

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

REALD SPARK, LLC,
Patent Owner.

Case No. IPR2023-01046

Patent No. 10,740,985

PETITION FOR *INTER PARTES* REVIEW OF

U.S. PATENT NO. 10,740,985 (CLAIMS 1-14 AND 16-19)

Mail Stop “PATENT BOARD”

Patent Trial and Appeal Board

U.S. Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	IDENTIFICATION OF CLAIMS AND GROUNDS.....	2
IV.	DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	3
V.	CERTIFICATION AND FEES	4
VI.	LEVEL OF SKILL	4
VII.	CLAIM CONSTRUCTION	5
VIII.	ARGUMENT.....	5
A.	GROUND #1: Ganin Anticipates Claims 1-3, 5, 10, 12-14, and 18	6
1.	Patent Owner’s Belated Disclosure of Ganin	6
2.	Claim 1	8
a.	____ 1[P].....	8
b.	____ 1[A].....	10
c.	____ 1[B].....	11
d.	____ 1[C].....	12
e.	____ 1[D].....	13
f.	____ 1[E]	18
g.	____ 1[F].....	22
h.	____ 1[G].....	25
3.	Claim 2	26
4.	Claim 3	30
5.	Claim 5	32
a.	____ 5[P] and 5[A]	32
b.	____ 5[B].....	33
6.	Claim 10	34
7.	Claim 12	34
8.	Claim 13	34

TABLE OF CONTENTS

(continued)

a.	____ 13[P].....	34
b.	____ 13[A].....	35
c.	____ 13[B].....	36
9.	Claim 14	37
10.	Claim 18	38
a.	____ 18[P] and 18[A-E]	38
b.	____ 18[F].....	38
B.	GROUND #2: Claims 16 and 19 are rendered obvious by Ganin in view of Liu.	38
1.	Claim 16	38
a.	____ 16[P].....	38
b.	____ 16[A-G].....	38
c.	____ 16[H].....	39
2.	Motivation to Combine Ganin with Liu	39
3.	Claim 19	41
a.	____ 19[P] and 19[A-E]	41
b.	____ 19[F-G]	41
C.	GROUND #3: Claims 1-3, 5-6, 8-14, and 16-19 are rendered obvious by the Chalom-Aslan Combination	41
1.	Motivation to Combine Chalom and Aslan	41
2.	Claim 1	47
a.	____ 1[P].....	47
b.	____ 1[A].....	50
c.	____ 1[B].....	51
d.	____ 1[C].....	53
e.	____ 1[D-E]	54
f.	____ 1[F-G]	59
3.	Claim 2	63
4.	Claim 3	65

TABLE OF CONTENTS

(continued)

5.	Claim 5	66
6.	Claim 6	67
7.	Claim 8	70
8.	Claim 9	70
9.	Claim 10	71
10.	Claim 11	71
11.	Claim 12	71
12.	Claim 13	72
	a. <u>13[P]</u>	72
	b. <u>13[A]</u>	72
	c. <u>13[B]</u>	73
13.	Claim 14	74
14.	Claim 16	75
	a. <u>16[P]</u>	75
	b. <u>16[A]-[C]</u>	77
	c. <u>16[D] and 16[F]</u>	77
	d. <u>16[E] and 16[G]</u>	78
	e. <u>16[H]</u>	78
15.	Claim 17	78
	a. <u>17[P] and 17[A-E]</u>	78
	b. <u>17[F]</u>	78
16.	Claim 18	79
	a. <u>18[P] and 18[A-E]</u>	79
	b. <u>18[F]</u>	79
17.	Claim 19	79
	a. <u>19[P] and 19[A-E]</u>	79
	b. <u>19[F]</u>	80
	c. <u>19[G]</u>	80

TABLE OF CONTENTS

(continued)

D.	GROUND #4: Claim 4 is rendered obvious by Chalom in view of Aslan and Grau.....	80
1.	Claim 4	80
2.	Motivation to Combine Chalom-Aslan with Grau	81
E.	GROUND #5: Claim 7 is rendered obvious by Chalom-Aslan and Pace.....	83
1.	Claim 7	83
2.	Motivation to Combine Chalom-Aslan with Pace	83
IX.	SECONDARY CONSIDERATIONS	85
X.	CONCLUSION.....	85

EXHIBIT LIST

EXHIBITS FILED BY PETITIONER	
EX-1001	U.S. Patent No. 10,740,985 to Eric Sommerlade and Alexandros Neophytou (“’985 Patent”)
EX-1002	Patent Prosecution History of U.S. Patent No. 10,740,985
EX-1003	Declaration of L. Karam Regarding Invalidity of U.S. Patent No. 10,740,985
EX-1004	Curriculum Vitae of Lina Karam
EX-1005	U.S. Patent No. 10,664,949 to Edmond Chalom and Or Shimshi (“Chalom”)
EX-1006	Ganin <i>et al.</i> “Deepwarp: Photorealistic image resynthesis for gaze manipulation.” <i>ECCV 2016: 14th European Conference on Computer Vision, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II (pp. 311-326)</i> . Springer International Publishing. (“Ganin”)
EX-1007	U.S. Patent Application Publication No. 2017/0132528 A1 to Aslan <i>et al.</i> (“Aslan”)
EX-1008	Webpage printout at https://link.springer.com/chapter/10.1007/978-3-319-46475-6_20 (DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation)

EX-1009	U.S. Patent Application Publication No. 2018/0158246 A1 to Oliver Grau and Daniel Pohl (“Grau”)
EX-1010	U.S. Patent No. 8,942,283 to Charles P. Pace (“Pace”)
EX-1011	U.S. Provisional Application No. 62/542,661 filed August 8, 2017 (“’661 Prov. Appl.”)
EX-1012	U.S. Provisional Application No. 62/543,587 filed August 10, 2017 (“’587 Prov. Appl.”)
EX-1013	Prosecution of European Patent Application No. 18844651
EX-1014	[Reserved]
EX-1015	[Reserved]
EX-1016	Liu <i>et al.</i> “Optical flow and principal component analysis-based motion detection in outdoor videos.” <i>Eurasip Journal on Advances in Signal Processing</i> , volume 2010, article id 680623, 6 pages, 2010. Hindawi Publishing Corporation. (“Liu”)
EX-1017	Webpage printout at https://www.reald.com/privacypolicy (RealD Privacy Policy)
EX-1018	Webpage printout at https://web.archive.org/web/20161231162955/https://www.skoltech.ru/en/2016/11/ckolteh-litsenziroval-tehnologiyu-obespecheniya-zritelnogo-kontakta-v-videokonferentsiyah-v-

	kompaniyu-realD/ (Skoltech Licenses Eye Contact Videoconferencing Technology to RealD)
EX-1019	Kononenko, Daniil. “Learnable Warping-Based Approach To Image Re-Synthesis With Application to Gaze Redirection.” (2017): 1-131. (Kononenko Doctoral Thesis)
EX-1020	Sotera Stipulation
EX-1021	Yang <i>et al.</i> “Eye gaze correction with stereovision for video-teleconferencing.” <i>ECCV 2002: 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28–31, 2002, Proceedings, Part II (pp. 479-494)</i> . Springer Berlin Heidelberg, 2002. (“Yang”)
EX-1022	Cham <i>et al.</i> “Analogous view transfer for gaze correction in video sequences.” <i>7th International Conference on Control, Automation, Robotics and Vision, Singapore, December 2-5, 2002, Proceedings (pp. 1415-1420)</i> . IEEE Xplore. (“Cham”)
EX-1023	Webpage printout at: https://ieeexplore.ieee.org/abstract/document/1234980 (Analogous view transfer for gaze correction in video sequences)
EX-1024	Wood <i>et al.</i> “Gazedirector: Fully articulated eye gaze redirection in video.” <i>arXiv: 1704.08763v1 [cs.CV]</i> , 27 April 2017.
EX-1025	Zitnick <i>et al.</i> “Manipulation of video eye gaze and head orientation for video teleconferencing.” <i>Technical Report MSR-TR-99-46, Microsoft Research, 16 June 1999</i> .

EX-1026	Webpage printout at: https://www.microsoft.com/en-us/research/publication/manipulation-of-video-eye-gaze-and-head-orientation-for-video-teleconferencing/ (Manipulation of Video Eye Gaze and Head Orientation for Video Teleconferencing – Microsoft Research)
EX-1027	Gemmell <i>et al.</i> “Gaze awareness for video-conferencing: A software approach.” <i>IEEE Multimedia</i> , volume 7, issue 4, October-December 2000 (pp. 26-35). (“Gemmell”)
EX-1028	Kononenko <i>et al.</i> “Learning to look up: Realtime monocular gaze correction using machine learning.” <i>CVPR 2015: IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , Boston, MA, USA, 7-12 June 2015 (pp. 4667-4675). (“Kononenko”)
EX-1029	Webpage printout at: https://github.com/BlueWinters/DeepWarp (GitHub - BlueWinters_DeepWarp_ Tensorflow implement for DeepWarp)
EX-1030	Aleš Hladnik, “Image Compression and Face Recognition: Two Image Processing Applications of Principal Component Analysis.” <i>International Circular of Graphic Education and Research</i> , No. 6, January 2013 (pp. 56-61).
EX-1031	Complaint: <i>RealD Spark, LLC v. Microsoft Corporation</i> , WDWA-2-22-cv-00942, Dkt. 1.
EX-1032	Hinton <i>et al.</i> Distilling the Knowledge in a Neural Network, arXiv:1503.02531v1 [stat.ML], March 2015. (“Hinton”)
EX-1033	C. Bucilua, R. Caruana, and A. Niculescu-Mizil. “Model Compression.” <i>KDD 2006: In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge</i>

	<i>Discovery and Data Mining, (KDD), Philadelphia, PA, USA, August 20-23, pages 535–541, 2006 (pp. 535–541).</i> (“Bucilua”)
EX-1034	[Reserved]
EX-1035	Fleet <i>et al.</i> , “Design and Use of Linear Models for Image Motion Analysis.” <i>International Journal of Computer Vision</i> , volume 36, issue 3, 2000 (pp. 171-193). Kluwer Academic Publishers. (“Fleet”)
EX-1036	Wulff <i>et al.</i> “Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers.” <i>CVPR 2015: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015 (pp. 120-130).</i> (“Wulff”)
EX-1037	Burt <i>et al.</i> “A Multiresolution Spline With Application to Image Mosaics.” <i>ACM Transactions on Graphics</i> , volume 2, issue 4, October 1983 (pp. 217-236). (“Burt”)
EX-1038	Giger <i>et al.</i> , “Gaze correction with a single webcam.” <i>ICME 2014: IEEE International Conference on Multimedia and Expo, Chengdu, China, July 14-18, 2014.</i> (“Giger”)
EX-1039	Ho <i>et al.</i> “Gaze correction using 3D video processing for videoconferencing.” <i>ChinaSIP 2015: China Summit and International Conference on Signal and Information Processing, Chengdu, China, July 12-15, 2015 (pp. 496-499).</i> (“Ho”)
EX-1040	Jain. <i>Fundamentals of Digital Image Processing.</i> Prentice-Hall, Inc., 1989 (excerpted). (“Jain”)

EX-1041	Vetterli <i>et al.</i> Wavelets and Subband Coding. Prentice-Hall, Inc., 1995 (excerpted). (“Vetterli”)
EX-1042	Dodge and Karam. Introduction to Machine Learning and Deep Learning: A Hands-On Starter’s Guide, 2017; revised 2019. Available online at https://deeplearningtextbook.org/ . (“Dodge”)
EX-1043	Goodfellow <i>et al.</i> Deep Learning, MIT Press, 2016. Available online at https://www.deeplearningbook.org/ (excerpted). (“Goodfellow”)
EX-1044	Murphy. Machine Learning: A Probabilistic Perspective, MIT Press, 2012. (“Murphy”)
EX-1045	Sonka <i>et al.</i> Image Processing, Analysis, and Machine Vision, 4 th edition, Cengage Learning, 2015.
Ex. 1046	U.S. Patent No. 10,043,088 (“Odry”)
EX-1047	Microsoft Computer Dictionary, fourth edition, 1999 (excerpted)
EX-1048	Internet Archive affidavit w exhibits
EX-1049	[Reserved]
EX-1050	Webpage printout at: https://link.springer.com/book/10.1007/978-3-319-46475-6 (Webpage Computer Vision – ECCV 2016)

EX-1051	Webpage printout at: https://asp-urasipjournals.springeropen.com/articles/10.1155/2010/680623 (SpringerOpen, Liu)
EX-1052	Webpage printout at: https://dl.acm.org/doi/abs/10.1155/2010/680623# (ACM, Liu)
EX-1053	U.S. Patent No. 9,338,449 to Joel Sole Rojals, Rajan Laxman Joshi, and Marta Karczewicz (“Rojals”)
EX-1054	U.S. Patent Application Publication No. 2008/0043833 A1 to Jane Perera and Joseph D. Young (“Perera”)
EX-1055	Joint Claim Construction and Prehearing Statement: <i>RealD Spark, LLC v. Microsoft Corporation</i> , WDWA-2-22-cv-00942, Dkt. 59.
EX-1056	George Wolberg, Digital Image Warping, IEEE Computer Society Press, 1990.
EX-1057	Webpage printout at: https://ieeexplore.ieee.org/document/7299098 (Learning to look up: Realtime monocular gaze correction using machine learning)
EX-1058	Webpage printout at: https://ieeexplore.ieee.org/document/6890306 (Gaze correction with a single webcam)

I. INTRODUCTION

Microsoft Corporation (“Petitioner”) requests *Inter Partes* Review (“IPR”) of claims 1-14 and 16-19 (the “Challenged Claims”) of U.S. Patent No. 10,740,985 (“’985 Patent,” EX-1001), purportedly owned by RealD Spark, LLC (“Patent Owner”).

II. MANDATORY NOTICES

Real Party-in-Interest: The real party-in-interest of this petition is Microsoft Corporation, located at One Microsoft Way, Redmond, WA 98052.

Related Matters: The ’985 Patent has been asserted against Petitioner in *RealD Spark, LLC v. Microsoft Corporation*, WDWA-2-22-cv-00942 (“District Court Case”). To the best of Petitioner’s knowledge, the ’985 Patent has not been asserted against other parties.

Lead Counsel: Alexander Stein (Reg. No. 71,397); Back-up Counsel: Andrew Devkar (Reg. No. 76,671).

Service: Service of any documents may be made at Morgan, Lewis & Bockius LLP, 1301 Second Ave., Suite 2800, Seattle, WA 98101 (Telephone: 206.274.6400; Fax: 206.274.6401).

Petitioner consents to e-mail service at: 985Patent-Microsoft-IPR@morganlewis.com.

III. IDENTIFICATION OF CLAIMS AND GROUNDS

'985 Patent: The application was filed on **August 7, 2018** and claims priority to Provisional Application No. 62/542,661, filed on August 8, 2017, and Provisional Application No. 62/543,587, filed on August 10, 2017. It is subject to the AIA provisions of 35 U.S.C. §102. Petitioner assumes only for this proceeding that the earliest effective filing date for claims 1-14 and 16-19 is August 8, 2017.

Ganin: “Deepwarp: Photorealistic image resynthesis for gaze manipulation” authored by Yaroslav Ganin *et al.* (“Ganin,” EX-1006) was published at least as early as **September 17, 2016** (EX-1008, EX-1048), and is prior art under §102(a)(1).

Chalom: U.S. Patent No. 10,664,949 titled “Eye contact correction in real time using machine learning” to Chalom *et al.* (“Chalom,” EX-1005), was filed **April 22, 2016** and granted May 26, 2020, and is prior art under §102(a)(2).

Aslan: U.S. Patent Application Publication No. 2017/0132528 titled “Joint model training” to Ozlem Aslan *et al.* (“Aslan,” EX-1007) was filed June 28, 2016 and published **May 11, 2017**, and is prior art under §102(a)(1).

Grau: U.S. Patent Application Publication No. 2018/0158246 to Grau *et al.* (“Grau,” EX-1009) is based on an application filed **December 7, 2016** so qualifies as prior art under §102(a)(2).

Pace: U.S. Patent No. 8,942,283 to Pace (“Pace,” EX-1010) issued on

January 27, 2015 (as well as earlier filing and priority dates) and based on the issue date alone qualifies as prior art under §102(a)(1).

Liu: Liu is an article titled “Optical Flow and Principal Component Analysis-Based Motion Detection in Outdoor Videos” (“Liu,” EX-1016) that published on **March 8, 2010** (EX-1048), so qualifies as prior art under §102(a)(1).

Applying these references as shown in the below Grounds, Petitioner requests that the Board find each of the Challenged Claims invalid:

Ground 1: Ganin anticipates claims 1-3, 5, 10, 12-14, and 18;

Ground 2: Claims 16 and 19 are obvious over Ganin in view of Liu.

Ground 3: Claims 1-3, 5-6, 8-14, and 16-19 are obvious over Chalom in view of Aslan;

Ground 4: Claim 4 is obvious over Chalom in view of Aslan and Grau; and

Ground 5: Claim 7 is obvious over Chalom in view of Aslan and Pace.

IV. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

Ganin is the only reference asserted here that was cited during prosecution. EX-1002, p.40. However, as explained in Section VIII.A.1, the Examiner was led by Patent Owner to overlook Ganin’s relevance and therefore error in a manner material to the patentability of many of the Challenged Claims.

All other references were not considered during prosecution, nor are they

cumulative of art or arguments presented during prosecution since the Chalom-Aslan combination raised in this Petition (and the underlying motivation theories) was never raised during prosecution. For at least the above reasons, a discretionary denial under 325(d) would be unwarranted.

Furthermore, as to 314(a), a discretionary denial would also be inappropriate. Following the guidance that “the PTAB will not discretionarily deny institution in view of parallel district court litigation” if a *Sotera* stipulation is filed (Memorandum on “Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation” issued on June 21, 2022), Petitioner herewith provides such a *Sotera* stipulation (EX-1020).

V. CERTIFICATION AND FEES

Petitioner certifies the '985 Patent is available for IPR and that Petitioner is not barred or estopped from requesting this IPR on the grounds identified herein.

Any additional fees for this IPR may be charged to Deposit Account No. 50-0310 (Order No. 117139-0008).

VI. LEVEL OF SKILL

A person having ordinary skill in the art (“PHOSITA”) at the time of the alleged invention would have had at least an undergraduate degree in computer science, electrical engineering, computer engineering, or equivalent education, and at least two years of work experience using machine learning for image processing.

Additional education might substitute for less work experience, and *vice versa*.

EX-1003, ¶22.

VII. CLAIM CONSTRUCTION

Petitioner applied plain and ordinary meanings of all terms and no express constructions are required here.¹

VIII. ARGUMENT

The Challenged Claims relate to eye-gaze correction, including training machine-learning models using pairs of input images (e.g., before and after gaze correction) and using the output of a first machine-learning algorithm (“MLA”) to train a second MLA. As explained below, this was unpatentable as of August 2017 due to (i) Ganin’s anticipating disclosures of machine-learning-based coarse and fine warping modules (i.e., two different MLAs) for gaze-correction purposes (*infra* Section VIII.A) and (ii) Chalom’s use of an MLA to perform gaze correction and the motivation to modify Chalom to employ the teacher-student-based machine-learning framework of Aslan to train a second MLA (*infra* Section VIII.C).

