2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Stereo-based Markerless Human Motion Capture for Humanoid Robot Systems

Pedram Azad[1] Aleš Ude[2] Tamim Asfour[1] Rüdiger Dillmann[1]

Abstract— **In this paper, we present an image-based markerless human motion capture system, intended for humanoid robot systems. The restrictions set by this ambitious goal are numerous. The input of the system is a sequence of stereo image pairs only, captured by cameras positioned at approximately eye distance. No artificial markers can be used to simplify the estimation problem. Furthermore, the complexity of all algorithms incorporated must be suitable for real-time application, which is maybe the biggest problem when considering the high dimensionality of the search space. Finally, the system must not depend on a static camera setup and has to find the initial configuration automatically.**

We present a system, which tackles these problems by combining multiple cues within a particle filter framework, allowing the system to recover from wrong estimations in a natural way. We make extensive use of the benefit of having a calibrated stereo setup. To reduce search space implicitly, we use the 3D positions of the hands and the head, computed by a separate hand and head tracker using a linear motion model for each entity to be tracked. With stereo input image sequences at a resolution of 320×240 pixels, the processing rate of our system is 15 Hz on a 3 GHz CPU. Experimental results documenting the performance of our system are available in form of several videos.

I. Introduction

The idea of markerless human motion capture is to capture human motion without any additional arrangements required, by operating on image sequences only. Implementing such a system on a humanoid robot and thus giving the robot the ability to perceive human motion would be valuable for various reasons. Captured trajectories, which are calculated in joint angle space, can serve as a solid base for learning human-like movements. Commercial human motion capture systems such as the VICON system [1], which are popular both in the film industry and in the biological research field, require reflective markers and time consuming manual post-processing of captured sequences. A real-time human motion capture system using the image data acquired by the robot's head would make one big step toward autonomous online learning of movements. Another application for the data computed by such a system is the recognition of actions and activities, serving as a perceptive component for human-robot interaction. However, providing data for learning of movements and actions – often referred to as learning-by-imitation – is the more challenging goal, since transforming captured movements in configuration space into the robot's kinematics and reproducing them on the robot sets the higher demands to smoothness and accuracy.

[1]Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany
[2]Jozef Stefan Institute, Ljubljana, Slowenia

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 potentially move. Furthermore, computations have to be performed in real-time, preferably at 30 Hz or higher, in order to achieve optimal results.

The general problem definition is to find the correct configuration of the underlying articulated 3D human model for each input image respectively image tuple when using multiple cameras. The main problem is that search space increases exponentionally with the number of Degrees Of Freedom (DOF). A realistic model of the human body has at least 14 DOF if only modeling the upper body without the neck (3 DOF for each shoulder, 1 DOF for each elbow, 6 DOF for base translation and rotation), or 25 DOF for the full body (plus 3 DOF for each hip joint, 1 DOF for each knee, 3 DOF for the neck), leading to a very high-dimensional search space.

There are several approaches to solve the general problem of markerless human motion capture, differing in the sensors incorporated and the intended application. When using multiple cameras, i.e. three or more cameras located around the area of interest, two different systems have shown very good results. The one class of approaches is based on the calculation of 3D voxel data, as done by [2], [3]. The other approach is based on particle filtering and became popular by the work of Deutscher et al. [4]. Other approaches depend on incorporation of an additional 3D sensor and the *Iterative Closest Point* (ICP) algorithm, such as the Swiss Ranger, as presented by [5]. Other approaches concentrate on deriving as much information as possible from monocular image sequences [6], and reducing the size of the search space by applying restrictions to the range of possible movements, e.g. by incorporating a task-specific dynamic model [7]. Our experience is that it is not possible to build a general 3D human motion capture system using monocular image sequences only, since in many cases a single camera is not sufficient to determine accurate 3D information, based on the principle *depth through scaling*. A further strategy to reduce search space is search space decomposition i.e. performing a hierarchical search, as done by [8]. However, by doing this, the power of the system is limited, since in many cases the global view is needed to determine the correct configuration, e.g. for rotations around the body axis, the information provided by the positions of the arms is very helpful.

