Fish-Eye Image Based Cross Traffic Alert System

Madhurima Bandyopadhyay(✉), Ankit Kumar, and Anoop Pathayapurakkal

Continental Automotive Components (India) Pvt. Ltd., Bengaluru, India
Madhurima.Bandyopadhyay@continental-corporation.com

Abstract. In the automotive industry, the use of advanced driver assisted systems (ADAS) is gaining a lot of traction. ADAS systems are designed to assist the driver by providing information regarding road users, lane information, traffic infrastructures etc., and improve the road safety aspects of an automobile. In this paper, we propose a real-time cross traffic alert (CTA) system based on fisheye camera images having ~180° field of view (FOV). The main purpose of CTA is to avoid unwanted collision with any approaching target by issuing an alert to the driver at a T-junction. The main components of the proposed CTA algorithm are, sparse optical flow vector tracking, object cluster formation and estimation of time to collision (TTC) for each object cluster. The TTC calculation can be performed without explicit depth reconstruction. We compute TTC based on homography estimation between similar features of consecutive image frames. Under good weather condition, the performance of the proposed CTA algorithm in detecting approaching target is reasonably good with a true positive rate of nearly 88%, but with significant false positive rate to the tune of 19%. A major contributing factor to this high false positive is identified to be the inability to distinguish a real crossing object and objects moving parallel to the host vehicle just using TTC estimates. To suppress the unintended false positive cases, we propose a novel solution based on statistics of the flow vector cluster and showcase its efficacy. The classification result for the proposed approach indicates that we have achieved to reduce the false positive rate to nearly 10% while maintaining the true positive rate.

Keywords: Cross traffic alert · Homography · Time to collision

1 Introduction

ADAS has great potential to enhance the driving comfort and more over the road safety aspects. Road accidents are one of the leading causes of death and health hazards in India. According to the Ministry of road transport and highways of India, during 2016, there were cases of 55 road accidents and 17 deaths in every hour. This emphasizes the necessity of ADAS, which can promote a safe driving by monitoring, warning, and reducing the controlling efforts of a driver. The research on ADAS has become the current trend in the automobile industry, which has been fueled by the consumer interest and guidelines of regulatory bodies to strengthen the road safety measures.

R. V. Babu et al. (Eds.): NCVPRIPG 2019, CCIS 1249, pp. 43–52, 2020.
https://doi.org/10.1007/978-981-15-8697-2_4

Now-a-days different ADAS functionalities are available, such as, adaptive cruise control, blind spot detection, collision avoidance system, traffic sign recognition, lane change assistance, pedestrian detection, parking assist among few of them. ADAS uses different environmental sensors like RADAR, LiDAR, ultrasound, visible/infrared imagers to assist the driver in recognizing and reacting to potentially dangerous traffic situations. Each sensor has its own benefits and limitations. Data fusion techniques provide best solution by utilizing the complimentary nature of data [1, 2]. However, such systems are expensive to enter the mass automobile market.

Designing any real-time system requires efficient algorithm with fast processing time. Different techniques have been proposed by various authors. Cui et al. (2010), used Haar and Adaboost classifier to detect moving objects [3]. In other research, Dagan et al. (2004), estimated collision time directly from the size and position of the vehicle in the image without computing a 3D representation of the scene [4]. To detect forward vehicle detection and warning system, Jheng et al. (2015), designed Bayes classifier along with vehicular symmetry detection and shadow detection technique [5]. Yet in another research, Deng et al. (2014), used Haar-like feature and Adaboost classifier together with SVM-based classifier with HOG feature to build forward collision warning system based on monocular vision [6].

In this paper, we are proposing a real-time solution for front CTA system based on fish-eye image for static host vehicle. The main steps in CTA are, feature detection, spare optical flow tracking, object cluster formation and finally estimation of the collision time, i.e., TTC. To calculate TTC, algorithm does not rely on the depth reconstruction between the host and the target object, rather TTC is estimated by determining the homography between cluster features obtained in the consecutive image frames. The proposed CTA module demonstrated good performance statistics, however, it had a serious limitation, where it was failing to differentiate between an object approaching the host and an object moving parallel to the host. This contributed in a high false positive number. To overcome this problem, we have come up with a solution which utilizes the statistical characteristic of the flow vectors of an object cluster. Using this approach, we are able to suppress the false positive cases to a greater extend. The rest of the paper is organized in four major sections. We will start with the method, followed by the results, discussion, and conclusion.

