Excel@FIT 2016
http://excel.fit.vutbr.cz

Multiple Vehicle Detection and Tracking from Surveillance Camera with Collision Prediction

Martin Kruták*

Abstract
This paper describes a system for detection and tracking of multiple vehicles from a surveillance camera with collision detection and prediction. Accurate vehicles' contour is obtained in the detection phase, and object centroids are calculated. Each detected vehicle is assigned to the specific lane and tracked separately. Calculated centroids are used for object tracking using a contour-based algorithm with movement prediction which provides sufficient amount of information to predict vehicle movement. A rectangle constructed around the ground part of the vehicle is used for vehicle collision prediction. Experimental results show a success rate of 71 % when constructing the ground rectangles, which are key for collision prediction.

Keywords: vehicle tracking – vehicle detection – collision prediction

Supplementary Material: Demonstration Video

*xkruta03@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Intelligent systems for traffic control support basic management systems such as car navigation, traffic signal control systems, license plate number recognition or speed cameras. In the modern world where, according to World Health Association [1], over 1.25 million people were killed in car accidents in 2015, we are missing one key feature to predict a collision. This early warning system would help to prevent the collisions.

In this paper, the way of tracking multiple vehicles with prediction and possible detection of a collision of vehicles from a surveillance camera is introduced.

The focus is on two main potential types of collisions that might occur on the highway in one direction. The first case is the one where one vehicle collides with another by hitting it from rear. In the second case, the vehicles might collide in the process of overtaking.

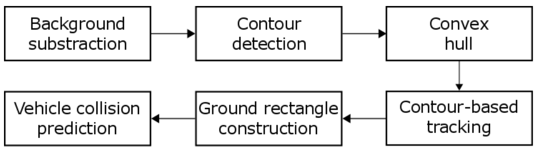

Figure 1. Process of collision detection.

One of the possible approaches is to construct a rectangle around the ground part of the vehicle. Then by enlarging this rectangle check for possible overlaps with other objects.

In the case of tracking multiple objects, we are focusing on the contour method which is efficient and fast at the same time (Figure 1).

2. Related work

To predict the collision of vehicles, they need to be tracked individually. Thus it is possible to study their movements. When the object is treated as individual, it is easier to study its behavior.

When tracking the objects, it is usually done by frame to frame tracking. There are four categories of tracking methods [2]: feature based, contour based, region based and model based.

Feature-based algorithms extract features and then match those between frames. Features to be tracked can be color, size, edges, shadow or symmetry [3].

Another way is to track the objects by their contour. The contour is updated in consecutive frames [4]. These algorithms are less complex than the first ones. However, this approach suffers from partial inability to segment vehicles that are partially occluded.

Model-based algorithms match the model of the object that is tracked [5]. These algorithms are robust, but slower.

Region based algorithms track objects given their region [6]. The motion region is usually obtained through background subtraction. These algorithms do not work well for multi-object tracking.

The more sophisticated way of vehicle tracking is a fusion of data from multiple cameras. It is possible to use features of location, intensity and geometry to match the vehicles taken by different cameras [7].

In the case of prediction of collisions, the range of possibilities is slightly thinner. One way is to predict the trajectory of the vehicle, based on subsequent learning of vehicles' movement [8]. By modeling those trajectories, it is possible to estimate the position of the vehicle and then compare the trajectory with trajectories of other vehicles. The focus of most of the papers is on collision at traffic intersections, as well as in [9], where the proposed method to predict the collision is by creating a three-dimensional polytopes. Those polytopes are then tested for overlaps. Vincent Hayward et al. [10] focus to predict a collision among many moving objects by defining an *urgent* and *closest* pairs of objects, testing only those which fit the conditions.

3. Implementation of tracking algorithm with collision prediction

The system comprises several modules. Firstly, a contour of the vehicle is obtained, then using the centroid of the contour, the vehicle is assigned to the specific lane. The vehicles are treated as objects with a few properties like exact position of the contour, line position or the coordinates of a centroid. A rectangle is constructed around the ground part of the vehicle. Using this rectangle the possible collision is predicted.

Figure 2. Sometimes the blob tends to merge when a vehicle is overtaking another (left) or given the imperfection of detected contour, shatter (right).

