A Real-Time Collision Warning System for Intersections

Kristen Stubbs, Hemanth Arumugam, Osama Masoud,

Colin McMillen, Harini Veeraraghavan, Ravi Janardan, Nikos Papanikolopoulos[1]

{kstubbs, hemanth, masoud, mcmillen, harini, janardan, npapas}@cs.umn.edu

Department of Computer Science and Engineering

University of Minnesota

4-192 EE/CSci Building

200 Union Street SE

Minneapolis, MN 55455

Phone: (612) 625-4002

Fax: (612) 625-0572

[1] Corresponding Author.

Abstract

Collisions between vehicles at urban and rural intersections account for nearly a third of all reported crashes in the United States. This has led to considerable interest at the federal level in developing an intelligent, low-cost system that can detect and prevent potential collisions in real- time. We propose the development of a system that uses video cameras to continuously gather traffic data at intersections (e.g., vehicle speeds, positions, trajectories, accelerations/decelerations, vehicle sizes, signal status etc.) which might eventually be used for collision prediction. This paper describes some of the challenges that face such a system as well as some of the possible solutions that are currently under investigation.

Introduction

Statistics from the crash database of the National Highway Traffic Safety Administration (NHTSA) show that in 1998 there were about 1.7 million vehicle crashes at intersections that accounted for as much as 27% of all reported crashes for the year and resulted in about 6,700 fatalities [1], [2]. The problem is expected to get worse with the continued proliferation of urban sprawl and the corresponding increases in traffic volumes and travel distances. Hence, there is considerable interest at the federal level [3] in the design and implementation of intelligent, real-time systems that can use knowledge of current traffic conditions at an intersection and its vicinity to predict potential collisions or near-misses and issue suitable countermeasures. We call this the *Intersection Collision Warning and Avoidanc*e (ICWA) problem.

Effective solutions to the ICWA problem must deal with a number of complex issues:

1. They must be able to integrate and synchronize temporal traffic information from a variety of sensors (e.g., multiple cameras from a computer vision-based system, radar, and GPS).
2. They must process this information, detect collisions or near-misses, and issue countermeasures in real-time (e.g., at 10 15 Hz.)
3. They must account for various trajectories of the vehicles. For instance, at the intersection, the vehicles in question may be moving at right angles to each other or they may be moving in opposite directions when one of them suddenly attempts a turn at the intersection.
4. They must account for different vehicle speeds and acceleration/deceleration in the vicinity of the intersection.
5. They must be able to process large numbers of vehicles moving relatively slowly (e.g., a suburban intersection) as well as few vehicles moving at high speed (e.g., a rural intersection).
6. They must be able to distinguish between different types of vehicles (e.g., buses are longer than cars, so they have a larger collision profile and also make wider turns).
7. They must account for pedestrians and cyclists crossing at the intersection (e.g., could these be treated as vehicles in their own right?).
8. They must have effective means for communicating countermeasures.
9. They must take into account other factors, such as the status of signals (if any) at the intersection and its vicinity, any signals issued by vehicles (e.g., flashing turn signals), the geometry of the intersection, current weather conditions (e.g., stopping distances in the winter are longer than in the summer), and the effect of countermeasures issued (e.g.,

does a suggested countermeasure such as a flashing warning light cause a vehicle to brake suddenly and create the potential for additional collisions?).

Developing a full-fledged system as discussed above is our long-term goal. We envision that such a system would consist of three interacting modules, as shown in Figure 1. At present, we have initiated this process with a technology feasibility study for a system that includes some of the above features. The system we are currently developing will incorporate computer vision techniques to gather traffic and other data at intersections. We plan to test our solution both via computer simulations and via field tests at actual intersections, such as an intersection between a highway and a major county road in a suburb of Minneapolis, MN, and a suburban intersection in St. Paul, MN.

