Real Time Person Tracking and Behavior Interpretation in Multi Camera Scenarios Applying Homography and Coupled HMMs

Dejan Arsić[1] and Björn Schuller[2]

[1] Müller BBM Vibroakustiksysteme GmbH, Planegg, Germany
DArsic@MuellerBBM-VAS.de
[2] Institute for Human-Machine Communication,
Technische Universität München, Germany
schuller@tum.de

1 Introduction

Visual surveillance systems, which are quite common in urban environments, aim at providing safety in everyday life. Unfortunately most CCTV cameras are unmonitored and the vast majority of benefits are either in forensic use or deterring potential offenders, as these might be easily recognized and detected [40]. Therefore it seems desirable to support human operators and implement automated surveillance systems to be able to react in time. In order to achieve this aim most systems are split into two parts, the detection and tracking application and the subsequent behavior interpretation part.

As video material may contain various stationary or moving objects and persons whose behavior may be interesting, these have to be detected in the current video frame and tracked over time. As a single camera usually is not sufficient to cope with dense crowds and large regions, multiple cameras should be mounted to view defined regions from different perspectives. Within these perspectives corresponding objects have to be located. Appearance based methods, such as matching color [32], lead to frequent errors due to different color settings and lighting situations in the individual sensors.

Approaches based on geometrical information rely on geometrical constraints between views, using calibrated data [43] or homography between uncalibrated views, which e.g. Khan [25] suggested to localize feet positions. However, as Khan's approach only localizes feet, it consequently tends to segment persons into further parts. In these respects a novel extension to this framework is presented herein, applying homography in multiple layers to successfully overcome the problem of aligning multiple segments belonging to one person. As convenient side effect the localization performance will increase dramatically [6]. Nevertheless this approach still creates some errors in complex scenes and is computationally quite expensive. Therefore a real time capable alteration of the initial homography approach will be presented in sec. 2. The results of the applied tracking approaches will be presented using the multi camera tracking databases from the Performance Evaluation of Tracking and Surveillance Challanges (PETS) in the years 2006, 2007 and 2009 [37,3,28]. All these databases have been recorded in public places, such as train stations or airports, and show at leas four views of the scene.

Subsequently an integrated approach for behavior interpretation will be presented in sec. 3. Although a wide range of approaches already exists, this issue is not yet solved. Most of these operate on 2D level using texture information to extract behaviors or gait [39,15]. Unfortunately it is not possible to guarantee similar and non-obscured views in real world scenarios, which are required by these algorithms. Hence it is suggested to operate on trajectory level. Trajectories can be can be extracted robustly by the previously mentioned algorithm, easily be normalized and compared to a baseline scenario with little to no changes and knowledge of the scene geometry. Nevertheless the positions of important landmarks and objects, which may be needed for the scenario recognition, should be collected. Other information is not required. Common approaches come at the cost of collecting a large amount of data to train Hidden Markov Models (HMM) [31] or behavioral maps [11]. Despite the scenario's complexity and large inter class variance, some scenarios are following a similar scheme, which can be modeled by a HMM architecture in two layers, where the first layer is responsible for the recognition of Low Level Activities (LLA). In the second layer complex scenarios are furthermore analyzed again applying HMMS, where only LLAs are used as features. High flexibility and robustness is achieved by the introduction of state transition between High Level Activities (HLA), allowing a detailed dynamic scene representation. It will be shown that this approach provides a high accuracy at low computational effort.

2 Object Localization Using Homography

2.1 Planar Homographies

Homography [22] is a special case of projective geometry. It enables the mapping of points in spaces with different dimensionality $\mathbb{R}^n$ [17]. Hence, a point p observed in a view can be mapped into its corresponding point $p^{'}$ in another perspective or even coordinate system. Fig. 1 illustrates this for the transformation of a point p in world coordinates $\mathbb{R}^3$ into the image pixel $p^{'}$ in $\mathbb{R}^2$

$$p^{'} = (x,y) \leftarrow p = (x,y,z). \tag{1}$$

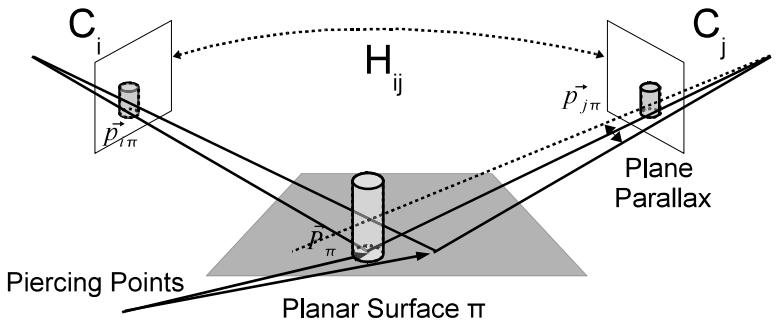

Fig. 1. The homography constraint visualized with a cylinder standing on a planar surface

