Multi Camera Person Tracking Applying a Graph-Cuts based Foreground Segmentation in a Homography Framework

Dejan Arsić, Atanas Lyutskanov, Gerhard Rigoll
Institute for Man Machine Communication
Technische Universität München
arsic@tum.de

Bogdan Kwolek
Computer and Control Engineering Chair
Rzeszow University of Technology
bkwolek@prz.rzeszow.pl

Abstract

Reliable tracking of objects is an inevitable prerequisite for automated video surveillance systems. As most object detection methods, which are based on machine learning, require adequate data for the application scenario, foreground segmentation is a popular method to find possible regions of interest. These usually require a specific learning phase and adaptation over time. In this work we will present a novel approach based on graph cuts, which outperforms most standard algorithms. It is commonly agreed that occlusions can only be resolved in multi camera environments. Applying multi layer homography will enable us to robustly detect and track objects applying only foreground data, resulting in a high tracking performance.

1. Introduction

One of the major aspects of automated visual surveillance systems is to detect objects in the scene and track these over time. The most challenging problem herein is to segment people in complex scenes, where high object density leads to occlusions. To model individual behaviors these have to be resolved robustly. Tracking techniques based on a single view, such as the mean shift algorithm [1], are able to track objects robustly, but require an initialization of single objects prior to the group formation and the subsequent handling of merge and split events [2]. However, in some cases a single view seems not sufficient to detect and track objects due to severe occlusion, which as a fact requires the utilization of multiple camera views.

Camera networks are frequently applied to extend the limited field of view of a camera, performing tracking in each sensor separately and fusing this information [3]. In order to deal with dense crowds, the cameras should be mounted to view defined regions from different perspectives. Within these, corresponding objects now have to be located. Approaches based on geometrical information rely on geometrical constraints between views using calibrated data [4] or homography between uncalibrated views, which e.g. Khan [5] used to localize feet positions. This approach, though very simple and effective, localizes feet and consequently tends to segment persons into further parts. This can be avoided by applying multi layer homography, as proposed in [6], which is capable to create a 3D representation of the scene. Up to now such tracking systems do not incorporate contextual knowledge about the scene [7], and therefore frequent failures can be observed. As the homography framework relies on the fusion of foreground regions visible in multiple views, it is not capable to detect a person behind a stationary object in case only one camera is observing the object. This problem will be addressed by incorporating prior knowledge into the homography framework, where obstacles will be incorporated into the fusion process.

As it is commonly agreed, the homography framework entirely relies on a stable foreground segmentation and its performance rises and falls with the reliability of the used method. Adaptive methods, such as Gaussian Mixture Models [8], have shown reliable results in indoor scenarios [9] but are frequently failing in outdoor scenes. Furthermore these require an initialization phase, where only the empty background image should be provided. To cope with this problem, we will present a novel foreground segmentation method which is based on so called graph cuts. Although it also requires a short initialization phase, we will show a huge difference in performance compared to GMMs and Eigenbackgrounds. Our approach is based on an initial reference image, which we extract automatically in advance, given images with moving targets. Afterwards we employ both region and pixel cues, which handle the illumination variations. In addition to this, we accommodate online the reference image against the illumination and scene changes.

In this work we will demonstrate the advantages of this new foreground segmentation method and the resulting reliability of a multi layer homography approach. Although finally only Kalman filtering is applied for tracking, we will show a high ID maintenance throughout the sequence. This paper is structured as follows: We will first introduce and evaluate a novel foreground segmentation technique in sec.2.

The extracted foreground is further used by the multi layer homography approach described in sec.3. Thereby we will also focus on the common problem of tracking and false positive handling in the ground plane. After a short evaluation of the presented tracking framework in sec.4, we will conclude our work with a short summary and outlook in sec.5.

2. Utilizing Graph Cuts for Foreground Segmentation

2.1 Relevant approaches

In case of absence of any a priori information about the objects of interest the most widely employed approach to extract moving targets is to use static cameras and a background subtraction technique. The idea of background subtraction is to identify non-stationary or new objects given a reference background model.
Moving or new objects are then detected by taking the difference between the current frame and the reference background model. In the most common paradigm, the background model is not fixed and it should adapt to illumination changes both sudden and gradual, as well as motion changes arising both due to camera jitter and high-frequency background fluctuations (such as tree brunches, sea waves). Typically, background subtraction is utilized in vision systems with static cameras. It is a first step in a sequence of image tasks, making it a critical part of the system. Afterwards, a post-processing is employed in order to enhance the subtraction results. The results of post-processing can be then utilized to get better the segmentation mask. This is achieved through feedback into the background subtraction algorithm in order to facilitate better updating of the model.