2 Method

The block diagram of the CTA system is shown in Fig. 1. To capture images, a video camera with fish-eye lens is placed at the front bumper of the car. The fish-eye lens has ~180° FOV to capture image of a wide region. The video camera captures 30 image frames per second, with image resolution of 1 megapixel. A pre-processing step is performed, such as noise removal, brightness adjustment etc. before sending the image to the CTA module. Next, CTA module detects any object approaching the host vehicle, it generates alert. The alert can be either in visual or acoustic form. To save further processing time CTA application runs only for two selected portions of the image. These are called region of interest (ROI). There are two ROIs, left and right ROI. The choice of ROI regions is to ensure that the movement of approaching objects (either from left to right for left ROI, or right to left for right ROI) can be captured unhindered.

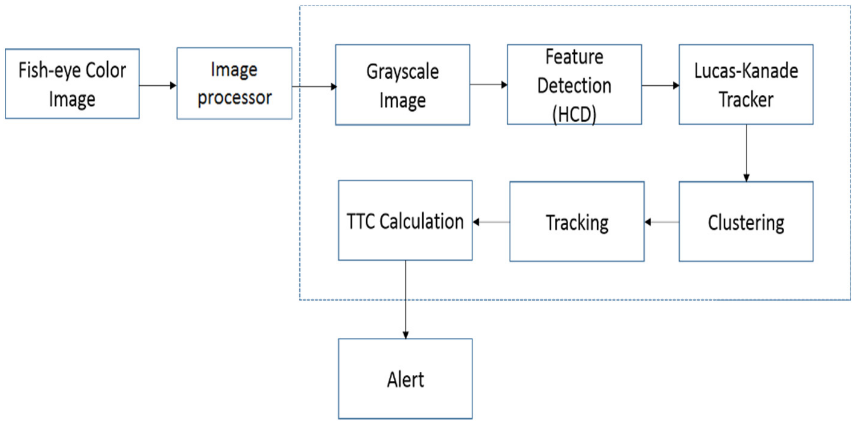

Fig. 1. The block diagram of CTA module, the main components are camera, image processor, CTA module.

Different components of the CTA module are discussed below.

2.1 Feature Detection

Harries corner Detector (HCD): Initial feature points were selected using HCD [7], followed by feature densification method to further enhance the number of feature points.

Feature Densification. Apart from the HCD points for each frame, we try to populate some more feature points, so that we do not miss any features which is not picked up by HCD. Feature densification works like this, it picks up a random point of the image. Over a 3x3 window region around that pixel it computes the luminance difference between the central and surrounding pixels. If majority number of pixels inside that window have luminance difference more than a predefined threshold, then we consider that pixel as a feature point.

Once the features points are identified our next task is feature tracking. We have used Lukas Kanade (LK) optical flow tracker for this purpose.

2.2 Lukas Kanade (LK) Optical Flow

Estimating the location of any image feature point between time frame t and t + 1 is called tracking. There are many tracking techniques available. Lucas kanade (LK) optical flow tracker [8] is most popularly used sparse tracking technique. However, with the conventional LK tracker the problem is to track larger movement or object with high speed of motion. We have implemented four pyramidal layer LK tracker [9] to overcome this issue.

Our next step is to group or cluster the flow vectors, which is performed in the next step.

2.3 Flow Filtering and Clustering

The Lucas-Kanade tracker has generated flow vectors between the subsequent frames. In the clustering stage, we group these flow vectors according to their movement and direction. Since we are concerned with the moving objects which are coming towards the host vehicle, we only consider the flow vectors which are dynamic in nature, as well as whose direction is towards the host vehicle. Clustering of these flow vectors is performed by an algorithm similar to GRIDCLUS algorithm [10].

