

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSARA INC.,
Petitioner

v.

MOTIVE TECHNOLOGIES, INC.,
Patent Owner

Case IPR2026-00034
U.S. Patent No. 12,136,276

**DECLARATION OF TREVOR J. DARRELL, PH.D.,
IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF
U.S. PATENT NO. 12,136,276**

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Samsara EX1003
Samsara v. Motive Tech. Inc.
IPR2026-00034

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MY BACKGROUND AND QUALIFICATIONS	3
III.	UNDERSTANDING OF APPLICABLE PATENT LAW	6
	A. Claim Construction.....	6
	B. Obviousness.....	7
IV.	TECHNOLOGY OVERVIEW.....	9
	A. Horizon Line Detection and Camera Calibration.....	9
	B. Overlaying and Modifying Annotations in Roadway Images	11
	C. Dynamic Recalibration Using Modified Lines	11
	D. Network-Based Transmission of Images and Calibration Data.....	12
	E. Transmission of Annotated or Overlaid Images	13
	F. Remote Calibration and GUI-Based Processing	14
	G. Verification and Consistency Across Annotators	15
	H. Human-in-the-Loop Calibration by Remote Experts.....	16
	I. Bandwidth-Efficient Transmission of Overlay Data	17
	J. Prior Art Relied on for the Grounds of Unpatentability	18
	1. Choe discloses a system that calibrates a camera for lane detection using a predicted horizon line and annotations.....	18
	2. Davies discloses a system that calibrates a vision-based sensor of a vehicle by transmitting roadway images to an external computing device, processing the images remotely, and then transmitting modified calibration data back to the vehicle.....	23
	3. Westmacot discloses a system for annotating roadway images captured by vehicle-mounted cameras, generating 3D road models, overlaying annotations such as lane boundaries, and computing camera parameters based on the detected or user- modified lines.....	27
	4. Tal discloses a system for capturing roadway images with vehicle-mounted devices, overlaying detected object data onto the images, transmitting the resulting overlaid images and	

associated data to remote servers over wireless or cellular networks, and providing access to the transmitted roadway and incident data through a web-based application..... 32

5. Kuehnle discloses a system that calibrates a camera based on time and the speed of the vehicle..... 34

V. PROSECUTION HISTORY 36

VI. PERSON OF ORDINARY SKILL IN THE ART 38

VII. GROUNDS OF UNPATENTABILITY..... 38

VIII. GROUNDS 1: CHOE RENDERS OBVIOUS CLAIMS 1-3, 5-10, 12-16, 18-20. 39

A. Independent Claim 140

1. [1Pre] “A method comprising” 40

2. [1A] “receiving an image of a roadway recorded by a camera device installed within a vehicle” 41

3. [1B] “detecting a horizon line in the image” 42

4. [1C] “overlaying a line on the image to generate an overlaid image” 48

5. [1D] “transmitting the overlaid image to a computing device over a network” 49

6. [1E] “receiving a modification of the line from the computing device, the modification comprising a new line at a second position” 55

7. [1F] “computing a camera parameter based on the new line”. 60

8. [1G] “transmitting data representing the camera parameter to the camera device.” 63

B. Claim 2: The method of claim 1, wherein the camera parameter comprises one of camera height, viewing angle, and road plane normal.66

C. Claim 3: The method of claim 1, wherein receiving the image of the roadway comprises receiving an image of the roadway while the vehicle is moving.....72

D.	Claim 5: The method of claim 1, wherein transmitting the overlaid image to the computing device over the network comprises transmitting the overlaid image to a web-based application.	74
E.	Claim 6: The method of claim 1, wherein receiving the modification of the line from the computing device comprises receiving an identification of a new horizon line by a user of the computing device.	79
F.	Claim 7: The method of claim 1, wherein transmitting the data representing the camera parameter to the camera device further comprises recomputing the camera parameter based on the modification of the line.	83
G.	Claims 8-10, 12-16 18-20.....	85
IX.	GROUND 2: CHOE IN VIEW OF DAVIES RENDERS OBVIOUS CLAIMS 1-3, 5-10, 12-16 AND 18-20.	87
A.	Motivation to Combine	89
1.	How and Why it would have been obvious to integrate transmission and external calibration	90
2.	How and Why it would have been obvious to enable remote display of overlaid calibration images.....	95
3.	How and Why it would have been obvious to complete the interactive calibration loop remotely	96
4.	How and Why it would have been obvious to extend human-in-the-loop calibration to remote operators.....	97
5.	How and Why it would have been obvious to optimize calibration through bandwidth-efficient overlay transmission	98
6.	Reasonable Expectation of Success.....	99
B.	Independent Claim 1	101
1.	[1Pre] -1[C], [1F] and [1G].....	102
2.	[1D]: “transmitting the overlaid image to a computing device over a network”	102
(a)	Choe.....	102
(b)	Davies	103
(c)	KSR.....	104

3.	[1E]: “receiving a modification of the line from the computing device, the modification comprising a new line at a second position”	109
	(a) Choe.....	109
	(b) Davies	110
	(c) KSR.....	110
C.	Claims 2-3 and 5-7	115
D.	Claims 8-10, 12-16, and 18-20.....	123
X.	FOUNDATIONS 3A/3B: CHOE IN VIEW OF KUEHNLE (GROUND 3A) AND CHOE IN VIEW OF DAVIES AND FURTHER IN VIEW OF KUEHNLE (GROUND 3B) RENDER OBVIOUS CLAIMS 4, 11, AND 17.....	123
XI.	GROUND 4: WESTMACOT IN VIEW OF TAL RENDERS OBVIOUS CLAIMS 1-20.....	128
A.	Westmacot.....	128
B.	Tal.....	129
C.	Motivation to Combine	132
	1. How and Why it would have been obvious to transmit Westmacot’s annotated images and parameters to remote servers for scalable training and calibration.....	135
	2. How and Why it would have been obvious to integrate Tal’s remote processing with Westmacot’s annotation interface for collaborative line modification	136
	3. How and Why it would have been obvious to apply Tal’s bidirectional communication model to transmit recalculated parameters back to the vehicle’s camera device.....	136
	4. Reasonable Expectation of Success	137
D.	Independent Claim 1	138
	1. [1Pre] “A method comprising”	138
	2. [1A] “receiving an image of a roadway recorded by a camera device installed within a vehicle”	138
	3. [1B] “detecting a horizon line in the image”	140
	4. [1C] “overlaying a line on the image to generate an overlaid image”	142

5.	[1D] “transmitting the overlaid image to a computing device over a network”	146
	(a) Westmacot	146
	(b) Tal	147
	(c) KSR.....	149
6.	[1E] receiving a modification of the line from the computing device, the modification comprising a new line at a second position;.....	151
	(a) Westmacot	151
	(b) Tal	153
	(c) KSR.....	154
7.	[1F] computing a camera parameter based on the new line;. 157	
	(a) Westmacot	157
	(b) Tal	158
	(c) KSR.....	159
8.	[1G] transmitting data representing the camera parameter to the camera device.....	162
	(a) Westmacot	162
	(b) Tal	163
	(c) KSR.....	164
E.	Claim 2	166
F.	Claim 3	168
G.	Claim 4	169
	1. Westmacot.....	169
	2. Tal	171
	3. KSR.....	171
H.	Claim 5: The method of claim 1, wherein transmitting the overlaid image to the computing device over the network comprises transmitting the overlaid image to a web-based application.	173
	1. Westmacot.....	173
	2. Tal	174

3.	KSR.....	175
I.	Claim 6	178
1.	Westmacot.....	178
2.	Tal	180
3.	KSR.....	180
J.	Claim 7	184
1.	Westmacot.....	184
2.	Tal	185
3.	KSR.....	186
K.	Claims 8-20	188
XII.	OBJECTIVE INDICIA OF NONOBVIOUSNESS	189
XIII.	THE ART CITED IN THE GROUNDS ARE SUBSTANTIVELY DIFFERENT THAN AND NOT CUMULATIVE TO THE REFERENCES CITED DURING PROSECUTION.	190
XIV.	CONCLUSION.....	194

I. INTRODUCTION

I, Trevor J. Darrell, declare as follows:

1. I have been retained on behalf of Samsara Inc. for the above-captioned *inter partes* review proceeding. I understand that this proceeding involves U.S. Patent No. 12,135,276 (“the ’276 patent”), which is titled “Camera Initialization for Lane Detection and Distance Estimation Using Single-View Geometry.”

2. I understand the ’276 patent is assigned to Motive Technologies, Inc. I also understand that the ’276 patent names three inventors: Ali Hassan, Ahmed Ali, and Syed Wajahat Ali Shah Kazmi. EX1001, (72).

3. I understand the ’276 patent is a continuation of U.S. Patent No. 11,875,580 (“the ’580 patent”), which is involved in a separate *inter partes* review proceeding. I previously submitted a declaration explaining why all challenged claims of the ’580 patent were unpatentable.

4. I understand the ’276 patent has been provided as Exhibit 1001. I will cite to the specification using the following format: EX1001, 1:1-10. This citation points to the specification of the ’276 patent at column 1, lines 1-10.

5. I have reviewed, and am familiar with, the file history of the ’276 patent. The file history of the ’276 patent has been provided as Exhibit 1002. A summary of my understanding of the prosecution history is in Section V.

6. My opinions are based on more than 30 years of working with computer vision and pattern recognition, which includes decades of experience teaching and researching predictive models for computer vision applications. My opinions are also based on investigation and study of the relevant materials, including the patent at issue and its file history, and the prior art.

7. For example, I have reviewed and am familiar with the following prior art used in the petition for inter partes review of the '276 patent:

- U.S. Publication No. 2020/0410704 to Choe et al. (EX1005) and its filed figures (EX1006);
- U.S. Publication No. 2014/0240500 to Davies (EX1007);
- International Publication No. WO 2019/175286 to Westmacot (EX1008);
- U.S. Publication No. 2022/0019829 to Tal (EX1009); and
- International Publication No. WO 2009/027090 to Kuehnle (EX1010).

8. I have been asked to provide my technical review, analysis, insights, and opinions about the '276 patent and the above-listed references that form the basis for the grounds of unpatentability.

9. A full list of documents I considered is in my Materials Considered, which is attached to this declaration as Appendix A.

10. To the best of my knowledge, the accompanying exhibits are true and accurate copies of what they purport to be, and that an expert in the field would reasonably rely on them to formulate opinions such as those in this Declaration.

11. I may rely upon these materials and/or additional materials to rebut arguments raised by the Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

12. My analysis of the materials relevant to this proceeding is ongoing, and I will continue to review any new material as it is provided. This report presents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

13. I am being compensated at \$810.00 per hour for my work in this case. My compensation does not depend on the outcome of the case. The fact that I am being compensated has not altered the opinions in this declaration.

II. MY BACKGROUND AND QUALIFICATIONS

14. I am currently a Professor in the Computer Science Division (EECS) at UC Berkeley. I am also currently a member of the steering committee for the Berkeley Artificial Intelligence Research Center (BAIR). I was previously a Professor in the Department of Electrical Engineering and Computer Science at

MIT from 1999 to 2008 and a Group Leader at the International Computer Science Institute (ICSI) from 2008 to 2014.

15. I have been researching and/or teaching in the area of computer science, with a particular focus on computer vision, machine learning, computer graphics, and perception-based human-computer interfaces for over 30 years. I currently lead a laboratory where our current research focuses on algorithms to enable visual recognition across a variety of platforms and applications. Additionally, I am the faculty director of UC Berkeley's DeepDrive Center, an industrial alliance focusing on deep learning and autonomous driving research.

16. I am a past or current member of a number of scientific societies that focus on computer vision and machine learning, including the DARPA Information Science and Technology Advisory Study Group (2003-2014) and various National Science Foundation (NSF) Panels (2002-present). I have been invited to chair numerous conferences related to computer vision and machine learning. For example, I have served as both the Area Chair and Program Chair for the Conference on Computer Vision and Pattern Recognition (CVPR), the Area Chair for the International Conference on Computer Vision (ICCV), the Area Chair and General Chair of the International Conference on Multimodal Interfaces (ICMI), the Area Chair for European Conference on Computer Vision (ECCV), the Area Chair for the International Conference on Machine Learning (ICML), the Area

Chair for the International Conference on Learning Representations (ICLR), and the Program Chair for the Conference on Perceptual User Interfaces (PUI).

17. I have been an editor for a number of leading journals in the field of computer vision and artificial intelligence, such as the Artificial Intelligence Journal (2004-2011) and the IEEE Transactions Pattern Analysis and Machine Intelligence Journal (2005-2011, 2015-2017).

18. I have published over 300 articles and conference papers related to computer algorithms, computer vision, autonomous vehicles, and image processing. My publications have won several awards and honors, including the International Conference on Machine Learning (ICML) Test of Time Award in 2024, the Computer Vision Foundation Longuet-Higgins Prize in 2024, the Association for Computer Machinery SIGMM Test of Time Paper Award in 2024, the North American Associate for Computation Linguistics Best Paper award in 2016, the Computer Vision and Pattern Recognition Best Paper Honorable Mention in 2015, the ACM Multimedia Best Open Source Software Award in 2014, and the Internet Vision Workshop Best Paper Award in 2008. In 2016, my team also won the Everingham Prize at the International Conference on Computer Vision, recognizing our work in providing a significant contribution to the computer vision community. To date, my publications have been cited by other scholars over 269,000 times.

19. I am a named inventor on several patents related to object recognition and image processing. I have also served as a consultant in the area of computer vision for a number of start-up and established companies, including Pinterest (IPO), IQ Engines (acquired by Yahoo), and BotSquare/Flutter (acquired by Google). I continue to advise a number of startup companies including Nexar and SafelyYou. I also have previously served as a consulting and/or testifying expert witness in a number of patent infringement cases, including two ITC investigations, as described more fully in my CV (EX1004).

III. UNDERSTANDING OF APPLICABLE PATENT LAW

20. I am not a lawyer. However, I have been advised that technical experts should apply certain legal standards in forming their opinions.

A. Claim Construction

21. I understand that claim terms should be interpreted under a claim construction standard called *Phillips*. I understand that, according to the *Phillips* standard, claim terms should be construed according to the terms' meaning as understood by a person having ordinary skill in the art (POSITA) at the time of the invention. I understand that the claim terms should be construed in light of the specification. I also understand that the prosecution history of the patent should be consulted in determining claim meaning. In this declaration, I have construed the

terms according to their ordinary and customary meanings as understood by a POSITA in view of the specification and prosecution history.

B. Obviousness

22. I understand a patent claim is unpatentable if the claimed invention would have been obvious to a person of ordinary skill in the art (POSITA) at the time of the alleged invention. This means that a patent claim can be unpatentable even if all of the claim elements cannot be found in a single prior-art reference.

23. I understand an invention is obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious to a POSITA at the time of the invention.

24. I understand that to prove that prior art renders a patent claim obvious, it is necessary to: (1) identify the particular references that, singly or in combination, teach or suggest every element of the claim; (2) identify which claim elements appear in which reference; and (3) explain why a POSITA would have sought to combine the prior-art references.

25. I understand that when considering whether a patent claim is obvious, one should consider whether at the time of the alleged invention a teaching, suggestion, or motivation to combine the references existed so as to avoid

impermissibly applying hindsight. I understand this test should not be rigidly applied, but that the test can be important to avoid such hindsight.

26. I understand the following rationales can support a motivation to combine prior-art references:

- (A) Combining prior-art elements according to known methods to yield predictable results;
- (B) Simple substitution of one known element for another to obtain predictable results;
- (C) Use of known technique to improve similar devices (methods, or products) in the same way;
- (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- (E) Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and
- (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art.

27. I understand objective indicia (also called secondary considerations) can be important evidence as to whether a patent claim is obvious. Such indicia include: (1) commercial success of products covered by the patent claim; (2) a long-felt need for the invention; (3) failed attempts by others to make the invention; (4) copying of the invention by others; (5) unexpected results achieved

by the invention; (6) praise of the invention by the infringer or others in the field; (7) taking of licenses under the patent by others; (8) expressions of surprise by experts and those skilled in the art at the making of the invention; and (9) the patentee proceeded contrary to the accepted wisdom of the prior art.

IV. TECHNOLOGY OVERVIEW

28. The '276 patent relates to “analyzing sensor data in” automotive systems by detecting a horizon line in roadway images and using that line, along with other annotations, to compute camera parameters. EX1001, 1:14-16; 1:43-59. In particular, the patent describes receiving an image of a roadway recorded by a vehicle-mounted camera, detecting a horizon line, overlaying line(s) such as a lane boundary, transmitting the overlaid image to a computing device, allowing receiving a modification of the line(s) to those lines (including the horizon line) by a user from the computing device, recomputing camera parameters such as camera height, pitch angle (viewing angle), and road plane normal based on the new line, and transmitting the camera parameter back to the camera device. EX1001, 1:35-55. These claimed methods are directed to calibration workflows that were well known in the art prior to the filing of the '276 patent.

A. Horizon Line Detection and Camera Calibration

29. Detecting a horizon line in roadway images and using it to determine camera pitch, viewing angle, or road plane orientation was well established before

the '276 patent. For decades, researchers have recognized that the horizon line serves as a geometric constraint linking the 2D image plane to the 3D world, and that its position is directly determined by the camera's pitch and height. For example, Russell describes methods for "labeling horizon lines" and using them to compute the pitch angle and camera height from traffic images. EX1014, 1, 3, 5-6. Similarly, Bartl discloses "fully automatic horizon estimation" for surveillance and traffic cameras, with human annotators correcting predicted horizons to improve accuracy. EX1015, Abstract, 1. These references confirm that horizon detection and human-in-the-loop correction were routine calibration tools long before the '276 patent. A POSITA would have understood human-in-the-loop means that part of the process relies on a human operator to review, adjust, or correct the output of an automated system. In calibration tools, it refers to situations where the system may automatically detect a feature, such as a horizon line, but then a person verifies or fine-tunes that detection to ensure accuracy. For example, in horizon detection for camera calibration, software may generate an initial horizon line based on image processing. A human operator can then move, tilt, or adjust that line if the automatic detection is off. The corrected line is fed back into the calibration process, ensuring that the final camera parameters are accurate. In short, "human-in-the-loop correction" means a hybrid approach: the machine does the heavy lifting, but a human supervises and provides targeted adjustments where

needed, as would have been understood by a POSITA. *See generally, e.g.,*
EX1016.

B. Overlaying and Modifying Annotations in Roadway Images

30. The concept of overlaying lane lines, road boundaries, or horizon lines onto images for calibration and annotation was also conventional. For example, Westmacot discloses projecting lane boundaries and horizon lines into camera images and displaying them in an annotation interface, where users could shift or adjust the lines for improved accuracy. EX1008, FIGs. 6 (UI 412), 25. This paradigm (displaying overlays in a 2D camera view and allowing annotators to make corrections) was a well-understood step in generating reliable training data and calibration parameters. By the time of the '276 patent's filing, annotation platforms routinely incorporated both automatic detection and manual correction of horizon lines and lane markings.

C. Dynamic Recalibration Using Modified Lines

31. The re-computation of camera parameters based on line modifications was likewise known in the field. Westmacot describes a parameter computation component that calculates camera parameters including height (H), lateral offset (S), forward point, horizon line, and lane width (W), and updates those parameters when lines are adjusted. EX1008, 32:27-33:6 (Component 408). Other references, such as Chongqing, emphasize dynamic recalibration in response to deviations in

lane line geometry caused by bumps, slopes, or vibrations, and disclose recomputing extrinsic parameters using optimization methods. EX1017, ¶¶[0036]-[0042]. These disclosures confirm that recomputing camera parameters based on detected or modified lines was a predictable, conventional practice.

D. Network-Based Transmission of Images and Calibration Data

32. The notion of transmitting roadway images and calibration data to and from remote computing devices was also well known. Tal discloses capturing roadway images, overlaying detected features, and transmitting both raw and processed images, including overlaid images, to a remote server over cellular or wireless networks. EX1009, ¶¶[0027], [0030], [0052], [0075], [0138]. Tal further explains that users can access the transmitted images and calibration data via a web application, enabling distributed annotation, calibration, and review. EX1009, ¶¶[0005]-[0013], FIG. 12. A POSITA would have understood that applying these networked techniques to horizon lines and camera parameters was a straightforward extension of these existing teachings.

33. In short, by the time of the '276 patent, it was already routine to: (i) capture roadway images from vehicle-mounted cameras, (ii) detect horizon lines and lane boundaries, (iii) overlay those lines on images for calibration and annotation, (iv) allow human annotators to correct or input new line positions, and (v) recompute camera parameters based on those annotations, with results

transmitted locally or remotely across networked systems. The '276 patent's claims therefore reflect the combination of well-known techniques in computer vision, annotation workflows, and vehicle calibration rather than any technical innovation.

E. Transmission of Annotated or Overlaid Images

34. Before the '276 patent, it was already conventional to generate and transmit annotated images or video streams in autonomous vehicle (AV) and robotics systems. Benchmark datasets such as KITTI and Cityscapes had long relied on annotated roadway imagery with bounding boxes, lane polylines, and segmentation masks, while AV middleware such as Autoware transmitted both raw frames and overlay topics for downstream visualization. *See* EX1018, 1231; EX1019, 1; EX1020, 289-90.

35. Industry teleoperation systems demonstrated the same concept. For example, Nissan's Seamless Autonomous Mobility ("SAM") system relied on remote operators viewing the AV's camera stream with rendered overlays to resolve ambiguous situations. *See generally* EX1036.

36. Academic work likewise presented remote operator GUIs with rasterized overlays superimposed on vehicle video feeds. *See generally* EX1022 (showing an overlaid "predictive corridor" that augments the teleoperator's video to compensate for latency and improve performance).

37. Patents also confirm the practice. *See, e.g.*, U.S. Pat. No. 5,652,849 (overlying control icons into live video for teleoperation); U.S. Pat. No. 9,775,682 (displaying visual indicators/overlays and diagnostic images during teleoperated procedures); U.S. Pat. No. 8,769,396 (calibrating video and displaying annotations tied to scene features to aid analysis).

F. Remote Calibration and GUI-Based Processing

38. Remote calibration of AV sensors was a known practice before the '276 patent. The Autoware open-source AV stack, for example, separated raw camera image streams and detection overlays into different ROS topics, enabling remote GUI nodes to recombine them and accept user edits. *See* EX1020, 289-90.

39. AV companies disclosed server-side calibration architectures where vehicles transmitted imagery and logs to a central server for recalibration or adjustment. *See* EX1025, 6:15-35 (describing an AV system architecture in which teleoperator systems and an AV management server exchanged logs, images, and calibration data).

40. Such remote calibration GUIs permitted operators to adjust horizon lines, lane boundaries, or other features remotely, recompute calibration parameters, and push updated values back to the vehicles.

G. Verification and Consistency Across Annotators

41. Verification and consistency were critical concerns in AV perception workflows. A consistent principle was that remote operators must observe real-world situations to make adjustments. Waymo's 2021 Safety Report described remote fleet response teams who reviewed and validated the AV's perception outputs to ensure consistent behavior. *See* EX1026, 20-23; 27.

42. Visualization tools and datasets ensured that overlays remained consistently aligned with the base image, thereby avoiding recomputation or divergence. For example, the Autoware middleware published raw sensor frames and overlay topics separately so they could be recombined with consistent registration in remote GUIs. *See* EX1020, 289-90. Likewise, benchmark datasets such as Cityscapes and ApolloScape required that annotated overlays (e.g., lane boundaries, segmentation masks) be pixel-aligned to the original camera frames to ensure consistent annotation quality. *See* EX1019, 1; EX1027, 1067-1073.

43. Dataset development practices reinforced the same principle: annotated frames were curated to ensure that multiple human labelers worked from the exact same imagery. *See* EX1018, 1231.

44. Patents also reflected this concern. *See, e.g.,* EX1028, 8:10-30 (teaching remote validation of AV perception outputs using annotated images to ensure operator and system consistency).

H. Human-in-the-Loop Calibration by Remote Experts

45. Human-in-the-loop (HITL) annotation and calibration were standard practices in AV development. HITL annotation involves the presentation of model outputs, intermediate feature maps, or sensor data to a human reviewer for the purpose of verifying, labeling, or correcting those outputs. The human-provided feedback or labels are then incorporated into the system's training or validation dataset, enabling supervised or semi-supervised refinement of the model parameters. HITL calibration refers to the use of human input to verify or adjust system parameters or alignment values, such as detection thresholds, transformation matrices, or sensor poses when automated calibration algorithms are incomplete or ambiguous. Through this process, a human operator may provide reference points, confirm predicted alignments, or supply corrective adjustments that directly inform the system's calibration state. For example, leading datasets (KITTI, Cityscapes, ApolloScape) relied on human annotators to correct roadway imagery and provide ground-truth data. *See* EX1018, 1231, *supra*; EX1019, 1; EX1027, 1067-1073.

46. Academic and legal literature acknowledged remote human oversight in AVs. *See generally, e.g.*, EX1029 (describing human-in-the-loop teleoperation and supervisory control in automotive systems).

47. Patents further documented the integration of remote operators into calibration and oversight loops. *See, e.g.*, EX1030, 12:5-28 (describing human-assisted correction of perception outputs via a remote interface).

48. Taken together, these references show that expanding human-in-the-loop calibration to include remote experts was a straightforward extension of established practice.

I. Bandwidth-Efficient Transmission of Overlay Data

49. AV engineers routinely balanced whether to transmit composite images, raw frames, or lightweight overlay metadata. Overlays typically imposed negligible bandwidth compared to the underlying video. *See* EX1031, 1700-1701.

50. Studies of teleoperation UIs emphasized that overlay graphics (e.g., lines, boxes) are “lightweight” relative to video, and therefore suitable for efficient transmission. *See* EX1022, *supra*.

51. Patent disclosures confirm the same. *See, e.g.*, EX1031, 9:1-25 (teaching that transmitting overlay parameters rather than full video reduces network load while preserving operator context).

52. These teachings demonstrate that there were no concerns about bandwidth when transmitting overlaid images: either rasterized overlays or metadata overlays were already routine design options.

53. In sum, by the time of the '276 patent, a POSITA would have readily understood: (i) transmitting annotated images was routine; (ii) remote calibration GUIs were widely disclosed; (iii) ensuring consistency across annotators required transmitting overlaid frames; (iv) human-in-the-loop calibration commonly included remote experts; and (v) overlay transmission was bandwidth-efficient.

J. Prior Art Relied on for the Grounds of Unpatentability

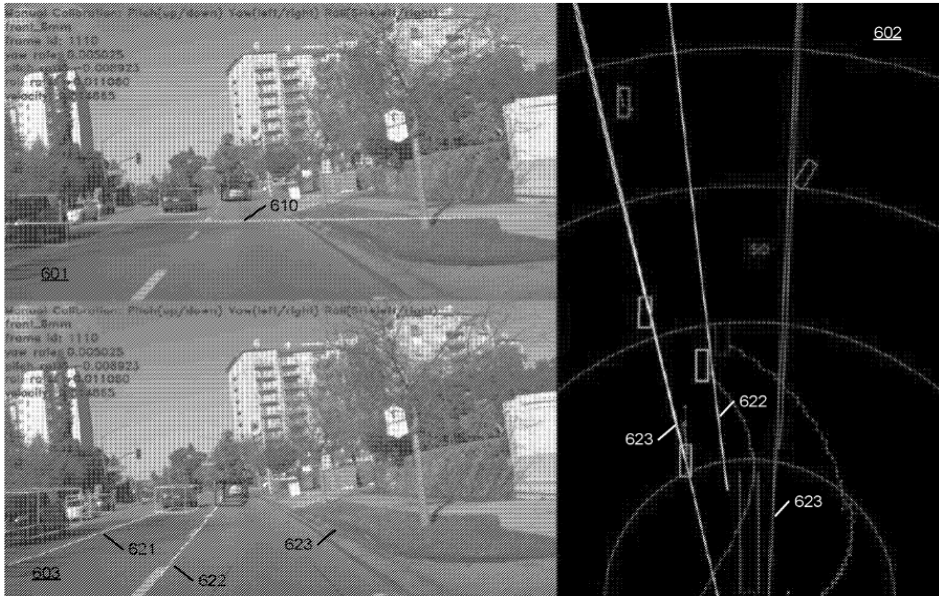
54. The following prior art used in the Grounds of unpatentability show that detecting a horizon line in roadway images, overlaying and modifying line annotations through an interface, and recomputing camera parameters based on those modified lines were all well known before the '276 patent.

1. Choe discloses a system that calibrates a camera for lane detection using a predicted horizon line and annotations.

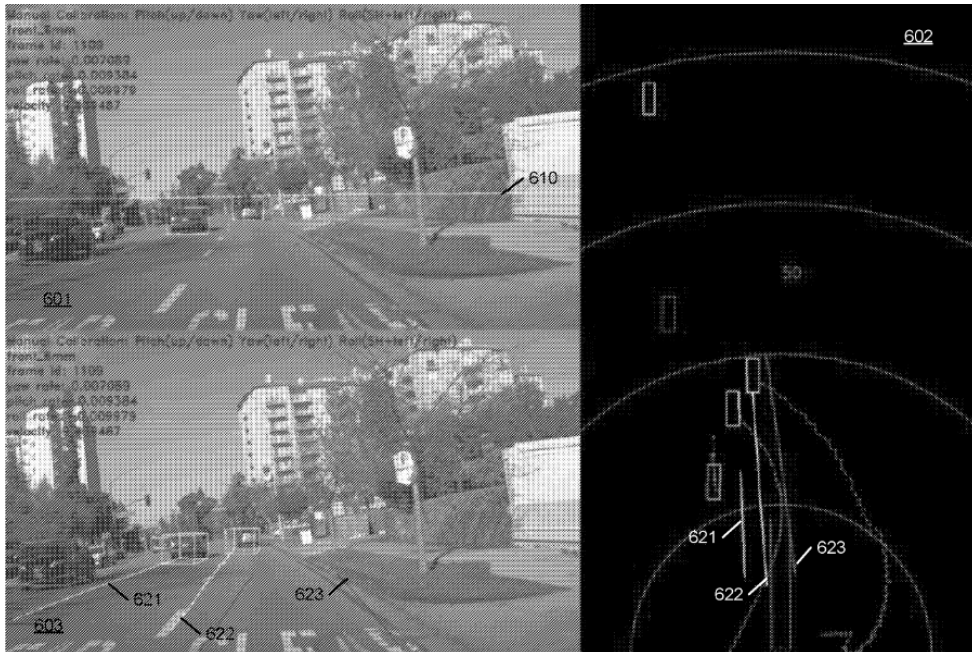
55. Choe is directed towards “calibrating a sensor system of an autonomous driving vehicle.” EX1005, ¶[0001]. As Choe explains, sensors in autonomous vehicles need to be periodically calibrated to ensure that they accurately detect and recognize objects and lane configurations of the driving environment. EX1005, ¶[0003]. At the time of Choe’s invention, however, there was “a lack of efficient sensor calibration system[s].” *Id.* One part of Choe’s sensor calibration system involves calibrating a camera for lane detection.

