Supporting Multi Camera Tracking by Monocular Deformable Graph Tracking

Nicolas Lehment, Dejan Arsić, Atanas Lyutskanov, Björn Schuller, Gerhard Rigoll
Institute for Human-Machine-Communication
Technische Universität München
Munich, Germany
{arsic, schuller, rigoll @tum.de} {lehment, lyutskanov @mytum.de}

Abstract

The reliable detection and tracking of objects, in particular humans, in video sequences is a requirement for video surveillance systems. This step enables automated threat detection systems to analyze trajectories and motion patterns. Thereby systems based on multiple overlapping fields of view have emerged in the last years. These are usually relying on simple foreground masks and discard texture information. In this work we propose to incorporate a 2D tracker into a multiple camera tracking system to avoid ID confusions during tracking. As both partial and total occlusions occur, trackers based on holistic person models usually fail. Therefore we propose to model object regions with an elastic feature graph, where the nodes are represented by SIFT features and are updated during the tracking process. This representation will enhance tracking performance in the 2D and can be applied in any view of a multi camera tracking system.

1. Introduction

Large public spaces have always been crucial to any society. It is in public spaces like town squares, airports, train stations, government buildings and shopping centers that we travel, meet, conduct our business and interact as a society. However, these spaces are therefore also vulnerable to abuse and attack. The safety concerns range from minor nuisances like pick-pockets to graver threads like terrorist attacks.

In order to reduce security risks, many of these places are under surveillance by video cameras (CCTV). These are typically monitored by police or private security personnel. Unfortunately human ability to track many different scenes, individuals, and interactions on multiple screen and in various scenes is limited. A human operator can usually observe only a fraction of a region covered by a sensor network. This leads naturally to blind spots which can be taken advantage of. By assuming a neutral appearance, criminals and terrorists are able to blend in with crowds. Drifting from one camera scene to another, they escape notice by moving continuously from one screen to another. All persons in the scene are thus able to travel freely through the scenery. Only a very experienced operator might notice them.

To overcome these limits and to support security in complex buildings and settings, several tools to support and automatize tracking and recognition of people in multi-camera scenarios have been investigated. Multi-layer homography, based on Khans homography constraint[1], is usually applied in multi sensor networks, to reliably track persons [2]. Current systems can automatically track persons and detect a set of special events like abandoned bags, theft and loitering based on simple heuristics [3, 4].

This multi camera approach allows for good individual tracking under most circumstances. However, under inconvenient conditions the system can lose track of an individual or confuses two tracked persons, and consequently confuses their assigned tracking ID. One of these situations is shown in fig. 1, where two previously tracked persons briefly obscure each other before parting again. Due to their close proximity and optical obstruction, the 3D tracking system registers the two individuals as one single entity and discards the second tracking ID-number. When they part, one person is assigned a new number while the other keeps the original tracking ID-number. To remedy this problem, we decided to introduce a second level of tracking. In addition to the existing multi-camera tracking system, we designed a secondary monocular tracking mechanism in order to resolve ambiguous situations during and after occlusions. In this first stage of development, we focus on using robust graph representations to track and identify individual persons. Such a graph consists of nodes, here represented by SIFT features, and edges, which are used to create geometrical relationships between the nodes.

1.1. Previous Work

After considering a number of options for the tracking mechanism, we decided to use a feature based tracker. The Scale Invariant Feature Transform (SIFT) was first de-

©2009 IEEE. All Rights Reserved.

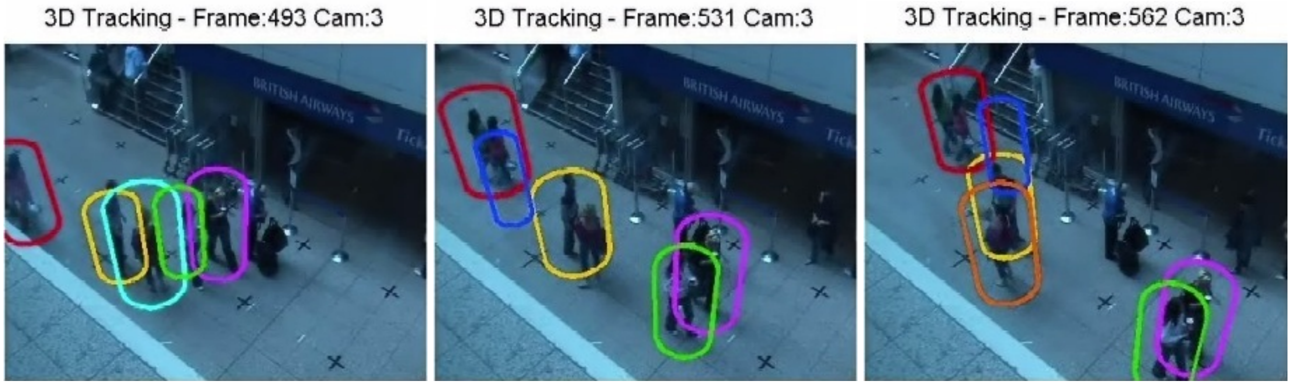
3D Tracking - Frame:493 Cam:3
3D Tracking - Frame:531 Cam:3
3D Tracking - Frame:562 Cam:3

