CAMERA ORIENTATION-INDEPENDENT PARKING EVENTS DETECTION

Piotr Dalka, Damian Ellwart, Grzegorz Szwoch

Gdansk University of Technology, Multimedia Systems Department, Gdansk, Poland
e-mail: piotr.dalka@sound.eti.pg.gda.pl

ABSTRACT

The paper describes the method for detecting precise position and time of vehicles parking in a parking lot. This task is trivial in case of favorable camera orientation but gets much more complex when an angle between the camera viewing axis and the ground is small. The method utilizes background subtraction and object tracking algorithms for detecting moving objects in a video stream. Objects are classified into vehicles and non-vehicles with SVM based on their shapes. Detection of the time moment when a vehicle parks utilizes spatial and temporal rules. Parking place identification (i.e. where a vehicle has parked) involves distance measurements between the vehicle position and right prisms that represent parking places in 3D space. The paper concludes with results description of a parking event detection based on the real-time analysis of several hours of video streamed from a real parking lot.

1. INTRODUCTION

With increasing number of video cameras in every city, there is a need for automatic analysis of video streams in order to detect various events in the real time. Depending on the camera placement, such systems may be used for many applications, such as general trespassing discovery [1], human activity analysis and suspicious behavior detection, like loitering [2], fighting [3], vandalism [4] or luggage abandonment [5].

In case of parking lots, especially the larger ones with high traffic, an automatic video surveillance system can be used to detect when and where a vehicle has parked; obtaining the same data by parking security employees is error prone and practically impossible. Such data (among others) is necessary for successful parking lot management, i.e. verification whether a vehicle parked at the right place or updating parking place occupancy map.

This paper present a module for parking event detection that forms the important part of a parking lot management system. Its goal is to detect the moment when a vehicle parks and to identify its parking place. The module consists of a few components. First, all moving objects are detected and tracked as long as that they stay in a camera field of view. This tasks is described in Section 2 of the paper. Section 3 contains details regarding the algorithm for vehicle classification. Its task is to distinguish vehicles from non-vehicles (e.g. humans). Without the algorithm, a human stopping for a longer time in a parking place would always trigger the parking event. Section 4 describes the parking event detection algorithm, including components for parking time and parking place identification. Results of experiments conducted in the real parking lot equipped with 7 cameras are presented in Section 5. Section 6 concludes the paper.

2. MOVING OBJECT DETECTION AND TRACKING

Objects moving in the parking lot are detected independently in each in video frame acquired from a fixed camera. The algorithm based on background modeling method utilizing Gaussian Mixtures is used for this purpose as it proved to be effective in earlier experiments [6]. The results of background modeling are processed by detecting and removing shadow pixels (basing on the color and luminance of pixels and on the texture of shaded regions) and by performing morphological operations on the detected objects in order to remove small areas and to fill holes inside objects.

Next, movements of the detected objects (blobs) are tracked in successive image frames using a method based on Kalman filters that allow predicting object position in the current frame based on observations in previous frames. By comparing results of background subtraction in the current frame with predicted object positions it is possible to correlate each tracker (Kalman filter representing one real object) with the detected moving object (including partial occlusions), so the movement of each object is tracked continuously. In case of conflicts, when the assignment of detected moving objects to trackers is ambiguous (overlapping, covered, splitting and merging objects, etc.), the correct association is provided by the algorithm that iteratively analyses blob-tracker clusters and assigns blobs to trackers based on their visual similarity [7].

In order to make parking event detection robust against possible false-positive results of moving object detection (e.g. in case of a sudden lighting change) or in case of background scene modifications (e.g. leaving an object, opening the door, etc), only objects entering the camera field of view from the outside are taken into account. The algorithm is as follows. In a video frame, hot areas are selected manually. They represent entry/leave regions in a video frame and are defined as polygons placed on roads and other areas suitable for vehicle movement (Fig. 1). Moving object is further analyzed only if:

1. It appeared in the vicinity of a hot area; the condition is implemented by checking intersection of a bounding rectangle of the new object with any hot area (object tracking parameters and hot area localizations guarantee that all interesting objects match this condition),
2. It entered the hot area (a center point of the bounding rectangle or other point defined relatively to the rectangle for the particular hot area is found inside the hot area),
3. It left the hot area (the point of interest is found outside the hot area)
4. And it moves in the direction of the frame center.

The last condition is examined by calculating the angle α between the short-term averaged direction of movement and the heading to the frame center. The angle α is given as:

$$\alpha = \arccos\left(\frac{(\mathbf{c}-\mathbf{p})\cdot\mathbf{v}}{\|\mathbf{c}-\mathbf{p}\|\cdot\|\mathbf{v}\|}\right) \tag{1}$$

where **c** denotes coordinates of a video frame center, **p** - coordinates of the current object position (the center of the bounding rectangle) and **v** - a vector containing short-term averaged velocity of the object along both axes. If angle α is less than 90°, it is assumed that the object moves in the direction of the frame center.

Fig. 1. An example of hot areas placement (white polygons) and a moving vehicle entering the frame in the right hot area (gray rectangle).

