ROBUST PERSON TRACKING WITH MULTIPLE NON-OVERLAPPING CAMERAS IN AN OUTDOOR ENVIRONMENT

S. Hellwig, N. Treutner

Humboldt-Universitaet zu Berlin
Institut fuer Informatik
Unter den Linden 6
10099 Berlin, Germany
hellwig@informatik.hu-berlin.de, treutner@informatik.hu-berlin.de
http://www.informatik.hu-berlin.de/sv

Commission V, WG V/5

KEY WORDS: Tracking, Camera, Multisensor, Video, Feature, On-line

ABSTRACT:

The aim of our work is to combine multiple cameras for a robust tracking of persons in an outdoor environment. Although surveillance is a well established field, many algorithms apply various constraints like overlapping fields of view or precise calibration of the cameras to improve results. An application of these developed systems in a realistic outdoor environment is often difficult. Our aim is to be widely independent from the camera setup and the observed scene, in order to use existing cameras. Thereby our algorithm needs to be capable to work with both overlapping and non-overlapping fields of views. We propose an algorithm that allows flexible combination of different static cameras with varying properties. Another requirement of a practical application is that the algorithm is able to work online. Our system is able to process the data during runtime and to provide results immediately. In addition to seeking flexibility in the camera setup, we present a specific approach that combines state of the art algorithms in order to be robust to environment influences. We present results that indicate a good performance of our introduced algorithm in different scenarios. We show its robustness to different types of image artifacts. In addition we demonstrate that our algorithm is able to match persons between cameras in a non-overlapping scenario.

1 INTRODUCTION

1.1 Motivation

Object tracking with cameras has been an actively researched topic in the past decades. As algorithms and hardware improved, so did performance and quality of the results. Today object tracking is applied in different kinds of applications, like security and traffic assessment.
In these scenarios automatic tracking and evaluation of pedestrian movements can be used in order to recognize and hopefully avoid dangerous situations or even disasters. As the number of used cameras usually by far exceeds the capacity of available personnel, automatic evaluation of video streams is required in order to enable a quick reaction to a possible threat. Especially large camera networks (e.g. on airports or during big sport events) make manual assessment of the data difficult. Results from evaluation algorithms can be used to direct the attention of the staff to peculiar situations.

In this paper we describe a new approach to find, track and evaluate moving objects and pedestrians. In order to cover a complex or wide area, we handle multiple video streams and combine the acquired data. One of our aims is to use existing cameras. Therefore, our algorithm is robust to the setup of the cameras and is able to work with both overlapping and non-overlapping field of views (FOV) at the same time. Furthermore, we choose to allow only fixed camera positions in order to reduce complexity, while not sacrificing much practicability.
In order to be robust to occurring problems like occlusions we try not to rely purely on a single approach. Thus, if a situation arises, which is difficult for an algorithm to process, the alternative algorithms still allow us to compute reliable trajectories.
We aim to detect patterns in the movement of the persons. These patterns also allow us to detect any unusual behavior that might indicate a dangerous situation and therefore requires special attention. Examples of such behavior would be a sudden change of preferred routes, which might indicate a blocked path or a panic. In order to react appropriately the results need to be available during runtime. We therefore chose to process the data ”online”, as opposed to analyze the data afterwards.

1.2 Related Work

The tracking of persons between multiple cameras requires a tracking within the single cameras. In many publications with static cameras a background estimation algorithm (Stauffer and Grimson, 1999) (Javed et al., 2002) (Elgammal et al., 2002) or an estimation of the optical flow (Lucas et al., 1981) is used as a basis for the segmentation of important objects in the images. Another approach for the segmentation is the use of features like SURF (Bay et al., 2006) or SIFT (Lowe, 1999) in combination with a cluster algorithm. Common approaches towards single-camera tracking also often include the use of histograms. These results can be enhanced by employing multiple histograms (Exner et al., 2009) or filters that predict the position (Wang et al., 2009) or appearance (Peng et al., 2005) of the object. Many approaches assume that every moving object is a person. Approaches, which distinguish between persons and other moving objects often use person detectors like the HOG person detector (Dalal and Triggs, 2005).
Our approach combines several of these algorithms in order to enhance the overall robustness of the system.

