Computer Vision Winter Workshop 2006, Ondřej Chum, Vojtěch Franc (eds.)
Telč, Czech Republic, February 6–8
Czech Pattern Recognition Society

3D Template-Based Single Camera Multiple Object Tracking

Michal Juza, Karel Marik, Jiri Rojicek, and Petr Stluka

Honeywell Prague Laboratory
{michal.juza, karel.marik, jiri.rojicek, petr.stluka} @honeywell.com

Abstract

This paper describes a 3D template-based tracking method that allows simultaneous tracking of multiple objects of different type. The assumption is that movements of all objects are constrained to a ground plane so that the tracking functionality can be accomplished using just one properly calibrated camera.

The tracking algorithm is based on particle filtering where each particle represents one hypothetical configuration of the scene. Tracked objects are modeled by instances of 3D templates, positions and dimensions of which are continually updated from frame to frame.

A novel method for initialization of new objects has been developed that can easily be adopted in any particle-based approach, where the measurement step is done on the pixel level.

Numerous experiments have been conducted that indicate the presented approach copes with occlusions in an efficient way. The results for scenes of various complexity also demonstrate that the real-time performance can be achieved on a standard PC.

1 Introduction

Object tracking is an area of active research in computer vision. There are many applications utilizing tracking including surveillance, gesture recognition, smart rooms, vehicle/human tracking, etc. The tracking of objects is a challenging problem due to the presence of noise, occlusion, clutter and changes in the scene. A variety of tracking algorithms have been proposed and implemented to overcome these difficulties.

Model-based tracking algorithms incorporate a priori information about the objects they are tracking. Models of 2D body shape [22, 10], 3D joint models of human [18, 17, 14] or simple 3D static objects [11, 27] are used for object modeling.

Many methods are proposed to track objects using single camera in 2D [10, 23, 1, 7, 6], while the others utilize spatial information provided as combined input from multiple cameras [9, 20, 21]. There are also methods which exploit 3D information from single calibrated camera assuming that objects are moving along known plane [27]. The 3D position of the object can be inferred from 2D position of object and 3D template along with the camera model and the ground plane constraint.

Different types of trackers are used for object tracking: Kalman filter [21], mean-shift [24, 5], etc. Recently different variations of particle filtering are very popular in computer vision [2, 4, 14, 17, 23, 15, 25, 1, 11]. A Tutorial on Particle Filters for on-line nonlinear non-Gaussian Bayesian tracking can be found in [2].

Various features may be measured on objects for the tracking. Color of the object is used as target model in [13]. Edge-based measurement can be found in [14]. Combination of shape and color features is presented in [23]. Combination of color and edge orientation histogram ordered in a cascade can be found in [25].

The presented approach utilizes two specific 3D models of a car and a human that are used to model moving objects on a ground plane monitored by a single calibrated camera. State of the tracker is estimated by a particle filter and measurements on the input image are based on binary matching of the templates of object shapes that have been preprocessed by a motion detection module.

The paper is structured as follows. Section 2 describes basic principles of the tracking algorithm. Object templates, their parameters and dynamics are summarized in Section 3, while Section 4 introduces a new probabilistic approach to addition and deletion of objects. Results of experiments performed on the testing sequences are demonstrated in Section 5.

2 Tracker

Tracking can be seen as a recursive calculation of belief degree that an object is in the state x_t at time t given measurements on input images $Q_t = \{q_1, \ldots, q_t\}$ up to time t. The objective of tracking is to recursively estimate the posterior probability density function:

$$p(x_t|Q_t). \tag{1}$$

In Bayesian framework it can be done in two steps: prediction (2) and update (3).

$$p(x_t|Q_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Q_{t-1})dx_{t-1}. \tag{2}$$

$$p(x_t|Q_t) = \frac{p(q_t|x_t)p(x_t|Q_{t-1})}{p(q_t|Q_{t-1})}, \tag{3}$$

where $p(q_t|x_t)$ is observation likelihood and Markovian first order model is assumed so that $p(x_t|X_{t-1}) = p(x_t|x_{t-1})$,

