Dynamic Integration of Generalized Cues for Person Tracking

Kai Nickel and Rainer Stiefelhagen

Universität Karlsruhe (TH), InterACT
Am Fasanengarten 5, 76131 Karlsruhe, Germany

Abstract. We present an approach for the dynamic combination of multiple cues in a particle filter-based tracking framework. The proposed algorithm is based on a combination of democratic integration and layered sampling. It is capable of dealing with deficiencies of single features as well as partial occlusion using the very same dynamic fusion mechanism. A set of simple but fast cues is defined, which allow us to cope with limited computational resources. The system is capable of automatic track initialization by means of a dedicated attention tracker permanently scanning the surroundings.

1 Introduction

Visual person tracking is a basic prerequisite for applications in fields like surveillance, multimodal man-machine interaction or smart spaces. Our envisioned scenario is that of an autonomous robot with limited computational resources operating in a common space together with its users. The tracking range varies from close distance, where the portrait of the user spans the entire camera image, to far distance, where the entire body is embedded in the scene. In order to tackle the problem, we present a multi-cue integration scheme within the framework of particle filter-based tracking. It is capable of dealing with deficiencies of single features as well as partial occlusion by means of the very same dynamic fusion mechanism. A set of simple but fast cues is defined, allowing to cope with limited on-board resources.

The choice of cues is a crucial design criterion for a tracking system. In real-world applications, each single cue is likely to fail in certain situations such as occlusion or background clutter. Thus, a dynamic integration mechanism is needed to smooth over a temporary weakness of certain cues as long as there are other cues that still support the track. In [1], Triesch and Von Der Malsburg introduced the concept of *democratic integration* that weights the influence of the cues according to their agreement with the joint hypothesis. The competing cues in [1] were based on different feature types such as color, motion, and shape. In this paper, we use the principle of democratic integration in a way that also includes the competition between different regions of the target object. We show that this allows us to deal with deficiencies of single feature types as well as with partial occlusion using one joint integration mechanism.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 514–526, 2008.

The combination of democratic integration and particle filters has been approached before by Spengler and Schiele [2]. In their work, however, the integration weights were held constant, thus falling short behind the real power of democratic integration. This has also been pointed out by Shen et al. [3], who did provide a cue quality criterion for dynamic weight adaptation. This criterion is formulated as the distance of the tracking hypothesis based on all cues and the hypothesis based on the cue alone. The problem with this formulation is that, due to resampling, the proposal distribution is generally strongly biased toward the final hypothesis. Thus, even cues with uniformly mediocre scores tend to agree well with the joint mean of the particle set. We therefore propose a new quality criterion based on weighted MSE that prefers cues which actually focus their probability mass around the joint hypothesis.

Democratic integration combines cues in the form of a weighted sum. In a particle filter framework, this means that all cues have to be evaluated simultaneously for all particles. As pointed out by Pérez et al. [4], this can be alleviated by *layered sampling*, if the cues are ordered from coarse to fine. In the proposed algorithm, we therefore combine two-stage layered sampling with democratic integration on each stage to increase efficiency by reducing the required number of particles.

For each object to be tracked, we employ one dedicated Condensation-like tracker [5]. By using separate trackers instead of one single tracker running in a joint state space, we accept the disadvantage of potentially not being able to find the global optimum. On the other hand, however, we thereby avoid the exponential increase in complexity that typically prevents the use of particle filters in high-dimensional state spaces. There are a number of approaches dealing with this problem, such as Partitioned Sampling [6], Trans-dimensional MCMC [7], or the Hybrid Joint-Separable formulation [8]. Although these approximations reduce the complexity of joint state space tracking significantly, they still require noticeably more computational power than the separate tracker approach.

The remainder of this paper is organized as follows: In section 2, we briefly describe the concept of particle filters and layered sampling. In section 3 we present our multi-cue integration scheme, which is the main contribution of this paper. It is followed, in section 4, by the definition of the cues that we actually use in the live tracking system. In section 5, the multi-person tracking logic including automatic track initialization and termination is described. Finally, section 6 shows the experiments and results.

