2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Markerless Human Motion Tracking with a Flexible Model and Appearance Learning

Florian Hecht, Pedram Azad and Rüdiger Dillmann

***Abstract*— A new approach to the 3D human motion tracking problem is proposed, which combines several particle filters with a physical simulation of a flexible body model. The flexible body model allows the partitioning of the state space of the human model into much smaller subsets, while finding a solution considering all the partial results of the particle filters. The flexible model also creates the necessary interaction between the different particle filters and allows effective semi-hierarchical tracking of the human body. The physical simulation does not require inverse kinematics calculations and is hence fast and easy to implement. Furthermore the system also builds an appearance model on-the-fly which allows it to work without a foreground segmentation. The system is able to start tracking automatically with a convenient initialization procedure. The implementation runs with 10 Hz on a regular PC using a stereo camera and is hence suitable for Human-Robot Interaction applications.**

I. INTRODUCTION

Finding and tracking the posture of a person over time is fundamental to several applications. It is used extensively in the animation industry to capture the performance of actors for films and computer games. The pose of a person is also very important in Human-Robot Interaction (HRI). When humans and robots interact, it is expected that the robots can understand the human body language, that is, they should be able to recognize certain actions such as waving or pointing. Also to teach a robot new actions it would be very helpful, if the actions could be taught by demonstration, where the robot learns the specific motions by observing the human performing them. To do all that the robot needs to have a notion of the body posture of the persons it is interacting with.

Motion capture applications in films and games use a large number of high-speed cameras in a studio environment to capture the performance of an actor in a tight suit with carefully placed markers. This expensive and complicated setup allows very precise measurements of the performed motions. For the application on a robot we cannot require the person being tracked to wear special clothing with markers or sensors. The robot also has only a limited view with a mono or stereo-camera and not a set of conveniently placed cameras around the person. With these restrictions it becomes much harder to track the movements of a person.

It would be beneficial to have sophisticated camera-based tracking systems that have only few requirements. Such a system should work with few regular cameras and should not require any preparations, neither of the person to be tracked nor of the environment. Such a system would be a lot cheaper and would not be restricted to certain locations. A person could work with such a system without special preparation or training. Ideally, such systems should have good performance in the following criteria at the same time: Robustness, precision and computational effort.

A. Previous Work

There is a plethora of different approaches to human motion capture that differ in the sensors used (accelerometers, cameras, depth cameras), the number of sensors, the model that is constructed (2D, 3D, with appearance, etc.) and the underlying algorithms. Since the proposed method uses particle filters and a 3D model with a single perspective view of the person, we will focus on previous work in these areas.

The use of short-baseline stereo cameras gives some additional 3D information about the scene [1], [8], [9] compared to a single camera, but has the same difficulties as monocular systems [3], [15] as only one fundamental perspective is available. These systems have to handle occlusions and the fact that motion in the depth direction is not directly visible.

There are several different approaches to determine the posture from a given set of images. For monocular 2D-3D registration, where the actual 3D pose is determined from a single view, a precise model and optimization algorithms are use to find the correct pose, as done in [15]. Systems that use stereo cameras usually extract the 3D location of key body parts like the head and hands and use other methods to solve the position of elbows and other body parts [1], [8]. Algorithms that use more cameras can extract 3D information from the different views and then fit an articulated model to this 3D data [9], [2], [4], [16], [11]. A different approach is to project a model into each view and determine the change of the model from each view either by optimization [3], [7], [15], filtering or sampling.

Particle filters are used in several algorithms [1], [5], [6], [8], [12], [14]. The biggest problem with human motion tracking with particle filters is the exponential growth of the needed number of particles with the increase in dimensions of the search space. For most whole body human models in 3D we have about 30 DoF, which is infeasible to solve with a regular particle filter. One of the tasks to solve when using particle filtering for human motion capture is to deal with

Institute for Technical Informatics, University of Karlsruhe, Germany
F. Hecht is a CS Diplom student, at the University of Karlsruhe `florian.hecht@ira.uka.de`
P. Azad is with the Institute for Technical Informatics, University of Karlsruhe `azad@ira.uka.de`
R. Dillmann is Professor of the IAIM chair at the Institute of Technical Informatics, University of Karlsruhe `dillmann@ira.uka.de`

978-1-4244-2789-5/09/$25.00 ©2009 IEEE

this high dimensionality. Several different approaches have been proposed:

One approach is to split the search space into smaller search spaces in combination with hierarchical search or the localization of certain body parts by other means, like the head, hands or the the dominant axis of the torso [12]. One big criticism of these approaches is that, say the two arms, are treated independently from each other, where in fact the position of the one-side influences the position of the other. In [6], an automatic partitioning scheme is proposed that reduces the needed number of particles while still creating the needed interactions.

