Adaptive Object Tracking in Dynamic Environments with User Interaction

Gabriele Peters and Martin Kluger

***Abstract*—In this article an object tracking system is introduced which is capable to handle difficult situations in a dynamically changing environment. We evaluate the concepts of the proposed system by means of the task of person tracking in crowded seminar rooms. The contributions are threefold. We propose an expansion of the condensation algorithm which results in a more stable tracking in difficult situations such as sudden camera movements. Secondly, a new way of particle diffusion is introduced which allows for the adaption of the tracking module to movements of the camera as, e.g., induced by the user. These two contributions apply not only to person tracking but to object tracking in general. The third contribution, which applies to person tracking only, consists in a flexible way how to represent a moving person. The proposed tracking module meets demands such as real-time tracking of a target person who is allowed to move freely within a group of other persons and thus can be occluded. An interactive selection of new target persons is possible and also the challenge of a dynamically changing background could be coped with.**

***Index Terms*—person tracking, video surveillance, automatic video production, intelligent rooms**

I. Introduction

Tracking objects in a dynamically changing environment belongs to one of the most difficult problems in computer vision. Solutions of this problem are crucial not only for person tracking, which is the application on which we evaluate our proposed tracking method in this article, but also, for example, for dynamic object acquisition where sequences of objects viewed from different view points are analysed. The appearance of objects can change dramatically from frame to frame in a video sequence due to changes in the environment. Intelligent systems should be able to react dynamically to variations in the object's appearance. These variations can have many causes. If, for example, a camera moves around an object the lighting conditions can change or parts of the object can appear or disappear. In addition, the background can change which increases the degree of difficulty to assign corresponding landmarks from frame to frame, i.e., track the object. On the other hand, not only the camera can move but the object can move by itself or, even worse, can change its shape, which holds true for walking persons. The most difficult task is on hand, when also the background varies dynamically. Thus, robust object tracking in dynamic environments should be by itself adaptive to changes in the environment.

In this article we introduce a tracking system which reacts dynamically to recognized changes in the environment (i.e., changes in the camera parameters and interaction by the user). As a result the acquired data depend on the environmental dynamics. These concepts of active, visual object acquisition are applied to moving persons who interact in seminar rooms. The demands of such a person tracking system are manifold and some of them are listed in the following. The person tracking system should be capable of:

1) tracking a selected target person in real-time,
2) tracking a target person within other interacting persons and in front of a dynamic background
3) adapting to changing camera parameters such as orientation and focal length, where changes are induced either by an active camera or are initiated by a user,
4) bridging occlusions of the target object,
5) allowing for an interactive selection of a new target person during runtime.

The tracking system we present in this article is part of a larger project called *Virtual Camera Assistant*. The Virtual Camera Assistant is a prototype of a semi-automatic system, which supports documentary video recordings of indoor events such as seminars. It combines an automatic person tracking system with a controllable pan-tilt-zoom camera. Depending on the output of the tracking module the camera parameters (i.e., orientation and focal length) are determined automatically in such a way that the observer gets a natural impression of the recorded video (e.g., without jerky movements) and the whole movie appears pleasant to the eyes. In addition, (that is the "semi" in "semi-automatic") the user of the Virtual Camera Assistant is able, on the one hand, to control the camera parameters manually as well and, on the other hand, to select the target person to be tracked interactively at any time during recording.

A distinctive feature in comparison to other systems for automatic video production is the possible scenario of person tracking in highly cluttered scenes and within groups of interacting persons.

In this article we will omit a detailed description of the camera control. It is covered in [1] where the whole system is described. Rather, we will concentrate here on the introduction of the tracking module, i.e., on the realization of an automatic tracking of a target person in a crowded room with an active camera. In section II we present the components of the tracking module, in section III we describe our experimental

Manuscript received May 13, 2009; revised August 02, 2009.

This research was funded by the German Research Association (DFG) under Grant PE 887/3-3.

Gabriele Peters is with the University of Applied Sciences and Arts, Computer Science, Visual Computing, Emil-Figge-Strae 42, D-44227 Dortmund, Germany, e-mail: gabriele.peters@fh-dortmund.de (see http://www.inf.fh-dortmund.de/personen/professoren/peters).

Martin Kluger was with the Technical University Dortmund, Computer Science, Computer Graphics, Otto-Hahn-Strae 16, D-44227 Dortmund, Germany.

data, and in section IV the results, including weaknesses and limitations, are summarized, after which we close in section V with the conclusions. But still let us familiarize the reader with systems related to our approach.

A. Related Systems

Direct comparisons of the proposed system are possible with approaches from automatic video production, approaches from intelligent rooms with active cameras, as well as approaches from person tracking in general, e.g., for surveillance.

1) Automatic Video Production: A completely automatic system for the recording of presentation events is proposed in [2]. It is able to track one or more persons on a stage based on two cameras. The first one is static and monitors the whole stage to detect movements. The second camera is flexible and its parameters are controlled by the first one, thus, it is automatically oriented towards events. A similar automatic video production system is presented in [3]. Its tracking component utilizes two cameras as well. Movement on a stage is detected via the difference of two successive frames. In addition, the valid region for movements of the speaker is strongly restricted to a small area at the podium. Summarizing, these systems concentrate on a composition of several video sources and restrict the scenario of person tracking much stronger than the system proposed in this article where also interactions between the target person and other persons are allowed.

2) Intelligent Rooms: Our system can also be compared to the person or head tracking modules of *intelligent* or *smart rooms* (see Fig. 1). However, smart rooms usually are equipped with a number of different cameras, such as static ones, 360 degree omnidirectional cameras [4], or special stereo cameras [5]. With static and omnidirectional cameras it is possible to remove static background of a scene with standard methods [6]. Then further analysis for person tracking can be confined to the foreground. The object position obtained in this way can be used to adjust the synchronized, active cameras with the purpose of recording frames of higher resolution from the best view.

In scenes with dynamic background and without additional information on the parameters of the active camera (as it applies to our system) models can be generated only with explicitly higher effort. In [7] the requirements are reduced by a restriction to pan and tilt movements of the camera only. In addition, during camera movements the segmentation results are improved by a template matching with the foreground recognized in the last step.

A different possibility is a static camera array which simulates a single, virtual, active camera on the overlapping fields of view of all cameras [8]. One advantage is a large static image space with a higher resolution compared with a wide angle camera, which can be segmented with standard background models.

Alternatively, one can dispense with a background model completely when active cameras are used. Instead, potential candidates for the next object position can be predicted model-based and verified afterwards. This is realized, e.g., in [9] and the references cited in subsection I-A3.

3) Person Tracking: As examples for the field of video surveillance [10], [11], [12] and many approaches to the problem of person tracking in general [13], [14], [15] we refer here to the works of Isard [16], Nummiaro et al. [17], and Perez et al. [18] only, as the person tracking of the proposed system is based on their ideas (see also Fig. 2).

By means of several examples of object tracking, mostly based on edge filters, Isard introduces the condensation algorithm. It is a simple but effective particle filter algorithm which works stable also under temporary occlusions and disturbances. Our system is based on this algorithm. Both the other approaches use particle filters as well, but utilize color-based histograms as object features. Whereas in [17] a color histogram is adapted all along the video sequence to compensate for illumination changes, two static color histograms are used in [18] to improve the object hypothesis. Both concepts are applied in our approach as well.

II. Components of the Object Tracking System

In this section we introduce the methods and functioning of the proposed tracking system. In subsection II-A we first give an overview of the object tracking as part of the Virtual Camera Assistant. In addition, we recall the condensation algorithm our system is based on. In subsection II-B we describe our improvement of the condensation algorithm which consists mainly in the reiteration of parts of the original algorithm. Subsections II-C to II-G introduce concepts necessary to understand the object tracking module, such as the definition of the object state, the dynamic model, two different object profiles, the used distance functions, and the adaption of the reference profile of the target object (i.e., the object representation learned so far) to the current measurements. In the last subsection we describe our approach, how the particle diffusion in the condensation algorithm can be dynamically adapted to events in the environment, clarified by means of the example of camera movements.

A. Overview

The Virtual Camera Assistant mentioned in the introduction consists of the modules *Object Tracking* and *Camera Control* (see Fig. 3). The first module extracts the position and size of the currently selected target object from the video stream in real-time. The information obtained by this process is used by the second module to realize the desired camera adjustment. Short-term occlusions of the target object and other disturbances of the footage are mostly recognized and incorporated by the algorithms of both components and bypassed if possible.

