Full Body Tracking from Multiple Views Using Stochastic Sampling

Roland Kehl[1], Matthieu Bray[1,2,4], Luc Van Gool[1,3]
[1]Computer Vision Laboratory (BIWI), ETH Zuerich, Switzerland
[2]Oxford Brookes University, Department of Computing, Oxford, UK
[3]ESAT-PSI, University of Leuven, Belgium
{rkehl,vangool}@vision.ee.ethz.ch
mbray@brookes.ac.uk

Abstract

We present a novel approach for full body pose tracking using stochastic sampling. A volumetric reconstruction of a person is extracted from silhouettes in multiple video images. Then, an articulated body model is fitted to the data with stochastic meta descent (SMD) optimization. By comparing even a simplified version of SMD to the commonly used Levenberg-Marquardt method, we demonstrate the power of stochastic compared to deterministic sampling, especially in cases of noisy and incomplete data. Moreover, color information is added to improve the speed and robustness of the tracking. Results are shown for several challenging sequences, with tracking of 24 degrees of freedom in less than 1 second per frame.

1. Introduction

Over the last few years, the tracking of articulated structures such as human bodies or hands has gained in popularity. Applications include surveillance, human-computer interaction and computer based animations in games and the movie industry. Most of these applications also require the tracking not only to be robust, but also fast.

Monocular approaches have the advantage that they work with a simple hardware setup, i.e. one, uncalibrated camera. At least for the moment, the advantages of multi-view systems can easily outweigh this feature, however, as problems with occlusions and appearance ambiguities can be strongly reduced. This leads to faster and more robust algorithms. Thus, fast, marker-less pose and gesture recognition as proposed in this paper but from monocular sequences seems quite remote still [1, 2, 3, 4].

Several multi-view approaches for marker-less body tracking have been published lately [5, 6, 7, 8, 9]. Also under multi-view conditions, the problem is challenging due to the high dimensionality of the state spaces (many degrees of freedom). Next, we discuss the novelty of our approach compared to these seminal contributions.

In their early work, Gavrila and Davis [6] used 4 calibrated cameras to track people. The goal function is based on the comparison of observed vs. predicted target contours. The use of tight-fitting clothes, with sleeves of contrasting colors, made the segmentation easier. In our case, the user can wear casual clothes, even with homogeneous colors. This said, contrasting colors are exploited when available.

Cheung *et al.* [7] introduced a shape-from-shilhouette method for full body tracking from both silhouette and color information. They use colored surface points (CSPs) to segment the hull into rigidly moving body parts, based on the results of the previous frames, and take advantage of the constraint of equal motion of parts at their coupling joints to estimate joint positions. A complex initialization sequence recovers the joint positions of an actor, which are used to track the same person in new video sequences. The results shown were obtained with 8 calibrated and color-balanced cameras. No processing times were given. In our case, a single frame is sufficient for robust initialization. Furthermore, higher robustness of our method for body parts coming close to each other widens the feasible range of motions.

Seidel *et al.* [8] fit an articulated body model to 2D data by minimizing the overlap between the observed 2D shapes and the projections of the model. Although their method does not require explicit 3D reconstruction, an exact body model and noiseless segmentation of the person in the different videos is crucial to reach a meaningful error measurement. The tracking is done by using Powell's method and via hardware acceleration. They achieve accurate results at about 2 seconds per frame on distributed hardware. In [9], they enhance their silhouette-based method by incorporating texture information into the tracking process. A 3D motion field is used to refine the pose estimate. In

[4] Currently funded by EPSRC grant GR/T21790/01 (Markerless Motion Capture for Humans in Video).

contrast, our optimizer uses gradient information to achieve tracking below 1 second per frame, even for incomplete or noisy measurements.

In this paper, we present a fast and robust model-based, multi-view approach for body tracking based on a volumetric reconstruction with texture information. In Section 2, we propose a simple but effective method for the volumetric reconstruction, which also supports efficient texturing. Section 3 describes the model and the skeletal structure underpinning its articulation. Section 4 focuses on our tracking framework. We propose stochastic sampling of our data set, which has proven to be advantageous. In particular, we compare tracking based on a simplified Stochastic Meta Descent (SMD) optimization vs. the widely used Levenberg-Marquardt method. Furthermore, texture information will be used to further improve both tracking speed and robustness. Section 5 shows experimental results for several challenging sequences, such as sticking the arms to the body or handling an object. Section 6 concludes the paper.

