Robust Real-time Stereo-based Markerless Human Motion Capture

Pedram Azad, Tamim Asfour, Rüdiger Dillmann
University of Karlsruhe, Germany *azad@ira.uka.de*, *asfour@ira.uka.de*, *dillmann@ira.uka.de*

Abstract— **The main problem of markerless human motion capture is the high-dimensional search space. Tracking approaches therefore utilize temporal information and rely on the pose differences between consecutive frames being small. Typically, systems using a pure tracking approach are sensitive to fast movements or require high frame rates, respectively. However, on the other hand, the complexity of the problem does not allow real-time processing at such high frame rates. Furthermore, pure tracking approaches often only recover by chance once tracking has got lost. In this paper, we present a novel approach building on top of a particle filtering framework that combines an edge cue and 3D hand/head tracking in a distance cue for human upper body tracking, as proposed in our earlier work. To overcome the mentioned deficiencies, the solutions of an inverse kinematics problem for a – in the context of the problem – redundant arm model are incorporated into the sampling of particles in a simplified annealed particle filter. Furthermore, a prioritized fusion method and adaptive shoulder positions are introduced in order to allow proper model alignment and therefore smooth tracking. Results of real-world experiments show that the proposed system is capable of robust online tracking of 3D human motion at a frame rate of 15 Hz. Initialization is accomplished automatically.**

I. Introduction

Markerless human motion capture means to capture human motion without any additional arrangements required, by operating on image sequences only. Commercial human motion capture systems such as the VICON system (`www.vicon.com`), which are popular in the film industry as well as in the biological research field, require reflective markers and time consuming manual post-processing of captured sequences. In contrast, a real-time markerless human motion capture system using the image data acquired by the robot's head would allow online imitation-learning in a natural way. Another application for the data computed by such a system is the recognition of human actions and activities, serving as a perception component for human-robot interaction.

For application on an active head of a humanoid robot, a number of restrictions has to be coped with. In addition to the limitation to two cameras positioned at approximately eye distance, one has to take into account that an active head can move. Furthermore, computations have to be performed in real-time, and most importantly for practical application, the robustness of the tracking must not depend on a high frame rate or slow movements, respectively.

In the following, a short overview of approaches to markerless human motion capture that are relevant for application on humanoid robot systems is given. Approaches operating on 3D data either extend the ICP algorithm for application to articulated object tracking ([1], [2]) or utilize an optimization method based on 3D-3D correspondences [3]. The 3D point clouds used as input are either acquired by disparity maps or a 3D sensor is used such as the SwissRanger (`www.mesa-imaging.ch`). Image-based approaches are either search-based ([4], [5]), utilize an optimization approach based on 2D-3D correspondences [6], [7], [8] (resp. [9] for articulated hand tracking), or are based on particle filtering. In [10], it was shown that human motion can be successfully tracked with particle filtering, using three cameras positioned around the scene of interest. In [11], it was shown that with the same principles, 3D human motion can be estimated from monocular image sequences to some degree, when learning a motion model. Recently, we have proposed the incorporation of stereo-based 3D hand/head tracking for an additional distance cue in [12]. In [13], in addition, a certain percentage of the particles is sampled with a Gaussian distribution around a single solution computed by an analytical inverse kinematics method for the purpose of re-initialization. Taking into account *all* relevant solutions of the inverse kinematics problem is not considered.

In Section II, the basic components of the used particle filtering framework are introduced, namely the utilized 3D human model and the used visual cues. The proposed approach consisting of the components hierarchical search, a prioritized fusion method, adaptive noise in sampling, adaptive shoulder positions, and the incorporation of the solutions of an inverse kinematics problem is presented in the Sections III–VII. The results of real-world experiments are presented in Section VIII, ending with a conclusion in Section IX.

II. Basic Components

A. Human Upper Body Model

In the proposed system, a kinematics model of the human upper body consisting of 14 DoF is used, not modeling the neck joint. The shoulder is modeled as a ball joint with 3 DoF, and the elbow as a hinge joint with 1 DoF. Additional 6 DoF are used for the base transformation. With this model, rotations around the axis of the forearm cannot be modeled. Capturing the forearm rotation would require tracking of the hand, which is regarded as a separate problem.

The shoulder joints are implemented with an axis/angle representation in order to avoid problems with singularities, which can occur when using Euler angles. The base rotation is modeled by Euler angles to allow a better imagination so that joint space restrictions can be defined easily. For the geometric model, the body sections are fleshed out by sections of a cone with circular cross-sections.

B. Image Processing Pipeline

The image processing pipeline transforms each input image pair into a binarized skin color image pair $I_{s,l}, I_{s,r}$ and a gradient image pair $I_{g,l}, I_{g,r}$, which are used by the likelihood functions presented in Section $II - C$. In Fig. 1, the input and outputs for a single image are illustrated. This pipeline is applied twice: once for the left and once for the right camera image. In order to allow for ego-motion, figure-ground segmentation is performed by shirt color segmentation, which is only needed for distinguishing edges that belong to the person's contour from edges belonging to the background. Details are given in [14].

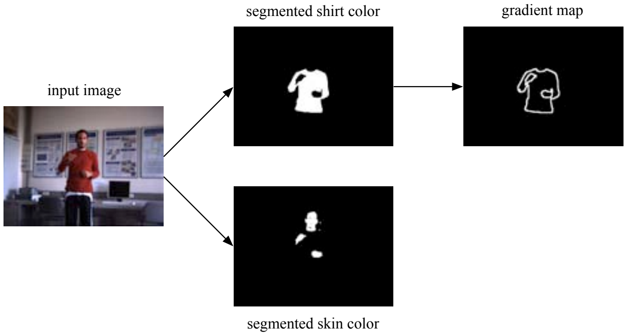

Fig. 1. Illustration of the input and outputs of the image processing pipeline.

