Visual Person Tracking Using a Cognitive Observation Model

Simone Frintrop[1] Achim Königs[2] Frank Hoeller[2] Dirk Schulz[2]

Abstract— **In this article we present a cognitive approach to person tracking from a mobile platform. The core of the technique is a biologically inspired observation model that combines several feature channels in an object and background dependent way, in order to optimally separate the object from the background. This observation model can be learned quickly from a single training image and is easily adaptable to different objects. We show how this model can be integrated into a visual object tracker based on the well known Condensation algorithm. Several experiments carried out with a mobile robot in an office environment illustrate the advantage of the approach compared to the Camshift algorithm which relies on fixed features for tracking.**

I. Introduction

An important skill for mobile service robots is the ability to detect and keep track of individual humans in their surrounding. Especially robots that are designed to provide services to individual persons need to be able to distinguish their client from the surrounding environment. During the last decade, several algorithms have been developed for detecting and tracking people with mobile robots using laser range data, vision, or both [1], [2], [3], [4], [5], [6]. Most of these approaches have in common that they rely on a single pre-specified feature domain to compute cues that allow to discriminate the robot's client from other objects in the sensor data. For example, in vision-based approaches color histograms are often employed, or shape information is used. Laser-based approaches mainly rely on range-features extracted from the laser rage scans. However, relying on a single feature leads to the problem that depending on the actual environment conditions, the chosen feature might not be discriminative enough; well known problems for color-histogram based approaches are changing lighting conditions or a cluttered multi-colored background.

In this article we propose to employ a visual attention system for choosing the cues which best distinguish a person from the background depending on the situation the robot currently faces [7]. Based on a cognitive perception model [8], the attention system utilizes a larger set of different simple features to discriminate particular objects from the background. Depending on the environment and the appearance of the object to detect, it automatically determines the suitable cues by computing a weighting of the different features available, such that the resulting mixture discriminates the object from the background best. The attention system being used is able to compute such weight vectors from a single training image. A similarity measure based on these weight vectors is then usually applied for finding the object within images. Instead of searching for the object, we employ the similarity measure within a CONDENSATION-based person tracker [9]. For this purpose, the similarity measure is converted to a likelihood function that is used as the observation model within the particle filter. Using this approach, the robot is able to quickly learn the current appearance of the person it wants to track. This leads to an improved tracking performance, compared to tracking approaches based on single feature cues. In order to evaluate the technique, we implemented an application, where the robot follows a person on its way through our laboratory environment. The experiments show that the approach is able to track the person in varying lighting conditions and backgrounds, and that it is considerably less prone to track loss than, for example, the purely color-based Camshift algorithm.

The remainder of the article is organized as follows. After discussing related work in Section II we explain the cognitive tracking system in Section III. In Section IV we briefly explain how the approach is integrated into a prototypical person following application and we present experimental results. We finally conclude in Section V.

II. Related work

In mobile robotics, person tracking can be performed with different sensors. Several groups have investigated person tracking with laser range finders [1], [2], [3]. These approaches usually only keep track of the motion of people and do not try to distinguish individuals. One approach which distinguishes different motion states in laser data is presented in [4]. Combinations of laser and vision data are presented in [5] and [6]. Both detect the position of people in the laser scan and distinguish between persons based on vision data. Bennewitz et al. [5] base the vision part on color histograms whereas Schulz [6] learns silhouettes of individuals from training data. This however requires a time-consuming learning phase for each new person.

In machine vision, people tracking is a well-studied problem. Two main approaches can be distinguished: *model-based* and *feature-based* methods. In model-based tracking approaches, a model of the object is learned in advance, usually from a large set of training images which show the object from different viewpoints and in different poses [10]. Learning a model of a human is difficult because of the dimensionality of the human body and the variability in human motion. Current approaches include simplified human

[1] Simone Frintrop is with the Institute of Computer Science III, Rheinische Friedrich-Wilhems-Universität, 53117 Bonn, Germany. Contact: frintrop@iai.uni-bonn.de.

[2] All other authors are with the Research Institute for Communication, Information Processing and Ergonomics (FKIE), 53343 Wachtberg, Germany. Contact: {koenigs,hoeller,schulz}@fgan.de

body models, e.g. stick, ellipsoidal, cylindric or skeleton models [11], [12], [13], or shape-from-silhouettes models [14]. While these approaches have reached good performance in laboratory settings with static cameras, they are usually not applicable in real-world environments on a mobile system. They usually do not operate in real-time and often rely on a static, uniform background.