2.4 Kalman Tracking

Once the clustering is done, we have the location as well as dimensions of the cluster as the output structure. In this step these clusters are tracked based on Kalman tracking [11]. The process of tracking has been described below:

1. For all the available tracks, we set the Kalman state transition matrix and predict the next state of the tracks. The state matrix of the track is given as:

$$X = \begin{bmatrix} xPos & xVel & yPos & yVel & wBox & 0\,hBox & 0 \end{bmatrix}^T \tag{1}$$

   The state transition matrix is based on the basic kinematics formula
We can write the above kinematics equation in the matrix form as

$$X(t) = A * X(t-1) \tag{2}$$

   The Kalman prediction of the next state [12] of the tracks is done using below equations:

- $X(t) = A * X(t-1)$

  Here we are predicting the mean of the next state. We assume that the target vehicles are travelling at a constant speed, so the acceleration of the vehicles is zero. Therefore, in the above equation, we are neglecting the control matrix since there is no known external force.

- $P(t) = A * P(t-1) * A^T + Q$

  Here we are predicting the next state covariance [12]. Q is the Process noise covariance matrix.

2. The track status is updated with respect to the objects in the frame. The types of operation done on the tracks are track association, track deletion, and track merge.

Once object tracking is done next the TTC is computed for the object cluster.

2.5 Time of Collision (TTC) Calculation

In this section we estimate the collision time between the host vehicle and target. TTC is purely computed based on homography estimation using the feature points of a cluster between two consecutive image frames. Hartley and Zisserman, 2000 have shown that the homography induced by a 3D plane between two views is given by,

$$H = K'\left(R - t\tilde{n}^T/D\right)K^{-1} \tag{3}$$

where R and t are the rotation and translation between two camera centres. K and K' are the intrinsic calibration matrix of the two cameras. D is the distance between the first camera centre and the 3D plane, and $\tilde{n}$ is the normal vector of the 3D plane.

Now, according to our setup, the host vehicle and the camera attached to it are static. Images of any 3D object which is approaching towards the host is captured by the camera at different time are being used to calculate homography H. These images are having similar normal vector $\tilde{n}$. Figure 2 depicts the situation.

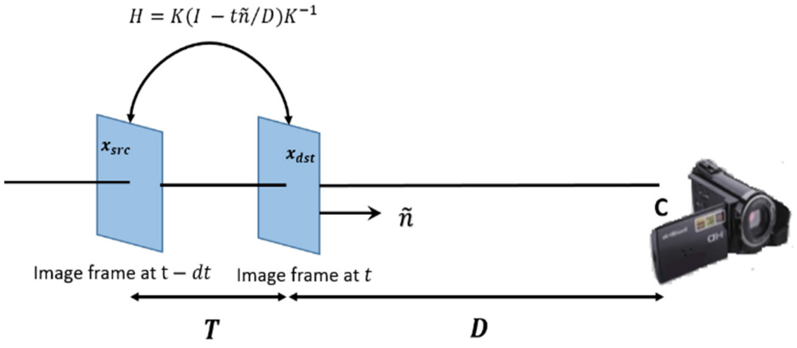

Fig. 2. Homography introduced by 3D plane

Based on the above condition, Eq. (5) becomes,

$$H = K\left(I - T\tilde{n}^T/D\right)K^{-1} \tag{4}$$

where, D is the distance between object and camera at time t. T is the translation/relative displacement between the object at $t - dt$ and t. The intrinsic calibration matrix K is same as only single camera is involded here. Also there is no rotation between two image frames thus rotation matrix R is an identity matix I. According to the above figure, the TTC can be expressed as,

$$\text{Time to collision } (TTC) = D\big/\|T\| * dt = D\big/V \tag{5}$$

where $V = T/dt$ is the relative speed between the object and the camera.

