Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models

S. Munder[1,2], C. Schnörr[2], and D. M. Gavrila[1,3]

[1] Environment Perception, Daimler Research, Ulm, Germany
dariu.gavrila@Daimler.com

[2] CVGPR Group, Department of Mathematics and Computer Science, University of Mannheim, Germany
schnoerr@uni-mannheim.de

[3] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

Abstract

This paper presents a robust multi-cue approach to the integrated detection and tracking of pedestrians in cluttered urban environment. A novel spatio-temporal object representation is proposed that combines a generative shape model and a discriminative texture classifier, both composed of a mixture of pose-specific submodels. Shape is represented by a set of linear subspace models, an extension of Point Distribution Models, with shape transitions modeled by a first-order Markov process. Texture, i.e. the shape-normalized intensity pattern, is represented by a manifold implicitly delimited by a set of pattern classifiers, while texture transition is modeled by a random walk. Direct 3D measurements provided by a stereo system are furthermore incorporated into the observation density function. We employ a Bayesian framework based on particle filtering to achieve integrated object detection and tracking. Large-scale experiments involving pedestrian detection and tracking from a moving vehicle demonstrate the benefit of the proposed approach.

I. INTRODUCTION

Pedestrians, and children in particular, are the most vulnerable participants of today's urban traffic. The goal of this paper is to present techniques for sensor-based driver assistance systems that detect potentially dangerous situations with pedestrians ahead of time. Such systems could then raise a warning to a possibly inattentive driver or, if no sufficient time remains, initiate protective measures such as automatic vehicle braking or the deployment of front airbags to either avoid an accident or to mitigate the impact of an otherwise unavoidable collision.

A prerequisite to such systems is the capability to accurately detect, localize, and track pedestrians. We make use of video sensors, because they provide fine-grained texture information that allows to distinguish pedestrians from other objects in the traffic environment. But the problem is challenging from a computer vision perspective due to the great variety of human appearances, background structure, and lighting conditions. While many different approaches to pedestrian detection and tracking have previously been studied in the literature, we seek to gain robustness by the use of multiple visual cues and their tight integration.

Building upon previous work [18], this paper introduces a novel spatio-temporal object representation that combines mixture models of shape and texture. An associated observation density function integrates multiple visual cues: *shape*, *texture*, and *depth*. Object shape is used, since it is distinctive yet sidesteps variation in object appearance due to texture. Furthermore, efficient matching techniques exist [17]. We utilize an extension of Point Distribution Models [6], [15]

to represent 2D object silhouettes by a set of distinct linear subspace models, each attuned to a distinct body pose or articulation. Temporal shape changes are represented by probabilities of subspace switching, by means of a Markov transition matrix. This shape model is generative in that it allows to synthesize shape hypotheses from the transition prior. Accurate segmentation of the object region is obtained by matching the shape hypotheses to image data using an active contour algorithm.

The object model is complemented by a texture component that employs a pattern classifier to make the distinction between object and background regions, after the image patch has been normalized for shape. The texture component also involves a mixture model, with separate classifiers for each shape subspace for enhanced specificity. Within a Bayesian tracking framework, this discrimination capability allows inference to be made not only about object configuration (position, shape, etc.) but also about the object class (target object versus background clutter). This enables the tracker to recognize false initializations and object disappearance, thus performing integrated object detection and tracking.

3D object kinematics (position and velocity) is modelled, which allows to incorporate available real-world knowledge and for which we obtain direct observations by means of stereo imaging ("depth"). We solely rely on learning-based approaches for constructing our object model from training data, since no prior object models are available that accurately describe human appearance in an arbitrary environment.

We apply our multi-cue object model within a Bayesian framework for detection and tracking based on particle filtering. The choice of particle filtering is due to the cluttered environment of our application and the use of highly nonlinear observations, which leads to a non-Gaussian, multi-modal posterior probability density function. An independently operating object detection module provides object hypotheses from single image frames, which are used to initialize new tracks and which serve as an additional source of information for particle sampling of existing tracks. Extensive evaluation of the proposed approach is performed on video data recorded on two half-hour drives in urban environment.

The outline of the paper is as follows. After reviewing previous work in Section II, our proposed multi-cue object model is described in Section III. Subsections IIIA–C cover details of the individual components shape, texture, and depth, their integration is described in Subsection III-D. In Section IV, details of the implementation of our particle filtering framework integrating

the proposed object model and the independent object detector are given. Experimental results that validate the proposed approach are presented in Section VI, and we conclude in Section VII.

II. Previous Work

There is extensive literature on the visual detection and tracking of pedestrians. A detailed survey of different sensor modalities and detection methods has recently been given in [13] and [14]. Previous research on detection and tracking systems can roughly be divided into two distinct lines. One line of research combines single-frame pedestrian detection with general-purpose object tracking. For the former, pattern classification techniques such as Support Vector Machines [8], [29] or AdaBoost [40] were applied to various feature extraction methods, e.g. Haar wavelets [29], [40] or orientation histograms [8]. See [27] for a recent comparative study. The subsequent tracking step builds upon Kalman filters [1], [5] or particle filtering [28]. Despite successful applications, such approaches only utilize position information for tracking and drop valuable information about temporal appearance transition. Alternatively, temporal image features have been integrated into texture classification [40], [41], but these approaches require additional means for ROI (region of interest) alignment or are applicable to static camera scenarios only. In this paper, discriminative texture classification is integrated into a probabilistic object model employed within a Bayesian tracking framework.

Bayesian tracking approaches form the second line of research. They involve an object model with an associated observation density function, and a mathematical method to sequentially infer the posterior probability density. See Table I for an overview. Regarding object models, an attractive way of representing pedestrians involves shape models, because they eliminate the need for modeling intensity variations that arise from varying lighting or clothing. The manifold of pedestrian shapes has either been represented by a set of exemplars in combination with efficient coarse-to-fine matching techniques [17], [37], [38], or by parametric representations of deformable contours [4], [6], [15], where shape matching involves iterative parameter estimation techniques. Recently, statistical field models have been introduced that directly model edge observation likelihoods [24], [43].

However, a vulnerability of purely shape-based approaches is their susceptibility to background clutter, since random background structure may lead to similar shape observations as the target

TABLE I

OVERVIEW OF VISUAL CUES AND TRACKING METHODS USED IN PREVIOUS WORK ON VISUAL TRACKING OF HUMANS

Authors	Object Model	Visual Cues				Tracking Approach
		Shape	Texture	Motion	Others?	
Deutscher *et al.*, 2000, [9]	3D assembly of cylinders	edge pixels			BG subtr.	"Annealed" PF
Isard and MacCormick, 2001 [22]	3D generalized cylinder	Mexican hat filter	color			PF
Soto and Khosla, 2001 [35]	2D appearance	edge pixels	color histogram		stereo	PF
Toyama and Blake, 2002 [38]	2D shape exemplars	edge pixels				PF
Fablet and Black, 2002 [10]	2D appearance			optical flow		PF
Sidenbladh and Black, 2003 [34]	3D assembly of truncated cones	edge, ridge		intensity diff.		PF
Spengler and Schiele, 2003 [36]	2D appearance		skin color		BG subtr.	Kalman or PF
Zhao and Nevatia, 2004 [45]	3D shape and locomotion			optical flow	BG subtr.	Kalman
Roth *et al.*, 2004 [32]	2D appearance	gradient pixels				PF
Okuma *et al.*, 2004 [28]	2D appearance		color histogram			"Mixture" PF
Kang and Kim, 2005 [24]	2D shape (SOM)	edge pixels				PF
Ramanan *et al.*, 2005 [31]	component-based 2D shape	edge pixels	color classifier			MAP search by DP
Wu and Yu, 2006 [43]	2D shape (Markov field)	edge pixels				PF
Wu and Nevatia, 2006 [42]	2D component appearance	"Edgelets"				data association or meanshift
Alonso *et al.*, 2007 [1]	2D component appearance	Canny, HoN	histogram, NTU			Kalman
***This paper*, 2007**	**mixture of 2D shape (PDM) and texture**	**edge pixels**	**texture classifier**		**stereo**	**PF**

BG subtr.: background subtraction; PF: particle filtering; PDM: Point Distribution Models [6]; SOM: Self-Organizing Map; DP: Dynamic Programming; HoN: Histogram of normalized gradients; NTU: "number of texture unit", see [1].

class. Increased robustness and accuracy has been obtained by combining shape and texture, such as the compound linear models described in [7], [11], [23]. Their application, however, is time consuming, since a model fit requires the combined estimation of shape and texture parameters by means of iterative, gradient descent-like methods. As a way out, separate models of shape and texture have been built, and their observations have been combined in a joint observation density function [22], [34], [35].