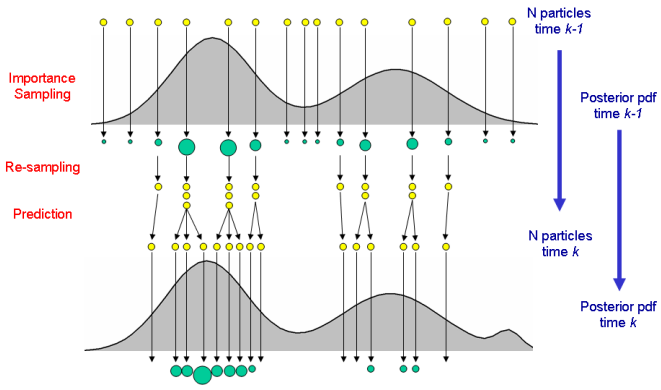

Figure 1: One step of particle filtering algorithm. Blob center represents sample value, size depict weight of sample.

where $X_{t-1} = \{x_1, \ldots, s_{t-1}\}$ is history of states up to time $t-1$. Overview of methods which can be used under different presumption to solve (1) can be found in [2].

2.1 Particle filter

Particle filtering is suboptimal method for recursive solving of (1). Its advantage is that there are no requirements on form of $p(x_t|Q_t)$. Key idea of particle filtering is to represent the required posterior density function (1) by a set of random samples with associated weights and to compute estimates based on these samples and weights. Particle filtering consists of three recurrent steps (see Figure 1):

1. Importance sampling - Weights of all particles are determined according to measurement on image.

2. Re-sampling - New set of N particles is generated based on the weights of particles.

3. Prediction - Particles are drifted in agreement with system model and some noise is added as well.

2.2 Structure of particles

There are two basic options for multiple object tracking. One can use a number of independent trackers that correspond to individual objects, or just one complex tracker for all objects in the scene. The presented approach is based on the second concept. Then each particle p_i is a collection of several objects of possibly different types $p_i = \{o_{1,i}, o_{2,i}, \cdots, o_{R_i,i}\}$, where R_i denotes actual number of distinct objects in the particle p_i. Number R_i of objects in the scene may vary for different particles. Each object $o_{j,i}$ is described by several parameters like position in 3D world coordinates, velocity etc. Object parameters will be described in detail in Section 3.

2.3 Importance sampling

Weight of each particle is determined based on the measurement on the input image. We assume binary motion images as input to our method. Weights of the particles are computed according to their ability to explain the current scene that is reflected in the input image. As described above, each particle p_i consists of R_i object instances (e.g. vehicles, humans). Weight of one particle is computed in following steps:

$q_i \setminus v_i$	0	1
0	0	1
1	1	0

Table 1: Matching penalty

1. Particle projection and matching - scene configuration coded in a particle p_i is projected from 3D coordinates to 2D image. This synthesized virtual scene v_i has the same size like the input image, which allows to compute the matching penalty m_i between the input image q and the virtual scene v_i in a straightforward way:

$$m_i = \sum_{w=1}^{W} \sum_{h=1}^{H} \text{diff}(q(w,h), v_i(w,h)), \tag{4}$$

   where W, H are image width and height, diff is logical XOR defined in Table 1.

2. Complexity penalty c_i is computed as follows:

$$c_i = \sum_{r=1}^{R(i)} \left[a_r + b_r e^{-\tau}\right], \tag{5}$$

   where a_r, b_r are static object parameters and τ is the period of time for which the given object is involved in given particle. Complexity penalty was introduced based on early experiments to avoid incorect placing more objects in very similar 3D position improving matching penalty m_i.

3. Weight computation - overall weight of the particle w_i is computed as follows:

$$w_i = e^{-\frac{m_i + c_i}{WH} s}, \tag{6}$$

   where W,H are dimensions of image and s is a scaling constant which controls greediness of the re-sampling.