Publications in multi-camera tracking often differ in either overlapping (Du and Piater, 2007) (Khan and Shah, 2006) or non-overlapping (Javed et al., 2008) field of views. Only a few approaches are able to handle both kinds of camera setups (Javed and Shah, 2003) (Nam et al., 2007). Many of the overlapping approaches transform the data of all cameras into a reference coordinate system. In this context stereo systems are also common, while most non-overlapping approaches do not use a reference coordinate system.
In order to establish the relationship between the objects in the different cameras other techniques like space-time probabilities (Javed et al., 2008), travel-transition time models (Nam et al., 2007) or FOV lines (Javed and Shah, 2003) are used. Under certain assumptions it is also possible to compute spatial relations of the cameras to each other (Rahimi et al., 2004).
Our approach combines the use of overlapping and non-overlapping FOVs and the transformation of all information into a reference coordinate system.

2 SYSTEM DESGIN

In the design of our tracking algorithm (Figure 1) we consider the use of multiple cameras with both overlapping and non-overlapping FOVs at the same time. Therefore, we process each camera stream independently and pass the person information to the "area tracker". The area tracker allows tracking of persons not only in a single camera domain, but in the whole area that is observed by connected cameras. It receives the tracking information from all the single-camera trackers and merges the data into a reference coordinate system. The area tracker is able to work with both overlapping and non-overlapping FOVs (Section 4).
Due to the fact that no image data is passed between any of the single camera trackers it is possible to separate them onto different machines and only transmit required data to the area tracker.

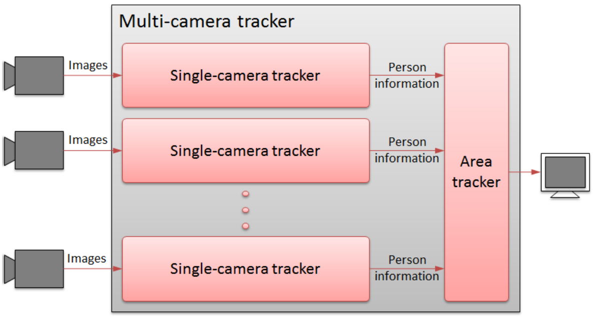

Figure 1: General design of our tracking system.

3 SINGLE-CAMERA TRACKING

The single-camera tracker (Figure 2) receives a video stream from one camera and processes it into tracking data. All information is processed in the image coordinate system. We implemented all algorithms in separate threads in order to achieve multitasking wherever possible. In the following sections we will describe the different processing steps of the single-camera tracker.

Normalization The first step of the single-camera tracker is the normalization of the images, in order to remove the lens distortion. We use the computation suggested by Brown (Brown, 1971).

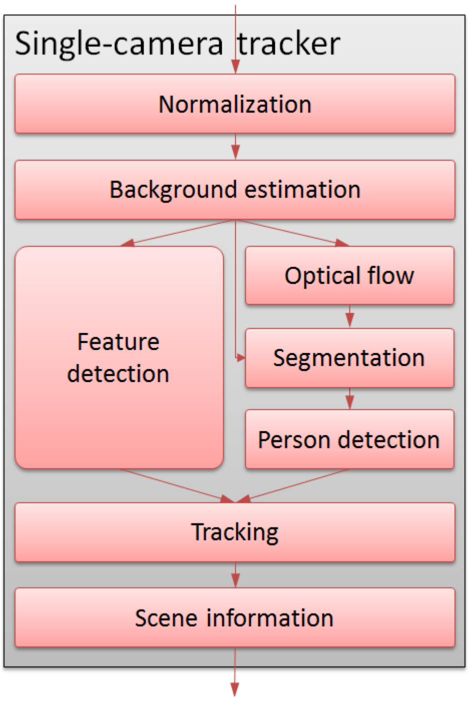

Figure 2: Design of our single-camera tracker.

