MULTI-CAMERA PEOPLE TRACKING WITH HIERARCHICAL LIKELIHOOD GRIDS

Lili Chen, Giorgio Panin and Alois Knoll
Fakultät für Informatik, Technische Universität München, Boltzmannstrasse 3, 85748 Garching bei München, Germany

Keywords: Edge-based background subtraction, Hierarchical likelihood grids, Oriented distance transform, Data association, Multi-view and Multi-target tracking.

Abstract: In this paper, we present a grid-based tracking by detection methodology, applied to 3D people tracking for multi-camera video surveillance. In particular, frame-by-frame detection is performed by means of hierarchical likelihood grids, using edge matching through the oriented distance transform on each camera view and a simple person model, followed by likelihood grids clustering in state-space. Subsequently, the tracking module performs a global nearest neighbor data association, in order to initiate, maintain and terminate tracks automatically. The proposed system can easily include additional features, such as color or background subtraction, it can be scaled to more camera views, and it can be used to track other items as well. We demonstrate it through experiments in indoor sequences, using a calibrated multi-camera setup.

1 INTRODUCTION

Nowadays automatic visual surveillance is becoming increasingly popular, because of its wide applications in indoor and outdoor environments with security requirements. Usually there are two major problems in an automatic surveillance system: one is to detect moving targets, and the other is to keep them tracked throughout the sequence. As the most representative application, detecting and tracking people is obviously the most challenging and attractive topic, due to people's huge variations in physical appearance, pose, movement and interaction. Therefore, people detection and tracking receives a significant amount of attention in the area of research and development.

Although some systems have been successfully developed towards this challenging task, it still remains difficult to detect and track multiple people precisely and automatically, only using generic models in a cluttered scene. This paper addresses the problem of employing a grid-based tracking-by-detection methodology, with a very simple shape model. The primary goal of our paper is to develop a fully automatic system for tracking multiple people in an overlapping, multi-camera environment, providing a 3D output robust to mutual occlusion between interacting people.

As a commonly used technique for segmenting out objects of interest, background subtraction has achieved a significant success in fixed camera scenarios. Most of the methods work by comparing color or intensities of pixels in the incoming video frame to the reference image (Stauffer and Grimson, 2000; Wren et al., 1997; Eng et al., 2004). However, it has the drawback of being susceptible to illumination changes, and provides a less precise localization. In contrast, we propose here an edge-based background subtraction, which employs the Canny edge map together with Sobel gradients, because edges are more precisely and stably localized, to a better extent in presence of illumination changes, so that the model has not to be adapted so often.

A second contribution of our system is frame-by-frame detection by means of hierarchical likelihood grids. This scheme, adapted from (Stenger et al., 2006), takes the advantage of multi-resolution grids that can, precisely and efficiently locate targets in cluttered scenes, without prior knowledge of their position. In particular, we compute the likelihood by edge matching through the oriented distance transform, which matches not only the location of edge points but also their orientation. And the likelihood is first computed on a coarse grid, then refined on the next level only the locations where likelihoods are higher than a given threshold. Subsequently, we perform state-space clustering on the high-resolution grid, in order to find the relevant peaks, possibly associated to people.

Chen L., Panin G. and Knoll A..
MULTI-CAMERA PEOPLE TRACKING WITH HIERARCHICAL LIKELIHOOD GRIDS.
DOI: 10.5220/0003316904740483
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2011), pages 474-483
ISBN: 978-989-8425-47-8

The third main issue consists in associating detected peaks to tracks, which is a classical data association problem, where a track can be updated by at most one measurement, and a measurement can be assigned to at most one track. Several approaches have been developed for this purpose, the most representative ones being (Fortmann et al., 1983; Reid, 1979); however, in place of complex methods, which require more complex models and parameter tuning, and further increase the computational complexity, our tracking module employs a Global Nearest Neighbor (GNN) approach in order to initiate, maintain and terminate tracks automatically.

The remainder of the paper is organized as follows: Section 2 reviews the state of the art and related work to our paper. Section 3 describes the general system overview with hardware setup and algorithmic flow of software. In Section 4, we provide the detailed detection procedure, including models, edge-based background subtraction, hierarchical grid evaluation as well as model-based contour matching and state-space clustering. Tracking by data association is presented in Section 5. The experimental results are discussed in Section 6. Finally, Section 7 summarizes the paper and proposes future development roads.

2 RELATED WORK

A vast amount of literature has been published on people detection and tracking. We can mainly classify it into four categories: region-based approaches, which are based on the variation of image regions in motion (Khan et al., 2001); feature-based (Wren et al., 1997; Fieguth and Terzopoulos, 1997; Li et al., 2003), that usually utilize information about color, texture, etc.; contour-based (Isard and Blake, 1996; Nguyen et al., 2002; Roh et al., 2007), that make use of the bounding contours to represent the target outline; and model-based methods (Gavrila and Davis, 1996; Andriluka et al., 2010) that explicitly require a 2D or 3D model of a person for tracking. However, a too detailed review of all the approaches is beyond the scope of our paper, therefore, in the following we will focus on people tracking-by-detection methodologies, more related to our work. There has been a number of literature on this approach (Okuma et al., 2004; Leibe et al., 2007; Wu and Nevatia, 2007), where detection of people in individual frame, as well as data association between detections in different frames, are the most challenging and ambiguous issues(Andriluka et al., 2008).

