# Computer Vision Analysis for Vehicular Safety Applications

Yuan-Fang Wang
Department of Computer Science
University of California
Santa Barbara, CA 93106
yfwang@cs.ucsb.edu

**Abstract**

In this paper, we present our research on using computer-vision analysis for vehicular safety applications. Our research has potential applications for both autonomous vehicles and connected vehicles. In particular, for connected vehicles, we propose three image analysis algorithms that enhance the quality of a vehicle's on-board video before inter-vehicular information exchange takes place. For autonomous vehicles, we are investigating a visual analysis scheme for collision avoidance during back up and an algorithm for automated 3D map building. These algorithms are relevant to the telemetering domain as they involve determining the relative pose between a vehicle and other vehicles on the road, or between a vehicle and its 3D driving environment, or between a vehicle and obstacles surrounding the vehicle.

**Keywords:** photo,video, vision, safety, autonomous, connected, vehicle

## 1 Introduction

Advances in video technology have enabled its wide adoption in the auto industry. Today, many vehicles are equipped with backup, front-looking, and side-looking cameras that allow the driver to easily monitor the traffic around the vehicle for enhanced safety.

Many vehicular technologies, particularly those for safety and guidance, can roughly be classified into autonomous (autonomous vehicles) [13, 12, 11, 18] and collaborative (connected vehicles) [16, 9] schemes. In an autonomous scheme, a vehicle relies mostly on its own on-board sensor to produce environmental data for navigation and collision avoidance. Significant strides have been made in autonomous navigation, e.g., Google's Self Driving Car [11] and similar projects from major auto manufacturers like BWM [12], Audi [2] and Benz [18]. Popular safety features including lane detection and departure warning, $360^o$ surround panoramic display, backover warning, and adaptive cruise control, to name a few [1, 14, 7, 8, 15, 25, 23, 22, 17, 21, 10, 24].

More recently, WAVE (wireless access in vehicular environments)/DSRC (dedicated short-range communications) [19] has become an attractive technology for vehicular safety applications. Vehicles with WAVE/DSRC devices can communicate with their neighboring vehicles to exchange information to achieve collaborative safety. For example, if a front vehicle applies brake suddenly, a warning signal can cascade into the vehicles that follow. Such vehicle-to-vehicle (V-2-V) communication is currently being tested for warning like collision, ice on the road, reduced visibility due to fog, etc., on the road ahead [18, 12].

Our own research in this exciting application area has been on both autonomous and collaborative schemes for vehicular safety and navigation applications. In collaborative schemes, we have developed three different algorithms to enhance a vehicle's on-board video before inter-vehicular information exchange. In autonomous schemes, we are investigating automated generation of back-over warning and 3D envoromental maping. The goal of these research is to develop intelligent image analysis algorithms that are robust and efficient for implementing on a vehicle's on-board processors to achieve both stand-alone and collaborative safety monitoring and situation awareness.

.

While Google's Self-Driving Car [1] has many sensors to enable safety monitoring and 3D map building (e.g., GPS, gyroscope, shaft encoder, and 3D LiDAR), these sensors are often expensive and bulky; requiring complex circuity to link together and on-board computing and recording devices to function. In particular, 3D data come from 3D laser radar [2] that can be many times more expensive than the host vehicle itself. In contrast, our analysis algorithms [5, 6] and 3D map building pipeline [28] have relied on a single video camera, but no other sensor to function. That is, we address the most difficult and data-deprived scenario for vehicular safety and navigation applications. Furthermore, these algorithms are relevant to the telemetering domain as they involve determining the relative pose (orientation and distance) between a vehicle and other vehicles on the road, or between a vehicle and its 3D driving environment, or between a vehicle and obstacles surrounding the vehicle.

The remainder of the paper will give brief introduction to these collaborative (Sec. 2) and autonomous (Sec. 3) safety and navigation algorithms, followed by a concluding remark pointing out possible future research directions.

# 2 Collaborative schemes

We envision that in the future, vehicles will be able to exchange a reasonable amount of information instantaneously. In particular, images and video captured by an on-board camera can be relayed to drivers of nearby vehicles to enhance situation awareness and provide road hazard warning. Before such information exchange commences, it is often desirable to clean up blockage and imperfection seen in such videos. These blockages may include vehicles on the road in front of the host vehicle that block the view of the host drive, activated wipers visible through the wind shield, and motion jitters caused by driving on rough roads. We propose three algorithms that allow such image imperfection to be compensated for, which are discussed below.

## 2.1 Translucentization of Road Obstacles

We propose a new vehicle blind spot elimination system which utilizes the on-board videos captured from multiple vehicles to make vehicles on the road translucent [5]. The particular scenario involves two vehicles, one following the other. As a result, the view of the back vehicle is often partially blocked by the front vehicle. If both vehicles are equipped with a front-looking camera and a V-2-V communication device which allows the exchange of the video streams, it is then possible to replace the image of the front vehicle in the video stream of the back vehicle by what the front vehicle sees, and thus eliminate the blind spot created by the front vehicle in the back cameras video stream.