56. To calibrate its camera, Choe describes processing logic that receives images from the camera over a network. *See* EX1005, ¶[0018]. The camera is

described as “still cameras and/or video cameras.” EX1005, ¶[0022]. When using a video camera, the “images” are video frames, as seen in Choe’s Figures 6A and 6B:



EX1005, FIG. 6A.



EX1005, FIG. 6B.

57. After receiving the video frames, Choe predicts the horizon line and lane lines. EX1005, ¶¶[0015]-[0016]. “[A] horizon line is determined based on the camera’s hardware settings.” EX1005, ¶[0015]. Additionally, “[o]ne or more lane lines are determined based on the first image via a perception process performed on the first image.” EX1005, ¶[0015]; *see also* EX1005, ¶¶[0056], [0059], FIG. 7. The prediction of horizon and lane lines is done by a “machine learning engine 122 [that] generates or trains a set of rules, algorithms, and/or predictive models 124.” EX1005, ¶[0030].

58. Choe further describes superimposing lines for the horizon line and lane lines on the frames of the video. EX1005, ¶¶[0016], [0050]. After determining the horizon line within a video frame, the horizon line is superimposed on the frame. EX1005, ¶[0050]. Then, “a second image is rendered” and “includes the lane lines superimposed thereon.” *Id.*



EX1006, FIG. 6A (excerpt, annotated); *see also* EX1005, FIG. 6A.

59. After superimposing the lines, Choe sends the video to a user to review the predicted lines. *See* EX1005, ¶¶[0014]-[0016]. Choe describes having a “visualizer” which “includes a user interface to visually display a two-dimensional (2D) image captured by a camera.” EX1005, ¶[0014]. On this visualizer, a user views the lines and manually lines them up with the real lane lines and horizon line. EX1005, ¶¶[0055]-[0057]. A user may do this manual correction using an “input device (e.g., keyboard, joystick, voice interactive command).” EX1005, ¶[0015]. “In response to a first input signal (e.g., an up or down arrow key) received from an input device (e.g., keyboard, joystick, voice interactive command), a position of the horizon line is updated based on the first input signal and a position of at least one of the lane lines is updated based on the updated

horizon line.” EX1005, ¶[0015]. The user manipulates the lines until they are satisfied with the results. See EX1005, ¶[0014].

60. Based on determining the lines, Choe then predicts camera parameters such as the pitch angle of the camera. EX1005, ¶[0015]. For example, “[a] first calibration factor or first correction value is determined for calibrating a pitch angle of the camera based on a difference between the initial horizon line and the updated horizon line.” EX1005, ¶[0015]. Choe also specifies that the system uses the visualizer to manually calibrate other camera parameters, including the camera’s “yaw angle, and/or a roll angle.” EX1005, ¶[0014].

61. Once the user is satisfied with the line placement, Choe transmits updated camera parameters back to the vehicle’s camera. *See* EX1005, ¶¶[0031]-[0044]. After a user has moved the horizon line, “[a] first calibration factor or first correction value is determined for calibrating a pitch angle of the camera based on a difference between the initial horizon line and the updated horizon line.” EX1005, ¶[0015]. “Similarly, the yaw angle and the roll angle of a camera can also be calibrated using the techniques as described above.” EX1005, ¶[0057].

2. Davies discloses a system that calibrates a vision-based sensor of a vehicle by transmitting roadway images to an external computing device, processing the images remotely, and then transmitting modified calibration data back to the vehicle.

62. Davies discloses a system that calibrates a vision-based sensor of a vehicle by transmitting roadway images to an external computing device, processing the images remotely, and then transmitting modified calibration data back to the vehicle.

63. Davies situates the work squarely in the field of vehicle-mounted imaging: “The present invention is generally related to vehicle mounted cameras. More particularly, example embodiments of the present invention are related systems and methods for adjusting an image, e.g., an image horizon, for a vehicle mounted camera.” EX1007, ¶[0002]. The specification identifies the prior-art shortcoming, fixed horizons that visually tilt as track angle changes, and the resulting need for correction: prior art “shows a change in horizon relative to the sky... due to a change in angle of the track” EX1007, FIGs. 1-2, and “[w]hat is needed in the art is a system and method that permits adjustment of an image from a vehicle mounted camera in a desired fashion,” EX1007, ¶¶[0004]-[0005].

64. Davies then discloses a telemetry-driven solution that automatically adjusts the displayed image based on vehicle sensors. The core teaching is “an adjustable image that adjusts in response to at least one vehicle mounted sensor,” EX1007, ¶[0006], where “telemetry of a vehicle from a plurality of sensors may be

used to automatically adjust an image, e.g. an image horizon, in a desired way,” EX1007, ¶[0007]. In embodiments, the system “automatically adjust[s] an image horizon to match a skyline horizon during tilting of a vehicle,” EX1007, ¶[0008]) and “both image horizon and zoom are automatically adjusted during tilting of a vehicle,” EX1007, ¶[0009]. Importantly, Davies implements this as post-capture processing: “such image horizon adjustment may be provided as a digital video effect, alleviating the need to actually adjust the angle of a camera during vehicle tilt.” EX1007, ¶[0010].

65. Davies details the breadth of inputs that can drive these adjustments. Sensor data is not limited to tilt; rather, “Sensor data may include any convenient type of data, including gyro data, vehicle angle, attitude, altitude, speed, acceleration, traction... navigational data,” and also “environmental conditions for the vehicle, such as weather, sensed track conditions, wind... [and] temperature.” EX1007, ¶[0031]. Correspondingly, the image operations extend beyond leveling the horizon to include “adjustment of image crop; selection of image portions; tracking of objects of interest in images; rendering selective high definition images from greater than high definition cameras; [and] selective capture of image points of interest, [and] adjustment of the image responsive to environmental conditions.” EX1007, ¶[0032].

66. Davies also teaches a high-resolution capture workflow with selective sub-frame output and a GUI “extraction window.” In particular, “a first image or video is captured at a first resolution, which resolution is greater than high definition... [and] a desired portion of the first image or video is then displayed at a second, lower resolution” EX1007, ¶[0038]. A concrete example shows a “full-raster 4K moving video image” with a “720p... selectable extraction window,” and even “simultaneous multiple extraction windows.” EX1007, ¶¶[0039]-[0040]. The GUI “allow[s] an operator to navigate within a captured image and select portions... adjust the size and position of the extraction window... [and] track or scan across moving images.” EX1007, ¶[0041]. Processing may be offsite or onsite with standard broadcast routing: it “may occur either offsite (FIG. 11) or onsite (FIG. 12),” with outputs provided “to a router ... switcher or ... server” for live or recorded use. EX1007, ¶¶[0042], [0045], [0046].

67. Davies sets out a straightforward flow (FIG. 3) that mirrors the core concept: “receiving image data from a vehicle mounted camera ... receiving data from at least one vehicle mounted sensor ... and adjusting the image horizon utilizing the data received from the at least one vehicle mounted sensor.” EX1007, ¶[0057]. Davies confirms again that the adjustment “may be applied as a digital video effect,” and notes it can operate “whether or not such adjustment results in matching image horizon with a skyline horizon.” EX1007, ¶[0058]. To address

bandwidth and deployment constraints, Davies contemplates “an on-board (on the vehicle) processor” performing “some or all of the image adjustment... particularly useful... in wireless transmission applications,” with operator communication on a “separate channel.” EX1007, ¶[0059]. It further underscores “automatic adjustment of image horizon based upon received vehicle telemetry data,” EX1007, ¶[0060], and illustrates skyline matching during tilt, EX1007, FIGs. 4-5, and tilt-plus-zoom, EX1007, FIGs. 6–7. EX1007, ¶¶[0061]-[0062].

68. Finally, Davies provides a networked system architecture that integrates telemetry, imaging, and processing. “The system 100 may include a server 101” hosting “vehicle telemetry information, static and continuous video images... algorithms and processing modules,” connected over a “network 106” and optionally to “third party data sources or servers 103.” EX1007, ¶¶[0063]-[0064]. Multiple client devices can operate online or offline using cached data, EX1007, ¶[0069], and, in sum, “any suitable device may be utilized to use vehicle telemetry data from at least one vehicle sensor to adjust image horizon from a vehicle mounted camera.” EX1007, ¶[0070].

69. Claim 24 of Davies confirms this functionality, expressly reciting “wherein said adjusted image is transmitted via wireless protocol to an external computing device.” Thus, Davies provides explicit support for remote calibration and parameter feedback.

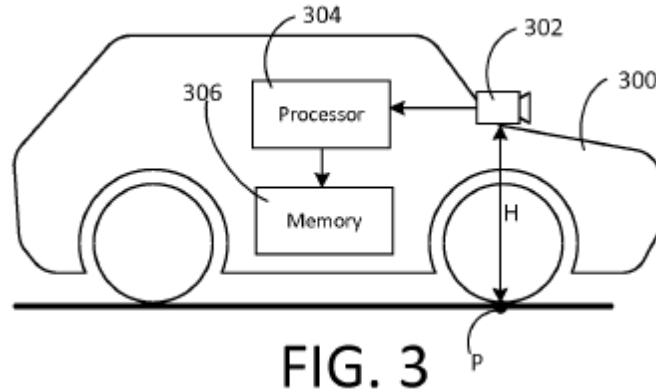
70. In sum, Davies discloses a calibration workflow in which vehicle-mounted cameras capture roadway images, transmit those images to a remote computing device, receive modified calibration data in return, and then update internal camera parameters accordingly. This two-way feedback loop, facilitated by wireless communication, enables scalable and accurate calibration of vehicle vision systems.

3. Westmacot discloses a system for annotating roadway images captured by vehicle-mounted cameras, generating 3D road models, overlaying annotations such as lane boundaries, and computing camera parameters based on the detected or user-modified lines.

71. Westmacot is directed to “methods of annotating road images” and, more specifically, to using vehicle-mounted cameras to capture image sequences and generate road/lane annotations that can be used to train or calibrate vision systems. EX1008, pp. 1-3, 7. It explains that training autonomous driving systems requires large datasets of annotated images (road surfaces, lane lines, and object boundaries) captured from the perspective of a traveling vehicle. *Id.*

72. Westmacot describes that “a time sequence of two dimensional images [is] captured by an image capture device of a travelling vehicle.” EX1008, p. 3. The system uses these frames, recorded by a dashcam or similar in-vehicle camera, to reconstruct a 3D road model of the environment. *See* EX1008, FIG. 3 (vehicle 300 with camera 302, processor 304, and memory 306). These captured

images form the basis for road modeling, lane detection, and annotation workflows.



EX1008, FIG. 3 (Vehicle & Camera System).

73. Figure 3 depicts a training vehicle 300 equipped with an “image capture device 302” mounted on the windshield, feeding images into processor 304 and memory 306. This figure confirms that the system begins with roadway images recorded by a camera installed within the vehicle. EX1008, FIG. 3, pp. 1, 3, 7.

74. After receiving images, Westmacot determines roadway features including the horizon line and lane boundaries. The disclosure explains that a horizon line can be calculated by estimating the road surface normal from the 3D vehicle path and intersecting it with the image plane, yielding a line across the image corresponding to the horizon. EX1008, Fig. 16; Camera Angular Offset discussion. This horizon line is expressly identified as such in the text.

Westmacott, pp. 30

75. Westmacot also discloses overlaying lane lines and annotations onto captured images for display to an annotator. For example, “annotation data may be displayed with the image (e.g., overlaid on the image).” EX1008, Fig. 6; Fig. 25. An annotation interface (UI 412) allows a user to view projected lane lines and adjust them (widen, narrow, move), with changes propagating across the batch of frames via the model adaptation component 410. See EX1008, Figs. 6, 18–24.

76. Figure 25 shows a roadway image overlaid with annotations of road surface, ego-lane, and lane instances. This exemplifies how the system generates overlaid images for human review and correction. EX1008, Fig. 25; annotation interface UI 412; rendering 416.

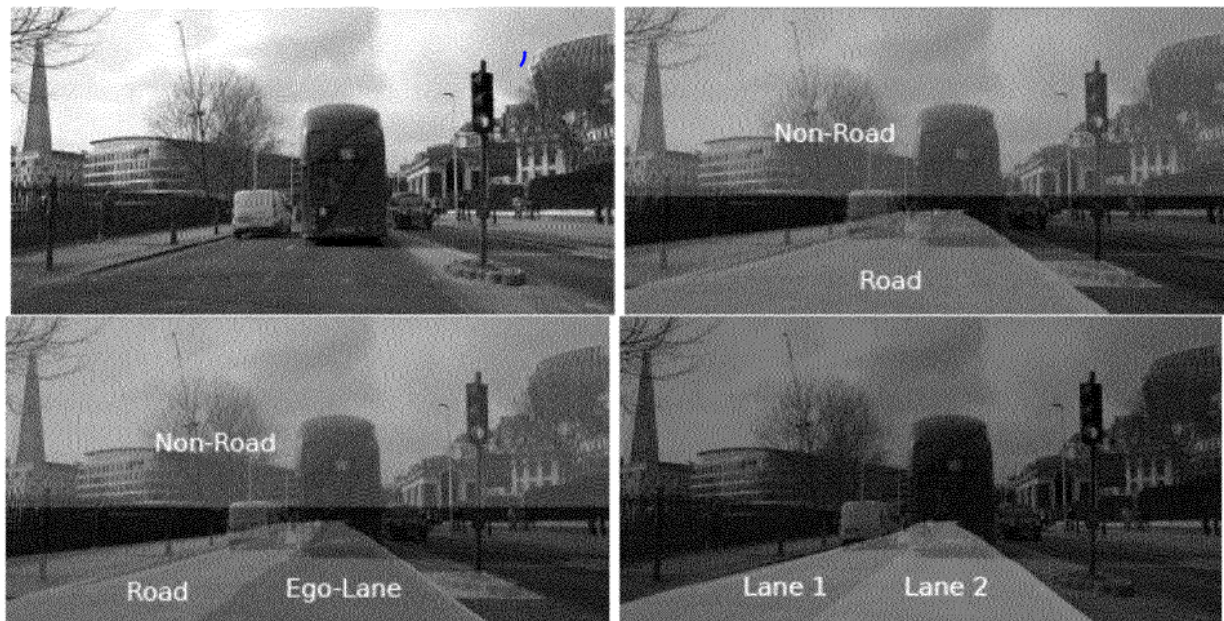


Fig. 25

EX1008, Fig. 25 (Annotated Road Image)

77. Based on detected or modified lines, Westmacot computes camera parameters through its parameter computation component 408. These parameters include camera height (H), horizontal offset (S), forward point, horizon line, lane width (W), and center position (C). EX1008, p. 32-33. The disclosure explains that these parameters are recomputed whenever the lane lines are adjusted, ensuring alignment between the 3D road model and the actual captured imagery.

78. Figure 15 illustrates how the angular offset of the camera relative to the vehicle's longitudinal axis is derived. Westmacot explains that this process defines the camera's forward point and horizon line in the image. EX1008, p. 40-41, Fig. 15.

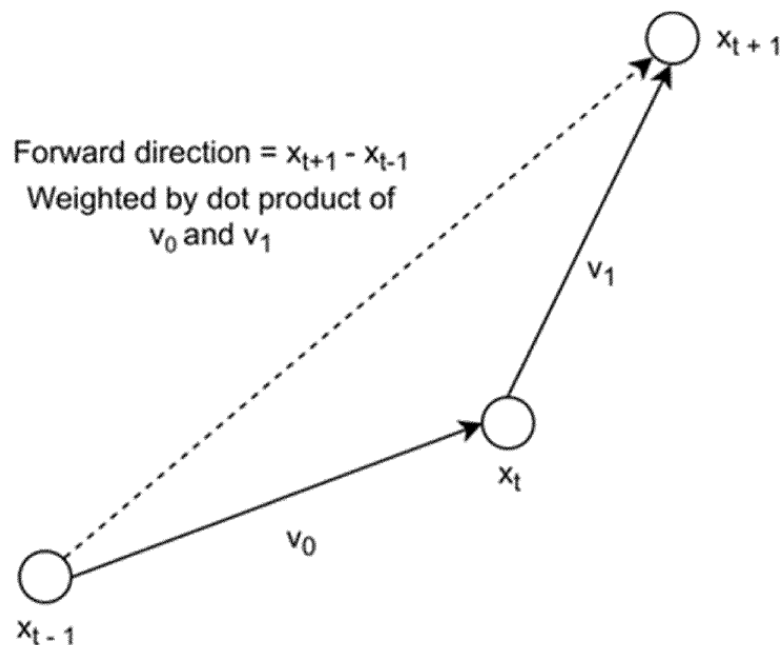


Fig. 15

EX1008, Fig. 15 (Camera Orientation / Horizon Line)

79. Westmacot further emphasizes that these functions are implemented in software executed on processors, and that annotation data (including overlays) is stored electronically in storage 414 for later retrieval. EX1008, pp. 33-34. This makes clear that annotated/overlaid images and computed camera parameters exist as digital data objects suitable for further use in calibration pipelines or machine learning training.

80. In sum, Westmacot discloses a vision-based calibration and annotation system that:

- receives roadway images from a vehicle-mounted camera,
- detects geometric features such as the horizon line and lane lines,
- overlays those lines onto the images for human review and correction,
- allows a user to modify the line positions through an annotation interface, and
- computes camera parameters (height, offset, horizon line, lane width, forward point, etc.) based on the detected or modified lines.

4. Tal discloses a system for capturing roadway images with vehicle-mounted devices, overlaying detected object data onto the images, transmitting the resulting overlaid images and associated data to remote servers over wireless or cellular networks, and providing access to the transmitted roadway and incident data through a web-based application.

81. Tal discloses a networked roadway imaging and incident-detection platform in which a vehicle-mounted device captures images, optionally performs local vision processing to detect/annotate objects, and transmits raw and/or processed (overlaid) imagery to remote servers over wireless/cellular networks for storage, analysis, and user access via web applications. EX1009, ¶¶[0012]-[0013], [0027], [0030], [0052], [0075], [0138], FIG. 12.

82. **Capture & Local Processing.** Tal’s “device 101” (a smartphone, embedded computer, or smart camera with one or more cameras) captures roadway images 16 and can locally run neural networks to infer objects 12 and produce “resultant processed data.” EX1009, ¶¶[0027] (“Digital images 16 are taken by one or more imaging devices 101 ... preferably with network connection capabilities ...”), [0030] (device “executes software ... [and] infers ... any objects 12 ... using the neural network(s) 905”). Tal expressly defines “data 19” as the processed/overlaid image: “data 19 (which is image 16 with object data 20 superimposed or overlaid thereon).” EX1009, ¶[0052].

83. **Overlaid Imagery (Annotated Frames).** Tal clarifies that processed imagery includes overlays of detected object data on top of the original frame, and

that such overlays may be flattened/merged into the image itself. EX1009, ¶[0052] (“[D]ata 19,20 ... can be overlaid on the image ... and flattened/merged into the image 17 ...”). Thus Tal expressly teaches generation of an “overlaid image.”

84. **Network Transmission to Remote Server(s).** Tal repeatedly teaches transmitting both raw images and processed/overlaid imagery to a remote server 107 via wireless/cellular networks. EX1009, ¶[0027] (“Only selected data portions 20 ... can be transmitted ... to a server 107 ... Alternatively, unprocessed image(s) 16 can be transmitted ... It is also recognized that both resultant processed image data 20 and unprocessed images 16 can be transmitted ...”), [0030] (device transmits data 20 “to the server 107 (via the network 18 utilizing the cellular connection 106 or wireless connection 105)”), [0075] (“processed image data 20 ... transmitted to the server 107 over the network 18”). Tal’s incident flow further illustrates sending images to the server as part of typical reporting. EX1009, ¶[0138] (“The Device 101 then transmits data 1202 to the Server(s) 107 ... including ... an Image: a picture of the incident ...”).

85. **Server as Gateway & Web Access (Web App UI).** Tal’s servers process/store incoming data and serve users via web access: “server(s) ... serve as gateway to users via web access and present data to users in a meaningful and intuitive manner,” and “client(s) 1208 ... access a user interface 1209, which may be either a web application that can be accessed using a web browser or a

client/server application.” EX1009, Summary, Fig. 12 discussion. This establishes a browser-based front end for viewing transmitted roadway imagery, including overlaid/annotated frames, and related incident/calibration data.

86. **User Interfaces on the Device.** Tal also provides a local UI (119) on the device to display the camera view, orientation, status, settings, and parameters, confirming that users can interact with roadway imagery and calibration-related information on a computing device. EX1009, UI 119 (device user interface).

87. **Two-Way/Distributed Workflows (Scalability).** Through the combination of on-device processing, network transmission, and server-hosted web access, Tal enables distributed data collection, centralized analysis/storage, and broad user access well-suited to scalable fleet or municipal deployments. EX1009, ¶¶[0012]-[0013], [0027], [0030], [0075], FIG. 12.

5. Kuehnle discloses a system that calibrates a camera based on time and the speed of the vehicle.

88. Kuehnle discloses a system “for online calibration of a video system ... in connection with an image-based road characterization ... for detecting roadway scenes in vehicles.” EX1010, p. 1. As part of this system, Kuehnle describes how, to measure the position of objects observed by the video camera, the system first establishes the orientation of the video camera. EX1010, p. 1. To determine the orientation of the video camera, Kuehnle detects vanishing points in the captured video to deduce the pitch and yaw of the camera. EX1010, p. 2. The

vanishing points are determined by looking at the “long term average vanishing point” that is “calculated with time-filtering methods.” EX1010, p.3.

89. Kuehnle’s time-filtering methods include looking at the speed of the vehicle and time. *See* EX1010, p. 11-12; p. 13. Kuehnle states that the conditions for extrapolation “require that the vehicle is moving with at least a certain speed, so that low speed maneuvering, with its possible large yaw angle, is not taking place.” *See* EX1010, p. 11-12.

90. Further, Kuehnle discloses how the system filters across multiple frames to meet a temporal requirement. Kuehnle discusses that “a long term average vanishing point location is calculated with time-filtering methods from a sequence of images,” and that “the static yaw and pitch angle of the camera is deduced” from this averaged data. *See* EX1010, p. 3. In one embodiment, Kuehnle applies a recursive averaging filter in which a “time-varying weighting value [] decreases with time (as in a recursive averaging filter),” such that each new frame contributes proportionally less to the overall calibration result. *See* EX1010, p. 13.

91. Therefore, Kuehnle discloses a calibration system for detecting roadway scenes in vehicles in part based on time and the speed of the vehicle.

V. PROSECUTION HISTORY

92. During the prosecution history of the '276 patent, there was only a single Office Action and response before allowance with no amendments.

93. In the single Office Action, the Examiner first rejected claims 21–22 and 28 on grounds of nonstatutory double patenting as being unpatentable over claims 1, 5, and 8 of its parent patent, the '580 patent. The Examiner explained that “although the claims at issue are not identical, they are not patentably distinct from each other.” FH, p. 119.

94. The Examiner also rejected claims 21–40 under § 103 over prior art. FH, p. 122. The cited prior art included: (1) U.S. Patent Application Publication No. 2019/0034740 A1 to Kwant et al. (“Kwant1”) and (2) U.S. Patent Application Publication No. 2019/0102674 A1 to Kwant et al. (“Kwant2”). FH, p. 122.

95. In response, the Applicant filed a terminal disclaimer to the '580 patent. With respect to the prior art rejections, the Applicant argued that Kwant1 did not disclose “any type of visualization of the horizon line.” FH, p. 137. The Applicant asserted that “instead, Kwant1 at most describes using a horizon line as a tool for other operations that do not include visualizing the horizon.” *Id.* The Applicant further alleged that “Kwant1 describes ‘annotating’ images. However, these annotations do not involve ‘overlaying’ the horizon as claimed. Instead, they

refer to annotating other objects (e.g., signs) by using the horizon line as a point of reference.” *Id.*

96. Applicant next argued that “[c]laim 21 recites ‘receiving a modification of the line from the computing device, the modification comprising a new line at a second position’” and that “[p]utting aside that Kwant2 describes a model training process (not a verification), nothing in Kwant2 describes computing a ‘new line at a second position.’” FH, p. 138. The Applicant explained that “[b]y contrast, Kwant2 describes predicting a ‘label’ and then comparing it to a known label” and that “[e]ven if the labels in Kwant2 were horizontal lines, there is no evidence that Kwant2 receives a horizontal line, predicts a new horizontal line, and then compares the new horizontal line to a third horizontal line.” FH, p. 139. The Applicant added: “[i]ndeed, at best, one would assume that Kwant2 receives an image with no horizon, compute the horizon, then compare the computed horizon to an expected horizon. In this example, there is no ‘modification’ of a line.” *Id.* Finally, the Applicant argued that “Even if Kwant2 were construed to describe a generative model (a point not conceded), Kwant2 fails to describe ‘computing a camera parameter based on the new line.’” *Id.* The Applicant emphasized that the “parameters tuned in Kwant2 are not camera parameters, they are internal weights and biases of a neural network. Simply stated, the parameters tuned in Kwant2

have no relation to a camera, but are related instead to how the machine learning system processes feature vectors.” *Id.*

97. The Examiner then allowed all claims.

VI. PERSON OF ORDINARY SKILL IN THE ART

98. A person of ordinary skill in the art (“POSITA”) in the technical field of the ’276 patent at the relevant time would typically have had a Bachelor’s in Computer Science with a focus on computer/machine vision, and at least two years of experience with each of: (1) machine learning (including neural network) methods as applied to machine/computer vision and (2) classical machine/computer vision algorithms (e.g., edge detection, line identification, computation of camera parameters from horizon line, etc.). Additional education could serve as a substitute for the experience requirement.

VII. GROUNDS OF UNPATENTABILITY

99. The four Grounds of unpatentability in this Declaration are shown below. This Declaration explains why each Ground renders obvious the claims.

Ground	Reference(s)	Basis	Claims
1	Choe	§103	1-3, 5-10, 12-16, 18-20
2	Choe and Davies	§103	1-3, 5-10, 12-16, 18-20
3A/3B	Choe and Kuhnle (3A); Choe, Davies and Kuhnle (3B)	§103	4, 11, 17
3	Westmacot and Tal	§103	1-20

VIII. GROUNDS 1: CHOE RENDERS OBVIOUS CLAIMS 1-3, 5-10, 12-16, 18-20.

100. Choe discloses or renders obvious each limitation of independent claims 1, 8, and 15, which recite the method, computer-readable medium, and device aspects of the same calibration workflow. Specifically, Choe teaches receiving an image from an in-vehicle camera ([0015], [0048]), detecting and superimposing a horizon line and lane lines ([0015], [0046], [0048]–[0051]), enabling a user to modify the line via input devices ([0015], [0049]–[0053]), and computing calibration parameters such as pitch, yaw, and roll from the updated line positions ([0015], [0049]–[0054], [0059]–[0060]). Those calibration parameters are then transmitted back for use in the vehicle’s perception software ([0031], [0044]).

101. To the extent Patent Owner argues that Choe does not expressly disclose transmitting the “overlaid image ... over a network,” Choe’s client-server architecture with remote calibration and GUI rendering ([0018], [0025], [0031], [0046]–[0047]) makes network transmission of the overlaid image at least obvious as a predictable deployment choice.

102. The challenged dependent claims in this Ground are also disclosed or, at a minimum, rendered obvious by Choe. For example, Choe’s computation of pitch, yaw, and roll parameters corresponds to the claimed “viewing angle,” and a POSITA would have readily derived road-plane normals or camera height from the

same geometry ([0015], [0049]–[0054], Figs. 5A–5C). Choe further teaches online calibration while the vehicle is moving ([0017], [0054]), and applying thresholds based on vehicle speed or duration would have been an obvious design choice using available CAN/GPS inputs ([0021]–[0024]). Likewise, Choe’s server-hosted calibration system readily encompasses a web-based application ([0018], [0025], [0031]), and the disclosed GUI shows a user identifying and updating a horizon line ([0015], [0049]–[0051], [0055]–[0057]). Finally, Choe expressly recomputes calibration parameters based on line modifications ([0015], [0049]–[0053], [0059]–[0060]).

A. Independent Claim 1

103. Choe renders obvious claim 1.

1. [1Pre] “A method comprising”

104. Choe discloses this element, as shown in the following excerpt:

The processes *or methods* depicted in the preceding figures may be performed by processing logic that comprises hardware (e.g. circuitry, dedicated logic, etc.), software (e.g., embodied on a non-transitory computer readable medium), or a combination of both.

EX1005, ¶[0064].

105. Choe’s specification and figures generally also describe its method.

2. [1A] “receiving an image of a roadway recorded by a camera device installed within a vehicle”

106. Choe discloses this element.

107. In Choe, an autonomous driving vehicle includes cameras 211 as part of its sensor system. Choe explains:

“Referring now to FIG. 2, in one embodiment, sensor system 115 includes, but it is not limited to, one or more cameras 211 ... Cameras 211 may include one or more devices to capture images of the environment surrounding the autonomous vehicle. Cameras 211 may be still cameras and/or video cameras.”

EX1005, ¶[0022], FIG. 2.

108. These cameras capture images of the roadway/environment surrounding the ADV, which are then used by the perception and calibration processes.