Here we will concentrate on the *Object Tracking* module. For the segmentation between foreground (which is the target object) and background one can follow a bottom-up or a top-down approach. In the bottom-up approach regions are constructed, starting from the image pixels, and assigned to foreground or background. In contrast to this, the top-down approach utilizes a priori knowledge on the object, e.g., in the form of an object model, generates hypotheses and verifies them in the image. This is the way we proceed. Fig. 4

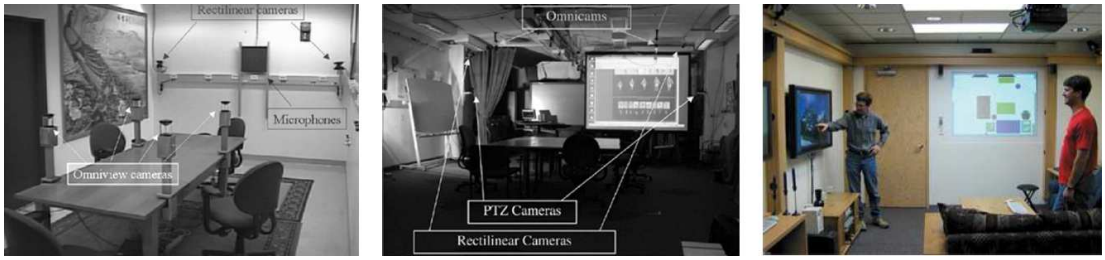

Fig. 1. Examples for smart rooms. Left and middle image taken from [4], right image taken from [5].

Fig. 2. Examples for person tracking with particle filter and different features. Left image taken from [16], middle image taken from [17], right image taken from [18].

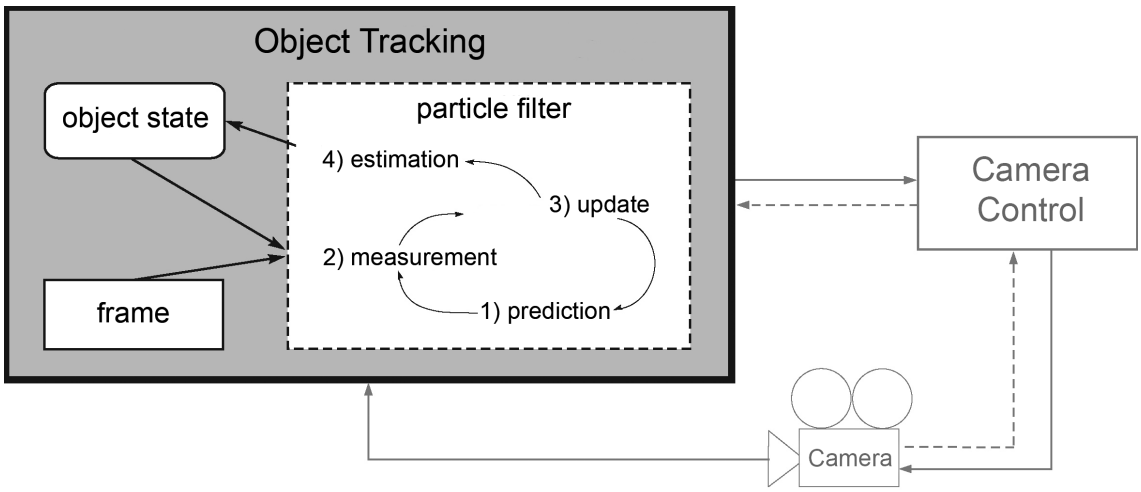

Fig. 4. Object Tracking Module. Object tracking is realized via a particle filter and operates in the four steps depicted here.

gives an idea of the information processing by the tracking module. Object tracking is realized via a particle filter which, roughly speaking, propagates multiple weighted hypotheses (called *particles*) and operates in four steps. First, for each particle the *object state* (see subsection II-C) - here it is the object's position and size in the current frame - is predicted by means of a *dynamic model* (see subsection II-D). Secondly, for each particle values of defined features are measured in the current frame. In the *object profile* (see subsection II-E) we define which features are observed. Thirdly, the hypotheses are compared with a reference profile via a similarity function (see subsections II-E and II-F). The results of the evaluation are used for an update of the particle weights. Fourthly, from all of those newly weighted hypotheses the pose and size estimation of the tracked object is calculated. After this, the reference profile is adapted via a new measurement in the current frame at the estimated position (not shown in this figure) (see subsection II-G). The particle filter is realized by an expansion of the condensation algorithm (see subsection II-B).

In a nutshell, after a manual selection of the target object by the user of the system, a number of features is initialized on the basis of which the hypotheses can be verified. From the current state of the object (e.g., her position and size) at time $t-1$ (i.e., in the current frame) a constant number of hypotheses for the state at time t (i.e., in the next frame) is calculated in a dynamic model. The validity of each hypothesis (i.e., each particle) is assessed in the following way. The features are measured for each particle in the frame at time t. Then they are compared with a reference profile (i.e., the description of the object learned up to this time) resulting in probability values,

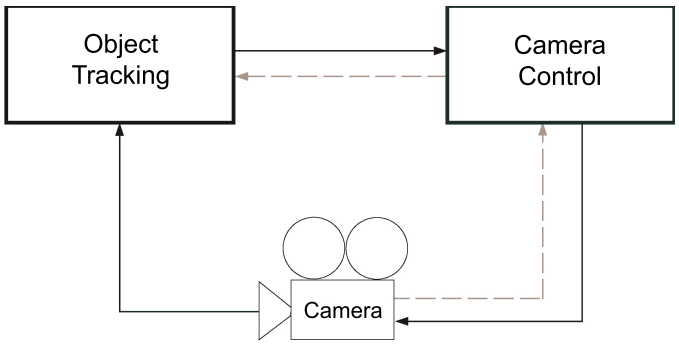

Fig. 3. Virtual Camera Assistant. The *Object Tracking* module recognizes a target object in a video frame, the *Camera Control* module adjusts the camera parameters in such a way that the object is shown in a favoured part of the frame.

which describe the validity of the hypotheses. (Particle weights are initialized by a uniform distriution.) Thus, the set of all hypotheses can be regarded as a discret approximation of the probability distribution for the current state of the target object. The estimated expectation value of this probability distribution, i.e., the weighted mean of all hypotheses, provides the new state of the target object at time t. Finally, the reference profile of the object is adapted by a second measurement at the estimated position of the object in frame t.

1) Condensation Algorithm: More formally spoken, the aim is the approximation of the *filtering distribution* $p(x_t|\mathcal{Z}_t)$ of the internal state x_t at time t on the basis of all previous measurements $\mathcal{Z}_t$. The hypotheses are represented by a constant number N of points $x_t^{(i)}, i = 1, \ldots, N$, at time t to which probabilities in form of normalized weights $\widetilde{w}_t^{(i)}$ are assigned. The points are called *particles*. The discrete approximation of $p(x_t|\mathcal{Z}_t)$ at time t is given by

$$\widehat{P}(x_t|\mathcal{Z}_t) \approx \sum_{i=1}^{N} \widetilde{w}_t^{(i)} \delta(x_t - x_t^{(i)})$$

with $\delta(.)$ as Dirac impulse. At the transition from $t-1$ to t the particles (and thus the hypotheses) are distributed anew via propagation through the dynamic model. Afterwards they are reassessed by a new measurement. That means, the new particles and weights $(x_t^{(i)}, \widetilde{w}_t^{(i)})$ are calculated from the old particles and weights $(x_{t-1}^{(i)}, \widetilde{w}_{t-1}^{(i)})$. This calculation is described by the condensation algorithm, as proposed by Isard [16], which consist of three steps:

1) *Resampling:* A new sample $\{\widetilde{x}_{t-1}^{(i)}, i = 1, \ldots, N\}$ of size N is drawn with replacement from the set of particles $\{x_{t-1}^{(i)}, i = 1, \ldots, N\}$ at time $t-1$. The weight of a particle defines its probability to be drawn.
2) *Prediction:* Based on this new sample a prediction is carried out via the dynamic model (described in subsection II-D). Thus, the particles become representatives of a sample $\{x_t^{(i)}, i = 1, \ldots, N\}$.
3) *Measurement:* The new particle weights $\widetilde{w}_t^{(i)}$ are calculated as follows. The consistency of the N hypotheses (represented by the N particles) with the current frame is evaluated by means of the similarity function given in formula 1. Finally, these weightings $w_t^{(i)}$ are normalized (resulting in $\widetilde{w}_t^{(i)}$) and assigned to their associated particles.

The weighting $w_t^{(i)}$ for one particle $x_t^{(i)}$ is calculated as follows:

$$w_t^{(i)} = \frac{1}{\sqrt{2\pi}\sigma} \cdot exp\left(-\frac{1}{2\sigma^2} \cdot \left[D_{\widetilde{\mathcal{R}}}\left(x_t^{(i)}\right)\right]^2\right), i = 1, \ldots, N. \quad (1)$$

It can be regarded as *confidence value* for the corresponding measurement. How the distance value $D_{\widetilde{\mathcal{R}}}\left(x_t^{(i)}\right)$ regarding the reference profile $\widetilde{\mathcal{R}}$ is calculated is defined in subsection II-E. The Gaussian shape of formula 1 secures that with a growing match of the hypothesis with the measured reference data the assigned weight increases as well. Throughout all experiments we used a value of 0.13 for σ.