In [14], a strong motion model is used to predict the state of the skeleton in the next frame. This learned motion model reduces the number of needed particles since the particles will be relatively close to the actual position. This system is limited to tracking one type of motion at a time (walking in that paper) and is therefore not universally applicable, but showed that a good prediction can significantly improve tracking results.

Another approach to deal with the high dimensional search space is the *annealed particle filter* as proposed in [5]. Similar to the optimization technique *simulated annealing*, several filtering runs are performed, with increasing detail in the weighting function. This guides the particle set to coarse peaks first, and then optimizes the result with finer details. With this approach the method can find the correct optimum with a reduced number of particles. The annealed particle filter is analyzed for application in non-studio-like environments in [13] and was found to depend on relatively noise free measurements for reliable results. In [6], extensions to this method are proposed including decreased noise that is dependent on the variance of each state space variable. The purpose of this is to focus attention to variables which are not yet determined precisely and to prevent losing an already precise localization of variables due to added noise.

Another algorithm is presented in [15], where *covariance scaled sampling* is proposed, which is a generalization of the variance-based noise extension of the annealed particle filter. The state space distribution is represented as a mixture of Gaussians. Samples are generated from each Gaussian, with a distribution that captures the dimensions with the most variance computed by eigen decomposition of the covariance matrix.

None of [14], [5], [15] achieve real-time results and are thus not applicable for use on a robot.

B. Outline

This paper presents a new method that was developed for the purpose of HRI. The focus here is on performance and robustness under certain conditions, like a short-baseline stereo camera and a frame rate of at least 10 Hz on a regular PC. The proposed method is not restricted to the application on a robot, but can be used in multi-camera scenarios with increased precision. The flexible model that represents the human pose is introduced in Section II. The integration of the particle filters into the complete tracking system is described in Section III, which is followed by experimental results in Section IV. Concluding remarks and ideas for future work can be found in Section V.

II. Flexible Model

The flexible model used in this application is a mass-spring system as illustrated in Fig. 1. Unlike a classical kinematic model, which uses a hierarchy of joints, we have a set of point masses, which are connected through springs. The springs represent abstract bones for the extremities, but are also used to construct a two-part torso, which is not completely rigid. The state of the model consists of the positions of the 16 points, which means the model has $16 \cdot 3 = 48$ DoF, which is more than the usual 30 DoF for a kinematic model, but still less than the $11 \cdot 6 = 66$ DoF for a model consisting of connected individual body parts[1].

Another thing to note is that the model does not represent rotations explicitly. Some rotations can be recovered from the positions of the mass-points, but the rotations around the arms' axes for instance cannot be derived. If the 24 springs in the system where completely rigid, then the number of DoF of the complete system would be $48 - 24 = 24$. But the springs are not 100% rigid and hence give the model more flexibility. Parametrizing the human model this way seems a bit of a waste, but enables a locality of change, since movements in a certain point can be implemented simply by moving that mass-point and do not require an inverse kinematics calculation. Note that the 48 DoF are estimated in a semi-hierarchical way as will be explained in Section III. Also this model allows splitting of the state space, while still retaining interaction between the subparts, which enables a good estimate for one part to fix the bad estimate for another part, thus greatly improving the robustness of the whole estimate.

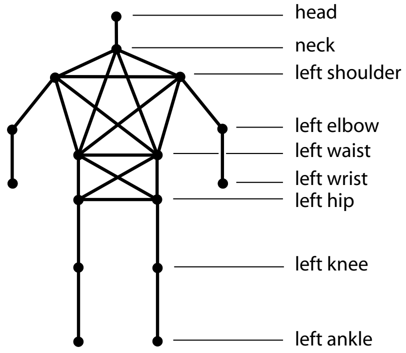

Fig. 1. Mass-Spring model of a human showing the mass-points and the springs.

The model is enhanced with additional constraints to limit the angular motions of the limbs. Since rotations are not modeled, several coordinate frames are derived from sets of the three positions of mass-points. In these frames the

[1] This assumes two parts per extremity, two for the torso and one for the head, thus 11 parts, where each part has 6 DoF as a rigid body.

rotational limits are defined. Further constraints prevent the self-penetration of the arms and the torso (by using an anisotropically scaled cylinder collision), as well as cross-over situations through distance constraints between the legs. This limits the number of states the model can get into further. The model is suspended in the air with gravity dangling from the position of the head.

