Tracking of Multiple Targets Using Online Learning for Reference Model Adaptation

Franz Pernkopf

***Abstract*—Recently, much work has been done in multiple object tracking on the one hand and on reference model adaptation for a single-object tracker on the other side. In this paper, we do both tracking of multiple objects (faces of people) in a meeting scenario and online learning to incrementally update the models of the tracked objects to account for appearance changes during tracking. Additionally, we automatically initialize and terminate tracking of individual objects based on low-level features, i.e., face color, face size, and object movement. Many methods unlike our approach assume that the target region has been initialized by hand in the first frame. For tracking, a particle filter is incorporated to propagate sample distributions over time. We discuss the close relationship between our implemented tracker based on particle filters and genetic algorithms. Numerous experiments on meeting data demonstrate the capabilities of our tracking approach. Additionally, we provide an empirical verification of the reference model learning during tracking of indoor and outdoor scenes which supports a more robust tracking. Therefore, we report the average of the standard deviation of the trajectories over numerous tracking runs depending on the learning rate.**

***Index Terms*—Genetic algorithms (GAs), multiple target tracking, particle filter, reference model learning, visual tracking.**

I. Introduction

Visual tracking of multiple objects is concerned with maintaining the correct identity and location of a variable number of objects over time irrespective of occlusions and visual alterations. Lim *et al.* [1] differentiate between intrinsic and extrinsic appearance variability including pose variation, shape deformation of the object and illumination change, camera movement, occlusions, respectively.

In the past few years, particle filters have become the method of choice for tracking. Isard and Blake [2] introduced particle filtering (condensation algorithm). Many different sampling schemes have been suggested in the meantime. An overview about sampling schemes of particle filters and the relation to Kalman filters is provided in [3].

Recently, the main emphasis is on simultaneously tracking multiple objects and on online learning to adapt the reference models to the appearance changes, e.g., pose variation, illumination change. Lim *et al.* [1] introduce a single-object tracker, where the target representation—a low-dimensional eigenspace representation—is incrementally updated to model the appearance variability. They assume, like most tracking algorithms, that the target region is initialized by hand in the first frame. Jepson *et al.* [4] use a Gaussian mixture model which is adapted using an online expectation maximization (EM) algorithm to account for appearance changes. Their $\mathcal{WSL}$ tracker uses a wavelet-based object model which is useful for tracking objects where regions of the objects (i.e., faces) are stable while other regions vary, e.g., mouth. McKenna *et al.* [5] employ Gaussian mixtures of the color distributions of the objects as adaptive model. In [6], simple color histograms are used to represent the objects (similar as in [7]). However, they introduce a simple update of the histograms to overcome the appearance changes of the object. All the aforementioned articles are focused on tracking a singe object. For tracking multiple objects, most algorithms belong to one of the following three categories: 1) Multiple instances of a single-object tracker are used [8]. 2) All objects of interest are included in the state space [9]. A fixed number of objects is assumed. Varying number of objects result in a dynamic change of the dimension of the state space. 3) Most recently, the framework of particle filters is extended to capture multiple targets using a mixture model [10]. This mixture particle filter—where each component models an individual object—enables interaction between the components by the importance weights. In [11], this approach is extended by the Adaboost algorithm to learn the models of the targets. The information from Adaboost enables detection of objects entering the scene automatically. The mixture particle filter is further extended in [12] to handle mutual occlusions. They introduce a rectification technique to compensate for camera motions, a global nearest neighbor data association method to correctly identify object detections with existing tracks, and a mean-shift algorithm which accounts for more stable trajectories for reliable motion prediction.

In this paper, we do both tracking of multiple persons in a meeting scenario and online adaptation of the models to account for appearance changes during tracking. The tracking is based on low-level features such as skin color, object motion, and object size. Based on these features, automatic initialization and termination of objects are performed. The aim is to use as little prior knowledge as possible. For tracking, a particle filter is incorporated to propagate sample distributions over time. Our implementation is related to the *dual estimation* problem [13], where both the states of multiple objects and the parameters of the reference models are simultaneously estimated given the observations. At every time step, the particle filter estimates the states using the observation likelihood of the current reference models while the online learning of the reference models is based on the current state estimates. Additionally, we discuss

Manuscript received July 24, 2007; revised January 30, 2008. This work was supported by the Austrian Science Fund under Project P19737-N15.

The author is with the Department of Electrical Engineering, Laboratory of Signal Processing and Speech Communication, Graz University of Technology, 8010 Graz, Austria (e-mail: pernkopf@tugraz.at).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.927281

the similarity between our implemented tracker based on particle filters and genetic algorithms (GAs). We want to emphasize this close connection since approaches what have independently been developed in one community might turn out to be very useful for the other community and vice versa. Numerous experiments on meeting data demonstrate the capabilities of our tracking approach. Additionally, we empirically show that the adaptation of the reference model during tracking of a indoor and outdoor scenes results in a more robust tracking. For this, we report the average of the standard deviation of the trajectories over numerous independent tracking runs depending on the learning rate.

The proposed approach differs from previous methods in several aspects. Recently, much work has been done in multiple object tracking on the one hand side and on reference model adaptation for a single-object tracker on the other side. In this paper, we do both tracking of multiple objects and online learning to incrementally update the representation of the tracked objects to model appearance changes. We use the Jensen–Shannon (JS) divergence [14] to measure the similarity between the tracked object and its reference model. Additionally, we discuss its advantages compared to the Kullback–Leibler divergence [15] and the Bhattacharyya similarity coefficient [16]. We automatically initialize and terminate tracking of individual objects based on low-level features, i.e., face color, face size, and object movement. Many methods unlike our approach assume that the target region has been initialized in the first frame.

This paper is organized as follows. Section II introduces the particle filter for multiple object tracking, the state-space dynamics, the observation model, automatic initialization and termination of objects, and the online learning of the models for the tracked objects. Section II-G summarizes the implemented tracker on the basis of pseudocode. Section III sketches the relationship to GA. The tracking results on a meeting scenario and for indoor/outdoor scenes are presented in Section IV. Additionally, we provide empirical verification of the reference model learning in this section. Section V concludes this paper.

II. Tracking Using Particle Filters

In many applications the states of a dynamic system have to be estimated from a time series of noisy observations. The Kalman filter [13], [17] is a linear dynamical system [18] that provides a linear time-discrete filter that estimates the states online over time once observations become available. This filter is recursive in a sense that each current state estimate is computed from the previous estimate and the current observed data. In contrast to linear dynamical systems, the hidden Markov model [19] assumes a discrete state space. Recently, many extentions of the basic linear dynamical system have been proposed [13] to overcome the assumption of the linear-Gaussian model used for the observations and state transition, e.g., the extended Kalman filter, unscented Kalman filter, or the switching state-space model [20]. Another approach for filtering is to use sequential Monte Carlo methods which are also known as particle filters [21]. They are capable to deal with any nonlinearity or distribution.

A. Particle Filter

A particle filter is capable to deal with nonlinear non-Gaussian processes and has become popular for visual tracking. For tracking, the probability distribution that the object is in state $\mathbf{x}_t$ at time t given the observations $\mathbf{y}_{0:t}$ up to time t is of interest. Hence, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ has to be constructed starting from the initial distribution $p(\mathbf{x}_0|\mathbf{y}_0) = p(\mathbf{x}_0)$. In Bayesian filtering, this can be formulated as iterative recursive process consisting of the prediction step

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})dx_{t-1} \quad (1)$$

and of the filtering step

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})dx_t} \quad (2)$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the dynamic model describing the state-space evolution which corresponds to the evolution of the tracked object (see Section II-B) and $p(\mathbf{y}_t|\mathbf{x}_t)$ is the likelihood of an observation $\mathbf{y}_t$ given the state $\mathbf{x}_t$ (see observation model in Section II-C).

In particle filters $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ of the filtering step is approximated by a finite set of weighted samples, i.e., the particles, $\{\mathbf{x}_t^m, w_t^m\}_{m=1}^M$, where M is the number of samples. Particles are sampled from a proposal distribution $\mathbf{x}_t^m \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{0:t})$ (importance sampling) [3]. In each iteration, the importance weights are updated according to

$$w_t^m \propto \frac{p\left(\mathbf{y}_t|\mathbf{x}_t^m\right) p\left(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m\right)}{q\left(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m, \mathbf{y}_{0:t}\right)} w_{t-1}^m \quad \sum_{m=1}^{M} w_t^m = 1. \quad (3)$$

One simple choice for the proposal distribution is to take the prior density $q(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m, \mathbf{y}_{0:t}) = p(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m)$ (bootstrap filter). Hence, the weights are proportional to the likelihood model $p(\mathbf{y}_t|\mathbf{x}_t^m)$

$$w_t^m \propto p\left(\mathbf{y}_t|\mathbf{x}_t^m\right) w_{t-1}^m. \quad (4)$$

The posterior filtered density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be approximated as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^m \delta\left(\mathbf{x}_t - \mathbf{x}_t^m\right) \quad (5)$$

where $\delta(\mathbf{x}_t - \mathbf{x}_t^m)$ is the Dirac delta function with mass at $\mathbf{x}_t^m$.

We use resampling to reduce the *degeneracy problem* [3], [21]. We resample the particles $\{\mathbf{x}_t^m\}_{m=1}^M$ with replacement M times according to their weights w_t^m. The resulting particles $\{\mathbf{x}_t^m\}_{m=1}^M$ have uniformly distributed weights $w_t^m = 1/M$. Similar to the sampling importance resampling filter [3], we resample in every time step. This simplifies (4) to $w_t^m \propto p(\mathbf{y}_t|\mathbf{x}_t^m)$ since $w_{t-1}^m = 1/M \quad \forall m$.

In the meeting scenario, we are interested in tracking the faces of multiple people. We treat the tracking of multiple objects completely independent, i.e., we assign a set of M particles to each tracked object k as $\{\{\mathbf{x}_t^{m,k}\}_{m=1}^M\}_{k=1}^K$, where K is the total number of tracked objects which dynamically

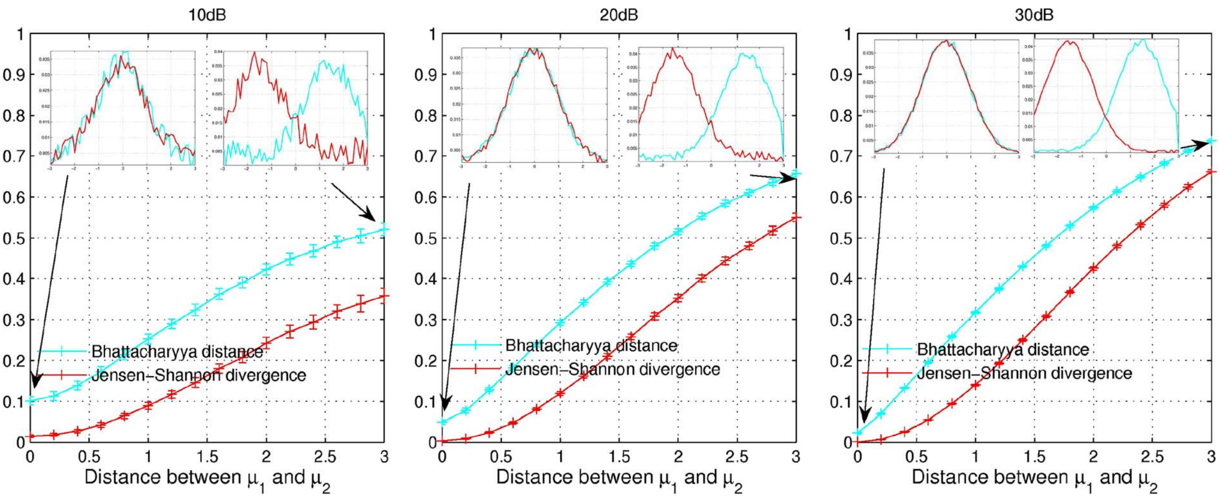


Fig. 1. JS divergence and Bhattacharyya similarity coefficient between two distributions estimated from samples. We added noise at a level of 10, 20, and 30 dB to the distributions.

changes over time. Hence, we use multiple instances of a single-object tracker similar to [8].

B. State-Space Dynamics

The state sequence evolution $\{\mathbf{x}_t; t \in \mathbb{N}\}$ is assumed to be a second-order autoregressive process which is used instead of the first-order formalism $(p(\mathbf{x}_t|\mathbf{x}_{t-1}))$ introduced in the previous section. The second-order dynamics can be written as first order by extending the state vector at time t with elements from the state vector at time $t-1$.

We define the state vector at time t as $\mathbf{x}_t = [x_t \;\; y_t \;\; s_t^x \;\; s_t^y]^{\mathrm{T}}$. The location of the target at t is given as x_t, y_t, respectively, and s_t^x, s_t^y denote the scale of the tracked region in the $x \times y$ image space. In our tracking approach, the transition model corresponds to

$$\mathbf{x}_{t+1}^{m,k} = \mathbf{x}_t^{m,k} + C\mathbf{v}_t + \frac{D}{2M}\sum_{m'=1}^{M}\left(\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1}^{m',k}\right) \qquad (6)$$

where $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I})$ is a simple Gaussian random noise model and the term $1/2M\sum_{m'=1}^{M}(\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1}^{m',k})$ captures the linear evolution of object k from the particles of the previous time step. Factor D models the influence of the linear evolution, e.g., D is set to 0.5. The parameters of the random noise model are set to $C = \mathrm{diag}([10 \quad 10 \quad 0.03 \quad 0.03])$ with the units of [pixel/frame], [pixel/frame], [1/frame], and [1/frame], respectively.

C. Observation Model

The shape of the tracked region is determined to be an ellipse [4] since the tracking is focused on the faces of the individuals. We assume that the principal axes of the ellipses are aligned with the coordinate axes of the image. Similarly to [7], we use the color histograms for modeling the target regions. Therefore, we transform the image into the hue-saturation-value (HSV) space [22]. For the sake of readability, we abuse the notation and write the particle $\mathbf{x}_t^{m,k}$ as $\mathbf{x}_t$ in this section. We build an individual histogram for hue (H) $h_{\mathrm{H}}^{\mathbf{x}_t}$, saturation (S) $h_{\mathrm{S}}^{\mathbf{x}_t}$, and value (V) $h_{\mathrm{V}}^{\mathbf{x}_t}$ of the elliptic candidate region at $\mathbf{x}_t$. The length of the principal axes of the ellipse are $A_{\mathrm{ref}}^k s_t^x$ and $B_{\mathrm{ref}}^k s_t^y$, respectively, where A_{ref}^k and B_{ref}^k are the length of the ellipse axes of the reference model of object k.