Figure 1: Exemplary ambiguous situation in 3D tracking illustrated with the PETS2007 [5] dataset

scribed by Lowe ([6]). Since then, SIFT features have been used extensively in image rectification and classification tasks. Several groups have considered using these features for tracking applications, notably Gomila [7] and Tang [8]. A central topic for most of the works utilizing SIFT based image processing was the problem of matching an original graph to the current observation. Kisku [9] and Luo ([10]) considered these problems in the context of face recognition, while Berg [11] was using low distortion correspondence methods to identify objects against a database of labeled samples. Although typically computationally very expensive, the basic methods explored in these works can be adapted for use in tracking applications.

2. Tracking the Dynamic Feature-Mesh

While the concept of tracking a relatively rigid object using SIFT features has been repeatedly explored, for instance in [8], tracking a number of persons in a crowded scene adds a number of interesting and challenging problems. Movement of the body, especially arms and legs with their wide, swinging motions, continuously alters the appearance of the tracked person. While there is an underlying geometry defined by the skeleton, the extra layers of clothing and accessories like purses, hats and backpacks have their own and complex dynamics. Additionally we may observe partial or even total occlusions by moving and stationary objects, possibly even by other tracked persons.

2.1 Local Correspondence Search

As the tracked person undergoes continuous changes in appearance, we use an adapting mesh of SIFT features for tracking. For each tracked person, we define an undirected graph $O_{TR} = \{P_{\text{TR},1\ldots N}, D_{\text{TR},1\ldots N}\}$ as tracking reference, where $P_{\text{TR},i}$ describes the position of a specific SIFT feature $D_{\text{TR},i}$ within the bounding box of the detected person. By matching this graph to the graph of SIFT features visible in the current image $O_{IM} = \{P_{\text{IM},1\ldots M}, D_{\text{IM},1\ldots M}\}$, we can find the most likely assignment in the current view. To reduce computational overhead and avoid ambiguous assignments from descriptors between the two graphs, we first perform a correspondence search for similar feature descriptors. While Lowe's originally proposed correspondence search used only the descriptor distances, we also utilize the spatial information by computing the spatially weighted feature distances

$$d(D_{\text{TR},k}, D_{\text{IM},i}) = (d_{\text{eucl}}(P_{\text{TR},k}, P_{\text{IM},i}) + 1)(D^T_{\text{TR},k} D_{\text{IM},i})^2 \tag{1}$$

The Euclidean distance $d_{\text{eucl}}(P_{\text{TR},k}, P_{\text{IM},i})$ is computed using the position and scale of the respective features, thereby penalizing excessive movement and shifts in scale. We then select only features satisfying

$$\omega d(D_{\text{TR},k}, D_{\text{IM},i}) < d(D_{\text{TR},k}, D_{\text{IM},j}) \rightarrow \{k, i\}, \tag{2}$$

where j is the next closest distance and ω a suitably selected constant. In addition we limit the deviation of scale sc and main orientation ψ of the descriptors:

$$sc_{\text{TR},k} - sc_{\text{IM},i} \overset{!}{<} sc_{\text{max}} \quad \text{and} \quad \measuredangle(\psi_{\text{TR},k}, \psi_{\text{IM},i},) \overset{!}{<} \theta_\psi \tag{3}$$

This significantly reduces ambiguous assignments by favoring local candidate features. The graph of matched features is now $O_{IM*} = \{P_{\text{IM*},1\ldots M*}, D_{\text{IM*},1\ldots M*}\}$.

©2009 IEEE. All Rights Reserved.