2 Particle Filter-Based Tracking

Particle filters represent a generally unknown probability density function by a set of random samples $\mathbf{s}_t^{(1..n)}$ and associated weights $\pi_t^{(1..n)}$ with $\sum \pi_t^{(i)} = 1$. In one of the simplest cases, the Condensation algorithm [5], the evolution of the particle set is a two-stage process which is guided by the observation and the state evolution model:

1. The prediction step (including resampling): randomly draw n new particles from the old set with a likelihood proportional to the particle weights. Propagate the new particles by applying the state evolution model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$.
2. The measurement step: adjust the weights of the new particles with respect to the current observation $\mathbf{z}_t$: $\pi_t^{(i)} \propto p(\mathbf{z}_t|\mathbf{s}_t^{(i)})$.

The final tracking hypothesis for the current time instance $\hat{\mathbf{s}}_t$ can be obtained from the sample set as

$$\hat{\mathbf{s}}_t = \sum_{i=0..n} \pi_t^{(i)} \mathbf{s}_t^{(i)} \tag{1}$$

2.1 Layered Sampling

Assuming that $\mathbf{z}$ is made up of M conditionally independent measurement sources, i.e. different cues, the observation likelihood of a particle $\mathbf{s}$ can be factorized as follows[1]:

$$p(\mathbf{z}|\mathbf{s}) = \prod_{m=1..M} p(\mathbf{z}^m|\mathbf{s}) \tag{2}$$

According to [4], the state evolution can then be decomposed into M successive intermediate steps:

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \int p_M(\mathbf{s}_t|\mathbf{s}^{M-1}) \cdots p_1(\mathbf{s}^1|\mathbf{s}_{t-1}) d\mathbf{s}^1 \cdots d\mathbf{s}^{M-1} \tag{3}$$

where $\mathbf{s}^1 \cdots \mathbf{s}^{M-1}$ are auxiliary state vectors[2]. In case of a Gaussian evolution model, this corresponds to a fragmentation into M successive steps with lower variances. Then, [4] make the approximation that the likelihood for the m-th cue $p(\mathbf{z}^m|\mathbf{s})$ can be incorporated after applying the m-th state evolution model $p_m(\mathbf{s}^m|\mathbf{s}^{m-1})$. This leads to a layered sampling strategy, where at the m-th stage new samples are simulated from a Monte Carlo approximation of the distribution $p_m(\mathbf{s}^m|\mathbf{s}^{m-1})\pi^{m-1}$ with an associated importance weight $\pi^m \propto p(\mathbf{z}^m|\mathbf{s}^m)$. As [4] point out, the benefit of layered sampling arises in cases where the cues can be ordered from coarse to fine, e.g. the first cue produces a reliable but rough estimation for the state, while the second cue produces a sharp and peaky estimation. Then, the layered sampling approach will effectively guide the search in the state space, with each stage refining the result from the previous stage. We will apply layered sampling in section 5 in combination with the multi-cue integration scheme described in the following.

3 Dynamic Multi-cue Integration

In the Bayesian tracking formulation used in this work, cues have the function of scoring the match between a state vector $\mathbf{s}$ and the observation $\mathbf{z}$. A joint score combining the cues from the set of all cues C can be formulated as a weighted sum

[1] The time index t is omitted for the sake of brevity wherever possible.
[2] We omit the according formula for splitting the proposal distribution, because in Condensation, the proposal distribution is identical to the evolution model.

$$p(\mathbf{z}|\mathbf{s}) = \sum_{c \in C} r_c p_c(\mathbf{z}|\mathbf{s}), \tag{4}$$

where $p_c(\mathbf{z}|\mathbf{s})$ is the the single-cue observation model, and r_c is the mixture weight for cue c, with $\sum_c r_c = 1$.

Democratic integration [1] is a mechanism to dynamically adjust the mixture weights r_c, termed reliabilities, with respect to the agreement of the single cue c with the joint result. For each cue, a quality measure q_c is defined that quantifies the agreement, with values close to zero indicating little agreement and values close to one indicating good agreement. The reliabilities are updated after each frame by a leaky integrator using the normalized qualities:

$$r_c^{t+1} = (1 - \tau) r_c^t + \tau \frac{q_c}{\sum_c q_c} \tag{5}$$

with the parameter τ controlling the speed of adaptation.