Orthogonal to combining multiple visual cues, the accuracy of appearance models has been increased by using mixtures of pose-specific models. For instance, Heap and Hogg [19] proposed a "Hierarchical PDM" (point distribution model) approach to 2D shape modeling, where a set of locally linear shape models is found by a k-means clustering of the training data. The adaptation of this approach in [18] has been used for our shape model in Section III-A. Other authors [42], [44] manually categorize by viewing angle (e.g., left, right, frontal, and back views), and build separate object models for each viewing direction.

Regarding pedestrian tracking, particle filtering has evolved as the standard tool because of its ability to estimate complex multi-modal posterior pdfs (probability density functions) that arise in cluttered environments. Following the seminal work of Isard and Blake [20] who re-introduced particle filtering to computer vision, many extensions have been proposed regarding mixed discrete/continuous state spaces [19], improved sampling strategies [9], [21], and the integration of multiple visual cues [21], [26], [36].

Tracking needs to handle a variable number of objects which randomly enter or leave the field of view. This can be achieved by a joint state space of variable dimension, where the number of objects is inferred in parallel with each object's configuration [22], [25]. However, the computational burden, increasing exponentially with dimension, is high. If complex object representations or complex observation models are involved, authors have generally refrained from joint state spaces but rather ran multiple (single target) tracker instances in parallel. Some heuristics are then typically used to handle track initialization and termination, and to implement target interactions. For example, a separate tracker instance with uniform prior distribution was employed in [24], while independent object detector processes were used in [21], [28]. Object detection performance is clearly limited by the initialization heuristic, which operates on single frames only. In this paper, inference about the object class is therefore made by the tracker in a sound Bayesian manner, based on the discriminatory components of our object model. For

processing cost reasons, one separate tracker is employed for each pedestrian hypothesis.

III. Multi-cue Object Representation

In this section, we introduce the cues that we use to represent the object class, their observation in video images, and their temporal transition. Three different visual cues are considered: shape, texture, and depth. Object *shape* is represented by its 2D contour for which we build a parametric mixture model (subsection III-A). *Texture* denotes the pixel intensity pattern within the object's contour obtained after shape normalization (subsection III-B). Direct 3D measurements from stereo imaging are incorporated by the *depth* cue (subsection III-C).

All three cues are represented within the object state vector $\mathbf{x}_t = (\mathbf{u}_t, \mathbf{s}_t, \mathbf{v}_t)$, which consists of object position and velocity in 3D, $\mathbf{u}_t$, shape $\mathbf{s}_t$, and texture $\mathbf{v}_t$, at time index t. The use of a Bayesian approach for tracking requires to model the temporal transition of the object state, $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, and its observation by means of image features $\mathcal{Z}_t$ extracted from the input image at time t, $p(\mathcal{Z}_t \,|\, \mathbf{x}_t)$. The observation density function is derived in subsection III-D below by integrating the three visual cues. For modeling the transition pdf (probability density function), we assume the decomposition

$$p(\mathbf{u}_t, \mathbf{s}_t, \mathbf{v}_t \,|\, \mathbf{u}_{t-1}, \mathbf{s}_{t-1}, \mathbf{v}_{t-1}) \;=\; p(\mathbf{u}_t \,|\, \mathbf{u}_{t-1})\, p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1})\, p(\mathbf{v}_t \,|\, \mathbf{s}_t, \mathbf{s}_{t-1}, \mathbf{v}_{t-1})\,. \tag{1}$$

The individual transition pdfs are derived below.

A. Shape

Our shape model is three-fold consisting of a parametric representation of static shapes of the object class, a model of temporal transition, and an observation function.

1) Static Shape Model: For modeling the 2D contours of pedestrians in still images, we employ the *Multi Point Distribution Models* described in [15]. A mixture of linear subspace models is used to handle the manifold of pedestrian shapes, each describing a certain body pose or viewing direction. The shape model is built from training data in a semi-automatic process consisting of the following two steps (see Figure 1):

Registration and clustering: In the first step, the training shapes are partitioned into K clusters (K given by the user), and point correspondences are established. These two operations are done jointly: Shape registration is only done within each cluster to ensure that physical point

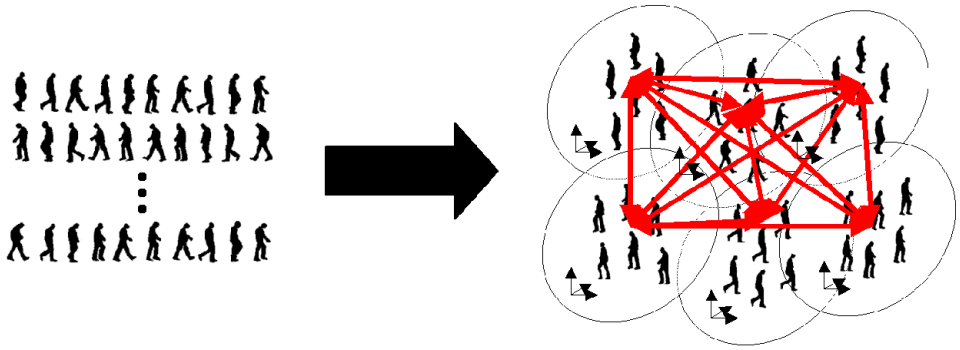

Fig. 1. Illustration of the construction of our shape model. The set of training shapes is first partitioned into clusters of distinct body poses, and a linear subspace model is built for each cluster. Temporal transitions between the clusters are represented by a Markov transition matrix.

correspondences can be found, and the distance function used for clustering is based on the actual point correspondences. Details of the integrated shape registration and clustering algorithm are given in [15]. This fully automatic procedure is followed by a manual refinement step to correct obvious errors in clustering or registration.

Linear subspace model: The result of the registration and clustering procedure is a set of K local vector spaces. A compact representation of each local vector space is obtained by principal component analysis (PCA), where the number of eigenvectors to retain is chosen such that a user-supplied fraction of the total variance (we use 95%) is explained. A Mahalanobis threshold is then determined that covers a user-supplied fraction of the training examples, e.g. 90%. Examples outside the resulting hyperellipsoid are considered outliers, while a truncated normal distribution is assumed within the hyperellipsoid.

2) Shape Observation: The chamfer distance is used to measure the similarity between an instance of our static shape model and an observed image, where the position and direction of edges found in image I are used as features. We make use of the multi-feature distance transform [17] to compute the shape observation

$$z_{shape}(I, \mathbf{x}) = \frac{1}{|S|} \sum_{s \in S} D_I(s)\,, \tag{2}$$

where S is the set of pixels resulting from the projection of the shape parameters into the image I, and $D_I(s)$ is the distance from s to the closest edge pixel in I with a matching edge direction. The shape observation z_{shape} is incorporated into the observation density function in subsection

III-D.

