Predict Vehicle Collision by TTC From Motion Using a Single Video Camera

Mehmet Kilicarslan, *Student Member, IEEE*, and Jiang Yu Zheng, *Senior Member, IEEE*

***Abstract*— The objective of this paper is the instantaneous computation of time-to-collision (TTC) for potential collision only from the motion information captured with a vehicle borne camera. The contribution is the detection of dangerous events and degree directly from motion divergence in the driving video, which is also a clue used by human drivers. Both horizontal and vertical motion divergence are analyzed simultaneously in several collision sensitive zones. The video data are condensed to the motion profiles both horizontally and vertically in the lower half of the video to show motion trajectories directly as edge traces. Stable motion traces of linear feature components are obtained through filtering in the motion profiles. As a result, this avoids object recognition and sophisticated depth sensing in prior. The fine velocity computation yields reasonable TTC accuracy so that a video camera can achieve collision avoidance alone from the size changes of visual patterns. We have tested the algorithm for various roads, environments, and traffic, and shown results by visualization in the motion profiles for overall evaluation.**

***Index Terms*— ADAS, TTC, collision avoidance, driving video, computer vision, motion profile, spatial-temporal filtering.**

I. Introduction

COLLISION avoidance has been studied extensively for driver assistance systems over decades. LiDAR and Radar are two main range sensors used in finding depth to target. However, the collision time not only depends on the depth, but also depends on the relative speed. On the other hand, video cameras have also been used on vehicles for detecting vehicles and pedestrians. They are also used in object recognition such as lane marks and road edges, for which LiDAR and Radar are incapable of doing in some cases.

Although there have been success of using cameras on target recognition coupling tracking with bounding boxes, these methods focus mainly on rear side appearance and they are computationally expensive for real time detection. Main challenges in recognition are vehicle variations, dynamic background, and disturbance in tracking. There are still errors in vehicle recognition and disturbances in tracking scenes with rapidly changing environment due to the vehicle shaking, scene occlusion, and shape deformation. The first fatal accident of autonomous vehicle was with a truck missed in object learning algorithms and recognition.

Manuscript received May 19, 2017; revised January 5, 2018; accepted March 11, 2018. The Associate Editor for this paper was Q. Ji. *(Corresponding author: Mehmet Kilicarslan.)*

The authors are with the Department of Computer Science, Indiana University—Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: mkilicar@indiana.edu; jzheng@iupui.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2819827

We have noticed that human drivers can perceive approaching vehicles from target motion in the field of view. Particularly, the collision danger can be estimated from an enlarging object over a short period of time, even if its depth is sensed inaccurately [1], [2]. In this work, we solely rely on motion feature in driving video to identify potential collision in all directions without requiring any shape recognition in prior and depth estimation with stereo cameras. We focus on the non-transitive flow in the video for approaching target during the vehicle motion. For those targets with zero-flow, the Time-to-collision (TTC) is computed from the flow diverging rate. For a certain direction, we know how long a collision will happen if the relative motion of both camera/vehicle and target are continued. Based on that, prec-ollision breaking or target avoidance can be applied.

In previous works on collision detection, a target vehicle has to be identified first with the Haar-type operators via training [3] and a bounding box is fitted onto it for tracking [4]. Most of the systems are outlined in survey paper [5] for both vision and range sensors. Recently, more progress has been reported on vehicle recognition based on deep learning. Such methods are based on exhaustive learning of huge data sets. Even the recognized object marked with a bounding box, it is not always precise and smooth for TTC estimation, particularly when a vehicle is viewed from side view or an occlusion happens.

The TTC based on point tracking [6] can only identify the motion in parallel to the vehicle heading direction, which yields the Time-to-passing (TTP) for most of the passing points, rather than TTC of vehicles approaching to the camera relatively. Therefore, other vehicle approaching nonparallel to the camera/vehicle heading direction, and vehicles on curved roads cannot be alarmed. A tracking of consecutive frames has to grasp the size and position of bounding box for understanding vehicle depth [7].

Different from previous works, our method uses simple motion cues to directly obtain TTC without vehicle recognition. A dangerous collision from mid-range happens when an object approaches to the camera in a certain direction. This generates a zero-flow (optical flow close to zero) in the view [6], [8]. The TTC of target thus can be obtained instantly those directions, which is computed further from the object size divided by its size change according to the rule in [9] and [10].

On a motion sensitive belt over the horizon in the video, we detect the horizontal zero-flow spots, and then monitor the scene divergence vertically in the crossing vertical zones in video frames to avoid the object recognition and tracking with bounding box. These steps are implemented efficiently in

the motion profiles condensed from the belt and zones [11]. We compute dense horizontal motion and detect the horizontal zero-flow spots in the motion profile. A longer motion than traditional between-frame optical flow [6] is estimated.

The motion profile summarizes distinct objects and blurs small details. This generates dense flow as strong evidence of targets, since linear features are stable as compared to corner points with rich occlusion in driving video. Only vehicles, object rims, and road edges become visible in the motion profile. Another benefit of condensing is to reduce image to one dimensional data for fast computation. The extraction of potential collision from zero-flow also ignores most background and non-danger vehicles at early stage [12].

At the same time, the horizontal orientation in the entire view is divided to many zones. In the zero-flow zones, the color is further condensed (averaged) horizontally for examining the vertical motion. Based on that, convergence/divergence factor is computed from clusters of motion trajectories to confirm approaching vehicles, exclude leaving vehicles, and follow the vehicles moving in parallel. The TTC is thus obtained for collision alarming.

In the next section, we introduce the collision scenarios and the TTC calculation from object size and size change. We describe our motion data collection in Section III for the zero-flow with possible danger. Section IV is to confirm the flow divergence for collision alarming. Section V computes the Time-to-collision supported by experiments and evaluation in Section VI.

II. Potential Collision and Alert Scenarios

A. Potential Collision on Road and Zero-Flow in Frames

The collision of a vehicle with other targets on road can be in different directions. To the camera mounted under the windshield of vehicle, such a collision has a relative velocity toward the camera as shown in Fig 1. The relative motion vector is aligning with a line of sight of the camera causing a zero-flow in the video. Even if a target moves on one line of sight, it may move away from the camera, stay at the same distance, and approach toward the camera with a potential collision. These actions will show the target size reduced, stays the same, and enlarged, respectively. In the real environments shown in Fig. 2, such collision can happen with front target vehicle on straight road, merging vehicle on highway ramp, upcoming vehicle on curved road, crossing vehicle at intersection, etc.

The Time-to-Collision (TTC) with the target is the distance between two vehicles divided by their relative speed. In the video, the TTC can be computed from the target size in the image divided by the size change during a short period, which will be proved in the following section.

B. Precaution Scenarios and Centered Image Flow

In addition to potential collision, another set of scenarios are required to pay attention. This is the case that the relative motion vector of targets intersects the heading direction of camera, rather than directly towards the camera. Although it does not imply immediate collision, it may switch to a potential collision at next moment. As shown in examples

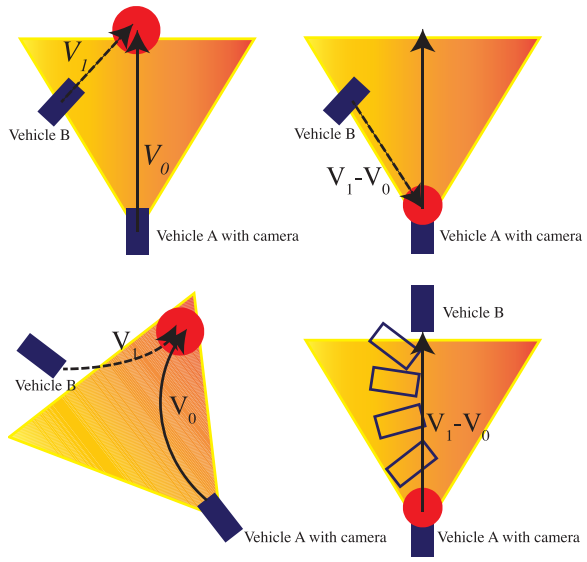

Fig. 1. The relative motion towards collision between camera/vehicle and target vehicles on straight and curved road. The self-vehicle has speed V_0 and target vehicle has speed V_1. Left column is the vehicle and target positions in the world coordinate system and right column is the camera centered coordinate system to see relative motion of targets. Red circles are the potential collision positions.

