Image-based Markerless 3D Human Motion Capture using Multiple Cues

Pedram Azad[1], Ales Ude[2], Tamim Asfour[1], Gordon Cheng[3], and Ruediger Dillmann[1]

[1] Institute for Computer Science and Engineering, University of Karlsruhe, Germany
azad|asfour|dillmann@ira.uka.de
[2] Jozef Stefan Institute, Ljubljana, Slowenia
ales.ude@ijs.si
[3] Computational Neuroscience Laboratories, ATR, Kyoto, Japan
gordon@atr.jp

1 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, 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. Therefore, such systems can only provide data for highly supervised offline learning algorithms. In contrast, 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 – 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.

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 a frame rate of 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. 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 25 DOF, or 17 DOF if only modeling the upper body, leading in both cases 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 [5, 13]. The other approach is based on particle filtering and became popular by the work of Deutscher et al. [6]. Recently, we have started to adapt and extend this system for real-time application on a humanoid robot head [3], presenting the newest results in the following. 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 [10]. However, for this system, the goal is not to acquire smooth trajectories but to classify the activities of a human into categories, such as walking, waving, bowing, etc. Other approaches concentrate on deriving as much information as possible from monocular image sequences [15], 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 [14]. Our experience is that it is not possible to build a general 3d human motion capture system, 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.

We use the Bayesian framework *Particle Filtering* to compute the probability distribution of the current configuration, as described in detail in [3]. Particle filtering, also known as the *Condensation Algorithm* in the context of visual tracking, as introduced in [9], has proven to be an applicable and robust technique for contour tracking in general [4] [11] [12], and for human motion capture in particular, as shown in [6] [15].

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 [6, 7]. 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 PIII CPU [7].

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 higher convergence of the probability distribution. 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 with a high smoothness and accuracy for a purely vision based algorithm, without using markers or manual post-processing. The processing rate of our algorithm is 15 Hz on a 3 GHz CPU.

2 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 distribution 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 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. A detailed description of using particle filters for human motion capture can be found in [3].

2.1 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 likelihood that the configuration leading to the set of projected edges is the proper configuration i.e. matching the gradient image the most. The basic

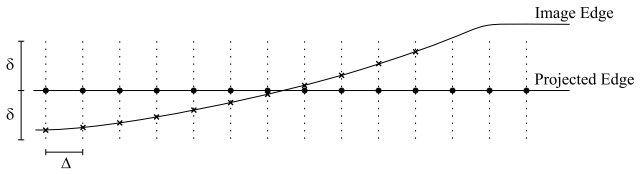

Fig. 1. Illustration of the search of edges

technique is to traverse the projected edges and search at fixed distances Δ for high-contrast features perpendicular to the projected edge within a fixed search distance δ (in each direction) i.e. finding edge pixels in the camera image, as illustrated in figure 1 [9]. For this purpose, usually the camera image is preprocessed to generate an edge image using a gradient based edge detector. The likelihood is calculated on the base of the Sum of Squared Differences (SSD). For convenience of notation, it is assumed that all edges are contained in one contiguous spline with $M = L/\Delta$ discretizations, where L denotes the sum of the length of all projected edges in the current image. The distance at which an edge feature has been found for the mth point is denoted as d_m and μ denotes a constant maximum error which is applied in case no edge feature could be found. The likelihood function can then be formulated as:

$$p(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma^2 M}\sum_{m=1}^{M} f(d_m, \mu)\right\} \tag{1}$$

where $f(\nu, \mu) = \min(\nu^2, \mu^2)$. Another approach 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 [6], rather than performing a search perpendicular to each pixel of the projected edge. By doing this, the computational effort is reduced significantly, even when picking the highest possible discretization of $\Delta = 1$ pixel. Furthermore, one does not have to make the non-trivial decision which gradient pixel to take for each pixel of the projected edge. Assuming that the spread gradient map has been remapped between 0 and 1, the modified likelihood function can be formulated as:

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

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

2.2 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. Both methods also assume 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 [16]. 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 [6]:

$$p_r(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_r^2 M_r}\sum_{m=1}^{M_r}(1-r_m)^2\right\} \tag{3}$$

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.