Planar homographies, here the matching of image coordinates onto the ground plane, in contrast only require an affine transformation from $\mathbb{R}^2 \rightarrow \mathbb{R}^2$. This can be interpreted as a simple rotation with R and translation with T

$$p^{'} = \mathbf{R}p + \mathbf{T}. \tag{2}$$

As has been shown in [25], projective geometry between multiple cameras and a plane in world coordinates can be used for person tracking. A point p_π located on the plane is visible as $p_{i\pi}$ in view C_i and as $p_{j\pi}$ in a second view C_j. $p_{i\pi}$ and $p_{j\pi}$ can be determined with

$$p_{i\pi} = \mathbf{H}_{i\pi}p_\pi \qquad \text{and} \qquad p_{j\pi} = \mathbf{H}_{j\pi}p_\pi, \tag{3}$$

where $H_{i\pi}$ denotes the transformation between view C_i and the ground plane π. The composition of both perspectives results in a homography [22]

$$p_{j\pi} = \mathbf{H}_{j\pi}\mathbf{H}_{i\pi}^{-1}p_{i\pi} = \mathbf{H}_{ij}p_{i\pi} \tag{4}$$

between the images planes. This way each pixel in a view can be transformed into another arbitrary view, given the projection matrices for the two views. A 3D point $p_{\overline{\pi}}$ located off the plane π, visible at location $p_{i\overline{\pi}}$ in view C_i, can also be warped into another image with $p_w = \mathbf{H}p_{i\overline{\pi}}$, and $p_w \neq p_{2\overline{\pi}}$. The resulting misalignment is called plane parallax. As illustrated in fig. 1 the homography projects a ray from the camera center C_i through a pixel p and extends it until it intersects with the plane π, which is referred to as piercing point of a pixel and the plane π. The ray is subsequently projected into the camera center of C_j, intersecting the second image plane at p_w. As can be seen, points in the image plane do not have any plane parallax, whereas those off the plane do have a considerable such.

Each scene point p_π located on an object in the 3D scene and on plane π will therefore be projected into a pixel $p_{1\pi}, p_{2\pi}, \cdots, p_{n\pi}$ in all available n views, if the projections are located in detected foreground regions FG_i with

$$p_{i\pi} \in FG_i. \tag{5}$$

Furthermore, each point $p_{i\pi}$ can be determined by a transformation between view i and an arbitrary chosen one indexed with j

$$p_{i\pi} = \mathbf{H}_{ij}p_{j\pi}, \tag{6}$$

where $\mathbf{H}_{ij}$ is the homography of plane π from view i to j. Given a foreground pixel $p_i \in FG_i$ in view C_i, with its piercing point located inside the volume of an object inside the scene, the projection

$$p_j = \mathbf{H}_{ij} p_i \in FG_j \tag{7}$$

lies in the foreground region FG_j. This proposition, the so called homography constraint, is segmenting pixel corresponding to ground plane positions of objects and helps resolving occlusions.

The homography constraint is not necessarily limited to the ground plane and can be used in any other plane in the scene, as will be shown in sec. 2.2. For the localization of objects, the ground plane seems sufficient to find objects touching it. In the context of pedestrians a detection of feet is performed, which will be explained in the following sections. Now that it is possible to compute point correspondences from the 2D space to the 3D world and vice versa, it is also possible to determine the number of objects and their exact location in a scene. In the first stage a synchronized image acquisition is needed, in order to compute the correspondences of moving objects in the current frames $C_1, C_2, \ldots, C_n$. Subsequently, a foreground segmentation is performed in all available smart sensors to detect changes from the empty background $B(x,y)$ [25] :

$$FG_i(x,y,t) = I_i(x,y,t) - B_i(x,y) \tag{8}$$

where the appropriate technique to update the background pixel, here based on Gaussian Mixture Models, is chosen for each sensor individually. It is advisable to set parameters, such as the update time, separately in all sensors to guarantee a high performance. Computational effort is reduced by masking the images with a predefined tracking area. Now the homography $\mathbf{H}_{i\pi}$ 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 we have

$$p(X|x_1, x_2, \ldots, x_n) \propto p(x_1, x_2, \ldots, x_n|X)p(X). \tag{9}$$

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

$$p(x_1, x_2, \ldots, x_n|X) = p(x_1|X) \cdot p(x_2|X) \cdot \ldots \cdot p(x_n|X). \tag{10}$$

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

$$p(x_i|X) \propto p(x_i), \tag{11}$$

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

$$p(X|x_1, x_2, \ldots, x_n) \propto \prod_{i=1}^{n} p(x_i) \tag{12}$$

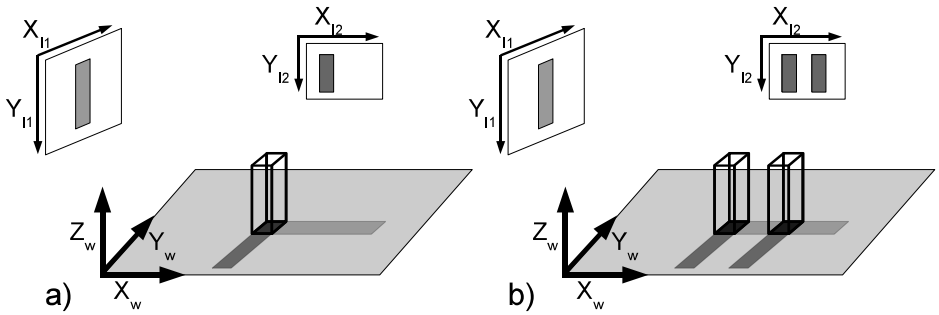

Fig. 2. a) Planar homography for object detection. b) Resolving occlusions by adding further views.