2.2 Iterative Refinement of the Feature Graph

After the initial correspondence search there still remain a number of incorrect or ambiguous assignments. Especially in occlusion scenarios or in sequences with fast motions, we may see a number of bad assignments from O_{TR} to O_{IM}. By taking advantage of graph matching techniques, we can filter out most of these. Modifying a technique described in [11], we treat the correspondence problem as a spatial graph-matching problem. A matrix $\mathbf{A}$ of angles between the vectors in O_{TR} and O_{IM^*} is calculated:

$$\mathbf{x}_{\text{TR}}{}^{ij} = \begin{pmatrix} x_{\text{TR}}{}^{ij} \\ y_{\text{TR}}{}^{ij} \end{pmatrix} = P_{\text{TR}}{}^{i} - P_{\text{TR}}{}^{j} \tag{4}$$

$$\mathbf{x}_{\text{IM}}{}^{ij} = \begin{pmatrix} x_{\text{IM}}{}^{ij} \\ y_{\text{IM}}{}^{ij} \end{pmatrix} = P_{\text{IM}}{}^{i} - P_{\text{IM}}{}^{j} \tag{5}$$

$$A_{ij} = (\alpha \left|\bar{\mathbf{x}}_{\text{TR}}^{ij}\right| + \beta) \left| acos \left(\frac{\mathbf{x}_{\text{TR}}{}^{ij\,T} \mathbf{x}_{\text{IM}}{}^{ij}}{|\mathbf{x}_{\text{TR}}{}^{ij}|\,|\mathbf{x}_{\text{IM}}{}^{ij}|} \right) \right| \tag{6}$$

The two factors α and β allow for flexible matching by stronger penalizing angular deviations to distant points. For this purpose, we also use a normalized $\bar{\mathbf{x}}_{\text{TR}}^{ij} \in (0, 1)$.
These angles now represent the deviation between the features in O_{TR} and O_{IM}. By taking the mean deviation for each feature, we get a measure for the quality of the correspondence. If there is any deviation above a preset limit, the correspondence with the maximum deviation is deleted and the filtering process repeated until a satisfying result is obtained.

$$Q_i = \frac{1}{N}\sum_{j=1}^{N} A_{ij} \overset{!}{<} \theta_Q \tag{7}$$

The iterative approach is necessitated by the impact of one bad feature on the quality metric for all other features. As the angles do not change by removing an outlier, we can instead just discard the filtered rows and columns in the matrix A.

3. Updating the Dynamic Feature-Mesh

To incorporate for the changing appearance of tracked pedestrians into the tracking process, we need to update the feature mesh at every frame. This includes removing instable features, updating the positions based on previous observations, adding new features, and deleting old ones.

3.1 Predicting Mesh Dynamics

Since our correspondence search is locally bound, we need to predict the feature positions for the next frame by considering previous observations of each feature in the graph. While a global prediction based on the overall movement of the tracked person is possible, experience has shown that the swinging movement of limbs, clothes and other accessories often greatly deviates. We therefore use previous observations of each feature to predict its future position.
As a feature changes its general path during the tracking, we need to find a balance between filtering out disturbances and allowing for changes in arbitrary directions. By introducing a lifetime factor $0 < \gamma < 1$, we are able to weight previous observations in the calculation of the new expected position $\hat{x}_i$:

$$\gamma = \begin{pmatrix} \gamma \\ \gamma^2 \\ \vdots \\ \gamma^{n-1} \end{pmatrix} \qquad \mathbf{v}_i = \begin{bmatrix} \mathbf{x}_{i,t}^T - \mathbf{x}_{i,t-1}^T \\ \mathbf{x}_{i,t-2}^T - \mathbf{x}_{i,t-3}^T \\ \vdots \\ \mathbf{x}_{i,t-n+1}^T - \mathbf{x}_{i,t-n}^T \end{bmatrix} \tag{8}$$

and

$$\hat{\mathbf{x}}_i = \mathbf{x}_i + \frac{\mathbf{v}_i^T \gamma}{\sum_{k=1}^{k=n-1} \gamma(k)}, \tag{9}$$

with v_i resembling the first deviation of the previous positions. This acts as a weighted mean estimate of the future feature position. While more sophisticated approaches (e.g. Kalman Filter) might enhance the tracking performance, this prediction performs reasonably well under most circumstances.