3.1 Cue Quality Measure

In the original paper [1], tracking is implemented as an exhaustive search over a support map, and the quality measure is defined over a single cue's support map. In [3], a different quality measure dedicated to particle filters is proposed: Based on the current particle set $\mathbf{s}^{(1..n)}$ and an auxiliary set of weights $\pi_c^{(1..n)} \propto p_c(\mathbf{z}|\mathbf{s}^{(1..n)})$, a tracking hypothesis $\hat{\mathbf{s}}_c$ is generated according to eq. 1 and compared to the joint hypothesis $\hat{\mathbf{s}}$. The L_2-norm distance $|\hat{\mathbf{s}}_c$ - $\hat{\mathbf{s}}|_2$ is normalized by means of a sigmoid function and then taken as quality measure.

Although this formulation looks straightforward, there is a problem associated with it: Imagine the common situation where a cue finds little or no support at all, and therefore assigns uniform likelihood values to all of the particles. Let's assume further that the state of the target has not changed for a while, so that in consequence, due to resampling, the particle distribution is equally spread around the actual state. In this case, the cue-based hypothesis $\hat{\mathbf{s}}_c$ will be close to $\hat{\mathbf{s}}$ resulting in a high quality value q_c despite the fact that the cue is actually not at all able to locate the target.

To address this problem, we need a quality measure that quantifies how well the probability mass agglomerates around the joint hypothesis $\hat{\mathbf{s}}$. The inverse mean-square error $(\sum_i \pi_c^{(i)} |\mathbf{s}^{(i)} - \hat{\mathbf{s}}|_2^2)^{-1}$ of the particle set weighted with the respective cue's weights π_c meets this requirement, but is dependent on the actual location of the particles. We eliminate this dependency by relating the cue's MSE to the MSE of a hypothetical baseline cue which assigns uniform weights $\frac{1}{n}$ to each particle. Because a good cue is not only supposed to converge to the target location but also to assign high values to the target, we multiply the term with the cue's non-normalized response at the joint hypothesis $p_c(\mathbf{z}|\hat{\mathbf{s}})$. Thus, we come to the following formulation for a universal cue quality measure in the context of particle-filter based tracking:

$$q_c = \frac{\sum_{i=1..n} \frac{1}{n} |\mathbf{s}^{(i)} - \hat{\mathbf{s}}|^\lambda}{\sum_{i=1..n} \pi_c^{(i)} |\mathbf{s}^{(i)} - \hat{\mathbf{s}}|^\lambda} p_c(\mathbf{z}|\hat{\mathbf{s}}) \tag{6}$$

The exponent $\lambda > 0$ can be used to tweak the volatility of the quality measure: high values of λ emphasize the quality difference between cues whereas low values produce more similar qualities for all cues.

3.2 Generalized Cue Competition

In order to allow for a fruitful combination, the set of cues should be orthogonal in the sense that different cues tend to fail under different circumstances. One way to reduce the chances of co-occurrence of failure is to use different cue-specific feature transformations $\mathcal{F}(\mathbf{z})$ like motion, color, or shape. Failure of one feature can thus more likely be compensated by other features.

$$p_c(\mathbf{z}|\mathbf{s}) = p_c(\mathcal{F}(\mathbf{z})|\mathbf{s}) \tag{7}$$

The other option to generate orthogonal cues is to use different state model transformations $\mathcal{A}(\mathbf{s})$:

$$p_c(\mathbf{z}|\mathbf{s}) = p_c(\mathbf{z}|\mathcal{A}(\mathbf{s})) \tag{8}$$

This is motivated by the fact that cues relying on certain aspects of the state vector may still be used while other aspects of the state are not observable. In our implementation, $\mathcal{A}(\mathbf{s})$ represents a certain projection from state space to image space, i.e. a certain image sub-region of the target. This is useful in a situation, where due to partial occlusion one region of the target object can be observed, while another region cannot.

In this work, we aim at combining the advantages of both strategies, i.e. dynamically combining cues that are based on different feature types as well as dynamically weighting cues that focus on different regions of the target but are based on the same feature type. Therefore, we use a generalized definition of the cues $c = (\mathcal{F}, \mathcal{A})$ that comprises different feature types $\mathcal{F}(\mathbf{z})$ and different state transformations $\mathcal{A}(\mathbf{s})$:

$$p_c(\mathbf{z}|\mathbf{s}) = p_{\mathcal{F},\mathcal{A}}(\mathcal{F}(\mathbf{z})|\mathcal{A}(\mathbf{s})), \tag{9}$$

All cues in this unified set will then compete equally against each other, guided by the very same integration mechanism. Thus, the self-organizing capabilities of democratic integration can be used to automatically select the specific feature types as well as the specific regions of the target that are most suitable in the current situation.

