Proceedings of the IEEE ICRA 2009
Workshop on People Detection and Tracking
Kobe, Japan, May 2009

Real-Time Object Tracking and Classification Using a Static Camera

Swantje Johnsen and Ashley Tews

***Abstract*— Understanding objects in video data is of particular interest due to its enhanced automation in public security surveillance as well as in traffic control and pedestrian flow analysis. Here, a system is presented which is able to detect and classify people and vehicles outdoors in different weather conditions using a static camera. The system is capable of correctly tracking multiple objects despite occlusions and object interactions. Results are presented on real world sequences and by online application of the algorithm.**

I. INTRODUCTION

It is important for vehicle operators around worksites to be aware of their surroundings in terms of infrastructure, people and vehicles. When an operator observes an object moving in a way that will impact on their operations, they take the necessary steps to avoid undesired interaction. Their response depends on recognising the type of object and its track. This skill is also important for autonomous vehicles. An autonomous vehicle needs to be able to react in a predictable and rational manner, similar to or better than a human operator. Onboard sensors are the primary means of obtaining environment information but suffer from occlusions. However, offboard sensors such as webcams commonly deployed around worksites can be used for this purpose. We present our system for offboard dynamic object tracking and classification using a static webcam mounted outside a building that monitors a typical open work area. As the preliminary step towards integrating the extracted information to improve an autonomous vehicle's situational awareness, information about the objects such as location, trajectory and type is determined using a tracking and classification system. The system consists of several existing subsystems with improvements in the detection and classification phases. The system is capable of working in different weather conditions and can distinguish between people and vehicles by identifying recurrent motion, typically caused by arm or leg motion in the tracked objects. Tests were conducted with different types and numbers of vehicles, people, trajectories and occlusions with promising results.

II. RELATED WORK

The common architecture of classification systems consists of the following three main steps: motion segmentation, object tracking and object classification [1] [2]. The steps are described as follows.

S. Johnsen is with the Institute for Reliability of Systems, Hamburg University of Technology, Eissendorfer Str. 40, 21073 Hamburg, Germany {swantje.johnsen}(at)tu-harburg.de

A. Tews is with the Commonwealth Scientific and Industrial Research Organisation (CSIRO) in Brisbane, Australia {ashley.tews}(at)csiro.au

In the motion segmentation step, the pixels of each moving object are detected. Generally, the motion segmentation consists of background subtraction and foreground pixel segmentation. Stauffer and Grimson [3] use the mixture of Gaussians to perform background subtraction and apply a two-pass grouping algorithm to segment foreground pixels. Simple and common techniques are based on frame differencing [4] or using a median filter [5]. In this work a technique based on the Approximated Median Filter [6] was used. Better results were obtained by introducing a step factor in the filter.

Following background subtraction, the mobile objects are tracked. Tracking of objects is the most important but error prone component. Problems arise when objects of interest touch, occlude and interact with each other, and when objects enter and leave the image. Israd and Blake [7] introduced a method termed CONDENSATION to track objects. Chen *et al*. [8] construct an invariant bipartite graph to model the dynamics of the tracking process. Stauffer and Grimson [3] use a linearly predictive multiple hypotheses tracking algorithm. Yang *et al*. [4] use a correspondence matrix and a merging and splitting algorithm to relate the measured foreground regions to the tracked objects. Many algorithms have been proposed in the literature, but the problem of multiple interacting objects tracking in complex scene is still far from being completely solved. Model based algorithms [9] are computationally more expensive, because the number of parameters to estimate the model is usually large. They are also sensitive to background clutter. Overall, many of those algorithms can only deal with partial object occlusions for a short duration and fail to deal with complete object occlusions.

In the classification step, the object type is determined. Classification of 3-dimensional moving objects from 2-dimensional images for known object classes is a highly complex task. Toth and Aach [10] use a feed-forward neural network to distinguish between human, vehicles, and background clutters. Rivlin *et al*. [11] use a Support Vector Machine to distinguish between a vehicle, a human and an animal. Zhang *et al*. [2] distinguish between cars, vans, trucks, persons, bikes and people groups. They introduced the error correction output code as a classifier. These techniques need to be trained via test sequences of the objects. Javed and Shah [1] produced an algorithm that does not need to be trained.

