2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Bridge Motion to Collision Alarming Using Driving Video

Mehmet Kilicarslan and Jiang Yu Zheng
Dept. of Computer Science
Indiana University Purdue University Indianapolis
Indianapolis, USA
mkilicar@iupuis.edu, jzheng@iupui.edu

***Abstract*— the objective of this work is to compute the Time-to-Collision (TTC) of surrounding vehicles of a vehicle using motion information in driving video. The key advantage in this work is the extraction of potential danger without vehicle detection and recognition in prior, but directly from the motion divergence in the video. We analyze the trace expansion both horizontally and vertically condensed in the collision sensitive zones in the driving video. Long term motion is stably obtained through filtering in the spatial-temporal video profiles at collision sensitive parts in the video. This overcomes the accuracy problem in object recognition and saved the computation cost tremendously in the real time 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.**

Keywords—image motion; spatio-temporal image; collision alarming; motion profile; time-to-collision

I. INTRODUCTION

A lot of sensors such as LiDAR and Radar have been used for depth acquisition to avoid collision, and have achieved great success in autonomous driving and safety improvement of vehicles. On the other hand, inexpensive cameras have been widely used on vehicles as well. Vehicle recognition and tracking in video have been intensively studied for collision avoidance. However, vision approaches may still have errors in understanding complex road scenes with rapidly changing background [9], even if it has a wider angle for identifying obstacles and road than LiDAR and radar signals.

In various works of potential collision detection, a target vehicle has to be identified first with Haar-type operators via training [1, 2] and a bounding box is fitted onto it for tracking [3]. Through tracking consecutive frames, the size and position of box are updated for understanding vehicle depth [4]. There are still errors in vehicle detection and disturbances in tracking scenes rapidly changing due to the vehicle shaking, scene occlusion, and shape deformation.

This work, however, uses the motion information in a vehicle borne video [5] to find potential Time-to-Collision with the minimum requirement of object recognition. This avoids overwhelmed object search in the field of view with classification windows of different scales, and saves the cost in the real time processing. Our approach is as follow. The dangerous collision from mid-range happens when an object approaches to the camera/vehicle. This approaching generates a zero-flow (optical flow close to zero) on the target [5, 11]. The time-to-collision of a target thus can be obtained instantly from the object size divided by its size change according to the well-known rule [6]. Around the zero-flow spots, we monitor the scene divergence horizontally and vertically in crossing zones in the video frame to avoid the tracking of bounding box. We focused on longer term flow continuously than traditional between-frame optical flow for obtaining stable motion information.

We focus on special image areas in the field of view that a target vehicle cannot be missed [7]. Instead of computing general optical flow on feature points [11], we compute fine horizontal motion flow condensed vertically in the area. This condensing provides more reliable motion evidences from larger and linear features than points. The potential collision vehicles with zero or small flow are extracted in order to ignore most background and non-danger vehicles [8]. At the same time, the horizontal orientation is divided to many zones with varied widths according to the attention angles. In the zones detected with zero-flow, the color is further condensed horizontally for the vertical motion. Moreover, convergence/ divergence factor is computed from the clusters of trajectories to confirm approaching vehicles, exclude leaving vehicles, and follow vehicles moving in parallel. Time-to-collision (TTC) can then obtained for collision alarming.

In the following sections, we describe our motion data collection in Section 2 for zero-flow with possible danger. Section 3 is to confirm flow divergence for alarming. Section 4 compute the Time-to-collision, supported by Experiment in Section 5.

II. MOTION PROFILING AND UNDERSTANDING

There are full of lines visible in the driving video. They include vertical lines on vehicles, poles, and side objects, as well as horizontal lines on vehicles such as bumper, window, roof, shadow, and on the ground such as road edges and surface marks. They are more robustly distinguishable than

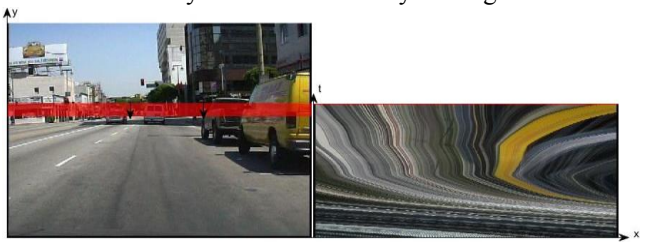

Fig. 1. Profiling motion in a belt around horizon by condensing color vertically to form an array, and then accumulated to the profile $P(x,t)$ in right image.