The likelihood of the observation model (likelihood model) $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$ must be large for candidate regions with a histogram close to the reference histogram. Therefore, we introduce the JS divergence [14] to measure the similarity between the normalized candidate and reference histograms, $h_{\mathrm{c}}^{\mathbf{x}_t}$ and $h_{\mathrm{c,ref}}^k$, $c \in \{H, S, V\}$, respectively. Since, the JS divergence is defined for probability distributions the histograms are normalized, i.e., $\sum_N h_{\mathrm{c}}^{\mathbf{x}_t} = 1$, where N denotes the number of histogram bins. In contrast to the Kullback–Leibler divergence [15], the JS divergence is symmetric and bounded between 0 and 1. The JS divergence between the normalized histograms is defined as

$$\mathrm{JS}_\pi\left(h_{\mathrm{c}}^{\mathbf{x}_t}, h_{\mathrm{c,ref}}^k\right) = H\left(\pi_1 h_{\mathrm{c}}^{\mathbf{x}_t} + \pi_2 h_{\mathrm{c,ref}}^k\right) - \pi_1 H\left(h_{\mathrm{c}}^{\mathbf{x}_t}\right) - \pi_2 H\left(h_{\mathrm{c,ref}}^k\right) \qquad (7)$$

where $\pi_1 + \pi_2 = 1, \pi_i \geq 0$ and the function $H(\cdot)$ is the entropy [15]. The JS divergence is computed for the histograms of the H, S, and V space, and the observation likelihood is

$$p\left(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k}\right) \propto \exp -\lambda\left[\sum_{c\in\{H,S,V\}} \mathrm{JS}_\pi\left(h_{\mathrm{c}}^{\mathbf{x}_t^{m,k}}, h_{\mathrm{c,ref}}^k\right)\right] \qquad (8)$$

where parameter λ is chosen to be five and the weight π_i is uniformly distributed. The number of bins of the histograms is set to $N = 50$. The JS divergence provides a lower and upper bound to the Bayes error and π_1 and π_2 can be viewed as *a priori* probabilities in a classification problem [14].

In contrast to the often used Bhattacharyya similarity coefficient $\sqrt{1 - \sum_N \sqrt{h_{\mathrm{c}}^{\mathbf{x}_t^{m,k}} h_{\mathrm{c,ref}}^k}}$ [16], the JS divergence is not so sensitive to local perturbations in the histogram (noise). This is shown in Fig. 1 where we compute the JS divergence and Bhattacharyya similarity coefficient on synthetic data. Therefore, we sample two Gaussian distributions with $\mu_1 = -\mu_2$, where μ_1 varies from 0 to 1.5, and unit variance. Noise is added to those distributions at a level of 10, 20, and 30 dB. Plots are averaged over 100 independent simulations.

D. Automatic Initialization of Objects

If an object enters the frame, a set of M particles and a reference histogram for this object have to be initialized. Basically, the initialization of objects is automatically performed using the following simple low-level features.

1) Motion: The images are transformed to gray scale $I^G_{x_t,y_t}$. The motion feature is determined for each pixel located at x, y by the standard deviation over a time window T_w as $\sigma^t_{x,y} = \sigma(I^G_{x_{t-T_w:t},y_{t-T_w:t}})$. Applying an adaptive threshold $T_{\text{motion}} = 1/10 \max_{x,y\in I^G} \sigma^t_{x,y}$ pixels with a value larger T_{motion} belong to regions where movement happens. However, $\max_{x,y\in I^G} \sigma^t_{x,y}$ has to be sufficiently large so that motion exists at all. A binary motion image $I^{B_{\text{motion}}}_{x_t,y_t}$ after morphological closing is shown in Fig. 2.

2) Skin Color: The skin color of the people is modeled by a Gaussian mixture model [23] in the HSV color space. A Gaussian mixture model $p(\mathbf{z}|\mathbf{\Theta})$ is the weighted sum of $L > 1$ Gaussian components, $p(\mathbf{z}|\mathbf{\Theta}) = \sum_{l=1}^{L} \alpha_l \mathcal{N}(\mathbf{z}|\mu_l, \mathbf{\Sigma}_l)$, where $\mathbf{z} = [z_{\text{H}}, z_{\text{S}}, z_{\text{V}}]^{\text{T}}$ is the 3-D color vector of one image pixel, α_l corresponds to the weight of each component $l = 1, \ldots, L$. These weights are constrained to be positive $\alpha_l \geq 0$ and $\sum_{l=1}^{L} \alpha_l = 1$. The Gaussian mixture is specified by the set of parameters $\mathbf{\Theta} = \{\alpha_l, \mu_l, \mathbf{\Sigma}_l\}_{l=1}^{L}$. These parameters are determined by the EM algorithm [24] from a face database.

   Image pixels $\mathbf{z} \in I^{\text{HSV}}_{x_t,y_t}$ are classified according to their likelihood $p(\mathbf{z}|\mathbf{\Theta})$ using a threshold T_{skin}. The binary map $I^{B_{\text{skin}}}_{x_t,y_t}$ filtered with a morphological closing operator is presented in Fig. 2.

3) Object Size: We initialize a new object only for skin-colored moving regions with a size larger than T_{Area}. Additionally, we do not allow initialization of a new set of particles in regions where currently an object is tracked. To this end, a binary map $I^{B_{\text{prohibited}}}_{x_t,y_t}$ represents the areas where initialization is prohibited. The binary combination of all images $I^B_{x_t,y_t} = I^{B_{\text{motion}}}_{x_t,y_t} \cap I^{B_{\text{skin}}}_{x_t,y_t} \cap \overline{I^{B_{\text{prohibited}}}_{x_t,y_t}}$ is used for extracting regions with an area larger T_{Area}. Target objects are initialized for those regions, i.e., the ellipse size $(A^k_{\text{ref}}, B^k_{\text{ref}})$ and the histograms $h^k_{\text{c,ref}}, c \in \{H, S, V\}$ are determined from the region of the bounding ellipse.

Fig. 2 shows an example for the initialization of a new object. The original image $I^{\text{HSV}}_{x_t,y_t}$ is presented in (a). A person entering from the right side should be initialized. A second person in the middle of the image is already tracked. The binary images of the thresholded motion $I^{B_{\text{motion}}}_{x_t,y_t}$ and the skin-colored areas $I^{B_{\text{skin}}}_{x_t,y_t}$ are shown in (b) and (c), respectively. The reflections at the table and the movement of the curtain produce noise in the motion image. The color of the table and chairs intersects with the skin-color model. To guarantee successful initialization the lower part of the image—the region of the chairs and desk—has to be excluded. This is reasonable since nobody can enter in this area. Also, tracking is performed in the area above the chairs only. Finally, the region of the new initialized object is presented as ellipse in (d). Resizing of the images is performed for computing the features to speed up the initialization of objects.

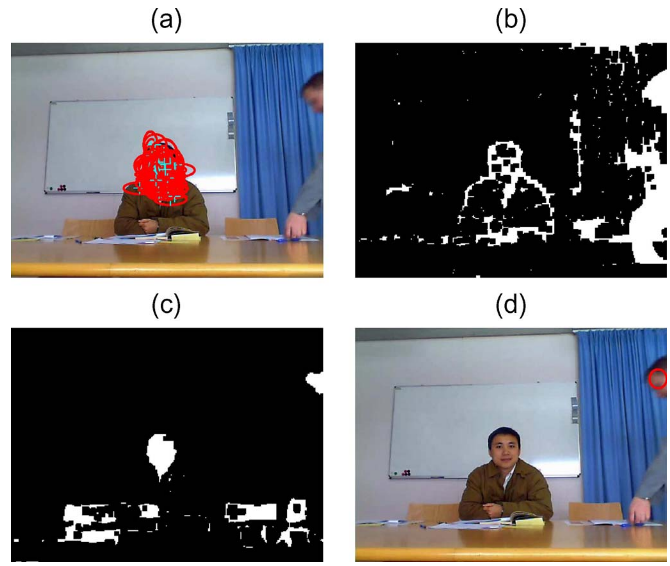


Fig. 2. Initialization of a new object. (a) Original image with one object already tracked. (b) Binary image of the thresholded motion $I^{B_{\text{motion}}}_{x_t,y_t}$. (c) Binary image of the skin-colored areas $I^{B_{\text{skin}}}_{x_t,y_t}$. (d) Image with region of initialized object.

1) Shortcomings: The objects are initialized when they enter the image. The reference histogram is taken during this initialization. There are the following shortcomings during initialization.

1) The camera is focused on the people sitting at the table and not on people walking behind the chairs. This means that walking persons appear blurred.
2) Entering persons are moving relatively fast. This also results in a degraded image quality (blurring).
3) During initialization, we normally get the side view of the person's head. When the person sits at the table the reference histogram is not necessarily a good model for the frontal view.

To deal with these shortcomings, we propose online learning to incrementally update the reference models of the tracked objects over time (see Section II-F). We perform this only in cases where no mutual occlusions between the tracked objects are existent.

E. Automatic Termination of Objects

Termination of particles is performed if the observation likelihood $p(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t)$ at state $\mathbf{x}^{m,k}_t$ drops below a predefined threshold T_{Kill} (e.g., 0.001), i.e.,

$$p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right) = \begin{cases} 0, & \text{if } p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right) < T_{\text{Kill}} \\ p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right), & \text{otherwise.} \end{cases} \tag{9}$$

Particles with zero probability do not survive during resampling. If the tracked object leaves the field of view all M

particles of an object k are removed, i.e., $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k}) = 0$ for all particles of object k.

F. Incremental Learning of Object Models

To handle the appearance change of the tracked objects over time, we use online learning to adapt the reference histograms $h_{\text{c,ref}}^k, \quad c \in \{H, S, V\}$ (similar to [6]) and ellipse size A_{ref}^k and B_{ref}^k. Therefore, a learning rate α is introduced, and the model parameters for target object k are updated according to

$$h_{\text{c,ref}}^k = \alpha \hat{h}_{\text{c}}^k + (1-\alpha) h_{\text{c,ref}}^k, \qquad c \in \{H, S, V\} \tag{10}$$

$$A_{\text{ref}}^k = \alpha \hat{A}^k + (1-\alpha) A_{\text{ref}}^k \tag{11}$$

$$B_{\text{ref}}^k = \alpha \hat{B}^k + (1-\alpha) B_{\text{ref}}^k \tag{12}$$

where $\hat{h}_{\text{c}}^k$ denotes the histogram and $\hat{A}^k$ and $\hat{B}^k$ are the principal axes of the bounding ellipse of the nonoccluded (i.e., no mutual occlusion between tracked objects) skin-colored region of the corresponding tracked object k located at $\{\mathbf{x}_t^{m,k}\}_{m=1}^M$. Again, this region has to be larger than T_{Area}. No update of the reference models is performed in the case where occlusion between the tracked objects occurs or the skin-colored region is not large enough. The latter condition is a simple way to ensure that the model update is only conducted for faces. This simplistic assumption can be appropriately extended by integrating more advanced face models.

The learning rate α introduces an *exponential forgetting process*, i.e., the contribution of a specific object exponentially decreases as it recedes into the past. Currently, the learning rate (value between 0 and 1) is fixed (a good value has been selected during experiments). However, α could be adapted depending on the dynamics of the scene.

Algorithm 1 Particle Filter Tracking
Input: $I_{x_{0:T}, y_{0:T}}^{\text{HSV}}$ (Color image sequence $0:T$),
  Skin-color model $\mathbf{\Theta}$
Parameters: $M, N, \lambda, C, D, T_w, T_{\text{motion}}, T_{\text{skin}}, T_{\text{Area}}, T_{\text{Kill}}, \alpha$
Output: $\{\{\mathbf{x}_{0:T}^{m,k}\}_{m=1}^M\}_{\forall k}$
$t \leftarrow 0$
$k \leftarrow 0$
while InitObjects **do**
  $k \leftarrow k+1$
  Obtain: $h_{\text{c,ref}}^k : c \in \{H, S, V\}$, A_{ref}^k, B_{ref}^k, $\mathbf{x}_{\text{ref}}^k$
  $\mathbf{x}_{t+1}^{m,k} \leftarrow \mathbf{x}_{\text{ref}}^k + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$ (Generate particles)
end while
$K \leftarrow k$
for $t = 1$ **to** T **do**
  $w_t^{m,k} \propto p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$
    $\forall k = 1, \ldots, K \quad \forall m = 1, \ldots, M$
  while KillObjects **do**
    $k \leftarrow$ Determine object to terminate
    Remove M particles $x_t^{m,k}$ of object k
    Remove reference histogram and ellipse size:
      $h_{c,ref}^k : c \in \{H, S, V\}, A_{\text{ref}}^k, B_{\text{ref}}^k$
    $K \leftarrow K - 1$
  end while
  for $k = 1$ **to** K **do**
    $w_t^{m,k} \leftarrow w_t^{m,k} / \sum_{m'=1}^M w_t^{m',k} \quad \forall m = 1, \ldots, M$
    $\{\mathbf{x}_t^{m,k}\}_{m=1}^M \leftarrow$ Resampling
      (with replacement): $\{\mathbf{x}_t^{m,k}, w_t^{m,k}\}_{m=1}^M$
    $\mathbf{x}_{t+1}^{m,k} \leftarrow \mathbf{x}_t^{m,k} + C\mathbf{v}_t + (D/2M) \sum_{m'=1}^M (\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1})^{m',k}$
      $\forall m = 1, \ldots, M$ (Apply state-space dynamics)
    if OnlineUpdate **then**
      Determine: $\hat{h}_{\text{c}}^k : c \in \{H, S, V\}$, $\hat{A}^k$, $\hat{B}^k$
      $h_{\text{c,ref}}^k \leftarrow \alpha \hat{h}_{\text{c}}^k + (1-\alpha) h_{\text{c,ref}}^k \quad c \in \{H, S, V\}$
      $A_{\text{ref}}^k \leftarrow \alpha \hat{A}^k + (1-\alpha) A_{\text{ref}}^k$
      $B_{\text{ref}}^k \leftarrow \alpha \hat{B}^k + (1-\alpha) B_{\text{ref}}^k$
    end if
  end for
  while InitObjects **do**
    $K \leftarrow K+1$
    Obtain: $h_{\text{c,ref}}^K : c \in \{H, S, V\}$, A_{ref}^K, B_{ref}^K, $\mathbf{x}_{\text{ref}}^K$
    $\mathbf{x}_{t+1}^{m,K} \leftarrow \mathbf{x}_{\text{ref}}^K + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$ (Generate particles)
  end while
end for

Fig. 3. Tracking scene. We track and initialize objects in the red rectangle.