2.4 Re-sampling

Not all particles are used in the prediction step. Particles are sampled according to their weights so that particles with a higher weight may be selected more than once, while some others may not be selected at all. Selection is done simply using cumulative weight of particles. Assume that we have indexed set of particles $P = \{p_1, \cdots, p_N\}$. Cumulative weight $w_{cum}(i)$ of particle p_i is given by:

$$w_{cum}(i) = \sum_{j=1}^{i} w_i, \tag{7}$$

w_i is weight of particle p_i, as defined in 2.3.

New set $P^n = \{p_1^n, \cdots, p_N^n\}$ of N particles is generated from the current particle set P as follows. First, set of random numbers $\Gamma = \{\gamma_1, \cdots, \gamma_N\}$ is generated from uniform distribution $U[0, w_{cum}(N)]$. Second, new particles are selected such that: $p_i^n = p_{s_i}$, where index s_i can be determined from: $w_{cum}(s_i) > \gamma_i \wedge w_{cum}(s_{i-1}) < \gamma_i$.

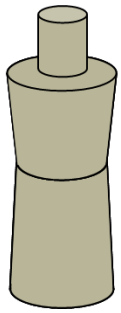

Figure 2: Template for human.

2.5 Prediction

Prediction consists of two separate steps:

1. Stochastic diffusion - Firstly objects are added or deleted in some particles during stochastic diffusion (for details see Section 4). Further the noise is added to some object parameters, as described in Section 3. Addition of Gaussian noise allows generation of new hypothesis about objects.
2. Deterministic drift - Parameters of all objects in each particle are updated according to template dynamics, which is described in the next section.

3 Templates

3D templates of a human and a vehicle were developed. Each template has a different set of parameters and also a different model of dynamics. Simplicity of the models was prefered because of lower computational effort.

3.1 Human

3D joint models of human have been already used for tracking, see for example [14]. Simplified 3D template (see Figure 2) has been found suitable for typical end-use applications of the presented approach. A complete state of the model can be described as follows:

$$\vec{o}_{human} = \{\vec{d}, \vec{l}, \vec{v}\},$$

where $\vec{d}$ is vector of body measurements (e.g. height, head radius, etc.), $\vec{l}$ is location of human in world coordinates and $\vec{v}$ is velocity vector. Model dynamics used in prediction step of the particle filter is:

$$\begin{aligned} \vec{d_t} &= \vec{d}_{t-1} + N(0, \sigma_d^2), \\ \vec{v_t} &= \vec{v}_{t-1} + N(0, \sigma_v^2), \\ \vec{l_t} &= \vec{l}_{t-1} + \vec{v}_t \Delta t, \end{aligned} \tag{8}$$

where $N(0, \sigma^2)$ is usual normal distribution with zero mean and variance σ_d^2, σ_v^2, resp.

3.2 Vehicle

The template used for vehicle modeling is shown in Figure 3. The template is designed to cover most types of common vehicles (sedan, pick-up, truck, etc.). In comparison to human template there is an additional parameter, namely orientation, which specifies vehicle orientation $\vec{\alpha}$. A complete state of the model can be described as follows:

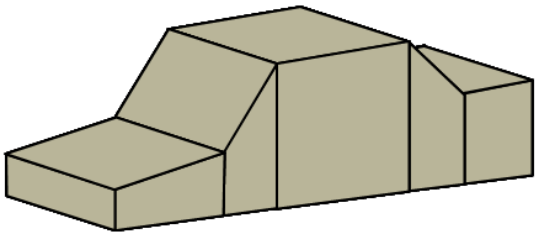

Figure 3: Template for vehicle.

$$\vec{o}_{vehicle} = \{\vec{d}, \vec{l}, \vec{\alpha}, \vec{\theta}, v\}$$

where $\vec{d}$ is vector of dimensions, $\vec{l}$ is location of the vehicle in world coordinates, $\vec{\alpha}$ is orientation the vehicle in ground plane, v is velocity of the vehicle and $\vec{\theta}$ is the angular velocity that characterizes car's turning.