III. SYSTEM OVERVIEW

A system that observes an outdoor environment by a single static camera is developed and tested. The goal is to track objects like walking people or moving vehicles in view of the camera and to determine their type and position. In Figure 1 the flow diagram of the system is shown. The motion segmentation step detects the moving objects using the current image in the image stream. This output (the moving objects) is required by the object tracking algorithm that provides the motion history of each object.

A particular characteristic of the tracking algorithm is its ability to track objects with complete occlusion for a long duration without knowledge about their shape or motion. The output of the tracking algorithm is used by the classification system. Our classification algorithm is a modified version of the system presented in Javed and Shah [1]. The algorithm uses on the motion history of each object and by determining the type of motion. Motion type is determined by any repeated, recurrent motion of the object's shape. This property is used to classify between people and vehicles.

The motion segmentation, tracking and classification steps are dependent on each other. Thus, the classification system would deliver inappropriate results, if one of the previous steps does not achieve good performance.

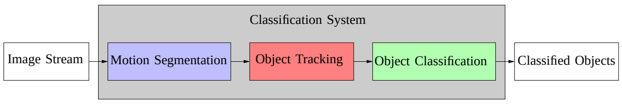


Fig. 1. Flow diagram of common classification systems.

The tests and experiments in this paper were conducted with a Canon VB-C50ir PTZ webcam. The maximal transmission rate of the camera is $25\,fps$ and it captures 768×576 resolution color images. Our system is developed in the c++ programming language on a 3.2 GHz Pentium D using the Open Source Computer Vision library (OpenCV).

IV. MOTION SEGMENTATION

An important condition in an object tracking algorithm as well as in an object classification algorithm is that the motion pixels of the moving objects in the images are segmented as accurately as possible. The common approach for motion segmentation consists of two steps: background subtraction and segmentation of foreground pixels.

A. Techniques of Background Subtraction

Background subtraction [12] identifies moving objects by selecting the parts of the image which differ significantly from a background model. Most of the background subtraction algorithms follow a simple flow diagram shown in Figure 2. Background modeling is a statistical description of the current background scene. Foreground pixel detection identifies the pixels in the current image that differ significantly from the background model and outputs them as a binary candidate foreground mask.

The Approximated Median Filter was chosen to perform background modeling. For our implementation, better results were obtained by scaling the increment and decrement by a step factor if the absolute difference between the current pixel and the median-modeled background pixel is bigger than a threshold.

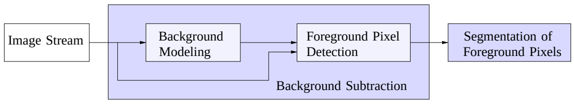


Fig. 2. Flow diagram of a general background subtraction algorithm.

Foreground pixels are detected by calculating the Euclidean norm at time t:

$$\|\mathbf{I_t}(x,y) - \mathbf{B_t}(x,y)\| > T_e \tag{1}$$

where $\mathbf{I_t}$ is the pixel intensity value, $\mathbf{B_t}$ is the background intensity value at time t and T_e is the foreground threshold or by checking

$$|I_{j,t} - B_{j,t}| > T_a \tag{2}$$

for $j = 1,\dots,c$ where T_a is the foreground threshold,

$$\mathbf{I_t} = \begin{bmatrix} I_{1,t} & \dots & I_{c,t} \end{bmatrix}^T, \quad \mathbf{B_t} = \begin{bmatrix} B_{1,t} & \dots & B_{c,t} \end{bmatrix}^T \tag{3}$$

and c is the number of image channels. The foreground thresholds T_e and T_a are determined experimentally. The foreground pixels were detected by determining the threshold T_a.

B. Segmentation of Foreground Pixels

In the next step, foreground pixels are segmented into regions. Using the two-pass connected component labeling method [3], a bounded box is applied to the connected regions. After this step, only grouped regions with bordered rectangles are considered. Any remaining noise is removed in the second noise reduction step using a size filter [13]. Finally, blobs are merged if they intersect or if the distances between them are below a threshold depending on the object distance to the camera.