exceeds a threshold θ. In order to keep computational effort low, it is feasible to transform only regions of interest [3]. 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 the corresponding probability maps. Therefore eq. 12 can be simplified to

$$p(X|x_1,x_2,\ldots,x_n) \propto \sum_{i=1}^{n} p(x_i) \tag{13}$$

without any influence on the performance. The value of θ_{low} is usually set dependent on the number n of camera sensors to $\theta_{low} = n-1$, 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 [25,12], where the entire probability map is transformed and probabilities are actually multiplied as in eq. 12. A small probability or even $x_i = 0$ would affect the overall probability and set it to small values, whereas the thresholded sum is not affected. Using the homography constraint hence solves the correspondence problem in the views $C_1, C_2, \ldots, C_n$, as illustrated in fig 2a) 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. 2a). Pixel regions located within the detected foreground areas, indicated in dark gray on the ground plane, 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 therefore 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. 2b), one cuboid is occluded by the other one in view C_1, as apparently foreground blobs are merged. The right object's bottom side is occluded by the larger object's body. Both objects are visible in view C_2, resulting in two detected foreground regions. A second set of foreground pixels, located on the ground plane π in view C_2, 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.

Fig. 3. Detection example applying homographic transformation in the ground plane. Detected object regions are subsequently projected into the third view of the PETS2006 data set. The regions in yellow represent intersecting areas. As can be seen, some objects are split into multiple regions. These are aligned in a subsequent tracking step.

Exemplary results of the object localization are shown in fig. 3, where the yellow regions on the left hand side represent possible object positions. For an easier post processing, the resulting intersections are interpreted as circular object regions OR_i with center point $p_j(x,y,t)$ and its radius $r_j(t)$, which is given by $r_j(t) = \sqrt{\frac{A_j(t)}{\pi}}$, where $A_j(t)$ is the size of the intersecting region.

2.2 3D Reconstruction of the Scene

The major drawback of planar homography is the restriction to the detection of objects touching the ground, which leads to some unwanted phenomena. Humans usually have two legs and therefore two feet touching the ground, but unfortunately not necessarily positioned next to each other. Walking people will show a distance between their feet of up to one meter. Computing intersections in the ground plane consequently results in two object positions per person. Fig. 3 illustrates the detected regions for all four persons present in the scene. As only the position of the feet is determined, remaining information on body shape and posture is dismissed. As a consequence distances between objects and individuals cannot be determined exactly. For instance, a person might try to reach an object with her arm and be just few millimeters away from touching it, though the computed distance would be almost one meter. Furthermore, tracking is only limited to objects located on a plane, while other objects, such as hands, birds, etc. cannot be considered.

To resolve these limitations, it seems reasonable to try to reconstruct the observed scenery as a 3D model. Therefore various techniques have already been applied: Recent works mostly deal with the composition of so called visual hulls from an ensemble of 2D images [27,26], which requires a rather precise segmentation in each smart sensor and the use of 3D constructs like voxels or visual cones. These are subsequently being intersected in the 3D world. A comparison of scene reconstruction techniques can be found in [35].

An approach for 3D reconstruction of objects from multiple views applying homography has already been presented in [24]. All required information can be gathered by fusion of silhouettes in the image plane, which can be resolved by planar homography. With a large set of cameras or views a quite precise object reconstruction can be

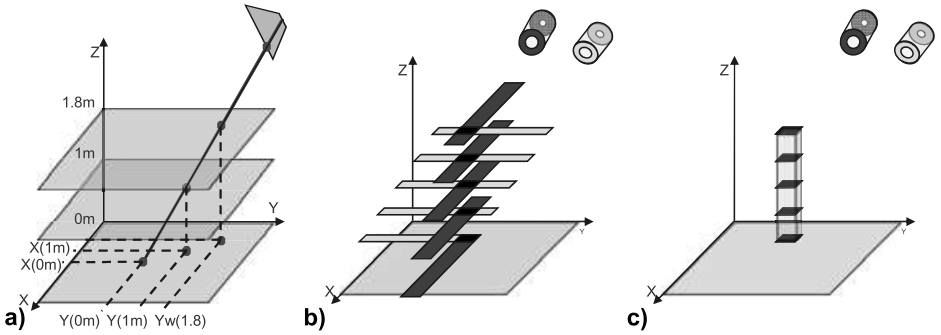

Fig. 4. a) Computation of layer intersections using two points. b) Transformed blobs in multiple layers. c) 3D reconstruction of a cuboid.

achieved, which is not required for this work. This approach can be altered to localize objects and approximate the occupied space with low additional effort [6], which will improve the detection and tracking performance.

The basic idea is to compute the intersections of transformed object boundaries in additional planes, as illustrated in fig. 4b). This transformation can be computed rapidly by taking epipolar geometry into account, which will be computationally more efficient than computing the transformation for each layer. All possible transformations of an image pixel $I(x,y)$ are basically located on an infinite line g in world coordinates (x_w, y_w, z_w). This line can be described by two points p_1 and p_2. Therefore only two transformations, which can be precomputed, are required for the subsequent processing steps. This procedure is usually only valid for a linear stretch in space, which can be assumed in most applied sensor setups.