C. Cues

In the following, the cues that are used in the proposed system are presented. The formulations are given for a single image; their application to stereo image pairs is explained in Section IV.

1) Edge Cue: According to [10], for the edge cue, the gradient values from the gradient image I_g are summed up along the projected model contours. Assuming that the gradient image has been remapped to the interval [0, 1], the evaluation function is defined as:

$$w_g(I_g, P) = 1 - \frac{1}{|P|} \sum_{i=1}^{|P|} I_g(\boldsymbol{p}_i) \,, \tag{1}$$

where P denotes the set of sampled 2D contour points. Note that compared to [10], squaring is omitted, which turned out not to have any significant effect. The likelihood function reads:

$$p_g(I_g \,|\, \boldsymbol{s}) \propto \exp\left\{ -\frac{1}{2\sigma_g^2} w_g(I_g, f_g(\boldsymbol{s})) \right\} \,, \tag{2}$$

where the function f_g computes the set of sampled 2D points P for a given model configuration $\boldsymbol{s}$.

2) Distance Cue: According to [12], the distance cue evaluates the squared distances between distinct model points and their absolute 3D measurements in the current stereo pair. In the proposed system, the hands and the head of the person are used as such points, which are tracked by a separate hand/head tracking system. The evaluation function of the distance cue is defined as follows:

$$w_d(I_d, P) = \sum_{i=1}^{|P|} |\boldsymbol{p}_i - \boldsymbol{p}'_i(I_d)|^2 \,, \tag{3}$$

where $P = \{\boldsymbol{p}_i\}$ denotes a set of transformed model points and $\boldsymbol{p}'_i(I_d)$ their measured positions that have been computed on the basis of the observations I_d. In the proposed system, these observations are the skin color segmentation results $I_{s,l}, I_{s,r}$ (see Section II-B). In order to apply this evaluation function for tracking, model points must be transformed into the coordinate system the measurements are accomplished in, yielding the point set P. For this purpose, the transformation $f_{d,i} : R^{\dim(\boldsymbol{s})} \to R^{\dim(\boldsymbol{p}_i)}$ is used, which maps a certain model point $\boldsymbol{p}_{m,i}$ to the coordinate system of the corresponding measured point $\boldsymbol{p}'_i$, given a model configuration $\boldsymbol{s}$. The function f_d performs this transformation $f_{d,i}$ for each desired model point and thereby computes the point set P. Finally, the likelihood function p_d can be formulated as follows:

$$p_d(I_d \,|\, \boldsymbol{s}) \propto \exp\left\{ -\frac{1}{2\sigma_d^2} w_d(I_d, f_d(\boldsymbol{s})) \right\} \,. \tag{4}$$

III. Hierarchical Search

The most general approach is to use one particle filter for estimating all degrees of freedom of the model, as done in our earlier work ([12], [14]). The advantage is that by estimating all degrees of freedom simultaneously, potentially the orientation of the torso can be estimated as well. In practice, however, the human model is not precise enough to benefit from this potential – if the sensor system is restricted to a single stereo camera system.

To reduce the number of particles, a hierarchical search is performed i.e. the search space is partitioned explicitly. Since the head is tracked for the distance cue anyway, the head's position can be used as the root of the kinematic chain. By doing this, only 3 DoF of the base transformation remain to be estimated. If not modeling the neck joint, these degrees of freedom describe the orientation of the torso. Since the torso orientation can hardly be estimated on the basis of 2D measurements only, it is regarded as a separate problem. In order to achieve robustness to small changes of the body rotation without actually knowing it, the shoulder positions are modeled to be adaptive, as will be described in Section VI.

With static shoulder positions (relative to the head), the final estimation problem for the particle filter would consist of 4 DoF for each arm; the 3 DoF of the base translation are estimated directly by a separate particle filter used for head tracking. Intuitively, estimating the 4 DoF of one arm with a separate particle filter sounds simple and one would assume that this approach would lead to an almost perfect result – given the restriction of a more or less frontal view of the person. However, various extensions are necessary to allow smooth and robust tracking of arm movements, which will be introduced in the following sections.

IV. Prioritized Fusion

The conventional approach for combining several cues within a particle filtering framework is to multiply the results of the respective likelihood functions. The quality and accuracy achieved by such an approach strongly depends on the cues agreeing on the way to the target configuration. In practice, however, different cues have different characteristics. While the likelihood functions of different cues in general have their global maximum in the vicinity of the true configuration, i.e. agree on the final goal, they often exhibit totally different local maxima. This circumstance often causes the likelihood functions to fight against each other, resulting in a typically noisy estimation.

In the proposed system, the edge cue and the distance cue have to be fused. Since the distance cue is the more reliable cue due to the explicit measurement of the 3D head and hand position, the idea is to introduce a prioritization scheme: If the distance error of the hand for the current estimation is above a predefined threshold, then the error of the edge cue is ignored by assigning the maximum error of 1; otherwise the distance error is set to zero. By doing this, the particle filter rapidly approaches configurations in which the estimated hand position is within the predefined minimum radius of the measured hand position – without being disturbed by the edge cue. All configurations that satisfy the hand position condition suddenly produce a significantly smaller error, since the distance error is set to zero and the edge error is < 1. Therefore, within the minimum radius, the edge cue can operate undisturbedly. Applying this fusion approach allows the two cues to act complementary instead of hindering each other.