From the new particles an estimation of the current position and size of the object is possible, for example, simply by averaging. We estimate the new object state $\hat{x}_t$ by a weighted sum of the particles:

$$\hat{x}_t = \sum_{i=1}^{N} \tilde{w}_t^{(i)} * x_t^{(i)} \quad (2)$$

The first step (resampling) has the purpose to prevent the particle weights from becoming degenerated [19]. By drawing particles with replacement light weighted particles tend to be excluded from further calculations.

B. Expansion of the Condensation Algorithm

The parameters of the dynamic model (described in subsection II-D) have been determined on the basis of a "regular", not overly fast movement of the target person in the context of a seminar scene. In practice, however, it is possible, that the object movement - offset with the camera movement - is not always covered by the model. In these particular cases too few of the propagated hypotheses still cover the "true" position of the object, normally resulting in a tracking loss. An example for this is shown in the first row of Fig. 5.

For this reason we improve the condensation algorithm by two measures.

- We use a first threshold α_1 for the confidence value of the currently estimated object state $\hat{x}_t$. If this confidence value falls below α_1, a second iteration of steps (1) to (3) of the condensation algorithm is carried out on the same frame with $N_2 \geq N$ particles.
- In case the newly estimated object state in turn provides a confidence value, which lies below a second threshold $\alpha_2 < \alpha_1$, we increase the number of particles to $N_{max} \gg N$ in the resampling step from the next frame on. As soon as the confidence value exceeds threshold α_1 in the further processing, the number of particles is reduced to N again.

The first measure results in a noticeable improvement of the tracking. The second resampling is based on the weights of the first run, and thus the strongest weighted particles (though not strongly weighted) can direct the second run into the right direction, while the object remains in place. This approach

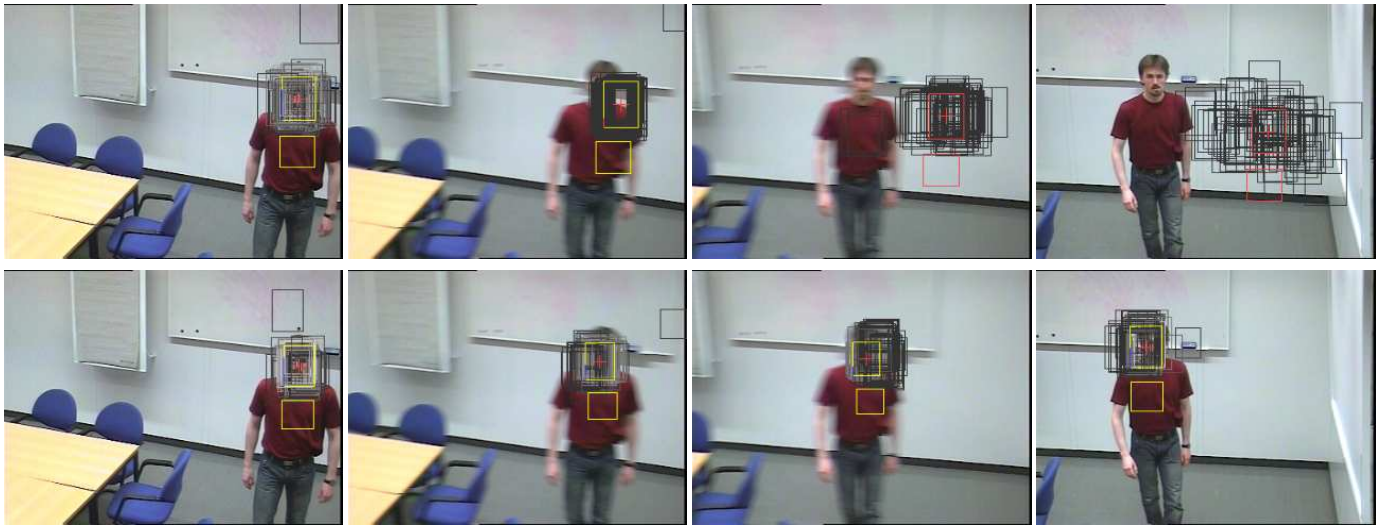

Fig. 5. Condensation algorithm with reiteration. For a description see subsection II-B.

resembles the annealed particle filter proposed by [20], who apply several iterations for one point in time with different weighting functions which mature over the course of iterations. This filter proved itself on person tracking under controlled conditions such as a neutral background.

To evaluate the proposed expansion we used N_2 between 10% and 20% and N_{max} between 50% and 75% larger than N. For α_1 and α_2 the values 0.4 and 0.2 proved to be feasible, respectively.

The first row of Fig. 5 shows frames 15, 20, 23, and 28 of a sequence as an example for a tracking loss due to a sudden camera pan by the user. It results from an application of the condensation algorithm with $N = 75$ particles as described by Isard [16]. The gray rectangles mark the particle distribution, the yellow (or red, depending on the success of the tracking) ones tag the mean of them and thus the estimated position and size of the tracked author. (The reason why there are always two yellow, or red respectively, rectangles becomes clear after reading subsection II-E.) The second row shows the same frames, this time tagged with the results of an application of our expansion of the condensation algorithm with reiteration. The second iteration of steps (1) to (3) of the condensation algorithm with $N = 50$, $N_2 = 75$, and $\alpha_1 = 0.4$ makes for a successful tracking.

C. Object State

As already mentioned above the current state of the object is described by features of the object valid for the current frame, such as its position and size in the frame under consideration. The object state is also the internal state $x_t \in \mathbb{R}^n$ of the system for a discrete point in time $t \in \mathbb{N}_0$ which is propagated in the dynamic model (see subsection II-D). (We omit the time index t in this subsection for simplification.) As we will see in subsection II-E we will compare different object profiles, which differ in the number and choice of the features constituting them. For the sake of a uniform formalism we include all possible features in the definition of the object state and later evaluate only those subsets of them which suit the object profile considered. The complete *object state* is thus defined by

$$x := (p_x, p_y, \dot{p}_x, \dot{p}_y, a, r, \rho)^T \tag{3}$$

with

- $(p_x, p_y)^T$ the position of the target object (i.e., the center of the rectangle selected by the user to mark the person)
- $(\dot{p}_x, \dot{p}_y)^T$ the horizontal and vertical velocity of the target object, respectively
- a width of the target object (in our application the head or face of a person is selected by the user)
- r ratio between width and height of the target object (i.e., $r = height/width$)
- ρ flexibility between two coupled regions of interest (for object profil $\mathcal{P}_2$, see subsection II-E)

With the exception of the last two parameters, units are given as pixels or pixels per time interval for velocities, respectively. The relation between width and height of the rectangle is restricted by a factor between 1.2 and 1.8 to avoid extreme shapes of the rectangle: $r \in [1.2; 1.8]$.

D. Dynamic Model

The dynamics of the system are defined by constant velocities only. The size of the target object is neglected in the dynamic model, because size variations caused by movements of a person in our application are quite small and thus are compensated for by the stochastic term of the model. Similar in design to the approach described in [17] we chose a simple model of first order:

$$x_t = \mathbf{A}x_{t-1} + \phi_t \tag{4}$$

Matrix $\mathbf{A}$ represents the deterministic part of the model, the independent random variable ϕ_t the stochastic part. For the

object state x_t, which was specified in the last subsection, $\mathbf{A}$ is defined by

$$\mathbf{A} := \begin{pmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{E_3} \end{pmatrix}, \qquad \mathbf{V} := \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

with E_3 the identity of size 3×3.

The elements of the stochastic term ϕ_t are independent and normally distributed with expected values zero, but with different standard deviations. Thus, the extent of the variability of even the static elements of the object state (such as the width a) can be controlled. This enables the user to manipulate the adaptivity of the elements indirectly, without the need of additional variables for periodic changes. The following values have proven to be useful:

$$\Sigma_\phi := \left(\sigma_{p_x}, \sigma_{p_y}, \sigma_{\dot{p}_x}, \sigma_{\dot{p}_y}, \sigma_a, \sigma_r, \sigma_\rho\right)^T = \left(3, 2, 2, 1, 1.5, 0.2, 9 \cdot \frac{2\pi}{360}\right)^T \tag{6}$$

1) Start Parameter: The start parameters for the object state depend on the position $(p_x^{t'}, p_y^{t'})$ and the selected object size $(a^{t'}, r^{t'})$ at time t' of the selection of a target object. As the target object is chosen by manual selection and interactively during the tracking of another object, one has to take into account that the target may have moved on already. For this reason the initial distribution of the particles $\{x_0^{(i)}, i = 1 \dots N\}$ is modeled as normally distributed random variable with a larger standard deviation Σ_0 and with the manually selected values for position and size as expected values: $x_0^{(i)} \in \mathcal{N}(\Theta_0, \Sigma_0)$,

$$\Theta_0 = \left(\mu_{p_x}, \mu_{p_y}, \mu_{\dot{p}_x}, \mu_{\dot{p}_y}, \mu_a, \mu_r, \mu_\rho\right)^T = \left(p_x^{t'}, p_y^{t'}, 0, 0, a^{t'}, r^{t'}, 0\right)^T \tag{7}$$

with $\Sigma_0 := \xi \cdot \Sigma_\phi, \xi > 1$. We chose $\xi = 5$.