Because of many practical applications, a variety of background estimation algorithms have been studied extensively by various research groups. Among many algorithms being in disposal, none of them can really operate 24 hours a day and seven days a week in almost every conditions. Existing algorithms can be classified as either predictive or non-predictive. The predictive methods are based on dynamical models of time series. However, even recently proposed predictive methods cannot cope with multiple modalities [10]. The non-predictive methods neglect the order of the observations and build a probabilistic representation of the distribution of pixel intensities. The method [11] assumes that each image pixel is a realization of random variable with a single Gaussian distribution. Grimson [12] extended this algorithm by using multiple Gaussians and a fast approximate method for model updates. Backgrounds with fast variations might not be easily modeled using just a few Gaussians [13]. Thus a non-parametric kernel estimation technique was employed for building a statistical representation of the background.

In outdoor scenes a background subtraction must be invariant to illumination change that might arise due to sun, moving clouds, or even to moving background, such as waves on a lake. In such conditions spatial gradients or texture features might be utilized for achieving some invariance to illumination [14]. In outdoor environment a codebook model, which originates from the video compression can be used since it employs a collection of different pixel values for each image coordinate. In such a non-parametric model the background is encoded pixel by pixel, where in a learning stage the intensities at each location are clustered into a set of codewords [15].

2.2 Foreground Subtraction

The use of linear combinations of Gaussian functions for modeling the probability of background pixels is the most common approach to background subtraction. In such an approach pixels are analyzed independently from the others and the only observed values are colors [11, 12]. Our approach is based on an initial reference image, which we extract automatically in advance, given images with moving targets. Afterwards we employ both region and pixel cues, which handle the illumination variations. In addition to this, we accommodate on-line the reference image against the illumination and scene changes.

In [16], a running median of the image sequence has been employed in the segmentation process. The segmentation was done through differencing the pixels from the current frame and the reference image. In our approach the median of pixel values is also used to compose the reference images. In our algorithm we do not only compare images pixel by pixel, but additionally utilize region cues that are tolerant against illumination and scene changes. The initial reference images were composed as medians of pixel values at each background location. Medians were calculated using quick sort algorithm. For PETS 2009 datasets the number of images needed to extract foreground free images ranges from 40 to 350 depending on the crowd and motion. In the current implementation we employ pixel intensities in the extraction of the color reference images. The extraction of the initial reference image can be assisted by object masks, which are extracted via one of the simple background subtraction algorithms. In such an approach the object pixels are not considered in the calculation of the median value.

The brightness invariant similarity between a reference template and the image can be obtained via cross covariance. In order to achieve also the insensibility to contrast the normalized cross-correlation can be employed instead. This is achieved by subtracting the mean and dividing by the standard deviation. That means that the cross-correlation of

a reference template $t(x, y)$ with a current subimage $f(x, y)$ can be expressed as follows:

$$NCC = \frac{1}{n-1} \sum_{x,y} \frac{(f(x,y) - \overline{f})(t(x,y) - \overline{t})}{\sigma_f \sigma_t} \quad (1)$$

where n denotes the number of pixels in $t(x, y)$ and $f(x, y)$, $\overline{t}$ and $\overline{f}$ are mean values, whereas σ_t and σ_f stand for standard deviations of $t(x, y)$ and $f(x, y)$, respectively. The normalized cross-correlation was computed very efficiently using the so called integral images. In our approach we utilize the normalized cross-correlation to generate the probability images between the reference images and the current image, see Fig. 1. Such a probability image is employed next in a classifier, which decides if pixel is a background, shadow or foreground. For shadowed pixels the normalized cross-correlation assumes values near to one.

Figure 1: a) Input image. b) Reference image. c) NCC-based probability image between the reference image and the input image.