¹ Petitioner proposed constructions in the District Court Case, but those are not material here (i.e., the prior art teaches all limitations with or without these constructions). EX-1055.

A. GROUND #1: Ganin Anticipates Claims 1-3, 5, 10, 12-14, and 18

1. Patent Owner's Belated Disclosure of Ganin

Ganin was cited, but only at a late stage and Patent Owner never revealed the extent of its involvement with Ganin's authors (and their associated university, Skoltech), nor ever explained its obvious relevance.

Specifically, in 2016 (before the '985 Patent's alleged earliest effective filing date), RealD Inc. ("RealD"),² obtained an exclusive license for the eye-contact videoconferencing technology developed by Skoltech. EX-1018.

Similarly, Ganin was published by a Skoltech group concerning eye-contact videoconferencing technology. According to Eric Sommerlade, the first inventor of the '985 Patent and Patent Owner's former employee, "Skoltech has the most elegant solution" to "the eye contact problem with video conferencing." *Id.*, p. 2.

Thus, Patent Owner was aware of Ganin as early as 2016. In addition to the licensing collaboration in 2016, RealD was interested in Dr. Lempitsky (an author of Ganin) and his team's gaze-correction research at least through September 2017. For example, Daniil Kononenko's (Dr. Lempitsky's then-PhD student and another Ganin author) doctoral thesis stated: "I thank Eric Sommerlade and his team at RealD for the interest shown in our work..." EX-1019, p.ii.

² RealD was formed by Patent Owner. EX-1031, ¶3; EX-1017.

Despite this 2016 knowledge of Ganin and discussing another Skoltech paper in the '985 Patent itself (EX-1001, 9:6-11), Patent Owner did not disclose Ganin to the Examiner until 2020 (EX-1002, p.40)—burying this relevant reference under 44 other references less than one month before a Notice of Allowance was issued (EX-1002, p.12).

The IDS itself shows that Patent Owner “retrieved” Ganin in January 2018 (EX-1002, p.40), yet it waited until 2020 to disclose it to the Examiner, while failing to ever inform the Examiner of the relevant background business relationship, let alone of Ganin’s material relevance. This material relevance is confirmed by Patent Owner’s actions before the European Patent Office as similar independent claims to those in the '985 Patent were rejected based on Ganin, and Patent Owner added limitations related to compressing editing instructions using a basis set, which are not present in any of the Challenged Claims. EX-1013, pp.214, 242. Thus, Patent Owner’s actions led the Examiner to overlook Ganin and therefore error in a manner material to the patentability of many of the Challenged Claims.

2. Claim 1³

a. 1[P]

If the preamble is limiting. Ganin discloses it by teaching a training process that generates “reference data for adjusting a digital representation of a head region,” e.g., for gaze correction. EX-1003, ¶¶97-102.

The “core of [Ganin’s] approach is the prediction of the **warping field** using a deep convolutional network.” EX-1006, p.13.⁴ Ganin’s proposed system is in its Figure 2, reproduced below.

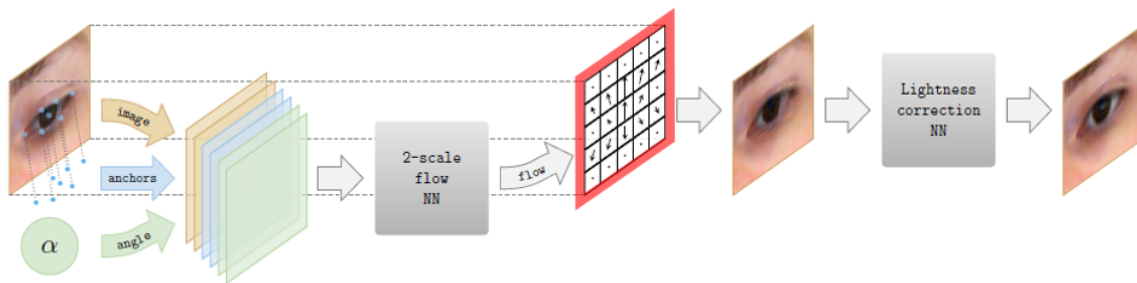


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2 (red frame added). As shown, given an input eye region (“image,”

³ A claim listing is at the end of this document.

⁴ Unless indicated otherwise, Petitioner added all coloring in Figures and emphases in quotations.

brownish) and other required information, the proposed “multi-scale neural network” (“2-scale flow NN,” gray) predicts a “warping field” (red)/“flow field”/“pixel-flow field”/“two-channel map,”⁵ which “is then **applied to the input image** to produce an image of a **redirected eye**.” EX-1006, p.4.

Ganin explains that “[t]he architecture is **trained** end-to-end in a supervised way using a specially collected dataset that depicts the change of the appearance under gaze redirection in real life.” EX-1006, p.2. Specifically, Ganin’s training process takes “**training pairs**... [of] two images with different gaze directions,” and passes them through the “multi-scale neural network” such that the weights of the nodes within the “multi-scale neural network” are optimized by minimizing the “ ℓ_2 -distance between the synthesized output O_{output} [e.g., gaze corrected eye regions, such as eye regions of the first images of the training pairs **warped by the predicted warping fields**] and the ground-truth O_{gt} [e.g., target eye regions, such as eye regions of the second images of the training pairs].” EX-1006, p.8.

In view of (i) Ganin’s training process comparing warped eye regions with target eye regions and (ii) the trained “multi-scale neural network” being able to

⁵ Ganin uses “warping field,” “flow field,” “pixel-flow field,” “two-channel map” interchangeably, so these terms are used interchangeably here as to Ganin. EX-1003, ¶99.

generate a warping field given an eye region input, Ganin's warping fields are expressly taught or, at a minimum, necessarily generated during the training process, which thus discloses the claimed "reference data." EX-1003, ¶101; *infra* Sections VIII.A.2.e-VIII.A.2.f.

Therefore, Ganin discloses "[a] method of generating reference data [e.g., warping fields generated during training; "first reference data" in 1[D]; "second reference data" in 1[E]] for adjusting a digital representation of a head region [e.g., gaze correction]." EX-1003, ¶¶97-102.

b. 1[A]

Ganin discloses that its proposed multi-scale neural network receives three types of input data: (1) an input eye region ("image," brownish), (2) feature points ("anchors," blue), and (3) a correction angle ("angle," green).

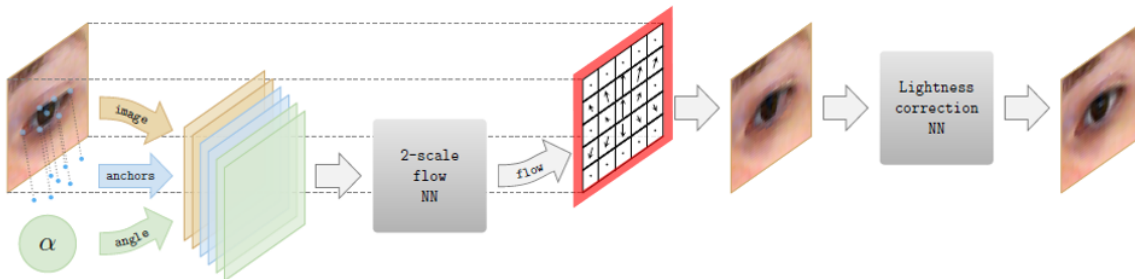


Fig. 2. The **proposed system** takes an input eye region, feature points (**anchors**) as well as a correction **angle** α and sends them to the multi-scale neural network (see Section 3.2) predicting a **flow** field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2 (red frame added), Fig.10(a). Since Ganin's neural networks are

“trained end-to-end in a supervised way,” the training of the NNs also includes these three types of inputs, plus the ground-truth eye regions. EX-1003, ¶103. As Ganin explains: “[t]he model is **trained on pairs of images** corresponding to eye appearance **before and after the redirection.**” EX-1006, pp.2, 4, 5 (referencing “**pairs of images**”). Ganin discloses using image pairs “as a **training example for the learning procedure** (Figure 4-Right).” EX-1006, p.5.



Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

EX-1006, Fig.4; EX-1003, ¶103.

Therefore, Ganin discloses “receiving training data [e.g., eye regions extracted from image pairs].” EX-1003, ¶¶103-104.

c. 1[B]

Ganin discloses that “[t]he model is **trained on pairs of images** corresponding to eye appearance **before and after the redirection.**” EX-1006, p.4. Ganin further discloses extracting the eye region by aligning the eyes, “comput[ing] a tight axis-aligned bounding box B' of the points in the **input image,**” and “cut[ting] out the interior of the estimated box from the **input image,**

and also from the **output image** of the pair (using exactly the same bounding box coordinates).” EX-1006, p.5. Thus, “[t]he resulting image pair serves as a **training example for the learning procedure** (Figure 4-Right).” *Id.* Thus, all training images, including the input images and the output images, would have contained eye regions and the eye region extraction process is performed for each image pair.

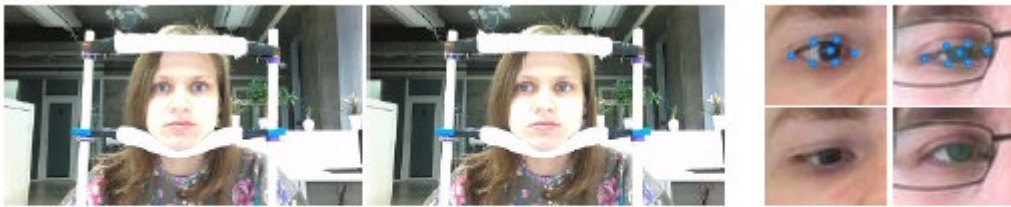


Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

EX-1006, Fig.4; EX-1003, ¶105.

Therefore, Ganin discloses “a set of input patches [e.g., extracted eye regions of input images corresponding to eye appearance before gaze redirection], each input patch comprising a target feature [e.g., eye region] of a digital representation [e.g., image] of a head region prior to adjustment [e.g., before gaze redirection] of the digital representation of the head region, wherein the target feature is the same for each input patch.” EX-1003, ¶¶105-106.

d. 1[C]

Ganin discloses that “[t]he model is **trained on pairs of images**

corresponding to eye appearance before and **after the redirection.**” EX-1006, p.4. “The head pose, the lighting, and all other nuisance parameters are (approximately) the **same between the two images in the pair.**” EX-1006, p.5; EX-1003, ¶107. Therefore, Ganin’s output images (and eye regions thereof) are in one-to-one correspondence with the input images (and the eye regions thereof). EX-1003, ¶108.

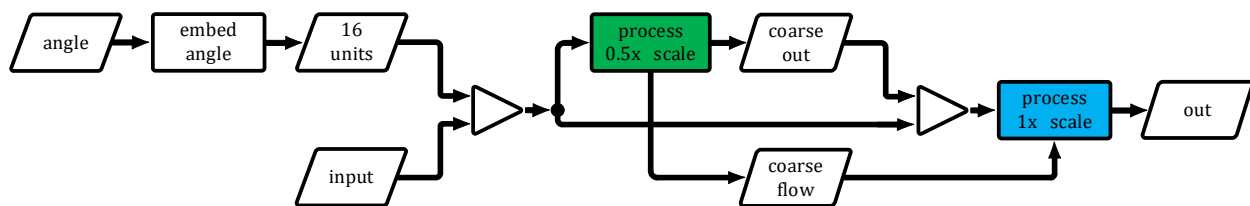
This one-to-one correspondence is further confirmed by Ganin’s process for extracting the eye region by aligning the eyes (EX-1006, p.5), because in order to “cut out the interior of the estimated box [i.e., eye region] from the output image of the pair []using exactly the same bounding box coordinates” (*id.*), the output image necessarily has an eye region located at approximately the same location within the output image as the corresponding eye region in the input image. EX-1003, ¶109.

Therefore, Ganin discloses “a set of output patches [e.g., extracted eye regions of input images corresponding to eye appearance after gaze redirection] in one-to-one correspondence with the input patches, each output patch comprising the target feature [e.g., eye region] of the digital representation of the head region after adjustment [e.g., after gaze redirection] of the digital representation of the head region.” EX-1003, ¶¶107-110.

e. 1[D]

Ganin proposes “warping the input image.” EX-1006, p.4. The warping

field to warp the image “is predicted in two stages in a coarse-to-fine manner,” and the “coarse warping” is implemented by a $0.5\times$ -scale processing module (green below) and the “fine warping” is implemented by a $1\times$ -scale processing module (blue below). Each of these “warping modules” warps the input image by predicting a pixel-flow field and applying the predicted pixel-flow field to the input image. EX-1006, p.4, Fig.10 caption.



(a) High-level pipeline

EX-1006, Fig.10(a).⁶ According to Ganin, each of the two processing modules is a MLA, comprising a neural network (“NN”) having “convolutional layer[s].” See, e.g., EX-1006, p.4 (“multi-scale neural network”); p.17 (“**The architectures of [the convolutional neural network within] the two warping modules are same (modulo the number of input maps) and are shown in Figure 10(d).**”); p.5 (“**Coarse warping. The last convolutional layer of the first (half-scale) warping module produces a pixel-flow field (a two-channel map)**”); p.6 (“**Fine warping. ... $1\times$ -**

⁶ Coloring for Ganin’s Figures 10(a), (b), and (c) is added by Petitioner unless otherwise noted.

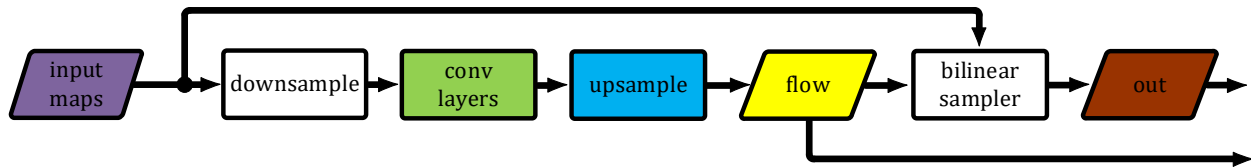
scale network”).



(d) Convolutional layers

EX-1006, Fig 10(d) (illustrating the convolutional layers for both warping modules).

Ganin’s coarse warping module, or $0.5\times$ -scale processing module, is further detailed in Fig.10(b), reproduced below.



(b) $0.5\times$ -scale processing module

EX-1006, Fig.10(b).

As shown, Ganin’s coarse warping module generates two outputs. One is the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ (yellow). “The last convolutional layer [green] of the first (half-scale) warping module produces a pixel-flow field (a two-channel map), which is then up-sampled [blue] [to generate the coarse pixel-flow field] $\mathbf{D}_{\text{coarse}}(I, \alpha)$ and applied to warp the input image.” EX-1006, p.5. The pixel-flow field instructs how pixels of the input image are warped and is therefore an “editing instruction[] for adjusting the digital representation of the head region” as required by 1[D]. *C.f.* EX-1001, 9:13-29 (“editing instructions... could be the

output of a convolution of the patch...”). The coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ instructs how the pixels of the input image may be warped to generate the other output of the coarse warping module “coarse estimate O_{coarse} ” (brown):

$$O_{\text{coarse}} = \mathbf{S}(I, \mathbf{D}_{\text{coarse}}(I, \alpha)) . \quad (1)$$

EX-1006, p.5, Eq.1. Therefore, both the “pixel-flow field” and the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ up-sampled therefrom are “editing instructions for adjusting the digital representation of the head region” required by 1[D]. EX-1003, ¶114.

During both training and runtime, Ganin’s coarse warping module receives “input maps,” which include “input images” before gaze redirection, “the desired redirection angle,” and “feature points” generated “from the raw input pixels.”

EX-1006, p.6. During training, Ganin teaches using “a regular ℓ_2 -distance between the synthesized output O_{output} and the ground-truth O_{gt} as the objective function.” EX-1006, p.8. A PHOSITA would have understood that the

synthesized output O_{output} could refer to both coarse estimate O_{coarse} (output of the coarse warping module) and final output O (output of the fine warping module).

Accordingly, the ground-truth O_{gt} refers to the “output images” having redirected gazes, which are provided to both the coarse warping module (to calculate the ℓ_2 -distance between coarse estimate O_{coarse} and O_{gt}) and the fine warping module (to calculate the ℓ_2 -distance between final output O and O_{gt}) during training. Thus,

Ganin's coarse warping module is trained using the claimed "training data" that includes both "input images" and "output images." EX-1003, ¶¶115-117.

In order to generate "coarse estimate O_{coarse} " during training, the corresponding pixel-flow fields and coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ up-sampled therefrom are expressly taught or, at a minimum, necessarily generated. EX-1003, ¶118.

Moreover, as explained in Section VIII.A.2.f *infra*, Ganin teaches that the fine warping module, or $1\times$ -scale processing module, receives as part of its input the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated from the training data. EX-1003, ¶¶119-120.

Finally, Ganin's editing instructions generated during training of the coarse warping module provide adjustments "for a range of possible digital representations of the head region." EX-1003, ¶121.

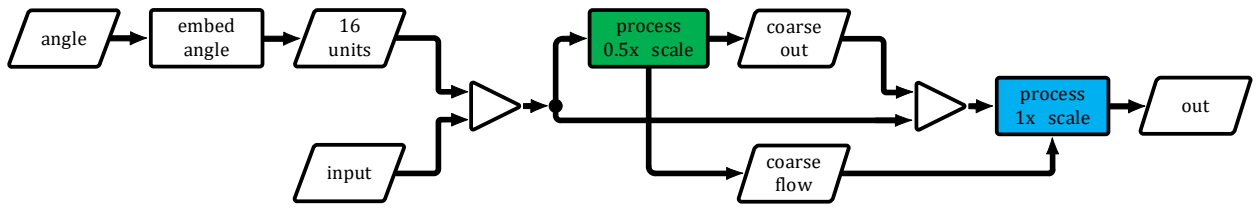
When preparing the training data, Ganin discloses that "[f]or each person we record 2 – 10 sequences, **changing the head pose** and light conditions **between different sequences**. Training pairs are collected, **taking two images** with different gaze directions **from one sequence**." EX-1006, p.8. After "exclud[ing] bad shots," the ultimate training data includes a "dataset of 33 persons and 98 sequences." *Id.* Since there are more sequences than persons, the training data expressly (or at a minimum, necessarily) includes multiple image pairs of the same

person taken from different sequences, i.e., having different head poses. EX-1003, ¶122; EX-1006, p.9 (“**Sampling training pairs.** We found that biasing the selection process for **more difficult and unusual head poses** and bigger redirection angles improved the results.”); p.13 (“different head poses”).

Therefore, Ganin discloses “using a first machine learning algorithm [e.g., coarse warping module including the $0.5\times$ -scale neural network] to generate first reference data [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] using the training data [e.g., “input maps” including input images, and output images as ground truth], the first reference data comprising editing instructions [e.g., pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ up-sampled therefrom] for adjusting the digital representation of the head region [e.g., gaze redirection] for a range of possible digital representations of the head region [e.g., different head poses].” EX-1003, ¶¶111-123.

f. 1[E]

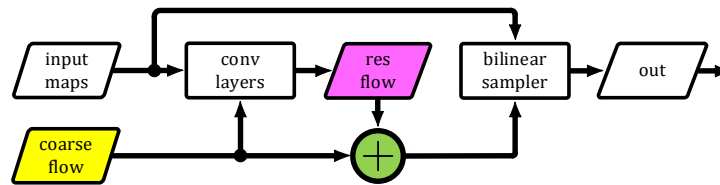
As mentioned above, Ganin proposes generating warping fields (i.e., editing instructions) in a “coarse-to-fine manner,” which includes a $0.5\times$ -scale coarse warping module (MLA1) and a $1\times$ -scale fine warping module (MLA2). EX-1006, pp.4-6.