Recently, we have started to adapt and extend the particle

1-4244-0602-1/07/$20.00 ©2007 IEEE.

filter based system for real-time application on a humanoid robot head ([9], [10]), presenting our newest results in the following. Particle filtering has proven to be an applicable and robust technique for contour tracking in general ([11], [12], [13]), and for human motion capture in particular, as shown in [4], [6]. However, in particle filters, a larger search space requires a greater number of particles. One strategy to cope with this problem is to reduce the dimensionality of configuration space by restricting the range of the subject's potential movements, as already mentioned, or to approach a linear relationship between the dimension of configuration space and the size of the search space by performing a hierarchical search. A general but yet effective way to reduce the number of particles is based on the idea of *Simulated Annealing*, presented in [4], [14]. However, the final system, which uses three cameras at fixed positions in the corners of a room, requires on average 15 seconds to process one frame on a 1 GHz CPU [14].

Theoretically, an edge-based cue would be already sufficient to track the movements of a human – if using an adequate number of particles. To span the search space with a sufficient resolution when using an edge-based cue only, millions of particles would be necessary for a successful tracker. Therefore, the common approach using particle filters for human motion capture is to combine edge and region information within the likelihood function, which evaluates a given configuration matching the current observation. Although this is a powerful approach, the computational effort is relatively high. Especially the evaluation of the region based cue is computationally expensive.

Our strategy is to combine as many cues derivable from the input images as possible to reduce search space implicitly by achieving a faster convergence. We present a running system on our humanoid robot ARMAR using the benefits of a stereo setup and combining edge, region and skin color information. The initial configuration is found automatically – a necessity for any perceptive component of a vision system. The system is able to capture real 3D motion without using markers or manual post-processing. The processing rate of our algorithm is 15 Hz on a 3 GHz CPU using stereo input image sequences with a resolution of 320×240 pixels.

II. Using Particle Filters for Human Motion Capture

Particle filtering has become popular for various visual tracking applications – often also referred to as the Condensation Algorithm. The benefits of a particle filter compared to a Kalman filter are the ability to track non-linear movements and the property to store multiple hypotheses simultaneously. The price one has to pay for these advantages is the higher computational effort. The probability density function representing the likelihood of the configurations in configuration space matching the observations is modeled by a finite set of N particles $S = \{(\mathbf{s_1}, \pi_\mathbf{1}), ..., (\mathbf{s_N}, \pi_\mathbf{N})\}$, where $\mathbf{s_i}$ denotes one configuration and π_i the likelihood associated with it. The core of a particle filter is the likelihood function $p(\mathbf{z}|\mathbf{s})$ computing the a-posteriori probabilities π_i, where $\mathbf{s}$ denotes a given configuration and $\mathbf{z}$ the current observations i.e. the current image pair. This likelihood function must be evaluated for each particle for each frame i.e. $N \cdot f$ times per second. As an example this means for $N = 1000$ particles and $f = 30$ Hz $N \cdot f = 30000$ evaluations per second.

For resampling the particle set, N particles from the last generation are picked proportional to their probability. The configuration of each of these particles is used as a base to build a new configuration, incorporating a motion model and adding Gaussian noise. We use a first-order linear motion model together with adaptive Gaussian noise, whose amount is decreased or increased depending on the current errors. A detailed description about the use of particle filters for human motion capture can be found in [9].

A. Edge Cue

Given the projected edges of a configuration $\mathbf{s}$ of the human model and the current input image $\mathbf{z}$, the likelihood function $p(\mathbf{z}|\mathbf{s})$ for the edge cue calculates the a-posteriori probability that the configuration leading to the set of projected edges is the proper configuration i.e. the one taht best matches the gradient image.

The approach we use is to spread the gradients in the gradient image with a Gaussian filter or any other suitable operator and to sum the gradient values along a projected edge, as done in [4]. Assuming that the spread gradient map has been remapped between 0 and 1, the modified likelihood function can be formulated as:

$$p_g(\mathbf{z}|\mathbf{s}) \propto \exp\left\{ -\frac{1}{2\sigma_g^2 M_g} \sum_{m=1}^{M_g} (1 - g_m)^2 \right\}$$

where g_m denotes the remapped gradient value for the mth point.