3. VEHICLE CLASSIFICATION

Depending on the type of an analyzed object, various actions could be defined. For the purpose of detecting parking events, only objects representing vehicles are relevant.

There are several methods which could be used for object classification. The simplest methods threshold object pixel dimensions and other pixel-based parameters. Unfortunately these values vary depending on the camera position and camera orientation. In this paper a shape based classification method is used which requires only binary image information to distinguish between object types.

In monitoring systems the shape of an observed object vary not only depending on the camera viewpoint but also depending on the object rotation around its own axis. Because of that, creating a universal shape based classifier becomes impossible. To solve this problem, a set of classifiers is created corresponding to three various camera observation angles: 40^o, 60^o and 80^o. To prepare two datasets required for proper classifiers training, 3D models representing two object types (vehicles and humans) are built. By projecting them from the defined observation angle a set of binary images is produced (Fig. 2). To lower the amount of data being processed and unify its description, such prepared data needs to be parameterized.

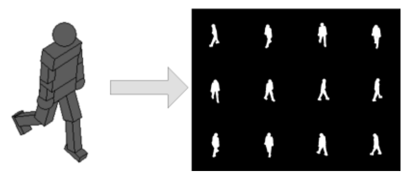

Fig. 2. Dataset generation for a defined observation angle.

To make the parameterization process insensitive to object scale, input masks are resized to a reference dimensions constraining their proportions. During research this value was set to 100x100 pixels. Vectors describing a single object, used afterwards for classifier training process, consist of 207 parameters. First 200 parameters are simply mask row and column projections normalized according to the defined reference dimension. Last 7 parameters are Hu invariant moments. Such dataset becomes an input for SVM classifier. While training a certain classifier the images representing the second class become the negative examples. For each classifier this set is additionally extended with images representing none of the recognized classes. Training process itself occurs using grid search method [8] assuring optimal cost and gamma parameter.

At this point many classification errors can be still present. Therefore results obtained from this block are stored and averaged through the object lifetime. Furthermore to prevent the classification algorithm from analyzing very small objects that could be misclassified, the minimum object size is set depending on the scene characteristics. With these restrictions the presented classifier achieves the efficiency of about 90% for recognizing between two classes: vehicles and non-vehicles. The classification results are sent to the following algorithms which decide whether a specified event could occur for that type of object.

4. PARKING EVENT DETECTION

Parking event detection consist of two stages. First, the fact that a vehicle parked at any parking place is detected. Then, the actual parking place is identified. Parking places are defined as polygons "drawn" in the image frame. Sample parking places are presented in Fig. 3.

Fig. 3. Parking places (light polygons) selected in video frames from two cameras with different viewing angles

4.1. Parking vehicle detection

Detection of vehicles parking in any parking place is implemented as a set of rules. First of all, only moving objects that entered the frame from the outside (Section 2) and are classified as vehicles (Section 3) are considered. A vehicle is considered as parked if it entered any parking place, stopped while still in the parking place and stayed in the vicinity of its current position for the defined amount of time. If the vehicle leaves the current position the countdown is terminated and the whole series of conditions must be repeated for the event to trigger.

Entering or leaving a parking place is detected when the center of vehicle bounding rectangle crosses the boundary of a polygon denoting the parking place, in a proper direction.

Stopping happens when a short-term average of a vehicle speed is smaller than the predefined threshold for a few video frames (e.g. five) in a row. When a vehicle stops, the current bounding rectangle of a vehicle becomes its stopping area. If the center of the vehicle leaves the stopping area, it is considered that the vehicle left its current position.

The time span required for the vehicle to stay in place cannot be too small in order to avoid false-positive errors caused, for example, by vehicles turning back in parking places. On the other hand, the higher the time span the greater delay in parking event alerting. Time span of 20 seconds has been chosen as a compromise.

The algorithm might seem over-complicated but it is necessary to limit false negatives and false positives errors caused by moving objects overlapping with each other and especially, by a driver (and passengers) leaving the vehicle and thus altering its bounding rectangle significantly.

4.2. Parking place identification

Whenever a parking vehicle is detected, its parking place is identified. This task is trivial if the camera looks straight down to the ground (the angle between the camera viewing axis and the ground is nearly perpendicular). However, the smaller the angle is the more difficult it is to determine where a vehicle parked based only on its bounding boundary and parking places marked on the ground level because of the increasing impact of the vehicle height on its perspective projection on the ground level (Fig. 3). Furthermore, in case of wide angle lenses, significant differences are also visible within the same video frame.

In order to make parking place identification possible for small camera viewing angles, the calibration procedure of a camera needs to be performed. Calibration is used to find an unambiguous conversion between the position of any point in the real-world, 3D coordinate system and the pixel (image) position of the same point in a video frame. The reverse conversion (2D to 3D) is not direct since one of coordinates in the real-world system has to be predefined (typically, it is z coordinate, describing the height of the point). The parameters for the conversion are calculated by the calibration model, fed with pairs of image and world coordinates for multiple calibration points. In the experiments, Tsai's calibration model was used, as it proved to be an efficient tool for camera calibration [9]. Using the provided calibration points, Tsai's method calculates a total of 11 transformation parameters, related to camera lenses (focal length, lens distortions) and camera orientation (translation and rotation).