The procedure described in sec. 2.1 is applied for each desired layer, resulting in intersecting regions in various heights, as illustrated in fig 4 b) and c). The object's height is not required as the polygons are only intersecting within the region above the person's position. In order to track humans it has been decided to use ten layers with a distance of $0.20\,m$ covering the range of $0.00\,m$ to $1.80\,m$, as this is usually sufficient to separate humans and only the head would be missing in case the person is by far taller. The ambiguities created by the planar homography approach are commonly solved by the upper body. Therefore the head, which is usually smaller than the body, is not required. The computed intersections have to be aligned in a subsequent step in order to reconstruct the objects' shapes. Assuming that an object does usually not float above another one, all layers can be stacked into one layer by projecting the intersections to the ground floor. This way a top view is simulated applying a simple summation of the pixel $P = (x_w, y_w, z_w)$ in all layers into one common ground floor layer with:

$$GF(x_w, y_w) = \sum_{l=1}^{n} P(x_w, y_w, z_l). \tag{14}$$

Subsequently, a connected component analysis is applied, in order to assign unique IDs to all possible object positions in the projected top view. Each ID is then propagated to the layers above the ground floor, providing a mapping of object regions in the single layers. Besides the exact object location, additionally volumetric information, such as

Fig. 5. Detection example on PETS2007 data [3] projected in two camera views. All persons, expect the lady in the ellipse, have been detected and labeled consistently in both views. The error occurred already in the foreground segmentation.

height, width, and depth, is extracted from the image data, providing a more detailed scene representation than the simple localization. Some localization examples are provided in fig. 5, where cylinders approximate the object volume. The operating area has been restricted to the predefined area of interest, which is the region with the marked up coordinate system. As can be seen, occlusions can be resolved easily without any errors. One miss, the lady marked with the black ellipse, appeared because of an error in the foreground segmentation. She has been standing in the same spot even before the background model has been created, and therefore not been detected.

2.3 Computational Optimization of the 3D Representation

The localization accuracy of the previously described approach comes at the cost of computational effort. Both the homography and the fusion in individual layers are quite demanding operations, although a simple mathematical model lies beneath them. Therefore a computationally more efficient variation will be presented in the following. As each detected foreground pixel is transformed into the ground plane, a vast amount of correspondences has to be post processed within the localization process. Instead of computing complex occupancy cones, the observed region is covered by a three dimensional grid with predefined edge lengths. Thus, we segment the observed space into a grid of volume elements, so called voxels. In a first step corresponding voxel and pixel positions in the recorded image are computed. This can be done by computing homographies in various layers, using occupancy rays cast from each image pixel in each separate camera view. Each voxel passed by a ray originating from one pixel is henceforth associated with that pixel. Due to the rough quantization of the 3D space, multiple pixel positions will be matched to each voxel. While slightly decreasing precision, this will result in a larger tolerance to calibration errors. As we now have a precomputed lookup table of pixel to voxel correspondences, it is possible to calculate an occupancy grid quickly for each following observation.

Each voxel is assigned a score which is set to zero at first. For each pixel showing a foreground object, all associated voxels' scores are incremented by one step. Going through all the foreground regions of all images, it is possible to compute the scores for each voxel in the occupancy grid. After all image pixels have been processed, a simple thresholding operation is performed on the scores of the voxels, excluding voxels with low scores and thus ambiguous regions. The remaining voxels with higher scores

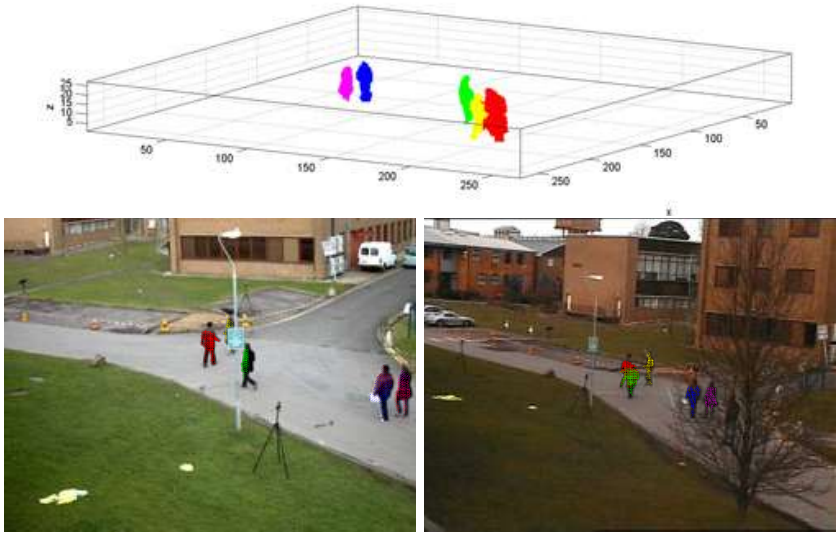

Fig. 6. 3D reconstruction and detection results of a scene from the PETS2009 [18] dataset

then provide an approximated volume of the observed object. The threshold is usually set equal to the number of cameras, meaning that a valid voxel needs an associated foreground/object pixel in each camera view.

After filling the individual grid elements, a connected components analysis, which is commonly used in image processing, is applied to the 3D voxel grid in order to locate objects. The only significant difference to the 2D operation is the number of possibly connected neighbor elements which rises from 8 to 26. An exemplary detection result is illustrated in fig. 6, using a scene from the PETS2009 workshop [18].