G. Implemented Tracker

In the following, we sketch our tracking approach for multiple objects (see Algorithm 1). The binary variable *InitObject* denotes that a new object for tracking has been detected. *KillObject* is set if an object should be terminated. *OnlineUpdate* indicates that object k located at $\{\mathbf{x}_t^{m,k}\}_{m=1}^M$ is nonoccluded, and the area of the skin-colored region is larger than T_{Area}, i.e., we perform online learning for reference model k.

Our implementation is related to the *dual estimation* problem [13], where both the states of multiple objects $\mathbf{x}_t^{m,k}$ and the parameters of the reference models are simultaneously estimated given the observations. At every time step, the particle filter estimates the states using the observation likelihood of the current reference models, while the online learning of the reference models is based on the current state estimates.

III. Relationship to GAs

GAs are optimization algorithms founded upon the principles of natural evolution discovered by Darwin. In nature, individuals have to adapt to their environment in order to survive in

Fig. 4. Tracking of people. Frames: 1, 416, 430, 449, 463, 491, 583, 609, 622, 637, 774, 844, 967, 975, 1182, 1400 (the frame number is assigned from left to right and top to bottom).

a process of further development. An introduction of GAs can be found in [25] and [26]. GA are stochastic procedures which have been successfully applied in many optimization tasks. GA operate on a population of potential solutions applying the principle of *survival of the fittest individual* to produce better and better approximations to the solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and assembling them together using operators inspired from nature. This leads to the evolution of individuals that are better suited to their environment than the parent individuals they were created from. GA model the natural processes, such as selection, recombination, and mutation. Starting from an initial population $P(0)$, the sequence $P(0)$, $P(1), \ldots, P(t)$, $P(t+1)$ is called population sequence or evolution. The end of an artificial evolution process is reached once the termination condition is met, and the result of the optimization task is available.

In this section, we want to point to the close relationship between GA and our particle filter for tracking. This analogy has been mentioned in [27]. As suggested in Section II, we treat the tracking of multiple objects completely independent, i.e., we have a set of M particles for each object k. In the GA framework, we can relate this to k instantiations of GA, one for each tracked object. Hence, each particle $\mathbf{x}_t^m$ of object k represents one individual in the population $P(t)$ which is value encoded. The population size is M. A new genetic evolution process is started once a new object is initialized for tracking (InitObject). The evolution process of the GA is terminated either at the end of the video $(t = T)$ or when the set of individuals is not supported by the fitness value (KillObject). The observation likelihood $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$ denotes the fitness function to evaluate the individuals. However, the scope of GA for tracking is slightly different. GA are generally used to find a set of parameters for a given optimization task, i.e., the aim is to find the individual with the best fitness after the termination of the GA. Whereas, in the tracking case, the focus lies on the evolution of the individuals, i.e., the trajectory of the tracked object.

The selection operator directs the search toward promising regions in the search space. *Roulette Wheel Selection* [28] is a widely used selection method which is very similar to sampling with replacement as used in Section II. To each individual, a reproduction probability according to $w_t^m \leftarrow w_t^m / \sum_{m'=1}^{M} w_t^{m'}$ is assigned. A roulette wheel is constructed with a slot size corresponding to the individuals reproduction probability. Then, M uniformly distributed random numbers on the interval [0, 1] are drawn and distributed according to their value around the wheel. The slots where they are placed to compose the subsequent population $P(t)$. The state-space dynamics of the particle filter (see Section II-B) is modeled by the recombination and mutation operator.

The framework of the GA for tracking one object k is presented in Algorithm 1. The incremental learning of the reference model is omitted for the sake of brevity.

Fig. 5. Partial occlusions. Frames: 468, 616, 974, 4363 (the frame number is assigned from left to right and top to bottom).

Algorithm 2 GA Tracking
Input: $I^{\text{HSV}}_{x_{t:T},y_{t:T}}$ (Color image sequence $t : T$),
Parameters: M, N, λ, C, D, T_{Kill}
Output: $\{\mathbf{x}^m_{t:T}\}^M_{m=1}$ (set of particle sequences $t : T$)
Initialize population $P(t)$:
  $\mathbf{x}^m_t \leftarrow \mathbf{x}_{ref} + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$
while $\overline{\text{KillObject}} \cap t < T$ (Loop over image sequence) **do**
  Evaluate individuals:
    $w^m_t \leftarrow p(\mathbf{y}^m_t|\mathbf{x}^m_t) \quad \forall m = 1, \ldots, M$
  Selection $P(t)$:
    $\{\mathbf{x}^m_t\}^M_{m=1} \leftarrow$ (Sampling with replacement) $\{\mathbf{x}^m_t, w^m_t\}^M_{m=1}$
  Recombination $P(t + 1)$:
    $\mathbf{x}^m_{t+1} \leftarrow \mathbf{x}^m_t + (D/2M) \sum^M_{m'=1}(\mathbf{x}^{m'}_t - \mathbf{x}^{m'}_{t-1})$
    $\forall m = 1, \ldots, M$
  Mutation $P(t+1)$: $\mathbf{x}^m_{t+1} \leftarrow \mathbf{x}^m_{t+1} + C\mathbf{v}_t \quad \forall m=1, \ldots, M$
  $t \leftarrow t + 1$
end while

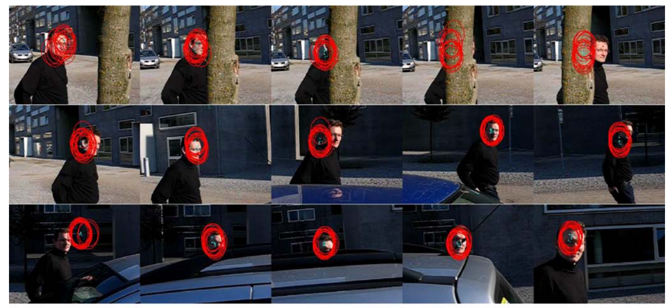
IV. Experiments

We present tracking results on meeting data in Section IV-A where we do both tracking of multiple persons and online adaptation of the reference models during tracking. In Section IV-B, we empirically show that the adaptation of the reference model during tracking (single object) of an indoor and outdoor scene results in a more robust tracking. Finally, in Section IV-C, tracking results using reference model adaptation for multiple objects of an outdoor scene are presented. For the outdoor scenes, we report the average standard deviation of the trajectories of independent tracking runs depending on the learning rate α.

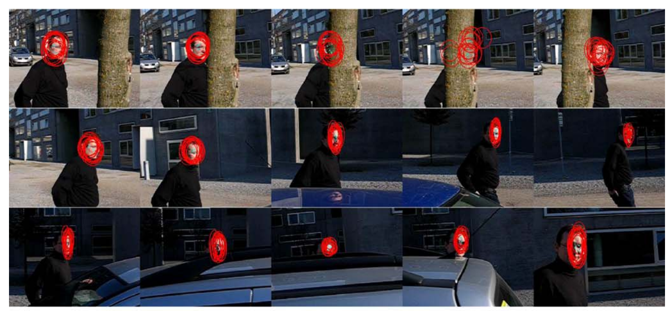
A. *Meeting Scenario*

The meeting room layout is shown in Fig. 3. The red rectangle [region of interest (ROI)] in the image marks the frame where tracking and initialization of objects are performed. People may enter and leave on both sides of the image. Currently, our tracker initializes a new target even if it enters from the

(a)



(b)



(c)

Fig. 6. Outdoor tracking. Frames: 7, 11, 12, 13, 14, 20, 42, 63, 80, 107, 136, 146, 158, 165, 192 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.2$).

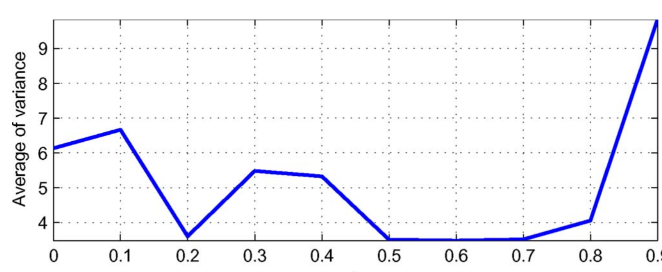


Fig. 7. Averaged standard deviation of the trajectories of 100 tracking runs depending on the reference model learning rate α.

bottom, e.g., a hand moving from the table into the ROI. The strong reflections at the table, chairs, and the white board cause noise in the motion image.

For testing the performance of our tracking approach, ten videos with ~7000 frames have been used. The resolution is 640 × 480 pixels. The meeting room is equipped with a table and three chairs. We have different persons in each video. The people are coming from both sides into the frame moving

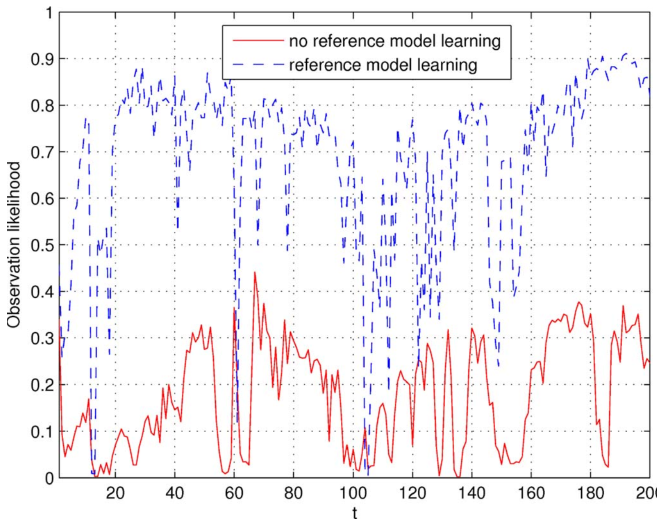


Fig. 8. Observation likelihood of outdoor sequence.

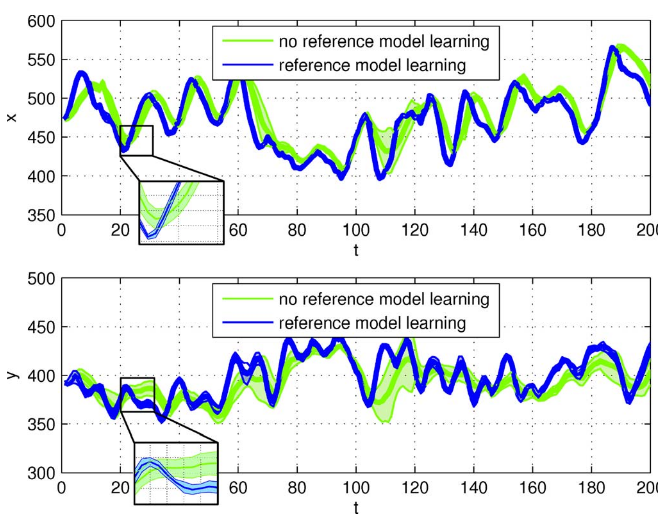


Fig. 9. Averaged trajectory with standard deviation in x and y of outdoor sequence (over ten runs).

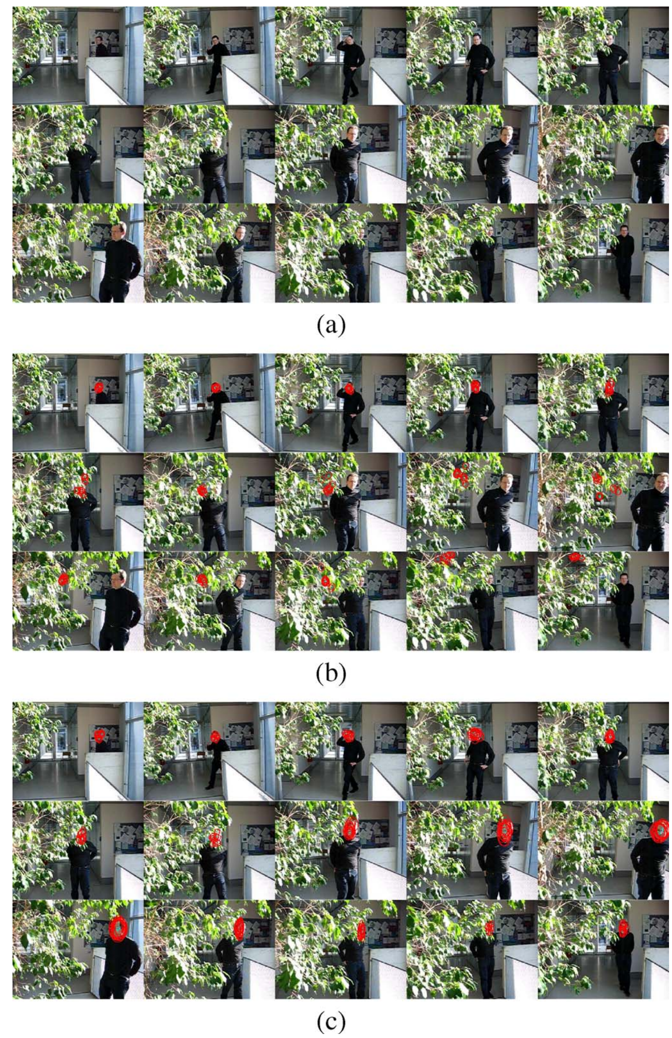


Fig. 10. Indoor tracking. Frames: 1, 12, 24, 31, 38, 41, 47, 54, 65, 71, 80, 107, 113, 120, 134 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.2$).

to chairs and sit down. After a short discussion, people are sequentially leaving the room, are coming back, sit down at different chairs and so on. At the beginning, people may already sit at the chairs. In this case, we have to automatically initialize multiple objects at the very first frame.

Fig. 4 shows the result of the implemented tracker for one video. All the initializations and terminations of objects are performed automatically. The appearance of an object changes over time. When entering the frame, we get the side view of the person's head. After sitting down at the table, we have a frontal view. We account for this by incrementally updating the reference histogram during tracking. We perform this only in the case where no mutual occlusions with other tracked objects are existent. The participants were successfully tracked over long image sequences.

