US 20110187713A1

(19) United States

(12) Patent Application Publication

Pershing et al.

(10) Pub. No.: US 2011/0187713 A1

(43) Pub. Date: Aug. 4, 2011

(54) GEOMETRIC CORRECTION OF ROUGH WIREFRAME MODELS DERIVED FROM PHOTOGRAPHS

(75) Inventors: **Chris Pershing**, Redmond, WA (US); **John F. Keane**, Kenmore, WA (US)

(73) Assignee: **EAGLE VIEW TECHNOLOGIES, INC.**, Redmond, WA (US)

(21) Appl. No.: **13/019,228**

(22) Filed: **Feb. 1, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/300,414, filed on Feb. 1, 2010.

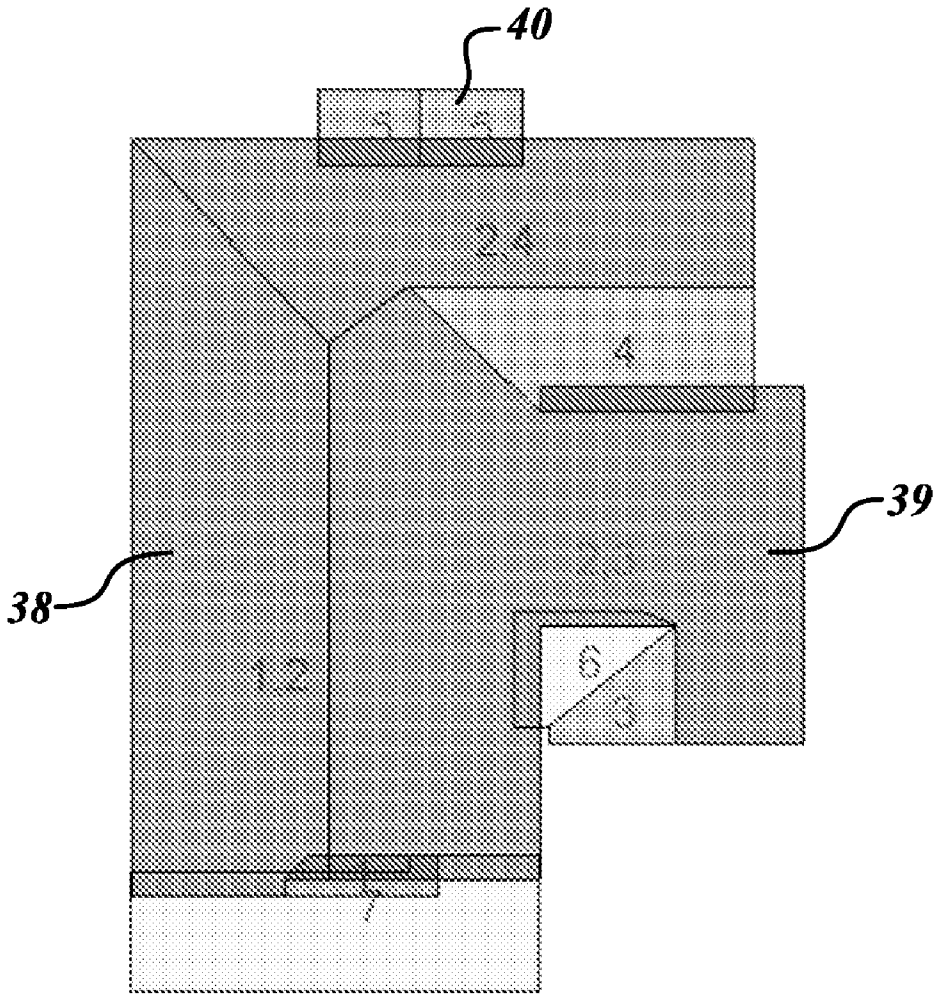
Publication Classification

(51) **Int. Cl.**
G06T 17/00 (2006.01)

(52) **U.S. Cl.** **345/420**

(57) **ABSTRACT**

Geometric correction of rough wireframe models derived from photographs may include rectification of either a 2D or 3D original wireframe model of a roof structure, derivation of metadata from the original wireframe, in-plane normalization of the wireframe, extrusion into a "rough" 3D wireframe based on the normalized wireframe, and correction of the "rough" 3D wireframe. The correction and normalization may be an iterative process based on initial pitch values, metadata derived from the original or corrected wireframe models and defined constraints regarding relationships between roof portions or segments. The iterative process may repeat adjusting the wireframe model until the adjusting converges to a stable state according to the various defined constraints.



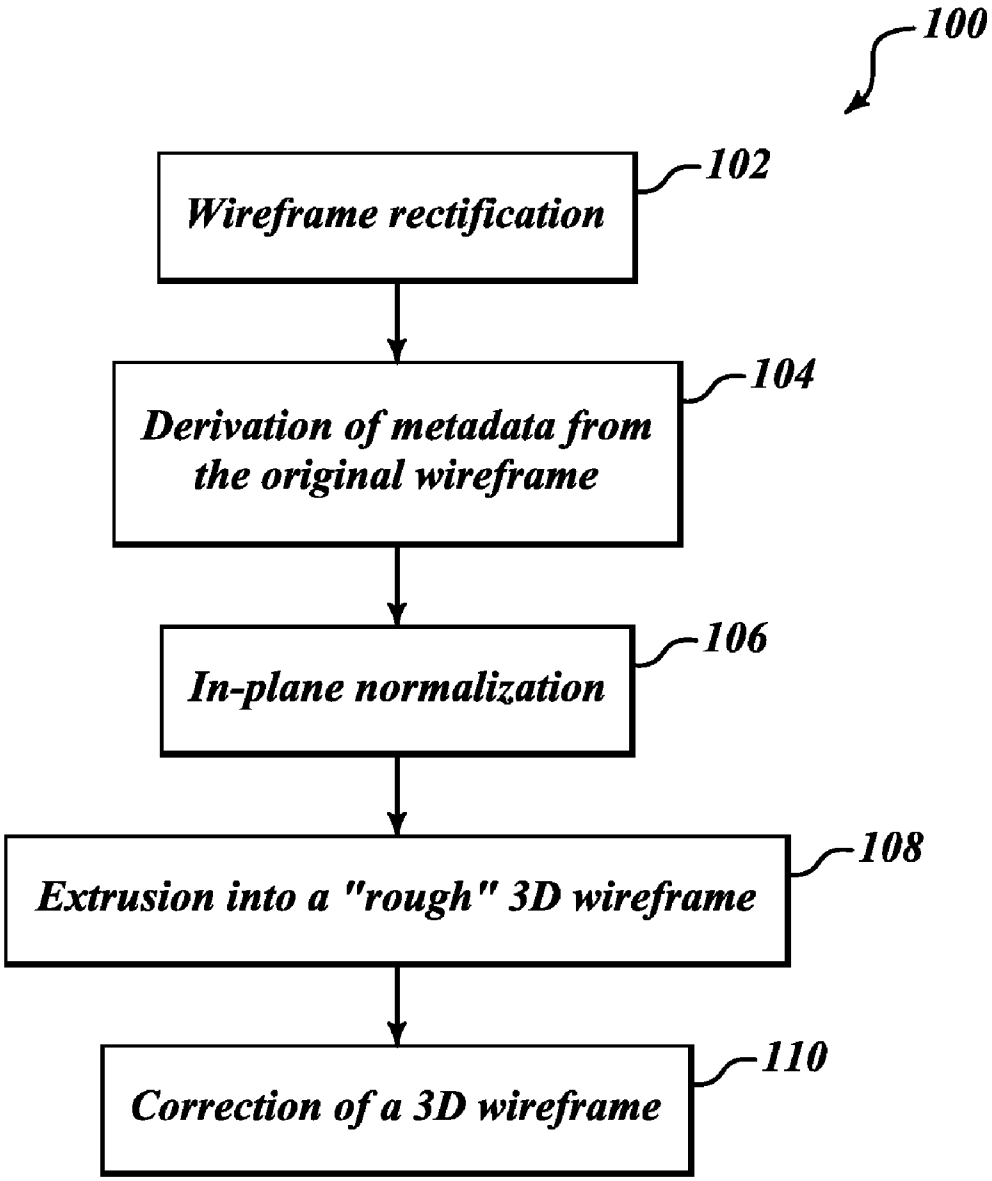

100
Wireframe rectification
102
Derivation of metadata from the original wireframe
104
In-plane normalization
106
Extrusion into a "rough" 3D wireframe
108
Correction of a 3D wireframe
110