Model dynamics used in the prediction step of particle filter is following:

$$\begin{aligned} \vec{d_t} &= \vec{d}_{t-1} + N(0, \sigma_d^2), \\ \vec{\theta_t} &= \vec{\theta}_{t-1} + N(0, \sigma_\theta^2), \\ \vec{\alpha_t} &= \vec{\alpha}_{t-1} + \vec{\theta}_{t-1} \Delta t, \\ v_t &= v_{t-1} + N(0, \sigma_v^2), \\ \vec{l_t} &= \vec{l}_{t-1} + v_{t-1} \Delta t \vec{\alpha}_t. \end{aligned} \tag{9}$$

4 Object addition and deletion

The overall accuracy of multiple object tracking techniques is influenced by object addition and deletion methods used. This topic is not frequently discussed in the literature, sometimes the manual initialization of new objects is assumed [16, 14, 17, 3]. Object initialization using unmatched motion cues is proposed in [9], which has disadvantage that one has to identify particular connected components in the input image. Appearance probability as a function of image coordinates is introduced in [8]. This probability distribution is dependent on concrete scene. Position is randomly sampled from uniform distribution in [11], while initialization based on color segmentation is used in [1]. In [26] a method for new object proposal is randomly selected from the following three methods: head detection on image cues, head detection from intensity and residue foreground analysis. Deletion of objects is usually done randomly.

We introduce novel statistics called Uncovered Object Histogram (UOH) that can improve the efficiency of new objects initialization. Based on this statistics we formulate addition of new object as minimization problem. Great advantage of UOH is that it can be computed very fast during importance sampling stage.

4.1 Uncovered Object Histogram

Uncovered Object Histogram (UOH) is 2D histogram of the same proportions like the input motion image. The value of the UOH in the position (w, h) is given by:

$$UOH(w,h) = \begin{cases} \sum_{i=1}^{N} \neg(v_i(w,h)), & \text{if } q(w,h) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Figure 4: (a) input image. (b) UOH.

where v_i is binary projection of particle p_i, q is input image and $\neg$ is logical unary operator NOT. Simply said, UOH expresses how many particles do not cover the input image at given position (m, n). Example of computed UOH can be seen in Figure 4. The right person is not tracked yet, hence UOH is clearly higher in this region than in area where the left person is situated.

4.2 Object Addition

In selected particles UOH is used for adding new objects to be tracked. New object is initialized in 3D so that decreases the overall sum of UOH as much as possible. Criterion for optimal initialization can be formulated in the following way:

$$o_{new} = \text{argmin}_o \sum_{w=1}^{W} \sum_{h=1}^{H} UOH_o(w, h), \qquad (11)$$

where UOH_o is defined by (10) but the object o is virtually added to all particles.

Object o_{new} is added randomly to particle p_i if:

$$rand_i < T_{add}, \qquad (12)$$

where $rand_i \in \text{U}[0, 1]$ is a random number and T_{add} is a predefined static threshold for objects addition.

4.3 Object Deletion

Situation when some object leaves the scene, or there is a wrongly placed template which does not correspond to any real object, must be covered as well. Object $o_{r,i}$ ($r - th$ object in particle p_i) is deleted from the given particle if:

$$rand_{r,i} < T_{del}, \qquad (13)$$

where $rand_{r,i} \in \text{U}[0, 1]$ is a random number and T_{del} is a predefined threshold for deletion of objects.

5 Experiments

Testing of the template-based concept has been conducted using ten different sequences that varied in both diversity and complexity. The moving objects were either humans, cars, or combinations of humans and cars in one scene. People were moving randomly or in a group. Cars were moving straight, crossing each other, or doing an U-turn. As a part of the testing, special attention was paid to occlusions that were present in a majority of sequences. Before applying the tracker all sequences were pre-processed by a background subtraction algorithm. Two specific approaches were applied with almost equivalent results: the algorithm of Stauffer and Grimson [19], and the background subtraction technique developed at University of Maryland [12].