109. Choe repeatedly describes receiving a first image captured by a camera of the ADV and using it for calibration. For example, Choe states:

“According to one embodiment, in response to a first image captured by a camera of an ADV, a horizon line is determined based on the camera’s hardware settings ... One or more lane lines are determined based on the first image via a perception process performed on the first image.”

EX1005, ¶[0015].

110. Similarly, Choe discloses that perception module 302 receives sensor data from camera 211, “sensor data may include an image ... The sensor data is then processed by perception module 302 such as image processing, including detecting an object within the image and determining a horizon line representing a vanishing point of a road.” EX1005, ¶[0045].

111. Choe again confirms that the system processes “a first image captured by a camera of an ADV,” from which the perception module determines a horizon line and lane lines. EX1005, ¶[0048]. Claim 1 of Choe itself also recites determining a horizon line from “a first image representing a two-dimensional (2D) view from the viewpoint of the ADV.” Thus, Choe teaches the step of “receiving an image of a roadway recorded by a camera device installed within a vehicle” as recited in claim 1 of the ’276 patent.

3. [1B] “detecting a horizon line in the image”

112. Choe discloses this element in two ways.

113. First, Choe explicitly teaches that, in response to an image captured by a camera of the ADV, the system determines a horizon line in the image. For example, Choe explains:

“According to one embodiment, in response to a first image captured by a camera of an ADV, a horizon line is determined based on the camera’s hardware settings. The horizon line represents a vanishing

point from a view point of the ADV based on an initial or default pitch angle of the camera.”

EX1005, ¶[0015]; *see also* EX1005, ¶[0048].

114. Similarly, Choe’s perception module processes images from camera 211 and expressly performs “image processing, including detecting an object within the image and determining a horizon line representing a vanishing point of a road.” EX1005, ¶[0045].

115. Choe further illustrates this detection in its GUI. In FIG. 6A, an image 601 captured by the camera includes horizon line 610, which Choe explains is “determined and placed on a location within image 601 based on the hardware setting of the targeted camera.” EX1005, ¶[0055].



EX1006, FIG. 6A (excerpt, annotated); see also EX1005, FIG. 6A.

116. Choe's process 700 (Fig. 7) confirms the same, reciting at block 701 that "processing logic determines a horizon line representing a vanishing point from a view point of the ADV based on a pitch angle of a camera that captured a first image." EX1005, ¶[0059].

700

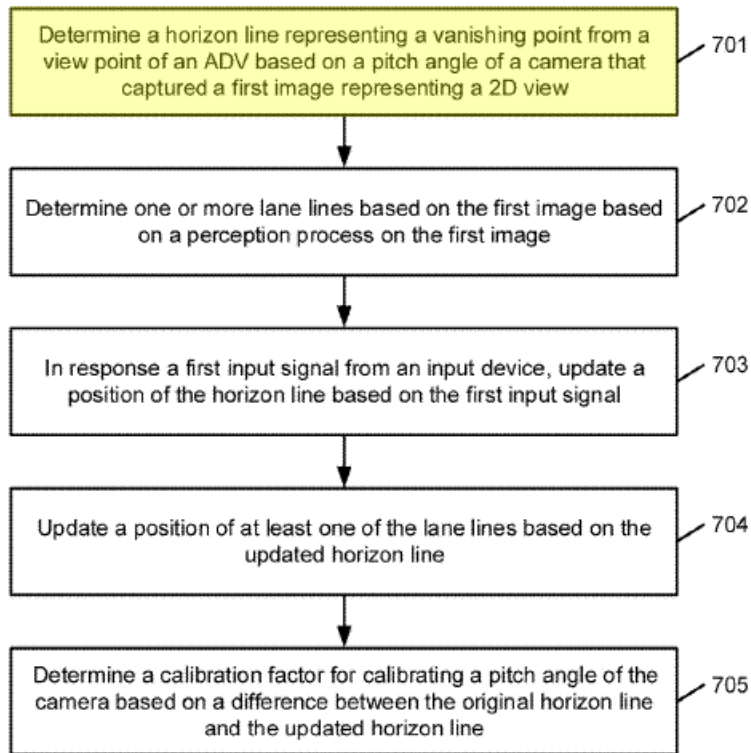


FIG. 7

117. Second, Choe teaches detecting a horizon line using a predictive model. Choe states that it uses machine learning engine 122 and predictive models 124 for a variety of purposes. EX1005, ¶[0030]. One of these purposes is camera calibration. EX1005, ¶[0030]; *see also* EX1005, FIG. 1 (showing algorithms/models 124 with the sensor calibration system 125).

Based on driving statistics 123, machine learning engine 122 generates or trains a set of rules, algorithms, and/or *predictive models 124* for a variety of purposes. In one embodiment, algorithms 124 may include algorithms for autonomous driving a vehicle as described further

below. *Algorithms 124 may further include sensor calibration algorithms*, which will be described further below. Algorithms 124 can then be uploaded on ADVs to be utilized during autonomous driving in real-time.

EX1005, ¶[0030].

118. In view of Choe’s express teachings of using predictive algorithms for camera calibration and its disclosure that perception module 302 determines a horizon line and lane lines (¶[0045]), a POSITA would understand that Choe’s predictive model 124 is directed to forecasting those features (horizon line, lane lines) for calibration purposes. This inference aligns with the well-established use in computer vision of horizon and lane geometry cues as inputs to calibration algorithms (e.g. line-based calibration methods using lane markings, and modern learned calibration methods that regress camera parameters from line and horizon features). *See* EX1033 (describing a method for estimating extrinsic parameters such as pitch, yaw, and roll from roadway lane boundary observations); EX1034 (proposing a transformer-based model that predicts calibration parameters from detected line and horizon cues); EX1035 (using deep networks to regress camera parameters from single images based on horizon and vanishing-point features).

119. Once the initial horizon and lane lines are identified, the user would then adjust the position of the lines. For example, “[t]he input signal may represent

an incremental adjustment for adjusting the position of the horizon line.” EX1005, ¶[0015].

120. After a predictive model 124 predicts the initial horizon and lane lines and a user manually adjusts the lines to ground truth, the lines are then re-identified and used by additional predictive models. Choe states:

The sensor data is then processed by perception module 302 such as image processing, including detecting an object within the image and determining a horizon line representing a vanishing point of a road. The result of the perception processing can be utilized by other modules such as prediction module 303 and planning module 305 for autonomous driving.

EX1005, ¶[0045].

121. This post-processing step provides additional support for [1B]. A POSITA would have therefore understood that Choe discloses identifying a horizon line using a predictive model, as well. Choe then describes how calibration module 308 computes calibration parameters such as pitch, yaw, and roll based on these detected lines (paragraphs [0046]–[0047]). Thus, the predictive models not only identify horizon and lane lines but also generate calibration factors from them.

122. Thus, Choe teaches detecting a horizon line in two ways, both satisfying element 1[B].

4. [1C] “overlaying a line on the image to generate an overlaid image”

123. In either form of horizon line detection described in Choe, direct detection by perception module 302 or predictive generation by models 124, Choe expressly discloses this element.

124. Choe expressly states that the detected horizon line (regardless of which form of detection) is superimposed on the image, which is an overlay, as would have been understood by a POSITA. For example, Choe states that the initial horizon line is superimposed on the image:

In one embodiment, the first image with *the horizon line superimposed thereon* is displayed within a first display area of a display device. When the position of the horizon line is updated, the first image is updated on the first display area to update the position of the horizon line to give a visual feedback to a user.

EX1005, ¶[0016]; *see also* EX1005, ¶[0015]. Further, when the position of the horizon line is updated, the image itself is updated to reflect the new overlay and provide visual feedback to the user. *See* EX1005, ¶¶[0046], [0048], [0055]-[0059], FIGs. 6A–6B.

125. Shown below, the horizon line is superimposed on the image with “frame id: 1110.” EX1005, FIG. 6A; EX1006, FIG. 6A. Shown below, this line is “too low” and will need to be adjusted by manual calibration. *See* EX1005, ¶[0057].



EX1006, FIG. 6A (excerpt, annotated); *see also* EX1005, FIG. 6A.

126. Accordingly, whether the horizon line originates from a predictive model or directly from the perception module, Choe teaches overlaying the detected horizon line on the image to generate an overlaid image.

5. [1D] “transmitting the overlaid image to a computing device over a network”

127. Choe renders obvious this element.

128. Choe teaches a fully networked architecture in which an autonomous vehicle communicates with remote servers over LAN, WAN, Internet, cellular, and satellite connections. Specifically, Choe explains:

“Network configuration 100 includes autonomous vehicle 101 that may be communicatively coupled to one or more servers 103-104 over a network 102. ... Network 102 may be any type of networks such as a local area network (LAN), a wide area network (WAN) such as the

Internet, a cellular network, a satellite network, or a combination thereof, wired or wireless.”

EX1005, ¶[0018].

129. Choe further underscores that the vehicle’s wireless communication system is expressly configured to communicate with servers and external devices:

“Wireless communication system 112 is to allow communication between autonomous vehicle 101 and external systems, such as devices, sensors, other vehicles, etc. For example, wireless communication system 112 can wirelessly communicate with one or more devices directly or via a communication network, such as servers 103-104 over network 102.”

EX1005, ¶[0025].

130. Critically, Choe discloses that calibration can be performed remotely based on images captured by the vehicle’s sensors:

“Sensor calibration system 125 may be hosted by server 103 to calibrate sensors such as cameras, LIDAR, and/or RADAR devices offline based on the images and/or point clouds captured by the sensors.”

EX1005, ¶[0031].

131. Choe also teaches that calibration results are then uploaded back to the vehicle:

“The functionalities of sensor calibration module 308 may be maintained in a data analytics system such as server 103 to perform a

sensor calibration of a sensor offline based on the images captured by the corresponding sensor. The calibration result can then be uploaded onto the vehicle to be utilized online during the image processing as a part of the perception process.”

EX1005, ¶[0044].

132. At the same time, Choe expressly discloses local/manual calibration in which images are generated with features, including a horizon line, “superimposed thereon” to provide real-time calibration feedback. *See* Choe ¶¶[0014]–[0017], [0049]–[0057]; Figs. 6A–6B. These passages confirm that the calibration imagery in Choe is not raw data alone but overlaid images with horizon and lane lines drawn across the scene.

133. Although Choe does not expressly state that the overlaid images are transmitted, Choe suggests such transmission and a POSITA would have understood it to be an obvious implementation detail in view of Choe’s networked calibration framework. Choe teaches both (i) generating annotated calibration views with horizon and lane lines superimposed locally (e.g., 2D view 411) and (ii) performing calibration remotely on server 103, with results communicated back to the vehicle. *See* Choe ¶¶[0014]–[0017], [0031], [0044], [0049]–[0057]. A POSITA would have recognized that for remote calibration and user interaction to function, the very same processed views used locally must be transmitted over the network so they can be rendered at the remote terminal. In other words, while Choe does

not describe the transmitting step explicitly, it would have been obvious that its architecture and functionality motivates and supports transmitting the overlaid image, and implementing that transmission would have been a routine and predictable extension of Choe's disclosed system.

134. Stated different, a POSITA would have recognized that Choe's own teachings naturally converge: the very same overlaid images generated and updated locally (with horizon and lane lines superimposed) would be transmitted to the remote calibration server over the disclosed network. On one side, Choe already generates annotated overlays and updates them with each user input; on the other, Choe expressly configures the ADV to transmit calibration imagery to remote servers. Combining these teachings (e.g., sending the already-generated overlaid image instead of only raw frames) would have been a straightforward, predictable extension of Choe's design. *See* Choe ¶¶[0018], [0025], [0031], [0044].

135. The technical motivations are clear. A POSITA would understand that transmitting the overlaid image ensures verification and consistency, allowing remote calibration systems or personnel to see what the vehicle sees rather than redundantly recomputing overlays and risking divergence. *See* EX1036; EX1026 (discussing remote assistance frameworks for rider-only AVs).

136. Further, a POSITA would have understood that overlays encode the results of local perception and user adjustments; transmitting them avoids

repeating detection on the server, improving operational efficiency and reducing compute demands. *See, e.g.*, EX1031 (explaining that on-vehicle preprocessing reduces network overhead and cloud compute).

137. A POSITA would have understood that Choe's server-side data collector and machine-learning engine, which generate predictive models for calibration, would further benefit from annotated frames for training, drift monitoring, and calibration audit trails. *See* EX1005, ¶¶[0029]–[0031]. The importance of curated annotated imagery for AV perception was already well-established through benchmark datasets in corroborating evidence. *See* EX1018, at 1231 (describing human-annotated AV images for detection/SLAM); EX1019, 1 (describing fine-grained pixel-level annotations of urban driving scenes); EX1027 (presenting large-scale per-pixel annotated urban driving data). *See also* EX1025, col. 6, ll. 15–35 (filed Mar. 2016, issued Mar. 2018) (Zoox) (describing an AV system architecture in which teleoperator systems and an AV management server exchange sensor logs and calibration data).

138. And in human-in-the-loop scenarios, a POSITA would have understood that transmitting the overlaid image would provide remote technicians with the identical UI view the vehicle uses, ensuring unambiguous calibration context. *See* EX1029.

139. A POSITA would have understood that implementation would have required only routine design choices. A POSITA could either rasterize overlays into the video frame and transmit the composite image (e.g., JPEG/PNG), as routinely done in AV teleoperation UIs and visualization tools. *See* EX1020, 289–90 (describing separate publication of sensor feeds and detection overlays). These alternatives reflect routine engineering trade-offs between sending composite images or metadata, both well-established by 2019. *See* EX1031, 462–63. Either approach would result in the remote device receiving the “overlaid image,” fully within Choe’s network architecture and overlay-generation framework. *See* Choe ¶¶[0014]–[0017], [0049]–[0057], [0018], [0025]. Either approach would have been obvious to try with a reasonable expectation of success.

140. Finally, potential counterarguments do not undermine this conclusion. Nor do potential counterarguments undercut this conclusion. While Choe does not explicitly state “transmit the overlaid image,” it does not need to. Choe already discloses both the generation of overlays and the network-based transmission of calibration imagery. Any concerns about bandwidth would be misplaced because overlay data adds negligible load compared to the base frame, and Choe already contemplates Internet and cellular links and even offline bulk processing. *See* EX1005, ¶¶[0018], [0031]. Finally, although overlays are described in connection with local UI, Choe’s server-hosted calibration and analytics provide a clear reason

to share annotated images remotely. *See* EX1005, ¶¶[0031], [0044]. Nothing in Choe teaches away from doing so.

141. Accordingly, Choe renders obvious this element.

6. [1E] “receiving a modification of the line from the computing device, the modification comprising a new line at a second position”

142. Choe renders obvious this element.

143. Choe expressly teaches modifying roadway features, including lane and horizon lines, once overlaid on an image. Choe explains: *“in response to a second input signal received from the input device (e.g., left or right arrow key), the position of at least one of the lane lines is modified ... [and] a second calibration factor ... is determined based on the modification”* (Choe ¶[0017]). Likewise, Choe describes that *“a user may modify a detected horizon line or lane line by providing input through an input device ... the calibration module may then compute a calibration parameter based on the modified line.”* Choe ¶¶[0052]–[0053]). These disclosures show that the calibration system receives a modification of an existing line, which produces a new line at a second position within the displayed image.

144. At the same time, Choe describes a networked calibration architecture in which images are transmitted to server 103 for calibration, with results uploaded back to the vehicle. *See* Choe ¶[0031] (*“Sensor calibration system 125 may be*

hosted by server 103 to calibrate sensors ... offline based on the images ... captured by the sensors”); ¶[0044] (“The calibration result can then be uploaded onto the vehicle to be utilized online during the image processing as a part of the perception process”). These passages make clear that calibration functionality, including user-driven adjustments, is not confined to the vehicle but can be implemented at a remote computing device.

145. A POSITA would have readily understood that the very same line-modification operation described locally in Choe (paragraph [0017]) would have been obvious to be performed within this remote/server workflow. Choe already teaches both (1) the precise act of receiving user input to modify an overlaid line, resulting in a new line at a second position, and (2) a distributed calibration system in which servers perform calibration using images transmitted from the ADV. Combining these teachings would have been a routine and predictable design choice: enabling a remote computing device to receive user-driven modifications of horizon or lane lines and returning the modified calibration data to the vehicle. Moreover, combining them according to their established functions produces nothing more than the expected result: line modification performed remotely. From a technical perspective, it would have been natural to combine these two aspects. The user-driven line adjustments are simply metadata applied to the image, and transmitting them to the server requires no more than sending the same annotation

along with the image stream already described. Doing so would relieve the vehicle of local compute burden, centralize calibration tasks for consistency across a fleet, and allow remote operators to perform or verify corrections in real time. These are straightforward engineering optimizations that would have followed directly from Choe's architecture and goals.

146. Further, transferring the line-adjustment step to the server environment would have been beneficial, offering benefits such as reducing on-vehicle compute load, centralizing calibration, and enabling distributed human-in-the-loop correction. The motivation for such an extension is straightforward.

147. **Motivation & industry context.** Remote, human-in-the-loop workflows were already recognized in AV practice to increase accuracy and provide independent checks on onboard automation. For example, Nissan's Seamless Autonomous Mobility (SAM), developed with NASA, explicitly put remote human operators in the loop to resolve difficult perception/decision situations for AVs, highlighting the role of off-vehicle human review and intervention in safety-critical contexts (2017–2019). *See* EX1036; EX1037. In parallel, the AV ecosystem widely relied on human annotation of vision data (including lane and road markings) to improve model accuracy and calibration quality; leading datasets (KITTI, Cityscapes) documented extensive, high-quality

manual labeling and quality control as foundational to perception fidelity. *See e.g.* EX1038; EX1039; EX1040; EX1019, 1; EX1041.

148. **Verification & validation (V&V) imperatives.** By the mid-2010s, U.S. automotive safety authorities and industry groups already mandated rigorous verification, validation, and independent review for electronic control systems. The National Highway Traffic Safety Administration's 2016 survey of safety standards stressed the importance of lifecycle V&V and independent assessment in automotive electronics. *See* EX1042; EX1043.

149. U.S. human factors evaluations conducted prior to 2020 likewise recognized the need to anticipate functional limitations in perception-driven automation and to mitigate them through design-time measures. For example, NHTSA's 2015 Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts examined automation insufficiencies, handoff strategies, and the role of human oversight in monitoring system outputs. *See* EX1044.

150. Similarly, NHTSA's 2018 Human Factors Design Guidance for Level 2 and Level 3 Automated Driving Concepts provided structured guidance on driver-vehicle interfaces, emphasizing proactive design to communicate automation limitations and preserve human-in-the-loop control. *See* EX1045.

151. Consistently, NHTSA's Federal Automated Vehicles Policy (2016) called for safety assessments and documentation across development and update

cycles, reflecting a regulatory expectation of robust validation processes for ADS functions. *See* EX1046.

152. Against that backdrop, shifting the line-adjustment step (already disclosed by Choe locally) to a remote computing device fits established AV operational patterns that (i) centralize expert oversight, (ii) reduce on-vehicle compute/UX burden, and (iii) create an auditable V&V trail, which are benefits expressly contemplated by the networked architectures and remote operations in the period. Industry trajectories toward remote oversight/teleoperation (e.g., Nissan SAM; Phantom Auto) show contemporaneous acceptance of off-board human review to resolve edge cases and validate perception outputs. *See* EX1047; EX1037.

153. Further, Choe's architecture contemplates server-hosted calibration and data analytics, meaning that enabling a remote computing device to receive and apply line modifications would be a predictable use of known elements according to their established functions. Nothing in Choe teaches away from such an arrangement, and the benefits of consistency, accuracy, and human oversight would have motivated a POSITA to implement this feature, making this feature obvious.

7. [1F] “computing a camera parameter based on the new line”

154. At the outset, this step does not specify where the computation occurs.

Read in context, it encompasses computation performed either locally or on a remote computing device; in either case, Choe discloses this element.

155. Choe expressly describes computing camera parameters (pitch, yaw, roll) based on updated or new line positions after user modification. For example, Choe states:

“According to one embodiment, in response to a first image captured by a camera of an ADV, a horizon line is determined based on the camera's hardware settings. ... In response to a first input signal (e.g., an up or down arrow key) received from an input device ... a position of the horizon line is updated ... A first calibration factor or first correction value is determined for calibrating a pitch angle of the camera based on a difference between the initial horizon line and the updated horizon line.”

EX1005, ¶[0015]; *see also* EX1005, ¶[0049].

156. Choe further discloses that

“[i]n response to a second input signal ... the position of at least one of the lane lines is modified ... A second calibration factor for calibrating a yaw angle of the camera is determined based on the modification of the positions of the lane lines. Similarly, in response to a third signal ... the lane lines are modified and a third calibration factor for calibrating a roll angle of the camera is determined.”

EX1005, ¶¶[0017], [0052]-[0053].

157. Choe's process flow confirms this functionality: "At block 705, processing logic determines a calibration factor or parameter for calibrating a pitch angle of the camera based on the difference between the initial horizon line and the updated horizon line." EX1005, ¶[0059], FIG. 7.

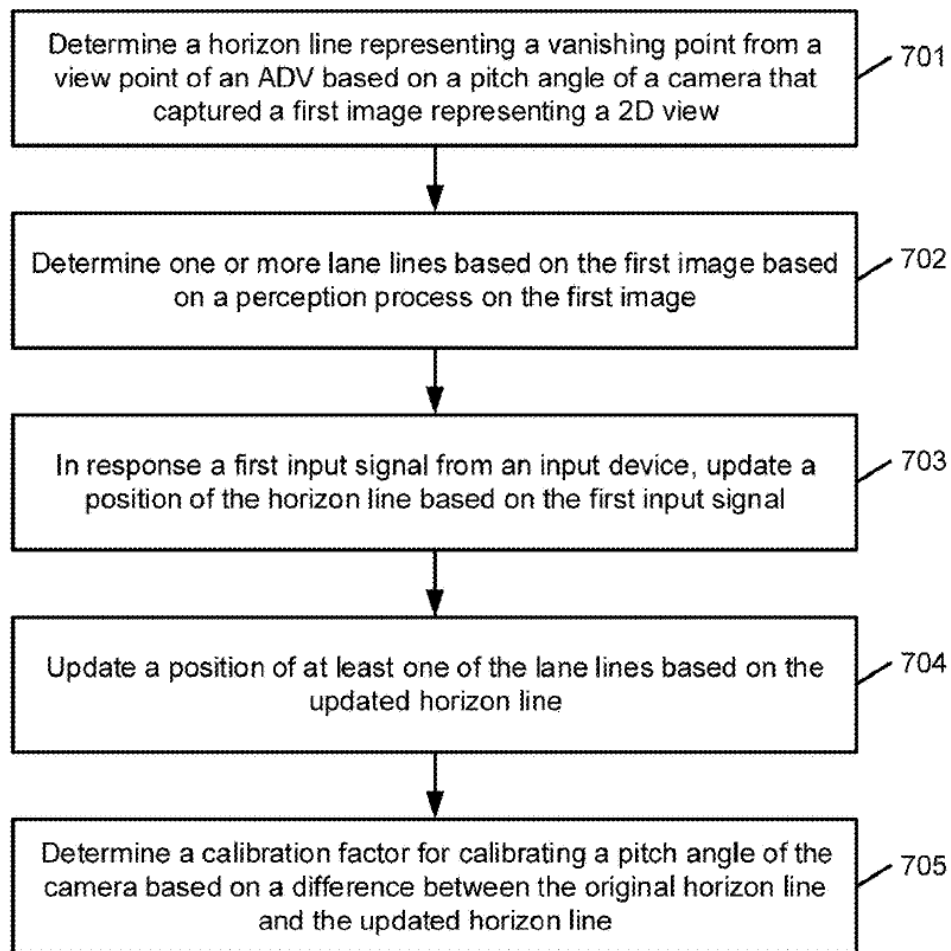


FIG. 7

158. These disclosures make clear that once a new line is generated at a second position (e.g., after receiving a user input to shift a horizon or lane line), Choe computes a calibration factor or parameter for the camera based on that new line. Thus, Choe directly teaches computing camera parameters based on new line positions.

159. **Local computation:** Choe describes this computation being performed onboard the ADV by calibration system 308, in real time, as the user provides incremental adjustments to the horizon and lane lines. *See* EX1005, ¶¶[0015], [0017], [0049]–[0053], [0059]. A POSITA would have understood that these “calibration factors” are the very camera parameters (pitch, yaw, roll) claimed.

160. **Remote computation:** Choe also discloses that the same calibration computations may be performed remotely on a server. For example, “Sensor calibration system 125 may be hosted by server 103 to calibrate sensors such as cameras, LIDAR, and/or RADAR devices offline based on the images and/or point clouds captured by the sensors.” EX1005, ¶[0031]. Likewise, “The functionalities of sensor calibration module 308 may be maintained in a data analytics system such as server 103 to perform a sensor calibration ... offline based on the images captured by the corresponding sensor. The calibration result can then be uploaded onto the vehicle to be utilized online during the image processing as a part of the

perception process.” EX1005, ¶[0044]. Choe’s networking disclosures confirm that the ADV is designed to transmit sensor imagery to these servers over LAN, WAN, Internet, or cellular links. *See* EX1005, ¶¶[0018], [0025].

161. In my opinion, a POSITA would readily understand that Choe teaches computing camera parameters based on new line positions both locally and remotely. Locally, calibration factors for pitch, yaw, and roll are explicitly calculated after a line is modified, producing updated camera parameters. Remotely, Choe discloses server-hosted calibration systems that perform the same computations offline based on uploaded sensor imagery, then return calibration results to the vehicle.

162. Accordingly, a POSITA would have understood that Choe discloses “computing a camera parameter based on the new line,” whether performed on-board the vehicle or by a remote server.

8. [1G] “transmitting data representing the camera parameter to the camera device.”

163. At the outset, this limitation does not specify “over a network.” Thus, it is satisfied by either (i) local/in-vehicle transmission (e.g., an ECU or calibration module sending updated pitch/yaw/roll to the camera device over an internal bus) or (ii) remote transmission where parameters computed off-board are sent back and then delivered to the camera device. Choe teaches both.

164. Choe teaches that once calibration factors (corresponding to camera parameters such as pitch, yaw, and roll) are determined, those parameters are transmitted back to the targeted sensor device, such as the camera, so that its internal parameters can be updated and the hardware compensated. For example, Choe explains:

“...The processes similar to the above process can be implemented to determine calibration factors for LIDAR and/or RADAR devices. The above processes can be utilized online while the vehicle is driving and real-time images are captured to calibrate the sensors in real-time. Alternatively, the above processes can be performed offline based on the previously captured images or point clouds. The calibration factors can be utilized by software applications such as a perception module to adjust the image processing parameters such as pitch, yaw, and roll angles to compensate the hardware during the image processing.”

EX1005, ¶[0017].

165. Choe further discloses that:

“...Once the calibration results are satisfied, calibration factor calculator 406 determines the calibration factor 410 and feeds back to perception module 302 or other software modules such that the software modules can perform proper adjustment at the software level to compensate the hardware settings of the sensors.”

EX1005, ¶[0047].

166. Choe also explains:

“Sensor calibration module or system 308 is configured to calibrate, using sensor calibration algorithms 313, a sensor based on an image or point cloud captured by the sensor to determine a set of calibration parameters. The calibration parameters can be utilized by the software to compensate the hardware settings of the sensors during a perception process. ... The calibration result can then be uploaded onto the vehicle to be utilized online during the image processing as a part of the perception process.”

EX1005, ¶[0044].

167. And Choe confirms that this process applies not only locally but also in remote calibration scenarios:

“...The sensor calibration process may be performed offline based on the previously captured sensor data captured by the targeted sensors of ADVs. The calibration parameters can then be uploaded to the corresponding vehicles and utilized by autonomous driving software to calibrate and compensate the corresponding sensors.”

EX1005, ¶[0031].

168. In my opinion, a POSITA would have understood that Choe describes transmitting data representing the determined calibration parameters (pitch, yaw, roll) back to the sensor device, specifically, to the camera being calibrated. When calibration is performed locally, Choe discloses feeding the correction values to

perception and control software, which then adjusts the camera's settings. When calibration is performed remotely, Choe explicitly states that the calibration parameters are uploaded back to the vehicle for use in compensating the corresponding sensors. In both scenarios, the effect is the same: the computed camera parameters are transmitted to the camera device so that its internal parameters are updated and proper calibration is achieved.

169. Accordingly, a POSITA would have recognized that Choe discloses, the step of “transmitting data representing the camera parameter to the camera device.”

B. Claim 2: The method of claim 1, wherein the camera parameter comprises one of camera height, viewing angle, and road plane normal.

170. Choe discloses or at least renders obvious this element.

171. Choe expressly teaches that its calibration process determines camera parameters corresponding to the geometric orientation of the camera, including pitch, yaw, and roll. These parameters describe the relationship of the camera to the road plane and the surrounding environment. For example, Choe explains:

“In response to a first input signal (e.g., an up or down arrow key) ... a position of the horizon line is updated ... A first calibration factor or first correction value is determined for calibrating a pitch angle of the camera based on a difference between the initial horizon line and the updated horizon line.”

EX1005, ¶[0015]; *see also* EX1005, ¶[0049].

172. Choe further discloses that:

“...in response to a second input signal ... a second calibration factor for calibrating a yaw angle of the camera is determined based on the modification of the positions of the lane lines. Similarly, in response to a third signal ... the lane lines are modified and a third calibration factor for calibrating a roll angle of the camera is determined.”

EX1005, ¶¶[0017], [0052]–[0053].

173. These calibration factors are directly tied to the geometric configuration of the camera relative to the roadway. Pitch corresponds to the vertical tilt of the camera and governs the apparent vertical placement of the horizon in the image; yaw corresponds to the camera’s azimuthal orientation and governs the horizontal placement of the road vanishing point and left–right alignment of lane features; roll describes rotation about the optical axis and governs the tilt of the horizon and the perceived skew of lane markings. Each parameter, as dictated by standard projective geometry, affects how the 3D road plane projects onto the 2D image plane.