Template-based methods have yielded nice results for locating targets with no prior knowledge in a cluttered scene. In (Gavrila, 2000), the efficiency of this method is illustrated, by using about 4,500 templates to match pedestrians in images. The core is the idea of using a Chamfer distance measure, so that matching a template with the DT image results in a similarity measure, that is a smooth function of the template transformation parameters. Meanwhile this approach enables the use of an efficient search algorithm that locks onto the correct solution. However, if only computing the location of edge pixels without considering their orientation when computing distance transform, it inevitably leads to a high rate of false alarms in presence of clutter.

Another highlight of this system is the utilization of a template hierarchy, which is generated automatically from available examples, and formed by a bottom-up approach, using a partitioned clustering algorithm. It only searches locations where the distance measure is under a given threshold, so that a speed-up of three orders of magnitude, compared to exhaustive searching, is demonstrated.

This idea was taken further by (Stenger et al., 2006), that however does not build the template hierarchy (or tree) by bottom-up clustering, rather by partitioning a state-space represented with an integral grid. The grid is hierarchically partitioned as the search descends into each region, so that regions at the leaf-level define the finest partition. This method is demonstrated to be capable of covering 3D motion, even with self-occlusion. Unfortunately, both approaches need a very specific model, only valid for a specific target.

Once the measurements have been obtained from the frame-by-frame detection, data association can be applied to solve the problem of measurement to track assignment. A simple nearest-neighbor approach(Bar-Shalom and Fortmann, 1988) uses only the closest observation to any predicted state in order to perform the measurement update, and it is commonly used for MTT systems because of its fast computation. More complex approaches, such as Joint Probabilistic Data Association Filter (JPDA) (Fortmann et al., 1983) and Multiple Hypothesis Tracking (MHT) (Reid, 1979) solve this problem by maintaining multiple hypotheses, until enough measurements can be collected to resolve the ambiguity. In particular, (Fortmann et al., 1983) combines all of the potential measurements into one weighted average, before associating it to the track, in a single update. By contrast, (Reid, 1979) calculates every possible update hypothesis, with a track, formed by previous hypotheses associated to the target. Both methods are known to be quite complex, and require a careful implementation in terms of parameters; in particular, the

latter cannot avoid the drawback of an exponentially growing computational complexity, with the number of targets and measurements involved in the resolution situation, so that sub-optimal solutions must be sought (Cox and Hingorani, 1996).

3 SYSTEM OVERVIEW

In this section, we describe the hardware setup and present an overview of our tracking system, which will be discussed in more detail afterwards.

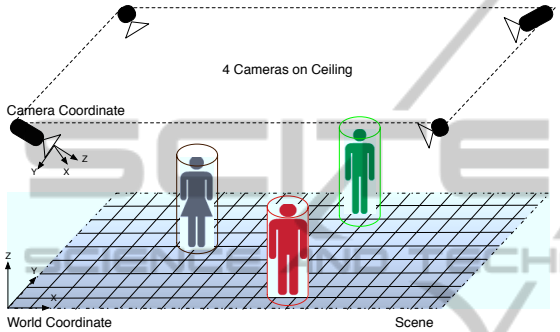


Figure 1: Hardware setup.

The overall setup is depicted in Figure 1. Four uEye usb cameras, with a resolution of 752×480, are mounted overhead on the corners of the ceiling, each of them observing the same 3D scene synchronously from different viewpoints. Furthermore, all the four cameras are connected to one multi-core PC. A necessary step before being able to get accurate 3D information, is calibration of the intrinsic and extrinsic camera parameters, that we perform with the Matlab Calibration Toolbox [1], with respect to a *world* coordinates system placed on the floor.

The detection and tracking software is designed and implemented in the *OpenTL* framework[2] (Panin et al., 2008; Panin, 2011), which is a structured, general purpose architecture for model-based visual tracking. We provide the block diagram of our tracking system in Figure 2, that consists of two main processing modules. Offline, we use a certain number of background frames to learn the background model. Moreover, grid states are sampled for each level, and the silhouettes are generated by projecting the external contours of the cylinder shape and keeping, for each contour and each camera view, a list of pixel positions and normals. Online, we have three main sub-modules: pre-process, detection and tracking.

