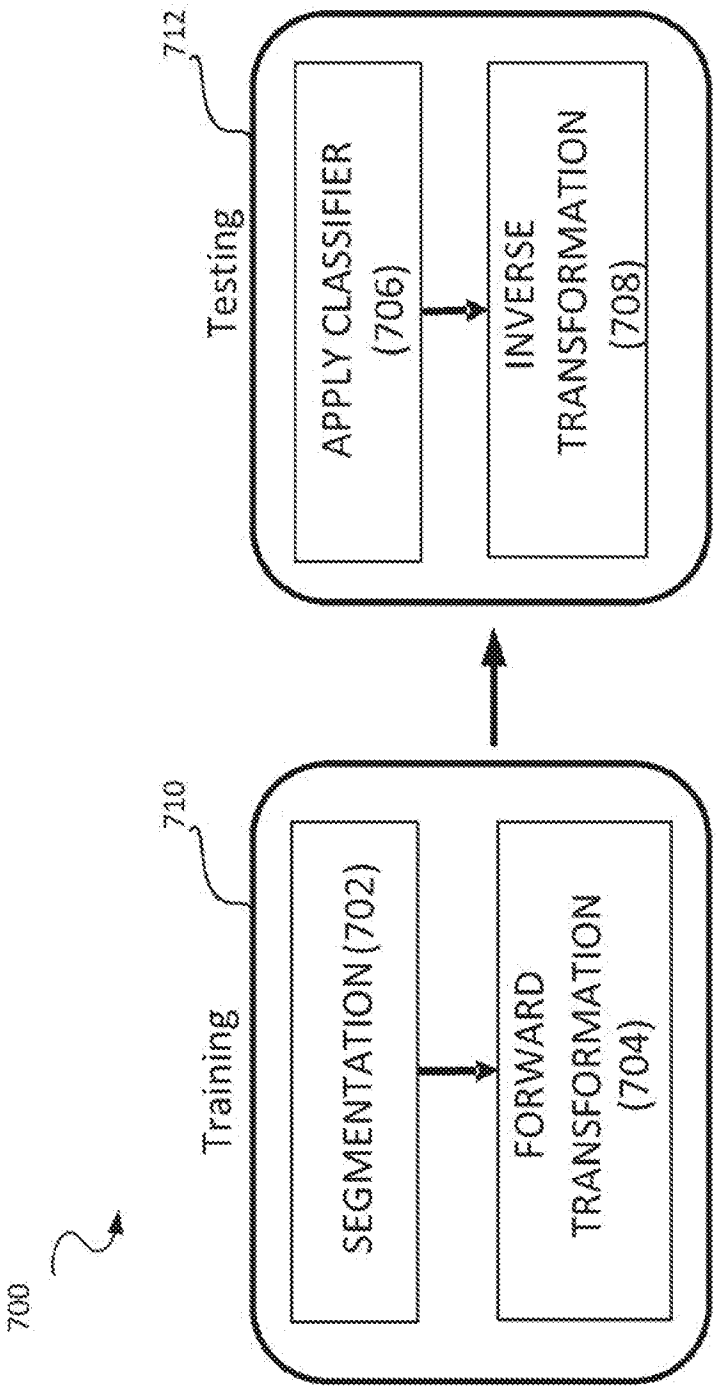

700
710
Training
SEGMENTATION (702)
FORWARD TRANSFORMATION (704)
712
Testing
APPLY CLASSIFIER (706)
INVERSE TRANSFORMATION (708)