Feature-based tracking approaches on the other hand do not learn a model but track an object based on simple features such as color cues or edges. One approach for feature-based tracking is the Mean Shift algorithm [15], [16] which classifies objects according to a color distribution. Variations of this method are presented in [17], [18]. While most approaches are not especially designed for person tracking, they might be applied in this area as well. One limitation with the above methods is that they operate only on color and are therefore dependent on colored objects.

Visual attention systems are especially suited to automatically determine the features which are relevant for a certain object. These systems are motivated by mechanisms of the human visual system and based on psychological theories on visual attention [8], [19]. During the last decade, many computational attention systems have been built, e.g., [20], [7], and recently, some systems came up that are able to operate in real-time [21], [22], [23]. Important for our application is that the systems compute a feature vector that describes the appearance of a salient region [24], [7].

Applications of visual attention systems range from object recognition to robot localization. However, they have rarely been applied to visual tracking. Some approaches track static regions, such as visual landmarks, from a mobile platform for robot localization [25]. This task is easier than tracking a moving object since the environment of the target remains stable. Another approach aims to track moving objects such as fish in an aquarium [26]. In this case however, the camera is static. The here presented VOCUS tracker is partly based on [27]. We have also applied a simpler approach based on visual attention (but without particle filters) to object tracking [28] and to person tracking [29].

III. The Cognitive Tracking System

The tracking system we present is based on a particle filter approach with a cognitive observation model. It employs the standard Condensation algorithm [9] which maintains a set of weighted particles over time using a recursive procedure based on the following three steps: First, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle. Second, the particles are transformed (predicted) according to a motion model. In vision-based tracking this step usually consists of a drift component in combination with random noise. Third, all particles are assigned new weights according to an observation model.

In the following, we first introduce the notation (sec. III-A), second mention how the system is initialized (sec. III-B), and third describe the motion model (sec. III-C). Finally, we specify in detail the observation model as core of the system (sec. III-D).

A. Notation

At each point in time $t \in \{1,..,T\}$, the particle filter recursively computes an estimate of the probability density of the person's location within the image using a set of J particles $\mathbf{\Phi}_t = \{\phi_t^1, ...\phi_t^J\}$ with

$$\phi_t^j = (\mathbf{s}_t^j, \pi_t^j, \mathbf{w}_t^j), \quad j \in \{1, ..., J\}.$$

Here, $\mathbf{s}_t^j = (x, y, v_x, v_y, w, h)$ is the state vector that specifies the particle's region with center (x, y), width w and height h – in the following, the region is also denoted as $\mathbf{R}_t^j = (x, y, w, h)$. The v_x and v_y components specify the current velocity of the particle in the x and y directions. Each particle additionally has a weight π_t^j determining the relevance of the particle with respect to the target, and a feature vector $\mathbf{w}_t^j$ that describes the appearance of the particle's region.

B. Initialization

In order to start the tracking process, the initial target region $\mathbf{R}^* = (x^*, y^*, w^*, h^*)$ has to be specified in the first frame. This can either be carried out manually or automatically using a separate detection module. Based on the initial target region $\mathbf{R}^*$, a feature weight vector $\mathbf{w}^*$ is computed that describes the appearance of the person. The initial particle set

$$\mathbf{\Phi}_0 = \{(\mathbf{s}_0^j, \pi_0^j, \mathbf{w}_0^j) \,|\, j = 1, ..., J\}. \tag{1}$$

is generated by randomly distributing the initial target location around the region's center (x^*, y^*). The velocity components v_x and v_y are initially set to 0 and the region dimensions of each particle are initialized with the dimensions of $\mathbf{R}^*$. The particle weights π_0^j are set to $1/J$.

C. Motion model

Currently, the object's motion is modeled by a simple first order autoregressive process in which the state $\mathbf{s}_t^j$ of a particle depends only on the state of the particle in the previous frame:

$$\mathbf{s}_t^j = \mathbf{M} \cdot \mathbf{s}_{t-1}^j + \mathbf{Q}.$$

Here, $\mathbf{M}$ is a state transition matrix of a constant velocity model and $\mathbf{Q}$ is a random variable that denotes some white Gaussian noise. This enables a flexible adaption of position and size of the particle region as well as of its velocity. Thus the system is able to quickly react to velocity changes of the object.