This data analysis and fusion paradigm is best understood by examples, and four are shown in Fig. 1 (one per row). The left column of Fig. 1 shows what the front vehicle sees, and the middle column shows what the back vehicle sees. Depending on the separation between the front and back vehicles, the view of the back vehicle is partially blocked by the front vehicle. In some cases, the blockage can severely limit the ability of the driver of the back vehicle to interpret the road condition ahead. On the right column, we show that by using our sensor data fusion algorithm, it is possible to make the front vehicle "translucent" in the back video frame, and hence, provide a much better visual feedback to the driver of the back vehicle of the road ahead.

The proposed algorithm uses image analysis to achieve sensor data registration and fusion, and *does not rely on any other external sensors such as GPS and gyroscope*. Furthermore, the visualization shown in the right column of Fig. 1 is from the point of view of the host driver, not simply displaying the videos of the front, blocking vehicle to the host driver.

The gist of the algorithm is to use image analysis methods to infer the relative pose between the two cameras, identify the image location of the front vehicle in the back image frame, and blend the front image content into the back image around the front vehicle location. The process makes use of feature correspondences extracted and identified in these images—that is, objects seen by both cameras. Such feature correspondences enable the following computations: (1) sensor registration: ascertain the relative pose (rotation and translation) of the two cameras, (2) vehicle localization:

.

Figure 1: Left: image seen by the front camera, center: image seen by the back camera, and right: the back image with blended-in content from the front image.

determine the locations of the epipoles, or the projected location of the (front, back) camera's optical center in the image plane of the (back, front) camera, and (3) data fusion: compute both a mapping equation and the size of the mapping region where pixel values of the front image are blended into the corresponding pixels in the back image.

## 2.2 Removal of Wipers

As illustrated above, in V-2-V communication, a vehicle's onboard video can be processed automatically, say, to remove undesirable motion blur and pixel blockage before re-transmission to others for heightened situation awareness and driver assistance. Here, we report yet another video refinement scheme in detecting and localizing wiper pixels and replacing them with the corresponding unblocked pixels from an adjacent non-wiper frame in the onboard video to improve the visual feedback to the drivers. We address two problems here: that of wiper detection and localization and, once the presence of a wiper is identified, to mask off the wiper pixels with suitable unblocked, non-wiper pixels from an adjacent video frame for enhanced viewing.

This process is best illustrated by an example. As shown in Fig. 2, during the raining days, many video frames may

.

Figure 2: (left) a video frame with the wiper presence, (middle) the wiper frame with the wiper detected and localized automatically by our computer program (wiper pixels are marked in red), and (right) wiper pixels in the wiper frame are replaced by the corresponding pixels in the adjacent non-wiper frame to provide better viewing of the surrounding.

have the wiper presence that blocks out the road and vehicles on the road (Fig. 2 left column). If we can detect and localize the wiper pixels in such a wiper frame (Fig. 2 middle column) and find a suitable adjacent video frame without the wiper presence, we can then replace those wiper pixels in the wiper frame with the corresponding pixels in the adjacent non-wiper frame (Fig 2 right column) to enhance the viewing feedback to other drivers through V-2-V communication.

Note that while in-painted regions in a wiper frame may not faithfully depict objects on the road at that particular time instance — as the in-painted pixels are extrapolated from an adjacent, non-wiper frame — the improved visual feedback to the driver is useful as wiper blockage can be significant as shown in Fig.2. Furthermore, as was shown in [6] , a wiper frame and the adjacent non-wiper frame used for masking the wiper pixels are separated by tens of milliseconds at most. Hence, the changes of road conditions are often negligible in such cases. We believe that such a wiper-masking system can be of great assistance to the driver in improving visualization and situation awareness.

## 2.3 Jitters Compensation

Motion jitters in a vehicle's on-board video is often unavoidable due to rough road condition. Our design of a jitter-compensation algorithm comprises two major components: a pair-wise jitter-compensation method that operates two frames (not necessarily consecutive) at a time, and a top-level driver routine that processes a whole video sequence, calling the pair-wise algorithm along the way. We describe these two processes in more details below.

For the pair-wise jitter-compensation algorithm, we model the corresponding pixel movements in between the back and front image frames as comprising two terms: (1) a zoom — induced by a 3D $z$ movement - around the optical center, and (2) jitter — induced by 3D $x$ and $y$ movements. The algorithm then tries to ascertain the parameters in this simple T+S model and removes the unwanted jitter component.

Let's denote the earlier frame the back frame and the later frame the front frame as the vehicle will have moved forward in time. The goal is to shift the front image in the $x - y$ plane to align with the back image, and optionally fill in the void in the front image — due to jitter compensation — with appropriate pixels from the back image. The processing flow is briefly summarized as: 1) Detect features in both the front and back image frames, 2) Match the detected features by descriptor similarity, 3) Filter out erroneous feature pairs by imposing the epipolar constraint, and 4) Use

.