B. Region Cue

The second cue commonly used is region-based, for which a foreground segmentation technique has to be applied. The segmentation algorithm to be picked is independent from the likelihood function itself. The most common approach is background subtraction. However, this segmentation method assumes a static camera setup and is therefore not suitable for application on a potentially moving robot head. Another option is to segment motion by using difference images or optical flow, but this method also assumes a static camera setup. It has to be mentioned that there are extensions of the basic optical flow algorithm that allow to distinguish real motion in the scene and ego-motion [15]. However, the problem with all motion-based methods – which does not include background subtraction – is that the quality of the segmentation result is not sufficient for a region-based cue. Only those parts of the image that contain edges or any other kind of texture can be segmented, and the silhouette of segmented moving objects often contains parts of the background, resulting in a relatively blurred segmentation result.

Having segmented the foreground in the input image, where foreground pixels are set to 1 and background pixels

are set to 0, the likelihood function commonly used can be formulated as [4]:

$$p_r(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_r^2 M_r}\sum_{m=1}^{M_r}(1-r_m)^2\right\} \quad (1)$$

where r_m denotes the segmentation value of the mth pixel from the set of pixels of all projected body part regions. Although this function can be optimized further, using the fact that $r_m \in \{0,1\}$, its computation is still rather inefficient. The bottleneck is the computation of the set of all M projected pixels together with reading the corresponding values from the segmentation map.

C. Fusion of Multiple Cues

The both introduced cues are fused by multiplying the two likelihood functions resulting in:

$$p_{g,r}(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2}\left(\frac{\sum_{m=1}^{M_g}(1-g_m)^2}{\sigma_g^2 M_g}+\frac{\sum_{m=1}^{M_r}(1-r_m)^2}{\sigma_r^2 M_r}\right)\right\}$$

Any other cue can be fused within the particle filter with the same rule. One way of combining the information provided by multiple cameras is to incorporate the likelihoods for each image in the exact same manner [4]. In our system, we additionally use 3D information which can be computed explicitly by knowing the stereo calibration. This separate cue is then combined with the other likelihoods with the same method, as will be described in Section III.

III. Multiple Cues in the proposed System

In this section, we want to introduce the cues our system is based on. Instead of the commonly used region-based likelihood function p_r, as introduced in Equation (1), we incorporate the result of foreground segmentation in a more efficient way, as will be introduced in Section III-A. In Section III-B we will present the results of studies regarding the effectivity of the introduced cues, leading to a new likelihood function. As already mentioned, we use the benefits of a stereo system in an additional explicit way, as will be introduced in III-C. The final combined likelihood function is presented in Section III-D.

A. Edge Filtering using Foreground Segmentation

When looking deeper into the region-based likelihood function p_r, one can state two separate abilities:

- Leading to a faster convergence of the particle filter
- Compensating the failure of the edge-based likelihood function in cluttered backgrounds

The first property is discussed in detail in Section III-B, and an efficient alternative is presented. The second property can be implemented explicitly by using the result of foreground segmentation directly to generate a filtered edge map, containing only foreground edge pixels. In general, there are two possibilities:

- Filtering the gradient image by masking out background pixels with the segmentation result
- Calculating gradients on the segmentation result

While the first alternative preserves more details in the image, the second alternative computes a sharper silhouette. Furthermore, in the second case, gradient computation can be optimized for binarized input images, which is why we currently use this approach. As explained in Section II-B, the only commonly used foreground segmentation technique is background subtraction, which we do not intend to use, since the robot head can potentially move. It has to be mentioned that taking into account that the robot head can move is not a burden, but there are several benefits of using an active head, which will be discussed in Section VI. As an alternative to using background subtraction, we are using a solid colored shirt, which allows us to perform tests practically anywhere in our lab. Since foreground segmentation is performed in almost any markerless human motion capture system, we do not restrict ourselves compared to other approaches, but only trade in the restriction of wearing a colored shirt for the need of having a completely static setup. Experiments have shown that the segmentation result using a colored shirt leads to a slightly smoother output of the human motion capture system compared to background subtraction. However, if the shirt color is not to be used, background subtraction still results in a robust system.