D. Observation model

In visual tracking, the choice of the observation model is the most crucial step since it decides which particles will survive. It therefore has the strongest influence on the estimated position of the target. Here, we use a cognitive observation model which favors the most discriminative features in the current setting based on concepts of human visual perception. It determines the feature description for the target and for each particle, enabling the comparison and weighting of particles.

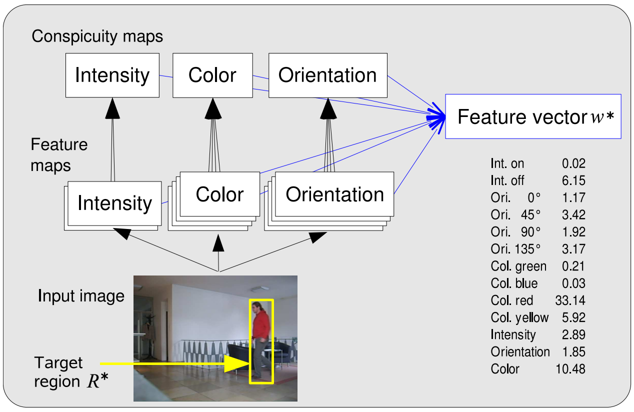

Fig. 1. Initialization: the attention system VOCUS learns the target appearance by computing feature and conspicuity maps for the image and determining a feature vector $\mathbf{w}^*$ for the manually provided search region $\mathbf{R}^*$ (yellow rectangle).

1) Computation of the feature vector: The feature vector is computed based on a cognitive perception model which computes the saliency of a region based on concepts of the human visual system (cf. Fig. 1). This *computational attention system* is called *VOCUS* and was originally built to simulate human eye movements [7]. It computes feature contrasts for different scales and feature types and assigns a saliency value to each image region. Additionally, a feature vector is computed for each salient region that determines the contribution of the different feature channels to the region.

In this paper, we use the system in a slightly different manner than the usual case: we do not determine the most salient regions in an image, but the feature saliency of predefined regions, the particle regions. However, the computation of the feature maps is the same.

The feature computations are performed on 3 different scales using image pyramids. The feature intensity is computed by *center-surround mechanisms* (similar to DoG filters); on-off (bright on dark) and off-on (dark on bright) contrasts are determined separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 color maps (green, blue, red, yellow) and 4 orientation maps $(0\,°, 45\,°, 90\,°, 135\,°)$ are computed. The color maps compute color contrasts based on the Lab color space (CIELAB), since this is known to approximate human perception well. To achieve real-time performance, the intensity and color maps are computed using integral images [30]. These provide an efficient way to determine the average value of a rectangular region of arbitrary size in constant time (4 operations per region), after once creating the integral image in linear time. For the orientation maps, Gabor filters highlight the gradients with a certain orientation (details in [7]).

Before the features are fused, they are weighted according to their *uniqueness*, i.e. a feature which occurs seldomly in a scene is assigned a higher saliency than a frequently occurring feature. This is a mechanism which enables humans to instantly detect outliers like a black sheep in a white herd. The uniqueness $\mathcal{W}$ of map X is computed as $\mathcal{W}(X) = X/\sqrt{m}$, where m is the number of local maxima that exceed a threshold. Here, '/' stands for the pixel-wise division of an image with a scalar. The weighted maps are summed up to 3 *conspicuity maps* for intensity, orientation, and color. In the following, we denote the 10 feature and 3 conspicuity maps for image I_t as $F_i(I_t), i \in \{1, .., 13\}$. In the original VOCUS system, the conspicuity maps are weighted again and fused into a saliency map. However, this map is not required in our approach.

For an arbitrary region in the image, a feature vector can be computed which describes the appearance of the region with respect to its surrounding. In the original system, feature vectors are computed for the most salient regions in a saliency map. Here, we compute a vector for each particle region. The feature vector $\mathbf{w} = (w_1, ..., w_{13})$ for a region $\mathbf{R}$ is computed as follows. For each map $F_i(I)$, the ratio of the mean saliency in the target region $\mathbf{R}$ and in the background $I \backslash R$ is determined as:

$$w_i = \frac{\text{mean}(\mathbf{R})}{\text{mean}(I \backslash \mathbf{R})}, \qquad i \in \{1, .., 13\}. \tag{2}$$

This computation does not only consider which features are the strongest in the target region, it also regards which features separate the region best from the rest of the image.