(a) High-level pipeline

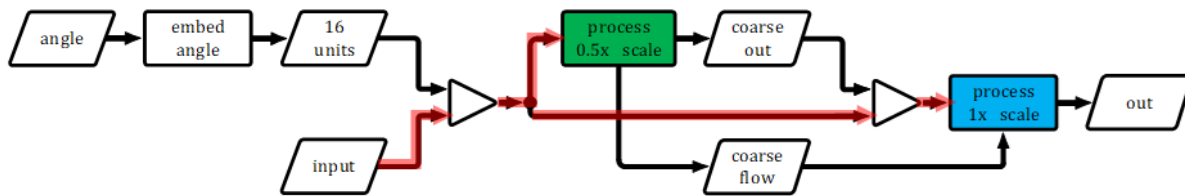
Id., Fig.10(a).

Ganin's fine warping module is further detailed in Fig.10(c), reproduced below.



(c) 1×-scale processing module

EX-1006, Fig.10(c). As shown, Ganin's fine warping module takes two inputs during runtime. First is the input maps. This is also confirmed by the "high-level pipeline" illustrating both NN-based warping modules:



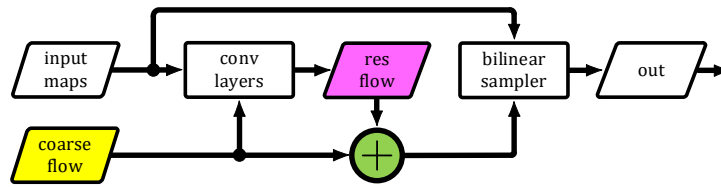
(a) High-level pipeline

EX-1006, Fig.10(a) (showing "input" being received (red) by both 0.5×-scale processing module (green) and 1×-scale processing module (blue)). The other

input to Ganin's $1\times$ -scale NN is the coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$. These two inputs are provided to Ganin's fine warping module during both runtime and training. EX-1003, ¶125.

In addition, Ganin teaches using “a regular ℓ_2 -distance between the synthesized output O_{output} and the ground-truth O_{gt} as the objective function.” EX-1006, p.8. The ground-truth O_{gt} refers to the “output images” having redirected gazes, which are provided to the fine warping module during training. Thus, Ganin's fine warping module is trained using the claimed “training data” that includes both “input images” and “output images.” EX-1003, ¶126.

Ganin further discloses that the fine ($1\times$ -scale) warping module's NN predicts “residual flow” $\mathbf{D}_{\text{res}}$ (pink), “another two-channel flow” that is added (green) to the coarse warping field $\mathbf{D}_{\text{coarse}}$ (yellow) to obtain the final predicted warping field $\mathbf{D}(I, \alpha)$.



(c) $1\times$ -scale processing module

EX-1006, Fig.10(c). This addition is reflected here:

$$\mathbf{D}(I, \alpha) = \mathbf{D}_{\text{coarse}}(I, \alpha) + \mathbf{D}_{\text{res}}(I, \alpha, O_{\text{coarse}}, \mathbf{D}_{\text{coarse}}(I, \alpha)), \quad (3)$$

EX-1006, Eq. 3. Such final warping field $\mathbf{D}(I, \alpha)$ is used to warp the original

image “to obtain the final output”:

$$O = S(I, D(I, \alpha)) . \quad (4)$$

EX-1006, p.6, Eq.4. Thus, both the “residual flow” $\mathbf{D}_{\text{res}}$ and the final warping field $\mathbf{D}(I, \alpha)$ are “for adjusting the digital representation of the head region” and are generated by the fine warping module during both runtime and training. EX-1003, ¶¶127-128.

For reasons explained in Section VIII.A.2.e, Ganin’s editing instructions generated during training of the fine warping module also provide adjustments “for a range of possible digital representations of the head region” because both the coarse and fine warping modules use the same training data having different poses. EX-1003, ¶129.

Therefore, Ganin discloses “using a second machine learning algorithm [e.g., fine warping module including the $1\times$ -scale NN] to generate second reference data [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] using the same training data as the first machine learning algorithm [e.g., “input maps” including input images, and output images as ground truth] and the first reference data generated by the first machine learning algorithm [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$], the second reference data comprising editing instructions [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] for adjusting

the digital representation of the head region [e.g., gaze redirection] for a range of possible digital representations of the head region [e.g., different head poses].”

EX-1003, ¶¶124-130.

g. 1[F]

Ganin discloses that its pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training are “for a range of possible configurations of the target feature.” EX-1003, ¶131. When collecting training data, Ganin “record[s] several images with eyes looking in different fixed directions” and therefore resulting in a “continuously varying [] angle.” EX-1006, p.8. Moreover, since Ganin’s trained system is able to perform gaze redirection for a “redirection angle... selected **arbitrarily from a certain range**” (EX-1006, Abstract), the training data necessarily includes images with different gaze directions. Accordingly, the “first editing instructions,” e.g., pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training using such training data would be “for a range of possible configurations of the target feature,” e.g., different gaze directions. EX-1003, ¶131.

Ganin also discloses that “the first reference data comprise... first selection instructions for selecting editing instructions for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch.” As discussed in Section VIII.C.2.f *infra*, the recitations of intended use are

not limiting; but to the extent they are, Ganin discloses them. EX-1003, ¶132.

Specifically, the training of Ganin's coarse warping module generates not only coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ but also the trained 0.5×-scale NN, represented by the weights of neurons of the 0.5×-scale NN. During runtime, the trained 0.5×-scale NN receives an input patch having an eye region before gaze redirection and uses these weights of neurons to generate a corresponding coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ for redirecting the input gaze by a specified angle. EX-1006, p.5; EX-1003, ¶133.

In addition, the pixel-flow fields and the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated **during runtime** are **selected from** the first editing instructions, i.e., the pixel-flow fields and the coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated **during training** and which are typically represented by a warping-field generating function that is learned by the NN. EX-1003, ¶134. This is because during training, the training set covers a wide range of eye regions and to-be-warped pixels, and the corresponding coarse warping fields generated for each of the wide range of eye regions cover all the possible coarse warping fields for various input eye regions. During training, the NN learns the corresponding warping-field generating function that covers all possible warping fields for the considered task. During runtime, given a particular input eye region, the machine-learning-based module (the coarse warping module specifically) selects a coarse warping field

from among all of the possible coarse warping fields. EX-1003, ¶134.

As shown in Figure 2 below, Ganin discloses generating feature maps and anchor maps for an input image, which are concatenated with the RGB channels of the input image to form an “input map stack” (brownish/blue/green, i.e., configuration of the target feature) for predicting the flow field for the input image.

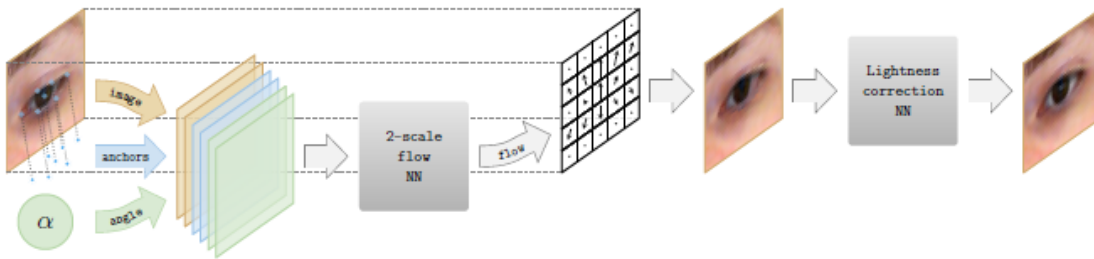


Fig. 2. The proposed system takes an input eye region, feature points (**anchors**) as well as a correction **angle** α and sends them to the multi-scale neural network (see Section 3.2) predicting a **flow** field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2 (colors original). In other words, Ganin’s configuration of the target feature of each input image is represented by the corresponding input map stack. EX-1003, ¶¶135-136.

Therefore, Ganin discloses that “the first reference data comprise first editing instructions [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] for a range of possible configurations of the target feature [e.g., different gaze directions] and first selection instructions [e.g., trained 0.5 $\times$ -scale NN, represented by the weights of its neurons] for selecting editing instructions [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] for a particular input patch from the first editing instructions [e.g., coarse warping

fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training or representation thereof learned by the network] based on the configuration of the target feature [e.g., input map stack] of the input patch.” EX-1003, ¶¶131-137.

h. 1[G]

Ganin discloses that “[t]he architectures of [the NNs within] the two warping modules are same (modulo the number of input maps).” EX-1006, p.17, Fig.10(d). As explained earlier, both modules receive the same training data and both generate warping fields, only at different scales. Thus, for reasons explained in Section VIII.A.2.g, Ganin discloses that the residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ generated during training are also “for a range of possible configurations of the target feature.” Similar to the “first selection instructions” discussed in Section VIII.A.2.g, Ganin’s “second reference data comprise... second selection instructions [e.g., trained $1\times$ -scale NN, represented by the weights of its neurons] for selecting editing instructions [e.g., residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$] for a particular input patch from the second image editing instructions [e.g., residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training or representations thereof learned by the network] based on the configuration of the target feature [e.g., input map stack] of the input patch.” EX-1003, ¶¶138-139.

* * * *

Because Ganin discloses each and every limitation of claim 1, claim 1 is anticipated and therefore unpatentable. EX-1003, ¶140.

3. Claim 2

Claim 2 depends from claim 1 (which is anticipated for reasons provided above), and adds aspects (discussed in turn below) taught by Ganin. EX-1003, ¶141.

First, Ganin discloses “plural of local descriptors of the input patch.” EX-1003, ¶142. Ganin discloses “employ[ing] an external face alignment library” to produce “ $N = 7$ feature points $\{(x_i^{anchor}, y_i^{anchor}) \mid i = 1, \dots, N\}$ **for the eye** (six points along the edge and also the pupil center).” EX-1006, p.5. Thus, these feature points (or anchor points) are “information about a local [eye] region” and therefore are “local descriptors.” EX-1003, ¶¶143-145.

The '985 Patent also identifies “a difference between the pixels at two points” to be a local descriptor. EX-1001, 9:57-60. Similarly, Ganin discloses “anchor maps,” where “for each... feature point located at $(x_i^{anchor}, y_i^{anchor})$, we compute a pair of maps:

$$\begin{aligned} \Delta_x^i[x, y] &= x - x_i^{anchor}, \\ \Delta_y^i[x, y] &= y - y_i^{anchor}, \end{aligned} \quad \forall (x, y) \in \{0, \dots, W\} \times \{0, \dots, H\}, \quad (5)$$

where W, H are width and height of the input image [e.g., input eye region]

respectively.” EX-1006, p.6. Here, $\Delta_x^i[x, y]$ and $\Delta_y^i[x, y]$ represent “difference[s] between the pixels at two points.” EX-1001, 9:59-60. Therefore, Ganin’s feature maps that include $\Delta_x^i[x, y]$ and $\Delta_y^i[x, y]$ are “local descriptors.” EX-1003, ¶146.

Second, for each type of these local descriptors, i.e., pixel values of RGB channels, feature or anchor points, feature or anchor maps, Ganin discloses deriving a feature vector from them. As discussed *supra* Section VIII.A.2.b, during training, Ganin’s multi-scale NN takes three types of input data: (1) an input eye region, (2) feature points (anchors), and (3) a correction angle.

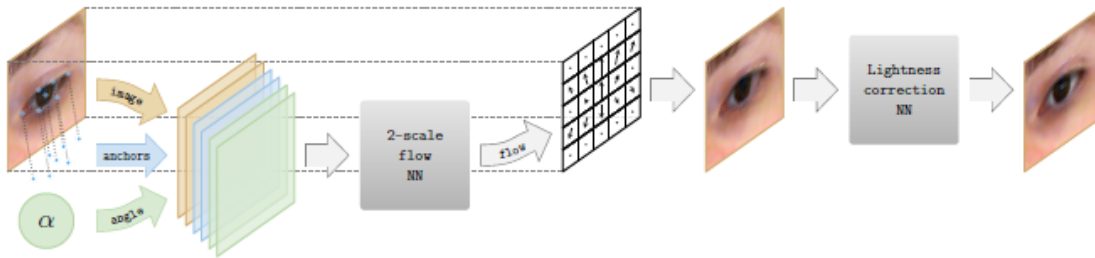


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2, Fig.10(a). According to Ganin, “[a]ll inputs are expressed as maps,” which are concatenated together into an “input map stack” fed to Ganin’s multi-scale NN. *Id.*, pp.5-6. The “input map stack” uses the RGB channels as the initial map stack. *Id.*, p.6. Ganin then discloses “[e]mbedding the [correction] angle” by deriving “a single position-independent 16-dimensional vector” from

the correction angle, which “is then expressed as 16 constant maps that are concatenated into the input map stack.” *Id.* The input map stack at that point consists of 3 RGB maps and 16 angle embedding maps. EX-1003, ¶147. Ganin further concatenates the feature maps mentioned above, forming “the input map stack consist[ing] of 33 maps ([3] RGB + 16 angle embedding maps + 14 feature point embedding maps).” EX-1006, p.6.

In the process of forming the “input map stack,” Ganin discloses deriving a feature vector in various stages. As a first example stage, when taking the pixel values of the RGB channels or the feature maps as local descriptors, the input map stack is a concatenation of them and is therefore a feature vector derived therefrom. EX-1003, ¶¶148-149.

As a second example stage, also when taking the pixel values of the RGB channels as local descriptors, the feature points are extracted from these pixel values using “an external face alignment library,” and therefore constitute a feature vector derived therefrom. EX-1006, p.5; EX-1003, ¶150.

As a third example stage, Ganin discloses “comput[ing] a pair of [distance] maps” using anchor points $(x_i^{anchor}, y_i^{anchor})$:

$$\begin{aligned}\Delta_x^i[x, y] &= x - x_i^{anchor}, \\ \Delta_y^i[x, y] &= y - y_i^{anchor},\end{aligned}\quad \forall (x, y) \in \{0, \dots, W\} \times \{0, \dots, H\}, \quad (5)$$

where W, H are width and height of the input image respectively.” EX-1006, p.6.

Thus, the “pair of [distance] maps” for an anchor point of an input image are local descriptors, and the combined pairs of distance maps for all the anchor points of the input image constitute a feature vector. EX-1003, ¶151. Alternatively, the anchor points themselves and corresponding pixel values may be considered local descriptors. Accordingly, either a pair of distance maps for an anchor point or the combined pairs of distance maps for all the anchor points constitute a feature vector. *Id.*

Aside from Ganin’s on-point teachings of local descriptors, the ’985 Patent itself admits that “local descriptors” were not new. The patent points to three prior-art publications teaching three types of local descriptors: “SIFT features,” “HOG features,” and “‘low level representations’ from pre-classification stages in deep learning neural networks.” EX-1001, 9:65-10:19. It also acknowledges a paper that uses “a distance of a pixel location with the position of an eye landmark.” EX-1001, 10:61-67. Thus, in addition to Ganin disclosing these features, the admitted prior art in the ’985 patent, additionally or alternatively, shows that the aspects recited in claim 2 were nothing new. EX-1003, ¶¶152-153.

Third, Ganin discloses that “the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch.” As explained in Section VIII.A.2, each of Ganin’s coarse and fine warping modules “takes as an input the image, the position of the feature points [embedded as

feature maps], and the redirection angle [embedded as constant maps],” i.e., the “input map stack.” This is true for both runtime and training. EX-1006, p.5; EX-1003, ¶154. The trained coarse warping model takes the “input image stack” and generates the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$. Similarly, the trained fine warping model also takes the “input image stack” and other parameters as appropriate, and generates “residual flow” $\mathbf{D}_{\text{res}}$ and then final warping field $\mathbf{D}(I, \alpha)$. EX-1003, ¶154.

Therefore, Ganin discloses that “the configuration of the target feature of each input patch is represented by a feature vector [e.g., “input map stack”] derived from plural local descriptors [e.g., $N=7$ feature points and pixel values within input patch] of the input patch, and the first and second selection instructions [e.g., trained coarse and fine warping modules] define how the feature vector is used to select editing instructions [e.g., coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ for coarse warping module, and “residual flow” $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ for fine warping module] for the input patch.” EX-1003, ¶¶141-155.

4. Claim 3

Claim 3 depends from claim 1 (which is disclosed by Ganin for reasons provided above) and adds that “the editing instructions comprise a displacement vector field defining how the input patch is to be transformed.”

Ganin discloses this added aspect by teaching using a “flow field,” or a “warping field” to transform an input eye region into a gaze-corrected eye region.

EX-1003, ¶157. Ganin transforms pixels of the input image I using coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$, resulting in the rough image estimate (i.e., coarse gaze correction):

$$O_{\text{coarse}}(x, y, c) = I\{x + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 1), y + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 2), c\}, \quad (2)$$

EX-1006, p.5, Eq.2. Similarly, Ganin also transforms pixels of the input image I using final warping field $\mathbf{D}(I, \alpha)$, resulting in the final estimate (i.e., fine gaze correction):

$$O = \mathbf{S}(I, \mathbf{D}(I, \alpha)). \quad (4)$$

EX-1006, p.6, Eq.4. The '985 Patent specifically acknowledges that its “displacement vector fields” may be “referred to as a ‘flow field’” (EX-1001, 9:11-12), which is another way Ganin refers to its “warping fields.” EX-1001, 9:6-12; EX-1006, pp.4, 18, Fig.2, Fig.10; EX-1003, ¶157.

Therefore, Ganin discloses that “the editing instructions comprise a displacement vector field [e.g., “flow field” or “warping field”] defining how the input patch is to be transformed [e.g., gaze correction by warping pixels].” EX-1003, ¶¶156-158.

In addition to Ganin disclosing the '985 Patent's “displacement vector field,” the '985 Patent admits that this aspect was nothing new. EX-1001, 9:6-12 (“the displacement vector field $\mathbf{F}$ may be derived as disclosed in Kononenko et

al.”); EX-1003, ¶159.