FIG. 7

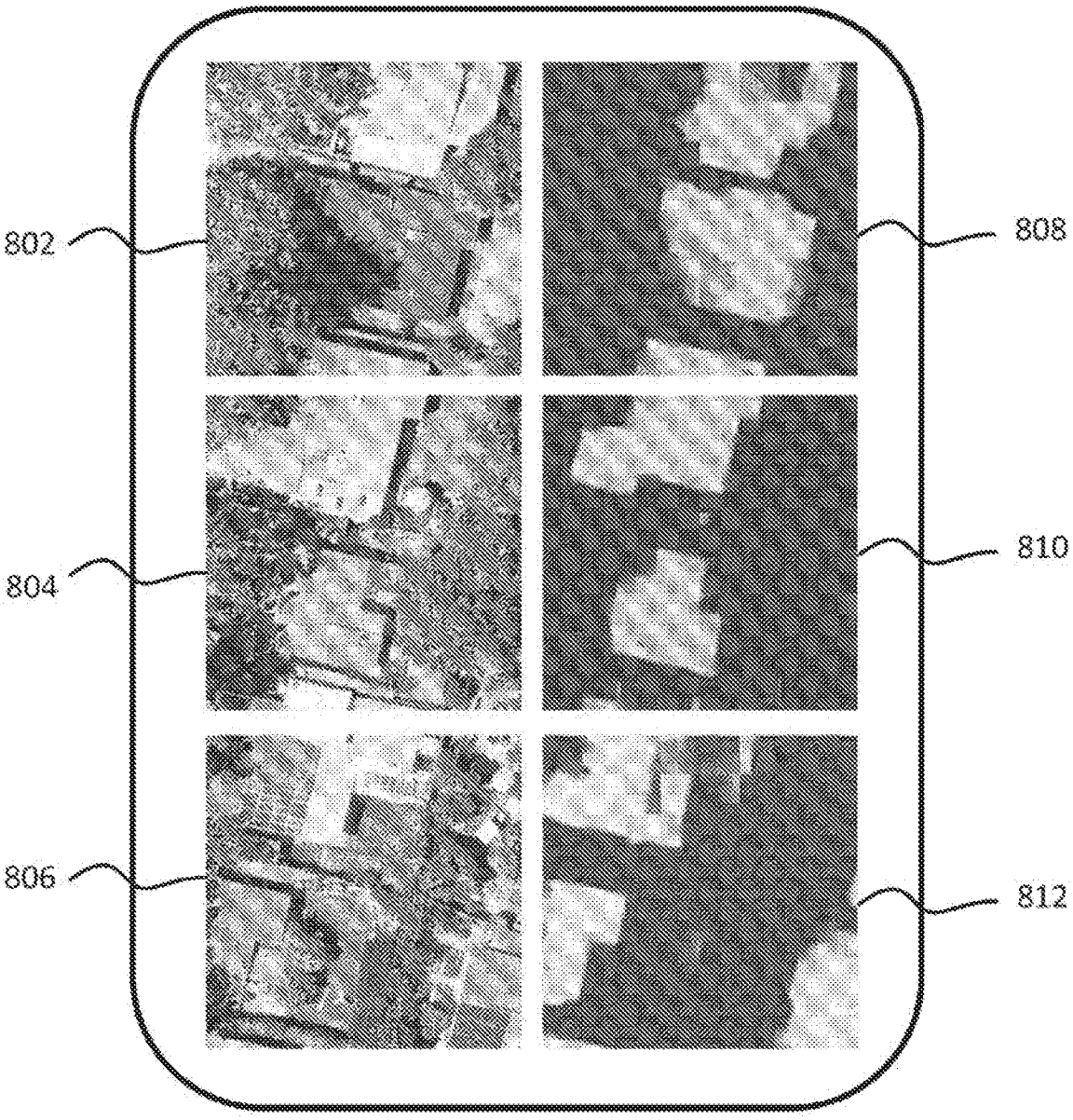


FIG. 8

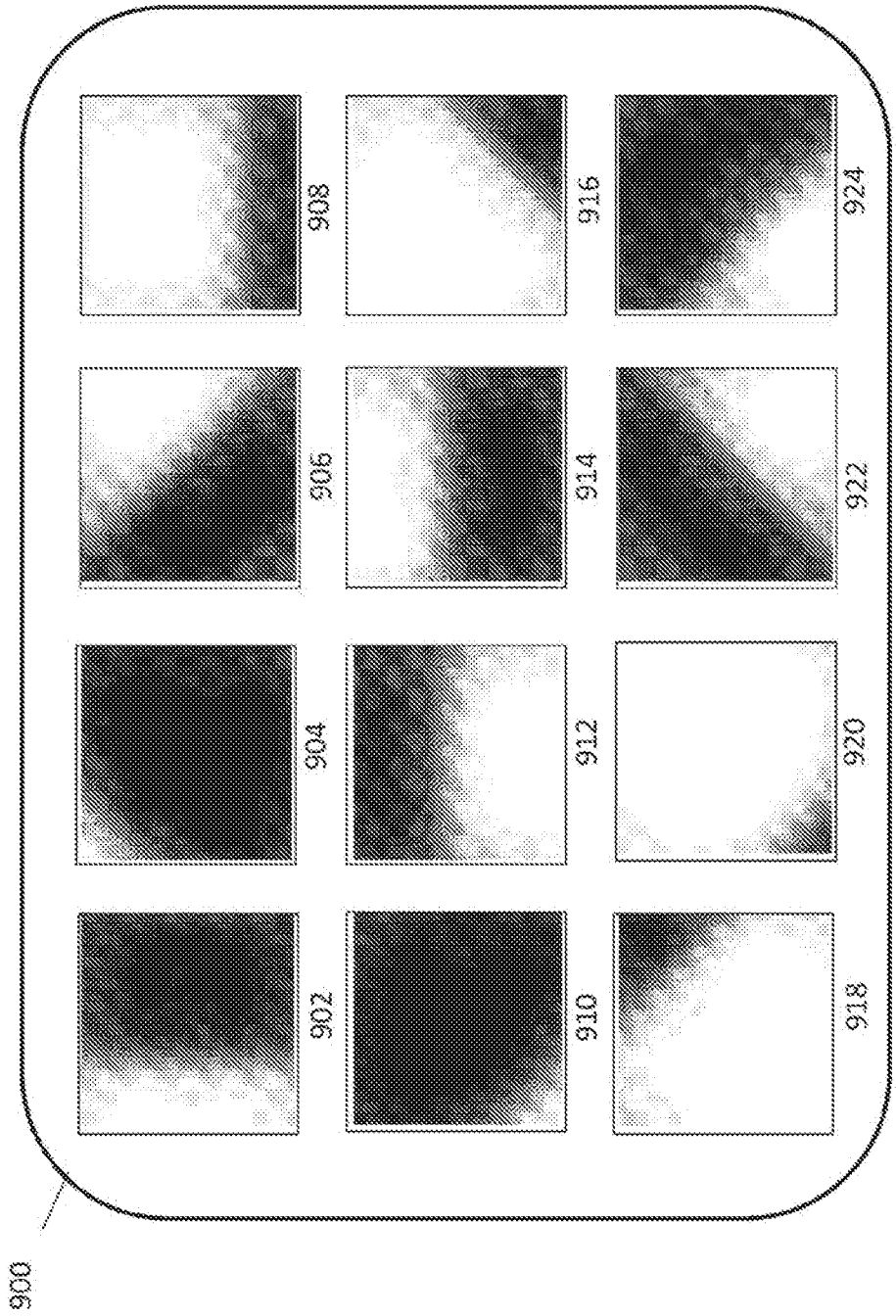


FIG. 9

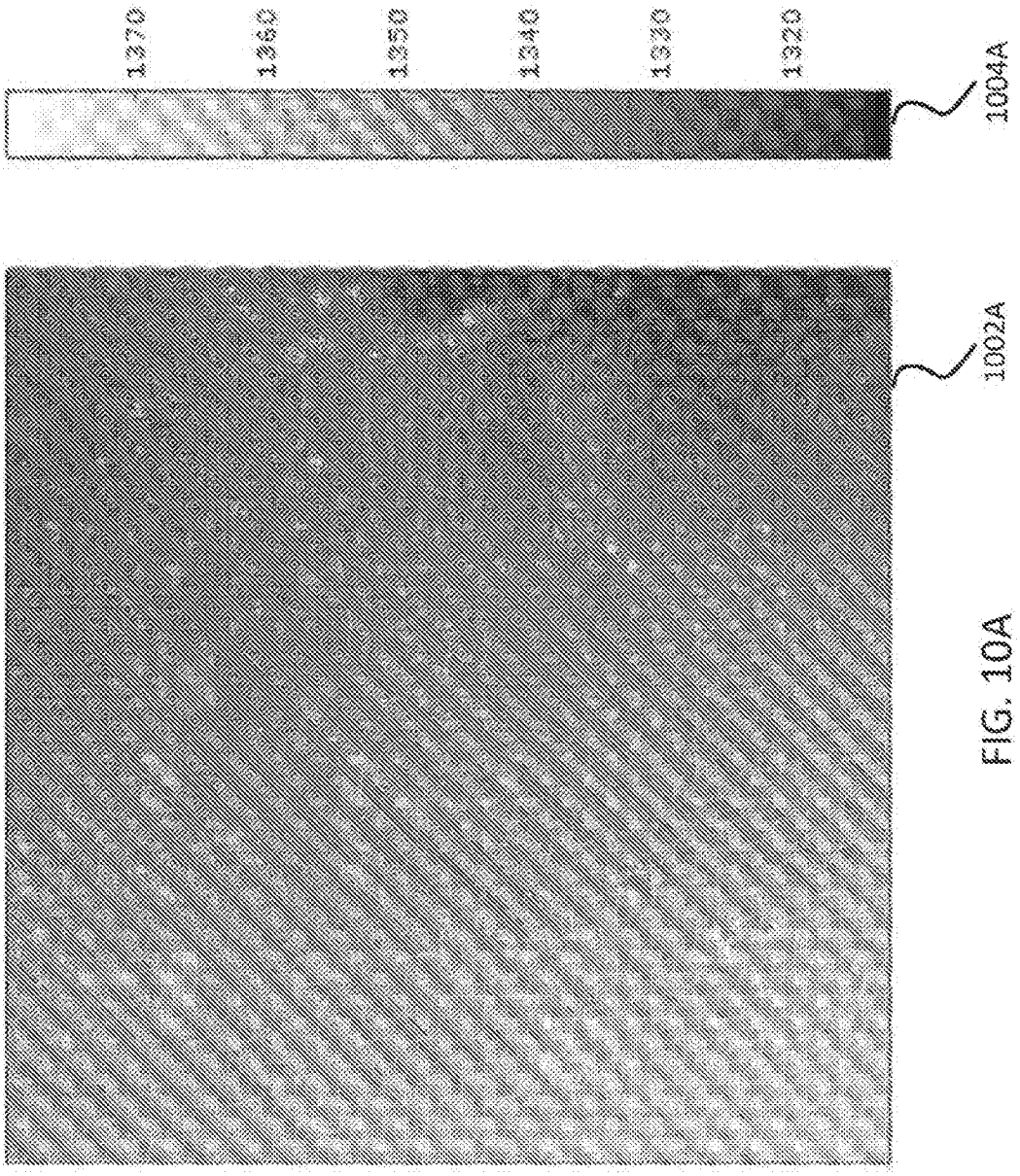


FIG. 10A

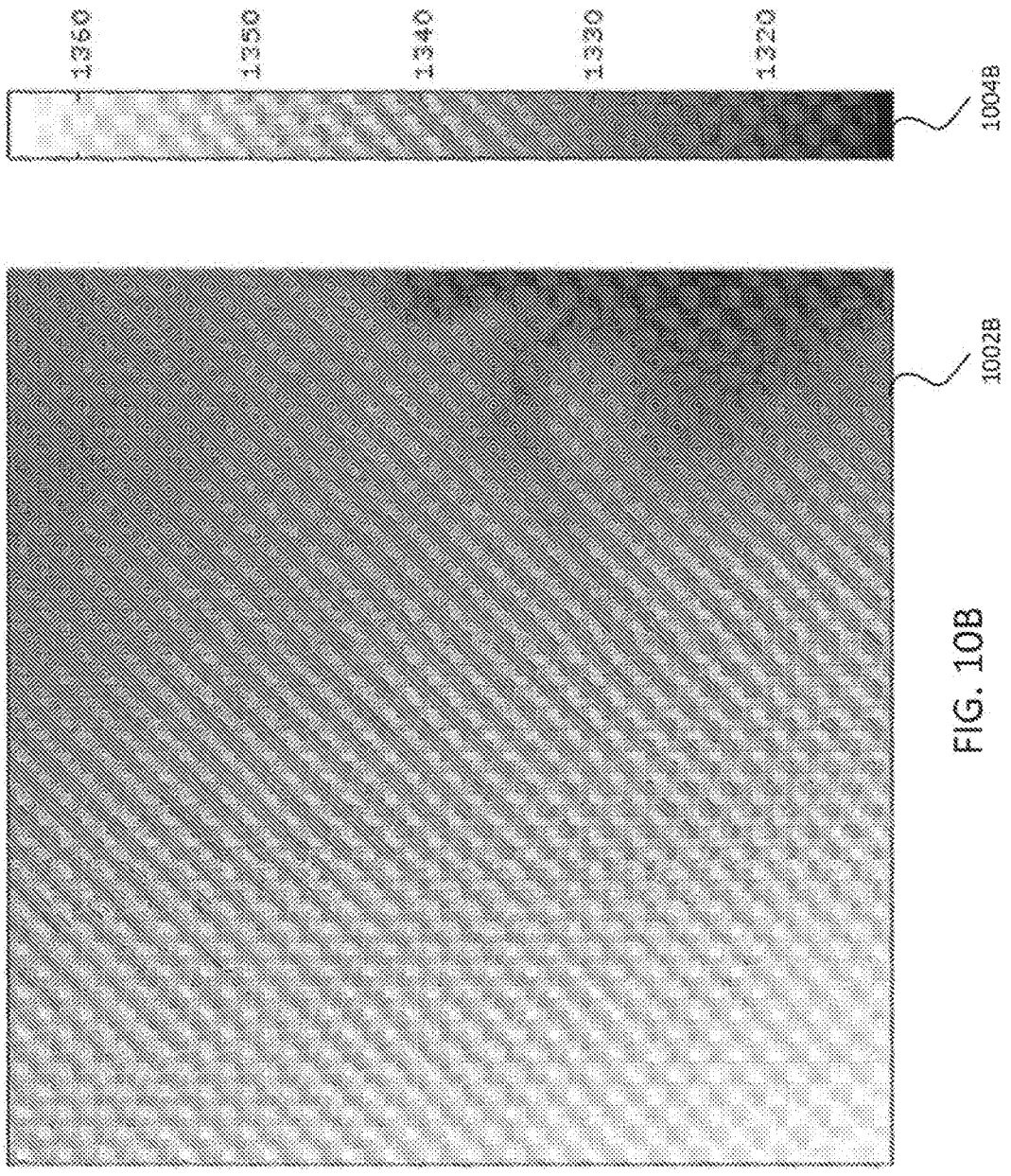


FIG. 10B

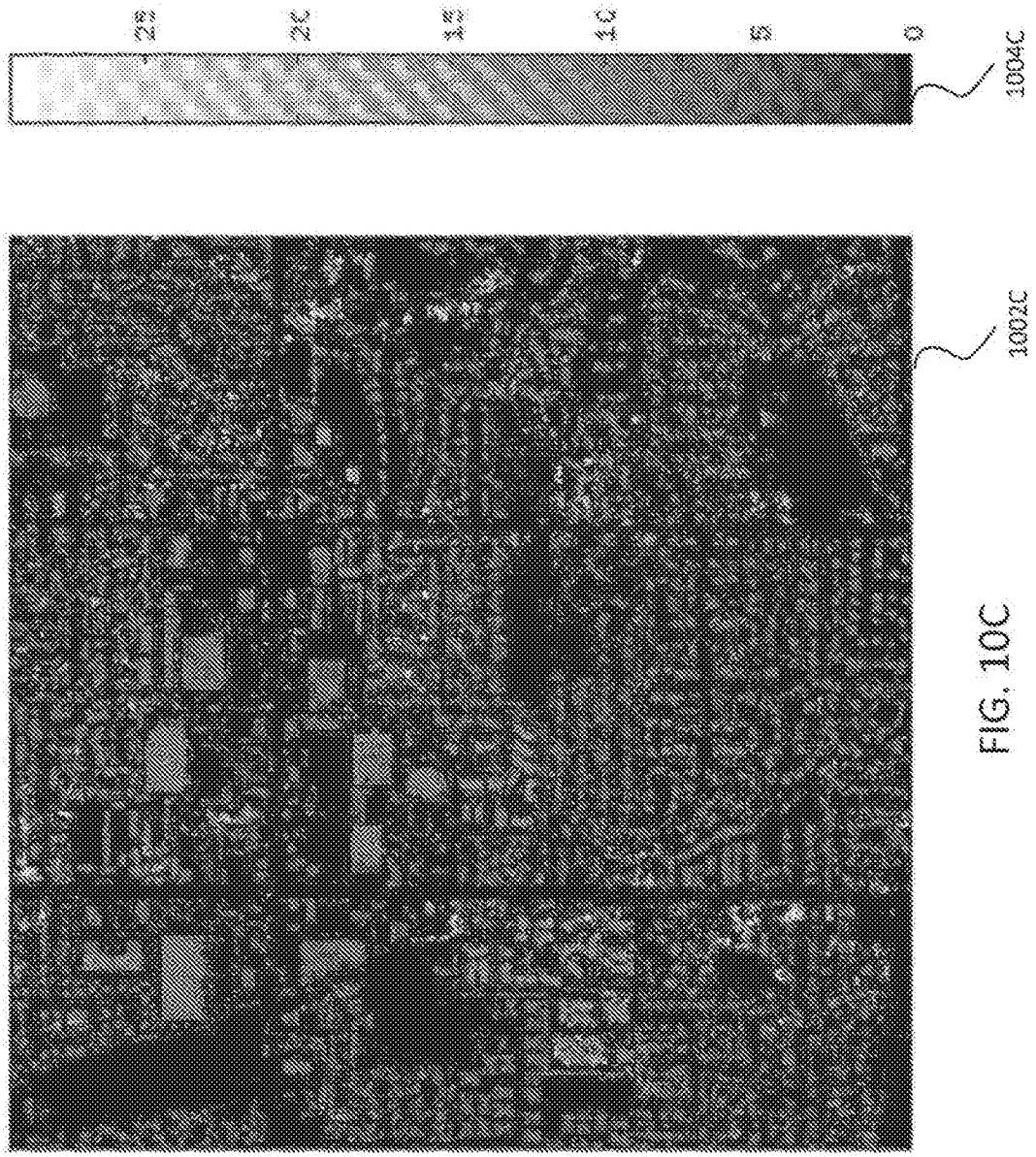


FIG. 10C

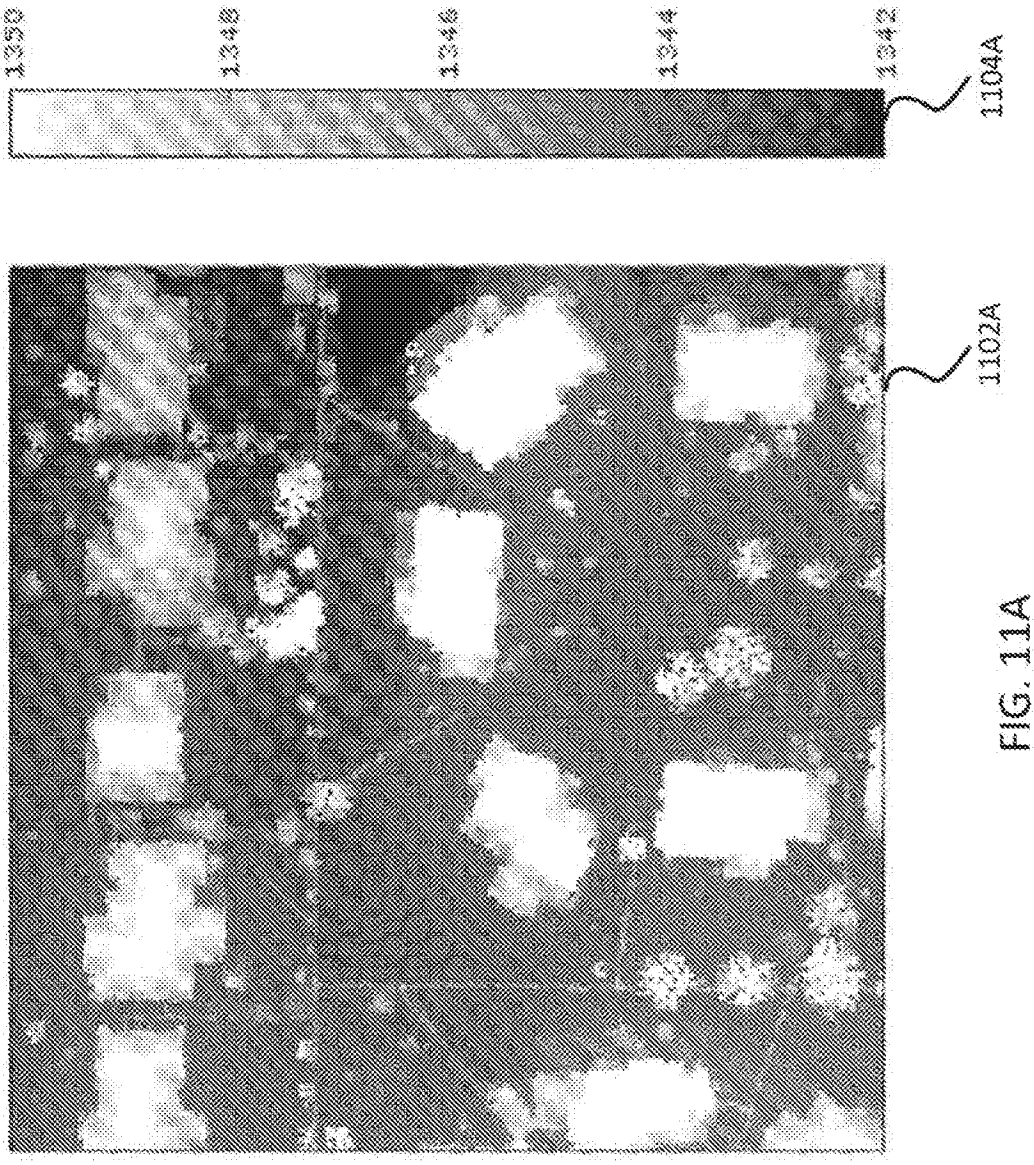


FIG. 11A

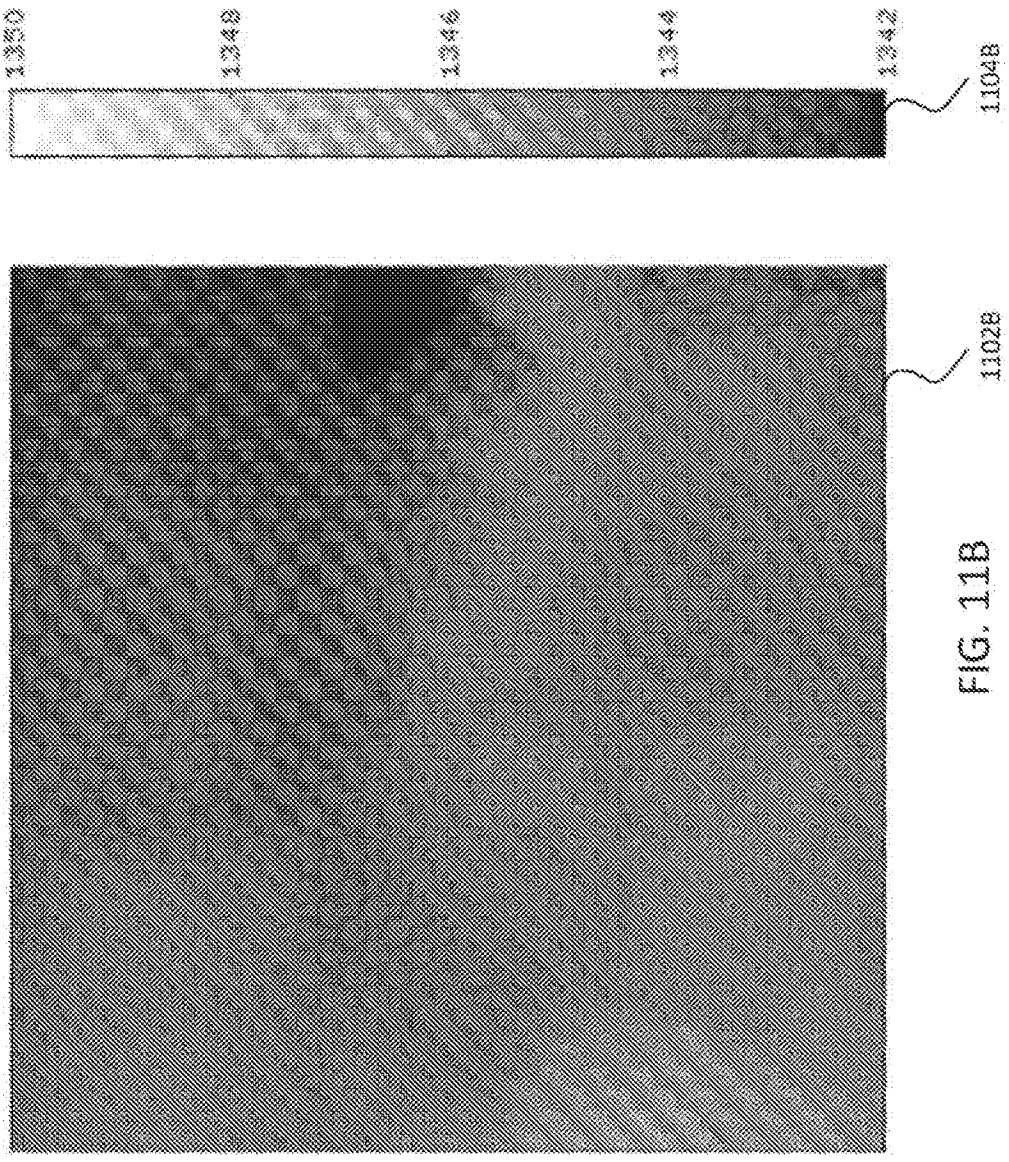


FIG. 11B

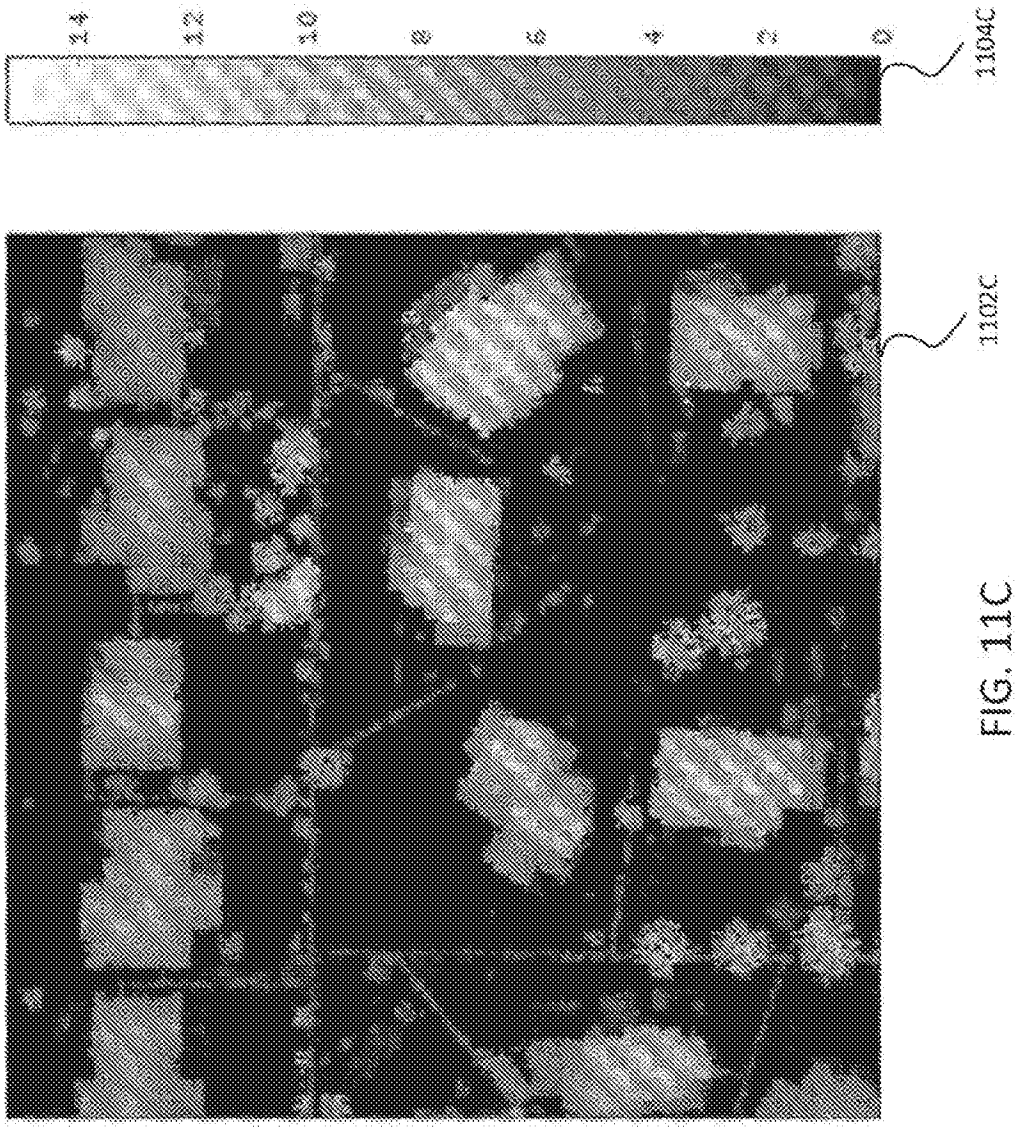


FIG. 11C

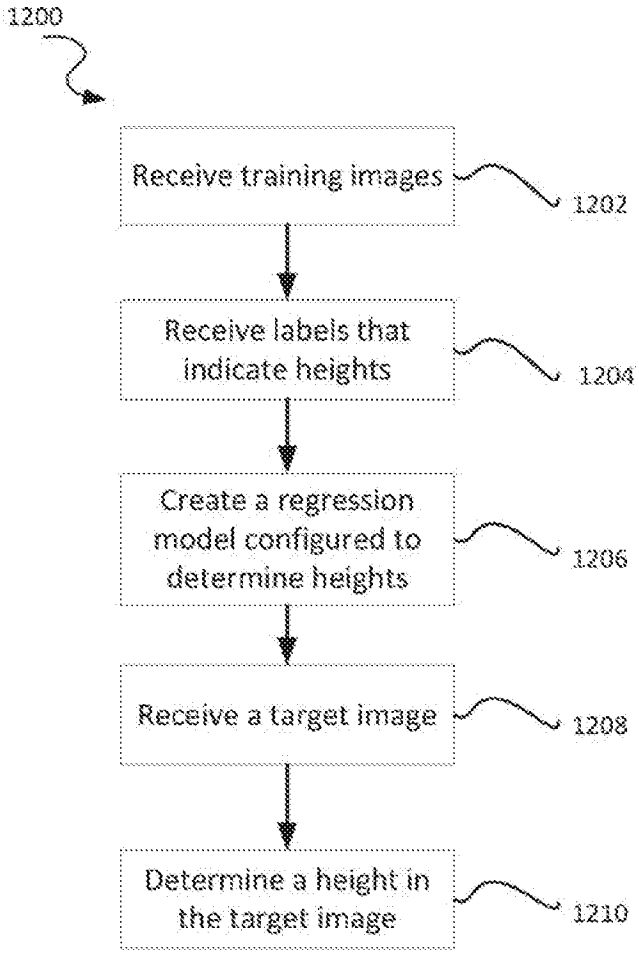

1200
Receive training images
1202
Receive labels that indicate heights
1204
Create a regression model configured to determine heights
1206
Receive a target image
1208
Determine a height in the target image
1210

FIG. 12

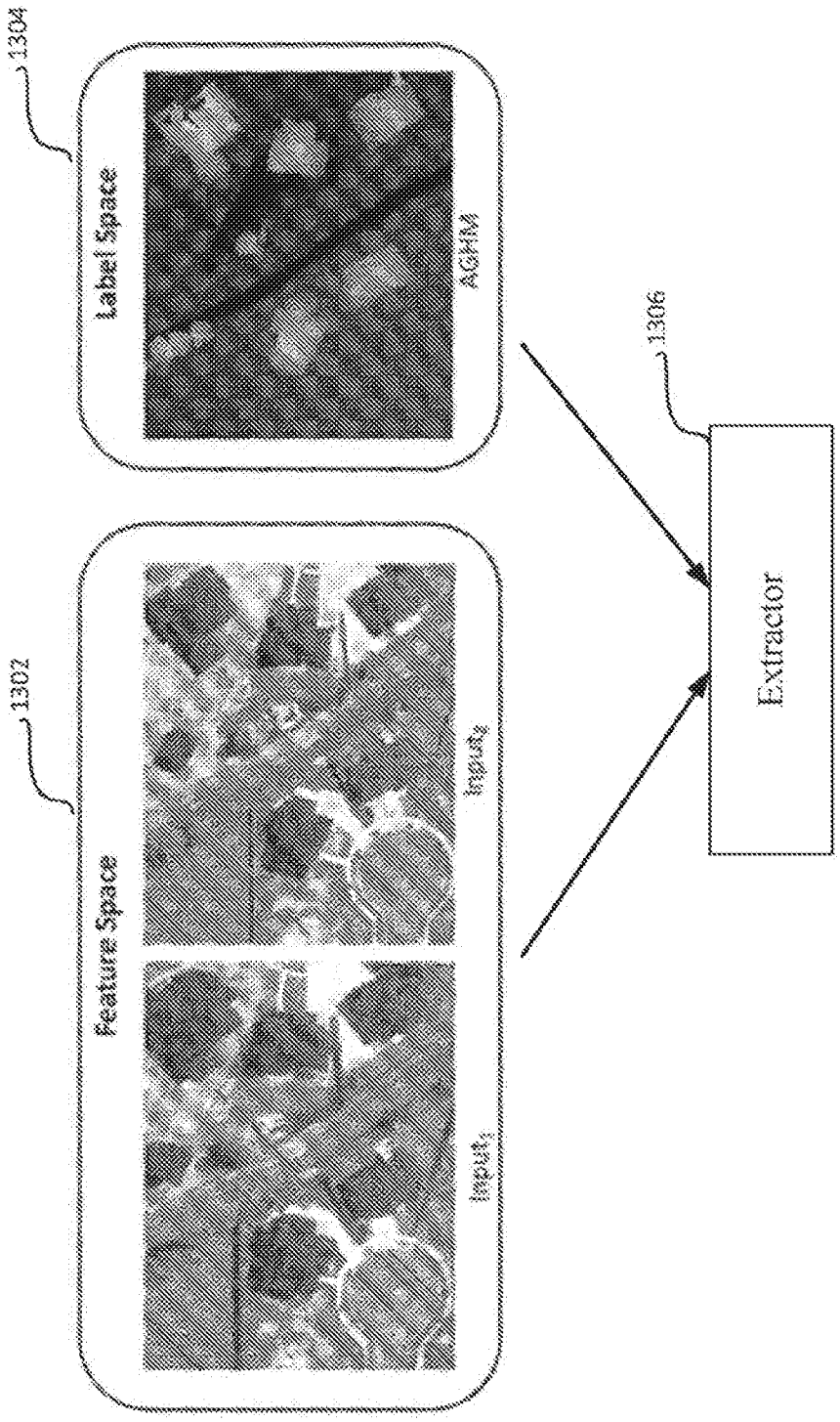

1304
Label Space
AGHM
1306
Extractor
1302