3) Shape Transition: The temporal transition of an object's shape is decomposed into pose cluster switching and shape changes within each cluster. The former is handled by a discrete first-order Markov process, where entry $T_{i,j}$ of the transition matrix describes the probability of switching from cluster $c_{t-1} = i$ to $c_t = j$. A Gaussian random walk is assumed for shape changes within the same cluster ($c_{t-1} = c_t$), while the shape prior is used in the case of a cluster switch. More precisely, if $(c_t, \mathbf{b}_t)$ is the shape state at time t consisting of the pose cluster c_t and PCA coefficients $\mathbf{b}_t$, then

$$p(c_t, \mathbf{b}_t \,|\, c_{t-1}, \mathbf{b}_{t-1}) = T_{c_{t-1},c_t} \cdot \begin{cases} g_{c_t}(\mathbf{b}_t \,|\, \mathbf{b}_{t-1}) & \text{if } c_t = c_{t-1} \\ p(\mathbf{b}_t \,|\, c_t) & \text{if } c_t \neq c_{t-1}\,, \end{cases} \tag{3}$$

where g_{c_t} is a Gaussian random walk and $p(\mathbf{b}_t \,|\, c_t)$ is the normal shape prior, both subject to the Mahalanobis threshold prescribed above.

B. Texture

The texture cue represents the variation of the intensity pattern across the image region of target objects. Much like in the *Active Appearance Models* by Cootes *et al.* [7], appearance variations that arise from differing shapes are eliminated by normalizing each object image for shape. Given the pose cluster and the parameters of the respective shape submodel, a Delauney triangulation method [3] is used to obtain a piece-wise affine warp to the mean shape of the respective pose cluster; see Figure 2 for a few examples. Let $V_I(\mathbf{u}, \mathbf{s})$ denote the texture vector such obtained from image I at position $\mathbf{u}$ (projected to the image plane) with shape parameters $\mathbf{s}$.

1) Static Texture Representation: Although appearance variation has already been significantly reduced by shape normalization, the foreground texture distribution of the pedestrian class is still very complex due to the great diversity of clothing, and due to varying lighting conditions. We hence do not attempt to build a generative model of this manifold, but instead make use of a generic pattern classifier to find the decision boundary between object and non-object texture patterns. In earlier research [27], we found a neural network with local receptive fields [41] particularly suitable for the task of pedestrian classification. One such neural network h_c is trained for each pose cluster $c = 1, \ldots, K$.

Fig. 2. Examples of shape normalization. The top row shows a few examples of the training set that fall into the same pose cluster. Texture warping to the pose cluster prototype is shown in the bottom row. Contour labels are superimposed on the examples images for visualization purposes, background pixels outside this contour are masked out for further processing.

Texture observation, given an input image and an hypothesized object configuration $\mathbf{x} = (\mathbf{u}, \mathbf{s}, \mathbf{v})$, then involves to feed the shape-normalized image patch $V_I(\mathbf{u}, \mathbf{s})$ into the neural network of pose cluster c to yield the texture observation value

$$z_{\textit{texture}}(I, \mathbf{x}) = h_c(V_I(\mathbf{u}, \mathbf{s})). \tag{4}$$

2) Temporal Texture Transition: The shape-normalized texture pattern of a pedestrian is assumed to remain constant over time, plus some unknown random noise. The transition pdf is therefore modeled by cross-correlating the two consecutive texture state vectors $\mathbf{v}_{t-1}$ and $\mathbf{v}_t$. If there is no pose cluster switch, i.e., $c_{t-1} = c_t$, then the shape-normalized texture vectors $\mathbf{v}_{t-1}$ and $\mathbf{v}_t$ have pixel-wise correspondence and we define

$$p(\mathbf{v}_t \,|\, \mathbf{v}_{t-1}, c_{t-1} = c_t) \propto \exp\big(-\alpha_1 \textit{ZNCC}(\mathbf{v}_{t-1}, \mathbf{v}_t) - \alpha_0\big)\,. \tag{5}$$

ZNCC denotes the zero-mean normalized cross-correlation given by

$$\textit{ZNCC}(\mathbf{a}, \mathbf{b}) = \frac{(\mathbf{a} - \bar{a}\mathbf{1}) \cdot (\mathbf{b} - \bar{b}\mathbf{1})}{\sqrt{(\mathbf{a} - \bar{a}\mathbf{1})^2 (\mathbf{b} - \bar{b}\mathbf{1})^2}}\,, \tag{6}$$

where $\bar{a}$ is the mean of vector $\mathbf{a}$. In the case of a pose cluster switch, $c_{t-1} \neq c_t$, we observe that texture transformations occur mainly in horizontal image direction, while the vertical intensity distribution remains approximately constant. This is exploited by matching the vertical profile

of the two texture patterns given by a projection to the image y axis and resampling to some fixed length. We thus have

$$p(\mathbf{v}_t \,|\, \mathbf{v}_{t-1}, c_{t-1} \neq c_t) \propto \exp\big(-\beta_1 \mathit{ZNCC}(H_{c_{t-1}}(\mathbf{v}_{t-1}), H_{c_t}(\mathbf{v}_t)) - \beta_0\big)\,, \tag{7}$$

where H is the vertical profile operator. The pdf parameters $\alpha_0, \ldots, \beta_1$ are learned from training data.

C. Depth

The *depth* cue represents the 3D position and velocity of target objects, the former observed by means of stereo imaging. Modeling object position in 3D space rather than 2D image space simplifies the dynamical model (we assume constant velocities), and allows to incorporate scene constraints such as the assumptions of pedestrians standing with at least one foot on the ground. Thus, $\mathbf{u} = (u_x, u_y, u_z, u_{vx}, u_{vy}, u_{vz})$.

1) Depth Observation: We make use of a feature-based, multi-resolution stereo algorithm developed by Franke [12]; alternative choices would have been possible, e.g. [39]. The outcome is a relatively sparse depth map that provides depth estimations along vertical image edges. Depth measurements are assumed normally distributed around the true depth, so we compute the mean difference as the depth observation value to be fed into the observation density function below,

$$z_{\mathit{depth}}(I, \mathbf{x}) = \overline{Z_I(\mathbf{u}, \mathbf{s})} - u_z\,, \tag{8}$$

where $\overline{Z_I(\mathbf{u}, \mathbf{s})}$ denotes the mean of depth measurements within the image region given by $(\mathbf{u}, \mathbf{s})$.

2) Dynamics: The dynamics of the position and velocity component $\mathbf{u}$ of the state vector is modeled as a first-order auto-regressive process by

$$\mathbf{u}_t = \begin{pmatrix} I_3 & I_3\Delta t \\ 0 & I_3 \end{pmatrix} \mathbf{u}_{t-1} \;+\; \mathbf{e}_u \Delta t\,, \quad \mathbf{e}_u \sim N(0, \Sigma_u)\,, \tag{9}$$

where Σ_u is the user-defined process noise, Δt the time interval, and I_3 denotes the 3×3 identity matrix.

D. Cue Integration

Having introduced the individual cues above, we now turn to their integration to model the density function $p(\mathcal{Z}_t \,|\, \mathcal{X}_t)$ of observing image features $\mathcal{Z}_t$ given the true state $\mathcal{X}_t$, by using the above cues $\mathbf{z} = (z_{\textit{shape}}, z_{\textit{texture}}, z_{\textit{depth}})$. The state of interest $\mathcal{X}_t$ to be inferred from the video sequences is the presence and positions of pedestrians. Here, we assume that either no target object is present at time t, or exactly one at position $\mathbf{x}_t$, which we denote by $\mathcal{X}_t = \mathcal{N}_t$ and $\mathcal{X}_t = (\mathcal{O}_t, \mathbf{x}_t)$, respectively. Multiple objects are handled by multiple trackers, one for each object, see Section V.