5. Claim 5

a. 5[P] and 5[A]

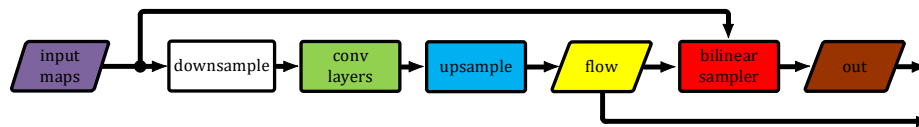
Ganin discloses “first editing instructions,” e.g., coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$. “ $\mathbf{D}_{\text{coarse}}(I, \alpha)$ [is] applied to **warp the input image** by means of a **bilinear sampler S** that finds the *coarse estimate*:

$$O_{\text{coarse}} = \mathbf{S}(I, \mathbf{D}_{\text{coarse}}(I, \alpha)) . \quad (1)$$

EX-1006, p.5. “Here, the sampling procedure S samples the pixels of O_{coarse} at pixels determined by the flow field:

$$O_{\text{coarse}}(x, y, c) = I\{x + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 1), y + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 2), c\} , \quad (2)$$

where c corresponds to a color channel (R,G, or B), and the curly brackets correspond to bilinear interpolation of $I(\cdot, \cdot, c)$ at a real-valued position. The sampling procedure (1) is piecewise differentiable.” *Id.* Ganin also illustrates this process in Fig.10(b) below, where the bilinear sampler (red) is used to apply the coarse warping fields (yellow) to input images (represented by input maps (purple)) to generate rough gaze-corrected image estimates.



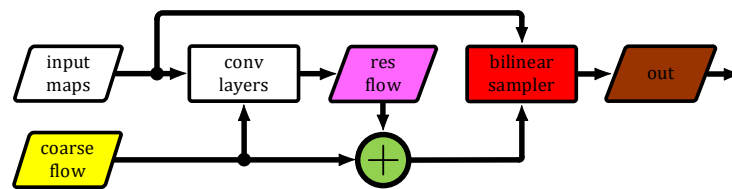
(b) 0.5×-scale processing module

EX-1006, Fig.10(b); EX-1003, ¶162.

Therefore, Ganin discloses “a first editing algorithm [e.g., bilinear sampler S] is used by the first machine learning algorithm [e.g., coarse warping module] to define how the first editing instructions [e.g., coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$] are to be applied to an input patch [e.g., “input image”] to derive an output patch [e.g., rough image estimates or “coarse estimate”].” EX-1003, ¶¶161-163.

b. 5[B]

Ganin’s fine warping module generates a residual flow $\mathbf{D}_{\text{res}}$, which is added to the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ to obtain the final warping field $\mathbf{D}(I, \alpha)$. EX-1006, pp.5-6. Like its coarse warping module, Ganin uses a bilinear sampler to apply the final warping field $\mathbf{D}(I, \alpha)$ to warp the input image I “to obtain the final output... $O = \mathbf{S}(I, \mathbf{D}(I, \alpha))$.” EX-1006, p.6. Ganin illustrates this process in Fig.10(c) below, where the bilinear sampler (red) is used to apply the combined residual flow (pink) and coarse warping field (yellow) to input images to generate final gaze-corrected images (brown).



(c) 1×-scale processing module

EX-1006, Fig.10(c); EX-1003, ¶164.

Therefore, Ganin discloses “a second editing algorithm [e.g., bilinear

sampler S] is used by the second machine learning algorithm [e.g., fine warping module] to define how the second editing instructions [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] are to be applied to an input patch [e.g., “input image”] to derive an output patch [e.g., final gaze corrected images].” EX-1003, ¶¶164-165.

6. Claim 10

Claim 10 depends from claim 1 (which is disclosed for reasons provided earlier) and adds that “**the first machine learning algorithm comprises one or more of the following: a neural network; a support vector machine; and a generative adversarial network (GAN).**” Ganin discloses that “the first machine learning algorithm comprises... a neural network [e.g., 0.5×-scale NN]”. EX-1003, ¶¶166-167.

7. Claim 12

Claim 12 depends from claim 1 (which is disclosed for reasons provided earlier) and adds aspects taught by Ganin because Ganin is directed to eye gaze correction and Ganin’s “target feature comprises... an eye region comprising at least part of an eye.” EX-1003, ¶¶168-169.

8. Claim 13

a. 13[P]

Ganin proposes an improved “method” for “gaze manipulation.” EX-1006,

p.1. Ganin’s method uses a “multi-scale neural network [to] predict[] a flow field,” which “is then applied to the input image to produce an image of a redirected eye.” *Id.*, p.4; VIII.A.2.

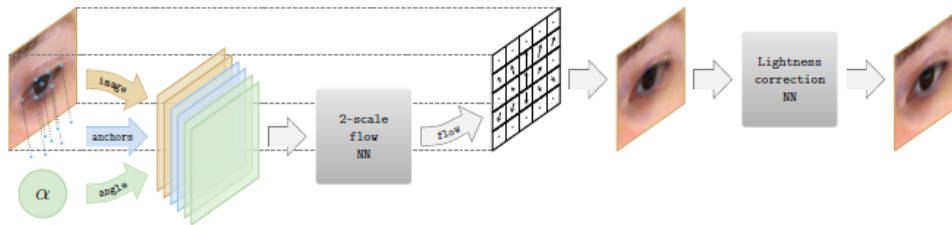


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2.

Therefore, to the extent 13[P] is limiting, Ganin discloses “a method of adjusting a digital representation of a head region [e.g., gaze redirection for an input eye region].” EX-1003, ¶¶171-172.

b. 13[A]

As shown in Fig. 2 below, the input to Ganin’s gaze redirection method includes “an **input eye region**, feature points (anchors) as well as a correction angle α .”

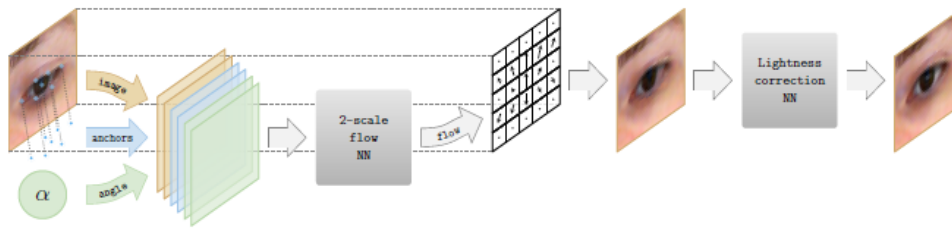


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

EX-1006, Fig.2; *supra* Section VIII.A.3 (the input eye region is combined with two other parameters to form an “input map stack” to be received by the multi-scale NN).

Therefore, Ganin discloses “receiving a digital representation of a head region [e.g., an input eye region].” EX-1003, ¶¶173-174.

c. 13[B]

Ganin discloses generating “second reference data” **during training**, which includes “second selection instructions” (e.g., trained 1×-scale NN) and “second editing instructions,” (e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$).

Supra Section VIII.A.2.h; EX-1003, ¶175.

Ganin also discloses using the “second reference data” generated by the method of claim 1, i.e., generated **during training**, for gaze redirection **during runtime**. EX-1003, ¶176.

First, Ganin uses the trained 1×-scale NN to generate residual flow $\mathbf{D}_{\text{res}}$ and

final warping field $\mathbf{D}(I, \alpha)$ during runtime. *Supra* Sections VIII.A.2.g-VIII.A.2.h. Thus, by using the “second selection instructions” (e.g., the trained $1\times$ -scale NN, which is part of the “second reference data” generated during training), Ganin discloses using the “second reference data” for runtime gaze redirection. EX-1003, ¶177.

Second, residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ are used in Ganin for gaze redirection and are generated **during both training and runtime**. *Supra* Section VIII.A.2.f. The residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training encompass all the possible values of residual flow $\mathbf{D}_{\text{res}}$ and/or final warping field $\mathbf{D}(I, \alpha)$. By picking the optimal residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ from the “second editing instructions” (e.g., the residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training or representation thereof, which are part of the “second reference data”), Ganin discloses using the “second reference data” for runtime gaze redirection. EX-1003, ¶¶178-179.

9. Claim 14

As explained above, Ganin discloses a method according to claim 1. *Supra* Section VIII.A.2. Ganin further discloses “computational performance of [its] method” “on a mid-range consumer GPU (NVIDIA GeForce-750M),” and also contemplates performance of its method “on CPU.” EX-1006, p.14. “GPU” and

“CPU” each are types of processors. EX-1003, ¶¶180-183.

10. Claim 18

a. 18[P] and 18[A-E]

18[P] and 18[A-E] are identical to 1[P] and 1[A-E], respectively, and are disclosed for the same reasons. *Supra* Sections VIII.A.2.a-VIII.A.2.f; EX-1003, ¶¶185-186.

b. 18[F]

18[F] is identical to an aspect in claim 12 that is taught by Ganin (*supra* Section VIII.A.7), so 18[F] is also taught by Ganin. EX-1003, ¶¶187-188.

B. GROUND #2: Claims 16 and 19 are rendered obvious by Ganin in view of Liu.

1. Claim 16

a. 16[P]

Most of 16[P] is identical to 1[P] and is thus disclosed in Ganin for the same reasons. *Infra* Section VIII.A.2.a. Ganin also discloses that its method may be performed on a processor. *Supra* Section VIII.A.9. The processor is a “data processing unit” and the device containing such processor (GPU or CPU) is the recited device in claim 16. EX-1003, ¶¶190-192.

b. 16[A-G]

16[A-G] collectively are substantively identical to 1[A-E] collectively and are therefore disclosed in Ganin for the same reasons. *Supra* Sections VIII.A.2.b-

VIII.A.2.f; EX-1003, ¶¶193-196.

c. 16[H]

16[H] recites that “the second editing instructions in the second reference data are provided in a compressed representation,” which is disclosed by Ganin in view of Liu. EX-1003, ¶197.

According to the '985 Patent, a compressed representation “may comprise a principal component analysis representation.” EX-1001, 15:59-61. Liu discloses applying “PCA... to the two-dimensional (2D) data of optical flows.” EX-1016, p.1. Ganin’s warping fields provide motions to pixels and therefore are a type of optical flows. As explained in Section VIII.B.2, a PHOSITA would have been motivated to use Liu’s PCA to compress Ganin’s warping fields, i.e., second editing instructions. EX-1003, ¶198.

Therefore, Ganin in view of Liu discloses that “the second editing instructions in the second reference data are provided in a compressed representation [e.g., PCA representation].” EX-1003, ¶¶197-199.

2. Motivation to Combine Ganin with Liu

A PHOSITA would have been motivated to apply Liu’s PCA to Ganin’s warping fields. EX-1003, ¶¶200-204.

Ganin and Liu qualify as analogous art to the '985 Patent because they are in the **same field of endeavor**. EX-1006, Fig.2; EX-1016, p.1 (Introduction); EX-

1001, 9:6-12. At a minimum, Liu is at least reasonably pertinent to a problem identified in the '985 Patent. EX-1001, 15:57-61; EX-1016, p.1; EX-1003, ¶201.

A PHOSITA would have been **motivated** to apply Liu's PCA to Ganin's warping fields to achieve the advantages disclosed in Liu, and Ganin itself encourages further improvements in its system. Specifically, Liu discloses that applying PCA to optical flows involves "project[ing] the original data onto an orthogonal subspace" and using "first several principal components" to represent "major data information." EX-1016, p.1. This can "reduce false alarms for videos with either static or dynamic background, and it is also useful to delineate the size of moving objects." *Id.* When applied to Ganin's warping fields, Liu's PCA compresses Ganin's warping fields such that Ganin's major warping information is represented by PCA components, thereby reducing unexpected pixel warping, possibly caused by noise, movement, or the person's facial appearance. Ganin also encourages further improvements to its system (EX-1006, p.14 ("Speed optimization")), so a PHOSITA would have naturally looked to an analogous reference like Liu to further improve Ganin's system. EX-1003, ¶202.

Finally, a PHOSITA would have had a **reasonable expectation of success** in applying Liu's PCA to Ganin's warping fields, at least because "[p]rinciple component analysis (PCA) is a typical approach in multivariate analysis." EX-1016, p.1. Not only does Liu provide a detailed description of its PCA approach,

PCA was also well known before the '985 Patent and widely used in image processing. Thus, applying PCA to Ganin's warping fields was a well-known process to implement and would have resulted in a working system. EX-1003, ¶203.

Therefore, Ganin-Liu renders obvious claim 16. EX-1003, ¶¶200-204.

3. Claim 19

a. 19[P] and 19[A-E]

19[P] and 19[A-E] are substantively identical to 1[P] and 1[A-E], respectively, and are disclosed by Ganin for the same reasons. *Supra* Sections VIII.A.2.a-VIII.A.2.f; EX-1003, ¶¶206-207.

b. 19[F-G]

Ganin-Liu includes having the second editing instructions provided in a PCA representation, and, because PCA is a form of compression, the additional aspects in 19[F-G] are rendered obvious. EX-1003, ¶208.

C. GROUND #3: Claims 1-3, 5-6, 8-14, and 16-19 are rendered obvious by the Chalom-Aslan Combination

1. Motivation to Combine Chalom and Aslan

As a threshold matter, Chalom and Aslan qualify as **analogous art** at least because they are in the **same field of endeavor** as the '985 Patent, *i.e.*, using machine learning for image processing. Both references (and the '985 Patent) disclose using training data, such as images, to train MLAs. EX-1001, Abstract;

EX-1005, 3:21-32; EX-1007, ¶24. Specifically, both references (and the '985 Patent) teach extracting features from input images (for training and for runtime) and providing these extracted features to the MLAs. EX-1001, Figs. 3, 8 and 11; EX-1005, Figs. 3, 8, 13 and 14; EX-1007, ¶24; EX-1003, ¶¶209-211.

At a minimum, Chalom and Aslan are at least reasonably pertinent to a problem identified in the '985 Patent of high accuracy MLAs often consuming more computational resources and being slower to run, as compared to efficient MLAs tending to provide less accurate predictions. EX-1001, 16:24-30; EX-1007, ¶45. Chalom and Aslan are at least reasonably pertinent to this efficiency-versus-accuracy problem. *See, e.g.*, EX-1005, 1:16-25, 9:48-55, 10:40-60, Fig.3; EX-1007, ¶45; EX-1001, 16:24-44, 17:14-17; EX-1003, ¶212.

Motivation also existed to incorporate Aslan's teacher-student architecture into Chalom's training of its RFC 306 for accuracy and/or efficiency gains. EX-1003, ¶213.

First, Chalom contemplates that its RFC 306 “may **apply different random forest models** responsive to or selectively based on camera 104 and display 102 having a particular relative position therebetween.” EX-1005, 10:36-39. In exploring ways to implement the use of different random forest models to construct the systems described by Chalom, a PHOSITA would have naturally consulted analogous references like Aslan and would have learned of Aslan's

teacher-student model architecture being applied to image processing. EX-1003, ¶214.

While Aslan describes the teacher-student architecture in the image-classification context, a PHOSITA would have been motivated to implement similar techniques for Chalom's image-editing context. This is at least because determining, using machine learning, an optimal motion vector field for a given eye region is still an image-classification task. Chalom's training process teaches all the possible motion vector fields to its MLAs, and these possible motion vector fields, although they will be used for image editing, are classes/labels themselves. During runtime, finding the optimal motion vector field corresponds to finding the best class/label and therefore involves a classification process. EX-1003, ¶215.

Alternatively, or in addition, Chalom expressly motivates looking for ways to further compress its random forest by indicating that it "may be compressed using any suitable technique or techniques." EX-1005, 15:4-7. This would have alternatively or additionally motivated a PHOSITA to look for such "suitable technique or techniques," and they would have been led to Aslan and its teachings related to "model compression" that allow for creating a second (student) model that "can comprise a much smaller machine learning model..." EX-1007, ¶44; EX-1003, ¶216.

Thus, upon learning of Aslan's teacher-student model architecture and its

application to image processing, a PHOSITA would have been motivated to apply that architecture in training the different random forest models described by Chalom, and would have done so to achieve Aslan's gains in efficiency and accuracy for the student machine learning model. EX-1007, ¶44; EX-1003, ¶217.

Second, a PHOSITA would have been motivated to incorporate Aslan's teacher-student architecture into Chalom's system in different ways. EX-1003, ¶218.

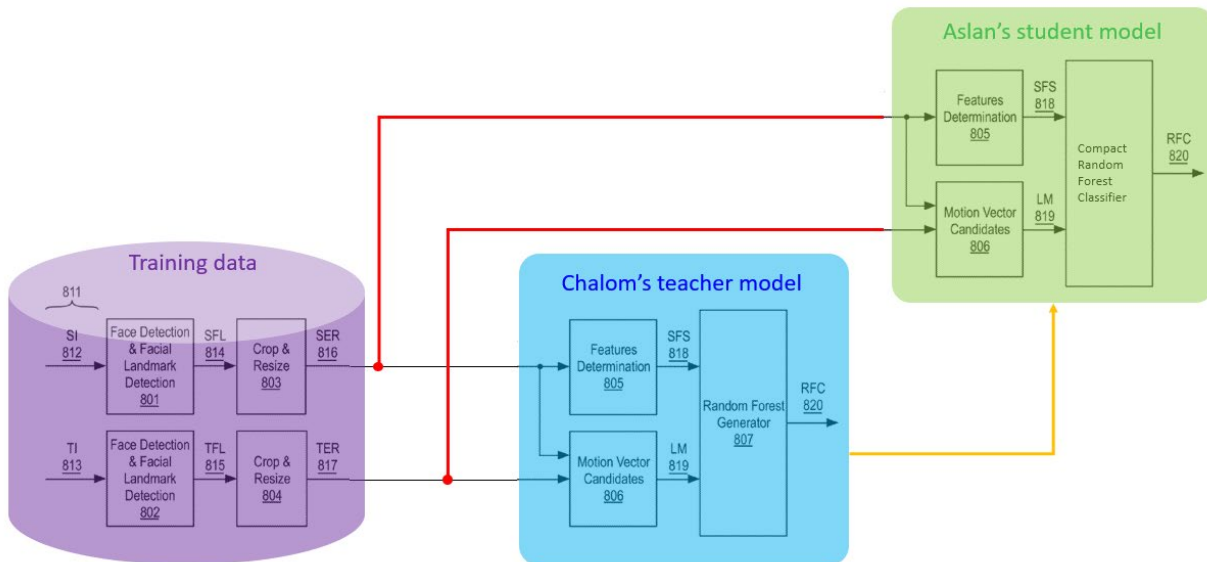
One way is using Chalom's RFC as a teacher model to train an even more compact MLA, e.g., a smaller random forest. A PHOSITA would have understood that Chalom's RFC is a machine-learning ensemble with a collection of decision trees. Aslan explains that ensembles "typically require a relatively **large storage footprint and powerful processing resources to execute at runtime**, they are not well suited for implementations where storage space and/or computational power is at a premium (such as with smart phones, wearables, hearing aids, etc.)." EX-1007, ¶3. Aslan therefore adopts an improved teacher-student architecture to solve this problem. Thus, a PHOSITA would have been motivated to incorporate Aslan's teacher-student training architecture by **using Chalom's already trained RFC as a teacher** to train a more compact RFC with fewer and/or less complex decision trees, thereby maintaining accuracy of the original RFC while achieving efficiency gains. EX-1003, ¶¶219-222.

Another way a PHOSITA would have been motivated to incorporate Aslan's teacher-student architecture is by using Chalom's RFC as a student model, which learns from a different teacher model. EX-1003, ¶223. It was well known at the time of the '985 Patent that using a random forest resulted in less optimal prediction accuracy as compared to deep convolutional neural networks for image-processing/classification tasks. A PHOSITA looking to improve the accuracy for Chalom's RFC would have been motivated to incorporate Aslan's teacher-student training architecture by **training a new accurate teacher model first, which is then used to train Chalom's RFC as a student model**. As taught by Aslan, such new teacher model could be, e.g., a deep neural network or another type of machine learning model. EX-1007, ¶45; EX-1003, ¶224.