2.3 Fusion of Multiple Cues

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

$$p_{g,r}(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2}\left(\frac{\sum_{m=1}^{M_g}(1-g_m)^2}{\sigma_g^2 M_g} + \frac{\sum_{m=1}^{M_r}(1-r_m)^2}{\sigma_r^2 M_r}\right)\right\} \tag{4}$$

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 [6]. 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 3.

3 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 (3), we incorporate the result of foreground segmentation in a more efficient way, as will be introduced in Section 3.1. In Section 3.2 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 3.3. The final combined likelihood function is presented in Section 3.4.

3.1 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 3.2, 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 2.2, the only commonly used foreground segmentation technique is background subtraction, which we cannot 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 7. 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. We want to point out that the proposed generation of a filtered edge map does not depend on the segmentation technique.

3.2 Cue Studies and Distance Likelihood Function

In order to understand which are the benefits and drawbacks of each likelihood function and thus getting a feeling of what a likelihood function can do and what not, it is helpful to take a look at the corresponding probability distributions in a simple one-dimensional example. The experiment we use in simulation is tracking a square of fixed size in 2d, which can be simplified furthermore 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 the following, we want to compare 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:

$$p_d(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_d^2}|f(\mathbf{s}) - \mathbf{c}|^2\right\} \tag{5}$$

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 3.3 and 3.4. For the experiments, we used $N = 15$ particles and picked $\sigma_g = \sigma_r = \sqrt{5}$ and $\sigma_d = 0.1$. 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. As one can see in Figure 2, the gradient-based likelihood function p_g produces the narrowest distribution. The probability distributions produced by p_r and p_d are relatively similar; their narrowness can be adjusted by varying σ_r respectively σ_d. The effect of each distribution can be seen in Figure 3. 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. In the figures 4-6, the initial distance from the goal Δx_0 was varied. As expected, p_g leads to the fastest and smoothest convergence for $\Delta x_0 = 5$. $\Delta x_0 = 15$ is already close to the border of the convergence radius for p_g; the particle filter first tends to the wrong direction and then finally converges to the goal position. With $\Delta x_0 = 80$, it is by far impossible for p_g to find the global maximum, it converges to the (wrong) local maximum, matching the right edge of the model with the left

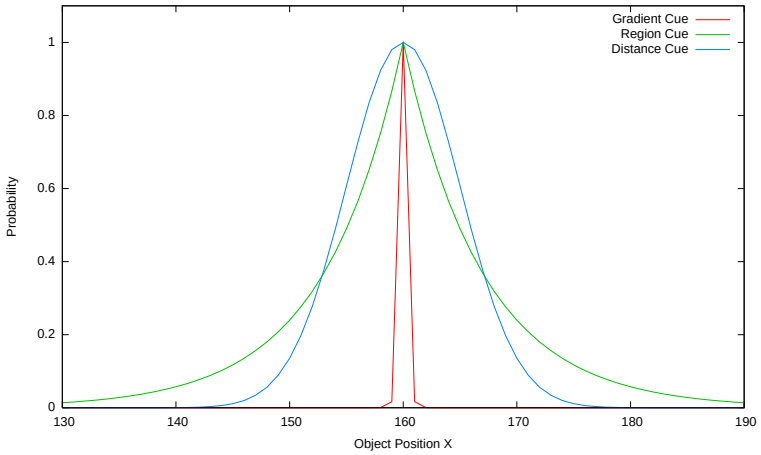

Fig. 2. Comparison of Probablity Distributions

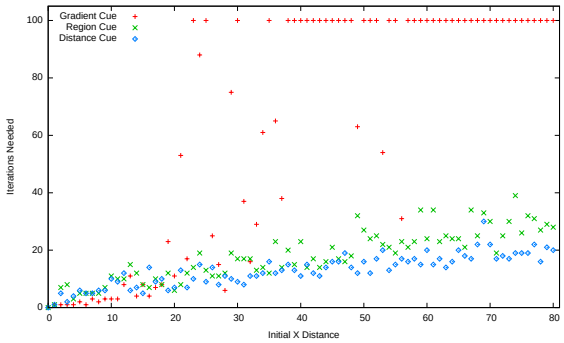

Fig. 3. Comparison of iteration numbers: an iteration number of 100 indicates that the goal was not found

edge of the square in the image. For $\Delta x_0 = 5$ and $\Delta x_0 = 15$, p_r and p_d behave quite similar. However, for $\Delta x_0 = 80$, p_d converges significantly faster, since it has the global view at any time. In contrast, p_r has to approach the goal slowly to reach the area, in which it can converge fast. 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. 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 17 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