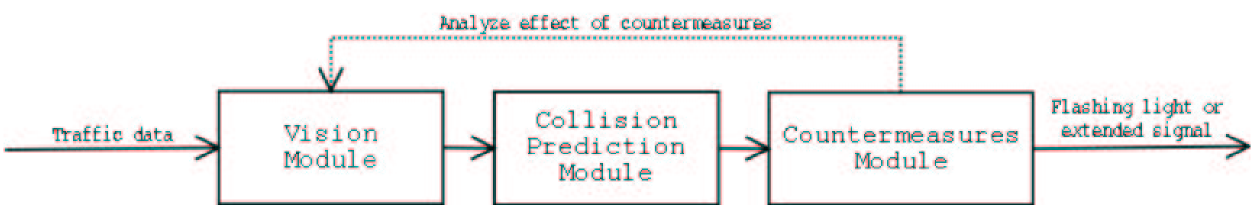

Figure 1. The complete ICWA system. This focus of this paper is on the Vision Module and the Collision Prediction Module.

Previous Work

Our research group's earlier projects on pedestrian tracking, vehicle detection, tracking and classification, and intersection control are the basis of the new system. In particular, O. Masoud, S. Gupte, and N. Papanikolopoulos have developed a method that can detect, track, and classify vehicles by establishing correspondences among vehicle entities and blobs (regions of motion) in the image. This technique has been used for the collection of data at weaving sections where vehicles need to be tracked as they change lanes. Unlike commercially available systems (Nestor's CrossingGuard and TrafficVision, Trafficon, PEEK, Solo, Odetics' Vantage), our approach treats the vehicles as separate entities with specific geometric and kinetic properties and constraints. This allows us to follow them as they move from lane to lane. The weaving system has already been used by Dr. Eil Kwon (Minnesota Department of Transportation) to collect data from weaving sections. Understanding characteristics of accident-prone intersections along with the design of new operational guidelines will help us to better monitor those intersections so that we can help to prevent accidents.

Some work done that is close to our approach is work in the area of three-dimensional vision-based tracking. Three-dimensional tracking uses models for vehicles and aims to handle complex traffic situations and arbitrary configurations. A suitable application would be conceptual descriptions of traffic situations [5]. Robustness is more important than computational efficiency in such applications. Kollnig and Nagel [6] developed a model-based system in which they proposed to use image gradients instead of edges for pose estimation. In another relevant piece of work, the same authors increased robustness by utilizing optical flow during the tracking process as well. Nagel et al. [7] and Leuck and Nagel [8] extended the previous approach to estimate the steering angle of vehicles. This was a necessary extension to handle trucks with trailers, which were represented as multiple linked rigid polyhedra. Experimental results in Leuck and Nagel [8] compared the steering angle and velocity of a vehicle to ground truth showing good performance. They also provided qualitative results for other vehicles showing an average success rate of 77%. Tracking a single vehicle took 2-3 seconds per frame. However, our proposed approach can handle a large number of vehicles in real-time. Finally, ours is the only effort of which we are aware that does vehicle tracking using a set of cameras (and can track a vehicle as it moves from one camera's field of view to another's).

Our group has also done work on collision prediction, motivated by applications in air traffic control and robotics, albeit under rather simple assumptions [4]. For instance, given a collection of point-objects, moving with different speeds and along given trajectories, the algorithms in [4] can compute very rapidly the potential collisions and near-misses in the system (near-misses are based on a user-specified threshold distance up to which the moving points can approach each other safely). These methods have been extended to deal with entities modeled by rectangular bounding boxes and moving along orthogonal trajectories. The solutions are based on advanced algorithmic and data representation techniques from the field of computational geometry.

Issues

While the technology for monitoring vehicles has been improving over the years, monitoring intersections and areas of congested traffic remains a very challenging problem. Several issues must be addressed in order to effectively monitor intersections and prevent accidents:

10. *Shadows*. Vehicles cast shadows as they travel, sometimes on top of other vehicles. Separating a vehicle from a shadow is a challenging problem, as shadows are not necessarily uniformly dark and may be the same color as the vehicle being tracked. Most methods that have been used to date lack an effective means to model shadows.
11. *Occlusions*. Occlusions occur when something obscures a vehicle on the road. This may be a stationary object such as a tree or another vehicle. Using an elevated camera minimizes occlusions, but in the case of an intersection this is not desirable as the goal is to monitor traffic in all directions.

12. *Stop-and-Go*. In congested traffic and at intersections, vehicles must slow down to a stop. Our system must keep track of vehicles even when they are not moving, meaning that usual tracking methods that separate moving objects from the background will fail.

Possible Solutions

The goal of our current work is to overcome these issues and create a system that can effectively predict vehicle collisions. At present, this system consists of a Vision Module for monitoring the intersection and a Collision Prediction Module to predict potentially dangerous situations.

