Direct Vehicle Collision Detection from Motion in Driving Video

Mehmet Kilicarslan[1], *Student Member, IEEE,* and Jiang Yu Zheng[2], *Senior Member, IEEE*

Abstract— The objective of this work is the instantaneous computation of Time-to-Collision (TTC) for potential collision only from 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, without applying vehicle recognition and depth measuring in prior. Both horizontal and vertical motion divergence are analyzed simultaneously in several collision sensitive zones. 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. The fine velocity computation yields reasonable TTC accuracy so that the video camera can achieve collision avoidance alone from size changes of visual patterns.

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.

Although there have been success 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 rapid 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.

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. 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, depth estimation with stereo cameras, and relative speed measuring. 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, precollision 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 [1] and a bounding box is fitted onto it for tracking [2]. Most of the systems are outlined in survey paper [3] 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 [4] 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 can not be alarmed. A tracking of consecutive frames has to grasp the size and position of bounding box for understanding vehicle depth [5].

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 [4], [6]. 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 [7], [8].

In 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 crossing vertical zones in the video frame to avoid the object recognition and tracking with bounding box. These steps are implemented efficiently in the motion profiles condensed (averaged) from the belt and zones [9]. We compute dense horizontal motion and detect the horizontal zero-flow spots in the motion profile. A long-term motion longer than traditional between-frame optical flow [4] is thus estimated.

The motion profile summarizes objects and, inherently, blurs small details. This generates dense flow as strong evidence of targets, since linear features are stable as compared

[1]Mehmet Kilicarslan is with the Department of Computer Science, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46402, USA phone: 812-391-4904; `mkilicar@iupui.edu`
[2]Jiang Yu Zheng is with the Department of Computer Science, and Transportation Active Safety Institute (TASI), Indiana University-Purdue University Indianapolis, Indianapolis, IN 46402, USA `jzheng@iupui.edu`

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 [10].

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 following sections, we describe our motion data collection in Section II for zero-flow with possible danger. Section III is to confirm flow divergence for alarming. Section IV compute the Time-to-collision supported by Experiment in Section V.

II. MOTION PROFILING TO CAPTURE OBJECT MOTION

Driving video environments have full of lines, which can be categorized mainly in three types in video frames: (1) Horizontal lines on rear side of vehicle, shadow and road marks on the ground; (2) Vertical lines on vehicles, poles, and side objects; and (3) Lines through depth on vehicle side view and adjacent lane marks. These lines are more robust to follow and continues than points in the moving scenes.

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 shown in Fig. 1. Vertical line features appear as trajectories in it.

The main advantages 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 viewing plane through the horizon in the image and camera focus is horizontal in the 3D space, when the vehicle is on a horizontal road. 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
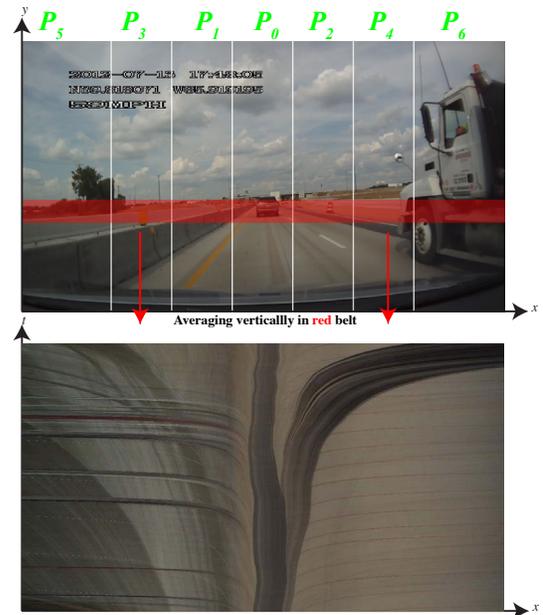
Fig. 1. One example of horizontal motion profile that contains motion trajectories of targets in front. (top) Setting a sampling belt marked in red at the horizon in the frame. (bottom) Motion profile. P_i's are vertical motion profile zones.

sampling of motion profile, we use large filters (9×9 pixels) in 5 degree interval for orientation. Horizontal image velocity u is computed from

$$u = \arctan(\theta) \quad \text{where} \quad \theta = \max_{-85 \leq \theta \leq 90} G_\theta \qquad (1)$$

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. 2, we lower down a threshold for picking meaningful gradient values as

$$G(x,t) \quad |G_\theta| > \delta_1 \qquad (2)$$

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 vehicle pitch may also cause abrupt color changes in the motion profile. These cause contrast edges orthogonal the time axis. Such an edge is not on real feature traces and is removed according to their close to horizontal orientation (u close to infinite) in the motion profile. Among traces, a flow expansion along the time axis means the enlargement of object as its depth Z decreases.