First, the person on the left side stands up and leaves the room on the right side (frame 416–491). When walking behind the two sitting people, partial occlusions occur which do not cause problems. Next, the person on the right (frame 583–637) leaves the room on the left side. His face is again partially occluded by the person in the middle. Then, the person on the center chair leaves the room (frame 774). After that, a person on the right side enters and sits at the left chair (frame 844). At frame 967, a small person is entering and moving to the chair in the middle. Here, again, a partial occlusion occurs at frame 975, which is also tackled. Finally, a person enters from the right and sits down on the right chair (frame 1182, 1400). The partial occlusions are shown in Fig. 5. Also, the blurred face of the moving person in the back can be observed in this figure. The reference model adaptation enables a more robust tracking. If we do not update the models of the tracked objects over time, the tracking fails in case of these partial occlusions. In [29], occlusions are handled using multiple cameras for tracking participants in a meeting.

B. Reference Model Adaptation for Single-Object Tracking

In the following, we show the benefit of the reference model adaptation during tracking of a short indoor and outdoor sequence. In contrast to the meeting scenario, we restrict the

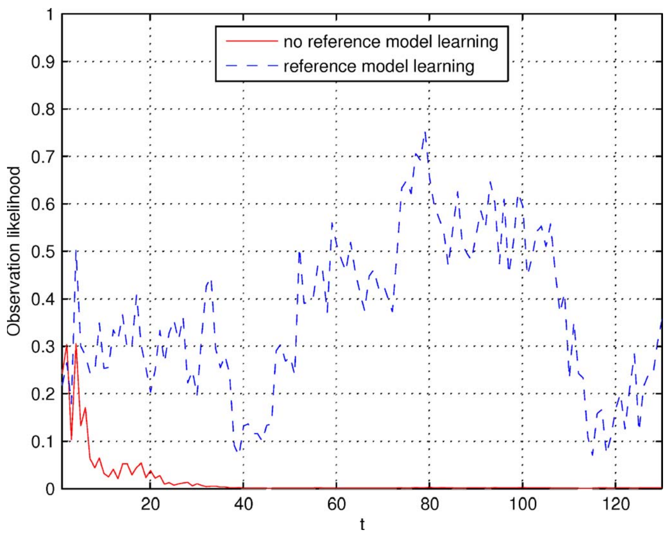


Fig. 11. Observation likelihood of indoor sequence.

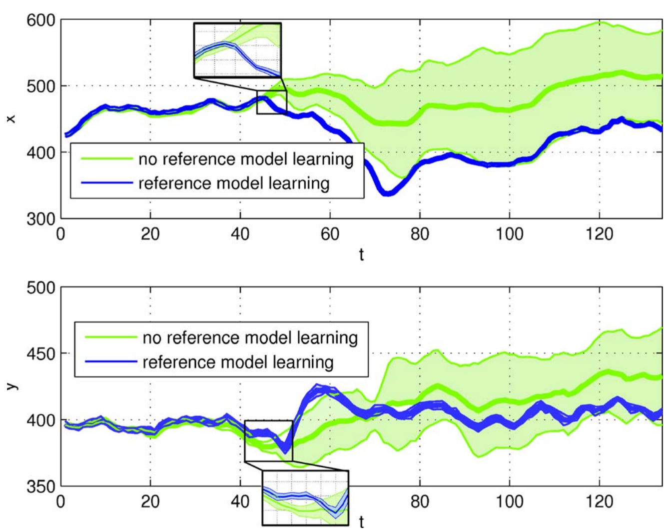


Fig. 12. Averaged trajectory with standard deviation in x and y of indoor sequence (over ten runs).

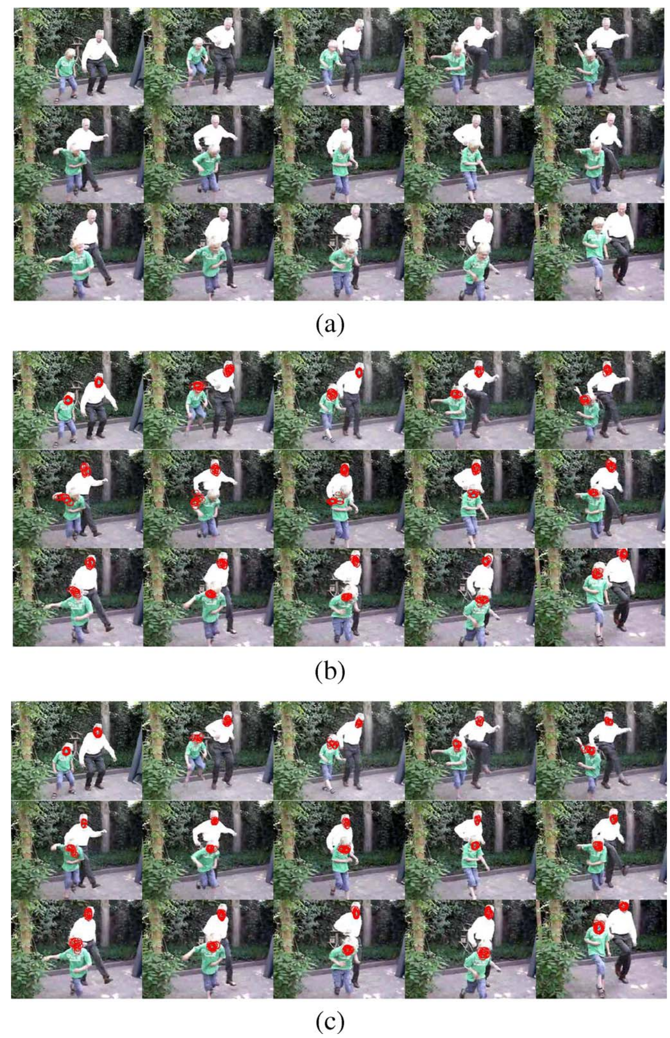


Fig. 13. Outdoor tracking of multiple objects. Frames: 1, 12, 30, 47, 49, 51, 53, 57, 59, 79, 105, 107, 109, 111, 149 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.1$).

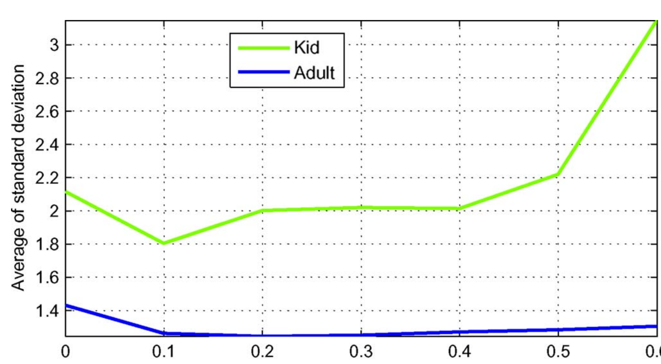


Fig. 14. Averaged standard deviation of the trajectories of ten tracking runs depending on the reference model learning rate α.

tracking to one single object, i.e., face. This means, in particular, that the automatic initialization and termination of the object is disabled. The object is initialized by hand in the first frame.

Fig. 6(a) shows a short outdoor sequence where a person is moving behind a tree and two cars with strongly changing lighting conditions. We have a total occlusion of the face in frames 12 and 13 and a partial occluded face in frames 146 to 165. We repeated the tracking without and with reference model learning ten times, and a typical result is shown in Fig. 6(b) and (c), respectively. We use $M = 50$ particles for tracking, whereas only 15 particles with the best observation likelihood are shown in the figures.

In Fig. 7, we present the average standard deviation of the trajectories over 100 tracking runs. The reference model learning rate α has been chosen in the range of $0, \ldots, 0.6$ (0 means that there is no learning). The optimal learning rate with respect to a low standard deviation of the trajectories over 100 independent runs is $\alpha = 0.2$ for this outdoor sequence.

Fig. 8 shows the observation likelihood of the best particle during tracking. At the complete occlusion (frames $t = 12$ and $t = 13$) and the partial occlusion (frames $t = 145, \ldots, 160$), the observation likelihood drops, however, with reference model learning a quick recovery is supported.

Fig. 9 summarizes the averaged trajectory with the standard deviation over ten different tracking runs performed for the outdoor scene. In the case of reference model learning, we can observe in the video sequences that the tracking of the face gives highly similar trajectories. The standard deviation is small and approximately constant over time. However, if no learning of the reference model is performed, the standard deviation is large in certain time segments. This leads to the conclusion that model adaptation results in a more robust tracking.

Fig. 10(a) shows an indoor video where a person is moving on a corridor, and a tree causes partial occlusion of the tracked face. Additionally, the lighting conditions are strongly varying. The face is partially occluded by the tree in frames 37–50 and 110–126. Again, the tracking without and with reference model learning is repeated ten times, and a typical result is shown in Fig. 10(b) and (c), respectively. Only 15 particles with the best observation likelihood are visualized. The parameter setting is the same as in the previous experiments. The tracker without reference model refinement fails during the first occlusion in all ten runs, whereas the tracker with online model update is successful in all cases. The optimal learning rate α is set to 0.2 (established during experiments).

This can be also observed in the observation likelihood of the best particle over time (see Fig. 11) and in the averaged trajectory over ten tracking results (see Fig. 12).

C. Reference Model Adaptation for Multiple Object Tracking

We show tracking results for an outdoor scene where a kid is showing an adult dancing steps (see Fig. 13). A typical tracking result without and with reference model learning is shown in Fig. 13(b) and (c), respectively. Again, $M = 50$ particles are used, whereas only 15 particles with the best observation likelihood are shown in the figures. Similar as in the previous section, we did a repeatability test, i.e., we tracked the objects over ten independent runs. The tracked objects are initialized by hand in the very first frame.

Fig. 14 shows the average standard deviation of the trajectories of ten tracking runs using a learning rate α in the range of $0, \ldots, 0.6$. The optimal learning rate for the *Kid* and the *Adult* is $\alpha = 0.1$ and $\alpha = 0.2$, respectively. Currently, α is fixed for the whole image sequence. Ideally, α could be adapted depending on the dynamics of the scene.

V. Conclusion

We propose a robust visual tracking algorithm for multiple objects (faces of people) in a meeting scenario based on low-level features as skin color, target motion, and target size. Based on these features, automatic initialization and termination of objects is performed. For tracking a sampling importance resampling, particle filter has been used to propagate sample distributions over time. Furthermore, we use online learning of the target models to handle the appearance variability of the objects. We discuss the similarity between our implemented tracker and GAs. Each particle represents an individual in the GA framework. The evaluation function incorporates the observation likelihood model and the individual selection process maps to the resampling procedure in the particle filter. The state-space dynamics is incorporated in the recombination and mutation operator of the GA. Numerous experiments on meeting data show the capabilities of the tracking approach. The participants were successfully tracked over long image sequences. Partial occlusions are handled by the algorithm. Additionally, we empirically show that the adaptation of the reference model during tracking of indoor and outdoor scenes results in a more robust tracking.

Future work concentrates on extending the tracker to other scenarios and to investigate an adaptive reference model learning rate α which depends on the dynamics of the scene. Furthermore, we aim to develop approaches for tackling occlusions.

Acknowledgment

This work was supported by the Austrian Science Fund (Project S106). The author would like to thank M. Grabner and M. Kepesi who collected the data during their involvement in the MISTRAL Project (www.mistral-project.at). The MISTRAL Project was funded by the Austrian Research Promotion Agency (www.ffg.at) within the strategic objective FIT-IT under Project 809264/9338. The author would also like to thank C. Kirchstätter for recording the indoor and outdoor video.

References

[1] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang, "Incremental learning for visual tracking," in *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2005.

[2] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[5] S. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixutre models," *Image Vis. Comput.*, vol. 17, no. 3/4, pp. 225–231, 1999.

[6] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[7] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. ECCV*, 2002, pp. 661–675.

[8] S. L. Dockstader and A. Tekalp, "Tracking multiple objects in the presence of articulated and occluded motion," in *Proc. Workshop Human Motion*, 2000, pp. 88–98.

[9] C. Hue, J.-P. Le Cadre, and P. Pérez, "Tracking multiple objects with particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 791–812, Jul. 2002.

[10] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multi-modality through mixture tracking," in *Proc. ICCV*, 2003, pp. 1110–1116.

[11] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. ECCV*, 2004, pp. 28–39.

[12] Y. Cai, N. de Freitas, and J. J. Little, "Robust visual tracking for multiple targets," in *Proc. ECCV*, 2006, pp. 107–118.

[13] S. Haykin, *Kalman Filtering and Neural Networks*. Hoboken, NJ: Wiley, 2001.

[14] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[15] T. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 142–149.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 82, pp. 34–45, 1960.
[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer Sci.+Bus. Media, 2006.
[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[20] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Comput.*, vol. 12, no. 4, pp. 963–996, 1998.
[21] A. Doucet, "On sequential Monte Carlo sampling methods for Bayesian filtering," Dept. Eng., Cambridge Univ., London, U.K., Tech. Rep. CUED/F-INFENG/TR. 310, 1998.
[22] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. London, U.K.: Int. Thomson, 1999.
[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ: Wiley, 2000.
[24] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. R. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 39, pp. 1–38, 1977.
[25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[26] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. London, U.K.: Oxford Univ. Press, 1996.
[27] T. Higuchi, "Monte Carlo filter using the genetic algorithm operators," *J. Stat. Comput. Simul.*, vol. 59, no. 1, pp. 1–23, Aug. 1997.
[28] L. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. Int. Conf. Genetic Algorithms Appl.*, 1987, pp. 14–21.
[29] H. Nait Charif and S. J. McKenna, "Tracking the activity of participants in a meeting," *Mach. Vis. Appl.*, vol. 17, no. 2, pp. 83–93, 2006.

Franz Pernkopf received the M.Sc. (Dipl.Ing.) degree in electrical engineering from the Graz University of Technology, Graz, Austria, in 1999, and the Ph.D. degree from the University of Leoben, Leoben, Austria, in 2002.

He was a Research Associate with the Department of Electrical Engineering, University of Washington, Seattle, from 2004 to 2006. Currently, he is an Assistant Professor with the Signal Processing and Speech Communication Laboratory, Graz University of Technology. His research interests include machine learning, Bayesian networks, feature selection, finite mixture models, vision, speech, and statistical pattern recognition.