Now, suppose, x_{src}, and x_{dst} are the source and destination point of a flow vector. These two points are connected by the homography matrix H, given by Eq. (6).

$$x_{src} = H * x_{dst}$$

$$x_{src} = K\Big(I - T\tilde{n}^T/D\Big)K^{-1} * x_{dst} \tag{6}$$

In the above equation, the only unknown is T/D, which we have termed as inverse TTC (iTTC). We can approximate the normal vector $\tilde{n}$, without introducing much error in the system, since we are only concern about the moving objects which are crossing the host vehicle. To solve the unknown quantity, we have built a system of linear equation using all the flow vectors inside the cluster obtained from the tracking module. The system of linear equation takes a form given by,

$$A * iTTC = B \tag{7}$$

By solving the above equation, we get iTTC, inverse of this gives us as the collision time TTC. The TTC is computed for each cluster which are having more than three flow vectors.

The algorithm design of the CTA module is discussed above. It can run real time with, with a processing speed of 33 ms and gives a reasonably good detection accuracy. However, we faced a problem in CTA is that, it was unable to differentiate between any incoming objects towards the host and the objects moving parallel to the host.

To overcome this kind of issue, we have applied suppression logic, which we are discussing below.

2.6 Suppressing the Turning Cases

To understand the problem better, when introspected the behavior of the flow vectors, we found out that, for each ROI, the flow vectors associated with an object which may (a) approach the host or (b) move perpendicular to the host, have similar flow direction, but shows different flow characteristics. Based on our observations the behavior of the flow vectors are as follows,

- Approaching object: The object cluster which is approaching the host vehicle gets bigger size due to the fact that flow vectors inside that cluster show divergence.
- Perpendicular moving/turning objects: The object cluster which is moving perpendicular to the host vehicle gets smaller in size due to the fact that flow vectors inside that cluster show convergence.

The above observations are attributed to the fish-eye distortion of the image. Figure 3 illustrates the flow characteristics. To suppress the detection of perpendicular moving vehicle, for each cluster the standard deviation of the source points and destination points was compared. If $\sigma_{src} > \sigma_{dst}$ – cluster is converging, perpendicular movement. If $\sigma_{src} < \sigma_{dst}$ - cluster is diverging, approaching towards host. The suppression logic removes the object cluster which shows $\sigma_{src} > \sigma_{dst}$.

3 Result and Discussion

The major challenge we have faced while designing this driving solutions is that we have to consider various weather conditions, different lighting condition throughout to the day

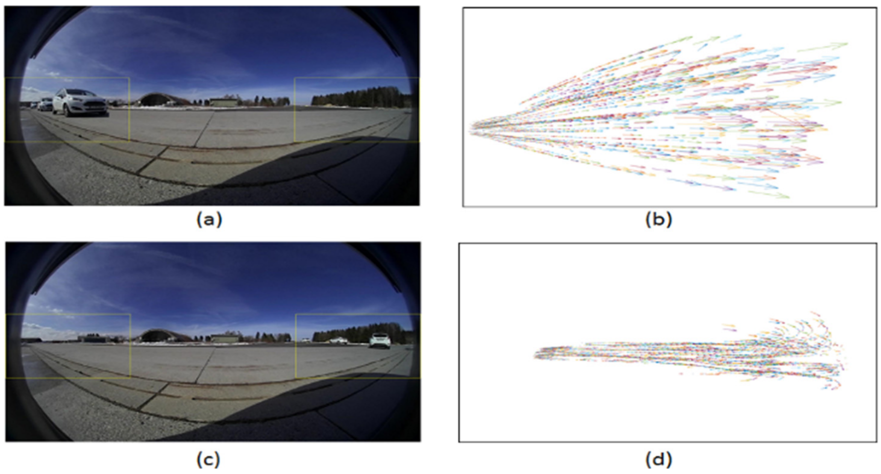

Fig. 3. The flow characteristics is shown in the figure. In figure (a) and (c) the images from three different time frames has been superimposed to show the object trajectory. Figure (b) shows the flow vectors associated with the (a) approaching vehicle (from left to right towards the host). Here the flow vectors are diverging in nature. Figure (d) shows the flow vectors associated with the (c) moving away vehicle. Here the flow vectors are converging in nature.

time, type of target objects and various road conditions (different road junctions, parking bay etc.). Therefore, we have tested our algorithm based on different 1) weather condition (rainy, fog, dry, wet), 2) time of the day (Early morning, Mid-day, Late afternoon), 3) target objects (bus, truck, car, two-wheeler etc.).