Vision Module

The images captured by the camera(s) are analyzed in the Vision Module. The input to the Vision Module consists of a sequence of images; the outputs from the module are the positions and trajectories of the tracked entities. An adaptive background segmentation scheme (like that used by Stauffer et al. [11]) is used for learning the model of the background during the course of tracking. This model is used to make the background subtraction robust to changes in lighting conditions in the scene. In the next step, the individual foreground regions are extracted using a connected components method. A region tracking method is used for tracking the moving vehicles and pedestrians in the image. For tracking, two levels of abstraction are used: the blob level and the object level. In the blob level, the individual regions extracted in the current frame are compared with those in the previous frame to establish correspondences. The relations are represented in the form of a bipartite graph. The tracking method used is similar to Masoud's pedestrian tracker [12]. In the next step, the information from the blob level is passed on to the object level.

Some of the issues faced in tracking are the following:

13. Splitting and merging of blobs due to the background estimation method
14. Occlusions
15. Cast shadows that move
16. Stopping of vehicles for long periods.

An inherent weakness of the background estimation method is that the individual vehicles are not always segmented into one whole region; instead, they are segmented into groups of small regions (Figure 2). These regions merge eventually, but might split again later. This leads to errors in the tracking process. This is an issue in the Collision Prediction Module as well. If a single vehicle is segmented into two regions (which will seem to be near one another), the system might raise a false collision alarm.

As discussed above, occlusions can be a problem, especially in typical outdoor scenes with a large number of vehicles and pedestrians. These occlusions are another source of errors in the Vision Module.

Moving shadows in the images cannot be segmented out by the background segmentation method. Shadows distort the shape of the moving objects and affect the segmentation of the individual objects. Accurate object shape estimation is very critical for the Collision Prediction Module. Hence, shadows cause very serious problems not only with tracking but also in the Collision Prediction Module.

In order to address the above issues, we are investigating the following:

17. Kalman filtering and

18. Shadow detection.

Kalman filtering:

We are developing a Kalman filter [13] for improving tracking accuracy. The Kalman filter can improve the estimates of the tracker by providing predictions based on past position measurements. A Kalman filter operates by using a set of predictor and corrector equations. The input to the filter consists of measurements in the form of the center position of the bounding box of the object in image coordinates. These are converted to scene coordinates and represented inside the filter. The filter estimates the object's position in scene coordinates. An extended Kalman filter is used due to the non-linearity in the transformation from the image to scene coordinates. The use of center coordinates to represent the position of the objects holds only for objects that have low height to width ratio. For objects like a bus this assumption no longer holds and estimation accuracy might be negatively affected.

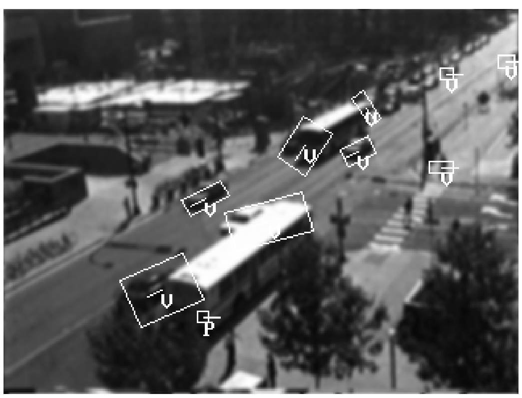

Figure 2: Splitting of individual vehicles into multiple vehicles. The figure also illustrates the merging of the height and length information into one.

Detecting shadows:

The Collision Prediction Module (CPM) relies on the Vision Module to furnish correct estimates of vehicle size and shape so that it can predict collisions with a high degree of confidence. The accuracy of the vehicle's size estimate is often limited by the presence of cast shadows. Our system uses a shadow detection and elimination scheme based on the fast marching method. This method is used to correct the vehicles' size and shape estimates by eliminating the shadows before these estimates are sent to the CPM.

