

Provisional Application for Patent Cover Sheet

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

Inventor(s)					
Inventor 1					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Wajahat		Kazmi	San Francisco	CA	US
Inventor 2					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Aamer		Zaheer	San Francisco	CA	US
Inventor 3					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Ali		Hassan	San Francisco	CA	US
Inventor 4					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Ahmed		Ali	San Francisco	CA	US
Inventor 5					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Hussam	Ullah	Khan	San Francisco	CA	US
Inventor 6					Remove
Given Name	Middle Name	Family Name	City	State	Country ;
Afsheen	Rafaqat	Ali	San Francisco	CA	US
All Inventors Must Be Listed – Additional Inventor Information blocks may be generated within this form by selecting the Add button.					
Title of Invention		DISTANCE ESTIMATION USING SINGLE VIEW CAMERA			
Attorney Docket Number (if applicable)		189908.100002			
Correspondence Address					
Direct all correspondence to (select one):					
☒ The address corresponding to Customer Number			☐ Firm or Individual Name		

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Customer Number	33717
-----------------	-------

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.
No.
Yes, the invention was made by an agency of the United States Government. The U.S. Government agency name is:
Yes, the invention was under a contract with an agency of the United States Government. The name of the U.S. Government agency and Government contract number are:

Entity Status

Applicant asserts small entity status under 37 CFR 1.27 or applicant certifies micro entity status under 37 CFR 1.29

- ☐ Applicant asserts small entity status under 37 CFR 1.27
- ☐ Applicant certifies micro entity status under 37 CFR 1.29. Applicant must attach form PTO/SB/15A or B or equivalent.
- ☒ No

Warning

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Signature

Please see 37 CFR 1.4(d) for the form of the signature.

Signature	Vivek Ganti/			Date (YYYY-MM-DD)	2020-02-12
First Name	Vivek	Last Name	Ganti	Registration Number (If appropriate)	71368

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **This form can only be used when in conjunction with EFS-Web. If this form is mailed to the USPTO, it may cause delays in handling the provisional application.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that : (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

DISTANCE ESTIMATION USING SINGLE VIEW CAMERA

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] Embodiments relate to analyzing image data to estimate relative distances of proximate objects.

2. Description of the Related Art

[0002] Multiple cameras or sensors may be installed in a vehicle to detect proximate objects. However, the use of multiple cameras may be expensive. Sensors such as RADAR and LiDAR may also be used as proximity sensors but such sensors may also be costly. In addition, a built-in camera may be used to detect the relative distance to proximate objects, however this may require the location of the camera to be predefined or preset.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The present disclosure relates to using a single view camera (e.g., an image sensor) that may be retrofitted and adjustable within the vehicle such as, for example, the vehicle's dashboard. Upon driving, the camera initializes and determines its height with respect to the ground. Thus, the camera may be installed at various heights and later adjusted. The height estimation is determined based on performing lane boundary calculations and using known and standardized lane widths. In this respect the single view approach exploits geometric features of the road for distance estimation.

[0004] Embodiments are directed to using cross-ratios of lane boundaries to estimate horizon. Embodiments are directed to determining an Inverse Perspective Mapping (IPM) and camera height from known lane width and a detected horizon. Distances of the vehicles on the road may be calculated by back projecting image point to a ray and intersecting that ray with the reconstructed road plane.

[0005] The present disclosure is directed to an efficient and reliable distance estimation framework that is not known to be present in ADAS (Advanced Driver Assistance Systems) literature. For example, single view cameras (e.g., a dashboard camera) may estimate distances to vehicles in real-time without the use of expensive sensor systems (e.g., RADAR, LiDAR) and without continuous use of resource intensive software like artificial intelligence and deep learning algorithms.

[0006] Embodiments of the present disclosure relate using simple dashcam images as input followed by an algorithm that calculates distance in real-time. For example, the images may include dashcam color (RGB) images. The camera maybe installed inside the driver's cabin, near the roof, looking forward onto the road. As soon as the images are acquired, the algorithm processes and finds the distance with respect to lead vehicle(s).

[0007] In some embodiments, a first part of the algorithm comprises a lane line detection procedure. In some embodiments, deep learning based lane line detection algorithms is executed upon initialization and is not executed after initialization. During initialization, the lane lines (lane boundaries) are detected and then using a geometric relationship of the lane lines (parallelism), the

horizon and thereby a transformation is estimated which can remove perspective distortion from images (find an Inverse Perspective Mapping).

[0008] Predefined real-world lane width may then used to find the camera height. The estimation of camera height completes the initialization phase. Camera initialization holds as long as the camera is not displaced from its position which is less likely since it is held in place in an rigid assembly. Thus, the initialization phase estimates camera height in the field while the vehicle is moving on a road with lane lines. The camera height may be stored for subsequent operations. In some embodiments, initialization may occur within less than a minute.

[0009] Once initialized, distance of the lead vehicles can be estimated in real-time on any onboard device using various arithmetic calculations. Embodiments of the present disclosure are described in Appendices A and B, attached hereto.

Lane detection using lane boundary marker network with road geometry constraints

Hussam Ullah Khan, Afsheen Rafaqat Ali, Ali Hassan, Ahmed Ali, Wajahat Kazmi, Aamer Zaheer
KeepTruckin Inc. (RnD)
Lahore, Pakistan

{hussam.khan, afsheen, ali.hassan, ahmed.ali, wajahat.kazmi, az} @keeptruckin.com

Abstract

Lane detection is of critical importance to both the self-driving cars as well as advanced driver assistance systems. While current methods use a range of features from low-level to deep features extracted from convolutional neural networks, they all suffer from the problem of occlusion and struggle to detect lanes with low or no evidence on the road. In this paper, we use a lane boundary marker network to detect keypoints along the lane boundaries. An inverse perspective mapping is estimated using road geometry which is then applied to the detected markers and lines/curves are fitted jointly on the rectified points. Finally, missing lane boundaries are predicted using lane geometry constraints i.e., equidistant and parallelism. Reciprocal weighted averaging ensures lane boundaries with strong evidence dominate their predicted alternatives. The results show a significant improvement of +7.8%, +6.8% and +1.2% of F1 scores over the state-of-the-art on CULane, Caltech and TuSimple datasets, respectively. This proves our algorithm's robustness against both occluded and missing lanes cases. Furthermore, we also show that our algorithm can be combined with other lane detectors to improve their lane retrieval potential.

1. Introduction

Fully autonomous vehicles in the form of self-driving cars are gradually appearing on the scene and we have Google Waymo [17], Tesla's Auto-pilot [19], Comma.AI's OpenPilot kit [16] and Intel's MobileEye solutions [18] just to name a few. Along with self-driving cars, Advanced Driver Assistance Systems (ADAS) have also captured a surge in the interest. Both self-driving cars and the ADAS require different levels of autonomy but some aspects of automation are still common. One of them is road lane detection and segmentation. Lane detection is critical in identifying and ensuring safe driving practices and an on-board

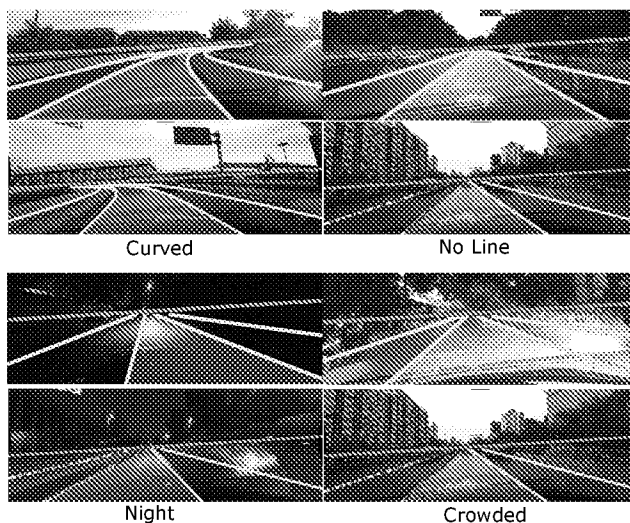


Figure 1. Sample results of our algorithm on examples from four different classes of CULane dataset [33] are shown here. Cyan lines are the detected lane boundaries, green region represents the ego lane and magenta line displays the estimated horizon. In the *No Line* class, there is actually no line markings on the road but the ground truth carries the lines shown.

system with this capability can alert the driver in case of an unsafe lane change and avoid it in the first place if it is a self-driving vehicle. Besides, it is also necessary in understanding the road layout and surrounding scene geometry for proper route planning for autonomous or semi-autonomous vehicles.

Lane detection has classically been done using either low-level edge features, histogram analysis, lane boundary detection algorithms such as Linear Hough Transform [15] or a combination. Although such methods are intuitive and efficient, they are however easily fooled by non-lane edges or lines appearing in the images such as road curbs, shadows as well as false negatives due to erased or inconsistent lane boundary markings. Recently, deep learning based solutions have also been proposed that produce more robust

and reliable results as they tend to learn the context of the road scene irrespective of occlusions. The advantage of using deep learning (DL) techniques lies in the discourse from traditional lane boundary detection approach towards semantic segmentation of the lane region [31]. Several state-of-the-art algorithms such as VPGNet [24], SCNN [33] and JLBNet [40] have shown impressive results and have set the bar very high. But there are challenging scenarios in which even DL based algorithms struggle to find the lane. One of the obvious cases is when the evidence of a lane line is either effaced or there is no lane marking on the road but the ground truth (GT) carries lane information such as CU-Lane's [33] *no line class*; examples are shown in Figure 1.

To address such cases, we proposed a novel lane detection algorithm that exploits the parallel nature of lane boundaries and fixed lane widths on the road to validate the detected lanes and predict lanes that are missed by the detector due to erased or effaced lane markings. In general, these assumptions apply on all through lanes that typically constitute the major length of the highway and our main focus in this study was to evaluate the effectiveness of the proposed approach in such scenarios. Although, the assumption of parallel lanes has been successfully exploited in earlier works to detect lanes in rectified view [5] but to the best of our knowledge, we are the first ones who used equidistant and parallelism property of the lanes jointly to fit the lanes and our results on *no line class* of CUlane dataset highlight the effectiveness of this approach.

We use a simple encoder-decoder based Convolutional Neural Network (CNN) to detect keypoints or markers (used interchangeably) along the lane boundaries. Inverse Perspective Mapping (IPM) estimated using scene geometry is then applied to the detected markers and lines/curves are fitted on the rectified points. The estimation of IPM is a one-off calculation for any given camera which is done over a small set of initial frames. Unlike earlier approaches that used single lane evidence to fit a curve for each lane line we exploited road geometry and jointly fitted lines/curves for all lanes. Our approach significantly improved the results on public datasets, specifically for crowded or missing lane scenarios. Lane region segmentation would descend from lane boundary detection as a boundary polygon. The sample results of our lane detection algorithm on four different classes of CUlane dataset, shown in Figure 1, demonstrate the efficacy of our approach.

The breakdown of the paper is as follows: Section 2 provides a quick review of currently explored lane detection techniques. Section 3 discusses our approach in detail with our keypoint detector in section 3.1 and the estimation of horizon and IPM in section 3.2.1 and 3.2.2. Our main contribution, the parallel lane fitting along with the missing lane prediction algorithm, is discussed in sections 3.3.1 and 3.3.2 respectively. Section 4 outlines the datasets, training and

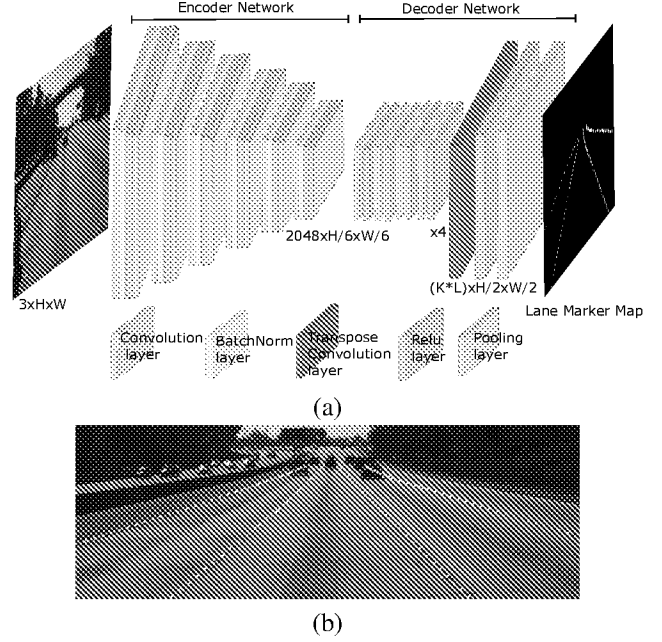


Figure 2. (a) The figure shows the architecture of the lane boundary marker network. (b) The sampled keypoints from the ground truth lane line are shown here.

testing frame work. Results are presented in sections 4.1, 4.2 and 4.3. Section 5 concludes the paper.