points in the scenes. To focus on a potential collision, a motion profile $P(x,t)$ is obtained from a horizontal belt at the height of projected horizon, y_0, in video frame $I(x,y,t)$ as shown in Figs. 1 and 2. We continuously project color vertically in the belt to obtain an array in each frame t, and connect temporal arrays to a spatial-temporal image $P(x,t)$ called motion profile [7]. The motion of vertical features appears as trajectories in it.

The motion profile has the advantages to avoid irrelevant background without target vehicles. Because the camera is set at a height lower than the roof of most vehicles, the vehicles on road are guaranteed to be covered by the sampling belt. The belt height also tolerant vehicle pitch changes to some extent in obtaining a smooth motion profile when a vehicle moves on an uneven or waved road. The depth changes of a front vehicle may smoothly change the color of its trajectory in the motion profile. This will not affect the trajectories of vertical features on the vehicle. Because of the vertical color condensing in the belt, the profile collects more vertical features even if they are small, which results in dense motion traces.

To obtain the target motion across multiple frames, we compute the direction of motion trajectories from its gradient orientation in the motion profile. This filtering is more stable than optical flow only from two consecutive frames. Also, the optical flow condition of consistent lighting and motion smoothness are frequently violated in driving video due to shadow and occlusion. Even if the trace color changes smoothly in the profile due to various reasons, the trace direction will not change significantly. We compute the trace orientation based on first derivative in the motion profile. To avoid the noise from digital sampling of motion profile, we use large horizontal and vertical filters (9×9 pixels) G_x and G_t for gradient. This will fill the velocity direction of traces almost everywhere in the motion profile.

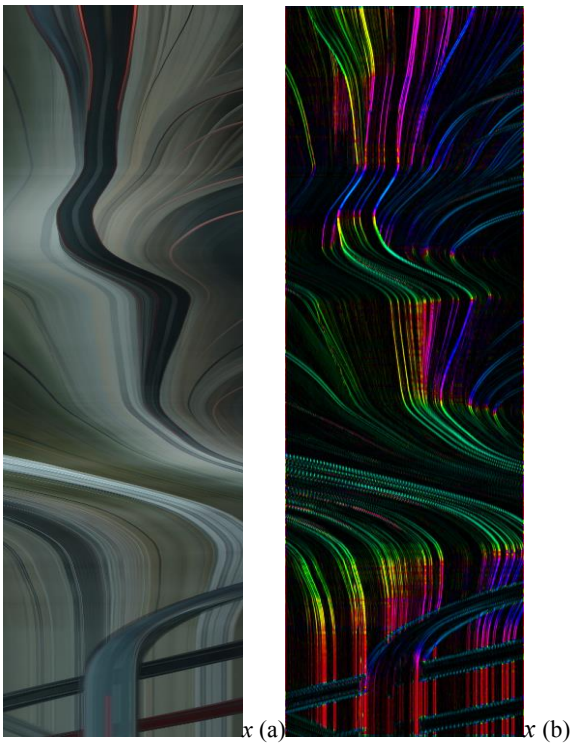

Fig. 2. Stable flow computation from long term traces. (a) Motion profile. (b) Detected motion in color. Red indicates zero flow in continuous $P(x,t)$. Blue and green shows velocities in two different directions.