Dr. Pernkopf was awarded the Erwin Schrödinger Fellowship in 2002.

Tracking of Multiple Targets Using Online Learning for Reference Model Adaptation

Franz Pernkopf

***Abstract*—Recently, much work has been done in multiple object tracking on the one hand and on reference model adaptation for a single-object tracker on the other side. In this paper, we do both tracking of multiple objects (faces of people) in a meeting scenario and online learning to incrementally update the models of the tracked objects to account for appearance changes during tracking. Additionally, we automatically initialize and terminate tracking of individual objects based on low-level features, i.e., face color, face size, and object movement. Many methods unlike our approach assume that the target region has been initialized by hand in the first frame. For tracking, a particle filter is incorporated to propagate sample distributions over time. We discuss the close relationship between our implemented tracker based on particle filters and genetic algorithms. Numerous experiments on meeting data demonstrate the capabilities of our tracking approach. Additionally, we provide an empirical verification of the reference model learning during tracking of indoor and outdoor scenes which supports a more robust tracking. Therefore, we report the average of the standard deviation of the trajectories over numerous tracking runs depending on the learning rate.**

***Index Terms*—Genetic algorithms (GAs), multiple target tracking, particle filter, reference model learning, visual tracking.**

I. Introduction

VISUAL tracking of multiple objects is concerned with maintaining the correct identity and location of a variable number of objects over time irrespective of occlusions and visual alterations. Lim *et al.* [1] differentiate between intrinsic and extrinsic appearance variability including pose variation, shape deformation of the object and illumination change, camera movement, occlusions, respectively.

In the past few years, particle filters have become the method of choice for tracking. Isard and Blake [2] introduced particle filtering (condensation algorithm). Many different sampling schemes have been suggested in the meantime. An overview about sampling schemes of particle filters and the relation to Kalman filters is provided in [3].

Recently, the main emphasis is on simultaneously tracking multiple objects and on online learning to adapt the reference models to the appearance changes, e.g., pose variation, illumination change. Lim *et al.* [1] introduce a single-object tracker, where the target representation—a low-dimensional eigenspace representation—is incrementally updated to model the appearance variability. They assume, like most tracking algorithms, that the target region is initialized by hand in the first frame. Jepson *et al.* [4] use a Gaussian mixture model which is adapted using an online expectation maximization (EM) algorithm to account for appearance changes. Their $\mathcal{WSL}$ tracker uses a wavelet-based object model which is useful for tracking objects where regions of the objects (i.e., faces) are stable while other regions vary, e.g., mouth. McKenna *et al.* [5] employ Gaussian mixtures of the color distributions of the objects as adaptive model. In [6], simple color histograms are used to represent the objects (similar as in [7]). However, they introduce a simple update of the histograms to overcome the appearance changes of the object. All the aforementioned articles are focused on tracking a singe object. For tracking multiple objects, most algorithms belong to one of the following three categories: 1) Multiple instances of a single-object tracker are used [8]. 2) All objects of interest are included in the state space [9]. A fixed number of objects is assumed. Varying number of objects result in a dynamic change of the dimension of the state space. 3) Most recently, the framework of particle filters is extended to capture multiple targets using a mixture model [10]. This mixture particle filter—where each component models an individual object—enables interaction between the components by the importance weights. In [11], this approach is extended by the Adaboost algorithm to learn the models of the targets. The information from Adaboost enables detection of objects entering the scene automatically. The mixture particle filter is further extended in [12] to handle mutual occlusions. They introduce a rectification technique to compensate for camera motions, a global nearest neighbor data association method to correctly identify object detections with existing tracks, and a mean-shift algorithm which accounts for more stable trajectories for reliable motion prediction.

In this paper, we do both tracking of multiple persons in a meeting scenario and online adaptation of the models to account for appearance changes during tracking. The tracking is based on low-level features such as skin color, object motion, and object size. Based on these features, automatic initialization and termination of objects are performed. The aim is to use as little prior knowledge as possible. For tracking, a particle filter is incorporated to propagate sample distributions over time. Our implementation is related to the *dual estimation* problem [13], where both the states of multiple objects and the parameters of the reference models are simultaneously estimated given the observations. At every time step, the particle filter estimates the states using the observation likelihood of the current reference models while the online learning of the reference models is based on the current state estimates. Additionally, we discuss

Manuscript received July 24, 2007; revised January 30, 2008. This work was supported by the Austrian Science Fund under Project P19737-N15.

The author is with the Department of Electrical Engineering, Laboratory of Signal Processing and Speech Communication, Graz University of Technology, 8010 Graz, Austria (e-mail: pernkopf@tugraz.at).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.927281

the similarity between our implemented tracker based on particle filters and genetic algorithms (GAs). We want to emphasize this close connection since approaches what have independently been developed in one community might turn out to be very useful for the other community and vice versa. Numerous experiments on meeting data demonstrate the capabilities of our tracking approach. Additionally, we empirically show that the adaptation of the reference model during tracking of a indoor and outdoor scenes results in a more robust tracking. For this, we report the average of the standard deviation of the trajectories over numerous independent tracking runs depending on the learning rate.

The proposed approach differs from previous methods in several aspects. Recently, much work has been done in multiple object tracking on the one hand side and on reference model adaptation for a single-object tracker on the other side. In this paper, we do both tracking of multiple objects and online learning to incrementally update the representation of the tracked objects to model appearance changes. We use the Jensen–Shannon (JS) divergence [14] to measure the similarity between the tracked object and its reference model. Additionally, we discuss its advantages compared to the Kullback–Leibler divergence [15] and the Bhattacharyya similarity coefficient [16]. We automatically initialize and terminate tracking of individual objects based on low-level features, i.e., face color, face size, and object movement. Many methods unlike our approach assume that the target region has been initialized in the first frame.

This paper is organized as follows. Section II introduces the particle filter for multiple object tracking, the state-space dynamics, the observation model, automatic initialization and termination of objects, and the online learning of the models for the tracked objects. Section II-G summarizes the implemented tracker on the basis of pseudocode. Section III sketches the relationship to GA. The tracking results on a meeting scenario and for indoor/outdoor scenes are presented in Section IV. Additionally, we provide empirical verification of the reference model learning in this section. Section V concludes this paper.

II. Tracking Using Particle Filters

In many applications the states of a dynamic system have to be estimated from a time series of noisy observations. The Kalman filter [13], [17] is a linear dynamical system [18] that provides a linear time-discrete filter that estimates the states online over time once observations become available. This filter is recursive in a sense that each current state estimate is computed from the previous estimate and the current observed data. In contrast to linear dynamical systems, the hidden Markov model [19] assumes a discrete state space. Recently, many extentions of the basic linear dynamical system have been proposed [13] to overcome the assumption of the linear-Gaussian model used for the observations and state transition, e.g., the extended Kalman filter, unscented Kalman filter, or the switching state-space model [20]. Another approach for filtering is to use sequential Monte Carlo methods which are also known as particle filters [21]. They are capable to deal with any nonlinearity or distribution.

A. *Particle Filter*

A particle filter is capable to deal with nonlinear non-Gaussian processes and has become popular for visual tracking. For tracking, the probability distribution that the object is in state $\mathbf{x}_t$ at time t given the observations $\mathbf{y}_{0:t}$ up to time t is of interest. Hence, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ has to be constructed starting from the initial distribution $p(\mathbf{x}_0|\mathbf{y}_0) = p(\mathbf{x}_0)$. In Bayesian filtering, this can be formulated as iterative recursive process consisting of the prediction step

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})dx_{t-1} \quad (1)$$

and of the filtering step

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})dx_t} \quad (2)$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the dynamic model describing the state-space evolution which corresponds to the evolution of the tracked object (see Section II-B) and $p(\mathbf{y}_t|\mathbf{x}_t)$ is the likelihood of an observation $\mathbf{y}_t$ given the state $\mathbf{x}_t$ (see observation model in Section II-C).

In particle filters $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ of the filtering step is approximated by a finite set of weighted samples, i.e., the particles, $\{\mathbf{x}_t^m, w_t^m\}_{m=1}^M$, where M is the number of samples. Particles are sampled from a proposal distribution $\mathbf{x}_t^m \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{0:t})$ (importance sampling) [3]. In each iteration, the importance weights are updated according to

$$w_t^m \propto \frac{p\left(\mathbf{y}_t|\mathbf{x}_t^m\right) p\left(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m\right)}{q\left(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m, \mathbf{y}_{0:t}\right)} w_{t-1}^m \quad \sum_{m=1}^M w_t^m = 1. \quad (3)$$

One simple choice for the proposal distribution is to take the prior density $q(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m, \mathbf{y}_{0:t}) = p(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m)$ (bootstrap filter). Hence, the weights are proportional to the likelihood model $p(\mathbf{y}_t|\mathbf{x}_t^m)$

$$w_t^m \propto p\left(\mathbf{y}_t|\mathbf{x}_t^m\right) w_{t-1}^m. \quad (4)$$

The posterior filtered density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be approximated as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^M w_t^m \delta\left(\mathbf{x}_t - \mathbf{x}_t^m\right) \quad (5)$$

where $\delta(\mathbf{x}_t - \mathbf{x}_t^m)$ is the Dirac delta function with mass at $\mathbf{x}_t^m$.

We use resampling to reduce the *degeneracy problem* [3], [21]. We resample the particles $\{\mathbf{x}_t^m\}_{m=1}^M$ with replacement M times according to their weights w_t^m. The resulting particles $\{\mathbf{x}_t^m\}_{m=1}^M$ have uniformly distributed weights $w_t^m = 1/M$. Similar to the sampling importance resampling filter [3], we resample in every time step. This simplifies (4) to $w_t^m \propto p(\mathbf{y}_t|\mathbf{x}_t^m)$ since $w_{t-1}^m = 1/M \quad \forall m$.

In the meeting scenario, we are interested in tracking the faces of multiple people. We treat the tracking of multiple objects completely independent, i.e., we assign a set of M particles to each tracked object k as $\{\{\mathbf{x}_t^{m,k}\}_{m=1}^M\}_{k=1}^K$, where K is the total number of tracked objects which dynamically

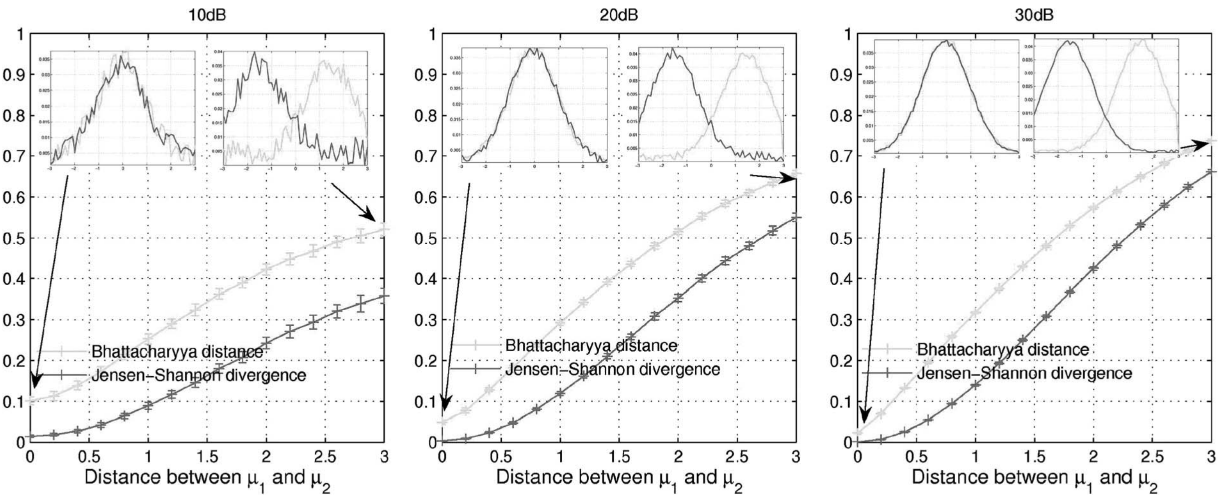


Fig. 1. JS divergence and Bhattacharyya similarity coefficient between two distributions estimated from samples. We added noise at a level of 10, 20, and 30 dB to the distributions.

changes over time. Hence, we use multiple instances of a single-object tracker similar to [8].

B. State-Space Dynamics

The state sequence evolution $\{\mathbf{x}_t; t \in \mathbb{N}\}$ is assumed to be a second-order autoregressive process which is used instead of the first-order formalism $(p(\mathbf{x}_t|\mathbf{x}_{t-1}))$ introduced in the previous section. The second-order dynamics can be written as first order by extending the state vector at time t with elements from the state vector at time $t-1$.

We define the state vector at time t as $\mathbf{x}_t = [x_t \;\; y_t \;\; s_t^x \;\; s_t^y]^{\mathrm{T}}$. The location of the target at t is given as x_t, y_t, respectively, and s_t^x, s_t^y denote the scale of the tracked region in the $x \times y$ image space. In our tracking approach, the transition model corresponds to

$$\mathbf{x}_{t+1}^{m,k} = \mathbf{x}_t^{m,k} + C\mathbf{v}_t + \frac{D}{2M}\sum_{m'=1}^{M}\left(\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1}^{m',k}\right) \qquad (6)$$

where $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I})$ is a simple Gaussian random noise model and the term $1/2M\sum_{m'=1}^{M}(\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1}^{m',k})$ captures the linear evolution of object k from the particles of the previous time step. Factor D models the influence of the linear evolution, e.g., D is set to 0.5. The parameters of the random noise model are set to $C = \mathrm{diag}([10 \quad 10 \quad 0.03 \quad 0.03])$ with the units of [pixel/frame], [pixel/frame], [1/frame], and [1/frame], respectively.

C. Observation Model

The shape of the tracked region is determined to be an ellipse [4] since the tracking is focused on the faces of the individuals. We assume that the principal axes of the ellipses are aligned with the coordinate axes of the image. Similarly to [7], we use the color histograms for modeling the target regions. Therefore, we transform the image into the hue-saturation-value (HSV) space [22]. For the sake of readability, we abuse the notation and write the particle $\mathbf{x}_t^{m,k}$ as $\mathbf{x}_t$ in this section. We build an individual histogram for hue (H) $h_{\mathrm{H}}^{\mathbf{x}_t}$, saturation (S) $h_{\mathrm{S}}^{\mathbf{x}_t}$, and value (V) $h_{\mathrm{V}}^{\mathbf{x}_t}$ of the elliptic candidate region at $\mathbf{x}_t$. The length of the principal axes of the ellipse are $A_{\mathrm{ref}}^k s_t^x$ and $B_{\mathrm{ref}}^k s_t^y$, respectively, where A_{ref}^k and B_{ref}^k are the length of the ellipse axes of the reference model of object k.