FIG.1A

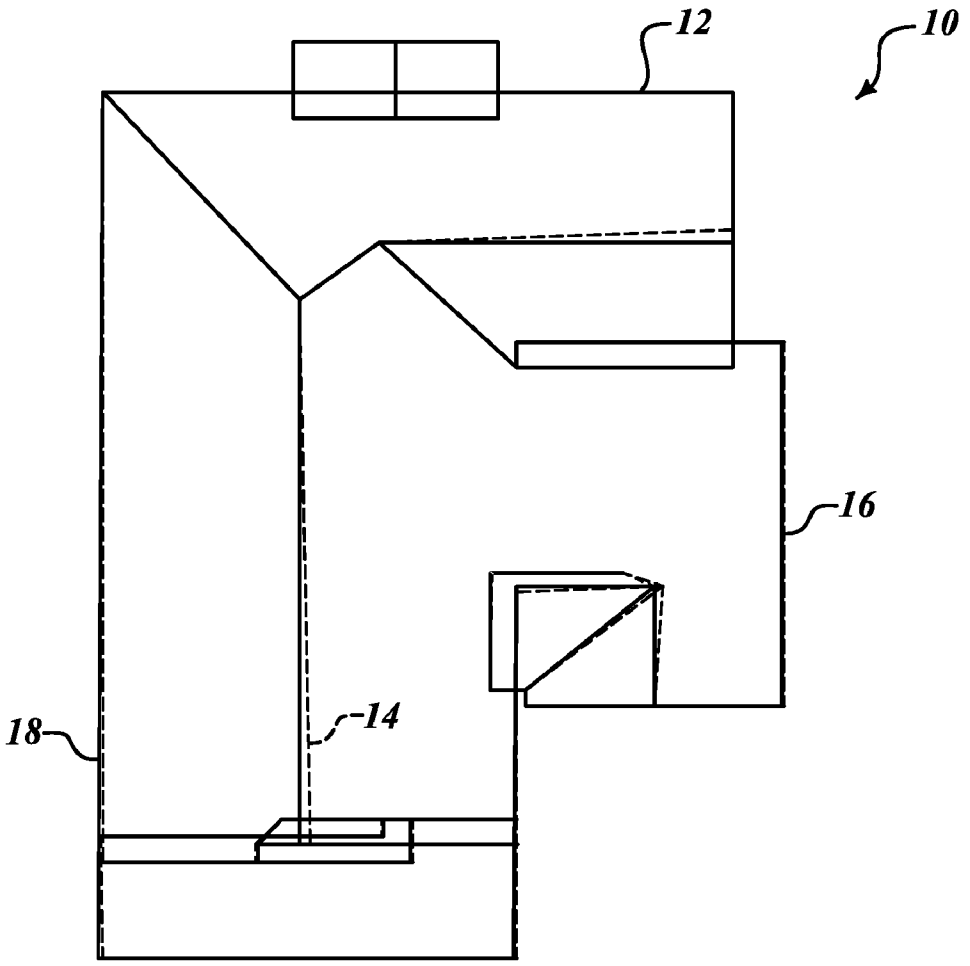


FIG.1B

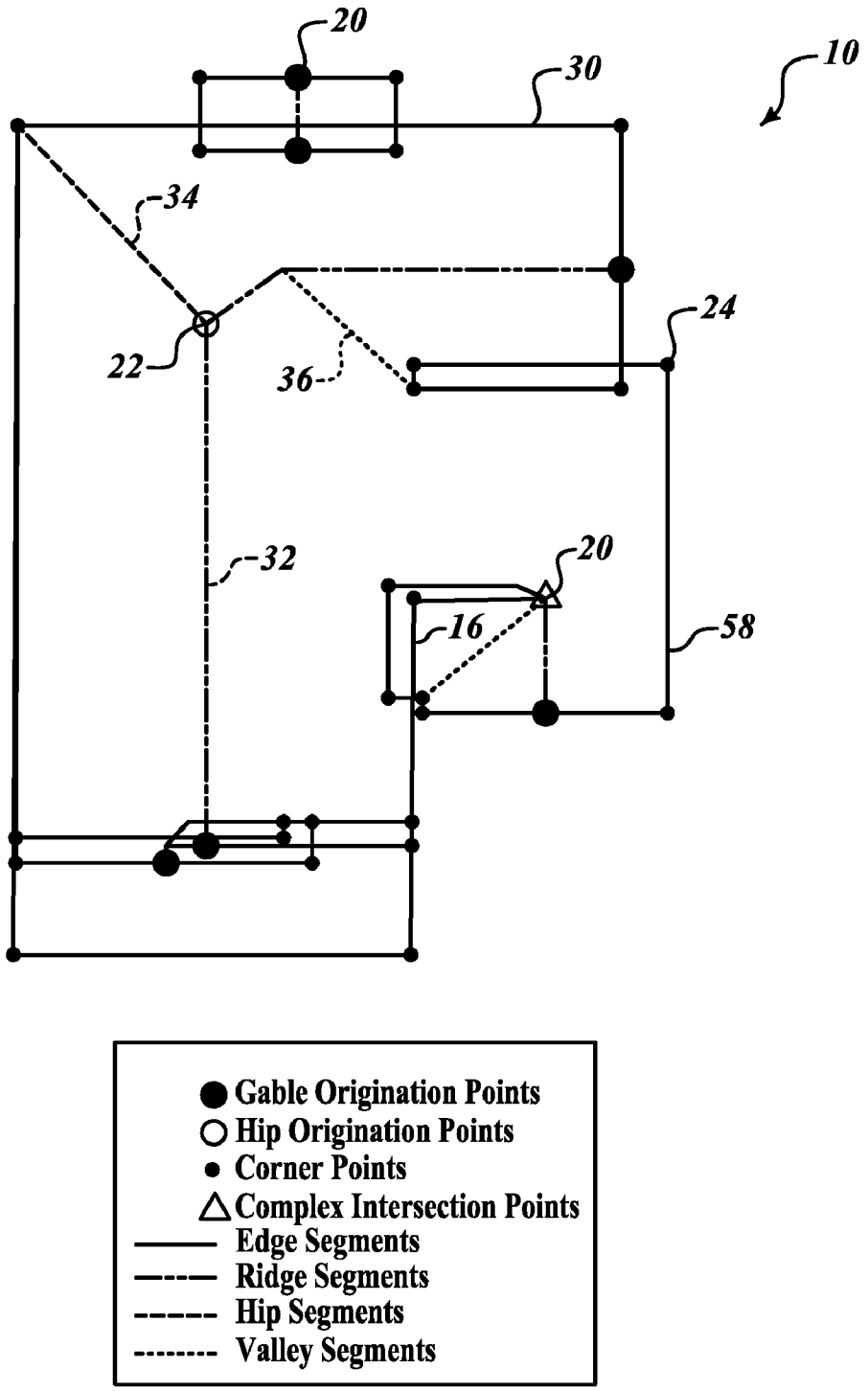

20
30
10
34
24
22
36
32
20
16
58
Gable Origination Points
Hip Origination Points
Corner Points
Complex Intersection Points
Edge Segments
Ridge Segments
Hip Segments
Valley Segments