3.3 Cue Model Adaptation

Certain cues, such as color models or templates, allow for online adaptation of their internal parameters to better match the current target appearance. In [1], this adaptation is described as a continuous update process with a fixed time constant τ_c:

$$P_c^{t+1} = (1 - \tau_c)P_c^t + \tau_c \hat{P}_c, \tag{10}$$

with P_c being the internal parameters of cue c, and $\hat{P}_c$ being new parameters acquired from the image region given by the joint hypothesis $\hat{\mathbf{s}}$.

One of the issues with adaptation is due to the fact that after an update step, the cue is not guaranteed to perform better than before. Although the update step always results in a higher score for the prototype region at $\hat{\mathbf{s}}$, it can happen that the updated model produces higher scores also for other regions than the correct one. This actually reduces the cue's discriminative power and, in consequence, its reliability r_c. We therefore propose the following test to be carried out before accepting an update:

1. Calculate q'_c (eq. 6) using the new parameters $\hat{P}_c$
2. Perform the update step (eq. 10) only if $q'_c > q_c$

4 Fast Cues for 3D Person Tracking

In the targeted application, one or more people are to be tracked in the vicinity of an autonomous robot featuring a calibrated stereo camera. As the on-board computational resources are strictly limited, cues have to be found that rely on features that can be evaluated rapidly. Our proposed cues are based on the following well-known feature types: difference image, color histogram back-projection, Haar-feature cascades and stereo correlation.

As motivated in section 3.2, we use different transformations of the state vector in order to handle partial occlusion: some cues focus on the human head region only, whereas other cues concentrate on the torso and legs region respectively. These regions are determined using the "3-box model" of the human body depicted in Fig. 1. The real-world extensions of the 3 cuboids are geared to model an average human being; their relative positions depend on the height of the head above the ground plane.

By combining the feature types motion, color and stereo with the 3 different body parts, and by using 4 different detectors, we obtain a total number of 13 cues that will be described in the following. Fig. 2 shows the different feature types as a snapshot from a test sequence.

In the following, we will use $\mathcal{F}(\mathbf{z})$ to denote a feature map, i.e. an image in which the intensity of a pixel is proportional to the presence of a feature, such as color or motion. An image region corresponding to a state vector $\mathbf{s}$ will be denoted as $\mathcal{A}(\mathbf{s})$ (see Fig. 1), $|\mathcal{A}(\mathbf{s})|$ is the size of the region, and $\sum_{\mathcal{A}(\mathbf{s})} \mathcal{F}(\mathbf{z})$ is the sum of pixel values of $\mathcal{F}(\mathbf{z})$ inside region $\mathcal{A}(\mathbf{s})$. All regions in our system are rectilinear bounding boxes, so the sum can be calculated efficiently by means of 4 table lookups in the integral image [9].

4.1 Motion Cues

The difference image $\mathcal{M}(\mathbf{z})$ is generated by pixel-wise thresholding the absolute difference of the current frame's and the previous frame's intensity images. For a moving object, we can expect high values of $\mathcal{M}(\mathbf{z})$ in the region $\mathcal{A}(\mathbf{s})$ around object's current location $\mathbf{s}$. The motion cue's observation likelihood is given as:

$$p_{\mathcal{M},\mathcal{A}}(\mathbf{z}|\mathbf{s}) = \frac{\sum_{\mathcal{A}(\mathbf{s})} \mathcal{M}(\mathbf{z})}{|\mathcal{A}(\mathbf{s})|} \cdot \frac{\sum_{\mathcal{A}(\mathbf{s})} \mathcal{M}(\mathbf{z})}{\sum \mathcal{M}(\mathbf{z})} \tag{11}$$

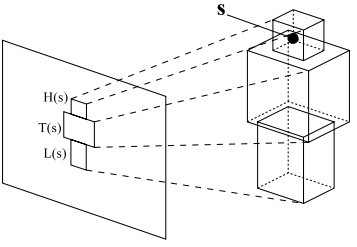
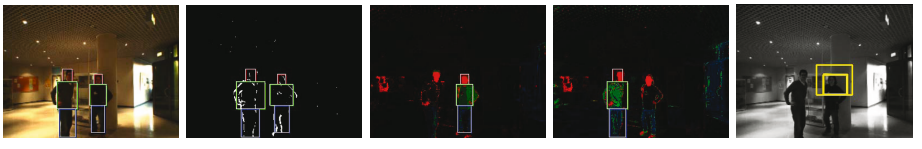


