Fast Audio-Visual Multi-Person Tracking for a Humanoid Stereo Camera Head

Kai Nickel and Rainer Stiefelhagen
Interactive Systems Labs, Universität Karlsruhe
76131 Karlsruhe, Germany
Email: nickel@ira.uka.de

***Abstract*— In this paper, we present an algorithm for real-time multi-person tracking with a humanoid sensor head featuring a stereo camera and multiple microphones. The proposed algorithm works with a dynamic combination of simple but fast features, which allow us to cope with limited on-board resources. By using a combination of democratic integration and layered sampling it can deal with deficiencies of single features as well as partial occlusion using the very same dynamic fusion mechanism. Both audio and video signals are processed to form a joint attention map of the surroundings. This map allows us to initialize tracks automatically and to control the robot's focus of attention dynamically.**

I. Introduction

Humanoid robots are defined by their human-like appearance as well as by their ability for human-like interaction. A basic prerequisite for this interaction is the ability of a robot to localize people in its surroundings. In this paper, we present a system for real-time person tracking using an on-board stereo camera in conjunction with 2 or more microphones. The video data is used for accurate 3-d localization of people, whereas the audio information is used for attention shifts of the robot head towards the sound source.

Person tracking with on-board cameras poses a number of challenges that we will address specifically:

- 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 visible and the user's face becomes as small as 10×10 pixels.
- The on-board computational resources are limited and cannot be used for person tracking exclusively.
- Tracks for a varying number of users have to be created and terminated automatically.

In order to tackle the aforementioned problems, we present a multi-cue integration scheme embedded into 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 situations, 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 [16], 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 [16] 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.

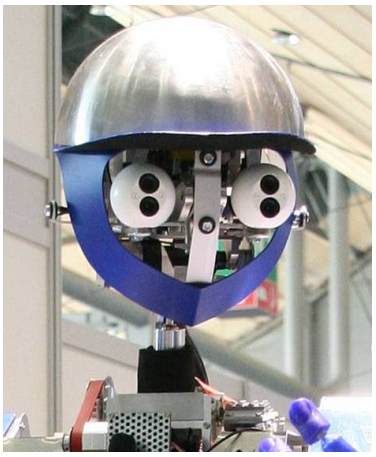


Fig. 1. Head of the humanoid ARMAR-III [1]. It is equipped with two stereo cameras and six microphones.

The combination of democratic integration and particle filters has been approached before by Spengler and Schiele [15]. 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. [13], 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. [11], this can be alleviated by *layered sampling*, if the measurement modalities 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 [2]. 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 [8], Trans-dimensional MCMC [14], or the Hybrid Joint-Seperable formulation [6]. Although these approximations reduce the complexity of joint state space tracking significantly, they still require noticeably more computational power than the separate tracker approach.

Related work on person tracking in the domain of mobile robots includes [3], who use democratic integration to track faces and objects based on motion, color and shape. They track in the 2D image space by fusing saliency maps as in [16]. In robotics, cameras are often used in combination with other sensors. For example [9] use cameras as well as ultrasonic and infrared sensors. They detect and track humans with an MCMC sampling scheme. In the work of [5], multiple people are tracked based on face detection, acoustic source localization and laser range finder data.

The remainder of this paper is organized as follows: In section II, we briefly describe the concept of particle filters and layered sampling. In section III we present our multi-cue integration scheme, which is the main contribution of this paper. It is followed, in section IV, by the definition of the cues used in the live tracking system. In section V, the multi-person tracking logic including automatic track initialization and termination is described. Section VI outlines the acoustic source localization, and section VII combines visual and acoustic stimuli into a joint attention map. Finally, section VIII shows the system in operation.

II. 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 [2], the evolution of the particle set is a two-stage process which is guided by the observation and the state evolution model[1]:

1) The prediction step (including resampling): randomly draw n new particles from the old set in consideration of their weights, and propagate them 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 $\hat{s}_t$ can be obtained from the sample set as

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

A. Layered sampling

Assuming that $\mathbf{z}$ is made up of M conditionally independent measurement sources, the observation likelihood of a particle $\mathbf{s}$ can be factorized as:

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

According to [11], the state evolution can then be decomposed into M successive intermediary steps:

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

where $\mathbf{s}^1 \cdots \mathbf{s}^{M-1}$ are auxiliary state vectors. In case of a Gaussian evolution model, this corresponds to a fragmentation into M successive steps with lower variances. 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 [11] point out, the benefit of layered sampling arises in cases where the measurement modalities can be ordered from coarse to fine. 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 V in combination with the multi-cue integration scheme described in the following.