B. Cue Comparison and Distance Likelihood Function

In order to understand the benefits and drawbacks of each likelihood function and thus getting a feeling of what a likelihood function can accomplish and what not, it is helpful to measure their effectivity in a simple one-dimensional example. The experiment we have used in simulation is tracking a square of fixed size in 2D, which can be further simplified to tracking the intersection of a square with a straight line along the straight line i.e. in one dimension.

The model of the square to be tracked is defined by the midpoint (x, y) and the edge length k, where y and k are constant and x is the one dimensional configuration to be predicted. In [10], we have compared three different likelihood functions separately: the gradient-based cue p_g, the region-based cue p_r, and a third cue p_d, which is based on the Euclidian distance:

$$p_d(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_d^2}|f(\mathbf{s})-\mathbf{c}|^2\right\}$$

where $\mathbf{c}$ is an arbitrary dimensional vector which has been calculated previously on the base of the observations $\mathbf{z}$, and $f : R^{\dim(\mathbf{s})} \rightarrow R^{\dim(\mathbf{c})}$ is a transformation mapping a configuration $\mathbf{s}$ to the vector space of $\mathbf{c}$. In our example, $\mathbf{c}$ denotes the midpoint of the square in the observation $\mathbf{z}$, $\dim(\mathbf{s}) = \dim(\mathbf{c}) = 1$, and $f(\mathbf{s}) = \mathbf{s}$. For efficiency considerations, we have used the squared Euclidian distance, practically resulting in the SSD. Evidently, in this simple case, there is no need to use a particle filter for tracking, if the configuration to be predicted $\mathbf{c}$ can be determined directly. However, in this example, we want to show the characteristic properties of the likelihood function p_d, in order to describe the performance in the final likelihood function of the human motion capture system, presented in the sections III-C and

III-D. In the update step of the particle filter, we applied Gaussian noise only, with an amplification factor of $\omega = 3$. The task was to find a static square with $k = 70$, based on the pixel data at the intersection of the square with the x-axis. The number of iterations needed depending on the initial distance to the goal is given in Figure 1 for each likelihood function. While with starting points in a close neighborhood of the goal the gradient cue leads to the fastest convergence, the region cue and the distance cue converge faster the farther the starting point is away from the goal. The results of the comparison of the cues is explained in detail in [10].

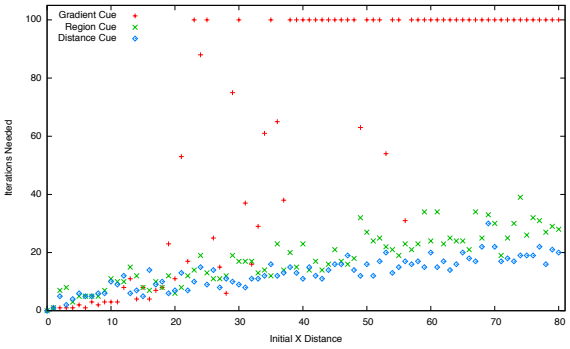

Fig. 1. Comparison of iteration numbers: an iteration number of 100 indicates that the goal was not found

As a conclusion, one can state that whenever possible to determine a discrete point directly, it is the best choice to use the likelihood function p_d rather than p_r. A second drawback of the region cue is explained in Section III-D. While it is not possible to do a successful tracking without the edge cue – especially when scaling has to be taken into account – it is also not possible to rely on the edge cue only. The higher the dimensionality of search space is, the more drastic the lack of a sufficient number of particles becomes. Thus, in the case of human motion capture with dimensions of 14 and greater, the configurations will never perfectly match the image observations. Note, that the simulated experiment examined a static case. In the dynamic case, the robustness of the tracker is always related to the frame rate at which images are captured and processed, and to the speed of the subject's movements. In the next section, we show how the likelihood function p_d is incorporated into our system using 3D points, leading to a significant implicit reduction of the search space.

C. Using Stereo Information

There are various ways to use stereo information in a vision system. One possibility is to calculate depth maps, however, the quality of depth maps is in general not sufficient and only rather rough information can be derived from them. Another option in a particle filter framework is to project the model into both the left and the right image and evaluate the likelihood function for both images and multiply the the resulting likelihoods, as already mentioned in Section II-C. This approach can be described as *implicit stereo*. A third alternative is to determine correspondences for specific features in the image pair and calculate the 3D position for each match explicitly by triangulation.