2. Related Work

In literature, the problem of lane detection has been addressed through a variety of sensor data. Taking into account the variation of ambient light owing to the time of the day and weather which may render passive color imaging useless, active sensors such as radars and lidars (Light/Laser Detection and Ranging) have also been explored [21, 9, 26]. With this approach, generally the white/colored lane boundary is identified in the range point cloud as having a different reflectance w.r.t the road surface. The identified lane boundary is then used to fit a line model on top of it. But this is not a very popular approach since the Lidar can be expensive with complex installation along with the challenges posed by road surface conditions such as the fading color of the lane boundary over time which is quite common across the globe.

The most straightforward approach, therefore, is using a forward looking monocular dashcam installed inside the driver cabin. Generally, images thus acquired are processed for low-level features. Veit *et al.* [37] presented a comprehensive overview of features used to detect road markings up till 2008. Most commonly, edge detection filters such as Canny and Sobel [32, 20, 38] were employed for detecting lane boundaries followed by model based approaches such as Hough Transform [6]. Bertozzi *et al.* used Inverse Perspective Mapping to rectify the image removing the per-

spective distortion to improve lane detection by employing parallel line constraint of the road lane boundaries [5]. But under all the constraints, these methods still suffered from occlusion and the false negatives generated by strong shadows. Vanishing Point (VP) constraint was imposed in [29, 6] to reduce the false positive rate. Shang *et al.* [34] used steerable filters to guide the edge orientation of the lane boundary markers and introduced a VP guided Parallel Hough transform which showed improved performance under complex lighting and weather conditions. To sum up the handcrafted features and model based lane boundary extraction, Narote *et al.* [30] compiled a very recent and a comprehensive literature review. The prime limitations in these algorithms remain illumination invariance and occlusion.

With the advent of deep learning for image classification [23] the landscape of algorithms quickly changed and supervised CNNs snatched the top slots in tasks related to image classification [23], semantic segmentation [27] and object detection [12]. Lane detection, similarly, also witnessed a significant improvement with deep CNNs [35, 10, 22, 7, 28]. Pan *et al.* [33] used Spatial CNNs to detect lane boundaries as line segments. They employ a cascade of convolutional kernels across the height, width as well as the depth of the input images. Using a sequential propagation scheme to integrate information from every pixel, they reduce the memory and computational burden in contrast to dense Markov Random Field (MRFs), Conditional Random Fields (CRFs) and Recurrent Neural Networks (RNNs). The output is a class specific probability map of lane boundaries which are then sampled horizontally every 20 rows to produce a set of points. The cubic splines are fitted on sampled points to estimate lane boundary curves. Their algorithm is the top performing one at TuSimple benchmark [36] (accuracy 96.53%). They also introduced their dataset (CULane) and reported 71.6% F1 measure on its test set.

Lee *et al.* [24] used a Vanishing Point Guided network (VPGNet) by annotating VP in their training data. The VP constraint improves the lane detection as previously found by Changzheng *et al.* [6] but their data is not publicly available. Recent works, such as LaneNet by Neven *et al.* [31] introduced a CNN for lane boundary segmentation. They cluster class-specific semantic segmentation of the lane boundaries into the available number of lanes. In contrast to previous approaches such as [5, 24] that estimate IPM with the assumption of a fixed camera pose, Neven *et al.* learned a small network, HNet, to estimate it at run time. On TuSimple dataset their algorithm performs almost as good as VPGNet (accuracy 96.4%) but its performance is limited by the fact that deep learning approaches have inherent dependence on training data domain, therefore, change in camera hardware, pose and scene will greatly influence prediction of HNet and hence the outcome.

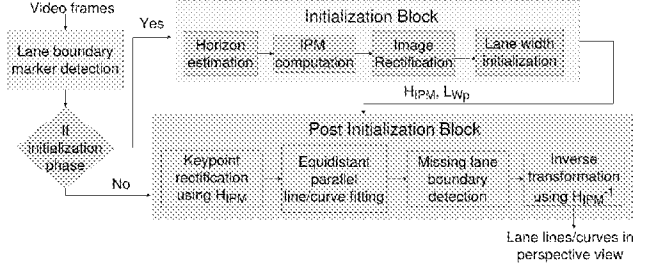


Figure 3. The block diagram of our end2end lane detection algorithm is shown here.

Deep learning is specifically good at learning context and semantic information of objects in the scene. Lane regions occupy more area than lane boundaries and hence are semantically strong contenders. With state-of-the-art performance in segmentation through Fully Convolutional Networks (FCN) [27] and Segmentation Networks (SegNets) [4], lane region segmentation has also been applied, but with not much success. A lane occupies a large region in the image and identifying all the pixels correctly, especially the boundary pixels becomes challenging. Generally, it is difficult to weight the training loss such that it penalizes the network more if border pixels get misclassified as they are more crucial in defining the lane boundary. Therefore, Zhang *et al.* [40] integrated both lane region segmentation and boundary pixel detection into two separate networks and combined them into a multi-task loss. The algorithm exceeded the performance of [33] with a total F1 measure of 73.1% on CULane. Recently, Hou *et al.* introduced Self Attention Distillation networks for lane detection that vary in performance (6 to 74 fps) and accuracy depending on the network architecture [14].

3. Our Approach

We believe that lane boundary detection can be generalized easily on a rather small dataset as compared to lane region. Tying boundary detection to region segmentation is not much helpful as is obvious from the reported F1 scores (71.6%) of Pan *et al.* [33] who used pure lane boundary detection contrast to Zhang *et al.* [40] (F1 73.6%) who complemented lane boundary and region detection in a joint framework on CULane dataset. Therefore, we focused only on lane boundaries. As generally is the case with deep learning methods for lane detection that output lane boundary maps which are sampled to produce boundary markers, we instead take motivation from keypoint detection in Mask-RCNN [12] and predict these lane markers directly instead of sampling them from segmentation maps. The main contribution of this paper is the post processing pipeline, shown in Figure 3, following the lane boundary markers detection. We fit lines to predicted keypoints and initialize lane width and camera rotation using the equidistant

tant and parallelism properties of the road lanes. At inference, if we are able to fit two lane boundaries (lines/curves) jointly using the initialized lane width within a reasonable tolerance (30%) and parallelism, only then we predict any missing lane boundaries and handle multiple predictions through weighted averaging. This pipeline greatly improves the lane detection accuracy and qualitative results are shown in Figure 1.

3.1. Lane Boundary Keypoint Detection

We represent each lane boundary using a fixed number of keypoints. The accurate estimation of horizon (which is critical for determining the IPM) is dependent upon the correct localization of Vanishing Point (VP). Therefore, during training, we sample more markers near the VP. Another reason for this is that points near the VP represent larger distance in the real world, so accurate localization of these points is crucial for precise line/curve fitting. The selected distribution of these markers along lane boundary on a sample TuSimple image is shown in Figure 2(b). To sample keypoints, we fit a cubic spline curve on the ground truth points for each lane boundary and then divide it vertically into three equal segments, where 50% of the total markers are sampled from a segment that is closer to the VP and $1/4^{th}$ of the total markers from the rest of the two segments each.

Inspired by the well known image segmentation networks (i.e. FCN [27] and SegNets [4]), our lane detection network is fully convolutional and consists of two blocks: an encoder block, responsible to encode all relevant low and high level features from the input image, and a decoder block that learns the relationship between provided low and high level features and makes the decision on top of it. The architecture diagram of our keypoint detector is shown in Figure 2(a). We used ResNet50 architecture [13] as backbone for information encoding while our decoder network consists of four convolution layers, a single transpose convolution layer, that upsamples the feature map by $4\times$ for precise localization of lane markers, and a final output convolution layer. In the output layer, we generate a one-hot mask for each marker just like Mask-RCNN, where only the marker's pixel is marked as foreground or one. To avoid inter-class competition among multiple keypoints, a per-mask cross-entropy loss is computed. The total loss of this network is the average of the loss for all K keypoints. At inference time, unique marker in each output map is computed by selecting the one with the highest score.

3.2. Camera Initialization

Camera initialization involves using camera intrinsics and road geometry to find Inverse Perspective Mapping (IPM) to synthetically rotate the viewpoint and lane width to fit equidistant parallel lanes in rectified view after the ini-

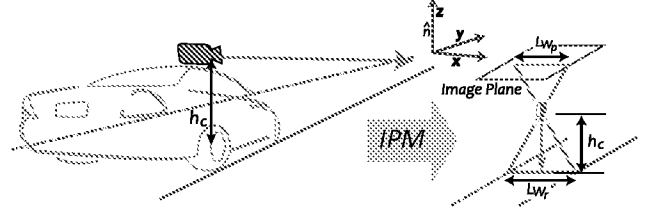


Figure 4. (Left) Perspective lane image demonstrating the right-hand camera system (Right) rectified camera view w.r.t road surface using IPM. h_c is the camera height, L_{Wr} is the lane width in real world in meters while L_{Wp} is the lane width on image plane in pixels given by Equation 1.

tialization.

Our approach is aimed at a forward looking dashcam installed inside the driver's cabin. Configuration is done in a right-handed frame of reference. The road plane normal is along the Z-axis (upwards), Y-axis is in the driving direction and X-axis towards the right side of the vehicle. As shown in Figure 4(a), the camera is expected to have some tilt w.r.t X-axis whereas the pan angle w.r.t Y-axis is negligible and therefore, the image mainly has perspective distortion [25].

IPM is used to remove the perspective distortion in an image. Application of the IPM synthetically rotates the camera into a top (bird's eye) view thus rendering the intersecting lines as parallel; the way they actually are in the real world. The use of IPM improves the curve fitting as explained in earlier works such as [5, 24]. Recently, Neven *et al.* [31] trained a small CNN (HNet) to estimate IPM but its performance is limited by the fact that deep learning approaches have inherent dependence on training data domain, therefore, change in camera hardware, pose and scene will greatly influence the outcome. On the other hand, for a geometric approach, knowing camera pose could be challenging as it may vary from time to time. Therefore, we introduce a novel, accurate and robust way of estimating an IPM without any prior knowledge of the camera pose. We instead use road geometry to find camera pose and in-turn find the IPM. The following subsections explain all modules in the *initialization block* of the Figure 3.

3.2.1 Horizon Estimation

The estimation of IPM assumes a planar road scene having no elevation and more than one lanes (3 or more lane boundaries), such as a typical highway scenario. We start by fitting lines to the detected markers. A best fit line must have a minimum number of inlier markers (i.e., 25% of total number of markers for each lane boundary). Inlier markers are those having horizontal distance of under 8 pixels to the best 1st degree line fit through RANSAC [8].

Critical part of initialization is the estimation of a vanishing line (horizon) from these lines. This requires two sets of vanishing points (VPs), forward and lateral VPs. Forward

VPs can be found from the cross product of the world parallel lines whereas the process of finding lateral vanishing point is a bit involved. For that we use cross-ratios of three real-world parallel lines which are the lane lines as used by Ali *et al.* [2]. Once a set of forward and lateral VPs is found, we use their horizon estimation algorithm to find the best fitting horizon l_h .

3.2.2 IPM and Image Rectification

After estimating horizon, we compute plane normal $\hat{n}$ using camera intrinsic matrix K and horizon l_h [11] by $\hat{n} = K^T l_h$. The IPM is then computed using $H_{IPM} = K R K^{-1}$, where $R = [l_h \times \hat{n}; (l_h \times \hat{n}) \times -\hat{n}; -\hat{n}]$ is the rotation matrix and H_{IPM} is the rectification homography which will rotate the camera view to align its Z-axis with the road's normal.

Estimation of horizon and IPM on a short video clip completes the initialization of camera. Frames can then be simply rectified into a bird's eye view by applying this IPM (Figure 4). As long as the camera intrinsics or pose do not change, re-initialization is not required. Therefore, it is a one-off calculation for every camera setting.

3.2.3 Lane Width Initialization

A frame is considered an inlier frame if it has at least one forward and one lateral vanishing point. A pair of consecutive parallel lines are selected and the distance between the lines is calculated per frame through:

$$L_{Wp} = \frac{\sum_f \frac{|c_2 - c_1|}{\sqrt{a^2 + b^2}}}{f} \quad (1)$$

where f is the number of inlier frames and L_{Wp} is the average initialized lane width in pixels.

In real world, standard lane width typically vary across countries and is usually decided by state or federal department of transportation. Lane widths are commonly narrower on low volume roads and wider on higher volume roads and depends on the assumed maximum vehicle width. The Interstate Highway standards for the U.S. Interstate Highway System use a 3.7 meter standard lane width while in Europe, the minimum widths of lanes are generally between 2.5 to 3.25 meters [39]. Hence the ratios of these lanes widths vary within nearly 0.7 to 1.3. Therefore, an empirically selected threshold on lane width ratio (initialized:detected) $L_{WR} = 0.3$ should suffice.