III. 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 different cues $c \in C$ can be formulated as a weighted sum

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

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

A. Democratic integration for particle filters

Democratic integration [16] is a mechanism to dynamically adjust the mixture weights, termed reliabilities, r_c 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

[1]The time index t is omitted for the sake of brevity wherever possible.

and values close to one indicating good agreement. The reliabilities are updated after each frame by a leaky integrator using the normalized qualities:

$$\tau \dot{r}_c = \frac{q_c}{\sum_c q_c} - r_c, \tag{5}$$

with the parameter τ controlling the speed of adaptation.

In the original paper [16], 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 [13], 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 $|\hat{\mathbf{s}}_c$ - $\hat{\mathbf{s}}|$ 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 equal 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 eliminate this problem, we need a quality measure that describes how well the probability mass agglomerates around the joint hypothesis $\hat{\mathbf{s}}$. We found the following weighted MSE formulation to be an appropriate quality measure for democratic integration:

$$q_c = \left(\sum_{i=1..n} \pi_c^{(i)} (\mathbf{s}^{(i)} - \hat{\mathbf{s}})^T (\mathbf{s}^{(i)} - \hat{\mathbf{s}}) \right)^{-\lambda} \tag{6}$$

The exponent $\lambda > 0$ can be used to tweak the volatility of the quality measure.

B. Generalized cue competition

In order to allow for a fruitful combination, the set of cues should be orthogonal in the sense that different cues fail under different circumstances. One way to achieve this is to use different cue-specific feature transformations $\mathcal{F}(\mathbf{z})$ like motion, color, or shape. Failure of one feature can thus be compensated by cues relying on 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 $\mathcal{A}(\mathbf{s})$, for example different projections from state space to image space.

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

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. This could for example happen in a situation, where due to partial occlusion a certain 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.

C. 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 [16], this adaptation is described as a continuous update process with a fixed time constant τ_c:

$$\tau_c \dot{P}_c = \hat{P}_c - P_c, \tag{10}$$

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

D. Cue normalization

The likelihood functions $p_c(\mathbf{z}|\mathbf{s})$ of different cues can differ strongly in terms of responsiveness and in the absolute range of values. One cue could, for example, produce a spiky output, whereas another cue might exhibit a higher ambient value. Combining the raw response from $p_c(\mathbf{z}|\mathbf{s})$ as weighted sum (cf. eq. 4) would then be problematic. Therefore, we normalize the cues using their average response statistics μ_c and the standard deviation σ_c:

$$p_c(\mathbf{z}|\mathbf{s}) \longleftarrow \frac{\max(0, p_c(\mathbf{z}|\mathbf{s}) - \mu_c)}{\sigma_c} \tag{11}$$

The values of μ_c and σ_c can either be learned from training data, or, as in our case, be acquired at runtime by continuously drawing random samples from the entire image, i.e. from mainly non-target regions.

IV. Fast Cues for 3D Person Tracking

As the humanoid's computational on-board resources are strictly limited, cues have to be found that rely on features that can be evaluated rapidly but still have the power to segment people from background. Our proposed cues are based on the following well-known feature types: difference image, color histogram back-projection, Haar-feature cascades and stereo correlation. While the first two features are known to be fast enough for real-time applications, we will show how to apply detector cascades and stereo correlation in a way that the computational complexity is low and well scaled to the specific needs.

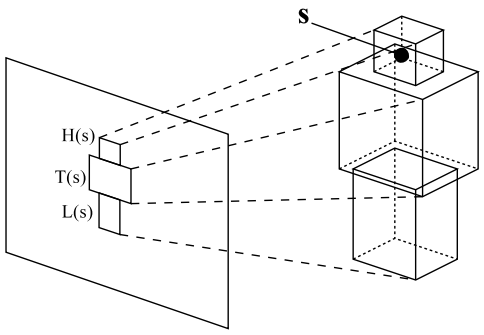


Fig. 2. The 3-box model of the human body: the state vector $\mathbf{s}$ is transformed into the image space as the projection of a cuboid representing either the head, torso, or leg region. The projection of the cuboid is approximated by a rectilinear bounding box.