V. MULTIPLE OBJECT TRACKING WITH OCCLUSION HANDLING

The goal of tracking is to establish correspondences between objects across frames. Robust classification of moving objects is difficult if tracking is inaccurate. The flow diagram of the implemented object tracking algorithm is shown in Figure 3.

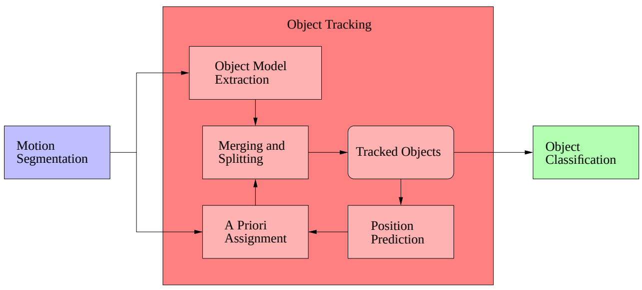


Fig. 3. Flow diagram of the multiple object tracking algorithm.

A. Object Model Extraction

A region-based model of the objects is extracted in this step. For every measured object, the normalized RGB color histogram is determined to uniquely identify an object. The histogram of an object was calculated by counting the number of pixels of the mask image within the rectangle that borders the object.

B. Position Prediction

In this step, the position of each tracked object on the plane is predicted by a Kalman filter. By using a homography the position measurement of each object is obtained. It is assumed that the objects are orthogonal to the plane and the lower points of the objects are touching the plane. Thus, the midpoint of the lower rectangle edge is chosen as the position and is projected onto the plane by the homography.

For the Kalman filter, a constant speed model is used. Thus, it is assumed that the accelerations of all objects are approximately zero except for noise to allow for non-constant object velocities. Each tracked object is modeled by one Kalman filter. The positions are also superimposed with noise since initially, the object velocities can not be estimated correctly due to absence of experience.

C. A Priori Assignment

In this step, the measured objects are *a priori* assigned to any existing tracks. Let $\hat{\mathbf{T}}_t^{1^-}, \hat{\mathbf{T}}_t^{2^-}, \ldots, \hat{\mathbf{T}}_t^{m^-}$ denote the predicted positions of tracked objects and $\mathbf{ME}_t^1, \mathbf{ME}_t^2, \ldots, \mathbf{ME}_t^n$ denote the positions of the measured objects on the plane at time step t. Then, the distance matrix $\mathbf{D}_t$ is computed based on the Euclidean norm as follows:

$$\mathbf{D}_t(i,j) = \|\hat{\mathbf{T}}_t^{i^-} - \mathbf{ME}_t^j\| < T_d, \tag{4}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. It stores the distances between the predicted positions of the tracked objects and the positions of the measured objects. The rows of the distance matrix correspond to the existing tracks and the columns to the measured objects. If the distance is above threshold T_d, the element in the matrix will be set to infinity. The threshold T_d is determined experimentally. Based on analyzing the distance matrix, a decision matrix $\mathbf{J}_t$ at time step t is constructed. The number of rows and columns are the same number as in the distance matrix and all elements are set to 0. For each row in $\mathbf{D}_t$, find the lowest valued cell and increment the corresponding cell in $\mathbf{J}_t$. The same is done for the columns. Thus each cell in $\mathbf{J}_t$ has a value between zero and two.

Only if an element value of the decision matrix $\mathbf{J}_t$ is equal to two, the measured object is assigned to the tracked object and their correspondence is stored. All elements in the same row and column of the distance matrix $\mathbf{D}_t$ are updated to infinity and a new decision matrix $\mathbf{J}_t$ is constructed. This process is repeated until none of the elements in the decision matrix equals to two. The correspondence between the objects is calculated by the Bhattacharya distance:

$$BD(HT, HM) = \sum_{i=1}^{N_r \cdot N_g \cdot N_b} \sqrt{HT(i) \cdot HM(i)} > T_{co} \tag{5}$$

where HT is the color histogram of the tracked object and HM is the measured object with $N_r \cdot N_g \cdot N_b$ bins. The values $HT(i)$ and $HM(i)$ are the normalized frequencies of the bin i. If the Bhattacharya distance of the object histograms is below the correspondence threshold T_{co}, a correspondence between the objects is not given. The threshold is 1 for a correspondence and 0 for a non-correspondence.