In the proposed system, we use both implicit stereo and stereo triangulation. As features we use the hands and the head, which are segmented by color and matched in a pre-processing step. Thus, the hands and the head can be understood as three natural markers. The image processing line for determining the positions of the hands and the head in the input image is described in Section IV.

There are two alternatives to use the likelihood function p_d together with skin color blobs: apply p_d in 2D for each image separately and let the 3D position be calculated implicitly by the particle filter, or apply p_d in 3D to the triangulated 3D positions of the matched skin color blobs. We have experienced that the first approach does not lead to a robust acquisition of 3D information. This fact is not surprising, since in a high dimensional space the mismatch between the number of particles used and the size of the search space is more drastic. This leads, together with the fact that the prediction result of the likelihood function p_d is noisy within an area of 1-2 pixels already in a very simple experiment, to a considerable error in the implicit stereo calculation in the real scenario. For this reason we apply p_d in 3D to the triangulation result of matched skin color blobs. By doing this, the particle filter is forced to always move the peak of the probability density function toward configurations in which the positions of the hands and the head from the model are very close to the real 3D positions, which have been determined on the base of the image observations.

D. Final Likelihood Function

In the final likelihood function, we use two different components: the edge cue based on the likelihood function p_g, and the distance cue based on the likelihood function p_d, as explained in the sections III-B and III-C. The region cue is left out for two reasons: it reduces the efficiency of the system significantly, and it can diminish the accuracy of the estimation. The reason for this is that the region cue is less precise than the edge cue; in many cases it evaluates wrong configurations with a similar or even equal likelihood as it does for the proper configuration, as illustrated in Figure 2.

Fig. 2. Illustration of a wrong and a proper configuration, which are assigned the same likelihood by the region cue

We have experienced that when leaving out the square in Equation (II-A), i.e. calculating the Sum of Absolute Differences (SAD) instead of the Sum Of Squared Differences (SSD), the quality of the results remains the same for our application. In this special case one can optimize p_g further,

resulting in:

$$p'_g(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_g^2}(1-\frac{1}{M_g}\sum_{m=1}^{M_g} g_m)\right\}$$

For a system intended for real-time application, we have decided to replace the region-based cue based on p_r completely by the distance cue based on p_d. In order to formulate the distance cue, first the function $d_i(\mathbf{s},\mathbf{c})$ is defined as:

$$d_i(\mathbf{s},\mathbf{c}) := \begin{cases} |f_i(\mathbf{s})-\mathbf{c}|^2 & : \quad \mathbf{c} \neq \mathbf{0} \\ 0 & : \quad \text{otherwise} \end{cases}$$

where $n := \dim(\mathbf{s})$ is the number of DOF of the human modal, $\dim(\mathbf{c}) = 3$, $i \in \{1,2,3\}$ to indicate the function for the left hand, right hand or the head. The transformation $f_i : R^n \rightarrow R^3$ transforms the n-dimensional configuration of the human model into the 3D position of the left hand, right hand or head respectively, using the forward kinematics of the human model. The likelihood function for the distance cue is then formulated as:

$$p'_d(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_d^2}(d_1(\mathbf{s},\mathbf{c_1})+d_2(\mathbf{s},\mathbf{c_2})+d_3(\mathbf{s},\mathbf{c_3}))\right\}$$

where the vectors $\mathbf{c_i}$ are computed on the base of the image observations $\mathbf{z}$ using skin color segmentation and stereo triangulation, as explained in Section III-C. If the position of a hand or the head can not be determined because of occlusions or any other disturbance, the corresponding vector $\mathbf{c_i}$ is set to the zero vector. Note that this does not falsify the resulting probability density function in any way. Since all likelihoods of a generation k are independent from the likelihoods calculated for any previous generation, the distribution for each generation is also independent. Thus, it does not make any difference that in the last image pair one $\mathbf{c_i}$ was present, and in the next image pair it is not. The final likelihood function is the product of p'_g and p'_d:

$$p(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma_d^2}\sum_{i=1}^{3} d_i(\mathbf{s},\mathbf{c_i}) + \frac{1}{\sigma_g^2}(1-\frac{1}{M_g}\sum_{m=1}^{M_g} g_m)\right)\right\}$$