3.2 Filtering Outliers

As we predict the future location for each feature independently, there is still the possibility that a bad feature, which does not belong to the tracked person, remains in the tracking graph undetected. Such features often enter the mesh in occlusion situations and can destabilize the tracking for the whole graph if these are not filtered out. Basically this corresponds to a multivariate outlier search.
To identify outliers, we use two metrices: filtering by position and filtering by motion. While the idea of position filtering is quite obvious, the motion filtering stems from the need to clear up mismatched features after occlusions. Since in most cases both graphs travel on in different directions, we can use motion vectors to identify outliers earlier than with position filtering. The vector of motion is easily obtained from the previous step of calculating $\hat{x}_i$.

©2009 IEEE. All Rights Reserved.

The tracking mesh has a roughly oval shape in the image. Clusters of features are normally located around the head, the shoulders and the torso with a few more unstable features on arms and legs. For these features in the more stable regions of the tracked person, the motion vectors usually have the same direction. This presents us with two clusters of 2-dimensional features. A multivariate analysis using Mahalanobis distances leads to a quick and reasonably reliable search for outliers:

$$d_{\text{Mahalanobis}}(\mathbf{x}, X) = \sqrt{(\mathbf{x}_i - \mu_X) S_X^{-1} (\mathbf{x}_i - \mu_X)} \quad (10)$$

$$\text{outlier}(i)_x = \delta(d_{\text{Mahalanobis}}(\mathbf{x}_i, P) > theta) \quad (11)$$

$$\text{outlier}(i)_v = \delta(d_{\text{Mahalanobis}}(\mathbf{v}_i, V_P) > \theta) \quad (12)$$

Another variant we tested included the generation of feature clusters by k-mean clustering before calculating the Mahalanobis distance in order to improve filtering during occlusions, where only some parts of the tracked person might be visible. However this brought no significant improvement and was therefore discontinued.

3.3 Updating the Tracking Mesh

As a person moves through the scene, the appearance and accordingly the SIFT features constantly change. While there is usually a number of relatively stable features, even those might disappear when the person turns around or undergoes other rapid changes. We therefore need a device to update the mesh constantly. A similar situation has been described in [8], where HMMs are used to judge the assignment of a feature to a tracking graph. We used a modified approach to our problem.
The tracked person is described by two separate graphs, namely the tracking graph O_{TR} and the candidate graph O_C. While O_{TR} contains the features used to track the person, O_C holds candidate features which were found in the vicinity of O_{TR} in previous frames. All features both in O_{TR} and O_C are assigned a score v which is updated as follows for each frame. Thereby current features are compared to previously detected features.

- Features TR+ in O_{TR} which were found in the current frame: $v(m+) = v(m+) + v_{\text{TR, pos}}$
- Features TR- in O_{TR} which were not found in the current frame: $v(m-) = v(m-) - v_{\text{TR, neg}}$
- Features C+ in O_{Temp} which were not found in the current frame: $v(n-) = v(n-) + v_{\text{Temp, pos}}$
- Features C- in O_{Temp} which were not found in the current frame: $v(n-) = v(n-) - v_{\text{Temp, neg}}$
- Features N_{up} in O_{Temp} where $v(n_{up}) > v_{stable}$ are moved to O_{TR}
- Features N_{down} in O_{TR} where $v(n_{up}) > v_{unstable}$ are moved to O_{Temp}
- Features N_{del} in O_{Temp} where $v(n_{del}) < v_{delete}$ are deleted entirely
- New features N_{New} are added to O_{Temp}, $v(n_{New}) = v_{\text{init}}$

By filtering for outliers we can now also modify the current score, so that normally a feature is not deleted instantly but instead is just weakened. A maximum score $v_{\text{max}} = 15$ avoids unrestricted strengthening of single features in order to facilitate removal of old features. An example of a possible, simplified tracking sequence is given in table 1.

t	v(1)	v(2)	v(3)	v(4)	v(5)	Events
1	5	5	5	5	-	-
2	6	6	6	4	-	[4] not found
3	7	5	7	3	1	[2,4] not found [5] → O_{Temp}
4	8	6	8	2	2	[4] not found [4] → O_{Temp}
5	9	7	9	1	3	[4] not found [5] → O_{TR}
6	10	8	10	-	4	[4] not found [4] deleted

Table 1: Example of the updating mechanism with $v_{stable} = 2$ and updating by steps of 1