Each parking place for the purpose of its identification is modeled with a right prism (usually a cuboid) in 3D real world coordinate system. Its lower base is formed by a polygon labeled in a video frame that is converted to 3D coordinates with the assumption, that it is placed on the ground level. Upper base of the right prism is formed by its lower base lifted 1.5 meters above the ground. The height of the prism corresponds to the average height of cars. Fig. 4 presents sample parking places with lower and upper bases of prisms displayed.

Fig. 4. Lower (gray) and upper (white) bases of 3D right prisms that are used to model four parking place seen in a video frame.

In order to determine where a vehicle has parked, its distance from all parking places is calculated according to the equation:

$$d_i = \|\mathbf{u_i} - \mathbf{p}\| + \|\mathbf{l_i} - \mathbf{p}\| \quad (2)$$

where d_i denotes distance of a vehicle from the i-th parking place, **p** - the vehicle position as the center of the vehicle bounding box in the moment when the parking event is detected, $\mathbf{u_i}$ and $\mathbf{l_i}$ - image plane projected coordinates of the centers of upper and lower bases of the prism corresponding to the i-th parking place.

The vehicle **p** is assumed to park in the parking place i if the distance d_i is the smallest comparing to distances to all other parking places.

5. EXPERIMENTS

For the purpose of experiments, the prototype video surveillance installation covering a parking lot around an office building has been used. The installation consists of 7 fixed cameras with different orientations in relation to the ground, covering a total number of 54 different parking places. Sample frames from a few cameras are presented in Fig. 1, 3 and 4. Hot areas for the purposes of finding object entering the video frame from the outside and polygons denoting parking place locations were labeled manually.

Approximately 24 hours of video streams covering 4 week days have been analyzed in the real time from all 7 cameras in order to detect all parking events. Results were verified manually and they are summarized in Table 1.

Table 1. Summary of parking events detection results

Number of parking events	Correct detections	False negatives	False positives
284	281	5	45

Results of parking event detection are characterized by the very small false negative rate and larger false positive rate. Over 98% of parking vehicles have been detected correctly. A few missed events were caused by object detection and tracking errors in a crowded scene, when a vehicle bounding rectangle was not allowed to remain still for the required period of time.

There are also false positives that account for almost 16% of real parking events. They are caused mainly by errors in object tracking (multiple trackers associated with the same real vehicle) and by mistakes in object classification (e.g. a person stopping in a parking place).

In all parking events, parking places have been identified correctly, regardless of various camera viewing angles.

6. CONCLUSIONS

The paper presents a solution for detecting vehicle parking for an automatic parking lot management system. The results achieved prove that the software is able to detect practically all parking events, with a small amount of false positives generated. The number of mistakes may be attenuated by improving accuracy of low-level image processing modules.

Future work will be focused on improving total efficiency and on adding additional functionality, including parking vehicle identification. Additionally, an attempt will be made to label parking places and hot areas automatically.

7. ACKNOWLEDGEMENTS

Research is subsidized by the European Commission within FP7 project “INDECT” (Grant Agreement No. 218086).

8. REFERENCES

[1] J. Black, S.A. Velastin, and B. Boghossian, “A real time surveillance system for metro-politan railways,” *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, pp. 189–194, 2005.

[2] N.D. Bird, O. Masoud, N.P. Papanikolopoulos, and A. Isaacs. “Detection of loitering individuals in public transportation areas,” *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 167–177, 2005.

[3] A. Datta, M. Shah, and N.D.V. Lobo, “Person-on-person violence detection in video data,” *Proc. of the 16th Int. Conf. on Pattern Recognition*, vol. 1, pp. 433–438, 2001.

[4] M. Ghazal, C. Vazquez, and A. Amer, “Real-time automatic detection of vandalism be-havior in video sequences,” *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pp. 1056–1060, 2007.

[5] G. Szwoch, and P. Dalka, “Automatic Detection of Abandoned Luggage Employing a Dual Camera System,” *Proc. of IEEE International Conference on Multimedia Communications, Services and Security*, pp. 56–61, Krakow, Poland, 2010.

[6] A. Czyzewski, P. Dalka, “Moving object detection and tracking for the purpose of multimodal surveillance system in urban areas,” *Proc. 1st Int. Symp. on Intell. Interactive Multim. Syst. and Services*, Piraeus, Greece, 2008.

[7] A. Czyzewski, P. Dalka, “Examining Kalman filters applied to tracking objects in motion,” *Proc. of 9th Int. Workshop on Image Analysis for Mult. Interact. Services*, pp. 175–178, 2008.

[8] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A practical guide to support vector classification,” *Technical report, Department of Computer Science, National Taiwan University*, July, 2003.

[9] P. Dalka, A. Ciarkowski, P. Szczuko, G. Szwoch, and A. Czyzewski, “Surveillance Camera Tracking of Geo positioned Objects,” *Studies in Computational Intelligence: New Directions in Intelligent Interactive Multimedia Systems and Services*, vol. 226, pp. 21-30, 2009.