Gevers [17] employed following color ratios between two image locations $x1$ and $x2$:

$$\frac{C^i_{x1} C^j_{x2}}{C^i_{x2} C^j_{x1}}, \;\; C^i \neq C^j \quad (2)$$

where C stands for a color channel of the RGB color space. Such color ratios are independent of the illumination, a change in viewpoint, and object geometry. Motivated by the discussed approach, we construct an image of color ratios between the reference image and the current image, where the color of each pixel is given by the following equation:

$$\left[\frac{R^c_{x1}}{R^r_{x1}} \;\; \frac{G^c_{x1}}{G^r_{x1}} \;\; \frac{B^c_{x1}}{B^r_{x1}} \right]^T \quad (3)$$

where c and r denote the current and reference image, respectively, whereas R, G, B stand for color components of the RGB color space. In the practical implementation the color of each pixel in such a color ratio image was calculated as follows:

$$\left[\arctan\left(\frac{R^c_{x1}}{R^r_{x1}}\right) \;\; \arctan\left(\frac{G^c_{x1}}{G^r_{x1}}\right) \;\; \arctan\left(\frac{B^c_{x1}}{B^r_{x1}}\right) \right]^T \quad (4)$$

Figure 2 depicts an example image of color ratios. We can observe that for the pixels belonging to the background the color assumes grey values. This happens because the color channels in the RGB color space are highly are correlated. Moreover, the color ratios are far smaller in comparison to ratios between foreground and background. However, as we might observe in the color ratio image there are noisy pixels. The majority of such noisy pixels can be excluded from the image using the probability images of the normalized cross-correlation, which can also be seen in fig. 1c.

Figure 2: Color ratios between reference and current image.

In our algorithm we compute on-line a reference image using the running median. Afterwards, given such an image we compute the difference image. The difference image is then employed in a classifier, which extracts the foreground objects. In the classifier we utilize also the probability image extracted via normalized cross-correlation, as well as color ratios. Optionally, in the final stage we use the graph-cut optimization algorithm [18] in order to fill small holes in the foreground objects.

In graph-based segmentation an image is mapped onto a weighted undirected graph $G =< \mathcal{V}; \mathcal{E} >$, where each pixel is represented as a node $v \in \mathcal{V}$ and each pair of neighboring pixels is connected by an edge $e \in \mathcal{E}$ called an n-link. Two additional terminal nodes, namely the source s and the sink t, stand for the object and the background. Each non-terminal node is connected to s and t through edges called t-links. A cut on the graph splits the nodes into two sets, where one on them is connected to the source s and the second one is connected to the sink t. The cost of a cut is the sum of weights of all the edges at the cut. The energy function undergoing minimization has the following form:

$$E(f) = \sum_{p \in \mathcal{V}} E_p(f_p) + \sum_{(p,q) \in \mathcal{E}} E_{p,q}(f_p, f_q) \quad (5)$$

where $f_p \in \{0, 1\}$ is the segmentation label of pixel p, where 0 and 1 correspond to the background and foreground, respectively. In Eq. 5 the first term is called the regional or data term as it incorporates regional constraints. In particular, it measures how well pixels fit into the object or background models. $E_p(f_p)$ is the penalty for assigning label f_p to pixel p. The more likely f_p is for p, the smaller should be $E_p(f_p)$. In our approach we utilize 4-neighborhood and $E_{p,q}$ assumes value 1 if two pixels are neighbors, and value 0 otherwise. E_p decreases at the exponential rate. For background we used the probabilities

generated by the classifier, whereas for the foreground the outcomes of NCC. Figure 3 depicts some segmentation

Figure 3: a) Input image. b) background probability image. c) Background refined by graph-cut

results. We can observe that owing to the use of graph-cut some separated patches were connected into the objects.

2.3 Evaluation

%	dr	fp	acc
gmm	39.7	0.14	94.7
eig	38.4	0.15	94.5
gc	78.3	0.07	97.8

Table 1: Evaluation of the propose graph (gc) cuts based foreground segmentation method. Both Eigenbackgrounds (eig) and Gaussian mixture models (gmm) are outperformed by far.