Background Estimation Since we made the restriction that the cameras are stationary, we are able to confine most of the computations to moving elements in the image. This allows the use of a background estimator. For outdoor use it is necessary that the background estimator is able to compensate changes of lighting and shadows of moving persons. Changes in the background are common in our test samples and therefore must be handled correctly. We chose to implement an algorithm that uses histograms to estimate the background image of each camera.

Optical Flow In order to further enhance our knowledge of the moving objects, we calculate the optical flow. We use the Farnebaeck algorithm (Farnebaeck, 2002) that computes a dense optical flow for the entire image.

Segmentation For the segmentation of the moving objects we combine the results of the background estimator and the optical flow in order to generate more precise contours. The reason for that is that the background estimator reduces the shadows and the use of the optical flow allows objects that encounter each other at different speeds to be separated. Each segment's location, movement, shape, and color histogram are calculated.

Feature Detection While the segments generated in the previous steps already allow us to track many objects in various scenes, we extend our system by also including features into the tracking algorithm. Among the most common feature descriptors are the SIFT (Lowe, 1999) and SURF descriptors (Bay et al., 2006). They provide good results, but computation and comparison of calculated features and descriptors can take a long time. Even SURF, which is a sped up alternative to SIFT, makes application in a real time framework difficult. Therefore, we chose to use the newly introduced BRIEF Features (Calonder et al., 2010). They are faster to compute and compare, and provide good results.

Basic Tracking Algorithm In this step, the previously generated segments are compared to existing objects, which were created in previous frames. In addition the computed features and their descriptors are compared to previously tracked features. All available properties of the object (spatial information, appearance, and features) are used to compute a distance. When calculating the spatial distance, we estimate the position of the previously detected object in the current frame by using a Kalman filter. The Kalman filter not only allows us to predict, but also to smooth the trajectory.

Using the computed distance we can estimate how likely the detected segments represent the already tracked object. All likely matches are combined and their position, appearance, and features updated. Additionally we check the features in every object for both inconsistent movements between frames and duplicate features that probably represent the same area.

Classification and Person Detection Although the described system can handle all kinds of moving objects, we focus on persons and therefore implemented a method to differentiate between persons and other objects, like cars or dogs. We used a person detector, which utilizes Histogram of Gradients and Support Vector Machines (Dalal and Triggs, 2005) to detect persons in the previously generated segments (Section 3).
By including this information, we can differentiate single persons in a group that would otherwise have been treated as a single big object.

Handling of Occlusions Our algorithm uses different approaches to tackle partial and complete occlusions. In case of a partial occlusion no segment in the current frame will match with the previously detected objects in the basic tracking step. But some of the object's saved features will match with the features of the visible part of the object. Here we compute the average movement of these features and use this movement to adjust the position of the object. Therefore we can track partly occluded objects, as long as some of the object features match in this frame. Especially in crowded scenes, where segmentation of single persons is difficult, the tracking in subsequent frames can benefit from the use of features.

In the case of a complete occlusion another approach is needed (Figure 3). First of all, we use Kalman filters to predict the position of moving objects. If a person continues its movement, it is often correctly matched after it reappears. But unlike cars, persons are highly mobile and can change their directions and speed very fast. Therefore, persons can appear at various locations after disappearing behind an obstacle, and so matching it with the help of the predicted position is not possible. Additionally, objects often reappear partly occluded and a matching with the original object fails. Instead of matching it to the previously tracked object, a new object is created and tracked subsequently. As the object fully emerges from the occlusion, the object's appearance eventually looks like the previously tracked object. This leads to the situation that one real object can have two representations in the tracker. This error can be detected by computing a distance between the two objects using all properties of the objects (spatial information, appearance, and features). If it is plausible that the two objects in the tracker represent the same real object, the objects in the tracker are fused and their information is combined.

Scene Information During the tracking of objects and persons, we also store information about the observed scene in order to improve tracking results in subsequent frames. These properties include the common direction and speed of objects at all positions as well as entry and exit zones.
The gathered scene information is used in various parts of the tracking algorithm. The direction and speed can be used in the motion model of the Kalman filters to support the prediction of the position. The entry and exit zones can help to support the matching in case of occlusions.
These patterns also allow us to detect any unusual behavior that might indicate a dangerous situation and requires special attention. Examples of such behavior would be a sudden change of preferred routes, which might indicate a panic or a blocked path or doorway.