FIG.2A

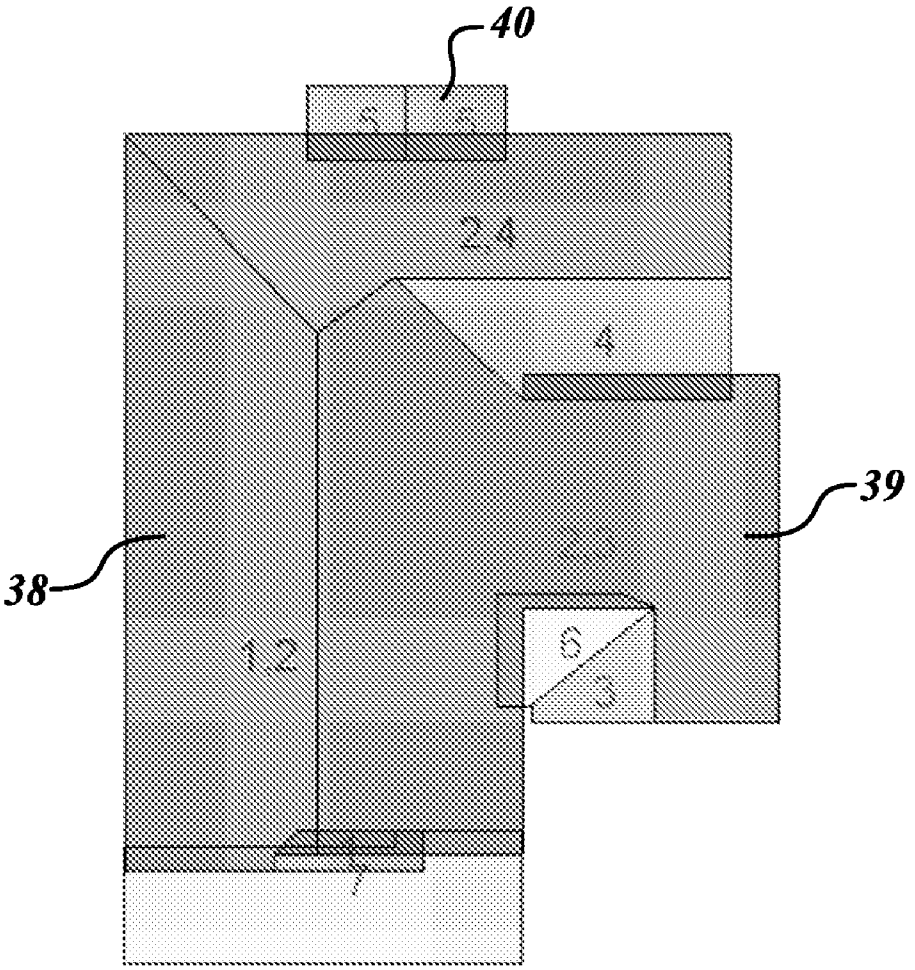


FIG.2B

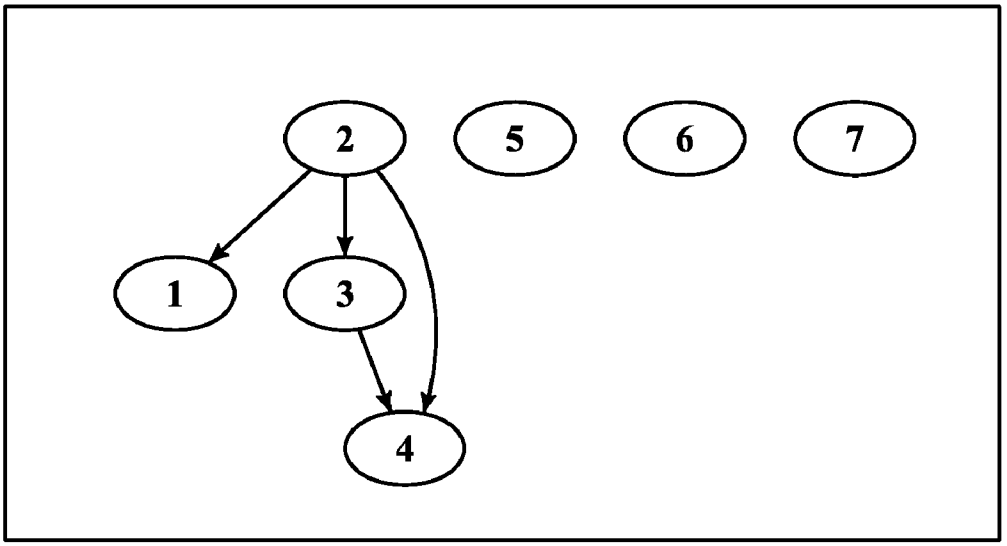


FIG.3

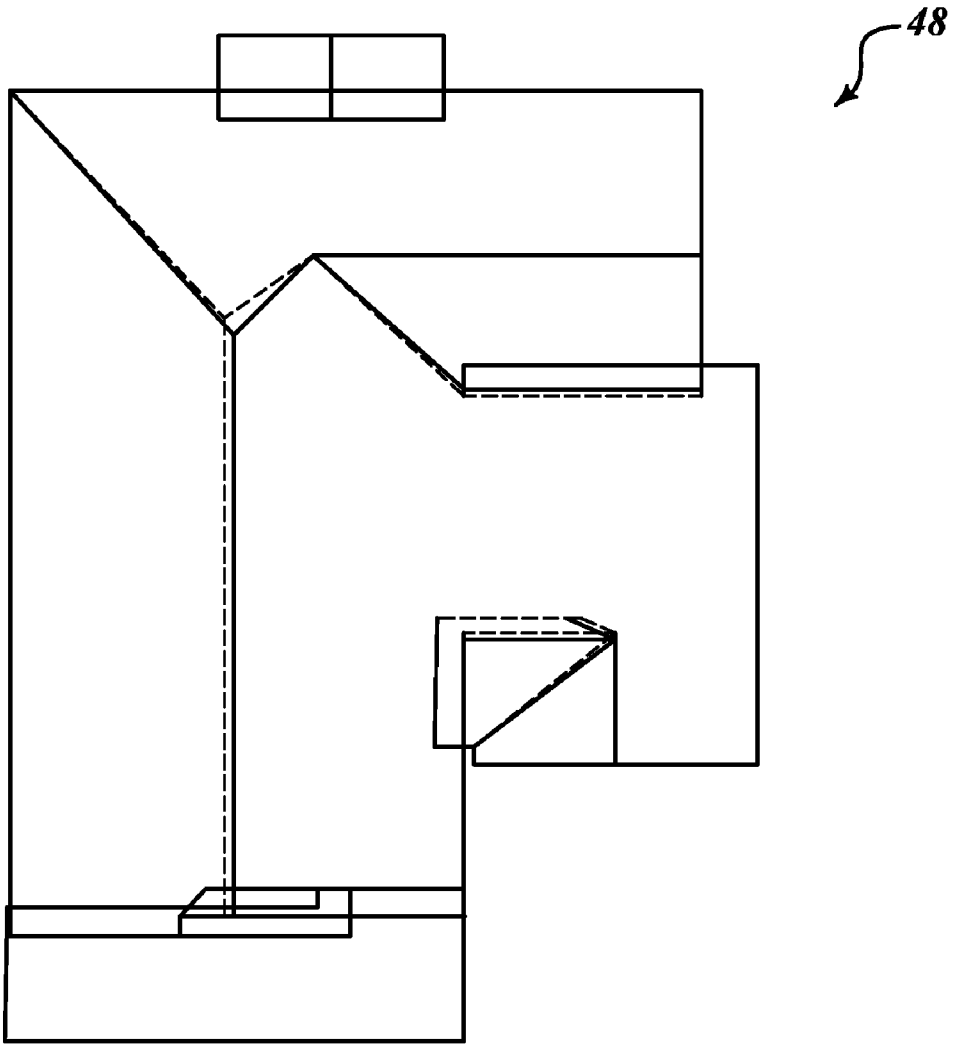


FIG.4

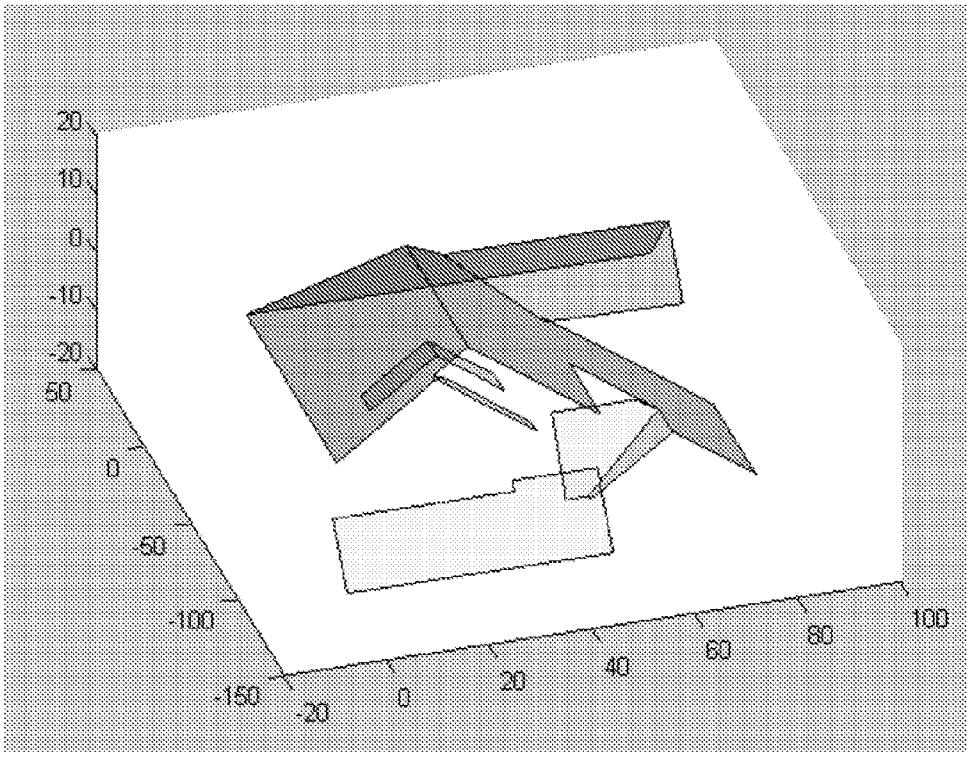


FIG.5

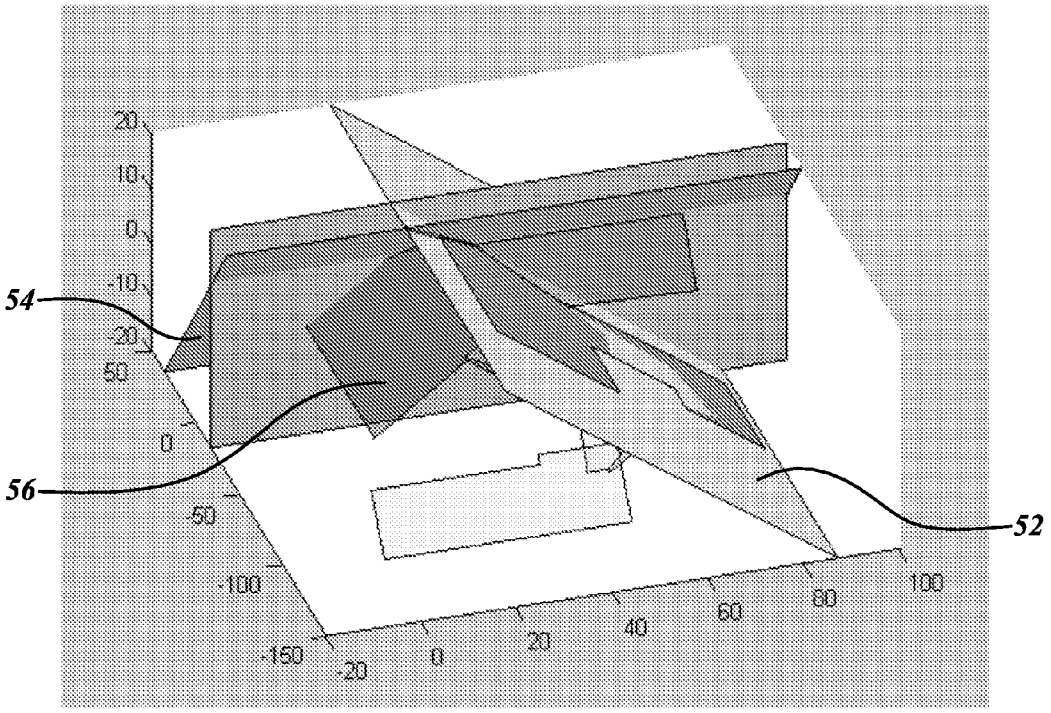


FIG.6

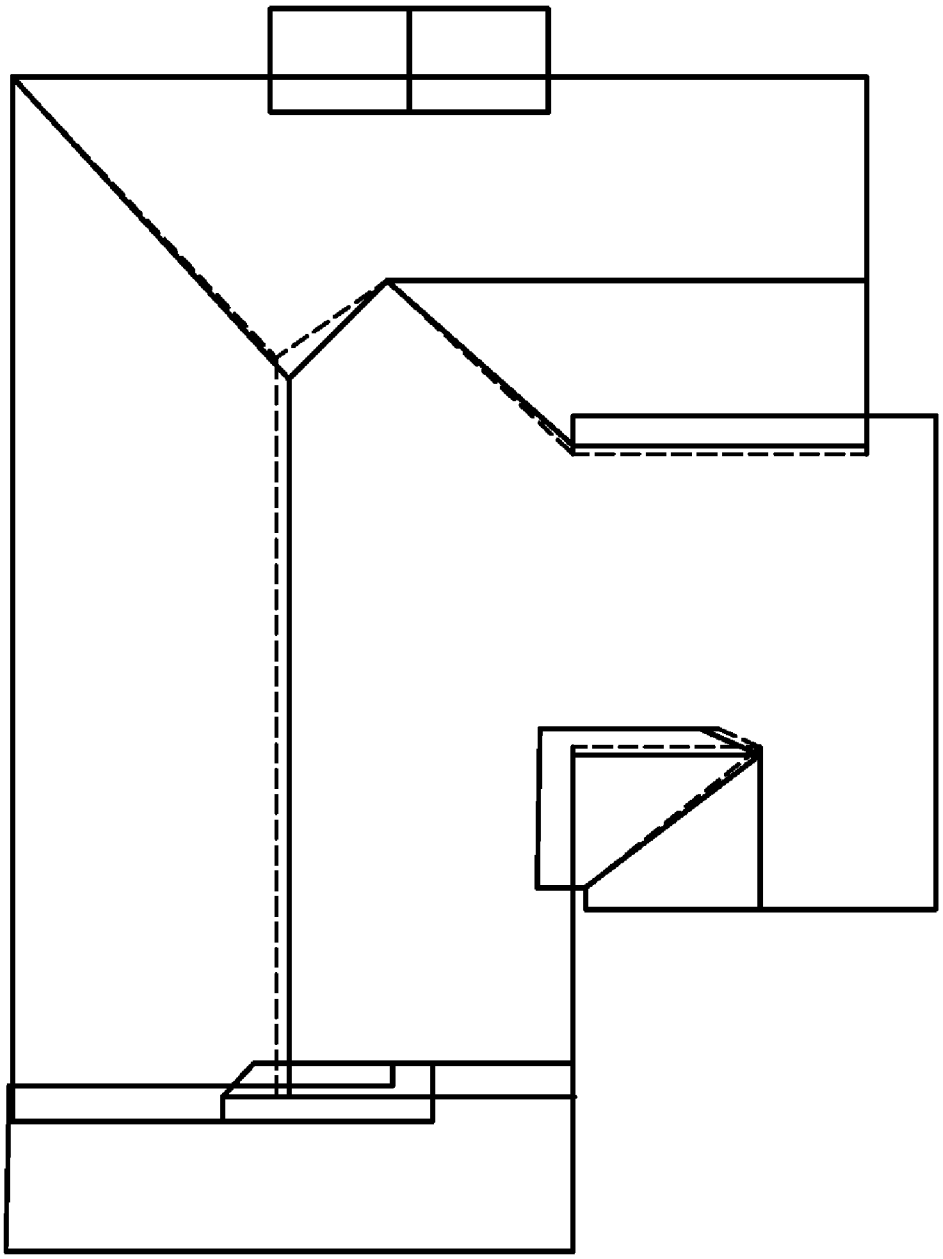


FIG. 7

FIG.8A

FIG.8B

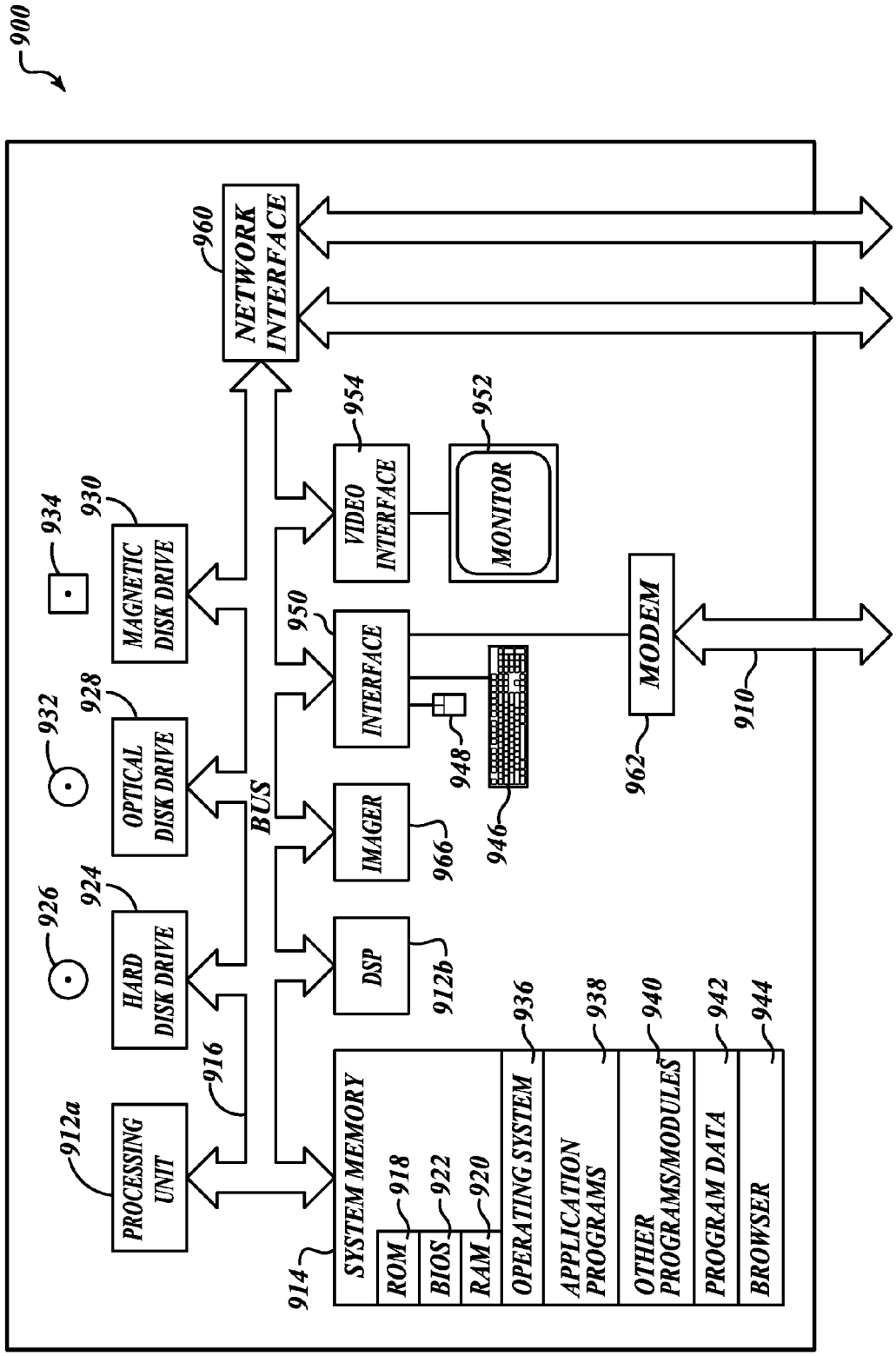

900
NETWORK INTERFACE
960
954
952
VIDEO INTERFACE
MONITOR
934
930
MAGNETIC DISK DRIVE
950
INTERFACE
MODEM
932
928
OPTICAL DISK DRIVE
BUS
948
946
962
910
IMAGER
966
926
924
HARD DISK DRIVE
DSP
912b
936
938
940
942
944
912a
PROCESSING UNIT
916
SYSTEM MEMORY
ROM
BIOS
RAM
OPERATING SYSTEM
APPLICATION PROGRAMS
OTHER PROGRAMS/MODULES
PROGRAM DATA
BROWSER
914
918
922
920

FIG. 9

GEOMETRIC CORRECTION OF ROUGH WIREFRAME MODELS DERIVED FROM PHOTOGRAPHS

BACKGROUND

[0001] 1. Technical Field

[0002] This invention is in the field of computer modeling of physical objects and, in particular, the generation of a three-dimensional model using one or more two-dimensional photographs as a starting data set.

[0003] 2. Description of the Related Art

[0004] Several methods can be employed to generate a three-dimensional computer model of a roof in the form of a wire-frame using one or more overhead photographs.

[0005] A first method is to trace the image in the photograph with or without user input. Using this tracing method on a photograph that provides a nearly top-down view of the structure of interest, a computer-aided design (CAD) tool allows lines and polygons to be drawn over the background image. The resulting wireframe model reflects the lines and contours of interest in the locations they are observed in the photograph.

[0006] While this method has the advantage that it is fast and compatible with many off-the-shelf CAD and drawing tools that do not require much training, there are a number of disadvantages. One disadvantage is that incorrect line lengths on sloped roof facets will result when the image is not taken from directly overhead, resulting in a "lean" of the roof within the image.

[0007] Another disadvantage is that there is a poor ability to correctly determine pitches on sloped roof facets.

[0008] Other problems are a poor determination of layered or underhanging roof sections. This first method requires a user to have previous knowledge of the roof facet pitches to obtain a wireframe without errors.

[0009] A second method is for a user to select and align roof 'primitives' over a photograph.

[0010] Like the previous method, an overhead photograph is selected as a backdrop starting point for a CAD tool. However, rather than tracing lines and polygons, the CAD tool allows the user to overlay, resize, and connect common geometric roof forms stored as primitives. Gables and hips are examples of commonly found primitive forms used in roof designs. The photograph provides a visual reference for the selection and placement by the user of these higher-level objects.

[0011] While this method has the advantage that a skilled operator can produce geometrically correct models, it has the disadvantage that it is harder to create from an image and it becomes more time consuming with increasing roof complexity.

[0012] In addition, complex user interface software is required, which requires greater training.

[0013] This second method also requires subjective technician judgments by the user to compensate for lean artifacts due to lean and shadows in images.

[0014] The user must also have previous knowledge of the roof facet pitches to correctly size the roof.

[0015] A third method is for a user to obtain and work with one or more photographs of the same roof taken from different angles in order to trace a 3D wire model of the roof.

[0016] This method starts with a set of photographs that are analyzed to derive a virtual 3D volume that is geometrically consistent with all the supplied viewpoints of the target roof. The line segments comprising the wireframe are then drawn in 3D-space and projected back onto the set of photographs to assess the correct placement within the derived 3D volume. The details of this method are described in prior U.S. patent applications owned by the assignee, EagleView Technologies, and bearing U.S. patent application Ser. Nos. 12/148,439; 12/253,092; 12/590,131; 12/467,244; and 12/467,250.

[0017] The advantage of this third method is that it does not require previous knowledge of roof facet pitches, rather, pitches are derived from the content of the photograph(s). It also produces geometrically correct models. Unfortunately, the detailed 3D model made this way cannot be easily imported to some widely used home construction CAD software programs.