Algorithm 1 ComputeLikelihoodArm($I_{g,l}$, $I_{g,r}$, $\boldsymbol{p}_h$, $\boldsymbol{s}$) → π

```
$e_g := \dfrac{w_g(I_{g,l}, f_{g,l}(\boldsymbol{s})) + w_g(I_{g,r}, f_{g,r}(\boldsymbol{s}))}{2}$
$e_d := |\boldsymbol{p}_h - f_d(\boldsymbol{s})|^2$
If $e_d < t_d^2$ then set $e_d := 0$ else set $e_g := 1$.
$e_d := \dfrac{s_d \cdot e_d}{e_d^{(t-1)}}$
If $e_d > 50$ then set $e_d := 50$.
$\pi := \exp\{-(e_d + s_g \cdot e_g)\}$
```

In addition, the range of the distance error is limited by division by the distance error $e_d^{(t-1)}$ for the estimated configuration of the previous frame. Otherwise the range of the distance error could become very large in some cases, potentially leading to numerical instabilities. Finally, the argument to the exponential function is cut off when it exceeds the value 50. The final likelihood function fusing the errors calculated by the edge cue and the distance cue is summarized in Algorithm 1. The inputs to the algorithm are the gradient map stereo pair $I_{g,l}, I_{g,r}$, the measured hand position $\boldsymbol{p}_h$, and the configuration $\boldsymbol{s}$ to be evaluated. For the weighting factors $s_g := \frac{1}{2\sigma_g^2}$ and $s_d := \frac{1}{2\sigma_d^2}$, $s_g = s_d = 10$ is used. As the minimum radius, $t_d = 30\,\text{mm}$ is used. The function f_d computes the 3D position of the hand for a given joint configuration $\boldsymbol{s}$ using the forward kinematics.

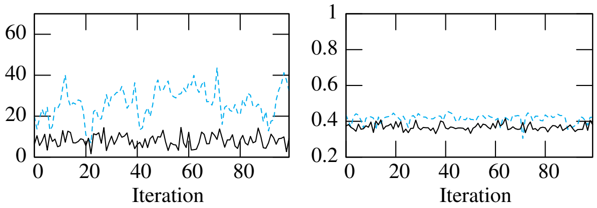

Fig. 2. Illustration of the effect of the proposed fusion method on the overall edge and distance error for a typical situation. The solid line indicates the result computed using the proposed fusion method, the dashed line using conventional fusion. Left: Euclidean distance error in [mm]. Right: edge error.

In Fig. 2, the results of 100 iterations of the particle filter are plotted, after the particle filter has already converged. As can be seen, using prioritized fusion does not only lead to smaller edge and distance errors, but the variances are also considerably smaller. The reason is that the cues do not agree on the same goal and thus cannot find the optimal configuration when using the conventional fusion method.

V. Adaptive Noise

In [15], the idea was raised to not apply a constant amount of noise for sampling new particles, but to choose the amount to be proportional to the variance of each parameter. Since the variance of a parameter is not necessarily related to an error of the parameter itself, we choose the amount for all degrees of freedom of an arm to be proportional to the current overall edge error of that arm. In Fig. 3, the overall errors are plotted for the same example as used for Fig. 2, comparing the application of adaptive noise to constant noise. In both cases, prioritized fusion was applied (see Section IV).

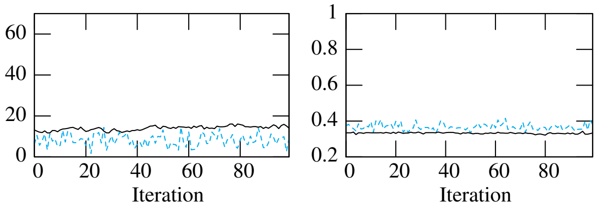

Fig. 3. Illustration of the effect of adaptive noise on the overall edge and distance error for a typical example. The solid line indicates the result computed using adaptive noise, the dashed line using a constant amount of noise. Left: Euclidean distance error in [mm]. Right: edge error.

Note that not only the standard deviation of the estimated trajectory is lower by a factor of approx. 2–3 – which is reasonable when reducing the amount of noise – but also the edge error exhibits a lower magnitude, compared to the application of a constant amount of noise. This means that the particle filter could find a better goal configuration when applying adaptive noise. The reason is that when applying a constant amount of noise, the amount must be chosen to be relatively high in order to cope with (unpredictable) motion. In the vicinity of the true configuration, however, this amount is too high to allow a fine search, whereas adaptive noise allows to search with a higher resolution in a smaller subspace. The reason for the slightly higher distance error is that the prioritized fusion method with $t_d = 30\,\text{mm}$ in Algorithm 1 gives the configurations the freedom to produce any distance error smaller than 30 mm. If desired, t_d could be chosen to be smaller. However, this would lead to less robustness to the effects of clothing, and in particular to loose sleeves.

VI. Adaptive Shoulder Position

In general, one of the main problems with real image data is that the model does not perfectly match the observations. In the case of motion capture of the upper body, the problem often occurs for the shoulder joint, which is usually approximated by a single ball joint, the glenohumeral joint. In reality, however, the position of this ball joint depends on two other shoulder joints, namely the acromioclavicular joint and the sternoclavicular joint. When not modeling these joints, the upper body model is too stiff to allow proper alignment; an exemplary situation is shown for the person's right arm in Fig. 4. Even more problematic situations occur, when the arm is moved to the back.