[1] http://www.vision.caltech.edu/bouguetj/calib_doc/
[2] http://www.opentl.org

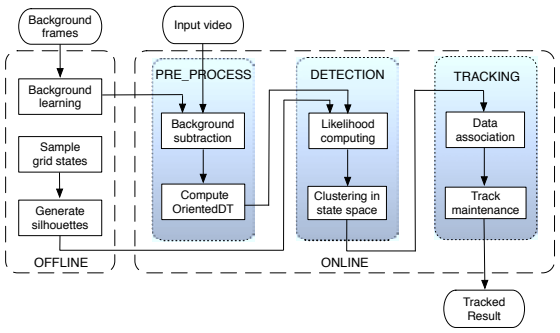


Figure 2: Block diagram of the tracking system.

In the pre-process part, for each camera view foreground contours are segmented by edge-based background subtraction, using the learned model. Afterwards, we compute an oriented distance transform onto this image, in order to match, for each template, both the location and the orientation of its contours. In particular, the oriented DT is efficiently computed over a finite set of orientations, so that the image is sampled over parallel scan lines that are precomputed. The advantage of using both edge position and orientation, during background subtraction as well as template matching, is a strong reduction of false alarms, i.e. false edge matching, that would arise when using only positional information.

Detection part first computes the likelihoods by matching projected templates and oriented DT for each camera view, where the likelihoods are computed on the coarse grid firstly, then refined on the next resolution only the locations where the likelihood is higher than a given threshold, the joint likelihoods can simply be multiplied then. The object-level measurements, or target hypotheses, are obtained by means of likelihood grid clustering, that is performed by Gaussian filtering of the high-resolution grid, and local maxima detection. Finally, the tracking module performs measurement-to-target association with the Global Nearest Neighbor approach, in order to initiate, maintain and terminate tracks automatically.

4 PEOPLE DETECTION

In this section, we provide more details about people detection, that serves as one of our key building blocks for our system.

4.1 Construction of Template Hierarchy

The idea to construct a template hierarchy is inspired by the paper (Stenger et al., 2006), as well as by the system developed by (Gavrila, 2000), extended

to multiple views, multiple targets, and with a more general template.

Assuming there are L levels of search, the state space is partitioned with a coarse-to-fine strategy. A graphical illustration is shown in Figure 3.

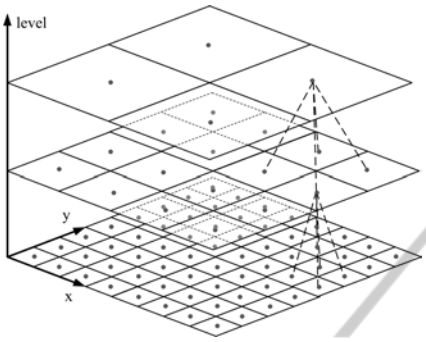


Figure 3: Grid based state space with hierarchical partition.

Each discrete region $\{R^{i,l}\}_{i=1}^{N_l}$, where N_l is the number of cells at level l, is sampled at its center, before the template hierarchy is generated. Meanwhile, we connect regions at a child level with its parent cell, by computing the nearest-neighbor in state-space, as well as its nearest neighbors within the same level, as it will be described in Section 4.4, in order to smooth the grid likelihoods.

After sampling the grid, templates are generated by rendering the 3D model at each state, under the respective camera projection. The model chosen in our approach is a simple cylinder, undergoing (x,y) translation on the floor, while its silhouette is generated by projecting the external contour. An example is shown in Figure 4, while a partial view of the hierarchy of silhouettes is depicted in Figure 5.

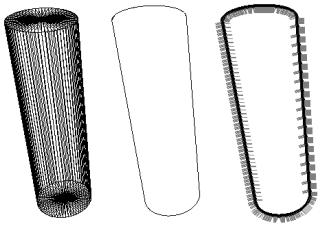


Figure 4: Our model. (a) Discretized cylinder. (b) Projected external contour. (c) Silhouette with normals.

For each silhouette, the position of each point as well as its normal are collected, as it will be described further in Section 4.3. As already emphasized, both grid sampling and template hierarchy generation are performed offline.

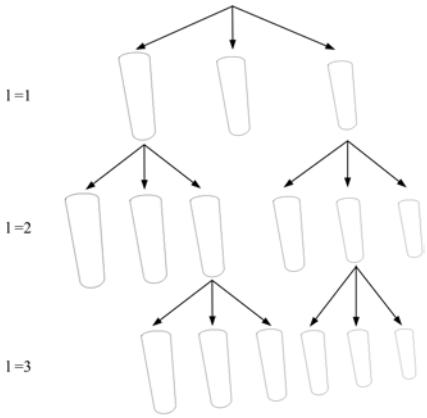


Figure 5: Hierarchy with silhouette of cylinder.

4.2 Background Learning and Foreground Segmentation

In order to match the image data with the templates, we first apply an edge-based background subtraction.