Type of the scene	FPS	Accuracy
People walking randomly (no cars, up to 5 objects)	12.5	> 0.9
Cars only (up to 3 objects)	12.2	> 0.88
Groups of people (no cars, 5 and more objects)	13.6	> 0.92
Combined scenes (2 cars, 2 persons in average)	13.2	> 0.8

Table 2: Results

5.1 Testing Process

Ground truth information was generated for all sequences so that each sequence was provided with the following details:

- Number of human tracks in each sequence
- Number of vehicle tracks in each sequence
- Length of each track in terms of the number of frames, start and end of each track in terms of the frame number

Then the testing was done by an automated comparison of results against the ground truth information. Due to the probabilistic nature of the tracker, each sequence was evaluated six times and results averaged. All presented results were achieved on a computer with 2.8 GHz processor and 1GB memory.

5.2 Performance Criteria

Two criteria were used to measure the performance one in terms of computation speed (frames per seconds), the other in terms of algorithms accuracy. The algorithms accuracy was defined as a product of the tracking accuracy, which equals to the percentage of correctly tracked objects, and the classification accuracy defined as the percentage of objects modeled by the right template.

5.3 Results

The average results for each specific type of scene are summarized in Table 2. The computation speed was always above 12 fps and more or less stable, which indicates that the presented approach is suitable for real-time applications. The achieved accuracy differs more significantly from scene to scene. The best results were obtained for scenes without cars where people were walking either randomly or in a group typical accuracy was above 90%, sometimes above 95%. Figure 7 illustrates results for a scene with five people walking in a longer distance from camera. Almost all occlussions were correctly captured in the synthesized scene. A little worse results correspond to scenes with car objects. Scenes including cars only were modeled with a similar accuracy exceeding 88%, while in the most complex scenes with cars and people the accuracy decreased to 80%. Figure 6 shows a part of the sequence where two cars were crossing each other in the middle of the scene. Also in this case the oclussion was modeled correctly. Figure 6 illustrates tracking of two turning cars.

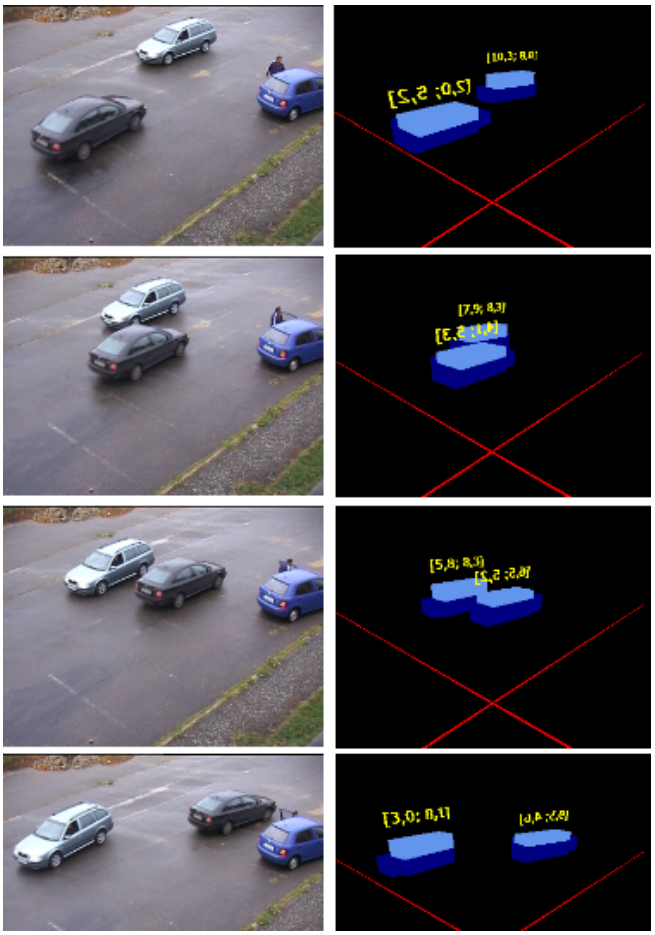

Figure 5: Tracking results on the cars sequence with oclussion. Yellow numbers in the synthesized scene (right) indicate the real world coordinates of each car.

