A Bayesian Framework for Multi-cue 3D Object Tracking

J. Giebel[1], D.M. Gavrila[1], and C. Schnörr[2]

[1] Machine Perception, DaimlerChrysler Research and Technology,
Wilhelm Runge Str. 11, 89089 Ulm, Germany
{jan.giebel,dariu.gavrila}@daimlerchrysler.com
[2] Computer Vision, Graphics and Pattern Recognition Group,
Department of Mathematics and Computer Science, University of Mannheim,
68131 Mannheim, Germany
schnoerr@uni-mannheim.de

Abstract. This paper presents a Bayesian framework for multi-cue 3D object tracking of deformable objects. The proposed spatio-temporal object representation involves a set of distinct linear subspace models or Dynamic Point Distribution Models (DPDMs), which can deal with both continuous and discontinuous appearance changes; the representation is learned fully automatically from training data. The representation is enriched with texture information by means of intensity histograms, which are compared using the Bhattacharyya coefficient. Direct 3D measurement is furthermore provided by a stereo system.
State propagation is achieved by a particle filter which combines the three cues shape, texture and depth, in its observation density function. The tracking framework integrates an independently operating object detection system by means of importance sampling. We illustrate the benefit of our integrated multi-cue tracking approach on pedestrian tracking from a moving vehicle.

1 Introduction

Object tracking is a central theme in computer vision with applications ranging from surveillance to intelligent vehicles. We are interested in tracking complex, deformable objects through cluttered environments, for those cases when simple segmentation techniques, such as background subtraction, are not applicable.

This paper presents a probabilistic framework for integrated detection and tracking of non-rigid objects. Detections from an independent source of information are modeled as "mixture" of Gaussians and are integrated by two means: They control initialization and termination by a set of rules and serve as additional source of information for the active tracks.

To increase robustness three independent visual cues are considered. Object shape is used, since it is (quite) independent of the complex illumination conditions found in real world applications and efficient matching techniques exist to compare shape templates with images [6]. Texture distributions are modeled

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3024, pp. 241–252, 2004.

as histograms [14,15], which are particularly suitable for tracking since they are independent of object shape, invariant to rotation, scale, and translation, and easy to compute. Finally stereo measurements are integrated into the system.

In this work, tracking proceeds directly in 3D-space, which allows a more natural incorporation of real-world knowledge (e.g. kinematical properties of objects) and simplifies reasoning about occlusion and data association.

The outline of the paper is as follows: Section 2 reviews previous work. Our proposed multi-cue object representation is described in Section 3. It consists of two parts; the first deals with the spatio-temporal shape representation and the second relates to the texture model. The proposed particle filtering approach for multi-cue 3D object tracking is presented in Section 4. It integrates an independently operating external detection system. We illustrate our approach in Section 5 on the challenging topic of pedestrian tracking from a moving vehicle. Finally, we conclude Section 6.

2 Previous Work

Bayesian techniques are frequently used for visual tracking. They provide a sound mathematical foundation for the derivation of (posterior) probability density functions (pdf) in dynamical systems. The evolution of the pdf can in principle be calculated recursively by optimal Bayesian filtering. Each iteration involves a prediction step based on a dynamical model and a correction step based on a measurement model. Analytical solutions for the optimal Bayesian filtering problem are known only for certain special cases (e.g. Kalman filtering). For others, approximate techniques have been developed, such as extended Kalman [1], particle [2,4], and "unscented" filters [13]. In particular particle filters have become widespread, because of their great ease and flexibility in approximating complex pdfs, and dealing with a wide range of dynamical and measurement models Their multi-modal nature makes them particularly suited for object tracking in cluttered environments, where uni-modal techniques might get stuck and loose track.