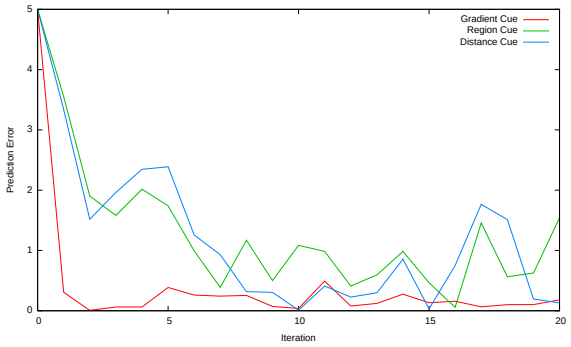

Fig. 4. Comparison of convergence for $\Delta x_0 = 5$

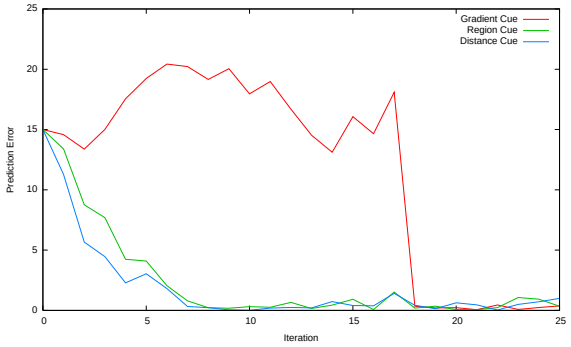

Fig. 5. Comparison of convergence for $\Delta x_0 = 15$

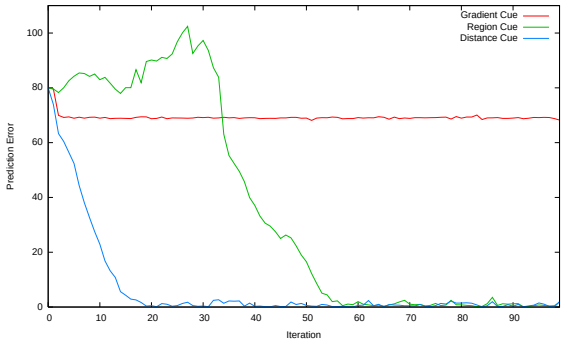

Fig. 6. Comparison of convergence for $\Delta x_0 = 80$

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 in our system in 3d, leading to a significant implicit reduction of the search space.

3.3 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 2.3. 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 preprocessing 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 4.

In principal, 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 circumstance 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 the in Figure 4 the prediction result of the likelihood function p_d is noisy within an area of 1-2 pixels in a very simple experiment, to a considerable error of the implicit stereo calculation in the real scenario. The accuracy of stereo triangulation decreases with the distance from the camera in a squared relationship. In order to observe the complete upper body of a human, the subject has to be located at a distance of at least 2-3 meters from the camera head. Thus, a potential error of two or more pixels in each camera image can lead to a significant error of the triangulation result. For this reason, in the proposed system, 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 distribution 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.

3.4 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 3.2 and 3.3. We have experienced that when leaving out the square in Equation (2), 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:

$$p'_g(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_g^2}(1-\frac{1}{M_g}\sum_{m=1}^{M_g} g_m)\right\} \tag{6}$$

For a system capable of 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. As our experimental results show, and as expected by the studies from Section 3.2, by doing this, a relatively small set of particles is sufficient for a successful system. The distance cue drags the peak of the distribution into a subspace in which the hands and the head are located at the true positions. Thus, search space is reduced implicitly, practically leaving the choice in this subspace to the cooperating gradient cue, based on the likelihood function p'_g. In order to formulate the distance cue, first the function $d_i(\mathbf{s}, \mathbf{c})$ is defined as:

$$d_i(\mathbf{s}, \mathbf{c}) := \begin{cases} |f_i(\mathbf{s}) - \mathbf{c}|^2 & : \quad \mathbf{c} \neq \mathbf{0} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{7}$$