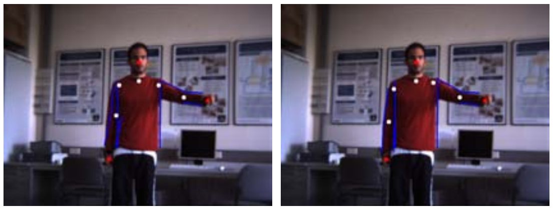

Fig. 4. Illustration of the effect of adaptive shoulder positions. The main difference can be observed for the person's right arm; the model edges cannot align with the image edges when using a static shoulder position, since the shoulder position is too much inside. The white dots indicate joint positions of the model, black dots mark the positions of the head and the hands of the model, and red dots mark the respective measured positions. Left: static shoulder position. Right: adaptive shoulder position.

In the proposed system, this problem becomes even more severe, since the shoulder positions are inferred by the head position, assuming a more or less frontal view. Our solution is to estimate the shoulder position within the particle filter of the arm, i.e. going from 4 DoF to 7 DoF. As it turns out, the higher dimensionality does not lead to any practical problems, whereas the freedom of the shoulder positions for aligning the model results in a significantly more powerful system. The three additional degrees of freedom define a translation in 3D space. The limits are defined as a cuboid, i.e. by $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$. The right image from Fig. 4 shows the improvement in terms of a better alignment of the person's right arm achieved by the adaptive shoulder position. As can be seen, the right shoulder has been moved slightly outwards in order to align the contour of the model with the image edges. Furthermore, the shoulder has been moved downwards so that the distance error is within the minimum radius, allowing the edge cue to operate undisturbedly.

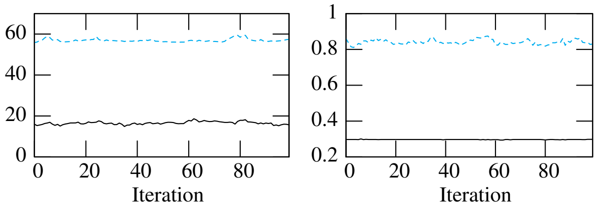

Fig. 5. Illustration of the effect of adaptive shoulder positions on the overall edge and distance error by the example of the person's right arm shown in Fig. 4. The solid line indicates the result computed using an adaptive shoulder position, the dashed line using a static shoulder position. Left: Euclidean distance error in [mm]. Right: edge error.

In Fig. 5, the overall errors are plotted for the person's right arm shown in Fig. 4, comparing a static to an adaptive shoulder position. As can be seen, both errors are significantly lower when modeling the shoulder position to be adaptive. The reason for the lower distance error is that the shoulder joint could move downwards so that the hand of the model can approach the hand in the image. The lower edge error is more significant: In the case of a static shoulder position, the edge error could not be minimized at all, while the adaptive shoulder position allows practically perfect alignment.

VII. Incorporating Inverse Kinematics

The system which has been presented so far performs well and can acquire smooth and accurate trajectories. The success of the tracker, however, depends on the speed of the person's movements with respect to the frame rate of the camera. This is typical for all pure tracking approaches, since they rely on the differences between consecutive frames being small. This leads to the main problem that once tracking has got lost, in general, tracking systems only recover by chance. The inclusion of the measured head and hand positions in the proposed system already leads to a considerable improvement, since the distance cue allows comparatively fast and reliable recovery.

One problem that remains are local minima. A typical situation is the automatic initialization of the tracking system. Here, the configuration must be found without the aid of temporal information. An example of such a local minimum is shown for the person's right arm in Fig. 6. Another problematic situation occurs when the arm is almost fully extended. In this

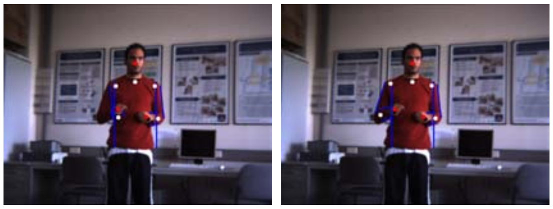

Fig. 6. Illustration of the effect of incorporating inverse kinematics. Left: without inverse kinematics. Right: with inverse kinematics.

case, one of the 3 DoF of the shoulder – namely the rotation around the upper arm – cannot be measured due to the lack of available information. Problems now occur when the person starts to bow the elbow, since the system cannot know at this point, in which direction the hand will move to. If the guess of the system is wrong, then the distance between the true configuration and the state of the particle filter can suddenly become very large and tracking gets lost.

In order to overcome these problems, the redundant inverse kinematics of the arm are incorporated into the sampling step of the particle filter. Given a 3D shoulder position $\boldsymbol{s}$, a 3D hand position $\boldsymbol{h}$, the length of the upper arm a, and the length of the forearm b, the set of all possible arm configurations is described by a circle on which the elbow can be located. The position of the elbow on this circle can be described by an angle α. Algorithm 2 analytically computes for a given angle α the joint angles $\theta_1, \theta_2, \theta_3$ for the shoulder and the elbow angle θ_4. The rotation matrix R_b denotes the base rotation from the frame the shoulder position $\boldsymbol{s}$ was measured in. Since the computations assume that the base rotation is zero, the shoulder position $\boldsymbol{s}$ and the hand position $\boldsymbol{h}$ are rotated back with the inverse rotation R_b at the beginning. The underlying geometric relationships are illustrated in Fig. 7.

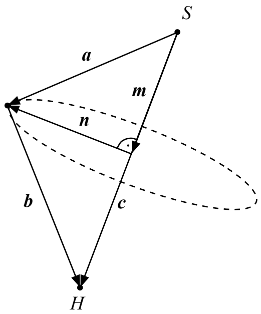

Fig. 7. Illustration of the geometric relationships for the inverse kinematics computations.