The likelihood of the observation model (likelihood model) $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$ must be large for candidate regions with a histogram close to the reference histogram. Therefore, we introduce the JS divergence [14] to measure the similarity between the normalized candidate and reference histograms, $h_{\mathrm{c}}^{\mathbf{x}_t}$ and $h_{\mathrm{c,ref}}^k$, $c \in \{H, S, V\}$, respectively. Since, the JS divergence is defined for probability distributions the histograms are normalized, i.e., $\sum_N h_{\mathrm{c}}^{\mathbf{x}_t} = 1$, where N denotes the number of histogram bins. In contrast to the Kullback–Leibler divergence [15], the JS divergence is symmetric and bounded between 0 and 1. The JS divergence between the normalized histograms is defined as

$$\mathrm{JS}_\pi\left(h_{\mathrm{c}}^{\mathbf{x}_t}, h_{\mathrm{c,ref}}^k\right) = H\left(\pi_1 h_{\mathrm{c}}^{\mathbf{x}_t} + \pi_2 h_{\mathrm{c,ref}}^k\right) - \pi_1 H\left(h_{\mathrm{c}}^{\mathbf{x}_t}\right) - \pi_2 H\left(h_{\mathrm{c,ref}}^k\right) \qquad (7)$$

where $\pi_1 + \pi_2 = 1, \pi_i \geq 0$ and the function $H(\cdot)$ is the entropy [15]. The JS divergence is computed for the histograms of the H, S, and V space, and the observation likelihood is

$$p\left(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k}\right) \propto \exp -\lambda\left[\sum_{c\in\{H,S,V\}} \mathrm{JS}_\pi\left(h_{\mathrm{c}}^{\mathbf{x}_t^{m,k}}, h_{\mathrm{c,ref}}^k\right)\right] \qquad (8)$$

where parameter λ is chosen to be five and the weight π_i is uniformly distributed. The number of bins of the histograms is set to $N = 50$. The JS divergence provides a lower and upper bound to the Bayes error and π_1 and π_2 can be viewed as *a priori* probabilities in a classification problem [14].

In contrast to the often used Bhattacharyya similarity coefficient $\sqrt{1 - \sum_N \sqrt{h_{\mathrm{c}}^{\mathbf{x}_t^{m,k}} h_{\mathrm{c,ref}}^k}}$ [16], the JS divergence is not so sensitive to local perturbations in the histogram (noise). This is shown in Fig. 1 where we compute the JS divergence and Bhattacharyya similarity coefficient on synthetic data. Therefore, we sample two Gaussian distributions with $\mu_1 = -\mu_2$, where μ_1 varies from 0 to 1.5, and unit variance. Noise is added to those distributions at a level of 10, 20, and 30 dB. Plots are averaged over 100 independent simulations.

D. Automatic Initialization of Objects

If an object enters the frame, a set of M particles and a reference histogram for this object have to be initialized. Basically, the initialization of objects is automatically performed using the following simple low-level features.

1) Motion: The images are transformed to gray scale $I^G_{x_t,y_t}$. The motion feature is determined for each pixel located at x, y by the standard deviation over a time window T_w as $\sigma^t_{x,y} = \sigma(I^G_{x_{t-T_w:t},y_{t-T_w:t}})$. Applying an adaptive threshold $T_{\text{motion}} = 1/10 \max_{x,y\in I^G} \sigma^t_{x,y}$ pixels with a value larger T_{motion} belong to regions where movement happens. However, $\max_{x,y\in I^G} \sigma^t_{x,y}$ has to be sufficiently large so that motion exists at all. A binary motion image $I^{B_{\text{motion}}}_{x_t,y_t}$ after morphological closing is shown in Fig. 2.

2) Skin Color: The skin color of the people is modeled by a Gaussian mixture model [23] in the HSV color space. A Gaussian mixture model $p(\mathbf{z}|\boldsymbol{\Theta})$ is the weighted sum of $L > 1$ Gaussian components, $p(\mathbf{z}|\boldsymbol{\Theta}) = \sum_{l=1}^{L} \alpha_l \mathcal{N}(\mathbf{z}|\mu_l, \boldsymbol{\Sigma}_l)$, where $\mathbf{z} = [z_{\text{H}}, z_{\text{S}}, z_{\text{V}}]^{\text{T}}$ is the 3-D color vector of one image pixel, α_l corresponds to the weight of each component $l = 1, \ldots, L$. These weights are constrained to be positive $\alpha_l \geq 0$ and $\sum_{l=1}^{L} \alpha_l = 1$. The Gaussian mixture is specified by the set of parameters $\boldsymbol{\Theta} = \{\alpha_l, \mu_l, \boldsymbol{\Sigma}_l\}_{l=1}^{L}$. These parameters are determined by the EM algorithm [24] from a face database.

   Image pixels $\mathbf{z} \in I^{\text{HSV}}_{x_t,y_t}$ are classified according to their likelihood $p(\mathbf{z}|\boldsymbol{\Theta})$ using a threshold T_{skin}. The binary map $I^{B_{\text{skin}}}_{x_t,y_t}$ filtered with a morphological closing operator is presented in Fig. 2.

3) Object Size: We initialize a new object only for skin-colored moving regions with a size larger than T_{Area}. Additionally, we do not allow initialization of a new set of particles in regions where currently an object is tracked. To this end, a binary map $I^{B_{\text{prohibited}}}_{x_t,y_t}$ represents the areas where initialization is prohibited. The binary combination of all images $I^B_{x_t,y_t} = I^{B_{\text{motion}}}_{x_t,y_t} \cap I^{B_{\text{skin}}}_{x_t,y_t} \cap \overline{I^{B_{\text{prohibited}}}_{x_t,y_t}}$ is used for extracting regions with an area larger T_{Area}. Target objects are initialized for those regions, i.e., the ellipse size $(A^k_{\text{ref}}, B^k_{\text{ref}})$ and the histograms $h^k_{\text{c,ref}}, c \in \{H, S, V\}$ are determined from the region of the bounding ellipse.

Fig. 2 shows an example for the initialization of a new object. The original image $I^{\text{HSV}}_{x_t,y_t}$ is presented in (a). A person entering from the right side should be initialized. A second person in the middle of the image is already tracked. The binary images of the thresholded motion $I^{B_{\text{motion}}}_{x_t,y_t}$ and the skin-colored areas $I^{B_{\text{skin}}}_{x_t,y_t}$ are shown in (b) and (c), respectively. The reflections at the table and the movement of the curtain produce noise in the motion image. The color of the table and chairs intersects with the skin-color model. To guarantee successful initialization the lower part of the image—the region of the chairs and desk—has to be excluded. This is reasonable since nobody can enter in this area. Also, tracking is performed in the area above the chairs only. Finally, the region of the new initialized object is presented as ellipse in (d). Resizing of the images is performed for computing the features to speed up the initialization of objects.

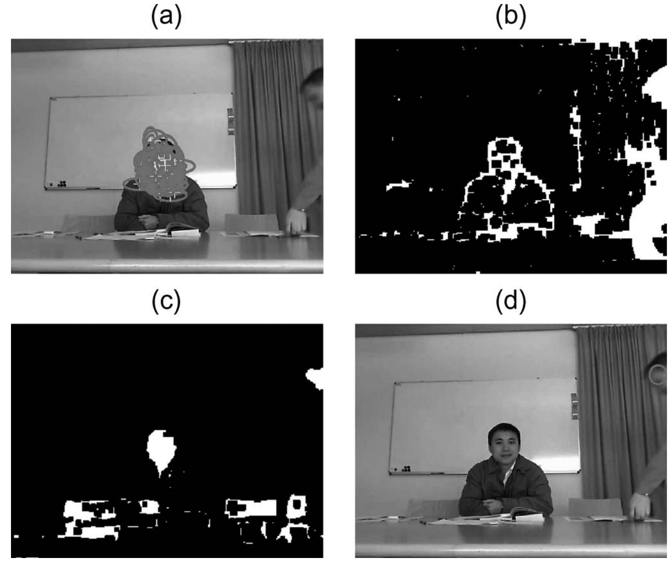


Fig. 2. Initialization of a new object. (a) Original image with one object already tracked. (b) Binary image of the thresholded motion $I^{B_{\text{motion}}}_{x_t,y_t}$. (c) Binary image of the skin-colored areas $I^{B_{\text{skin}}}_{x_t,y_t}$. (d) Image with region of initialized object.

1) Shortcomings: The objects are initialized when they enter the image. The reference histogram is taken during this initialization. There are the following shortcomings during initialization.

1) The camera is focused on the people sitting at the table and not on people walking behind the chairs. This means that walking persons appear blurred.
2) Entering persons are moving relatively fast. This also results in a degraded image quality (blurring).
3) During initialization, we normally get the side view of the person's head. When the person sits at the table the reference histogram is not necessarily a good model for the frontal view.

To deal with these shortcomings, we propose online learning to incrementally update the reference models of the tracked objects over time (see Section II-F). We perform this only in cases where no mutual occlusions between the tracked objects are existent.

E. Automatic Termination of Objects

Termination of particles is performed if the observation likelihood $p(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t)$ at state $\mathbf{x}^{m,k}_t$ drops below a predefined threshold T_{Kill} (e.g., 0.001), i.e.,

$$p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right) = \begin{cases} 0, & \text{if } p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right) < T_{\text{Kill}} \\ p\left(\mathbf{y}^{m,k}_t|\mathbf{x}^{m,k}_t\right), & \text{otherwise.} \end{cases} \tag{9}$$

Particles with zero probability do not survive during resampling. If the tracked object leaves the field of view all M

particles of an object k are removed, i.e., $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k}) = 0$ for all particles of object k.

F. Incremental Learning of Object Models

To handle the appearance change of the tracked objects over time, we use online learning to adapt the reference histograms $h_{\mathrm{c,ref}}^k, \quad c \in \{H, S, V\}$ (similar to [6]) and ellipse size A_{ref}^k and B_{ref}^k. Therefore, a learning rate α is introduced, and the model parameters for target object k are updated according to

$$h_{\mathrm{c,ref}}^k = \alpha \hat{h}_{\mathrm{c}}^k + (1-\alpha) h_{\mathrm{c,ref}}^k, \qquad c \in \{H, S, V\} \tag{10}$$

$$A_{\mathrm{ref}}^k = \alpha \hat{A}^k + (1-\alpha) A_{\mathrm{ref}}^k \tag{11}$$

$$B_{\mathrm{ref}}^k = \alpha \hat{B}^k + (1-\alpha) B_{\mathrm{ref}}^k \tag{12}$$

where $\hat{h}_{\mathrm{c}}^k$ denotes the histogram and $\hat{A}^k$ and $\hat{B}^k$ are the principal axes of the bounding ellipse of the nonoccluded (i.e., no mutual occlusion between tracked objects) skin-colored region of the corresponding tracked object k located at $\{\mathbf{x}_t^{m,k}\}_{m=1}^M$. Again, this region has to be larger than T_{Area}. No update of the reference models is performed in the case where occlusion between the tracked objects occurs or the skin-colored region is not large enough. The latter condition is a simple way to ensure that the model update is only conducted for faces. This simplistic assumption can be appropriately extended by integrating more advanced face models.

The learning rate α introduces an *exponential forgetting process*, i.e., the contribution of a specific object exponentially decreases as it recedes into the past. Currently, the learning rate (value between 0 and 1) is fixed (a good value has been selected during experiments). However, α could be adapted depending on the dynamics of the scene.

Algorithm 1 Particle Filter Tracking
Input: $I_{x_{0:T}, y_{0:T}}^{\mathrm{HSV}}$ (Color image sequence $0:T$),
  Skin-color model $\mathbf{\Theta}$
Parameters: $M, N, \lambda, C, D, T_w, T_{\mathrm{motion}}, T_{\mathrm{skin}}, T_{\mathrm{Area}}$,
  $T_{\mathrm{Kill}}, \alpha$
Output: $\{\{\mathbf{x}_{0:T}^{m,k}\}_{m=1}^M\}_{\forall k}$
$t \leftarrow 0$
$k \leftarrow 0$
while InitObjects **do**
  $k \leftarrow k+1$
  Obtain: $h_{\mathrm{c,ref}}^k : c \in \{H, S, V\}$, A_{ref}^k, B_{ref}^k, $\mathbf{x}_{\mathrm{ref}}^k$
  $\mathbf{x}_{t+1}^{m,k} \leftarrow \mathbf{x}_{\mathrm{ref}}^k + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$ (Generate particles)
end while
$K \leftarrow k$
for $t = 1$ **to** T **do**
  $w_t^{m,k} \propto p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$
    $\forall k = 1, \ldots, K \quad \forall m = 1, \ldots, M$
  while KillObjects **do**
    $k \leftarrow$ Determine object to terminate
    Remove M particles $x_t^{m,k}$ of object k
    Remove reference histogram and ellipse size:
      $h_{c,ref}^k : c \in \{H, S, V\}, A_{\mathrm{ref}}^k, B_{\mathrm{ref}}^k$
    $K \leftarrow K - 1$
  end while
  for $k = 1$ **to** K **do**
    $w_t^{m,k} \leftarrow w_t^{m,k} / \sum_{m'=1}^M w_t^{m',k} \quad \forall m = 1, \ldots, M$
    $\{\mathbf{x}_t^{m,k}\}_{m=1}^M \leftarrow$ Resampling
      (with replacement): $\{\mathbf{x}_t^{m,k}, w_t^{m,k}\}_{m=1}^M$
    $\mathbf{x}_{t+1}^{m,k} \leftarrow \mathbf{x}_t^{m,k} + C\mathbf{v}_t + (D/2M) \sum_{m'=1}^M (\mathbf{x}_t^{m',k} - \mathbf{x}_{t-1})^{m',k}$
      $\forall m = 1, \ldots, M$ (Apply state-space dynamics)
    if OnlineUpdate **then**
      Determine: $\hat{h}_{\mathrm{c}}^k : c \in \{H, S, V\}$, $\hat{A}^k$, $\hat{B}^k$
      $h_{\mathrm{c,ref}}^k \leftarrow \alpha \hat{h}_{\mathrm{c}}^k + (1-\alpha) h_{\mathrm{c,ref}}^k \quad c \in \{H, S, V\}$
      $A_{\mathrm{ref}}^k \leftarrow \alpha \hat{A}^k + (1-\alpha) A_{\mathrm{ref}}^k$
      $B_{\mathrm{ref}}^k \leftarrow \alpha \hat{B}^k + (1-\alpha) B_{\mathrm{ref}}^k$
    end if
  end for
  while InitObjects **do**
    $K \leftarrow K + 1$
    Obtain: $h_{\mathrm{c,ref}}^K : c \in \{H, S, V\}$, A_{ref}^K, B_{ref}^K, $\mathbf{x}_{\mathrm{ref}}^K$
    $\mathbf{x}_{t+1}^{m,K} \leftarrow \mathbf{x}_{\mathrm{ref}}^K + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$ (Generate particles)
  end while
end for

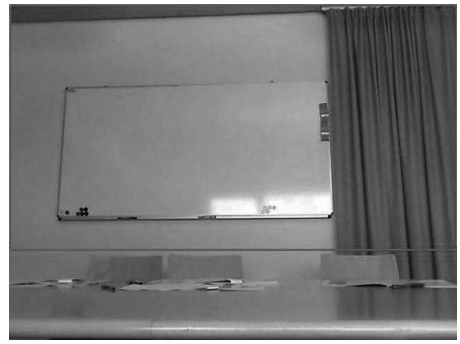


Fig. 3. Tracking scene. We track and initialize objects in the red rectangle.