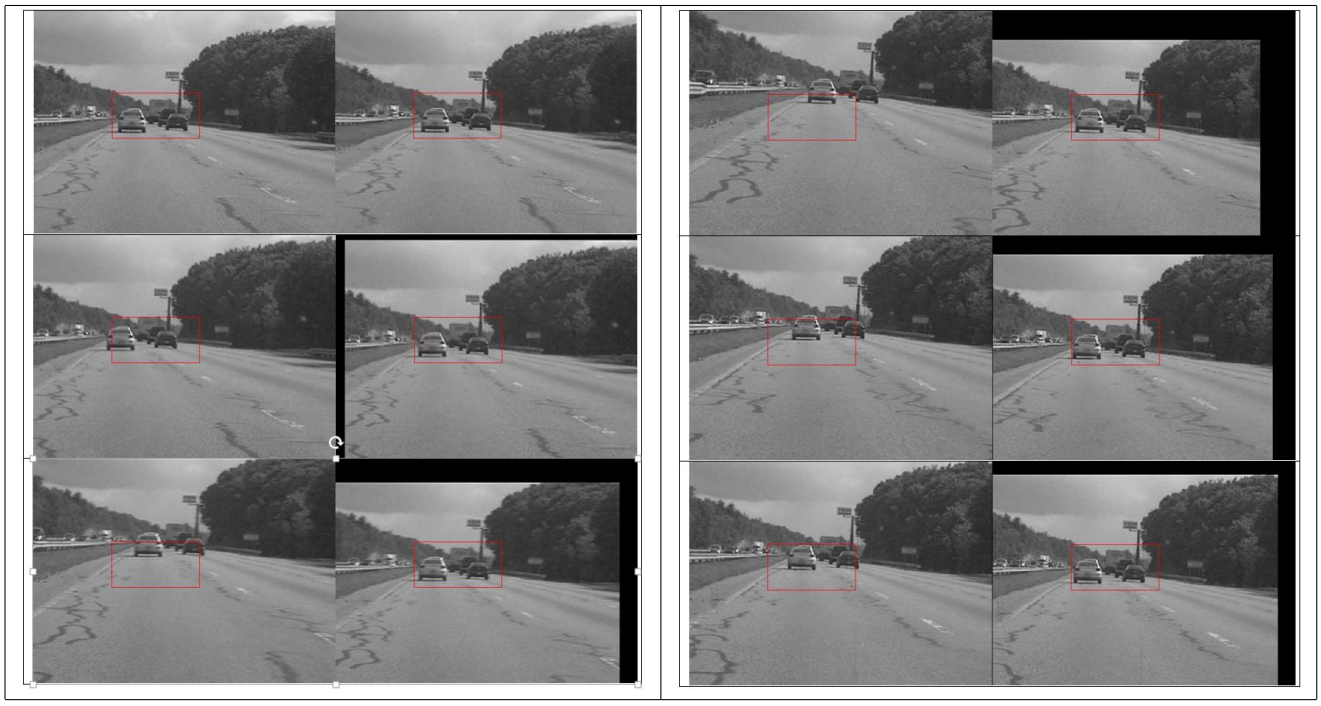

Figure 3: Left: images without jitters compensation and right: images with jitters compensation. The results are shown from top to bottom in two columns.

the cleaned feature correspondences to compute the amount of zoom and jitter in between the front and back images.

Finally, jitter compensation is a four-step process executed in the following order: a) Map the back image to temporary output (1), b) Map the front image to temporary output (2), c) Properly merge the two temporary output images, and d) Blend over the border region in between the temporary output (1) and (2).

in Fig. 3 a number of frames of a driving sequence where large jitter and vibration was observed. Each pair of images (left and right in two columns) in the figure shows the original image on the left and jitter-compensated image on the right. The red box marks the position of the vehicles on the road as observed in the first frame. As the time progressed, the position of the road vehicles experienced very large $x - y$ shift as shown on the left while the jitter-compensated vehicle positions were quite stable as shown on the right. The amount of black swaths around the image periphery gave a good indication as to the size of the jitter.

# 3 Autonomous Schemes

Autonomous schemes, as the name suggests, rely on the on-board sensor data to make navigation and safety decisions. A vehicle can perform such processing without the help rendered by neighboring vehicles. We look at two specific problems here: 3D map building and backover warning.

## 3.1 3D Environmental Map Building

The particular application of 3D environmental map building deserves some discussion. While Google's Self-Driving Car [1] has many sensors to enable building such a 3D environmental map (e.g., GPS, gyroscope, shaft encoder, and 3D LiDAR), these sensors are often expensive and bulky; requiring complex circuity to link together and on-board

.

computing and recording devices to function. In particular, 3D data come from 3D laser radar [2] that can be many times more expensive than the host vehicle itself. In contrast, *our 3D map building pipeline [22] has relied on a single, front-looking video camera, but no other sensor to function*. That is, we address the most difficult and data-deprived scenario. Furthermore, we have developed a new tracker design that makes video-based 3D environmental map building much more efficient.