3.3. Post-Initialization

In the post initialization phase, we use the computed homography and estimated lane width to rectify keypoints and fit equidistant parallel lines/curves on them. Following subsections explain all the modules shown in post-initialization block of Figure 3.

3.3.1 Equidistant Parallel Line/Curve Fitting

The lane boundary markers detected by CNN are rectified using the H_{IPM} found in Section 3.2.2. Second degree quadratic curves are fitted to randomly selected subset of markers. Three constraints are imposed on the curves in order to ensure correctness. First, the number of inliers, which have to be more than a minimum numbers (generally 25% of the total number of markers for lines and 50% for curves). For this, the perpendicular distance between the curve and a given marker is used as a measure. This process is repeated until the number of inliers cross a threshold or maximum iterations are reached. If it fails, then straight lines are tried.

After line/curve fitting, the second and the third constraints, parallelism and equidistant nature of lane lines, are ensured. If the two lines are straight, the parallelism is estimated from the difference between their slopes and the distance between them is computed using the Equation 1. In case of curved lane boundaries, the difference between the slopes of their tangents is used to judge parallelism and the lane width is calculated by intersecting a line perpendicular to the tangent of first curve to a point at the second curve and then finding the distance between the two points.

In its current form, our lane detector is trained for four lane boundaries (three lanes, i.e. left, center and right). If four lane boundaries do not satisfy all the three constraints, three lane boundaries are fitted and then two. At less than two lines/curves, the frame is ignored. Algorithm 1 explains this process for two lane boundaries.

3.3.2 Detecting Weak or Missing Lane Boundaries

In order to estimate the missing lanes, we used the initialized lane width in pixels (L_{Wp}). We calculate the lane boundary offset for the missing or low confidence lanes from Equation 1 as:

$$c_o = L_{Wp} \sqrt{a^2 + b^2} \quad (2)$$

where c_o is the required offset in x-intercept for the missing lines. Adding c_o to the x-intercept of the detected lane boundaries in the rectified view, we predict the missing lane boundaries. For example, for a detected first lane's first boundary, we predict the number of missing lane boundaries on each side, i.e. $x = -\frac{b}{a}y + (c + c_o)$, $x = -\frac{b}{a}y + (c - c_o)$ (see Figure 5 (a)).

After the missing lane boundary prediction, we perform a reciprocal weighted average of their x-coordinates near the forward VP and bottom of the image. If there are two predicted boundaries on the right side of a initial lane boundary as shown in Figure 5 (a), we use:

$$x'_1 = \frac{(w'x_1 + wx'_1)}{w + w'}, \quad x'_2 = \frac{(w'x_2 + wx'_2)}{w + w'} \quad (3)$$

where the weights w, w', w'' are $\{0,1,2\}$ respectively and are selected, based on the distance of the predicted

Algorithm 1: Lane boundary detection algorithm

Input: Detected markers from lane detector for a pair of consecutive lanes boundaries, $fit2$

Output: Detected lane boundaries

```
1 initialize;
2  $inliers_{tot} = 0$ 
3 Fit lines to lane boundaries:
4 Set  $M = \min$  num of inliers;
5 while  $iter \leq MAX\_ITER$  do
6   Set  $inliers_1 = 0$ ;
7   Set  $inliers_2 = 0$ ;
8   Rectify lane markers using  $H_{IPM}$ ;
9   Randomly sample  $n$  markers per line;
10  if  $fit2 == 1$  then
11    Fit  $2^{nd}$  degree curve (Section 3.3.1);
12  else
13    Fit line;
14  end
15 end
16 Count  $inliers_1$  and  $inliers_2$  (Section 3.3.1);
17 Check Constraint1:
18 if  $inliers_1 \geq M \ \& \ inliers_2 \geq M$  then
19   Get slopes  $m_1 = |a_1/b_1|, m_2 = |a_2/b_2|$  (from
    Equation  $a_n x + b_n y + c_n = 0$ );
20   Get detected lane width  $L_{WD}$  (Section 3.2.3);
21   Read Initialized lane width  $L_{Wp}$  (Equation 1);
22   Check Constraints 2 and 3:
23   if  $|m_1 - m_2| \leq 0.1 \ \& \ |1 - \frac{L_{WD}}{L_{Wp}}| \leq L_{WR} \ \&$ 
     $inliers_{tot}^n > inliers_{tot}^{n-1}$  then
24     Set  $lane\_detected = True$ ;
25     Update  $inliers_{tot} = inliers_1 + inliers_2$ ;
26   end
27 end
28 end
```

lane boundary from detected lane boundary. Predicted lane boundary that is one lane width distance away is assigned a weight of 1, detected lane boundary is given a weight of 0 and so on. This enables robust retrieval of missing lane boundaries.

4. Evaluations and Discussion

We evaluated our algorithm on a number of publicly available lane datasets, namely, CULane [33], TuSimple [36] and Caltech [3]. In order to match lane boundaries with the ground truth we followed the TuSimple's criteria [1] for all datasets which allows prediction of up to two extra lane boundaries. The computational times listed in Table 1 were noted with GPU: GTX-1080 Ti 11 GB RAM and CPU: Core i7 6900K 3.20 GHz, except for the SAD-Nets [14] which were computed on GTX Titan X GPU by their authors.

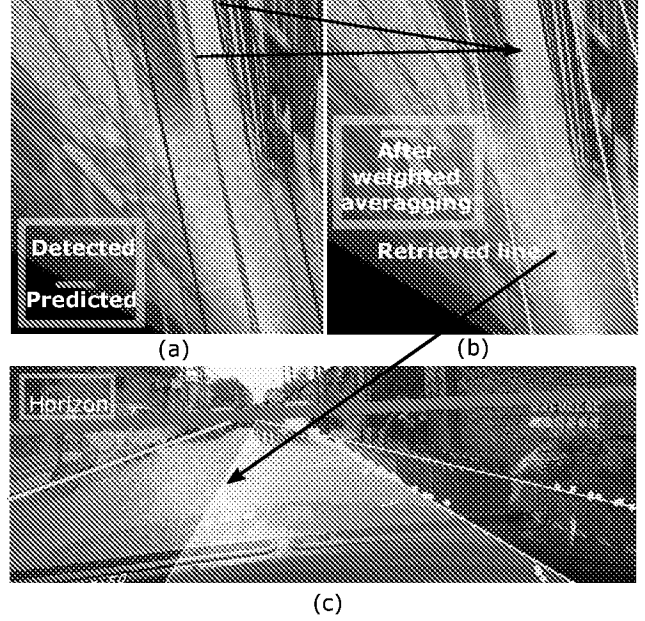


Figure 5. Detecting missing lane boundaries. (a) Rectified view. Lane boundaries are predicted in red using c_o from section 3.3.2. (b) Filtered lane boundaries after weighted averaging (c) Recovered perspective view with all the four lane boundaries.

4.1. Results on CULane

CULane dataset consists of 55 hours of traffic video providing a total of 133,235 images. The training set contains 88,880 images, while the validation and test sets contain 9,675 and 34,680 each respectively. Depending on the scene categories, the test set is divided into eight subsets. Our lane marker detector was trained on the training set. We created lane boundaries in prediction and ground truth masks each of 30 pixels as used by [33] and [40], unless stated otherwise. Our results on the CULane test set and their comparison is given in Table 1.

As can be observed in the table, our method $Ours_{50}$ outperformed the state-of-the-art in lane boundary detection i.e. JLBNet [40], SCNN [33] and SAD-Nets [14] by an order of magnitude. In the *Normal*, *Crowded*, *Night*, *Arrow* and *Dazzle Light* categories, we had an improvement in F1 scores from 4 to 12% while in the *No Line* and *Shadow* categories, there had been an increase of almost 25% and 17% respectively. One odd case though was the *Curve* category in which we performed slightly better than SCNN but 3% lower than JLBNet and 5 to 6% lower than SAD-Nets. We will discuss the cause in the ablation studies. But we were able to improve the overall performance from 73.1% [40] and 72% SCNN [33] to almost 81%.

4.1.1 Ablation Studies

For the sake of ablation studies, we modified the lane boundary keypoint detector's architecture by changing the

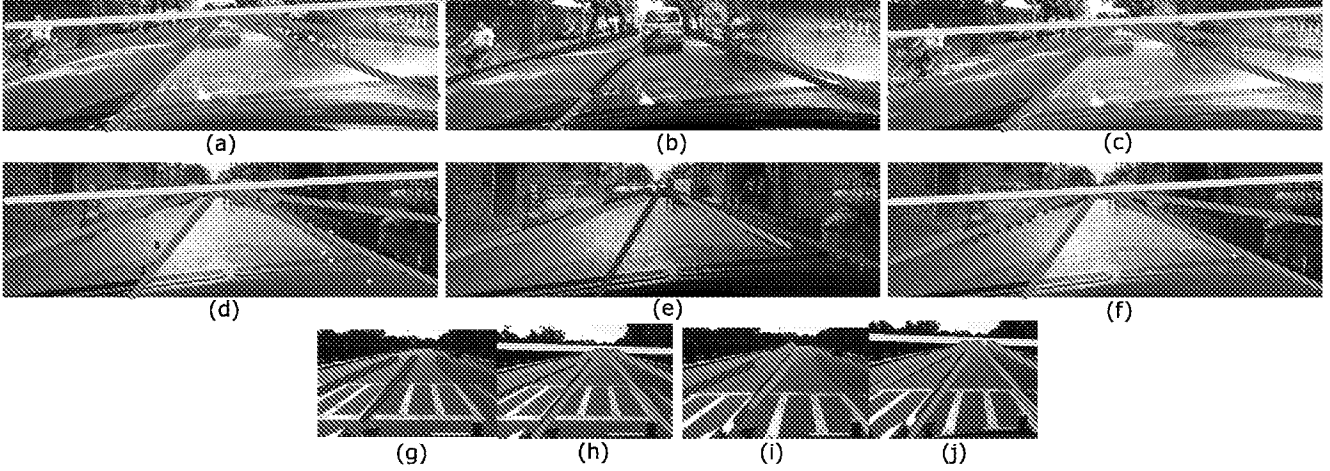


Figure 6. Results comparison (CULane samples: a to f, CalTech samples g to j). Detections in red and ground truth in blue and horizon line in green (a,d) Ours₄₀ (b,e) SCNN (c,f) SCNN lane boundary map + Ours_{pp} post processing. (g,i) SCNN trained on CULane tested on Caltech. (h,j) Ours₄₀ trained on CULane and tested on CalTech.

Method	Normal	Crowded	Night	No Line	Shadow	Arrow	Dazzle Light	Curve	Total	FPS
SCNN[33]	0.883	0.753	0.686	0.365	0.593	0.821	0.531	0.594	0.72	9
JLBNNet[40]	0.897	0.765	0.687	0.351	0.655	0.822	0.674	0.632	0.731	—
ENet-SAD[14]	0.901	0.688	0.660	0.416	0.659	0.840	0.602	0.657	0.708	74
R-18-SAD[14]	0.898	0.681	0.642	0.425	0.675	0.839	0.598	0.655	0.705	40
R-34-SAD[14]	0.899	0.685	0.646	0.422	0.677	0.838	0.599	0.660	0.707	20
R-101-SAD[14]	0.907	0.700	0.663	0.435	0.67	0.844	0.599	0.657	0.718	6
Ours ₂₀	0.925	0.790	0.753	0.564	0.782	0.898	0.738	0.572	0.796	4.8
Ours ₃₀	0.935	0.812	0.760	0.586	0.822	0.910	0.746	0.587	0.804	4.5
Ours ₄₀	0.934	0.808	0.762	0.586	0.794	0.911	0.745	0.604	0.806	7.8
Ours _{40c4}	0.931	0.801	0.759	0.581	0.799	0.911	0.747	0.573	0.802	4.2
Ours ₅₀	0.936	0.815	0.769	0.592	0.826	0.915	0.751	0.607	0.809	3.7
SCNN _{0.3,18} + Ours _{pp}	0.898	0.736	0.722	0.395	0.645	0.839	0.626	<u>0.476</u>	0.7500	4.3
Diff. with SCNN (%)	+1.48	-1.71	+3.58	+3.0	+5.19	+1.76	+9.54	-11.79	+3.09	—
SCNN _{0.2,40} + Ours _{pp}	0.930	0.797	0.775	0.556	0.803	0.902	0.735	<u>0.572</u>	0.808	3.8
Diff. with SCNN (%)	+4.73	+4.38	+8.9	+19.11	+21.03	+8.13	+20.4	-2.25	+8.94	—

Table 1. F1 score for lane boundary detection on CULane [33]. Both Ground Truth (GT) and prediction were mapped to 30 pixel wide masks with IoU threshold of 0.5 or higher for True Positives. The lane detector used ResNet50 backbone. Ours₂₀: Conv₅ → Deconv block and 20 markers/boundary. Ours₃₀: Conv₅ → Deconv block and 30 markers/boundary and so on. Ours_{40c4}: Conv₄ → Deconv block and 40 markers/boundary. F1 scores of SCNN and JLBNNet were copied from JLBNNet’s paper [40] while for SAD-Nets from their paper [14].

total number of markers/boundary as well as the depth of the backbone to observe the effect on the lane detection results. We varied the number of markers/boundary to 20, 30 and 40 as well. Although the best results were achieved by Ours₅₀ (subscript represent the number of markers/boundary) as shown in Table 1, Ours₄₀ and Ours₃₀ also closely follow. These results show that reduction or increment in the number of points within the range 30 to 50 had no significant effect on the F1 scores but just a subtle improvement is observed with increasing the number of markers. We also reduced the depth of backbone and attached the decoder after the fourth block of ResNet architecture instead of the last block. This reduced

the efficiency due to the removal of one max-pooling layer which increased the size of the output feature map while also marginally reducing the F1 scores. Ours₂₀ showed some reduction in overall F1 score (still higher than SCNN, JLB and SAD-Nets). This implies that with reduction in the number of markers below 30, our lane detection algorithm started to struggle. Ours₃₀ has the best trade-off between accuracy and efficiency. Please note that our algorithm was not optimized for efficiency and therefore, the FPS in the table is primarily for reference.