Several extensions have been proposed to the early particle filter techniques, e.g. dealing with discrete/continuous state spaces [9,11], multiple target tracking [12,15,21], or multiple sources of information [10,17]. The latter has involved techniques such as importance sampling [10] or democratic integration [17,19], and have been used to combine visual cues such as edge and texture in a particle filter framework. Particle filters have furthermore been applied in combination with low-level [14,15], high-level [9], exemplar-based [18], or mixed-level [11] object representations.

In terms of representation, compact low-dimensional object parameterizations can be obtained by linear subspace techniques, e.g. using shape (PDMs) [3,9], or texture [20]. However, these methods have some limitations concerning the global linearity assumption: nonlinear object deformations have to be approximated by linear combinations of the modes of variation. They are not the most compact representations for objects undergoing complex (non-linear)

deformations, nor do they tend to be very specific, since implausible shapes can be generated, when invalid combinations of the global modes are used.

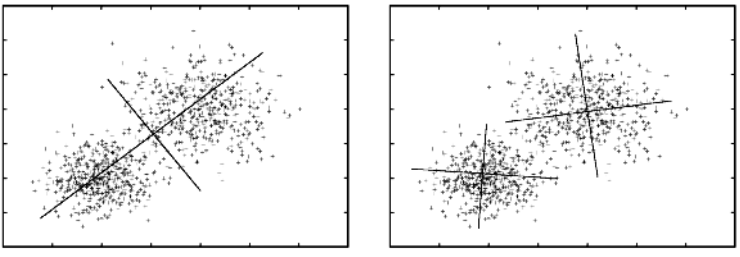

Fig. 1. Feature spaces: Linear and locally-linear feature spaces

Our approach, discussed in the next sections, builds upon the locally linear shape representation of [9] (see Figure 1). We extend this by a spatio-temporal shape representation, which does not utilize a common object parameterization for all possible shapes. Instead, a set of unconnected local parameterizations is used, which correspond to clusters of similar shapes. This allows our spatio-temporal shape representation to be fully automatically learned from training sequences of closed contours, without requiring prior feature correspondence.

We model texture by means of histograms similar to [14,15]. However, we do not rely on circular/rectangular region primitives, but take advantage of the detailed shape information to derive appropriate object masks for texture extraction. Furthermore, unlike previous work, we derive a 3D tracking framework also incorporating stereo measurements for added robustness.

Finally, our tracking framework integrates an independently operating object detection system by means of importance sampling.

3 Multi-cue Object Representation

3.1 Spatio-temporal Shape Representation

Dynamic point distribution models capture object appearance by a set of linear subspace models with temporal transition probabilities between them. This spatio-temporal shape representation can be learned automatically from example sequences of closed contours. See Figure 2. Three successive steps are involved.

Integrated registration and clustering: At first an integrated registration and clustering approach [8] is performed. The idea of integration is motivated by the fact, that general automatic registration methods are not able to find the physically correct point correspondences, if the variance in object appearance is too high. This is in particular the case for self occluding objects, when not all object parts are visible for all time. Our proposed approach therefore does not try to register all shapes into a common feature space prior to

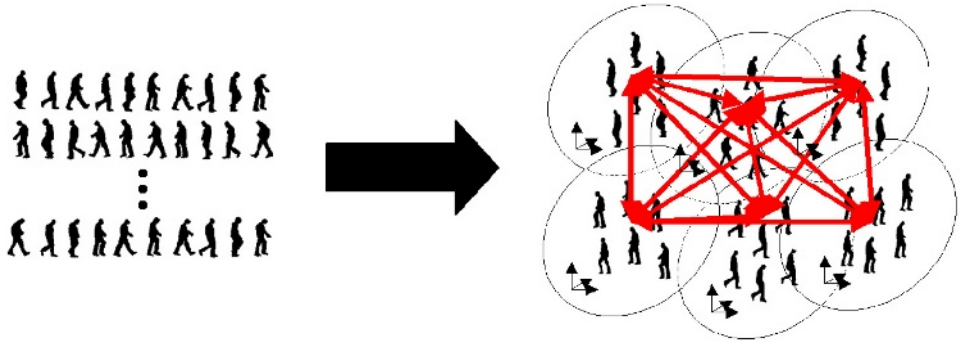

Fig. 2. Learning Dynamic Point Distribution Models

clustering. Instead the clustering is based on a similarity measure derived from the registration procedure. To be specific, the average distance between corresponding points after alignment. Only if this distance is lower than a user defined threshold, the shapes fall into the same cluster and the registration is assumed valid. For details, the reader is referred to [8].