Finally, a PHOSITA would have had a **reasonable expectation of success** in incorporating Aslan's teacher-student training architecture into Chalom's system. EX-1003, ¶225.

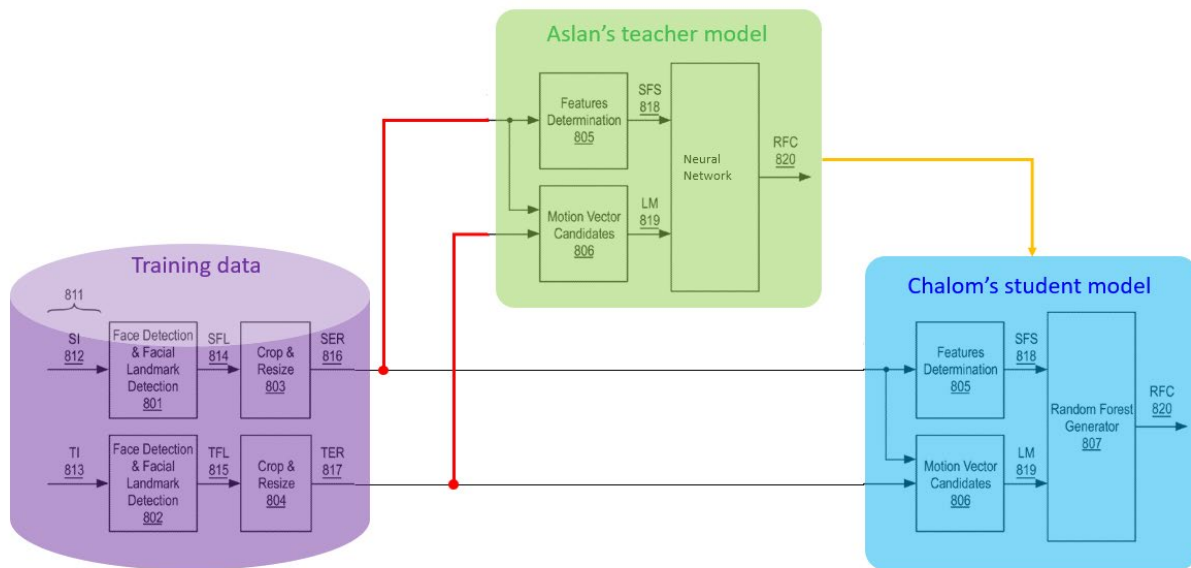
When using Chalom's RFC as a teacher to train a more compact student, a PHOSITA would have needed only to follow the instructions provided in Aslan, such that after training the RFC (blue, below) the exact way Chalom does, feed the same training data (purple) and the output of Chalom's random forest (orange) to train a student model, e.g., a smaller or more compact random forest (green). Such training of a random forest was a well-known process (as acknowledged by the

'985 Patent, EX-1001, 11:1-6), and would have been an easy-to-achieve repetition of the teacher's training process. An example illustration of the Chalom-Aslan combination is shown below.



EX-1005, Fig.8, modified based on Aslan to incorporate a student model; *c.f.* EX-1007, Fig.1. As shown, the same training data, e.g., source eye regions 816 and target eye regions 817 are provided to both the teacher model and the student model. Further optimizations and predictable variations also would have been apparent to a PHOSITA, including, e.g., (i) having one set of features determination module 805 and motion vector candidates module 806 to provide source feature sets 818 and likelihood maps 819 to both the teacher and student models; and (ii) using another type of MLA (other than a RFC) as the student model. EX-1003, ¶¶226-230.

Similarly, when using Chalom's RFC as a student, a PHOSITA would still only have needed to follow the instructions provided in Aslan, such that before training Chalom's RFC (blue, below), use the same training data (purple) to first train a more accurate teacher model (green).



EX-1005, Fig.8, modified based on Aslan to add a teacher model; *c.f.* EX-1007, Fig.1; EX-1003, ¶231.

Accordingly, a PHOSITA would have been motivated to incorporate Aslan's well-known teacher-student training architecture for training Chalom's RFC (using it as either the teacher or student model, as was explained above) to yield a working system. EX-1003, ¶¶209-232.

2. Claim 1

a. 1[P]

If the preamble is limiting, Chalom discloses it. EX-1003, ¶¶230-242.

Chalom discloses “a **training component** (e.g., performed offline) and a **real-time component** (e.g., performed during [] runtime) that provides **eye contact correction.**” EX-1005, 3:21-24; EX-1003, ¶235-238.

During runtime, as shown in Fig. 3 of Chalom (reproduced below), RFC 306 (yellow) “receive[s] feature sets 312 [red]” and “each feature set of feature sets 312... traverses branches of the forest based on decisions at each branch until a leaf is reached such that the leaf corresponds to or provides a motion vector field [e.g., motion vector field 313 (green)] for the feature set.” EX-1005, 9:17-22.

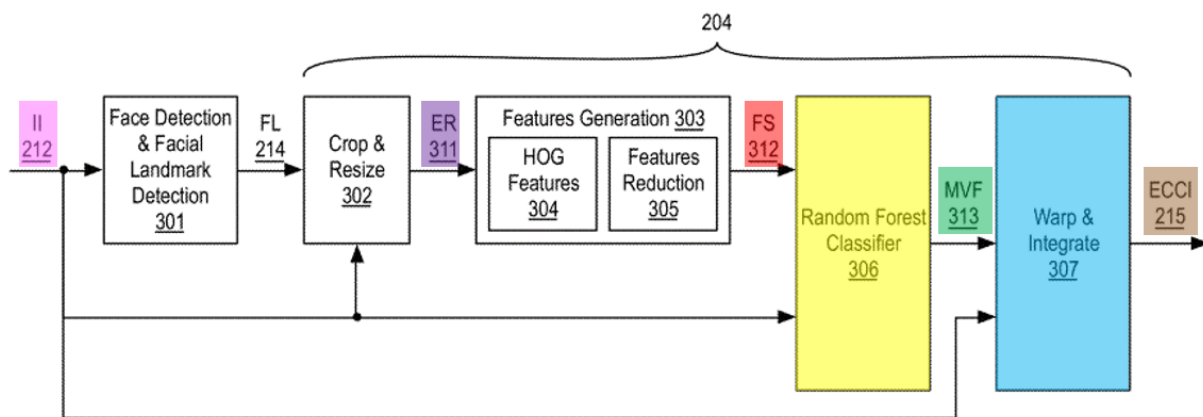


FIG. 3

EX-1005, Fig.3. The input image 212 (pink) may then be warped by warp and integrate module 307 (blue) “based on motion vector fields 313 [green].” EX-1005, 9:26-30; EX-1003, ¶239-240.

Chalom further describes the **training process** for its RFC 306 in Figure 8, reproduced below.

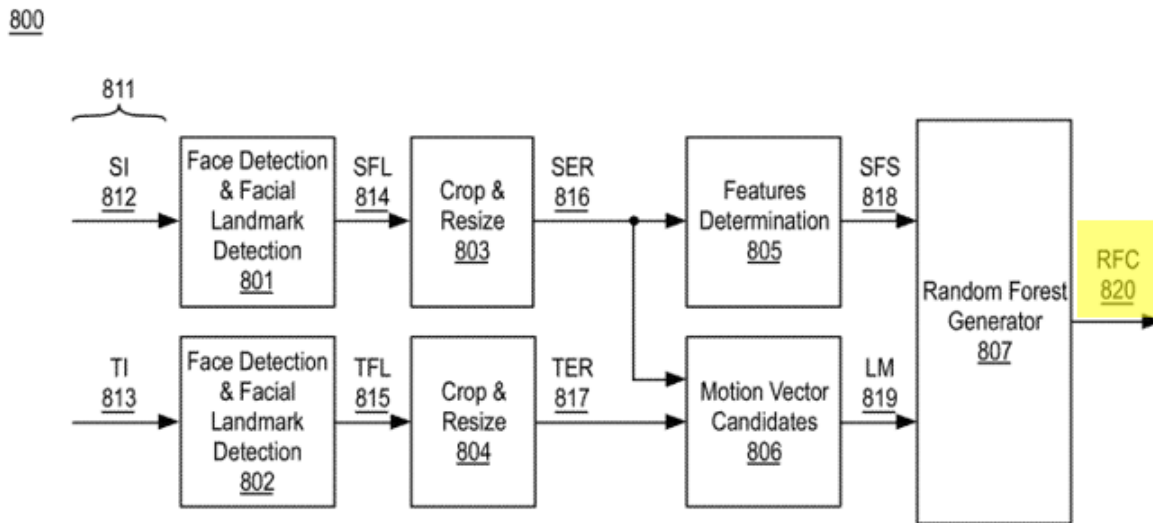


FIG. 8

EX-1005, Fig.8. Chalom discloses that “[t]raining may be performed such that **each leaf of a tree in random forest classifier 820 [yellow] represents a [motion vector] likelihood map (e.g., a SAD map) for every pixel in a source eye region**” and that “source features sets 818 and likelihood maps 819 [both of which are determined from the training data SER 816, extracted from SI 812 and/or TER 817, extracted from TI 813, in Fig. 8] may be provided to random forest generator module 807, which may train and determine random forests classifier 820 based on source feature sets 818 [extracted from the training data SER 816] and likelihood maps 819 [derived using the training data SER 816 and TER 817].” EX-1005, 13:41-55, 9:21-22. Furthermore, Chalom discloses that, during the training process, the “**random forest classifier 820 may be trained to determine optimal**

motion vector fields [for the input training data] from [the training data's] source feature sets 818.” EX-1005, 13:49-52, Fig.8. Chalom also discloses that such **training process** results in a trained RFC that generalizes to unseen data samples and “determines an **optimal motion vector field** based on the feature set [of input data] during the **implementation phase.**” EX-1005, 4:58-60.

Additionally, a PHOSITA would have understood that in order for the RFC to be able to generate an optimal motion vector field during the implementation phase, optimal motion vector fields are also generated for the training data during the training process. EX-1003, ¶241; *infra* Section VIII.C.2.e.

Therefore, Chalom discloses “[a] method of generating reference data [e.g., motion vector fields generated during training, including the recited “first reference data” in 1[D] and the recited “second reference data” of 1[E]] for adjusting a digital representation of a head region [e.g., gaze correction].” EX-1003, ¶¶234-242.

b. 1[A]

Chalom discloses that “during a training phase of the principal component analysis and/or the pretrained classifier, training may be performed based on a **training set of images** having known gaze angle difference therebetween.” EX-1005, 4:19-22. “During training, the training set images may be cropped to eye regions and the eye regions may be aligned to provide **pairs of training eye**

region images.” EX-1005, 4:25-28; Fig.8 (receiving source images 812 and target images 813 (pink), cropped and resized to generate source eye regions 816 and target eye regions 817 (purple)).

800

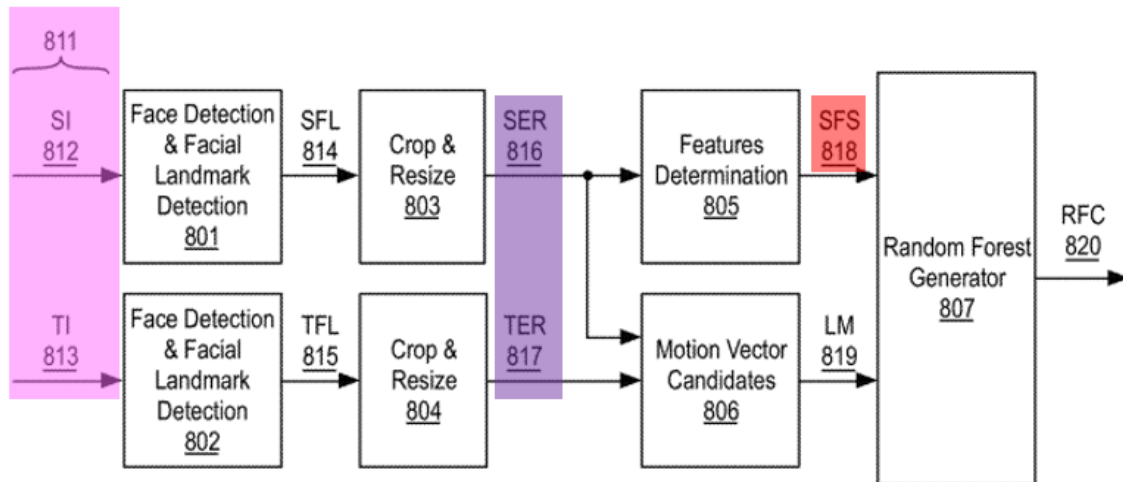


FIG. 8

EX-1005, Fig.8.

Therefore, Chalom discloses “receiving training data [e.g., pairs of training eye region images, including source eye regions 816 and target eye regions 817].”

EX-1003, ¶¶243-245.

c. 1[B]

Chalom’s training data includes two sets of image patches. As shown below, Chalom’s training data “include pairs of **source images 812** and target images 813 [pink].” EX-1005, 11:24-26.

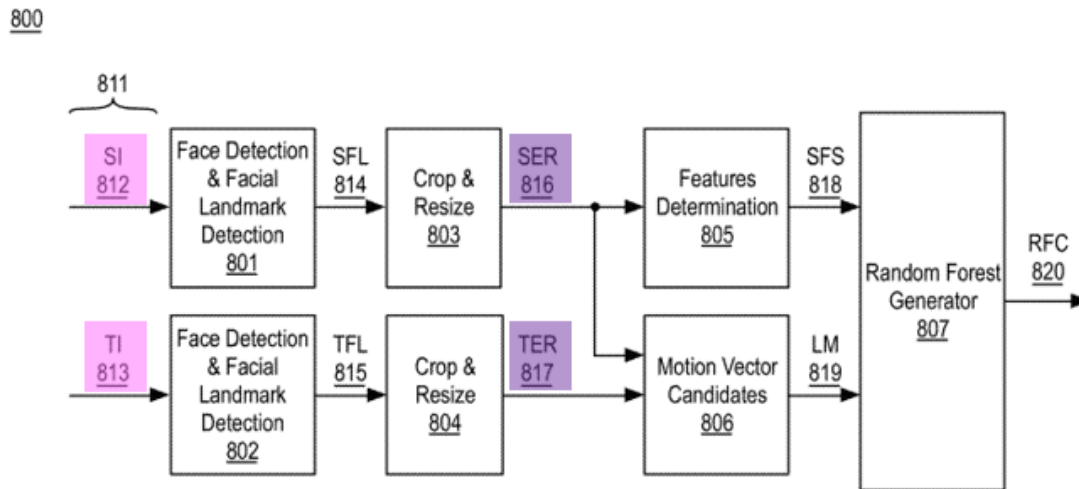


FIG. 8

EX-1005, Fig.8. Chalom’s source images 812 and target images 813 (pink) are analyzed by “face detection and facial landmark detection modules 801, 802” and then cropped and resized (803 and 804) to form source eye regions 816 and target eye regions 817 (purple), both of which are then used for training. EX-1005, 11:13-14. Since face and landmark detection and cropping and resizing are performed on each of the source images 912 and target images 913, they each have the same target feature, e.g., an eye region, of a digital representation of a head region. EX-1005, 12:1-22; EX-1003, ¶¶246-248.

Therefore, Chalom discloses “a set of input patches [e.g., source eye regions 816], each input patch comprising a target feature [e.g., eye region] of a digital representation [e.g., image] of a head region prior to adjustment [e.g., before gaze correction] of the digital representation of the head region, wherein the target

feature is the same for each input patch.” EX-1003, ¶¶246-249.

d. 1[C]

As explained in Section VIII.C.2.c, Chalom’s training data includes a set of output patches, e.g., target eye regions 817 (purple) obtained by cropping and resizing the target images 813 (pink) after face & landmark detection 802.

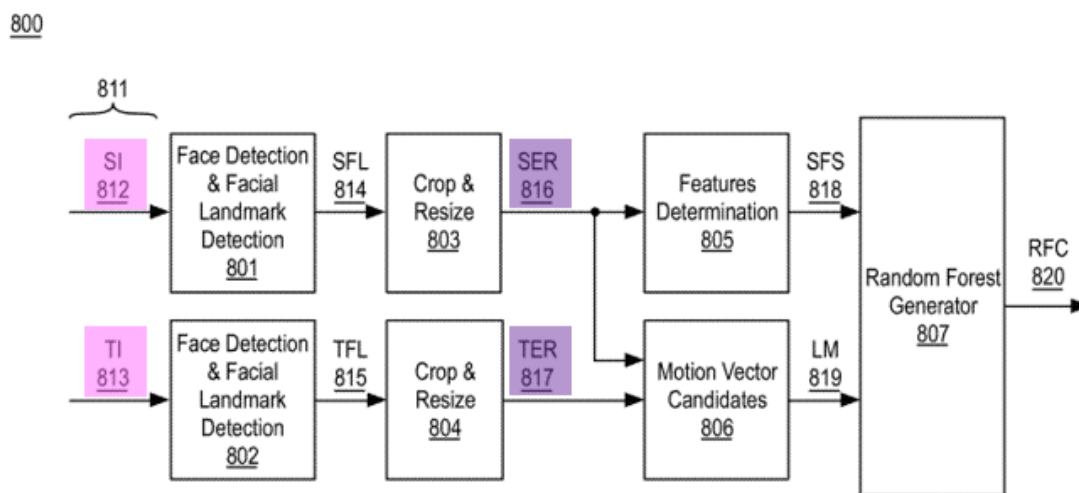


FIG. 8

EX-1005, Fig.8; EX-1003, ¶251.

As shown in Fig. 8 above, source image 812 and target image 813 (pink) correspond to each other such that source image 812 includes “a user looking in a downward direction” (i.e., gaze to be adjusted) and the corresponding target image 813 includes the same user “looking at the camera” (i.e., gaze has been adjusted in the desired direction). *Id.*, 11:36-39, 12:10-22, Fig.11.

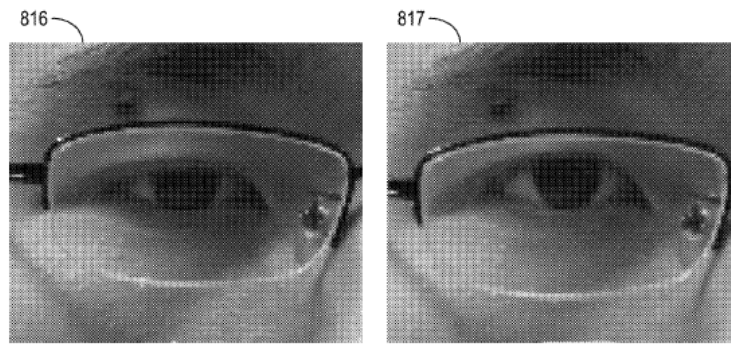


FIG. 11

Id., Fig.11; EX-1003, ¶¶252-254.

Therefore, Chalom discloses “a set of output patches [e.g., target eye regions 817] in one-to-one correspondence with the input patches, each output patch comprising the target feature [e.g., eye region] of the digital representation of the head region after adjustment [e.g., after gaze correction, e.g., “with correct gaze”] of the digital representation of the head region.” EX-1003, ¶¶250-255.

e. 1[D-E]

Chalom-Aslan teaches 1[D] and 1[E]. EX-1003, ¶256.