The mass-spring system is solved in a Verlet framework [10], [17], which makes the implementation very simple and the physics simulation very stable. The system of constraints is solved by iteratively applying them individually several times, similar to Gauss-Seidel iterations for linear systems. The Verlet integration step is

$$\boldsymbol{x}_{t+1} = 2\,\boldsymbol{x}_t - \boldsymbol{x}_{t-1} + \boldsymbol{a}_t \Delta t^2,$$

where $\boldsymbol{x}_t$ denotes the position of a point at time t, $\boldsymbol{a}_t$ is the acceleration which is computed from the accumulated forces during a frame, and Δt is the time step. The significant difference to the regular Euler integration is the absence of the velocity, which is implicitly calculated by the difference to the previous position. This implicit velocity makes the solution stable, since position and velocity cannot get out of sync. It also simplifies the implementation of various constraints, since the current position is simply projected to a valid state without having to calculate a new velocity, which is implicitly handled by the previous position.

The weights of the mass points are chosen so that the torso points are heavier and the points of the extremities are lighter. The head has an infinite weight, which makes it immovable, i.e it is only moved by assigning the position measured by the head tracker. The strengths of the springs have been chosen in a way to reflect the movability of the mass-points with regard to their corresponding joints in humans, which gives the shoulder points a certain amount of play. The springs also allow the model to adapt to a limited amount of size change of the tracked person.

The state of the mass-spring model is used to calculate the state of a cylinder model, which consists of two cylinders for each extremity, two for the torso and one cylinder for the head. This model is used for projections in the measurement model of the particle filters, as well as for the occlusion model, and for visualizations. See also Fig. 2.

The various parameters of this model have been determined empirically and may not be realistic with regard to a real human, but the simulation produces quite realistic results and enables good predictions. This model can be used for people of a similar stature, but the lengths and diameters would have to adapted for persons of different stature (depending on the body height). The constraints and spring coefficients can stay the same.

III. Tracking with a Flexible Model

The tracking of the body takes place in a two-step semi-hierarchal way: The head is tracked by a special particle filter-based face tracker, using skin color segmentation. This is the only place where the stereo-camera is actually required, as the rest of the system could be using only one of the images. Unfortunately, the 3D estimate of the particle filter is not precise enough for a robust 3D localization of the head. Therefore we use stereo correspondences with features on the face to get a better depth estimate. The head point of the mass-spring system is moved to the head position and the rest of the points are moved 80% of the translation that the head point moved.

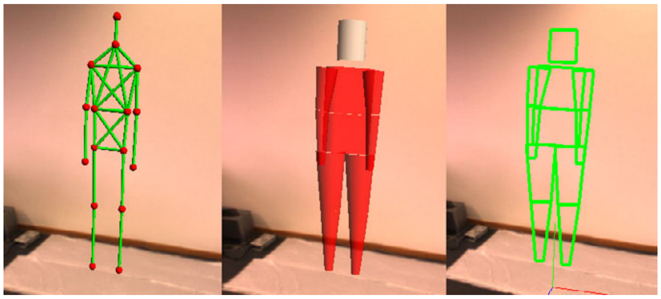

Fig. 2. Overlays of the model over the rectified image. From left to right: The mass-spring system, the cylinder model and the simplified cylinder projection.

When the model is moved with the head, the rest of the body pose is determined by tracking the extremities with particle filters and solving the mass-spring system, which creates an implicit solution for the torso. The approach to deal with the *curse of dimensionality* is to split the problem into smaller sub parts and to integrate the partial solutions into an overall solution. However, one cannot simply split the state space into smaller parts without considering the interactions between the sub-parts. The needed interaction is achieved by the mass-spring system, which influences and is influenced by the four particle filters of the extremities, as shown in Fig. 3. The particle filter for a limb has a formal state space with $3{\cdot}3 = 9$ dimensions, combining the positions of the three mass-points that define the state of that limb, e.g. shoulder, elbow, and wrist position for an arm filter. However, for the same reason as before, the actual DoF is lower, since with rigid springs only 4 DoF would be used for each extremity. So with not completely rigid springs, as was used here, we have more than 4 DoF but much less than the formal 9 DoF.