As motivated in section III-B, 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. 2. 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. As all regions in our system are rectilinear bounding boxes, the sum of pixel values inside the regions can be calculated efficiently by means of 4 table-lookups in the integral image [18].

By combining the 4 different feature types with the 3 different body parts, we obtain a total number of 13 cues that will be described in the following.

A. Motion cues

The difference image can be considered as a short-time background model that requires the camera to be static only for two successive frames. It 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 the difference image both in the region around object's current location $\mathbf{x}$, as well as in the region around the object's previous location $\mathbf{x} - \dot{\mathbf{x}}$. Thus, the response of the motion cue is based on the amount of foreground pixels within the regions in question.

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. 2. We rely on the ability of the integration mechanism (see section III) 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.

B. 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 dedicated 32^3-bin histogram in RGB color space that automatically adapts to the target region using the mechanism described in section III-C. 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 for a color cue. The response of the color cue is based on the sum of pixels within the histogram back-projection.

C. Detector cues

The face detection algorithm proposed by Viola and Jones [18] employs simple features that can be efficiently computed using the integral image. In the original approach, a variable-size search window is repeatedly shifted over the image, and overlapping detections are clustered. Thus, the number of detector runs required for each scale grows quadratically with image size, making video-rate processing of high-res images computationally expensive.

In the proposed particle filter framework however, it is not necessary to scan the image exhaustively: the places to search are directly given by the particle set $\mathbf{s}^{(1..n)}$, which by definition is an optimal prior for the target's location. For each particle, the head region is projected to the image plane, and the bounding box of the projection defines the search window that is to be classified. Thus, the evaluation of a particle takes only one run of the classifier.

The response of the detector cue is based on the maximum overlap between the particle's query region and all positively classified regions from the whole particle set.

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 [4], [7] as provided by the OpenCV library.

D. Stereo correlation cues

In traditional stereo processing [12], a dense disparity map is generated by exhaustive area correlation followed by several post-filtering steps. The result are the z-values of those image pixels which lie within sufficiently textured regions. 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. For particle filter-based tracking, however, this issue can be solved in an elegant way: it is the function of the stereo cue to verify that an object exists at the hypothesized target location $\mathbf{s}$. For this purpose, it uses the entire target region $\mathcal{A}(\mathbf{s})$ as correlation window and searches for optimal correlation along the epipolar lines. The response of the stereo cue is then given by the inverse distance of the discovered disparity $\hat{d}(\mathcal{A}(\mathbf{s}))$ and the hypothesized disparity $d(\mathcal{A}(\mathbf{s}))$.

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 [17] for dense disparity calculation. We employ 3 stereo cues, one for the head (S-H), torso (S-T), and legs (S-L).

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.

V. 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 $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ of the person's head centroid in 3-dimensional space: $\mathbf{s}^{(i)} = (\mathbf{x}^{(i)}, \dot{\mathbf{x}}^{(i)})$. 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.

A. 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 II-A. 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.

Apart from the geometrical constraints implied by the stereo cues, there is a 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 other tracks. Thereby, we guarantee mutual exclusion of tracks.

B. 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 sometimes become more difficult than the actual tracking problem. In our system, we define the quality measure for a tracker to be the joint response from both stereo and regular cues at the tracker's hypothesis $\hat{\mathbf{s}}$:

$$Q(\hat{\mathbf{s}}) = \sum_{c\in C_S} r_c p_c(\mathbf{z}|\hat{\mathbf{s}}) \cdot \sum_{c\in C_R} r_c p_c(\mathbf{z}|\hat{\mathbf{s}}) \tag{12}$$

This formulation of $Q(\hat{\mathbf{s}})$ is feasible because the cues are normalized using their statistics μ_c and σ_c (section III-D). The final quality measure Q is a result of temporal filtering with a time constant ν:

$$\nu\dot{Q} = Q(\hat{\mathbf{s}}) - Q \tag{13}$$

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 *visual 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.