IV. Image Processing Pipeline and Tracking of Hands and Head

The image processing pipeline transforms each image of the stereo pair into a skin color map and a gradient map, which are then used by the likelihood function presented in Section $III-D$. In Figure 3, the pipeline is shown for one image; in the system, the pipeline is applied twice: once for each image of the stereo pair. After the input images are smoothed with a 3×3 Gaussian kernel, the HSV image is computed. The HSV image is then filtered twice, once for skin color segmentation and once for foreground segmentation by segmenting the shirt color. A simple combination of a 2×1 and a 1×2 gradient operator is applied to the segmented foreground image, which is sufficient and the most efficient for a binarized image. Finally, a gradient pixel map is generated by blurring the gradient image, as done in [4].

Currently, the hands and the head are segmented using a fixed interval color model in HSV color space. Similar to the idea presented in [16], each color blob is tracked with a linear motion model, which makes it possible to track through images in which blobs are occluded or blobs fall together. In these images, the state of the affected blobs from the last image is propagated to the current image, predicting their current position by using the linear motion model. The resulting color blobs are matched between the left and right image, taking into account their size, the ratio between the height and width of the bounding box, and the epipolar geometry. By doing this, false regions in the background can be discarded easily. Finally, the centroids of matched regions are triangulated using the parameters of the calibrated stereo setup.

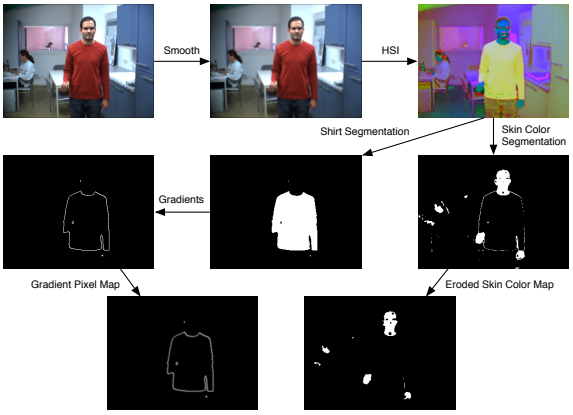

Fig. 3. Visualization of the image processing line

All image processing routines and mathematical computations are implemented using the *Integrating Vision Toolkit* (IVT) [17]. It is an Open Source vision library, offering a clean interface to image devices and camera calibration, and a variety of filters, segmentation methods, stereo routines, mathematical function and data structures.

V. Experimental Results

The experiments being presented in this section were performed on the humanoid robot ARMAR. In the robot head, two Dragonfly cameras are positioned at a distance of approximately eleven centimeters. As input for the image processing line, we used a resolution of 320×240, captured at a frame rate of 25 Hz. The particle filter was run with a set of $N = 1000$ particles. The computation times for one image pair, processed on a 3 GHz CPU, are listed in Table I. As one can see, the processing rate of the system is 15 Hz, which is not yet real-time for an image sequence captured at 25 Hz, but very close. Of course, when moving slowly, a processing rate of 15 Hz is sufficient.

In Figure 4, six screenshots are shown which demonstrate how the system automatically initializes itself. No initial configuration is given; it autonomously finds the only possible configuration matching the observations. Figure 5 shows four screenshots of the same video sequence, showing the performance of the human motion capture system tracking a punch with the left hand. We have also run successful experiments with a half a minute sequence with a resolution of 640×480 captured at 57 Hz.