The estimation of the particle filter for each extremity is put into the mass-spring system by overwriting the position of the points that are filtered. The mass m_i of the i-th point is changed to reflect the confidence in the estimate of the particle filter, which is based on the variance σ_i of the i-th point in the particle filter. While unrealistic in a physical simulation sense, it works very well as way to integrate the 4 weighted results into the complete state estimate.

$$m_i = m_{base} + m_{variable} e^{s\sigma_i},$$

with m_{base} being the base mass and a variable part $m_{variable}$, and $s < 0$. It hence gives more weight to points it is confident about and less weight to points for which the estimate is not reliable. The solution of the mass-spring system will then determine a state which reflects these confidence measures. The four extremity particle filters are run independently from each other and use the current state of the mass-spring system in their motion models. The estimates are put into the mass-spring system at the same time.

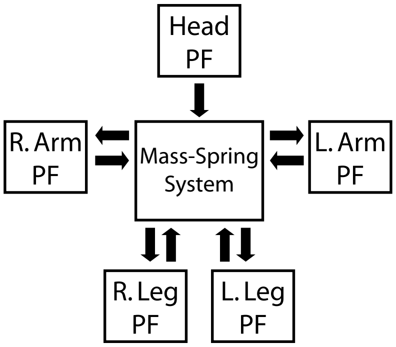

Fig. 3. The particle filter (PF) of the head influences the mass-spring system, while there is a two-way coupling of the extremity PFs and the physics simulation.

A. Motion Model

Each particle of the particle filters consists of three points $(\boldsymbol{p}_1^t, \boldsymbol{p}_2^t, \boldsymbol{p}_3^t)$, representing an alternative state to the one in the mass-spring system at time t. The points are first moved with a constant velocity model

$$\bar{\boldsymbol{p}}_i^{t+1} = \boldsymbol{p}_i^t + \eta(\boldsymbol{p}_i^t - \boldsymbol{p}_i^{t-1}),$$

where η is a factor that determines the trust in this constant velocity model (empirically set to $50\% - 80\%$). The previous position is stored for all particles and points. After that, noise is added, which is a mixture of two Gaussian distributions: a process noise and a "depth" noise that is intended to push particles into the direction that is not visible in the camera image to improve the search for the correct state, similar to the use of covariance scaled sampling in [15]. The new position of the point is drawn according to the probability distribution of the new position $p(\boldsymbol{p}_i^{t+1})$, which is modeled as:

$$p(\boldsymbol{p}_i^{t+1}) = (1-\alpha)\,\mathcal{N}(\bar{\boldsymbol{p}}_i^{t+1}, \sigma_i \mathbf{I}) + \alpha\,\mathcal{N}(\bar{\boldsymbol{p}}_i^{t+1}, \boldsymbol{\Sigma}_i),$$

where $\bar{\boldsymbol{p}}_i^{t+1}$ is the prediction from the constant velocity model, $\mathbf{I}$ is the identity matrix and σ_i the variance of the first normal distribution $\mathcal{N}(\bar{\boldsymbol{p}}_i^{t+1}, \sigma_i \mathbf{I})$. The second normal distribution has a covariance matrix instead of a uniform variance. The mixture weight α determines how many particles are drawn in average by the second distribution. We use a value of $\alpha = 0.25$. The first variance σ_i, which represents the unknown motion, consists of three components: a base noise level, one that depends on the variance of the point positions[2], and one that depends on the motion in the image at the projected position of the particle. The amount of motion is sampled from a motion image for each of the three points of the particle. For this purpose the motion image is a down-sampled and heavily blurred thresholded difference image between the previous and current image. If more computational power is available, this could be improved by the use of optical flow.

The second part of the noise distribution is the depth component with the covariance matrix $\boldsymbol{\Sigma}_i$. This matrix is constructed as follows:

$$\boldsymbol{\Sigma}_i = \mathbf{T}\,\mathbf{R} \begin{pmatrix} \sigma_x & & \\ & \sigma_y & \\ & & \sigma_z \end{pmatrix} \mathbf{R}^T\,\mathbf{T}^T$$

The diagonal covariance matrix that expresses the different scale factors in a frame where the z-axis is the depth direction. This covariance matrix is rotated into camera coordinates by $\mathbf{R}$ and then into world coordinates by $\mathbf{T}$. The transformation into world coordinates $\mathbf{T}$ comes from the camera projection. The rotation matrix $\mathbf{R}$ is constructed for each point, as a coordinate frame which has the z-axis in the depth direction. The scale factors are chosen $\sigma_x = \sigma_y = \epsilon$ to be very small and the σ_z is chosen differently for each point of a particle, to give the end of the extremities, wrists and ankles, more depth noise than the other points[3]. Since the noise in the x- and y-directions in the rotated frame are small and since the construction has to be done three times for every particle, an approximation to this Gaussian distribution is used, which is simply Gaussian noise along the depth direction.