Since this computation involves computing the average value of a particle region of arbitrary size for a usually large collection of particles and for 13 feature maps, the process can be time consuming. To maintain real-time performance, the computations are also performed with integral images. This increased the average processing speed of VOCUS considerably from 10 Hz to 40 Hz. The result of the computations in this section is a feature vector $\mathbf{w}_t^j$ for each particle.

2) Weighting of the particles: The feature vector $\mathbf{w}_t^j$ of a particle ϕ_t^j is now used to determine the similarity of the particle region $\mathbf{R}_t^j$ with the initial target region $\mathbf{R}^*$. As similarity measure we use the Tanimoto-coefficient

$$T(\mathbf{w}^*, \mathbf{w}_t^j) = \frac{\mathbf{w}^* \cdot \mathbf{w}_t^j}{||\mathbf{w}^*||^2 + ||\mathbf{w}_t^j||^2 - \mathbf{w}^* \cdot \mathbf{w}_t^j}.$$

The Tanimoto coefficient produces values in the interval $[0, 1]$, the higher the value the higher the similarity. If the two vectors are identical, the coefficient is 1. Compared to Euclidean distance, it turned out that the Tanimoto coefficient is better suited to distinguish between true and false matches [28]. Based on the Tanimoto coefficient the weight of a particle is computed as

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\mathbf{w}^*, \mathbf{w}_t^j)}.$$

This function prioritizes particles which are very similar to the target vector $\mathbf{w}^*$ by assigning an especially high weight. A value of $\lambda = 14$ has shown to be useful in our experiments. The parameter c is a normalization factor which is chosen so that $\sum_{j=1}^{J} \pi_t^j = 1$.

3) Determining the target state: From the weighted particle set, the current target state, including target position and size, can be estimated by

$$\mathbf{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \mathbf{s}_t^j.$$

IV. Experiments and Results

The experiments were carried out using a RWI B21 robot equipped with a simple USB web camera mounted on a pan-tilt unit (see Fig. 2, left). The camera captures 15 frames/sec, with a resolution of 320×240. The complete software runs on a 2GHz dual core PC onboard the robot. For the experiments, the tracking application was implemented within the software framework RoSe developed at FKIE [31]. This framework consists of roughly 30 modules which exchange information over a UDP-based communication infrastructure. The RoSe framework is specifically designed to allow for the easy assembly of multi-robot applications, which extensively use wireless ad-hoc communication. However, for the tracking experiments, we only required two modules on a single robot:

1) A visual tracking module, which captures the images and employs the tracking algorithm (VOCUS or Camshift) for tracking a single person within the image. Based on the pixel location of the person computed by the vision-based tracker, the module computes a heading direction relative to the robot, steers the pantilt unit in order to center the person within the image and commands the robot to follow the person. This is achieved by continuously instructing the reactive collision avoidance component of the robot to drive to goal locations behind the moving person.

2) The collision avoidance component of the robot. It is specifically designed for the task of following moving persons based on motion tracking information. It does so by applying an expansive spaces tree algorithm, which carries out a search for admissible paths in time and space, based on information about static obstacles provided by a laser range scanner, as well as motion information, i.e. position and velocity vectors of moving obstacles and the person being followed, provided by the external tracking component [32].

We performed two series of experiments with this system within the hallways of the FKIE building – an outline of the floorplan is shown in Figure 2, right. The first series of experiments illustrates the benefit of the VOCUS tracker for the actual people tracking task; the second series evaluates the robustness of the image-based tracker using the VOCUS system, compared to simpler feature-based techniques like Camshift.

Both series were performed during normal working hours with people walking around. The lighting conditions varied strongly during the experiments: some areas show natural daylight (see Fig. 2, right), others artificial light. In some parts, the light was switched off resulting in rather poorly illuminated areas. These conditions resulted in several images with very poor quality (cf. Fig. 5). Furthermore, after quick camera movements the camera was out of focus for some frames and capturing images was sometimes delayed resulting in large changes between consecutive frames.