Feature Space
$input_2$
$input_1$

FIG. 13

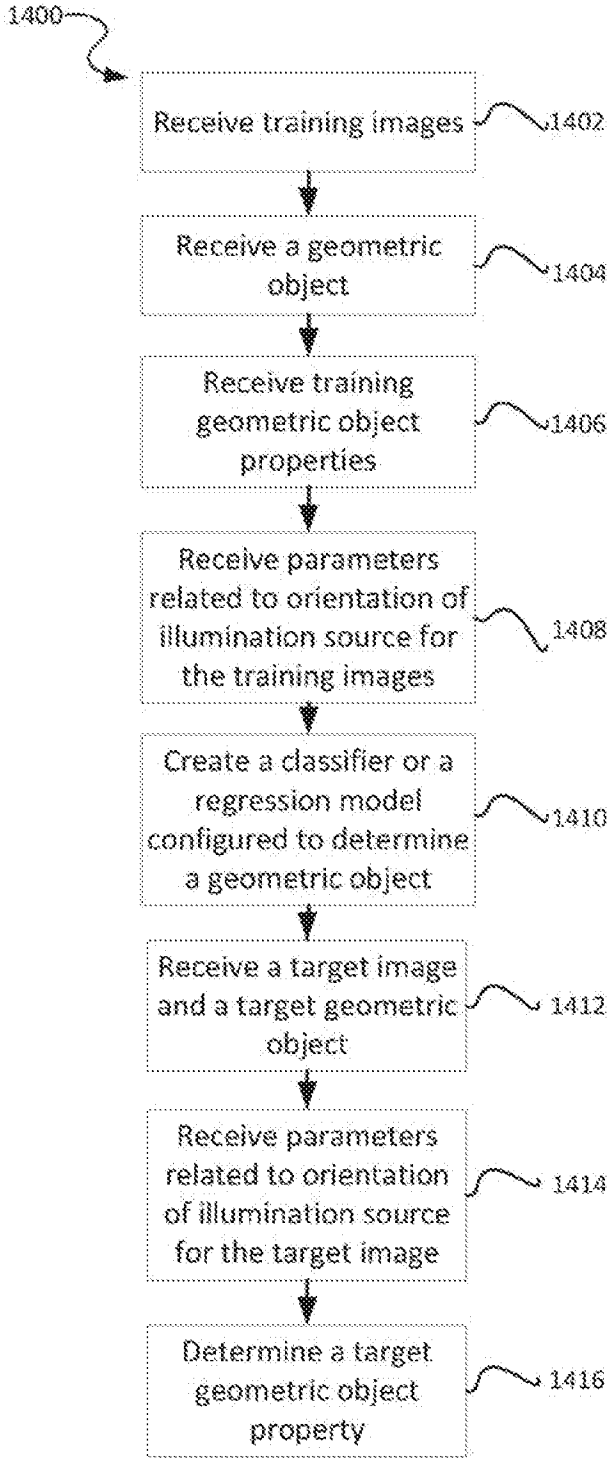

1400
Receive training images
1402
Receive a geometric object
1404
Receive training geometric object properties
1406
Receive parameters related to orientation of illumination source for the training images
1408
Create a classifier or a regression model configured to determine a geometric object
1410
Receive a target image and a target geometric object
1412
Receive parameters related to orientation of illumination source for the target image
1414
Determine a target geometric object property
1416

FIG. 14

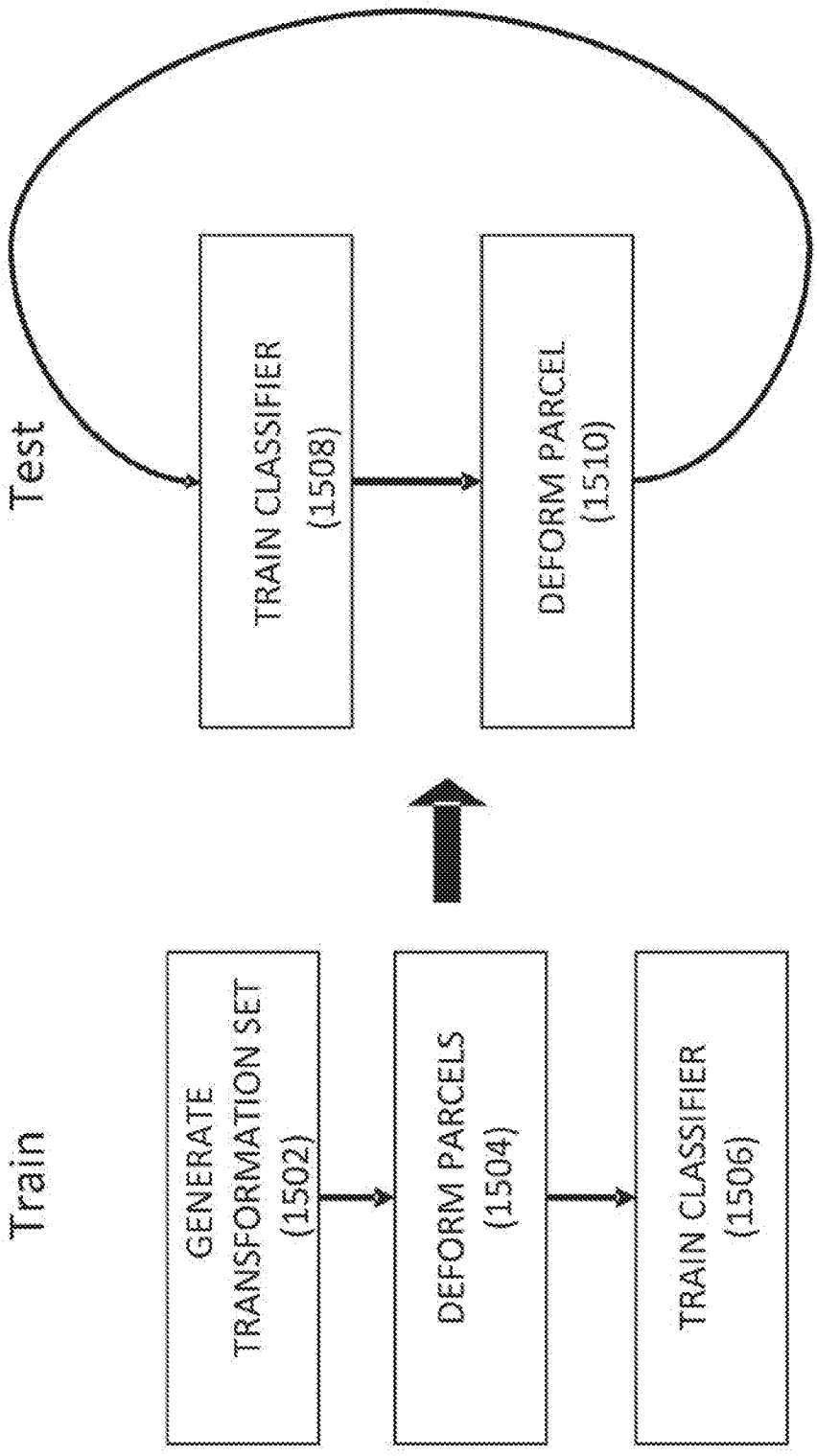

Test
TRAIN CLASSIFIER (1508)
DEFORM PARCEL (1510)
Train
GENERATE TRANSFORMATION SET (1502)
DEFORM PARCELS (1504)
TRAIN CLASSIFIER (1506)