[0018] The disadvantages of this third method are that greater training is required as compared to the first method and it becomes harder and more time consuming with increasing roof complexity without further processing or touch-up of the model to meet import requirements of the 3rd party software. To summarize, the first method provides a fast and simple means of generating a wire frame model of a roof at the expense of decreased accuracy and a dependence on having previous knowledge of the roof facet pitches to complete the model.

[0019] The second method sacrifices some speed and ease of use in exchange for a more consistent model and a dependency on having previous knowledge of the roof facet pitches to complete the model.

[0020] The third method also sacrifices speed and requires more user skill than the first method, but the resulting model is a very accurate reflection of the correct geometric form of the house in the photograph and does not depend on previous knowledge of the roof facet pitches.

SUMMARY OF THE INVENTION

[0021] The present invention provides a method of forming both a 2D and a 3D wireframe model in computer software that are accurate and geometrically correct. Further, the technique does not require prior knowledge of the roof facet pitches. The produced model can be easily converted into formats compatible with import into 3rd party CAD software.

[0022] Five technology components have been developed as part of the inventive solution:

[0023] 1) Wireframe rectification, either 2D or 3D;

[0024] 2) Derivation of metadata from the original wireframe;

[0025] 3) In-plane normalization;

[0026] 4) Extrusion into a "rough" 3D wireframe; and

[0027] 5) Correction of a 3D wireframe.

[0028] While each of the five components contributes to the overall solution, each can be used independently or in various combinations to yield improvements in obtaining a 2D or 3D wire model as discussed below.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0029] FIG. **1**A is a flow chart showing an example process of geometric correction of rough wireframe models derived from photographs, according to one embodiment of the present invention.

[0030] FIG. **1**B shows a top side 2D view of wireframe correction according to one embodiment of the present invention.

[0031] FIG. **2**A illustrates a precursor step of deriving metadata from a proposed wireframe according to principles of the present invention.

[0032] FIG. **2**B is a top side view illustrating the membership of various roof facets within geometric primitives identified from the metadata for the roof of FIG. **2**A.

[0033] FIG. **3** illustrates the hierarchy operations in deriving the primitives of FIG. **2**B.

[0034] FIG. **4** illustrates the correction of a "rough" 3D wireframe from a normalized 2D wireframe using rules from the metadata.

[0035] FIG. **5** illustrates a 3D wireframe model of the roof of FIG. **2**A according to one embodiment.

[0036] FIG. **6** illustrates a plurality of geometrical objects representing the state of the correction algorithm during the process of correcting the spatial location of a point in the wireframe model of FIG. **5**.

[0037] FIG. **7** shows a fully normalized and corrected 2D wireframe model.

[0038] FIGS. **8**A and **8**B illustrate the finalized 3D wireframe model superimposed back onto the photograph of the original home.

[0039] FIG. **9** is a schematic diagram of a computing environment in which geometric correction of rough wireframe models derived from photographs may be implemented or of which it may be a part.

DETAILED DESCRIPTION

[0040] FIG. **1**A is a flow chart showing an example process **100** of geometric correction of rough wireframe models derived from photographs, according to one embodiment. In particular, FIG. **1**A illustrates process **100** that may be performed or implemented by, for example, one or more software modules or components or any combination of suitable hardware, firmware or software components or devices including those that are a part of or configure the computing environment of FIG. **7**.

[0041] While each of the steps shown in FIG. **1**A contributes to the overall solution, each can be used independently or in various combinations to yield improvements in obtaining a 2D or 3D wire model as discussed below. Below is an overview of each step in the process, which will be followed by a more detailed discussion of each step.