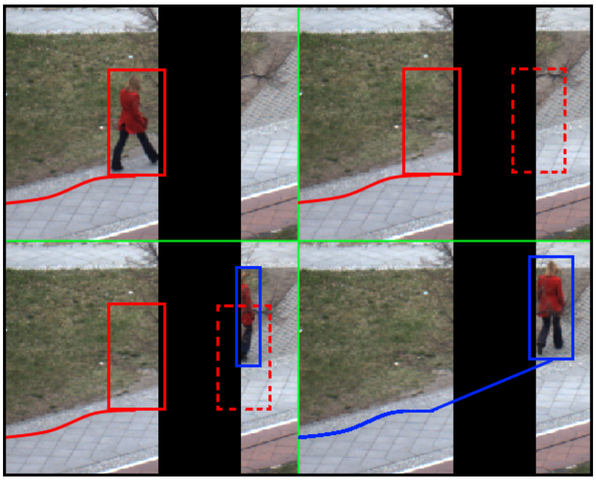

Figure 3: Top left: Tracked person (red) approaching occlusion. Top right: Person hidden by occlusion, tracker searches for person at the predicted position (red dashed). Bottom left: Person emerges partially at an unpredicted position. A new tracked person is created (blue). Person emerges fully, is matched with both red and blue tracked persons. The data is fused and stored for the next frames.

An example of generated entry zones can be seen in figure 4. The zones were generated over 22000 frames with 166 used persons. Among some smaller erroneous entry zones along the person's path, larger and intense entry zones at the border of the image can be seen. Additionally, a few obstacles (e.g. trees) are surrounded by entry zones, as the occlusions can lead to temporarily lost tracking.
The corresponding exit zones are similar to the entry zones. Both passing behind an occlusion and entering and leaving the field of view produces both entry- and exit points.

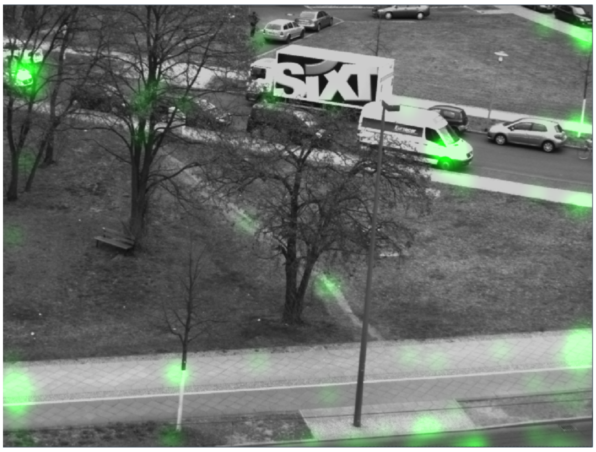

Figure 4: Example of generated entry zones within 22000 frames with 166 used persons.

4 AREA TRACKER

The area tracker receives and merges the tracking data from all single-camera trackers. For this, all the data in image coordinates is transformed into the reference coordinate system. This transformed data is used to create an own representation of all observed persons in all cameras in the reference coordinate system.
In the following sections the processing steps of the area tracker are described.

Transformation into the Reference Coordinate System For the transformation of the camera coordinates into the reference coordinate system we assume that all objects move on a ground plane. The transformation is done by a projective transformation, for which a 3x3 projection matrix is needed for each camera. It can be shown that the determination of this projection matrix needs only four point correspondences. However, in this case we want to establish the projection matrix by possibly more than four correspondences, resulting in a least-squares best fit projection. The transformation between the two coordinate systems can be written as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x'/z \\ y'/z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z \end{pmatrix} = P * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

The point (u,v) is the point on the assumed ground plane. By applying the direct linear transformation (DLT), the matrix P can be stacked as a column vector. Each homologous point pair contributes with two equations to the linear system, which determines P. Let (u,v) and (x,y) be such a point pair. Then the nine unknown entries of P can be computed from:

$$H = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & u*x & u*y & u \\ 0 & 0 & 0 & -x & -y & -1 & v*x & v*y & v \end{pmatrix} \tag{2}$$