FIG. 15

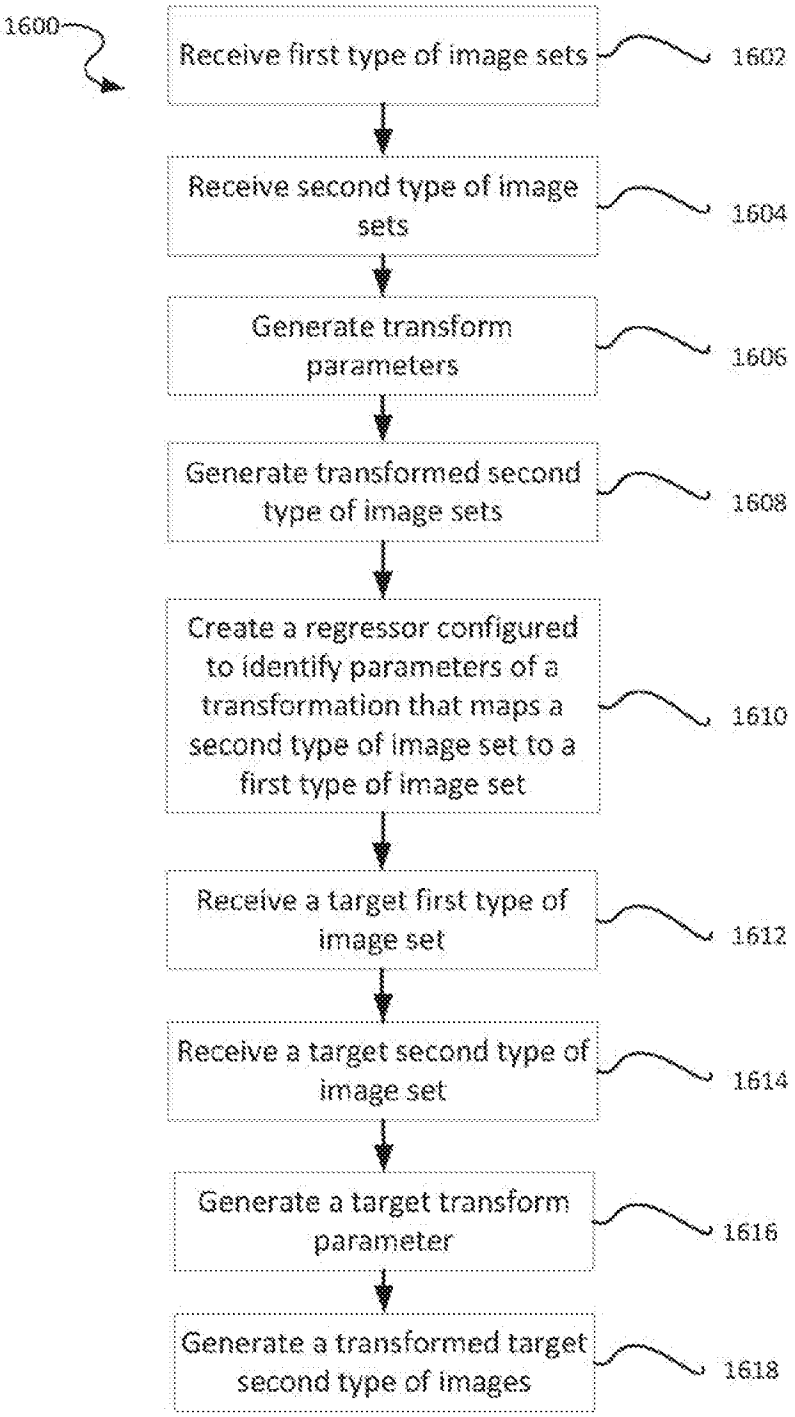

1600
Receive first type of image sets
1602
Receive second type of image sets
1604
Generate transform parameters
1606
Generate transformed second type of image sets
1608
Create a regressor configured to identify parameters of a transformation that maps a second type of image set to a first type of image set
1610
Receive a target first type of image set
1612
Receive a target second type of image set
1614
Generate a target transform parameter
1616
Generate a transformed target second type of images
1618

FIG. 16

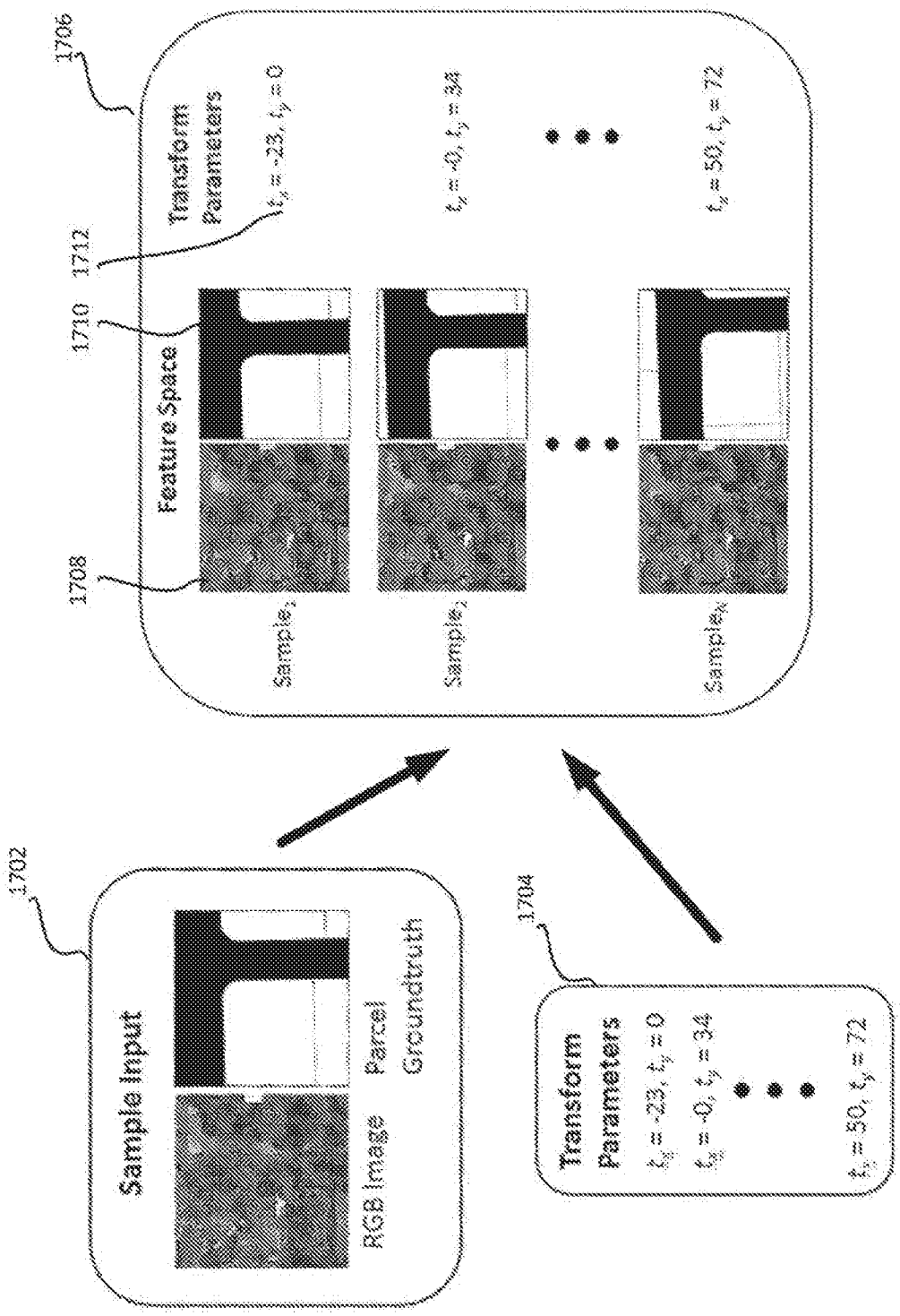

1706
1712
Transform Parameters
$t_x = -23, t_y = 0$
$t_x = -0, t_y = 34$
$t_x = 50, t_y = 72$
1710
Feature Space
1708
$Sample_1$
$Sample_2$
$Sample_N$
1702
Sample Input
RGB Image
Parcel Groundtruth
1704
Transform Parameters
$t_x = -23, t_y = 0$
$t_x = -0, t_y = 34$
$t_x = 50, t_y = 72$

FIG. 17

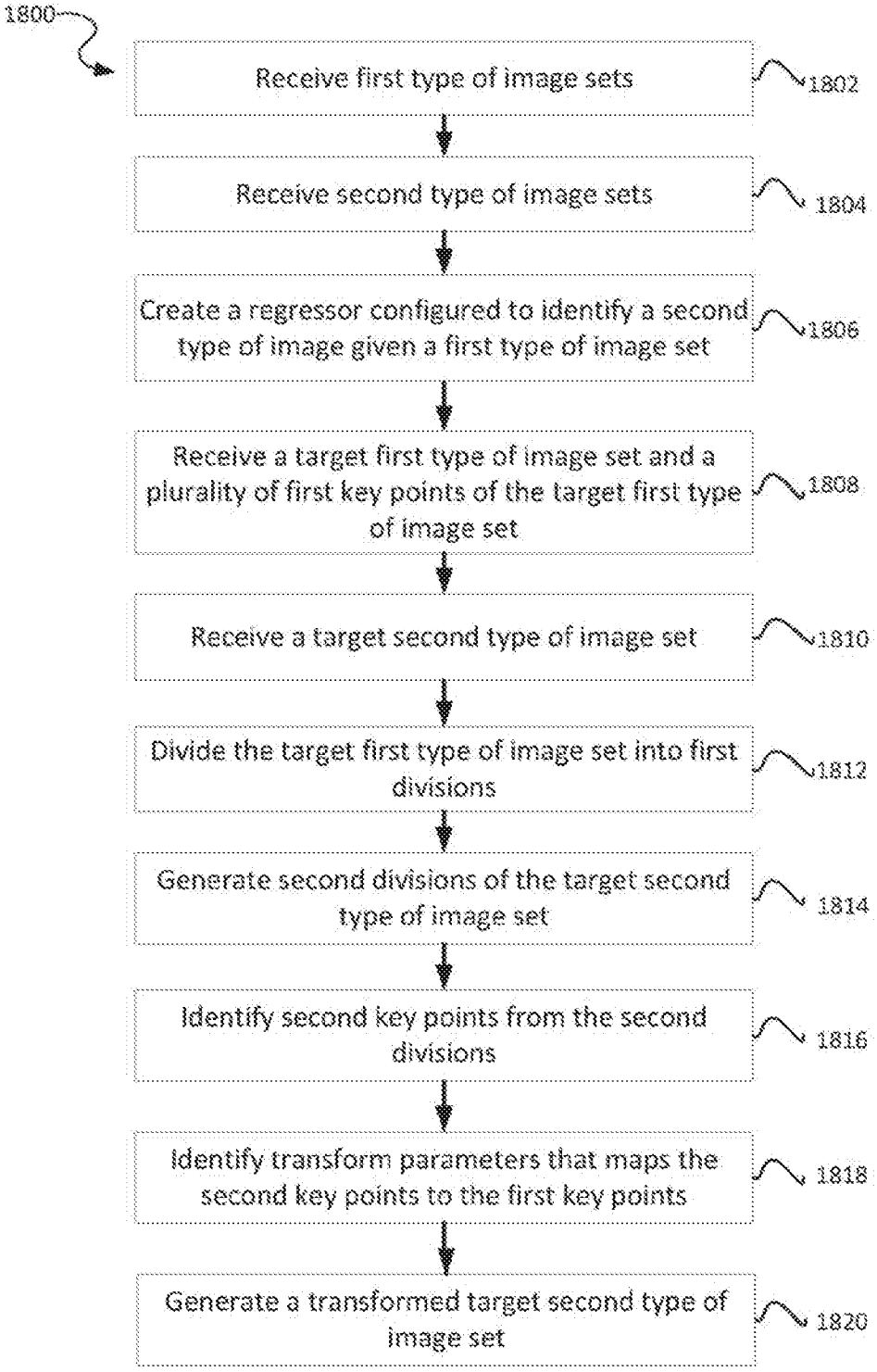

1800
Receive first type of image sets
1802
Receive second type of image sets
1804
Create a regressor configured to identify a second type of image given a first type of image set
1806
Receive a target first type of image set and a plurality of first key points of the target first type of image set
1808
Receive a target second type of image set
1810
Divide the target first type of image set into first divisions
1812
Generate second divisions of the target second type of image set
1814
Identify second key points from the second divisions
1816
Identify transform parameters that maps the second key points to the first key points
1818
Generate a transformed target second type of image set
1820