SCNN with Our Lane Prediction Algorithm

We also experimented by replacing lane detector (lane boundary marker detection block in Figure 3) in our pipeline with SCNN model [33] shared by the authors. The output lane boundary masks were sampled using criteria mentioned in [33] i.e., 18 points from every boundary with confidence ≥ 0.3 . Results are shown in the table as SCNN+Ours_{pp=0.3,18}. The results were a bit of a mixture. F1 score improved from +1.48% in *Normal* to +9.54% in *Dazzle Light*. While it reduced by -1.71% in *Crowded* and -11.79% in *Curve*. Due to unbalanced dataset, the influence of compromised categories was mitigated and an overall F1 of 75% was achieved which was an improvement of about 3% from solo SCNN.

We also customized the sampling criteria in SCNN and used 40 samples at a threshold ≥ 0.2 . Results are reported in the table as SCNN+Ours_{pp=0.2,40}, where *No Line/Shadow/Dazzle Light* showed improvements between 19.11% to 20.4%. An example of general improvement can be seen in Figure 6 (b and e) where the second lane boundary from the left was not correctly detected and marked as FN with solo SCNN. Our post processing of the markers detected the line with evidence (c) and without evidence (f).

Although the overall F1 score reached close to our best of 80.9%, the *Curve* category still remained 2% below the score of solo SCNN. The problem with the *Curve* category was that there were primarily fewer images (*Curved*: 421 images and for example *Normal*: 9621) and therefore they had less representation in the training. Secondly, during marker detection, occlusions reduce the number of detected markers. Therefore, during the parallel lane boundary fitting (section 3.3.1), the combined inliers remain fewer and such cases were labeled as FNs.

4.2. Results on CalTech

CalTech data has four video sets with a total of 1224 images. It's an old database (2008) and due to inconsistent ground truth, we only test it for one lane, preferably ego-lane as the annotations are there for at least two lanes in worst cases. Ego-lanes were identified through the slopes of their boundaries or their tangents (in case of curves) as in a perspective view, they have the steepest slopes in any direction. Since only SCNN's model was available, we tested SCNN and Ours₄₀ both trained on CULane. We mapped ground truth and predicted lanes each to 15 pixels and then to 30 pixels for two different tests. Results are shown in Table 2 (Left). Once again, our algorithm out performs SCNN noticeably in both tests with F1 scores of 89.90% (15:15) and 96.8% (30:30) respectively. Some sample images for comparison are shown in Figure 6 (g to j).

Data	Algorithm	F1	Data	Algorithm	Acc.
CalTech	SCNN(30:30)	0.900	TuSimple	SCNN	0.967
	Ours₄₀ (30:30)	0.968		ENet-SAD	0.966
	SCNN(15:15)	0.520		Ours ₄₀	0.955
	Ours₄₀ (15:15)	0.899		Ours_{40-gtRef}	0.979

Table 2. (Left) F1 score on CalTech data. GT and predicted boundaries were mapped to 15 pixel (15:15) and 30 pixel (30:30) wide masks (Right) TuSimple Accuracy. SCNN results copied from [33]. In TuSimple, for Ours_{40-gtRef} only, pixels along the ground truth lines marked as background are also kept as background in predicted lines.

4.3. Results on TuSimple

TuSimple [36] data contains 3626 training and 2782 test images. We trained Ours₄₀ on the training data and the results on the test set are shown in Table 2 (Right), where the results for SCNN were obtained from their paper [33]. As we can observe, Ours₄₀ accuracy (95.5 %) is 1% lower than SCNN (96.5%). We designed our framework to be occlusion robust, so that lane boundaries across the road users and obstacles can also be detected. This was the idea behind using lane boundary markers and line fitting as argued in section 3.1 followed by missing lane boundary prediction in section 3.3.2. But in the TuSimple's ground truth, lane boundary occlusions are marked as background producing a discontinued ground truth. If we also treat those pixels as background for prediction during evaluation (Ours_{40-gtRef}), then our accuracy reaches almost 98% surpassing SCNN and SAD-Nets by more than 1.5%.

5. Conclusion

In this paper we have presented a robust and efficient lane detection algorithm. We have shown that using equidistant parallel lines constraint on detected lane lines, retrieval of missing and occluded lanes can be greatly enhanced. The results on three different publicly available and diverse datasets show substantial improvement from state-of-the-art which is a proof in itself that assumption holds. Our approach is modular and the missing lane detection block can be appended to any lane boundary detection network by sampling markers from lane boundaries.

For future work, we want to address auxiliary and loading lanes where the assumption of parallelism does not apply. We also want to improve the detection of curved lane boundaries. In its current form, our research takes the state-of-the-art forward and highlights new ways of dealing with one of the most critical challenges in autonomous driving by unifying deep learning and geometric computer vision to build a system utilizing best of both the worlds.

References

- [1] https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection. *TuSimple evaluation criteria*, 2017.
- [2] A. Ali, A. Hassan, A. Razaqat, H. Khan, W. Kazmi, and A. Zaheer. Real-time vehicle distance estimation using single view geometry. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.
- [3] M. Aly. Real time detection of lane markers in urban streets. *Intelligent Vehicles Symposium*, pages 7–12, 2008.
- [4] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, dec 2017.
- [5] M. Bertozzi and A. Broggi. Real-time lane and obstacle detection on the GOLD system. In *Proceedings of Conference on Intelligent Vehicles*, pages 213–218, 1996.
- [6] H. Changzheng, H. Jin, and Y. Chaochao. An efficient lane markings detection and tracking method based on vanishing point constraints. *35th Chinese Control Conference (CCC)*, pages 6999–7004, 2016.
- [7] S. Chougule, N. Koznek, A. Ismail, G. Adam, V. Narayan, and M. Schulze. Reliable Multilane Detection and Classification by Utilizing CNN as a Regression Network. In L. Leal-Taixé and S. Roth, editors, *European Conference on Computer Vision Workshops*, pages 740–752, Cham, 2018. Springer International Publishing.
- [8] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, jun 1981.
- [9] F. Ghallabi, F. Nashashibi, G. El-Haj-Shhade, and M.-A. Mitter. LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map. In *21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii., United States, nov 2018. IEEE.
- [10] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali. DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, jun 2016.
- [11] R. Hartley and A. Zisserman. More Single View Geometry. In *Multiple view geometry in computer vision*, chapter 8, pages 205,216–219. Cambridge Univ. Press, 2nd edition, 2003.
- [12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [14] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [15] P. Hough. Method and Means for Recognizing Complex Patterns. U.S. Patent 3.069.654, dec 1962.
- [16] (<https://comma.ai>). Comma AI’s Open Pilot: An open source driving agent. 2019.
- [17] (<https://waymo.com>). Google’s Waymo: A self-driving taxi service. 2019.
- [18] (<https://www.mobileye.com>). Intel’s Mobile eye: ADAS and self-driving systems. 2019.
- [19] (<https://www.tesla.com/autopilot>). Tesla’s Autopilot: Full self-driving hardware. 2019.
- [20] C. R. Jung and C. R. Kelber. Lane following and lane departure using a linear-parabolic model. *Image and Vision Computing*, 23(13):1192–1202, 2005.
- [21] S. Kammel and B. Pitzer. Lidar-based lane marker detection and mapping. *2008 IEEE Intelligent Vehicles Symposium*, pages 1137–1142, 2008.
- [22] J. Kim and C. Park. End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1194–1202, 2017.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [24] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. H. Lee, H. S. Hong, S. H. Han, and I. S. Kweon. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. In *International Conference on Computer Vision (ICCV)*, pages 1965–1973. IEEE, 2017.
- [25] C. Lin and M. Wang. A vision based top-view transformation model for a vehicle parking assistant. *MDPI Sensors Journal*, pages 4431–4446, 2012.
- [26] P. Lindner, E. Richter, G. Wanielik, K. Takagi, and A. Isogai. Multi-channel lidar processing for lane detection and estimation. In *12th International Conference on Intelligent Transportation Systems*, pages 1–6, oct 2009.
- [27] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [28] X. Ma, A. Ismail, A. Soni, M. E. Carazo, V. Narayan, and M. Schulze. An efficient encoder-decoder CNN architecture for reliable multilane detection in real time. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1444–1451, 2018.
- [29] J. B. McDonald and J. B. M. Donald. Application of the Hough Transform to Lane Detection and Following on High Speed Roads. In *Proceeding of Irish Signals and Systems Conference*, 2001.
- [30] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 73:216–234, 2018.
- [31] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool. Towards End-to-End Lane Detection: an Instance Segmentation Approach. *IEEE Intelligent Vehicles Symposium (IV)*, 2018.
- [32] Y. Otsuka, S. Muramatsu, H. Takenaga, Y. Kobayashi, and T. Monji. Multitype lane markers recognition using local

- edge direction. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 604–609 vol.2, jun 2002.
- [33] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI Conference on Artificial Intelligence*, pages 7276–7283, 2018.
 - [34] E. Shang, J. Li, X. An, and H. He. A real-time lane departure warning system based on FPGA. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1243–1248, oct 2011.
 - [35] P. Y. Shinzato, V. Grassi, F. S. Osório, and D. F. Wolf. Fast visual road recognition and horizon detection using multiple artificial neural networks. *Intelligent Vehicles Symposium*, pages 1090–1095, 2012.
 - [36] TuSimple. TuSimple Velocity Estimation Challenge. pages <https://github.com/TuSimple/tusimple-benchmark/tre>, 2017.
 - [37] T. Veit, J. Tarel, P. Nicolle, and P. Charbonnier. Evaluation of Road Marking Feature Extraction. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 174–181, oct 2008.
 - [38] J.-G. Wang, C.-J. Lin, and S.-M. Chen. Applying fuzzy method to vision-based lane detection and departure warning system. *Expert Systems with Applications*, 37(1):113–126, 2010.
 - [39] www.eurotestmobility.net/eurotest.php. EuroTest.
 - [40] J. Zhang, Y. Xu, B. Ni, and Z. Duan. Geometric Constrained Joint Lane Segmentation and Lane Boundary Detection. In *European Conference on Computer Vision*, pages 37–47, 2018.

Real-time vehicle distance estimation using single view geometry

Ahmed Ali, Ali Hassan, Afsheen Rafaqat Ali, Hussam Ullah Khan, Wajahat Kazmi, Aamer Zaheer
KeepTruckin Inc.
Lahore, Pakistan

{Ahmed.Ali, Ali.Hassan, Afsheen, Hussam.Khan, Wajahat.Kazmi, AZ} @KeepTruckin.com

Abstract

Distance estimation is required for advanced driver assistance systems (ADAS) as well as self-driving cars. It is crucial for obstacle avoidance, tailgating detection and accident prevention. Currently, radars and lidars are primarily used for this purpose which are either expensive or offer poor resolution. Deep learning based depth or distance estimation techniques require huge amount of data and ensuring domain invariance is a challenge. Therefore, in this paper, we propose a single view geometric approach which is lightweight and uses geometric features of the road lane markings for distance estimation that integrates well with the lane and vehicle detection modules of an existing ADAS. Our system introduces novelty on two fronts: (1) it uses cross-ratios of lane boundaries to estimate horizon (2) it determines an Inverse Perspective Mapping (IPM) and camera height from a known lane width and the detected horizon. Distances of the vehicles on the road are then calculated by back projecting image point to a ray intersecting the reconstructed road plane. For evaluation, we used lidar data as ground truth and compare the performance of our algorithm with radar as well as the state-of-the-art deep learning based monocular depth prediction algorithms. The results on three public datasets (Kitti, nuScenes and Lyft level 5) showed that the proposed system maintains a consistent RMSE between 6.10 to 7.31. It outperforms other algorithms on two of the datasets while on KITTI it falls behind one (supervised) deep learning method. Furthermore, it is computationally inexpensive and is data-domain invariant.