The general idea of the used inverse kinematics method is as follows. The starting point is the calculation of the vector $\boldsymbol{m}$, which points from the shoulder position to the center of the circle. Subsequently, for each α a vector $\boldsymbol{n}$ is calculated that points from the center to the position of the elbow. Then, one possible rotation matrix R_e for the shoulder joint is calculated that moves the elbow to the computed position. For this rotation matrix, the rotation matrix $R_y(\varphi)$ for the rotation around the upper arm is calculated that satisfies the hand constraint. The final rotation matrix R for the shoulder joint satisfying both the elbow and the hand constraint is composed of the rotations R_e and $R_y(\varphi)$. The elbow angle θ_4 is given by $\gamma - \pi$, where γ is the angle between $-\boldsymbol{a}$ and $\boldsymbol{b}$ (see Fig. 7), since $\theta_4 \leq 0$ and for a fully extended arm, it is $\theta_4 = 0$.

In order to take into account joint constraints, not all possible vectors $\boldsymbol{n}$ are considered, but only a subset. For this, the shoulder rotation that is necessary for bringing the hand to the target position $\boldsymbol{c}$ is reproduced in a defined way. Since the computation of this rotation is ambiguous and a defined elbow position is desired, the rotation is decomposed into two single rotations. The first rotation moves the hand to the proper position in the sagital plane (yz), the second rotation finally moves the hand to the target position. By applying the same two rotations to the vector $(0,\, 0,\, -a \sin \beta)^T$, which defines $\boldsymbol{n}$ in a canonical way, the vector $\boldsymbol{n}_0$ is calculated as a reference. For this $\boldsymbol{n}_0$, according to human-like joint constraints, plausible values for the bounds of $\alpha \in [\alpha_{min},\, \alpha_{max}]$ are $\alpha_{min} = -0.2,\ \alpha_{max} = \pi$ for the left arm, and $\alpha_{min} = -\pi,\ \alpha_{max} = 0.2$ for the right arm, respectively.

Algorithm 2 ComputeInverseKinematics(R_b, $\boldsymbol{s}$, $\boldsymbol{h}$, a, b, α) → $\theta_1, \theta_2, \theta_3, \theta_4$

```
$\boldsymbol{c} := R_b^T(\boldsymbol{h} - \boldsymbol{s})$
If $|\boldsymbol{c}| > 0.95\,(a+b)$ then set $\boldsymbol{c} := 0.95\,(a+b)\dfrac{\boldsymbol{c}}{|\boldsymbol{c}|}$.
If $|\boldsymbol{c}| < |a-b|$ then set $\boldsymbol{c} := |a-b|\dfrac{\boldsymbol{c}}{|\boldsymbol{c}|}$.
$c := |\boldsymbol{c}|$
$\beta := \arccos \dfrac{a^2 + c^2 - b^2}{2ac}$
$\gamma := \arccos \dfrac{a^2 + b^2 - c^2}{2ab}$
$\boldsymbol{u}_1 := (0,\, c,\, 0)^T$
$\boldsymbol{u}_2 := (0,\, c_y,\, \mathrm{sign}(c_z)\sqrt{c_x^2 + c_z^2})^T$
$\boldsymbol{n}_0 := \mathrm{Rotate}((0,\, 0,\, -a \sin \beta)^T,\, (1,\, 0,\, 0)^T,$
    $\mathrm{Angle}(\boldsymbol{u}_1,\, \boldsymbol{u}_2,\, (1,\, 0,\, 0)^T))$
$\boldsymbol{n}_0 := \mathrm{Rotate}(\boldsymbol{n}_0,\, (0,\, 1,\, 0)^T,\, \mathrm{Angle}(\boldsymbol{u}_2,\, \boldsymbol{c},\, (0,\, 1,\, 0)^T))$
$\boldsymbol{n} := \mathrm{Rotate}(\boldsymbol{n}_0,\, \boldsymbol{c},\, \alpha)$
$\boldsymbol{m} := \dfrac{\boldsymbol{c}}{|\boldsymbol{c}|} a \cos \beta$
$\boldsymbol{a} := \boldsymbol{m} + \boldsymbol{n}$
$\boldsymbol{b} := \boldsymbol{c} - \boldsymbol{a}$
$\boldsymbol{u}_1 := (0,\, 1,\, 0)^T$
$\boldsymbol{u}_2 := \dfrac{\boldsymbol{a}}{|\boldsymbol{a}|}$
$R_e := \mathrm{RotationMatrixAxisAngle}(\boldsymbol{u}_1 \times \boldsymbol{u}_2,$
    $\mathrm{Angle}(\boldsymbol{u}_1,\, \boldsymbol{u}_2,\, \boldsymbol{u}_1 \times \boldsymbol{u}_2))$
$\varphi := \mathrm{Angle}(R_e \cdot R_x(\gamma - \pi) \cdot (0,\, b,\, 0)^T,\ \boldsymbol{b},\ \boldsymbol{a})$
$R := R_e \cdot R_y(\varphi)$
$(\theta_1,\, \theta_2,\, \theta_3) := \mathrm{GetAxisAngle}(R)$
$\theta_4 := \gamma - \pi$
```

Finally, the inverse kinematics method must be incorporated into the sampling step of the particle filter. For this purpose, the general idea of annealed particle filtering [10] is exploited, which is running the particle filter several times on the same frame while adapting the parameters for each run in a suitable way in order to support faster convergence. In [10], the adapted parameter was the weighting factor for the evaluation function, with which the broadness of the resulting probability distribution can be modified.