This approach can be divided into two phases: background learning (offline) and foreground segmentation (online). In the first phase, we utilize a certain number N of frames without people, in order to learn the background model. Let $\Theta_b(t), G_{bx}(t), G_{by}(t)$ respectively be the Canny edge map, and Sobel x-gradient and y-gradient images, detected at frame $I_b(t)$. The Canny map Θ_b is accumulated by binary OR, from frame $\Theta_b^{(I)}(1), \ldots, \Theta_b^{(I)}(N)$, while Sobel gradients are accumulated in a running average over the same frames. At the end, we normalize the accumulated Sobel image

$$G_{bx}^2 + G_{by}^2 = 1, \ \forall (x,y) \tag{1}$$

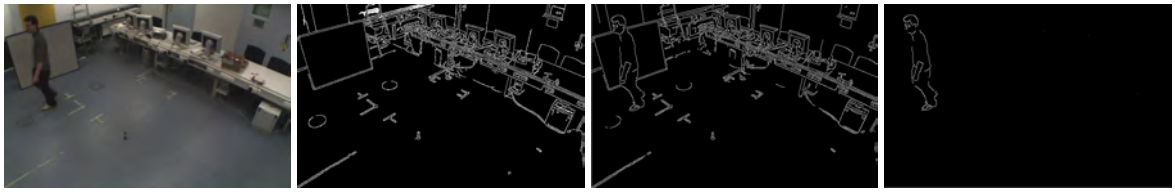
Subsequently, standard distance transform is applied to the accumulated background Canny map, and thresholded to a few pixels, providing a binary mask $\Theta_{DT} \in \{0,1\}$, where potential background edges are found.

Online, from the foreground Canny map and Sobel gradients $\Theta_f(t), G_{fx}(t), G_{fy}(t)$ of camera frame $I_f(t)$, we test the position and orientation of each edge pixel: edges $\Theta_f(t) \neq 0$ that lie near to a background edge $\Theta_{DT} \neq 0$ are candidate for removal.

Then, we further test these edges for orientation with the Sobel masks, and if the scalar product is higher than another threshold θ

$$\frac{G_{bx}G_{fx} + G_{by}G_{fy}}{\sqrt{G_{fx}^2 + G_{fy}^2}} > \theta \tag{2}$$

the point is removed from $\Theta_f(t)$. Figure 6 shows an example of this procedure: as we can see, the result-



(a) (b) (c) (d)

Figure 6: Edge-based background subtraction. (a) Original frame. (b) Learned background model. (c) Unsegmented foreground edge. (d) Segmented foreground edge.

ing edge map robustly preserves the person contours, while discarding most of the background edges.

4.3 Matching based on the Oriented Distance Transform

The next step is to match foreground edges with the model silhouettes. One possibility would be to use the Chamfer distance transform on the edge map, that is tolerant to small shape variations, and has already been applied in several works, such as (Borgefors, 1988; Gavrila, 2000). However, in case of images with considerable clutter, a significant rate of false alarms would be present. This problem can be reduced by matching not only the location of edge points, but also their orientation (Olsen and Huttenlocher, 1997).

Therefore, we propose here another approach, using the *oriented* distance transform. We define the oriented DT by scanning the edge image along parallel lines $\mathcal{L}_\gamma(a)$ through pixel $a = (x, y)$ for a given orientation γ, and repeat it for a finite set of N_γ directions $\Gamma = \{\gamma_i\}_{i=1}^{N_\gamma}$. The algorithm is illustrated in Figure 7.

Figure 7: Scanning single line for one direction. From left to right: Multiple single line scanning; Distance value to the nearest edge point on the line; Multiple scanning directions.

In particular, for each direction and each scan line, the oriented DT is a mono-dimensional function, looking for the nearest edge point b on either direction

$$b = DT_\gamma(a) = \min_{b \in \mathcal{L}_\gamma(a)} \|a - b\| \tag{3}$$

An example of oriented distance transform is shown in Figure 8.

Once oriented DTs are computed, template matching simply amounts to compute the likelihood, by summing up all values over the silhouette pixels, in the corresponding direction of the normal. To formalize the idea, a projected template s is represented by a set of pixel positions and normals $\{x_i, y_i, g_i\}_{i=1}^{N}$, obtained by re-projection through a 3×4 camera projection matrix P, where g_i selects the nearest $\gamma \in \Gamma$, from which the DT value will be taken. Therefore, the likelihood for state hypothesis s is given by:

$$P(z|s) = \exp\left(-\frac{1}{2NR^2}\sum_{i=1}^{N}\min\left(DT_{\gamma(g_i)}(x_i,y_i)^2, D_{max}^2\right)\right) \tag{4}$$

where $\gamma(g_i)$ denotes the closest available direction to the normal, and the sum is performed over all values $\{x_i, y_i, g_i\}_{i=1}^{N}$. R is the measurement standard deviation, and an outlier threshold is usually fixed at $D_{max} = 3R$, which is our validation gate for a more robust matching. Also notice that, in order to avoid problems with different scales, the sum is further normalized by N.