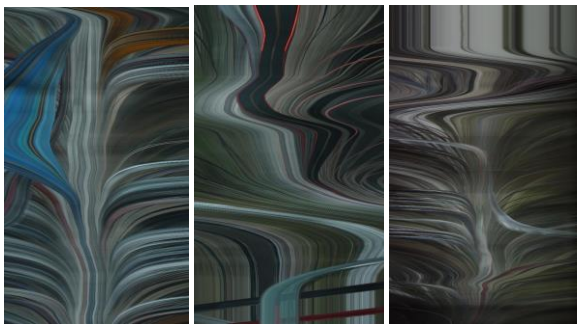

Fig. 3. Profiles containing motion of surrounding vehicles and background. The horizontal axis is image x and vertical axis upward is time t. Background scenes have divergent flow traces to two sides. Blue car trace is visible in left profile. Stopping period yields vertical traces in parallel. Breaking light on of a front car yields red diverged traces

To obtain flow as dense as possible for the motion at all orientation as shown in Fig. 3, we lower down a threshold for picking meaningful vertical gradient values as

$$G(x,t) \quad |G_x| > \delta_1 \tag{1}$$

to compute horizontal image velocity u from

$$u(x,t) = -(G_x/G_t) \tag{2}$$

For those locations x with $G(x,t)<\delta_1$, u is not reliable as noise.

A temporal illumination change when a vehicle goes under a shadow area. A vehicle pitch may also post 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 are removed according to their close to horizontal orientation (u close to infinite) in the motion profile. Among traces, a flow expansion along time means the enlargement of object due to decreasing depth Z (target is approaching).

III. Potential Collision at Zero Flow Position

A collision of target toward the camera has a relative velocity towards the camera as shown in Fig. 4. The zero-flow

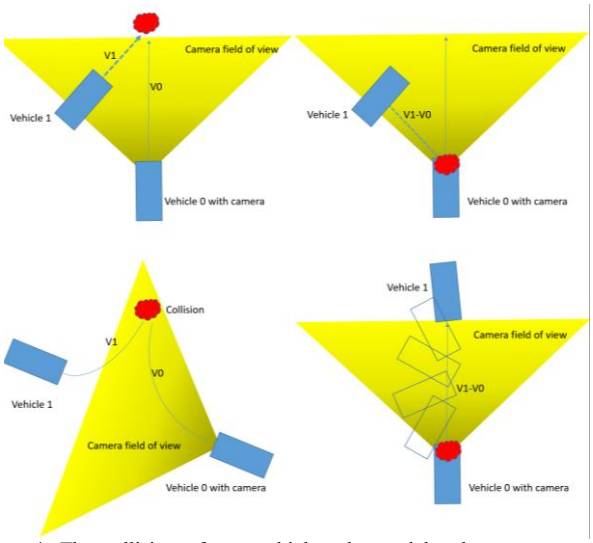

Figure 4. The collision of two vehicles observed by the camera on one vehicle (v_0). Left: absolute speeds of two vehicles. Right: relative speed of the collided vehicle v_1- v_0 towards the camera, which appear as zero flow of vertical features. Even if the two vehicles are moving on curved paths, the same mechanism applies. Top: vehicles move along straight paths. Bottom: vehicles turn on curved paths.

means a motion along the line of sight of the camera, which can be (1) approaching to, (2) leaving, and (3) keeping same distance from the camera. Other flow directions mean scenes passing by. Extracting zero flow can ignore background region to process further. To identify the zero-flow in the profile, the image velocity satisfies

$$|u(x,t)| < \delta_2 \qquad (3)$$

which removes safe passing objects, vehicles, and instant changes of profile colors due to vehicle pitch/shaking and illumination changes. Figure 5c is an example of such result. This operation may still contain digital error and instant illumination changes. We thus apply median filter to $u(x,t)$ to obtain reliable clusters of zero-flow in the motion profile. At the area with homogeneous color, i.e., $G=0$, the random noise of u generates small dots after applying (3). After median filter, the noise points are removed as shown in Fig. 5d.