1. Introduction

Human error is one of the major causes of driving accidents. At the heart of safe driving practices lies the driver's judgment to estimate their vehicle's braking distance w.r.t other road users. For this reason, Advanced Driver Assistance Systems (ADAS) have evolved substantially over the last few years and features such as pedestrian and tailgating

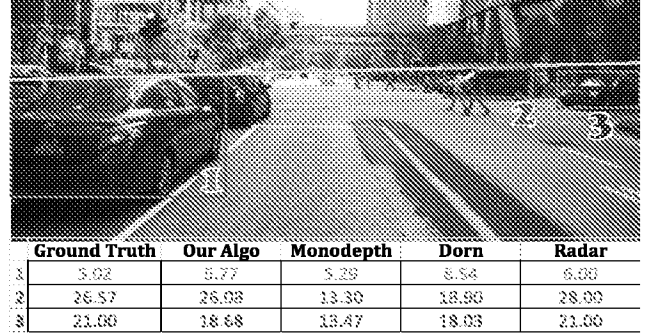


Figure 1. Sample image from nuScenes [3]. Estimated horizon (green line) and computed distances (meters) by all methods for each vehicle (1, 2 and 3) from the ego vehicle are displayed. Our method is compared against deep learning based depth estimation as well as radar. GT is the ground truth which is provided by the high resolution lidar.

detection have been introduced to alert drivers in the run-up to perilous maneuvers. Real-time logging of this information is crucial in this regard and can help fleet managers in scoring a driver's behavior. It also makes an integral part of autonomous driving in path planning, obstacle and collision avoidance and hence in overall safety assessment of the self-driving cars.

Active sensing in the form of Radars (Radio Detection and Ranging) and Lidars (Light Detection and Ranging) are a common choice for measuring distances to surrounding obstacles. Radars have ranges up to 150m [29] but they generally have low resolution. Lidars, on the other hand, offer higher spatial resolution [19] but they are too expensive (cheapest costs US\$ 7,500 [1]). The biggest advantage of using active sensing is its real-time operation as minimal post processing is required. Scene depth is estimated in a quick scan and simple thresholding may do the job. While improvement in the sensor technology and mass production is expected to reduce the cost of lidars by 2022 [19], they still have limitations of compromised performance in adverse weather conditions such as rain, snow and fog. In contrast, standard dashcams provide low cost and high reso-

lution RGB images of the scene but will demand intelligent post processing using robust algorithms and higher compute power on board.

In computer vision, the problem of distance estimation falls in the broader area of scene depth estimation. To this end, various data-driven and geometric approaches have been explored in literature. In data driven approaches, depth of objects is learnt through either training a supervised depth regressor [10] or unsupervised disparity estimator [13, 45]. On the other hand, geometric approaches take into account the geometry of the features in the scene such as Multi-View Stereo (MVS) and Single View Metrology [43]. MVS has remained a primary focus for depth recovery in the past [21]. It provides depth for each pixel via dense matching of more than one image. MVS is limited by the availability of texture in the scene, the efficiency of the algorithms as well as the complexity of multi-camera configuration. The accuracy of the depth map may not be consistent throughout. Single view metrology, instead, uses a single image to compute 3D measurements using perspective geometry [7]. Being ill-posed, it has remained limited to images of structured environments where a combination of parallel and orthogonal lines and surfaces are substantially available. In this paper, we use monocular dashcam images to estimate relative distances of vehicles on the road. For evaluation, we used lidar data as ground truth and established a comparison of the data-driven approaches with the radar data. We propose an efficient and data independent algorithm for real-time distance estimation for flat road surfaces using single view geometry. With the help of any available lane and vehicle detection modules (which are integral components of ADAS and self driving cars), it fully exploits the structure in the scene. We will show that it competes well with deep learning methods and under the geometric constraints, it outperforms end-to-end deep learning based methods as well as the radar data both in accuracy and efficiency making it suitable for edge devices. Since the autonomous driving is currently limited to high-way driving scenarios, it is well suited for such tasks. And except for the camera initialization phase, which is a one-time computation, it comes at almost no computational expense.

The breakdown of the paper is as follows: Section 2 provides a quick review of depth estimation literature. Section 3 discusses our approach in detail with ADAS modules for lane and vehicle detection in section 3.1, camera initialization in section 3.2 and distance estimation in section 3.3. Section 4 outlines the datasets, training and testing. Results and a discussion on the results are presented in section 4.1 and section 5 concludes the paper.

2. Related work

In literature, only a few works have employed approaches similar to ours. Park and Hwang [32] used monoc-

ular imaging and average vehicle width in real world to estimate horizon. Following that, the distance of the bottom of the vehicle's detected bounding box from the horizon provided a hint of its relative distance. But they assumed an already known camera height which can be difficult if done on a multitude of vehicle across a fleet. In 2017, top performing submission to TuSimple's velocity estimation challenge employed deep learning based depth prediction techniques combined with object (vehicle) detection to find relative velocities [37]. Keeping this in view, we will have to look into the research work done in the two domains i.e. our proposal of single view geometry for depth estimation and the state-of-the-art deep learning based depth prediction.

2.1. Single view geometry based depth estimation

In computer vision, single view based recovery of scene geometry have been thoroughly studied [4, 36, 30, 17]. Leibowitz used geometric properties such as parallelism and orthogonality of structures to recover 3D models from one or two views without explicitly knowing the metric properties of objects [26]. Lee *et al.* showed that even in the presence of clutter, full 3D model can be recovered from a single image using geometric reasoning on detected line segment produced from planar surfaces [24]. Recently, Zaheer *et al.* used orthogonal lines in a multi planar single view, to recover full 3D reconstruction of the scene [43]. These approaches are particularly interesting for indoor man-made environments or for outdoor scenes with man made structures around. The reason is that human architecture involves a profusion of parallel and orthogonal lines emanating from planar surfaces.

Apart from 3D model recovery from a single view, substantial work has also been done in single view metrology. Leibowitz and Zisserman proposed metric rectification of planes from single perspective view and used it to measure angles, lengths and ratios under given constraints of one known angle, two unknown but equal angles and a known length ratio [27]. Zaheer and Khan showed that projectively distorted views of man-made 3D structures can be metric rectified using orthogonal plane-pairs [42]. Liu *et al.* estimated depth from a single view given semantic segmentation under geometric constraints [28]. On this topic, Criminisi *et al.* explain single and multi view metric reconstructions in detail, especially determining camera pose, computing distance between planes parallel to a reference plane and also finding area and length ratios on any plane parallel to the reference plane [7, 6].

2.2. Data-driven depth prediction

Digressing from a geometric approach, data-driven depth prediction techniques have used both supervised and unsupervised learning for stereo as well as monocular images. Sinz *et al.* used stereo images in Gaussian process re-

gression to estimate pixel wise depth map avoiding complications of stereo calibration [35]. Saxena *et al.* used monocular images of a variation of unstructured outdoor scenes for training Markov Random Fields (MRFs) on both local and global features for the recovery of fairly accurate depth maps [34]. Ladicky *et al.* designed a stereo matching classifier for predicting likelihood of matching pixel in one image offset by disparity in second image [23].

Owing to little success, the data-driven approaches for depth estimation could not acquire a lot of attention until the advent of deep learning. Deep learning based depth estimation made a huge impact in terms of holistic and accurate depth prediction so much so that a significant number of contributions have surfaced over a short span of time [40, 46, 25, 41, 45]. Žbontar and LeCun trained a small convolutional neural network for matching stereo pairs instead of handcrafted feature matching and showed results better in terms of both the accuracy of depth maps as well as efficiency [44]. Eigen *et al.* used two deep networks in cascade [8] to predict high-level global depth followed by refinement to low-level local depth in a supervised regression framework. This approach acquired a lot of traction and several improvements followed such as adding Conditional Random Fields (CRFs) by Li *et al.* [39]. Cao *et al.* transformed monocular depth estimation from regression to a supervised classification problem [5]. Fu *et al.* used an ordinal regression loss in supervised training to eliminate slow convergence during training and sub-optimal minimas [10]. Their method is the best performing one on Kitti benchmark [2] with iRMSE 12.98 (1/km), RMSE 2.6 (m).

Supervised learning requires ground truth depth data which in case of deep learning becomes challenging to acquire. Therefore, recently, some unsupervised approaches have appeared that have performed exceptionally well. Garg *et al.* proposed a convolutional encoder network in which they use one of the stereo-pair style images and predict a disparity map [11]. An inverse warp of the second image is then produced from the known inter-view displacement and the disparity map, which is then used to reconstruct the first image. The photometric error in the reconstruction defines the network loss. An improvement to this monocular depth learning from stereo pairs was introduced by Godard *et al.* by integrating left-right consistency check [13]. Kundu *et al.* employed unsupervised adversarial learning for depth prediction [22].

3. Our approach

We use single view geometry for distance estimation. As argued earlier, single view geometry depends on the perspective of the parallel or orthogonal lines and planes in the image. As lane lines are generally parallel, we exploit the parallelism for this matter. We start with unknown camera extrinsics because it is usually difficult to obtain this infor-

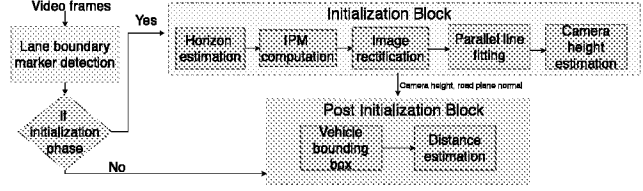


Figure 2. Block diagram of our algorithm.

mation as it may change from time to time. Instead, we propose an automated way of finding camera height using horizon through cross-ratios of parallel lines [15]. Determination of camera height completes camera initialization which only needs to be done once for each rigid camera assembly. Road plane is then reconstructed and the distance from other vehicles on the road can be calculated through simple math. A block diagram of our full pipeline is shown in Figure 2. For camera initialization, we assume a straight and flat road surface. Lane boundaries and vehicles can be detected using any off-the-shelf lane and vehicle detection algorithms. In the following section, we will now briefly describe lane and vehicle detectors that are used in evaluation for distance estimation in this paper.

3.1. Lane and vehicle detection

To exploit road geometry, detection of lane boundaries is a crucial step for which we used the lane detection network by Khan *et al.* [20] which uses a fully convolutional deep network for detecting keypoints sampled along the lane boundaries. The network architecture diagram is shown in Figure 3(a). It consists of an encoder block (ResNet50 [16] in this case), responsible for feature extraction, and a decoder block that learns the relationship among extracted features and makes decision on top of it. The decoder block comprised of four convolutional layers, a single transpose-convolution layer (that upsamples feature map by 4x) followed by the output convolution layer. In the output layer, just like Mask-RCNN, one-hot mask for each keypoint is generated, where only the keypoint pixel is encoded (marked high or foreground).

For training the lane detection network, CULane[31] dataset has been used. Every lane boundary ground truth was sampled into 40 keypoints, this number was chosen based on the best possible trade-off between efficiency and accuracy [20], with greater density of points closer to the road plane horizon (Figure 3(b)). This is done by splitting every lane into three vertical segments and sampling equally spaced keypoints (20, 10 and 10) from each segment starting from the one closer to the horizon. At inference time, line/curve is fitted on the predicted keypoints using RANSAC [9]. A best fit line must have a minimum number of inlier markers, which in case of 40 markers/lane boundary were kept at 10.