At this point we have combined a constant velocity model with a model for the assumed and estimated random distribution of the particles. But since we estimate these values independently for the three points of the particle, the particles might have reached positions which do not correspond to physically or anatomically correct states of the extremity. To force the particles back to valid states as defined by the constraint system of the body model, the three points are put into the mass-spring system and the active constraints that affect these points are applied to them for several iterations. The other points in the mass-spring system are assigned an infinite mass and do not move (See Fig. 4). This is the other direction of the coupling between the mass-spring system and the particle filters, since the state of the mass-spring system influences the state the particles can get into.

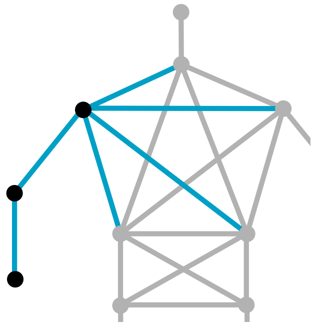

Fig. 4. Constrains in the particle filter of the right arm — points from the particle are black, the active constraints are light blue and the disabled constraints and mass points are gray. Only the spring constraints are displayed.

[2] This is the same variance that is used to determine the mass of the points in the mass-spring system, that are estimated by the particle filters.

[3] Empirically set to $\sigma_z = 1.0m/0.5m/0.15m$ for wrists/elbows/ shoulders. Same for the legs.

B. Measurement Model

The measurement model is using the projection of cylinders into the images to calculate two scores per particle, an edge score and a surface score. The edge score is calculated by comparing the projected long edges of the cylinders against an edge image (like in [5]), which is a blurred version of the combination of a thresholded difference image and a thresholded gradient image. This combination has the benefit of having a higher score for moving edges that are assumed to belong to the moving person. The second benefit is that this scheme does not depend on a foreground segmentation. The drawbacks are that it is not as clean as silhouette edges from a segmented image and that it also features edges that do not belong to the person, which is compensated to an extent through the moving edges.

The surface score is calculated by sampling the surface of the simplified cylinder projection with a regular grid, as illustrated in Fig. 5. One possibility for calculating the score is counting the number of foreground samples, as it is coming from a foreground segmentation. But this requires a good segmentation, which is hardly possible with motion segmentation. Therefore we use an appearance model instead, where each sample on the grid gets a reference color, which is compared to the actual color in the current image (sum of absolute differences). The color model for each body part (left/right upper arm, left/right lower arm, etc.) consists of ten reference colors along the cylinder for that part and assumes that the color is constant on a ring around the surface of the limb. The colors are linearly interpolated to get the reference color for a sample, by using the normalized distance from the base of the cylinder as an index.

The color model is learned during the first couple of frames, with an running average scheme. The color in a frame is sampled on a regular grid and the points that are classified as foreground (from the segmented image) are averaged together per ring of the cylinder and learned over several frames. Once the learning is complete, a foreground segmentation is no longer needed and an active head could start moving again.

The whole measurement model makes use of an occlusion model, which is based on the current state of the mass-spring system. The occlusion model consists of the convex quadrilaterals of the simplified projections of the cylinders. The quadrilaterals have a minimum and maximum depth and are indexed through a spatial grid, which makes the query whether or not a sample point is visible very fast. All samples, for the edge and the surface score, are tested and receive special default scores if a sample is occluded or is outside of the view. These default values have to be chosen carefully to prevent a preference for occluded or unoccluded states.

C. Initialization

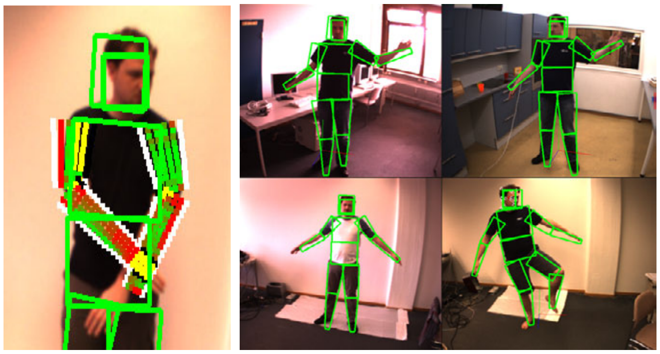
Fig. 5. The left image is showing the sampling grid for the arms displayed on the mean estimates. The surface samples show the quality of the match in a red to green scale. Occluded sample points are yellow. Edge samples are white or black (occluded). The right image shows different backgrounds and clothes.