3.4 Finding New Features and New Persons

New features are usually detected by an exhaustive search in foreground regions. These are obtained using a Gaussian Mixture Model, although most other robust foreground extraction methods can be used as well. The Gaussian Mixture Model we use is based on an adaptive algorithm proposed by Zivkovic in [12]. Deviating from the usual four Gaussian components, this algorithm computes the number K of components for each pixel individually. This yields local background models $\hat{p}(\mathbf{x}|\mathcal{X}, \text{BG})$ estimated from the set of previous observations $\mathcal{X}$. Assuming $p(FG) = p(BG)$ and and a uniform distribution for the foreground appearance $\mathbf{x}^{\text{t}} = c_{\text{FG}}$, we can build a Bayes classifier to determine if a pixel belongs to the background:

$$\frac{p(\text{BG}|\mathbf{x}^{\text{t}})}{p(\text{FG}|\mathbf{x}^{\text{t}})} = \frac{p(\mathbf{x}^{\text{t}}|\text{BG})p(\text{BG})}{p(\mathbf{x}^{\text{t}}|\text{FG})p(\text{FG})} \quad (13)$$

$$p(\mathbf{x}^{\text{t}}|\text{BG}) \approx \hat{p}(\mathbf{x}^{\text{t}}|\mathcal{X}, \text{BG}) \overset{?}{>} \theta_{\text{BG}} \quad (14)$$

©2009 IEEE. All Rights Reserved.

Once we have extracted the foreground, we assign each tracked mesh to its respective image region by considering the predicted feature positions. From the image region, we then add all unassigned features to the temporary mesh $O_{C,i}$. The procedure can be summarized as follows:

- Find the foreground region which contains most detected features for each tracking mesh and assign them to each other
- Check for double-assignments, perform person segmentation when necessary
- Initialize unassigned regions as new tracking meshes

Once each region is assigned to a tracking mesh, all new, unmatched features in that region are added to the respective temporary meshes $O_{C,i}$. It is important to note that the algorithm has no way of separating two persons entering the scene close to each other. If the two newly detected features are extracted as a single region, they are also initialized as a single person. Normally this problem resolves itself once they part briefly so that independent tracking is started.

4. Person Segmentation and Occlusion Handling

Most tracking algorithms suffer of handling either full or partial occlusions robustly. The fundamental question in these situations is the segmentation of the foreground. There is normally no clear indicator which pixel belongs to which person or whether a person is visible at all. Nevertheless we need a basic segmentation of the foreground to update the tracking graph. One of the primary difficulties is the lack of information on the nature of the occlusion. For example, in case two tracked persons cross paths, we do not necessarily know which one is standing in front of the other. In order to find a satisfying segmentation, we will therefore need to analyze the information contained in the detected parts of the tracking meshes, ${O_{TR}}^t$. We evaluated three different techniques:

- Naive Bayes Classifier without previous clustering
- Naive Bayes Classifier with k-means clustering
- Identification of overlapping regions + k-Nearest Neighbor Classification

While other, even more sophisticated methods were also considered, we found that due to the transient nature of the moving tracking features simple methods typically worked just as well: although frequent missclassifications are made, these are usually quickly rectified by the shifting of detected features in the next frame. ${O_{TR,\mathrm{i}}}^t$ and ${O_{TR,\mathrm{i}}}^{t-1}$ often differ significantly during occlusion situations. So although frequently smaller parts of the meshes get mixed up, they usually separate cleanly at the end of the occlusion. This is due to a stable core of features outside of the occluded zone. Therefore, we prefer simple and fast methods over more elaborate and costly procedures.

4.1 Naive Bayes Classifier

In an occlusion situation, there is always an increased risk of misclassified feature correspondence. A feature belonging to person ID_1 can easily be mistaken for a feature belonging to person ID_2 when the two are standing close to each other and share similar features. As we want to use the detected features to classify regions, these misclassifications are liable to destabilize the segmentation. We applied simple stochastic classification techniques hoping to reducing the effect of such outliers.

The positions of the detected features in the current frame are used to estimate the parameters of a Gaussian probability density function $p(\mathbf{y}|\Omega_i) = N(\mu_i, \Sigma_i)$ for each mesh $O_{TR,\mathrm{i}}$. We assume equal priors, so that we simply need to find the assigned class $\Omega_i = \max\limits_i p(\mathbf{y}|\Omega_i)$.