E. Object Profiles

In this subsection we introduce two different so-called *object profiles* $\mathcal{P}_1$ and $\mathcal{P}_2$ we used for our tracking system. The object profiles define which features $\mathcal{F}(x)$ describe an object state x. Due to these features a person is recognized and tracked from frame to frame. Furthermore, for each object profile we define a distance function $D_{\widetilde{\mathcal{R}}}(x) := D\left(\widetilde{\mathcal{R}}, \mathcal{F}(x)\right)$ which determines the similarity (or better *dissimilarity*) between a *reference profile* $\widetilde{\mathcal{R}} := \mathcal{F}(\hat{x}_0)$ characterizing the target object and the features $\mathcal{F}(x)$ of a current hypothesis x. (The result of the distance function $D_{\widetilde{\mathcal{R}}}$ is finally converted into the particle weight of the considered hypothesis.)

We use color histograms as features. Let $H(p_x, p_y, a, a \cdot r)$ be a color histogram of a rectangular area with center (p_x, p_y), width a, and height $a \cdot r$. Furthermore, let $\Delta(H_1, H_2)$ be a distance function between two histograms H_1 and H_2. (It is defined in formula (15).) The two used object profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ are illustrated in Fig. 6 and are defined now as follows.

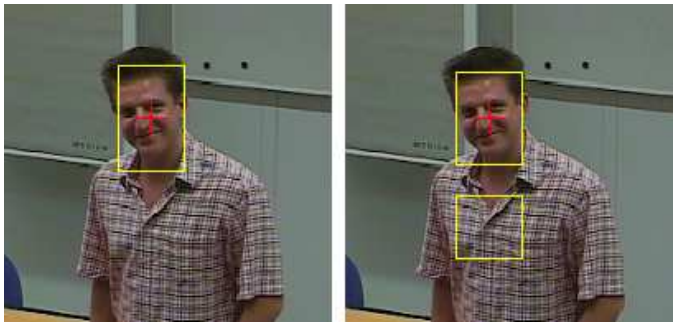

Fig. 6. Two different types of object profiles. Left: profile $\mathcal{P}_1$, right: profile $\mathcal{P}_2$.

1) Profile $\mathcal{P}_1$: Restricted Ratio of Rectangle Width and Height: This profile consists of one histogram H_T only. If the target object is a person H_T should cover the head of the person (see left image of Fig. 6). To prevent the rectangle from degeneration we control the ratio of width and height by restricting r of the state vector $x := (p_x, p_y, \dot{p}_x, \dot{p}_y, a, r, \rho)^T$ as described in subsection II-C ($r \in [1.2; 1.8]$). Features $\mathcal{F}$ and reference profile $\widetilde{\mathcal{R}}$ are then defined as follows:

$$\begin{aligned} \mathcal{F}(x) &:= H_T(x) = H(p_x, p_y, a, a \cdot r) \\ \widetilde{\mathcal{R}} &:= H_T(\hat{x}_0) \end{aligned} \tag{8}$$

For the calculation of the similarity of a hypothesis $x := x_t^{(i)}$ with the reference profile $\widetilde{\mathcal{R}}$ the histogram distance function Δ can be applied directly:

$$D_{\widetilde{\mathcal{R}}}(x) := \Delta\left(H_T(x), \widetilde{\mathcal{R}}\right) \tag{9}$$

2) Profile $\mathcal{P}_2$: Two Rectangles: This object profile is customized especially for the application of the tracking system to moving persons. As a person's front view does not only contain skin colors in the face area, but also, e.g., at the hands, these areas can easily be confused with the face if only one color histogram is evaluated. To prevent confusion we integrate a second rectangular area in profile $\mathcal{P}_2$ the histogram H_B of which is evaluated. This approach is similar to that described in [18]:

$$\begin{aligned} \mathcal{F}(x) &:= \begin{pmatrix} H_T(x) \\ H_B(x) \end{pmatrix} \\ \widetilde{\mathcal{R}} &:= \begin{pmatrix} \widetilde{\mathcal{R}}_T(x) \\ \widetilde{\mathcal{R}}_B(x) \end{pmatrix} \end{aligned} \tag{10}$$

H_B has a square shape with side length a and is (as in [18]) positioned below the region of the first histogram H_T (see right image of Fig. 6). But in contrast to the approach described in [18] we employ a dynamic positioning of the second rectangle on a circular arc with center (p_x, p_y) and radius $b = 1.7a$. These circumstances are depcited in Fig. 7. The extent of the circular arc is determined by the parameter $\rho \in [-\rho_{max}; \rho_{max}]$ which is the last component of the object state vector (see formula (3)). We obtained the best results with $\rho_{max} = 20 \cdot \frac{2\pi}{360}$. This flexible positioning of the second rectangle should improve the tracking in situations where the person displays a laterally crooked posture or when partial

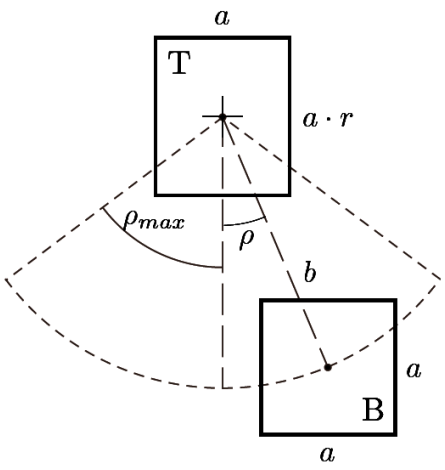

Fig. 7. Object profile $\mathcal{P}_2$. The second rectangle is positioned dynamically on a circular arc.

occlusions below the upper rectangle occur. The histograms are calculated as follows:

$$\begin{array}{rcl} H_T\left(x\right) & := & H\left(p_x, p_y, a, a\cdot r\right) \\ H_B\left(x\right) & := & H\left(p_x + 1.7a\cdot sin(\rho), p_y + \right. \\ & & \left. +1.7a\cdot cos(\rho), a, a\right) \end{array} \quad (11)$$

For the evaluation of the similarity of a hypothesis x a weighted mean of both histogram distances $D_{\widetilde{\mathcal{R}}_T}(x)$ and $D_{\widetilde{\mathcal{R}}_B}(x)$ is calculated:

$$\begin{array}{rcl} D_{\widetilde{\mathcal{R}}}\left(x\right) & := & \gamma^+\left(x\right)\cdot\left[0.6\cdot D_{\widetilde{\mathcal{R}}_T}(x) + 0.4\cdot D_{\widetilde{\mathcal{R}}_B}(x)\right] \\ D_{\widetilde{\mathcal{R}}_T}\left(x\right) & := & \gamma^-\left(\Delta\left(H_T(x), \widetilde{\mathcal{R}}_T\right)\right) \\ D_{\widetilde{\mathcal{R}}_B}\left(x\right) & := & \gamma^-\left(\Delta\left(H_B(x), \widetilde{\mathcal{R}}_B\right)\right) \end{array} \quad (12)$$

The face region is weighted stronger whereas the rectangle in the bottom can be regarded as an auxiliary region. In addition to this, increases and decreases of the different distance values are carried out depending on heuristic measures governed by the following two rules:

1) Decrease of both single histogram distances by γ^- in case they already fall below a threshold:

$$\gamma^-(s) \quad := \quad \left\{ \begin{array}{ll} 0.9s & s < 0.3 \\ s & otherwise \end{array} \right. \quad (13)$$

2) Increase of the total distance by γ^+ in case both single histogram distances still exceed a threshold after the application of rule (1):

$$\gamma^+(x) := \left\{ \begin{array}{ll} 1.1 & D_{\widetilde{\mathcal{R}}_T}(x) \geq 0.4 \; and \; D_{\widetilde{\mathcal{R}}_B}(x) \geq 0.4 \\ 1 & otherwise \end{array} \right. \quad (14)$$

These rules reinforce a positive as well as a negative bias of the distance values based on experimentally derived thresholds.