There are three cases of horizontal zero-flow as mentioned above in (1)-(3). Only approaching case will cause collision if no 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

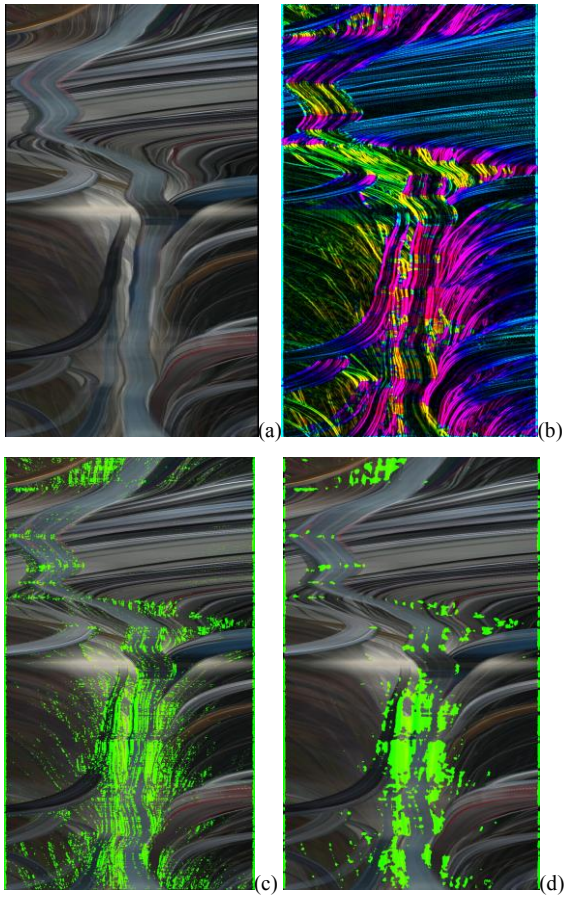

Fig. 5. Computing zero-flow in horizontal profile. (a) Original motion profile (b) Horizontal image velocity (c) Zero flow time in green, (d) Zero flow regions after median filtering.

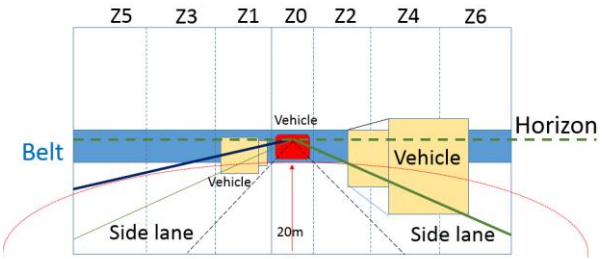

Fig. 6. Defining vertical zones in the video frame for vertical profiles. Red arc is the ultimate distance to sense vehicles.

point does not reflect true motion. (c) Between-vehicle space may expend or shrink in video. The flow divergence or convergence there does not imply a depth change of space. (d) Empty background has less feature. Overall, there is no guarantee to find an object robustly from color, parallel or coherent traces, etc. Therefore, we will not segment an object for its size, rather we examine the size change vertically to identify approaching object.

The horizontal orientation of the view is divided to different vertical zones as $Z_0, Z_1, Z_2, \ldots, Z_n$ depicted in Fig. 6, with Z_0 at center, odd number zones on left and even number zones on right respectively. The scene convergence/divergence is determined in each zone with zero-flow including zone without vertical features (no traces in P). In such vertical zones, horizontal color condensing is carried out to produce a series of vertical motion profiles, $P_0, P_1, P_2, \ldots, P_n$. We find the distinct flow in each zone where zero-flow has been detected and figure out the enlargement of objects in both horizontal and vertical profiles as in Fig. 7 for alarming approaching vehicles.

The width of Z_0 is set such that a front car *20m* ahead can still have a distinct output in the profile P_0, i.e., the width of Z_0 is about the width of front car at *20m* away. Beyond *20m*, a car may not be large enough to appear as trace in the vertical profile. Zone Z_0 certainly is filled up by a car closer than *20m*.

For each zone, color pixels are horizontally averaged for a vertical motion in $P_l(y,t)$. Because of the scanning effect of side zones with respect to the side scenes of the road [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. These scanned scenes provide no information on the object speed. We thus only compute the vertical image velocity in the zones where the horizontal zero-flow points are the majority. 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, etc. Finding the traces in a vertical profile can provide the speed information of target relative to the camera in that direction. The total cost to obtain vertical profiles is equivalent to averaging the entire image frame once. This is much smaller than the vehicle detection with a scalable window shifted in the view.