Linear subspace decomposition: A principal component analysis is applied in each cluster of registered shapes to obtain compact shape parameterizations known as "Point Distribution Models" (PDMs) [3]. From the N^c shape vectors of cluster c given by their u- and v-coordinates

$$\mathbf{s}_i^c = (u_{i,1}^c, v_{i,1}^c, u_{i,2}^c v_{i,2}^c, ..., u_{i,n^c}^c, v_{i,n^c}^c), i \in \{1, 2, ..., N^c\} \tag{1}$$

the mean shape $\bar{\mathbf{s}}^{\mathbf{c}}$ and covariance matrix K^c is derived. Solving the eigensystem $K^c \mathbf{e^c}_j = \lambda_j^c \mathbf{e^c}_j$ one obtains the $2n^c$ orthonormal eigenvectors, corresponding to the "modes of variation". The k^c most significant "variation vectors" $E^c = (\mathbf{e}_1^c, \mathbf{e}_2^c, ..., \mathbf{e}_{k^c}^c)$, the ones with the highest eigenvalues λ_j^c, are chosen to cover a user specified proportion of total variance contained in the cluster. Shapes can then be generated from the mean shape plus a weighted combination of the variation vectors

$$\mathbf{s}^c = \bar{\mathbf{s}}^{\mathbf{c}} + E^c \mathbf{b}. \tag{2}$$

To ensure that the generated shapes remain similar to the training set, the weight vector $\mathbf{b}$ is constrained to lie in a hyperellipsoid about the subspace origin. Therefore $\mathbf{b}$ is scaled so that the weighted distance from the origin is less than a user-supplied threshold M_{max}

$$\sum_{l=1}^{k^c} \frac{b_l^{c2}}{\lambda_l^c} \leq M_{max}^2 \tag{3}$$

Markov transition matrix: To capture the temporal sequence of PDMs a discrete Markov model stores the transition probabilities $T_{i,j}$ from cluster i to j. They are automatically derived from the transition frequencies found in

the training sequences. An extension for covering more complex temporal events (e.g. [5]) is conceivable and straightforward.

3.2 Modeling Texture Distributions

The texture distribution over a region $\mathbf{R} = (u_1, v_1, u_2, v_2, ..., u_{n_R}, v_{n_R})$, given by its n_R u- and v-coordinates, is represented by a histogram $\theta_{\mathbf{R}} = \{\theta^r\}_{r=1,...,m}$, which is divided into m bins. It is calculated as follows

$$\theta^r_{\mathbf{R}} = \frac{1}{n_R} \sum_{i=1}^{n_R} \delta(h(u_i, v_i) - r), \tag{4}$$

whereas $h(u_i, v_i)$ assigns one of the m bins for the grey value at location u_i, v_i and δ is the Kronecker delta function.

To measure the similarity of two distributions $\theta_1 = \{\theta^r_1\}_{r=1,...,m}$ and $\theta_2 = \{\theta^r_2\}_{r=1,...,m}$ we selected (among various possibilities [16]) the Bhattacharyya coefficient, which proved to be of value in combination with tracking [14,15]

$$\rho(\theta_1, \theta_2) = \sum_{r=1}^{m} \sqrt{\theta^r_1 \theta^r_2}. \tag{5}$$

$\rho(\theta_1, \theta_2)$ ranges from 0 to 1, with 1 indicating a perfect match. The Bhattacharyya distance $d(\theta_1, \theta_2) = \sqrt{1 - \rho(\theta_1, \theta_2)}$ can easily be calculated from the coefficient.