This approach proved to work only in situations with small occlusions (like a handshake) and good feature detection. In situations with more extensive occlusions, the clustering of features around a few significant regions like head and shoulders led to distorted estimates for $N(\mu_i, \Sigma_i)$. This led to an increasing number of misclassifications, where for example the leg regions of person A were classified as belonging to person B. Eventually these misclassifications would push the smaller, weaker tracking mesh out off the foreground region.

4.2 Naive Bayes Classifier with k-means clustering

In an attempt to encounter the distortion effects created by the strong clusters around a few significant regions, an additional step has been introduced: by using k-means clustering we hope to find a number of smaller, local clusters which would provide a more robust segmentation. The k_i-value was set to $k_i = \lceil \frac{\mathrm{num}(O_{TR,\mathrm{i}})}{20} \rceil$. As before, for each subgroup $p(\mathbf{y}|\Omega_k) = N(\mu_k, \Sigma_k)$ was estimated and eventually $\Omega_k = \max\limits_k p(\mathbf{y}|\Omega_k)$ determined. By remapping $\Omega_k \rightarrow \Omega_i$ the original were classes retrieved.

Again this approach proved to be too reliant on strong local clusters. While the problem of the distortion by a few local features was relieved, it would often happen that the mesh with more detected features would simply overwhelm a smaller, more distributed cluster, effectively taking over

©2009 IEEE. All Rights Reserved.

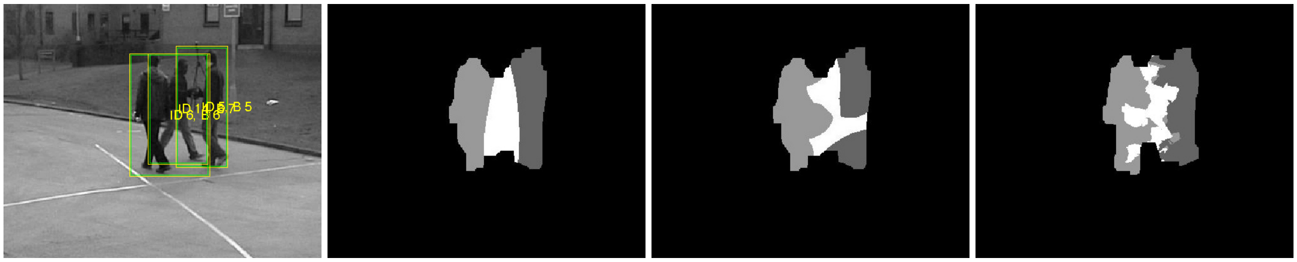

Figure 2: Comparison of segmentation for Bayes classifier, k-mean clustered Bayes classifier and K-NN segmentation.

the region of the other tracked person. We conclude that Bayes classifiers show only mediocre performance in cases where $|(O_{TR,\,\mathrm{i,\ matched}})| \gg |(O_{TR,\,\mathrm{j,\ matched}})|$

4.3 Identification of overlapping regions combined with K-Nearest Neighbor Classification

To remedy these shortcomings, we decided not to classify all points in the disputed foreground region, but instead to identify those regions with the most significant occlusions. We subsequently try to classify points by using a k-nearest neighbor approach only in regions creating confusions.

To find the overlapping regions, we construct polygons R_i over the mesh of found and safely identified features $O_{TR,\,\mathrm{i}}$ for each tracked mesh in the occluding situation. After identifying the regions $R_{\mathrm{occl}} = R_i \cap S_j \cap \ldots$, we identify the unmatched features $\{P_{\mathrm{occl}}, D_{\mathrm{occl}}\}$ contained within these regions. Next a 3-nearest neighbor classification is performed, using the spatial information in $O_{TR,\,\mathrm{i}}$, $O_{TR,\,\mathrm{j}}$ and possible further meshes as reference points.

Using the 3-nearest neighbor classification avoids some of the problems arising from the different clustering behaviors of occluded and occluding meshes. When two persons enter an occlusion, usually entire clusters of features situated on the edge of the silhouette are instantly lost. This consequently leads to a situation where the interior clusters located inside the person's silhouette are dominating the segmentation process, which is therefore heavily influenced by the distribution of those features. A Bayes classification often suffers from the scattering or wider distribution of some clusters. The k-Nearest Neighbor classification also depends on the distribution, but is not as heavily influenced by the scattering of a cluster. We therefore achieve a better segmentation even with widely differing number of matched features in $O_{TR,\,\mathrm{i}}$ and $O_{TR,\,\mathrm{j}}$.