	Time [ms]
Image Processing Line	14
1000 Forward Kinematics and Projection	23
1000 Evaluations of Likelihood Function	29
Total	**66**

TABLE I

PROCESSING TIMES WITH $N = 1000$ PARTICLES ON A 3 GHZ CPU USING A 320×240 STEREO INPUT IMAGE SEQUENCE

Fig. 4. Screenshots showing automatic initialization

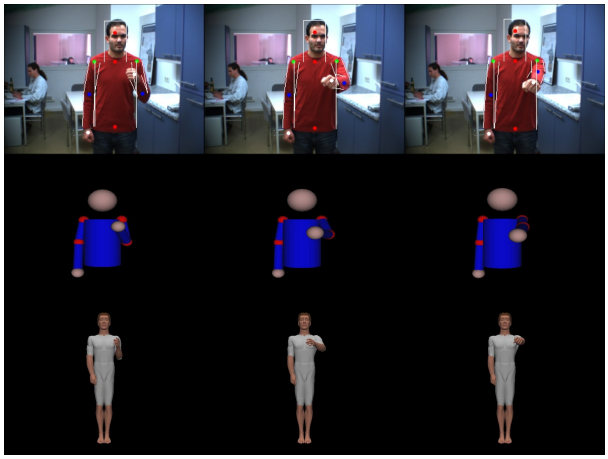

Fig. 5. Screenshots showing tracking performance. Top: input images of the left camera with projected estimation. Middle: Visualization with a simple 3D model. Bottom: Visualization by mapping to a realistic human model

VI. CONCLUSION

We have presented an image-based markerless human motion capture system for application on a humanoid robot. The system is capable of computing motion trajectories in the configuration space of a human body. We presented our strategy for fusing multiple cues within the particle filter. The proposed strategy is supported by the results of a study examining the properties of commonly used image cues and the newly introduced distance cue. We showed that by using the 3-D distance cue we could find optimal configuration withs less particles than standard approaches. This implicit reduction of the search space allows us to capture human motion with a particle filter using as few as 1000 particles, which results in a processing rate of 15 Hz on a 3 GHz CPU for a 320×240 stereo input video stream.

In the near future, we plan to extend the system by incorporating the legs and feet into the human model. Furthermore, we intend to make use of the active head to follow the human subject, thus keeping her/him in the robot's field of view. This would not be possible with a static head.

ACKNOWLEDGMENT

The work described in this paper was partially conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) and funded by the European Commission and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft). We also would like to thank Gordon Cheng of ATR/JST for his comments and suggestions.

REFERENCES

[1] "Vicon Peak," http://www.vicon.com.
[2] F. Caillette and T. Howard, "Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction," in *British Machine Vision Conference*, vol. 2, Kingston, UK, 2004, pp. 597–606.
[3] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human Body Model Acquisition and Tracking using Voxel Data," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 199–223, 2003.
[4] J. Deutscher, A. Blake, and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," in *Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, USA, 2000, pp. 2126–2133.
[5] 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.
[6] H. Sidenbladh, "Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, 2001.
[7] K. Rohr, "Human movement analysis based on explicit motion models," *Motion-Based Recognition*, pp. 171–198, 1997.
[8] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, 1996, pp. pp. 73–80.
[9] P. Azad, A. Ude, R. Dillmann, and G. Cheng, "A Full Body Human Motion Capture System using Particle Filtering and On-The-Fly Edge Detection," in *International Conference on Humanoid Robots (Humanoids)*, Santa Monica, USA, 2004.
[10] 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.
[11] A. Blake and M. Isard, *Active Contours*. Springer, 1998.
[12] J. MacCormick, "Probabilistic models and stochastic algorithms for visual tracking," Ph.D. dissertation, University of Oxford, UK, 2000.
[13] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *European Conference Computer Vision (ECCV)*, Dublin, Ireland, 2000, pp. 3–19.
[14] J. Deutscher, A. Davison, and I. Reid, "Automatic Partitioning of High Dimensional Search Spaces associated with Articulated Body Motion Capture," in *Computer Vision and Pattern Recognition (CVPR)*, Kauai, USA, 2001, pp. 669–676.
[15] K. Wong and M. Spetsakis, "Motion Segmentation and Tracking," in *International Conference on Vision Interface*, Calgary, Canada, 2002, pp. 80–87.
[16] A. A. Argyros and M. I. Lourakis, "Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera," in *European Conference on Computer Vision (ECCV)*, vol. 3, Prague, Czech Republic, 2004, pp. 368–379.
[17] P. Azad, "Integrating Vision Toolkit," http://ivt.sourceforge.net.