Fig. 1. The 3-box model of the human body: the state vector **s** is transformed into the image space as the projection of a cuboid representing either the head, torso, or leg region. The projection of the cuboid is approximated by a rectilinear bounding box.



camera image | motion | color person 1 | color person 2 | detectors

Fig. 2. Snapshot from a test sequence showing the different feature types. In this visualization, the color support maps for head, torso and legs of the respective person are merged into the RGB-channels of the image. The tracking result is superimposed.

The left factor seeks to maximize the amount of foreground within the region. The right factor seeks to cover all foreground pixels in the image. It prevents the motion cue from preferring tiny regions filled with motion, while ignoring the rest.

We employ 3 motion cues, termed M-H, M-T and M-L, dedicated to either the head, torso or legs region as depicted in Fig. 1. We rely on the ability of the integration mechanism (see section 3) to automatically cancel the influence of the motion cues in case of camera motion. This is justified by the fact that the agreement of the motion cues with the final tracking hypothesis will drop whenever large portions of the image exceed the threshold.

4.2 Color Cues

We employ three adaptive color cues C-H, C-T, C-L for the three body regions. For each of the cues, we use a 3-dimensional histogram with 16 bins per channel in RGB color space that automatically adapts to the target region using the mechanism described in section 3.3. A second histogram is built from the entire image; it acts as a model for the background color distribution. The quotient histogram of the target histogram and the background histogram is back-projected and forms the support map $\mathcal{C}(\mathbf{z})$ for a color cue. The observation likelihood is given analogous to eq. 11 as:

$$p_{\mathcal{C},\mathcal{A}}(\mathbf{z}|\mathbf{s}) = \frac{\sum_{\mathcal{A}(\mathbf{s})} \mathcal{C}(\mathbf{z})}{|\mathcal{A}(\mathbf{s})|} \cdot \frac{\sum_{\mathcal{A}(\mathbf{s})} \mathcal{C}(\mathbf{z})}{\sum \mathcal{C}(\mathbf{z})} \tag{12}$$

4.3 Detector Cues

For each particle, the head region $\mathcal{A}(\mathbf{s})$ is projected to the image plane, and the bounding box of the projection is being classified with a single run of the detector proposed by [9]. The detectors are organized stages that need to be passed one by one in order to produce a positive response. The ratio $m(\mathcal{A}(\mathbf{s})) = (\frac{\text{stages passed}}{\text{stages total}})^{\omega}$ can be interpreted as a confidence value for the detection, with the exponent ω controlling the steepness of decay for each stage that is not being passed.

In order to smooth the scores of nearby particles, we define the score of a particle $\mathbf{s}$ as the highest overlap between its region $\mathcal{A}(\mathbf{s})$ and all the positively classified regions $\mathcal{A}' \in \{\mathcal{A}(\mathbf{s}^{(i)}) | \mathcal{A}(\mathbf{s}^{(i)}) \text{is face}\}_{i=1..n}$ by any of the other particles:

$$p_{\mathcal{D},\mathcal{A}}(\mathbf{z}|\mathbf{s}) = \max_{\mathcal{A}'} m(\mathcal{A}') \cdot d(\mathcal{A}', \mathcal{A}(\mathbf{s})), \tag{13}$$

with d being a distance metric based on rectangle overlap.

We use four detector cues in total: one for frontal faces (D-F), one for left (D-L) and one for right (D-R) profile faces, and one for upper bodies (D-U). Implementation and training of the detectors is based on [10,11] as provided by the OpenCV library.

4.4 Stereo Correlation Cues

In traditional stereo processing [12], a dense disparity map is generated by exhaustive area correlation followed by several post-filtering steps. Apart from the computational effort of generating a dense disparity map, there is another, more fundamental problem, namely the choice of the size of the area correlation window. If a windows is too large, it smoothes over fine details, if it is too small, it tends to produce noisy results. In our approach, we can avoid these issues: we use the entire target region $\mathcal{A}(\mathbf{s})$ as correlation window and search for optimal correlation along the epipolar lines. The adaptive correlation window is thus as large as possible and as small as necessary given the current size of the target.