During the computation of likelihood, a coarse-to-fine search strategy is applied by evaluating it, at each level, only for locations where the parent cell likelihood is higher than a given threshold, which is usually obtained as the average likelihood (Stenger et al., 2006). For those cells where the parent likelihood is under the threshold, its value is simply inherited, thus saving a large amount of computation.

4.4 Likelihood Grid Clustering

In order to obtain the object-level measurements, or target hypotheses, after likelihood computation we employ a clustering procedure on the high-resolution grid, where each cluster is a local maximum, potentially corresponding to a person.

This approach is similar to mean-shift, but explicitly done on discrete states. First of all, a Gaussian filtering is applied to the grid, where the isotropic Gaussian corresponds to the filtering kernel. For each cell s_i within the grid, we take the nearest neighbor

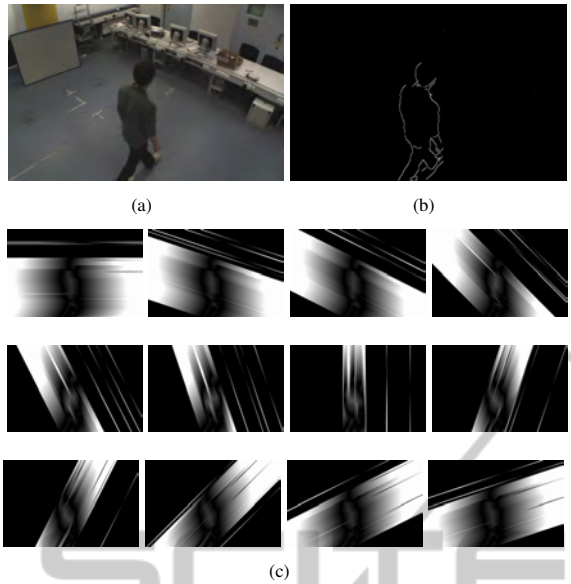


Figure 8: Results of oriented distance transform. (a) Input image. (b) Foreground edge map. (c) Oriented DT (at 12 discrete orientations).

s_j by looking at the connected states with distance $d_{i,j} = \|s_i - s_j\|$ up to a validation gate $D_{max} = 3\sigma_s^2$, where σ_s^2 is the measurement covariance in *state-space*, these neighbors are pre-computed in the off-line phase. For each neighbor, the Gaussian weight is also pre-computed by

$$W_{i,j} = exp(-\frac{d_{i,j}^2}{2\sigma_s^2}) \tag{5}$$

the computed weights are also normalized to 1, so that the smoothed likelihood for state cell s_i is given by

$$P(z|s)_{weighted}(i) = \sum_{i,j} W_{i,j} \cdot P(z|s)(j) \tag{6}$$

Subsequently, local maxima are detected (within the same neighborhood), to obtain the target hypotheses, or measurements. The final step will be to associate these hypotheses to tracks, as it will be described in next section.

5 MULTIPLE PEOPLE TRACKING

In this section we deal with the problem of multi-target tracking, by associating measurements obtained from our detector to individual tracks, also performing automatic track initiation and termination.

In particular, our track management follows a strategy indicated in (Bar-Shalom and Li, 1995):

- *Track Initiation.* In case of new targets entering into the scene, they will generate measurements that are too far from the existing targets, and therefore can be used to start new tracks. In this case, they are labeled with a unique ID, and a counter for the number of consecutive, successful detections for this target is also initialized to 1.
- *Track Maintenance.* During tracking, a target is successfully detected whenever the data association algorithm provides one valid measurement for it, so its counter is increased up to a maximum value (which can be taken as a confirmation time), while in case of misdetection it will be decreased. Those targets which are successfully detected over the confirmation time, can be considered as stable targets and maintained by the algorithm. In this way, if a target is misdetected for a few frames in case of occlusion, it can still be recovered until the counter goes to 0.
- *Track Termination.* When a target exits the scene, or after occlusion for a too long time, its misdetection counter goes to 0, and its track is terminated.

A pseudo-code of the whole procedure is shown in Algorithm 1, where the GNN algorithm is called in (line 25).

The data association problem consists in deciding which measurement should correspond to which track. Although our detection algorithm is fairly robust, it is also not person-specific, and therefore in a small indoor environment there are always ambiguities, arising from neighboring targets, as well as from missing detections and false alarms caused by background clutter. To this respect we employ the Global Nearest Neighbor (GNN) approach, that gives a good solution for this problem (Konstantinova et al., 2003), while requiring relative low computational cost.

The first step of the GNN is to set up a distance (or cost) matrix: assuming that, at time t, there are M existing tracks and N measurements, the cost matrix is given by

$$D = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M1} & d_{M2} & \cdots & d_{MN} \end{pmatrix} \tag{7}$$

where d_{ij} is the Euclidean distance between track i and measurement j, and $i = 1,2,\ldots,M; j = 1,2,\ldots,N$. In particular, d_{ij} is set to ∞ if it exceeds the validation gate, which is a circle with fixed radius around the predicted position, eliminating unlikely observation-to-track pairs. Moreover, it is commonly required that a target can be associated with at most one measurement (none, in case of misdetection), and

a measurement can be associated to at most one target (none, in case of false alarms).