For tracking, a reference distribution θ is calculated at track initialization, which is updated over time to compensate for small texture changes. As in [14] the update is done with the mean histogram $\bar{\theta}$ observed under the shape of all particles

$$\theta^r_{t+1} = \alpha \bar{\theta}^r_t + (1 - \alpha)\theta^r_t. \tag{6}$$

The user specified parameter α controls the contribution of the previous reference and the observed mean histograms.

4 Bayesian Tracking

In this work particle filtering is applied to approximate optimal Bayesian tracking [2,4] for a single target. The state vector $\boldsymbol{x} = (\Pi, \Sigma)$ of a particle comprises the position and velocity $\Pi = (x, y, z, v_x, v_y, v_z)$ in three dimensional space (a fixed object size is assumed), and the shape parameters $\Sigma = (c, \mathbf{b})$ introduced in Section 3.1. For tracking, the dynamics $p(\boldsymbol{x_{k+1}} | \boldsymbol{x_k} = s^i_k)$ and the conditional density $p(\boldsymbol{z_k} | \boldsymbol{x_k} = s^i_k)$ have to be specified, whereas s^i_k is the i^{th} sample at time k.

4.1 Dynamics

Object dynamics is assumed independent for the two components Π and Σ of our state vector and is defined separately as follows.

During each sampling period T_k the position and velocity vector Π is assumed to evolve according to the following dynamic equation

$$\Pi_{k+1} = \begin{pmatrix} 1 & 0 & 0 & T_k & 0 & 0 \\ 0 & 1 & 0 & 0 & T_k & 0 \\ 0 & 0 & 1 & 0 & 0 & T_k \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Pi_k + \Gamma_k \nu_k, \tag{7}$$

whereas ν_k is the user defined process noise, which has to be chosen to account for velocity changes during each sampling interval T_k, and Γ_k is the time dependent noise gain [1,2].

The shape component $\Sigma = (c, \mathbf{b})$ is composed of a discrete parameter c_k modeling the cluster membership and the continuous valued weight vector $\mathbf{b}$. To deal with this "mixed" state the dynamics is decomposed as follows

$$p(\Sigma_{k+1}|\Sigma_k) = p(\mathbf{b}_{k+1}|c_{k+1}, \Sigma_k) p(c_{k+1}|\Sigma_k). \tag{8}$$

Assuming that the transition probabilities $T_{i,j}$ of our discrete Markov model are independent of the previous weight vector $\mathbf{b}_k$, the second part of Equation 8 reduces to

$$p(c_{k+1} = j | c_k = i, \mathbf{b}_k) = T_{i,j}(\mathbf{b}_k) = T_{i,j}. \tag{9}$$

For the continuous parameters we now have to consider two cases: In case of $i = j$, when no PDM transition occurs, we assume

$$p(\mathbf{b}_{k+1}|c_{k+1} = j, c_k = i, \mathbf{b}_k) = p_{i,j}(\mathbf{b}_{k+1}|\mathbf{b}_k) \tag{10}$$

to be a Gaussian random walk. For $i \neq j$ the cluster is switched from i to j and the parameters $\mathbf{b}$ are assumed to be normally distributed about the mean shape of PDM j.

4.2 Multi-cue Observation

Three cues are integrated in this work, which contribute to the particle weights: shape, texture, and stereo. Their distributions are assumed conditionally independent so that

$$p(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i) = p_{shape}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i)\; p_{texture}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i)\; p_{stereo}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i). \tag{11}$$

Since the shape and texture similarity measures between the prediction and observation are defined in the image plane, the shape of each particle is generated using Equation 2. Its centroid coordinates u, v and the scale factor s are derived using a simple pinhole camera model with known (intrinsic/extrinsic) parameters from the 3D-coordinates x, y, z and the specified 3D object dimensions.