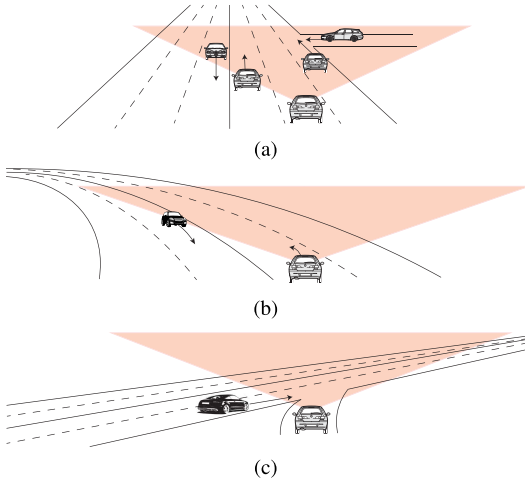

Fig. 2. Possible collision on different types of roads with relative motion between vehicles. Pink regions show the camera field of view. (a) Straight road and crossing road, with side lane vehicle cutting in, or front vehicle slowing down. (b) Curved road with opposite vehicle upcoming. (c) Merging road with collision danger.

in Fig. 3, the cut-in target vehicle from side lane may cause a collision if the target slows down further. On a curved road, the front vehicle slows down. Upcoming vehicle approaches. At an interaction, crossing targets move towards the vehicle heading direction. At a merging road, a vehicle speeds up from side without yield. These actions of target vehicles can be summarized in Fig. 4, where the relative motion vectors of target vehicles are toward the heading line of the camera/vehicle. These velocity vectors in the video frame are accompanied with a horizontal flow towards the Focus of Expansion (FOE), which is the penetrating point of camera translating direction with the image plane. If the relative motion of a target has a vector going behind the camera, the camera/vehicle passes it without danger. In such a case, an outgoing flow to a side appears on the target in the video.

Fig. 3. Different road collision cases cause horizontal motion and vertical flow expansion. The red arrows indicate motion direction of potential collision, green arrows mean safe motion, and orange arrows mean centered motion direction requiring attention. The vehicle heading direction is at the image center.

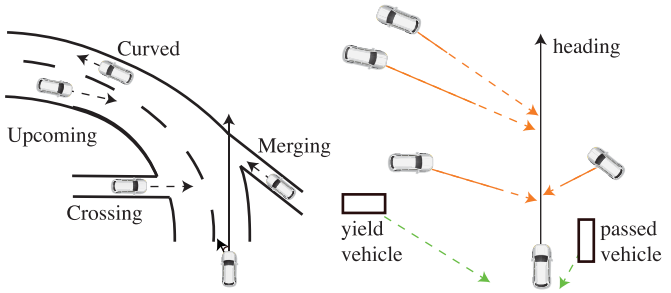

Fig. 4. Some precaution cases where target vehicles generate centered flow in the video. (left) Actions on different roads: crossing, merging, upcoming and slow down with respect to the translation and rotation of camera on a curved road. (right) Relative motion that needs alert. If the extension of motion vector intersects the camera/vehicle heading direction (orange), it is a precaution case. If the extension is toward the back of camera (green), the image flow is outgoing in video and no collision will happen.

C. Vehicle Borne Camera and TTC

After a camera is mounted on vehicle in the forward direction, its forward translation direction is determined by the vehicle and will not change during its driving on a straight path. The FOE is thus fixed at a position in the video. Because our camera is set at a height lower than all the vehicle tops, the horizontal plane through the camera focus passes all the vehicles (lower than all the vehicle tops). If the vehicle is moving on a horizontal plane, the FOE is on the horizon projected in the image. In the video, the projected horizon cuts all the vehicles running on the same plane as the self-vehicle.

We set the camera coordinate aligned with the vehicle, i.e., Z axis on the heading direction and the x axis with the horizon in the image. The image in video is denoted as $I(x, y, t)$, where t is frame number, and the image flow vector or image velocity at point (x, y) is denoted as (u, v). The zero-flow for potential collision is described by $u = 0$ and the centered flow for alert can be described by $u * x < 0$. The outgoing flow in the frame can be described by $u * x > 0$, which is on passing target without danger of collision.

If the vehicle/camera moves along a straight path, the points on background and vehicles moving in parallel toward the camera. A point passes line $Z = 0$ at the Time-to-Pass (TTP). In such cases, TTP can be computed as $TTP = x/u$, where u is the derivative of x, i.e., horizontal image velocity.

For the points moving in a direction different from Z axis, e.g., a vehicle moving in its own direction on a curved road, above formula does not apply. It is not difficult to prove that TTP for an object can be computed by $TTP = D/D'$ for all target moving directions, where D is the object size $D = x_2 - x_1$ in the image, and D' is the size change $D' = u_2 - u_1$ in the video [10]. For a short proof, the perspective projection of camera is

$$x = \frac{Xf}{Z}, \quad x_1 = \frac{X_1 f}{Z}, \quad x_2 = \frac{X_2 f}{Z} \tag{1}$$

Target width at the same depth (e.g., vehicle frontal or back surface) is

$$\Delta X = X_1 - X_2 \neq 0, \tag{2}$$

and it is reflected to the image width Δx according to (1) as

$$\Delta X = \frac{x_1 Z - x_2 Z}{f} = \frac{(x_1 - x_2)Z}{f} = \frac{(\Delta x)Z}{f}, \tag{3}$$

The target width is constant during its approaching. Therefore,

$$\Delta X' = 0, \frac{\Delta x' Z - \Delta x Z'}{f} = 0 \tag{4}$$

from (3). Thus, we have

$$TTC = \frac{Z}{Z'} = \frac{\Delta x}{\Delta x'} = \frac{D}{D'} \tag{5}$$

This means that the TTC computation is not related to camera property like focal length, but a precise time counting of target size in the video. All types of camera can implement this task. On the other hand, at least two lines are necessary to be paired on the same object in order to measure the object

size. Only the motion with zero-flow may cause the collision, which yields real TTC. However, it is not easy to couple two vertical lines on an object without target recognition.

This work does not attempt to perform whole frame vehicle or object recognition such that the proposed method will be more robust on general road environments. We rely on lines appearing horizontally and vertically in the video frame, and response to the potential collision directly. Driving environments are full of lines, which can be categorized mainly in three types in video frames: (1) Horizontal lines on rear side of vehicle such as bumper, window, and top, shadow and road marks on the ground; (2) Vertical lines on vehicles and background such as poles, and side objects; and (3) Lines through depth on vehicle side view and adjacent lane marks. These lines are more continuous and robust to follow in video than points in the moving scenes. The size change of targets and background can be viewed as the convergence and divergence of motion flows of these lines in the video.

III. Motion Profiling to Capture Object Motion

A. Vertical Lines for Understanding Horizontal Movement

To acquire vertical lines in the environment, multiple horizontal belts are placed near the horizon in each frame for vertical color condensing. Pixels in the belt are averaged vertically to produce a pixel line. Lines from consecutive frames are connected along the time axis to form a spatial-temporal image called Motion Profiles $P(x,t)$ as in Fig. 5. Vertical features in video appear as trajectories in $P(x,t)$.

The main advantage of motion profile is to ignore most of the background objects. The vehicles on road are guaranteed to be covered by the sampling belt because the camera positioning is lower than the roof of most vehicles. The belt height can also tolerate small vehicle pitch changes to obtain smooth motion trajectories when the vehicle moves on uneven roads. Motion profile reflects both long and short vertical features, which increases density of motion traces.

The direction of motion trajectory is computed from the gradient orientation that provides the image motion of objects. This motion computation is more stable than optical flow based on two consecutive frames. In addition, the optical flow assumptions on invariant lighting and motion smoothness between frames are frequently violated in driving videos. Even if the trace color changes smoothly in the profile, the trace direction will not change.