Roughly, input image I_t is expected to contain only non-object features in the case $\mathcal{X}_t = \mathcal{N}_t$, whereas in the state $\mathcal{X}_t = (\mathcal{O}_t, \mathbf{x}_t)$, object features are expected within the image region given by $\mathbf{x}_t$, and non-object features elsewhere. This decomposition is realized by making a simplifying assumption: Projecting object state $\mathbf{x}_t$ to the image plane subdivides the image into a foreground and a background region. Although not strictly true, we assume that features extracted from these regions are statistically independent, and that these features obey a common foreground (FG) or a common background (BG) distribution, respectively. With this assumption at hand, we follow the reasoning of Sidenbladh and Black [34] and other authors [22], [33] to get

$$p(\mathcal{Z}_t \,|\, \mathcal{X}_t) \;\propto\; \begin{cases} \dfrac{p(\mathbf{z}(I_t, \mathbf{x}_t) \,|\, \mathrm{FG})}{p(\mathbf{z}(I_t, \mathbf{x}_t) \,|\, \mathrm{BG})} & \text{for } \mathcal{X}_t = (\mathcal{O}_t, \mathbf{x}_t) \\ 1 & \text{for } \mathcal{X}_t = \mathcal{N}_t \,, \end{cases} \tag{10}$$

with the proportionality factor $p(\mathcal{Z}_t \,|\, \mathcal{N}_t)$.

In order to find a parametric model of the above likelihood ratio, we notice that:

- The distribution of Chamfer distances $z_{\textit{shape}}$ is approximated by an exponential distribution [30], [38].
- The neural network output $z_{\textit{texture}}$ is approximately normally distributed about the class means.
- Both cues, shape and (shape-normalized) texture, represent complementary image features, so dependencies are relatively weak.

This motivates the use of a quadratic function to approximate the log of the likelihood ratio

$$\log \frac{p(\mathbf{z} \,|\, \mathrm{FG})}{p(\mathbf{z} \,|\, \mathrm{BG})} \approx \mathbf{z}^T R \mathbf{z} + \mathbf{r}^T \mathbf{z} + r_0 \,, \tag{11}$$

which covers multivariate normal distributions and univariate exponential distributions as special cases. Non-zero off-diagonal elements of matrix R represent statistical dependencies between the respective cues. An illustration of the quadratic approximation is given in Figure 3, which, for visualization purposes, shows two separate quadratic approximations of the two marginal distributions of shape and texture.

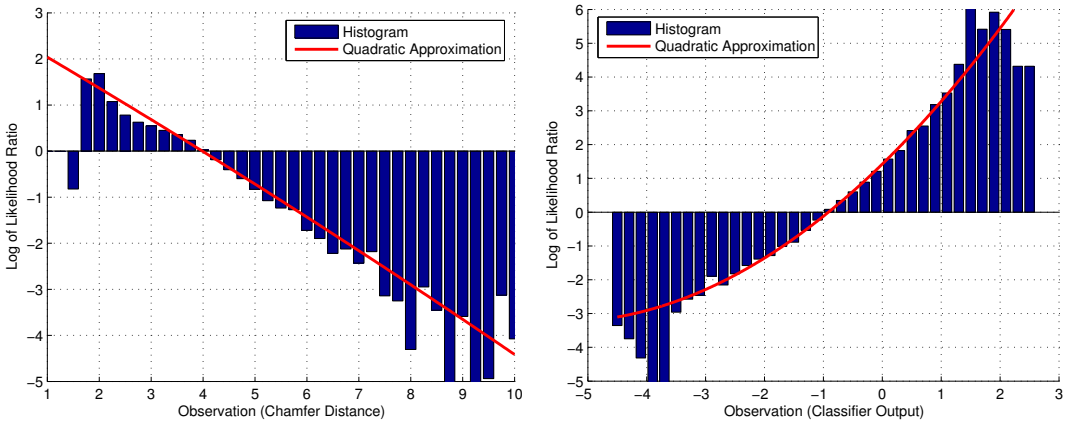

Fig. 3. Quadratic approximation of the log of the likelihood ratio, shown for the marginal distributions of z_{shape} (left), and $z_{texture}$ (right). The ragged borders of both histograms plots arise from sparsely populated histogram bins, whereas the more densely populated regions around the graph centers are well approximated by the quadratic function.

IV. Particle Filtering in an Hybrid State Space

For tracking, we make use of a standard sequential importance resampling (SIR, [2]) particle filter that has been successfully employed in many similar applications (e.g., [22]). Multiple pedestrians are handled by the simplified approach of instantiating one independent particle filter per object, initialized from a separate object detection module. However, each particle filter makes inference about the hybrid state space of object class (target object vs. background clutter) and object configuration (position, velocity, etc.), in order to cope with false or inaccurate initializations. In this section, we derive the particle filtering equations for this hybrid state space.

Filtering denotes the recursive computation of the posterior probability $p(\mathcal{X}_t \mid \mathcal{Z}_{1:t})$ of state $\mathcal{X}_t$ given the series of image observations $\mathcal{Z}_{1:t} = (\mathcal{Z}_1, \ldots, \mathcal{Z}_t)$. Our state of interest $\mathcal{X}_t$ is either $\mathcal{N}_t$, which denotes that no target object is present at time t, or $(\mathcal{O}_t, \mathbf{x}_t)$, which denotes the existence of an object of configuration $\mathbf{x}_t$. Whereas the transition of object configuration $\mathbf{x}_t$ has been

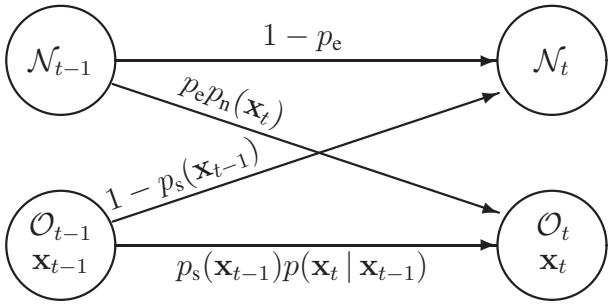

Fig. 4. Transition PDF

defined by our object representation above, we still need to define the transitions of states $\mathcal{O}_t$ and $\mathcal{N}_t$, i.e. the appearance and disappearance of objects as illustrated in Fig. 4:

- $p_\mathrm{s}(\mathbf{x}_{t-1}) = p(\mathcal{O}_t \,|\, \mathcal{O}_{t-1}, \mathbf{x}_{t-1})$ specifies the probability that an object remains within the detection area given its previous position ("Stay"),
- $p_\mathrm{e} = p(\mathcal{O}_t \,|\, \mathcal{N}_{t-1})$ denotes the probability of the event that a new object enters the detection area, and
- $p_\mathrm{n}(\mathbf{x}_t) = p(\mathbf{x}_t \,|\, \mathcal{O}_t, \mathcal{N}_{t-1})$ describes where new objects enter the detection area.