The response of the stereo cue is given by the distance of the discovered disparity $\hat{d}(\mathcal{A}(\mathbf{s}))$ and the hypothesized disparity $d(\mathcal{A}(\mathbf{s}))$:

$$p_{\mathcal{S},\mathcal{A}}(\mathbf{z}|\mathbf{s}) = \left(1 + |\hat{d}(\mathcal{A}(\mathbf{s})) - d(\mathcal{A}(\mathbf{s}))|^{\kappa}\right)^{-1}, \tag{14}$$

with κ being a parameter to control the volatility of the cue. The complexity of the local search for the disparity $\hat{d}(\mathcal{A}(\mathbf{s}))$ is scale-invariant because it can be implemented efficiently by means of integral images, as proposed by [13] for dense disparity calculation. We employ 3 stereo cues, one for the head (S-H), torso (S-T), and legs (S-L).

5 Multi-person Tracking Logic

As motivated in the introduction, we run one dedicated particle filter for each person to be tracked. The state space consists of the location and velocity of the person's head centroid in 3-dimensional space: $\mathbf{s}^{(i)} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$. The state evolution $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is implemented as a 1st-order motion model with additive Gaussian noise on the velocity components.

5.1 Democratic Integration and Layered Sampling

Multi-cue integration as described by eq. 4 is suitable for all kinds of cues that are *optional* for the target, which means that the target may or may not have the property implied by the cue at the moment. There are, however, cues that are indispensable as track foundation and therefore must not be ruled out by the fusion mechanism. In our application, this applies to the stereo cues: a track should not be able to exist if it is not supported by at least one of the stereo cues as these represent strict geometrical constraints. One way of ensuring this would be to multiply the response of the stereo cues with the response of the regular cues. A more efficient way is layered sampling as described in section 2.1. We use it to evaluate the stereo cues $C_S \subset C$ before the regular cues $C_R \subset C$, as shown in Fig. 3. By evaluating the mandatory stereo cues first, followed by a resampling step, the resulting particle set $\mathbf{s}_t^{1,(1..n)}$ clusters only in those regions of the state space that are well supported by the stereo cues. The particles on the second stage can now more efficiently evaluate the regular cues.

1st layer:

- resample $\mathbf{s}_{t-1}^{(1..n)}$ wrt. $\pi_{t-1}^{(1..n)}$
- propagate with partial evolution model (cf. eq. 3)
  $\mathbf{s}_t^{1,(1..n)} \longleftarrow p_1(\mathbf{s}_t^{1,(i)}|\mathbf{s}_{t-1}^{(i)})$
- evaluate stereo cues: $\pi_t^{1,(i)} \propto \sum_{c\in C_S} r_c p_c(\mathbf{z}|\mathbf{s}_t^{1,(i)})$
- apply collision penalty: $\pi_t^{1,(i)} \longleftarrow \pi_t^{1,(i)} - v(\mathbf{s}_t^{1,(i)})$

2nd layer:

- resample $\mathbf{s}_t^{1,(1..n)}$ wrt. $\pi_t^{1,(1..n)}$
- propagate with partial evolution model (cf. eq. 3)
  $\mathbf{s}_t^{(1..n)} \longleftarrow p_2(\mathbf{s}_t^{(i)}|\mathbf{s}_t^{1,(i)})$
- evaluate regular cues: $\pi_t^{(i)} \propto \sum_{c\in C_R} r_c p_c(\mathbf{z}|\mathbf{s}_t^{(i)})$

Dem. integration:

- calculate track hypothesis $\hat{\mathbf{s}}_t = \sum_i \pi_t^{(i)} \mathbf{s}_t^{(i)}$
- update reliabilities (cf. eqs. 5 and 6)
  $r_{c\in C_S} \longleftarrow \hat{\mathbf{s}}_t, \mathbf{s}_t^{1,(1..n)}, \pi_t^{1,(1..n)}$
  $r_{c\in C_R} \longleftarrow \hat{\mathbf{s}}_t, \mathbf{s}_t^{(1..n)}, \pi_t^{(1..n)}$

Fig. 3. Two-stage layered sampling algorithm with democratic cue integration

Apart from the geometrical constraints implied by the stereo cues, there is another strict constraint, namely the collision penalty, which is enforced in the 1st layer of the algorithm in Fig. 3. The function $v(\mathbf{s})$ penalizes particles that are close to those tracks with a higher track quality than the current track (see following section). Thereby, we guarantee mutual exclusion of tracks.