FIG. 18

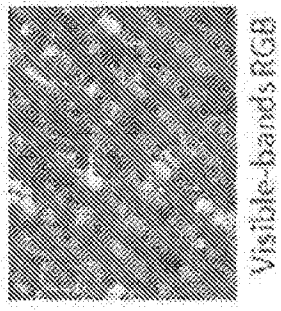


FIG. 19A

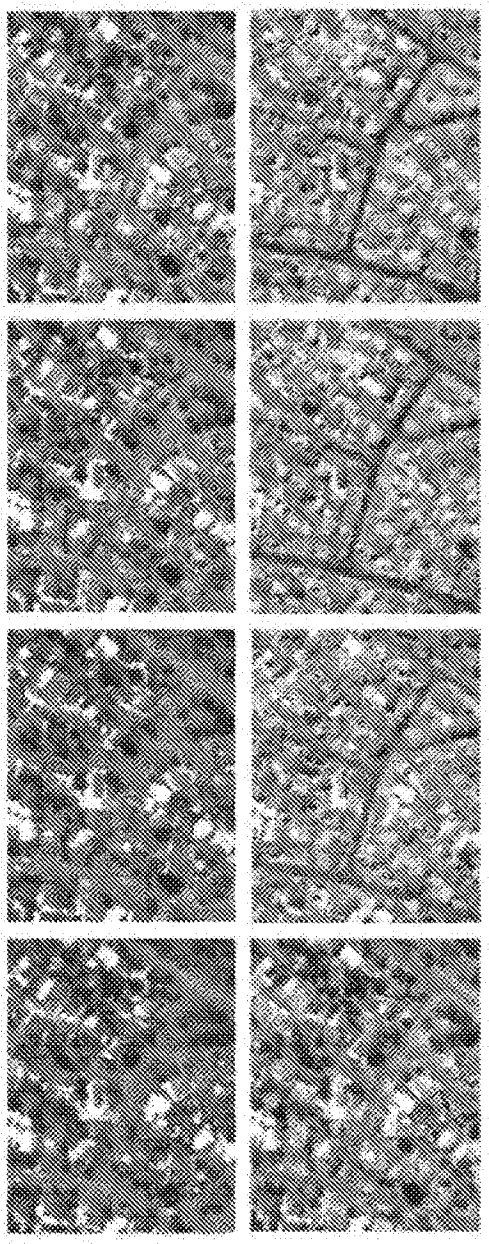


FIG. 19B

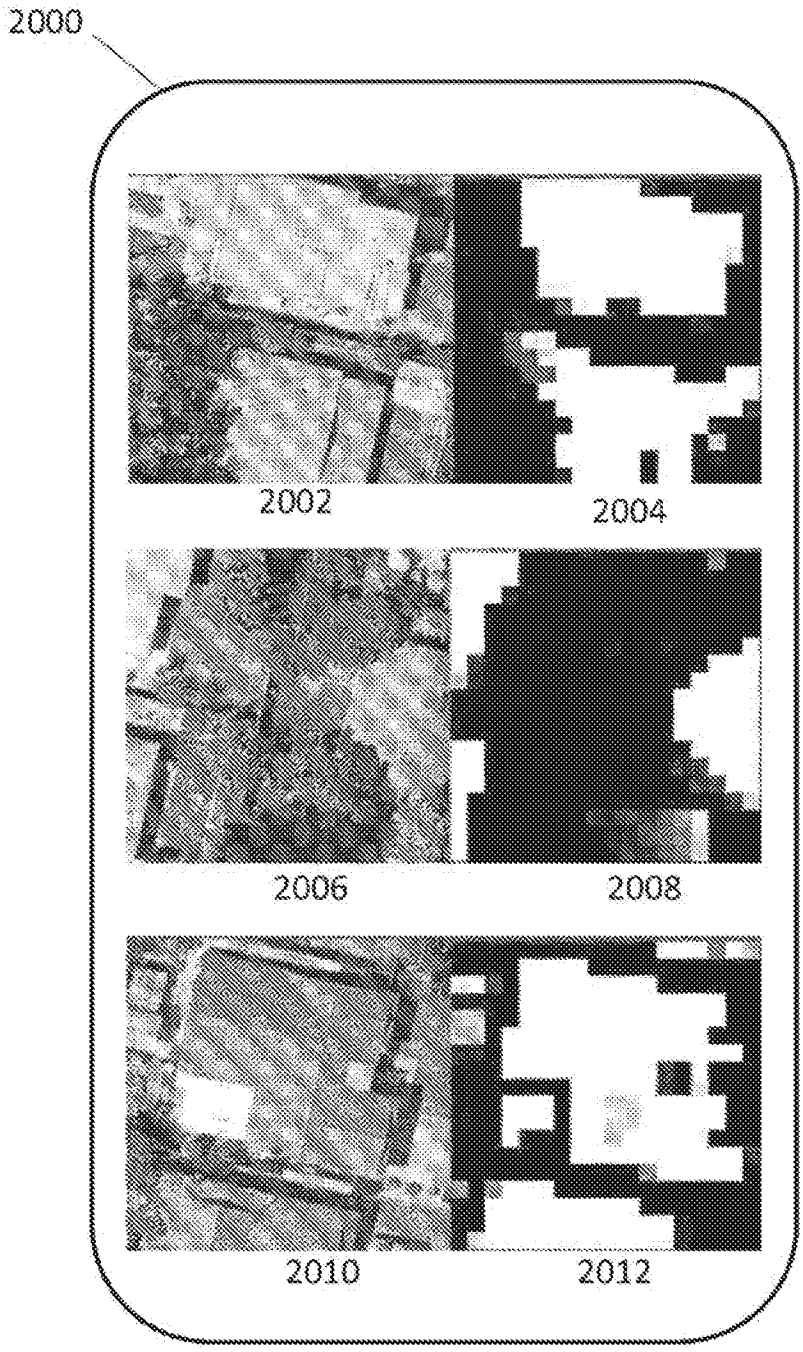


FIG. 20

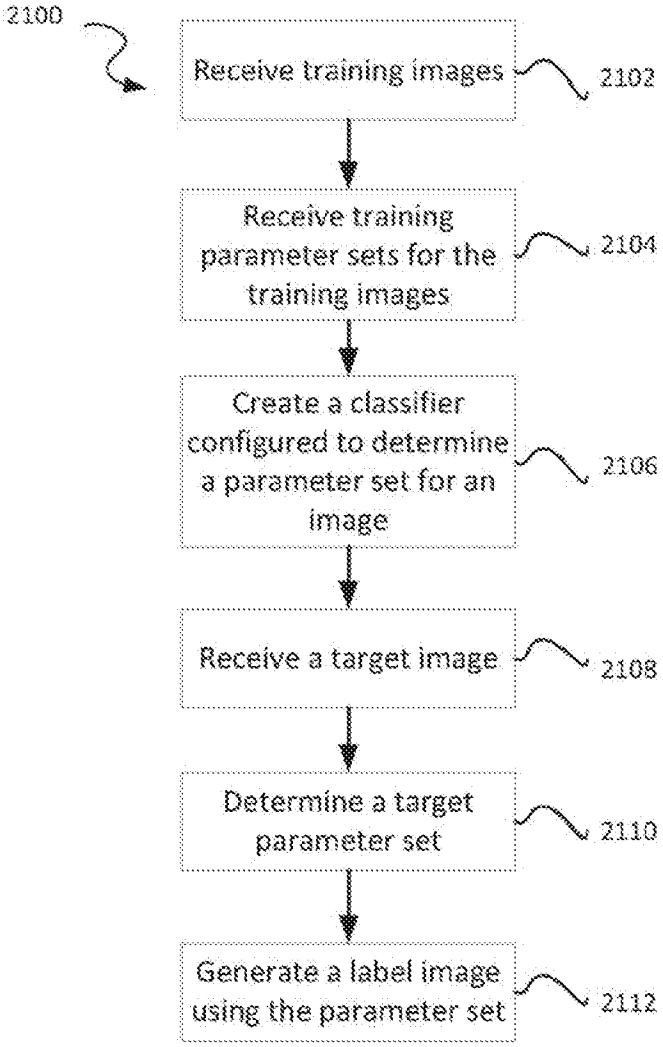

2100
Receive training images
2102
Receive training parameter sets for the training images
2104
Create a classifier configured to determine a parameter set for an image
2106
Receive a target image
2108
Determine a target parameter set
2110
Generate a label image using the parameter set
2112