VI. Acoustic Source Localization

For the task of person tracking on a mobile robot, acoustic source localization has some drawbacks compared to visual localization: it is limited to speaking people, it gets distracted by non-speech noises, and the localization accuracy especially along the z-axis (distance to robot) is lower than for visual tracking. On the other hand, the advantage of acoustic source localization lies in its ability for permanent 360° covering of the surroundings. Furthermore, an acoustic event is a strong indication for the existence of a person at a certain direction. For these reasons, we use acoustic source localization not as an additional cue for the scoring of regular tracks, but rather as a central component of the robot's focus-of-attention control mechanism (see section VII).

In order to localize acoustic events, we evaluate the space of possible sound sources in the surroundings of the robot by means of dedicated particle filter named the *acoustic attention tracker*. Like for the visual attention tracker, 50% of the particles are drawn randomly at each time instance, while the remaining 50% are propagated regularly.

Given a pair of microphones and a hypothesized speaker position $\mathbf{x}$, the speech signal arrives with a certain *time delay of arrival* (TDOA) $\tau(\mathbf{x})$ depending on the spatial geometry

of the setup. To measure how well the signals from a given microphone pair support the hypothesis of a sound source at $\tau(\mathbf{x})$, we calculate the *phase transform* (PHAT) [10], which can be expressed as

$$R(\tau(\mathbf{x})) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{X_1(e^{j\omega\tau(\mathbf{x})})X_2^*(e^{j\omega\tau(\mathbf{x})})}{\left|X_1(e^{j\omega\tau(\mathbf{x})})X_2^*(e^{j\omega\tau(\mathbf{x})})\right|} e^{j\omega\tau(\mathbf{x})}\, d\omega \tag{14}$$

where $X_1(\omega)$ and $X_2(\omega)$ are the Fourier transforms of the signals of the microphone pair.

The observation model $p(\mathbf{z}|\mathbf{x})$ of the acoustic attention tracker given a particle $\mathbf{x}$ and the acoustic observation $\mathbf{z}$ is defined by interpreting the PHAT as a pseudo probability density function. We integrate the scores from all those microphone pairs $\mathcal{M}(\mathbf{x})$ that are exposed to direct sound given the particle's location $\mathbf{x}$:

$$p(\mathbf{z}|\mathbf{x}) \propto \frac{1}{|\mathcal{M}(\mathbf{x})|}\sum_{m\in\mathcal{M}(\mathbf{x})} \mathcal{S}(R_m(\tau_m(\mathbf{x}))) \tag{15}$$

with $\mathcal{S}$ being a function that cuts off negative values and smoothes the PHAT.

In addition to the acoustic attention tracker, we also evaluate the positions of all regular tracks with the acoustic observation model. If a significant correlation can be observed at a track's position, we assume that the respective person is currently speaking.

VII. Attention-based Head Control

Natural head control for a humanoid robot requires selecting one of many competing targets for the robot's focus-of-attention. In the framework of person tracking, the task is to find and to follow people with the camera head in order to keep track of the surroundings. In addition, a humanoid robot should signal its interest in a human communication partner by actively focusing him or her – while ignoring the rest of the scene. The decision rules for head control in this situation can be classified into two categories:

- Top-down: Focus on one of the existing tracks. Prefer people that are likely interaction partners because they are speaking, standing close to the robot, facing the robot, etc.
- Bottom-up: Focus on the source of a visible or audible event in order to discover new people. Scan areas that have not been observed for a while.

Due to the conflicting nature of these requirements we propose to integrate all factors that influence the robot's focus-of-attention in a joint multi-modal *attention map*. The attention map (see Fig. 4) is a 2-dim histogram in the discretized space of all possible pan/tilt positions. The value of a bin represents the attractiveness of the respective viewing direction. For each frame, the attention map is generated as follows:

- The particle distributions of all tracks are projected onto the attention map. Each bin's value increases by the accumulated particle weights $\pi^{(i)}$ of all particles falling into the respective bin. A sigmoidal weighting function is used to boost the influence of particles that are close to the robot. Another boosting factor is applied to particles belonging to the track that is currently speaking.
- The particles of the acoustic as well as of the visual attention tracker are likewise projected to the attention map. The attention trackers represent information about visual or acoustic events that are currently below the track level, but may become a regular track in the future.
- Before processing the next frame, the bins outside the current field-of-view (FoV) are multiplied with an aging factor $f_o < 1$. The bins inside the FoV are multiplied with a stronger aging factor $f_i < f_o < 1$. As a result, the non-observed regions become more and more attractive over time – unless a strong attractor within the current FoV compensates the effect.