B. Potential Collision in Motion Profile

A collision of target toward the camera has a relative velocity along the line of sight as shown in Fig. 3. In a potential collision, horizontal velocity is towards to the camera, which generates a zero-flow in the video. 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

$$|u(x,t)| < \delta_2 \qquad (3)$$
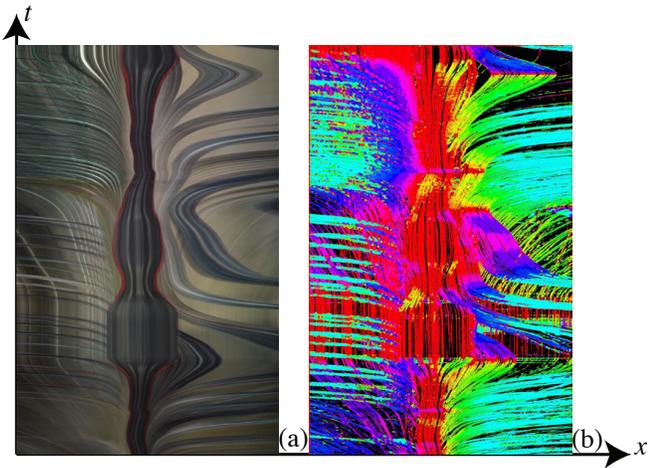
Fig. 2. Computing target flow from the orientation of target trajectories in the motion profile. Color from green, red, to blue indicates positive, zero, and negative flow on the traces towards right, vertical, and left respectively.

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 because such a flow, as listed in Table I, can be (1) a cut-in of target from a side lane, which will cause collision at front sooner after it is in the same lane, (2) a fast vehicle on merging roads, or (3) a slowing vehicle at front on curved roads. 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. Focusing on zero-flow regions prevents further processing of non-collision objects and background in the video.

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 (3) 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. 4.

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
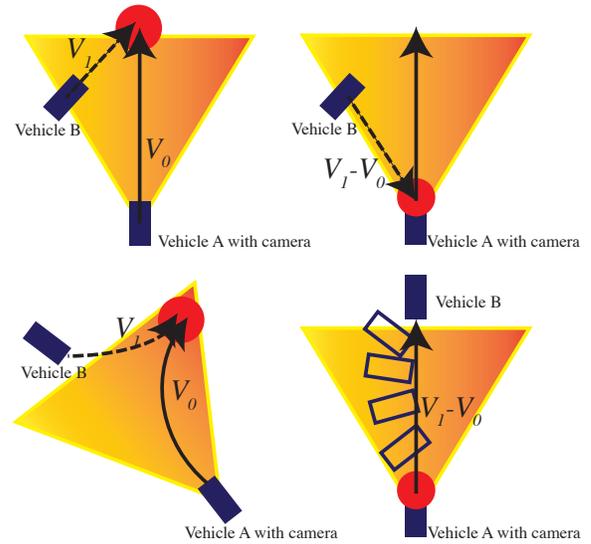
Fig. 3. The relative motion towards collision between camera/vehicle and target vehicle on straight and curved road. Left column is the vehicle and target positions in world coordinate system and right column is the camera centered coordinate system to see relative motion of targets. Red circles are the potential collision positions.
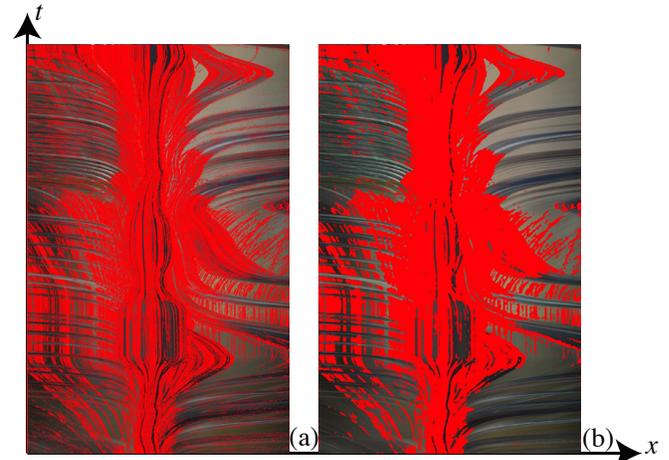
Fig. 4. Zero-flow locations shown in red color over a long period before and after median filtering.