After the *a priori* assignment the tracked and measured objects can be classified into the following three categories:

- matched tracked and measured objects,
- unmatched tracked objects and
- unmatched measured objects.

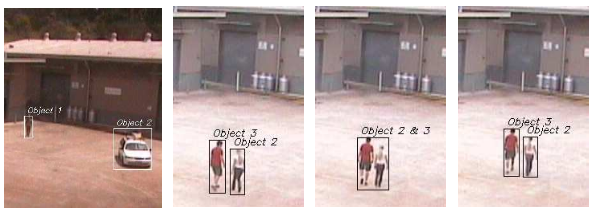
This step can not handle merging and splitting events, in which one measured object may be assigned to multiple tracks and one track may be assigned to multiple measured objects. A merging and splitting algorithm was developed to solve this problem.

D. Merging and Splitting

In this step, merging and splitting events are handled. Here, it is a valid assumption that as soon as objects touch each other, a large rectangle containing all objects is generated. Thus, the objects are not occluding each other at that time step. For tracked objects that are not matched to the measured objects, a merging detection algorithm is used to decide whether the track is merged with another track or it remains unmatched. If the track remains unmatched, its age increases until the object is assumed to be lost and therefore no longer significant. For unmatched measured objects, a splitting detection algorithm is developed. It decides whether the measured object is split from a tracked object or it is a new track.

E. Experimental Results

Three different scenes are chosen to represent the tracking algorithm. The first two scenes are demonstrated in Figure 4. A moving car and a walking person is shown in the leftmost figure. In the right three subfigures, two people merge and split. After the splitting, the individuals were identified correctly.



(a) (b) Before the merging. (c) After the merging. (d) After the splitting.

Fig. 4. Multiple object tracking (left). Merging and splitting of two people in a scene (right).

In figure 5, the third scene is demonstrated. In this scene, two people cross each other. During the crossing, one person occludes the other person. The persons are identified correctly after crossing. Note that complete occlusion of objects via other moving objects is handled correctly.

(a) Before the crossing. (b) During the crossing. (c) Occlusion. (d) After the crossing.

Fig. 5. Crossing of two people in a scene.

VI. OBJECT CLASSIFICATION

The goal is to classify each moving object visible in the input images as a single person, group or vehicle. Our approach to classify people and vehicles is based on [1]. The algorithm requires an appearance history of the object from the tracking algorithm by means of a bounding box (smallest possible rectangle bordering the mask of the object) and correspondence of each object over the frames. In most cases, the whole object is moving along with local changes in shape (mask of the object). Thus, the objects are classified by detecting repetitive *changes* in their shapes. In Figure 6, the flow diagram of the classification algorithm is presented.

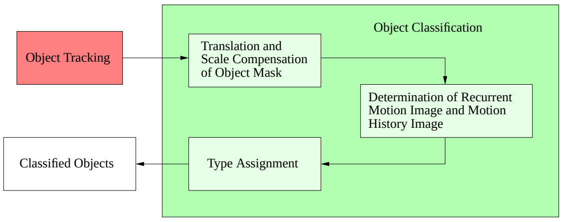


Fig. 6. The flow diagram of the classification algorithm.

These steps are explained in the following sections where an object mask is defined as the part of the mask image within the bounding box of the object.

A. Translation and Scale Compensation of Object Masks

A moving object often changes its position within the bounding box and its size. To eliminate effects of mask changes that are not due to shape changes, the translation and change in scale of the object mask over time needs to be compensated. The assumption is that the only reason for changes in the shape size is the variation of the object distance from the camera. The translation is compensated by aligning the objects in the images along its centroid. For compensation of scale, the object mask is scaled in horizontal and vertical directions such that its bounding box width and height are the same as of the first observation.