2. Volumetric reconstruction

For the 3D reconstruction we use a voxel representation. Multipe video streams are simultaneously captured and foreground/background segmentation is performed continuously on each. The intersection of the projection cones defined by the foreground masks yields the 3D shape. Our camera environment allows us to activate up to 16, statically mounted IEEE1394 cameras placed all around a single working volume. The cameras are synchronized through external trigger hardware and are calibrated by the automatic self-calibration procedure proposed by Svoboda *et al.* [10]. After acquisition, all images undergo a foreground/background segmentation by the illumination invariant method proposed by Mester *et al.* [11]. Assuming static backgrounds, previously captured sequences of the empty working space are compared to the actual image by using a statistic criterion that measures the collinearity of expected background and actual colors in color space. Fig. 1 shows the segmentation result for a frame captured from four different views.

2.1. 3D Reconstruction

Based on the foreground/background segmentations in the different images, a volumetric representation is created of the person to be tracked. This is based on the intersection of the backprojection cones for the foreground masks. The main drawback of voxel based procedures has been their computational cost [12]. Each voxel has to be projected into the image plane of each camera, leading to one matrix multiplication per voxel per camera. Most implementations speed up this process by using an octree representation to compute the result from coarser to finer resolutions [13]. Others exploit hardware acceleration [14].

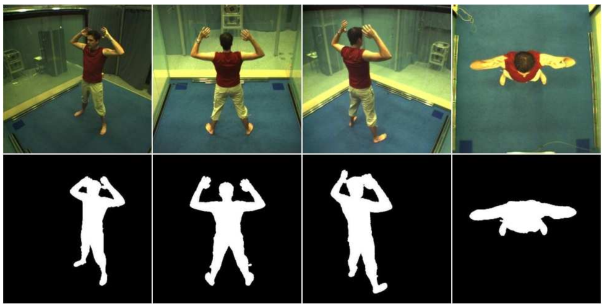


Figure 1. Segmentation results for 4 views.

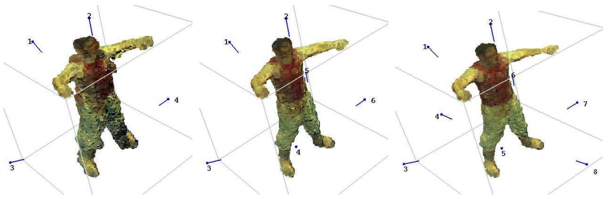
The proposed implementation addresses the problem the other way around. Instead of projecting the voxels into the camera views at each frame, we keep a fixed look-up table (LUT) for each camera view and store a list at each pixel with pointers to all voxels that project onto that particular pixel. To reach real-time reconstruction even at fine voxel resolutions, the projection of a voxel into the image planes is approximated by a patch of constant shape around the projected voxel center (for the size of voxels used in the experiments this simply was the 4-neighbour cross around the pixel onto which the center projects). Furthermore, each voxel is represented by a bitmask, where each bit b_i is 1 if its projection lies in the foreground of camera i and 0 otherwise. Thus, a voxel belongs to the subject if its bitmask only contains 1's. This can be evaluated rapidly by byte comparisons. This bitmask needs to be updated online.

The reconstruction itself is done by going pixel by pixel through all segmented (binary) images. If a pixel of the current view i has changed its value compared to the previous frame, the corresponding bit b_i for all voxels contained in the reference list of this pixel is set to the new value. For matching a model to the 3D observations, surface voxels suffice. An effective way to test for surface voxels is to check if at least one of their six nearest neighbors does *not* belong to it.

2.2. Texturing

In case of inaccurate reconstruction or body parts making contact, the model-observation matching can become ambiguous. Additional color information is used to further improve the matching process. In this Section, a method is presented that assigns a representative color to each surface voxel. This information will be used to group voxels into regions of similar colors, as discussed in Section 4.4.