We compute the trace orientation based on the first derivative in the motion profile. To avoid the noise from digital sampling of motion profile, we use large filters (9×9 pixels) in 5 degree interval for orientation. Horizontal image velocity u is computed from

$$u = \arctan(\theta) \quad \text{where} \quad \theta = \max_{-85 \le \theta \le 90} G_\theta \tag{6}$$

This will fill the velocity direction of traces almost everywhere in the motion profile. To obtain flow as dense as possible for the motion at all orientation as shown in Fig. 6, we lower down a threshold for picking meaningful gradient values as

$$G(x,t) \quad |G_\theta| > \delta_1 \tag{7}$$

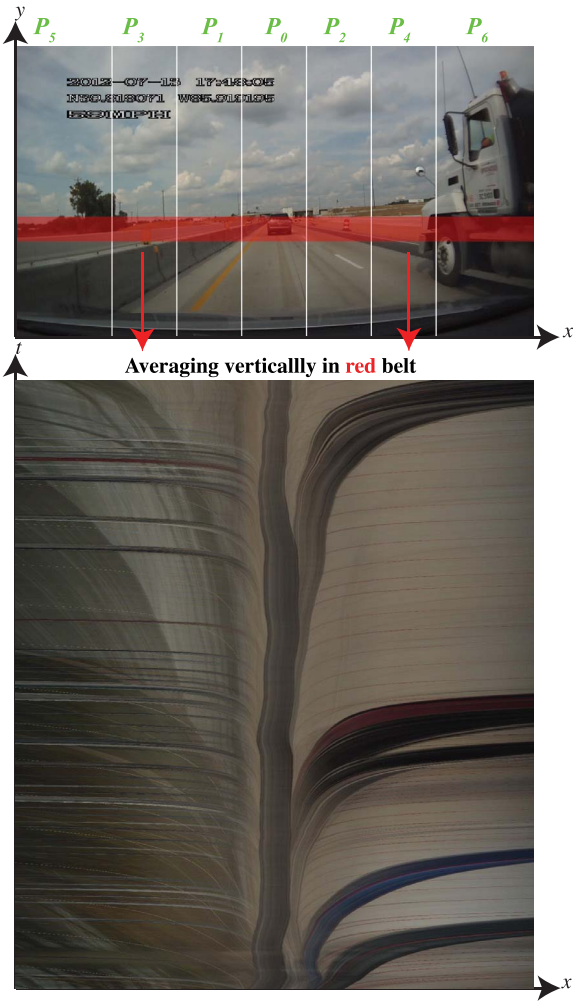

Fig. 5. One example of horizontal motion profile with motion trajectories of front targets. (top) Setting a sampling belt marked in red at the horizon in the frame. (bottom) Motion profile. P_i's are vertical motion profile zones.

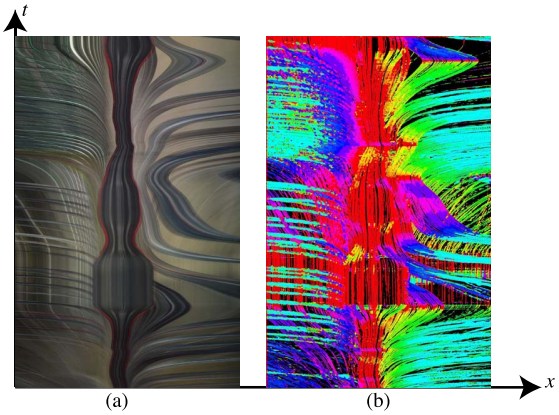

Fig. 6. Computing target flow from the orientation of target trajectories in the motion profile. Color from green, red, to blue indicates positive, zero, and negative flow on the traces towards right, vertical, and left respectively.

For those locations x with $G(x,t) < \delta_1$, u is not reliable as noise. On the other hand, a temporal illumination change can occur when a vehicle goes under a shadow area. A large vehicle pitch may also cause abrupt color changes in the motion profile. These cause contrast edges orthogonal the time axis. Such edges are not real feature traces and are removed according to their close-to-horizontal orientation

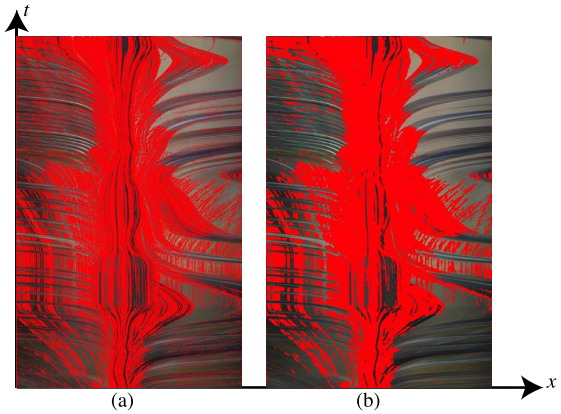

Fig. 7. Zero-flow locations shown in red color over a long period before (a) and after (b) median filtering.

(u close to infinity) in the motion profile. Among all traces, a flow expansion along the time axis means object enlargement as its depth Z decreases.

B. Potential Collision Estimated in Horizontal Motion Profile

A potential collision of target toward the camera has a zero-flow in the video, which is a trace along the time axis. In a potential collision, Considering the physical size of the vehicle wider than the camera spot, the velocity slightly deviated from the line of sight may also cause collision to the body of self-vehicle. Thus, zero-flow region is defined as small flow as

$$|u(x,t)| < \delta_2 \tag{8}$$

which removes safe passing objects including vehicles, and instant changes of profile colors due to vehicle pitch/shaking and illumination changes.

In addition to zero-flow, we pay attention to the flow towards image center (FOE). Thus, a non-zero-flow trace towards the image center (FOE) up to 80 degree in its orientation is included for attention, as long as it is constrained by $u(x,t)x < 0$. Rest of the flow directions indicates passing by objects without danger. This prevents further processing of non-collision objects and background in the video [12].

This processing may still contain digital errors. We further apply median filter in 9×9 regions to motion profile, $u(x,t)$ to obtain reliable clusters of zero-flow regions. In details, in the homogeneous color regions obtained from (8) will produce discretized random noise due to insufficient time sampling of video on fast target motion. After median filter, the noise points are reduced as shown in Fig. 7.

There are three cases in the horizontal zero-flow: target (1) approaching to, (2) leaving, and (3) keeping the same distance from the camera. Only approaching case will cause collision if no breaking or avoidance is taken. This can be confirmed from the flow divergence around the zero-flow spot, where an object is enlarged due to depth reducing. However, it is not reliable to segment the horizontal flow $u(x,t)$ to individual objects from the motion differences, because (a) Multiple vehicles may have the same flow. (b) Complex occlusion between vehicles and background may not reveal entire objects. Flow at occluding point does not reflect true motion. (c) Background space between two target vehicles may expend or shrink in video, which is not the motion of a physical object. The flow divergence or convergence there does not imply a depth change of space. (d) Empty background, e.g., unpainted barrier has less feature on it. Overall, there is no guarantee on finding an object robustly from color, parallelism, and coherence of traces in the horizontal motion profile. Therefore, we will not segment an object for its horizontal size, rather we examine the size changes vertically to identify approaching objects. These circumstances are summarized in Table I and are also illustrated in Figures 2 and 3.

IV. Vertical Flow Divergence Estimation

Since neither target size nor depth are known under the horizon, the video frame is divided into vertical zones for further investigation. For simplicity, these zones are equal in size in order to compensate both straight and curved roads. The size is decided by considering the target scale at close and mid ranges. For example, the center zone that has the far distance is set approximately at the width of front or opposite vehicles 20 m ahead. From these zones, a series of vertical motion profiles are obtained by condensing the color horizontally. In these vertical motion profiles, horizontal features on vehicle, crossing marks on the ground, and a part of road edges stretching in depth are strongly captured. Denote vertical zones as $P_0, P_1, P_2, ..., P_n$ depicted in Fig. 5, with P_0 at center, odd number zones on left and even number zones on right respectively. The scene convergence/divergence is determined in the zones. We compute the distinct flow in each profile where the zero-flow has been detected in order to measure the enlargement of objects in vertical profiles as in Fig. 8.

Because of the scanning effect of side zones on the scenes sideways [13], the profiles may contain shapes of scenes rather than motion traces repeated by the same objects, if the zone does not have a zero-flow in the horizontal motion profile. Such scanned scenes provide no information on the object speed. We thus use the zero-flow weights obtained from the horizontal motion profile to limit the computation only on reliable vertical motion values.