Our 3D modeling pipeline is called PhotoModel3D. PhotoModel3D [26, 27, 28] employs a photo- and video-based analysis paradigm known as structure from motion (SfM) [3,21]. The principles are to exploit the motion parallax effect exhibited in multiple images taken by a moving camera to infer the 3D scene structures and the camera poses. PhotoModel3D (1) works with both discrete images and continuous videos taken by any consumer-market digital camera, camcorder, or phone, (2) uses no special equipment (e.g., lens and tripod), active projection, artificial lighting, prior camera calibration, and man-made registration markers, (3) functions both indoor and outdoor and over long range, unlike many commercial RGBD sensors that are mostly for indoor, short-range applications, (4) requires no user training (just point and shoot), (5) is fully automated and end-to-end (from photographs to fully colored and textured 3D models) without manual intervention or data-specific parameter tuning, (6) is a software-based solution that runs on commodity Linux and Windows servers without the need of special hardware (GPU, DSP, etc.) acceleration.

A video-based 3D pipeline like PhotoModel3D often involves detecting and matching features across the video frames. These detection and correspondence processes must be carried out with care as they form the basis of all ensuing analysis. Our experience has been that the feature analysis process can take anywhere from 15% to 40% of the total processing time in SfM. Exploiting a fast feature tracker can be advantageous in reducing the computational power on a mobile device and allowing appropriate client-server partition of the 3D mapping pipeline.

To this end, we have performed research on evaluating many existing object tracking techniques, and proposing a new tracker design and its application for 3D environmental mapping in vehicular technology applications [4]. The contribution of our research is 4-fold: (1) We evaluate a large collection of state-of-the-art trackers using multiple criteria relevant to vehicular technology applications, (2) we show how to derive useful evaluation metrics from public-domain, real-world driving videos that do not come with ground-truth information on pixel tracking, (3) we propose a new tracker that is geared specifically for vehicular technology application and show that it achieves tracking accuracy that outperforms SIFT and is on-par with state-of-the-art optical-flow tracking algorithm, which has the best accuracy in our evaluation. Furthermore, we show that our tracker is 600 times more efficient than optical flow and 7 times more efficient than SIFT, and (4) we validated our new tracker design for 3D environmental map building application and showed that the new tracker can obtain comparable results as SIFT but with a significant saving in runtime.

Our new feature tracker is called ZBRIEF. It combined the FAST corner detector [20] with the BRIEF descriptor [3] for feature tracking. We further extended the BRIEF descriptor with a zoom factor to better describe the zoom image flow, which is prevalent in driving applications [4]. We have conducted experiments to replace SIFT and SURF with ZBRIEF in our 3D pipeline, while keeping all other components the same, and then compare the resulting 3D models. Two such examples are shown in Fig. 4. The left column shows a sample image in the video sequence of length 35 (top) and 45 (bottom), respectively. The middle column shows the 3D models based on SIFT feature analysis while the right column the corresponding 3D models using ZBRIEF analysis. Note that these models differ only in the 2D feature used; all other processing components, parameter settings, and input photos are exactly the same. The ZBRIEF models are qualitatively similar to the SIFT models but were generated about 30% faster.

## 3.2 Obstacle Detection during Back up

Collision detection while backing up is a much more challenging problem than that on highways. Driving on a highway, one expects to encounter certain classes of objects, such as trucks, cars, and motorcycles mostly. Potential collision implies observing the front (back collision) or back (front collision) of a vehicle at a close range. Wheels, headlights, and license plates have a semi-regular placement and similar appearance that enables a machine-learning algorithm to discern the telltale signs of a vehicle in close proximity to sound the collision alarm. However, obstacles

.