First, Chalom discloses “using a [] **machine learning algorithm** to generate [] reference data using the training data, the [] reference data comprising **editing instructions for adjusting** the digital representation of the head region.” EX-1003, ¶257. As shown in Fig. 8 of Chalom below, training data such as source eye regions 816 and target eye regions 817 are used by system 800 “to generate a random forest classifier 820.” EX-1005, 11:29-30.

800

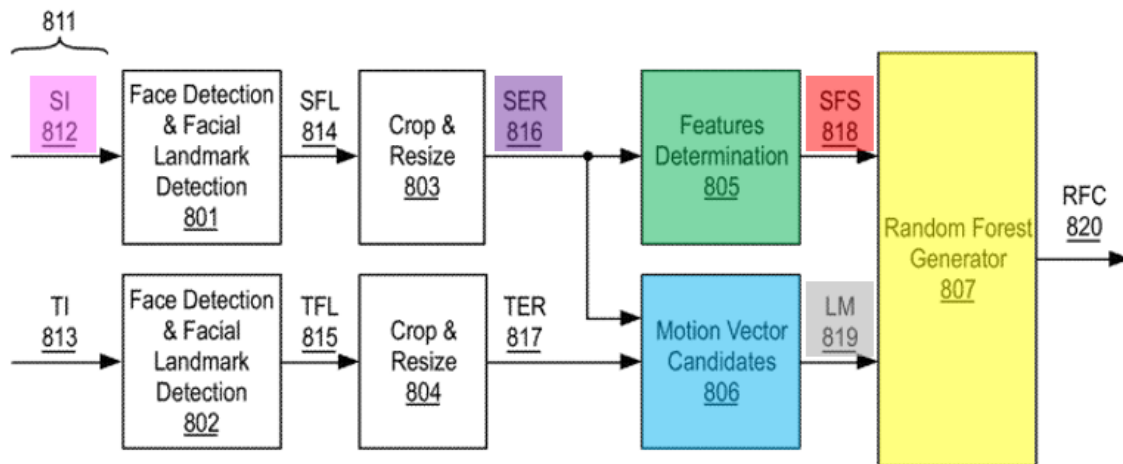


FIG. 8

Id., Fig.8. Specifically, “features determination module 805 [green] may receive source eye regions 816 [purple]... [and] determine source feature sets (SFS) 818 [red].” *Id.*, 13:8-11. Motion vector candidates module 806 (blue) receives both source eye regions 816 and target eye regions 817 to generate candidate motion vectors and likelihood maps 819 (gray). *Id.*, 12:23-27. The source feature sets 818 and likelihood maps 819 are then provided to random forest generator module 807 (yellow) to “train and determine random forest classifier 820.” *Id.*, 13:45-49; EX-1003, ¶258.

Chalom’s trained RFC 820 comprises “tree[s]” each having multiple “branches” and “lea[ves].” *Id.*, 9:9-22, 13:52-55. “[E]ach leaf of the tree of the random forest classifier may **represent a likelihood map** (e.g., a SAD map) for every pixel in the eye region” and “corresponds to or provides a **motion vector**

field for the feature set.” *Id.*, 4:62-64, 9:21-22, 13:49-55. The “motion vector field may be used to **warp the eye region** to generate the desired output image having corrected eye contact.” *Id.*, 3:36-38. Thus, Chalom’s training results in each leaf of the trees in Chalom’s RFC 820 providing a motion likelihood map for every pixel in source eye region and a corresponding motion vector field. EX-1003, ¶¶259-260.

Second, Chalom discloses 1[D]’s further recitation of “adjusting the digital representation of the head region for **a range of possible digital representations of the head region.**” Chalom discloses that its “[eye] contact correction module [] may be capable of implementing eye contact correction for **multiple orientations or relative positions between camera 104 and display 102.**” EX-1005, 10:12-14. To allow for such various “relative position[s] between camera 104 and display 102,” “camera 104 may be moved from a central position of display 102 to a position off center but still within an edge of display 102,” for example. *Id.*, 5:40-50. Thus, when moving camera 104 relative to display 102, with the person’s head region being captured by camera 104, at least the “orientations” of the person’s head would change. *Id.*, 10:10-14; EX-1003, ¶¶261-263.

Third, as to the use of a second MLA, Chalom-Aslan discloses this as well. As explained earlier, there was motivation to modify Chalom’s system to use Aslan’s teacher-student architecture, which includes a teacher MLA and a student

MLA, such that Chalom's training processes then involve use of two MLAs. EX-1003, ¶264.

As shown in Aslan's Fig. 1 below, during training, both the teacher model 100 (green) and the student model 102 (blue) "receiv[e], or hav[e] access to, the [same] training data 104." EX-1007, ¶23.

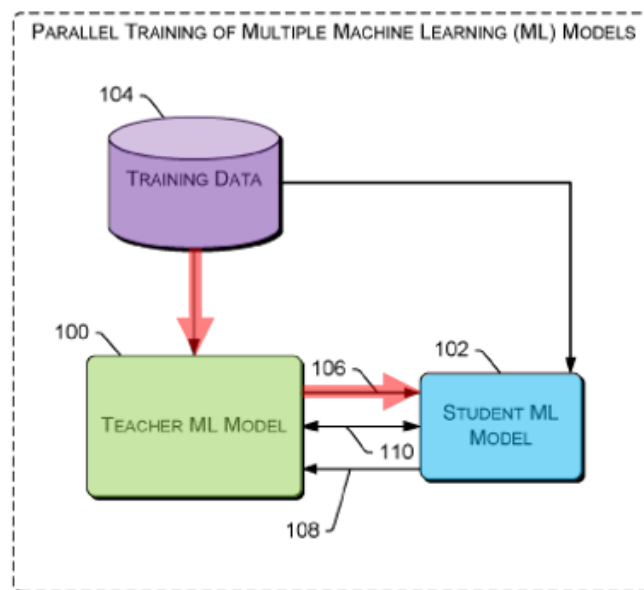


Fig. 1

Id., Fig.1. In Chalom-Aslan, the training data would have been the source eye regions and target eye regions from Chalom, and this training data would have covered a range of possible digital representations of the head region based on the different camera angles utilized (as explained above). EX-1003, ¶¶365-368.

According to Aslan, both the teacher and student models may be trained to

perform the same task based on the same training data (the source and target eye regions from Chalom). EX-1007, ¶25. In the Chalom-Aslan system, the “task” is generating motion vector fields for gaze correction, and the teacher’s output, e.g., first editing instructions including motion vector fields, is used to train the student that also generates motion vector fields, i.e., second editing instructions. EX-1003, ¶¶269-272.⁷

Therefore, the Chalom-Aslan combination discloses “using a first machine learning algorithm [e.g., Chalom-Aslan’s teacher model] to generate first reference data using the training data, the first reference data comprising editing instructions [e.g., motion vector fields generated by the teacher model] for adjusting the digital representation of the head region [e.g., gaze correction] for a range of possible digital representations of the head region [e.g., different head orientations],” as well as “using a second machine learning algorithm [e.g., Chalom-Aslan’s student model] to generate second reference data using the same training data as the first machine learning algorithm [e.g., training data for the teacher model] and the first

⁷ Aslan also discusses sequential training of teacher and student models, such that it was known to use sequential or parallel training. EX-1007, ¶5. Additionally, the claims of the ’985 Patent do not require sequential learning and thus they are taught by parallel or sequential learning. EX-1003, ¶273.

reference data generated by the first machine learning algorithm [e.g., motion vector fields generated by the teacher model], the second reference data comprising editing instructions [e.g., motion vector fields generated by the student model] for adjusting the digital representation of the head region for a range of possible digital representations of the head region.” EX-1003, ¶¶256-274.

f. 1[F-G]

1[F] and 1[G] are also taught by Chalom-Aslan. First, as explained in Section VIII.C.2.e *supra*, Chalom discloses generating reference data that comprises editing instructions, e.g., motion vector fields, for gaze correction, during training. Chalom’s motion vector fields generated during training are “for a range of possible configurations of the target feature,” because “random forest classifier 306 may apply **different random forest models** responsive to or selectively based on camera 104 and display 102 **having a particular relative position therebetween**,” EX-1005 10:35-39, which thereby results in having training data that expressly, or at least inherently includes different gaze directions. Also, in Chalom-Aslan, since the training data for both the teacher and student models are the same, the first editing instructions and second editing instructions generated from them respectively are both “for a range of possible configurations of the target feature,” e.g., different gaze directions. EX-1003, ¶¶275-278.

Second, Chalom in view of Aslan also discloses aspects related to first and

second selection instructions and their intended uses for selecting respective editing instructions, as recited in 1[F-G]. EX-1003, ¶280.

While claim 1 is directed to a machine-learning training process, it nonetheless recites intended uses of the outputs of the training process for runtime purposes. For example, 1[F] recites the “first reference data” generated by the “second machine learning algorithm” **during training**, but also recites that “the first reference data comprise... first selection instructions **for selecting editing instructions** for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch,” which reflects how generated “first reference data” comprising “first selection instructions” is intended to be used **during runtime**. *See also* 1[G] (“second selection instructions **for selecting editing instructions...**”). These intended-use recitations are non-limiting, but, even if limiting, Chalom-Aslan discloses them. EX-1003, ¶281.

In Chalom, “each leaf of the tree of the random forest classifier may represent a likelihood map (e.g., a SAD map) for every pixel in the eye region” and “corresponds to or provides a motion vector field.” EX-1005, 4:60-64, 9:21-22. Therefore, the training of Chalom’s RFC generates not only motion vector fields, but also the trained RFC 820 (yellow) as shown in Fig. 8 below (and RFC 306 in Fig. 3), represented by a group of trees having branches and leaves with each leaf corresponding to a motion vector field. *Id.*, 9:9-22, 13:52-55.

800

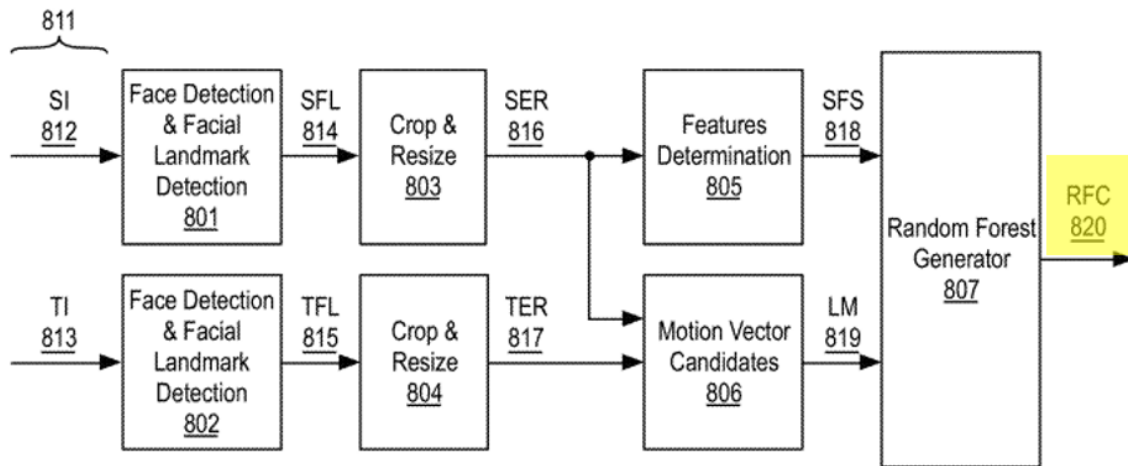


FIG. 8

Id., Fig.8; EX-1003, ¶282.

For a particular input source eye region, the corresponding motion vector field output by the trained RFC **during runtime** is selected from motion vector fields generated **during training**. This is because each leaf of Chalom’s RFC (generated during training) corresponds to a possible motion vector field, and the runtime implementation is the process of finding the optimal motion vector field among these leaves, i.e., possible motion vector fields. Thus, Chalom’s trained RFC provides “selection instructions for selecting editing instructions [e.g., optimal motion vector field] for a particular input patch from editing instructions [e.g., motion vector fields generated during training].” EX-1003, ¶283.

Moreover, Chalom’s selection of an optimal motion vector field is based on

“the configuration of the target feature of the input patch” required by 1[F-G].

Specifically, Chalom discloses extracting histogram of oriented gradient (HOG) features from input images, which are then “reduced using principal component analysis (PCA) techniques” to “convert[] the histogram of oriented gradient

features to feature sets 312.” EX-1005, 8:59-61. As shown in Figure 3 below,

Chalom’s selection of optimal motion vector fields (MVF 313, green) is based on

the “feature set 312 [red],” i.e., configuration of the target feature of the input

patch.

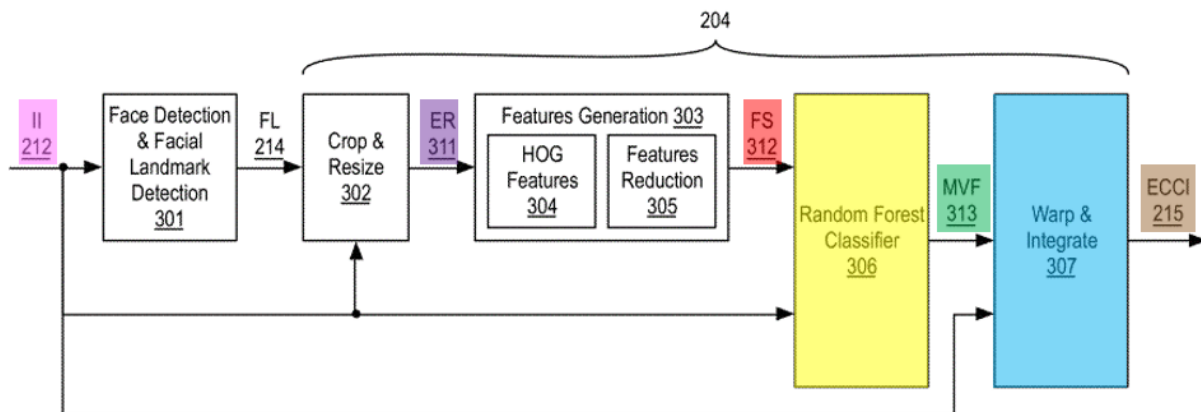


FIG. 3

Id., Fig.3; claim 2; EX-1003, ¶284.

Therefore, Chalom-Aslan discloses that “the first reference data comprise first editing instructions [e.g., teacher’s motion vector fields generated during training] for a range of possible configurations of the target feature [e.g., different gaze directions] and first selection instructions [e.g., trained teacher model] for

selecting editing instructions [e.g., optimal motion vector field] for a particular input patch from the first editing instructions [e.g., teacher's motion vector fields generated during training] based on the configuration of the target feature of the input patch [e.g., Chalom's feature set 312],” and that “the second reference data comprise second editing instructions [e.g., student's motion vector fields generated during training] for a range of possible configurations of the target feature [e.g., different gaze directions] and second selection instructions [e.g., trained student model] for selecting editing instructions [e.g., optimal motion vector field] for a particular input patch from the second image editing instructions [e.g., student's motion vector fields generated during training] based on the configuration of the target feature of the input patch [e.g., Chalom's feature set 312].” EX-1003, ¶¶275-286.

3. Claim 2

Claim 2 depends from claim 1 (disclosed by Chalom-Aslan as explained above) and adds aspects (each discussed below) taught by Chalom-Aslan. EX-1003, ¶287.

First, like the '985 Patent, Chalom discloses extracting HOG features from

input images.⁸ Chalom's Fig. 3 discloses "a features generation module 303 that may include a histogram of oriented gradients (HOG) features module 304 [green]," "which may generate or determine histogram of oriented gradient features for each eye region of eye regions 311." EX-1005, 7:27-29, 8:30-34.

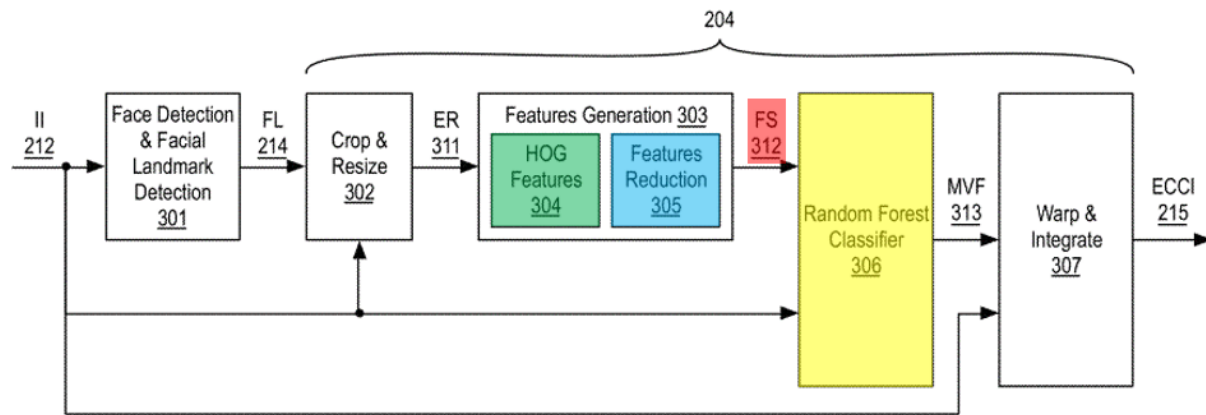


FIG. 3

EX-1005, Fig.3; EX-1003, ¶¶288-290.

Chalom further discloses a feature vector generated from these HOG features. As shown in Fig. 3 above, "[t]he histogram of oriented gradient features generated by features generation module 303 may be reduced by features reduction module 305 [blue]." EX-1005, 8:49-51. The HOG features "may be **reduced using principal component analysis (PCA)** techniques... [which] may include

⁸ The '985 Patent admits that "local descriptors" were not new. *Supra* Section VIII.A.3.

converting the histogram of oriented gradient features to **feature sets 312 [red]** such that feature sets 312 are linearly uncorrelated variables derived from the histogram of oriented gradient features.” EX-1005, 8:57-63. According to Chalom, each “feature set correspond[s] to each eye region” (EX-1005, 9:4-6) and therefore the recited “configuration of the target feature of each input patch” is represented by such feature set. EX-1003, ¶291.

Second, as to selecting editing instructions, Chalom-Aslan includes a trained teacher model and a trained student model, each of which receives input images and generates optimal motion vector fields based on the feature sets extracted from those images. EX-1003, ¶292.

Therefore, Chalom discloses “the configuration of the target feature of each input patch is represented by a feature vector [e.g., Chalom’s feature set 312] derived from plural local descriptors [e.g., Chalom’s HOG features] of the input patch, and the first and second selection instructions [e.g., the trained teacher model and the trained student model in Chalom-Aslan system] define how the feature vector is used to select editing instructions [e.g., motion vector fields] for the input patch.” EX-1003, ¶¶287-293.

4. Claim 3

Claim 3 depends from claim 1 (disclosed by Chalom-Aslan for reasons

provided above) and adds aspects taught by Chalom-Aslan.⁹ EX-1003, ¶294.