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. Furthermore:

$$g(\mathbf{c}) := \begin{cases} 1 & : \quad \mathbf{c} \neq \mathbf{0} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{8}$$

The likelihood function for the distance cue is then formulated as:

$$p'_d(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2\sigma_d^2}\frac{d_1(\mathbf{s}, \mathbf{c_1}) + d_2(\mathbf{s}, \mathbf{c_2}) + d_3(\mathbf{s}, \mathbf{c_3})}{g(\mathbf{c_1}) + g(\mathbf{c_2}) + g(\mathbf{c_3})}\right\} \tag{9}$$

where the vector $\mathbf{c_i}$ are computed on the base of the image observations $\mathbf{z}$ using skin color segmentation and stereo triangulation, as explained in Section 3.3. 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 distribution 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:

$$p(\mathbf{z}|\mathbf{s}) \propto \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma_d^2}\sum_{i=1}^{3}\frac{d_i(\mathbf{s}, \mathbf{c_i})}{g(\mathbf{c_i})} + \frac{1}{\sigma_g^2}(1-\frac{1}{M_g}\sum_{m=1}^{M_g} g_m)\right)\right\} \tag{10}$$

4 Image Processing Line

The image processing line is a pipeline, transforming the input image pair into a skin color map and a gradient map, which are then used by the likelihood function presented in Section 3.4. In Figure 7, the image processing line for one image is shown; in the system the pipeline is applied twice: once for the left and once for the right input image. After the input images are smoothed with a 3×3 Gaussian kernel, the HSI image is computed. The HSI image is then filtered twice, once for skin color segmentation and once for foreground segmentation by segmenting the shirt color. A simple 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 applying a 3×3 or 5×5 Gaussian kernel, as done in [6]. Currently, the hands

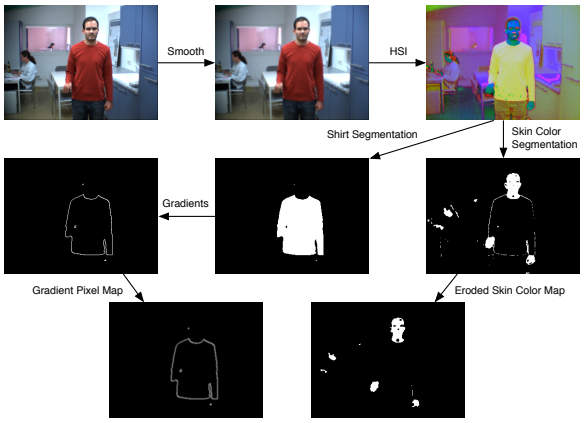

Fig. 7. Visualization of the image processing line

and the head are segmented using a fixed interval color model in HSI color space. The resulting color blobs are matched, 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. As will be discussed in Section 7, we are currently working on implementing a more sophisticated hand-/head-tracking system, which allows to deal with occlusions of skin colored regions.

5 Integrating Vision Toolkit

The complete system has been implemented using the *Integrating Vision Toolkit* (IVT) extensively [2]. With the IVT, the complete image processing

line presented in Section 4 could be implemented in less than 50 lines of code. The IVT provides a clean interface to capture devices of any kind, providing a convenient application for stereo camera calibration based on the OpenCV. For implementing Graphical User Interfaces, QT is integrated optionally, as well as the OpenCV library for image processing routines which are not yet available. The library is implemented in an easy-to-use software architecture, hiding all dependencies behind clean interfaces. The IVT fully supports the operating systems Linux, Mac OS and Windows. The project is available on Sourceforge; the link is included in the References.