Shape: A method based on multi-feature distance transforms [6] is applied to measure the similarity between the predicted shapes and the observed image

edges. It takes into account the position and direction of edge elements. Formally, if the image I is observed at time k and S is the shape of particle s_k^i we define

$$p_{shape}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i) \propto \exp(-\alpha_{shape}\big(\frac{1}{|S|}\sum_{s\in S} D_I(s)\big)^2), \tag{12}$$

whereas $|S|$ denotes the number of features s in S, $D_I(s)$ is the distance of the closest feature in I to s, and α_{shape} is a user specified weight.

Texture: For texture, $p_{texture}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i)$ is defined as

$$p_{texture}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i) \propto \exp(-\alpha_{texture} d^2(\omega, \theta)), \tag{13}$$

whereas $d(\omega, \theta)$ is the Bhattacharyya distance described in Section 3.2 between the reference distribution θ and the observed texture distribution ω under the shape of particle s_k^i. Like above, $\alpha_{texture}$ is a user defined weighting factor.

Stereo: A stereo vision module generates a depth image I_{depth}, which contains the distance to certain feature points. To measure the depth d_{stereo} of particle s_k^i the distance of the feature points under its shape are averaged. Given the predicted distance z of s_k^i and the measurement d_{stereo}, we define

$$p_{stereo}(\boldsymbol{z_k}|\boldsymbol{x_k} = s_k^i) \propto \exp(-\alpha_{stereo}(d_{stereo} - z)^2), \tag{14}$$

whereas α_{stereo} is a weighting factor.

4.3 Integrated Detection and Tracking

A set of particle filters is used to track multiple objects in this work. Each is in one of the states *active* or *inactive*. An *active* track is either *visible* or *hidden*.

A detection system provides possible object locations, which are modeled as "mixture" of Gaussians, whereas one component corresponds to one detection. The mixture is exploited in two ways: As importance function for the particle filters and for the initialization and termination of tracks.

The following rules, which depend on the actual detections, observations, and geometric constraints control the evolution of tracks.

A track is initialized, if no mean state of an *active* track is in the 3σ-bound of a detection. To suppress spurious measurements it starts *hidden*. Initialization involves drawing the 3D position and velocity of the particles according to the Gaussian of the detection. Since no shape information is provided by the detection system, the cluster membership and the continuous parameters are randomly assigned. After the first particle weighting the reference texture distribution is initialized with the mean histogram observed under the shape of all particles.

A track is *visible*, if it has at least t_1 associated detections, if the last associated detection is at most t_2 time steps old, and if the actual match values were better than user defined thresholds for the last t_3 times. Otherwise the track is *hidden*.

A track becomes *inactive*, if the prediction falls outside the detection area or image, if the actual match values were worse than user specified thresholds for t_4 successive times, or if a second track is tracking the same object.

4.4 Sampling

For particle filtering an algorithm based on *icondensation* [10] is applied. It integrates the standard factored sampling technique of *condensation* [2], importance sampling, and sampling from a "reinitialization" prior probability density.

This allows us to integrate the mixture of Gaussians from the detection system as an importance function into the tracking framework. Like in [10], it is also used as a reinitialization prior, which gives us the possibility to draw samples independently of the past history using the Gaussian of the nearest detection.

Like in [9,11] a two step sampling approach is followed for the decomposed dynamics of the mixed discrete/continuous shape space. At first the cluster of our shape model is determined using the transition probabilities $T_{i,j}$ and afterwards the weight vector $\mathbf{b}$ is predicted according to the Gaussian assumptions described in Section 4.1.