F. Histogram Distance

What remains now is the definition of the distance $\Delta(H_1, H_2)$ between two histograms H_1 and H_2. The structure of the histograms we use is based on those utilized in [17] and [18] and is characterized by the color space and the number of the bins. To achieve a larger robustness of the acquired color information against lightness variations we use the HSL color space and partition it into 32 bins. 24 of them are reserved for hue and saturation ($H \times S$), 8 for lightness (L). However, the lightness information is utilized only for those pixels that do not provide reliable color information. We threshold the minimal saturation with $\chi = 16$ out of a maximum of 255, because below this threshold the signal starts to become noisy. The assignment of a pixel to a bin of the histogram is described by the function $\mathcal{B} : H \times S \times L \longrightarrow \{1, \ldots, 32\}$:

$$\mathcal{B}(u_h, u_s, u_l) \quad := \quad \left\{ \begin{array}{ll} \mathcal{B}_L(u_l) & 0 \leq u_s < \chi \\ \mathcal{B}_{HS}(u_h, u_s) & \chi \leq u_s \leq 255 \end{array} \right.$$

The case differentiation represents the separation of the bin partitioning and $\mathcal{B}_L$ and $\mathcal{B}_{HS}$ are the corresponding lookup tables. We employ a uniform allocation of the pixels to the bins. In Fig. 8 the described calculation of the histograms is depicted.

Summarizing, we have the following formal description of a normalized histogram H on the rectangle image section denoted by $\square$ with center (p_x, p_y) and size $a \times a \cdot r$:

$$H(p_x, p_y, a, a\cdot r) \quad := \quad \frac{1}{a\cdot ar}\cdot\left(\begin{array}{c} h(1, \square) \\ \vdots \\ h(32, \square) \end{array}\right)$$

with $h(i, \square)$ as function which counts the number of pixels in $\square$ for bin i:

$$h(i, \square) \quad := \quad \sum_{u\in\square} \delta\left(\mathcal{B}(u) - i\right), \qquad i \in \{1, \cdots, 32\}.$$

The normalization by the factor $1/(a \cdot ar)$ allows for the comparison of histograms derived from base areas of different sizes. For the purpose of an efficient calculation of the histograms we enhanced the concept of *integral histograms* ([21], [22]) for particle filters. This is described in detail in [1].

Finally we are able to define the distance between two histograms H_1 and H_2 as follows:

$$\Delta(H_1, H_2) \; := \; \sqrt{1 - \sum_i \sqrt{H_{1,i}\cdot H_{2,i}}}\,, i \in \{1, \cdots, 32\}. \quad (15)$$

where $H_{1,i}$ and $H_{2,i}$ are the i-th bins in the corresponding histogram vector. Δ is the Bhattacharyya distance, which is used in [17] and [18] as well.

G. Adaption of the Reference Profile

For a static, color-based reference profile $\widetilde{\mathcal{R}}$ as introduced in subsection II-E, which is calculated only once, color variations due to changes in lightning or viewpoint pose a serious problem. For this reason we modeled the reference profile, similar to the approach described in [17], dynamically. In case the particle weighting w_t for the currently estimated object state $\hat{x}_t$ (see formulas (1) and (2)) exceeds a minimum value

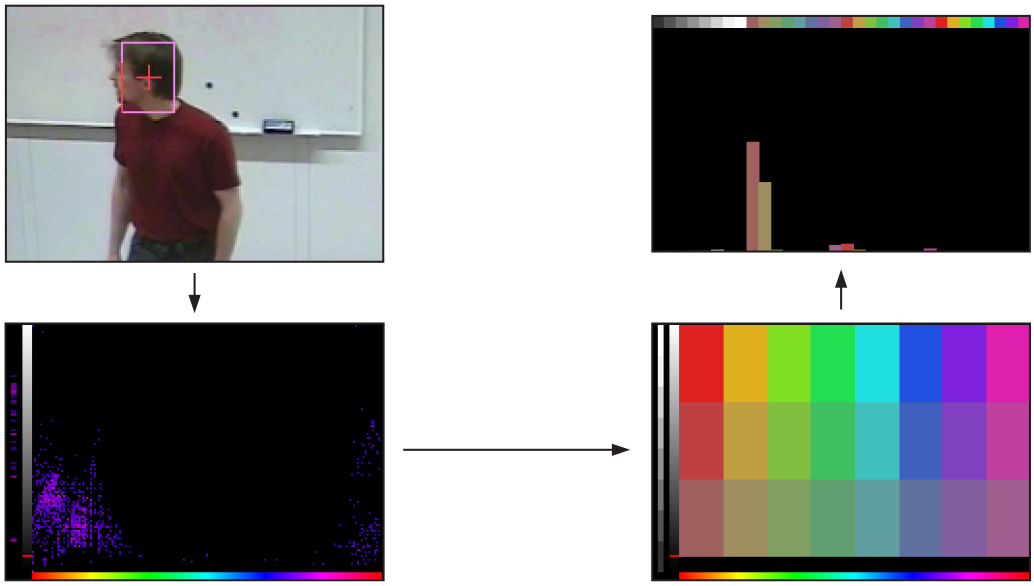

Fig. 8. Calculation of the histograms. The image in the bottom left shows the detailed $(H \times S, L)$ histogram coding the pixel occurrences of the rectangle displayed in the upper left image. In the bottom right image the quantization into $8 \cdot 3$ hue and saturation bins and 8 lightness bins (left hand side of this image) is depicted. This results in the used final histogram form displayed in the upper right image.

w_{min} ($w_t \geq w_{min}$) the data of the estimated object state are integrated into the current reference profil:

$$\begin{aligned} \widetilde{\mathcal{R}}_0 &:= \mathcal{F}(\hat{x}_0) \\ \widetilde{\mathcal{R}}_t &:= \kappa\widetilde{\mathcal{R}}_0 + (1-\kappa)\left[\lambda\widetilde{\mathcal{R}}_{t-1} + (1-\lambda)\mathcal{F}(\hat{x}_t)\right] \end{aligned}$$

$\widetilde{\mathcal{R}}$ is updated componentwise, i.e., for object profile $\mathcal{P}_2$ both histograms are refreshed separately. In contrast to the approach described in [17] the initial object profil $\widetilde{\mathcal{R}}_0$ remains in the current profile with a fixed fraction κ. This prevents the color histograms from beeing totally changed. We chose the following values of the parameters: $w_{min} = 0.25, \kappa = 0.15$, and $\lambda = 0.15$.

H. Competing Reference Profiles

To prevent the reference profile $\widetilde{\mathcal{R}}$ from beeing adapted to wrongly tracked image parts the parameters mentioned in subsection II-G should be chosen carefully. Nevertheless, it is necessary to incorporate different color distributions for one target object as exemplified in Fig. 9. For this reason we utilize, similar to the approach described in [9], several competing reference profiles $\widetilde{\mathcal{R}}^{(j)}$. The similarity of each hypothesis $x_t^{(i)}$ is then calculated as maximum of the similarities for each of the competing reference profiles:

$$D_{\widetilde{\mathcal{R}}}\left(x_t^{(i)}\right) := \max_j \left\{ D_{\widetilde{\mathcal{R}}^{(j)}}\left(x_t^{(i)}\right) \right\}$$

For best results the competing reference profiles should describe clearly distinct representations of the target object, such as front and side views. The competing profiles have been arranged in a circular buffer which means that a new initialization of the profile by the user overwrites an older reference profile in case all positions in the circular buffer are allocated. In our experiments we provided two competing reference profiles for each target object. For all different parameter configurations we analyzed in the experiments the user has initialized for each sequence both reference profiles always in the same frame. An example of a target object successfully tracked because of competing reference profiles is shown in Fig. 15.

I. Adaptive Particle Diffusion

As mentioned in the overview (subsection II-A) the complete Virtual Camera Assistent consists of the two modules *Object Tracking* and *Camera Control*. They interact with each other inter alia by an adaption of the tracking module to the movements of the cameras. In doing so it does not matter whether the camera movements have been induced by the autonomous feedback loop between both modules as indicated in Fig. 3 or the camera - as part of the environment - has been moved by the user. In any case a dynamic adaption of the particle diffusion to the recognized camera movement takes place. This means that the target object is acquired depending on the recognized state of the environment.

As each camera pan causes a shift of the current target object in the image in the opposite direction, the basic idea of the camera movement-controlled particle diffusion consists in a coupling of the particle distribution to the movement of the camera in such a way that the particles are placed smarter before they are weighted by formula 1. The velocity and direction of the camera movement should affect the random particle distribution in the dynamic model. For this purpose we decide for each particle with probability ψ if, instead of ϕ_t (see formula (4)), we use a different distribution ϕ_t^* for the random part of the dynamic model (as in subsection II-D we again omit index i, which indicates a single particle):

$$x_t = \mathbf{A}x_{t-1} + \phi_t^* \tag{16}$$

Fig. 9. Necessity for competing reference profiles. Different views of a target person imply different color distributions in the histograms.