5.2 Automatic Track Initialization

The question of when to spawn a new tracker and when to terminate a tracker that has lost its target is of high importance, and can become more difficult than the actual tracking problem. We define the quality measure for a tracker to be the joint response from both stereo and regular cues at the tracker's hypothesis $\hat{\mathbf{s}}$:

$$Q(\hat{\mathbf{s}}) = \sum_{c \in C_S} r_c p_c(\mathbf{z}|\hat{\mathbf{s}}) \cdot \sum_{c \in C_R} r_c p_c(\mathbf{z}|\hat{\mathbf{s}}) \tag{15}$$

The final quality measure Q is a result of temporal filtering with a time constant ν:

$$Q^{t+1} = (1 - \nu)Q^t + \nu Q(\hat{\mathbf{s}}) \tag{16}$$

Trackers falling below a certain threshold $Q < \Theta$ for a certain amount of time Γ will be discarded.

In order to discover potential targets, we employ an additional tracker termed *attention tracker*. The attention tracker permanently scans the state space, searching for promising regions. It is, however, repelled by existing tracks by means of the collision penalty $v(\mathbf{s})$. Unlike regular trackers, 50% of the attention tracker's particles are not propagated by means of the state evolution model, but are drawn randomly from the state space. This guarantees good coverage of the state space and still allows some clustering around interesting regions. As the attention tracker must remain general, its cues' parameters are not allowed to adapt. After each frame, the distribution of the attention tracker's particles is clustered with a k-means algorithm. If one of the clusters exceeds the threshold Θ, a new regular tracker is initialized at that location.

6 Experiments

We evaluated the algorithm on 11 test sequences, some of them including camera motion. The head's bounding box was manually labeled in 3 of the 15 frames per second to obtain the ground truth. In total, 2312 frames were labeled. From the 3D tracking output, a head-sized box was projected to the image and compared to the manually labeled box. If there was no overlap between the boxes, the frame was counted as a miss and a false positive. As the tracker was free to output 0, 1 or more tracks, the number of misses and false positives do not need to be identical.

Overall, the tracker showed solid performance throughout the experiments. Critical situations for track loss – although it occurred rarely – were periods in which the user rested virtually motionless either at far distance or in a turned-away position, so that in consequence the detectors did not respond. Then, the tracker had to rely solely on the automatically initialized color models, which were not always significant enough. Another issue were phantom tracks that were triggered by non-human motion or false detections. They were sometimes kept alive by the color models which adapted to the false positive region. In most

Table 1. Tracking results on the evaluation set

	misses	false pos.
Fixed reliabilities (baseline)	10.2%	8.1%
Dynamic integration (Shen et al.)	11.1%	8.8%
Dynamic integration (equation 6)	4.6%	4.6%