The main idea behind the fast marching method is that - given the existence of two regions that are separated by a coarsely defined boundary, a moving front can be made to align itself with this boundary. The speed of this front is given by a speed function, F, which is dependent on the intensity of the image pixels on which the front moves. Hence, if the vehicle and its shadow are considered as two regions that are separated by a vehicle-shadow boundary, a moving front can be placed at a region near the boundary. This initial placement is called "seeding," and a seed can be just a point. This seed will gradually grow into a closed region and continue growing until it reaches an equilibrium which will be at the vehicle-shadow boundary – for a carefully chosen speed function F. Once the seed has reached the boundary, the resulting region can be then classified as either a vehicle or a shadow, and this information is conveyed to the CPM.

Collision Prediction Module

The Collision Prediction Module (CPM) employs techniques from the field of computational geometry to formulate and solve collision prediction problems. The approach we use is to find novel ways to represent features of an intersection as geometric entities (Figure 3). Such a representation allows the use of efficient data structures and algorithms which are essential for the CPM to work in real-time. The CPM represents all of the features in an intersection as simple geometric shapes like polygons or circles. These features are then categorized as fixed, transient, or moving obstacles based on the nature of the real world entities they represent. Further, these obstacles may be classified as rigid, quasi-rigid, or deformable depending on the temporal variation of the size and shape of the obstacle. Finally, the notion of obstacle modeling derives from the fact that every vehicle views every other entity in the intersection as a possible obstacle.

The CPM currently employs a simple array of vertices to represent each geometric entity. This representation together with the obstacle classifications completes the definition of real-world intersection entities. The obstacle classification helps in optimizing the collision prediction process by identifying classes that should not be checked against each other. For instance, fixed obstacles are not checked for collisions with each other, and pedestrians are never checked for collisions with fixed obstacles. Currently, the CPM has two collision-prediction schemes to choose from:

19. A simple, brute force algorithm that checks for collisions between each vehicle-type obstacle with every other obstacle for a specified number of future time steps.
20. A more sophisticated two-step algorithm that consists of dividing the entire intersection into a fixed number of polygons (e.g., rectangles), each maintaining an occupancy list of

obstacles that it contains (either partially or fully). Each vehicle-type obstacle is then checked for collisions with every other obstacle in all of the occupancy lists in which it is listed.

The CPM chooses the first approach when the number of vehicles in the intersection falls below a pre-determined number and switches to the more efficient two-step algorithm when the intersection traffic increases. Both the approaches rely on a polygon-clipping algorithm to detect possible collisions between the obstacles. Clipping in two dimensions is the process of 'trimming' a geometric shape about a clipping line, just like cutting a paper along straight lines with a pair of scissors. The clipping algorithm operates on two polygonal entities by choosing one of them as reference. The edges of the reference polygon are then used to 'clip' the other polygon sequentially until all the edges of the reference polygon have been used. If the clipped area of the other polygon (i.e., the area it shares with the reference polygon) is zero, then we conclude that there is no collision; otherwise, there is a collision. Once the CPM has been tested and integrated into the Vision Module, we will explore techniques for speeding up collision prediction using efficient geometric techniques [4].