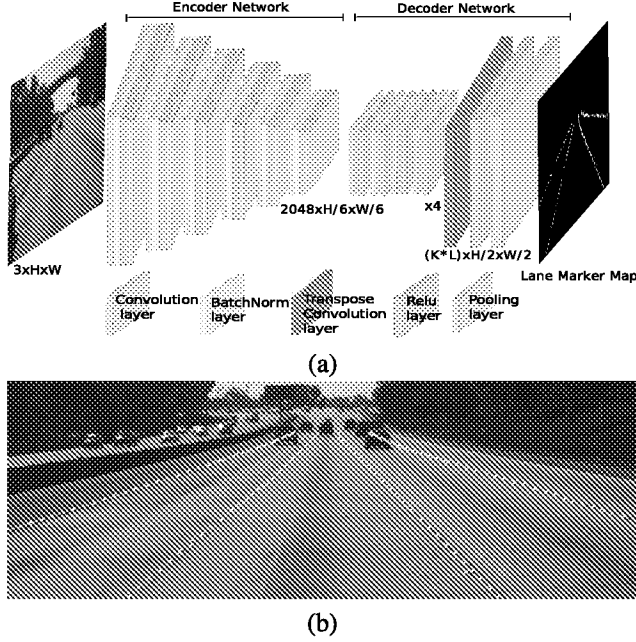


Figure 3. (a) The architecture of our lane detection network. The encoder is ResNet50 and the decoder consists of 4 convolution layers, 1 transpose-convolution layer followed by a fully convolutional output layer with one channel for each keypoint (b) The input image and keypoints sampled from the ground truth lane line are overlaid on the image. Number of keypoints increase towards the vanishing point.

For vehicle localization, any detection framework can be used. We fine-tuned COCO pre-trained Faster-RCNN model [33] with Resnet-50 backbone on Kitti vehicle dataset [12].

3.2. Camera initialization

Camera initialization involves finding camera height and road plane normal. This is our first contribution in this paper as along with camera intrinsics, we use road geometry and lane width which has not been used before in this regard. Integrating these known priors makes the process of initialization self-sufficient and fully automatic. It's a one-off calculation for every camera and stays constant until the camera is displaced or replaced altogether. It involves the following sub-modules:

3.2.1 Inverse Perspective Mapping

Inverse Perspective Mapping (IPM) is used to remove the perspective distortion in an image. Application of an IPM can synthetically rotate the camera into a top (bird's eye) view thus rendering the intersecting lines as parallel; the way they actually are in the real-world. This is required in order to find the camera height which is required for distance estimation. Therefore, we introduce a novel, accurate

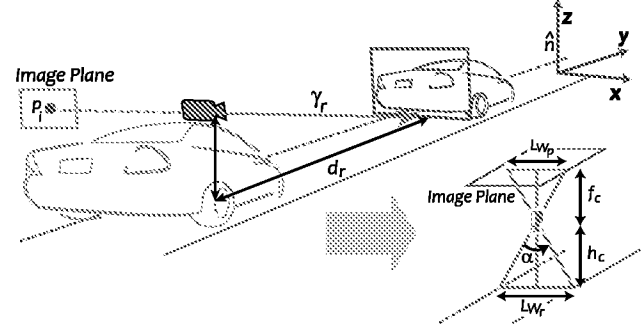


Figure 4. (Left) Point p_i from the bottom of the detected vehicle is back projected to a ray γ_r intersecting road plane at a point which gives the distance d_r of the point from the camera on road plane with normal vector $\hat{n}$ (Right) Synthetically rotated camera view to estimate camera height h_c given focal length f_c , lane width in rectified image Lw_p and lane width in real world Lw_r .

and robust way of estimating an IPM by exploiting road geometry with no prior knowledge of the camera pose.

Our approach is aimed for a forward looking dashcam installed inside the driver's cabin. Configuration was done in a right-handed frame of reference. The road plane normal $\hat{n}$ was along the Z-axis (upwards), Y-axis was in the driving direction and X-axis towards the right side of the driver as shown in Figure 4(Left).

The estimation of IPM used a planar road scene having no elevation and more than one lane (3 or more lane boundaries), such as a typical highway scenario (Figure 5). Critical part of initialization was the estimation of a vanishing line or horizon which required at least three parallel lines in the rectified image.

3.2.2 Horizon estimation

The horizon must be known to estimate the required camera rotation for an IPM. A horizon is a vanishing line of a plane which, in our case, is the road. In order to find vanishing line, we need two vanishing points. The first one can be simply calculated through the cross product of the vectors of any two lane lines expressed in homogeneous coordinates (see Figure 5 (a)). We call it the forward vanishing point (V). Finding a second vanishing point, however, is a bit involved.

According to Hartley and Zisserman Chapter 2 [14], if there are three co-linear points in an image a' , b' and c' (as shown in Figure 5 (a)) with known real world length ratios ($\frac{ab}{bc}$) of their corresponding real world points (a , b and c) as shown in Figure 5 (b), we can find a point on the vanishing line of the plane on which these points lie. Lets call this point the lateral vanishing point or V' .

We take this concept and map it to our problem i.e. the road plane. For this, we will need three co-linear points existing on the road plane having known real world length

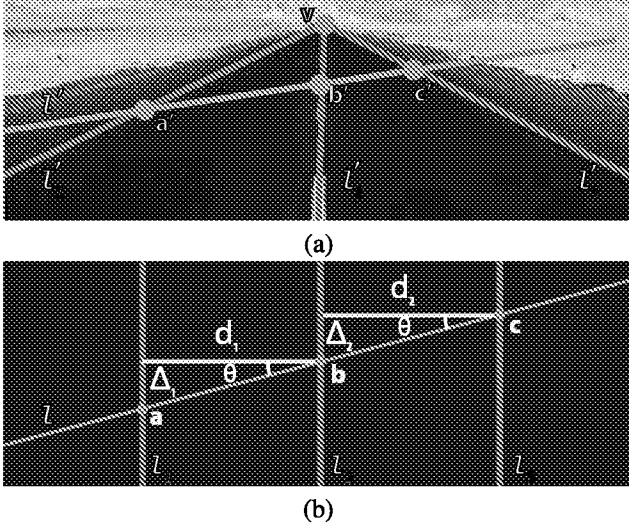


Figure 5. (a) Perspective view of road plane. Lane lines (l'_1 , l'_2 and l'_3) are shown in red and an arbitrary line l' (blue) cuts all lane lines at points a' , b' and c' . Forward vanishing point (V) is obtained by the cross product of any two lane lines (b) Real world top-down view of the road plane. d_1 and d_2 are the distances between lane lines l_1, l_2 and l_2, l_3 respectively. Since all lane lines are parallel Δ_1 and Δ_2 are similar triangles which implies $\frac{ab}{bc} = \frac{d_1}{d_2}$

ratios. Since we already have detected lane lines in the image which lie on the road plane, we can find three co-linear points by drawing any line l' which intersects all three lane lines at points a' , b' and c' respectively.

The points a' , b' and c' are in fact images of their real world counter parts a , b , and c which lie on a line l as shown in Figure 5 (b). The line l' is the image of line l . In order to find the lateral vanishing point of the road plane, according to the description above, we need to know the length ratios between these real world points. The fact that these three points lie on three (parallel) lane lines (l_1, l_2 and l_3), it can be established through trigonometry that the ratio of the line segments $\frac{ab}{bc}$ is equal to $\frac{d_1}{d_2}$ where d_1 and d_2 are the distances between lines l_1, l_2 and l_2, l_3 respectively, as shown in figure (b). Here d_1 and d_2 are the lane widths.

Although the algorithm will still work if neighbouring lanes don't have same width (only requires ratio of the widths i.e. $\frac{d_1}{d_2}$ to be known), we assume the ratio to be 1 ($d_1=d_2$ =lane width), since the lane lines are generally equidistant. Since the initialization is required only once for a camera, it can be done on a highway or a section of the highway where lanes are consistent/equidistant (using gps).

After getting two vanishing points (VPs) per frame, we collected all the sets of the vanishing points across all the frames of a video clip. The horizon was found as the best fitted line to the VPs as described in Algorithm 1.

Algorithm 1: Best fitting Horizon (l_h) estimation

Input: Detected markers by lane detector
Output: Estimated Horizon line

- 1 Find forward and lateral VP for each frame in a video:
- 2 initialize;
- 3 for $i \leq \text{num_frames in video}$ do
- 4 Fit lines to keypoints ($B = \text{num_lines}$) ;
- 5 if $B \geq 2$ then
- 6 for $j \leq {}^B C_2$ do
- 7 forward_vpi_j = crossprod(line_1, line_2)
- 8 end
- 9 end
- 10 if $B \geq 3$ then
- 11 for $k \leq {}^B C_3$ do
- 12 Get lateral_vpi_k using cross ratios as described in Section 3.2.2
- 13 end
- 14 end
- 15 end
- 16 RANSAC for best fitting horizon across video:
- 17 initialize;
- 18 $s = \text{random_int between 1 to num_forward_vps}$;
- 19 $t = \text{random_int between 1 to num_lateral_vps}$;
- 20 while iterations $\leq t$ OR num_inliers $\leq \text{max_inliers}$ do
- 21 Fit line l_h to forward_vp(s) and lateral_vp(t);
- 22 if $\text{dist}(l_h, \text{forward_vp}(s)) \leq \text{threshold}_s$ & $\text{dist}(l_h, \text{lateral_vp}(t)) \leq \text{threshold}_t$ then
- 23 record and count inlier points
- 24 end
- 25 end

3.2.3 Image rectification

Now that the horizon has been determined, the IPM could be found. We computed road plane normal $\hat{n}$ using camera intrinsic matrix K and horizon l_h [14] by $\hat{n} = K^T l_h$. The IPM is then computed using $H_{IPM} = K R K^{-1}$, where $R = [l_h \times \hat{n}; (l_h \times \hat{n}) \times -\hat{n}; -\hat{n}]$ is the rotation matrix and H_{IPM} is the rectification homography which will rotate the camera view to align its Z-axis with the road's normal as shown in Figure 4(Right). Frames could now be simply rectified into a bird's eye view by applying this IPM (Figure 4(Right)).

3.2.4 Parallel line fitting

Parallel lines can be represented as:

$$a_1x + b_1y + c_1 = 0, a_2x + b_2y + c_2 = 0$$

where the co-efficients $a_1 = a_2$ and $b_1 = b_2$ and only the intercept c_1 and c_2 are different. After a minimum number of inlier constraint is satisfied, parallel lines are fitted to the inlier markers using least squares by solving $Ax = 0$.

Here A contains the inlier points (subscript) for each parallel line (superscript) in the rectified view. This system is solved using singular value decomposition for the least singular value.

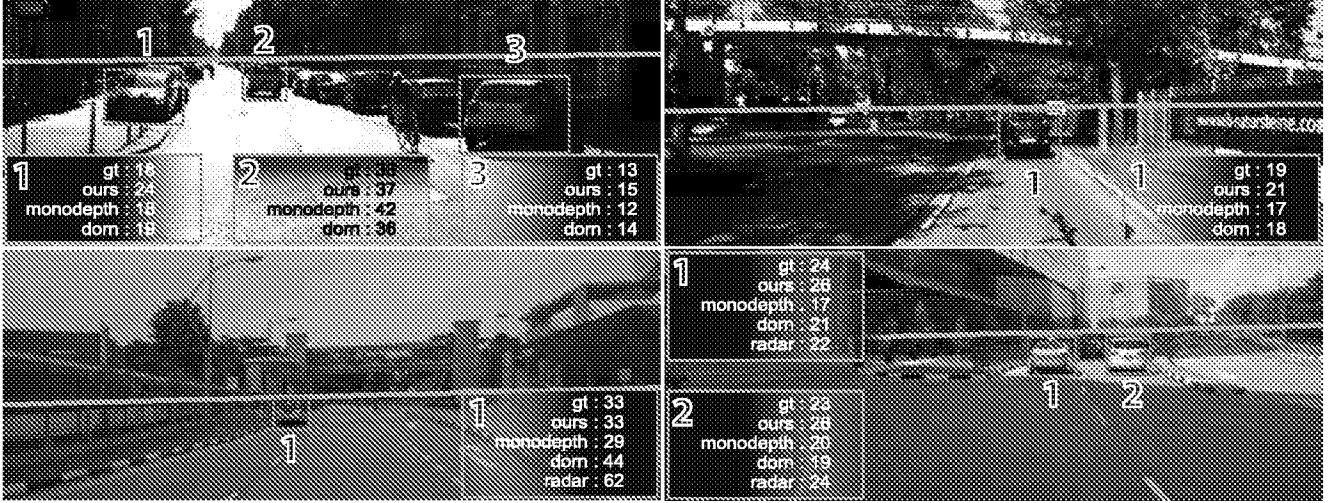


Figure 6. Sample results on Kitti [12] (top) and nuScenes [3] (bottom). Displayed numbers are the distances in meters. Vehicles are numbered for clarity.

3.2.5 Lane Width Initialization

In the rectified view of a given frame, a pair of consecutive parallel lines were selected and the distance between the lines was calculated per frame through:

$$L_{Wp} = \frac{\sum_f \frac{|c_2 - c_1|}{\sqrt{a^2 + b^2}}}{f} \quad (1)$$

where f is the number of inlier frames and L_{Wp} is the average initialized lane width. A frame is considered an inlier frame if it has at least one forward and one lateral vanishing point.