6 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 1. 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, if moving more slowly, a processing rate of 15 Hz is sufficient. The relationship between the speed of the movements to be tracked and the frame rate at which the images are captured (and for real-time application processed) is briefly discussed in Section 7. In Figure

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

Table 1. Processing times with $N = 1000$ particles on a 3 GHz CPU

8, six screenshots are shown which show how the system automatically initializes itself. No configuration is told the system; it autonomously finds the only possible configuration matching the observations. Figure 9 shows four screenshots of the same video sequence, showing the performance of the human motion capture system tracking a punch with the left hand. The corresponding video and videos of other sequences can be downloaded from `http://i61www.ira.uka.de/users/azad/videos`.

Fig. 8. Screenshots showing automatic initialization

Fig. 9. Screenshots showing tracking performance

7 Discussion

We have presented an image-based markerless human motion capture system for real-time application. The system is capable of computing very smooth and accurate trajectories in configuration space for such a system. We presented our strategy of multi-cue fusion within the particle filter, and showed the results of studies examining the properties of the cues commonly used and a further distance cue. We showed that by using this distance cue combined with stereo vision, which has not yet been used in markerless human motion capture, we could reduce the size of the search space implicitly. This reduction of search space allows us to capture human motion with a particle filter using as few as 1000 particles with a processing rate of 15 Hz on a 3 GHz CPU. We plan to investigate and implement several improvements of the system:

- Currently, the subsystem for detection of the hands and the head in the images is not powerful enough to deal with occlusions of skin-colored regions in the image. To overcome this problem, we are currently working on implementing a more sophisticated hand and head tracking system, as presented by Argyros et al. [1]. By doing this, we expect the system to be

able to robustly track long and complicated sequences, since it will not be required to try to avoid occlusions between the hands and the hand.

- For any kind of tracking, the effective size of the search space increases exponentially with the potential speed of the movements respectively decreases exponentially with the frame rate at which images are captured. For this reason, the human motion capture systems with the most convincing results use a framerate of 60 Hz or higher, as done by [6]. Commercial marker-based tracking systems use a frame rate of 100 Hz up to 400 Hz and higher, to acquire smooth trajectories. For this reason, we want to perform further tests with the new Dragonfly2 camera, which is capable of providing the same image data as the Dragonfly camera, but at a frame rate of 60 Hz instead of 30 Hz.
- In the theory of particle filters, there exist several methods to decrease the effectively needed number of particles by modification of the standard filtering algorithm. For this purpose, we want to investigate the work on Partitioned Sampling [12] and Annealed Particle Filtering [6].
- We plan to extend the human model by incorporating the legs and feet into the human model. Especially for this purpose, we want to use the benefits of an active head, since with a static head it is hardly possible to have the complete human in the field of vision of the robot at one time step.

To our best knowledge, the proposed system is the first purely image-based markerless human motion capture system designed for a robot head which can track human movements with such accuracy and smoothness, and being suitable for real-time application at the same time. The system does not assume a static camera in any way; future work will also concentrate on running experiments using this benefit of being able to capture human motion while tracking the subject actively.

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).

References

1. A. A. Argyros and M. I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision (ECCV)*, volume 3, pages 368–379, Prague, Czech Republic, 2004.
2. P. Azad. Integrating Vision Toolkit. `http://ivt.sourceforge.net`.

3. 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.
4. A. Blake and M. Isard. *Active Contours*. Springer, 1998.
5. F. Caillette and T. Howard. Real-time markerless human body tracking with multi-view 3-d voxel reconstruction. In *British Machine Vision Conference*, volume 2, pages 597–606, Kingston, UK, 2004.
6. J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2133, Hilton Head, USA, 2000.
7. 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)*, pages 669–676, Kauai, USA, 2001.
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)*, pages pp. 73–80, San Francisco, USA, 1996.
9. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
10. 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.
11. J. MacCormick. *Probabilistic models and stochastic algorithms for visual tracking*. PhD thesis, University of Oxford, UK, 2000.
12. J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference Computer Vision (ECCV)*, pages 3–19, Dublin, Ireland, 2000.
13. I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.
14. K. Rohr. Human movement analysis based on explicit motion models. *Motion-Based Recognition*, pages 171–198, 1997.
15. H. Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.
16. K. Wong and M. Spetsakis. Motion segmentation and tracking. In *International Conference on Vision Interface*, pages 80–87, Calgary, Canada, 2002.