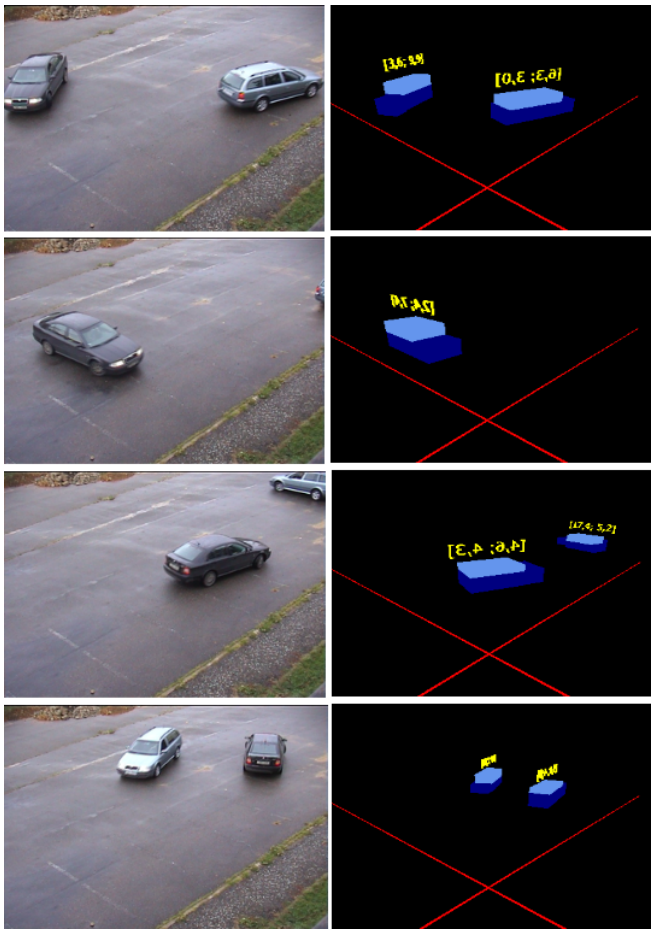

Figure 6: Tracking results for cars doing U-turn.

6 Conclusion

A model-based technology for tracking multiple objects using 3D templates has been presented. The core part of this concept is based on particle filtering, which provides updated estimates of the state vector comprising positions, dimensions and other parameters of all moving objects. Specific approach to probabilistic initialization of new objects has been proposed. The algorithm was tested on a number of sequences of different type that correspond to realistic scenarios and included cars and humans as visual objects to be tracked. The achieved results have confirmed the solution complies with real-time requirements and copes efficiently with oclussions.

References

[1] Zia Khan amd Tucker Balch and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(11):1805–1819, Nov. 2005.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. In *IEEE Transactions on Signal Processing, 50(2)*, pages 174–188, 2002.

[3] Cheng Chang, R. Ansari, and A. Khokhar. Multiple object tracking with kernel particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005.*, volume 1, pages 566 – 573, 2005.

[4] H.T. Chen and C.S. Fuh. Probabilistic tracking with adaptive feature selection. In *International Conference on Pattern Recognition*, pages II: 736–739, 2004.

[5] R.T. Collins. Mean-shift blob tracking through scale space. In *IEEE Computer Vision and Pattern Recognition*, pages II: 234–240, 2003.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[7] B. Han, Y. Zhu, D. Comaniciu, and L.S. Davis. Kernel-based bayesian filtering for object tracking. In *IEEE Computer Vision and Pattern Recognition*, pages I: 227–234, 2005.

[8] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. In *IEEE Computer Vision and Pattern Recognition*, pages I: 864–871, 2004.

[9] I. Haritaoglu, D. Harwood, and L. S. Davis. W4s: A real-time system for detecting and tracking people in 2 1/2-d. In *European Conference on Computer Vision*, 1998.