The system is initialized by taking the first frame as a background model, which is used for the background subtraction. The head tracker is looking for a suitable skin blob to track as the head. When a head is found, the model is moved to the measured position of the head and the physics simulation starts — but at this time without the particle filters, which results in a relaxed state of the model with arms and legs simply dangling. The configuration from the physical simulation is run through the measurement model and the edge score is used to determine when a match between the relaxed state and the image occurs. When a match is found an initial appearance model is captured and the tracking with the filters starts. During the next frames the color model is updated as already explained. The learning rate depends on the local confidence values, which are interpolated along the extremities. When the learning stops, the appearance model is fixed and the foreground segmentation is no longer needed, allowing the camera to be moved.

Note that this initialization procedure assumes that the person will get into a pose that is close to the relaxed state at the beginning.

IV. Experimental Results

A. System Parameters

The single threaded tracking application runs on a Pentium 4 3.2 GHz CPU with 10 Hz. The calibrated stereo camera consists of two Dragonfly cameras by Point Grey Research Inc. Image processing is performed on 640 × 480 color images. The used lenses are 4 mm M12 micro lenses, which have a significant radial distortion, but provide a big enough field of view, to see the complete human at a distance of 2–3 m. The software was built using the *Integrating Vision Toolkit*[4] which offers a clean camera abstraction and a generic camera model.

The particle filters used 200 particles per extremity and 100 particles for the head. The processing time is ≈35 ms for the image processing and ≈50 ms for the filtering, where the motion model is consuming about half the processing power.

B. Range of Motions

The tracking of the arms is able to follow the motions of a single arm very robustly, even through complicated

[4] http://ivt.sourceforge.net

and ambiguous situations. This is achieved through the combination of the angular constraints, the occlusion model and the motion model, which guide the estimate through ambiguous situations. The tracking is even able to detect if parts of a limb are hidden behind the body or the head, due to the occlusion model. The interactions of the two arms are tracked as long as both arms are visible to a certain extent, cross-over situations are successfully tracked due to the occlusion model. Fig. 6 shows tracked arms positions. Furthermore folded arms are captured, but the estimate does not reflect which one of the arms is visible, since it yields fluctuating depth estimates. When opening the folded arms, the tracking can get confused, but usually recovers through the opening motion.

The legs are tracked individually very well, including the detection of knee bends and folded up calves, which works also when standing on one leg. As with the arms the interaction of the two legs is more complicated to track. As they are closer to another than the arms, they have a tendency to both capture the same leg. To compensate that, a plane separation constraint forces the points from the two legs away from each other similar to the self penetration constraint, which improves the tracking, but makes it impossible to capture natural cross-over situations. When a person turns side ways, one leg is completely occluding the other, which can cause significant confusion for the filters.

As the tracking depends strongly on the localization of the head, more precisely the face, the person cannot turn away more than 90° from the camera. The suspension of the head together with the simulation of gravity implies that the person has to be standing. The waist area is also relatively stiff at the moment, and thus does not model the flexibility of a human. The constraint framework allows easy implementation of external constraints, like collisions with objects or furniture, which should allow the integration of specific knowledge into sitting and other special motions, but this has not been explored yet.

C. Precision

To measure the precision of the estimate one needs ground truth to compare to. Unfortunately, a motion capture system was not available, so the tracking of an object in the right hand, which was localized with the stereo camera, is compared to the estimate of the right wrist point of the human motion tracking application. This is a substitute for better ground truth and is not very precise, but confirms that most of the error – as one would expect – is in the depth direction and that the correct estimation of bent limbs needs a certain amount of foreshortening of the limbs before the model will capture it. Comparing the 3D renderings of the estimates to images gives enough insight into how good the estimate is and where the precision deteriorates. See also the accompanying video. The focus of the proposed system is not on precision, but on speed and robustness, since the estimated body posture is used for the recognition of states and gestures and not the capturing of performances for animations.