G. Implemented Tracker

In the following, we sketch our tracking approach for multiple objects (see Algorithm 1). The binary variable *InitObject* denotes that a new object for tracking has been detected. *KillObject* is set if an object should be terminated. *OnlineUpdate* indicates that object k located at $\{\mathbf{x}_t^{m,k}\}_{m=1}^M$ is nonoccluded, and the area of the skin-colored region is larger than T_{Area}, i.e., we perform online learning for reference model k.

Our implementation is related to the *dual estimation* problem [13], where both the states of multiple objects $\mathbf{x}_t^{m,k}$ and the parameters of the reference models are simultaneously estimated given the observations. At every time step, the particle filter estimates the states using the observation likelihood of the current reference models, while the online learning of the reference models is based on the current state estimates.

III. Relationship to GAs

GAs are optimization algorithms founded upon the principles of natural evolution discovered by Darwin. In nature, individuals have to adapt to their environment in order to survive in

Fig. 4. Tracking of people. Frames: 1, 416, 430, 449, 463, 491, 583, 609, 622, 637, 774, 844, 967, 975, 1182, 1400 (the frame number is assigned from left to right and top to bottom).

a process of further development. An introduction of GAs can be found in [25] and [26]. GA are stochastic procedures which have been successfully applied in many optimization tasks. GA operate on a population of potential solutions applying the principle of *survival of the fittest individual* to produce better and better approximations to the solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and assembling them together using operators inspired from nature. This leads to the evolution of individuals that are better suited to their environment than the parent individuals they were created from. GA model the natural processes, such as selection, recombination, and mutation. Starting from an initial population $P(0)$, the sequence $P(0)$, $P(1), \ldots, P(t)$, $P(t+1)$ is called population sequence or evolution. The end of an artificial evolution process is reached once the termination condition is met, and the result of the optimization task is available.

In this section, we want to point to the close relationship between GA and our particle filter for tracking. This analogy has been mentioned in [27]. As suggested in Section II, we treat the tracking of multiple objects completely independent, i.e., we have a set of M particles for each object k. In the GA framework, we can relate this to k instantiations of GA, one for each tracked object. Hence, each particle $\mathbf{x}_t^m$ of object k represents one individual in the population $P(t)$ which is value encoded. The population size is M. A new genetic evolution process is started once a new object is initialized for tracking (InitObject). The evolution process of the GA is terminated either at the end of the video $(t = T)$ or when the set of individuals is not supported by the fitness value (KillObject). The observation likelihood $p(\mathbf{y}_t^{m,k}|\mathbf{x}_t^{m,k})$ denotes the fitness function to evaluate the individuals. However, the scope of GA for tracking is slightly different. GA are generally used to find a set of parameters for a given optimization task, i.e., the aim is to find the individual with the best fitness after the termination of the GA. Whereas, in the tracking case, the focus lies on the evolution of the individuals, i.e., the trajectory of the tracked object.

The selection operator directs the search toward promising regions in the search space. *Roulette Wheel Selection* [28] is a widely used selection method which is very similar to sampling with replacement as used in Section II. To each individual, a reproduction probability according to $w_t^m \leftarrow w_t^m / \sum_{m'=1}^{M} w_t^{m'}$ is assigned. A roulette wheel is constructed with a slot size corresponding to the individuals reproduction probability. Then, M uniformly distributed random numbers on the interval [0, 1] are drawn and distributed according to their value around the wheel. The slots where they are placed to compose the subsequent population $P(t)$. The state-space dynamics of the particle filter (see Section II-B) is modeled by the recombination and mutation operator.

The framework of the GA for tracking one object k is presented in Algorithm 1. The incremental learning of the reference model is omitted for the sake of brevity.

Fig. 5. Partial occlusions. Frames: 468, 616, 974, 4363 (the frame number is assigned from left to right and top to bottom).

Algorithm 2 GA Tracking
Input: $I^{\mathrm{HSV}}_{x_{t:T}, y_{t:T}}$ (Color image sequence $t:T$),
Parameters: M, N, λ, C, D, T_{Kill}
Output: $\{\mathbf{x}^m_{t:T}\}^M_{m=1}$ (set of particle sequences $t:T$)
Initialize population $P(t)$:
  $\mathbf{x}^m_t \leftarrow \mathbf{x}_{ref} + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$
while $\overline{\text{KillObject}} \cap t < T$ (Loop over image sequence) **do**
  Evaluate individuals:
    $w^m_t \leftarrow p(\mathbf{y}^m_t | \mathbf{x}^m_t) \quad \forall m = 1, \ldots, M$
  Selection $P(t)$:
    $\{\mathbf{x}^m_t\}^M_{m=1} \leftarrow$ (Sampling with replacement) $\{\mathbf{x}^m_t, w^m_t\}^M_{m=1}$
  Recombination $P(t+1)$:
    $\mathbf{x}^m_{t+1} \leftarrow \mathbf{x}^m_t + (D/2M) \sum^M_{m'=1} (\mathbf{x}^{m'}_t - \mathbf{x}^{m'}_{t-1})$
    $\forall m = 1, \ldots, M$
  Mutation $P(t+1)$: $\mathbf{x}^m_{t+1} \leftarrow \mathbf{x}^m_{t+1} + C\mathbf{v}_t \quad \forall m = 1, \ldots, M$
  $t \leftarrow t+1$
end while

IV. Experiments

We present tracking results on meeting data in Section IV-A where we do both tracking of multiple persons and online adaptation of the reference models during tracking. In Section IV-B, we empirically show that the adaptation of the reference model during tracking (single object) of an indoor and outdoor scene results in a more robust tracking. Finally, in Section IV-C, tracking results using reference model adaptation for multiple objects of an outdoor scene are presented. For the outdoor scenes, we report the average standard deviation of the trajectories of independent tracking runs depending on the learning rate α.

A. Meeting Scenario

The meeting room layout is shown in Fig. 3. The red rectangle [region of interest (ROI)] in the image marks the frame where tracking and initialization of objects are performed. People may enter and leave on both sides of the image. Currently, our tracker initializes a new target even if it enters from the bottom, e.g., a hand moving from the table into the ROI. The strong reflections at the table, chairs, and the white board cause noise in the motion image.

For testing the performance of our tracking approach, ten videos with ~7000 frames have been used. The resolution is 640 × 480 pixels. The meeting room is equipped with a table and three chairs. We have different persons in each video. The people are coming from both sides into the frame moving

(a)

(b)

(c)

Fig. 6. Outdoor tracking. Frames: 7, 11, 12, 13, 14, 20, 42, 63, 80, 107, 136, 146, 158, 165, 192 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.2$).

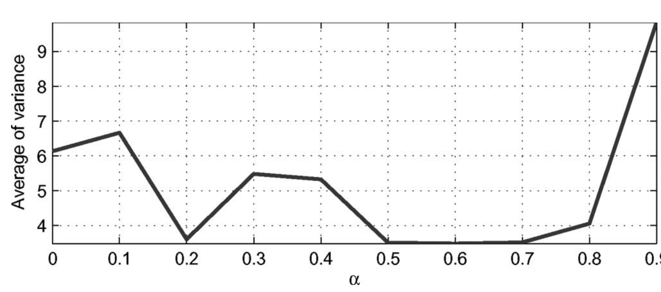


Fig. 7. Averaged standard deviation of the trajectories of 100 tracking runs depending on the reference model learning rate α.

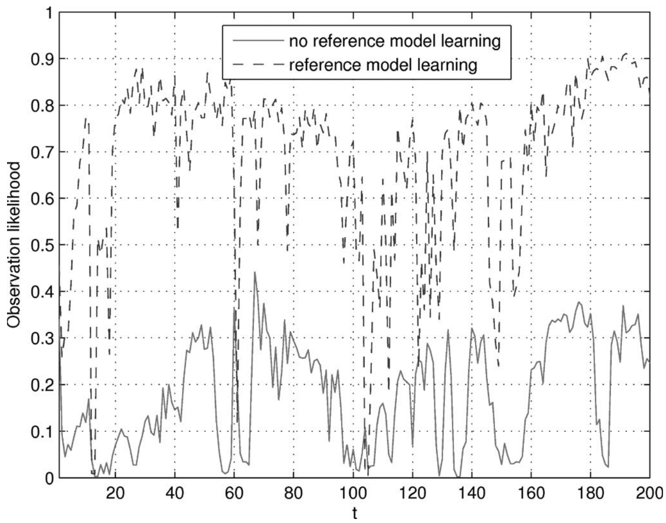


Fig. 8. Observation likelihood of outdoor sequence.

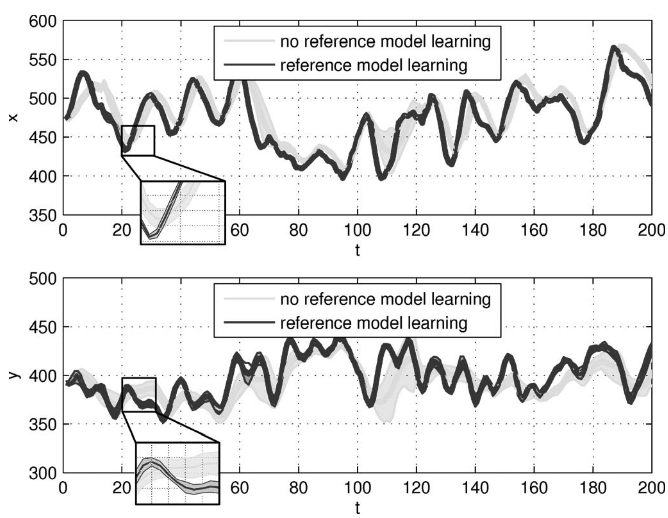


Fig. 9. Averaged trajectory with standard deviation in x and y of outdoor sequence (over ten runs).

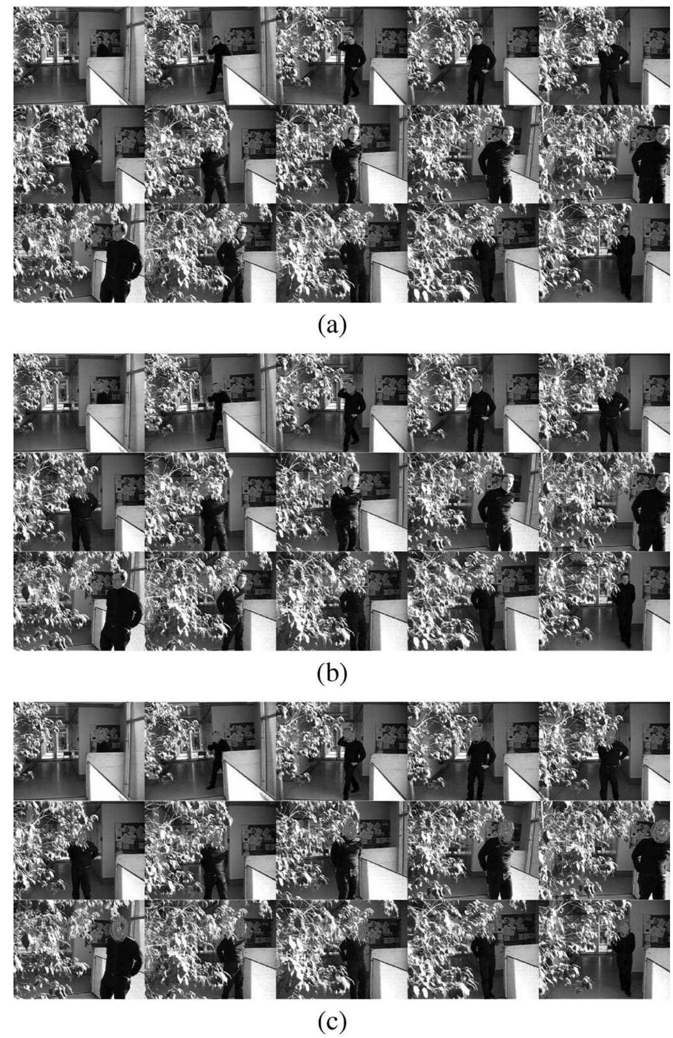


Fig. 10. Indoor tracking. Frames: 1, 12, 24, 31, 38, 41, 47, 54, 65, 71, 80, 107, 113, 120, 134 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.2$).

to chairs and sit down. After a short discussion, people are sequentially leaving the room, are coming back, sit down at different chairs and so on. At the beginning, people may already sit at the chairs. In this case, we have to automatically initialize multiple objects at the very first frame.

Fig. 4 shows the result of the implemented tracker for one video. All the initializations and terminations of objects are performed automatically. The appearance of an object changes over time. When entering the frame, we get the side view of the person's head. After sitting down at the table, we have a frontal view. We account for this by incrementally updating the reference histogram during tracking. We perform this only in the case where no mutual occlusions with other tracked objects are existent. The participants were successfully tracked over long image sequences.