5 Experiments

To evaluate our framework we performed experiments on pedestrian tracking from a moving vehicle. The dynamic shape model, outlined in Section 3.1, was trained from approximately 2500 pedestrian shapes of our training set. The resulting cluster prototypes and the temporal transition probabilities between the associated PDMs are illustrated in Figure 3. As expected (and desired), the

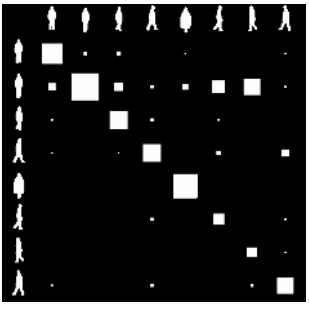

Fig. 3. Cluster transition matrix: The squares represent the transition probabilities from a PDM of column j to row i.

diagonal elements of the transition matrix contain high values, so that there is always a high probability of staying in the same cluster during tracking. Figure 4 shows three random trajectories generated with the proposed dynamic shape model assuming that a camera is moving at 5m/s towards the object in 3D-space, which is moving laterally at 1m/s. Each greyscale change corresponds to a PDM transition.

Pedestrian detection is performed by the ChamferSystem in the experiments. It localizes objects according to their shape in a coarse to fine approach over a

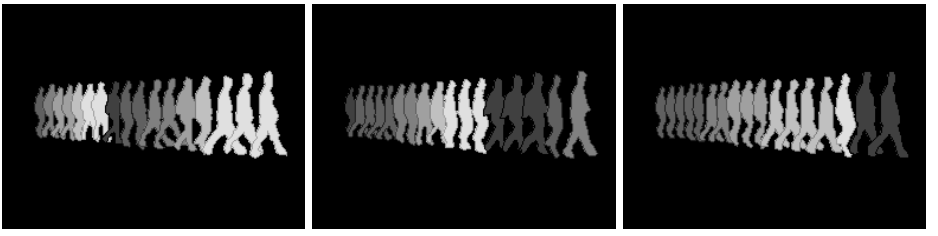

Fig. 4. Predicting shape changes using Dynamic Point Distribution Models: Three random trajectories assuming that the camera is moving at 5m/s towards the object, which is moving laterally at 1m/s. Each greyscale change corresponds to a PDM transition.

template hierarchy by correlating with distance transformed images. For details the reader is referred to [7]. The 3D position is derived by backprojecting the 2D shape with our camera model, assuming that the object is standing on the ground.

The tracking system was tested on a 2.4GHz standard workstation and needs about 300ms/frame for an active track and an image resolution of 256×196. The number of particles is set to 500 in the experiments.

During tracking, the a-priori and a-posteriori probability of each PDM can be observed online, as shown in Figure 5. The size of the dark and light grey boxes indicate the a-priori and a-posteriori probability respectively. The more similar they are, the better the prediction.

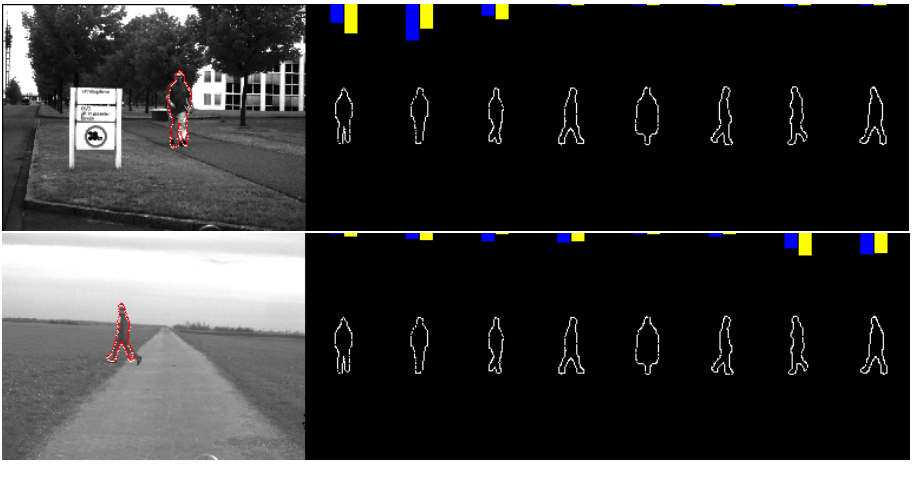

Fig. 5. Tracking results: The dark box indicates the a-priori and the light the a-posteriori confidence of each cluster. The larger the box the higher the probability.