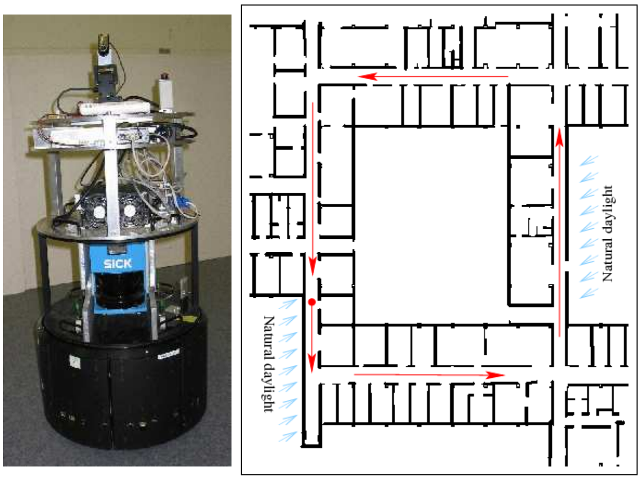

Fig. 2. Left: the RWI B21 robot *Blücher* used for the experiment. The images were taken using the small pantilt mounted webcam on top of the robot. Right: An outline of the environment used for the experiments. The robot tracked the person through the indicated round trip tour (red arrows) and encountered different lighting conditions on its path. The start and end location is marked with a small red circle.

A. *Autonomous Person Tracking*

In the first series of experiments, the robot followed a person autonomously through the hallways (red arrows in Fig. 2, right). We performed 4 runs with 2 different persons and 3 different kinds of clothing. Initialization of the target was done with user interaction by marking the person in the first frame. After that, the robot estimated the position of the person in each frame and drove autonomously into the direction of the estimated target state. The camera was controlled to center the target in the frame.

To evaluate the tracking, we counted the number of *detections* manually. A detection occurs if the center of the target state was on the person[1]. The results are shown in Tab. I. Images in which the target was not visible were not considered for the detection rate but are shown in Tab. I. In three of the runs, the detection rate was about 80%. In the 2nd run, the detection rate is considerably lower. The reason was that the center of gravity of the particle cloud was in many frames next to the target (cf. Fig. 5, right).

B. *Comparison with Camshift*

Most similar to the here presented VOCUS tracking are color-based trackers such as trackers based on the MeanShift algorithm [16]. One well-known modification is the Camshift algorithm [17] that is able to adapt dynamically to the target it is tracking[2]. It is a statistical method of finding the peak of a probability distribution, usually obtained with a color histogram. In the 2nd series of experiments, we used Camshift as benchmarking system for our approach.

[1] This is an approximation which is actually too optimistic since the region might include a part of the background and still have its center on the region. It is reasonable here anyway since the center is the point the robot uses as target direction.

[2] Camshift is publically available from the OpenCV library: http://opencvlibrary.sourceforge.net/

	# Frames	detections [%]	# frames without target
1	1918	81	1
2	1486	58	37
3	1202	87	8
4	559	80	79
Average	1291	77	31

TABLE I

VOCUS TRACKING IN ONLINE EXPERIMENTS

	# Frames	correct detections [%]			
		VOCUS	Cam (HSV)	Cam (RG)	Cam (Lab)
1	1477	79	51	88	39
2	1158	96	53	62	54
3	1596	65	5	28	50
4	1392	54	13	1	10
5	1519	71	46	47	46
	Average	73	33	45	40

TABLE II

COMPARISON OF VOCUS AND CAMSHIFT TRACKING. CAMSHIFT IS INVESTIGATED FOR DIFFERENT COLOR SPACES (HSV, RG, LAB). THE ROWS SHOW THE RESULTS FOR THE 5 PERSONS IN FIG. 3.

Although the Camshift algorithm has shown good results in other applications, it is only of limited use for a flexible online tracker. Usually, it is necessary to adapt the parameters of the algorithm for each object to obtain good results. While this may be acceptable for some applications like face tracking in which each face has a similar hue value, it is difficult for targets like persons which vary strongly in appearance due to different clothing. Since our VOCUS tracker is applicable to different objects without adapting parameters, we used the Camshift algorithm with the standard parameter set of the OpenCV implementation for all test sequences to make the approaches comparable. The Camshift usually uses the HSV color space. Additionally to this implementation, we used it with two other color spaces: RG chromaticity space and Lab space.

To be able to compare the approaches on the same data, several image sequences were acquired by teleoperating the robot and processed offline. We tested 5 different runs, each covering one circle in our environment (approx. 160 m per run). Each run was performed with a different person as target, with different clothing (cf. Fig. 3). The runs consisted of 1000–1600 frames each. Tab. III shows the initial feature vectors w^* that were learned from the frames in Fig. 3. The results are displayed in Tab. II. All approaches clearly have difficulties with the challenging conditions, mainly resulting from the strong changes in illumination. In most cases, the VOCUS tracker performed best, with an average detection rate of 73%. The Camshift approaches perform considerably worse (33, 45 and 40%). All approaches had most difficulties with person 4. This is partly due to the white shirt which is similar to the color of the walls. For all approaches it turned out that the clothing of the person made a strong difference in performance: the larger the contrast and difference to the background, the easier the tracking.