Due to the rough quantization of the tracking region, calibration errors and unreliable foreground segmentation could be partially eliminated, and a by far higher tracking accuracy has been reached applying this method, which has been evaluated using the PETS2007 database. While the localization accuracy of the multi layer homography approach (MLH) and the presented voxel based tracking achieved the same localization accuracy of $0.15m$, the number of ID changes has been decreased drastically from 18 to 3. this result is comparable to a combined MLH and 2D tracking approach, as presented in [8], where a graph based representation using SIFT features [30] has been applied [28]. Speaking in terms of tracking accuracy, the performance has not risen drastically. Computational effort has been decreased by the factor seven at the same time. This makes this approach by far more efficient than comparable ones.

3 Behavior Interpretation

The created trajectories and changes in motion patterns can now be used by a behavior interpretation module, which subsequently either triggers an alarm signal or reacts to the observed activity by other appropriate means [23]. This module is basically matching an unknown observation sequence to stored reference samples and performing a comparison. The basic problem is to find a meaningful representation of human behavior, which is a quite a challenging task even for highly trained human operators, who

indeed should be 'experts in the field'. A wide range of classifiers, based on statistical learning theory, has been employed in the past, in order to recognize different behavior. The probably most popular approaches involve the use of dynamic classifiers, such as HMMs [31] or Dynamic Time Warping [36]. Nevertheless static classifiers, e. g. Support Vector Machines (SVM) or Neural Networks (NN) are being further explored, as these may outperform dynamic ones [4]. All these approaches have in common that they are data driven approaches, which usually requires a vast amount of real world training data. This is usually not available, as authorities usually do not provide or simply do not have such data, and preparing data and model creation are quite time consuming. Therefore an effective solution has to be found to overcome this problem.

In order to be able to pick up interesting events and derive so called 'threat intentions', which may for instance include robberies or even the placement of explosives, a set of Predefined Indicators (PDI), such as loitering in a defined region, has been collected [13]. These PDIs have been assembled to complex scenarios, which can be interpreted as combination and temporal sequence of so called Low Level Activities. Hence, the entire approach consists of two subsequent steps: The Low Level Activity detection and the subsequent scene analysis using the outputs of the LLA analysis.

3.1 Feature Extraction

The recognition of complex events on trajectory level requires a detailed analysis of temporal events. A trajectory can be interpreted as an object projected onto the ground plane, and therefore techniques from the 2D domain can be used. According to Francois [20] and Choi [16], the most relevant trajectory related features are defined as follows: continue, appear, disappear, split, and merge. All these can be handled by the tracking algorithm, where the object age, meaning the number of frames a person is visible, can also be determined reliably. Additionally, motion patterns, such as speed and stationarity, are being analyzed.

- **Motion Features:** In order to be able to perform an analysis of LLAs from a wide range of recordings and setups, it is reasonable to remove the position of the person in the first place. It is important to detect running, walking or loitering persons, where the position only provides contextual information. Therefore only the persons' speed and acceleration are computed directly on trajectory level. The direction of movement can also be considered as contextual information, which leads to the conclusion to just record changes in the direction of motion on the xy plane.
- **Stationarity:** For some scenarios, such as left luggage detection, objects not altering their spatial position have to be picked up in a video sequence. Due to noise in the video material or slight changes in the detector output, e. g. the median of a particle filter, the object location is slightly jittering. A simple spatial threshold over time is usually not adequate, because the jitter might vary in intensity over time. Therefore the object position $p_i(t)$ is averaged over the last N frames:

$$p_i = \frac{1}{N} \sum_{t'=t-N}^{t} p_i(t') \tag{15}$$

Subsequently, the normalized variance in both $x-$ and $y-$ direction

$$\sigma_i(t) = \left| \frac{1}{N} \sum_{t'=t-N}^{t} p_i(t') - p_i)^2 \right| \tag{16}$$

is computed [9,3]. This step is required to smooth noise created by the sensors and errors during image processing. Stationarity can then be assumed for objects with a lower variance than a predefined threshold θ:

$$\text{stationarity} = \begin{cases} 1 & \text{if } var < \theta \\ 0 & \text{else} \end{cases}, \tag{17}$$

where 1 indicates stationarity and 0 represents walking or running. Given only the location coordinates, this method does not discriminate between pedestrians and other objects, enabling the stationarity detection for any given object in the scene. A detection example is illustrated in fig. 7.

– **Detection of Splits and Mergers:** According to Perera [33], splits and merges have to be detected in order to maintain IDs in the tracking task. Guler [21] tried to handle these as low level events describing more complex scenarios, such as people getting out of cars or forming crowds. A merger usually appears in case two previously independent objects $O_1(t)$ and $O_2(t)$ unite to a normally bigger one

$$O_{12}(t) = O_1(t-1) \cup O_2(t-1). \tag{18}$$

This observation is usually made in case two objects are either located extremely close to each other or touch one another in 3D, whereas in 2D a partial occlusion might be the reason for a merger. In contrast two objects $O_{11}(t)$ and $O_{12}(t)$ can be created by a splitting object $O_1(t-1)$, which might have been created by a previous merger.