3.2.6 Camera height estimation

Now that we have lane width in pixels L_{Wp} , we can use the camera focal length (f_c) to compute viewing angle α alpha using trigonometry

$$\alpha = 2 \times \tan^{-1}\left(\frac{L_{Wp}/2}{f_c}\right) \quad (2)$$

Once camera viewing angle α is known, camera height h_c is given by

$$h_c = \frac{\frac{L_{Wr}}{2}}{\tan(\frac{\alpha}{2})} \quad (3)$$

Here L_{Wr} is the lane width in real world and considering the fact that lane widths are usually standard across highways, L_{wr} is generally known. Determining camera height completes camera initialization. This is depicted in Figure 4 (Right).

3.3. Distance estimation

Once the camera is initialized, we know the camera height and road plane normal. We can then reconstruct the road plane:

$$\pi_r = \begin{bmatrix} \hat{n} \\ -h_c \end{bmatrix} \quad (4)$$

where $\hat{n}$ is the road plane normal and π_r is 4×1 reconstructed road plane vector. A ray from the point p_i in the image plane of the camera is given by $\gamma_r = K^{-1}p_i$. So the distance from the image plane to point where this ray intersects the reconstructed road plane in the real-world can be found by solving:

$$\pi_r^T \begin{bmatrix} d_r \gamma_r \\ 1 \end{bmatrix} = 0 \quad (5)$$

where d_r is the distance where γ_r intersects the plane:

$$d_r = \frac{h_c}{\hat{n} \cdot \gamma_r} \quad (6)$$

This is depicted in Figure 4 (Left). It is this simple arithmetic calculation that estimates the distance of the detected objects on the reconstructed road plane which makes it computationally inexpensive. Shift in the horizon which can be caused by changing slope of the road can impart difference to the observed distances. The best way to investigate the extent of error is to evaluate our algorithm on publicly collected datasets and compare them against the computationally expensive but state-of-the-art deep learning approaches.

4. Experimental setup

To evaluate our distance estimation algorithm, we used the well known datasets i.e. KITTI [12], nuScenes [3] and the recently released Lyft level 5 dataset [18]. All the datasets have color images along with Lidar data which is used as distance ground truth. nuScenes also provide front Radar data. In order to localize vehicles in the scenes, we used our vehicle detector as described in section 3.1. During the assessment, among the overlapping vehicle boxes only

Algorithm	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs.Rel	RMSE	RMSE log	Sq.Rel
(a) KITTI dataset							
DORN[10]	0.92	0.96	0.98	0.11	2.44	0.18	0.44
MonoDepth[13]	0.82	0.94	0.98	0.13	6.34	0.21	1.36
Ours	<u>0.53</u>	0.89	0.98	0.29	<u>6.24</u>	0.30	2.02
(b) nuScenes dataset							
DORN[10]	0.55	0.85	0.96	0.25	7.43	0.33	1.83
MonoDepth[13]	0.56	0.81	0.93	0.23	8.38	0.36	2.16
Radar	0.73	0.80	0.85	0.48	11.66	0.71	13.25
Ours	<u>0.78</u>	0.94	0.97	0.15	<u>6.10</u>	0.24	1.29
(c) Lyft Level 5 dataset							
DORN[10]	0.11	0.30	0.67	0.73	11.43	0.57	7.91
MonoDepth[13]	0.03	0.12	0.45	0.50	12.23	0.77	5.50
Ours	<u>0.69</u>	0.87	0.93	0.17	<u>7.31</u>	0.32	1.53

Table 1. Distance estimation results on (a) KITTI (b) nuScenes (c) Lyft Level 5 datasets. Metrics in Red means higher is better. Metrics are explained in [8, 13]. Metrics in Blue means lower is better. Best scores are in bold. The scores of our algorithm are underlined.

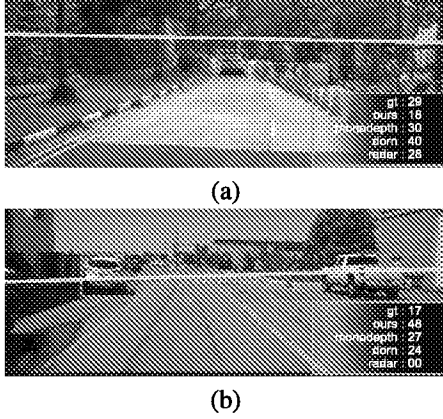


Figure 7. Sample images from nuScenes [3]. Displayed numbers are the distances in meters. Sub-optimal horizon as compared to Figure 1 due to (a) downwards slope and (b) upwards slope. Recalculation required in this case.

the foremost box was selected. Hence only the un-occluded vehicles were used. Maximum distance cap of 70m was set since some lidars do not perform at larger distances. Besides, 70m braking distance provides enough room for a fair assessment.

Distance was in meters and we used the metrics defined in [8, 13], namely, Ratio threshold δ , Absolute Relative difference (Abs Rel), Squared Relative difference (Sq Rel), RMSE (linear) and RMSE log for evaluation. Apart from our algorithm which is based on single view geometry, we also compared distance estimates of a supervised deep learning based method, DORN by Fu *et al.* [10] which has shown the best performance on Kitti depth estimation benchmark [2]. We used their model which is trained on the KITTI data for color images with lidar as ground truth. From unsupervised deep learning methods, we use MonoDepth by Godard *et al.* [13]. We used the model shared by the authors which is trained on Kitti’s stereo data. MonoDepth predicts disparity which is converted into depth using the stereo configuration used in the data acquisition [Depth in meters = (FocalLength $\times$ Baseline in meters) / Disparity]. Lidar and radar data were projected on the image plane using camera matrix to produce depth maps. The vehicle depth was computed by averaging all the non zero depth values inside the vehicle box for lidar, MonoDepth and DORN. The detected box size was reduced by 50% in height and width with the same center in order to reduce the influence of the background. For radar however, we took the non zero minimum value inside the full sized box due to the sparsity of the data (works best for radar). For our algorithm, we took the middle point of the bottom of the full sized box for distance estimation because our algorithm requires the point to be on the road plane.

oDepth predicts disparity which is converted into depth using the stereo configuration used in the data acquisition [Depth in meters = (FocalLength $\times$ Baseline in meters) / Disparity]. Lidar and radar data were projected on the image plane using camera matrix to produce depth maps. The vehicle depth was computed by averaging all the non zero depth values inside the vehicle box for lidar, MonoDepth and DORN. The detected box size was reduced by 50% in height and width with the same center in order to reduce the influence of the background. For radar however, we took the non zero minimum value inside the full sized box due to the sparsity of the data (works best for radar). For our algorithm, we took the middle point of the bottom of the full sized box for distance estimation because our algorithm requires the point to be on the road plane.

4.1. Results and Discussion

4.1.1 Tests on KITTI

KITTI dataset includes images from two color 1.4 MP PointGrey Flea2 cameras (left one is used in these experiments) and a Velodyne HDL-64E lidar data. Lidar data was mapped to the cameras. It does not have radar data. We used its six video clips which have the lidar as well as the corresponding video data in the KITTI validation split.

Table 1 (a) show the results and Figure 6 top-row shows some sample KITTI results. Based on the RMSE, we can see that supervised depth estimation by DORN performed much better than either MonoDepth or our algorithm. Both data driven approaches, DORN and MonoDepth were trained on the training and quiet possibly the validation sets of KITTI before the authors submitted the results to the KITTI test set benchmark [2]. Please note that the KITTI data was recorded by driving around the city of Karlsruhe, Germany, in the rural areas as well as on the highways [38]. And although our algorithm fell behind DORN in performance, still it had lower RMSE than the un-

supervised deep learning model (MonoDepth). Therefore, even after aiming for a highway driving scenario, it is still competing the state-of-the-art data driven approaches on a variety of road types. For a fair assessment of robustness and domain in-variance of the algorithms in consideration, we tested these models along with our algorithms on some other public datasets.

4.1.2 Tests on nuScenes

nuScenes dataset contains recorded videos of mainly city scenes. It has 1000 video clips out of which 150 comprise the validation set that we used in this assessment. The dataset contains clips from multiple cameras installed on the vehicle. We used the front mounted camera CAM_FRONT for evaluation. Table 1 (b) show the results.

The results show that unlike KITTI, the deep learning networks both supervised and unsupervised, do not perform as well. Our algorithm however, performs better than all including radar in both δ (0.78) and RMSE (6.10). Figure 6 bottom row shows some results of nuScenes where horizon was estimated correctly and our algorithm performed well. In contrast, Figure 7 display example with sub-optimal horizon estimation and hence incorrect distance estimate which is a limitation of our system.

4.1.3 Tests on Lyft Level 5 Dataset

Lyft dataset is the newest and the largest of all datasets. The setup is similar to nuScenes. We used the data from the forward looking Cam0 which includes 22680 sample images and collected on relatively planar roads as opposed to nuScenes making it the largest dataset in our evaluation setup. Table 1 (c) show the results. Since Monodepth and DORN are trained on KITTI, their performance is compromised on a new camera and its settings with RMSE of 12.23 and 11.43, respectively. Our algorithm, however, outperforms with an RMSE of 7.31.

4.2. Processing efficiency

In USA, the traffic rules require 3 to 5 secs headway from the vehicle in the front. Therefore, for an ADAS system, real-time operation is imperative as the response time is very short and compute power onboard is very limited. As argued earlier, this is one of the reasons for using lidars and radars. In this regard, we compared our algorithm with others for the processing time and the results are shown in Table 2. Deep networks require high compute power. As far as our method is concerned, there is no specific processing requirements. It is just a single division and a dot product which takes $2.31e^{-05}$ secs on Core i7 CPU. The initialization part can use any available Lane detection framework which will be in operation anyway within an ADAS or self-driving car. Our lane detection network discussed in section 3.1 runs at 0.125 secs/frame. But it's a one-off calculation for a limited number of frames and re-Initialization

Algorithm	DORN	MonoDepth	Ours
GPU	2.10	0.124	—
CPU	130	0.795	$23e^{-06}$

Table 2. Processing time per frame in secs averaged over 100 frames of nuScenes. GPU: Nvidia GTX-1080 with 8 GB on chip RAM, CPU: Intel Core i7 7820HK 2.9 GHz with 32 GB system RAM. Our algorithm's time is post initialization run-time for distance calculation. During initialization, our lane detector was also used at 0.124 secs/frame (one-time execution per camera).

will not be required until the camera configuration is disturbed. This brings our algorithm at par with radar and lidar data processing without demanding any excessive compute power on onboard edge devices.

5. Conclusion

In this paper we have presented a single view geometry based relative distance estimation algorithm for road vehicles. We have compared our algorithm with data driven deep learning (DL) based methods. Supervised deep learning performed exceptionally well on KITTI data. The compromise of the DL based algorithms on nuScenes and Lyft dataset was perhaps due to domain variance since the models were trained on KITTI data only. Therefore, although data driven approaches are the future, they do have their short comings. We instead proposed a geometry based alternative that is simple, intuitive, independent of data and performs almost equally on all the tested datasets keeping RMSE within a very narrow range i.e. 6.10 to 7.31. Our algorithm uses ADAS module of lane detection only for the initialization part which is computationally expensive since it's usually a DL based method. But initialization is a one-off calculation and after that, our distance estimation works real-time.

In this work, we also conclude that both geometric as well as deep learning frameworks are strong contestants to replace active sensors for ADAS systems as well as self-driving cars which will be a great help in reducing the cost of the intelligent transport systems of the future.

Acknowledgements

This work was funded by KeepTruckin Inc. Authors greatly acknowledge the assistance of Mr. Numan Sheikh and Mr. Abdullah Hasan in the data collection and review process.

References

- [1] <https://www.businessinsider.com/googles-waymo-reduces-lidar-cost-90-in-effort-to-scale-self-driving-cars-2017-1>. *Business Insider*, Google Waymo reduces cost of Lidars by 90%, 2017.
- [2] http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction. *KITTI Depth prediction benchmark*.