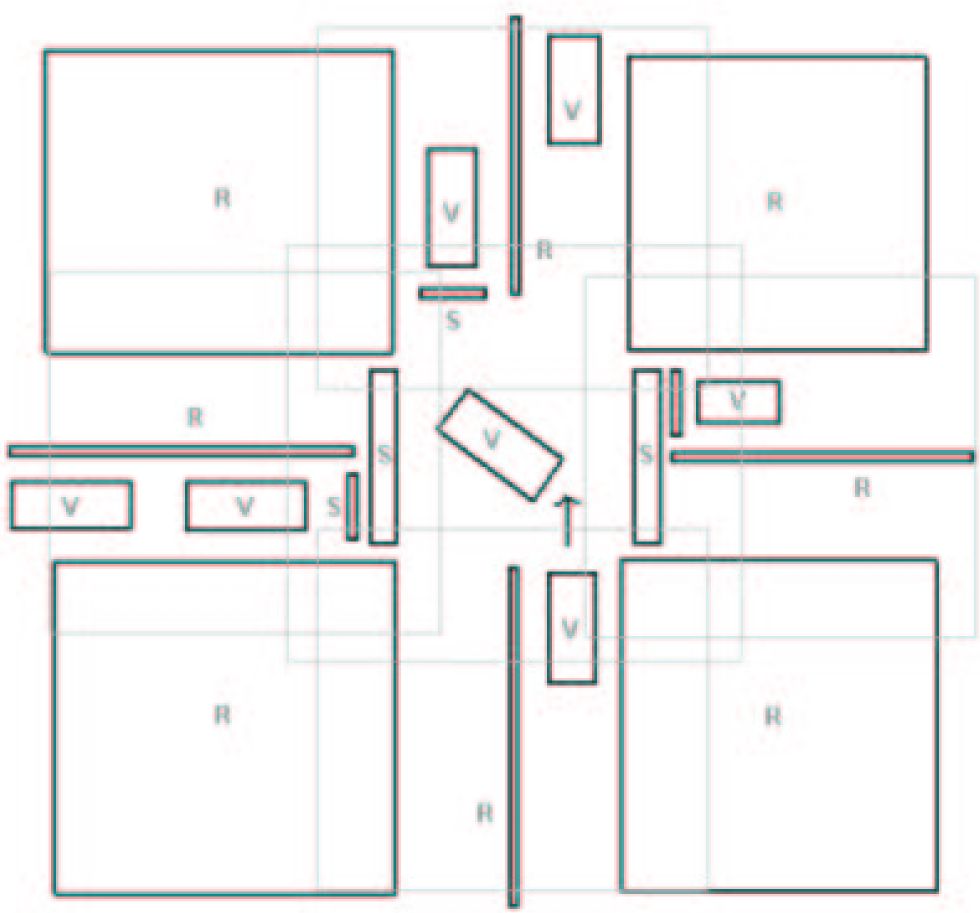

Figure 3: Intersection features modeled as geometric entities. R, V, and S are geometric representation of road features, vehicles and signals respectively. The thin lines show the partition scheme used for computing the occupancy list in the two-step collision prediction algorithm.

Conclusions

Intersection monitoring and collision prediction is a challenging problem, yet one that has the potential to save thousands of lives once it has been solved. We are currently working on a system to deal with this problem and the issues associated with it, such as occlusions, shadows, and the stop-and-go problem. At present we have created a Vision Module, to track vehicles, as well as a Collision Prediction Module, to predict potential collisions. As we continue in our research and refine our system, we look forward to finding a solution to these problems and enhancing traffic safety.

Acknowledgments

This work has been supported by National Science Foundation through grant CMS-0127893, the ITS Institute (University of Minnesota), and the Minnesota Department of Transportation.

Endnotes

21. D. Smith and W. Najm, “Analysis of crossing path crashes.” Volpe Center Technical Report, 1999.
22. T. Penney. “Intersection collision warning system.” Pub. No. FHWA-RD-99-103, 1999.
23. B. Ferlis. “Analysis of Infrastructure-based system concepts – Intersection Collision Avoidance Problem Area.” HRDO, 1999.
24. P. Gupta, R. Janardan, and M. Smid. “Fast algorithms for collision and proximity problems involving moving geometric objects.” *Computational Geometry: Theory and Applications,* 371(6), 1996.

25. M. Haag and H.-H. Nagel. “Incremental recognition of traffic situations from video image sequences.” *Image and Vision Computing*, 137(18), 2000.

26. H. Kollnig and H.-H. Nagel. “Matching object models to segments from an optical flow field.” Fourth European Conference on Computer Vision, 15(2), 1996.

27. H.-H. Nagel, T. Schwarz, H. Leuck, and M. Haag. “T3wT: tracking turning trucks with trailers.” IEEE Workshop on Visual Surveillance, 65, 1998.

28. H. Leuck and H.-H. Nagel. “Automatic differentiation facilitates OF-integration into steering-angle-based road vehicle tracking.” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 360(2), 1999.

29. J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

30. D. Halperin, L. Kavraki, and J.-C Latombe. “Robotics.” *Handbook of Discrete and Computational Geometry*, Ch. 41. J. Goodman and J. O'Rourke (eds.)

31. C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-timetracking.” Proc. Computer Vision and Pattern Recognition Conf. (CVPR '99), June 1999.

32. O. Masoud, “Tracking and analysis of articulated motion with application to human motion,” Ph.D. Thesis, Dept. of Computer Science and Engineering, University of Minnesota, 2000.

33. Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

34. J. Sethian, “A Fast Marching Level Set Method for Monotonically Advancing Fronts”, Proc. National Academy of Sciences, 93(4), 1996.