If we now use n correspondences (n $>$ 4), we obtain an over determined equation system on the entries of P with 2*n equations. This can be solved in least squares sense by using the singular value decomposition (SVD):

$$U * \Sigma * V^T = H \tag{3}$$

U and V are orthonormal matrices of dimensions 2*n and 9, respectively. Σ contains the sorted singular values as diagonal matrix. The last column vector of V is the basis transformation vector for the smallest singular value. It contains the entries for the best projection matrix P in least squares sense.

Area Tracking Algorithm The area tracker handles the representation of persons in a similar to the single-camera tracker (Section 3). The person properties include a color histogram, a trajectory, features, and a few other appearance properties.
The most important difference between the area tracking and the single-camera tracking is that we receive no segment information from the actual image. Instead we process the person information stored in the single-camera trackers. Every time one of the single-camera trackers processes a frame, the persons in this tracker are transformed into the reference coordinate system (Section 4) and matched with the existing representations in the area tracker. For this matching all available properties of the persons are used to compute a distance.
This approach already allows us to track a person through multiple cameras if their FOVs overlap. In order to handle the problem of the non-overlapping cameras both a Kalman filter and a fuse algorithm, similar to the one in the single-camera tracker (Section 3), are used.

Area Information Similar to the scene information (Section 3) we compute various values, which characterize the area in the reference coordinate system. The computed properties include the common speed and direction of the persons at the different positions as well as entry and exit zones. For the calculation of the entry and exit zones we transform the scene information from the single-camera trackers into our reference coordinate system. Additionally we calculate probabilities between the entry and exit zones to establish common links between them. The entry and exit zones are used to help solving problems like occlusion or prediction of positions. In the reference coordinate system these zones are used to support the matching between non-overlapping cameras.

5 RESULTS AND DISCUSSION

The main focus of our approach is to provide robust and reliable tracking of persons in an outdoor environment. To show the robustness of our tracking system we tested it in two different scenarios. The first scenario shows the robustness in the single-camera domain, by using intentionally impaired input data. The second scenario focuses on the transition of persons between two cameras and shows the robustness of our system to gaps between two non-overlapping FOVs.

5.1 Results: Single-Camera Tracking

The first scenario shows the robustness of our system to difficult situations like occlusion and image artifacts. For an analysis of the tracking performance we selected scenes from a camera that include both single persons and groups. The camera provided a resolution of 600x400 at 7.5 fps. The observed persons in the images are 50 to 100 pixels tall.
An occlusion was added manually to all images in order to create a more difficult challenge for the different tests. This test is labeled ”normal”. We then compared these tracking results with results that were obtained after we added noise, blur, and even more occlusions to the images. An example image of four different test can be seen in figure 5. The ”noise” test had a 0.1% salt and pepper noise on each of the three color channels. The ”blur” test had a gaussian noise added with a neighborhood of 5 pixels and a standard deviation of 2.5. In the ”more occlusions” test two more occlusions are added, each of them partly occluding a frequently used path.

To compare the results of our system in the four tests we analyzed the trajectories of 35 persons and determined the rate of persons, which were successfully tracked as ”completed”. Some persons were tracked on both sides of the occlusion but could not be joined, these were counted as ”disconnected”. Persons that were not tracked at all, or generated only interrupted tracks were enumerated as ”failed”.