While others usually analyze object texture and luminance [38], the applied rule based approach only relies on the object position and the regions' sizes. Disappearing and appearing objects have to be recognized during the tracking process, in order to incorporate a split or merge:

- **Merge:** one object disappears but two objects can be mapped on one and the same object during tracking. In an optimal case both surfaces would intersect with the resulting bigger surface

$$O_1(t-1) \cap O_{12}(t) \;\&\; O_2(t-1) \cap O_{12}(t). \tag{19}$$

- **Split:** Similar to the object split two objects at frame t are mapped to one object at time $t-1$, where the objects both intersect with the old splitting one

$$O_{11}(t) \cap O_1(t-1) \;\&\; O_{12}(t) \cap O_1(t-1). \tag{20}$$

– **Proximity of Objects:** As in various cases persons are interacting with each other, it seems reasonable to model combined motions. This can be done according to the direction of movement, proximity of objects, and velocity. As the direction of

Fig. 7. Exemplary recognition results for Walking, Loitering and Operating an ATM

motion can be simply computed, it is possible to elongate the motion vector v and compute intersections with interesting objects or other motion vectors. Further the distance between object positions can be easily detected with

$$d_{ij} = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}. \tag{21}$$

Distances in between persons and objects are usually computed scenario relatedly and require contextual knowledge, as the positions of fixed objects are known beforehand and these objects cannot necessarily be detected automatically. In case interactions between persons are required, it is sufficient to analyze only the objects with the smallest distance.

3.2 Low Level Activity Detection

The classification of Low Level activities has been performed applying various different techniques. Thereby rule based approaches [6] and Bayesian Networks [14] have been quite popular. As it is hard to handle continuous data streams with both approaches and to set up a wide set of rules for each activity, dynamic data driven classification should be preferred. Though it has previously been stated that data is hardly available, this accounts only for complex scenarios, such as robberies or theft. It is therefore reasonable to collect LLAs from different data sources and additionally collect a large amount of normal data containing none of the desired LLAs, as this will be the class usually appearing.

Hidden-Markov-Models [34] are applied for the trajectory analysis task in the first stage, as these can cope with dynamic sequences with variable length. Neither duration, start or end frame of the desired LLAs is known before the training phase. Only the order and number of activities for each sample in the database are defined. Each action is represented by a four or five state, left-right, continuous HMM and trained using the Baum-Welch-Algorithm [10]. During the training-process the activities are aligned to the training data via the Viterbi-Algorithm in order to find the start and end frames of the contained activities. The recognition task was performed applying the Viterbi-Algorithm.

For this task all features expect the contextual information, such as position or proximity, have been applied. Table 1 illustrates the desired classes and the recognition results. This approach has been evaluated on a total amount of 2.5 h of video including the

Table 1. Detection (det) results and false positives (fpos) for all five LLAs within the databases. The HMM based approach obviously outperforms the followed static Bayesian Networks approach.

Event	[#]	detBN	detHMM	fpos BN	fpos HMM
Running	14	10	13	1	0
Stationarity	7	7	0	0	0
Drop luggage	18	0	15	12	1
Pick up luggage	12	0	10	0	2
Loitering	60	60	60	3	1

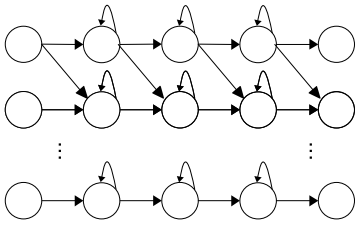

Fig. 8. a) Structure of the coupled HMMs

PETS2006, PETS2007, and the PROMETHEUS [1] datasets. As such a detailed analysis of the datasets has not yet been performed, a comparison to concurring approaches is not possible. Nevertheless results applying Bayesian Networks, as presented in [7] are provided if available. Note that the activities of interest only cover a small part of the databases. It is remarkable that for all classes only few misses can be reported and a very small amount of false positives is detected. A confusion matrix is not provided, as usually misses were confused with neutral behavior, while this was usually responsible for false positives. Walking is handled as neutral behavior and due to the large amount of data not especially considered for the evaluation task. Nevertheless it can be recognized almost flawlessly, although longer sequences of walking are frequently segmented into shorter parts. This problem can be covered by summing up continuous streams of walking.

3.3 Scenario Recognition

Having extracted and detected all required LLAs, either with HMMs or using the tracking algorithm, these can now be further analyzed by a scenario interpretation module. Recent approaches were frequently based on a so called Scenario Description Language (SDL), which contains examples for each possible scenario [13]. Applying the SDL based approach can be interpreted as rule based reasoning, which can be achieved with a simple set of rules [8]. Current approaches use a wide range of LLA features and perform the analysis of behaviors or emotions with Dynamic Bayesian Networks (DBN) [41], which usually require a vast amount of data to compute the inferences. A simple form of the DBN, also data driven, is the well-known HMM. It is capable to segment and classify data streams at the same time. Current implementations usually analyze the trajectory created by one person, not allowing the interaction of multiple persons.

Table 2. Detection (det) results and false positives (fpos) for all five complex scenarios within the evaluated databases. Rules obviously perform by far weaker than DBNs, which are outperformed by HMMs.

Event	[#]	det DBN	det Rules	det HMM	fpos DBN	fpos Rules	fpos HMM
Left Luggage	11	9	5	10	3	6	2
Luggage Theft	6	2	0	4	3	1	2
Operate ATM	17	17	17	17	2	5	0
Wait at ATM	15	15	10	15	3	7	1
Robbery at ATM	3	3	2	3	0	4	0

Furthermore, it seems hard to compute transition probabilities when a wide range of states and orders is allowed, if only little data is available. Therefore it has already been proposed to couple Markov chains [13]. A DBN based approach has been presented in [7], where the outputs of individually classified trajectories have been combined to an overall decision. In contrast to the previously used simple Markovian structure, now a HMM based implementation is used to allow for more complex models and scenarios. As fig. 8 illustrates, the applied implementation allows transitions between several HMMs being run through in parallel. This has the advantage that not each and every scenario has to be modeled individually and links between individually modeled trajectories or persons can be established. In a very basic implementation it can be assumed that these state transitions are simple triggers, which set a feature value, allowing to leave the actual state, which has been repeated a couple of times.