Figure 8 shows the pairs of horizontal and vertical profiles simultaneously obtained from video. Zero-flow regions are marked in horizontal profile $P(x,t)$ and the vertical flow v is marked in the corresponding vertical profiles. The identified traces in the vertical profiles are mainly from horizontal features such as vehicle bumper, shadow, window, top, as well as from crossing road marks and shadows. Very slanted road edges in the image from a curved road or a merging road also respond to the condensing and leave trajectories in the vertical profiles, as summarized in Table I. Fortunately, only those horizontal lines supported by the approaching vertical lines on targets are examined for potential collision. Other horizontal lines are mostly road edges and surface lines that can be ignored here and pursued by other lane tracking modules.

Finding the traces in a vertical profile can provide the speed information of targets relative to the camera in that direction. We also use oriented differential filters with 5 degree interval to pick the highest response as the vertical motion direction. The cost to obtain vertical profiles and computing flow are equivalent to averaging the entire image frame once, plus

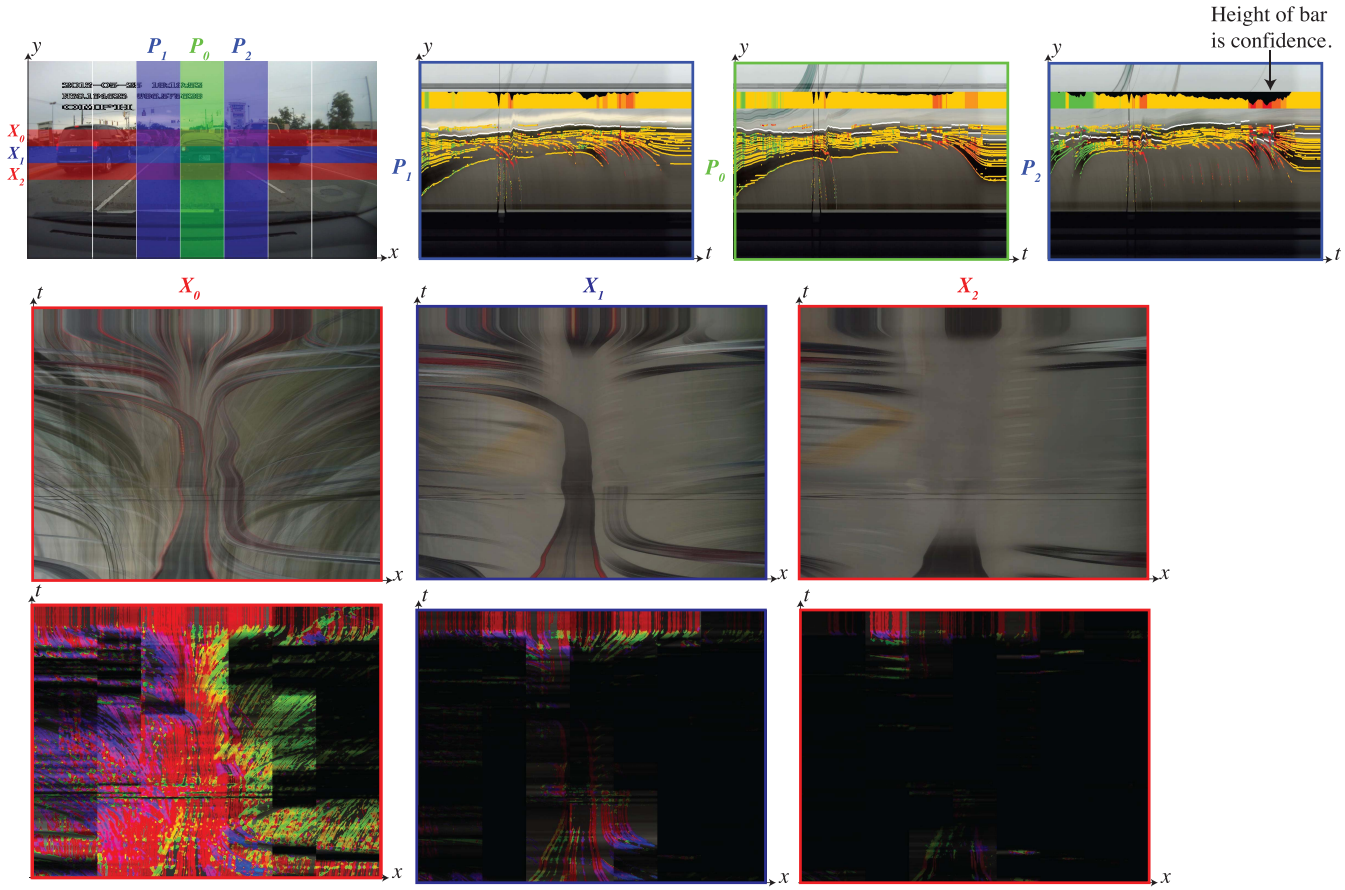

Fig. 8. Example of TTC computation in vertical and horizontal profiles. (top) Frame, horizontal belts, and vertical zones, as well as vertical profiles from three zones around the center. Confidence level is presented as the height of vertical bars at top of the profiles. (middle) Horizontal profiles at different heights starting from horizon to lower positions. The one close to the horizon captures the motion of all objects. The lower ones not only capture the danger at closer depth but also scan road surface. (bottom) Zero-flow in each profiles above. Non-zero-flow regions have lower weights of confidence displayed in dark.

TABLE I

AN OVERVIEW OF HORIZONTAL AND VERTICAL FEATURES IN MOTION PROFILES AND THEIR CLASSIFICATION BY DANGEROUS LEVELS

	Flow	Size change	Straight road	Curved road	Merge road	Crossing road
Horizontal motion profile from vertical features in image	Zero-flow u = 0 (potential collision)	Divergence	Approaching on straight road (collision)	Upcoming vehicle along tangent of curved road (collision)	Merging vehicle causes collision on highway (collision)	Crossing and collision if car is not stopped (collision)
		u = 0	Same distance on straight road (attention)	Same distance on curved road (attention)	Impossible	Impossible
		Convergence	Leaving ahead at front (safe)	Leaving on curved road or turning (safe)	Impossible	Impossible
	Centered flow (attention)	u>0, x<0	Cut in from left (attention)	Approaching on left curved road (attention)	Merge (attention)	Crossing approaching from left (attention)
		u<0, x>0	Cut in from right (attention)	Approaching on right curved road (attention)	Merge (attention)	Crossing approaching from right (attention)
	Outgoing flow (safe)	u>0, x>0	Taking over (safe)	Leaving (safe)	Yield (safe)	Passed (safe)
		u<0, x<0	Taking over (safe)	Leaving (safe)	Yield (safe)	Passed (safe)
Vertical motion profile from horizontal features	Vertical flow downward v>0		Front vehicles, Parked cars (collision) Crossing road marks (attention) and Shadow (safe)	Curved road edge slanted in image when camera/vehicle moving toward road edge before road departure (collision)	Merging vehicle side appearance (collision)	Side road intersecting driving path (safe)

filtering in multiple orientations in y profiles. This is much smaller than the vehicle detection and recognition algorithms with a scalable window shifted in the field of view.

V. TIME-TO-COLLISION COMPUTATION

As in the horizontal direction, if the road is flat locally such that surrounding vehicles are on the same plane, the TTP of POINTS can be calculated from their y coordinates divided by the vertical image velocity v, i.e., $TTP = y/v$. However, if a road has rolling and a camera/vehicle has shaking in pitch all the time, we switch to the vertical motion profiles to observe the motion of horizontal LINES for the TTP.