174. **Camera Height:** Although Choe does not expressly recite the term “camera height,” a POSITA would have understood that pitch calibration as disclosed in Choe (paragraph [0015]) accounts for the physical mounting height of the camera. The location of the horizon line in the image is governed jointly by (i) the pitch angle and (ii) the height of the camera above the road surface. For a fixed

pitch, a higher mounting pushes the horizon line lower in the frame, while a lower mounting raises it. Thus, when Choe adjusts pitch calibration based on displacement of the horizon line, the geometry being solved depends on the camera's mounting height relative to the roadway. A POSITA would have recognized that this linkage between horizon displacement, pitch, and camera height makes camera height a parameter implicit or at least obvious in Choe's calibration process.

175. **Viewing Angle:** Pitch also defines the vertical viewing angle of the camera relative to the road plane. As Choe explains in ¶[0015], modifying the horizon line position by user input alters the calibration factor for pitch. A POSITA would have understood that pitch is the rotation of the camera about its lateral (side-to-side) axis. In practical terms, pitch represents whether the camera is tilted upward or downward relative to the ground plane. A POSITA would have understood that pitch directly defines the *viewing angle* of the roadway scene, because even a small adjustment in vertical tilt substantially alters how far ahead the horizon appears in the captured image and how the road surface is projected onto the image plane. For example, tilting the camera downward increases the portion of the near roadway visible in the image, while tilting it upward raises the horizon line and increases the visible distance ahead.

176. **Road Plane Normal.** Choe's disclosures of pitch, yaw, and roll calibration collectively define the camera's full 3D orientation relative to the road plane. Pitch is rotation about the lateral axis, yaw is rotation about the vertical axis, and roll is rotation about the optical axis. Together, these three angles specify the direction of the road plane's normal vector in the camera coordinate frame. Choe explains that these calibration factors are applied so that "the calibration factors can be utilized by software applications such as a perception module to adjust the image processing parameters such as pitch, yaw, and roll angles to compensate the hardware during the image processing" (Choe ¶[0017]). A POSITA would have understood that the purpose of adjusting pitch, yaw, and roll is to correctly recover the road plane geometry, which in practice is represented by the road plane normal.

177. Yaw (rotation about the vertical axis) controls the horizontal position of the road vanishing point and the left-right symmetry of projected lane lines. When yaw is off, parallel lane boundaries converge toward a vanishing point that is laterally offset, and lane centerlines appear slanted relative to the image vertical. Choe's adjustment of lane-line positions and computation of a yaw correction factor ([0017], [0052]) is consistent with estimating yaw by minimizing lateral residuals between observed lane projections and their expected symmetric alignment in the camera frame.

178. Roll (rotation about the optical axis) manifests as a tilted horizon and a corresponding slant of lane markings relative to the image horizontal. A nonzero roll causes the road plane's horizon to deviate from horizontal in the image and introduces a systematic slope to features that should be level. Choe's disclosure of modifying lane lines and determining a roll calibration factor ([0017], [0053]) aligns with standard practice of estimating roll from the slope of the horizon/road-edge ensemble or from the aggregate orientation of lane markings, thereby recovering the road-plane normal in the camera frame

179. Thus, Choe's calibration process explicitly uses these geometric parameters to align the vehicle's perception system with the road surface. For example, Choe explains that "the calibration factors can be utilized by software applications such as a perception module to adjust the image processing parameters such as pitch, yaw, and roll angles to compensate the hardware during the image processing." EX1005, ¶[0017]. A POSITA would have understood that the purpose of adjusting these angles is to ensure proper estimation of the road plane geometry, which in practice is represented by the road plane normal vector, and to account for the effective camera mounting position (camera height) and orientation (viewing angle).

180. In my opinion, a POSITA would have recognized that Choe's explicit disclosure of pitch, yaw, and roll calibration factors encompasses the claimed

camera parameters. Pitch angle calibration corresponds to the effective viewing angle of the camera relative to the road. Yaw and roll adjustments, when combined with pitch, define the camera's orientation in 3D space, which is the mathematical basis for computing the road plane normal. Moreover, pitch calibration derived from the horizon line implicitly depends on the relative camera height above the ground, as the vanishing point and horizon shift with camera mounting height. Thus, even if Choe does not use the exact terms "camera height," "viewing angle," or "road plane normal," it plainly renders obvious these parameters through its calibration of pitch, yaw, and roll angles. In my opinion, a POSITA would have recognized that computing "camera height," "viewing angle," and "road plane normal" requires nothing more than applying the calibration process Choe already describes. These parameters are simply alternate formulations of the same geometric concepts (pitch/yaw/roll) that Choe explicitly discloses and uses to calibrate the vehicle camera system.

181. Accordingly, even if Choe does not expressly disclose the camera parameter comprises one of camera height, viewing angle, and road plane normal itself it would have been obvious for a POSITA to implement this feature in view of Choe's teachings. The only difference is the terminology: Choe calls the outputs "calibration factors" for pitch, yaw, and roll, while the claim refers to "camera parameters" such as camera height, viewing angle, or road plane normal. To a

POSITA, these are interchangeable descriptors of the same underlying geometry, and Choe's system clearly teaches or at least renders obvious computing the claimed parameters from the newly positioned line

C. Claim 3: The method of claim 1, wherein receiving the image of the roadway comprises receiving an image of the roadway while the vehicle is moving.

182. Choe discloses this element.

183. Choe teaches that its calibration processes can be performed in real time while the autonomous vehicle is in motion. For example, Choe states:

“...The above processes can be utilized online while the vehicle is driving and real-time images are captured to calibrate the sensors in real-time. Alternatively, the above processes can be performed offline based on the previously captured images or point clouds.”

EX1005, ¶[0017].

184. Choe further explains that calibration updates to horizon and lane lines may occur during driving, with corresponding pitch, yaw, and roll calibration factors computed while the vehicle is operating. *See* EX1005, ¶¶[0052]–[0054].

The process flow confirms this functionality: “At block 705, processing logic determines a calibration factor or parameter for calibrating a pitch angle of the camera based on the difference between the initial horizon line and the updated horizon line.” EX1005, ¶[0059].

185. Choe also confirms that roadway images and environmental data are continuously received while the vehicle is moving. For example, Choe states:

“While autonomous vehicle 101 is moving along the route, perception and planning system 110 may also obtain real-time traffic information ... [and] real-time local environment data detected or sensed by sensor system 115 (e.g., obstacles, objects, nearby vehicles).”

EX1005, ¶[0028].

186. In my opinion, a POSITA would have understood that these disclosures collectively confirm that Choe receives roadway images while the vehicle is moving. The explicit reference to “online while the vehicle is driving” and “real-time images” (¶[0017]) shows that the calibration system processes camera images during vehicle operation. Moreover, Choe’s disclosure that perception and planning continuously obtain “real-time local environment data” while the vehicle moves along its route (¶[0028]) reinforces that roadway images are received during motion. Together, these disclosures demonstrate that Choe supports the claimed limitation that “receiving the image of the roadway comprises receiving an image of the roadway while the vehicle is moving.”

D. Claim 5: The method of claim 1, wherein transmitting the overlaid image to the computing device over the network comprises transmitting the overlaid image to a web-based application.

187. Choe renders obvious this claim element

188. Choe teaches a networked architecture in which autonomous vehicles are communicatively coupled to remote servers over a variety of networks, including the Internet. For example, Choe explains:

“Network configuration 100 includes autonomous vehicle 101 that may be communicatively coupled to one or more servers 103-104 over a network 102. ... Network 102 may be any type of networks such as a local area network (LAN), a wide area network (WAN) such as the Internet, a cellular network, a satellite network, or a combination thereof, wired or wireless.”

EX1005, ¶[0018].

189. Choe further states:

“Wireless communication system 112 is to allow communication between autonomous vehicle 101 and external systems, such as devices, sensors, other vehicles, etc. For example, wireless communication system 112 can wirelessly communicate with one or more devices directly or via a communication network, such as servers 103-104 over network 102.”

EX1005, ¶[0025].

190. Choe discloses that calibration may be performed remotely on a server, and calibration results are transmitted back to the vehicle. For example:

“Sensor calibration system 125 may be hosted by server 103 to calibrate sensors ... offline based on the images and/or point clouds captured by the sensors. ... The calibration parameters can then be uploaded to the corresponding vehicles and utilized by autonomous driving software to calibrate and compensate the corresponding sensors.”

EX1005, ¶[0031]; *see also* EX1005, ¶[0044].

191. In my opinion, a POSITA would have understood that these disclosures render it obvious to implement transmission of overlaid images to a web-based application to achieve scalability, accessibility, and centralized verification within Choe’s existing networked architecture. Choe already discloses (i) generating overlaid images locally for calibration (¶[0015]–[0017], [0055]–[0057]), (ii) transmitting sensor images and calibration data over the Internet to servers (¶[0018], [0025], [0031], [0044]), and (iii) server-hosted calibration functionality (¶[0031], [0044]). A POSITA would recognize that a “web-based application” is simply one of the conventional ways to implement server-hosted processing over the Internet.

192. Conditioning the remote transmission on a web-based architecture would have been a predictable use of known design alternatives. At the time of the invention, it was well known that Internet-based applications (including browser-

accessible or cloud-based services) were routinely used to deliver data and visualization to remote computing devices. Selecting a web-based application as the interface to receive and display the overlaid image is merely a design choice among finite, predictable options (e.g., desktop client, native application, or web application), yielding no unexpected results. Choe emphasizes that calibration may be performed remotely on a server (§[0031]) and that autonomous vehicles exchange image and calibration data with those servers over the Internet (§[0018], [0025], [0044]). A POSITA would recognize that enabling remote technicians, engineers, or quality-assurance personnel to review and verify calibration imagery (such as overlaid lane and horizon lines) requires an interface capable of displaying those images at the server side without installing specialized software on each client machine.

193. By 2017–2019, production AV stacks already exposed browser UIs for viewing overlaid perception data. Baidu Apollo’s Dreamview is “a web application” used to visualize modules and driving outputs. *See* EX1048 (describing Dreamview as “a web application that helps developers visualize the output of autonomous driving modules”). Likewise, Cruise released Webviz, explicitly “visualizing robotics data in the browser” and described as “a web-based application for playback and visualization of ROS bag files.” *See* EX1049; EX1050; EX1051. In robotics generally, the Robot Web Tools initiative (IROS

2015) formalized browser/WebSocket clients for robot data and visualization, i.e., web apps as front-ends for remote robot telemetry and control. *See* EX1052.

194. In parallel, mainstream cloud/IoT stacks provided web dashboards for device data well before 2020, normalizing browser delivery of streamed telemetry and visuals. *See, e.g.*, EX1053. In automotive telematics specifically, patents taught Internet-delivered vehicle data to remote UIs: U.S. Patent No. 7,650,210 B2 (2010) directs diagnostic outputs “via the Internet” to “an Internet-enabled device,” and U.S. Pub. No. 2013/0246135 A1 provides a “remote user-interface via the network interface” for vehicle information, both illustrating web-based remote presentation as routine.

195. Against this backdrop, implementing the “receiving ... from the computing device” step through a web application (rather than a native desktop client) would have been a conventional substitution among well-known delivery channels that AV teams were already using to transmit and display overlaid imagery and telemetry to remote operators. The motivation would have been clear: using a web-based application allows calibration personnel or remote servers to view, process, or verify the exact annotated images (horizon lines, lane lines, overlays) generated by the ADV without requiring specialized local software installation. This supports Choe’s goals of enabling server-hosted calibration, fleet-

wide analytics, and human-in-the-loop verification. *See* EX1005, ¶¶[0029]–[0031], [0044].

196. Accordingly, a POSITA would have understood that transmitting the overlaid image to a web-based application is an obvious implementation of Choe’s disclosed Internet-based calibration system.

E. Claim 6: The method of claim 1, wherein receiving the modification of the line from the computing device comprises receiving an identification of a new horizon line by a user of the computing device.

197. Choe renders obvious this element.

198. Choe already discloses that a user may interactively adjust roadway feature overlays, including the horizon line, through an input device. For example, Choe explains: “in response to a second input signal received from the input device (e.g., left or right arrow key), the position of at least one of the lane lines is modified ... and a second calibration factor ... is determined based on the modification” (¶[0017]). Choe further states: “a user may modify a detected horizon line or lane line by providing input through an input device ... the calibration module may then compute a calibration parameter based on the modified line.” EX1005, ¶¶[0052]–[0053]. These passages make clear that the calibration system is configured to accept user adjustments to the horizon line and then use the updated line for calibration. Functionally, this is equivalent to the system receiving the user’s identification of a new horizon line.

199. The claimed term “identification” would have been readily understood by a POSITA to describe exactly what Choe already shows: the user setting the horizon line to a new, explicit position through input actions. In Choe, the horizon-line “modification” is not merely a suggestion, it is an affirmative action that specifies the new line geometry in image coordinates. Once that new position is

accepted, the system recognizes it as the current horizon line and uses it to calculate pitch, roll, and other calibration values. *See* EX1005, ¶¶[0015], [0052]–[0053]. Thus, receiving this updated geometry is technically the same as receiving an “identification” of the new horizon line by the user.

200. The same conclusion applies whether the calibration occurs locally on the vehicle or remotely on a server. Additionally, Choe itself suggests implementing the same user-adjustment at the server node it discloses: Choe teaches a client–server calibration framework in which images from the ADV are processed by a server-hosted sensor calibration system and results uploaded back to the vehicle. EX1005 ¶¶[0018], [0025], [0031], [0044]. As discussed in connection with element [1E], Choe discloses a distributed architecture in which images are transmitted to a server for calibration and results are returned to the vehicle. EX1005, ¶¶[0031], [0044]. In that framework, a POSITA would have understood that the same user identification step (i.e, adjusting the horizon overlay) could be carried out through a remote computing device or web interface. The server would then receive the updated horizon line and compute calibration from it, producing exactly the same effect as the local operation. Moving this well-defined adjustment step from the in-vehicle UI to the remote environment would have been a straightforward engineering choice using the same disclosed mechanisms for transmitting images and calibration results.

201. In my view, Choe already teaches the three essential elements: (i) an overlaid horizon line, (ii) user-driven modification of that line, and (iii) computation of calibration factors based on the updated horizon. The additional phrasing of Claim 6 (“receiving an identification of a new horizon line by a user”) is simply another way of describing the same process. To the extent the claim is read to require receipt of a user-supplied horizon line at a remote device, Choe’s server-based calibration model makes that an ordinary and predictable extension and therefore obvious. Further motivation is outlined below.

202. **Known technique applied in a disclosed context.** Choe already teaches a human-in-the-loop adjustment of the horizon/lane lines and a networked calibration pathway. Specifically, Choe discloses receiving user inputs that move the horizon line and then computing calibration from the difference between the initial and updated line (¶¶[0015], ¶¶[0049], with visual feedback at ¶¶[0016], and GUI examples at ¶¶[0055]–[0057], FIGS. 6A–6B), and separately discloses a client–server calibration architecture with images sent to and results returned from a computing device “over a network” (¶¶[0018], ¶¶[0025], ¶¶[0031], ¶¶[0044]). Implementing the user identification step on the computing device that is already in Choe’s networked loop is thus the use of a known technique (user-guided line setting) in the very architecture Choe discloses for performing calibration, with a reasonable expectation of success and predictable results.

203. Combination of known elements with predictable results. Choe

teaches two elements: (i) horizon-line adjustment via user input (¶[0015], ¶[0049], ¶[0055]–[0057]) and (ii) network communication of calibration artifacts between the vehicle and a computing device (¶[0018], ¶[0025], ¶[0031], ¶[0044]).

Combining them so that the same computing device that displays the overlay also returns the user-identified line is a straightforward integration of known, co-taught features that works exactly as expected: the overlaid image is sent out; the user identifies the corrected horizon on that device; the updated line (or its parameters) is returned; calibration proceeds from the updated line (¶[0015], ¶[0016], ¶[0049]).

204. Design incentives and common sense. Choe highlights a networked architecture and server-side calibration (¶[0018], ¶[0025], ¶[0031], ¶[0044]).

Placing the user-identification step on the computing device that is already processing calibration reduces on-vehicle UX/compute burden, centralizes review, and keeps the calibration loop coherent, which are all ordinary design incentives for safety-critical perception workflows. Implementing that step and returning the updated line is, in my opinion, a common-sense optimization of Choe's disclosed flow, not a departure from it.

F. Claim 7: The method of claim 1, wherein transmitting the data representing the camera parameter to the camera device further comprises recomputing the camera parameter based on the modification of the line.

205. Choe discloses this element.

206. Choe explicitly teaches that when a line (horizon or lane line) is modified based on user input, the system recomputes calibration parameters (pitch, yaw, or roll) based on that modification. For example, Choe explains:

“In response to a first input signal ... a position of the horizon line is updated ... A first calibration factor or first correction value is determined for calibrating a pitch angle of the camera based on a difference between the initial horizon line and the updated horizon line.”

EX1005, ¶[0015]; *see also* EX1005, ¶[0049].

207. Choe further states:

“According to one embodiment, in response to a second input signal ... the position of at least one of the lane lines is modified ... A second calibration factor for calibrating a yaw angle of the camera is determined based on the modification of the positions of the lane lines. Similarly, in response to a third signal ... the lane lines are modified and a third calibration factor for calibrating a roll angle of the camera is determined.”

EX1005, ¶[0017]; *see also* ¶¶[0052]–[0053].

208. Choe’s flow diagram further reinforces this point: “At block 705, processing logic determines a calibration factor or parameter for calibrating a pitch angle of the camera based on the difference between the initial horizon line and the updated horizon line.”

EX1005, ¶[0059].

209. Choe also teaches that once these calibration factors are determined, they are transmitted back to adjust the camera device or used by perception software to compensate hardware settings:

“The calibration factors can be utilized by software applications such as a perception module to adjust the image processing parameters such as pitch, yaw, and roll angles to compensate the hardware during the image processing.”

EX1005, ¶[0017]; *see also* EX1005, ¶[0047].

210. In my opinion, a POSITA would have understood that Choe discloses recomputing the camera parameters based on modifications of the line and then transmitting those recomputed parameters back to the camera device (or to perception/control modules acting on the camera). Every time the horizon or lane line is adjusted by the user, the calibration system computes a new correction factor, EX1005, ¶¶[0015], [0017], [0049], [0052]–[0053]. In other words, it recomputes the camera parameter in light of the line modification. These updated

parameters are then fed back into the system, EX1005, ¶[0047], or uploaded from a server, EX1005, ¶¶[0031], [0044], to be applied to the sensor hardware.

211. Thus, Choe discloses not only transmitting calibration parameters to the camera device, but also that such transmission follows recomputation of the parameters after each line modification. Without recomputation, the calibration parameters would not reflect the updated line position.

212. Accordingly, Choe discloses the limitation of claim 7 requiring that transmitting the data representing the camera parameter to the camera device further comprises recomputing the camera parameter based on the modification of the line.

G. Claims 8-10, 12-16 18-20

213. Claims 8-10, 12-16 18-20 recite the same substantive limitations as claims 1–7, but expressed in different statutory formats (computer-readable medium and device claims rather than method claims).

214. Claim 8 mirrors the method of claim 1, except that it is directed to “a non-transitory computer-readable storage medium” storing program instructions. Claims 9–10 and 12-14 then correspond directly to dependent claims 2-3 and 5-7, adding the same minor refinements such as specifying the type of camera parameter (claim 9 vs. claim 2), conditioning image receipt on vehicle motion (claims 10 vs. claims 3), transmitting to a web-based application (claim 12 vs.

claim 5), receiving a new horizon line from a user (claim 13 vs. claim 6), and recomputing parameters after line modification (claim 14 vs. claim 7).

215. Likewise, claim 15 recites the same functional steps as claims 1 and 8 but in the context of a “device” with a processor and storage medium, and claims 16 and 18-20 track the same refinements as claims 3, 5-7 and 9-10 and 12-14.

216. The only differences between claims 1-3 and 5-7 and claims 8-10, 12-16 and 18-20 are the claim formats (method vs. storage medium vs. device).

217. A POSITA would have understood that the same disclosed calibration functionality could be implemented as software instructions stored on a non-transitory medium (claims 8-10 and 12-14) or as logic executed on a processor within a device (claims 15-16 and 18-20). Choe discloses as much.

218. Choe already describes implementing calibration through software and hardware modules, including perception module 302, calibration module 308, and associated processors and storage. *See, e.g.*, EX1005, ¶¶[0032]–[0034], [0044], [0047]. Thus, recasting the same operations into “computer-readable medium” or “device” form involves no substantive difference, but merely reflects conventional statutory claiming techniques.

219. Accordingly, for the same reasons set forth with respect to claims 1–7, claims 8-20 are rendered obvious by Choe.

IX. GROUND 2: CHOE IN VIEW OF DAVIES RENDERS OBVIOUS CLAIMS 1-3, 5-10, 12-16 AND 18-20.

220. As I explained in Ground 1, Choe discloses a complete calibration workflow that (i) receives roadway images, (ii) overlays horizon and lane lines, (iii) allows user input to modify those lines, and (iv) recomputes calibration factors (pitch, yaw, roll) based on the new line positions. *See* EX1005, ¶¶[0015], [0017], [0049]–[0053], [0059]. Choe also expressly discloses that calibration may be performed locally within the vehicle or remotely at servers via network 102, with calibration results transmitted back to the ADV. *See* EX1005, ¶¶[0018], [0025], [0031], [0044].

221. However, I understand Patent Owner may argue that Choe does not expressly disclose two-way communication of modified overlay data between remote servers and the vehicle. It is true that some of Choe’s examples describe line modification in a local context, where a user manipulates horizon or lane lines via a keyboard, joystick, or voice system. *See* EX1005, ¶¶[0015], [0017], [0049]–[0053]. But Choe also goes further, teaching that calibration functionality can be hosted remotely on server 103, based on images transmitted from the vehicle, with calibration results uploaded back for in-vehicle use. *See id.* ¶¶[0031], [0044]. And as I discussed above for Ground 1, element [1D], a POSITA would have recognized that for such remote calibration and user interaction to function, the very same processed, annotated views used locally must be transmitted over the

network so they can be rendered at the remote terminal. The only aspect not expressly detailed in Choe is the specific mechanics of how the processed and annotated images are made available for remote calibration. A POSITA seeking to understand the implementation would naturally look to Choe's disclosed networked architecture and then consult complementary references that provide such transmission and display details. Davies supplies exactly those implementation details, showing how annotated images can be transmitted and rendered in a remote, web-based environment.

222. Davies confirms and complements Choe's teachings by expressly disclosing transmission of adjusted or processed image data to and from external computing devices. For example, Davies describes a distributed calibration and vision-processing system in which images and calibration data are transmitted wirelessly, and modifications made at a remote computing device are sent back to update parameters locally. *See* EX1007, ¶¶[0059], [0063]–[0069], claim 24.

223. A POSITA would have recognized that Davies fills any arguable gap in Choe by making clear that two-way communication of calibration adjustments, including user modifications of lines, is a conventional and predictable feature of networked calibration systems.

A. Motivation to Combine

224. A POSITA would have found it obvious to combine Choe and Davies for at least five reasons. First, a POSITA would have been motivated to integrate Davies's conventional methods of transmitting annotated or adjusted images over a network into Choe's server-hosted calibration framework, enabling transmission of overlaid calibration views to remote devices. *See* IV.E. Second, a POSITA would have been motivated to leverage Davies's teachings on remote processing and GUI-based interaction to extend Choe's calibration loop beyond the vehicle, so that calibration factors recomputed from line modifications could be derived externally and returned to the system. *Cf.* Autoware on Board, *supra*, at 289–90 (describing separation of raw frames and overlay topics combined at a remote node). Third, a POSITA would have been motivated to ensure verification and consistency across annotators by transmitting the same overlaid horizon and lane images from Choe to remote operators, as Davies explicitly discloses. *See* EX1026, *supra*, at 20–21 (describing remote assistance paradigms for rider-only AVs). Fourth, a POSITA would have been motivated to expand Choe's human-in-the-loop calibration to include remote experts, as Davies teaches, to enhance oversight, error correction, and fleet-level quality control. *Cf.* EX1029, *supra*. Fifth, a POSITA would have been motivated to adopt Davies's disclosure of bandwidth-efficient transmission by sending lightweight overlay adjustments

rather than raw video streams, reducing network load while maintaining calibration accuracy. *See* Chen, *Edge Computing for Autonomous Driving*, *supra*, at 462–63.

1. How and Why it would have been obvious to integrate transmission and external calibration

225. In my opinion, it would have been obvious to a POSITA to combine Choe and Davies to teach the limitation “transmitting the overlaid image to a computing device over a network.” The rationale for this combination hinges on Choe’s clear teaching of remote calibration capability, coupled with Davies’ disclosure of routine methods for transmitting adjusted vehicle images over a network for remote viewing and user interaction. The combination directly addresses the need to move the visual feedback step (the “overlaid image”) of Choe’s calibration process out of the vehicle and onto a remote operator’s screen, using conventional networking means expressly taught by Davies.

226. **Choe Teaches the Need for Networked/Remote Calibration.** Choe discloses sensor calibration methods that include detecting a horizon line, overlaying it on an image (e.g., 2D view 411), and allowing a user to adjust the line. For example, “in response to a second input signal received from the input device (e.g., left or right arrow key), the position of at least one of the lane lines is modified ... a second calibration factor ... is determined based on the modification” EX1005, ¶[0017]). Similarly, a user may modify a detected horizon line which “update[s] the position of the horizon line to give a visual feedback to

the user” and the calibration module may then compute “[a] second calibration factor for calibrating ... the camera.” EX1005, ¶¶[0050], [0052]. These passages show that calibration is driven by interactive user adjustment of overlaid images.

227. While Figure 4 of Choe depicts these overlays on a local Display Device 405, Choe also discloses a fully networked environment: “[A]utonomous vehicle 101 that may be communicatively coupled to one or more servers 103–104 over a network 102,” where “[n]etwork 102 may be any type of networks such as a LAN, a WAN such as the Internet, a cellular network, [or] a satellite network” EX1005, ¶[0018].

228. Critically, Choe explains that “sensor calibration system 125 may be hosted by server 103 to calibrate sensors ... offline based on the images ... captured by the sensors” EX1005, ¶[0031] and that “the functionalities of sensor calibration module 308 may be maintained in a data analytics system such as server 103 ... [and] the calibration result can then be uploaded onto the vehicle” EX1005, ¶[0044]. These passages make clear that calibration tasks, including the manual modification of overlaid lines described in ¶¶[0017], [0052]–[0053], are not confined to local execution but can be carried out in a remote/server setting. A POSITA would have recognized that for such remote calibration to function, the annotated calibration views must be transmitted across the network so that the remote user can view and adjust them.

229. Davies Teaches Transmitting Adjusted Vehicle Images Over a Network. Davies fills any gap in Choe by disclosing routine mechanisms for transmitting adjusted vehicle images to external computing devices. Davies describes a system 100 with a server 101 and local computing apparatuses 104, 105, and 106 connected via a network 106 such as “the Internet, wide area network, and/or a local network” EX1007, ¶[0066]. The local apparatuses “may be any suitable computer apparatus including ... a cellular telephone [or] a portable tablet computer” EX1007, ¶[0067].

230. Davies further explains that “apparatuses 104, 105, and 106 may manipulate, share, transmit, and/or receive different data previously or currently produced at any one of the illustrated elements of the system 100” (Davies ¶[0070]). This expressly includes adjusted image data.

231. Davies also discloses offsite video/image transmission. In Figure 11, for example, “a camera 1116 captures 4K images onsite, e.g., at a field (shown generally at 1118)” and the 4K images are then transported by transport mechanism 1120 “to an operations base (‘OB’) (shown generally at 1122), e.g., a production truck away from the field 1118.” EX10070, ¶[0041]. At the operations base, a GUI may enable the operator to resize or zoom around the image. EX1007, ¶¶[0043]–[0044]. The processed output 1128 is then provided to downstream systems EX1007, ¶[0045]. Davies further teaches that portable tablet controllers

may serve as the GUI EX1007, ¶¶[0047], underscoring that annotated or adjusted images can be transmitted to and manipulated on remote computing devices.

214. Davies explicitly contemplates horizon and zoom adjustments as part of the transmitted and remotely manipulated imagery EX1007, ¶¶[0031], [0061]–[0063]. This shows that transmitting adjusted/annotated images to remote GUIs for operator interaction was well known and conventional.

232. A POSITA would have found it obvious to combine these teachings.

233. Choe requires remote execution. Choe expressly teaches server-hosted calibration EX1005, ¶¶[0031], [0044] and user modification of horizon line/lanes EX1005, ¶¶[0017], [0052]–[0053]. A POSITA would recognize that for the server-hosted calibration to support user interaction, the overlaid calibration views must be transmitted to the remote user's device. Davies provides the conventional solution. Davies discloses exactly how to transmit adjusted vehicle imagery, including horizon-adjusted views, over a network EX1007, ¶¶[0066]–[0070] to external computing devices, where a GUI enables operator viewing and interaction EX1007, ¶¶[0041]–[0043], [0047].

234. **Predictable result.** Combining Choe with Davies would simply apply known transmission techniques to Choe's calibration architecture, yielding the expected outcome: annotated calibration images displayed on a remote computing device so that a user can make horizon/line modifications. This combination would

have been routine, predictable, and motivated by efficiency, auditability, and operator oversight benefits already emphasized in both references.

235. A POSITA would have been motivated to integrate Davies's transmission framework into Choe's annotator system to improve scalability, robustness, and efficiency. Specifically: **Computational efficiency.** Remote servers can devote substantially greater computational power to calibration than the ADV's onboard system. Instead of relying on lightweight calculations performed locally, a remote server could employ more sophisticated optimization techniques (*e.g.*, gradient descent, probabilistic error modeling) to refine calibration parameters derived from user adjustments of the horizon and lane lines. **Quality assurance and consistency.** By transmitting user-modified line data from multiple vehicles, a centralized system could aggregate these adjustments to detect systematic errors or drifts, improving reliability across an entire fleet. Remote operators could also audit or validate local adjustments, ensuring that horizon and lane annotations are consistently applied across annotators and vehicles. *See* Sec. IV.G. **Fleet-level calibration.** Choe already contemplates and discloses server-hosted calibration systems and uploading calibration results back to the vehicle. EX1005, ¶¶[0031], [0044]. Integrating Davies's explicit teachings of transmitting line modifications and recalculated parameters makes this architecture directly applicable to fleet-wide calibration workflows. The result is a predictable benefit:

every vehicle in a fleet could receive the same updated calibration factors, improving consistency of perception models. **Reduced reliance on in-vehicle resources.** A POSITA would have understood that in-vehicle processors are constrained by power, thermal, and cost limitations. Offloading parameter recomputation to a remote server reduces this burden and frees the vehicle's hardware for other real-time perception and planning tasks.