IV. Time-to-Collision Computation

If the vehicle/camera is moving along a straight path, the points on static background and on the vehicles moving along the same direction (on parallel lanes) pass $Z=0$ in the camera

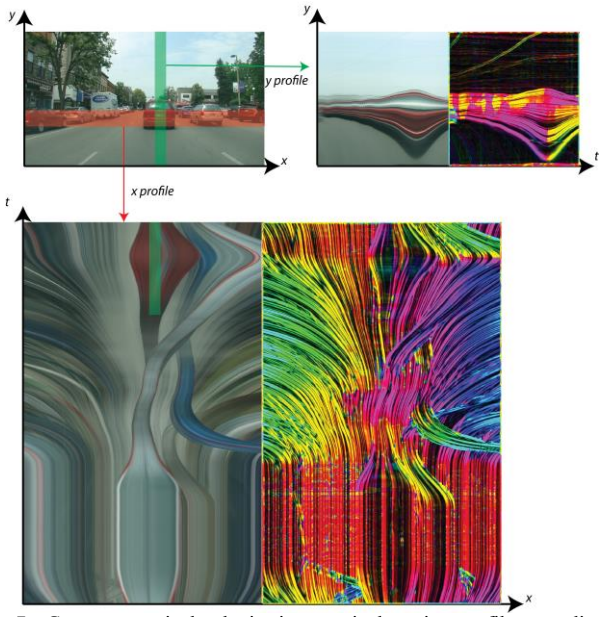

Fig. 7. Compute vertical velocity in a vertical motion profile according to the zero-flow in the horizontal profile. (Top) Vertical profile and computed image velocity in color, (bottom) horizontal profile and dense image velocity.

coordinate system in the Time-to-Pass (TTP, many works call it Time-to-collision traditionally). It can be computed as TTP=x/u, where x is the image coordinate of the point and u is its derivative, i.e., horizontal image velocity. The relative motion of these points is parallel to the Z direction. For the points not moving along the Z axis, e.g., a vehicle moving in its own direction, above formula does not apply. In such a situation, if the road is highly flat so that the surrounding vehicles are on the same plane, the TTP of points on such vehicles or even on background can be calculated from their y coordinates divided by the vertical image velocity v, i.e., TTP=y/v. However, a road may be rolling and a camera/vehicle has shaking in pitch all the time.

We therefore use LINE segments to compute TTC because they are more reliable in tracking as compared to points. It is not difficult to prove that TTP for an object with two identified vertical lines can be computed from TTP=D/D' in general for all target moving directions, where D is the object size and D' is the size change in the video. This means at least two lines are necessary to be marked on the same object in order to measure size $D=x_2-x_1$, and scale change $D'=u_2-u_1$ on two vertical lines, which equals the size divided by size changes. However, it is not easy to couple two vertical lines on the same object without vehicle recognition, i.e., to mark two trajectories on the same object in the horizontal motion profile. We thus switch to the VERTICAL motion profiles at each orientation to observe the motion of horizontal lines for the TTP to those lines.

Two types of outcomes are obtained from our vertical zones in the obtained motion profiles. When an object stays at an orientation that causes zero-flow horizontally, its vertical motion is captured repeatedly in the corresponding vertical profile. On the other hand, if a scene passes that orientation, the zone is scanning different scene. Generally, such transition of different scenes does not provide motion information. Nevertheless, if we use horizontal line segments condensed in the vertical motion profile, we still can obtain the motion

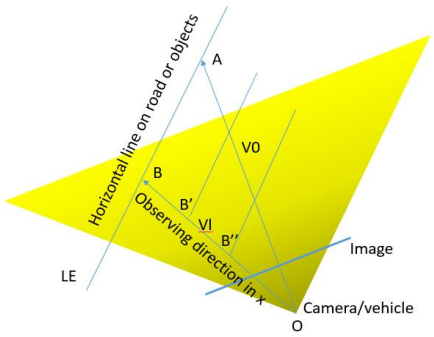

Figure 8. The approaching of vehicle toward a line in a certain angle. The line is viewed by a vertical sampling zone as a trajectory in the corresponding motion profile.

orthogonal to the lines according to the *aperture problem*. In the following, we prove that the trajectory of a line in a motion profile provides its Time-to-Collision (TTC) to the camera.