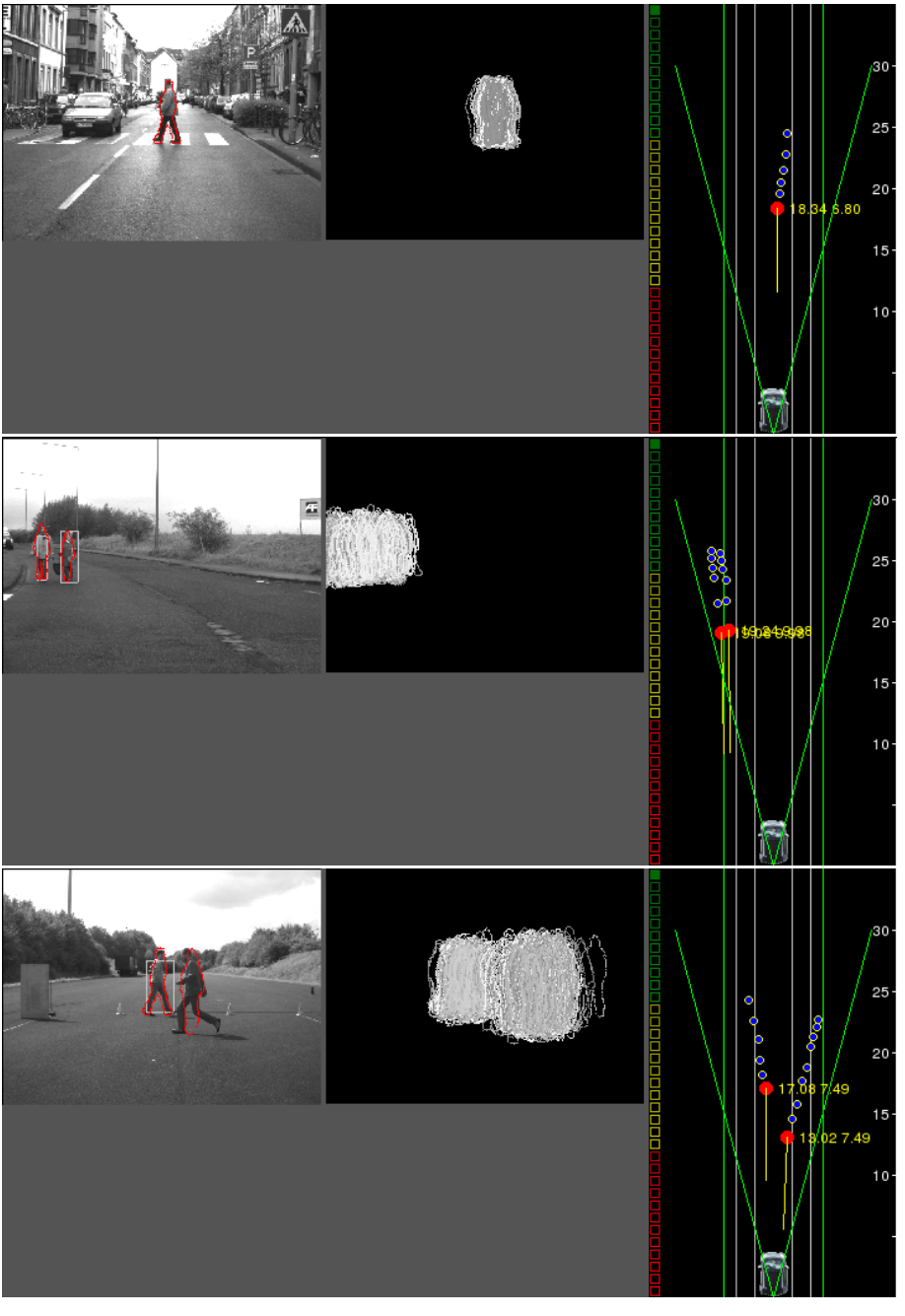

Fig. 6. Tracking results: In the left images the best sample is shown for each track. In addition the detections are illustrated as boxes. The shapes of all particles, which approximate the posterior pdf, are drawn in the middle. Finally a top view of the scene can be viewed on the right.