These functions are application-specific and need to be provided the user. The desired posterior is then obtained using Bayes rule as (cf. Eq. (4) in [2])

$$p(\mathcal{X}_t \,|\, \mathcal{Z}_{1:t}) \;\propto\; p(\mathcal{Z}_t \,|\, \mathcal{X}_t)\, p(\mathcal{X}_t \,|\, \mathcal{Z}_{1:t-1})\,, \tag{12}$$

where the transition prior $p(\mathcal{X}_t \,|\, \mathcal{Z}_{1:t-1})$ is given by the Chapman-Kolmogorov equation (cf. Eq. (3) in [2])

$$\begin{aligned} p(\mathcal{O}_t, \mathbf{x}_t \,|\, \mathcal{Z}_{1:t-1}) \;&= p_\mathrm{n}(\mathbf{x}_t)\, p_\mathrm{e}\, p(\mathcal{N}_{t-1} \,|\, \mathcal{Z}_{1:t-1}) \\ &\quad + \int p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p_\mathrm{s}(\mathbf{x}_{t-1})\, p(\mathcal{O}_{t-1}, \mathbf{x}_{t-1} \,|\, \mathcal{Z}_{1:t-1})\, \mathrm{d}\mathbf{x}_{t-1}\,, \end{aligned} \tag{13}$$

$$\begin{aligned} p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t-1}) \;&= (1 - p_\mathrm{e})\, p(\mathcal{N}_{t-1} \,|\, \mathcal{Z}_{1:t-1}) \\ &\quad + \int (1 - p_\mathrm{s}(\mathbf{x}_{t-1}))\; p(\mathcal{O}_{t-1}, \mathbf{x}_{t-1} \,|\, \mathcal{Z}_{1:t-1})\, \mathrm{d}\mathbf{x}_{t-1}\,. \end{aligned} \tag{14}$$

In particle filtering, the posterior is approximated by a set of weighted samples or particles. For representing our hybrid state space, we dedicate one special particle with index 0 and weight $w_t^{(0)}$ to the case $\mathcal{N}_t$, while the remaining N_s particles $\{(\mathbf{x}_t^{(i)}, w_t^{(i)}) \;:\; i = 1, \ldots, N_s\}$ represent

$(\mathcal{O}_t, \mathbf{x}_t)$. Formally,

$$\begin{array}{rcl} p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t}) & \approx & w_t^{(0)} \\ p(\mathcal{O}_t, \mathbf{x}_t \,|\, \mathcal{Z}_{1:t}) & \approx & \sum_{i=1}^{N_s} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \,. \end{array} \tag{15}$$

At each time step t, a new particle set is drawn from a proposal distribution q_t. Here, we draw exactly one particle of state $\mathcal{N}_t$, and N_s particles of state $\mathcal{O}_t$ with object configuration $\mathbf{x}_t$ sampled from the proposal $q_t(\mathbf{x}_t)$, i.e.

$$\begin{array}{rcll} q_t(\mathcal{N}_t) & = & \frac{1}{N_s+1} & \propto \; \frac{1}{N_s} \\ q_t(\mathcal{O}_t, \mathbf{x}_t) & = & \frac{N_s}{N_s+1} q_t(\mathbf{x}_t) & \propto \; q_t(\mathbf{x}_t) \,. \end{array} \tag{16}$$

Particles are then weighted to represent the posterior,

$$w_t^{(i)} \;=\; \begin{cases} \dfrac{p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t})}{q_t(\mathcal{N}_t)} & \propto \; p(\mathcal{Z}_t \,|\, \mathcal{N}_t)\, p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t-1})\, N_s & \text{for } i = 0 \,, \\ \dfrac{p(\mathcal{O}_t, \mathbf{x}_t^{(i)} \,|\, \mathcal{Z}_{1:t})}{q_t(\mathcal{O}_t, \mathbf{x}_t^{(i)})} & \propto \; \dfrac{p(\mathcal{Z}_t \,|\, \mathcal{O}_t, \mathbf{x}_t^{(i)})\, p(\mathcal{O}_t, \mathbf{x}_t^{(i)} \,|\, \mathcal{Z}_{1:t-1})}{q_t(\mathbf{x}_t)} & \text{for } i = 1, \ldots, N_s \,, \end{cases} \tag{17}$$

where the transition priors given in Eqs. (13) and (14) are now approximated by the particle set:

$$p(\mathcal{O}_t, \mathbf{x}_t \,|\, \mathcal{Z}_{1:t-1}) \;\approx p_{\mathrm{n}}(\mathbf{x}_t)\, p_{\mathrm{e}}\, w_{t-1}^{(0)} \;+\; \sum_{j=1}^{N_s} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^{(j)})\, p_{\mathrm{s}}(\mathbf{x}_{t-1}^{(j)})\, w_{t-1}^{(j)} \tag{18}$$

$$p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t-1}) \;\approx (1 - p_{\mathrm{e}})\, w_{t-1}^{(0)} \;+\; \sum_{j=1}^{N_s} (1 - p_{\mathrm{s}}(\mathbf{x}_{t-1}^{(j)}))\, w_{t-1}^{(j)} \tag{19}$$

Proportionalities in the above equations are resolved by normalizing the particle weights to sum to one.

The choice of a good proposal density is a crucial design step in the implementation of a particle filter [2]. The most convenient and most frequent choice is to use the transition prior $p(\mathcal{X}_t \,|\, \mathcal{Z}_{1:t-1})$ as approximated in Eqs. (18) and (19), because this greatly simplifies the particle weight computation in Eq. (17). But this choice is not necessarily optimal, as it may lead to many "wasted" particles with negligible weight, in particular in cases of noisy state prediction (widespread transition prior) and peaked observation densities. It is hence desirable to incorporate the current measurements into the proposal density in order to have particles generated close to the posterior distribution [2], [21]. Since sampling from $p(\mathcal{X}_t \,|\, \mathcal{Z}_t)$ is computationally expensive, the output of the independent target detector is used as an approximation, and the proposal density is then designed as a mixture of both sources of information; details are given in the next Section.