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. Outgoing flow from the image center is ignored for the safe action of corresponding targets. These circumstances are summarized in Table 1 and are also illustrated in Figures 5 and 6.

III. 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. 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
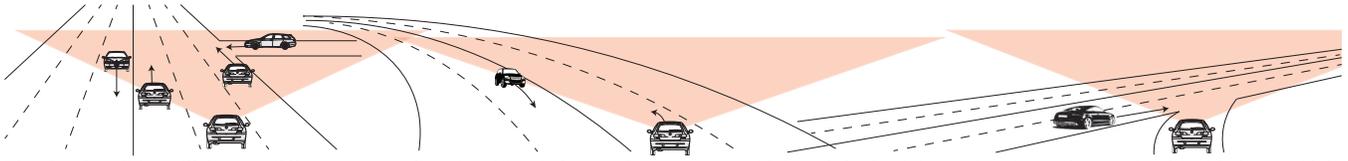
Fig. 5. Possible collision on different types of roads with relative motion between vehicles. (left) Straight road and crossing road, with side lane vehicle cutting in, or front vehicle slowing down. (middle) Curved road with opposite vehicle upcoming. (right) Merging road with collision.

Fig. 6. 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 required attention. The vehicle heading direction is at the image center.

TABLE I

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

	Flow	Size change	Straight road	Curved road	Merge road	Crossing road
Horizontal motion profile from Vertical Lines	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 Lines	Vertical flow $v > 0$		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)

stretching in depth are strongly captured. Denote vertical zones as $P_0, P_1, P_2, ..., P_n$ depicted in Fig. 1, 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. In such vertical zones, horizontal color averaging is carried out to produce a series of vertical motion profiles, $P_0, P_1, P_2, ..., P_n$. 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. 7.

Because of the scanning effect of side zones on the scenes sideways [11], 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 7 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 edges, 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 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.

IV. Time-to-collision Computation

If the vehicle/camera moves along a straight path, the points on background and vehicles moving in parallel toward the camera (i.e., Z axis in the camera coordinate system). A point passes line $Z = 0$ at the Time-to-Pass (TTP). In such cases, TTP can be computed as $TTP = x/u$, where x is the image coordinate of point and u is its derivative, 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 and D' is the size change in the video [8]. This means at least two lines are necessary to be paired on the same object in order to measure size $D = x_2 - x_1$, and size change $D' = u_2 - u_1$. 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.

On the other hand, 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.

For horizontal lines in the static environment such as road edges, guardrails, and crossing marks on the ground, similar

conclusion of TTP calculation can be derived. In general, if we condense a horizontal line segment into a vertical motion 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.

The proof is as follows. Assume a horizontal line, LE, in the 3D space as in Fig. 8, 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,

$$TTC = \frac{OA}{V_0} = \frac{OB}{V_i} \qquad (4)$$

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

$$y = \frac{Yf}{Z} \qquad (5)$$

where Y is fixed for horizontal lines in the 3D space, and Z is the depth of point B when the line is approaching. Taking the derivative of (5) with respect to time t, we have

$$v = -\frac{Yf}{Z^2}\frac{dZ}{dt} = \frac{-YfV_z}{Z^2} \qquad (6)$$

where $V_z = dZ/dt$ and $V_y = 0$ due to fixed Y of horizontal line. The TTC thus can be computed from (5) according to (6), which results the same TTC as for points.

$$TTC = \frac{OB}{V_i} = \frac{Z}{V_z} = -\frac{Yf}{vZ} = -\frac{y}{v} \qquad (7)$$