2. How and Why it would have been obvious to enable remote display of overlaid calibration images

236. Choe teaches generating annotated calibration views, such as 2D view 411 with horizon and lane lines superimposed, and allowing users to modify those overlays locally through input devices. EX1005, ¶¶[0015], [0017], [0049]–[0053]. At the same time, Choe discloses that calibration functions may be hosted on a remote server 103 and results uploaded back to the vehicle. EX1005, ¶¶[0031], [0044]. A POSITA would have understood that for remote calibration to operate in the same manner as local calibration, the overlaid calibration images shown in the vehicle's visualizer must also be displayed to the remote operator.

237. Davies teaches precisely how to accomplish this, disclosing that adjusted or annotated vehicle images may be transmitted to remote computing devices, including servers, tablets, or mobile phones, over a network such as the Internet. EX1007, ¶¶[0066]–[0070]. For example, Davies explains that images captured onsite (*e.g.*, Fig. 11) may be transmitted to a remote operations base 1122,

where an operator manipulates them via a GUI 1114, and that portable devices such as tablets may serve as the interface. EX1007, ¶¶[0041]–[0043], [0047]. A POSITA would therefore have found it obvious to combine Choe’s annotated calibration overlays with Davies’s network transmission methods to enable remote operators to view and interact with the same images

3. How and Why it would have been obvious to complete the interactive calibration loop remotely

238. In Choe, calibration is interactive: when a user modifies the horizon line or lane line, the system recomputes calibration parameters such as pitch or yaw in real time. EX1005, ¶¶[0015]–[0017], [0052]–[0053]. However, Choe does not explicitly explain how this interactive loop would function in the disclosed server-hosted calibration scenario.

239. Davies fills this gap by teaching that remote computing devices can not only receive adjusted image data but also allow the operator to manipulate those images through a GUI and return modified information back to the system. EX1007, ¶¶[0066]–[0070]. For example, Davies discloses that remote apparatuses “may manipulate, share, transmit, and/or receive different data previously or currently produced at any one of the illustrated elements of the system 100.” *Id.* ¶[0070]. A POSITA would have recognized that integrating this capability with Choe’s server-hosted calibration naturally enables remote operators to adjust

overlaid horizon and lane lines, with the updated geometry returned over the network for use in recalculating calibration parameters.

4. How and Why it would have been obvious to extend human-in-the-loop calibration to remote operators

240. Choe already supports user-driven line modification locally, where a user adjusts the position of the horizon and lane lines through an input device such as a keyboard or joystick. EX1005, ¶¶ [0015], [0017], [0049]–[0053]. These manual adjustments directly update the overlays displayed in Choe’s visualizer and trigger recomputation of calibration parameters (*e.g.*, pitch, yaw, roll).

241. Extending this capability to remote operators, as Davies teaches, would have been an obvious and advantageous design choice. Davies expressly describes transmitting image data to an external computing device for review, refinement, and return to the vehicle. EX1007, ¶¶ [0059], [0063]. A POSITA would have recognized that integrating Davies’s approach into Choe’s visualizer framework allows expert annotators or supervisors located at a central facility to validate or correct line placement in real time. This is particularly beneficial in edge cases (*e.g.*, poor visibility, faded lane markings, complex intersections) where local annotators may struggle or where central teams can leverage additional contextual data.

242. By extending human-in-the-loop calibration beyond the vehicle to a distributed team of remote reviewers, the system gains robustness, quality control,

and scalability. *See* Sec. IV.H. A POSITA would have reasonably expected success in doing so, as both Choe and Davies rely on conventional, well-understood networking and annotation interfaces. Combining them merely shifts where the human annotation occurs (local vs. remote), while preserving the same established correction process.

5. How and Why it would have been obvious to optimize calibration through bandwidth-efficient overlay transmission

243. A POSITA would have been motivated by Davies’s teaching that image adjustment can be distributed between on-board and external processors specifically to conserve bandwidth. Davies explains that “some or all of image adjustment may be performed on the vehicle ... [and] [a]llocating processing power to the vehicle may be particularly useful, e.g., in wireless transmission applications where a reduced data package can take advantage of bandwidth limitations” EX1007, ¶¶[0059]. This disclosure confirms that rather than transmitting full raw video streams, a smaller data package, containing adjusted or annotated image information, may be sent to an external device.

244. A POSITA would have recognized that this approach is consistent with Choe’s existing networked calibration framework, which already contemplates bidirectional communication between the ADV and remote servers. EX1005, ¶¶[0018], [0025], [0031], [0044]. In practice, this means that instead of transmitting every frame of raw video to a remote server, the vehicle could

transmit an annotated image with horizon/line overlays or even lightweight coordinate data representing operator adjustments. *See* EX1022, *supra*. These incremental updates are negligible in size compared to the full image payload yet preserve all calibration context necessary for parameter recomputation. *See* Sec. IV.I.

245. Thus, adopting Davies’s reduced-data transmission model into Choe’s distributed calibration system would have been an efficient and predictable optimization. The expected benefits include reduced bandwidth consumption, lower latency, and faster feedback loops, all while maintaining calibration accuracy. Importantly, such an integration reflects nothing more than applying known compression and incremental update techniques to improve the performance of a calibration architecture that Choe already discloses as network-enabled.

6. Reasonable Expectation of Success

246. **Predictable improvement.** Choe’s calibration parameters are recomputed whenever a user modifies the position of a horizon or lane line, with the new calibration factor determined “based on the difference between the initial horizon line and the updated horizon line.” EX1005, ¶¶ [0015], [0049], [0052]–[0053], [0059]. Whether the updated overlay is adjusted by a user inside the vehicle (Choe) or by a remote user viewing the transmitted overlay (Choe, Davies),

the computational result is the same: recomputation of pitch, yaw, or roll. A POSITA would have recognized that integrating Davies's remote transmission and GUI interaction into Choe's calibration loop represents nothing more than a straightforward application of known elements, yielding the same predictable output.

247. **Known techniques applied to similar systems.** Choe already teaches generating overlaid calibration images, enabling user modification of those overlays, and recomputing calibration parameters based on the modification. Davies, in turn, teaches transmitting adjusted or annotated vehicle images to remote computing devices and enabling operator interaction through a GUI. EX1007, ¶¶[0041]–[0043], [0047], [0066]–[0070]. Combining these teachings simply allows Choe's overlay modifications to be carried out not only locally but also remotely, using conventional networking methods, thereby improving calibration systems in the same manner.

248. **Obvious to try.** A POSITA would have recognized that there are only a small number of predictable implementation choices: (i) keep user-driven calibration strictly local, as in Choe's Figure 4 embodiment, or (ii) enable user-driven calibration remotely via server 103 and a GUI-equipped device, as both Choes and Davies teaches. Both pathways use the same calibration workflow: overlaying horizon/lanes, receiving user modifications, and recomputing

parameters. Given Choe’s express disclosure of network-based calibration and Davies’s routine mechanisms for transmitting adjusted imagery, a POSITA would have had a reasonable expectation of success in combining them.

249. **No teaching away.** Nothing in Choe discourages extending calibration input to remote users. On the contrary, Choe expressly provides that “various modifications may be made [to embodiments] without departing from the broader spirit and scope of the disclosure.” EX1005, ¶[0066]. Likewise, Davies emphasizes flexibility in transmitting and manipulating calibration imagery across networks EX1007, ¶¶[0059]–[0063]. This reinforces that integrating remote annotation into Choe’s calibration framework would have been viewed as a routine design choice, not as a departure.

250. Accordingly, a POSITA would have been motivated to combine Choe and Davies, with a reasonable expectation of success, and the combination renders claims 1–20 obvious.

B. Independent Claim 1

251. As I explained above: Choe alone discloses or at least renders obvious, every step of claim 1 performed via a computing device communicating over a network, including elements 1[D]–1[G] (networked architecture and server-hosted calibration with upload/download of calibration artifacts; user-driven line adjustment and recalculation). To the extent Patent Owner contends that Choe

alone does not render obvious elements 1[D]-1[G], if additional corroboration were deemed useful, Davies expressly describes wireless transmission of adjusted/processed images and image data to and from external computing devices, which further confirms the obviousness of performing elements 1[D]-1[G] over a network.

1. [1Pre] -1[C], [1F] and [1G]

252. As I explained above, Choe expressly discloses elements 1[Pre]-[1C], [1F] and [1G]. *See* Sec. VIII.A.1-4, 7, 8

2. [1D]: “transmitting the overlaid image to a computing device over a network”

253. Choe in view of Davies renders this limitation obvious.

(a) Choe

254. Choe teaches a networked ADV architecture communicatively coupled to remote servers EX1005, ¶[0018], [0025]; the ability to transmit calibration images and point clouds for remote processing and to upload calibration results back to the vehicle EX1005, ¶[0031], [0044]; and the local generation and real-time updating of overlaid images with horizon and lane lines EX1005, ¶[0014]-[0017], [0049]-[0057], FIGs. 6A-6B. A POSITA would have found it obvious to combine these teachings by transmitting not just raw images but the already-generated overlaid images to a remote computing device over the disclosed network, since this is nothing more than using known elements for their established

functions. Doing so provides predictable and practical benefits: ensuring verification and consistency between local and remote calibration, improving efficiency by avoiding redundant processing, and supporting fleet-level analytics and human-in-the-loop troubleshooting. Whether rasterized as composite images or transmitted as raw frames plus overlay metadata, both obvious implementation options would have yielded the same result, delivery of overlaid images to a remote device within Choe's existing framework, with a reasonable expectation of success.

255. To the extent Patent Owner argues that Choe does not expressly disclose transmitting the overlaid image to a computing device, it would have been obvious in view of Davies.

(b) Davies

256. Davies discloses a vision-processing and calibration system in which image data captured by a vehicle-mounted camera may be transmitted to an external computing device for operator review and adjustment. For example, Davies describes capturing 4K images onsite and "transport[ing] the captured images to an operations base ('OB') ... e.g., a production truck away from the field" EX1007, ¶[0042], where an operator uses a GUI "to navigate the full raster 4K image and maneuver the selective ... extraction window" to pan, zoom, and select areas of interest EX1007, ¶[0044]. Davies further explains that "the output

... is provided to a router ... [and] can be taken live ... or ... ingested at a server ... for later playout” EX1007, ¶[0045], thereby allowing the adjusted image output to be routed back into the system for broadcast or further processing. In addition, Davies contemplates dividing processing tasks between the vehicle and remote servers to optimize bandwidth, noting that “some or all of image adjustment may be performed on the vehicle ... particularly useful ... in wireless transmission applications where a reduced data package can take advantage of bandwidth limitations” EX1007, ¶[0059].

(c) KSR

257. Thus, Davies provides explicit support for transmitting high-resolution images from a vehicle to remote operator interfaces, enabling review, adjustment, and return of processed outputs, making it a natural complement to Choe’s disclosure of overlay generation and server-based calibration.

258. A POSITA would have understood that combining this with Choe’s existing architecture, which already generates and updates overlaid images, EX1005, ¶¶[0014]–[0017], [0049]–[0057], and transmits calibration data to and from remote servers, EX1005, ¶¶[0018], [0025], [0031], [0044], would have been nothing more than the predictable use of known elements. The result is entirely foreseeable: remote servers or operators receive the same annotated calibration context already available in-vehicle, enabling consistency, efficiency, and human-

in-the-loop verification. Accordingly, even if Choe were not read to disclose this limitation, Davies makes clear that a POSITA would have been motivated, with a reasonable expectation of success, to implement remote transmission of overlaid images in Choe's system.

259. A POSITA would have been motivated to combine Choe's local overlay generation and networked calibration framework with Davies's explicit teaching of two-way image and data transmission for all the reasons I articulated at the outset. But to emphasize the following reasons: **(1) Verification & Consistency:** Choe already generates overlays EX1005, ¶¶[0014]–[0017], [0049]–[0057] and transmits calibration data remotely EX1005, ¶¶[0018], [0025], [0031], [0044]. Integrating Davies ensures that the very same overlaid images displayed locally can be shared with remote servers or operators, allowing them to “see what the vehicle sees” without recomputing overlays; **(2) Operational Efficiency:** Overlays compactly represent perception outputs and user-driven adjustments. Transmitting these overlays to a remote server avoids duplicative local computation and allows centralized handling of calibration, consistent with Davies's disclosure that “processing of the captured images may occur either offsite ... or onsite” and that the external processor and GUI “allow the operator to navigate ... and maneuver the selective ... extraction window” to generate modified image outputs EX1007, ¶¶[0042]–[0044]; **(3) Human-in-the-Loop**

Calibration: Davies teaches transmitting images to remote operators for adjustment and returning the modified outputs, with processing flexibly shared between vehicle and external devices EX1007, ¶¶[0042]–[0045], [0059], [0063]. Combined with Choe’s overlay UI, this allows remote technicians to work from the same calibration context as local users, ensuring consistent and efficient verification.

260. **Verification & Consistency:** Choe already generates overlaid images in real time by superimposing horizon and lane lines on camera frames and updating them with each user input. *See* EX1005, ¶¶[0014]–[0017], [0049]–[0057]. Choe also discloses transmitting calibration data over a network to remote servers for offline processing and then uploading calibration results back to the vehicle. *See* EX1005, ¶¶[0018], [0025], [0031], [0044]. Davies complements this by teaching that captured images may be transported outward to an offsite operations base for operator review and adjustment, with the modified outputs then routed back into the system EX1007, ¶¶[0042]–[0045], and further that image adjustment may be divided between on-vehicle and external processors “particularly useful ... in wireless transmission applications where a reduced data package can take advantage of bandwidth limitations” EX1007, ¶[0059]. These disclosures ensure that the same calibration views generated locally, whether overlays or modified image data, can also be transmitted and displayed remotely, allowing remote

technicians to work from the identical calibration context. This avoids requiring the server to regenerate overlays from raw imagery, reducing risk of misalignment or inconsistency. *See* Sec. IV.G; EX1026, *supra* at 20–21. In short, combining Choe with Davies yields consistent calibration feedback loops across local and remote environments.

261. **Operational Efficiency:** In Choe, overlaid images encode both perception outputs and user-driven adjustments. Transmitting these annotated frames, rather than only raw imagery, avoids requiring the remote server to re-detect horizon and lane lines, thereby reducing redundant computation and bandwidth. Davies teaches exactly this type of division of labor, explaining that “some or all of image adjustment may be performed on the vehicle ... particularly useful ... in wireless transmission applications where a reduced data package can take advantage of bandwidth limitations” EX1007, ¶[0059]. Davies further describes that captured images can be transported outward to a remote operations base for processing, with adjusted outputs returned into the system EX1007, ¶¶[0042]–[0045]. A POSITA would have recognized that sending Choe’s already-generated overlays instead of raw frames directly serves this purpose: it minimizes redundant work, accelerates calibration workflows, and improves responsiveness of remote or cloud-based calibration checks. This predictable efficiency benefit would have motivated a POSITA to combine Choe and Davies.

262. **Human-in-the-Loop Calibration:** Davies teaches that captured images may be transmitted outward to an external operations base, where an operator can manipulate the imagery through a GUI (e.g., panning, zooming, or extracting a selective window) to generate adjusted outputs EX1007, ¶¶[0042]–[0045]. Davies further explains that image processing tasks may be divided between the vehicle and external processors, “particularly useful ... in wireless transmission applications where a reduced data package can take advantage of bandwidth limitations” EX1007, ¶[0059], and that adjusted image outputs can be routed back into the system for continued use *Id.* These disclosures make clear that like Choe, Davies contemplates two-way interaction: images are transmitted outward, modified remotely, and returned to the vehicle. When integrated with Choe’s overlay-based calibration interface, a POSITA would have understood that remote technicians could directly view the same overlaid images used locally, adjust calibration lines in real time, and transmit the modified data back to update the vehicle’s parameters. This provides remote personnel with the identical calibration context seen by the local operator, eliminating ambiguity and enabling effective human-in-the-loop verification and correction. *See* Sec. IV.H.

3. [1E]: “receiving a modification of the line from the computing device, the modification comprising a new line at a second position”

263. Choe in view of Davies renders this limitation obvious.

(a) Choe

264. Choe expressly teaches modifying roadway features, including lane and horizon lines, once overlaid on an image. For example, Choe explains that “in response to a second input signal received from the input device (e.g., left or right arrow key), the position of at least one of the lane lines is modified ... [and] a second calibration factor ... is determined based on the modification” EX1005, ¶¶[0017]. Likewise, Choe describes that a user may modify a detected horizon line which “update[s] the position of the horizon line to give a visual feedback to the user” and the calibration module may then compute “[a] second calibration factor for calibrating ... the camera.” EX1005, ¶¶[0050],[0052]. These passages show that Choe’s calibration system already receives a modification of an existing line, producing a new line at a second position.

265. Choe also describes a networked calibration architecture in which calibration may be hosted remotely: “Sensor calibration system 125 may be hosted by server 103 to calibrate sensors ... offline based on the images and/or point clouds captured by the sensors” EX1005, ¶[0031]; and “the calibration result can then be uploaded onto the vehicle to be utilized online during the image processing

as a part of the perception process” EX1005, ¶[0044]. These disclosures make clear that calibration, including user-driven adjustments, can be performed on a remote computing device as well as locally.

266. To the extent Patent Owner argues that Choe does not expressly disclose receiving a modified line from a remote computing device, it would have been obvious in view of Davies.

(b) Davies

267. Davies complements Choe’s teachings by expressly showing how captured images can be transported to a remote operations base and adjusted by an operator through a GUI. Davies describes a GUI that allows an operator to “navigate the full raster 4K image and maneuver the selective ... extraction window” to adjust the position and size of a region of interest EX1007, ¶[0044], and explains that “processing of the captured images may occur either offsite ... or onsite” EX1007, ¶[0042]. The adjusted outputs are then routed back into the system for further use EX1007, ¶[0045]. Davies also emphasizes that processing can be flexibly divided between on-vehicle processors and remote servers, including across wireless networks, depending on bandwidth. EX1007, ¶[0059].

(c) KSR

268. A POSITA would have recognized that the same principle applies directly to Choe’s overlaid horizon and lane lines: once those annotated images are

transmitted outward for remote calibration (as explained in the 1D analysis), it would have been obvious and routine for the remote operator to modify the overlaid line using a GUI, and for that modified line to be received back by the calibration system. This is exactly what Davies demonstrates: user adjustments made remotely and reintegrated into the system.

269. Combining Choe and Davies yields no more than the predictable result: line-modification operations, already disclosed in Choe's local interface, are performed remotely using Davies's conventional GUI-based transmission and adjustment workflow. Technically, the modification is just metadata (new coordinates for the horizon or lane line), which can be transmitted with negligible overhead. Implementing this step relieves the vehicle of compute burden, enables centralized or fleet-wide calibration management, and allows human-in-the-loop validation.

270. A POSITA would have been motivated to combine Choe's overlay-based calibration system with Davies's explicit teaching of returning modified calibration data for all the reasons I articulated at the outset. And I emphasize the following reasons, similar to the ones I discussed for element 1[D].

271. **Verification & Consistency.** Choe already generates and updates overlays locally, EX1005, ¶¶[0014]–[0017], [0049]–[0057], while Davies teaches that captured images may be transmitted outward for operator adjustment via a

GUI and that the adjusted outputs are routed back into the system for further use EX1007, ¶¶[0042]–[0045], [0059]. Integrating these disclosures ensures that both local and remote calibration systems operate on the same updated line data, maintaining consistency and preventing divergence between local annotations and remote corrections, avoiding divergence.

272. **Operational Efficiency.** Allowing remote servers or operators to adjust the line and return the modification avoids repeating line detection locally or recalculating overlays from scratch. This reduces redundant computation and bandwidth, improving calibration throughput. Davies expressly contemplates external devices making such adjustments. EX1007, ¶[0059].

273. **Human-in-the-Loop Calibration.** Davies teaches that captured images may be transmitted to an offsite operations base where an operator, through a GUI, can adjust portions of the image, such as by navigating and modifying a selective extraction window, and that the adjusted outputs are then routed back into the system for further use EX1007, ¶¶[0042]–[0045]. Davies further explains that processing can be flexibly allocated between on-vehicle and external processors, with data transmitted across networks depending on bandwidth constraints EX1007, ¶[0059]. These teachings make clear that remote technicians can refine calibration outputs and return the modified data to the vehicle. When combined with Choe’s overlay UI and similar teaching of remote calibration, this provides

remote operators with the identical calibration context available in-vehicle, enabling unambiguous and effective human-in-the-loop feedback.

274. This combination would have been nothing more than the predictable use of known elements according to their established functions. Choe already teaches local user modification of horizon and lane lines and describes a networked calibration architecture in which calibration data is transmitted between the vehicle and a remote server EX1005, ¶¶[0017], [0031], [0044], [0052]–[0053]. Davies makes explicit that captured images may be transmitted outward for operator adjustment via a GUI, with the adjusted outputs then routed back into the system for continued use EX1007, ¶¶[0042]–[0045], [0059]. A POSITA would have recognized that combining these disclosures simply extends the same line-modification operation into the remote setting, yielding only the expected result.

275. A POSITA would have had a reasonable expectation of success in combining these teachings to achieve the claimed “receiving a modification of the line from the computing device, the modification comprising a new line at a second position.” First, Choe already proves the concept works locally. Choe discloses that line positions (horizon and lane lines) are modified in real time based on user input, and that calibration factors are recomputed from the updated line positions EX1005, ¶¶[0015], [0017], [0052]–[0053]. This establishes that once a “new line at a second position” is identified, the system readily updates calibration

parameters. No novel or unpredictable step is needed: Choe already demonstrates the entire loop locally.

276. Second, Davies makes explicit the outward and return communication of image-based adjustments. Davies teaches that captured images may be transmitted to an offsite operator for modification via a GUI, and that the adjusted outputs are then routed back into the system for further use EX1007, ¶¶[0042]–[0045], [0059]. This shows there is nothing speculative about external modification or return of calibration data; Davies expressly contemplates it, including across wireless channels. Third, the combination involves routine, well-understood implementation choices. The vehicle in Choe is already network-enabled EX1005, ¶¶[0018], [0025], and overlay data is already generated for calibration EX1005, ¶¶[0014]–[0017], [0049]–[0057]. Extending Choe’s communication framework to send and receive line modifications would have been obvious as a routine application of known techniques to improve performance in the same field. Engineers were already familiar with transmitting annotation or overlay data efficiently in other image-processing systems, including Davies’s disclosure of transmitting operator-modified image outputs back into the system EX1007, ¶¶[0042]–[0045], [0059]. A POSITA would have recognized and found it obvious that the same approach could be applied to Choe’s calibration overlays, since transmitting lightweight line adjustments rather than regenerating them remotely

improves bandwidth use and ensures consistency across local and remote workflows. In this way, the combination merely applies a well-understood communication technique to Choe's system, yielding the expected benefit without any inventive step.

C. Claims 2-3 and 5-7

277. Claims 2–3 and 5-7 are also obvious in view of Choe in combination with Davies for the same reasons already discussed for Ground 1, with Davies providing further reinforcement where Patent Owner may allege a gap regarding any transmission to/from the remote computer or remote processing. Each of these dependent claims simply refine the basic calibration method of claim 1 in ways already taught or rendered obvious by Choe, and in any event obvious to a POSITA in view of Davies's explicit two-way transmission/adjustment framework.

278. **Claim 2** specifies that the computed camera parameter comprises camera height, viewing angle, or road plane normal. As discussed above for Grounds 1, Choe already computes pitch, yaw, and roll angles EX1005, ¶¶ [0015], [0017], [0049], [0052], [0053], which directly define the viewing angle and, by geometric relationship, the effective height of the camera relative to the road plane. *See* Sec. VIII.B. Thus, for all the reasons I articulated above in Ground 1, Choe in view of Davies also renders obvious claim 2.

279. **Claim 3** recites that the roadway image is received while the vehicle is moving. For all the reasons I discussed above in Ground 1, Choe discloses exactly this, teaching that “[t]he above processes can be utilized online while the vehicle is driving and real-time images are captured to calibrate the sensors in real-time” EX1005, ¶[0054]. Choe also emphasizes continuous perception and planning while “autonomous vehicle 101 is moving along the route” EX1005, ¶[0028]. Thus, for all the reasons I articulated above in Ground 1, Choe in view of Davies also renders obvious claim 2.

280. **Claim 5** specifies that the overlaid image is transmitted to a web-based application. As discussed above for Ground 1, Choe already contemplates this very framework. It teaches that calibration images, results, and associated analytics may be transmitted to remote servers over a variety of networks, including LAN, WAN, the Internet, cellular, and satellite connections EX1005, ¶¶[0018], [0025], [0031], [0044]. While Choe does not explicitly identify a browser-based or web-hosted interface, it establishes the same client–server data-exchange foundation that underlies a web-based application. These disclosures directly align with how a web-based application would be accessed in practice: through Internet-connected servers hosting an application interface. A POSITA would have recognized that, given Choe’s reliance on Internet-based communication with remote calibration servers, implementing the remote

visualization and review component as a browser-accessible web interface would have been a straightforward design choice enabling access without proprietary client software Choe therefore establishes that calibration data can be exported from the vehicle environment and made available over the same network pathways used by web-based platforms.

281. Davies explicitly discloses remote capture, transmission, and GUI editing. Specifically, it teaches that images captured on the vehicle can be transmitted to an offsite operations base, where an operator manipulates the imagery through a graphical user interface (GUI). *See* EX1007, ¶¶[0042]–[0044]. This already establishes the basic building block of remote human-in-the-loop calibration, exactly the concept a POSITA would equate with a client/server or web-based setup. Although Davies refers to a workstation GUI rather than a browser interface, the underlying architecture, a remote client interacting with a server to receive and edit calibration imagery, is the same technical paradigm implemented by web applications. A POSITA would therefore have recognized that deploying such a GUI within a browser would merely substitute a known interface technology for the same client-server workflow

282. Davies further teaches distributed processing depending on bandwidth. It explains that the calibration or image-adjustment workload can be allocated between the vehicle and remote processors, with data exchanged over

networks. *See* EX1007, ¶[0059]. This disclosure shows that Davies was not limited to one embodiment but contemplated routine engineering trade-offs, the same design choices a POSITA would consider in deciding whether to rasterize overlays locally versus transmitting raw frames with overlay metadata, techniques already described in AV frameworks such as Autoware and edge-computing literature. Such bandwidth-dependent workload partitioning is precisely the type of optimization that web-based architectures facilitate, as browser-based clients can dynamically request imagery or metadata from centralized servers depending on available network conditions. A POSITA implementing Choe's Internet-connected calibration system would have seen this as compatible with a web-based interface capable of rendering overlays from streamed data

283. Davies also discloses the bidirectional exchange of adjusted calibration data. After operator edits, the modified calibration parameters are transmitted back to the vehicle for continued use. *See* EX1007, ¶[0063]. This closes the loop — image capture → remote GUI adjustment → recalculated calibration → retransmission to the vehicle — which mirrors the functional architecture of a web-based calibration tool where a client connects to a backend that receives images, hosts the editing UI, and returns updated parameters. A POSITA would thus have recognized that implementing this same bidirectional exchange through a web interface would streamline deployment, allowing multiple

operators to access the calibration UI remotely through standard browsers, consistent with the distributed calibration workflow already described in Davies

284. Taken together, these teachings demonstrate that Davies is not describing a static or one-way image transfer, but a bidirectional, network-based calibration workflow where imagery and annotations are exchanged between the vehicle and external computing resources. To a POSITA, this architecture would have been functionally similar to, and readily implemented as, a web-based calibration application, since both (i) host the calibration UI remotely, (ii) exchange images and annotations over a network, and (iii) return updated calibration factors back into the vehicle's system. In other words, Davies anticipates the very same system design choices, remote GUI, client/server flow, and bandwidth trade-offs, that define a web-based calibration application

285. A POSITA would have readily understood that employing a web-based application for this purpose is a routine and obvious implementation of the disclosed architectures. By the time of the invention, Internet-hosted applications were the standard mechanism for enabling distributed access to remote computing resources. Leveraging such a platform would predictably provide well-known benefits, including accessibility from any location, the ability to scale processing across fleets, and efficient collaboration among multiple users. No inventive leap is required to apply Choe's transmission of calibration data over the Internet to the

web-based interface paradigm expressly illustrated in Davies. Instead, it represents the straightforward use of conventional web technologies to implement a function that both references already describe sending calibration images out of the vehicle, receiving operator modifications, and returning the updated calibration data back into the system.

286. **Claim 6** requires that the modification of the line from the computing device comprises receiving an identification of a new horizon line by a user. As discussed above for Ground 1, Choe discloses that its calibration interface superimposes horizon and lane lines over captured frames, and the user can incrementally adjust those lines through input devices such as keys, joystick, or voice, EX1005, ¶¶[0015], [0017], [0049], [0052]. In Choe, those user-provided adjustments directly update the overlaid image and feed into the calculation of calibration parameters. Davies complements this by teaching that captured images may be transmitted to an external computing device where an operator, using a GUI, manipulates portions of the image (e.g., via a selective extraction window) and that the adjusted outputs are then routed back into the system for further use EX1007, ¶¶[0042]–[0045], [0063]. Davies further explains that processing can be flexibly divided between on-vehicle and external processors depending on bandwidth EX1007, ¶[0059]. A POSITA would have readily recognized that combining these disclosures yields exactly the claimed feature. In particular, it

would have been obvious to implement the same line-adjustment capability through a remote GUI, enabling an external user to identify a new horizon line, transmit that identification over the network, and have the vehicle update its calibration parameters accordingly. This combination represents nothing more than applying the predictable extension of Choe's local adjustment mechanism to the remote processing framework taught by Davies, achieving the well-understood benefit of allowing calibration to occur outside the vehicle while preserving the same functional input/output relationship.