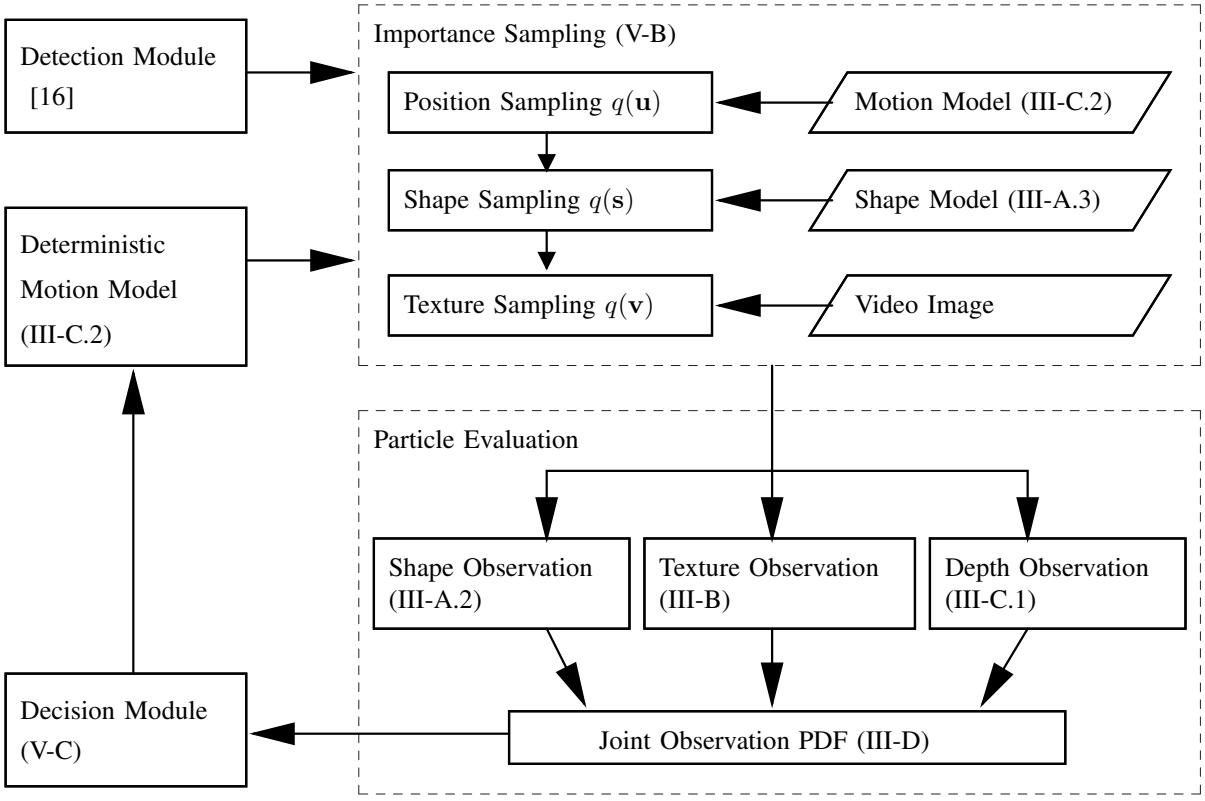

Fig. 5. Overview of our pedestrian detection and tracking system. Section numbers for details of each module are given in brackets.

V. IMPLEMENTATION

We now turn to our pedestrian detection and tracking application, integrating the proposed multi-cue object model (Section III) into the particle filtering framework (Section IV). See Figure 5 for a system overview.

A. Target Object Detector

A computationally efficient target object detector provides a rough approximation about the presence of target objects and their positions, given a single input image [16]. A cascade of system modules is applied to an input image, each utilizing complementary image features, to successively narrow down the search space. Processing starts with stereo-based ROI generation which utilizes a depth map (as in Subsection III-C) and the so-called ground plane constraint to generate a list of ROIs for the shape-based detection module. Based on a hierarchy of exemplary

shape templates, this module performs coarse-to-fine template matching to efficiently locate target objects, by means of chamfer matching. The shape matching results are then fed into a texture-based pattern classification module to make the distinction pedestrian versus non-pedestrian. The pattern classifier used has the same architecture as the one used for the texture representation (Subsection III-B), a neural network with local receptive fields, applied to the raw input intensity pattern. Finally, a stereo-based pedestrian verification heuristic filters out false detections that contain an appreciable amount of background; see [16] for details.

The result of the detection process is a list of 3D positions of potential pedestrians. Deviations from the true positions are assumed to obey a normal distribution, with parameters learned from a training set. In order to use the list of detections in the proposal density of a particle filter, detections are associated to (possibly multiple) existing trackers, by means of a maximum distance to the mean track position. For each tracker, a detector density $g_t(\mathbf{u}_t)$ is built as a mixture of Gaussians from the list of associated detections. (g_t is left unspecified if this list is empty.) Detections not associated to any track are used to initialize new trackers, see below.

B. Proposal Density

The proposal density $q_t(\mathbf{x}_t)$ of our particle filter needs to possess two properties: First, both sampling from and evaluation of the proposal density needs to be computationally efficient. Second, two sources of information, the transition prior and the detector density are to be incorporated into the proposal density by means of a mixture density. Given the decomposition of the transition prior in Eq. (1), sampling is done incrementally for each of the state vector components position, shape, and texture:

- Pdfs of the position component $\mathbf{u}_t$ are given as a Gaussian or mixture of Gaussians (Eq. (9) and definition of g_t above), for which efficient sampling and evaluation is possible. Therefore,

$$q_t(\mathbf{u}_t) = \rho g_t(\mathbf{u}_t) \; + \; (1-\rho) p(\mathbf{u}_t \,|\, \mathcal{O}_t, \mathcal{Z}_{1:t-1}) \,, \tag{20}$$

  where the mixing coefficient ρ is set to 0 if g_t is undefined (i.e. no associated detections), otherwise, we choose $\rho = 0.5$. The component transition prior in the second term,

$$p(\mathbf{u}_t \,|\, \mathcal{O}_t, \mathcal{Z}_{1:t-1}) = \frac{p(\mathcal{O}_t, \mathbf{u}_t \,|\, \mathcal{Z}_{1:t-1})}{1 - p(\mathcal{N}_t \,|\, \mathcal{Z}_{1:t-1})}$$

  is obtained from Eqs. (18) and (19) by replacing $\mathbf{x}_t$ with $\mathbf{u}_t$.

- The transition pdf of the shape component (3) is composed of a discrete first-order Markov process and a Gaussian random walk and allows efficient sampling and evaluation, so we let

$$q_t(\mathbf{s}_t) = p(\mathbf{s}_t \mid \mathcal{O}_t, \mathcal{Z}_{1:t-1}) \,, \tag{21}$$

  where the RHS is computed analogously to the position transition prior above.
- The transition pdf of the texture component (5),(7) based on cross-correlation is easily evaluated, but does not permit direct sampling. Instead, the texture component of particles is always sampled directly from the input image given the particle's position and shape:

$$\mathbf{v}_t \mid \mathbf{u}_t, \mathbf{s}_t = V_{I_t}(\mathbf{u}_t, \mathbf{s}_t) \,. \tag{22}$$

The joint proposal density is then composed of the above parts as

$$q_t\big(\mathbf{x}_t = (\mathbf{u}_t, \mathbf{s}_t, \mathbf{v}_t)\big) = \begin{cases} q_t(\mathbf{u}_t) q_t(\mathbf{s}_t) & \text{if } \mathbf{v}_t = V_{I_t}(\mathbf{u}_t, \mathbf{s}_t) \,, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

In order to explore the high-dimensional state space with only a limited number of particles, we employ a particle optimization step as proposed in [19]. After each new sample is drawn from the proposal density, an active contour algorithm is used to refine the shape and to obtain an accurate segmentation of the foreground region, which is crucial for subsequent texture observations.

C. Track Initialization and Termination

New tracks are initialized if a detection made by the target object detector is not associated to an existing track. In order to suppress spurious detections, tracks start *hidden*, i.e., their output is suppressed. A track becomes *visible* if the probability of not tracking a target object, $p(\mathcal{N}_t \mid \mathcal{Z}_{1:t})$, falls below a threshold $\theta_{\textit{visible}}$, while we switch back to *hidden* if $p(\mathcal{N}_t \mid \mathcal{Z}_{1:t}) > \theta_{\textit{hidden}}$. Tracks are terminated if $p(\mathcal{N}_t \mid \mathcal{Z}_{1:t})$ exceeds the user-defined threshold $\theta_{\textit{term}}$. For the experiments below, we chose $\theta_{\textit{visible}} = 0.5$, $\theta_{\textit{hidden}} = 0.7$, and $\theta_{\textit{term}} = 0.9$. To avoid that multiple trackers "jump" onto the same target, the track with higher non-target posterior is discarded if the mean object positions of two tracks coincide (subject to some user-defined tolerances).

VI. EXPERIMENTS

The proposed multi-cue object representation has been extensively evaluated in the application of tracking pedestrians from a moving vehicle. A training set of 6,522 pedestrian instances occurring in 170 pedestrian trajectories with manually labeled contour points has been used to learn the static and dynamic shape and texture representations (Sections III-A and III-B), respectively. Non-pedestrian patterns were randomly extracted from a set of 7,000 video images without pedestrians. 12 pose clusters have been defined according to four different viewing directions (frontal, back, left, right) and three different leg articulations (feet closed, knees closed but feet apart, knees apart). The number of particles to use in each tracker has been determined in preliminary experiments. As a result of the efficient proposal density and the particle optimization strategy employed, 500 particles per track were found sufficient.