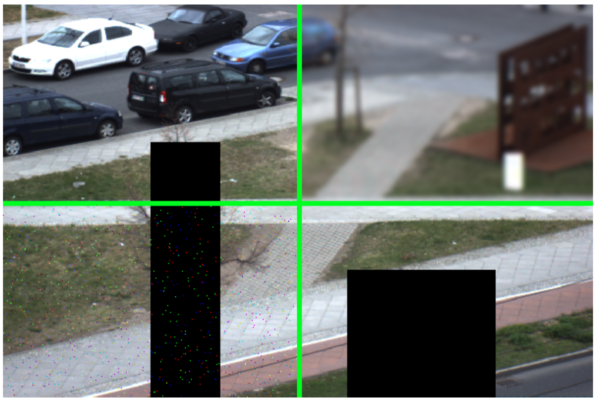

Figure 5: Sample images from four tests: upper left - ”normal”, upper right - ”blur”, lower left - ”noise”, lower right - ”more occlusion”. For tracking results see Table 1

The results (Table 1) indicate good tracking, even under difficult circumstances. Most of the ”disconnected” and ”failed” persons can be explained by frequent mutual occlusions between the persons while they where walking in a group.
We found that although some parts of our system could not cope with specific alterations, the final outcome was still good. E.g. in the ”blur” test much less features were detected, because a blurred image lacks distinctive edges and corners and thereby makes keypoint detection difficult. While the lack of features degrades the tracking close to the occlusion, the remaining algorithms still work properly.

	completed	disconnected	failed
normal	74%	17%	9%
noise	58%	33%	9%
blur	77%	17%	6%
more occlusion	69%	22%	9%

Table 1: Results of the single-camera tracking scenario. 35 persons were analyzed. ”Completed” are persons that were tracked correctly. ”Disconnected” indicates that the track was predominantly correct, but was lost at one point and a new representation of the person was created. ”Failed” represents the number of persons that could not be tracked correctly.

5.2 Results: Multi-Camera Tracking

The second scenario shows the robustness of our system to non-overlapping FOVs. Our test scenario consists of two cameras with a resolution of 600x400 at 7.5 fps. The observed persons in right camera have a height of 50 to 100 pixels and 25 to 60 pixels in the left camera. The gap between the FOVs is approximately 7 meters at the front path and 8 meters at the rear path. The persons need approximately 7 seconds to pass the gap. Figure 6 shows an example of a tracking of persons in the reference coordinate system.

When evaluating a scene with 33 persons passing between the two cameras, our tracking algorithm managed to correctly track 73% of all persons. Most of the errors occur because of mutual occlusion of the persons walking in groups. Thus, it is often difficult to track the person in the single-camera tracker, and so a tracking between two cameras is not possible. Of all persons that were tracked properly in both cameras, 83% could be tracked across both cameras.

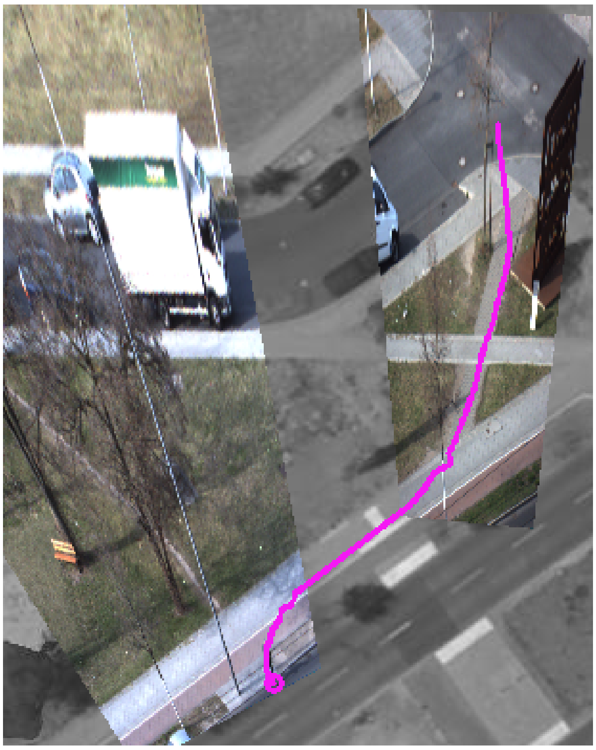

Figure 6: Example of a tracking in the reference coordinate system. Both camera images are transformed onto the ground plane. The ground plane is indicated with an aerial image of the area.