According to Chalom, motion vector fields “may adjust the input (e.g., having an uncorrected eye region) image to a desired output (e.g., having corrected eye contact)” by “warp[ing] the eye region.” EX-1005, 3:28-31, 3:36-38; 9:30-38 (“motion vector fields 313 may be applied to eye regions 311 by **determining, for each pixel location of eye regions 311 a warped pixel value** as a pixel value corresponding to pixel value as indicated by the motion vector for that pixel.”). EX-1003, ¶¶295-298.

Therefore, Chalom-Aslan discloses “the editing instructions comprise a displacement vector field [e.g., Chalom’s motion vector field] defining how the input patch is to be transformed [e.g., how an eye region is warped].” EX-1003, ¶¶294-299.

5. Claim 5

Claim 5 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above) and adds aspects about first and second editing algorithms used by the first and second MLAs, respectively, which are taught by Chalom-Aslan. EX-1003, ¶300.

⁹ The ’985 Patent admits that the added feature of Claim 3 is not new. *Supra* Section VIII.A.4.

As explained in Section VIII.C.2.e, both the teacher and student models in Chalom-Aslan generate “editing instructions,” e.g., motion vector fields. According to Chalom, “the eye region(s) of the source image may be warped based on the motion vector field(s) and integrated into the source image (e.g., into the remaining, non-eye region portion of the source image) to generate the eye contact corrected image.” EX-1005, 4:12-17; EX-1003, ¶301.

Therefore, Chalom in view of Aslan discloses “a first editing algorithm [e.g., Chalom’s algorithm to warp the eye regions] is used by the first machine learning algorithm [e.g., Chalom-Aslan’s teacher model] to define how the first editing instructions [e.g., motion vector fields generated by the teacher model] are to be applied to an input patch [e.g., eye regions to be warped] to derive an output patch [e.g., warped eye regions]; and a second editing algorithm [e.g., Chalom’s algorithm to warp the eye regions] is used by the second machine learning algorithm [e.g., Chalom-Aslan’s student model] to define how the second editing instructions [e.g., motion vector fields generated by the student model] are to be applied to an input patch [e.g., eye regions to be warped] to derive an output patch [e.g., warped eye regions].” EX-1003, ¶¶300-302.

6. Claim 6

Claim 6 depends from claim 5 (disclosed by Chalom-Aslan for reasons provided above) and adds aspects concerning “principle component analysis” that

are taught by the combination.¹⁰ EX-1003, ¶303.

Chalom discloses using principal component analysis to compress the input to its RFC. As shown in Fig. 3 below, Chalom discloses a “histogram of oriented gradients features module 304 [green], which may generate or determine histogram of oriented gradient [HOG] features for each eye region of eye regions 311.” EX-1005, 8:31-34.

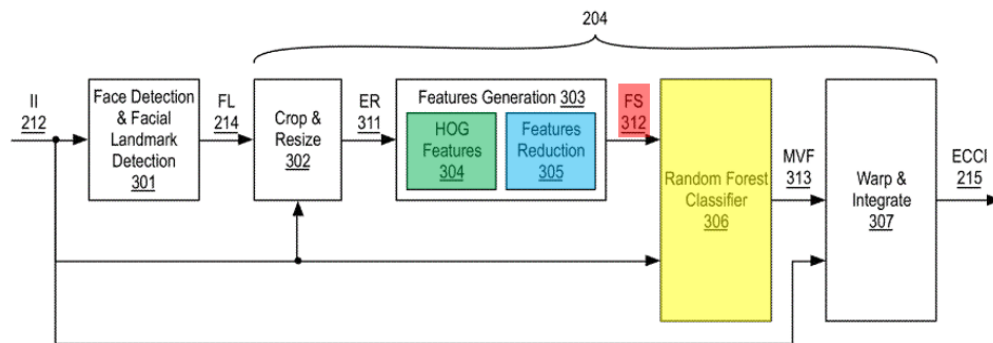


FIG. 3

EX-1005, Fig.3. These HOG features “may be reduced [by features reduction model 305 (blue)] using **principal component analysis** (PCA) techniques.” *Id.*, 8:57-59; 13:28-30. The reduced features are “feature sets 312 [red],” which “may be received by random forest classifier 306 [yellow].” *Id.*, 9:8-9. Such dimensionality reduction using PCA is also applied to HOG features generated

¹⁰ To the extent that “**principle** component analysis” in the ’985 Patent claims is not indefinite, it refers to “**principal** component analysis.” EX-1003, ¶304.

from input patches during training. As shown in Fig. 8, Chalom discloses that “features determination module 805 [green] may generate histogram of oriented gradient (HOG) features and apply principal component analysis to reduce the number of features.” *Id.*, 13:11-14.

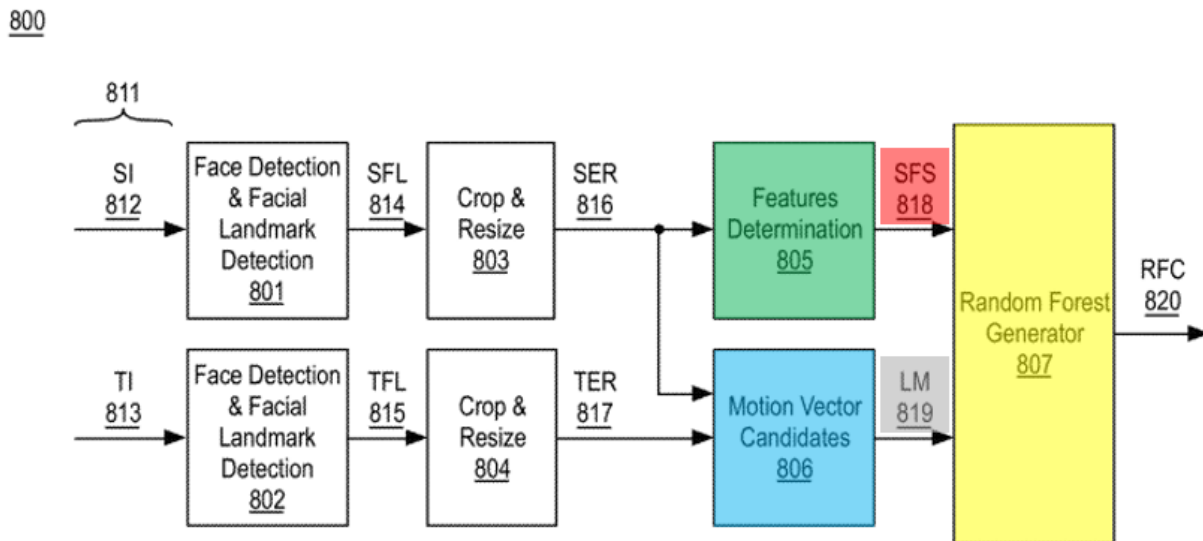


FIG. 8

Id., Fig.8; EX-1003, ¶306.

While Chalom discloses compressing one of the two inputs to its RFC, a PHOSITA would have found it obvious to apply Chalom’s PCA to compress the other input to the teacher model, the input to the student model, or both. Since the student takes both training data and “first editing instructions” as input, a PHOSITA would have been motivated to apply PCA to the training data, the “first editing instructions,” or both. EX-1003, ¶307.

When applying PCA to compress the “first editing instructions,” which are

then used to train the student model, the trained student model would generate “second editing instructions” as the PCA components, and such “second editing instructions” would have been converted back to approximate the “first editing instructions” by inverse principal component analysis transform. EX-1003, ¶¶308-309.

7. Claim 8

Claim 8 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above) and adds aspects (discussed below) taught by the combination. EX-1003, ¶¶310-311.

As explained in Section VIII.C.1, Chalom-Aslan can include a teacher model being a complex RFC and a student model being a more compact RFC. A more compact RFC may have fewer trees and/or have each tree with fewer leaves. Since such leaves correspond to editing instructions, the student RFC in the Chalom-Aslan system has fewer leaves, and therefore fewer editing instructions than the teacher RFC. EX-1003, ¶¶312-315.

8. Claim 9

Claim 9 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above) and adds that the two claimed MLAs are of different types. EX-1003, ¶¶316-317. As was explained in Section VIII.C.1, it would have been obvious for Chalom-Aslan to have the teacher model as a neural network and the

student model as a random forest. *See, e.g.*, EX-1005, 9:13-17; EX-1007, ¶22; EX-1003, ¶318.

Therefore, Chalom in view of Aslan discloses “the first machine learning algorithm [e.g., teacher model, such as a NN] is of a different machine learning algorithm type than the second machine learning algorithm [e.g., student model, such as a random forest].” EX-1003, ¶¶316-319.

9. Claim 10

As just explained, in Chalom-Aslan, the first MLA can be a NN, so claim 10 is rendered obvious. EX-1003, ¶¶320-321.

10. Claim 11

As just explained, in Chalom-Aslan, the second MLA can be a regression forest, so claim 11 is rendered obvious. EX-1003, ¶¶322-323.

11. Claim 12

Claim 12 depends from claim 1 (which is disclosed by Chalom-Aslan for reasons provided above). Because Chalom is directed to eye gaze correction and Chalom’s “target feature comprises... an eye region comprising at least part of an eye,” claim 12’s additional limitation is rendered obvious. *Supra* Sections VIII.C.2.c-VIII.C.2.d; EX-1003, ¶¶324-325.

12. Claim 13

a. 13[P]

13[P] (to the extent it is limiting) focuses on runtime use of the reference data from claim 1. EX-1003, ¶327.

Chalom discloses the same in its Figure 3, which teaches “[a] machine implemented **method** for providing **eye contact correction**.” EX-1005, claim 1.

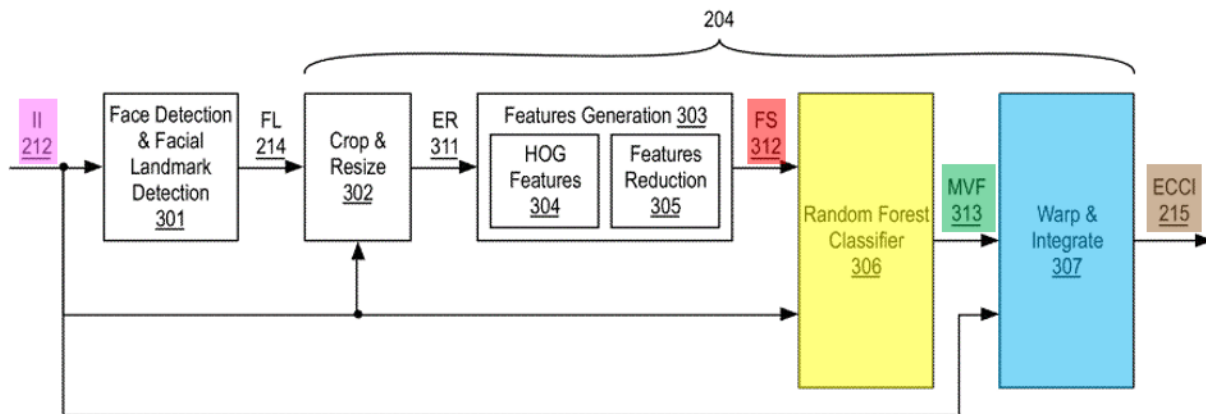


FIG. 3

Id., Fig. 3; EX-1003, ¶¶327-329.

b. 13[A]

As shown below, Chalom discloses receiving input image 212 (pink).

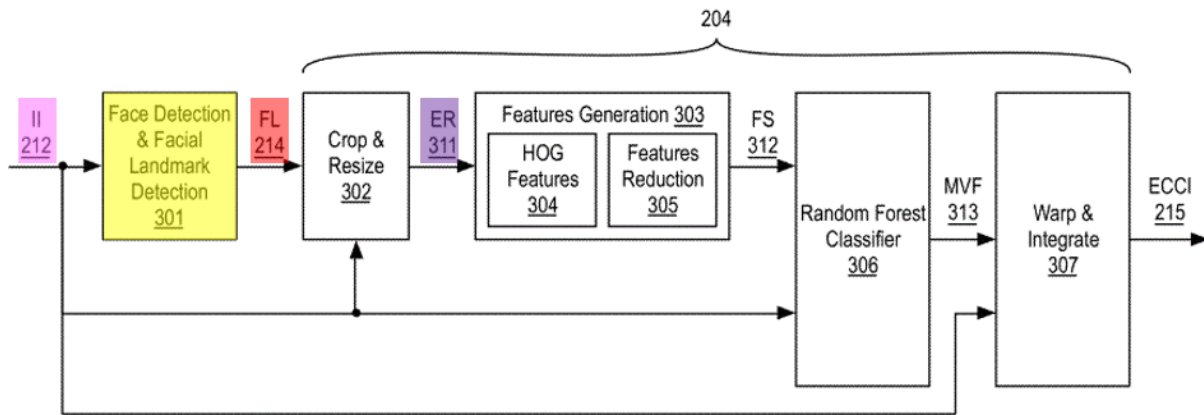


FIG. 3

EX-1005, Fig.3; 7:48-52.

Chalom's Fig. 3 also illustrates runtime use of trained RFC 306 in Chalom-Aslan, where Aslan's teacher-student architecture is adopted for training Chalom's MLA, e.g., RFC 306. *Supra* Section VIII.C.1; EX-1003, ¶332.

Therefore, Chalom-Aslan discloses "receiving a digital representation of a head region [e.g., input image 212 having a face]." EX-1003, ¶330-333.

c. 13[B]

Chalom-Aslan uses a student model to generate motion vector fields during training. *Supra* Section VIII.C.2.f. Thus, by using the "second selection instructions" (e.g., the trained student model, which is part of the "second reference data" generated during training), Chalom-Aslan discloses using the "second reference data" generated by the method of claim 1, i.e., generated **during training**, for gaze redirection **during runtime**. EX-1003, ¶¶334-336.

13. Claim 14

As shown in Fig. 16 below, Chalom discloses that “face detection module 202, facial landmark detection module 203, eye contact correction module 204, and video compression module 205 may be implemented via **central processor 1601** [yellow].” EX-1005, 15:47-50.

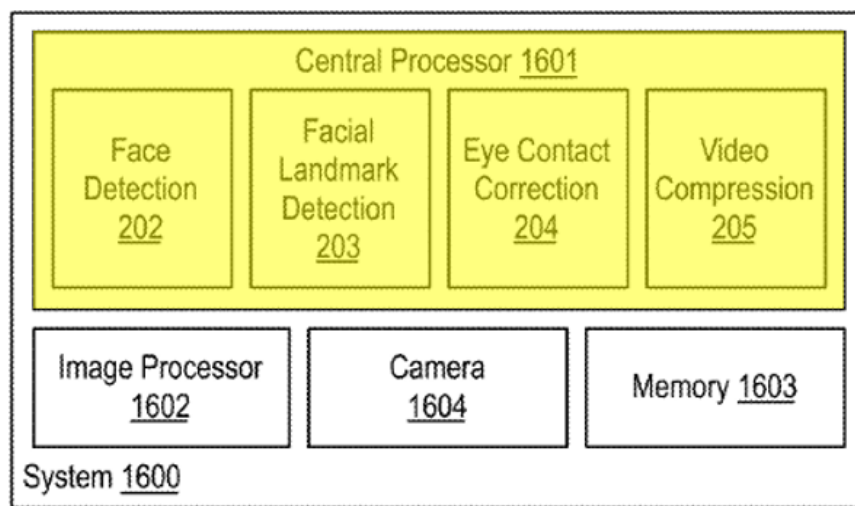


FIG. 16

Id., Fig. 16; EX-1003, ¶339.

Aslan also discloses, in Figure 8 below, “an exemplary computing system environment 800 for implementing the joint training techniques and systems.” EX-1007, ¶65.

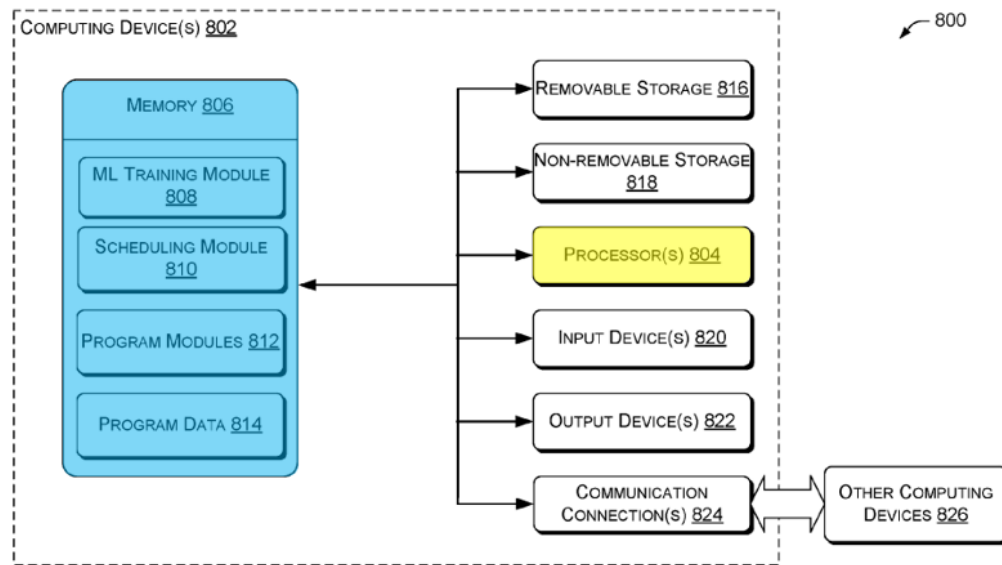


Fig. 8

Id., Fig.8. As shown, Aslan discloses a computing device including “one or more processors 804 [yellow] and computer-readable memory 806 [blue].” *Id.*, ¶66.

“The processor(s) 804 can be configured to execute instructions, applications, or programs stored in the memory 806.” *Id.*; EX-1003, ¶340.

Therefore, in view of these teachings, it would have been obvious to run the Chalom-Aslan teacher-student architected training process on a processor. Claim 14 is obvious. EX-1003, ¶¶337-341.

14. Claim 16

a. 16[P]

Most of 16[P] is identical to 1[P] and is thus rendered obvious by Chalom-Aslan. *Supra* Section VIII.C.2.a; EX-1003, ¶343.

As to the “device” and “processing unit” recitations, Chalom discloses in Fig. 14 “a flow diagram illustrating an example process 1400 for **pretraining an eye contact correction classifier.**” EX-1005, 13:66-67.