A naive approach would be to apply the inverse kinematics for sampling all particles of the first run. Doing this would reset the complete state of the particle filter, including the elimination of all hypotheses, which are stored in the probability distribution. To keep the characteristics and benefits of a particle filter, only a certain percentage of the particles is sampled according to the inverse kinematics; all other particles are sampled in the conventional way. By doing this, new particles created by the inverse kinematics sampling get the chance to establish themselves, while particles with great likelihoods from the last generation, i.e. frame, can survive according to the particle filtering principle. For each frame, we use one such mixed run, followed by three normal runs of the particle filter. These additional runs allow the particle filter to sort out weak particles from the inverse kinematics sampling and to converge to a representative probability distribution. In the first run, 60% of the particles are sampled according to the inverse kinematics, while the other 40% are sampled in the conventional way. In Algorithm 2, the hand position $\boldsymbol{h}$ is measured by hand tracking, and for the shoulder position $\boldsymbol{s}$, the estimated shoulder position offset from the previous frame is applied to the measured head position.

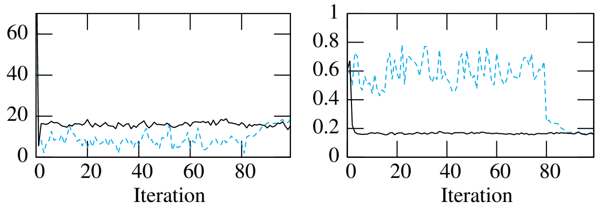

Fig. 8. Illustration of the effect of inverse kinematics sampling on the overall edge and distance error by the example of the person's right arm shown in Fig. 6. The solid line indicates the result computed with inverse kinematics sampling, the dashed line without. Left: Euclidean distance error in [mm]. Right: edge error.

In Fig. 8, the overall errors are plotted for the person's right arm shown in Fig. 6, comparing conventional sampling to sampling taking into account inverse kinematics. As can be seen, conventional sampling searches for 80 frames within the minimum distance radius until the true configuration is found and thus the edge error decreases. The corresponding joint angle trajectories are shown in Fig. 9. The proposed combined inverse kinematics sampling leads to almost immediate convergence, in contrast to sampling without inverse kinematics. To allow comparison of the results, the particle filter was run four times in one iteration of the conventional sampling method.

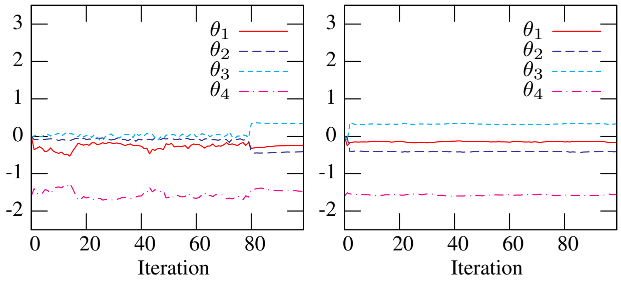

Fig. 9. Illustration of the effect of inverse kinematics sampling on the trajectory of the person's right arm shown in Fig. 6. Left: without inverse kinematics. Right: with inverse kinematics. The standard deviations for the iterations 3–99 for the angles $\theta_1, \theta_2, \theta_3, \theta_4$ are 0.011, 0.0070, 0.0076, 0.015, when using inverse kinematics sampling. The units are in radians.

VIII. Experimental Results

A. Runtime

In Table I, the runtimes for the different processing stages are given for the proposed system. The runtimes have been measured on a 3 GHz single core CPU for a test sequence consisting of 840 24 bit RGB stereo images with a resolution of 640×480 each. For arm motion tracking, 150 particles with four runs were used. The total processing time of 66 ms yields a processing rate of 15 Hz.

	Time [ms]
Skin color segmentation	4
Shirt color segmentation	20
Edge image calculation	6
Particle filters for hand/head tracking	6
Particle filters for arm motion tracking	30
Total	**66**

TABLE I

Processing times for the proposed system.

B. Real-world Experiments

For the results presented in this section, an exemplary sequence consisting of 840 frames captured at a frame rate of 30 Hz was processed and analyzed. The sequence was processed with the proposed system once on all 840 frames and once using every second frame only. By doing this, the degradation of the accuracy with lower frame rates can be observed. As will be shown, the proposed system operates robustly on lower frame rates as well, which is crucial for robust online application. The system proved to be applicable for online reproduction of movements on the humanoid robot ARMAR III, as presented in [16].

The estimated trajectories of the left and right arm are plotted in the Fig. 10 and Fig. 11, respectively. The angles θ_1–θ_4 are the direct output of the particle filter. The angles θ_1–θ_3 define a vector whose direction represents the rotation axis and whose magnitude the rotation angle. As can be seen, the trajectories acquired at 15 Hz and 30 Hz mostly equal. The greatest deviations can be observed for the first 100 frames of the left arm in Fig. 10. However, the magnitude of the deviation is not representative for the actual error. The elbow angle for these frames is near zero, and the different values result from the uncertainty of the estimation of the upper arm rotation – a problem that is not related to the frame rate. Due to the small elbow angle, the projections of both trajectories look similar. The deviation for the angle θ_2 of the right arm for the frames 670–840 in Fig. 11 is due to the same ambiguity; again, the elbow angle is near zero. Judging from the visualized model configurations in 2D and 3D, both alternatives are plausible. For stable recognition or reproduction of such configurations with a humanoid robot system, the trajectories must post-processed in order to ensure continuity and uniqueness. This post-processing can be performed online at run-time, as applied for reproduction of movements on the humanoid robot ARMAR III presented in [16].