Feature	1)	2)	3)	4)	5)
intensity on-off	0.14	0.14	0.19	0.62	0.44
intensity off-on	2.48	4.06	4.36	1.95	4.30
orientation 0°	1.2	1.58	2.00	2.56	1.86
orientation 45°	1.66	2.35	1.25	1.75	1.69
orientation 90°	1.08	1.90	1.40	1.65	1.81
orientation 135°	1.27	1.59	1.21	1.52	2.07
color green	0.35	2.62	0.90	0.75	1.10
color blue	5.55	2.68	3.24	3.02	6.02
color red	1.53	31.40	3.41	1.67	6.88
color yellow	1.48	3.71	0.80	1.54	1.41
intensity	1.26	1.86	2.18	1.14	2.85
orientation	1.21	1.81	1.38	1.80	1.84
color	1.93	10.44	1.60	1.81	2.61

TABLE III

FEATURE VECTORS $\mathbf{w}^*$ THAT ARE LEARNED FOR THE TARGET PERSONS IN FIG. 3 (THE COLUMNS CORRESPOND TO THE IMAGES).

V. CONCLUSION

In this paper, we have presented a cognitive approach for person tracking from a mobile platform. The appearance of an object of interest is learned from an initially provided target region and the resulting target feature vector is used to search for the target in subsequent frames. Advantages of the system are that it uses several feature channels in parallel, that it considers not only the target appearance but also the appearance of the background, and that it is quickly adaptable to a new target without a time-consuming learning phase. Furthermore, it is capable to work on a mobile platform since it works in real-time, does not rely on a static background, and copes with varying illumination conditions.

We obtained promising first results in different settings. However, the task of person tracking in natural conditions is very challenging and we just scratched the surface of the problem. Although our image sequences are more difficult than most of the data used in research groups for similar tasks, they show by far not the most difficult settings. Persons with similar clothing to the background, bright sunlight, and crowded environments in which the person is temporarily occluded would make the problem worse. We will investigate such settings in future work.

There are several ways the current approach could be improved. Currently, we learn target appearance from a single frame. While this works reasonably well in many cases, it will fail if the environment changes strongly. Learning target appearance online from several frames and adapting the feature vector to new conditions is subject to future work. We also plan to integrate additional features, e.g. motion cues, into the tracking system.

REFERENCES

[1] M. Montemerlo, S. Thrun, and W. Whittaker, “Conditional particle filters for simultaneous mobile robot localization and people-tracking,” in *Int'l Conference on Robotics and Automation (ICRA)*, 2002.

[2] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, “People tracking with mobile robots using sample-based joint probabilistic data association filters,” International Journal of Robotics Research, 22(2), 2003.

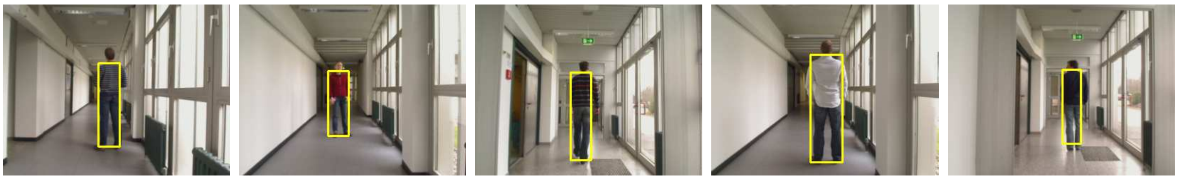

Fig. 3. Initial frames and initial target regions $\mathbf{R}^*$ (yellow rectangles) used to learn the appearance of the 5 persons.

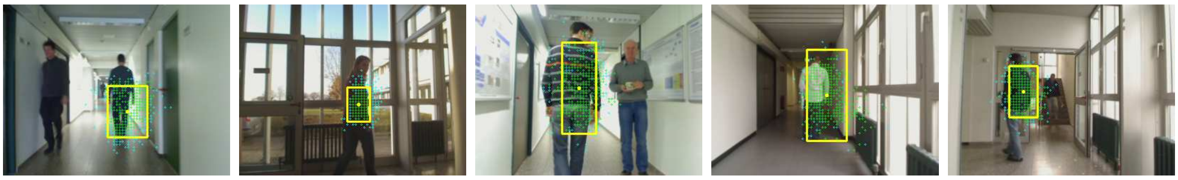

Fig. 4. Successful tracking. Green points: particles that matched to target; cyan points: particles that didn't match. Rectangles show estimated target state.