B. Determination of Recurrent Motion Image and Motion History Image

Let $A_t^i(x,y)$, for $i = 1,\ldots,m$, be the pixel value of the translation and scale compensated object mask i at position (x,y) and at time t. Then, a difference image $D_t^i(x,y)$ is generated for each object $i = 1,\ldots,m$ by using the exclusive-or operator $\oplus$ as follows:

$$D_t^i(x,y) = A_{t-1}^i(x,y) \oplus A_t^i(x,y). \quad (6)$$

The value $D_t^i(x,y)$ indicates the shape changes of the object. After this step, the Recurrent Motion Image (RMI) is calculated as follows:

$$RMI_t^i(x,y) = \frac{\sum_{k=0}^{\tau} D_{t-k}^i(x,y)}{\tau} \quad (7)$$

where τ is the time interval that should be large enough to capture the recurrent shape changes. The recurrent motion image has high values at those pixels whose shape changes repeatedly and low values at pixels where there are little shape changes or no shape changes at all.

Our classification algorithm is based on the work of Javed and Shah [1]. However, we found that it did not always correctly classify objects that change shape through turning. Henceforth, we enhanced their algorithm to increase robustness by providing a second metric for analysing motion - termed a 'Motion History Image'.

The Motion History Image (MHI) is a mask image that indicates where motion of the object occurred during the time interval τ. It is calculated as follows:

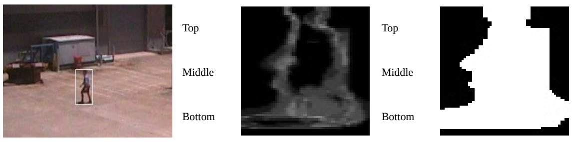
$$MHI_t^i(x,y) = \begin{cases} 0 & \text{if } \sum_{k=0}^{\tau} A_{t-k}^i(x,y) = 0 \\ MHI_{max} & \text{otherwise} \end{cases} \quad (8)$$

where MHI_{max} is the maximum value of the MHI.

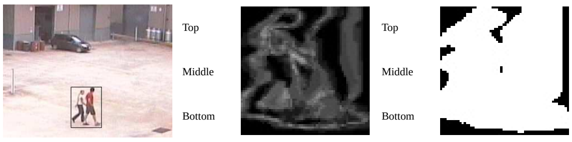
C. Type Assignment

Once the recurrent motion and the MHI of the object is obtained, the type of the object needs to be classified. Therefore, the recurrent motion is divided into $o \times o$ equal sized square blocks and the mean value for each block is computed. The partitioning reduces the computation and the averaging reduces noise. Then, the corresponding MHI is computed by scaling it to an $o \times o$ image. In Figure 7, examples of averaged recurrent motion and scaled MHI are shown in three different scenes. As it can be seen, the ratio of recurrent motion to motion occurrence of the single person and the group in the bottom of the images is bigger than that of the van, because a van has no repeated changes in its shape.

The type assignment is also different to [1]. A Repeated Motion Ratio is introduced to distinguish between people and vehicles. The sum S_t^i of all mean values of the blocks in the bottom of the recurrent motion image at which the corresponding blocks of the MHI has its maximum value (motion has occurred) is determined for the objects $i = 1,\ldots,m$ at time t. During this step, the number of the blocks $o_{p,t}^i$ in the bottom of the MHI with maximum value is counted. In the next step, the Repeated Motion Ratio is calculated by dividing the sum S_t^i by the number of blocks $o_{p,t}^i$ times the maximal value RMI_{max} of the recurrent motion image. The Repeated Motion Ratio is 1, if the recurrent motion image has its maximum mean value in every block at which the corresponding MHI indicates motion. That is, if the shape of the object changes repeatedly. If the recurrent motion image has its minimum mean value 0 in every block, the Repeated Motion Ratio is 0 as well which means that the shape of the object does not change repeatedly. Thus, the



(a) Single person walking. (b) Averaged RMI. (c) Scaled MHI.



(d) Two people walking. (e) Averaged RMI. (f) Scaled MHI.

(g) A moving van. (h) Averaged RMI. (i) Scaled MHI.

Fig. 7. Examples of RMIs and MHIs in different scenes.

object type single person or group is assigned to the object, if

$$RMR_t^i = \frac{S_t^i}{o_{p,t}^i \cdot RMI_{max}} > T_{ot} \tag{9}$$

where T_{ot} is the fixed decision threshold of the object type. If RMR is below that threshold, the object is classified as a vehicle. The threshold T_{ot} is determined experimentally.