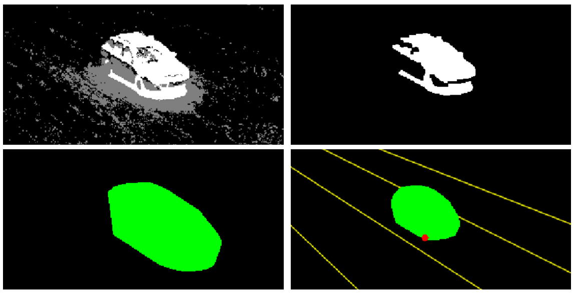

Figure 3. The mask before amendment (top left) – detected shadow is gray, mask after amendment (top right), the final contour (bottom left) and the point moved to the bottom of the contour (bottom right)

3.1 Vehicle detection and blob amendment

The vehicles in the video sequence are detected by Gaussian mixture-based background/foreground segmentation algorithm [11]. The algorithm estimates the background and extracts the foreground which are preferably only the objects which are moving – estimated with the threshold of the reference image (background) and the current frame.

The output mask is not perfect though (Figure 3). In the next step, the shadow is removed and morphological operations are applied to get rid of the noise.

Even though the mask is yet not impeccable. To get the desirable result, a contour of the mask is obtained [12]. Around the contour a convex hull [13] is constructed (Figure 3).

3.2 Centroid calculation

The moments of the polygon (contour) are calculated to get the centroid of the object:

$$m_{ij} = \sum_{x,y} (array(x,y) \cdot (x-\bar{x})^j \cdot (y-\bar{y})^i) \qquad (1)$$

where $(\bar{x}, \bar{y})$ is the mass center:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \qquad (2)$$

The point given by centroid's coordinates is moved to the bottom of the contour (Figure 3) to ensure its position in the lane.

3.3 Assigning the lane

To assign the object to the lane, we will compute the distance of the calculated point at the bottom of the

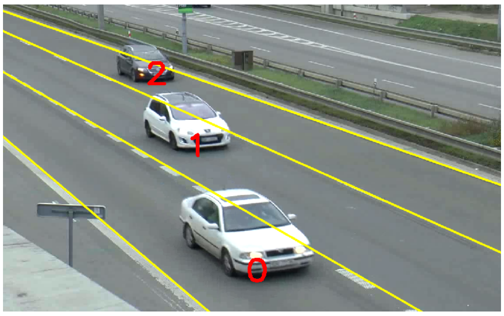

Figure 4. The number at the vehicle represents the lane the vehicle is on.

contour to the lines. Lines are not detected automatically, those have to be detected a priori and are then loaded separately. We expect the line to be given in general form $p : ax+by+c=0$. The point has coordinates $A[x,y]$, then the distance is computed as follows:

$$v(A,p) = \frac{|a \cdot x + b \cdot y + c|}{\sqrt{a^2+b^2}} \tag{3}$$

The algorithm stores the information to which of the lines is the point closest to. Those two lines define a lane. This lane is then assigned to the object (Figure 4). In this step, the vehicles going in other direction (away from camera) are removed. Those are the objects whose distance to the top line is shorter than to the line one below.

3.4 Tracking of multiple objects

The contour of each vehicle is treated as individual object. Deciding the identity of the object in the subsequential frame in time *t* is done by comparing them to detected objects in time *t-1*. In the process of the object to object comparison, several cases may occur (Figure 2). 1) No intersection are detected – that means a new object has appeared in the frame. 2) There is only one intersection – that means that we have found the object from the last frame in the current frame. 3) There are two intersections – that means, that in the current frame two blobs have merged into one or the blob have shattered into two smaller blobs (that also applies for the possibility of more than two intersections). In the first case the object in not updated, but moved in expected direction. Moving of the object is done as follows. If we know what distance in which direction the object traveled in past, it is assumed that this object will follow that direction and travel the similar distance in the future. If this information is unknown, it is obtained from closest known objects. In the second case, it is checked that all parts ale in one lane (means one object). Those parts are then merged together.

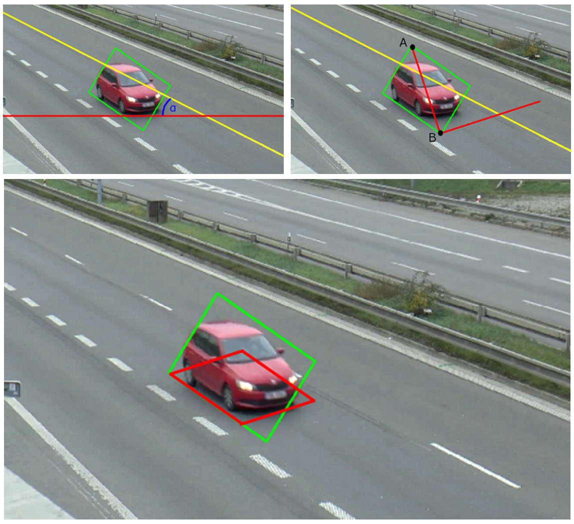

Figure 5. The rotated rectangle with angle of rotation α (top left), construction of the main guiding line (top right), final ground rectangle (bottom).