Finally, in Fig. 12 snapshots of the state of the tracker are given for the test sequence. Each snapshot corresponds to a frame $1+k\cdot 60$ from the Fig. 10 and Fig. 11, respectively. Note that not only the projection of the human model configuration to the left camera image is plausible, but also the estimated 3D pose illustrated by the 3D visualization of the human model is correct.

IX. Discussion and Outlook

We have presented a stereo-based markerless human motion capture system that is capable of robust real-time tracking of upper body motion. The processing rate amounts to 15 Hz on a 3 GHz single core CPU operating on stereo color image pairs with a resolution of 640×480. We introduced a prioritized fusion method for combining the edge cue and the distance cue, the latter operating on 3D positions acquired by a 3D hand/head tracking system. It was shown that this fusion method together with adaptive noise leads to substantially smoother and more accurate trajectories. Accurate model alignment is accomplished by modeling the shoulder position to be adaptive – in contrast to conventional models using a stiff ball joint for the shoulder. The introduced incorporation of the solutions of an inverse kinematics problem with a redundancy degree of one into particle sampling reduces the problem of local minima drastically, allowing for immediate recovery and automatic initialization.

In the current system, 3D hand/head tracking is performed separately in a pre-processing step for each frame. In the near future, we plan to resolve ambiguities that can occur throughout hand/head tracking by utilizing the evaluation function of the particle filters used for arm tracking. In this way, hand/head tracking and arm tracking can mutually support each other, rather than arm tracking benefitting from hand/head tracking only.

Acknowledgment

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) and funded by the European Commission.

References

[1] D. Demirdjian, T. Ko, and T. Darrell, "Constraining Human Body Tracking," in *International Conference on Computer Vision (ICCV)*, Nice, France, 2003, pp. 1071–1078.

[2] S. Knoop, S. Vacek, and R. Dillmann, "Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP," in *International Conference on Humanoid Robots (Humanoids)*, Tsukuba, Japan, 2005.

[3] D. Grest, , J. Woetzel, and R. Koch, "Nonlinear Body Pose Estimation from Depth Images," *Lecture Notes in Computer Science*, vol. 3663, pp. 285–292, 2005.

[4] D. Gavrila and L. Davis, "3-D Model-based tracking of humans in action: a multi-view approach," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, 1996, pp. 73–80.

[5] K. Rohr, "Human Movement Analysis based on Explicit Motion Models," *Motion-Based Recognition*, pp. 171–198, 1997.

[6] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Santa Barbara, USA, 1998, pp. 8–15.

[7] S. Wachter and H.-H. Nagel, "Tracking Persons in Monocular Image Sequences," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 174–192, 1999.

[8] D. Grest, D. Herzog, and R. Koch, "Monocular Body Pose Estimation by Color Histograms and Point Tracking," in *DAGM-Symposium*, Berlin, Germany, 2006, pp. 576–586.

[9] T. E. de Campos, B. J. Tordoff, and D. W. Murray, "Recovering Articulated Pose: A Comparison of Two Pre and Postimposed Constraint Methods," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 1, pp. 163–168, 2006.

[10] J. Deutscher, A. Blake, and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, USA, 2000, pp. 2126–2133.

[11] H. Sidenbladh, "Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, 2001.

[12] P. Azad, A. Ude, T. Asfour, G. Cheng, and R. Dillmann, "Image-based Markerless 3D Human Motion Capture using Multiple Cues," in *International Workshop on Vision Based Human-Robot Interaction*, Palermo, Italy, 2006.

[13] M. Fontmarty, F. Lerasle, and P. Danes, "Data Fusion within a modified Annealed Particle Filter dedicated to Human Motion Capture," in *International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, 2007, pp. 3391–3396.

[14] P. Azad, A. Ude, T. Asfour, and R. Dillmann, "Stereo-based Markerless Human Motion Capture for Humanoid Robot Systems," in *International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007, pp. 3951–3956.

[15] J. Deutscher, A. Davison, and I. Reid, "Automatic Partitioning of High Dimensional Search Spaces associated with Articulated Body Motion Capture," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Kauai, USA, 2001, pp. 669–676.

[16] M. Do, P. Azad, T. Asfour, and R. Dillmann, "Imitation of Human Motion on a Humanoid Robot using Nonlinear Optimization," in *International Conference on Humanoid Robots (Humanoids)*, Daejeon, Korea, 2008.

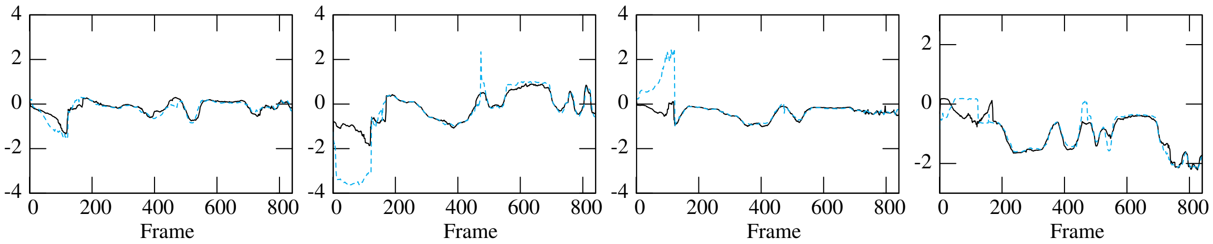

Fig. 10. Exemplary arm trajectory for the left arm acquired by the proposed human motion capture system. The solid line indicates the tracking result acquired at the full temporal resolution of 30 Hz; for the dashed line every second frame was skipped, i.e. 15 Hz. The angles θ_1–θ_4 are plotted from left to right. The units are in radians.