287. **Claim 7** recites recomputing the camera parameter based on the modification of the line. As discussed above for Ground 1, Choe squarely teaches this process: its calibration system relies on the superimposed horizon and lane lines as reference features, and when those lines are adjusted by the user, the system recalculates the underlying calibration factors to correct for pitch, yaw, or roll of the camera EX1005, ¶¶ [0015], [0017], [0049], [0052], [0053], [0059]. For example, Choe explains that user-driven movement of the horizon line directly results in recalculation of the pitch factor, while lane line adjustments yield updated yaw or roll corrections.

288. Davies reinforces and extends this teaching by making explicit that calibration and image adjustment are not confined to in-vehicle computation. Davies discloses that captured images can be transmitted outward to an offsite

operations base, where an operator uses a GUI to adjust the image EX1007, ¶¶[0042]–[0045], and that the adjusted output is then routed back into the system for continued use EX1007, ¶[0045]. Davies further explains that some or all of the image adjustment may be performed either on-vehicle or externally, particularly in bandwidth-limited scenarios, with processing tasks flexibly allocated between the vehicle and remote processors EX1007, ¶[0059]. Thus, Choe in view of Davies therefore makes clear that recomputation or refinement of calibration-related outputs can occur remotely, with the resulting adjusted data transmitted back into the system.

289. A POSITA would have had a reasonable expectation of success in implementing this functionality because the mechanism is straightforward and entirely predictable. Choe already establishes the direct functional link between line adjustments and parameter recomputation. Davies simply applies that same link in a distributed setting, where external servers or user interfaces participate in the process. The combination yields the claimed feature in a natural way: line modifications (local or remote) trigger the recomputation of camera parameters, and those updated parameters are then applied to improve system accuracy. This represents a routine, well-understood design choice to extend Choe's core recalibration logic into the external communication framework described by

Davies, producing no unexpected results but rather the anticipated benefit of maintaining accurate camera calibration across both local and remote workflows.

D. Claims 8-10, 12-16, and 18-20.

290. For the same reasons I discussed above for Grounds 1 (see Sec. VIII.H) and claims 1-3 and 5-7 here (see Sec. IX.A-C), claims 8-10, 12-16, and 18-20 are obvious over Choe in view of Davies.

X. GROUNDS 3A/3B: CHOE IN VIEW OF KUEHNLE (GROUND 3A) AND CHOE IN VIEW OF DAVIES AND FURTHER IN VIEW OF KUEHNLE (GROUND 3B) RENDER OBVIOUS CLAIMS 4, 11, AND 17.

291. Grounds 1 and 2 already demonstrated that the independent claims are obvious in light of Choe alone (Ground 1) and Choe in view of Davies (Ground 2). Building on that foundation, Grounds 3A and 3B show that dependent claims 4, 11, and 17 are likewise unpatentable. These claims add the additional limitation that “receiving the image of the roadway while the vehicle is moving comprises detecting that the vehicle is traveling above a pre-defined speed or for a pre-defined duration.” Claim 4 is dependent upon claim 3, and similarly claims 11 and 17 depend upon claims 10 and 16. In my opinion, to meet the limitations of claim 3, 10, and 16, one would necessarily have to also meet the limitations of claim 4. Further, as I explain below, this additional condition was well known in the art, and a POSA would have found it obvious to incorporate into Choe (Ground 3A), or

into Choe as further modified by Davies (Ground 3B), in view of Kuehnle's express teaching of speed and time gating for calibration.

292. **Choe's Online Calibration and Telemetry.** Choe teaches online calibration "while the vehicle is driving and real-time images are captured." *See* EX1005 ¶¶[0017]. During driving, Choe obtains "real-time ... data detected ... by sensor system 115," which includes telemetry from vehicle sensors. *See id.* ¶¶[0028]. Choe also makes clear that vehicle dynamics such as "speeds, accelerations, [and] decelerations" are logged in association with calibration data. *See id.* ¶¶[0029]. These passages show that Choe already contemplates capturing roadway images and using contemporaneous telemetry (including speed) during calibration. However, Choe does not explicitly disclose tying calibration initiation to a pre-defined minimum speed or elapsed time threshold. Instead, Choe assumes the vehicle is in motion and relies on captured data streams without specifying conditions for when recalibration should occur.

293. **Davies's Use of Telemetry and Multi-Frame Refinement.** Davies extends these concepts by expressly teaching that calibration and image adjustment depend on vehicle dynamics. Davies discloses that telemetry such as "gyro data, vehicle angle, attitude, altitude, speed, acceleration, traction, [and] navigational data" may be used in calibration processes. *See* EX1007, ¶¶[0044]. It also emphasizes that calibration is not based on a single image but rather involves

“receiving image data from a vehicle mounted camera ... [and] receiving data from at least one vehicle mounted sensor” with horizon refinement performed over multiple frames. *See* EX1007, ¶¶[0042], [0045]. While Davies expressly incorporates vehicle speed and other dynamics into the calibration loop, it does not explicitly condition calibration on a minimum speed or elapsed duration. Instead, Davies leaves the precise trigger criteria to implementation, describing in general terms that networked calibration can be distributed between vehicle and server processors depending on available bandwidth. *See id.* ¶[0059]. Davies therefore reinforces the use of telemetry but does not provide the explicit safeguards of speed and time thresholds.

294. Kuenhnle’s Explicit Teaching of Speed and Time Gating:

Kuenhnle squarely fills this gap. Kuenhnle discloses an online calibration method for vehicle-mounted video systems in which pitch and yaw of the camera are deduced from vanishing points in roadway images. In doing so, Kuenhnle emphasizes that calibration must be gated by both speed and time. Kuenhnle states that “the conditions for extrapolation ... require that the vehicle is moving with at least a certain speed, so that low speed maneuvering, with its possible large yaw angle, is not taking place.” *See* EX1010, p. 11-12. Kuenhnle further explains that the calibration circuit evaluates “the current speed data of the vehicle.” *See id.*, p. 6.

295. In addition, Kuenhnle discloses a temporal requirement by filtering across multiple frames. It states that “a long term average vanishing point location is calculated with time-filtering methods from a sequence of images,” and that “the static yaw and pitch angle of the camera is deduced” from this averaged data. *See* EX1010, p. 4, 20. In one embodiment, Kuenhnle applies a recursive averaging filter in which the “weighting value that decreases with time (as in a recursive averaging filter),” such that each new frame contributes proportionally less to the overall calibration result. *See* EX1010, p. 13. These teachings are precisely the claim limitations: calibration only when the vehicle is traveling above a pre-defined speed or for a pre-defined duration

296. **Motivation to Combine:** A POSA would have been motivated to incorporate Kuenhnle’s speed and time conditions into Choe’s calibration framework (Ground 3A), or into Choe as combined with Davies (Ground 3B), for multiple reasons.

297. First, speed and time gating directly address the same risks already identified in Choe and Davies: spurious calibration updates when the vehicle is stopped, parking, maneuvering at low speeds, or subject to transient states. Kuenhnle’s safeguards, minimum speed and duration thresholds, ensure that recalibration occurs only under stable and representative conditions.

298. Second, Choe already supplies the telemetry signals (speed, acceleration, elapsed time) necessary to implement Kuenhnle's conditions. *See* EX1005, ¶¶[0028]–[0029]. Thus, incorporating Kuenhnle's approach requires no new hardware or sensing capability, only the application of a known gating condition to signals already disclosed.

299. Third, Davies reinforces the rationale for incorporating Kuenhnle's teachings. Davies already discloses the use of speed, acceleration, and other dynamics to influence calibration, and it teaches multi-frame refinement of horizon lines. *See* EX1007, ¶¶[0044]–[0045]. A POSA would naturally apply Kuenhnle's speed and time thresholds in that multi-frame context to ensure stability, filtering out transient noise and improving consistency across calibration updates.

300. This is a classic example of a combination of familiar elements according to their established functions. Speed gating and temporal averaging were finite, predictable design choices, well-known in the field, and their application in the context of online camera calibration would have been straightforward. A POSA would have expected success in combining these teachings because each reference is directed to the same technical problem (camera calibration in moving vehicles), and Kuenhnle's explicit thresholds directly address the risks that Choe and Davies implicitly recognized. The result would be predictable: recalibration triggered only

when the vehicle is moving at a representative speed and for a sufficient duration to yield stable horizon and lane measurements.

301. Accordingly, claims 4, 11, and 17 would have been obvious in view of Choe in view of Kuenhnle (Ground 3A) or in view of Choe in combination with Davies and further in view of Kuenhnle (Ground 3B). The combination merely applies known elements in their expected manner, using Kuenhnle's explicit safeguards to improve the robustness of calibration already disclosed in Choe and Davies.

XI. GROUND 4: WESTMACOT IN VIEW OF TAL RENDERS OBVIOUS CLAIMS 1-20.

302. In my opinion, claims 1-20 would have been obvious to a POSITA in view of Westmacot in combination with Tal.

A. Westmacot

303. As discussed in Section IV.E.3, Westmacot describes a road annotation and calibration system for autonomous vehicles that already discloses many elements of claims 1-20. Westmacot explains that roadway images are captured by a camera device installed within a travelling vehicle (*see* EX1008, pp. 1-3, 7; Figs. 3, 7, 15, 30–32). Westmacot further teaches that the system computes geometric information from those images, including a horizon line, and overlays annotations such as lane lines on the images for display to an annotator (*e.g.*, annotation interface UI 412; Figs. 6, 25; Fig. 16 and related text).

304. Westmacot also discloses that the annotation interface allows a user to modify lane lines by shifting boundaries or adding new lines, and that those changes propagate through the 3D road model via the model adaptation component 410. Finally, Westmacot explains that its parameter computation component 408 calculates camera-related parameters such as height (H), offset (S), forward point, horizon line, lane width (W), and center position (C), and that these values are updated when lane boundaries are modified. These parameters are then shared internally among system components (*e.g.*, model adaptation 410, rendering 416).

B. Tal

305. Tal discloses network-based image processing systems, expressly teaching that both raw and processed images, including images with overlays, may be transmitted to remote servers over a communications network. For example, Tal explains that “only selected data portions 20 (*e.g.* image frames 16a, b, c, d) of the images 16 can be transmitted over the network 18 by the device 101 to a server 107 ... Alternatively, unprocessed image(s) 16 can be transmitted by the device 101 over the network 18, as they are acquired in real time” EX1009, ¶[0027] Tal further teaches that the device may connect via wireless LAN or Wi-Fi “to upload a large volume of data without the cost of cellular data usage” EX1009, ¶[0064], or via cellular interfaces for constant or periodic communication with the server EX1009, ¶[0065]. The server 107 in turn “is responsible for the organizing,

storing, processing and disseminating of the object data 21 uploaded by the device(s) 101 ... [and] can store image data 16, sensor data 17, object data 21, resultant processed data 20 ...” EX1009, ¶¶[0066]. Critically, Tal describes a distributed processing framework (FIG. 4), in which “a first portion 402 of the image 16 processing can be performed on the device 101 itself ... while a second portion 404 of the image 16 processing can be performed on the server 107 post receipt of the object data 21” EX1009, ¶¶[0050]–[0051]. Tal also makes clear that transmitted images can include overlays. In terms of format, Tal explains that a digital image 16 can represent raw image data (e.g., YUV, RGB, HSL, HSV) with pixel values or vector representations, and that processed data “can be overlaid on the image in a vector graphics or a separate raster image and flattened/merged into the image 17 itself to display the detection 12 in the image 16. The data 21 may be overlaid/incorporated into the image data 20 or stored and sent separately, with association to the respective image 16” EX1009, ¶[0052].

306. This confirms that image and calibration data may be exchanged back and forth between vehicles and servers to support storage, analysis, and reporting.

307. Thus, Tal provides the potentially missing elements from Westmacot: (i) transmitting overlaid images across a network, (ii) implementing modification and processing steps at a remote computing device, and (iii) transmitting calibration results or parameters back to the vehicle.

308. **Complementary Teachings.** Westmacot does disclose that its computed outputs are shared across components within the local system, but it is silent on transmitting those outputs externally. For example, the parameter computation component 408 calculates camera parameters such as height, offset, horizon line, lane width, and center position, and those values are provided to other internal modules such as the model adaptation component 410 and rendering component 416 to update the 3D road model and generate annotated displays for the annotation interface UI 412. *See* EX1008, pp. 32-35; Figs. 6, 25. Similarly, user modifications entered through UI 412 are propagated through the local road model. Thus, Westmacot clearly contemplates internal distribution of annotated images and parameters, but it does not expressly disclose transmitting those artifacts outside the local architecture to remote servers or distributed users. However, given the system's goal of generating large numbers of training images efficiently (implying centralized processing and annotation), remote access to the UI is strongly implied. However, Westmacot does not explicitly state that the overlaid image itself is transmitted over a network to a separate computing device for viewing and modification. That omission is directly addressed by Tal, which teaches transmitting raw and processed image data, including images with overlays, to and from external computing devices via wireless or cellular networks. *See* EX1009, ¶¶[0027], [0050-0052], [0064-0066].

C. Motivation to Combine

309. A POSITA would have found it obvious to combine Westmacot and Tal for at least three reasons. First, a POSITA would have been motivated to leverage Tal's networked transmission framework to distribute Westmacot's annotated roadway images and computed camera parameters to remote servers, enabling scalable training, validation, and fleet-wide calibration. Second, a POSITA would have been motivated to integrate Tal's remote processing capabilities with Westmacot's annotation interface, allowing line modifications (e.g., new or adjusted lane and horizon lines) to be performed collaboratively by distributed annotators, thereby improving throughput and dataset consistency. Third, a POSITA would have been motivated to apply Tal's bidirectional communication model so that parameters computed in Westmacot, such as height, offset, lane width, and horizon line, could be transmitted back to the vehicle's camera device, closing the calibration loop and improving accuracy.

310. A POSITA would have understood that one of the most compelling reasons to distribute Westmacot's computed outputs externally is to enable scalable, centralized training, validation, and calibration of autonomous driving systems. Multiple patents preceding the priority date of the '276 patent explicitly support this point and form corroborating evidence of my analysis. For example, U.S. Patent No. 9,916,703 B2 (2018) discloses a backend AV management system

with an offline calibration module that ingests logged vehicle data, performs calibration remotely, and provides results back to vehicles, demonstrating centralized calibration and teleoperator involvement. U.S. Patent Application Publication No. 2017/0010106 A1 (2017) and its related U.S. Patent No. 9,665,100 B2 (2017) describe vehicles uploading observations to a server, where data are aggregated into a sparse map and redistributed to vehicles: evidence of the same “upload–aggregate–redistribute” architecture applied to perception artifacts. Likewise, U.S. Patent Application Publication No. 2016/0214533 A1 (2016) discloses an image server that receives uploaded sensor images and generates calibration information off-vehicle, expressly showing remote calibration computation. Finally, U.S. Patent No. 8,612,136 B2 (2013) teaches central servers aggregating roadway data from multiple devices to create validated maps distributed back to users. Taken together, these patents confirm that before the priority date it was routine to offload vehicle-captured outputs to centralized servers for processing and then push the validated results back to vehicles. A POSITA would have recognized that applying this well-established paradigm (as disclosed in Tal) to Westmacot would have been an obvious, predictable extension, yielding scalable and consistent calibration across a fleets

311. Westmacot discloses generating precisely the types of data that are critical for machine-learning pipelines: annotated roadway images with overlaid

horizon and lane lines (UI 412), EX1008 Figs. 6, 25, and computed camera parameters such as height, offset, horizon line, lane width, and center position (parameter computation component 408). These outputs are essential for training and refining perception models, yet Westmacot limits their use to distribution/transmission between internal system components (e.g., model adaptation 410, rendering 416). Tal directly addresses this omission by teaching that both raw and processed image data, including images with overlays, may be transmitted via wireless or cellular networks to remote servers for analysis, storage, and further processing. EX1009, ¶¶[0027], [0050-0052], [0064-0066], [0075], [0138]. A POSITA would have recognized that combining Tal's transmission capabilities with Westmacot's annotated imagery and calibration parameters would allow large volumes of this high-value data to be aggregated from many vehicles or annotation stations into central servers, thereby improving the scalability of training datasets, supporting distributed annotation workflows, and enabling fleet-wide calibration consistency. This is a straightforward application of known networking techniques to data that Westmacot already generates, yielding predictable and highly valuable results.

1. How and Why it would have been obvious to transmit Westmacot's annotated images and parameters to remote servers for scalable training and calibration

312. By the relevant priority date, centralized machine learning and dataset aggregation were well-established practices in the autonomous driving industry. Systems routinely transmitted large volumes of annotated roadway imagery and calibration data from distributed vehicles or annotation tools to central servers to train perception models at scale. Westmacot's disclosure of generating annotated/overlaid images and updating camera parameters provides precisely the type of ground-truth data that was conventionally shared for such training pipelines. Tal reinforces this by explicitly teaching that both raw and processed images, including those with superimposed object data, can be transmitted across networks for analysis and storage. EX1009, ¶[0052]. A POSITA would have understood that combining Westmacot's annotated imagery with Tal's network architecture yields a scalable, fleet-wide solution: local vehicles or annotators generate the overlays and parameters, and those artifacts are transmitted to central servers for aggregation, validation, and model training. This combination was a natural and predictable step aligned with industry practice.

2. How and Why it would have been obvious to integrate Tal's remote processing with Westmacot's annotation interface for collaborative line modification

313. Westmacot emphasizes that its annotation interface (UI 412) allows a user to adjust lane lines, add new lines, and thereby refine the underlying 3D road model. *See* EX1008, p. 35. Tal, in turn, teaches that images and overlays can be transmitted over networks for processing on remote servers. EX1009, ¶¶[0027], [0030]. A POSITA would have recognized that integrating Tal's network functionality with Westmacot's annotation interface allows line-modification workflows to occur remotely. This enables multiple annotators or supervisors to collaborate in real time, improves annotation throughput, and ensures consistency across datasets. By distributing Westmacot's annotation interface via Tal's network, annotators in different locations could access the same imagery, propose or approve modifications, and update shared calibration models. This distributed approach was entirely conventional by the priority date, consistent with Tal's architecture, and predictably improved the scalability and quality of annotation pipelines.

3. How and Why it would have been obvious to apply Tal's bidirectional communication model to transmit recalculated parameters back to the vehicle's camera device

314. Westmacot discloses computing key camera-related parameters such as height, offset, horizon line, lane width, and center position, which are updated

when line boundaries are modified. *See* EX1008, pp. 32-33. These values are essential for calibrating the camera and associated perception modules. A POSITA would have recognized that transmitting these parameters back to the vehicle's camera device (or embedding them in the captured imagery as metadata) closes the calibration loop, enabling real-time adjustment and improved accuracy. Tal confirms that such bidirectional communication was conventional: "processed image data" may be transmitted across networks, and the modified outputs returned. *See* EX1009, ¶¶[0027], [0030], [0052], [0075], [0138]. This disclosure squarely supports transmitting calibration outputs, whether parameters, overlays, or metadata, back to the source system. Combining Tal with Westmacot therefore yields a system where parameters computed in Westmacot are fed back to the vehicle via Tal's communication channels, a straightforward extension offering clear utility and predictability.

4. Reasonable Expectation of Success

315. **Predictable Use of Prior Art Elements.** Both Westmacot and Tal operate in the same technological domain: software-based image processing for autonomous vehicles, leveraging sensors, annotations, and networking.

Westmacot's contribution lies in generating annotated imagery, supporting user-driven line modifications, and computing calibration parameters. Tal's contribution lies in transmitting such imagery and data across networks for remote processing,

analysis, and return. Applying Tal’s teachings to Westmacot’s data is precisely the type of “predictable use of prior art elements according to their established functions.” Nothing unexpected arises from the combination: annotated images and parameters generated in Westmacot are simply shared using Tal’s well-known communication methods, yielding remote review, storage, and calibration feedback. A POSITA would have had every reason to make this combination and every expectation of success.

D. Independent Claim 1

1. [1Pre] “A method comprising”

316. Westmacot explains that its system is designed for use in autonomous or semi-autonomous driving, where images of a road scene are captured by one or more cameras mounted on a vehicle and then analyzed for lane and road modeling. *See, e.g.*, EX1008, pp. 1-3, 7, and Figs. 3, 7, 15, 30–32. These disclosures establish that the invention operates in the context of images recorded by a camera device installed within a vehicle. Moreover, Westmacot presents its invention as a method of annotating roadway images and updating camera calibration parameters based on user modifications of overlaid lines, which frames the steps of claim 1.

2. [1A] “receiving an image of a roadway recorded by a camera device installed within a vehicle”

317. Westmacot discloses this element.

318. Westmacot teaches that roadway images are captured by a camera device installed within a travelling vehicle. For example, the background explains:

“Such networks require large numbers of training images. These training images are like the images that will be seen from cameras in the autonomous vehicle, but they have been annotated with the information that the neural network is required to learn. For example, they will have annotation that marks which pixels on the image are the road surface and/or which pixels of the image belong to lanes.”

EX1008, p.1.

319. Westmacot further discloses that “a first aspect of the invention provides a method of annotating road images ... captured by an image capture device of [a] travelling vehicle.” Westmacot, pp. 3, 1, 14, 68. Similarly, “a second aspect of the invention provides a method of measuring a relative angular orientation of an image capture device within a vehicle ... receiving a time sequence of two dimensional images as captured by an image capture device of a travelling vehicle.” EX1008, p. 7. In another embodiment, “receiving a time sequence of two dimensional images as captured by an image capture device of travelling vehicle; processing the images to reconstruct, in three-dimensional space, a path travelled by the vehicle.” EX1008, p. 3.

320. Westmacot also provides a concrete example of such image capture: “videos and associated GPS data were captured with a standard Nextbase 402G

Professional dashcam ... mounted on the inside of the car windscreen, roughly along the centre line of the vehicle and approximately aligned with the axis of motion.”EX1008, p. 22. And Figure 3 illustrates “a vehicle 300 that can be used to capture road images to be annotated... The training vehicle 300 is shown to comprise an image capture device 302... The processor 304 receives the captured images from the image capture device 302, and stores them in a memory 306, from which they can be retrieved for use.” EX1008, pp. 22-23. *See also* EX1008, Figs. 7, 15, 30–32 (depicting captured frames and corresponding camera locations).

321. Accordingly, a POSITA would have understood that Westmacot expressly discloses “receiving an image of a roadway recorded by a camera device installed within a vehicle,” as required by claim element [1A].

3. [1B] “detecting a horizon line in the image”

322. Westmacot discloses this element.

323. Westmacot explicitly teaches that in response to roadway images captured by a vehicle-mounted camera, the system determines a horizon line in the image. For example, the specification explains:

“The plane that is perpendicular to this road surface normal vector can be intercepted with the image plane and this will provide a line across the image that would match the horizon if the car were driving along a perfectly straight and level road (the ‘horizon line’ as that term is used herein).”

EX1008, p. 44, Fig. 16.

324. Similarly, Westmacot explains how camera orientation is derived from roadway geometry:

“This is illustrated in Figure 15. In other words, this exploits the observation that the vector difference $x_{t+1} - x_{t-1}$ lies (approximately) parallel to the vehicle’s longitudinal axis a_L when the vehicle 300 is travelling in an (approximately) straight line, in order to estimate the angular offset of the camera 302 relative to the vehicle 300.”

EX1008, p. 41

325. Estimating angular offset is equivalent to detecting the horizon line, because the horizon line provides the reference for the pitch orientation of the camera.

326. Westmacot also teaches determining the 3D location of the camera for each image:

“...For each of the captured images $img(n-1)$, $img(n)$, $img(n+1)$, a 3D location of the camera 302 at the time that image was captured...is estimated...”

EX1008, p. 37

327. Detecting the horizon line is necessary to align image frames with real-world geometry, since the horizon defines how the 2D image maps to world coordinates.

328. Finally, Westmacot notes that these calculations enable automatic lane annotation:

“...By exploiting all of the above, it is possible to fully automatically generate annotation of the lane driven by the car in all images of the video, using only images captured from a low-cost and un-calibrated image capture device.”

EX1008, p. 46

329. A POSITA would have understood that generating lane annotations requires normalization against the horizon line across frames, confirming that horizon detection occurs in the process.

330. Accordingly, Westmacot expressly discloses “detecting a horizon line in the image,” as required by claim element [1B].

4. [1C] “overlaying a line on the image to generate an overlaid image”

331. Westmacot discloses this element.

332. Westmacot explicitly teaches overlaying lines and annotations on roadway images to generate an overlaid image that is then displayed to the user.

For example, it states:

“Figure 6 shows a schematic block diagram that illustrates an example of the **types of adjustment that can be performed at the human fixer stage**. The rendering component can render data of any one of the captured images on a display of the user interface 412, overlaid with its

associated annotation data A(n). One or more selectable options 600 are also rendered, which a user can select in order to cause the model adaptation component 410 to adapt the 3D road model 602, in a fast and intuitive manner. As indicated in Figure 6, as the road model 602 is updated, not only is the annotation data A(n) that is currently being rendered modified in accordance with the adapted model 602, but the adapted model 602 can also be used to update the annotation data 604 for other images in the same video sequence. By way of example, the top part of Figure 6 shows a marked road or lane within an image. Although the right-hand side boundary R2 accurately coincides with a real-world right-side boundary visible in the image, the left-side boundary R1 is slightly off. The user can fix this quickly by shifting the position of the left-side boundary in the 3D model, using the current image and the annotation data rendered on the current image as a reference to manually adapt the location of the left-side boundary R2 until it corresponds with a real-world left-side boundary visible in the image. This change will apply across multiple images in the sequence, meaning that the left-side road boundary will be accurately placed in those images for as long as the road/lane remains the same width.” EX1008, p. 49.

333. Westmacot also describes an “annotation interface [that] provides a human annotator with the ability to view the frames ... with the currently rendered lanes projected into the images. These lanes can be widened, narrowed and moved by the annotator ... All annotations are performed in the estimated 3D road plane, but immediate feedback is provided via projection in the 2D camera view.”

EX1008, pp. 33-34. This makes clear that projected lane lines are visually overlaid on the roadway images.

334. The figures confirm this teaching. Figure 25, for example, shows a captured road image “including annotations for road ... ego-lane ... and lane instance,” all rendered on top of the original image. EX1008, p. 22. Figure 6 similarly depicts lane boundaries R1 and R2 overlaid on a road image, where “the user can fix this quickly by shifting the position of the left-side boundary in the 3D model, using the current image and the annotation data rendered on the current image as a reference.” EX1008, p. 49. The rendering component is described as “render[ing] data of any one of the captured images on a display of the user interface 412, overlaid with its associated annotation data A(n).” *Id.*

335. Westmacot also identifies the horizon line as a specific overlaid feature:

“The plane that is perpendicular to this road surface normal vector can be intercepted with the image plane and this will provide a line across the image that would match the horizon if the car were driving along a perfectly straight and level road (the ‘horizon line’ as that term is used herein).”

EX1008, p. 44, Fig. 16.

336. Finally, Westmacot teaches creating a “3D road model” that “can be projected into the image plane of images to automatically generate 2D annotation

data” which allows “for fast manual ‘tweaks.’” EX1008, p. 10-11. The annotation data is shown to the user via an “image ... marked with the expected road annotations determined for that image.” EX1008, p. 34.

337. Accordingly, a POSITA would have understood that Westmacot expressly discloses overlaying lines such as lane boundaries and horizon lines onto roadway images to generate overlaid images, as required by claim element [1C].

5. [1D] “transmitting the overlaid image to a computing device over a network”

338. Westmacot in view of Tal renders obvious this element.

(a) Westmacot

339. Westmacot teaches the generation, storage, and display of annotated roadway images. For example, it discloses that “the processor 304 receives the captured images from the image capture device 302, and stores them in a memory 306, from which they can be retrieved for use” and that the rendering component may render an image, “overlaid with its associated annotation data.” EX1008, p. 23, 49, Figs. 3, 6, 25. The disclosure further explains that a “rendering component can render data of any one of the captured images on a display of the user interface 412, overlaid with its associated annotation data A(n),” EX1008, p. 49 and that the annotation data is stored in electronic storage 414 “from which it can be accessed or retrieved for use in the training process described above.” EX1008, p. 33.

Westmacot thus establishes that overlaid images exist as digital data objects stored and retrievable within the system. To the extent it is argued that Westmacot does not teach the transmission of the overlaid image over a network, a POSITA would have recognized that once such overlaid images exist digitally, transmission to another computing device over a network would have been obvious in view of Tal.

(b) Tal

340. Tal discloses systems and methods for processing and transmitting roadway images and associated object data in the autonomous driving context. Vehicle-mounted imaging devices capture digital images 16 of the roadway, which may be processed locally to detect objects and generate object data 20.

341. Tal emphasizes two-way communication between the vehicle and external computing resources. Tal explains that “the processed image data 20 and the sensor data 17 is transmitted to the server 107 as object data 21” EX1009, ¶[0099] and that “unprocessed images 16 [can be] sent to the server 107 by the device 101 ... [and] once received, then the server 107 would then process the images 16 as processed image data 20.” EX1009, ¶[0101]. In this architecture, the server 107 can perform additional neural network processing and inference workflows to classify or segment objects of interest. EX1009, ¶[0100]. Tal also confirms that communication is bidirectional: “the Server(s) 107 communicate with Client(s) 1208 (which may be the devices 101 or separate computing devices ...) which allow users to handle the incidents 12 using a user interface 1209” EX1009, ¶[0138]. Through this interface, users or client devices can view and interact with the server-processed results (e.g., incident detections, annotated images) in different formats such as map view, list view, or gallery view. EX1009, ¶¶[0148]–[0153]. Taken together, these disclosures show that Tal supports a

feedback loop where images (raw or partially processed) are transmitted to the server, the server applies additional processing, and the results are disseminated back to client devices, including the vehicle itself.

342. Tal therefore squarely teaches that both raw images and overlaid/processed images can be shared with remote servers for additional processing, storage, and reporting, and that updated calibration information may be returned to the vehicle. A POSITA would have recognized that these teachings provide conventional, well-understood mechanisms for transmitting overlaid images and calibration data in autonomous driving systems.