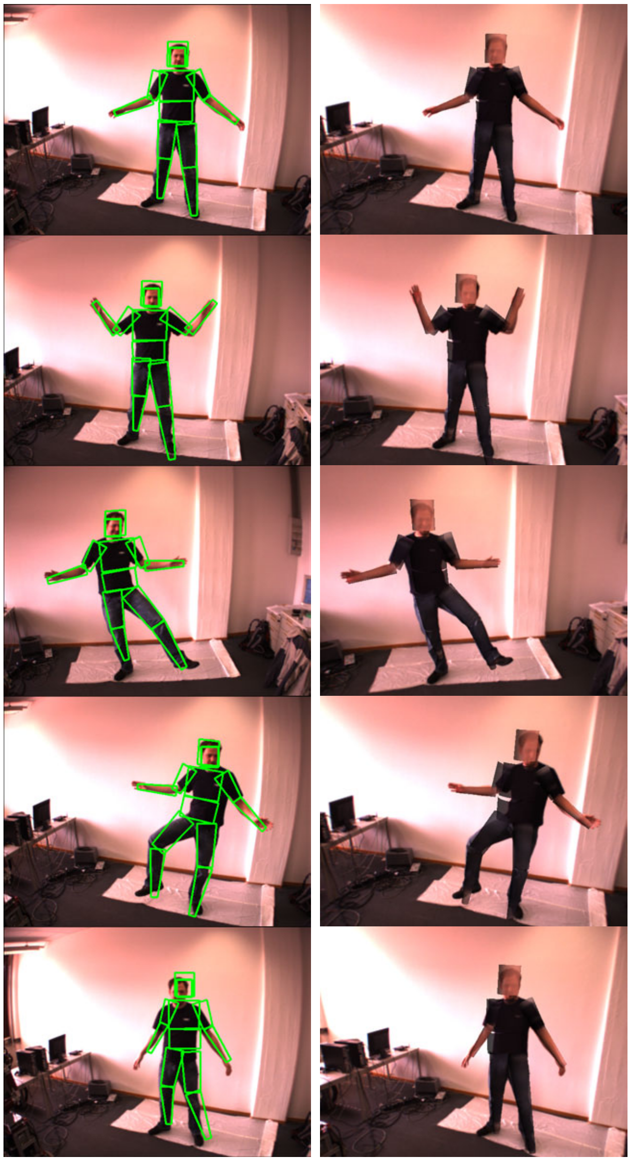

Fig. 6. A series of pictures from a tracking session. The left side shows the simplified cylinder projection, while the right side shows a 3D overlay over the rectified image. The 3D model is transparent and uses the color from the appearance model. The head and torso color are derived from the left arm.

D. Tracking Failures

Unfortunately, the system has problems when the person is seen from the side, since two limbs are not visible. The estimate for the parts that are not visible will deteriorate rapidly which will affect the estimates of the other limbs, since they are connected through the flexible model. Turning back to face the camera again, does usually not recover the correct estimate.

If two limbs are close to each other and one is moving while the other is still, the moving limb can affect the estimate of the still limb, by pulling the estimate away, since moving edges yield a higher score than still gradients. Particles from the still limb will follow the moving edges.

Naturally, motions which are too fast for the 10 Hz processing rate of the system cannot be tracked

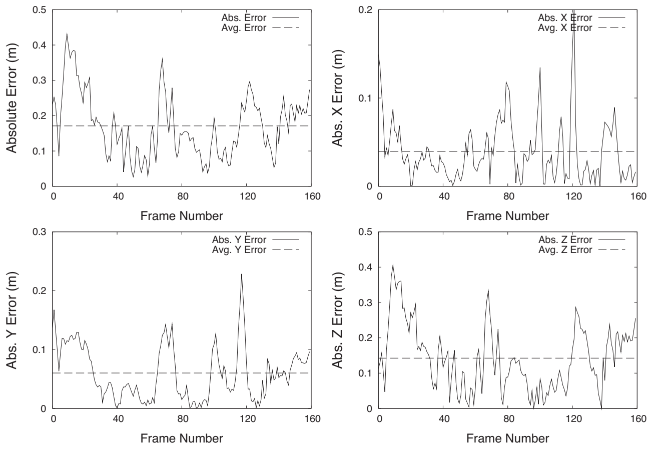

Fig. 7. Plot of the absolute error between the tracked object position and the wrist estimate of the human motion tracking, showing that the biggest error is in the depth direction, which also correlates the most with the overall error. Note the different scales!

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We have presented a new way to approach the human motion tracking problem, by coupling a physical mass-spring simulation with probabilistic particle filters to get an estimate for the pose of a human. Due to the reduced dimensionality of the particle filters achieved by semi-hierarchical decomposition, the system can run with 10 Hz on a regular PC, which allows real-time tracking of the motions of a person. The flexible model enables the tracking system to adapt to the person being tracked and creates a natural interaction between different parts of the body model.