[0042] At **102**, the process performs wireframe rectification. This is a pre-cursor step that can optionally be used within or as a precursor to each of the other four steps that follow. The algorithm can work with either a 2D or 3D wireframe as an input. The wireframe comes into this step as a set of connected points forming a set of closed polygons shown as roof **10** in FIG. **1**B. From this set of points, some basic geometric flaws are identified and corrected.

[0043] At **104**, the process derives metadata from the original wireframe. A series of analysis steps are made to derive contextual information that becomes metadata for the wireframe. Each step of the analysis yields an additional layer of metadata derived from the wireframe and previous layers of derived metadata. This combined set of metadata provides valuable context for later processing steps.

[0044] At **106**, the process performs in-plane normalization which takes as input a "rough" wireframe (2D or 3D) generated by a manual or automated process and then performs an 'in-plane' normalization of the wireframe to yield a new wireframe. Any elevation or 'z' coordinate values of the input wireframe are ignored during this normalization step. Therefore, while this step is typically processes 2D wireframes, it remains compatible with a wireframe input that already contains elevation data.

[0045] At **108**, extrusion into a "rough" 3D wireframe is performed by transforming output of the in-plane normalization of step **106** to a "rough" 3D wireframe suitable for import into a subsequent 3D wireframe correction algorithm. Extrusion of a normalized 2D or 3D wireframe is performed to produce a "rough" 3D form using a set of rules combined with metadata that the algorithm derived from the original wireframe in step **104**.

[0046] At **110**, correction of a 3D wireframe is performed. For example the correction step may take as an input the "rough" 3D wireframe generated by the output of the step **108**, or directly from prior art Method 3 or any similar manual or automated method yielding a 3D wireframe. A new wireframe model is generated by correcting the original wireframe to satisfy a set of user-defined constraints and to remove geometric distortions/problems found in the original wireframe that are inconsistent with the selected constraints. This includes systematic processing of primitives, facet polygons, and line segments, which may be continued throughout the wireframe until the corrections converge to a final roof shape with no further changes.

[0047] FIG. **1**B shows a wireframe rectification of a 2D image of a roof. In particular, FIG. **1**B illustrates a roof **10** shown as a 2D top plan view. This is the type of view that would be seen looking down on the top side of a roof from an overhead image. This view can be obtained from an aircraft that flies over the roof **10** and takes a picture using standard photographic equipment.

[0048] A number of potential difficulties arise when a photograph is taken from a top side view of a home of the type shown in FIG. **1**B. A first difficulty is that the camera mounted on the aircraft, satellite, or other aerial platform might not be directly over the roof **10** at the moment that the photograph is taken. If the camera is directly overhead, then an exact top plan view of the roof **10** is provided in the image which is created in the photograph. However, if the camera is off to one side, for example several degrees to the left or right of directly overhead, then the relative position of objects of different elevations (height) within the image of the roof **10** will appear to be shifted when compared to a truly top-down viewpoint. In some cases, presence of this 'lean' in the image can be observed as the ability to see the side of a vertical wall or chimney. It is difficult to correct for the amount of lean unless it is exactly known. In addition, the presence of lean may affect the perception of objects in the image differently depending on the pitch of the various roof faces and the orientation of the roof **10** with respect to the aircraft at the time the photograph was taken. A further problem is that the original images typically undergo a series of processing steps such as compression, resampling, and orthorectification before image is published into an image library. This processing may degrade details and/or introduce geometric distortions that were not present in the original image. Yet a third problem of the top side photograph is the difficulty in determining the pitch of the roof. The pitch is a measure of the slope of a roof facet. Typically, pitch is defined as the ratio change in height (in inches) for every 12 inches of of lateral movement in the direction of steepest descent. This may be reported as a ratio of **6/12** or simple as 6 with an implied denominator of 12 inches. A roof pitch of 0/12 represents a flat roof while a 9/12 pitch denotes a steeper slope than a 6/12

roof pitch. From a single top-side-only photograph, determining the exact pitch is difficult and often not possible. Accordingly, the image of the roof **10** is rectified according to principles of the present invention as set forth herein.

Wireframe Rectification

[0049] FIG. **1**B shows wireframe rectification being performed. This is a pre-cursor step that can optionally be used within or as a precursor to each of the other four processing algorithms that follow. The algorithm can work with either a 2D or 3D wireframe as an input.

[0050] The wireframe comes into this algorithm as a set of connected points forming a set of closed polygons shown as roof **10** in FIG. **1**B.

[0051] From this set of points, a series of processing steps are to identify and correct some basic geometric flaws.

[0052] These steps may include:

[0053] a) Orientation/Rotation: Principal axes of the wireframe are determined to gain an understanding of the orientation of the structure represented by the wireframe. This orientation information can then be used to rotate the coordinate system of the wireframe to any desired orientation.

[0054] b) De-skew: The wireframe is analyzed to determine if a global 'skew' is present, namely, if the two principal axes of the structure are not orthogonal to each other. This type of distortion makes rectangular objects look like parallelograms. If detected, the set of points are transformed to move the lines to become orthogonal and thus remove the skew.

[0055] c) Rectify: Points in the wireframe are shifted in the plane as needed to straighten lines, square off corners, and adjust lines to be either parallel or perpendicular to other lines based on a set of rules using information regarding their proximity to each other, the relation of each major line, and their nearest neighbors. For example, according to one set of rules, long extending edges of the roof **10** on opposite sides are made parallel to each other while corners between connect lines that are very close to 90° are made to be exactly 90° square.

[0056] As can be seen in FIG. **1**B, the solid dark lines indicated by line **12** is one example of the fully corrected final lines of the roof, while the dashed lines **14** represent the position of the line in the original image. The rectified, solid line wireframe has been overlaid on top of the original photograph of wireframe **14** to better illustrate the corrections which have been made and the realignment to each of the lines. As can be seen, the line **12** did not require any changes from the photograph because it was properly aligned, however one more lines **14** required a deskew in order to align the principle axes of the structure with each other. In addition, some alignment was required of the two sides **16** and **18** to ensure that they were parallel with respect to each other. Other appropriate orientation and rotation was carried out in order to properly represent the roof **10** to obtain the rectified solid line pattern overlaid on the original dashed line pattern shown in FIG. **1**B.

Derivation of Metadata from the Original Wireframe

[0057] FIG. **2** shows another pre-cursor step that can be made prior to each of the processing algorithms that follow. The wireframe of the roof **10** is input to this algorithm as a set of connected points forming a set of closed polygons. The algorithm is equally applicable to whether the original wireframe is flat 2D representation of the roof or a 3D representation of the roof.

[0058] From this set of connected points, a series of analysis steps are made to derive contextual information that becomes metadata for the wireframe. Each step of the analysis yields an additional layer of metadata derived from the wireframe and previous layers of derived metadata. This combined set of metadata provides valuable context for later processing steps.

[0059] Basic metadata includes a catalog of nearest neighbor points, nearest neighbor segments, segment membership in polygons, etc.

[0060] This basic metadata is then used to determine more abstract information such as whether segments are edges or ridges, groups of points and/or line segments that will likely reside at the same elevation (Z coordinate) in the final model, and which points look like hip junctions, gable junctions, or corners on the perimeter of the structure, for example.

[0061] Additional metadata layers are then derived to identify the presence of common geometric forms, referred to herein as 'primitives,' that are components of many roof architectures. Examples of such 'primitives' include hips and gables. The identification of the primitives facilitates an association of individual roof facets with one or more primitive objects in standard root geometry.

[0062] Finally, an ordered hierarchy is established between these roof primitives that provides context regarding the spatial relationships that exist between the primitives. This hierarchy is used to establish an order of operations in later processing steps.

[0063] FIG. **2**A shows a first layer of analysis and labeling. The software has analyzed the wireframe and determined some base points and layers of metadata. Shown in this diagram of FIG. **2**A are gable origination points **20**, large dark circles ●; hip origination points **22**, open circles ○; edge segments **30**, dark solid line —; ridge segments **33**, dash-dot line ·····; hip segments **39**, dash lines -----; junction and valley segments **36**, dotted line ·····; corner points **24**, closed black circles •; and 'complex' intersection points **26**, triangles Δ. Of course, the various lines and points can be shown on a computer screen in different colors rather than as different sizes and styles of black to provide ease of use for an operator. For example, the gable origination points can be red, the hip points green or blues and the corner points yellow or black. Similarly, the various lines can be in solid colors, such as red lines for the ridge segments, green lines for the edge segments and magenta or blue or purple for the junction and valley segments. The different metadata are shown here with different black and white representation patterns for the patent application, but in some embodiments, different colors on the computer screen will be preferred.

[0064] In FIG. **2**B, the wireframe and metadata from FIG. **2**A are further processed to identify hip and gable primitives, as well as uncategorized geometric primitives. Facet polygons labeled with a **2** are members of the hip associated with the hip origination point **22** identified in FIG. **2**A. Facets labeled with **1**, **3**, **4**, **5** are members of the respective gable primitives associated with the gable origination points **20** identified in FIG. **2**A. Facets labeled with **6** and **7** are identified as geometric primitives that are not hips, gables, or any other common geometric form recognized by the software. As seen in FIG. **2**B, the mapping of facet polygons to primitives is a one-to-many relationship—e.g., a facet may be a member of more than one identified primitive. For example, the facet **38** is both a gable primitive **2** and a hip primitive **1**, whereas facet **40** is only a gable primitive **5**.

[0065] As shown in FIG. **3**, the metadata derived from the wireframe in previous steps represented by FIGS. **2**A and **2**B is analyzed further to determine hierarchy among the identified primitives. Information from this hierarchy can then be used by other algorithms to establish a preferred order of processing when progressing through the wireframe.

[0066] As seen in FIG. **3**, the algorithm determined the hierarchy of the connected hip and gable primitives. Primitive **2**, the gable on the main ridge line of the entire home, was determined to be higher in the hierarchy than the subordinate (or child) primitives **1**, **3**, and **4**. Multiple parentage is allowed during this process such that Primitive **3** is also identified as parent of Primitive **4**. Algorithms consuming this hierarchy may choose to prune multiple parent linkages such that each child only has a one parent.

[0067] The uncategorized primitives **5**, **6**, **7** are shown without parents or children. For **5** and **7**, this is due to the fact that the underlying polygons are physically disconnected from the main portion of the wireframe containing **1**, **2**, **3**, **4**. Primitive **6** is also shown as disconnected in the hierarchy even through it shares a line segment with the wireframe containing **1**, **2**, **3**, **4**. In this case, the software determined that pruning this primitive from the main hierarchy would result in a better order of operations for down-stream processing. In other situations, the algorithm may have joined Primitive **6** to the main hierarchy as a child of **2** and **3**.