Given that most non-vertical lines under the camera height are horizontal in the road environment such as road edges, guardrails, crossing marks on the ground, a similar conclusion of TTP calculation as point can be derived. In general, if we condense a horizontal line segment into a vertical motion

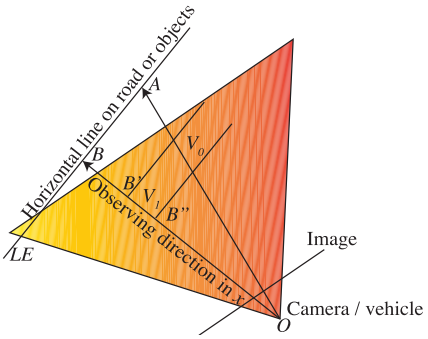

Fig. 9. The approaching of vehicle toward a 3D line in a certain angle. The line is viewed by a vertical sampling zone as a trajectory in its motion profile.

profile, we can prove that the TTP for the camera/vehicle to run over such a line or its extension is also y/v, even if the line is scanned by a vertical zone during the camera motion. This TTP passed under camera is actually the TTC, because the vehicle body runs over the line to cause a collision or road departure, unless the line stops and has a non-zero horizontal flow that leaves away from the vehicle heading.

Theorem: the TTP the camera reaches a line on a horizontal plane is y/v in any direction x.

Proof: Assume a horizontal line, LE, in the 3D space as in Fig. 9, which can be a surface line or road edge. The vehicle moves straight forward in direction OA at speed V_0, while a vertical zone P_i samples LE at the orientation OB. The TTC to arrive LE at A is OA/V_0, where OA is the distance to collision. In the direction of P_i, the observed point B is shifting to $B^{'}$, $B^{''}$..., A gradually on line LE. Because line LE is approaching to camera in parallel, the TTC is equal to OB/V_i, where V_i and OB are the approaching speed and distance of line LE in the orientation of P_i. Therefore,

$$TTC = \frac{OA}{V_0} = \frac{OB}{V_i} \tag{9}$$

In video frame, the depth of a point is projected to the camera at coordinate y as

$$y = \frac{Yf}{Z} \tag{10}$$

where Y is fixed for horizontal lines in the 3D space, and Z is the depth of point. Taking the derivative of (10) with respect to time t, we have vertical image velocity

$$v = -\frac{Yf}{Z^2}\frac{dZ}{dt} = \frac{-YfV_z}{Z^2} \tag{11}$$

where $V_z = dZ/dt$ and $V_y = 0$ due to fixed Y of horizontal line. The TTC thus can be computed from (10) according to (11), which results the same TTC as for points.

$$TTC = \frac{OB}{V_i} = \frac{Z}{V_z} = -\frac{Yf}{vZ} = -\frac{y}{v} \tag{12}$$

This allows us to use the vertical profile in the collision estimation of road edges, guard rails, and stopping lines in the same way as lines on vehicle bumpers and windows in the vertical motion profiles, regardless whether the observed point is sampled by a zone constantly at the same 3D position or is shifted on a line during camera motion. By examining vertical profile $P_i(y,t)$, we found phenomena as:

- Feature traces on a vehicle such as bumper, window, and roof lines scale up and down coherently during depth changes; they have the same TTC.
- Road surface has ground features such as white surface marks, shadows, etc. Their motion is fast approaching in hyperbolic function of vehicle speed. Vision is incapable of sensing feature heights above the ground as LiDAR. However, we can compute the TTC to that surface line using (12). For surface marks along curved road, we can still estimate the time to departure based on piecewise line segments that approximate the curve.
- The trace expansion on a vehicle is mainly observable below the horizon in the frames. However, due to road unevenness and vehicle shaking, the y coordinates of horizontal features are simultaneously waved (Fig. 8).

For single line surface mark, we use skip-one-line policy to ignore it, because a vehicle normally passes a stopping line at street crossings as signal is on green. However, if multiple lines are detected on the ground, they indicate a prohibited region or parked vehicles that must pass with caution or stop. Such a case is treated as collision alarming as well. To implement this, we classify single-line surface marks in bright color in the vertical profile, i.e., a single narrow trace at the lowest position in the vertical profile to ignore. If multiple bright lines are crowed in front of the vehicle, we take them as an area to pay attention and remind driver to slow down. In general, our work to predict collision is not necessary to respond to every ground line, because we assume the surface line marks should be tracked by other modules like road/lane following.

For each time instance t in the zero-flow profile as shown in Fig. 8, TTC is computed from multiple traces at their peaks of gradient starting from the horizon, after ignoring the surface marks as the outlier. The velocities v of traces at y positions are obtained in the vertical profile through filtering. Selecting the highest contrast trace at each moment as a reference with y_0 and v_0, a trace at y_i in the profile has its size $D = y_i - y_0$ and the size change $D^{'} = v_i - v_0$. The TTC of an object is obtained according to (12) as

$$TTC = \sum_{i=1}^{n} \frac{\alpha_i (y_i - y_0)}{v_i - v_0} \tag{13}$$

where coefficient α_i is related to $|y_i|$ and $\sum \alpha_i = 1$. n is the number of traces from horizontal features in the zone. More weights are put on lower features away from the horizon, because a large y_i has larger expansion rate. With the reference trace, we can cancel the TTC shaking and non-horizontal motion of target vehicle in the TTC estimation. If TTC is a negative value, the traces are converging and the target vehicle is leaving away from the camera, which has no danger of collision. The common expansion rate of car shadow, bumper, window, and roof of a vehicle is then obtained for alarming collision.

Instead of using TTC both for real computation and color scale display, we have used $\frac{1}{TTC}$ [14] for result visualization. Figure 10 shows the color scale used in visualization. With the center shifted to the horizon position in the image, we can pre-compute a lookup table to directly obtain the $\frac{1}{TTC}$ in real time estimation. Besides real TTC values, we display four

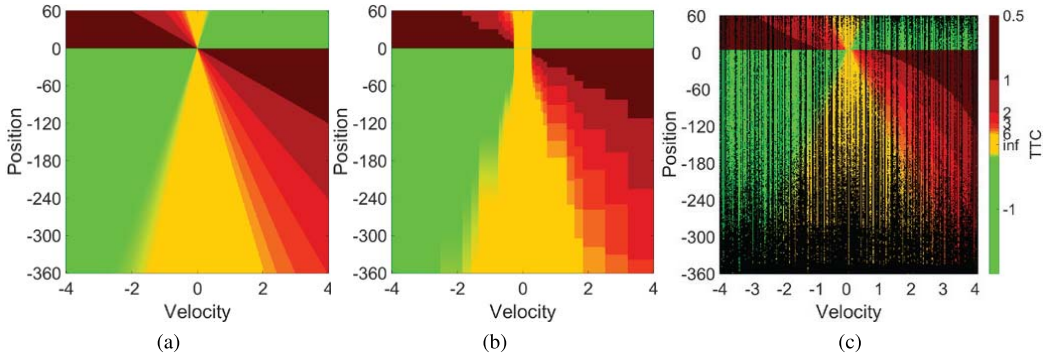

Fig. 10. Colormap visualizes the dangerous degree of potential collision measured in TTC, which is the function of vertical image position y displayed vertically in pixel, and vertical motion v displayed horizontally in pixel/frame. The image position of traces are mostly below the horizon at $y = 0$. The color bar on right gives the TTC ranging from 0.5 second to infinity. (a) TTC is computed with simulated velocity, which forms a look-up table for fast collision prediction. (b) TTC from the approximated velocity using 5-degree rotated 9×9 filters in real motion estimation. (c) TTC data from a large driving video dataset. Black means there is no data resulted in TTC computation due to the discrete output of orientation filtering.

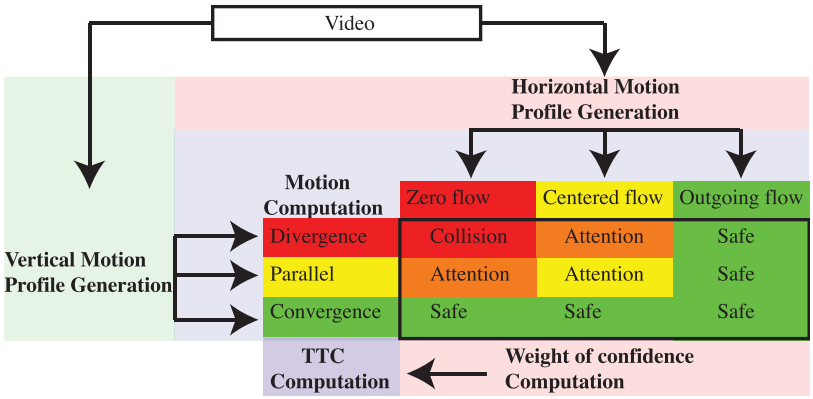

Fig. 11. Processing overview. (1) Simultaneous horizontal and vertical motion profiles; (2) Horizontal and vertical motion detection; (3) Weight computation in Horizontal Motion Profile for confidence; (4) TTC computation in Vertical Motion profiles.

levels of collision status in video. Safe orientations are colored in green. Precaution areas close to zero-flow horizontally are painted in yellow. The approaching objects are marked as orange and then dangerous situation is alarmed in red.