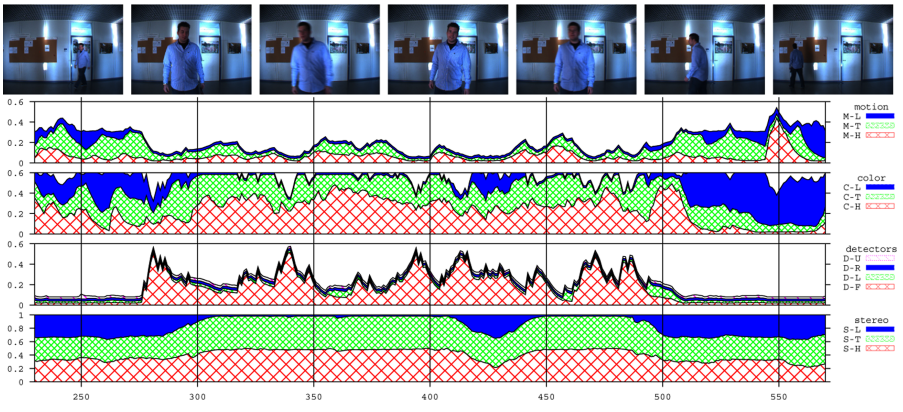


Fig. 4. Evolution of cue reliabilities in an example sequence. The three stereo cues constitute the first layer of the algorithm, their reliabilities sum up to 1. The remaining ten cues are used in layer 2 and sum up to 1 likewise. In the beginning of the interval, the subject approaches the camera. While he is walking (frame 250), the motion cues for legs and torso (M-L,M-T) contribute significantly to the track. At around frame 300, the subject's legs disappear, and in consequence the reliabilities of all leg-related cues (M-L, C-L, S-L) drop automatically. While the subject is standing in front of the camera (frames 300-500), the frontal face detection cue D-F and the the head color cue C-H dominate the track. The influence of the head color cue C-H drops dramatically, when the subject turns around (frame 520) and walks in front of the wooden pinboard, which has a skin-color like appearance.

cases, however, this could be avoided by the adaptation control mechanisms described in section 3.3.

Table 1 shows the results of the evaluation. The proposed algorithm was compared to a baseline system with static reliabilities, and to a system using the dynamic cue quality formulation by Shen et al. [3]. The proposed algorithm clearly outperforms the two other systems both in the number of misses and false positives. Figure 4 discusses the evolution of cue reliabilities for an example sequence.

6.1 Implementation Details

In the implementation, we made the following additions to the algorithm: The color cue for the head region (C-H) is expected to converge to general skin color;

its model is therefore shared among all trackers. An new box-type for the upper body detector was used; it comprises head and upper half of the torso. To avoid dominance, we limited the range for a cue's influence to $0.03 \leq r_c \leq 0.6$. We found, however, that these situations rarely occur. Boxes that get projected outside the visible range or that are clipped to less than 20% of their original size, are scored with a minimum score of 0.001. The approximate runtime of the algorithm was $30ms$ per frame for an empty scene, plus another $10ms$ per person being tracked. These values are based on an image size of 320×240 pixels, and a 2.4GHz Pentium CPU. The most important parameter values are given in Table 2.

Table 2. Parameters of the algorithm

# of particles per tracker	$n = 150$
Track threshold / timeout	$\Theta = 0.25$, $\Gamma = 2s$
Track quality time constant	$\nu = 0.33$
Cue reliability time constant	$\tau = 0.25$
Color update time constant	$\tau_c = 0.01$
Cue tweaking factors	$\lambda = 4, \kappa = 4, \omega = 10$

7 Conclusion

We have presented a new approach for dynamic cue combination in the framework of particle filter-based tracking. It combines the concepts of democratic integration and layered sampling and enables a generalized kind of competition among cues. With this method, cues based on different feature types compete directly with cues based on different target regions. In this way, the self-organizing capabilities of democratic integration can be fully exploited. In an experimental validation, the proposed new cue quality measure has been shown to improve the tracking performance significantly.

Acknowledgments

This work has been funded by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoid Robots".

References

1. Triesch, J., Malsburg, C.V.D.: Democratic integration: Self-organized integration of adaptive cues. Neural Comput. 13(9), 2049–2074 (2001)
2. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. Machine Vision and Applications 14, 50–58 (2003)
3. Shen, C., Hengel, A., Dick, A.: Probabilistic multiple cue integration for particle filter based tracking. In: International Conference on Digital Image Computing - Techniques and Applications, pp. 309–408 (2003)

4. Pérez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proceedings of the IEEE 92(3), 495–513 (2004)
5. Isard, M., Blake, A.: Condensation–conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
6. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. International Journal of Computer Vision 39(1), 57–71 (2000)
7. Smith, K., Gatica-Perez, D., Odobez, J.M.: Using particles to track varying numbers of interacting people. In: IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, pp. 962–969 (2005)
8. Lanz, O.: Approximate bayesian multibody tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9), 1436–1449 (2006)
9. Viola, P., Jones, M.: Robust real-time object detection. In: ICCV Workshop on Statistical and Computation Theories of Vision (July 2001)
10. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: ICIP, vol. 1, pp. 900–903 (September 2002)
11. Kruppa, H., Castrillon-Santana, M., Schiele, B.: Fast and robust face finding via local context. In: IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (October 2003)
12. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 47(1/2/3), 7–42 (2002)
13. Veksler, O.: Fast variable window for stereo correspondence using integral images. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 556–561 (2003)