Our pedestrian tracking system was then tested on two long, continuous sequences recorded on the same route through suburbia and inner city of Aachen, Germany, lasting 27 min and 24 min and consisting of 21,053 and 17,390 frames, respectively. We required the system to detect pedestrian within the range of 10m to 25m ahead and 4m to each side. In total, 92 pedestrian trajectories and 1,427 pedestrian instances appeared within this observation area.

System performance was measured by means of *detection rate* and *false alarm rate*. For additional insight, we consider the two performance metrics on both the frame- and trajectory-level. For the latter, we further distinguish two types of trajectories: "class-B" and "class-A" trajectories that have at least one entry or at least 50 percent of their entries matched, respectively. Thus, all "class-A" trajectories are also "class-B" trajectories, but "class-A" trajectories pose stronger detection demands that might be necessary in some applications. Furthermore, we distinguish "system detection rate" which measures the combined detection and tracking performance by counting the percentage of all true pedestrians correctly detected by the system, and "tracking rate" which only considers true pedestrians after (and including) the first detection made by the detector process. For the latter, ground truth instances prior to the first detection are ignored, and true pedestrian trajectories not detected at least once are entirely ignored. Thus, "tracking rate" only refers to the tracking component of the system.

The localization tolerance for matching ground truth and system output was specified in 3D, relative to the true object distance, which better fits application constraints than, e.g., a 2D pixel

TABLE II

EXPERIMENTAL RESULTS COMPARING THE PROPOSED MULTI-CUE DETECTION AND TRACKING APPROACH TO THE PERFORMANCE OF THE DETECTION MODULE ALONE AND TO THAT OF A SIMPLE α-β TRACKER [16]. SEE TEXT FOR DETAILS.

	Detector Only [16]	α-β Tracker [16]			Our Approach		
	F	F	A	B	F	A	B
System Detection Rate	51.6%	58.2%	57.6%	72.8%	64.3%	65.2%	73.9%
Tracking Rate	N/A	76.8%	80.0%	92.9%	84.6%	91.4%	95.7%
FPR (per 10^3fr, min)	15	14	2.5	2.4	17	2.0	2.0

Columns "F" show frame-level performance, "A" and "B" denote class-A and class-B trajectory performance, respectively. "FPR" denotes the number of false positives and is given per 10^3 frames for frame-level performance (F), and per driving minute for trajectory-level performance (A,B). "Tracking Rate" denotes the rate of object detection after the first track initialization.

tolerance. We chose to tolerate 10% in lateral and 30% in longitudinal direction, e.g., at 10m distance we tolerate 1m lateral and 3m longitudinal deviation, respectively. A true pedestrian was considered matched if there was at least one system output within the localization tolerance, and vice versa, so that many-to-many correspondences were allowed.

Evaluation results on the two test sequences are given in Table II. The first column, "Detector Only", refers to the object detector, without any tracking, and serves to quantify the benefit of the tracking component added to the detector. The second column lists the performance obtained with a very basic tracking approach ("α-β Tracker") described in [16]. The outputs of the object detector are simply concatenated by means of the Hungarian method for object association and α-β filtering for predicting object positions. The third column ("Our approach") lists the performance of the system proposed in this paper.

Comparing the second and third column of Table II, we see that false positive rates of both tracking approaches are similar; a slight increase on the frame level is compensated by a decrease on the trajectory level. The benefit of our approach is shown by the increased detection rate. Compared to the detector alone, our tracking approach raised the frame-level detection rate from 51.6% to 64.3%, while only one-half of that increase was achieved by the simpler α-β tracker (58.1%). The effect becomes even more visible when considering the tracking rate on the "class-

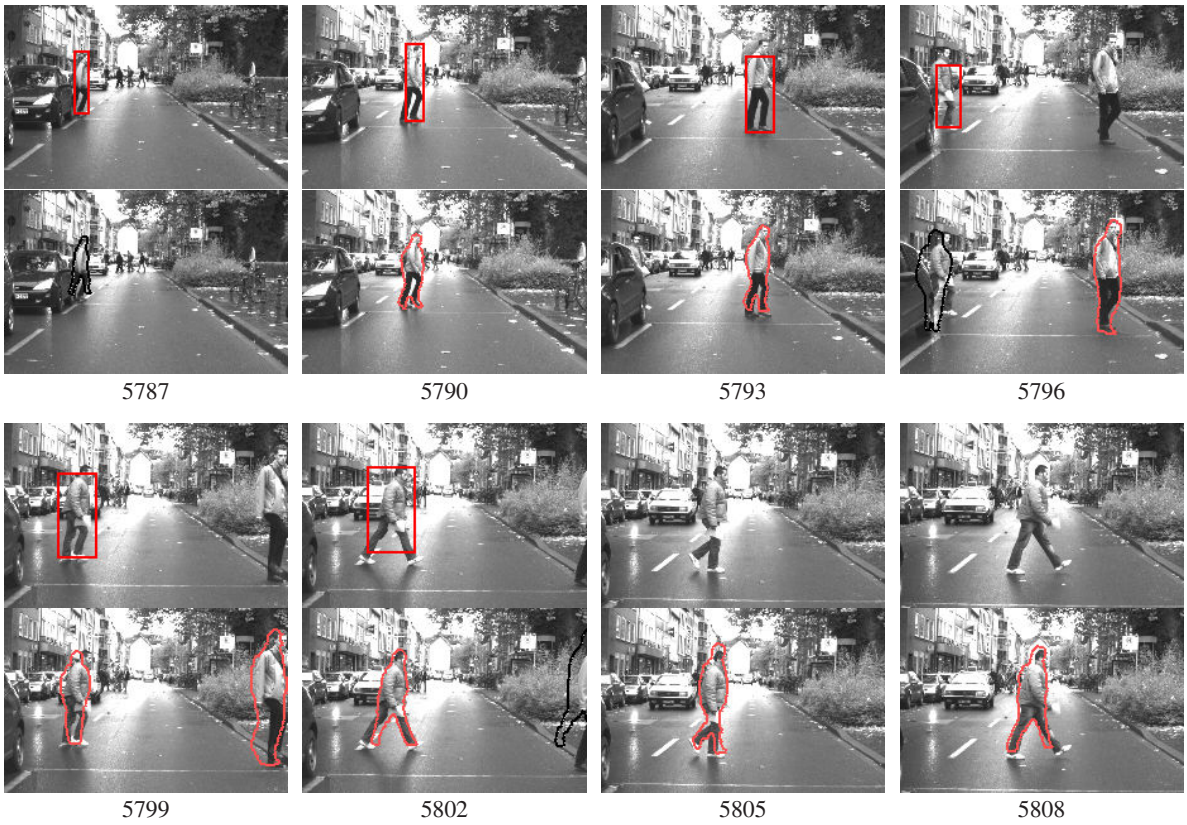

Fig. 6. Example results obtained on one of the test sequences. In each row, the top subimages show results of the detector module, bottom subimages show results of our tracker. Tracks classified as *visible* are shown by red contours, i.e. they denote the actual system output, while *hidden* tracks are shown in black. Frame numbers are given below each image.

A" trajectory level, where our approach achieved the considerably higher rate of 91.4% compared to 80.0% of the α-β tracker.

Figures 6 and 7 show tracking results for a few example frames of the two test sequences. Top subimages show the output of the independent target detector (red boxes). The maximum a-posteriori output (i.e., the particle with maximum weight) of our tracker is given in the bottom subimages. Red contours denote *visible* tracks which are classified as tracking a target object, while black contours denote *hidden* tracks that are more likely not to contain a target object. In Fig. 6, all pedestrians are correctly detected and tracked after few initialization frames, even though the detector process only provides sporadic detections. Fig. 7 shows an example of false positives of the detector process that leads to a false initialization, but which is correctly recognized as such by the tracker.