FIG. 21

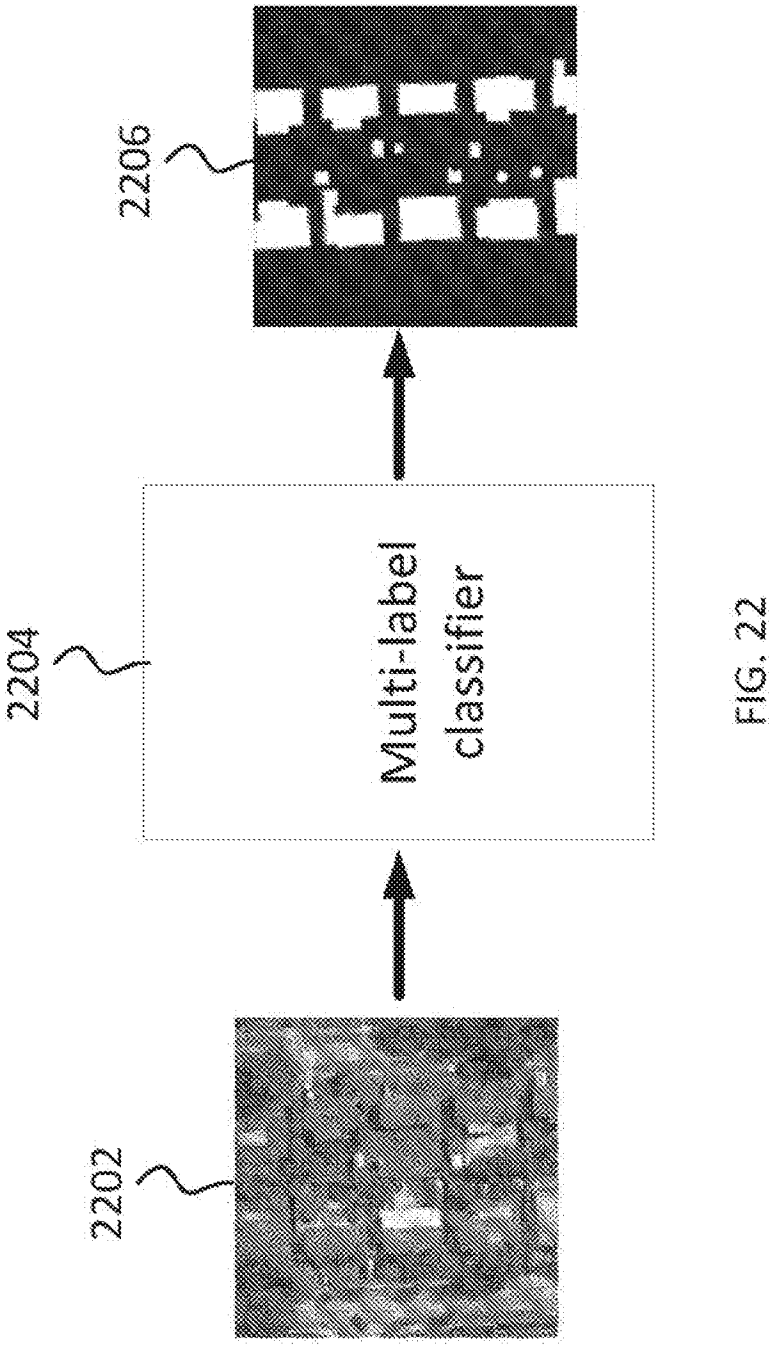


FIG. 22

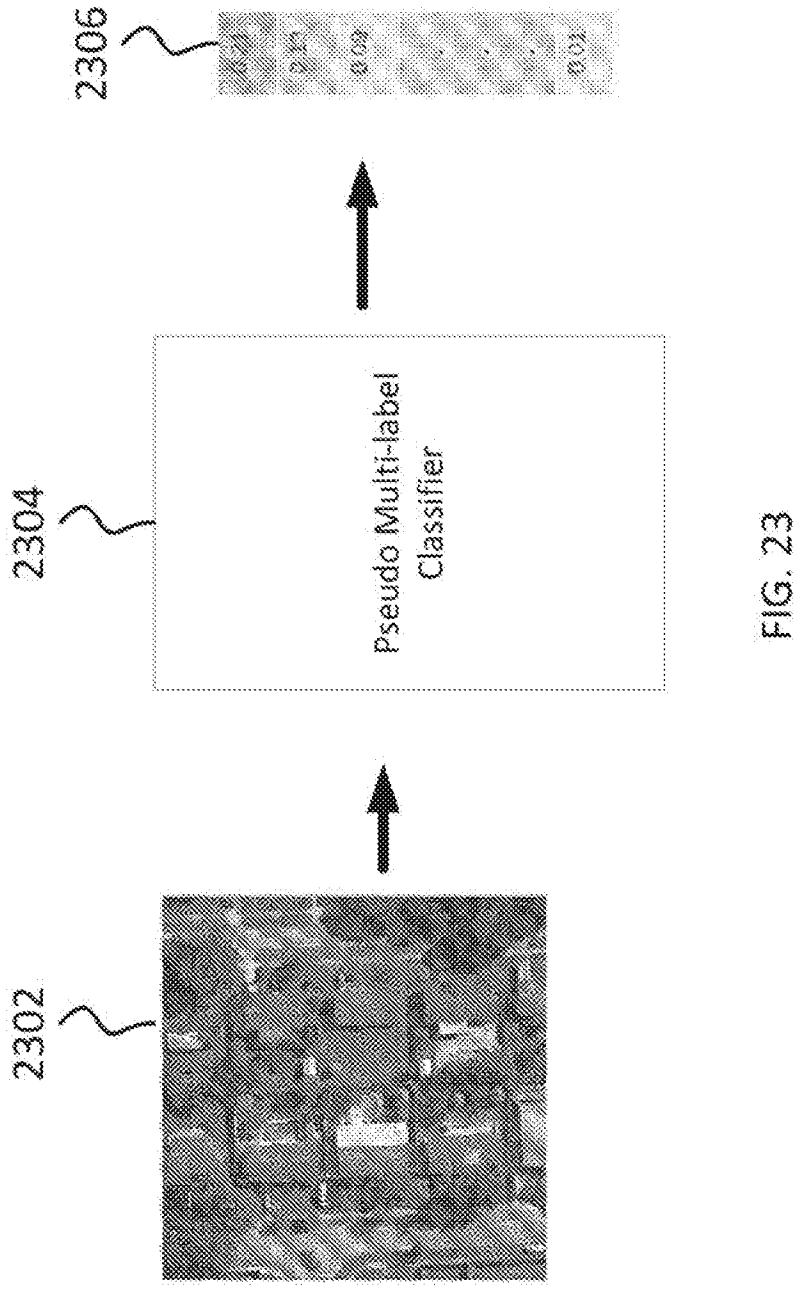


FIG. 23

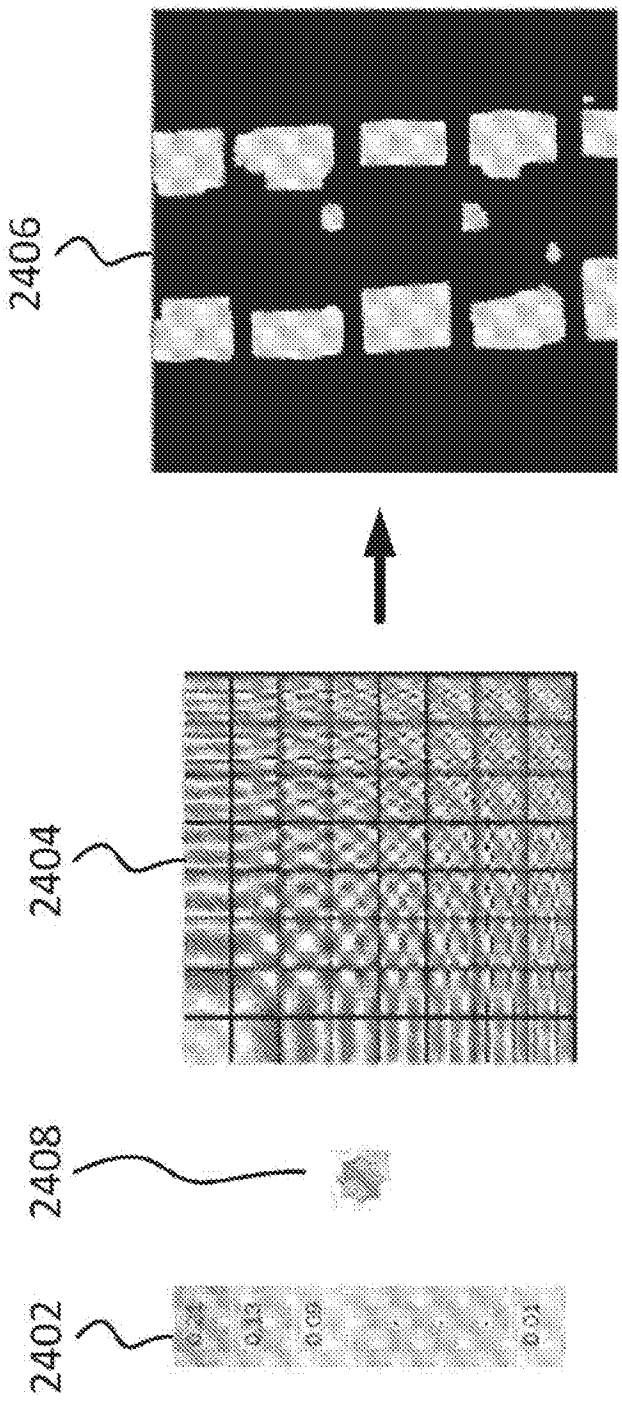


FIG. 24

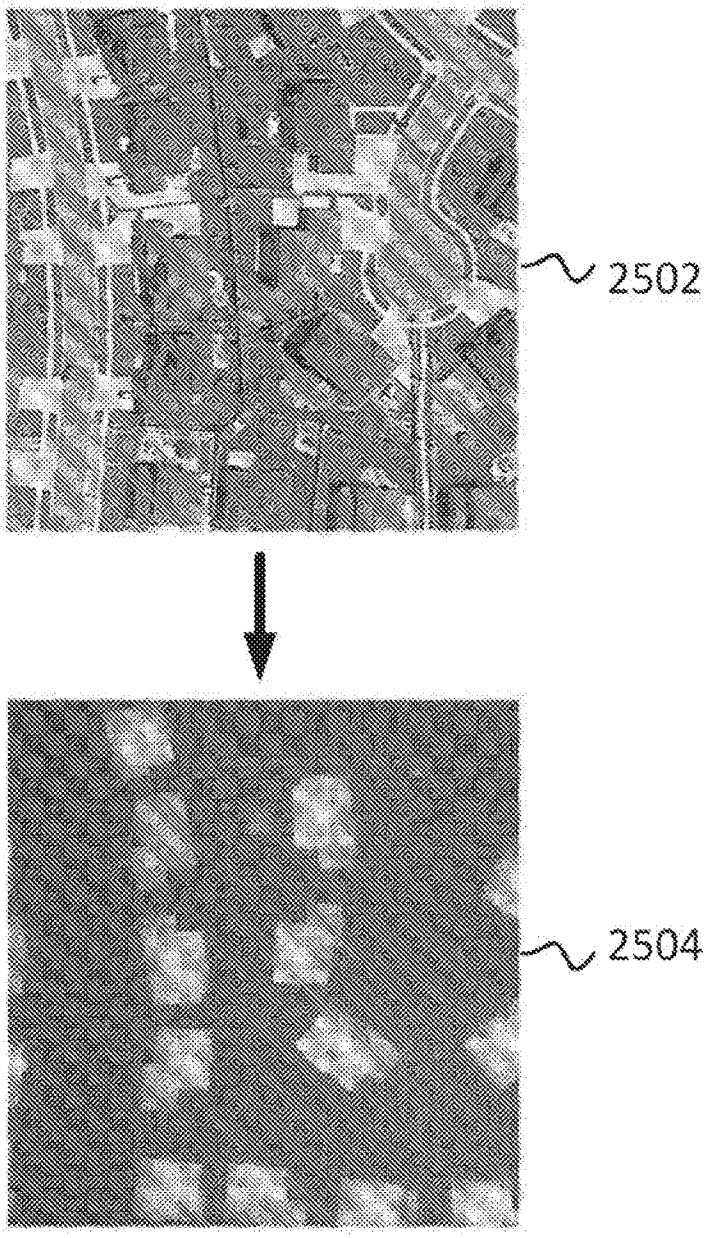


FIG. 25

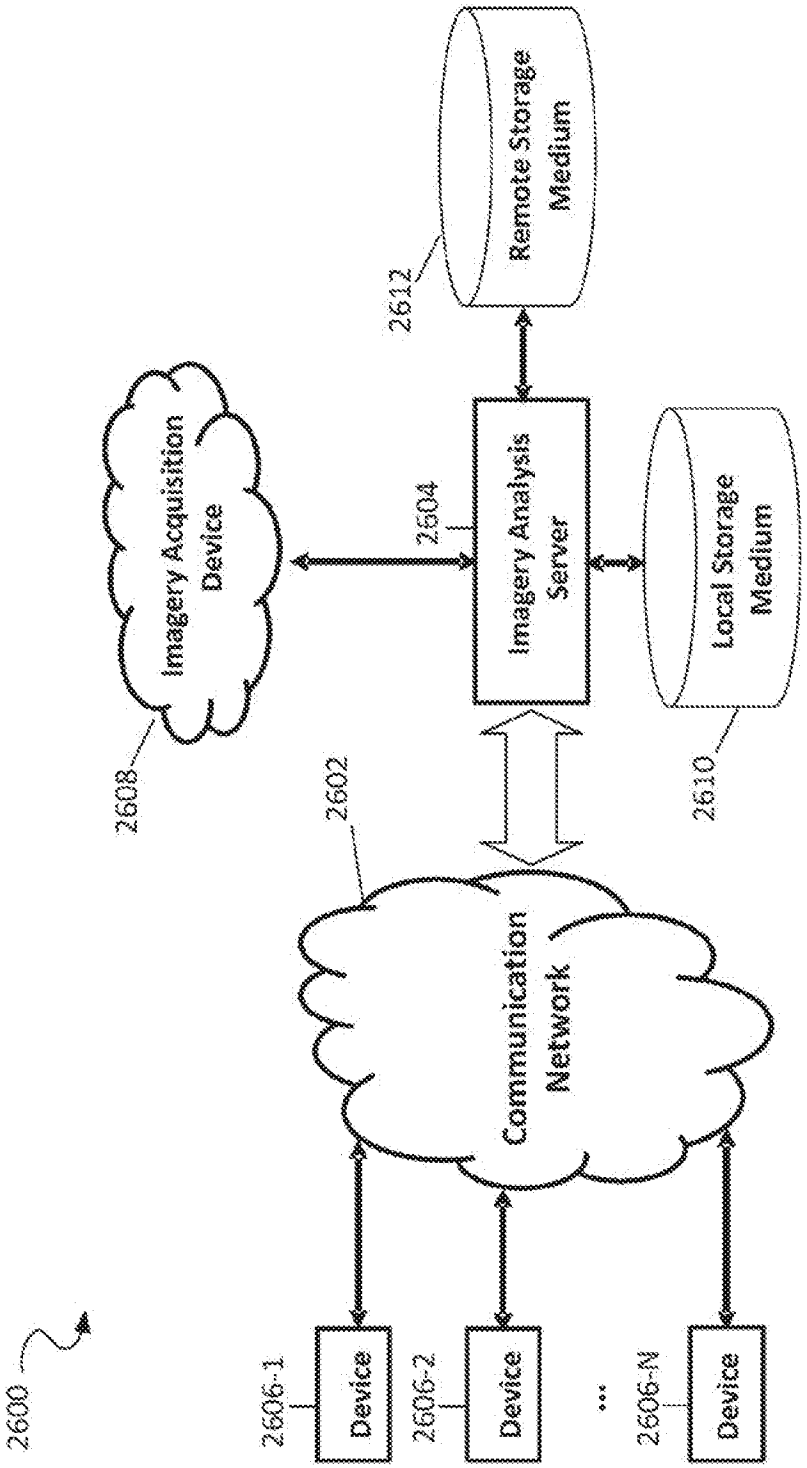

2600
2606-1
Device
2606-2
Device
...
2606-N
Device
2602
Communication Network
2608
Imagery Acquisition Device
2604
Imagery Analysis Server
2610
Local Storage Medium
2612
Remote Storage Medium

FIG. 26

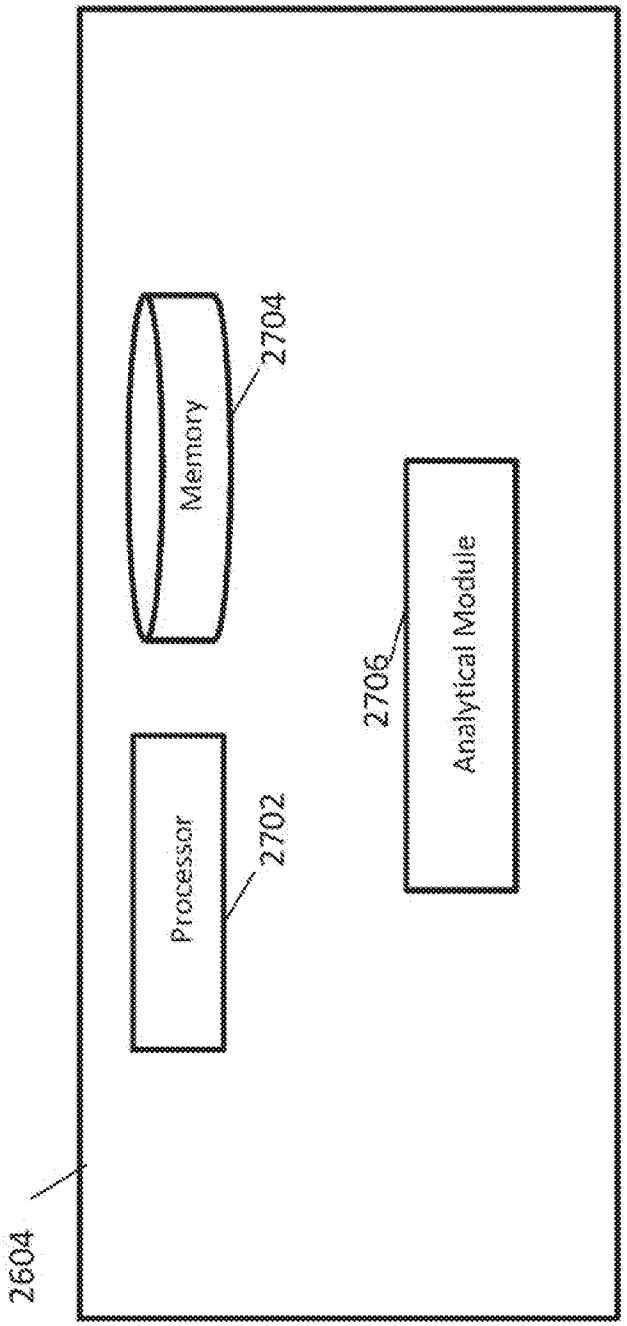


FIG. 27

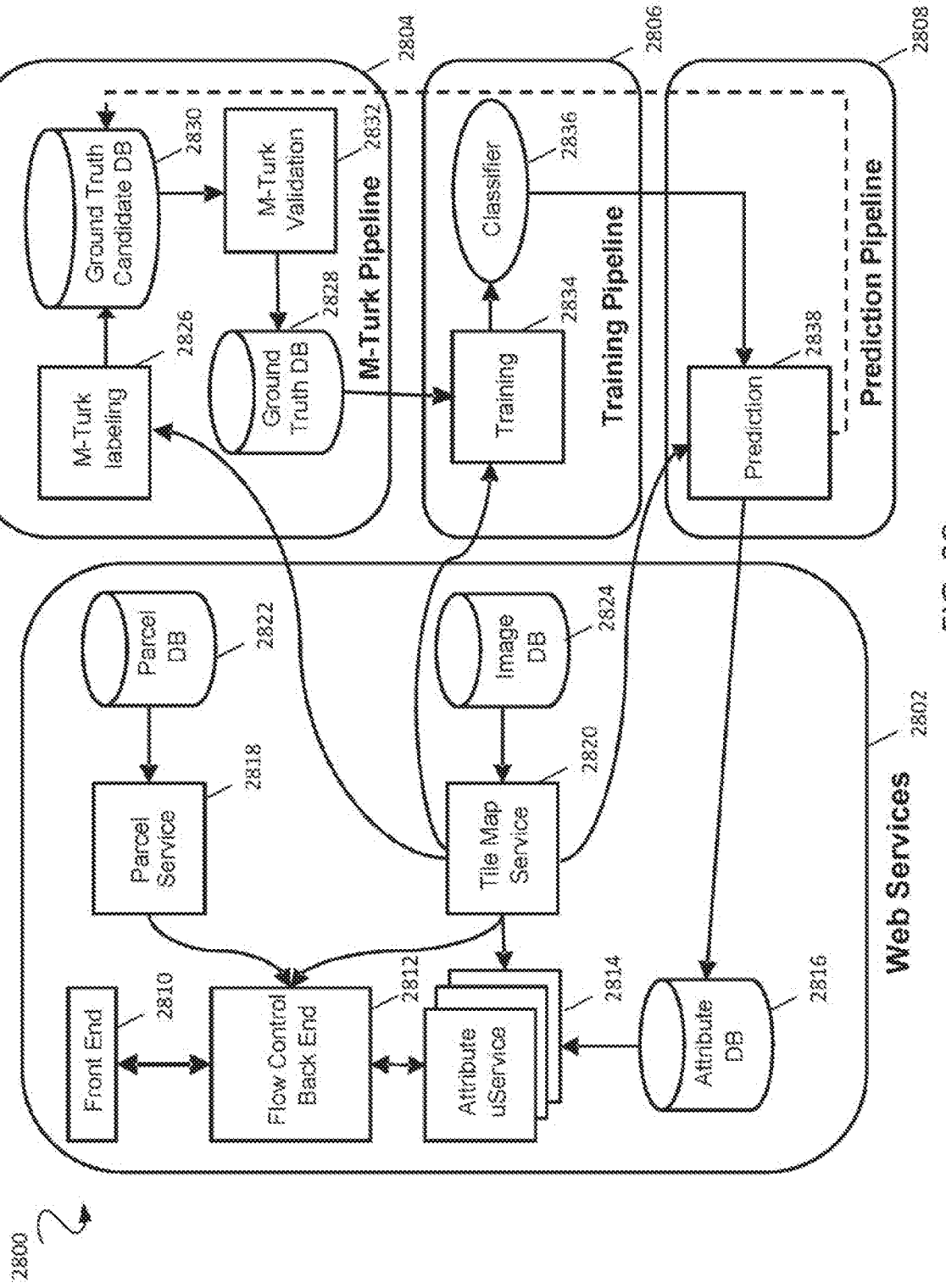

2800
2
Web Services
2802
Front End
2810
Flow Control Back End
2812
Attribute uService
2814
Attribute DB
2816
Parcel Service
2818
Tile Map Service
2820
Parcel DB
2822
Image DB
2824
M-Turk labeling
2826
Ground Truth DB
2828
Ground Truth Candidate DB
2830
M-Turk Validation
2832
M-Turk Pipeline
2804
Training
2834
Classifier
2836
Training Pipeline
2806
Prediction
2838
Prediction Pipeline
2808

FIG. 28

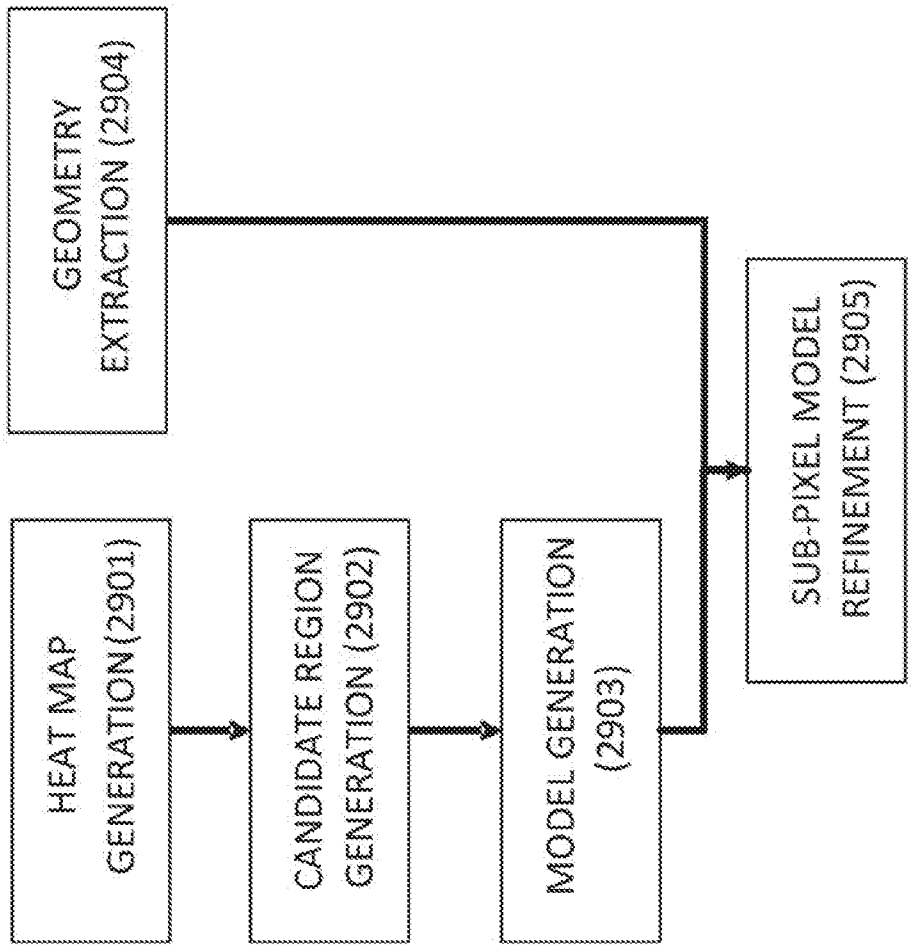

GEOMETRY EXTRACTION (2904)
HEAT MAP GENERATION (2901)
CANDIDATE REGION GENERATION (2902)
MODEL GENERATION (2903)
SUB-PIXEL MODEL REFINEMENT (2905)