[0068] The algorithm for the selection of hierarchy is based on a set of rules that evaluates the intersections between the identified primitives. These rules normally produce a hierarchy predicting a top-down elevation ordering of connected primitives. In this case, primitives **2** (associated with ridgeline **32**) is ranked above the primitives **1**, **3**, and **4**. This ranking is consistent with relative elevations of the ridglines observed in FIG. **5** resulting from additional downstream processing steps.

In-Plane Normalization

[0069] FIG. **4** shows in-plane normalization in which the algorithm takes as input a "rough" wireframe (2D or 3D) generated by the first technique of FIG. **1**B or by a manual process similar to Method 1 of the prior art, or by an automated process employing computer vision techniques, or by some other free-hand method that may or may not employ a photograph as a background image during the tracing process (e.g., a hand-drawn sketch for example).

[0070] The algorithm then performs an 'in-plane' normalization of the wireframe to yield a new wireframe. As the name suggests, this algorithm ignores any elevation or 'z' coordinate values of the input wireframe during this normalization step.

[0071] Therefore, while this algorithm is typically used to process 2D wireframes, it remains compatible with a wireframe input that already contains elevation data.

[0072] A top-down view of the normalized result **48** shows a wireframe that has been corrected for lean effects and many other geometric distortions and imperfections that may be present in an original wireframe. Although the elevation values of the points within this wireframe may not have been adjusted yet, the shapes of the polygons have been modified to yield a true top-down viewpoint for each facet that is consistent with a set of initial pitch values assigned to the roof facets. In another embodiment, a precursor step to the In-Plane Normalization may have made adjustments/corrections to the 'z' coordinate values (elevation values) of the points.

[0073] One great value of the removal of the lean and other distortions is realized when this normalized model is used as an input to the Extrusion and 3D -correction algorithms discussed later in this document. The errors associated with those lean artifacts and edge distortions would otherwise become amplified during the subsequent extrusion and correction steps and degrade the quality of the end result.

[0074] When the algorithms of the invention are used in this sequence, the speed and ease of use of prior art Method 1 provides a precursor model that can be further processed to yield a model that is equivalent in quality and accuracy to the output of prior art Method 3.

[0075] This algorithm provides several competitive benefits:

[0076] a) Significant reduction of manual drawing time, which was the primary benefit of prior art Method 1.

[0077] b) Minimal technician training time, another primary benefit of prior art Method 1.

[0078] c) Maintains all the benefits of prior art Method 3—accuracy, consistent geometry, and no dependence on previous pitch knowledge.

[0079] d) Maintains compatibility with existing construction methods.

[0080] e) Supports and enhances models derived from automated computer vision methods.

[0081] Following are the in-plane normalization algorithm details:

[0082] a) Precursor steps: Wireframe metadata has been determined and the structure has been processed by the rectification algorithm as set forth in FIGS. **1**B-**3**. The use of either or both of these techniques as a precursor step is preferred but not required.

[0083] b) Seeding of Pitch Values: An initial set of pitch values and orientations are assigned to each roof facet. The assignment could be via user assignment, by an algorithm that assigns a pitch based on metadata derived from the 2D or 3D wireframe, or by an algorithm that gleans pitch information from the existing "rough" 3D wireframe provided as an input.

[0084] c) Adjustment of ridgelines: A systematic pass is made through the wireframe to adjust the lateral position of the points associated with line segments identified as 'ridges'. The adjustments are made to accommodate pitch values associated with the polygons that share the ridge segment and/or the ridge segment's endpoints.

[0085] d) Correction of Junctions between Primitives: Line segments that have been identified as junctions or seams between adjacent primitives are likewise adjusted to accommodate the specified pitches and the adjusted positions of the ridgelines. An iterative loop repeats steps (c) and (d) until the repositioning of lines and points converges to a stable state. In an alternative embodiment, the loop repeats steps (b), (c), and (d) allowing flexibility in adapting the pitch of facets when converging to the final solution.

[0086] FIG. **4** shows the result **48** of the In-Plane Normalization, shown by the solid lines in FIG. **4**, that is overlaid on the rectified wireframe of FIG. **1**B, which is shown by the dashed lines of FIG. **4**.

[0087] The shift of the line positions resulted from a repositioning of the hip and gable origination points **20** and **22** and junction line segments **32** to accommodate a set of pitch values that were assigned to each of the facets. This adjusts for lean and rotation that may have been present in the first image. In this example, the corrections have been made based on an

assigned pitch of 7/12 for all facets except for the facet associated with primitive **6**, which has been assigned a pitch of 9/12.

Extrusion into a "Rough" 3D Wireframe

[0088] FIG. **5** shows the results of an algorithm as a transition step that transforms output of the in-plane normalization to a "rough" 3D wireframe suitable for import into a subsequent 3D wireframe correction algorithm.

[0089] This algorithm performs an extrusion of a normalized 2D or 3D wireframe to produce a "rough" 3D form using a set of rules combined with metadata that the algorithm derived from the original wireframe in FIGS. **2**A and **2**B and **4**, for example.

[0090] The resulting "rough" 3D model may contain some geometric inconsistencies that were not anticipated or corrected during the in-plane normalization step or during the extrusion step.

[0091] The "rough" 3D model is then used as an input into the aforementioned 3D correction algorithm for further processing.

[0092] Extrusion algorithm details:

[0093] a) Precursor steps: Wireframe metadata has been determined and pitch values have been assigned to each of the facets. The structure has optionally been processed by the rectification algorithm and/or optionally has undergone an in-plane normalization step.

[0094] b) Extrusion of the wireframe into 3D: A systematic pass is made through the extruded wireframe to correct the locations of points in the wireframe to accommodate the desired pitches and facet orientations. Information from the primitive hierarchy provides context for the order of primitive processing. Combined with other derived metadata, the primitive hierarchy also provides contextual clues about the relative elevations between groups connected by primitives. If the input wireframe is already in a "rough" 3D form, elevation information from this model may optionally be used by the Extrusion algorithm to "seed" initial relative elevations of points, lines and/or primitives with respect to other points, lines, and/or primitives present in the mode. If the input wireframe is 2D, it does not contain elevation information. In such a case, metadata containing hints about relative elevations between objects in the model (such as a disconnected patio or bay window cover) can be used to provide context for placing some objects above or below others in the extruded 3D output.

[0095] As shown in FIG. **5**, the in-plane optimized wireframe of FIG. **3** has been extruded into a "rough" 3D wireframe.

Correction of 3D Wireframes

[0096] FIG. **6** shows algorithm that takes as an input the "rough" 3D wireframe generated by the output of the 'Extrusion' algorithm of FIG. **5**, or directly from prior art Method 3 or any similar manual or automated method yielding a 3D wireframe.

[0097] The algorithm then generates a new wireframe model by correcting the original wireframe to satisfy a set of user-defined constraints and to remove geometric distortions/problems found in the original wireframe that are inconsistent with the selected constraints. As can be seen in FIG. **6**, whole planes of the proper geometric shape are overlaid in an interest to assist in correction of prior possible error.

[0098] For example, an original wireframe originating from Method 3 may contain some erroneously shaped roof polygons due to errors made by an inexperienced technician or even by a highly experienced technician that can quickly create a "rough" model that may contain a series of small correctible defects. In another example, an automated computer vision process produces a wireframe that contains correctable defects resulting from the small statistical uncertainties in its ability to determine the exact location of poorly resolved boundaries. In a third example, the output from the 'Extrusion' algorithm of FIG. **5** may contain defects resulting from complex intersection from several planes which require further normalization to correctly resolve.

[0099] In these situations, the correction algorithms process the "rough" input wireframe to yield a wireframe that is consistent with a set of selected constraints. Examples of such constraint may include:

- **[0100]** Preserving the footprint (e.g., perimeter) of the original wireframe
- **[0101]** Correcting angles of intersection between line segments
- **[0102]** Straightening line segments
- **[0103]** Specifying a particular pitch for one or more facets
- **[0104]** Allowing the pitch of one or more facets to conform with other constraints
- **[0105]** Enforcing relationships between line segments and facets to satisfy symmetry requirements.
- **[0106]** Correcting of 'lean' artifacts that are identified in the original model

[0107] Additional constraints that have proved to be useful may include:

- **[0108]** Constraining groups of points and/or line segments to reside at the same elevation (elevation groups). For example, such line segments include the line segments (and points they contain) that are parallel and equidistant from a ridge line as measured by the perpendicular distance between the ridge and the line segments. See the two line segments **58** and **60**, which meet these criteria for the ridge **20** in FIG. **2**A.
- **[0109]** Constraining two or more elevation groups to maintain a particular difference in elevation. For example, Group **1** is the ridge **20** in FIG. **2**A. Group **2** are the line segments identified in the above example of an elevation group.
- **[0110]** Enforcing symmetry between nearly symmetric objects in the model, such as between the left and right planes **5** in FIG. **2**B.

[0111] The constraints above are additional examples of the use of metadata derived from the input model and/or collected interactively from the user. Also, the constraints above may be followed according to an assigned priority. For example, if one constraint cannot be adhered to without violating another constraint, the constraint to adhere to may be decided based upon which constraint has a higher priority. The priorities may be assigned by a user to be followed automatically during the processes described herein or created dynamically during the process of 3D wireframe generation. The metadata used in the examples above derived from the input model and/or collected interactively from the user may also or instead be used in the process for extrusion into a "rough" 3D wireframe described above.

[0112] This algorithm provides several competitive benefits:

[0113] a) It automates small corrections that would otherwise require manual correction by quality control staff that review the work of other technicians.

[0114] b) It decrease the time spent drawing a 3D wireframe using manual processes. Technicians can be less precise when "roughing" out a 3D model. This time savings increases with the complexity of the roof model.

[0115] c) It reduces some skill and training time required for new technicians

[0116] d) It processes constraints specified to produce models that satisfy a set of conditions required by third party CAD applications that desire to import the model for further editing.

[0117] e) It provides mechanisms to "clean up" or adjust models generated from other processes or programs.

[0118] Some software used by insurance claim adjustments has a set of constraints for their CAD software that renders roof images. Some software tools have a specific set of requirements that must be satisfied in order to import a roof diagram into their CAD software tool. The ability to modify a model derived from a photograph image to conform to the software requirements of specific CAD software tools is provided by the features of this invention. This yields the advantage of compatibility across the industry, one of the benefits of the inventive techniques described herein.

[0119] 3D correction algorithm details:

[0120] a) Precursor steps: Wireframe metadata has been determined and pitch values been assigned to facets. The pitch value assignment could be via user assignment, by an algorithm that automatically assigns a pitch based on metadata derived from the 2D or 3D wireframe, or by an algorithm that derives or assesses pitch information from the existing "rough" 3D wireframe provided as an input. The structure has optionally been processed by the rectification algorithm and/or optionally has undergone an in-plane normalization step and/or has optionally undergone an extrusion step.