6 CONCLUSIONS AND FUTURE WORK

This paper contributes a new approach to track persons in a multi-camera environment. We aim for a robustness in both overlapping and non-overlapping scenarios. Our approach is also robust to environment conditions through the use of various parallel algorithms like background estimation, optical flow, person detection, and feature descriptors.
The system we described, still depends on external information to transform the data from the image coordinate systems into the reference coordinate system. Thus the next step in order to make application in unknown areas faster and easier is to develop methods to automatically calculate the used transformation matrices.
Furthermore, the scene information and area information can be utilized in even more steps during the tracking, e.g. to deduce sophisticated motion models, including theories of social force in order to predict each person’s path. The detection of unusual events, based on the area information, could be used to generate alerts that draw attention of human staff to peculiar and probably dangerous events.

ACKNOWLEDGEMENTS

We would like to thank our colleague Sascha Stuebing for helping us greatly with designing, programming and evaluating the algorithm for this paper. We would also like to thank Prof. Dr. Beate Meffert for her continued support. The described tracking system is part of the project ZONE (”Charakterisierung des Verhaltens von Personen oder Personenstroemen in abgegrenzten Zonen (ZONE)”), a cooperation of the Humboldt-Universitaet zu Berlin with LUETH & DUEMCHEN Automatisierungsprojekt GmbH. The project is funded as a part of ProFIT by the Investitionsbank Berlin.

We would like to thank the organizers of the XXII Congress of the International Society for Photogrammetry and Remote Sensing.

REFERENCES

Bay, H., Tuytelaars, T. and Van Gool, L., 2006. Surf: Speeded up robust features. Computer Vision–ECCV 2006 pp. 404–417.

Brown, D., 1971. Close-range camera calibration. Photogrammetric engineering 37(8), pp. 855–866.

Calonder, M., Lepetit, V., Strecha, C. and Fua, P., 2010. Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010 pp. 778–792.

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 1, pp. 886–893.

Du, W. and Piater, J., 2007. Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In: Proceedings of the 8th Asian conference on Computer vision-Volume Part I, Springer-Verlag, pp. 365–374.

Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L., 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE 90(7), pp. 1151–1163.

Exner, D., Bruns, E., Kurz, D., Grundhöfer, A. and Bimber, O., 2009. Fast and reliable camshift tracking.

Farnebaeck, G., 2002. Polynomial expansion for orientation and motion estimation. PhD thesis, Universitetet i Linkping Department of Electrical Engineering.

Javed, O. and Shah, M., 2003. Knightm: A multi-camera surveillance system. In: IEEE International Conference on Multimedia and Expo, Citeseer.

Javed, O., Shafique, K. and Shah, M., 2002. A hierarchical approach to robust background subtraction using color and gradient information.

Javed, O., Shah, M., Shafique, K., Rasheed, Z. et al., 2008. Tracking across multiple cameras with disjoint views. US Patent 7,450,735.

Khan, S. and Shah, M., 2006. A multiview approach to tracking people in crowded scenes using a planar homography constraint. Computer Vision–ECCV 2006 pp. 133–146.

Lowe, D., 1999. Object recognition from local scale-invariant features. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Vol. 2, Ieee, pp. 1150–1157.

Lucas, B., Kanade, T. et al., 1981. An iterative image registration technique with an application to stereo vision. In: International joint conference on artificial intelligence, Vol. 3, Citeseer, pp. 674–679.

Nam, Y., Ryu, J., Choi, Y. and Cho, W., 2007. Learning spatio-temporal topology of a multi-camera network by tracking multiple people. World Academy of Science Engineering and Technology 24, pp. 175–180.

Peng, N., Yang, J. and Liu, Z., 2005. Mean shift blob tracking with kernel histogram filtering and hypothesis testing. Pattern Recognition Letters 26(5), pp. 605–614.

Rahimi, A., Dunagan, B. and Darrell, T., 2004. Simultaneous calibration and tracking with a network of non-overlapping sensors. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 1, IEEE, pp. I–187.

Stauffer, C. and Grimson, W., 1999. Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., Vol. 2, IEEE.

Wang, Z., Yang, X., Xu, Y. and Yu, S., 2009. Camshift guided particle filter for visual tracking. Pattern Recognition Letters 30(4), pp. 407–413.