One of the major issues with this approach is the need of real data. As this is not available in vast amounts, training has been performed using real data and an additional set of definitions by experts, where artificial variance has been included by insertions and deletions of observations. The trained models have been once more evaluated with the previously mentioned three databases, namely PETS2006, PETS2007 and PROMETHEUS. A brief overview on the results is given in table 2, which compares the HMM based approach to previous ones applying either rules [3] or the previously mentioned Dynamic Bayesian Networks (DBN) [7]. Obviously both DBNs and HMMs perform better than rule based approaches. The presented coupled HMM approach nevertheless performs slightly better than the previous DBN based implementation, which only allowed state transitions from left to the right and not between individual models. Especially the lower false positive rate of the coupled HMM approach is remarkable.

Fig. 9. Exemplary recognition of a robbery at an ATM

Two exemplary recognition results from the Prometheus database are provided in fig. 7 and fig. 9, where a person is either operating an ATM or being robbed at an ATM machine. As it can be seen, the activities in the scene are correctly picked up, assigned to the corresponding persons, and displayed in the figures.

4 Conclusion and Outlook

We have presented an integrated framework for the robust interpretation of complex behaviors utilizing multi camera surveillance systems. The tracking part has been conducted in a voxel based representation of the desired tracking regions, which has been based on Multi Layer Homography. This approach has been improved both in speed and performance by this rough quantization of space. Nevertheless tracking performance can be further enhanced by creating a 3D model of the person using information retrieved from the original images, as proposed for Probability Occupancy Maps [19]. Furthermore the introduction of other sensors, such as 3D cameras or thermal infrared, could provide a more reliable segmentation of the scene [5].

Further it has been demonstrated that a complex behavior can be decomposed into multiple easy to detect LLAs, which can be detected either during the tracking phase or applying HMMs. The detected LLA are subsequently fed into a behavior interpretation module, which uses coupled HMMs and allows transitions between concurring models. Applying this approach resulted in a high detection and low false positive rate for all three evaluated databases. For future development it would be desired to analyze persons in further detail, which would include the estimation of the person's pose [2,29], which will also allow the recognition of gestures [42]. Besides the introduction of further features and potential LLAs, the scenario interpretation needs further improvement. While a limited amount of behaviors can be modeled with little data, ambiguities between classes with low variance may not be distinguished that easily.

Summed up the presented methods can be used as assistance for human operated CCTV systems, helping staff to focus attention on noticeable events at a low false positive rate, though at the same time ensuring minimal false negatives.

References

1. Ahlberg, J., Arsić, D., Ganchev, T., Linderhed, A., Menezes, P., Ntalampiras, S., Olma, T., Potamitis, I., Ros, J.: Prometheus: Prediction and interpretation of human behavior based on probabilistic structures and heterogeneous sensors. In: Proceedings 18th ECCAI European Conference on Artificial Intelligence, ECAI 2008, Patras, Greece, pp. 38–39 (2008)
2. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. In: Proceedings International IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), pp. 623–630 (2010)
3. Arsić, D., Hofmann, M., Schuller, B., Rigoll, G.: 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, pp. 55–62 (2007)