D. Classification Results

The classification algorithm was applied to a variety of video sequences. They contain people walking and vehicles moving. Each sequence consists of 600 to 1000 frames. The tracking algorithm provides the bounding box and correspondence of each object over the images of each sequence. The classification algorithm was applied for each object after it has completely entered the image. The number of frames over which the recurrent motion and the motion history image were calculated is $\tau = 20$. Thus, a wrong data association do not have quite an impact on the recurrent motion and the motion history image The decision threshold of the object type is $T_{ot} = 0.12$. In Table I, the results of the classification algorithm distinguishing between people and vehicles are given. Even in presence of noisy mask images accurate classifications were obtained.

TABLE I

RESULTS OF THE OBJECT CLASSIFICATION ALGORITHM.

TYPE OF OBJECT	Classified as People	Classified as Vehicle
Single People	38	0
Vehicle	1	20

VII. ONLINE APPLICATION OF THE CLASSIFICATION SYSTEM

The classification system was applied online. The input image stream is handled by the DDX framework (Dynamic Data eXchange) developed by Corke *et al*. [14]. To acquire video live streams and controlling a camera the DDXVideo framework is used [15].

Three representative scenarios were chosen. In the first, a moving car enters the scene, stops, and a person egresses and both leave. Two people crossing each other are displayed in the second. During the crossing, one person occludes the other. In the third scenario, two people merge and split. The people occlude each other repeatedly when they are merged. The results are shown in Figures 8 to 10.

Fig. 8. First scene: Person and car.

In all tests, the objects are correctly tracked and identified. Further tests have shown that the classification system can achieve frame rates $33-50 fps$.

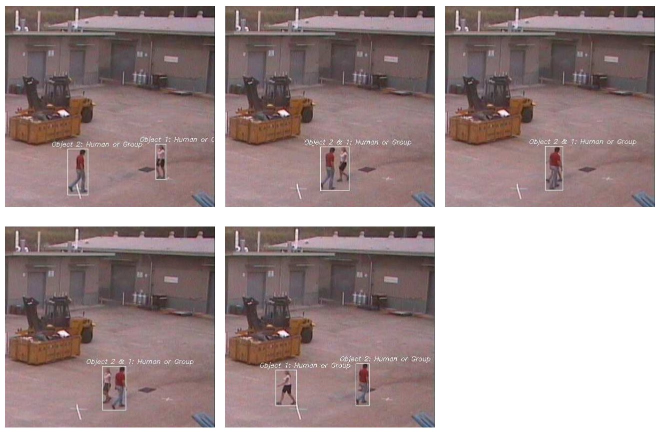


Fig. 9. Second scene: Two people cross each other.

We have also tested the algorithm on various vehicle types and in different types of weather. Figure 11 below show samples of a forklift in sunlight, and a bicycle rider in the rain - both mounted and unmounted. The bicycle rider case is interesting since the recurrent motion has a higher vertical component than in walking cases. The classifier gave the correct predictions in all cases.

VIII. CONCLUSIONS AND FUTURE WORK

We have demonstrated a vision based system for tracking and classifying dynamic objects in an outdoor environment. The system is based on [1] and shows improvements in the detection and classification of people and vehicles. The system can handle occlusions and has demonstrated good results over multiple objects in varying weather conditions. In each test case, the system accurately labeled the dynamic

(a) Before the merging. (b) After the merging. (c) During occlusion 1. (d) After occlusion 1.

(e) During occlusion 2. (f) After occlusion 2. (g) After the splitting.

Fig. 10. Third scene: Two people merge, occlude each other repeatedly and split.

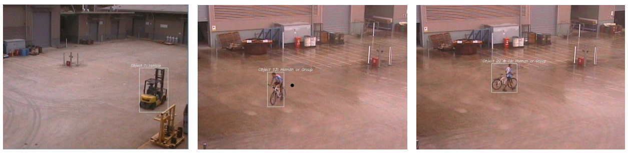


Fig. 11. Various types of dynamic objects have been used for testing the system in different weather conditions.

objects and tracked them correctly. The system works in real time and achieves a frame rate of $33-50fps$ for 768×576 resolution color images on a 3.2 GHz Pentium D computer. Our approach differs from existing approaches in that multiple objects are reliably tracked, even presence of occlusions, and the combination of using recurrent motion and Motion History Images improves classification and tracking performance.