VI. Experiments and Discussion

Figure 11 shows the processing flow from driving video to output of different levels of safety alert. Using neither the distance to the targets nor the vehicle speed itself, we have to obtain image velocity precisely to facilitate the TTC computation. We have applied our algorithm onto naturalistic driving videos without accidents, and the output shows the sensitivity of the algorithm to the moments that need breaking. A large number of video have been examined through visualizing the intermediate results on motion profiles, and total results are superimposed onto the original video for humans to verify the correctness of output. It is not necessary to make a real collision case for verification because the direct computation of TTC in (5) does not include any complex recognition that may bring in high missing and false positive rates.

A. Performance and Effectiveness of Our Method

The experiments are carried out using a large driving video database taken by video cameras facing forward. The videos have the resolution of 1280×720 pixels sampled at 30 frames per second. The computer processor is i7-3770 3.40 GHz with

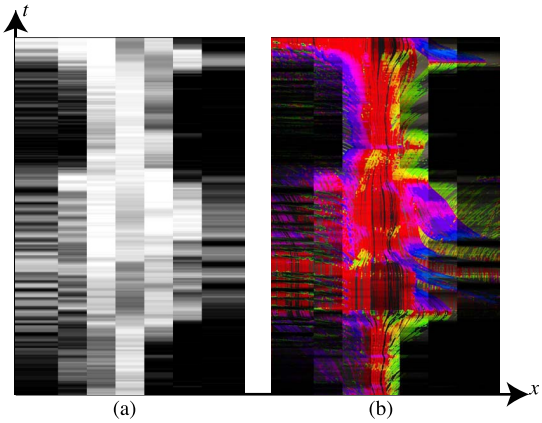

Fig. 12. Zero-flow weights in one zone for confidence level. (a) Zero-flow weight computed from the amount of zero-flow points in the horizontal motion profile. (b) Flow direction in color.

16GB RAM. The implementation has been done using Matlab 2014b on Windows 7. The horizon is provided in advance for pixel condensing to the motion profile.

The efficiency of our method lies in processing several profiles rather than entire video volume, excluding safe directions with zero-flow, filtering of traces without object recognition, computing image motion without iterative procedures like optical flow, and predicting collision independent on depth acquisition as stereo camera. The pixel condensing in selected

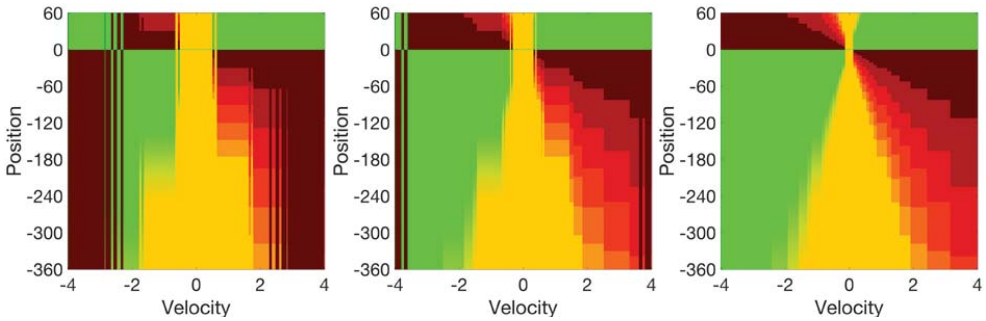

Fig. 13. Error rate of filtering with respect to different filter sizes. (a) 5×5 (b) 7×7 (c) 15×15 pixels.

belt and zones for profiles cost a fixed amount of time, which is 2ms. The delay of the process in alarming is about 4 frames ($<$ 130ms) caused by the filtering with 9-pixel window in the motion profiles. This delay is still tolerable in real time collision prediction. There are average of three Horizontal Motion Profiles per video. The horizontal profile filtering and generating weights calculated at the speed of in 2.8ms. The computation for vertical motion is 2.2ms. Hence, overall computation time is $138ms$ including the filtering delay. Current TTC is the block-wised value over entire field of view. If we want to obtain higher resolution in horizontal orientation for detailed TTC, more overlapped vertical zones can be set to make the dense values in TTC, which may take more time than these values.

B. Precision and Accuracy of Measured TTC

Zero-flow weights are the ratio of zero-flow points in each vertical zone, P_i, as in Fig. 12. These weights are used as confidence level visualized in Fig. 8 in vertical profiles at each moment. Figure 8 shows one example where zero-flow is detected in the horizontal profile, and corresponding vertical profiles are triggered for processing in the zero-flow periods. Because we have displayed the major features of vehicles by their trajectories in the motion profiles, their positions and velocity changes are more visible and countable than verifying bounding boxes in a tracked video.

According to (12), the accuracy of TTC is mainly related to the image position of trace and the image velocity estimation. The position can be localized at the trace peak within $1 \sim 2$ pixels in the motion profiles. The errors in the velocity is yielded from the digital error of 9×9 pixel filters.

It can be easily derived that the TTC error is inversely proportional to Δv^2, i.e., the divergence rate of object traces. This rate is more obvious for close targets than distant ones according to the perspective projection of video. From (12), we can derive

$$\Delta TTC = \frac{1}{v} dy - \frac{y}{v^2} dv \tag{14}$$

where $|dy|$ is the edge location error less than 2 pixels. The error of dv ranges differently according to v. Using a 9×9 filter size, the detectable dv only results in limited levels. These discrete levels cause blockwise output of TTC in Fig. 10b emphasizing small TTC values in critical moment. Figure 13 visualizes TTC error rates of different filter sizes. Size smaller than 9×9 does not have enough resolution to capture the motion. On the other hand, larger ones do not decrease the error significantly. Also, enlarging the filter size is less affective in measuring far objects (small scales) for the low sensitivity. A large filter requires more frames to process, which results in a large delay in the processing as well. In the ideal case, TTC is smooth as in Fig. 10a.

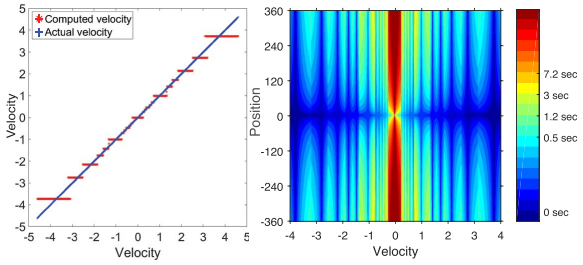

Fig. 14. Precision of TTC: Resolution of TTC with respect to the motion when digitized error from filtering is taken into account. (Left) Discrete velocity output from skewed line orientations using 5-degree rotated filters. The velocity results in 20 distinct levels over the range of $[-5, 5]$ pixel/frame. (Right) Upper bound of TTC error estimated from (14). A parallel moving target with the camera ($v = 0$) has a high uncertainty and should not be used for TTC, until the target shows an obvious motion. Though small, the repetitive patterns appearing horizontally in the distribution are due to the digitized error in v calculation from (left).

We test the accuracy of dv for ideal lines skewed in all directions by using 5-degree spacing filters. The result is insensitive to some range of dv, which causes error distributed in large angle correspond to high image velocity as in Fig. 14. Filtered results of image velocity are compared with the true velocity and the error is displayed. Nevertheless, the real large error of TTC is not at the small values before collision (close to red color in Fig. 14), but in the range when v is close to 0 (yellow color) according to (14). The TTC is as large as infinite momentarily when a target is moving at the same speed as the camera. The upper bound of ΔTTC yields distribution in Fig. 14 from absolute values of two terms in (14). Curved traces with changing velocity within a short period in the motion profile may further randomize the output levels.