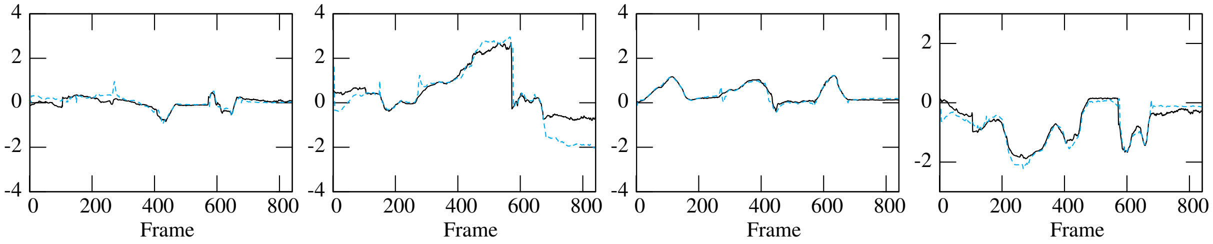

Fig. 11. Exemplary arm trajectory for the right arm acquired by the proposed human motion capture system. The solid line indicates the tracking result acquired at the full temporal resolution of 30 Hz; for the dashed line every second frame was skipped, i.e. 15 Hz. The angles θ_1–θ_4 are plotted from left to right. The units are in radians.

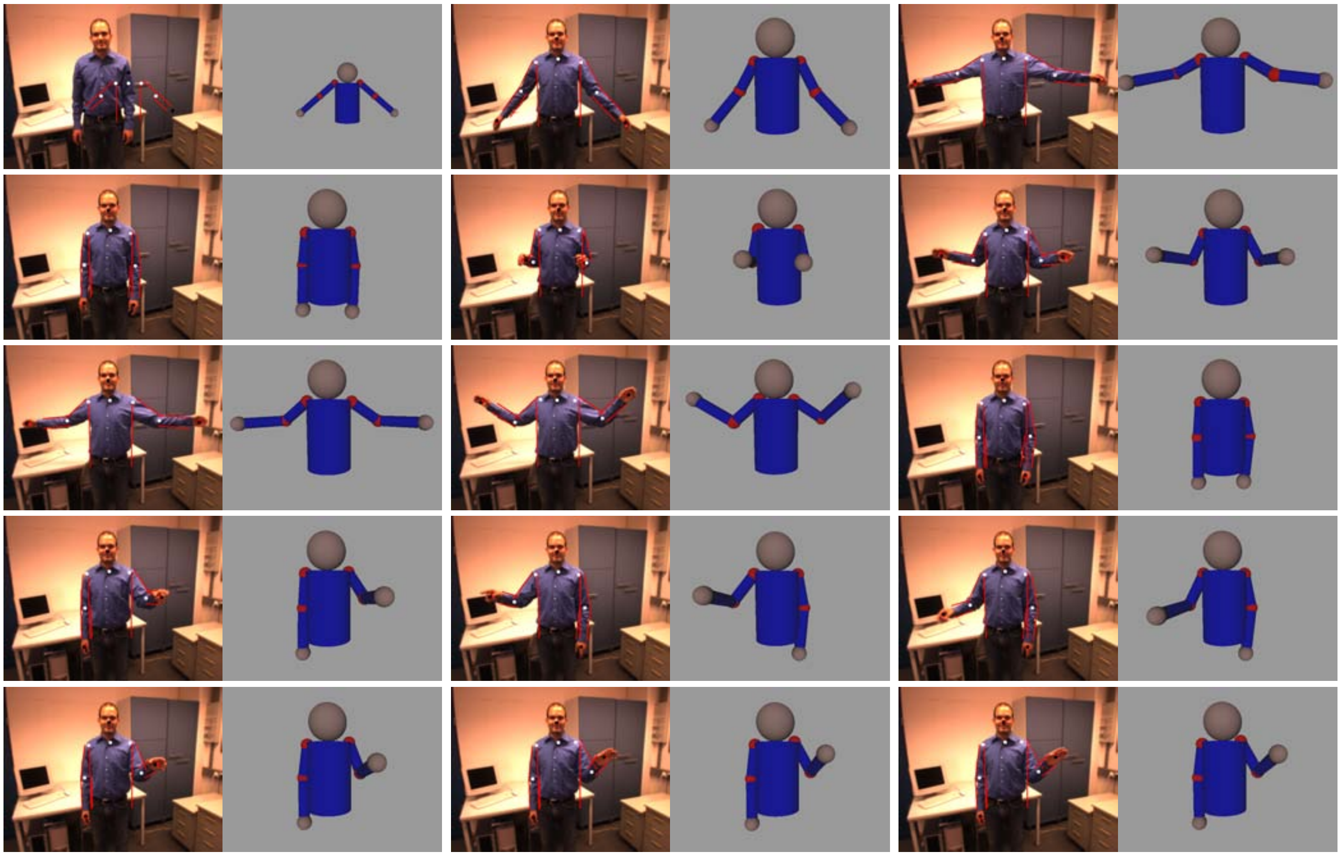

Fig. 12. Snapshots of the results computed for a test sequence consisting of 840 frames, which were captured at a frame rate of 30 Hz. Every 60th frame is shown; the frames are ordered row-wise from top left to bottom right. The red dots mark the measured positions computed by the hand/head tracking system. The black dots mark the corresponding positions according to the estimated model configuration. The first frame illustrates the initial state of the particle filter.