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 (7). For surface marks along curved road, we can still estimate the collision 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
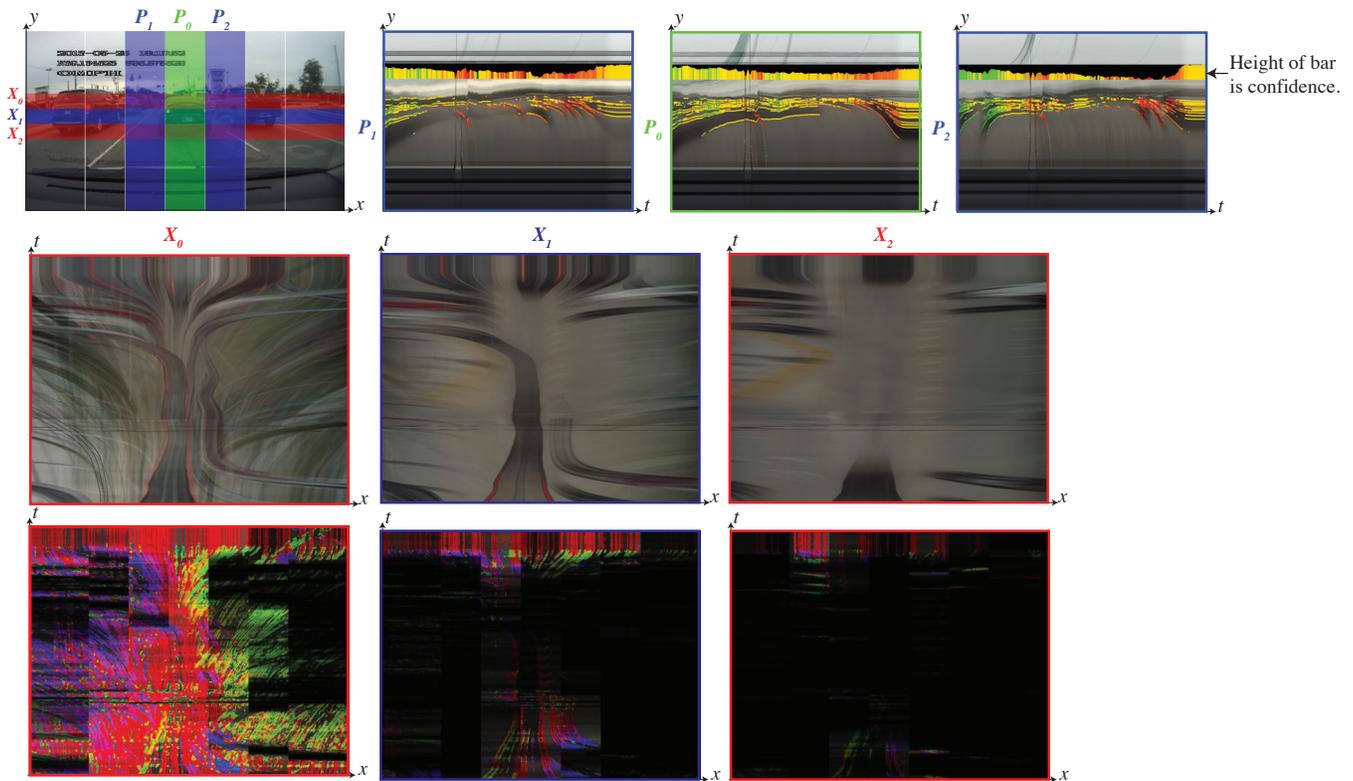
Fig. 7. Example of TTC computation in vertical and horizontal profiles. (top) Frame, belt, and zones, as well as vertical profiles from three zones. Confidence level is presented as the height of vertical bars. (middle) Horizontal profiles at different heights starting from horizon. (bottom) Zero-flow in each profiles above. Non-zero-flow regions have lower weights of confidence displayed in dark.
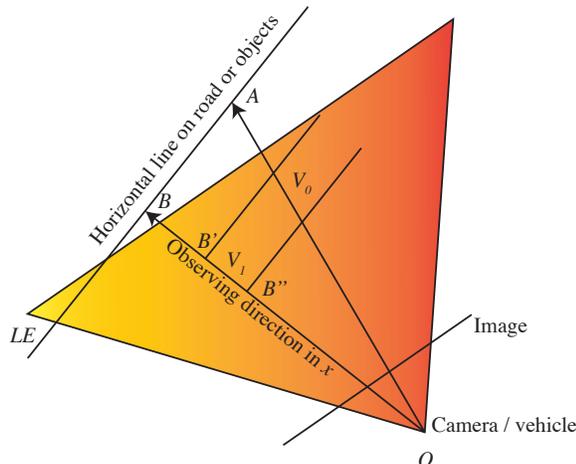
Fig. 8. The approaching of vehicle toward a line in a certain angle. The line is viewed by a vertical sampling zone as a trajectory in the corresponding motion profile.

unevenness and vehicle shaking, the y coordinates of horizontal features are simultaneously waved (Fig. 7).