The system is a preliminary step towards improving the situational awareness of either human-operated or autonomous vehicles working in joint workspaces. Being more aware of the environment makes operations safer and improves efficiency since better local path planning can result from knowing where potential path conflicts will occur and anticipatory steps taken to avoid them.

Within this work a basis of classification system was created. It is very efficient in terms of computational and space requirements. The next step is to develop a cast shadow algorithm in the motion segmentation step to create a good prerequisite for object tracking and classification under all lighting conditions. During the course of this research, several cast shadow algorithms were tested [8], [16] but none were robust or reliable enough in our test environment.

The object classifier of the system is also a basis for investigating further improvements. For example, a classifier could be developed that distinguishes between the different types of vehicles like cars, vans, trucks etc. or between single persons and groups. Furthermore the system could be optimized in its implementation to improve its speed. Introducing multiple camera viewing the scene in different angles would improve the object tracking and classification performance and robustness of the system.

REFERENCES

[1] O. Javed and M. Shah, *Computer Vision - ECCV 2002*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2002, vol. 2353/2002, ch. Tracking And Object Classification For Automated Surveillance, pp. 439–443.

[2] L. Zhang, S. Z. Li, X. Yuan, and S. Xiang, “Real-time Object Classification in Video Surveillance Based on Appearance Learning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[3] C. Stauffer and W. E. L. Grimson, “Learning Patterns of Activity Using Real-Time Tracking,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, pp. 747–757.

[4] T. Yang, S. Z. Li, Q. Pan, and J. Li, “Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes,” in *Proceedings of the 2005 IEEE Computer Society on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 970–975.

[5] R. Cuccharina, C. Grana, M. Piccardi, and A. Prati, “Detecting Moving Objects, Ghosts, and Shadows in Video Streams,” in *Transactions on Pattern Analysis and Machine Intellegence*, 2003.

[6] N. McFarlane and C. Schofield, “Segmentation and Tracking of Piglets in Images,” in *Machine Vision and Applications*, vol. 8, 1995, pp. 187–193.

[7] M. Israd and A. Blake, “CONDENSATION - Conditional Density Propagation for Visual Tracking,” in *Int. J. Computer Vision*, 1998, pp. 5–28.

[8] H.-T. Chen, H.-H. Lin, and T.-L. Liu, “Multi-Object Tracking Using Dynamical Graph Matching,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 210–217.

[9] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle, “Appearance models for occlusion handling,” in *Proceedings 2nd IEEE Int. Workshop on PETS*, 2001.

[10] D. Toth and T. Aach, “Detection and Recognition of Moving Objects Using Statistical Motion Detection and Fourier Descriptors,” in *Proceedings of the 12th International Conference on Image Analysis and Processing*, 2003, pp. 430–435.

[11] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lepchev, “A Real-Time System for Classification of Moving Objects,” in *16th International Conference on Pattern Recognition*, vol. 3, 2002.

[12] C. Stauffer and W. E. L. Grimson, “Adaptive Background Mixture Models for Real-time Tracking,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[13] S. Gupte, O. Masoud, and N. P. Papanikolopoulos, “Vision-Based Vehicle Classification,” in *Proceedings 2000 IEEE Intelligent Transportation Systems*, 2000, pp. 46–51.

[14] P. Corke, P. Sikka, J. Roberts, and E. Duff, “DDX: A Distributed Software Architecture for Robotic Systems,” in *Proceedings of the Australian Conference on Robotics and Automation*, 2004.

[15] E. Duff, “DDXVideo: A Lightwight Video Framework for Autonomous Robotic Platforms,” in *Proceedings of the Australian Conference on Robotics and Automation*, 2005.

[16] J. C. S. J. Jr and C. Jung, “Background Subtraction and Shadow Detection in Grayscale Video Sequences,” in *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*, 2005.