First, the person on the left side stands up and leaves the room on the right side (frame 416–491). When walking behind the two sitting people, partial occlusions occur which do not cause problems. Next, the person on the right (frame 583–637) leaves the room on the left side. His face is again partially occluded by the person in the middle. Then, the person on the center chair leaves the room (frame 774). After that, a person on the right side enters and sits at the left chair (frame 844). At frame 967, a small person is entering and moving to the chair in the middle. Here, again, a partial occlusion occurs at frame 975, which is also tackled. Finally, a person enters from the right and sits down on the right chair (frame 1182, 1400). The partial occlusions are shown in Fig. 5. Also, the blurred face of the moving person in the back can be observed in this figure. The reference model adaptation enables a more robust tracking. If we do not update the models of the tracked objects over time, the tracking fails in case of these partial occlusions. In [29], occlusions are handled using multiple cameras for tracking participants in a meeting.

B. Reference Model Adaptation for Single-Object Tracking

In the following, we show the benefit of the reference model adaptation during tracking of a short indoor and outdoor sequence. In contrast to the meeting scenario, we restrict the

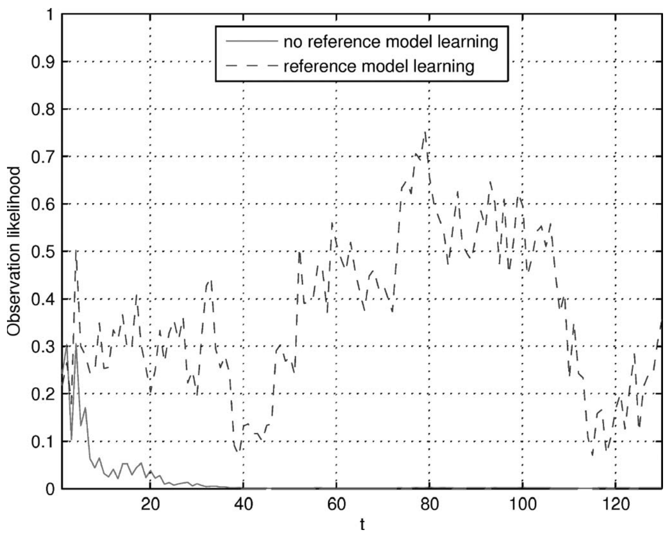


Fig. 11. Observation likelihood of indoor sequence.

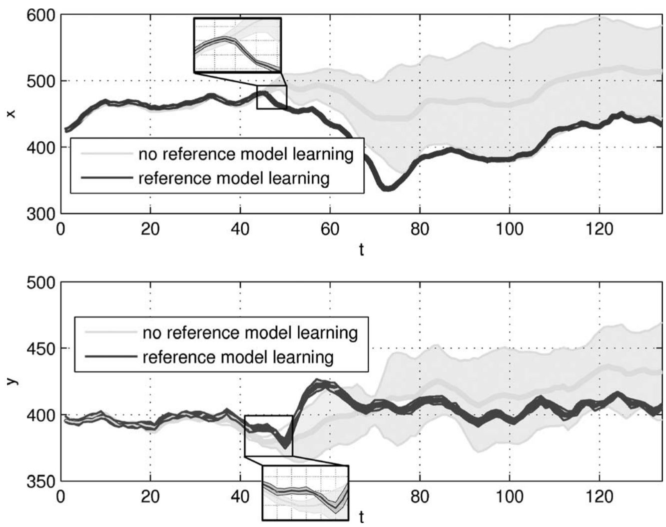
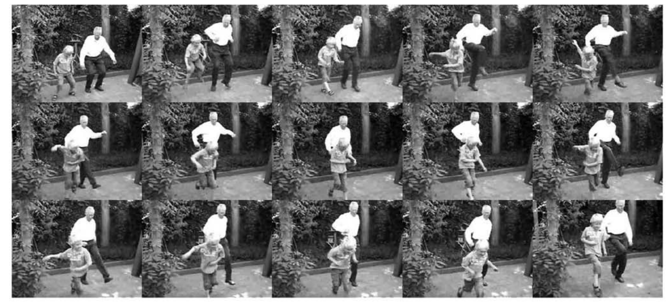
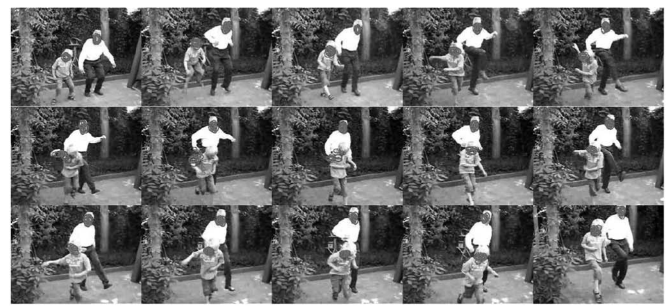


Fig. 12. Averaged trajectory with standard deviation in x and y of indoor sequence (over ten runs).



(a)



(b)

(c)

Fig. 13. Outdoor tracking of multiple objects. Frames: 1, 12, 30, 47, 49, 51, 53, 57, 59, 79, 105, 107, 109, 111, 149 (the frame number is assigned from left to right and top to bottom). (a) Original image sequence. (b) Tracking without reference model adaptation ($\alpha = 0$). (c) Tracking with online reference model learning ($\alpha = 0.1$).

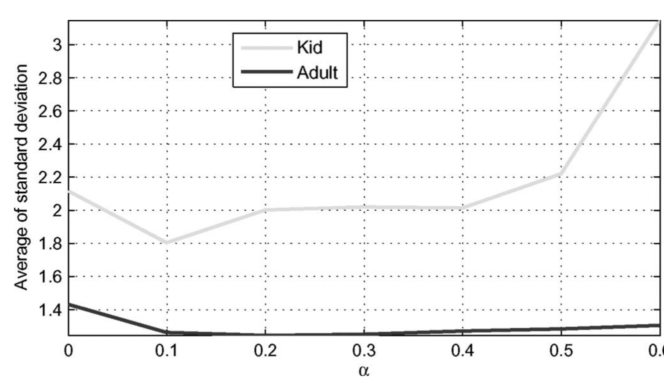


Fig. 14. Averaged standard deviation of the trajectories of ten tracking runs depending on the reference model learning rate α.

tracking to one single object, i.e., face. This means, in particular, that the automatic initialization and termination of the object is disabled. The object is initialized by hand in the first frame.

Fig. 6(a) shows a short outdoor sequence where a person is moving behind a tree and two cars with strongly changing lighting conditions. We have a total occlusion of the face in frames 12 and 13 and a partial occluded face in frames 146 to 165. We repeated the tracking without and with reference model learning ten times, and a typical result is shown in Fig. 6(b) and (c), respectively. We use $M = 50$ particles for tracking, whereas only 15 particles with the best observation likelihood are shown in the figures.

In Fig. 7, we present the average standard deviation of the trajectories over 100 tracking runs. The reference model learning rate α has been chosen in the range of $0, \ldots, 0.6$ (0 means that there is no learning). The optimal learning rate with respect to a low standard deviation of the trajectories over 100 independent runs is $\alpha = 0.2$ for this outdoor sequence.

Fig. 8 shows the observation likelihood of the best particle during tracking. At the complete occlusion (frames $t = 12$ and $t = 13$) and the partial occlusion (frames $t = 145, \ldots, 160$), the observation likelihood drops, however, with reference model learning a quick recovery is supported.

Fig. 9 summarizes the averaged trajectory with the standard deviation over ten different tracking runs performed for the outdoor scene. In the case of reference model learning, we can observe in the video sequences that the tracking of the face gives highly similar trajectories. The standard deviation is small and approximately constant over time. However, if no learning of the reference model is performed, the standard deviation is large in certain time segments. This leads to the conclusion that model adaptation results in a more robust tracking.

Fig. 10(a) shows an indoor video where a person is moving on a corridor, and a tree causes partial occlusion of the tracked face. Additionally, the lighting conditions are strongly varying. The face is partially occluded by the tree in frames 37–50 and 110–126. Again, the tracking without and with reference model learning is repeated ten times, and a typical result is shown in Fig. 10(b) and (c), respectively. Only 15 particles with the best observation likelihood are visualized. The parameter setting is the same as in the previous experiments. The tracker without reference model refinement fails during the first occlusion in all ten runs, whereas the tracker with online model update is successful in all cases. The optimal learning rate α is set to 0.2 (established during experiments).

This can be also observed in the observation likelihood of the best particle over time (see Fig. 11) and in the averaged trajectory over ten tracking results (see Fig. 12).

C. Reference Model Adaptation for Multiple Object Tracking

We show tracking results for an outdoor scene where a kid is showing an adult dancing steps (see Fig. 13). A typical tracking result without and with reference model learning is shown in Fig. 13(b) and (c), respectively. Again, $M = 50$ particles are used, whereas only 15 particles with the best observation likelihood are shown in the figures. Similar as in the previous section, we did a repeatability test, i.e., we tracked the objects over ten independent runs. The tracked objects are initialized by hand in the very first frame.

Fig. 14 shows the average standard deviation of the trajectories of ten tracking runs using a learning rate α in the range of $0, \ldots, 0.6$. The optimal learning rate for the *Kid* and the *Adult* is $\alpha = 0.1$ and $\alpha = 0.2$, respectively. Currently, α is fixed for the whole image sequence. Ideally, α could be adapted depending on the dynamics of the scene.

V. Conclusion

We propose a robust visual tracking algorithm for multiple objects (faces of people) in a meeting scenario based on low-level features as skin color, target motion, and target size. Based on these features, automatic initialization and termination of objects is performed. For tracking a sampling importance resampling, particle filter has been used to propagate sample distributions over time. Furthermore, we use online learning of the target models to handle the appearance variability of the objects. We discuss the similarity between our implemented tracker and GAs. Each particle represents an individual in the GA framework. The evaluation function incorporates the observation likelihood model and the individual selection process maps to the resampling procedure in the particle filter. The state-space dynamics is incorporated in the recombination and mutation operator of the GA. Numerous experiments on meeting data show the capabilities of the tracking approach. The participants were successfully tracked over long image sequences. Partial occlusions are handled by the algorithm. Additionally, we empirically show that the adaptation of the reference model during tracking of indoor and outdoor scenes results in a more robust tracking.

Future work concentrates on extending the tracker to other scenarios and to investigate an adaptive reference model learning rate α which depends on the dynamics of the scene. Furthermore, we aim to develop approaches for tackling occlusions.

Acknowledgment

This work was supported by the Austrian Science Fund (Project S106). The author would like to thank M. Grabner and M. Kepesi who collected the data during their involvement in the MISTRAL Project (www.mistral-project.at). The MISTRAL Project was funded by the Austrian Research Promotion Agency (www.ffg.at) within the strategic objective FIT-IT under Project 809264/9338. The author would also like to thank C. Kirchstätter for recording the indoor and outdoor video.

References

[1] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang, “Incremental learning for visual tracking,” in *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2005.

[2] M. Isard and A. Blake, “Condensation—Conditional density propagation for visual tracking,” *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, “Robust online appearance models for visual tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[5] S. McKenna, Y. Raja, and S. Gong, “Tracking colour objects using adaptive mixutre models,” *Image Vis. Comput.*, vol. 17, no. 3/4, pp. 225–231, 1999.

[6] K. Nummiaro, E. Koller-Meier, and L. Van Gool, “An adaptive color-based particle filter,” *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[7] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Proc. ECCV*, 2002, pp. 661–675.

[8] S. L. Dockstader and A. Tekalp, “Tracking multiple objects in the presence of articulated and occluded motion,” in *Proc. Workshop Human Motion*, 2000, pp. 88–98.

[9] C. Hue, J.-P. Le Cadre, and P. Pérez, “Tracking multiple objects with particle filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 791–812, Jul. 2002.

[10] J. Vermaak, A. Doucet, and P. Pérez, “Maintaining multi-modality through mixture tracking,” in *Proc. ICCV*, 2003, pp. 1110–1116.

[11] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, “A boosted particle filter: Multitarget detection and tracking,” in *Proc. ECCV*, 2004, pp. 28–39.

[12] Y. Cai, N. de Freitas, and J. J. Little, “Robust visual tracking for multiple targets,” in *Proc. ECCV*, 2006, pp. 107–118.

[13] S. Haykin, *Kalman Filtering and Neural Networks*. Hoboken, NJ: Wiley, 2001.

[14] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[15] T. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.

[16] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 142–149.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 82, pp. 34–45, 1960.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer Sci.+Bus. Media, 2006.

[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[20] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Comput.*, vol. 12, no. 4, pp. 963–996, 1998.

[21] A. Doucet, "On sequential Monte Carlo sampling methods for Bayesian filtering," Dept. Eng., Cambridge Univ., London, U.K., Tech. Rep. CUED/F-INFENG/TR. 310, 1998.

[22] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. London, U.K.: Int. Thomson, 1999.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ: Wiley, 2000.

[24] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. R. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 39, pp. 1–38, 1977.

[25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[26] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. London, U.K.: Oxford Univ. Press, 1996.

[27] T. Higuchi, "Monte Carlo filter using the genetic algorithm operators," *J. Stat. Comput. Simul.*, vol. 59, no. 1, pp. 1–23, Aug. 1997.

[28] L. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. Int. Conf. Genetic Algorithms Appl.*, 1987, pp. 14–21.

[29] H. Nait Charif and S. J. McKenna, "Tracking the activity of participants in a meeting," *Mach. Vis. Appl.*, vol. 17, no. 2, pp. 83–93, 2006.

Franz Pernkopf received the M.Sc. (Dipl.Ing.) degree in electrical engineering from the Graz University of Technology, Graz, Austria, in 1999, and the Ph.D. degree from the University of Leoben, Leoben, Austria, in 2002.

He was a Research Associate with the Department of Electrical Engineering, University of Washington, Seattle, from 2004 to 2006. Currently, he is an Assistant Professor with the Signal Processing and Speech Communication Laboratory, Graz University of Technology. His research interests include machine learning, Bayesian networks, feature selection, finite mixture models, vision, speech, and statistical pattern recognition.

Dr. Pernkopf was awarded the Erwin Schrödinger Fellowship in 2002.