FIG. 29

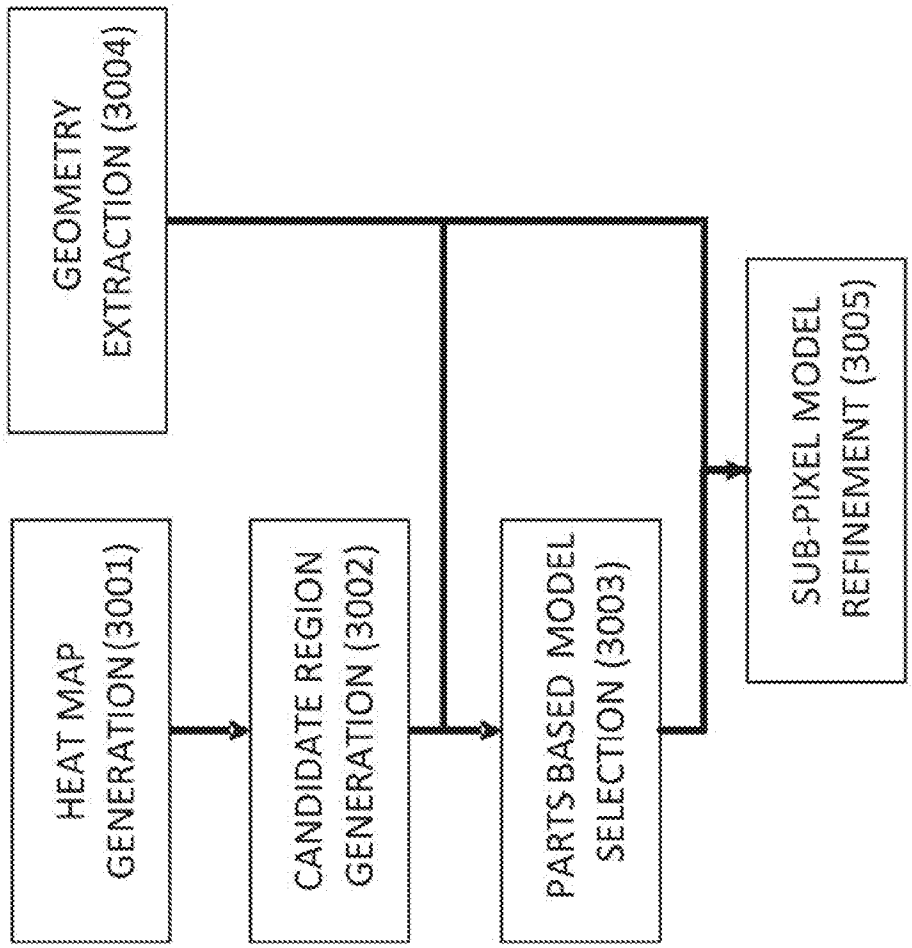

GEOMETRY EXTRACTION (3004)
HEAT MAP GENERATION(3001)
CANDIDATE REGION GENERATION (3002)
PARTS BASED MODEL SELECTION (3003)
SUB-PIXEL MODEL REFINEMENT (3005)

FIG. 30

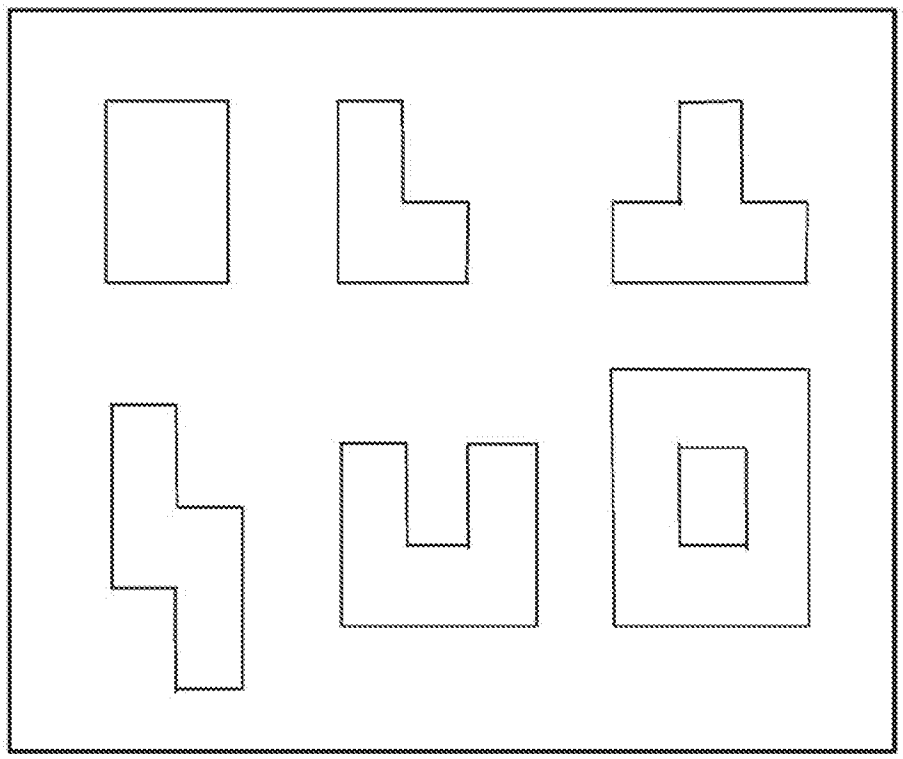


FIG. 31

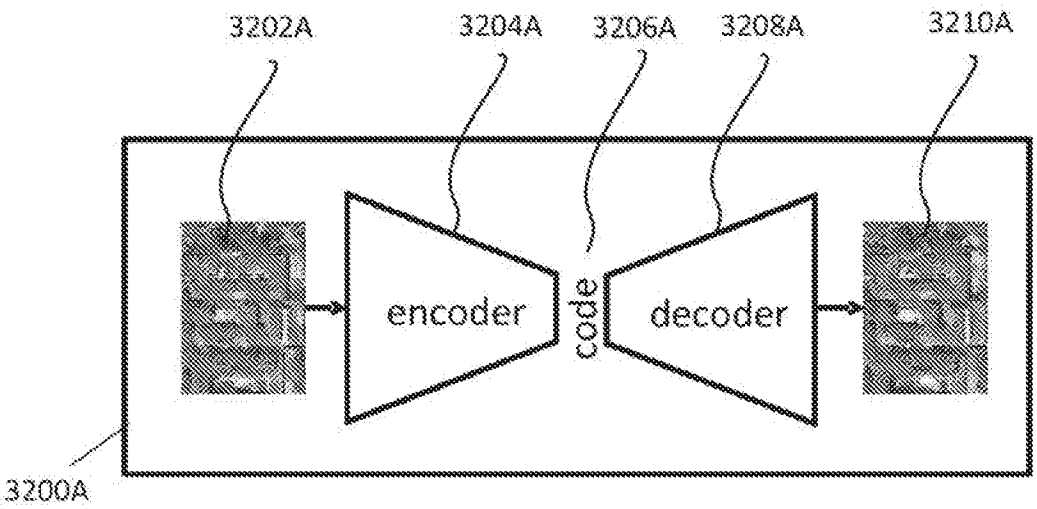


FIG. 32A

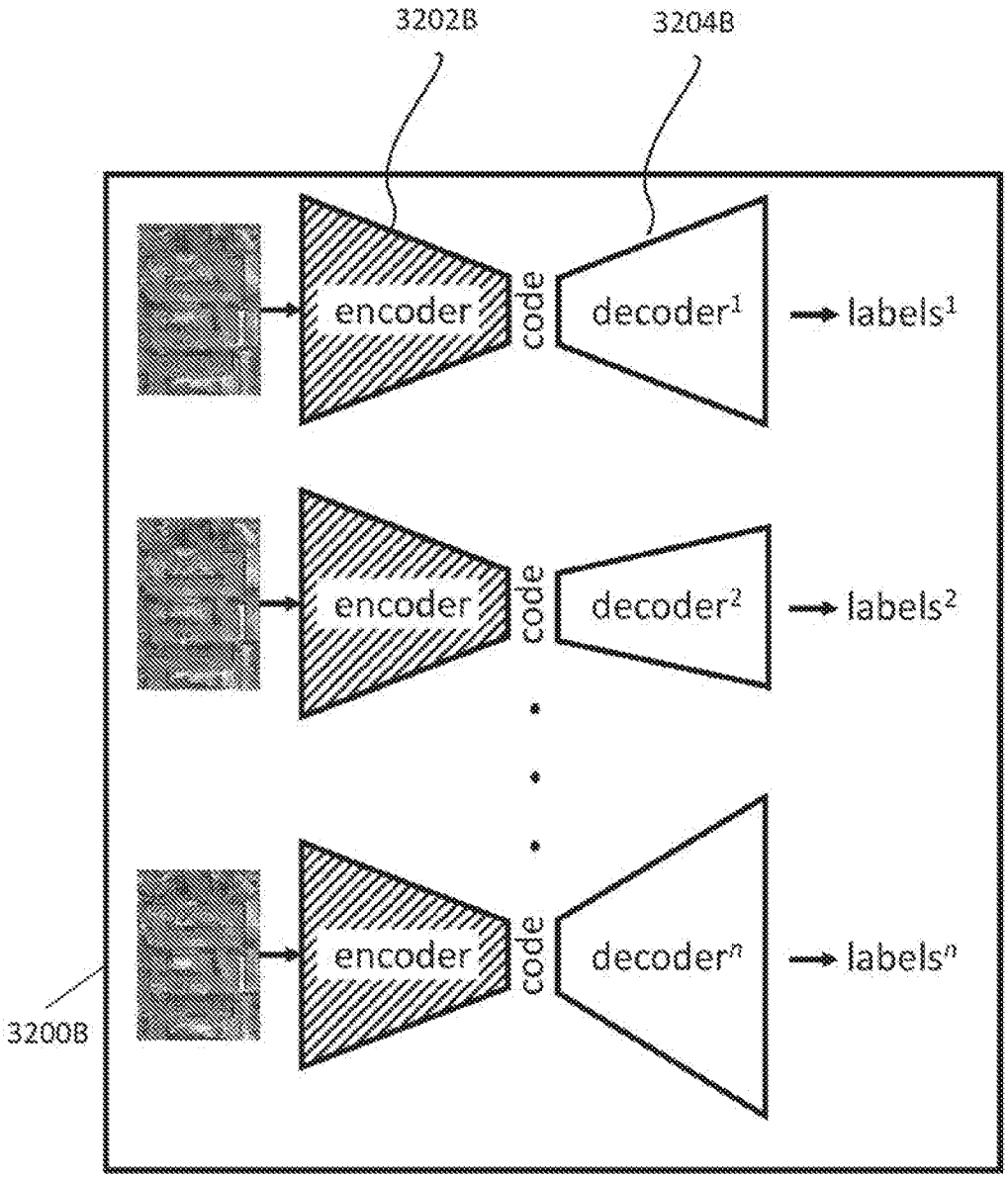

3202B
3204B
encoder
code
decoder¹
labels¹
encoder
code
decoder²
labels²
encoder
code
decoderⁿ
labelsⁿ
3200B