We have experimented with 27 videos of different scenarios, which have one hour in total, and have examined the results in the visualization as Fig. 15. An enlarged detail is also displayed in Fig. 16. For all the calculated traces, we plot their distributions of vertical position and velocity that has discrete levels in output. TTC are yielded from the distributions of v and y as shown in Fig. 10c.

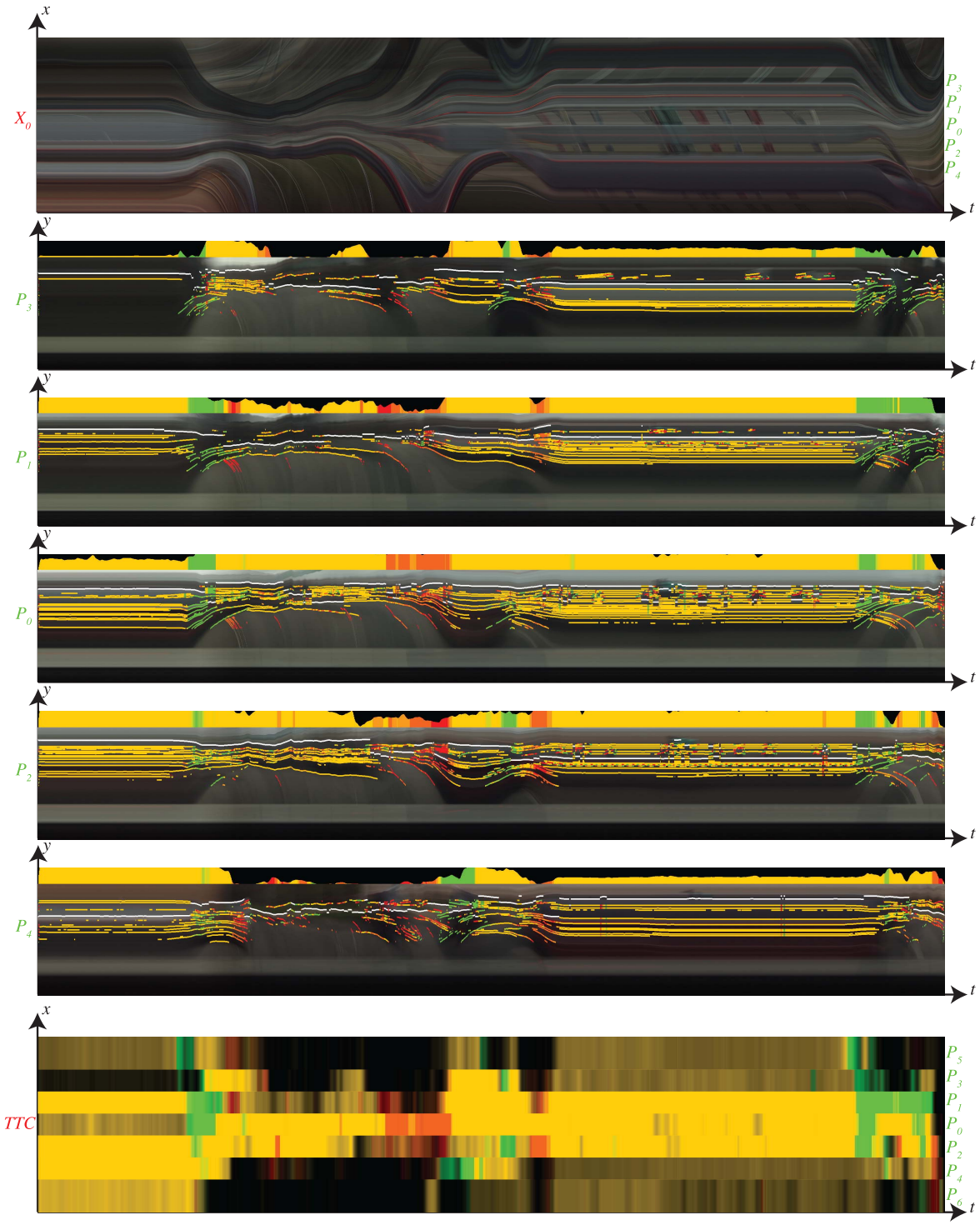

Fig. 15. Visualization of 1min long driving video with TTC and confidence levels. The horizontal motion profile, vertical profiles, and $1/TTC$ values in the field of view are displayed in order from top. The time axes are all horizontal. The top row in each vertical profile shows bars of potential collision in that zone (orientation). The bar color shows the dangerous level $1/TTC$ and the height indicates the confidence level. Below the bar sequence are the traces from horizontal features in that zone colored for TTC according to their convergence/divergence motion. Green is safe, yellow is pay attention, orange and red are dangerous. White traces are the references with highest contrast. The bottom figure colors TTC over time in the field of view showing dangerous level. The intensity indicates the confidence level. Black regions are safe due to the non-zero flow there in the horizontal motion profile.

The visualized color results show 94% of accuracy and 93% percent of precision. That indicates the correctness in the computation fitting with real situation visualized from video. We also generated the video to show the results superimposed on to the input video and confirmed that results visualized in color have no conflict to the relative movement between the camera and the targets. Most of the false positives are from road surface related features in central zones. Main source of error is vehicle shaking due to road unevenness or on vehicle breaking. In these cases, localization of y_0 is not

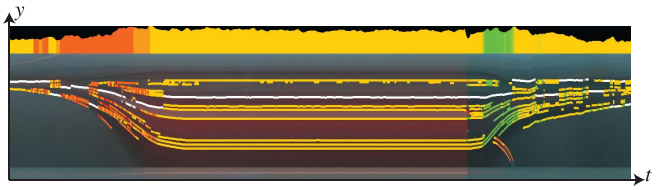

Fig. 16. Enlarged vertical profile with traces colored by their orientation in a stop-and-go driving. The white trace is the most distinct one below the horizon as the reference trace. The TTC is computed from the divergence/convergence of traces with respect to the reference trace. The TTC is displayed also in color bars overtime at the top of the profile. The height of bars are confidence level of the TTC.

precise. This error constitutes approximately 6% in overall data set. Certainly, smoothly paved road or vehicles with good suspension can reduce these type of disturbances.

C. Sensitivity to Various Environments

Different types of roads, as well as a variety of driving actions ranging from sudden breaking behind stopped cars, to cut in from side lanes, from truck merging to curved road vehicle following are selected for testing. The overall evaluation is satisfactory as an alarming of collision dangers.

The belt height and zone widths are set 60 and 160 pixels respectively for horizontal and vertical motion profiles. They are set to cover a vehicle up to $20m$ ahead, and they certainly cover a closer vehicle in obtaining distinct motion. We have observed that scenes as small as $1/4th$ to $1/3th$ of the zone width response to the horizontal pixel condensing and leave trace in the vertical profile. That means our method is sensitive to front vehicles as far as $60m$ if scenes have a good visibility.

The environment changes mainly affect our methods as follows. (1) For the video with poor visibility such as night and heavy raining, the condensing of pixels in the belt and zones will obtain less contrast in the images, and then weak trajectories in the motion profiles. The confidence level in the final TTC computation is thus low because of fewer zero-flow points will be marked in the motion profiles. (2) The sudden illumination changes such as irregular head light of upcoming vehicles at night and specular reflection on vehicle bodies destroy the continuous flow of scenes. This affects our motion based method more or less in estimating TTC, which is more significant than other vision approach based on single frame under insufficient illuminations. (3) The floating edges on objects and road from shadow, highlight reflection, and painted road patterns also produce fake motion different from real vehicle motion, which is a problem in the frame based object recognition as well. In such cases, the motion violates the motion smoothness criterion and can be ignored if a more careful tracking of the motion traces is carried out. On the other hand, our proposed method can be applied to other sensors such as infrared video cameras to overcome the problem, because we are not using object recognition algorithm as a pre-condition, and infrared video satisfies the contrast requirement.