Assume a 3D line LE is horizontal in the 3D space as in Fig. 8. It can be a surface line or road edge. The vehicle moves straight in forward direction at speed V_0, while a vertical profile samples LE at a fixed orientation Z_i (OB). The TTC to LE at A is the OA/V_0, where OA is the distance to collision. In the direction Z_i, the observed point B is shifting to B', B''…, A on LE and TTC is equal to OB/V_l, where V_l is the approaching speed of line in that orientation. Therefore,

$$TTC = \frac{OA}{V_0} = \frac{OB}{V_1} \tag{4}$$

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

$$y = \frac{Yf}{Z} \tag{5}$$

where Y is fixed for horizontal line 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

$$v = -\frac{Yf}{Z^2}\frac{dZ}{dt} = -\frac{YfV_Z}{Z^2} \tag{6}$$

where $V_z=dZ/dt$ and $V_y=0$ due to fixed Y. The TTC thus can be computed from (4) according to (6), which results the same TTC as for points.

$$TTC = \frac{OB}{V_1} = \frac{Z}{V_Z} = -\frac{Yf}{vZ} = -\frac{y}{v} \tag{7}$$

This allows us to use the vertical profile in the collision estimation of road edges, vehicle bumpers, and a stopped lines regardless if the point viewed by the sampling zone is at the same position or shifted on line. Figure 9 gives an example of viewing horizontal road edge in a vertical profile.

Similarly as in the horizontal motion profile $P(x,t)$, the vertical motion velocity v at the traces with strong contrast is computed from the gradient orientation as shown in Fig. 8. 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, which means that they have the same TTC.

• Road surface has ground features such as white surface marks, shadows, and their motion is fast approaching in hyperbolic function of vehicle speed. Vision is incapable of sensing feature heights above the ground as radar and LiDAR. However, we can compute the TTC to that surface line using (7). For curved surface mark, we still can estimate collision

Figure 9 Example of scanning effect of lane change event in highway. Top images are first and last frames. Divergence of parallel lines to the heading direction is visible in the y profile in the bottom image. One curb line is highlighted in yellow both in frames and motion profile.

based on piecewise line segments approximating 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 from time to time (Fig. 7).

We classify the ground surface marks in white color and exclude them in the alarming decision. Single narrow trace of horizontal mark at the lowest position of video frame is ignored and so as in the vertical profile. 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.

For each time instance t in the zero-flow profile as shown in Fig. 9, 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 and with velocity v_0 computed from trace, the velocity v of a trace at y position is obtained in the vertical profile through filtering. For all the traces 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

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

where coefficients α_i is related to $|y_i|$ and has $\sum \alpha_i = 1$. More weights are put on lower features away from the horizon, because a large y_i has large an 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 the traces from car shadow, bumper, window, and roof of vehicles is then obtained for alarming collision.

V. Experiments

The experiments are carried out using a large driving video database [9] and [10] taken by video cameras facing forward under different weathers and illumination conditions. The horizon is provided in a motion-blurred image by condensing the entire video clip temporally. No other camera calibration is necessary and the horizon is only a rough estimation of the location for pixel condensing to the motion profile. The belt is to cover a vehicle *20m* ahead, and it certainly covers a closer vehicle in obtaining distinct motion. The video has the