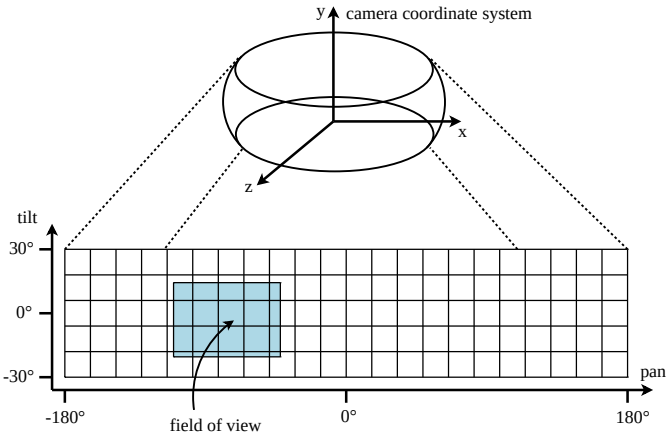


Fig. 4. The attention map is a 2-dim histogram in the discretized space of all possible pan/tilt positions.

After each frame, the attention map is searched for a pan/tilt position that maximizes the bin values in a neighborhood having the size of the FoV. To prevent the system from shifting the focus too frequently, positions far away from the current pan/tilt position are penalized by means of a sigmoid function. The influence factor of this penalizing function reduces over time to allow for big attention shifts every once in a while. After such a big shift, the influence factor is reset again to a prohibitively high value.

VIII. Experiments

Overall, the tracker showed solid performance throughout our 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. A bigger issue were phantom tracks that were triggered by non-human motion or false detections. They were sometimes kept alive for some seconds by the color models which adapted to the false positive region.

A. Implementation details

In the implementation, we made the following modifications 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. None of the 3 boxes depicted in Fig. 2 was used for the upper body detector; instead the detector employs an additional box-type that comprises head and upper half of the torso. To avoid dominance, we limited the range for a cue's influence to $0.03 \leq r_c \leq 0.5$. 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 $20ms$ 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 single core CPU. The number of particles was $n = 100$, the track threshold was set to $\Theta = 2.5$.

B. Example sequences

The first video sequence, comprising 5148 frames in total, was recorded in an office: a person stands, sits, walks around, and pretends to have a conversation with the robot head from time to time. The camera motion is controlled by a human operator. Figure 5 discusses the evolution of cue reliabilities for a selected interval of sequence 1.

Sequence 2, comprising 886 frames, was recorded without camera motion. It shows two people walking around at a distance of 1-6m from the camera; they sometimes leave the visible range or walk behind the cabinet in the center of the scene. Fig. 6 shows a snapshot from this sequence. Track loss occurred when a person was not visible for more than about $3s$. A new track got initialized as soon as the person entered the scene again.

The attention map is work-in-progress and could not be evaluated methodically yet. Fig. 7 shows a snapshot from an attention map at runtime. Both effects like smooth pursuit of people as well as sudden attention-shifts could well be observed.

IX. Conclusion

We presented a complete 3-d person tracking system for a humanoid robot head. It implements a new approach for dynamic cue combination by combining the concepts of democratic integration with layered sampling, and enables a generalized kind of competition among cues. Cues based on different feature types compete directly with cues based on different target regions so that the self-organizing capabilities of democratic integration can be fully exploited. The proposed stereo and detector cues demonstrate the increase in efficiency that lies in particle filter-based tracking: the sampled representation of the search space allows for local evaluation of features that would otherwise be expensive to process.

The same consideration led to the development of the audio-visual attention trackers: instead of searching the state space exhaustively, the attention trackers sample the search space and cluster around the most promising regions. For humanoid head control, we proposed a multi-modal attention map that fuses both top-down knowledge like the positions of the known tracks as well as bottom-up stimuli which are below the track level. Future work is needed to reduce the number of parameters involved in the composition of the attention map to a small set of descriptive factors.