As the camera parameters can be separated into horizontal (pan) and vertical (tilt) movement and the variation of the focal length, for each of these parameters an element of the object state vector can be modified. According to this, the random vector ϕ_t^* differs from ϕ_t only in those three entries which describe the random part of the position (p_x, p_y) and width a of the target object:

$$\phi_t^* := \begin{pmatrix} -s_x \cdot |\phi_{p_x}| \\ -s_x \cdot |\phi_{p_x}| \\ \bullet \\ \bullet \\ \phi_a \\ \bullet \\ \bullet \end{pmatrix} \quad with \quad \begin{array}{c} \phi_{p_x} \in \mathcal{N}(0, \sigma_{p_x}), \\ \phi_{p_y} \in \mathcal{N}(0, \sigma_{p_y}), \\ \phi_a \in \mathcal{N}(\mu_a, \sigma_a). \end{array} \tag{17}$$

The used values for $\sigma_{p_x}, \sigma_{p_y}, \mu_a$, and σ_a are listed at the end of this section. $s_x, s_y \in [-1, 1]$ refer to the current pan and tilt direction, respectively. A change of the focal length usually causes only small changes in the projected width of the target object. This is the reason why changes of a are caused only by noise. A slight shift of the expected value or a slight boost of the standard deviation yields the desired effect. In contrast to this, the shift of the position of the object is quite distintive in terms of the horizontal and vertical movements so that only a wide particle distribution can cover the shift sufficiently. But on the other hand, a strongly shifted expectation value leads to an insufficient covering of the object at the old position, whereas a large standard deviation implies a wider positioning also in the opposite direction. For this reason the absolute values of the random variables in formula (17) ensure the necessary, unambiguous bias in the distribution of the particles. The negative sign secures the desired opposite direction of the distribution. Summarizing, the additive noise of the dynamic model does not display an expectation value of zero anymore, rather it takes a componentwise shift of the mean value into the opposite direction of the camera movement into account.

In Fig. 10 two examples for the adaptive particle diffusion are shown. The combination of the first and second image illustrates one example, the combination of the third and fourth image a second example. First example: The left image is a snapshot from a sequence with a camera pan to the left. The right image displays the final frame of this sequence. To illustrate the effect of the modified distribution of the particles by the adaptive particle diffusion we used a large value for the standard deviation of the horizontal movement, namely $\sigma_{p_x} = 50$. As probability ψ for the choice of this modified distribution we set $\psi = 0.3$ throughout all experiments. The modified particles are marked by a beige colored top bar. (Thus about 30% of the present particles are marked.) In the left image one can recognize that during a camera pan to the left the modified particles have a stronger bias to diffuse to the opposite direction. The unmodified particles realize the movement only after their modification by the measurements as can be seen in the right image. Second example: Here the same holds true only for the opposite direction of the camera pan to the right.

We employ a value of $\psi = 0.3$ for the probability of a modified distribution. Table 1 summarizes the used values of the parameters of formula (17). To allow for a more flexible differentiation of camera velocities we separate the camera parameters (pan, tilt, and focal length) into three velocity categories, namely slow, medium, and fast velocities. For each of these categories we define separate parameters for the underlying distribution. They are summarized in table 1. The

parameter	slow velocity	medium velocity	fast velocity
σ_{p_x}	10.0	20.0	50.0
σ_{p_y}	8.0	15.0	30.0
μ_a	0.5	1.0	1.5
σ_a	1.0	2.0	3.0

Tab. 1. Parameter values for different velocities of camera movements.

division of the velocities into slow, medium, and fast depends on the adjustments the used camera (here camera JVC TK-C655) allows for. For pan and tilt the camera provides 8 levels, for focal length 3. These levels are quantized heuristically into the three categories. Details are described in [1].

III. Experiments

We analysed statistically different parameter configurations of the proposed tracking system. In particular, we examined the gain of the expansion of the condensation algorithm on the one hand and the account of the enhanced object profil $\mathcal{P}_2$ on the other hand. These issues represent two of the main contributions of this paper. The results of these experiments are reported in subsections IV-A and IV-B. Furthermore, we carried out several tracking experiments which reflect the

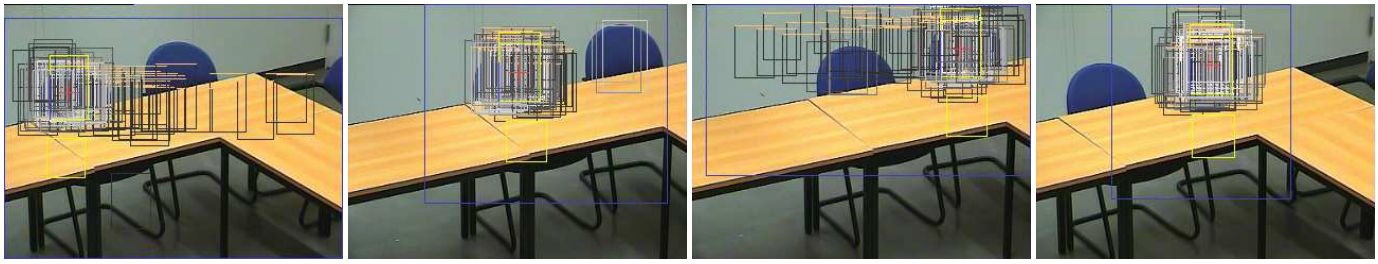

Fig. 10. Adaptive particle diffusion. For an explanation see subsection II-I.

capacities of the person tracking system according to the demands pronounced in the introduction. In the remaining subsections of section IV the results of those experiments are reproduced. For example, the adaption to changing camera parameters (third demand) is addressed in the examples of subsections IV-D and IV-E, results for sequences with occluded target persons are given in subsection IV-F (fourth demand), and the flexibility of the system in terms of user interaction (fifth demand) is the subject of subsection IV-G.

The experiments have been carried out on 3 GHz machines with image resolutions of 352×288 pixels. Although we used 200 particles only, this ensured tracking with even 1500 particles at a frame rate of 25 fps without difficulty.

A. Test Sequences

We carried out our experiments on three sequences with interacting persons in a seminar scene. The target person is selected interactively by the user and tracked autonomously according to the methods described in the last section. The test sequences (TS) are characterized as follows with increasing degree of complexity:

(TS1) Mainly one slightly moving person only at a time, relatively static camera, (typical seminar scene with moving instructor and relatively static audience)

(TS2) Mainly one moving person only at a time, but additionally a zooming and panning camera following the person's movements (and thus a dynamically changing background)

(TS3) Many simultaneously but non-uniformly moving persons with manycrossing roads and mutual occlusions; in addition, we have interaction by the user,who selects different target persons

A subsequence of TS1 is shown in Fig. 14, subsequences of TS2 are displayed in Fig. 15, Fig. 16, and Fig. 17, and Fig. 18, Fig. 19, and Fig. 20 show subsequences of TS3.

B. Error Calculation

Besides the visual analysis of the results from test sequences we examine differences between single applied methods and their variations to evaluate the quality of our tracking approach. As the particle filter is a randomized technique the comparisons reported in the next section are based on 100 independent runs with identical adjustments and predefined start parameters for the target person. For each frame the position and size of the head of a tracked person was manually selected to define the ground truth for the experiments. Deviations from these values by the autonomously tracked positions and sizes are for each frame t of a sequence expressed by a *frame error* ϵ_t and summarized to a *totel error* ϵ of the whole sequence. For frames in which the visible part of the head of the target person is smaller than one third of its real size we evaluate a positive recognition with a maximal error $\epsilon_{max} := 100$. In addition, we penalize a loss of the target person with ϵ_{max} as well. Summarizing, the total error ϵ of a sequence with t_{max} frames is the mean of all single frame errors ϵ_t:

$$\epsilon_t := \begin{cases} \epsilon_{max} & tracking\ failure \\ min\{\epsilon_{max}, \|\hat{x}_t - g_t\|_2\} & otherwise \end{cases}$$

$$\epsilon := \frac{1}{t_{max}} \sum_{t=1}^{t_{max}} \epsilon_t$$

$\|\cdot\|_2$ is the L_2-norm, $\hat{x}_t$ is the estimated object state as defined in formula (2), and g_t denotes the ground truth for frame t.

To recognize differences in terms of ϵ between different configurations of the system we carry out a comparison of means by a significance test (Duncan's test, significance niveau $\alpha = 0.01$). For illustration the 99% confidence intervals of single configurations are displayed in the figures of subsection IV-B.

IV. Results

In this section results from experiments with moving persons in realistic seminar scenes are reported. These results are obtained mainly by visual inspection of tracking examples of subsequences of the test sequences TS1, TS2, and TS3. But we also analyse tracking errors for different parameter configurations of the proposed tracking system. In the image examples the current estimated position and size of the target person are marked by a yellow rectangle with a red cross in its center. As soon as the rectangles are colored red the target person was lost (as, e.g., shown in Fig. 14). If the tracking was successful blue bars in the bottom left of the yellow rectangles visualize the confidence value of the estimation.