Fig. 7. Example of a false initialization caused by the detector process (red boxes in top subimages) that are correctly classified as "non-pedestrian" by our tracker (black contours in bottom subimages).

Our unoptimized implementation of the system runs at about 5s per frame on a 2.66GHz Intel PC. The texture warping component of the shape normalization code turned out to be the bottleneck of the current system. Large speed-ups can be expected from careful software optimization (e.g. memory management, loop unrolling) and the use of specialized hardware. For example, texture warping can be performed on the GPU (graphics processing unit) of modern video hardware. Provided that both optimizations yield about a factor of 10, real-time processing speed is achievable.

VII. Conclusion

This paper described a Bayesian, multi-cue approach to the integrated detection and tracking of pedestrians in cluttered urban environment. A novel spatio-temporal object model has been presented that combines dynamic shape representation and shape-invariant texture classification, thus enabling accurate segmentation of the object region and discrimination from non-object patterns. Specificity has been gained by utilizing a mixture of sub-models, each attuned to a particular body pose. The efficacy of the proposed object model has been demonstrated within a Bayesian framework for pedestrian detection and tracking based on particle filtering. By enabling the tracker to make inference about both, object class and configuration, the system detection rate and the tracking rate could be significantly improved without degrading the false alarm rate.

What detection performance is actually necessary for a production system clearly depends on the particular application. The deployment of a front airbag, for example, poses more stringent

performance requirements than an acoustical warning system. See [13], [14] for detailed discussions. Especially for critical applications, the performance of our system certainly needs to be further improved. For example, component-based approaches [34], [44] could improve the robustness of our object model against partial occlusions. For tracking groups of pedestrians, recent advances in multi-target tracking [24], [25] that deal with object interactions seem to be a worthwhile direction of future research. Finally, our system would benefit from a higher detection rate of the object detector.

REFERENCES

[1] I. P. Alonso, D. F. Llorca, M. A. Sotelo, L. M. Bergasa, J. P. R. de Toro Nuevo, M. Ocana, and M. A. G. Garrido, "Combination of feature extraction methods for svm pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 292–307, 2007.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[3] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996. [Online]. Available: http://www.qhull.org

[4] M. Bergtholdt, D. Cremers, and C. Schnörr, "Variational segmentation with shape priors," in *Mathematical Models in Computer Vision: The Handbook*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Springer, 2005.

[5] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascoli, and A. Tibaldi, "Shape-based pedestrian detection and localization," in *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, Bejing, China, 2003, pp. 328–333.

[6] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and applications," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, San Diego, CA, USA, 2005, pp. 886–893.

[9] J. Deutscher, A. Blake, and I. D. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 126–133.

[10] R. Fablet and M. Black, "Automatic detection and tracking of human motion with a view-based representation," in *ECCV*, 2002, pp. 476–491.

[11] L. Fan, K. Sung, and T. Ng, "Pedestrian registration in static images with unconstrained background," *Pattern Recognition*, vol. 36, no. 4, pp. 1019–1029, April 2003.

[12] U. Franke, "Real-time stereo vision for urban traffic scene understanding," in *Proc. of the IEEE Intelligent Vehicle Symposium*, Detroit, USA, 2000.

[13] T. Gandhi and M. M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 413–430, 2007.

[14] D. M. Gavrila, "Sensor-based pedestrian protection," *IEEE Intelligent Systems*, vol. 16, no. 6, pp. 77–81, 2001.

[15] D. M. Gavrila, J. Giebel, and H. Neumann, "Learning shape models from examples," in *Proc. of the Annual Symposium of the German Association for Pattern Recognition (DAGM)*, Munich, Germany, 2001, pp. 369–376.

[16] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007.

[17] D. M. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," in *Proc. of the International Conference on Computer Vision (ICCV'99)*, Kerkyra, Greece, 1999, pp. 87–93.

[18] J. Giebel, D. M. Gavrila, and C. Schnörr, "A Bayesian framework for multi-cue 3d object tracking," in *Proc. of the European Conference on Computer Vision (ECCV'04)*. Prague, Czech Republic: Springer-Verlag, May 11-14 2004.

[19] T. Heap and D. Hogg, "Wormholes in shape space: Tracking through discontinuous changes in shape," in *Proc. of the International Conference on Computer Vision*, Bombay, India, 1998, pp. 344–349.

[20] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[21] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," in *Proc. of the European Conference on Computer Vision*, vol. 1, 1998, pp. 893–908.

[22] M. Isard and J. MacCormick, "BraMBLE: a Bayesian multiple-blob tracker," in *Proc. of the International Conference on Computer Vision*, vol. II, 2001, pp. 34–41.

[23] M. J. Jones and T. Poggio, "Multidimensional morphable models," in *Proc. of the International Conference on Computer Vision*, 1998, pp. 683–688.

[24] H.-G. Kang and D. Kim, "Real-time multiple people tracking using competitive condensation," *Pattern Recognition*, vol. 38, no. 7, pp. 1045–1058, 2005.

[25] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[26] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. of the European Conference on Computer Vision (ECCV'00)*, vol. II. Dublin, Ireland: Springer-Verlag, June 2000, pp. 3–19.

[27] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1863–1868, 2006.

[28] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. of the European Conference on Computer Vision (ECCV'04)*, vol. I. Prague, Czech Republic: Springer-Verlag, May 11-14 2004, pp. 28–39.

[29] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[30] V. Philomin, R. Duraiswami, and L. S. Davis, "Quasi-random sampling for condensation," in *Proc. of the European Conference on Computer Vision (ECCV'00)*. Dublin, Ireland: Springer-Verlag, June 2000, pp. 134–149.

[31] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a pose: Tracking people by finding stylized poses," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, San Diego, CA, USA, 2005, pp. 271–278.

[32] S. Roth, L. Sigal, and M. J. Black, "Gibbs likelihoods for bayesian tracking," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, June 2004, pp. 886–893.

[33] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[34] H. Sidenbladh and M. J. Black, "Learning the statistics of people in images and video," *International Journal of Computer Vision*, vol. 54, no. 1/2/3, pp. 183–209, 2003.

[35] A. Soto and P. Khosla, "Probabilistic adaptive agent based system for dynamic state estimation using multiple visual cues," in *10th Int'l Symposium of Robotics Research (ISRR 2001)*, Lorne, Victoria, Australia, November 9-12 2001.

[36] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," *Machine Vision and Applications*, vol. 14, no. 1, pp. 50–58, 2003.

[37] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Filtering using a tree-based estimator," in *Proc. of the International Conference on Computer Vision (ICCV'03)*, vol. II, Nice, France, October 2003, pp. 1063–1070.

[38] K. Toyama and A. Blake, "Probabilistic tracking with exemplars in a metric space," *International Journal of Computer Vision*, vol. 48, no. 1, pp. 9–19, 2002.

[39] W. van der Mark and D. M. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 38–50, 2006.

[40] P. A. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[41] C. Wöhler and J. Anlauf, "An adaptable time-delay neural-network algorithm for image sequence analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1531–1536, 1999.

[42] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, 2006, pp. 951–958.

[43] Y. Wu and T. Yu, "A field model for human detection and tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 753–765, 2006.

[44] J. Zhang, R. Collins, and Y. Liu, "Bayesian body localization using mixture of nonlinear shape models," in *Proc. of the International Conference on Computer Vision (ICCV'05)*, October 2005, pp. 725–732.

[45] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1208–1221, September 2004.