4. Arsić, D., Hörnler, B., Schuller, B., Rigoll, G.: A hierarchical approach for visual suspicious behavior detection in aircrafts. In: Proceedings 16th IEEE International Conference on Digital Signal Processing, Special Session "Biometric Recognition and Verification of Persons and their Activities for Video Surveillance", DSP 2009, Santorini, Greece (2009)
5. Arsić, D., Hörnler, B., Schuller, B., Rigoll, G.: Resolving partial occlusions in crowded environments utilizing range data and video cameras. In: Proceedings 16th IEEE International Conference on Digital Signal Processing, Special Session "Fusion of Heterogeneous Data for Robust Estimation and Classification", DSP 2009, Santorini, Greece (2009)
6. Arsić, D., Lehment, N., Hristov, E., Hrnler, B., Schuller, B., Rigoll, G.: Applying multi layer homography for multi camera tracking. In: Proceeedings Second ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2008, Stanford, CA, USA, pp. 1–9 (2008)
7. Arsić, D., Lyutskanov, A., Kaiser, M., Rigoll, G.: Applying bayes markov chains for the detection of atm related scenarios. In: Proceedings IEEE Workshop on Applications of Computer Vision (WACV), in Conj. with the IEEE Computer Society's Winter Vision Meetings, Snowbird, Utah, USA, pp. 1–8 (2009)
8. Arsić, D., Schuller, B., Rigoll, G.: Multiple camera person tracking in multiple layers combining 2d and 3d information. In: Proceedings Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), Marseille, France (2008)
9. Auvinet, E., Grossmann, E., Rougier, C., Dahmane, M., Meunier, J.: Left luggage detection using homographies and simple heuristics. In: Proceedings Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2006, New York, NY, USA, pp. 51–59 (2006)
10. Baum, L.E.: An inequality and associated maximalization technique in statistical estimation for probabilistic function of markov processes. Inequalities 3, 1–8 (1972)
11. Berclaz, J., Fleuret, F., Fua, P.: Multi-camera tracking and atypical motion detection with behavioral maps. In: Proceedings 10th European Conference on Computer Vision, Marseille, France (2008)
12. Broadhurst, A., Drummond, T., Cipolla, R.: A probabilistic framework for space carving. In: Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, pp. 388–393 (2001)
13. Carter, N., Ferryman, J.: The safee on-board threat detection system. In: Proceedings International Conference on Computer Vision Systems, pp. 79–88 (May 2008)
14. Carter, N., Young, D., Ferryman, J.: A combined bayesian markovian approach for behaviour recognition. In: Proceedings 18th International IEEE Conference on Pattern Recognition, ICPR 2006, Washington, DC, USA, pp. 761–764 (2006)
15. Chen, D., Liao, H.M., Shih, S.: Continuous human action segmentation and recognition using a spatio-temporal probabilistic framework. In: Proceedings Eighth IEEE International Symposium on Multimedia, ISM 2006, Washington, DC, USA, pp. 275–282 (2006)
16. Choi, J., Cho, Y., Cho, K., Bae, S., Yang, H.S.: A view-based multiple objects tracking and human action recognition for interactive virtual environments. The International Journal of Virtual Reality 7, 71–76 (2008)
17. Estrada, F., Jepson, A., Fleet, D.: Planar homographies, lecture notes foundations of computer vision. University of Toronto, Department of Computer Science (2004)
18. Ferryman, J., Shahrokni, A.: An overview of the pets 2009 challenge. In: Proceedings Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2009, Miami, FL, USA, pp. 1–8 (2009)
19. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 30(2), 267–282 (2008)
20. Francois, A.R.J.: Real-time multi-resolution blob tracking. In: IRIS Technical Report, IRIS-04-422, University of Southern California. Los Angeles, USA (2004)

21. Guler, S.: Scene and content analysis from multiple video streams. In: Proceedings 30th IEEE Workshop on Applied Imagery Pattern Recognition, AIPR 2001, pp. 119–123 (2001)
22. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
23. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34(3), 334–352 (2004)
24. Khan, S.M., Yan, P., Shah, M.: A homographic framework for the fusion of multi-view silhouettes. In: Proceedings Eleventh IEEE International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, pp. 1–8 (2007)
25. Khan, S., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 133–146. Springer, Heidelberg (2006)
26. Kutulakos, K., Seitz, S.: A theory of shape by space carving, technical report tr692. Tech. rep., Computer Science Deptartment, University Rochester (1998)
27. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(2), 150–162 (1994)
28. Lehment, N., Arsić, D., Lyutskanov, A., Schuller, B., Rigoll, G.: Supporting multi camera tracking by monocular deformable graph tracking. In: Proceedings Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2009, Miami, FL, USA, pp. 87–94 (2009)
29. Lehment, N., Kaiser, M., Arsic, D., Rigoll, G.: Cue-independent extending inverse kinematics for robust pose estimation in 3d point clouds. In: Proceeding IEEE International Conference on Image Processing (ICIP 2010), Hong Kong, China, pp. 2465–2468 (2010)
30. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
31. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis Machine Intelligence 22(8), 831–843 (2000)
32. Orwell, J., Remagnino, P., Jones, G.: Multi-camera colour tracking. In: Proceedings Second IEEE Workshop on Visual Surveillance, VS 1999, Fort Collins, CO, USA, pp. 14–21 (1999)
33. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: Proceedings 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, Washington, DC, USA, pp. 666–673 (2006)
34. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286 (1989)
35. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, CVPR, New York, NY, June 17-22, vol. 1, pp. 519–528 (2006)
36. Takahashi, K., Seki, S., Kojima, E., Oka, R.: Recognition of dexterous manipulations from time-varying images. In: Proceedings 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 23–28 (1994)
37. Thirde, D., Li, L., Ferryman, J.: Overview of the pets2006 challenge. In: Proceedings Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2006, pp. 1–8. IEEE, New York (2006)
38. Vigus, S., Bul, D., Canagarajah, C.: Video object tracking using region split and merge and a kalman filter tracking algorithm. In: Proceedings International Conference On Image Processing, ICIP 2001, Thessaloniki, Greece, vol. x, pp. 650–653 (2001)

39. Wang, L.: Abnormal walking gait analysis using silhouette-masked flow histograms. In: Proceedings 18th International Conference on Pattern Recognition, pp. 473–476. IEEE Computer Society, Washington, DC (2006)
40. Welsh, B., Ferrington, D.: Effects of closed circuit television surveillance on crime. Campbell Systematic Reviews 17, 110–135 (2008)
41. Wöllmer, M., Schuller, B., Eyben, F., Rigoll, G.: Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening. IEEE Journal of Selected Topics in Signal Processing 4(5), 867–881 (2010); special Issue on "Speech Processing for Natural Interaction with Intelligent Environments
42. Wu, C., Aghajan, H.: Model-based human posture estimation for gesture analysis in an opportunistic fusion smart camera network. In: Proceedings IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007, pp. 453–458 (2007)
43. Yue, Z., Zhou, S., Chellappa, R.: Robust two-camera tracking using homography. In: Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, vol. 3, pp. 1–4 (2004)