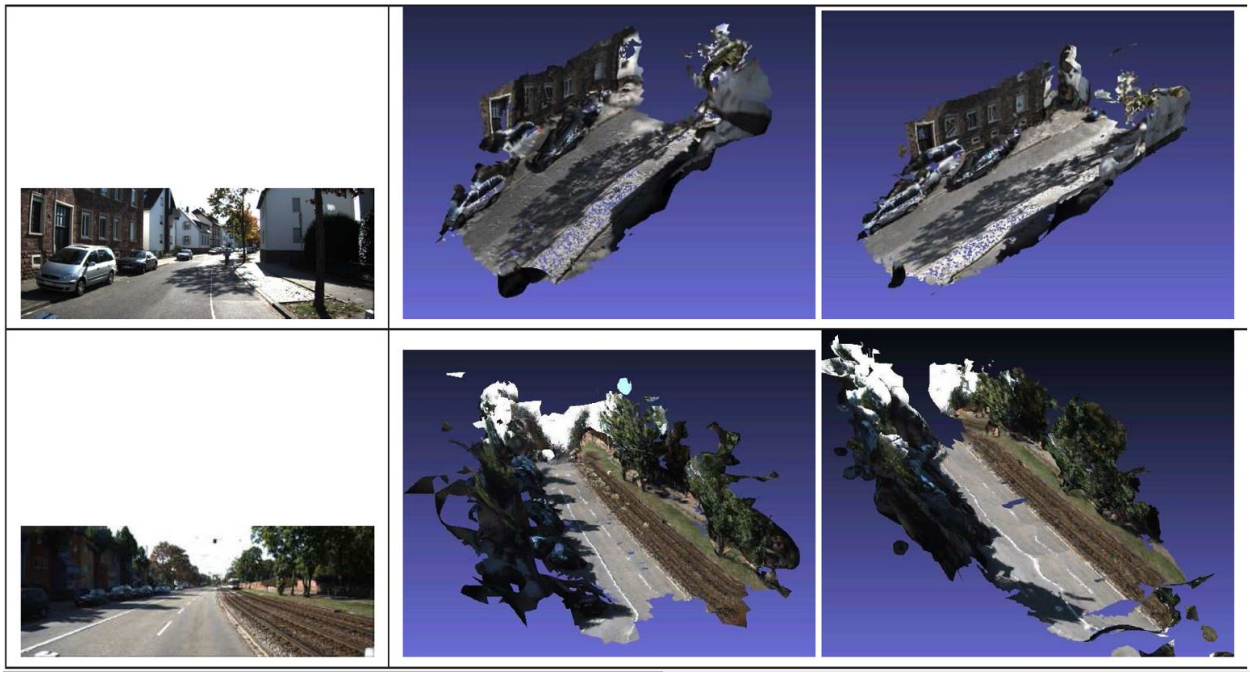

Figure 4: Left: a sample image in the video sequence, middle: 3D models using SIFT features and right: the corresponding models using ZBRIEF features.

that one can back into vary greatly (fence, bush, tree, light post, traffic sign, garbage can, mailbox, water pound, people, dog, cat, toy, furniture, grocery cart, bike, motorcycle, other vehicle, just to name a few). It is hard to imagine that an exhaustive list of obstacles can be defined and their appearance learned *a priori* for collision avoidance during backup. Hence, if it is possible to design a backup collision detection system without prior learning is important.

As is well known, time-to-contact can be estimated by a simple ratio expression $TTC = -Z/V$, where $Z$ is the distance to the obstacle and $V$ is the vehicle's speed. A collision warning is issued when TTC is lower than a certain threshold — typically 2 seconds. If certain length quantities, e.g., the width of a vehicle, can be tracked over time, it is possible to relate the rate of change of such a length measure to the TTC. This is because the perceived dimension ($w$) is proportional to the true dimension ($W$) by $w = fW/Z$, where $f$ is the focal length and $Z$ is the distance to the camera. Hence, the ratio of the perceived lengths at two time instants $t$ and $t' = t + \Delta t$ are related by $S = w'/w = (fW/Z')/(fW/Z) = Z/Z'$. Furthermore, $Z' = Z + V\Delta t$ where $V$ is the speed of the vehicle. Hence, we have

$$S = \frac{Z' - V\Delta t}{Z'} = \frac{Z'/V - \Delta t}{Z'/V} = \frac{TTC - \Delta t}{TTC} \tag{1}$$

Or

$$TTC = \frac{\Delta t}{1 - S} \tag{2}$$

However as noted before, there are myriads of potential obstacles during backing up, and hence, no well defined telltale signs—such as a vehicle's width and contact point with the road—are to be tracked always. In fact, in a static scene, distant obstacles and pavement can all approach the vehicle as it backs up. TTC can be calculated based on the temporal ratio of the distance between, say, a pair of tracked points on these objects. Hence, there must be a way to tell pavement apart from true obstacles with a similar TTC signature to avoid generating many falsely positive warnings. Absent a general segmentation scheme, a reasonable assumption is that pavement is flat while obstacles most likely are not. This assumption is violated if one is to back into an upright wall, which is also a flat structure. That is, planar structures of different orientations and heights have to register differently with the back-over warning system.

.