3.5 Collision prediction

The contour, as it is, is not a reliable feature to be directly used in collision prediction. Because of that, we focus on the part that will presumably collide; that is the ground part of the car. First, we create a rectangle around the bottom part of the vehicle (Figure 5). This is achieved by enclosing a rectangle around the vehicle's contour. This rectangle is rotated in a way of vehicle's movement. A diagonal of the rectangle is constructed and a line perpendicular to the diagonal is the first side of the rectangle. This line is moved to the point where it touches the contour, being the front of the vehicle. A line at the back of the car is parallel with the front one, going through the far bottom corner of the rectangle. The third line is parallel with the side of the rectangle and the fourth line is parallel to the third one and moved so far so it does not go out of the lane the vehicle is in.

When constructing the rectangle, the proposed method is efficient only in left or right side view of the camera.

The ground rectangle is constructed for every vehicle detected in the frame. Then, as stated in [8], ”Given the position, orientation, and size, at each time step, of n oriented rectangles in 2-D, find all possible pairs of rectangles that are within a distance δ in the current and future time steps.” Given δ is the maximum distance between vehicles, the size of every rectangle is enlarged about $\delta/2$ and checked for overlaps with other vehicles.

4. Experimental results of the ground rectangle construction

The subject to testing was the ground rectangle constructed around the ground part of the vehicle. The precision of construction of the ground rectangle is highly dependent on the vehicle detection process with association to the consistency of the blob.

Detection Rate – is a ratio of precisely constructed rectangles to all constructed rectangles. It is defined as follows:

$$DetectionRate = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4)$$

Recall – is a ratio of precisely constructed rectangles around the objects to all the objects that should have to have their rectangle constructed around them. It is defined as follows:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

From Table 1 it is evident that the detection rate (precision) is at 71 %.

Table 1. System Performance

	Detection Rate	Recall
Implemented system	0.708	0.661

5. Conclusion and future work

A vision-based system of tracking multiple objects in video and their collision prediction is introduced in this paper. The tracking method is a contour-based method, where contours of each detected object are compared in subsequent frames. The collision prediction offers the approach of prediction by constructing the rectangle enclosing a ground part of the vehicle, enlarging the rectangle and then checking this rectangle for the overlaps with objects' rectangles around.

Certain parts of the system need user-defined inputs (e.g. lines on the road), so they are not obtained automatically at this time. Automatic camera calibration to the scene and a use of Kalman filter for tracking the objects would improve the system. There are further possibilities to fit other modules to the system like size, speed measurement or collision detection.

Acknowledgements

I would like to thank my supervisor Ing. Jakub Špaňhel for his help and valuable advices during the writing of this paper and the work on the project.

References

[1] World Health Organization. *Global status report on road safety 2015*. World Health Organization, 2015.

[2] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[3] Dieter Koller, J Weber, T Huang, J Malik, G Ogasawara, B Rao, and S Russell. Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126–131. IEEE, 1994.

[4] Zhimin Fan, Jie Zhou, and Dashan Gao. Contour extraction and tracking of moving vehicles for traffic monitoring. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 84–87. IEEE, 2002.

[5] David Serby, EK Meier, and Luc Van Gool. Probabilistic object tracking using multiple features. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 184–187. IEEE, 2004.

[6] Boyoon Jung and Gaurav Sukhatme. A generalized region-based approach for multi-target tracking in outdoor environments. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2189–2195. IEEE, 2004.

[7] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 3–10. IEEE, 2000.

[8] Stefan Atev, Hemanth Arumugam, Osama Masoud, Ravi Janardan, and Nikolaos P Papanikolopoulos. A vision-based approach to collision prediction at traffic intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 6(4):416–423, 2005.

[9] Nicolas Saunier, Tarek Sayed, and Clark Lim. Probabilistic collision prediction for vision-based automated road safety analysis. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 872–878. IEEE, 2007.

[10] Vincent Hayward, Stéphane Aubry, André Foisy, and Yasmine Ghallab. Efficient collision predic-

tion among many moving objects. *The International journal of robotics research*, 14(2):129–143, 1995.

[11] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

[12] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[13] Jack Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, 1(2):79–83, 1982.