D. Comparison With TTC Measure From Other Methods

Most of the public datasets containing videos with LiDAR data are 10Hz because of their frame based methods in vehicle recognition and tracking. Although a lower temporal resolution

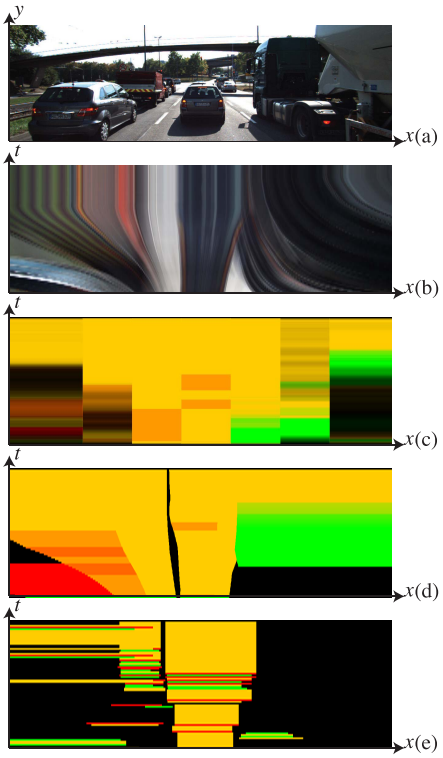

Fig. 17. Comparison of $TTCs$ with different methods using a typical KITTI video. (a) First frame in video. (b) Horizontal motion profile. (c) $1/TTC$ value of our method in color. The intensity shows the confidence level from zero-flow weights in horizontal motion profiles. (d) $1/TTC$ from KITTI 3D LiDAR data. Black part means missing data. In left part, TTP (red) but not TTC is counted. TTP is safe due to its out-going flow. LiDAR from depth can not distinguish TTP. (e) $1/TTC$ from KITTI's vision detected vehicle bounding box, which is not stable due to detection miss (in black) and the miss-alignment of vehicle window in consecutive frames.

is not ideal for our motion based method, we compare our TTC result with depth based TTC from LiDAR, and vision recognition of vehicles for the verification of our method. From one of video databases, KITTI [15] with LiDAR ground truth, we generate a sequence of TTC changes on surrounding vehicles, and display them along with the horizontal motion profile as in Fig. 17. The camera/vehicle is about to stop due to a frontal vehicle waiting for signal. Left lane has slowing down cars and right lane has a speeding up truck.

LiDAR data surrounding the camera is provided. 3D boxes fitted on to front vehicle are used for TTC estimation. Other vehicle's depth change on side cannot be obtained without point-to-point matching and tracking in the dynamic depth map. The point matching and tracking is not feasible for the sparse LiDAR map (32 or 64 lines) from fast transition scenes and traffic in real time driving. Therefore, either an approximation on a large flat surface has to be applied, or a recognition of vehicle in 3D is required. An example of such TTC from LiDAR is given in Fig. 17d. 3D box on the right side truck was not located until enough large parts become visible. Although the truck is partial in the video frame, our method using lines as features is capable of sensing divergent flow (green color) (Fig. 17c). In addition, the depth discontinuity locations measured by LiDAR may frequently yield incorrect TTC values at the boundary of vehicles.

On the other hand, a vehicle recognition module can also provide bounding box for TTC calculation (Fig. 17e), which is a higher level feature component than the lines we are focused on. Vehicle detection in frames has achieved a certain degree of accuracy [16]. Even if we do not count the error in recognition, which is reported to be reduced greatly by the deep learning method [17], the bounding box obtained from a shift window is generally a little larger than the exact vehicle size in order to capture the vehicle outer edges. The size of bounding box jumps randomly from time to time due to the included background scenes during driving, and it is thus hard to grasp the size change accurately. This makes the 2D box inherently inaccurate for TTC estimation. Moreover, the discrete size and shift position of the window further lower the precision of the TTC computation. Particularly, the bounding box becomes error-prone when a vehicle at a close range has only a part visible by the camera, or a truck different from normal cars (not trained in deep learning) enters the field of view partially. In Fig. 17e, side view vehicles are missed in most of the time because the bounding box is not trained well to cover all aspect views and parts in the recognition. Figure 17e gives such an example of frame based recognition in which the discrete error of TTC happens frequently due to the jumping box. Our method depends on lines and responses on partially occluded vehicles and environments. The results obtained in Fig. 17c is more stable than Fig. 17e.

Our TTC from the motion is partially bothered by heavy shadow casted on a front vehicle, which generates stronger traces than vehicle motion traces in the vertical profiles. The vertical profile may also be affected by painted patterns on road that approaches in the inverse speed of vehicle. It is a little difficult to distinguish such false positive lines (safe in driving though) from the vehicle features and road edges that may cause collision. Such false positive lines, if they are long, can be removed in the x-profile as an instantaneous light change.

VII. Conclusion

Inspired by our human driving ability, our method purely uses the motion from a cluster of linear features to compute TTC, which is in principle applicable to all background and vehicles. It avoids complicated vehicle searching and recognition in the video, as well as depth estimation such that it has the computational efficiency for the real time processing. Spatial-temporal profiling and filtering of motion in selected regions have improved the stability of motion estimation with a single video camera. The divergence of motion at zero-flow directions achieves a prompt alarming for potential collision in all directions. The method is an original work using motion only and the test has been carried out on various driving videos and environments.

References

[1] M. Kilicarslan and J. Y. Zheng, "Bridge motion to collision alarming using driving video," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 1870–1875.

[2] M. Kilicarslan and J. Y. Zheng, "Direct vehicle collision detection from motion in driving video," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 1558–1564.

[3] W. C. Chang and C. W. Cho, "Online boosting for vehicle detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 892–902, Jun. 2010.

[4] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," SAE Tech. Paper 2005-01-1470, 2005.

[5] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2318–2338, May 2015.

[6] A. Schaub and D. Burschka, "Spatio-temporal prediction of collision candidates for static and dynamic objects in monocular image sequences," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2013, pp. 1052–1058.

[7] H. T. Niknejad, A. Takeuchi, S. Mita, and D. McAllester, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 748–758, Jun. 2012.

[8] M. Kilicarslan and J. Y. Zheng, "Towards collision alarming based on visual motion," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 654–659.

[9] A. Nègre, C. Braillon, J. L. Crowley, and C. Laugier, "Real-time time-to-collision from variation of intrinsic scale," in *Proc. 10th Int. Symp. Experim. Robot.*, 2008, pp. 75–84.

[10] E. Dagan, O. Mano, G. Stein, and A. Shashua, "Forward collision warning with a single camera," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2004, pp. 37–42.

[11] M. Kilicarslan and J. Y. Zheng, "Visualizing driving video in temporal profile," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 1263–1269.

[12] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 583–595, Jun. 2011.

[13] J. Y. Zheng, "Digital route panoramas," *IEEE Multimedia Mag.*, vol. 10, no. 3, pp. 57–67, Jul. 2003.

[14] V. E. Balas and M. M. Balas, "Driver assisting by inverse time to collision," in *Proc. World Autom. Congr.*, Jul. 2006, pp. 1–6.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[16] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3D estimation of objects and scene layout," in *Proc. Adv. Neural Inf. Processing Syst. (NIPS)*, 2011, pp. 1467–1475.

[17] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.

Mehmet Kilicarslan (S'15) received the B.S. degree from Selcuk University, Konya, Turkey, in 2008, the master's degree from Indiana University, Bloomington, IN, USA, in 2012, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2018. His current research interests include computer vision, image and video processing, and intelligent transportation systems. He has been involved in the motion analysis of driving video at IUPUI.

Jiang Yu Zheng (M'90–SM'05) received the B.S. degree from Fudan University, Shanghai, China, in 1983 and the Ph.D. degree from Osaka University, Osaka, Japan, in 1990. He is currently a Professor with the Department of Computer Science, Indiana University—Purdue University Indianapolis, Indianapolis, IN, USA. His research interests include image and video processing, computer vision, intelligent transportation system, multimedia, virtual reality, robotics, and digital forensics. He has published over 150 research papers as a primary author in journals and refereed international conferences in his research areas. He was a recipient of the best paper awards from the Information Processing Society of Japan in 1991 and the ACM Virtual Reality Software and Technology Award in 2004 for creating the first digital panorama and scene tunnel images, respectively.