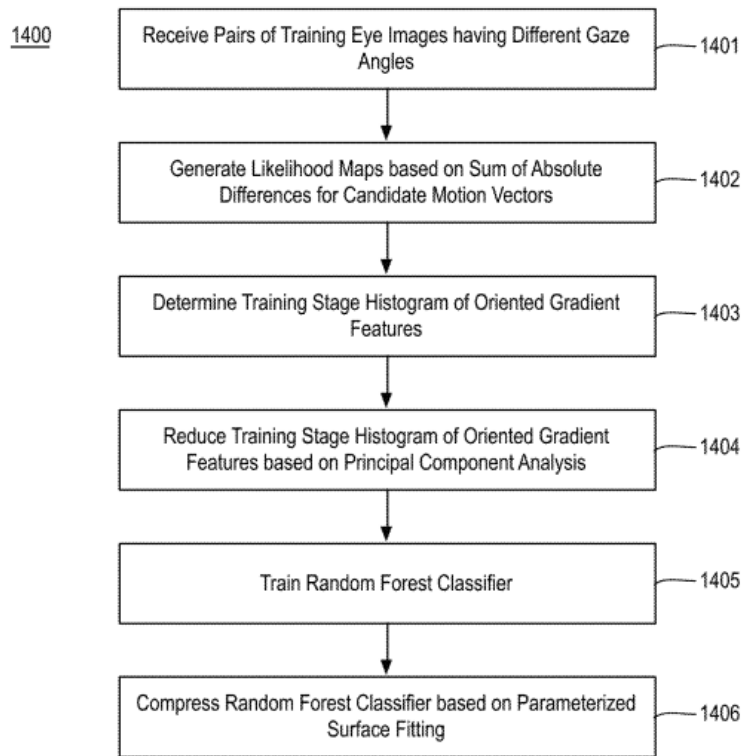


FIG. 14

Id., Fig.14. “[P]rocess 1400 may be performed by system 1600” as shown in Fig. 16 below. *Id.*, 14:8-10.

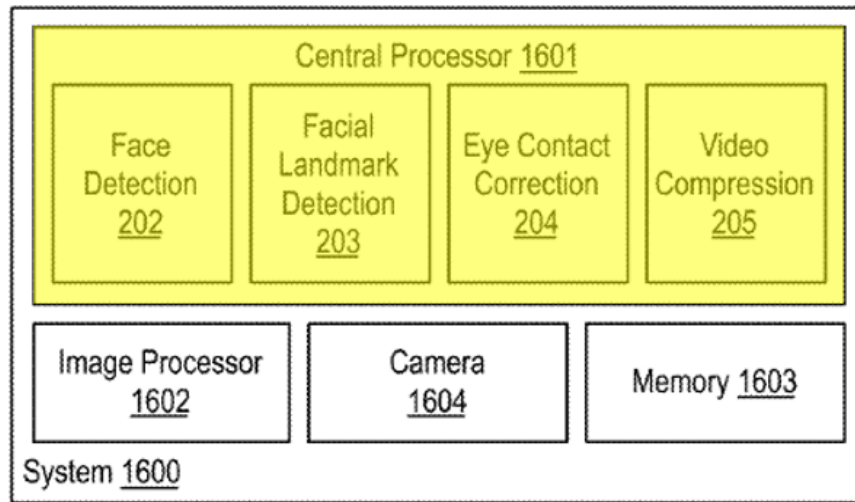


FIG. 16

EX-1005, Fig.16; 15:31-40.

In view of the above, it would have been obvious to have a data processing unit on a device arranged to perform the training steps of the combined Chalom-Aslan system. EX-1003, ¶¶343-345.

b. 16[A]-[C]

16[A]-[C] collectively are substantively identical to 1[A]-1[G] and are disclosed by Chalom-Aslan. *Supra* Sections VIII.C.2.b-VIII.C.2.f; EX-1003, ¶¶347-348.

c. 16[D] and 16[F]

16[D] and 16[F] are substantively identical to 1[D] and are disclosed by Chalom-Aslan. *Supra* Section VIII.C.2.e; EX-1003, ¶¶349-350.

d. 16[E] and 16[G]

16[E] and 16[G] are substantively identical to 1[E] and are disclosed by Chalom-Aslan. *Supra* Section VIII.C.2.e; EX-1003, ¶¶351-352.

e. 16[H]

16[H] recites that “the second editing instructions in the second reference data are provided in a compressed representation,” which is disclosed by Chalom-Aslan because the second editing instructions can be the PCA components of the first editing instructions, which means that they are provided in a compressed representation. *Supra* Section VIII.C.6; EX-1003, ¶¶353-356.

15. Claim 17

a. 17[P] and 17[A-E]

17[P] and 17[A-E] are identical to 1[P] and 1[A-E], respectively, and are thus taught by Chalom-Aslan. *Supra* Sections VIII.C.2.a-VIII.C.2.e; EX-1003, ¶¶358-359.

b. 17[F]

The teacher model in the Chalom-Aslan system could be a NN. *Supra* Section VIII.C.1. While Section VIII.C.1 acknowledges two ways of combining Chalom with Aslan, a PHOSITA would have found it obvious to implement the implementation in a third way, i.e., the student model also being a NN. *See* EX-1005, 9:13-17 (“Although illustrated with respect to random forest classifier 306,

any suitable pretrained classifier may be applied...”); EX-1003, ¶¶360-361.

According to Aslan, the student model “can comprise a much smaller machine learning model (e.g., a neural net with **1000 times fewer parameters** than the first model 100) that has the **size and/or speed that is advantageous** at run-time in particular scenarios.” EX-1007, ¶44. To achieve Aslan’s objective to convert a bigger but slower teacher NN into a smaller and faster student NN, the student NN would have had fewer layers, smaller convolutional fields, or both. EX-1003, ¶¶362-363.

16. Claim 18

a. 18[P] and 18[A-E]

18[P] and 18[A-E] are identical to 1[P] and 1[A-E], respectively, and are thus taught by Chalom-Aslan. *Supra* Sections VIII.C.2.a-VIII.C.2.e; EX-1003, ¶¶365-366.

b. 18[F]

18[F] is identical to an aspect in claim 12, so it is taught by Chalom-Aslan (*supra* Section VIII.C.11). EX-1003, ¶¶367-368.

17. Claim 19

a. 19[P] and 19[A-E]

19[P] and 19[A-E] are identical to 1[P] and 1[A-E], respectively, and are taught by Chalom-Aslan. *Supra* Sections VIII.C.2.a-VIII.C.2.e; EX-1003, ¶¶370-

371.

b. 19[F]

19[F] is substantively identical to 16[H] and is taught by Chalom-Aslan.

Supra Section VIII.C.14.e; EX-1003, ¶372.

c. 19[G]

Chalom-Aslan teaches the “principle component analysis representation” in 19[G], as was explained in Section VIII.C.6. EX-1003, ¶373.

D. GROUND #4: Claim 4 is rendered obvious by Chalom in view of Aslan and Grau

1. Claim 4

Claim 4 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above), and adds that “the editing instructions comprise a filter field, a brightness adjustment field, or a texture blending field,” which is in Grau. EX-1003, ¶¶374-375.

While Chalom discloses “displacement vector fields” as “editing instructions” (*supra* Section VIII.C.4), Grau discloses two of the other three options in claim 4: a “filter field,” and a “brightness adjustment field” in the context of IR-to-color mapping and IR-and-lighting mapping. EX-1003, ¶¶376-377.

Regarding IR-to-color mapping, Grau discloses that “the IR to color conversion may be performed by a mapping function **using a limited**

neighborhood of pixels (such as 1, 4, 8, higher order) to form a combination value (RGB) for a single color value for a pixel.” EX-1009, ¶79. EX-1003, ¶378.

Regarding the IR-and-lighting mapping, Grau uses “a neural network... to map the IR [infa-red] and lighting from non-occluded parts of the face.” EX-1009, ¶80. “Neural networks (like CNNs) can take multiple inputs, ...and maps this to the anticipated facial image.” *Id.* Changing the lighting/brightness of the occluded parts of the face to match those of the non-occluded part of the face is a lighting adjustment operation, and the lighting/brightness adjustment amount is a “brightness adjustment field.” EX-1003, ¶¶379-380.

2. Motivation to Combine Chalom-Aslan with Grau

Grau, like Chalom and Aslan, qualifies as analogous art to the '985 patent as it is in the **same field of endeavor**, *i.e.*, using machine learning for image processing. EX-1009, ¶79; EX-1001, Abstract; EX-1005, 3:21-32; EX-1007, ¶24. At a minimum, Grau is reasonably pertinent to at least one problem identified by the '985 Patent. EX-1003, ¶¶381-382.

Although Chalom discloses a “warp and integrate module 307,” it does not detail how “the warped eye regions may be integrated into the remaining portion of input image.” EX-1005, 9:26-38. A PHOSITA seeking to integrate with high quality the warped eye regions would have been motivated to look to analogous references, such as Grau, for more detailed teachings on how to achieve this, such

as by “mapping IR and lighting from non-occluded parts of the face by using convolutional neural networks (CNNs).” EX-1009, ¶42. A PHOSITA would also have been **motivated** to apply the Chalom-Aslan system to additional image-optimization tasks that complement the basic gaze-correction task. Chalom explicitly motivates a PHOSITA to employ its “techniques and/or arrangements... in a variety of other systems and applications other than what is described.” EX-1005, 2:14-17, 24:1-5. Thus, a PHOSITA would have been motivated to expand the application of Chalom-Aslan and would have looked to Grau for specific facial-image editing instructions beyond displacement vector fields or even beyond gaze correction. EX-1003, ¶¶383-385.

Finally, a PHOSITA would have had a **reasonable expectation of success** in using the Chalom-Aslan system to perform the facial-image editing tasks described in Grau. Like the gaze correction in the Chalom-Aslan system, Grau’s “IR to color conversion” and “IR and lighting” mapping are also implemented via machine learning. EX-1009, ¶¶79-80. A PHOSITA would have understood that applying Chalom-Aslan to Grau’s facial image editing merely involves modifying Chalom’s warping algorithm; training data and objective functions could stay the same. To the extent that more robust “IR and lighting” mapping is expected, Grau discloses the type of training data that can be used—“a training set that contains many examples of different persons under different lighting conditions.” EX-

1009, ¶¶80. Thus, it would have been straightforward to implement Grau's teachings into Chalom-Aslan. EX-1003, ¶¶386-387.

E. GROUND #5: Claim 7 is rendered obvious by Chalom-Aslan and Pace

1. Claim 7

Claim 7 (depends from claim 5, which is disclosed by Chalom-Aslan for reasons provided above) is like claim 6 except wavelet transformation replaces principal component analysis. EX-1003, ¶¶388-389.

Pace discloses using discrete wavelet transform (DWT) as a “[d]ata reduction technique” used to make “the input data... sparser.” EX-1010, 21:4-11. This is “because data is often more sparse in the wavelet domain” and “DWT transformation is the least ‘invasive’ and theoretically allows full recovery of the input data.” *Id.*, 21:7-13. A PHOSITA would have found it obvious to use wavelet transformation for compressing the editing instructions in Chalom-Aslan. EX-1003, ¶¶390-391.

2. Motivation to Combine Chalom-Aslan with Pace

Pace, like Chalom and Aslan, qualifies as analogous art to the '985 Patent because it is in the **same field of endeavor**. EX-1010, 21:4-11, EX-1001, Abstract; EX-1005, 3:21-32; EX-1007, ¶24. At a minimum, Pace is at least reasonably pertinent to a problem identified in the '985 Patent. EX-1003, ¶¶392-393.

While Chalom-Aslan utilizes PCA to compress the “first editing instructions,” a PHOSITA would have found it obvious to replace PCA with wavelet transformation. For example, Chalom teaches that “[t]he histogram of oriented gradient features may be reduced **using any suitable technique or techniques,**” and that PCA is just one example. EX-1005, 8:51-53. Thus, a PHOSITA would have been motivated to look to other references, such as Pace, where different compression techniques are used. EX-1003, ¶394.

In addition, both PCA and DWT were widely used as alternative data-compression techniques before the '985 Patent. *See* EX-1040, pp.163-164; EX-1041. Pace itself discusses both PCA and DWT. EX-1010, 17:49-52, 20:26-29. “Of the data reduction techniques, DWT transformation is the least ‘invasive’ and theoretically allows full recovery of the input data.” EX-1010, 21:11-13. Moreover, Chalom notes that PCA is a machine learning process and requires training. EX-1005, 8:64-9:1. In contrast, DWT is a deterministic process and does not involve training (and therefore easier to implement). Thus, a PHOSITA would have been motivated to replace the PCA compression in the Chalom-Aslan system with another suitable technique of DWT, which is a less invasive data compression technique to be implemented more easily. EX-1003, ¶395.

Finally, a PHOSITA would have had a reasonable expectation of success in replacing Chalom-Aslan’s PCA with Pace’s DWT. As mentioned above, PCA and

DWT were widely used as alternatives and using one or the other would have been well within a PHOSITA's capabilities, and since DWT does not require training, replacing PCA with DWT simplifies compression calculations. EX-1003, ¶¶396-397

IX. SECONDARY CONSIDERATIONS

The Examiner did not rely on any secondary considerations during prosecution, so this petition is not required to address any purported secondary considerations. *See, e.g., Zurn Industries, LLC v. Sioux Chief MFG. Co., Inc.*, IPR2018-00975, Paper 9 at 39 (PTAB Nov. 8, 2018).

X. CONCLUSION

The Board should institute a trial and cancel the Challenged Claims as unpatentable.

Dated: June 16, 2023

Respectfully Submitted,

/ Alexander B. Stein /
Alexander B. Stein (Reg. No. 71,397)

U.S. PATENT NO. 10,740,985 – Claim Listing

No.	Claim Elements
1[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
1[A]	receiving training data comprising:
1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
1[F]	wherein the first reference data comprise first editing instructions for a range of possible configurations of the target feature and first selection instructions for selecting editing instructions for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch; and
1[G]	wherein the second reference data comprise second editing instructions for a range of possible configurations of the target feature and second selection instructions for selecting editing instructions for a particular input patch from the second image editing instructions based on the configuration of the target feature of the input patch.
2[P]	The method of claim 1, wherein
2[A]	the configuration of the target feature of each input patch is represented by a feature vector derived from plural local descriptors of the input patch, and

2[B]	the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch.
Claim 3	3. The method of claim 1, wherein the editing instructions comprise a displacement vector field defining how the input patch is to be transformed.
Claim 4	4. The method of claim 1, wherein the editing instructions comprise a filter field, a brightness adjustment field, or a texture blending field.
5[P]	5. The method of claim 1, wherein:
5[A]	a first editing algorithm is used by the first machine learning algorithm to define how the first editing instructions are to be applied to an input patch to derive an output patch; and
5[B]	a second editing algorithm is used by the second machine learning algorithm to define how the second editing instructions are to be applied to an input patch to derive an output patch.
6[P]	6. The method of claim 5, wherein:
6[A]	the second editing instructions are principle component analysis components of a principle component analysis of the first editing instructions; and
6[B]	the second image editing algorithm is configured to transform the second editing instructions into the first editing instructions by inverse principle component analysis transform.
7[P]	7. The method of claim 5, wherein:
7[A]	the second editing instructions are wavelet components of the first editing instructions; and
7[B]	the second editing algorithm is configured to transform the second editing instructions into the first editing instructions by inverse wavelet transform.
Claim 8	8. The method of claim 1, wherein the first selection instructions for the first reference data are able to select between a larger number of alternative editing instructions than the second selection instructions for the second reference data.
Claim 9	9. The method of claim 1, wherein the first machine learning algorithm is of a different machine learning algorithm type than the second machine learning algorithm.
Claim 10	10. The method of claim 1, wherein the first machine learning algorithm comprises one or more of the following: a neural network; a support vector machine; and a generative adversarial network (GAN).

Claim 11	11. The method of claim 1, wherein the second machine learning algorithm comprises one or more of the following: a regression forest; regression ferns, cluster centres, a lookup table, and separable filter banks.
Claim 12	12. The method of claim 1, wherein the target feature comprises one or more of the following: an eye region comprising at least part of an eye, a nose region comprising at least part of a nose, a mouth region comprising at least part of a mouth, a chin region comprising at least part of a chin, a neck region comprising at least part of a neck, and a hair region comprising hair.
13[P]	13. A method of adjusting a digital representation of a head region, the method comprising
13[A]	receiving a digital representation of a head region; and
13[B]	using reference data comprising editing instructions to adjust the digital representation of the head region, wherein the reference data comprises the second reference data generated by the method of claim 1.
Claim 14	14. A non-transitory computer readable storage medium storing a computer program capable of execution by a processor and arranged on execution to cause the processor to perform a method according to claim 1.
16[P]	16. A device for generating reference data for adjusting a digital representation of a head region, the device comprising a data processing unit arranged to:
16[A]	receive training data comprising:
16[B]	a set of input patches, each input patch comprising information about a target feature of a digital representation of the head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
16[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target portion of the digital representation of the head region after adjustment of the digital representation of the head region;
16[D]	use a first machine learning algorithm to generate first reference data using the training data; and
16[E]	use a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data output by the first machine learning algorithm

16[F]	wherein the first reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
16[G]	wherein the second reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region, and
16[H]	wherein the second editing instructions in the second reference data are provided in a compressed representation.
17[P]	17. A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
17[A]	receiving training data comprising:
17[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
17[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
17[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
17[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
17[F]	wherein the first machine learning algorithm comprises a first neural network and the second machine learning algorithm comprises a second neural network, and wherein the second neural network comprises fewer layers and/or smaller convolution fields than the first neural network.
18[P]	18. A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
18[A]	receiving training data comprising:
18[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital

	representation of the head region, wherein the target feature is the same for each input patch; and
18[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
18[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
18[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
18[F]	wherein the target feature comprises an eye region comprising at least part of an eye and the adjustment of the digital representation of the head region comprises adjusting a gaze direction, and/or wherein the target feature comprises a nose region comprising at least part of a nose and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the nose, and/or wherein the target feature comprises a chin region comprising at least part of a chin and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the chin, and/or wherein the target feature comprises a neck region comprising at least part of a neck and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the neck, and/or wherein the target feature comprises a hair region comprising hair and the adjustment of the digital representation of the head region comprises adjusting a color of the hair.

19[P]	19. A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
19[A]	receiving training data comprising:
19[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
19[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
19[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
19[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
19[F]	wherein the second editing instructions in the second reference data are provided in a compressed representation, and
19[G]	wherein the compressed representation comprises one or more of the following: a principle component analysis representation; a wavelet representation; Fourier and/or discrete cosine transform components; cluster centers.

CERTIFICATION OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Petition includes 13,997 words as counted by Microsoft Word and is therefore in compliance with the 14,000-word limit established by 37 C.F.R.

42.24(a)(1)(i). Accordingly, pursuant to 37 C.F.R. 42.24(d), lead counsel for the Petitioner hereby certifies that this petition complies with the type-volume limits established for a petition requesting IPR.

Dated: June 16, 2023

Respectfully Submitted,

/ Alexander B. Stein /

Alexander B. Stein (Reg. No. 71,397)

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. 42.6(4) and 42.105, lead counsel for Petitioner hereby certifies that on June 16, 2023, copies of this Petition and all supporting exhibits were sent via Priority Mail Express to the correspondence address of record for the '985 patent:

RealD Spark, LLC - Patents
Stefan Bump
1930 Central Avenue
Suite A-2
Boulder, CO 80301

A courtesy copy of this Petition and supporting exhibits was also served via email on Patent Owner's counsel of record in *RealD Spark, LLC v. Microsoft Corporation*, WDWA-2-22-cv-00942:

Ashley N Moore (amoore@mrlp.com)
Kaleigh N Boyd (kboyd@tousley.com)
Jennifer A Mauri (jmauri@mrlp.com)
Jessica E Pharis (jpharis@mrlp.com)

Dated: June 16, 2023

Respectfully Submitted,

/ Alexander B. Stein /
Alexander B. Stein (Reg. No. 71,397)