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 if 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 by alarming them. 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. 7, TTC is computed from multiple traces at their peaks of gradient starting from the horizon, after ignoring the surface marks as the outlier. Selecting the closest trace to the horizon at a position y_0 with velocity v_0, the velocity v of a trace at y position is obtained in the vertical profile through filtering. For all the traces at y_i in the profile, their sizes are $D = y_i - y_0$ and the size changes are $D' = v_i - v_0$. The TTC of an object is obtained according to (7) as

$$TTC = \sum_{i=1}^{n} \frac{\alpha_i(y_i - y_0)}{v_i - v_0} \tag{8}$$

where coefficient α_i is related to $|y_i|$ and $\sum \alpha_i = 1$. n is the number of horizontal features. More weights are put on lower features away from the horizon, because a large y_i has larger expansion rate. 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.

V. EXPERIMENTS

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 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 belt height is to cover a vehicle up to $20m$ ahead, and it certainly covers a closer vehicle in obtaining distinct motion. The belt heights and zone widths are set 60 and 160 pixels for horizontal and vertical motion profiles, respectively. The condensing of selected belt and zones for profiles cost a fixed amount of time, which is 2ms in average. 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 speed is 138ms including the filtering delay.

Zero-flow weights are the ratio of zero flow points in vertical zone, P_i, as in Fig. 7. These weights are used as confidence level visualized in Fig. 7 in vertical profiles at each moment. Figure 7 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 (7), 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 (7), we can derive

$$\Delta TTC = \frac{1}{v}dy - \frac{y}{v^2}dv \qquad (9)$$

where $|dy|$ is the edge location error less than 2 pixels. And dv has an upper bound of 2.5 degrees due to filter spacing of 5 degree. We skew an edge in the spatial domain to simulate traces in various angles.

Main source of error is vehicle shaking due to road unevenness or on vehicle breaking. In these cases, although velocity v is within 2.5 degree error range, localization of y_0 is not precise. This error constitutes approximately %4 in overall data set. Certainly, smoothly paved road or vehicles with good suspension can reduce these type of disturbances.

Instead of using color scale to show TTC values, we have used $\frac{1}{TTC}$ for visualization [12]. 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 levels of collision status in video. Safe orientations are colored in green. Pay-

attention 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.

Using neither the real 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 the naturalistic videos without accidents, and the output shows the sensitivity of the algorithm.

VI. CONCLUSION

Our method purely uses motion from a cluster of linear feature to compute TTC, which is in principle applicable to all background and avoids complicated vehicle searching and recognition in the video. Selective regions for spatial-temporal profiling of motion have achieved the alarming of dangerous collision as well as improved computational efficiency for real time processing. The method is an original work using motion only and the test has been carried out on various videos and environments.

REFERENCES

[1] W. C. Chang and C. W. Cho, "Online boosting for vehicle detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 892–902, jun 2010.

[2] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," *SAE World Congress & Exhibition*, apr 2005.

[3] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle Detection Techniques for Collision Avoidance Systems: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, oct 2015.

[4] A. Schaub and D. Burschka, "Spatio-temporal prediction of collision candidates for static and dynamic objects in monocular image sequences," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2013, pp. 1052–1058.

[5] H. Tehrani Niknejad, A. Takeuchi, S. Mita, and D. McAllester, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 748–758, 2012.

[6] M. Kilicarslan and J. Y. Zheng, "Towards collision alarming based on visual motion," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2012, pp. 654–659.

[7] A. Nègre, C. Braillon, J. L. Crowley, and C. Laugier, *Real-Time Time-to-Collision from Variation of Intrinsic Scale*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 75–84.

[8] E. Dagan, O. Mano, G. Stein, and A. Shashua, "Forward collision warning with a single camera," in *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, pp. 37–42.

[9] M. Kilicarslan and J. Y. Zheng, "Visualizing driving video in temporal profile," in *IEEE Intelligent Vehicles Symposium, Proceedings*. IEEE, jun 2014, pp. 1263–1269.

[10] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 583–595, 2011.

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

[12] V. E. Balas and M. M. Balas, "Driver Assisting by Inverse Time to Collision," in *2006 World Automation Congress*. IEEE, jul 2006, pp. 1–6.