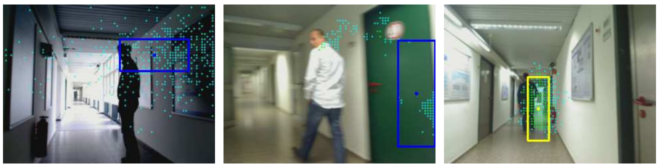

Fig. 5. Failed tracking. The points denote the particles (green points: particles that matched to the target, cyan points: particles that did not match). Rectangles denote the estimated target state (blue: less than 50% of particles match, otherwise yellow).

[3] K. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. of IEEE Int'l Conference on Robotics and Automation (ICRA'08)*, Pasadena, USA, 2008.

[4] G. Taylor and L. Kleeman, "A multiple hypothesis walking person tracker with switched dynamic model," in *Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[5] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[6] D. Schulz, "A probabilistic exemplar approach to combine laser and vision for person tracking," In Proc. of the International Conference on Robotics Science and Systems (RSS 2006), 2006.

[7] S. Frintrop, "VOCUS: a visual attention system for object detection and goal-directed search," Ph.D. dissertation, University of Bonn, Germany, July 2005, published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.

[8] A. M. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980.

[9] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Int. J. of Computer Vision (IJCV)*, vol. 29, no. 1, pp. 5–28, 1998.

[10] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP – Image Understanding*, vol. 59, no. 1, pp. 94–115, 1994.

[11] C. Breglera, J. Malik, and K. Pullen, "Twist based acquisition and tracking of animal and human kinematics," *International Journal of Computer Vision (IJCV)*, 2004.

[12] R. Urtasun, D. J. Fleet, and P. Fua, "Temporal motion models for monocular and multiview 3d human body tracking," *Computer Vision and Image Understanding (CVIU), special issue Modeling People*, 2006.

[13] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 199–223, 2003.

[14] K. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette across time part II: Applications to human modeling and markerless motion," *International Journal of Computer Vision (IJCV)*, 2005.

[15] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, 2002.

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," Proc. Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, 2000.

[17] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, 1998.

[18] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," Proceedings of the 7th European Conference on Computer Vision (ECCV) London, UK, Springer-Verlag, 2002.

[19] J. M. Wolfe, "Guided search 2.0: A revised model of visual search," *Psychonomic Bulletin and Review*, vol. 1, no. 2, pp. 202–238, 1994.

[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[21] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. of the 5th Int'l Conf. on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

[22] S. May, M. Klodt, and E. Rome, "GPU-accelerated Affordance Cueing based on Visual Attention," in *Proc. of Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 3385–3390.

[23] M. Björkman and J.-O. Eklundh, "Vision in the real world: Finding, attending and recognizing objects," *Int'l Journal of Imaging Systems and Technology*, vol. 16, no. 2, pp. 189–208, 2007.

[24] V. Navalpakkam, J. Rebesco, and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[25] S. Frintrop and P. Jensfelt, "Active gaze control for attentional visual SLAM," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA'08)*, 2008.

[26] M. Veyret and E. Maisel, "Attention-based target tracking for an augmented reality application," Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, 2006.

[27] M. Kessel, "Aufmerksamkeitsbasiertes Objekt-Tracking," Master's thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2008.

[28] S. Frintrop and M. Kessel, "Most salient region tracking," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA'09)*, Kobe, Japan, 2009.

[29] ——, "Cognitive data association for visual person tracking," in *Proc. of the 1st IEEE Workshop on Human Detection from Mobile Platforms (HDMP) at ICRA*, Pasadena, CA, May 2008.

[30] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, May 2004.

[31] A. Tiderko, T. Bachran, F. Hoeller, D. Schulz, and S. F. E., "RoSe – a framework for multicast communication via unreliable networks in multi-robot systems," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1017–1026, 2008.

[32] F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider, "Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps," in *Proc. of the International Conference on Robots and Systems (IROS)*. IEEE, 2007, pp. 1260–1265.