To validate the performance of the proposed CTA algorithm, we compared its result with ground truth videos. These videos were captured by placing the differential global positioning systems (DGPS) in the host and the target object. The DGPS is accurate up to 10 cm. Videos of various length were captured ranging from 1 min up to 15 min. Ground truth data were prepared by manually labelling the approaching objects towards the host from either left or right ROI. The DGPS measured distance between the host and target was used to calculated collision time per frame, which was treated as ground truth TTC to compare the accuracy of the TTC alert of CTA.

3.1 Assessment of CTA Measured TTC

As mentioned earlier CTA estimated TTC was compared based on the DGPS derived TTC. Figure 4 shows the plot of CTA TTC vs. DGPS measured TTC, for a target vehicle of speed 35 kph.

As shown in the above graph, the CTA estimated TTC and the DGPS TTC almost follow a 1:1 relation. We can see some deviation, which is more prominent when the target vehicle is far from the host. This was expected, as we are calculating TTC based on homography between two clusters of an image frame. This method of TTC calculation is adopted to reduce the run time. So, it can be said, to improve the run time, we have compromised in the accuracy of collision time. However, as the target comes nearer to

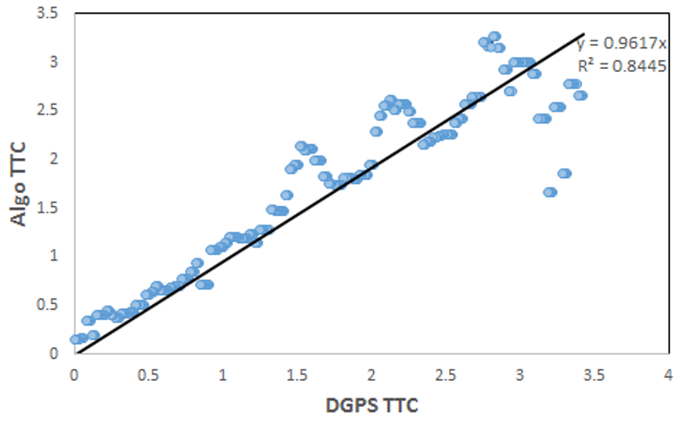

Fig. 4. Comparison of Algo TTC vs. DGPS derived TTC

the host, say between 1.5 s to 0 s, (in this particular case target is ~15 m away from the host) the CTA derived TTC is accurate, which is crucial, as chances of accident is more in this zone.

3.2 Classification Assessment of CTA

We have tested the CTA application under varying environmental conditions. Data catalogue was created to cover all weather conditions, road scenarios, target objects. Table 1 shows the classification accuracy of the proposed CTA module, along with the number of video clips used to test CTA under different weather and lighting conditions. Table 1 shows that except high luminance and foggy condition, the performance of CTA is reasonably good with true positive rate of ~86%. The reason behind the poor performance of CTA for above mentioned cases is that there is a drop in overall scene contrast, which affects the algorithm performance. As mentioned earlier, before the implementation of flow suppression logic, CTA was wrongly detecting parallel (w.r.t host vehicle) moving target objects, which was degrading the false positive rate. For example, the 1st row of Table 1 has total 1049 videos, among them nearly 400 videos have turning case scenario. Due to this reason, the false positive rate was ~19% before the flow vector suppression. Our proposed solution, based on the statistical characteristics of the flow vectors shows a good suppression of the turning cases, and thus the false positive rate has improved to 15%. If we compare the overall performance of the CTA module we can see the overall false positive detection has been improved from 11% to 9% without affecting the true positive detection rate.