In order to evaluate the proposed foreground segmentation method, we decided to manually annotate 140 images from the PETS 2009 S2L1 sequence, resulting in 20 images from each available views. Thereby binary masks containing the foreground have been manually created using the Gimp. The automatically created foreground images were subsequently compared to the ground truth. In order to create a meaningful set of metrics, we decided to use the detection rate (dr), denoting correctly assigned foreground pixels, false positive rate (fp), denoting background pixels being assigned to the foreground, and the overall accuracy (acc). The computed numbers of the graph cuts (gc) based method are illustrated in tab. 1. Furthermore the results are compared to a background model applying Gaussian Mixtures (gmm) and so called Eigenbackgrounds (eig) [19]. While the accuracy of all approaches is considerably high, here in the high nineties, and differs by a maximum of 3%, the detection rates differ significantly. It is remarkably that the graph cuts based method outperforms the remaining approaches by apx. 30%. The small difference in accuracy can nevertheless be explained by the small amount of positive pixels, compared to the over all amount of pixels, in the randomly chosen evaluation set. We also managed to lower the already considerably low false positive rates, which results in just some additional Gaussian noise in the foreground image.
Fig. 4 illustrates the results of a GMM and the proposed method, where frame 15 of camera 4 in scene S2-L1 has been chosen as example. It is obviously that the objects are all detected with little to none error. Further it can be seen that even with only 15 frames the segmentation is already possible, while the GMM still needs training, as the initial frame has not been entirely empty.

Figure 4: a) Input image. b) GMM based foreground segmentation c) Output of the proposed method.

3. Multi Camera Tracking

3.1 Person Localization Applying Multi Layer Homography

In the first stage a synchronized image acquisition is needed, in order to compute the correspondences of moving objects in the corresponding views $C_1, C_2, \ldots, C_n$. Additionally the sensors should be set up keeping in mind that the observed region should be as large as possible and direct occlusions of the sensor should be avoided. Therefore a field of view looking down on the scene from an elevated point would be preferable.
Subsequently a foreground segmentation is performed in all available smart sensors to detect changes from the empty background BG [5] :

$$FG_n(x, y, t) = I_n(x, y, t) - BG_n(x, y) \qquad (6)$$

where a Gaussian Mixture Model is applied for foreground segmentation. Now the homography H_i between a pixel

p_i in the view C_i and the corresponding location on the ground plane π can be determined. In all views the observations $x_1, x_2, \ldots, x_n$ can be made at the pixel positions $p_1, p_2, \ldots, p_n$. Let X resemble the event that a foreground pixel p_i has a piercing point within a foreground object with the probability $P(X|x_1, x_2, \ldots, x_n)$. With Bayes' law

$$P(X|x_1, x_2, \ldots, x_n) \propto P(x_1, x_2, \ldots, x_n|X)P(X) \quad (7)$$

the first term on the right side is the likelihood of making an observation $x_1, x_2, ..., x_n$ given an event X happens. Assuming conditional independence, the term can be rewritten to

$$P(x_1, \ldots, x_n|X) = P(x_1|X) \times \ldots \times P(x_n|X) \quad (8)$$

According to the homography constraint, a pixel within an object will be part of the foreground object in every view

$$P(x_i|X) \propto L(x_i) \quad (9)$$

where $L(x_i)$ is the probability of x_i belonging to the foreground. An object is then detected in the ground plane when

$$P(X|x_1, x_2, \ldots, x_n) \propto \prod_{i=1}^{n} L(x_i) \quad (10)$$

exceeds a threshold θ. In order to keep computational effort low it is feasible to transform only regions of interest. These are determined by thresholding the entire image, resulting in a binary image, before the transformation and the detection of blobs with a simple connected component analysis. This way only the binary blobs are transformed into the ground plane instead of probabilities. Therefore eq. 10 can be simplified to

$$P(X|x_1, x_2, \ldots, x_n) \propto \sum_{i=1}^{n} L(x_i) \quad (11)$$

without any influence on the performance. The value of theta θ is usually set dependent on the number n of camera sensors to $\theta = n - 1$, in order to provide some additional robustness in case one of the views accidentally fails. The thresholding on sensor level has a further advantage compared to the so called *soft threshold* [5], where the entire probability map is transformed and probabilities are actually multiplied as in eq. 10. A small probability or even $x_i = 0$ would result in a small overall probability, whereas the thresholded sum is not affected that dramatically. Using the homography constraint hence solves the correspondence problem in the views $C_1, C_2, \ldots, C_n$, as illustrated in fig 5a) for a cubic object. In case the object is human, only the feet of the person touching the ground plane will be detected. The homography constraint additionally resolves occlusions, as can be seen in fig. 5a). Pixel regions located