A. Expansion of the Condensation Algorithm

The reiteration of the Condensation algorithm as described in subsection II-B improves the tracking accuracy particularly

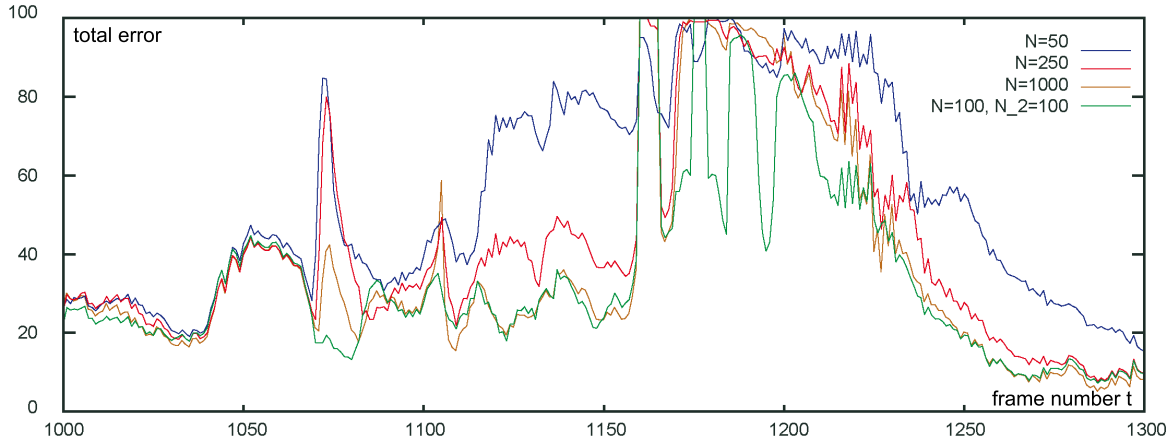

Fig. 11. Gain of the expansion of the condensation algorithm. In this diagram frame errors ϵ_t are plotted against a series of frames t of TS2. Four curves are displayed: three for three different numbers N of particels and one (the green one) for the configuration with reiteration. The chosen subsequence of TS2 contains a strong horizontal camera movement around frame $t = 1075$. As one can see, a sole increase of the number of particles cannot compensate for the camera pan, whereas the proposed expansion of the condensation algorithm can cope with it.

Fig. 12. Significance test for the expansion of the condensation algorithm. Total errors ϵ are plotted for different parameter configuration for the three test sequences TS1, TS2, and TS3. The errors are depicted as 99% confidence intervals.

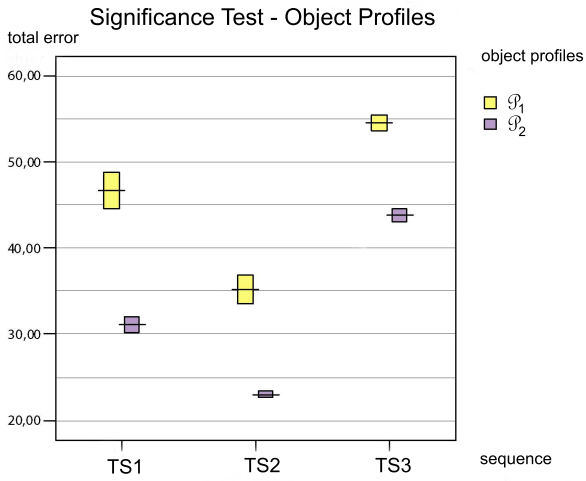

Fig. 13. Significance test for object profiles. Total errors ϵ are plotted for the two object profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ for the three test sequences TS1, TS2, and TS3. The errors are depicted as 99% confidence intervals.

in the case of intense camera pans, which, e.g., occur in TS2. In Fig. 11 a diagram is presented which illustrates this effect.

In Fig. 12 results of significance tests for the three test sequences are displayed. A consideration of the results for different parameters suggests that the second iteration (i.e., the first measure of improvement reported in subsection II-B) is crucial for an improvement of the tracking. Neither the increase of the number of particles for the second iteration nor the increase of N_{max} for the next step indicates a significant improvement. Summarizing, the second iteration of the condensation algorithm is a reasonable alternative to the increase of the number of particles. It can be applied selectively for problems such as camera pans by the adjustment of the threshold α_1 for its activation.

B. Object Profiles in Comparison

In subsection II-E we introduced two different object profiles $\mathcal{P}_1$ and $\mathcal{P}_2$. A significance test for both profiles is displayed in Fig. 13. As expected, the tracking results are better for profil $\mathcal{P}_2$, because the additional color area can resolve ambiguities in a more efficient way than profile $\mathcal{P}_1$. The examples reported in the remains of this section have been obtained with profile $\mathcal{P}_2$.

C. Lost and Found - Functionality of the Particel Filter

Fig. 14 illustrates the functionality of the particel filter, which is able to relocate a totally lost target object by employing multiple hypotheses. In particular the third, fourth, and fifth image show the expansion of the particle cloud in the case of a loss of the target person. But the cloud is

Fig. 14. Lost and found - functionality of the particel filter. Frames of TS1 for $t = 1309, 1324, 1337, 1383, 1384,$ and 1403 are displayed. (The additional partitionings in the top of the upper rectangles in this and the following figures belong to a third object profile not reported in this article.)

attracted again by the correct target person as soon the target is recognized again by at least one hypothesis.

D. Exposure to Changes in Camera Parameters - Pan

Fig. 15 shows a scene from TS2 with a moving person and a continuously panning camera. The person tracking is stable and precise even if the background is partly cluttered with other persons, although the confidence values in images 6 and 7 decrease significantly as visualized by the short confidence bars.

Fig- 16 shows another example of a sequence with a panning camera. Here the target person has a large velocity itself, and in addition, also the focal length of the camera varies. Even if the camera captures the target object only partly at the edges of a frame the estimations remain on the target.

E. Exposure to Changes in Camera Parameters - Focal Length

The other camera parameter that should be variable without affecting the tracking module is the focal length. Fig. 17 shows an example where the camera zooms into the scene. The person is tracked robustly across the different focal lengths and with high confidence values.

In Fig. 18 another example for the exposure of the tracking system to varying focal lengths of the camera is displayed. The camera zooms out of the scene and as soon as the person who disappeared in the previous frames reappears, its position and size are estimated correctly again, although now displayed at a different focal length.

F. Occlusions

Occlusions belong to the most difficult challenges to deal with in tracking systems. The number of occlusions in sequence TS3 is quite high. But the target persons are relocated altogether after a short period of time while they were occluded, frequently only a few frames after they reappeared in the captured scene. Fig. 19 shows three example scenes from TS3 which are effortlessly bridged by the tracking module. In the third image of the bottom row, for example, one can infer from the length of the confidence bar the decreased weighting of the estimation for the tracked woman. Nevertheless, she is tracked on robustly after her total occlusion.

G. Interactive Selection of Another Target Object

The selection of a new target person is done by the user of the system. She has to mark the head of the new target person. In TS3 the user has changed the target person several times. Fig. 20 shows an example where the system immediately takes the control and tracks the newly selected person.

H. Weaknesses of the Proposed Tracking System

Some weaknesses of the system striked during the experiments. One problem consists in the fact that the object profile consists of color information only. Although the expansion of profile $\mathcal{P}_1$ by a second, dynamic color histogram area improves the tracking significantly there still exist cases in which the particle cloud expands after an occlusion of the target object, and rearranges at a wrong image area with similar colors. It can happen that, even if the target object reappears in the scene, the particles remain at the wrong position until the

Fig. 15. Exposure to changes in camera parameters - pan. Depicted are frames $t = 475, 500, 545, 580, 615, 655, 675,$ and 700 of TS2 which render snapshots of a smoothly panning camera. Two competing reference profiles used as described in subsection II-H are crucial for the successful tracking of this sequence. The proposed system can keep track with the camera movement.

Fig. 16. Exposure to changes in camera parameters - truncations at frame edges. We see frames $t = 1897, 1919, 1935, 1960, 1980, 2000, 2010,$ and 2020 of sequence TS2.

similarity drops below the threshold and some particles detect the target again by chance.

Another weakness is the strong dependence of the tracking success on the initial selection of the target object by the user in form of a rectangle. Especially during camera movements the marking of the target can be too unprecise, resulting in the incorporation of wrong color information in the profil. This holds true as well for target persons who display a laterally crooked posture when profil $\mathcal{P}_2$ is employed. As the initial selection always places the bottom rectangle vertically below the head region, in this case it is positioned partly on the background, resulting in wrong color histograms in the object profil.

V. Conclusions

In this article we introduced an object tracking system which is capable to handle difficult situations in a dynamically changing environment. We evaluated the concepts of the proposed system (e.g., an improved version of the condensation algorithm or particle diffusion adaptive to variations in the environment) by applying it to the task of person tracking in crowded seminar rooms. The demands made on the system comprise robust real-time tracking of a target person who is allowed to move freely within a group of other persons and thus can be occluded. Furthermore, the background may change from frame to frame, and the tracking method should cope with dynamically varying camera parameters as, for example, induced by a user. In addition, the user of the system should be enabled to interactively select a new target person during tracking of another person.