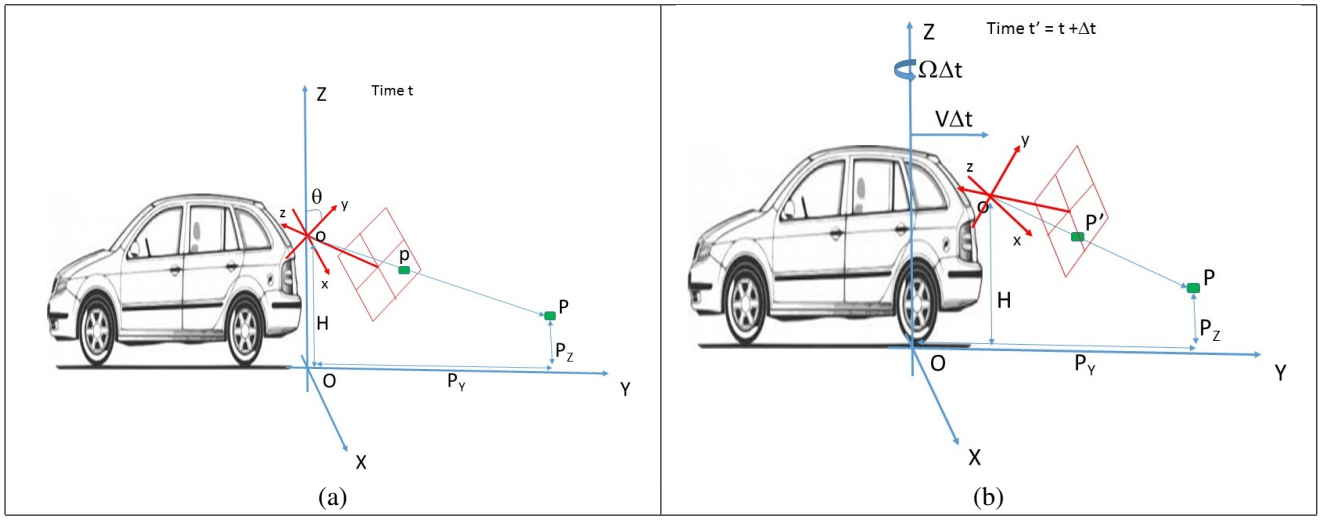


Figure 5: Homography relations for the ground plane.

Collision warning should not be issued when tracked points are identified to be on a horizontal planar structure at the height of the ground plane (i.e., the pavement).

Hence, the ability to identify the ground plane is important. Anything that is above the ground plane, even of a planar structure, should signal the presence of an obstacle. One way to identify a planar structure is to observe that tracked points on a planar structure obey the homography constraint, or $\mathbf{p}' = \mathbf{H}\mathbf{p}$, where $\mathbf{p}$ and $\mathbf{p}'$ are tracked 2D feature points at time $t$ and $t'$ of a planar 3D structure. Essentially, homography represents a 3D cue without explicit 3D scene reconstruction, as an explicit 3D analysis using a small number of video frames can be quite challenging. However, as stated earlier, tracked points on an upright wall will also satisfy the homography constraint. It therefore important to distinguish homography induced by planar structures of different height and orientation and only accepts the homography relation for a horizontal plane at the ground level to be pavement.

In more detail, consider the diagram shown in Fig. 5(a), which depicts a vehicle with a backup camera. We assume that the height ($H$) and the pitch angle ($\theta$, measured downward from the horizontal plane at height $H$) of the camera are known. We further assume that the camera's roll and yaw angles are zero, as the backup camera should be mounted upright and has no left or right preference in the viewing direction.

We consider two coordinate systems here: one associated with the world (**O-X-Y-Z**) and the other with the camera (**o-x-y-z**). Without loss of generality, we assume that, at time $t$, the world system is centered below the focal point of the camera on the ground, with the $X$ axis aligned with the width of the vehicle, the $Y$ axis aligned with the length of the vehicle, and the $Z$ axis pointing straight up. The origin of the camera's coordinate frame is at a height of $H$ above that of the world system, the $x$ axis points to the right, the $y$ axis points at the pitch angle $\theta$ relative to the world's $Z$ axis , and the $z$ axis points toward the vehicle, as shown in Fig. 5(a).

At time $t' = t + \Delta t$, the vehicle backs up to a new location. The motion is described by a linear velocity $V$along $Y$ of and an angular velocity $\Omega$ about $Z$ as shown in Fig. 5(b). The same 3D point $\mathbf{P}$ is imaged at $\mathbf{p}$ at time $t$ and **p'** at time $t'$. We would like to derive the special homography matrix for the set of ground-plane points $\mathbf{P}$ where $P_z = 0$. For such ground-plane point $\mathbf{P}$, we define $\hat{\mathbf{P}} = [P_x, P_y, 1]^T$ (because $P_z = 0$, we can drop that component). Homography relation implies

$$\begin{aligned} \mathbf{K^{-1}p} &= \mathbf{H}\hat{\mathbf{P}} \\ \mathbf{K^{-1}p}' &= \mathbf{H}'\hat{\mathbf{P}} \end{aligned} \tag{3}$$

where $\mathbf{K}$ represents the intrinsic camera matrix, Or we have

$$\mathbf{p}' = \mathbf{K}\mathbf{H}'\mathbf{H}^{-1}\mathbf{K}^{-1}\mathbf{p} \tag{4}$$

.