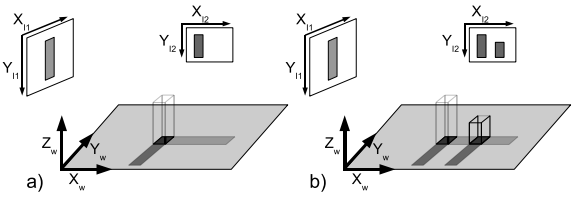

Figure 5: a) Planar homography for object detection.b) Resolving occlusions by adding further views.

within the detected foreground areas, indicated in gray on white ground and representing the feet, will be transformed to a piercing point within the object volume. Foreground pixels not satisfying the homography constraint are located off the plane, and are being warped into background regions of other views. The piercing point is located outside the object volume. All outliers indicate regions with high uncertainty, as there is no depth information available. This limitation can now be used to detect occluded objects. As visualized in fig. 5b) the smaller cuboid is occluded by the large one in view C_1, as apparently foreground blobs are merged. The smaller object's bottom side is occluded by the larger object's body. In contrast both objects are visible in view C_2, resulting in two detected foreground regions. A second set of foreground pixel, located off the ground plane π, in view C_1 will now satisfy the homography constraint and localize the occluded object. This process allows the localization of feet positions, although they are entirely occluded, by creating a kind of see through effect.

The implemented algorithm can be described as following:

- Foreground objects ψ_{in} are detected in all n views and a binary map is created. Subsequently n object boundaries can be extracted utilizing connected components analysis in the binary image
- Object boundaries are then being transformed into a predefined reference view

  $$\Psi_{in} = \mathbf{H}\psi_{in}. \quad (12)$$

  Though any of the views can be chosen, the most convenient one is a top view on the ground plane, visualizing spatial relationships between objects.
- Next the intersections of the polygons are computed. These can be calculated by a plane-sweep algorithm within the reference view. The binary represented regions B_n

  $$B_n(x, y) = \left\{ \begin{array}{c} 1 \text{ if } P_n(x, y) \in \Psi_{in} \\ 0 \text{ else} \end{array} \right\} \quad (13)$$

  located within detected foreground, are now transformed into the ground plane. In a subsequent step

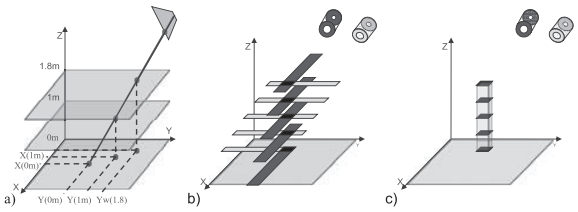

Figure 6: Exemplary object detection utilizing multiple layers.

these values are summed up to

$$B(x,y) = \sum_{i=1}^{n} B_i(x,y). \tag{14}$$

- The resulting map B(x,y) is subsequently thresholded with the previously defined parameter θ to encounter possible object regions

$$S(x,y) = \begin{cases} 1 \text{ if } B(x,y) \geq \theta \\ 0 \text{ else} \end{cases} \tag{15}$$

  This has been frequently computed with $\theta = n - 1$ to obtain higher reliability in the tracking process [20]. Experience has shown that this fixed threshold should only be applied in regions which are covered by all available cameras. As this region is usually rather small and cannot be granted due to camera positioning, the threshold can be set to a lower value. In order to keep the false positive rate at a considerably low level, exhaustive experiments have shown that three intersecting blobs are sufficient and should be preferred to the minimum amount of two intersections.

- Finally coherent regions indicating feet positions are indexed applying a simple connected component analysis.

Although the approach of Khan et al. localizes feet positions quite exactly, the performance is not sufficient in crowded situations. As only the feet are detected and these are not necessarily located next to each other, most persons are split into at least two parts. In order to combine the single feet it has been suggested to transform the detected foreground regions in multiple layers [6]. Fig. 6 illustrates this process. As can be seen the upper body parts of the object are drawn above the original feet position. By stacking the layers into the ground plane it is possible to align the feet. Furthermore it is possible to approximate the object position in 3D.