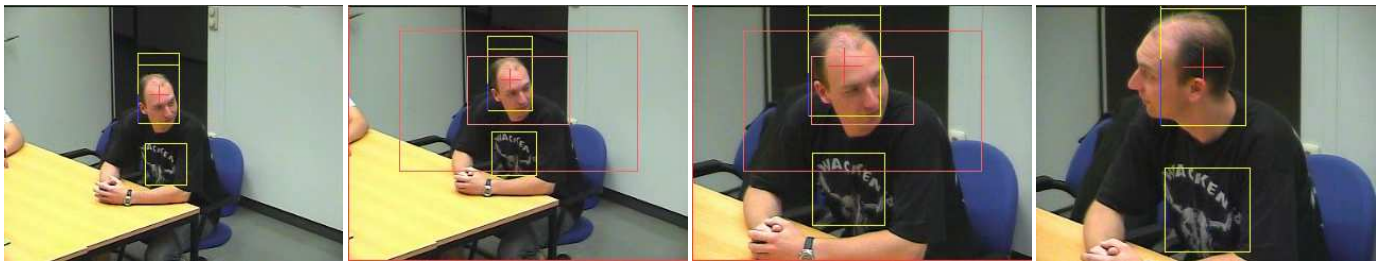

Fig. 17. Exposure to changes in camera parameters - focal length. These are the frames $t = 2620, 2661, 2715$, and 2760 of sequence TS2. (The additional red rectangles belong to the camera control module, which is not subject of this article.)

Fig. 18. Exposure to changes in camera parameters - successful relocation after zooming out. Frames $t = 1350, 1375, 1400, 1425, 1450, 1475, 1502$, and 1527 of sequence TS3 are shown.

Our contributions are threefold. First, we proposed an expansion of the condensation algorithm which results in a more stable tracking in difficult situations such as sudden camera movements. Secondly, we introduced a new way of particle diffusion which allows for the adaption of the tracking module to movements of the camera. These two contributions apply not only to person tracking but to object tracking in general. The third contribution consists in a more flexible way how to represent a person than propagated in previous publications. This contribution applies to person tracking only. Summarizing, the proposed tracking module mostly meets the postulated demands. Real-time tracking and the interactive selection of new target persons are possible. The challenges of a dynamically changing background and multiple occlusions could largely be coped with.

Ongoing research concentrates mainly on an expansion of the current object profiles, which are based on color information only. The incorporation of additional features could disambiguate situations where the current system fails because of similar color distributions in target and background. Moreover, it seems to be promising to synthesize a behavioral model which allows for the prediction of future directions of moving persons. This could represent the basis for an even more advanced and intelligent automatization.

REFERENCES

[1] Kluger, M., Partikelfilterbasierter, virtueller Kameraassistent zur Verfolgung einer Person in einer Gruppe von Menschen, *Diploma Thesis*, University Dortmund, September 2006.

[2] Bianchi, M., Automatic Video Production of Lectures Using an Intelligent and Aware Environment, *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia*, ACM Press New York, USA, pp. 117-123, 2006.

[3] Rui, Y., Gupta, A., Grudin, J., and He, L., Automating Lecture Capture and Broadcast: Technology and Videography, *ACM Multimedia Systems Journal*, 10(1), pp. 3-15, 2004.

[4] Trivedi, M., Huang, K., and Mikic, I., Dynamic Context Capture and Distributed Video Arrays for Intelligent Spaces, *IEEE Transactions on Systems, Man, and Cybernetics*, 35(A), pp. 145-163, 2005.

[5] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S., Multi-Camera Multi-Person Tracking for EasyLiving, *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, Washington, USA, IEEE Computer Society, 2000.

[6] Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L., Background Modeling and Subtraction by Codebook Construction, *IEEE International Conference on Image Processing*, pp. 3061-3064, 2004.

[7] Lin, C., Chang, Y., Wang, C., Chen, Y., and Sun, M., A Standard-Compliant Virtual Meeting System with Active Video Object Tracking, *EURASIP Journal on Applied Signal Processing*, 6, pp. 622-634, 2002.

[8] Nicolescu, M. and Medioni, G., Electronic Pan-Tilt-Zoom: A Solution for Intelligent Room Systems, *IEEE International Conference on Multimedia and Expo*, pp. 1581-1584, 2000.

[9] Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., and Gool, L., Color-Based Object Tracking in Multi-Camera Environments, *Proceedings of the 25th DAGM Symposium on Pattern Recognition*, Lecture Notes in Computer Science 2781, pp. 591-599, 2003.

Fig. 19. Occlusions. Three sample scenes for TS3 with occlusions are shown. Upper row: frames $t = 700, 717, 726$, and 735; middle row: frames $t = 2735, 2745, 2749$, and 2757; bottom row: $t = 4950, 4958, 4964$, and 4975.

Fig. 20. Interactive selection of another target object. The frames $t = 1815, 1845, 1870$, and 1923 of TS3 are displayed. In the second image the man with the white T-shirt is selected for tracking instead of the previously tracked man with the red T-shirt. He is successfully taken over by the tracking module.

[10] Hu, W., Tan, T., Wang, L., and Maybank, S., A Survey on Visual Surveillance of Object Motion and Behaviors, *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3), pp. 334-352, 2004.

[11] Martínez-Tomás, R., Rincón, M., Bachiller, M. and Mira, J., On the Correspondence Between Objects and Events for the Diagnosis of Situations in Visual Surveillance Tasks, *Pattern Recogn. Lett.*, 29(8), pp. 1117-1135, 2008.

[12] Park, S. and Trivedi, M. M., Understanding Human Interactions with Track and Body Synergies (TBS) Captured from Multiple Views, *Computer Vision and Image Understanding*, 111(1), pp. 2-20, 2008.

[13] Moeslund, T. and Granum, E., A Survey of Computer Vision-Based Human Motion Capture, *Computer Vision and Image Understanding*, 81(3), pp. 231-268, 2001.

[14] Zhao, T., Nevatia, R., and Wu, B., Segmentation and Tracking of Multiple Humans in Crowded Environments, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7), pp. 1198-1211, 2008.

[15] Ess, A., Leibe, B., Schindler, K., and van Gool, L., Moving Obstacle Detection in Highly Dynamic Scenes, *IEEE Int. Conf. on Robotics and Automation, ICRA 2009*, 2009.

[16] Isard, M., Visual Motion Analysis by Probabilistic Propagation of Conditional Density, *PhD Thesis*, Oxford University, 1998.

[17] Nummiaro, K., Koller-Meier, E., and Gool, L., An Adaptive Color-Based Particle Filter, *Image and Vision Computing*, 21(1), pp. 99-110, 2002.

[18] Perez, P., Hue, C., Vermaak, J., and Gangnet, M., Color-Based Probabilistic Tracking, *Proceedings of the 7th European Conference on Computer Vision*, Lecture Notes In Computer Science 2350, pp. 661 - 675, 2002.

[19] Doucet, A., Godsill, S, and Andrieu, C., On Sequential Monte Carlo Sampling Methods for Bayesian Filtering, *Statistics and Computing*, 10(3), pp. 197-208, 2000.

[20] Deutscher, J., Blake, A., and Reid, I., Articulated Body Motion Capture by Annealed Particle Filtering, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2, pp. 126-133, 2000.

[21] Porikli, F., Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 829-836, 2005.

[22] Woelk, F., Schiller, I., and Koch, R. An Airborne Bayesian Color Tracking System, *IEEE Intelligent Vehicles Symposium*, pp. 67-72, 2005.

Gabriele Peters received the PhD (Dr. rer. nat.) degree from the Faculty of Technology of the University of Bielefeld, Germany, in 2002. She carried out her PhD studies at the Institute for Neural Computation, Ruhr-University Bochum, Germany. Afterwards she worked as postdoctoral research assistant at the Department of Computer Graphics of the Technical University of Dortmund, Germany, where she focused on machine learning for computer vision and human computer interaction. For several months she worked as research scientist at the Academy of Sciences of the Czech Republic in Prague in the field of image processing and as visiting professor in the Computational Vision Group at the California Institute of Technology in Pasadena, USA, in the field of computational photography. Since 2007 she is professor at the University of Applied Sciences and Arts in Dortmund, Germany, where she heads the Visual Computing Group. She is author of more than 40 peer-reviewed scientific publications. Among other awards, in 2003 Prof. Peters received the Rudoploh Chaudoire award of the Technical Universitiy of Dortmund for her scientific achievements. Since 2004 she is an elected member of the executive committee of the Association for Informatics (GI) and is granted since 2005 by the German Research Association (DFG).