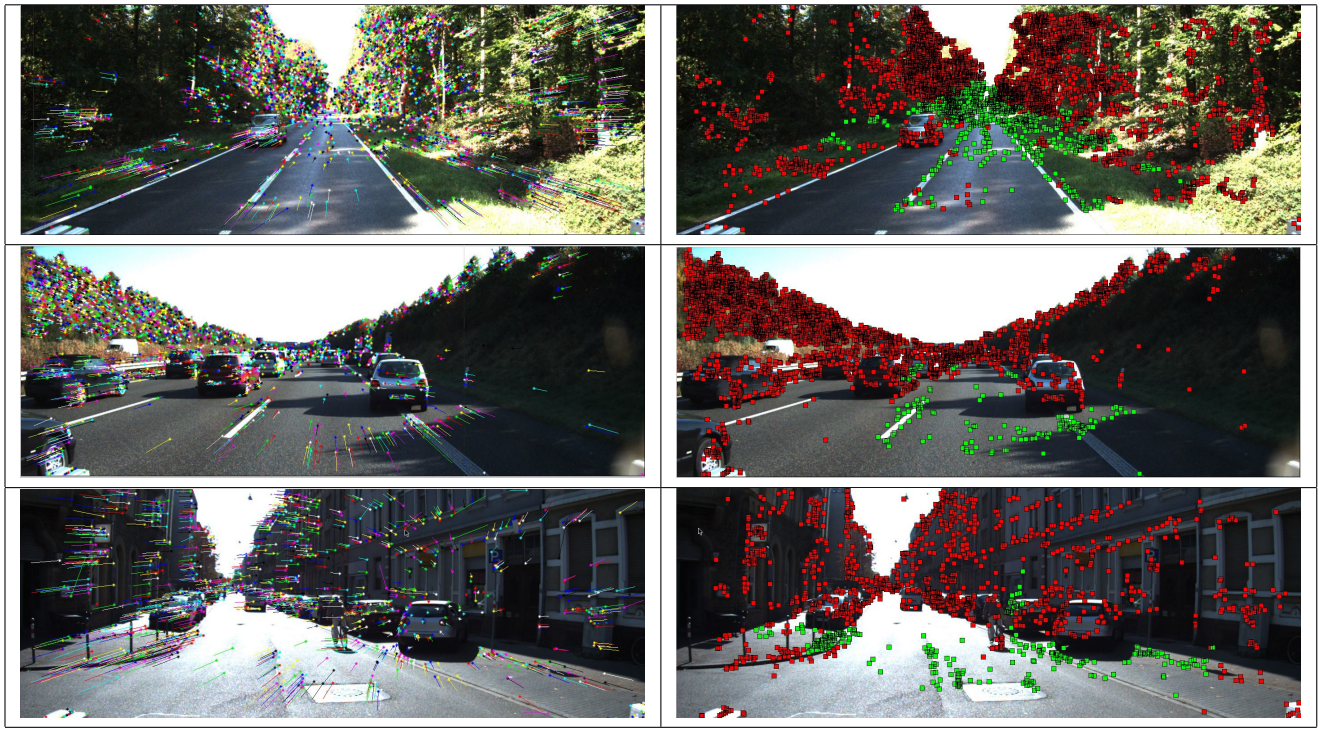

Figure 6: Homography relations for the ground plane. Left: detected features and motion vectors. Right: ground plane features in green and others in red.

which is a homography mapping using matrix $\mathbf{KH'H}^{-1}\mathbf{K}^{-1}$. Certainly, points on any 3D planar structure should satisfy such a relation, but with a different homography matrix. The goal is then to ascertain what the particular matrix for the ground plane is. Omitting the detailed derivation, it can be shown that Eq. 5 represents the homography relation between a plane at height $P_Z$ and its image projection.

$$\begin{aligned} \mathbf{p}' &= \mathbf{K}\begin{bmatrix} 1 & 0 & 0 \\ 0 & sin\theta & cos\theta \\ 0 & -cos\theta & sin\theta \end{bmatrix}\begin{bmatrix} cos\Omega\Delta t & sin\Omega\Delta t & 0 \\ -sin\Omega\Delta t & cos\Omega\Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & V\Delta t sin(\Omega\Delta t) \\ 0 & 1 & -V\Delta t cos(\Omega\Delta t) \\ 0 & 0 & P_Z - H \end{bmatrix}\begin{bmatrix} P_X \\ P_Y \\ 1 \end{bmatrix} \\ &= \mathbf{KH'}[P_X, P_Y, 1]^T \end{aligned} \tag{5}$$

In Fig. 6, we show some sample results (one per row). On the left the movement of tracked features (using our ZBRIEF detector) are overlayed on one of the image frames. The detected movements are then compared with those predicted by the homography model in Eq. 5. The deviation of the computed movement and the movement predicted by Eq. 5 is then thresholded to classify a feature as either on the ground (green) or not on the ground (red). As can be seen, vehicles (both stationary and moving) on the road were correctly identified as not on the ground plane with the host vehicle executing both straight-line (top two) and turning (bottom one) motion in Fig. 6. However, this simple model does suffer from some deficiency: (1) For objects in the far field, the feature movements will be very small (close to zero). Homography prediction could be inconclusive, and (2) other syntactic information, such as color and texture pattern, can provide powerful cues to the identity of a pixel. Hence, some sensor data integration scheme should be used to improve the robustness of the classification mechanism.

# 4 Concluding Remarks

In this paper, we presented our research on vehicular navigation and safety applications. A common trait of these research is their reliance on a single on-board video camera but no other sensors, which makes the schemes viable

.

with a low hardware cost. Another common trait is that they require telemetering processing to determine the relative pose and motion of the vehicle with respect to other vehicles on the road, environments and obstacles.

# References


[1] Ahlers Florian and Westhoff Daniel. Laserscanner-based Lane Detection and Relative Positioning. In *Proceedings of the 16th World Congress on ITS*, 2009.

[2] Audi. Audi Piloted Driving. http://www.audi.com/content/com/brand/en/vorsprung_durch_technik/content/2014/10/piloted-driving.html, 2014.

[3] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Trans. Pattern Analy. Machine Intell.*, 2012.