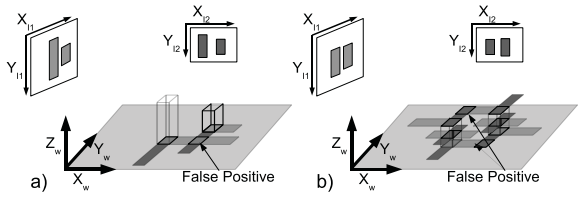

Figure 7: Creation of false positive regions in case multiple objects are located in a scene: Example for the creation of false positives in the ground plane and floating ghost objects due to multi layer homography.

3.2 Including Prior Knowledge

The presented approach obviously requires a robust detection of foreground regions, which can be provided by applying adaptive models. These of course cannot cope with the problem of static objects, such as trees, traffic signs, buildings, etc, which are occluding the scenery. Some of these objects have been identified in the PETS2009 data set. In order to incorporate the static obstacles, experiments have shown that it is sufficient to label the object position manually and include these as a foreground object. It can be transformed as any other object and be considered during the fusion process. In case further intersections, which are created by other fields of view, appear in the region behind the obstacle, it can be assumed that an occluded object is present in the scene.

3.3 False Positive Detection

The ability to detect partially occluded objects applying homography with high accuracy comes at the cost of a possibly large number of false positives [5] or so called ghost objects [21]. In case only one single object is present in the scene no errors will occur. As soon as there are two or more objects visible additional post processing steps have to be performed. Depending on the constellation of objects and cameras the boundaries of the transformed blobs may create additional intersections. These usually appear in regions covered by all objects and are hence not being visible in all views, as illustrated in fig. 7. Ambiguities like these can be resolved by adding further fields of view. It has thus been commonly agreed to use more cameras to reduce the number of false positives and increase the number of true positives [5, 22]. In real world applications the amount of hardware and the computational effort are supposed to be held as low as possible. All possible object locations are further examined in the field of view of each camera applying geometrical constraints [21, 20]. In order to detect a false positive we first analyze their creation and appearance in each camera view independently. Our experience has shown, that the upper part of an extracted foreground

Figure 8: Tracking result at frame 45 of S2-L1. 2 bounding boxes could not be created in the sixth frame, as these were projected outside the image boundaries

region is usually responsible for most of the false positives. Cutting these parts off is misleading, as partially occluded objects can not be detected anymore. This also shows the dilemma: is the detected object occluded, even only partially, or not? In order to answer this question, we propose to transform detected object positions in world coordinates O_{wi} back into the 2D domain in each camera view. This procedure will provide detailed information on the object arrangement in each view. Using the spatial arrangement of the transformed blobs in 2D and their approximated distance from the camera it is possible to determine whether an object is occluded by another one or not. Thereby it is important to determine the degree of occlusion. In case an object is entirely occluded by another one, the detected region located further away from the camera will be entirely located within the region of the occluding object. Partially occluded objects in contrast are not entirely located within other object regions. Therefore it can be assumed that an object, which is partially visible in at least one view, is a real object. If it is not visible in any of the available views it is probably a false positive candidate. Such a candidate should be incorporated into the tracking process nevertheless, as it could actually be a real object.

The presented approach is unfortunately only capable to localize objects without the capability of associating single detections. Therefore it is required to combine the localization procedure with a tracking algorithm. As only the object position and the approximated occupied area are known, it is not possible to rely on highly sophisticated methods that rely on textural information. In order to cope with this problem, it has been decided to apply simple Kalman filtering, although commonly only linear motion can be modeled.

4 Tracking Evaluation

Quantitative results of the PETS2009 evaluation are given in tab. 2, while a typical detection result is illustrated in fig. 8. As proposed in the PETS2009 evaluation methodology, MODA and MODP have been chosen as metrics. The homography approach obviously performs weaker than already reported results. One of the reasons is the bad estimation of the bounding box position in the real image, although the centroid seems to fit. A possible explanation could be errors in the calibration, as the views with low results were usually not considered in the localization task because the transformed blobs would not create intersections with other blobs. As the bounding box position has usually not been further analyzed, which resulted in frequent misalignments.