Acknowledgments

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

References

[1] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An integrated humanoid platform for sensory-motor control. In *IEEE-RAS International Conference on Humanoid Robots*, Genoa, Italy, December 2006.

[2] M. Isard and A. Blake. Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[3] H. Kim, B. Lau, and J. Triesch. Adaptive object tracking with an anthropomorphic robot head. In *Proc. of the 8th International Conference on the Simulation of Adaptive Behaviors (SAB'04)*, 13-17 July 2004.

[4] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2003.

[5] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer. Providing the basis for human-robot-interaction: a multi-modal attention system for a mobile robot. In *Proc. of the 5th international conference on Multimodal interfaces*, pages 28–35, 2003.

[6] O. Lanz. Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, September 2006.

[7] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP*, volume 1, pages 900–903, September 2002.

[8] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[9] T. Miyashita, M. Shiomi, and H. Ishiguro. Multisensor-based human tracking behaviors with markov chain monte carlo methods. In *Proc. of the 4th IEEE/RAS International Conference on Humanoid Robots*, volume 2, pages 794–810, November 2004.

[10] M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. In *Proc. ICASSP '94*, pages II–273–II–276, Adelaide, Austrailia, April 1994.

[11] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, March 2004.

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, April-June 2002.

[13] C. Shen, A. Hengel, and A. Dick. Probabilistic multiple cue integration for particle filter based tracking. In *International Conference on Digital Image Computing - Techniques and Applications*, pages 309–408, 2003.

[14] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 962–969, Washington, DC, USA, 2005.

[15] M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, 14:50–58, 2003.

[16] J. Triesch and C. V. D. Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Comput.*, 13(9):2049–2074, 2001.

[17] O. Veksler. Fast variable window for stereo correspondence using integral images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 556–561, 2003.

[18] P. Viola and M. Jones. Robust real-time object detection. In *ICCV Workshop on Statistical and Computation Theories of Vision*, July 2001.

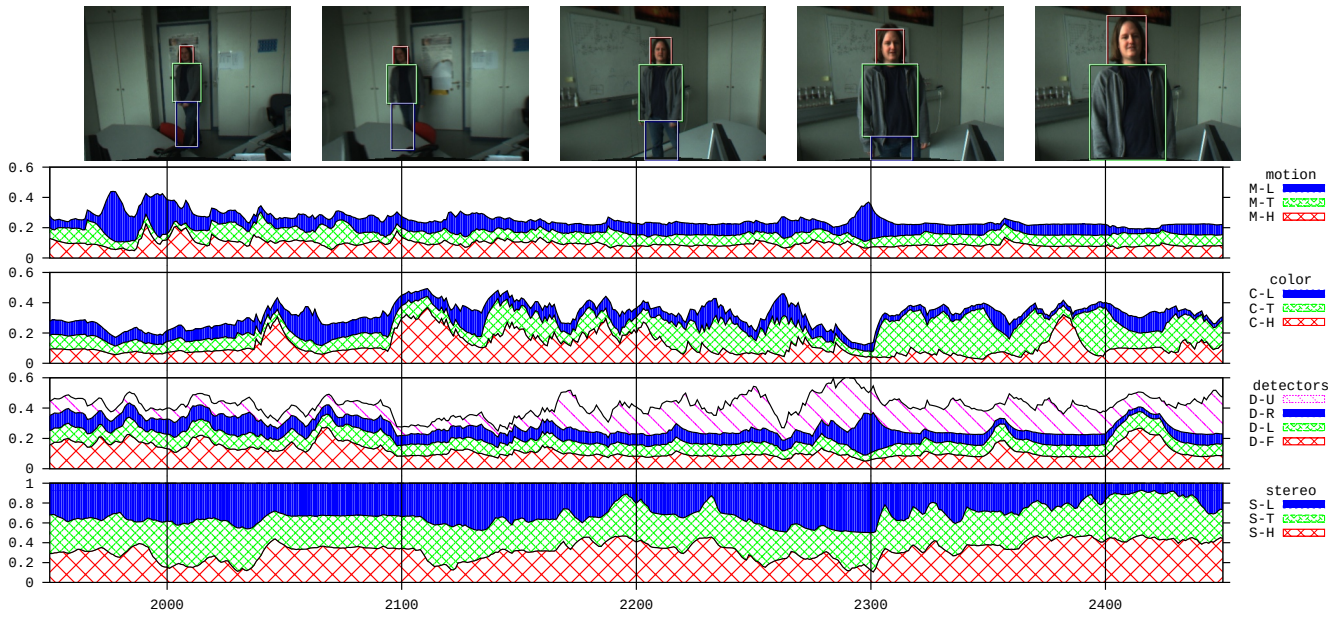
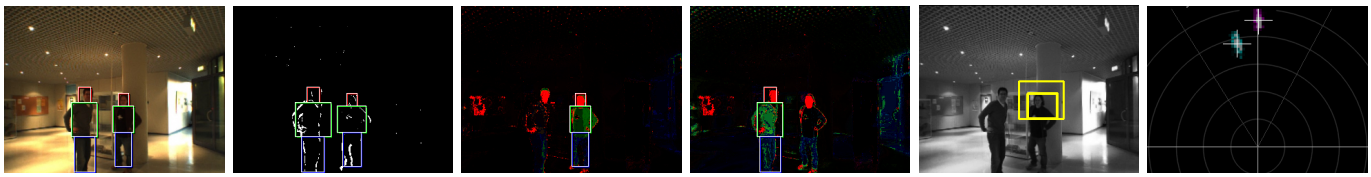