5. Evaluation

The presented approach has been tested on the close-distance views of the PETS2009 benchmark data set S1.L1 *walking*. It showed promising performance in lightly and moderately crowded situations. We used the close distance camera views 5, 6, 7 and 8 for testing. Therefore the occurring occlusions have been manually annotated to evaluate the tracking performance, where especially ID maintenance has been in our focus of attention. The tracker was able to handle most long partial and brief full occlusions between two and three persons, provided that all persons had been tracked with at least stable 20 features for 5 frames previously. In several sequences, occlusions with significant change in direction and appearance were resolved successfully.

The evaluation results are summarized in table 2. Two factors were measured independently: the resolution of occlusions with regard to the number of people involved, here two, three or more than three, and the overall tracking success. An occlusion, i. e. two or more tracking meshes assigned to the same foreground region, was considered to be resolved successfully if all tracked persons entering the occlusion are assigned the same ID as before after the foreground regions split again. Switching tracking assignments between two persons has also been recorded. A tracking was considered successful, in case a person was assigned the same ID for the whole time spent in the view of the camera. We did not consider situations arising while people were entering the scene, since at that time no tracking of that person took place yet.

The current, non-optimized implementation was tested under Matlab 2007a on a MacBook (Intel Core2Duo 2 GHz, 2, GB RAM) with an average of 13 s/frame. As we expect a C-implementation to run considerably faster, realtime tracking is considered feasible.

It is important to note that the system can only perform adequately, if a sufficient amount of stable features is detectable. Our experience has shown that 10 features are the minimum number recquired. While tracking with fewer features is possible, this greatly impairs person segmentation

©2009 IEEE. All Rights Reserved.

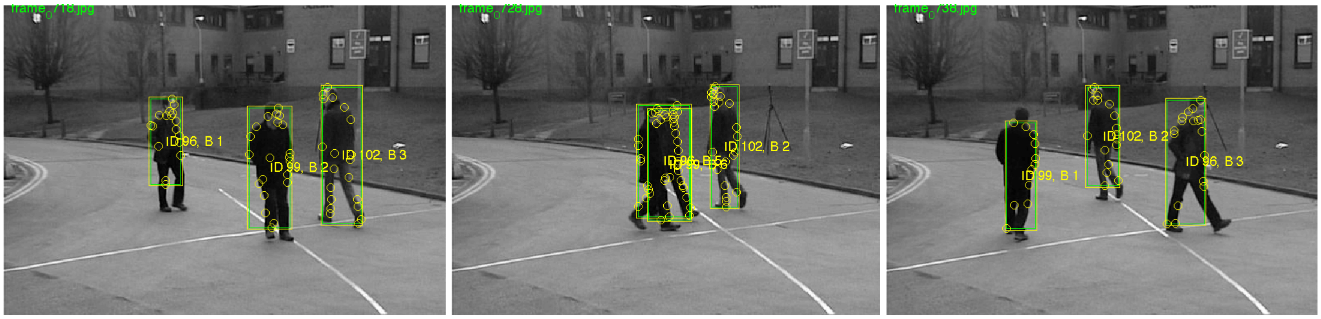

Figure 3: Example of resolved occlusion event with changing directions and rapid movement

Categories		C5	C6	C7	C8	Accuracy	
Occlusion Resolution							
2 Persons	+	10	12	8	14	44	72.1 %
	-	2	5	4	4	15	
	∼	0	1	1	0	2	
3 Persons	+	6	6	5	9	26	78.8 %
	-	0	0	0	3	3	
	∼	0	0	2	2	4	
>3 Persons	+	2	0	2	0	4	33.3 %
	-	1	3	1	1	6	
	∼	0	0	1	1	2	
Overall Tracking Performance							
All Tracks	+	18	15	13	16	62	68.9 %
	-	5	9	9	5	28	

Table 2: Summary of tracking performance and occlusion resolution. Overall success rate for categories given in percent. (+) denotes success, (-) a failure and (∼) a switching of ID assignments.

and therefore robustness in occlusion situations. Persons in drab and dark clothing are especially challenging. In these cases, nearly all features are located in the face or on the unstable silhouette of the person. Furthermore, wide area views suffered from too few discernible features for prolonged stable tracking and were therefore not used in the evaluation.