Table 1. Average distances of the true and estimated pedestrian locations for a sequence of 34 images.

cues	distance	lateral error	error in depth
edge	1.37m	0.085m	1.34m
edge + texture	1.28m	0.14m	1.25m
edge + texture + stereo	1.05m	0.11m	1.03m

Results of the overall system are given in Figure 6 for urban, rural, and synthetic environments. In the original images (left column) the best sample of each active track is shown. Whenever detections are observed, they are also represented there as grey boxes. The shapes of all particles, which approximate the posterior pdf, are drawn in the middle column. Finally, a top view of the scene can be viewed on the right. The past trajectories are represented by small circles while the current position estimate is marked by big circles. The text contains the actual distance and velocity estimates.

To substantiate the visually observable improvement due to the integration of shape, texture, and stereo information, the position estimates of the system were compared against ground truth. Table 1 shows the results for the first sequence of Figure 6, which consists of 34 images. As expected, the average error in depth is higher than the lateral and an improved performance due to the integration of multiple cues can be observed.

6 Conclusions

This paper presented a general Bayesian framework for multi-cue 3D deformable object tracking. A method for learning spatio-temporal shape representations from examples was outlined, which can deal with both continuous and discontinuous appearance changes. Texture histograms and direct 3D measurements were integrated, to improve the robustness and versatility of the framework. It was presented how measurements from an independently operating detection system can be integrated into the tracking approach by means of importance sampling. Experiments show, that the proposed framework is suitable for tracking pedestrians from a moving vehicle and that the integration of multiple cues can improve the tracking performance.

References

1. Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, editors. *Estimation with applications to tracking and navigation.* Wiley, 2001.
2. A. Blake and M. Isard. *Active Contours.* Springer, 1998.
3. T.F. Cootes, C.J. Taylor, D.C. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.
4. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer, 2001.

5. R.J. Elliot, L. Aggoun, and J.B. Moore. *Hidden Markov Models*. Springer, 2nd edition, 1997.
6. D. M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Proc. of the ICPR*, pages 439–444, Brisbane, 1998.
7. D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. of the ECCV*, pages 37–49, 2000.
8. D.M. Gavrila and J. Giebel. Virtual sample generation for template-based shape matching. In *Proc. of the IEEE CVPR Conf.*, pages I:676–681, 2001.
9. T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. of the ICCV*, pages 344–349, 1998.
10. M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of the ECCV*, pages 893–908, 1998.
11. M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *Proc. of the ICCV*, pages 107–112, 1998.
12. M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proc. of the ICCV*, pages 34–41, 2001.
13. S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.
14. K. Nummiaro, E. Koller-Meier, and L. Van Gool. Object tracking with an adaptive color-based particle filter. In *Proc. of the Deutsche Arbeitsgemeinschaft für Mustererkennung*, Zurich, Switzerland, 2002.
15. P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. of the ECCV*, pages 661–675, 2002.
16. Y. Rubner, J. Puzicha, C. Tomasi, and J.M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *CVIU*, 84(1):25–43, 2001.
17. M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Machine, Vision and Applications*, 14:50–58, 2003.
18. K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. of Computer Vision*, 48(1):9–19, 2002.
19. J. Triesch and C. von der Malsburg. Self-organized integration of adaptive visual cues for face tracking. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Los Alamitos, 2000.
20. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.
21. J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. of the ICCV*, 2003.