[0121] b) Assembly: The "rough" wireframe input and the associated metadata provide a prototype from which a new wireframe is assembled while enforcing a set of constraints on the wireframe. These constraints may include such things as: i) A requirement that the set of points in each identified roof facet are coplanar to within a specified tolerance. ii) The area footprint in the XY plane of the wireframe's perimeter remains within a specified tolerance of the original "rough" wireframe's footprint area. iii) ridgelines and eavelines remain parallel to the XY plane.

[0122] FIG. **6** is a "rough" wireframe of the 3D frame shown in FIG. **4** overlayed with several known geometric objects. This figure illustrates a snapshot of the processing being performed to correct the spatial location of one point within the "rough" 3D form of the wireframe. The large plane **52** highlights the facet polygon **39** from FIG. **2**B currently being corrected. The plane **54** highlights the adjoining planar facet sharing the currently processed line segment with the main plane **52**. The line segment in this example originates from the ridgeline of Primitive **2** and terminates at the ridgeline of Primitive **4**. The darker plane **56** is a plane orthogonal to the XY plane that intersects the ridgeline of Primitive **4**. The coordinates of the terminal endpoint of his line segment are then calculated as the intersection of the plane **56** with the line segment computed from the intersection of the planes **54** and **52**, respectively. In FIG. **5**, the various planes and geometric shapes are shown in different shades of gray; of course, on a computer screen, each of the geometric shapes or planes may be a different color to provide highlight for a user. For example, the plane **52** may be green, the plane **54** red, the polygon **39** blue and the darker plane **56** a brown or orange. This diagram serves to illustrate one of many similar internal processing steps that are not typically displayed to the end user because they typically do not require user input. The systematic processing of primitives, facet polygons, and line segments is continued throughout the wireframe until the corrections converge to a final roof shape with no further changes. If convergence does not occur, the software may exit or produce a corresponding message or alert due to a non-convergence condition.

Typical Usage Scenarios:

[0123] In the "background" section of this document, three prior art methods were identified and discussed as the motivation for inventing new methods that leverage best efficiency and accuracy attributes of the present methods.

[0124] Various new technology components, such as, for example, those listed below as items 1-5 resulted from this effort:

[0125] 1) Wireframe rectification

[0126] 2) Derivation of metadata from the original wireframe

[0127] 3) In-plane normalization

[0128] 4) Extrusion into a "rough" 3D wireframe

[0129] 5) Correction of a 3D wireframe

[0130] Although the above technology components can be used independently, there are a number of usage embodiments envisioned involving these various technology components in various scenarios. The following labels for each step in each scenario below are for ease of identification and recollection of common steps within the various scenarios for the sake of brevity and do not limit the content or activities in each step.

Scenario 1

[0131] In scenario 1 below, the output is a corrected 3D wireframe.

[0132] a) "Wireframe": A 2D or 3D wireframe is used as input along with (optional) registration information that enables the system to overlay, project, and/or map the points of the input 2D or 3D wire frame onto one or more photographs containing an image of the structure of interest.

[0133] b) "Rectify": Wireframe Rectification is optionally applied to the input 2D or 3D wire frame.

[0134] c) "Metadata": Metadata is derived from the input 2D or 3D wire frame and/or the output of (b).

[0135] d) "AssignPitch": Pitches are either assigned by algorithms that analyze neighboring polygons to infer relative pitch values or the pitches are assigned by the user.

[0136] e) "Normalize": The in-plane normalization algorithm is applied to the wireframe.

[0137] f) "Extrude": The Extrusion algorithm is applied.

[0138] g) "Correct 3D": The 3D correction algorithm is applied.

[0139] h) "Verify":

[0140] The following are example steps takes during the "Verify" process:

[0141] 1. Validation: As assessment of the resulting wireframe is made to determine if the wireframe conforms to the

above mentioned pitches and constraints to within a desired tolerance. If the model does not conform, the algorithm to returns to (d) or (e) until the wireframe is successfully validated or the algorithm determines that insufficient progress is being made toward converging to a valid model.

[0142] The following steps h(2) and h(3) may be optional since they depend on the result of the image registration process and could be omitted if the registration information is not provided as part of the input.

[0143] 2. Projection: The validated wireframe is transformed and projected onto each of the photographs registered into the coordinate system determined provided in (a), as described in U.S. patent application Ser. Nos. 12/467,244 and 12/467,250.

[0144] 3. Quality of Fit evaluation (shown in FIGS. **8**A and **8**B): The projected wireframes are evaluated for a quality of fit. This may be a combination of an automated quality of fit determination as well as a subjective visual assessment by a quality control technician. If quality of fit is not acceptable, the algorithm returns to step (b) following adjustments to the facet pitch values and/or edits to the wireframe.

[0145] FIG. **7** shows the noncorrected 3D wireframe dark lines, overlaid onto the rectified wireframe of FIG. **1**B, shown by the dashed lines of FIG. **7**. Note that the top-down view of this final 3D wireframe in FIG. **7** is not distinguishable from the output of the (2D) in-plane optimized wireframe in FIG. **4**. This is a desirable result for this simple roof example since it illustrates how the in-plane normalization produced a wireframe that correctly represented a top-down projection of what became the final 3D form. In situations of greater wireframe complexity, the correction in 3D may result in a repositioning of lines in the top-down projection that could not be predicted by the in-plane calculations.

Scenario 2

[0146] In scenario 2, the output is a corrected 2D wire frame instead of a 3D wire frame as in scenario 1 above. The sequence of steps is: "Wireframe", "Rectify", "Metadata", "Assign Pitch", and "Normalize". The steps of scenario 2 correspond to those steps of scenario 1 having the same labels.

Scenario 3

[0147] In scenario 3, the output is a corrected 3D wire frame. Scenario 3 uses a "rough" 3D model input needing minor corrections, such as a model that was already extruded by using the new registration algorithm mentioned in this document above. The sequence of steps is: "Wireframe", "Rectify", "Metadata", "Assign Pitch", "Normalize", "Extrude", "Correct3D", and "Verify". The steps of scenario 3 correspond to those steps of scenario 1 having the same labels. However, in the "AssignPitch" step of scenario 3, pitches are typically assigned by algorithms that generate a "rough" 3D wireframe. Also, the "Extrude" step of scenario 3 leverages existing elevation information to "seed" elevation placements of model points and/or lines, etc.

Scenario 4

[0148] Scenario 4 uses a "rough" 3D model input needing major corrections, such as a major pitch correction or after adding, removing, and/or changing major parts of the model. The sequence of steps is: "Wireframe", "Rectify", "Metadata", "Assign Pitch", "Normalize", "Extrude", "Correct3D", and "Verify". The steps of scenario 4 correspond to those steps of scenario 1 having the same labels. However, in the "AssignPitch" step of scenario 4, pitches are typically assigned by algorithms that that generate a "rough" 3D wireframe and/or are overridden by a user. Also, the "Extrude" step of scenario 4 may ignore existing elevation "seed" information where needed to complete a successful extrusion.

[0149] FIGS. **8**A and **8**B shows the 3D wireframe of FIG. **7** transformed and then projected back on the original source images. Note that the transformation process slightly distorts the appearance of overlayed wireframe to account for the 'lean' and other artifacts that were calculated to be in the original source photographs during the registration process. These distortions are most obvious at the gables of Primitive **1** and **3**.

[0150] By mapping the distortions back into the wireframe, a quality of fit (QOF) metric can be applied to judge the alignment of the wireframe against the object(s) it represents in the original source image.

[0151] In one embodiment a wireframe is generated via prior art Method 2 or 3, then correction is done in 3D as shown in FIGS. **5**-**7**.

[0152] This is an extension of Methods 2 and 3 whereby a "rough" 3D model resulting from a manual or automated wireframe generating process is optimized to a desired set of pitch and orientation constraints.

[0153] In yet another embodiment, the system will generate a metadata and a rectified 2D model, then normalize in-plane to achieve a final 2D or 3D wireframe.

[0154] In some situations, the content of the photograph(s) are insufficient to complete a 3D model without previous knowledge of the pitches of some or all of the roof facets. Furthermore, it may be preferable in some situations to defer the final application of pitch values to selection by an operator at a later time. In such situations, only steps 1, 2, and 3 as set forth under the heading of "Typical Usage Scenarios" above are carried out or, alternatively, step 3 is carried out on the in-plane normalized wireframe traced via prior art Method 1.

[0155] FIG. **9** is a schematic diagram of computing environment in which geometric correction of rough wireframe models derived from photographs may be implemented or of which it may be a part. For example, process **100** described above in conjunction with FIG. **1**A may be performed or implemented by, for example, one or more software modules or components or any combination of suitable hardware, firmware or software components or devices including those that are a part of, stored in, or configure the computing environment of FIG. **9**.

[0156] The computing environment **900** will at times be referred to in the singular herein, but this is not intended to limit the embodiments to a single device since in typical embodiments there may be more than one computer system or device involved. Unless described otherwise, the construction and operation of the various blocks shown in FIG. **9** are of conventional design. As a result, such blocks need not be described in further detail herein, as they will be understood by those skilled in the relevant art.

[0157] The computing environment **900** may include one or more processing units **912***a*, **912***b* (collectively **912**), a system memory **914** and a system bus **916** that couples various system components including the system memory **914** to the processing units **912**. The processing units **912** may be any logic processing unit, such as one or more central pro-

cessing units (CPUs) **912***a,* digital signal processors (DSPs) **912***b,* digital video or audio processing units such as coder-decoders (codecs) or compression-decompression units, application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), etc. The system bus **916** can employ any known bus structures or architectures, including a memory bus with memory controller, a peripheral bus, and a local bus. The system memory **914** includes read-only memory ("ROM") **918** and random access memory ("RAM") **920**. A basic input/output system ("BIOS") **922**, which can form part of the ROM **918**, contains basic routines that help transfer information between elements within the computing environment **900**, such as during start-up.

[0158] The computing environment **900** may include a hard disk drive **924** for reading from and writing to a hard disk **926** (including a solid state memory device), an optical disk drive **928** for reading from and writing to removable optical disks **932**, and/or a magnetic disk drive **930** for reading from and writing to magnetic disks **934**. The optical disk **932** can be a CD-ROM, while the magnetic disk **934** can be a magnetic floppy disk or diskette. The hard disk drive **924**, optical disk drive **928** and magnetic disk drive **930** may communicate with the processing unit **912** via the system bus **916**. The hard disk drive **924**, optical disk drive **928** and magnetic disk drive **930** may include interfaces or controllers (not shown) coupled between such drives and the system bus **916**, as is known by those skilled in the relevant art. The drives **924**, **928** and **930**, and their associated computer-readable storage media **926**, **932**, **934**, may provide nonvolatile and non-transitory storage of computer readable instructions, data structures, program modules and other data for the computing environment **900**. Although the depicted computing environment **900** is illustrated employing a hard disk **924**, optical disk **928** and magnetic disk **930**, those skilled in the relevant art will appreciate that other types of computer-readable storage media that can store data accessible by a computer may be employed, such as magnetic cassettes, flash memory, digital video disks ("DVD"), Bernoulli cartridges, RAMs, ROMs, smart cards, etc. For example, computer-readable storage media may include, but is not limited to, random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory, compact disc ROM (CD-ROM), digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, solid state memory or any other medium which can be used to store the desired information and which may be accessed by processing unit **912***a*.