Most observed tracking failures can be ascribed to badly tracked or untracked persons. Especially persons entering the scene at a brisk speed are hard to track, since the prediction of future feature position fails (all speeds are initialized to zero). Untracked persons are not registered in occlusion events and consequently no person segmentation is performed in the occluded areas. Since now the whole foreground region is assumed to belong to the same, tracked person, foreign features are introduced into the existing mesh. On separation, the original graph may then be dragged away by the previously untracked person, depending on occlusion time and feature clustering.

Another unresolved problem are longer occlusions between two persons or a longer disappearance behind a stationary object. Two things may happen here: the obscured person changes pose and appearance, so the original mesh fits no longer the observations after the person is visible again, and the track is consequently lost. Alternatively the person might be obscured for such a long time that all the features in the tracking mesh are deleted due to missing observations. Both cases might lead to similar effects as an occlusion with an untracked person.

It is important to note that the drawbacks described above are not unsurmountable problems. The issue of too low initialization speeds for newly tracked persons might be solved by considering SIFT-flow in the image region. The loss of track after prolonged occlusion is basically a person recognition problem. We are therefore confident that further development of the SIFT tracking method described in this paper will yield reliable and robust tracking.

In case multiple cameras are available and correspondences of the bounding boxes can be determined via homography, classical multi camera tracking techniques can be supported by a simple majority voting. As it is rather unlikely that tracking fails in all fields of view, some of the IDs should be maintained in any case. As seen in the distribution of the test material occlusions don not necessarily appear in each view at the same time. This fact can be used to rise the overall ID maintenance rate.

6. Conclusion and Outlook

As explained at the beginning of this work, the long term goal of our research into SIFT tracking is to support and extend the capabilities of the existing homographic transform tracker. The integration into our 3-D tracking system now requires the design and implementation of a 3-D model for the tracked persons. Using the position information from the 3-D tracking, one may observe and update the SIFT model of a tracked person simultaneously

©2009 IEEE. All Rights Reserved.

from several different angles. A total integration into the existing tracking mechanism with a unified tracking model would then enable sharing of tracking information between the two levels of tracking and the various cameras, effectively allowing for uninterrupted tracking of the person. First experiments with the PETS2007 dataset have already shown a by far smaller amount of ID changes [13], where homography tracking created 15 confusions, while combined 2D-3D tracking created only two confusions. The introduction of clustering techniques and the obviously more elaborate updating technique outperform the former approach by far.

We expect that further study of deformable graphs based on SIFT features and faster, more robust non-rigid graph matching techniques will enable systems which are based entirely on SIFT-features, making the foreground extraction by Gaussian Mixture Model obsolete. Especially considering the restraints arising from the hardware normally used in surveillance settings (low resolution, grayscale images, frequent occlusions), we expect SIFT-based tracking to become an essential tool in surveillance and event-detection systems.

References

[1] 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.

[2] 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, pp. 1–9.

[3] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in *Proceedings of the ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2006, IEEE, New York, NY, USA*, Oct. 2006.

[4] D. Arsić, M. Hofmann, B. Schuller, and G. Rigoll, "Multi-camera person tracking and left luggage detection applying homographic transformation," in *Proc. of Tenth IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2007, Rio de Janeiro, Brazil*, J. M. Ferryman, Ed. 2007, pp. 55–62, University of Reading, UK, 14.10.2007 ISBN 0-7049-1423-9.

[5] J. Ferryman and D. Tweed, "An Overview of the PETS 2007 Dataset," in *Proceedings Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2007, Rio de Janeiro, Brazil*, October 2007.

[6] D.G. Lowe, "Object recognition from local scale-invariant features," *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157 vol.2, 1999.

[7] C. Gomila and F. Meyer, "Graph-based object tracking," *Proceedings IEEE International Conference on Image Processing, ICIP2003*, vol. 2, pp. II–41–4 vol.3, Sept. 2003.

[8] F. Tang and H. Tao, "Object tracking with dynamic feature graph," *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 25–32, Oct. 2005.

[9] D.R. Kisku, A. Rattani, E. Grosso, and M. Tistarelli, "Face identification by sift-based complete graph topology," *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, pp. 63–68, June 2007.

[10] Jun Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and Bao-Liang Lu, "Person-specific sift features for face recognition," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, pp. II–593–II–596, April 2007.

[11] A.C. Berg, T.L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 26–33 vol. 1, 2005.

[12] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, Washington, DC, USA, 2004, pp. 28–31, IEEE Computer Society.

[13] 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), October 12-18, 2008, Marseille, France*, Oct. 2008.

©2009 IEEE. All Rights Reserved.