- [3] <https://www.nuscenes.org/>. *NUScenes*, 2018.
- [4] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast Automatic Single-View 3-d Reconstruction of Urban Scenes. In *European Conference on Computer Vision*, 2008.
- [5] Y. Cao, Z. Wu, and C. Shen. Estimating Depth From Monocular Images as Classification Using Deep Fully Convolutional Residual Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, nov 2018.
- [6] A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer-Verlag, Berlin, Heidelberg, 2001.
- [7] A. Criminisi, I. Reid, and A. Zisserman. Single View Metrology. *International Journal of Computer Vision*, 40(2):123–148, nov 2000.
- [8] D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2014.
- [9] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, jun 1981.
- [10] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [11] R. Garg, G. VijayKumarB., and I. D. Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *European Conference on Computer Vision*, 2016.
- [12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32:1231–1237, 2013.
- [13] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [14] R. Hartley and A. Zisserman. More Single View Geometry. In *Multiple view geometry in computer vision*, chapter 8, pages 205,216–219. Cambridge Univ. Press, 2nd edition, 2003.
- [15] R. Hartley and A. Zisserman. Projective Geometry and Transformation of 2D. In *Multiple View Geometry in Computer Vision*, chapter 2, pages 51–52. Cambridge University Press, 2nd edition, 2003.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [17] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *10th IEEE International Conference on Computer Vision*, volume 1, pages 654–661, oct 2005.
- [18] V. Kesten, R. and Usman, M. and Houston, J. and Pandya, T. and Nadhamuni, K. and Ferreira, A. and Yuan, M. and Low, B. and Jain, A. and Ondruska, P. and Omari, S. and Shah, S. and Kulkarni, A. and Kazakova, A. and Tao, C. and Platinsky, L. and Jiang, W. and. Lyft Level 5 Dataset. <https://level5.lyft.com/dataset/>, 2019.
- [19] M. Khader and S. Cherian. An Introduction to Automotive LIDAR (Texas Instruments). Technical report, 2018.
- [20] H. Khan, A. Rafayat, A. Hassan, A. Ali, W. Kazmi, and A. Zaheer. Lane detection using lane boundary marker network with road geometry constraints. In *IEEE Winter Conference on Applications of Computer Vision, Snowmass, Co.*. IEEE, 2020.
- [21] V. Kolmogorov and R. Zabih. Multi-camera Scene Reconstruction via Graph Cuts. In *Proceedings of the 7th European Conference on Computer Vision-Part III, ECCV '02*, pages 82–96, London, UK, 2002. Springer-Verlag.
- [22] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu. AdaDepth: Unsupervised Content Congruent Adaptation for Depth Estimation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [23] L. Ladicky, C. Häne, and M. Pollefeys. Learning the Matching Function. *CoRR*, abs/1502.0, 2015.
- [24] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, 2009.
- [25] Z. Li and N. Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In IEEE, editor, *IEEE Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.
- [26] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating Architectural Models from Images. *Computer Graphics Forum*, 18(3):39–50, 1999.
- [27] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *International Conference on Computer and Pattern Recognition*. IEEE Comput. Soc, 1998.
- [28] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1253–1260, jun 2010.
- [29] J. Oberhammer, N. Somjit, U. Shah, and Z. Baghchehsaraei. RF MEMS for automotive radar. In D. Uttamchandani, editor, *Handbook of Mems for Wireless and Mobile Applications*, Woodhead Publishing Series in Electronic and Optical Materials, chapter 16, pages 518–549. Woodhead Publishing, 2013.
- [30] J. Pan, M. Hebert, and T. Kanade. Inferring 3D layout of building facades from a single image. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2926, 2015.
- [31] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI Conference on Artificial Intelligence*, pages 7276–7283, 2018.
- [32] K.-Y. Park and S.-Y. Hwang. Robust Range Estimation with a Monocular Camera for Vision-Based Forward Collision Warning System. *The Scientific World Journal*, (923632):9, 2014.
- [33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. D. Lawrence, D. D. Lee,

- M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 91–99. Curran Associates, Inc., 2015.
- [34] A. Saxena, S. H. Chung, and A. Y. Ng. Learning Depth from Single Monocular Images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS’05, pages 1161–1168, Cambridge, MA, USA, 2005. MIT Press.
 - [35] F. H. Sinz, J. Q. Candela, G. H. Bakir, C. E. Rasmussen, and M. O. Franz. Learning Depth from Stereo. In C. E. Rasmussen, H. H. Bülthoff, B. Schölkopf, and M. A. Giese, editors, *Pattern Recognition*, pages 245–252, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
 - [36] P. Sturm and S. Maybank. A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images. In *British Machine Vision Conference*, pages 265–274, 1999.
 - [37] TuSimple. TuSimple Velocity Estimation Challenge. pages <https://github.com/TuSimple/tusimple-benchmark/tree/2017>.
 - [38] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity Invariant CNNs. In *International Conference on 3D Vision (3DV)*, 2017.
 - [39] B. L. . C. S. . Y. D. . A. van den Hengel ; Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, jun 2015.
 - [40] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning Depth from Monocular Videos using Direct Methods. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.
 - [41] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
 - [42] A. Zaheer and S. Khan. 3D Metric Rectification using Angle Regularity. *IEEE Winter Conference on Applications of Computer Vision*, pages 31–36, 2014.
 - [43] A. Zaheer, M. Rashid, M. A. Riaz, and S. Khan. Single-View Reconstruction using orthogonal line-pairs. *Computer Vision and Image Understanding*, 172:107–123, 2018.
 - [44] J. Žbontar and Y. LeCun. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *Journal of Machine Learning Research*, 17(1):2287–2318, jan 2016.
 - [45] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
 - [46] Y. Zhang and T. Funkhouser. Deep Depth Completion of a Single RGB-D Image. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2:

☐	Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)
--------------------------	---

Inventor Information:

Inventor	1				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Wajahat		Kazmi		
Residence Information (Select One) US Residency ☒ Non US Residency Active US Military Service					
City	Islamabad	Country of Residence ⁱ	PK		
Mailing Address of Inventor:					
Address 1	65 Hawthorne St 4th floor				
Address 2					
City	San Francisco	State/Province	CA		
Postal Code	94105	Country ⁱ	US		
Inventor	2				Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Aamer		Zaheer		
Residence Information (Select One) US Residency ☒ Non US Residency Active US Military Service					
City	Islamabad	Country of Residence ⁱ	PK		
Mailing Address of Inventor:					
Address 1	65 Hawthorne St 4th floor				
Address 2					
City	San Francisco	State/Province	CA		
Postal Code	94105	Country ⁱ	US		
Inventor	3				Remove
Legal Name					

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Prefix	Given Name	Middle Name	Family Name	Suffix
	Ali		Hassan	
Residence Information (Select One) US Residency ☒ Non US Residency Active US Military Service				
City	Islamabad	Country of Residence ⁱ	PK	

Mailing Address of Inventor:

Address 1	55 Hawthorne St 4th floor			
Address 2				
City	San Francisco	State/Province	CA	
Postal Code	94105	Country ⁱ	US	
Inventor	4	Remove		
Legal Name				

Prefix	Given Name	Middle Name	Family Name	Suffix
	Ahmed		Ali	
Residence Information (Select One) US Residency ☒ Non US Residency Active US Military Service				
City	Islamabad	Country of Residence ⁱ	PK	

Mailing Address of Inventor:

Address 1	55 Hawthorne St 4th floor			
Address 2				
City	San Francisco	State/Province	CA	
Postal Code	94105	Country ⁱ	US	
Inventor	5	Remove		
Legal Name				

Prefix	Given Name	Middle Name	Family Name	Suffix
	Hussam	Ullah	Khan	
Residence Information (Select One) US Residency ☒ Non US Residency Active US Military Service				
City	Islamabad	Country of Residence ⁱ	PK	

Mailing Address of Inventor:

Address 1	55 Hawthorne St 4th floor			
Address 2				
City	San Francisco	State/Province	CA	
Postal Code	94105	Country ⁱ	US	

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Inventor	6 Remove			
Legal Name				
Prefix	Given Name	Middle Name	Family Name	Suffix
	Afsheen	Rafaqat	Ali	
Residence Information (Select One) ☐ US Residency ☒ Non US Residency ☐ Active US Military Service				
City	Islamabad	Country of Residence ⁱ	PK	
Mailing Address of Inventor:				
Address 1	55 Hawthorne St 4th floor			
Address 2				
City	San Francisco	State/Province	CA	
Postal Code	94105	Country ⁱ	US	
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button. Add				

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	33717		
Email Address	gtipmail@gtlaw.com	Add Email	Remove Email

Application Information:

Title of the Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		
Attorney Docket Number	189908.100002	Small Entity Status Claimed	☐
Application Type	Provisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		Suggested Figure for Publication (if any)	

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☒ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not be** the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	83717		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status		Remove
Application Number	Continuity Type	Prior Application Number
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.		Add

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)ⁱ the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ⁱ (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.

NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant	1	Remove
If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section. Clear		
Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
☐ Person to whom the inventor is obligated to assign.		☐ Person who shows sufficient proprietary interest
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:		
Name of the Deceased or Legally Incapacitated Inventor:		
If the Applicant is an Organization check here. ☒		
Organization Name	Keep Truckin, Inc.	
Mailing Address Information For Applicant:		
Address 1	55 Hawthorne St. 4th floor	
Address 2		
City	San Francisco	State/Province
Country	US	Postal Code
Phone Number		Fax Number
Email Address		
Additional Applicant Data may be generated within this form by selecting the Add button. Add		

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

Assignee	1			
Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.				
				Remove
If the Assignee or Non-Applicant Assignee is an Organization check here.				☐
Prefix	Given Name	Middle Name	Family Name	Suffix
Mailing Address Information For Assignee including Non-Applicant Assignee:				
Address 1				
Address 2				
City		State/Province		
Country ⁱ		Postal Code		
Phone Number		Fax Number		
Email Address				
Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.				Add

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). However, if this Application Data Sheet is submitted with the **INITIAL** filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Vivek Ganti/		Date (YYYY-MM-DD)	2020-02-12
First Name	Vivek	Last Name	Ganti	Registration Number
				71,368
Additional Signature may be generated within this form by selecting the Add button.				Add

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	189908.100002
		Application Number	
Title of Invention	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA		

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- 1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
- 2 A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4 A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5 A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6 A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7 A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8 A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
- 9 A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal

Application Number:				
Filing Date:				
Title of Invention:	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA			
First Named Inventor/Applicant Name:	Wajahat Kazmi			
Filer:	Vivek Ganti/Belinda Love			
Attorney Docket Number:	189908.100002			
Filed as Large Entity				
Filing Fees for Provisional				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
PROVISIONAL APPLICATION FILING	1005	1	280	280
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				280

Electronic Acknowledgement Receipt

EFS ID:	38564285
Application Number:	62975360
International Application Number:	
Confirmation Number:	9473
Title of Invention:	DISTANCE ESTIMATION USING SINGLE VIEW CAMERA
First Named Inventor/Applicant Name:	Wajahat Kazmi
Customer Number:	33717
Filer:	Vivek Ganti/Belinda Love
Filer Authorized By:	Vivek Ganti
Attorney Docket Number:	189908.100002
Receipt Date:	12-FEB-2020
Filing Date:	
Time Stamp:	14:15:43
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	\$280
RAM confirmation Number	E20202BE16104543
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	189908-100002_Transmittal_PROV_dated_2-12-20.pdf	2032430	no	4
			4bc21bef68e908fd3d66cb78a7e77e19ca9f8bc4		
Warnings:					
Information:					
2	Specification	189908-100002_Specification_PROV_dated_2-12-20.pdf	139025	no	3
			b632241b6b54d9d908b9acb5fecb9021e550ce64		
Warnings:					
Information:					
3	Appendix to the Specification	Appendix_A.PDF	10994178	no	10
			700cc4acdeb16e9b9fb5158d15750ae441d6e5bf		
Warnings:					
Information:					
4	Appendix to the Specification	Appendix_B.PDF	19656572	no	10
			91d1feb41a12f627583b28acd47786414f23b3f3		
Warnings:					
Information:					
5	Application Data Sheet	189908-100002_ADS_PROV_dated_2-12-20.pdf	1256283	no	10
			1362b9624b291babbbd1e258164e45ead476eed7		
Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	30036	no	2
			99c35cba80b19454ccb4d3db62f69c1fbb06c3b0		
Warnings:					
Information:					

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
62/975,360	02/12/2020		280	189908.100002		

CONFIRMATION NO. 9473

FILING RECEIPT



0000000114778987

33717
GREENBERG TRAURIG LLP (GT)
C/O: GREENBERG TRAURIG
INTELLECTUAL PROPERTY DEPARTMENT
77 WEST WACKER DRIVE, SUITE 3100
CHICAGO, IL 60601

Date Mailed: 02/19/2020

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt identifying the requested changes, preferably by including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

Wajahat Kazmi, Islamabad, PAKISTAN;
Aamer Zaheer, Islamabad, PAKISTAN;
Ali Hassan, Islamabad, PAKISTAN;
Ahmed Ali, Islamabad, PAKISTAN;
Hussam Ullah Khan, Islamabad, PAKISTAN;
Afsheen Rafaqat Ali, Islamabad, PAKISTAN;

Applicant(s)

Keep Truckin, Inc., San Francisco, CA

Power of Attorney:

Vivek Ganti--71368

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/18/2020

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 62/975,360**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No

Title

DISTANCE ESTIMATION USING SINGLE VIEW CAMERA

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.