The GNN solution to this problem is the one that maximizes the number of valid assignments, while minimizing the sum of distances of the assigned pairs. To this aim, we adopt the extended Munkres' algorithm (Burgeois and Lasalle, 1971), where the input is the cost matrix D, and output are the indices (row, col) of assigned track-measurement pairs.

Algorithm 1: Track management with GNN.

```
if nMeasurements = 0 then
    for i = 0 to nTargets do
        DecreaseCounter(target[i]);
        if Counter(target[i]) > 0 then
            MaintainTarget(target[i]);
        else
            TerminateTarget(target[i]);
        end if
    end for
else
    if nTargets = 0 then
        for j = 0 to nMeasurements do
            newTarget = CreateTarget(meas[j]);
            ResetCounter(newTarget);
        end for
    else
        for i = 0 to nTargets do
            for j = 0 to nMeasurements do
                D(i, j) = Distance(target[i], meas[j]);
                if D(i, j) > ValidGate then
                    D(i, j) = ∞;
                end if
            end for
        end for
        (i ↔ j) = GNN(D);
        for i = 0 to nAssocTargets do
            if D(i, j(i)) ≤ ValidGate then
                MoveTarget(target[i], meas[j]);
                IncreaseCounter(target[i]);
                if Counter(target[i]) > MaxC then
                    Counter(target[i]) = MaxC;
                end if
            else
                DecreaseCounter(target[i]);
                if Counter(target[i]) = 0 then
                    TerminateTarget(target[i]);
                end if
            end if
        end for
        for j = 0 to nUnassocMeas do
            newTarget = CreateTarget(meas[j]);
            ResetCounter(newTarget);
        end for
    end if
end if
```

6 EXPERIMENTAL RESULTS

We evaluated the proposed algorithms through pre-recorded video sequences, with multiple people entering and leaving the scene, as well as interacting with each other. The sequences have been simultaneously recorded from four cameras, as described in Section 3, with a resolution of (752×480), and a frame rate of 25 fps.

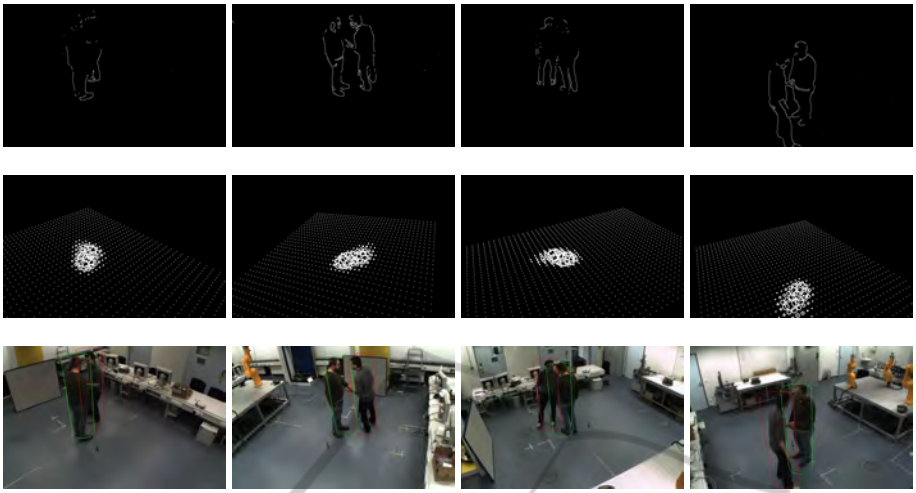
Before carrying out detection and tracking, state grids are set up at all levels, respectively 10×10, 20×20 and 40×40 from the coarsest to the finest, resulting in a total of 2100 grid cells, and the same amount of silhouette templates are sampled off-line. Since the area of interest is $(6m \times 4.2m)$, the corresponding grid on the finest level has a resolution of $(150mm \times 105mm)$.

Our current implementation of the oriented distance transform uses 12 discrete orientations, ranging from 0 to π. As it computes each orientation separately, they overall require about 0.25 sec/frame for four images, whereas a single, standard distance transform is computed in 0.1 sec/frame. Therefore, the speed of our oriented distance transform is acceptable and reasonable in comparison with standard distance transform. And the subsequent matching is done very quickly for each hypothesis.