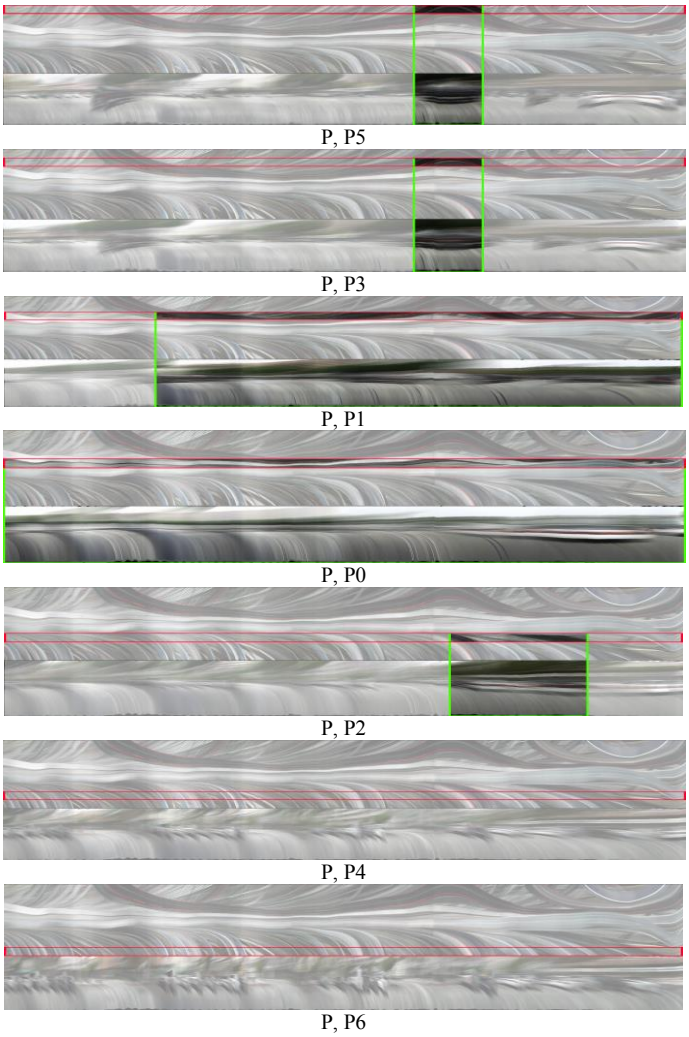

Fig. 10. Horizontal profile displayed vertical profiles (P, P_i). Vertical profiles P_0~P_6 are generated from the red zones in the horizontal profile P stacked at top. Non-zero-flow are shaded and ignored. Only zero-flow zones marked in red in certain periods in P are examined in Green regions of vertical profiles. The horizontal axes are time, and vertical axes are x and y respectively.

resolution of 1280×720 and 640×480 pixels in [9] and [10], respectively, and are sampled 30 frames per second.

Before we condense the color for profiles, we have also tried traditional optical flow computation in our experiments. The optical flow is unable to generate reasonably accurate velocity values for TTC computation; the flow between consecutive frames is noisy. The separated frames also contain large error because of heavy occlusion and fast scene changes in driving video. We thus employ the color condensing method for obtaining motion traces from video. The window height and width are set differently for two data sets. The vertical zones are condensed in parallel with the horizontal profile condensing, although only the vertical profiles with zero-flow in the horizontal profile are used for alarming approaching depth. The condensing of selected belt and zones for profiles cost a fixed amount of time in averaging pixels. The delay of the process in alarming is about 4 frames (<0.2 second) caused from the filtering with 9-pixel window. This delay is still tolerable in real time collision prediction. The main computation is the averaging, filtering, and parameter

calculation with fixed size windows, which greatly reduce the complexity of system and can be reduced further to satisfy the real time video processing requirement during vehicle driving.

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. More examples can be found in supplemental video. Because we have displayed the major features of a vehicle in trajectories, their positions and velocity changes are more visible and countable than verifying a bounding box in 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 located at the trace peak with an accuracy of 1~2 pixels in the profile. The errors in the velocity is yielded from the digital error of 9×9pixel filters. It can be easily derived that the TTC error is inversely proportional to $(\Delta v)^2$, 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 have

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

where $|dy|$ is the edge location error in 2 pixels. However, dv is not fixed. To simulate the error rate of velocity detection, we skew an edge in the spatial domain to simulate traces in various directions in 9×9pixel window. Filtered results of image velocity are compared with the true velocity in Fig. 11. The error is distributed in the scope of large angles correspond to high image velocity. The overall accuracy is also estimated by using Eq. 7, and the distribution is shown in Fig. 12.

Figure 13 shows the estimated TTC in second from the image position and image velocity in the motion profile. The image size is 720 pixels with the center shifted to the horizon position in the image. Red color is short TTC in danger and blue is safe status. We can pre-compute a lookup table and directly obtain the TTC in real time estimation.