343. In sum: Tal describes how image data, including images with overlays, is transmitted to a remote computing device (server 107) over a communications network. Specifically, Tal states: “Digital images 16 are taken by one or more imaging devices 101 ... preferably with network connection capabilities to a communications network 18 (e.g. wireless network 105, cellular network 106, etc.),” and that “unprocessed image(s) 16 can be transmitted by the device 101 over the network 18, as they are acquired in real time ... [or] resultant processed image data 20 ... can be transmitted to the server 107 by the device 101 over the network 18.” EX1009, ¶[0027]. Tal further discloses that “Neural Network(s) Inference results 1000 of a processed image 16 [include] bounding boxes 1002, polygons 1003, masks 1004, and landmarks 1005 as part of the

resultant processed image data 20.” EX1009, ¶[0114]. These inference results, when superimposed on the original image, constitute overlaid images. Tal makes clear that “the processed image data 20 and the sensor data 17 is transmitted to the server 107 as object data 21” EX1009, ¶[0099] and further that client devices may access server-stored data through interfaces that present incident images in map, list, or gallery views EX1009, ¶[0138]–[0148].

(c) KSR

344. A POSITA would have been motivated to combine Westmacot with Tal for several reasons including those I outlined at the outset of Ground 3. *See* Section X.C. First, the references are directly complementary: Westmacot already generates and stores annotated (overlaid) images but is silent on transmitting them, while Tal teaches the routine step of transmitting raw and overlaid images across a network. Second, by the priority date, centralized machine-learning workflows and distributed training pipelines in the autonomous driving field relied heavily on networked sharing of annotated and calibrated images. A POSITA would have recognized that transmitting Westmacot’s annotated images, as in Tal, was an obvious and predictable way to support large-scale training, validation, and remote access. Third, once an image exists as a digital object stored in memory (as Westmacot discloses), transmission across a network is nothing more than an

obvious design alternative to local display, especially in light of Tal's explicit teaching of using standard wireless/cellular links for that purpose.

345. The combination would also have been entirely predictable and achievable. Both Westmacot and Tal describe software-implemented image processing systems in the autonomous driving domain. Networking technologies such as TCP/IP and wireless protocols were mature and routinely used in 2016-2018 to share digital image data. Thus, a POSITA would have reasonably expected success in applying Tal's network transmission mechanism to Westmacot's digital overlaid images with no technical barrier.

346. Accordingly, even if Westmacot does not expressly disclose transmitting the overlaid image to a computing device over a network, Tal fills that gap. A POSITA would have understood that combining Westmacot's annotated image generation with Tal's explicit disclosure of transmitting overlaid images to remote servers represents nothing more than the predictable use of known elements to achieve the well-understood goal of enabling remote review, validation, and training of annotated roadway imagery.

6. [1E] receiving a modification of the line from the computing device, the modification comprising a new line at a second position;

347. Westmacot in view of Tal renders obvious this limitation.

(a) Westmacot

348. Westmacot teaches receiving a modification of a line from a computing device, where the modification comprises a new line at a second position. Westmacot repeatedly explains that its annotation interface (UI 412) enables a human annotator to view images with overlaid lane lines and to adjust or correct those lines. For example, “an annotation interface provides a human annotator with the ability to view the frames in a batch individually with the currently rendered lanes projected into the images. These lanes can be widened, narrowed and moved by the annotator ... providing a clear advantage over annotation of all of the image[s] individually.” EX1008, p. 33, Fig. 18–24.

349. Similarly, the disclosure explains that “the user can modify the annotation data A(n) for an individual image via the UI 412, in order to better align the assumed lane boundaries R1, R2 with the actual lane boundaries in that image,” and that a “model adaptation component 410” allows efficient manual adaptation of annotation data across images. EX1008, p. 35. Figures 6 and 25 show specific examples where an annotator shifts an existing lane boundary to a new position or adds additional lanes, bus lanes, or cycle lanes. EX1008, p. 35, Fig. 6, Fig. 25.

These modifications generate new line positions, i.e., “a new line at a second position”, which are then propagated across multiple frames of the dataset.

350. Westmacot expressly discloses computation and projection of the horizon line as part of its 3D model of the roadway. For example, Figure 16 and related text describe computing a road surface normal and intersecting it with the image plane to “provide a line across the image that would match the horizon ... (the ‘horizon line’ as that term is used herein).” EX1008, p. 44, Fig. 16 discussion. Because the system both (i) overlays this computed horizon line on the image and (ii) provides a user interface for modifying line annotations (UI 412; model adaptation 410), a POSITA would have readily understood that the same modification mechanism applies equally to the horizon line. Indeed, the ability to adjust the horizon line is functionally identical to adjusting a lane boundary: both are linear annotations derived from the 3D model and both affect calibration outputs. A POSITA would therefore have recognized it as a straightforward extension, well within routine skill, to enable user-driven adjustment of the horizon line using the same tools that Westmacot already provides for lane lines. Thus, Westmacot supplies the “what”: user-generated line modifications that create a new line position but does not expressly disclose the “how”: the network-based receipt of the user’s line modification.

(b) Tal

351. Tal supplies the how: “over a network.” Tal discloses a vehicular device 101 that captures roadway images 16 and communicates with a remote server 107 over a network 18; the device may buffer data locally and then transmit the processed “data 20/sensor information 17 ... over the network 18,” which the server ingests as object data 21. EX1009, ¶[0030]. Tal further states that “the server 107 is responsible for ... processing ... the object data 21 uploaded by the device(s) 101,” including when “the server 107 receives object data 21 from a device 101 through the internet 18.” EX1009, ¶[0060]. Processing can occur on the device 101, on the server 107, or be shared/distributed across both in a “shared/distributed processing environment 400.” EX1009, ¶[0030]. Tal repeatedly describes this split pipeline: a first image processing portion 402 runs on the device using image processing instructions 905, after which the resulting object data 21 is sent to the server 107 for a second image processing portion 404 (also using instructions 905), with the server storing the processed image data 20'. Tal also explains that both device 101 and server 107 can host the image processing instructions 905, and that the device-side processing generates processed image data 20 and object data 21 is “transmitted to the server 107.” EX1009, ¶[0050]-[0051].

352. Thus, with Westmacot’s human-in-the-loop line editing running on a computing device (e.g., a server-hosted UI or workstation) and Tal’s networked device–server pipeline, the edited line (expressed as updated line parameters/coordinates) is a routine message payload that the other computing node receives. If the edit occurs on the server-side UI, the vehicle device 101 receives the updated line definition from server 107; if the edit occurs on a client UI attached to the device, the server 107 receives the updated line definition from device 101. Tal’s shared/distributed architecture expressly supports either direction of transmission and processing between device 101 and server 107: sending “unprocessed image(s) 16” or “processed image data 20” to the server and having the server “organizing, storing, processing and disseminating” results back to clients. EX1009 ¶¶[0027], [0050]–[0052], [0066], [0138].

(c) KSR

353. The combination of Westmacot and Tal therefore makes clear that receiving a modification comprising a new horizon line at a second position would have been obvious, since it merely combines Westmacot’s disclosed line-adjustment framework and Tal’s remote transmission/feedback.

354. **Combine known elements with predictable results.** Westmacot already produces user-corrected lines (new position). Tal already provides the networked channel and shared processing to carry camera/overlay/state between

computing devices. Combining them so the edited line is transmitted and received between Tal's device 101 and server 107 is a textbook combination of complementary teachings; the outcome: one node receiving a modification comprising the new line at a second position, is entirely predictable.

355. **Use of a known technique in an analogous context.** Westmacot's UI captures edits as updated geometry (endpoints, parametric spline, or line pose in the 3D model). Tal's pipeline already communicates processed/unprocessed image data and associated metadata over network 18; encapsulating a line-edit as a small metadata message is the same, known technique applied in Tal's networked vision system.

356. **Finite, predictable options / common sense implementation.** A POSITA would recognize only a few routine ways to represent the modification (e.g., new line endpoints in image coords, delta from prior line, polynomial/spline coefficients, or a "replace-line" control message). Any of these is trivially packaged as Tal's "resultant processed data 20 / object data 21" or an instruction to update overlays and sent between device 101 and server 107 using the already-disclosed communications mechanism.

357. A POSITA would have recognized that extending Westmacot's annotation-modification workflow to operate on a remote computing device, consistent with Tal's network-based architecture, would have been nothing more

than the predictable use of known elements. The motivations to do so were clear:

(1) Scalability and efficiency: offloading computationally intensive 3D reconstruction and line interpolation tasks to more powerful servers reduces the burden on the annotator's local system; (2) Collaboration and consistency: a networked implementation allows multiple annotators to contribute to the same dataset in real time, ensuring consistency and reducing duplication; and (3) Integration with cloud-based workflows: Tal expressly teaches transmitting annotated/overlaid images to a server, naturally complementing Westmacot's local modification process.

358. Accordingly, Tal makes clear that a POSITA would have been motivated, with a reasonable expectation of success, to adapt Westmacot's system to receive modifications of lines at a remote computing device. The result: (receiving a modification comprising a new line at a second position) is the predictable outcome of combining Westmacot's local annotation interface with Tal's conventional teachings of network-based processing.

7. [1F] computing a camera parameter based on the new line;

359. Westmacot in view of Tal renders obvious this feature.

(a) Westmacot

360. Westmacot expressly discloses computing camera parameters based on detected or modified line geometry. Westmacot describes a “parameter computation component 408” that calculates camera-related parameters such as camera height (H), horizontal offset (S), orientation or forward point (f), horizon line, and lane width (W). EX1008, pp. 34-37, 42-49, Fig. 3; component 408. These parameters are updated whenever lane boundaries (R1, R2) are detected or adjusted, ensuring that the 3D road model remains consistent with the captured images. For example, “a user can adjust the assumed lane width W to better fit one image, and because the road model is being adapted to the new lane width, the annotation data for other images can be automatically adapted based on the adapted model with the new lane width W.” EX1008, p. 35. Westmacot further discloses that the forward point (camera orientation) can be derived using tangents from lane geometry, and that the horizon line is obtained by intersecting the road surface normal with the image plane. EX1008, pp. 29, 30, 34-35, Fig. 15–16 discussion. Camera height (H) is calculated from a reconstructed road surface, and offsets (S, C) are estimated by detecting lane boundaries and measuring their relationship to the camera’s transverse axis. *Id.*, pp. 39-40, 47

361. This disclosure makes clear that Westmacot computes “camera parameters” directly from the geometry of detected or user-modified lines. The system explicitly recalculates parameters such as height, lateral offset, forward point, and lane width whenever line positions change. The “human fixer” stage reinforces this link, noting that corrections to lane boundaries cause updates to the calculated camera height, lateral position, and lane width. EX1008, p. 47-48. These recalculations represent precisely what claim 1 requires: computing a camera parameter based on the new line.

(b) Tal

362. While Westmacot describes this computation locally, Tal provides the explicit teaching of distributing and recomputing calibration data across a network. Tal explains that captured roadway images and associated detection data can be transmitted by the imaging device to a remote server over wireless or cellular networks, either as unprocessed images or as processed data with object overlays. EX1009, ¶¶[0027], [0030], [0052], [0075], [0138]. At the remote site, the external server may process the received image data to determine calibration-related outputs or generate modified image information, which can then be accessed through a web application for annotation, calibration, or review. EX1009, Summary ¶¶[0005]-[0013]; Fig. 12. These disclosures confirm that Tal teaches a distributed calibration framework, in which calibration operations may be

performed remotely and results made available back to the vehicle or user, complementing Westmacot's local computations.

(c) KSR

363. A POSITA would have understood that combining Westmacot's explicit parameter computation engine with Tal's network-based architecture is nothing more than applying conventional distributed processing techniques. The motivations would have been clear and predictable and similar to what I have articulated at the outset of Ground 3 and for the additional specific reasons below. *See Sec. X.C.*

364. **Verification and consistency.** Westmacot's parameters (H, S, W, forward point, horizon) are derived from detected or modified lines. Allowing those computations to occur remotely, as Tal teaches, ensures consistency across distributed training or calibration environments. Remote operators thereby see what the vehicle sees and recompute identical parameters from the same modified line data.

365. **Operational efficiency.** Offloading calibration computations to remote servers reduces the computational burden on the annotator's or vehicle's local system while still operating on the same roadway imagery and detected line geometry captured locally. Tal expressly teaches that "digital images 16 are taken by one or more imaging devices 101 ... [and] only selected data portions 20 ... can

be transmitted ... to a server 107 for subsequent processing ... Alternatively, unprocessed image(s) 16 can be transmitted ... to the server 107 for subsequent processing. It is also recognized that both resultant processed image data 20 and unprocessed images 16 can be transmitted to the server 107 by the device 101.” EX1009, ¶¶[0027], [0030]. Tal further explains that “processing of images 16 can be performed by the device 101 alone, by the server 107 alone, or by both the server 107 and the device 101 as provided for in FIG. 4 for the shared/distributed processing environment 400.” EX1009, ¶¶[0030]. Once transmitted, “the server 107 is responsible for the organizing, storing, processing and disseminating of the object data 21 uploaded by the device(s) 101” and “can facilitate as gateway to users ... to make data available ... in a meaningful and intuitive manner.” EX1009, ¶¶[0031], [066], [0138]. Tal also highlights efficiency gains, noting that “any discarded image data ... constitutes bandwidth transmission savings ... [and] the system 10 can take advantage of a split image processing framework ... such that a first portion ... can be performed on the device ... while a second portion ... can be performed on the server 107.” EX1009, ¶¶[0050], [0075]. These disclosures confirm that Tal’s architecture enables resource-constrained vehicles or client devices to rely on more powerful remote processors for parameter determination, improving efficiency without altering the underlying data inputs.

366. **Human-in-the-loop calibration.** A POSITA would recognize that when a user modifies a line (*e.g.*, shifts a lane boundary or horizon line), the resulting recalculated parameters could be computed at a remote server, returned to the vehicle, and used to update camera orientation or positioning in real time, exactly as Tal teaches. Tal explains that “processing of images 16 can be performed by the device 101 alone, by the server 107 alone, or by both the server 107 and the device 101 as provided for in FIG. 4 for the shared/distributed processing environment 400.” EX1009, ¶[0030]. Tal further teaches that “the server 107 is responsible for the organizing, storing, processing and disseminating of the object data 21 uploaded by the device(s) 101” EX1009, ¶[0066], thereby making clear that calibration-relevant processing performed remotely can be transmitted back and applied in the vehicle context.

367. Accordingly, Westmacot discloses computing camera parameters based on detected or modified line geometry, and in view of Tal, it would have been obvious to implement this functionality in a networked architecture, where parameter computation could occur locally or remotely. Tal expressly discloses that “both resultant processed image data 20 and unprocessed images 16 can be transmitted to the server 107 by the device 101” EX1009 ¶[0027] and that “a first portion 402 of the image 106 processing can be performed on the device 101 itself ... while a second portion ... can be performed on the server 107 post receipt of

the object data 21” EX1009, ¶[0050]. This is nothing more than the predictable use of known elements according to their established functions, yielding the foreseeable result of updated camera parameters based on new line positions, with a reasonable expectation of success.

8. [1G] transmitting data representing the camera parameter to the camera device.

368. Westmacot in view of Tal renders obvious this limitation.

(a) Westmacot

369. Westmacot discloses computing a set of camera parameters, including camera height (H), lateral offset (S), lane width (W), forward point, horizon line, and orientation, based on detected or modified lane boundaries. These parameters are calculated by the “parameter computation component 408” and then passed internally to other components such as the model adaptation component 410, rendering component 416, and user interface 412 to update the road model and displayed overlays. *See* EX1008, pp. 34-37, 42-49, Figs. 3, 6, 25, 30–32; components 408, 410, 412, 416. In this sense, Westmacot shows that computed camera parameters are actively transmitted within the system to control annotation display and model alignment. However, Westmacot does not expressly disclose transmitting those parameters back to the camera device (302) itself.

(b) Tal

370. Tal directly addresses this gap by teaching network-based transmission of image-derived data, including calibration-related outputs, to and from external devices over a communications network. For example, Tal explains that “digital images 16 are taken by one or more imaging devices 101 ... and ... resultant processed image data 20 ... can be transmitted to the server 107 by the device 101 over the network 18.” EX1009, ¶[0027]. Tal further teaches that “processing of images 16 can be performed by the device 101 alone, by the server 107 alone, or by both the server 107 and the device 101 as provided for in FIG. 4 for the shared/distributed processing environment 400” EX1009, ¶[0030], and that “the server 107 is responsible for the organizing, storing, processing and disseminating of the object data 21 uploaded by the device(s) 101” EX1009, ¶[0066]. Tal also makes clear that “a first portion 402 of the image 106 processing can be performed on the device 101 itself ... while a second portion ... can be performed on the server 107 post receipt of the object data 21” EX1009, ¶[0050], with efficiency gains from “bandwidth transmission savings” when only refined outputs are returned. EX1009, ¶[0050]. These disclosures establish two-way communication in which calibration data derived from imagery may be computed remotely and transmitted back for use by the originating camera device, thereby filling the gap left by Westmacot.

(c) **KSR**

371. A POSITA would have found it obvious to combine Westmacot with Tal to transmit the camera parameters computed in Westmacot back to the camera device (302) for all the reasons I articulated at the outset and the additional specific reasons below.

372. **Calibration and Feedback:** Westmacot's camera device 302 provides the raw roadway images used for annotation and parameter computation. Transmitting the computed parameters (*e.g.*, orientation, horizon line, offset) back to that same device would allow the camera to self-calibrate, automatically correct for pitch/yaw/roll offsets, adjust capture settings (*e.g.*, cropping or field-of-view corrections), or embed metadata into the captured images for downstream use. Tal expressly teaches a distributed system in which image and sensor data are uploaded from the in-vehicle device 101 to a remote server 107, where they are processed and organized, and then made available back to users or client devices through the server gateway. EX1009, ¶¶[0027]–[0031], [0066]–[0068]. Although Tal frames this in terms of object/incident data rather than calibration parameters (which Westmacot does disclose), the architecture plainly establishes a routine feedback loop: images and metadata flow outward for processing, and the results are disseminated back to the vehicle or external users. A POSITA would have recognized that extending this loop to include Westmacot's calibration value rather

than just object classifications, was a straightforward, predictable application of the same framework.

373. **Distributed Processing:** Westmacot already demonstrates intra-system transmission of computed parameters from one module to another. Tal shows that transmitting image-derived parameters across devices and networks is conventional. Extending Westmacot's internal communication to include the originating camera device is therefore nothing more than a predictable application of distributed design, ensuring that the camera device has awareness of and can utilize its calibrated parameters.

374. **Predictable Results:** By transmitting the computed camera parameters back to the camera device, a POSITA would have expected improved calibration accuracy and operational efficiency. The device could embed calibration metadata in each captured frame or apply real-time corrections, thereby enhancing the fidelity of downstream annotation and perception processes. These benefits flow directly from closing the feedback loop between the processing system (computing the parameters) and the data source (the camera device), a well-understood and conventional design choice in networked image-processing architectures

375. Accordingly, even though Westmacot does not expressly disclose transmitting computed camera parameters to the camera device, Tal makes clear

that doing so would have been obvious. Combining Westmacot's explicit parameter computation with Tal's explicit teaching of transmitting calibration data back to the device represents the routine use of known elements for their established purposes, yielding predictable improvements with a reasonable expectation of success.

E. Claim 2

376. Westmacot discloses claim 2.

377. Claim 2 recites that the "camera parameter comprises one of camera height, viewing angle, and road plane normal." Westmacot expressly discloses each of these.

378. **Camera Height (H).** Westmacot teaches that its parameter computation component explicitly computes the camera height above the road:

"The height H of the camera 302 above the road" EX1008, p. 32.

"The camera height above the road ft is easy to measure manually. However, in case this cannot be done ... it is also possible to obtain the height of the camera using the estimated mesh of the road surface obtained from OpenSfM." EX1008, p. 26.

"The height of the camera above the road (H) can be calculated by generating a surface mesh of the road from the SLAM process ... and then averaging the height of all points on the camera path above this mesh." EX1008, p. 45.

“The height H at a given point on the camera path CP is defined [as] the distance between a point on the camera path CP and the mesh along the vehicle’s vertical axis aV at that point.” EX1008, p. 45

“It is straightforward to determine the vehicle path VP at ground level from the camera path CP if the height of the camera 302 above the surface of the road, labelled H in Figures 3, 3A and 5, the horizontal offset of the camera 302 from the center line C, labelled S in Figures 3A and 5, and the orientation of the camera relative to the vehicle are known.” EX1008, p. 30.

379. Viewing Angle (captured as forward point and horizon line).

Westmacot also discloses that the orientation of the camera, *i.e.*, its viewing angle, is captured by computing both a forward point and a horizon line:

“The orientation of the camera 302 relative to the vehicle, which can be captured as: a. A ‘forward point’ of the camera 302 ... and b. A ‘horizon line.’” EX1008, p. 33.

“This orientation is captured as the forward point of the camera, defined to be the pixel that the car appears to head towards when travelling in a straight line ... This pixel is the point which the car would seem to be moving towards if it were driving on a perfectly straight and flat road (the ‘forward point’ as that term is used herein).” EX1008, pp. 40-41.

“The plane that is perpendicular to this road surface normal vector can be intercepted with the image plane and this will provide a line across the image that would match the horizon ... (the ‘horizon line’ as that term is used herein).” EX1008, p. 44.

380. **Road Plane Normal (n).** Finally, Westmacot discloses computing a road surface normal vector, which is directly used as a calibration parameter:

“The road normal n is estimated based on the fact that, when the car moves around a turn, the vectors representing it[‘]s motion m will all lie in the road plane, and thus taking the cross product of them will result in the road normal, see Fig. 27.” EX1008, p. 26.

“The estimated road normal at frame i (in camera coordinates) is [given by] the cross-product ... (see Fig. 27).” EX108, p.27.

381. Accordingly, Westmacot explicitly discloses camera height, viewing angle (forward point and horizon line), and road plane normal as parameters calculated during its annotation and road-modeling process. A person of ordinary skill in the art would therefore have understood that Westmacot meets the limitation of claim 2.

F. Claim 3

382. Westmacot discloses claim 3: “receiving an image of the roadway while the vehicle is moving.”

383. Westmacot expressly teaches that roadway images are captured from a vehicle-mounted camera while the vehicle is in motion. For example, it describes “receiving a time sequence of two dimensional images as captured by an image capture device of travelling vehicle.” EX1008, p. 3. Likewise, the Second Aspect explains “receiving a time sequence of two dimensional images as captured by the

image capture device of the vehicle; processing the images to reconstruct, in three-dimensional space, a path travelled by the vehicle.” EX1008, Abstract. Figure 3 illustrates a “vehicle 300 that can be used to capture road images ... preferably ... captured as frames of short video segments recorded as the vehicle 300 drives along a road.” EX1008, p. 22. Westmacot further “recognizes that driving the car is itself a form of annotation” and emphasizes that its methods are “particularly well suited to batch-annotation of video frames as captured by a moving training vehicle.” EX1008, p. 18-19. The experimental setup confirms the point: “videos and associated GPS data were captured with a standard Nextbase 402G Professional dashcam ... [and] in order to remove parts where the car moves very slow or stands still ... only frames that are at least 1m apart according to GPS are included.” EX1008, p. 22. This filtering step ensures that the images are captured while the vehicle is moving, not while stopped.

G. Claim 4

384. Westmacot in view of Tal renders obvious claim 4: “receiving the image of the roadway while the vehicle is moving comprises detecting that the vehicle is traveling above a pre-defined speed or for a pre-defined duration.”

1. Westmacot

385. Westmacot expressly teaches filtering roadway images to ensure that they are captured only while the vehicle is in motion, using a distance-based

threshold derived from GPS data. Specifically, Westmacot explains: “In order to remove parts where the car moves very slow or stands still (which is common in urban environments), only frames that are at least 1m apart according [to] GPS are included.” EX1008, p. 22. This disclosure makes clear that the system excludes frames captured when the vehicle is moving too slowly or not at all, thereby ensuring that annotation is performed only on images recorded while the vehicle is traveling.

386. Although Westmacot implements this filter as a 1-meter distance criterion, a POSITA would understand that this is effectively equivalent to detecting whether the vehicle exceeds a minimum speed or has traveled for a minimum duration. Because the frames are sampled at fixed time intervals, requiring that successive frames be at least 1 meter apart sets a minimum speed threshold (*e.g.*, if captured at one-second intervals, the threshold is ~1 m/s or 3.6 km/h). Similarly, the distance requirement also functions as a duration condition, since it ensures that the vehicle has been in motion for enough time to traverse the specified distance before the next frame is considered. In this way, Westmacot discloses the claimed concept of detecting that the vehicle is traveling above a pre-defined speed or for a pre-defined duration.

387. To the extent Patent Owner argues that Westmacot does not expressly disclose speed or time thresholds, it would nonetheless have been obvious to a

POSITA to implement such thresholds as interchangeable and routine alternatives.

In the art of autonomous driving and data curation, it was conventional to filter imagery based on vehicle telemetry, whether expressed as distance traveled, speed over ground, or elapsed driving time. Each approach achieves the same practical goal: excluding idle or low-speed frames that add noise rather than useful roadway content. Substituting a minimum speed (*e.g.*, 5 km/h) or a duration threshold (*e.g.*, 2 seconds of continuous travel) for Westmacot's 1-meter GPS filter would have been nothing more than a predictable design choice well within ordinary skill.

2. Tal

388. Finally, Tal reinforces the obviousness of such an implementation because it teaches transmitting roadway imagery and associated metadata, including position, speed, and other telemetry, over a network to remote servers for further processing. *See* EX1009, ¶¶[0027], [0030], [0052], [0075], [0138]. A POSITA would have recognized that leveraging this telemetry to enforce speed or duration thresholds before transmitting images was a straightforward application of known techniques to improve efficiency, scalability, and data quality.

3. KSR

389. A POSITA would have been motivated to combine Westmacot's explicit teaching of filtering frames based on a minimum distance traveled with Tal's disclosure of transmitting roadway imagery and associated telemetry

(including vehicle speed and movement data) to remote servers. *See* EX1009, ¶¶[0027], [0030], [0052], [0075], [0138]. Both references address the same problem: ensuring that roadway imagery used for annotation, calibration, or navigation reflects meaningful driving conditions rather than idle or low-speed noise.

390. Specifically, Westmacot’s requirement that frames be “at least 1m apart according [to] GPS” reflects a de facto speed/duration threshold, while Tal demonstrates that speed, distance, and time telemetry were routinely transmitted and relied upon in autonomous vehicle systems. A POSITA would have recognized that substituting or supplementing Westmacot’s GPS distance filter with a speed threshold (*e.g.*, >5 km/h) or duration threshold (*e.g.*, >2 seconds of continuous travel) was an obvious design choice, since these criteria are functionally interchangeable and widely used in the field to gate data quality.

391. Combining the teachings yields predictable results: filtering roadway imagery to ensure it is captured only when the vehicle is traveling at meaningful speeds or for a sufficient duration, thereby improving calibration accuracy, training data consistency, and system efficiency. This reflects nothing more than applying known telemetry-based filtering techniques in the context of Westmacot’s annotation pipeline, with a reasonable expectation of success.

392. Accordingly, Westmacot in view of Tal discloses, or at the very least renders obvious, the limitation of claim 4.

H. Claim 5: The method of claim 1, wherein transmitting the overlaid image to the computing device over the network comprises transmitting the overlaid image to a web-based application.

393. Westmacot in view of Tal renders this limitation obvious.

1. Westmacot

394. Westmacot discloses the capture of roadway images by a vehicle-mounted camera, the overlay of annotation data onto those images, and the ability to transmit annotation data to or from a remote server for training and calibration purposes. *See, e.g.*, EX1008, Fig. 3 (vehicle 300 with image capture device 302); annotation interface/UI 412; rendering component 416 (overlaid display); electronic storage 414 (annotation data storage and retrieval).

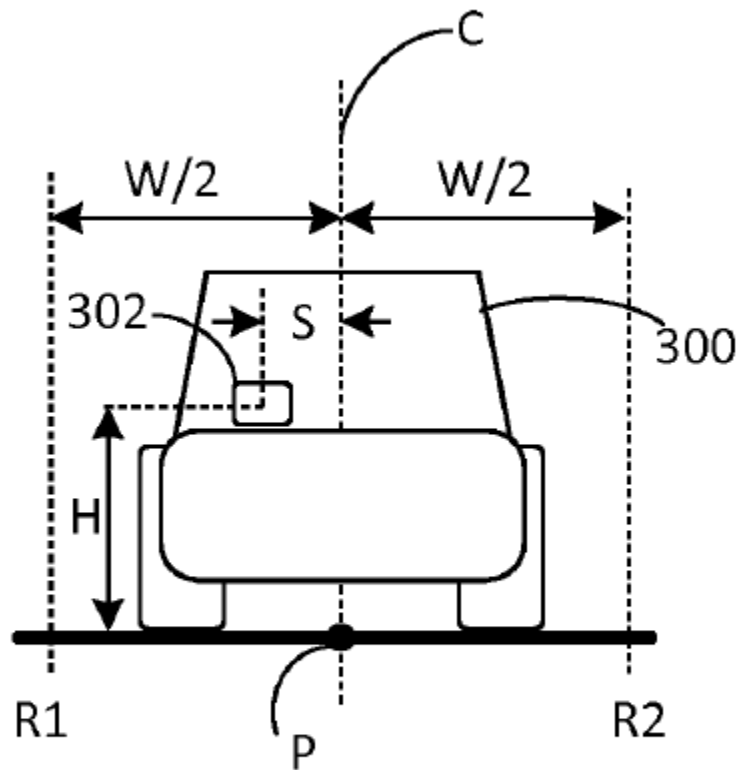


FIG. 3A

395. Westmacot therefore discloses digitized overlaid images transmitted to a computing system for annotation and calibration. However, it does not expressly describe transmitting those images to a web-based application. Its focus is on transmission to remote servers for annotation and training.