Fig. 5. Evolution of cue reliabilities in sequence 1. The cues are grouped with respect to their feature types. The three stereo cues constitute the 1st layer of the algorithm, their reliabilities sum up to 1. The remaining ten cues are used in layer 2 and sum up to 1 likewise. In the beginning of the interval around frame 2000, we can observe that the three motion cues gain influence as the person starts walking. This is compensated a moment later when the camera starts to move in order to follow the person. Throughout the selected interval, the detector cues are the most dominant ones because the person is close enough for the detectors to respond. Exceptions are frames 2100-2150, where motion blur prevents detections. The automatic invalidation of body regions can be observed twice for the legs region of the stereo cue: the first time around frame 2200 when the person gets occluded by the desk, and a second time around frame 2400 when the person stands in front of the camera in a way that the legs region is unobservable.



a) left cam. image b) diff. image c) color person 1 d) color person 2 e) detectors f) plan-view

Fig. 6. Snapshot from sequence 2. In c) and d), the color support maps for head, torso and legs of the respective person are merged into the red/green/blue channels of the image. Image f) is a plan view of the ground plane displaying the particle distribution of the two active tracks; the camera is located at the origin of the depicted coordinate system.

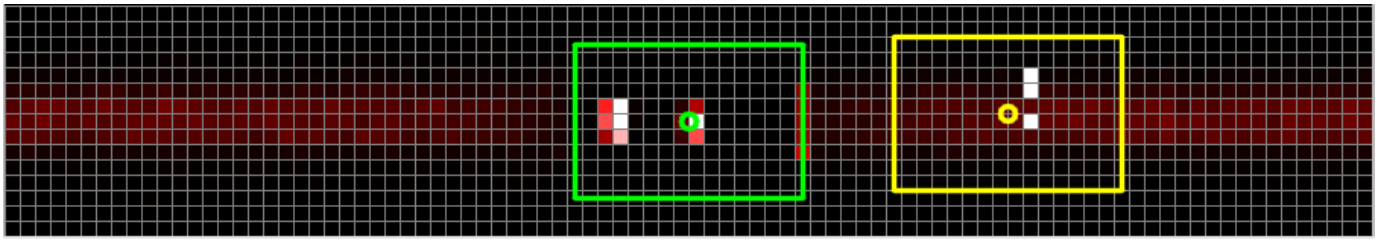


Fig. 7. Attention map at runtime. The green box in the center represents the robot's current field-of-view. The system proposes an attention shift towards an acoustic stimulus at approx. 90° to the right (yellow box). The horizontal belt of medium bin values (shades of red) stems from random a/v attention particles that explore visually un-observed regions. The resolution of the attention map was chosen to be 4° in both pan and tilt.