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. Besides real TTC values, we display four levels of collision status in video. Safe orientations are not colored. Pay-attention areas close to zero-flow at the horizontal are painted in yellow. The approaching objects are marked as orange and then dangerous red. We have applied our algorithm onto the naturalistic videos without accidents, and the output shows the sensitivity of 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 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 vehicle objects. The avoidance of vehicle recognition makes the method irrelevant to target vehicle shape and types.

References

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

[2] S Sivaraman, MM Trivedi, Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis- IEEE Transactions on Intelligent Transportation Systems, 14 (4), 1173-1195.

[3] I. Gat, M. Benady, and A. Shashua, A Monocular Vision Advance Warning System for the Automotive Aftermarket, SAE Technical Paper 2005-01-1470, 2005.

[4] H. T. Niknejad, A, Takeuchi, S. Mita, D. McAllester, On-road Multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. IEEE Transactions on Intelligent Transportation System, Vol. 13, (2), 748-758, 2012.

[5] M Kilicarslan, JY Zheng, Towards collision alarming based on visual motion, IEEE 15th International Conference on Intelligent Transportation Systems (ITSC), 2012, 654-659.

[6] Nègre, A., Braillon, C., Crowley, J.L., & Laugier, C. (2008). Real-Time Time-to-Collision from Variation of Intrinsic Scale. In O. Khatib, V. Kumar, & D. Rus (Eds). *Experimental Robotics: The 10th International Symposium on Experimental Robotics (pp. 75-84). Springer Berlin Heidelberg*

[7] M Kilicarslan, JY Zheng, Visualizing driving video in temporal profile, IEEE Intelligent Vehicles Symposium 2014, 1263-1269.

[8] A Jazayeri, H Cai, JY Zheng, M Tuceryan, Vehicle detection and tracking in car video based on motion model, Intelligent Transportation Systems, IEEE Transactions on 12 (2), 583-595

[9] R. Tian, L. Li, K. Yang, S. Chien, Y. Chen, R. Sherony, "Estimation of the vehicle-pedestrian encounter/conflict risk on the road based on TASI 110-car naturalistic driving data collection. IEEE Intelligent Vehicles Symposium 2014, 623-629.

[10] P. Dollár, C. Wojek, B. Schiele, and P. Perone, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. PAMI, 34, 743-761, 2012.

[11] A. Schaub, D. Burschkaka, Spatio-Temporal Prediction of Collision Candidates for Static and Dynamic Objects in Monocular Image Sequences" IEEE IV2013, 1052-1058.

[12] J. Y. Zheng, Y. Bhupalam, H. Tanaka, Understanding Vehicle Motion via Spatial Integration of Intensities, ICPR2008,

[13] J. Y. Zheng, Digital route panarama, IEEE Multimedia, 10(3), 57-67, 2003.

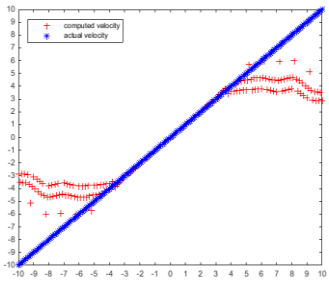

Figure 11. Error rate of computed velocity and actual velocity. The unit is pixel.

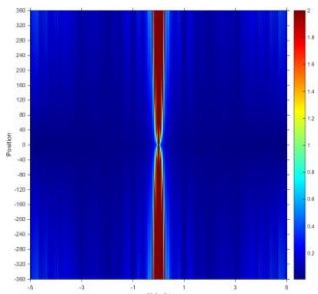

Figure 12. Time-to-collision error in seconds. Red regions are high errors and more uncertainty while blue regions are robust.

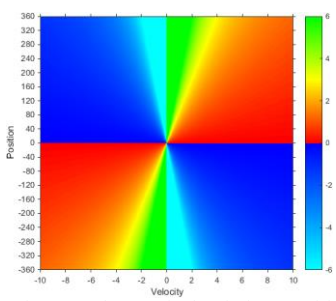

Fig. 13 TTC in seconds. Vertical axis is *y* position centered at horizon, and horizontal axis is the image velocity in pixels. Red is collision regions and blue is safe regions of TTC.