[0159] Program modules can be stored in the system memory **914**, such as an operating system **936**, one or more application programs **938**, other programs or modules **940** and program data **942**. Application programs **938** may include instructions that cause the processor(s) **912** to perform directional geometric correction of rough wireframe models derived from photographs, and store and display input images or images generated by geometric correction of rough wireframe models. Other program modules **940** may include instructions for handling security such as password or other access protection and communications encryption. The system memory **914** may also include communications programs, for example, a Web client or browser **944** for permitting the computing environment **900** to access and exchange data including digital images of structures with sources such as Web sites of the Internet, corporate intranets, extranets, or other networks and devices, as well as other server applications on server computing systems. The browser **944** in the depicted embodiment is markup language based, such as Hypertext Markup Language (HTML), Extensible Markup Language (XML) or Wireless Markup Language (WML), and operates with markup languages that use syntactically delimited characters added to the data of a document to represent the structure of the document. A number of Web clients or browsers are commercially available such as those from Mozilla, Google, and Microsoft of Redmond, Wash.

[0160] While shown in FIG. **9** as being stored in the system memory **914**, the operating system **936**, application programs **938**, other programs/modules **940**, program data **942** and browser **944** can be stored on the hard disk **926** of the hard disk drive **924**, the optical disk **932** of the optical disk drive **928** and/or the magnetic disk **934** of the magnetic disk drive **930**.

[0161] An operator can enter commands and information into the computing environment **900** through input devices such as a touch screen or keyboard **946** and/or a pointing device such as a mouse **948**, and/or via a graphical user interface in order to receive, process, store and send data on which geometric correction of rough wireframe models derived from photographs has been or will be performed as described herein. Other input devices can include a microphone, joystick, game pad, tablet, scanner, etc. These and other input devices are connected to one or more of the processing units **912** through an interface **950** such as a serial port interface that couples to the system bus **916**, although other interfaces such as a parallel port, a game port or a wireless interface or a universal serial bus ("USB") can be used. A monitor **952** or other display device is coupled to the system bus **916** via a video interface **954**, such as a video adapter which may be configured to display images used by or generated by geometric correction of rough wireframe models derived from photographs. The computing environment **900** can include other output devices, such as speakers, printers, etc.

[0162] The computing environment **900** can operate in a networked environment using logical connections to one or more remote computers and/or devices. For example, the computing environment **900** can operate in a networked environment using logical connections to one or more other computing systems, mobile devices and other service providers or information servers that provide the digital images in various format or by other electronic delivery methods. Communications may be via a wired and/or wireless network architecture, for instance wired and wireless enterprise-wide computer networks, intranets, extranets, telecommunications networks, cellular networks, paging networks, and other mobile networks.

[0163] The various embodiments described above can be combined to provide further embodiments. All of the U.S. patents, U.S. patent application publications, U.S. patent applications, foreign patents, foreign patent applications and non-patent publications referred to in this specification and/or listed in the Application Data Sheet are incorporated herein by reference, in their entirety. Aspects of the embodiments can be modified, if necessary to employ concepts of the various patents, applications and publications to provide yet further embodiments.

[0164] These and other changes can be made to the embodiments in light of the above-detailed description. In general, in the following claims, the terms used should not be construed

to limit the claims to the specific embodiments disclosed in the specification and the claims, but should be construed to include all possible embodiments along with the full scope of equivalents to which such claims are entitled. Accordingly, the claims are not limited by the disclosure.

1. A method comprising:

assigning an initial set of pitch values to a plurality of facets of a roof;

automatically adjusting lateral positions of points associated with one or more first line segments of the roof identified as one or more ridge segments to accommodate pitch values of the initial set of pitch values associated with facets that share the one or more ridge segments or share endpoints of the one or more roof segments; and

automatically adjusting lateral positions of points associated with one or more second line segments of the roof identified as a junction or seam between adjacent facets of the roof to accommodate pitch values of the initial set of pitch values associated with the adjacent facets.

2. The method according to claim **1** further comprising repeating the automatically adjusting lateral positions of points associated with the one or more first line segments until the adjusting of lateral positions of points converges to a stable state.

3. The method according to claim **2** further comprising automatically repeating the automatically adjusting lateral positions of points associated with the one or more second line segments until the adjusting of lateral positions of points converges to a stable state.

4. The method according to claim **3** further comprising repeating the assigning an initial set of pitch values before repeating the automatically adjusting lateral positions of points associated with the one or more first line segments and before repeating the automatically adjusting lateral positions of points associated with the one or more second line segments.

5. The method according to claim **1**, further comprising assigning an initial set of orientations to the plurality of roof facets.

6. The method according to claim **1**, further comprising automatically generating a first 3D wireframe of the roof based on the adjusted lateral positions of points associated with the one or more first line segments and the adjusted lateral positions of points associated with the one or more second line segments.

7. The method according to claim **6** wherein the automatically generating the first 3D wireframe is further based on and initial set of input roof elevation values.

8. The method according to claim **7** further comprising automatically generating a second 3D wireframe of the roof to satisfy a set of constraints by changing geometric characteristics of the first 3D wireframe that are inconsistent with the constraints.

9. The method according to claim **8** wherein the changing geometric characteristics of the first 3D wireframe that are inconsistent with the constraints includes one or more of the following: preserving a footprint of the first 3D wireframe; correcting angles of intersection between line segments of the first 3D wireframe; straightening line segments of the first 3D wireframe; specifying a particular pitch for one or more facets of the roof; allowing a pitch of one or more facets of the roof to conform with other constraints; enforcing relationships between line segments and facets of the first 3D wireframe to satisfy symmetry requirements; constraining groups of points or line segments of the first 3D wireframe to reside at a same elevation; constraining two or more different groups of points or line segments of the first 3D wireframe to maintain a particular difference in elevation; enforcing symmetry between nearly symmetric objects in the first 3D wireframe.

10. The method of claim **8** wherein the automatically generating a second 3D wireframe of the roof to satisfy a set of constraints by changing geometric characteristics of the first 3D wireframe that are inconsistent with the constraints includes using metadata regarding relationships between roof line segments and points.

11. The method according to claim **1**, further comprising deriving metadata from an initial wireframe of the roof, the metadata including information regarding one or more of the following: nearest neighbor points of the points associated with the one or more first line segments, nearest neighbor line segments of the one or more first line segments, nearest neighbor points of the points associated with the one or more second line segments, nearest neighbor line segments of the one or more second line segments, roof line segment membership in polygons, whether line segments of the roof are edges or ridges, which points resemble roof hip junctions of the roof, which points resemble roof gable junctions of the roof, which points resemble roof corners on a perimeter of the roof, or identification of roof geometric shape primitives;

wherein the metadata is used in one or both of: the automatically adjusting lateral positions of points associated with one or more first line segments, and the automatically adjusting lateral positions of points associated with one or more second line segments.

12. The method according to claim **11** wherein the initial wireframe is a 2D wireframe.

13. The method according to claim **11** wherein the initial wireframe is a 3D wireframe.

14. The method according to claim **1** wherein the assigning an initial set of pitch values to a plurality of facets of a roof comprises automatically analyzing neighboring polygons of the roof to infer relative pitch values.

15. The method according to claim **1** wherein the initial set of pitch values are automatically assigned based on received input pitch values from a user.

16. A method comprising:

generating metadata from a wireframe model of a roof;

moving a location of roof junction points from the wireframe model to a normalized roof outline based on the generated metadata;

modifying a shape of the roof facets of the wireframe model; and

outputting a revised wireframe model based on the moved junction points and modified shapes to more closely approximate the actual shape of the roof.

17. The method according to claim **16** wherein the modifying the shape of the roof facets of the wireframe model includes:

comparing the roof facets of the wireframe model to a database of roof geometric shape primitives; and

modifying the shape of at least one of the roof facets of the wireframe model to match the geometric shape of a previously stored primitive.

18. The method according to claim **17** where in one of the primitives is a hip or gable.

19. The method according to claim **16** wherein the modifying the shape of the roof facets of the wireframe model is based on the generated metadata.

20. The method according to claim **16** wherein the wireframe model is a 2D top view of the roof.

21. The method according to claim **20** further including:

generating a 3D wireframe of the roof based on the revised wireframe model.

22. A non-transitory computer readable medium having computer executable instructions thereon that, when executed, cause a computer processor to perform the following:

automatically adjusting lateral positions of points associated with one or more first line segments of a roof identified as one or more ridge segments to accommodate pitch values of an initial set of pitch values associated with facets that share the one or more ridge segments or share endpoints of the one or more roof segments; and

automatically adjusting lateral positions of points associated with one or more second line segments of the roof identified as a junction or seam between adjacent facets of the roof to accommodate pitch values of the initial set of pitch values associated with the adjacent facets.

23. The method according to claim **22** wherein the computer executable instructions thereon, when executed, further cause the computer processor to perform automatically generating a first 3D wireframe of the roof based on the adjusted lateral positions of points associated with the one or more first line segments and the adjusted lateral positions of points associated with the one or more second line segments.

24. The method according to claim **23** wherein the generating the first 3D wireframe is further based on and initial set of input roof elevation values.

25. The method according to claim **23** wherein the computer executable instructions thereon, when executed, further cause the computer processor to perform automatically generating a second 3D wireframe of the roof to satisfy a set of constraints by changing geometric characteristics of the first 3D wireframe that are inconsistent with the constraints.

26. A non-transitory computer readable medium having computer executable instructions thereon that, when executed, cause a computer processor to perform the following:

generating metadata from a wireframe model of a roof, the wireframe model having roof junction points and roof facets;

moving a location of roof junction points from the wireframe model to a normalized roof outline based on the generated metadata;

modifying a shape of the roof facets of the wireframe model; and

outputting a revised wireframe model based on the moved junction points and modified shapes to more closely approximate the actual shape of the roof.

27. The non-transitory computer readable medium of claim **26** wherein the modifying the shape of the roof facets of the wireframe model includes:

comparing the roof facets of the wireframe model to a database of roof geometric shape primitives; and

modifying the shape of at least one of the roof facets of the wireframe model to match the geometric shape of a previously stored primitive.

28. The non-transitory computer readable medium of claim **26**, wherein the computer executable instructions thereon, when executed, further cause the computer processor to perform generating a 3D wireframe of the roof based on the revised wireframe model.

* * * * *