[4] Che-Tsung Lin and Long-Tai Chen and Yuan-Fang Wang. Evaluation, Design and Application of Object Tracking Technologies for Vehicular Technology Applications.

[5] Che-Tsung Lin and Yu-Chen Lin and Long-Tai Chen and Yuan-Fang Wang. Front Vehicle Blind Spot Translucentization Based on Augmented Reality. In *Proceedings of IEEE Vehicular Technology Conference*, 2013.

[6] Che-Tsung Lin and Yu-Chen Lin and Long-Tai Chen and Yuan-Fang Wang. Enhancing Vehicular Safety in Adverse Weather using Computer Vision Analysis. In *Proceedings of IEEE Vehicular Technology Conference*, 2014.

[7] E. Bertolazzi and F. Biral and M. Da Lio and A. Saroldi and F. Tango. Supporting Drivers in Keeping Safe Speed and Safe Distance. *IEEE Transactions on Intelligent Transportation Systems*, page 525538, Sep. 2010.

[8] N. Enache, S. Mammar, M. Netto, and B. Lusetti. Driver Steering Assistance for Lane-Departure Avoidance based on Hybrid Automata and Composite lyapunov Function. *IEEE Transactions on Intelligent Transportation Systems*, page 2839, Mar. 2010.

[9] E. T. S. I. ETSI. Intelligent Transport Systems Standards. http://www.etsi.org/index.php/technologies-clusters/technologies/intelligent-transport, 2015.

[10] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. S. asnd H. Ito, and T. Mikami. An Efficient Computational Architecture for a Collision Early Warning System for Vehicles, Pedestrians, and Bicyclists. *IEEE Transactions on Intelligent Transportation Systems*, page 942953, Dec. 2011.

[11] E. Guizzo. How Google's Self-Driving Car Works. *IEEE Spectrum*, Feb. 2013.

[12] HAVEit. European Union Research Project HAVEit. http://www.haveit-eu.org/, 2015.

[13] Hoeger Reiner and Amditis Angelos and Kunert Martin and Hoess Alfred and Flemish Frank and Krueger Hans-Peter and Bartels Arne and Beutner Achim and Pagle Katia. Highly Automated Vehicles for Intelligent Transport: HAVEit Approach. In *Proceedings of the 15th World Congress on ITS*, 2008.

[14] K. Kusano and H. Gabler. Safety Benefits of Forward Collision Warning, Brake Assist, and Autonomous Braking Systems in Rear-End Collisions. *IEEE Transactions on Intelligent Transportation Systems*, pages 1546–1555, 2012.

[15] D. F. Llorca, V. Milanes, I. P. Alonso, M. Gavilan, I. G. Daza, and M. A. Sotelo. Autonomous Pedestrian Collision Avoidance using a Fuzzy Steering Controller. *IEEE Transactions on Intelligent Transportation Systems*, page 390401, Jun. 2011.

[16] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. Mark. Connected Vehicles: Solutions and Challenges. *IEEE Internet of Things Journal*, Aug. 2014.

[17] Nissan. Nissan Around View Monitor. http://www.nissan-global.com/EN/NEWS/2007/_STORY/071012-01-e.html, 2007.

[18] I. of Engineering and Technology. Autonomous vehicles from Mercedes, BMW and Audi debut at CES. http://eandt.theiet.org/news/2015/jan/autonomous-cars-ces.cfm, 2015.

[19] U. D. of Transportation. IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE), Apr. 2013.

[20] E. Rosten and T. Drummond. Fusing Points and Lines for High Performance Tracking. In *Proc. Int. Conf. Comput. Vision*, volume 2, pages 1508–1511, October 2005.

[21] R. Schubert, K. Schulze, and G.Wanielik. Situation Assessment for Automatic Lane-Change Maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, page 607616, Sep. 2010.

[22] B. Southall, M. Bansal, and J. Eledath. Real-time Vehicle Detection for Highway Driving. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 2009.

[23] Z. Sun, G. Bebies, and R. Millier. On-Road Vehicle Detection Using Optical Sensors: A Review. *IEEE Trans. Pattern Analy. Machine Intell.*, 2014.

[24] K.-J. Tu and J.-F. Kiang. Estimation on Location, Velocity, and Acceleration with High Precision for Collision Avoidance. *IEEE Transactions on Intelligent Transportation Systems*, page 374379, Jun. 2010.

[25] V.Milanes, J. Godoy, J. Villagra, and J. Perez. Automated On-ramp Merging System for Congested Traffic Situations. *IEEE Transactions on Intelligent Transportation Systems*, page 500508, Jun. 2011.

[26] Y.-F. Wang. UCSB Computer Vision Lab Video Demo Pages. http://www.cs.ucsb.edu/~yfwang/demos.html, 2011-2014.

[27] Y.-F. Wang. Photomodel3D: UCSB Computer Vision Lab Public-Use 3D Modeling Server (primary). http://rogue.cs.ucsb.edu/PhotoModel3D/webUpload.html, 2015.

[28] Yuan-Fang Wang. Photo- and Video-based Ranging and Modeling. In *International Telemetering Conference*, 2014.