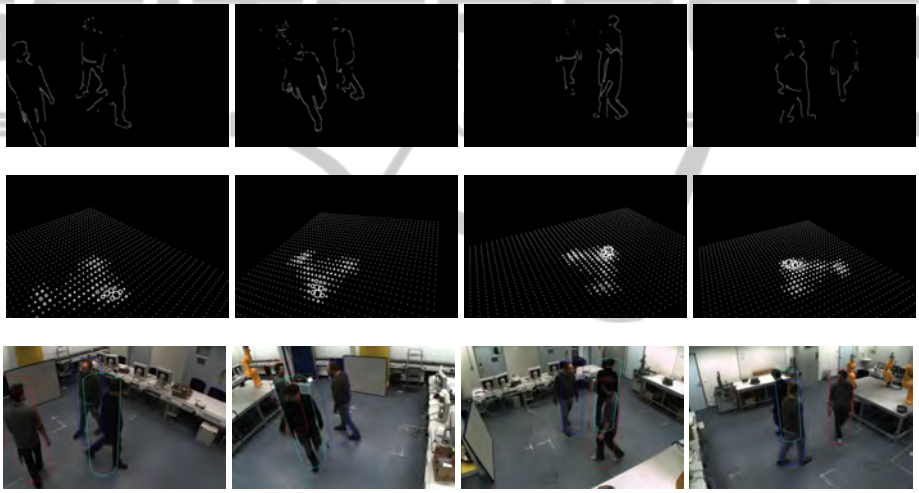
Figure 9 shows qualitative tracking results in a multi-camera environment, with a complex background. In particular, the top row shows foreground edges after edge-based background subtraction. Here 30 frames have been used for background learning, where the threshold θ mentioned in Eq. (2) is set to 0.9. The middle row shows likelihood values onto the finest grid, and the bottom row shows the corresponding tracking results after data association, with the projected cylinder silhouettes.

During data association, we keep a confirmation time of 10 frames (which is the maximum value for the consecutive detections counter) for keeping or removing tracks. As can be seen from the results, there are situations with significant occlusion from one or more views. For instance, at frame 345, each two targets are occluded from some views, however, since for the same pairs there are no occlusions from another camera view, all targets are successfully detected, thanks to the robustness of multi-camera fusion and oriented DT matching. The system also successfully handles targets entering and leaving the scene.

In order to better evaluate the performance of our system, we manually label the ground truth data for our sequences, and compare the results of our tracker, both in terms of position accuracy and robustness of



(a) Frame 185



(b) Frame 345

Figure 9: Performance of 3D people tracking. Shown are edge-based background subtraction, likelihood grids, and the corresponding tracking results, on four camera views.

detection. Ground truth trajectories, labeled on the finest grid, are depicted in Figure 10, where we can see the challenges due to targets that keep close most of the time, with mutual interactions and position exchanges.

Figure 11 shows the (X,Y) position errors of our tracking system. Because of the above mentioned occlusions and dynamics, for each target the system temporarily loses track, and recovers it again shortly afterwards. That happens about 4-5 times per target over the 550 frames of sequence, leading to several sub-tracks with different IDs, as shown in Figure 11 by the green boxes.

Overall these results indicate that, despite the cluttered situation, position errors are considerably low for all people, being most of the time under 100-150mm, that corresponds to one cell of the high-resolution grid. This is because of the local edge-based matching which, despite the simplicity of the model, is more precise with respect to global statistics such as color histograms (Stillman et al., 1998), or histograms of oriented gradients (Dalal and Triggs, 2005).

The execution time of the whole tracking procedure is currently 2 FPS, on a desktop PC with Intel Core 2 Duo CPU (1.86 GHz), 1GB RAM and an Nvidia GeForce 8600 GT graphic card.

7 CONCLUSIONS

In this paper, we presented a novel system for multi-

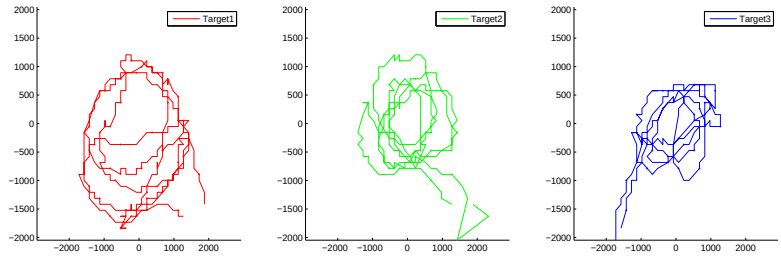


Figure 10: Ground-truth trajectories, sampled on the discretized grid (high-resolution).

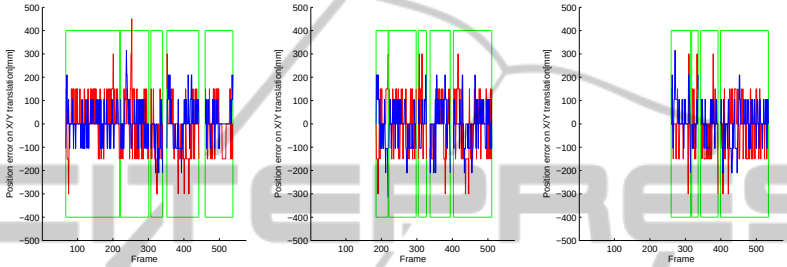


Figure 11: Position error on X(red) and Y(blue) in world coordinates. From left to right are shown target 1, 2 and 3. The green boxes correspond to sub-tracks estimated by our system.