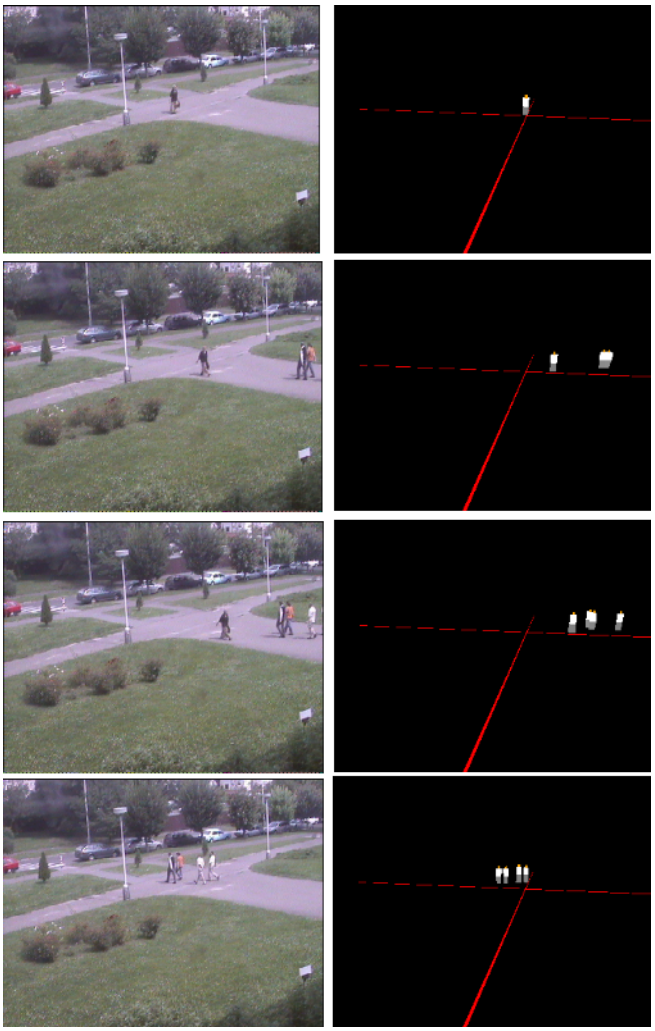

Figure 7: Tracking results for a group of people.

[10] H. S. Sawhney H.Tao and R. Kumar. A sampling algorithmfor tracking multiple objects. In *IEEE International Workshop on Vision Algorithms*, page 53, Corfu, Greece, 1999.

[11] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Int. Conf. Computer Vision*, volume 2, pages 34–41, 2001.

[12] K. Kim, T.H. Chalidabhongse, D. Harwood, and L.S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, June 2005.

[13] K. Nummiaro, E. Koller-Meier, and L.J. Van Gool. Object tracking with an adaptive color-based particle filter. In *German Pattern Recognition Symposium*, page 353 ff., 2002.

[14] E. Poon and D.J. Fleet. Hybrid monte carlo filtering: Edge-based people tracking. In *IEEE Workshop on Motion and Video Computing*, pages 151–158, 2002.

[15] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 786–793, Kauai, Hawaii, 2001.

[16] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages II: 702–718, 2000.

[17] H. Sidenbladh, M. J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages 702–718, Dublin, Ireland, 2000.

[18] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conf. on Computer Vision, ECCV2002*, pages 784–800, 2002.

[19] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, Fort Collins, CO, 1999.

[20] Black J. Ellis T.J. Multi camera image measurement and correspondence. *Measurement - Journal of the International Measurement Confederation*, 35:61–71, July 2002.

[21] Black J. Ellis T.J. Multi camera image tracking. *Image and Vision Computing*, 2005.

[22] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[23] Y. Wu and T.S. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *International Journal of Computer Vision*, 58(1):55–71, June 2004.

[24] C. Yang, R. Duraiswami, and L.S. Davis. Efficient mean-shift tracking via a new similarity measure. In *IEEE Computer Vision and Pattern Recognition*, pages I: 176–183, 2005.

[25] C. Yang, R. Duraiswami, and L.S. Davis. Fast multiple object tracking via a hierarchical particle filter. In *IEEE International Conference on Computer Vision*, pages I: 212–219, 2005.

[26] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE Computer Vision and Pattern Recognition*, pages II: 459–466, 2003.

[27] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *IEEE Computer Vision and Pattern Recognition*, pages II: 406–413, 2004.