2. Tal

396. On the other hand, Tal expressly teaches that images and incident data captured by a device (*e.g.*, smartphone, embedded computer, or smart camera mounted in a vehicle) are processed locally and transmitted to a remote server,

which then provides access to users through a web application. For example, Tal explains:

“The system comprising of: device—a mobile computing device such as smartphone, or embedded computer system with built in or peripheral camera(s) and application utilizing neural network(s) and machine learning work together with server(s) that process and store data received from single or multiple devices, serve as gateway to users via web access and present data to users in a meaningful and intuitive manner.” EX1009, ¶[0009].

“The server 107 also provides for client(s) 1208 to securely log in to access a user interface 1209, which may be either a web application that can be accessed using a web browser or a client/server application that uses physical installation to a computer or a smartphone 101.” EX1009, ¶[0148], Fig. 12.

397. Tal therefore explicitly identifies that transmitted roadway/incident imagery can be presented to users through a browser-based web application.

3. KSR

398. A POSITA would have been motivated to combine Tal’s explicit teaching of web-based applications with Westmacot’s disclosure of transmitting annotated roadway images to remote servers because:

399. **Complementary Teachings.** Westmacot already teaches the generation of annotated roadway images by overlaying lane lines and road structures on captured camera frames, and further describes transmitting annotation

data to and from remote servers for calibration and training. *See* EX1008, p. 34.

Tal, in turn, discloses that images and processed data transmitted to a server can be accessed by end users via a web application. Tal explains that its server “serve[s] as gateway to users via web access and present[s] data to users in a meaningful and intuitive manner,” and that client users can log in to a “user interface ... which may be either a web application that can be accessed using a web browser or a client/server application.” EX1009, ¶¶[0009], [0148], Fig. 12. Taken together, the references are mutually reinforcing: Westmacot produces the overlaid image data that is suitable for transmission, and Tal provides the conventional mechanism (*e.g.*, web-based access) for delivering that data to remote users. This pairing fills Westmacot’s gap (how overlaid images reach end users) with a known and routine technique disclosed by Tal.

400. **Accessibility & Collaboration.** By the priority date, web applications were an entrenched mechanism for providing broad access to server-stored data. They enabled distributed users to view, edit, and collaborate on shared datasets through nothing more than a browser, eliminating the need for specialized client-side software. A POSITA would have recognized that implementing Tal’s web interface in Westmacot’s system would allow annotated roadway images to be distributed to a wide range of stakeholders (*e.g.*, municipalities, contractors, fleet operators, or research collaborators) without requiring proprietary software

installations. Web access also supports cross-platform compatibility (*e.g.*, desktop, tablet, mobile), which would be particularly valuable in annotation workflows involving large teams or field-based personnel. Thus, integrating Tal's web-based application with Westmacot's annotated imagery would predictably improve accessibility, lower deployment costs, and facilitate multi-party collaboration.

401. **Efficiency & Scalability.** A POSITA would also appreciate that Tal's architecture, where the server acts as a gateway to users through a web application, was designed for scalability and centralized processing. Applying this to Westmacot's annotation and calibration system would allow geographically distributed annotators and supervisors to work from the same server-based dataset, ensuring consistency and reducing duplicative local processing. This integration would further enable centralized machine learning training pipelines, where annotated frames transmitted from vehicles (Westmacot) are aggregated, validated, and managed centrally before being used to train perception models. Web-based distribution was already well-established as a means of scaling annotation throughput, supporting version control, and enabling real-time review of results by remote experts. In short, combining Westmacot and Tal would not only provide technical compatibility but also confer well-recognized advantages of scalability, efficiency, and quality control that a POSITA would have reasonably expected to achieve.

402. **Predictable Use of Prior Art Elements.** Applying Tal’s web-based access mechanism to Westmacot’s existing framework of annotated image generation and transmission is nothing more than using known elements for their intended function (remote image access and review) with predictable results and benefits.

403. Accordingly, even if Westmacot does not expressly disclose web-based delivery, combining it with Tal would have been obvious. A POSITA would have had a reasonable expectation of success in implementing this combination, which reflects no more than a straightforward application of conventional web-based distribution techniques to Westmacot’s already-digitized and transmitted overlaid images.

I. Claim 6

404. Westmacot in view of Tal renders obvious this claim.

1. Westmacot

405. Westmacot expressly teaches computation of the *horizon line* by deriving a road surface normal from the vehicle’s motion and intersecting it with the image plane:

“...the plane that is perpendicular to this road surface normal vector can be intercepted with the image plane and this will provide a line across the image that would match the horizon if the car were driving along a

perfectly straight and level road (the ‘horizon line’ as that term is used herein).” EX1008, p. 44, Fig. 16.

406. Westmacot further explains that camera parameters such as height (H), horizontal offset (S), and orientation (forward point, horizon line) are computed from detected lane geometry, but acknowledges that these parameters “may not be exactly correct” and thus require manual correction during the “human fixer” stage. *See* EX1008, p. 47, parameter computation component 408; human fixer stage.

407. At the same time, Westmacot’s annotation interface (UI 412) allows human operators to directly manipulate roadway feature lines within camera images. For example, lane boundaries (R1, R2) can be shifted, widened, or narrowed by dragging them within the image, and those modifications are propagated across the 3D road model. *See* EX1008, pp. 35, 51, Fig. 6; Fig. 25. A POSITA would have recognized that the *horizon line* (already explicitly defined and computed) is another geometric reference line in the same image space, functionally no different from lane lines. Both lane lines and the horizon are linear annotations tied to the underlying 3D road model and both affect recalculated camera parameters (pitch for horizon; lateral offset and width for lane lines).

2. Tal

408. Tal further confirms this understanding by disclosing that its roadway incident detection system employs a computing device with a user interface to display roadway images and calibration parameters and to receive user input. For example, Tal teaches that “the device 101 can have a user interface 119 including a display ... to display information ... such as... the camera 500 field of view (viewfinder), the orientation of the device 101, status indicators, settings, parameters, and other information related [to] the installation, configuration operation, and maintenance of the device 101.” EX1009, ¶[0035]. Tal also explains that the server “serve[s] as gateway to users via web access and present[s] data to users in a meaningful and intuitive manner.” EX1009, ¶[0009]. And Tal notes that road surveyors historically identified roadway issues manually using tablets or smartphones, confirming that user identification of roadway features through a computing device was conventional.

3. KSR

409. Taken together, these references render obvious claim 6. Westmacot shows that line modifications (shifting, resizing, adding) are received from a human annotator through a computing device. Tal shows that roadway image data and parameters are presented to users through mobile or web-based computing interfaces, which accept user identification and input. A POSITA would have

found it obvious to apply Tal's computing-device interface to Westmacot's line-modification framework, thereby allowing a user to identify a new horizon line via the device. Horizon lines, like lane lines, are geometric features central to calibration; enabling a user to confirm or correct them is simply another application of the same disclosed functionality.

410. A POSITA would have been motivated to extend Westmacot's disclosed annotation framework to include manual identification and modification of horizon lines, particularly in view of Tal's explicit disclosure of user-facing computing interfaces for roadway images and parameters.

411. Functional Similarity of Lane Lines and Horizon Lines.

Westmacot discloses that its annotation interface allows annotators to directly manipulate lane boundaries (R1, R2) projected into the camera image, with adjustments propagated through the underlying 3D road model. *See* EX1008, pp. 36-37, 51 UI 412; Fig. 6; Fig. 25. A POSITA would have readily appreciated that the horizon line plays a parallel geometric role in calibration. Both lane boundaries and the horizon line are fundamental visual references derived from the camera's perspective, and both constrain the relationship between the vehicle coordinate system and the camera coordinate system. Lane line geometry provides lateral and width constraints, whereas the horizon line provides vertical and pitch constraints. Just as modifying a lane line affects the computed lane width (W), lateral offset

(S), and center (C), adjusting the horizon line affects pitch orientation and the effective viewing angle of the roadway scene. Because both lane lines and the horizon line are simultaneously visualized in the same captured image, they can be adjusted using the same interface mechanics (dragging, repositioning, or confirming line placement) without any change in the principle of operation. To a POSITA, extending Westmacot's annotation interface to include user identification of the horizon line would have been a straightforward and predictable step, leveraging the same UI paradigm already disclosed for lane boundaries.

412. **Calibration Needs and Error Correction.** Westmacot itself acknowledges that camera parameters such as height, offset, and horizon may not be "exactly correct" when computed automatically and require manual correction through the "human fixer" stage. This expressly suggests human-in-the-loop verification of camera parameters to ensure calibration accuracy. A POSITA would have understood that the horizon line, in particular, is a visually salient and easily recognizable reference feature that could be manually identified by an annotator to correct errors in pitch calibration. Because the horizon line directly reflects the alignment of the camera's vertical axis with the road plane, a misestimated horizon line translates into pitch misalignment, producing systematic error in camera calibration. Allowing an annotator to identify or confirm the horizon line therefore provides a direct, efficient means to reduce error and improve calibration

robustness. Moreover, horizon lines were widely recognized in the art as key geometric primitives used in camera calibration workflows. Thus, empowering users to adjust the horizon line through the same annotation interface already used for lane line adjustments would have been viewed by a POSITA as a natural extension of Westmacot's correction process.

413. **Tal's Computing Device Interface.** Tal discloses that its system provides roadway images and calibration-related parameters to users via mobile and web-based interfaces, allowing input and identification of roadway features. *See* EX1009, UI 119; Summary. A POSITA would have viewed this as confirming that human operators could interact with roadway images and calibration data through a computing device. Applying Tal's interface paradigm to Westmacot's annotation system makes clear that users could not only adjust lane lines but also identify a new horizon line.

414. **Predictable Use of Prior Art Elements (KSR Rationale).** Extending user-modification functionality from lane lines to horizon lines represents nothing more than the predictable use of prior art elements according to their established functions. Both are geometric calibration references visualized in camera images, both are subject to automatic estimation with potential error, and both benefit from human-in-the-loop correction. Tal shows that computing devices were already being used to capture, display, and allow user interaction with roadway imagery,

confirming that adding horizon-line input to Westmacot's annotation system would have been straightforward. Accordingly, a POSITA would have had both motivation and a reasonable expectation of success in extending Westmacot's annotation interface to support user identification of horizon lines through a computing device.

J. Claim 7

415. Westmacot in view of Tal renders obvious claim 7.

1. Westmacot

416. Westmacot discloses nearly all aspects of claim 7, namely that camera parameters are recomputed when a user modifies a line annotation. Westmacot describes a "parameter computation component 408 [that] computes various 'reference parameters' ... [including] the height H of the camera ... the horizontal offset S ... the orientation of the camera ... captured as a 'forward point' ... and a 'horizon line' ... and additionally: the width W of the road/lane." EX1008, pp. 32-33, Fig. 4. It further explains that when lane boundaries (R1, R2) are adjusted by the user, the parameters are recomputed to maintain alignment: "a user can adjust the assumed lane width W to better fit one image, and because the road model is being adapted to the new lane width, the annotation data for other images can be automatically adapted based on the adapted model with the new lane width W." EX1008, p. 35. The disclosure also notes at the "human fixer" stage that camera

height, lateral offset, and lane width may not be “exactly correct” and require human correction, which directly results in recomputation of those parameters.

EX1009, p. 47.

417. What Westmacot does not expressly disclose is transmitting those recomputed parameters to the camera device itself. Tal fills this gap.

2. Tal

418. Tal teaches transmitting roadway imagery and processed data, including images with overlays, to and from a remote server over a network: “Digital images 16 are taken by one or more imaging devices 101 ... Only selected data portions 20 (e.g., image frames 16a,b,c,d) ... can be transmitted over the network 18 by the device 101 to a server 107 for subsequent processing/reporting ... It is also recognized that both resultant processed image data 20 and unprocessed images 16 can be transmitted to the server 107 by the device 101 over the network 18 ...” EX1009, ¶[0027]. Tal also states that “the device 101 transmits the data 20 ... to the server 107 (via the network 18 utilizing the cellular connection 106 or wireless connection 105, for example)” EX1009, ¶[0030], and that processed/overlaid images “can be ... sent separately” to the server EX1009, ¶[0052], ¶[0075]. Finally, Tal makes clear that data flows back as well: “The Device 101 then transmits data 1202 to the Server(s) 107 ... The data transmitted 1202 illustrates what typical incident data 12 contains ... including ... an Image: a

picture of the incident ...” EX1009, ¶¶ [0138]-[0146]; *see also* EX1009, Fig. 12, ¶[0009] (servers act as a gateway to users via web access).

3. KSR

419. A POSITA would have found it obvious to combine Westmacot’s disclosure of recomputing camera parameters when lines are modified with Tal’s explicit teaching of network-based transmission between a device and a server. The predictable result is that the recomputed camera parameters (height (H), offset (S), lane width (W), forward point, and horizon line) would be transmitted back to the camera device, closing the calibration loop. This is nothing more than using known elements (line modification–based recalibration in Westmacot; distributed network feedback in Tal) for their established functions, yielding the expected result of a self-calibrating system.

420. A POSITA would have been motivated to do so for similar reasons that I have discussed throughout:

421. **Calibration Feedback** – Westmacot’s parameter computation component 408 expressly calculates camera parameters such as height (H), offset (S), forward point, and horizon line based on user-modified lane geometry. EX1008, Fig. 4; parameter computation discussion. However, in Westmacot these parameters remain within the local processing pipeline (*e.g.*, model adaptation 410, rendering 416). A POSITA would have recognized that transmitting these

recomputed parameters back to the camera device itself provides direct functional benefits. For example, the camera device could embed the parameters as metadata in each captured image, ensuring downstream processes (such as perception or training) are informed of the current calibration state. Alternatively, the camera device could adjust its own capture settings, such as orientation corrections, region-of-interest cropping, or exposure adjustments, based on the updated calibration data. Tal confirms that image-related data is routinely transmitted over a network and back to devices, *e.g.*, “the device 101 transmits the data 20 ... to the server 107 (via the network 18 ... cellular connection 106 or wireless connection 105)” EX1009, ¶[0030], and that image data can flow back as part of incident reporting EX1009, ¶[0138]. Thus, returning calibration parameters to the camera device is not only feasible but a predictable way to improve image accuracy and integrity.

422. **Consistency Across Devices** – Tal expressly discloses that servers act as gateways to users via web access and client interfaces, EX1009, ¶[0012]-[0013]; Fig. 12, and that both processed and unprocessed image data may be transmitted across networks for centralized handling EX1009, ¶¶[0027], [0030], [0052], [0075]. This architecture supports distribution of calibration data across multiple devices. A POSITA would have understood that if recomputed parameters in Westmacot were transmitted back to the originating camera device, those same

parameters could also be propagated to other components in the system (*e.g.*, storage servers, training databases, or other vehicles in a fleet). Doing so ensures that all stakeholders, whether local devices, remote servers, or web-based users, operate on a consistent calibration baseline. In fleet-scale deployments, this prevents divergence between locally adjusted parameters and remotely stored models, thereby enhancing consistency, reproducibility, and reliability.

423. **Predictable Use of Known Techniques** – By the priority date, feedback loops where centrally computed calibration data was transmitted back to sensors were conventional. Integrating Westmacot’s recomputation with Tal’s transmission methods is a straightforward and predictable application of known design practices.

K. Claims 8-20

424. As I mentioned earlier, the only differences between claims 1–7 and claims 8–20 are the statutory formats (method vs. storage medium vs. device). A POSITA would have understood it to be obvious that the calibration functionality disclosed by Westmacot (annotating roadway images, overlaying and modifying lines, computing parameters) and Tal (transmitting annotated data over a network, including to web-based applications) could be implemented equally as (i) software instructions stored on a non-transitory medium (claims 8–14), or (ii) logic executed on a processor within a computing device (claims 15–20). Both Westmacot and Tal

expressly describe software-based implementations running on processors and storing data in memory. *See, e.g.*, EX1008, p. 23, processor 304 and memory 306 (“The processor 304 receives the captured images ... and stores them in a memory 306”); EX1009, ¶[0027], [0030], [0138] (device 101 with software executing on a mobile or embedded computer, transmitting image and annotation data to a server). Thus, recasting the same operations into “computer-readable medium” or “device” form involves no substantive difference, but merely reflects conventional statutory claiming techniques.

425. Accordingly, for the same reasons set forth with respect to claims 1–7, and because Westmacot in view of Tal renders obvious the claimed calibration operations regardless of format, claims 8–20 would have been obvious to a POSITA.

XII. OBJECTIVE INDICIA OF NONOBVIOUSNESS

426. In reaching my opinions regarding obviousness in this declaration, I also considered whether there were any objective indicia of nonobviousness. The basis for my understanding of obviousness and objective indicia of nonobviousness is provided above in Section III.

427. I understand that any objective indicia of nonobviousness must have a nexus, or a close connection, to the novel aspects of the claims. I understand that objective indicia of nonobviousness include the following: commercial success,

long-felt but unmet need, failure of others, copying, praise in the industry, and unexpectedly superior results as compared with the closest prior art.

428. Considering the information I have reviewed, I am not aware of any objective evidence that might support the patentability of the claims of the '276 patent. Further, from my review of the file history (EX1002) of the '276 patent, I did not see that the applicant raised any objective indicia of nonobviousness during prosecution. I am not otherwise aware of any publicly available evidence of objective indicia of nonobviousness.

429. Therefore, it is my opinion that no objective indicia of nonobviousness support the patentability of the claims of the '276 patent.

XIII. THE ART CITED IN THE GROUNDS ARE SUBSTANTIVELY DIFFERENT THAN AND NOT CUMULATIVE TO THE REFERENCES CITED DURING PROSECUTION.

430. During prosecution of the '276 patent, the Examiner rejected claims 21–40 over Kwant1 and Kwant2, and the Applicant distinguished the claims by arguing that the cited prior art did not disclose three key features: (1) visualization of the horizon line; (2) receiving a modification of a line to produce a new line at a second position; and (3) computing a camera parameter based on the new line. FH, pp. 137–39. The prior art cited in the Grounds, Choe, Davies, Westmacot, Tal and Kuehnle, disclose these features in ways that are substantively different from Kwant1 and Kwant2 and not cumulative.

431. First, the Applicant argued that Kwant1 did not disclose “any type of visualization of the horizon line” and that, at most, it used the horizon line as a reference for other annotations. EX1002, p. 137. In contrast, the references in the Grounds expressly disclose visualization. Choe teaches generating a “first image with the horizon line superimposed thereon” and updating the displayed image as the line is moved. EX1005, ¶[0016]; *see also* EX1005, FIGs. 6A–6B. Westmacot describes overlaying both lane lines and the “horizon line” directly on roadway images in its annotation interface. EX1008, p. 22, 44, Fig. 16. Davies explains that external computing devices may process captured image data to generate calibration outputs such as modified image information, which does not exclude visualized overlays. EX1007, ¶[0059]. Tal explicitly describes “[object] data 19,20 ... can be overlaid on the image” and transmitting that overlaid image to a remote server. EX1009, ¶[0052]. These disclosures are substantively different from Kwant1’s alleged indirect use of a horizon line as a geometric reference.

432. Second, the Applicant argued that Kwant2 did not disclose “receiving a modification of the line from the computing device, the modification comprising a new line at a second position.” EX1002, p. 138. Instead, Kwant2 described predicting a label and comparing it to a known label, which the Applicant argued was not a “modification.” EX1002, pp. 138–39. The references in the Grounds, however, explicitly disclose line modification workflows. Westmacot teaches an

annotation interface (UI 412) where a user can “shift” an existing boundary or add “new lanes” and other lines, and those modifications are propagated throughout the 3D road model. EX1008, 33:20–25; Figs. 6, 18–24. Choe discloses that a user can update the horizon line and lane lines in real time using keyboard, joystick, or voice inputs, with the image display updated accordingly. EX1005, ¶¶[0015]–[0016], [0055]–[0057]. Tal confirms that user input can be provided through computing devices with displays and interfaces for annotation and calibration. EX1009, ¶[0119], Fig. 12. These systems clearly disclose or contemplate receiving modifications to lines, including horizon lines, and generating new line positions.

433. Third, the Applicant argued that Kwant2 failed to disclose “computing a camera parameter based on the new line” because the only parameters in Kwant2 were internal weights and biases of a neural network, not camera parameters. FH, p. 139. In contrast, the references in the Grounds expressly compute camera parameters from line data. Choe discloses determining calibration factors for pitch, yaw, and roll angles of the camera, derived from horizon and lane line modifications. EX1005, ¶¶[0015], [0057], [0059]. Westmacot describes calculating the camera height (H), lateral offset (S), forward point, horizon line, lane width (W), and center position (C) based on detected or modified lines. EX1008, p. 26–30. Tal and Davies both describe transmission of image for adjustment and return to the vehicle.

434. Accordingly, Choe, Davies, Westmacot, Tal and Kuehnle are substantively different from and not cumulative to Kwant1 and Kwant2. Unlike the arguments raised against the prosecution art, these references explicitly disclose (i) visualization of the horizon line, (ii) receiving and applying user modifications of lines to produce new line positions, and (iii) computing camera parameters from those new lines. They directly address the specific elements Applicants relied on to distinguish the claims, and thus demonstrate that the claims of the '276 patent would have been obvious.

XIV. CONCLUSION

In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the allotted time.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Respectfully submitted,



Trevor J. Darrell, Ph.D.

Date: October 17, 2025

Appendix A

MATERIALS CONSIDERED

Exhibit	Description
1001	U.S. Patent No. 12,136,276 (Hassan)
1002	Prosecution History of U.S. Patent No. 12,136,276
1004	<i>Curriculum Vitae</i> of Dr. Trevor Darrell
1005	U.S. Publication No. 2020/0410704 (Choe)
1006	Prosecution History Figures of U.S. Publication No. 2020/0410704 to Choe
1007	U.S. Publication No. 2014/0240500 (Davies)
1008	International Publication No. WO2019/175286 (Westmacot)
1009	U.S. Publication No. 2022/0019829 (Tal)
1010	International Publication No. WO2009/027090 (Kuehnle)
1011	U.S. Publication No. 2019/0034740 (Kwant1)
1012	U.S. Publication No. 2019/0102674 (Kwant2)
1013	<i>Motive Techs., Inc. v. Samsara Inc.</i> , No. 24-cv-00902, Complaint (NDCA July 9, 2025)
1014	Russell, B.C. & Torralba, A., “Building a database of 3D scenes from user annotations,” <i>2009 IEEE Conference on Computer Vision and Pattern Recognition</i> , pp. 2711-2718 (2009)
1015	Bartl, V. & Herout, A., “Fully Automatic Horizon Estimation for Surveillance Cameras,” <i>2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)</i> , pp. 1-8 (2017)
1016	U.S. Patent No. 9,201,421 to Fairfield et al.
1017	Chinese Publication No. 112509054 to Chongqing, including certified English-language translation
1018	Geiger, A., et al., “Vision Meets Robotics: The KITTI Dataset,” <i>Int’l J. Robotics Res.</i> , 32(11):1231-1237 (Sept. 2013)
1019	Cordts, M., et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding,” <i>2016 IEEE Conf. on Computer Vision & Pattern Recognition (CVPR)</i> , 3213-3223 (2016)

Exhibit	Description
1020	Kato, S., et al., “Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems,” <i>Proc. 2018 ACM/IEEE Int’l Conf. on Cyber-Physical Systems</i> , pp. 287-296 (April 2018)
1021	U.S. Patent No. 8,769,396 to Chen et al.
1022	Graf, G., et al., “The Predictive Corridor: A Virtual Augmented Driving Assistance System for Teleoperated Autonomous Vehicles,” International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments, pp. 61-69 (2020)
1023	U.S. Patent No. 5,652,849 to Conway et al.
1024	U.S. Patent No. 9,775,682 to Quaid et al.
1025	U.S. Patent No. 9,916,703 to Levinson et al.
1026	Waymo Safety Report: On the Road to Fully Self-Driving (Feb. 2021)
1027	Huang, X., et al., “The ApolloScape Open Dataset for Autonomous Driving and Its Application to Network Benchmarking,” <i>Proc. IEEE CVPR Workshops</i> (2018)
1028	U.S. Patent No. 9,905,949 to Hartmann
1029	Saparia, S., et al., “Active Safety System for Semi-Autonomous Teleoperated Vehicles,” <i>IEEE</i> (2021)
1030	U.S. Patent No. 10,027,031 to Arai et al.
1031	Liu, S., et al., “Edge Computing for Autonomous Driving: Opportunities and Challenges,” <i>IEEE</i> , (99):1-20 (2019)
1032	U.S. Patent No. 9,792,569 to Ikawa
1033	Lee, J., et al., “Online Extrinsic Camera Calibration Using Lane Boundary Observations,” pp. 1-6 (2020)
1034	Lee, J., et al., “CTRL-C: Camera Calibration Transformer With Line-Classification,” <i>Proceedings of the IEEE/CVF Int’l Conf. on Computer Vision</i> , pp. 1-14 (2021)
1035	Andrés M. Lopez et al., <i>Deep Single Image Camera Calibration With Radial Distortion</i> , in <i>Proceedings of the IEEE/CVF Conf. on Computer Vision & Pattern Recognition</i> 9290 (2019)

Exhibit	Description
1036	Alex Davies, <i>Nissan’s Path to Self-Driving Cars? Humans in Call Centers</i> , WIRED (Jan. 5, 2017), https://www.wired.com/2017/01/nissans-self-driving-teleoperation/
1037	Aarian Marshall, <i>Self-Driving Cars Have a Secret Weapon: Remote Control</i> , WIRED (Feb. 1, 2018), https://www.wired.com/story/phantom-teleops/
1038	Andreas Geiger, Philip Lenz & Raquel Urtasun, <i>Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite</i> , in 2012 IEEE Conf. on Computer Vision & Pattern Recognition (CVPR) 3354 (2012), http://www.cvlibs.net/publications/Geiger2012CVPR.pdf
1039	<i>The KITTI Vision Benchmark Suite—Road/Lane Detection Evaluation</i> , CVlibs (dataset site), http://www.cvlibs.net/datasets/kitti/eval_road.php
1040	<i>The KITTI Vision Benchmark Suite</i> (home page), CVlibs, https://www.cvlibs.net/datasets/kitti/
1041	<i>The Cityscapes Dataset</i> , Cityscapes (dataset site), https://www.cityscapes-dataset.com/
1042	U.S. Dep’t of Transp., Nat’l Highway Traffic Safety Admin., <i>Assessment of Safety Standards for Automotive Electronic Control Systems</i> (2016), https://www.nhtsa.gov/sites/nhtsa.gov/files/812285_electronicreliabilityreport.pdf
1043	J. Brewer et al., <i>Functional Safety Assessment of a Generic Automated Lane Centering System and Related Foundational Vehicle Systems</i> (Volpe Nat’l Transp. Sys. Ctr. & NHTSA, Rep. No. DOT HS 812 572, Aug. 2018), https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13496_812572_alcsynthesis_080318.pdf
1044	Nat’l Highway Traffic Safety Admin., <i>Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts</i> (DOT HS 812 182, Aug. 2015), https://www.nhtsa.gov/sites/nhtsa.gov/files/812182_humanfactorseval-1213-automdrivingconcepts.pdf

Exhibit	Description
1045	Nat’l Highway Traffic Safety Admin., Human Factors Design Guidance for Level 2 and Level 3 Automated Driving Concepts (DOT HS 812 555, Aug. 2018), https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13494_812555_1213automationhfguidance.pdf
1046	U.S. Dep’t of Transp. & Nat’l Highway Traffic Safety Admin., Federal Automated Vehicles Policy: Accelerating the Next Revolution in Roadway Safety (Sept. 2016), https://www.transportation.gov/AV/federal-automated-vehicles-policy-september-2016
1047	Aarian Marshall, <i>Sacramento Eases Into the Self-Driving Scene</i> , WIRED (Aug. 1, 2018), https://www.wired.com/story/sacramento-phantom-auto-self-driving-car-partnership/
1048	Apollo, Dreamview Plus: README (2020), https://apollo.baidu.com/docs/apollo/9.x/md_modules_2dreamview_plus_2README.html
1049	Cruise, Webviz (2019), https://webviz.io ; https://web.archive.org/web/20191229152903/https://webviz.io/
1050	Cruise, Webviz GitHub Repository (2019), https://github.com/cruise-automation/webviz
1051	Kirsten Korosec, <i>Cruise Is Sharing Its Data Visualization Tool with Robotics Geeks Everywhere</i> , TechCrunch (June 18, 2019), https://techcrunch.com/2019/06/18/cruise-is-sharing-its-data-visualization-tool-with-robotics-geeks-everywhere
1052	Robot Web Tools, https://robotwebtools.github.io
1053	Amazon Web Servs., Inc., Amazon QuickSight—Fast, Easy Business Analytics for the Cloud, https://aws.amazon.com/quicksight ; https://web.archive.org/web/20151118211525/https://aws.amazon.com/quicksight/
1054	U.S. Patent No. 7,650,210 to Breed
1055	U.S. Publication No. 2013/0246135 to Wang
1056	U.S. Patent Application Publication No. 2017/0010106 to Shashua
1057	U.S. Patent No. 9,665,100 to Shashua

Exhibit	Description
1058	U.S. Patent Application Publication No. 2016/0214533 to Doyle
1059	U.S. Patent No. 8,612,136 to Levine
1060	Emami, Yousef, et al. “Human-in-the-loop machine learning for safe and ethical autonomous vehicles: Principles, challenges, and opportunities.” <i>arXiv preprint arXiv:2408.12548</i> (2024).
1061	Chen, Qi, et al. “F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds.” <i>Proceedings of the 4th ACM/IEEE Symposium on Edge Computing</i> . 2019.
1062	Xu, Jiaxuan, et al. “An automated learning-based procedure for large-scale vehicle dynamics modeling on baidu apollo platform.” <i>2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)</i> . IEEE, 2019.
1063	Jiang, Shu, et al. “DRF: A framework for high-accuracy autonomous driving vehicle modeling.” <i>arXiv preprint arXiv:2011.00646</i> (2020).
1064	Tang, Jie, et al. “LoPECS: A low-power edge computing system for real-time autonomous driving services.” <i>IEEE Access</i> 8 (2020): 30467-30479.