ple people tracking in a multi-camera environment, using a grid-based tracking by detection methodology. A template hierarchy is constructed off-line, by partitioning the state space. And frame-by-frame detection is performed by means of hierarchical likelihood grids and clustered on the finest level, followed by data association through the GNN approach. Moreover, edge-based background subtraction has been proposed for foreground segmentation, which is quite robust to illumination changes, together with an oriented distance transform, matching the silhouette templates by taking gradient orientations into account, thus significantly reducing the rate of false alarms. Our system initiates, maintains and terminates tracks in a fully automatic way. Experimental results over the video sequences also show that our proposed system deals fairly well with mutual occlusions.

As a future work, this system can be easily extended to include additional features, such as color or motion, also can be scaled to more camera views, as well as being used for tracking different objects, for example 3D indoor tracking of flying *quadrotors*. In addition, the individual components can still be further optimized, both with respect to speed and performance, graphics hardware is possibly need to be exploited. Moreover, we plan to address the issue of heavy occlusions between people, taking place for longer periods. Re-identification after occlusions are going to be done by using more specific features, such as color or texture.

Besides these straightforward improvements, we also plan to test and extend our system to more challenging scenarios, such as outdoor tracking with multiple models (such as people and cars), as well as people tracking on mobile robots, with a non-static background and viewpoint.

REFERENCES

Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *IEEE conference on Computer Vision and Pattern Recognition(CVPR)*.

Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and data association*. Academic Press, San Diego.

Bar-Shalom, Y. and Li, X. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing.

Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):849–865.

Burgeois, F. and Lasalle, J. C. (1971). An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14:802–806.

Cox, I. J. and Hingorani, S. L. (1996). An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):138–150.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, Washington, DC, USA. IEEE Computer Society.

Eng, H., Wang, J., Kam, A., and Yau, W. (2004). A bayesian framework for robust human detection and occlusion handling using a human shape model. In *International Conference on Pattern Recognition*, volume 2, pages 257–260.

Fieguth, P. and Terzopoulos, D. (1997). Color-based tracking of heads and other mobile objects at video frame rates. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 21–27, San Juan, Puerto Rico.

Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184.

Gavrila, D. M. (2000). Pedestrian detection from a moving vehicle. In *Proc. of European Conference on Computer Vision*, pages 37–49, Dublin, Ireland.

Gavrila, D. M. and Davis, L. S. (1996). 3-d model-based tracking of humans in action: a multi-view approach. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 73–80, San Francisco.

Isard, M. and Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. In *Proceedings of the European Conference on Computer Vision*, pages 343–356, Cambridge, UK.

Khan, S., Javed, O., Rasheed, Z., and Shah, M. (2001). Human tracking in multiple cameras. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, pages 331–336, Vancouver, Canada.

Konstantinova, P., Udvarev, A., and Semerdjiev, T. (2003). A study of a target tracking algorithm using global nearest neighbor approach. In *Proceeding of International Conference on Computer Systems and Technologies*.

Leibe, B., Schindler, K., and Gool, L. V. (2007). Coupled detection and trajectory estimation for multi-object tracking. In *International Conference on Computer Vision*.

Li, L., Huang, W., Gu, I. Y. H., and Tian, Q. (2003). Foreground object detection from videos containing complex background. In *Proceedings of the 11th ACM International Conference on Multimedia*, pages 2–10.

Nguyen, H. T., Worring, M., van den Boomgaard, R., and Smeulders, A. W. M. (2002). Tracking nonparameterized object contours in video. *IEEE Trans. Image Process*, 11(9):1081–1091.

Okuma, K., Taleghani, A., Freitas, N. D., Little, J., and Lowe, D. (2004). A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*.

Olsen, C. F. and Huttenlocher, D. P. (1997). Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:103–113.

Panin, G. (2011). *Model-based visual tracking: the OpenTL framework*. Wiley-Blackwell. (to appear).

Panin, G., Lenz, C., Nair, S., Roth, E., Wojtczyk, M., Friedlhuber, T., and Knoll, A. (2008). A unifying software architecture for model-based visual tracking. In *IS&T/SPIE 20th Annual Symposium of Electronic Imaging*, San Jose, CA.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):843–854.

Roh, M. C., Kim, T. Y., Park, J., and Lee, S. W. (2007). Accurate object contour tracking based on boundary edge selection. *Pattern Recognition*, 40(3):931–943.

Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):747–757.

Stenger, B., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1372–1384.

Stillman, S., Tanawongsuwan, R., and Essa, I. (1998). A system for tracking and recognizing multiple people with multiple cameras. In *In Proceedings of Second International Conference on Audio-Visionbased Person Authentication*, pages 96–101.

Wren, C., Azarbayejani, A., Darrel, T., and Pentland, A. (1997). Pfinder, real time tracking of the human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

Wu, B. and Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266.