Table 1. Classification matrix of CTA module

	Test conditions		Total video clips	Before turning case suppression logic		After turning case suppression logic	
	Weather	Luminance		TP (%)	FP (%)	TP (%)	FP (%)
1	Dry	Low	1059	88.88	19.19	88.07	15.66
2	Dry	Medium	952	94.24	3.35	93.88	2.86
3	Dry	High	813	68.21	10.69	67.39	9.36
4	Wet	Low	909	91.71	7.92	91.25	5.86
5	Wet	Medium	576	90.07	8.06	89.66	6.63
6	Wet	High	365	81.79	18.85	81.07	15.07
7	Snow/Rain	Low	910	86.21	12.36	85.75	11.05
8	Snow/Rain	Medium	339	73.56	18.69	73.32	16.59
9	Fog	Low	328	55.66	9.98	54.65	10.22
10	Fog	Medium	418	54.24	1.2	53.45	1.33
		Average (%)		78.45	11.03	77.84	9.46

4 Conclusion

In this paper, we have presented real-time CTA application with run time of 33 ms. The main functionality of CTA is to alert the driver for approaching target, at T-junction or parking bay. Proposed system purely works based on feature detection and tracking. The TTC calculation is based on homography estimation between two planes. Estimated TTC is accurate when the target vehicle is in the vicinity of the host (Fig. 6). Overall the detection accuracy of the CTA is 80%, with a false positive rate of 9%. Earlier, CTA was suffering from inability of detecting crossing traffic targets with the targets moving parallel to the host. We have addressed this by exploiting the converging/diverging nature of the optical flow to improve the overall performance. This is evident from Table 1, which shows a drop in false positive rate from 11% to 9%, without affecting the true positive detection (~80%). Our proposed CTA module can deal with the cases of static as well as moving host vehicle with small velocity between 0–10 KMPH. In the case of dynamic host, we use ego-motion compensation. The ego motion compensation part is not mentioned in this paper as our main purpose here was to present the main CTA algorithm. Since there is no real-time benchmarking data available, so there is no baselining for CTA. But we have extensively studied the performance of the proposed system with collected ground truth data (Table 1), and able to achieve intended performance. In future, we will try to improve the classification accuracy, without compromising the processing time. Currently, CTA works under day light condition, so in future, we are planning to increase its applicability by running it under dark, where the possibility of accident is more. Also, our plan is to not to constraint the application inside ROI, rather build a semantic segmentation-based detection system.

Acknowledgement. This paper is the result of the collective work of all teams involved in the project. Particularly we want to thank to Boris Lugez (Continental Automotive, France) and Behshad Memari (Continentla Automotive, UK) for their technical support and guidance.

References

1. Tokoro, S., Moriizumi, K., Kawasaki, T., Nagao, T., Abe, K., Fujita, K.: Sensor fusion system for pre-crash safety system. In: 2004 IEEE Intelligent Vehicles Symposium, pp. 945–950. IEEE (2004)
2. Hsieh, Y.-C., Lian, F.-L., Hsu, C.-M.: Optimal multi-sensor selection for driver assistance systems under dynamical driving environment. In: 2007 IEEE Intelligent Transportation Systems Conference, ITSC 2007, pp. 696–701. IEEE (2007)
3. Cui, J., Liu, F., Li, Z., Jia, Z.: Vehicle localisation using a single camera. In: 2010 IEEE Intelligent Vehicles Symposium (IV), pp. 871–876. IEEE (2010)
4. Mano, O.S., Stein, G.P., Dagan, E., Shashua, A.: Forward collision warning with a single camera. In: 2004 IEEE Intelligent Vehicles Symposium, pp. 37–42. IEEE (2004)
5. Jheng, Y.-J., Yen, Y.-H., Sun, T.-Y.: A symmetry-based forward vehicle detection and collision warning system on android smartphone. In: 2015 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 212–213. IEEE (2015)
6. Deng, Y., Liang, H., Wang, Z., Huang, J.: An integrated forward collision warning system based on monocular vision. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1219–1223. IEEE (2014)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, vol. 15. Citeseer (1988). https://doi.org/10.5244/C.2.23
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (1981)
9. Bouguet, J.-Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corp. **5**(1–10), 4 (2001)
10. Schikuta, E.: Grid-clustering: an efficient hierarchical clustering method for very large data sets. In: 1996 Proceedings of the 13th International Conference on Pattern Recognition, vol. 2, pp. 101–105. IEEE (1996)
11. Kalman, Rudolf E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
12. Faragher, Ramsey: Understanding the basis of the kalman filter via a simple and intuitive derivation. IEEE Signal Process. Mag. **29**(5), 128–132 (2012)
13. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)