FIG. 32B

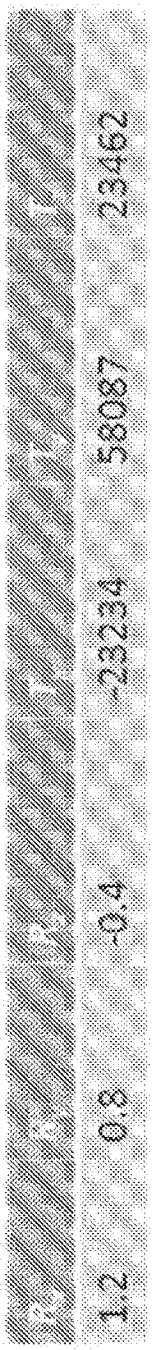


FIG. 33A

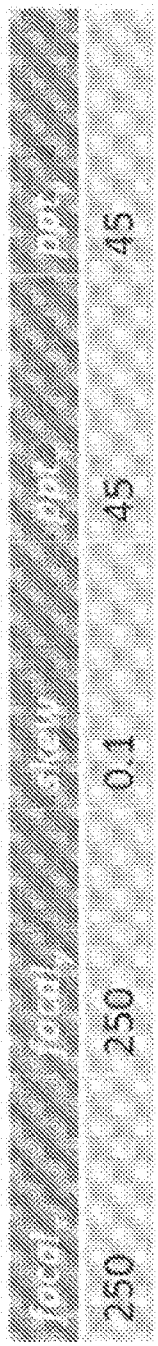


FIG. 33B

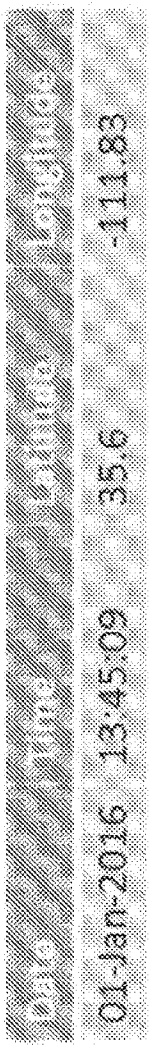

Date
Time
Temperature
Longitude
01-Jan-2016
13:45:09
35.6
-111.83

FIG. 33C

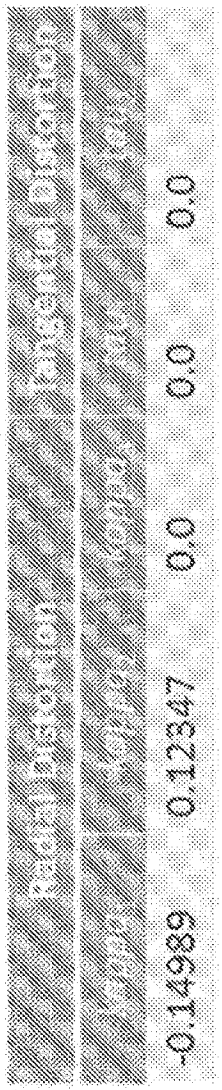


FIG. 33D

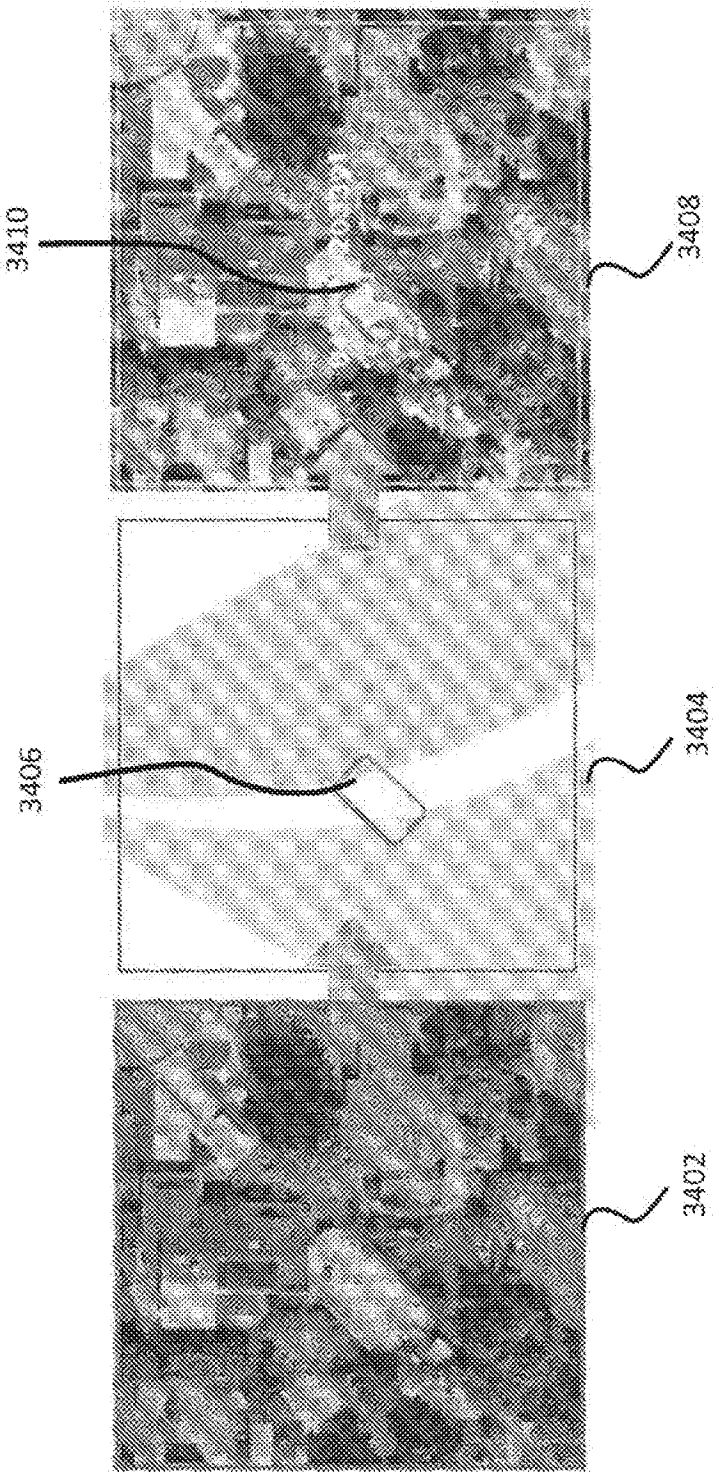


FIG. 34

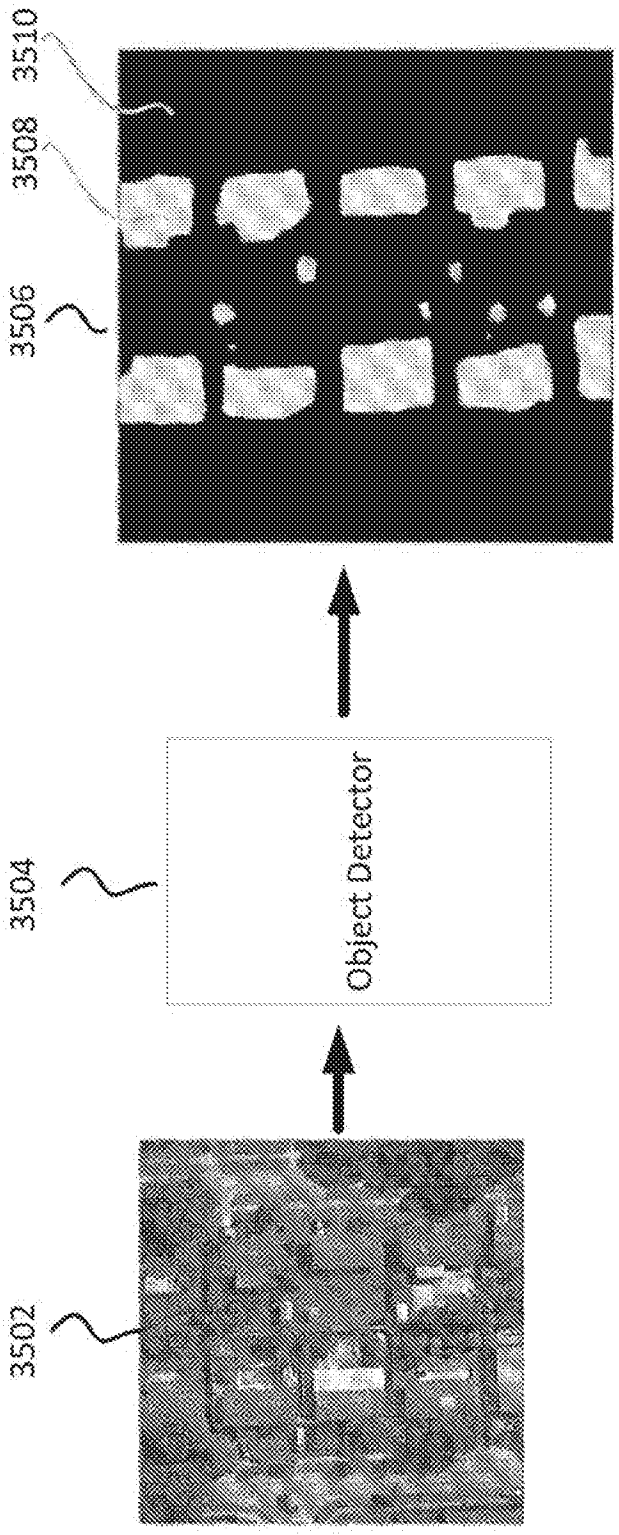


FIG. 35

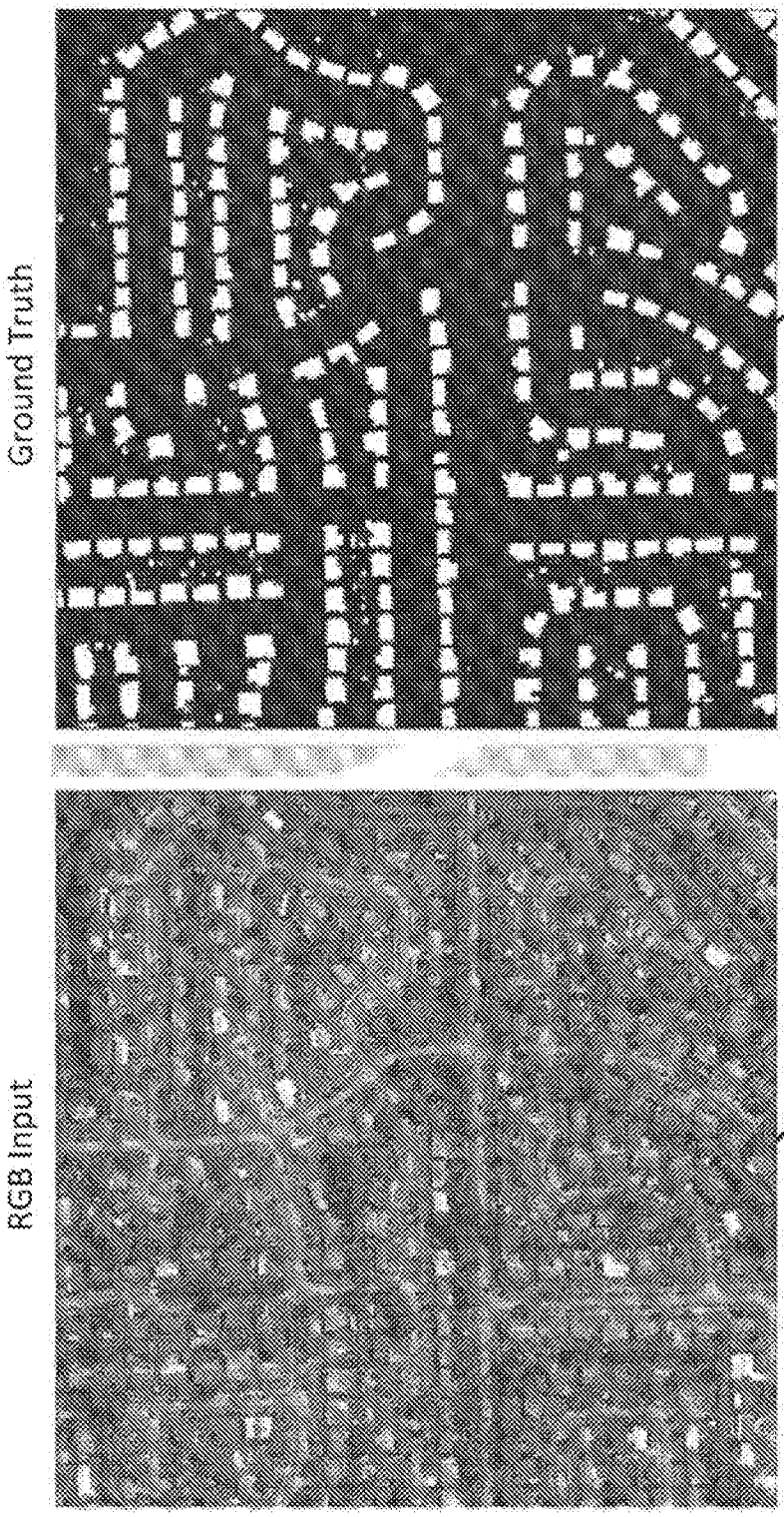


FIG. 36

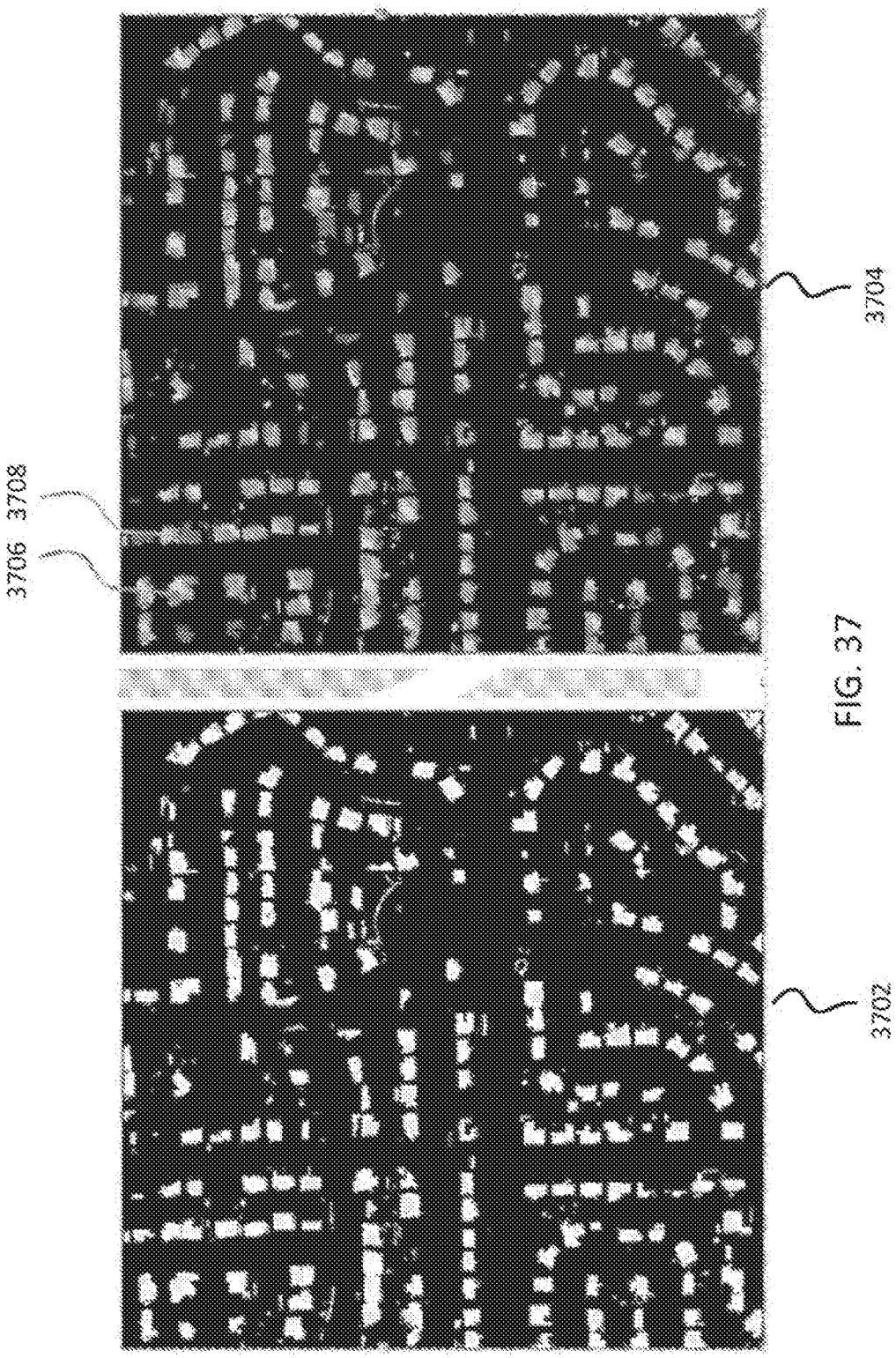


FIG. 37

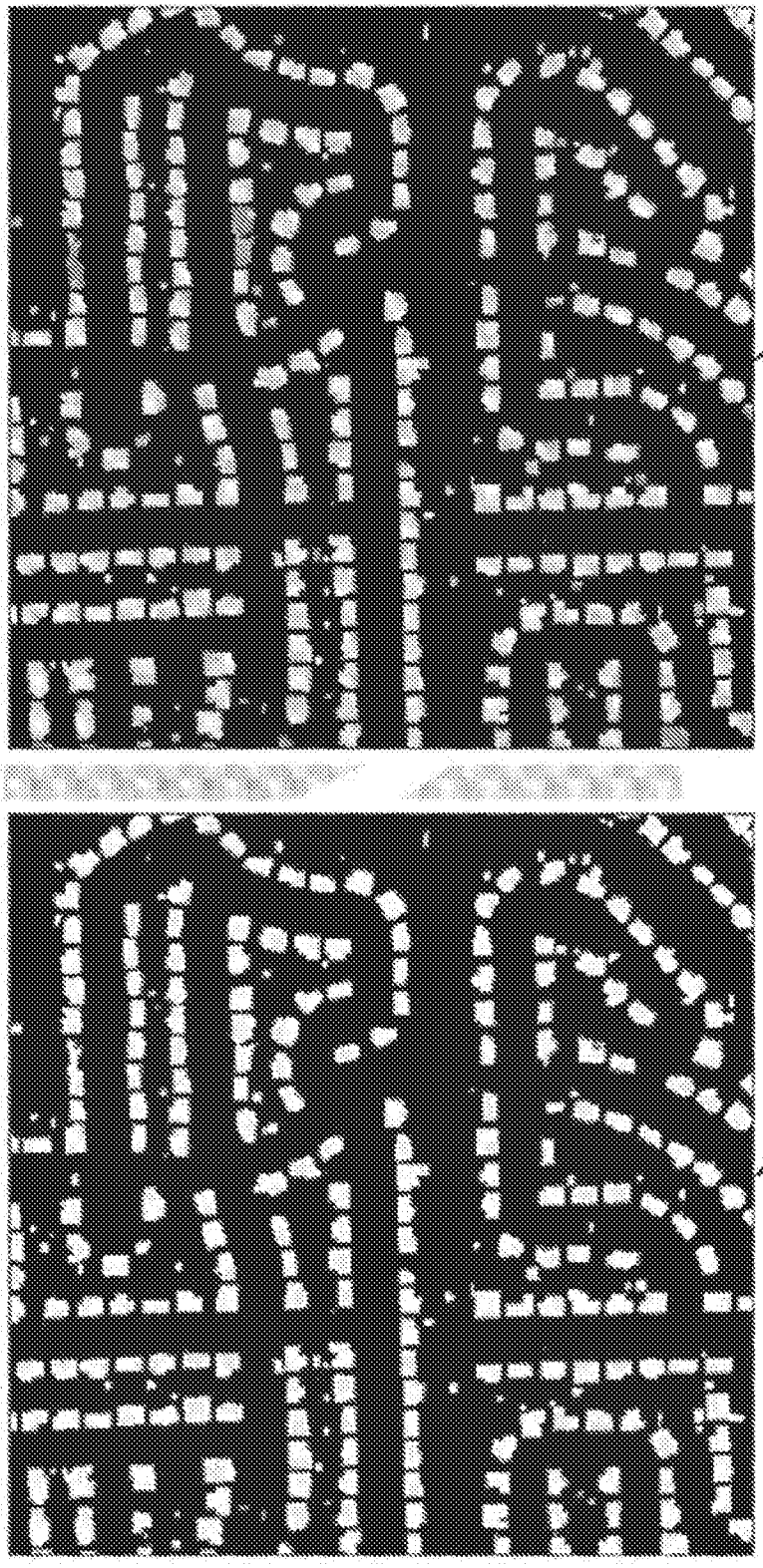


FIG. 38