A simple appearance model with reference colors is learned on-the-fly and allows the system to work without background subtraction, which enables the camera to be moved after the appearance model has been learned. The use of a strong motion model, which takes motion cues from the images and enforces the constraints of the system, allows tracking through ambiguous situations. Together with the occlusion model this creates a robust system that is able to track through complicated situations. All this is done with a stereo camera, where the stereo is only used for the head tracking. The filtering is basically monocular[5].

B. Future Works

Particle filtering is very well-suited for a parallel implementation. The speedup from such an implementation should be substantial and would increase the range of motions that can be tracked, due to the current speed limit.

The current mass-spring system is flat and has been chosen because it is the simplest model for the purpose, which results in few springs. A more elaborate mass-spring skeleton could enable more realistic deformations for the waist area and more freedom for the hip.

The current system uses only a stereo-camera and is therefore suitable for application on a humanoid robot head. However, we plan to evaluate how a system with several cameras can benefit from the proposed flexible body model.

VI. ACKNOWLEDGMENTS

The work described in this paper was partially conducted within the EU Cognitive Systems projects GRASP (FP7-215821) and PACO-PLUS (FP6-027657) and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft).

REFERENCES

[1] P. Azad, A. Ude, T. Asfour and R. Dillmann, ”Stereo-based Markerless Human Motion Capture for Humanoid Robot Systems”, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3951–3956, 2007

[2] F. Caillette and T. Howard, ”Real-Time Markerless Human Body Tracking Using Colored Voxels and 3-D Blobs”, *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 266–267, 2004

[3] Y. Chen, J. Lee, R. Parent and R. Machiraju, ”Markerless Monocular Motion Capture Using Image Features and Physical Constraints”, *Computer Graphics International*, June, pp. 36– 43, 2005.

[4] G. K. M. Cheung, S. Baker and T. Kanade, ”Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 77–84, 2003

[5] J. Deutscher, A. Blake and I. Reid, ”Articulated Body Motion Capture by Annealed Particle Filtering”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 126–133, 2000

[6] J. Deutscher, A. Davidson and I. Reid, ”Automatic Partitioning of High Dimensional Search Spaces associated with Articulated Body Motion Capture”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 669–676, 2001

[7] T. Drummond and R. Cippola, ”Real-time Tracking of Highly Articulated Structures in the Presence of Noisy Measurements”, *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 315–320, 2001

[8] M. Fontmarty, F. Lerasle and P. Danès, ”Data Fusion within a modified Annealed Particle Filter dedicated to Human Motion Capture”, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3391–3396, 2007

[9] P. Fua, A. Gruen, N. D' Apuzzo and R. Plänkers, ”Markerless Full Body Shape and Motion Capture From Video Sequences”, *Symposium on Close Range Imaging, International Society for Photogrammetry and Remote Sensing*, Corfu, Greece, 2002

[10] T. Jakobsen, ”Advanced Character Physics”, *Game Developer Conference 2001*, www.teknikus.dk/tj/gdc2001.htm, link verified June, 26th, 2008

[11] S. Knoop, S. Vacek, R. Dillmann, ”Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model”, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1686–1691, 2006

[12] M. W. Lee, I. Cohen and S. K. Jung, ”Particle Filter with Analytical Inference for Human Body Tracking”, *Proceedings of the Workshop on Motion and Video Computing (MOTION)*, p. 159ff, 2002

[13] P. Peursum, ”On the Behaviour of Body Tracking with the Annealed Particle Filter in Realistic Conditions”, *Technical Report*, November, 2006

[14] H. Sidenbladh, M. J. Black and D. J. Fleet, ”Stochastic Tracking of 3D Human Figures Using 2D Image Motion”, *Proceedings of the 6th European Conference on Computer Vision*, Part II, pp. 702–718, 2000

[15] C. Sminchisescu and Bill Triggs, ”Estimating Articulated Human Motion With Covariance Scaled Sampling”, *International Journal of Robotics Research*, Vol. 22, No.6, pp. 371–393, 2003

[16] A. Sundaresan and R. Chellappa, ”Markerless Motion Capture using Multiple Cameras”, *Proceedings of the Computer Vision for Interactive and Intelligent Environment (CVIIE)*, pp. 15–26, 2005

[17] L. Verlet, ”Computer Experiments on Classical Fluids”, *PhysRev. Vol. 159*, No. 98, July 1967

[5] We actually alternate between the two images, but the benefits are minimal.