5 Conclusion and Outlook

We have presented an integrated approach for object tracking in multi camera surveillance systems applying an extension of the common homography approach. The system's reliability can be easily enhanced by transformation of foreground blobs in further layers. As the entire localization system solely relies on the extracted foreground, we introduced a robust method based on graph cuts. As we showed it easily outperforms common algorithms such as GMMs and Eigenbackgrounds. Utilizing the foreground maps we have been able to track the persons in the PETS2009 data set reliably.

Although the system has a high localization accuracy, ID changes appear frequently, as only the objects' positions in the ground plane are used for tracking. Therefore we propose to incorporate the texture into the tracking process, in order to avoid mix ups. Furthermore a more reliable cal-

camera	1	3	4	5	6	7	8
MODA	0.25	0.20	0.29	0.11	0.13	-0.01	0.13
MODP	0.32	0.32	0.38	0.35	0.35	0.25	0.33

Table 2: Evaluation of the proposed homography approach.

ibration method seems to be necessary, as even the back projected positions do not necessarily correspond.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Hilto Head Island, SC, USA*, 2000, vol. 2, pp. 142–149.

[2] W. Niu, J. Long, D. Han, and Y.-F. Wang, "Human activity detection and recognition for video surveillance," in *IEEE International Confenrence on Multimedia and Expo, Taipei, Taiwan*, June 2004, pp. 719–722.

[3] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Journal on Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[4] Z. Yue, S.K. Z, and R. Chellappa, "Robust two-camera tracking using homography," *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP2004, 17-21 May 2004, Montreal, Quebec, Canada*, vol. 3, pp. 1–4, May 2004.

[5] S.M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *Proceedings of the 10th European Conference on Computer Vision, ECCV 2006, Graz, Austria*, 2006, pp. 133–146.

[6] D. Arsić, N. Lehment, E. Hristov, B. Hörnler, B. Schuller, and G. Rigoll, "Applying multi layer homography for multi camera tracking," in *Proceeedings Second ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC2008, Stanford, CA, USA*, sep 2008.

[7] Antonio Torralba, "Contextual priming for object detection," *International Journal on Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[8] C. Stauffer, "Adaptive background mixture models for real-time tracking," in *Proceedings IEEE conference on Computer Vision and Pattern Recoginition, CVPR, Fort Collins, USA*, 1999, pp. 246–252.

[9] D. Arsić, M. Hofmann, B. Schuller, and G. Rigoll, "Multi-camera person tracking and left luggage detection applying homographic transformation," in *Proceedings Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2007, Rio de Janeiro, Brazil*, Oct. 2007.

[10] J. Zhong and S. Sclaroff, "Segmenting foreground objects form a dynamic textured background via a robust kalman filter," in *ICCV, IEEE, Nice, France*, 2003, pp. 44–50.

[11] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time tracking of the human body," *PAMI*, vol. 19, no. 7, pp. 780–785, 1997.

[12] W. Grimson and C. Staufer, "Adaptive background mixture models for real-time tracking," in *CVPR, IEEE, Ft. Collins, USA*, 1999, pp. II: 246–252.

[13] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European Conf. on Computer Vision*, 2000, pp. II:751–767.

[14] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. of the Workshop on Motion and Video Computing*. 2002, pp. 22–27, IEEE Comp. Socjety,Washington, DC, USA.

[15] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *IEEE Int. Conf. on Image Processing*, 2004, pp. V:3061–3064.

[16] N. J. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, pp. 187–193, 1995.

[17] T. Gevers and A. W. Smeulders, "Color based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[18] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[19] J. D. Rymel, J.-P. Renno, D. Greenhill, J. Orwell, and G. A. Jones, "Adaptive eigen-backgrounds for object detection," in *Proceedings IEEE International Conference on Image Processing (ICIP) 2005*, 2004, pp. 1847–1850.

[20] D. Arsić, B. Schuller, and G. Rigoll, "Multiple camera person tracking in multiple layers combining 2d and 3d information," in *In Proceedings Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), Marseille, France*, Oct. 2008.

[21] B. Michoud, S. Bouakaz, E. Guillou, and H. Briceno, "Largest silhouette-equivalent volume for 3d shapes modeling without ghost object," in *In Proceedings Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), October 12-18, 2008, Marseille, France*, Oct. 2008.

[22] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, CVPR2008, June 24-26, 2008, Anchorage, Alaska, USA*, June 2008, pp. 1–8.