For the sake of speed, multiple passes through the set of surface voxels have to be avoided. On the other hand, for robustness, all colors of the views where the voxel is not occluded should be mixed together. To meet this two re-



(a) 4 views. (b) 6 views. (c) 8 views.

Figure 2. Reconstruction results. Note that the quality of the surface as well as the texture quality increase with the number of cameras used.

quirements, we use a depth buffer to detect occlusions and a four-component representation for the color mixture. Three color coordinates are accumulated whereas a fourth component counts the number of contributing views. This counter allows the algorithm to also subtract a previously added color from the mixture, as required in the case of occlusion. Furthermore, the depth buffer also contains a pointer to the current visible voxel V_n for each entry, allowing to subtract previously added colors from it if an occlusion occurred.

The texturing algorithm linearly traverses the set of surface voxels. If a voxel V_i is projected into the image plane of camera C_j, the color of the corresponding pixel is added to V_i unless the depth buffer detects an occlusion. If no occlusion occurred, V_i also becomes the new visible voxel V_n in the depth buffer. As the algorithm traverses the set of surface voxels only once, occlusion can also happen to V_n if V_i is closer to the camera. In this case, the color has to be subtracted from V_n, where it has been previously added, and the algorithm continues as in the other case. As only the surface has to be textured, the coloring algorithm can be performed at the same time the surface voxels are identified.

2.3. Reconstruction Results

Our reconstruction algorithm lowers the computational complexity by using precomputed information (the visibility LUTs of Section 2.1). For each reconstruction, the process makes just one pass trough all camera images and through all voxels, with only fast bit-operations and memory look-up's. Furthermore, for subsequent frames, the reconstruction is updated instead of being computed from scratch since only the changes of the foreground masks have to be taken into account. Table 1 shows average processing times on a PIV 3GHz for the reconstructions in Fig. 2. We used 128^3 voxels (i.e. about 2 million) with a volume of $10cm^3$ each. The extracted surface consists of appr. 4000-7000 voxels. Looking at Table 1, one sees that times scale better than linear when the number of views or voxels is increased.

No. Views	Reconstruction	Texturing	Total
4	46.28 ms	15.29 ms	61.57 ms
6	49.71 ms	23.28 ms	72.99 ms
8	60.28 ms	30.51 ms	90.79 ms

Table 1. Reconstruction processing times.

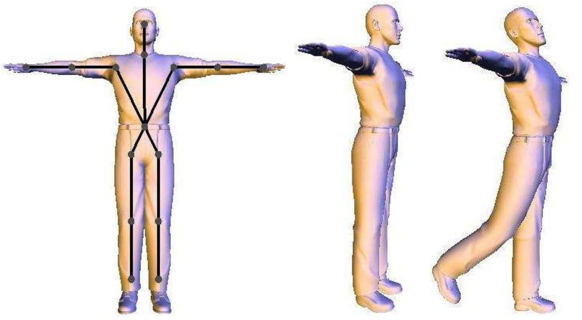
3. Model

During tracking, an articulated body model is fitted to the reconstruction. We use a human body model exported from Poser (http://www.curiouslabs.com/) which is bound to an articulated skeletal structure. The surface consists of approximately 22000 vertices. The skeletal structure is modeled by 24 degrees of freedom (DOF): a 6 DOF joint for the torso, a 3 DOF joint for the shoulders and hips each, 2 $DOF's$ for the neck and a 1 DOF joint for each of the knees and the elbows. Fig. 3 a) shows the model and its joints. The three rotational parameters of the torso, shoulder and hips are modeled by an axis-angle representation, which can be easily converted to a rotation matrix by using Rodrigues formula [15]. Two angles define the orientation of the limb whereas the third angle corresponds to a twist around its own axis. Therefore, the Gimbal Lock effect can be avoided.

To allow the model to undergo realistic physical deformations, the skin is bound to the skeleton by linear blend skinning [16]. This technique allows surface vertices to be influenced by more than just one joint. Without such blending, unnatural deformations appear near the joints. With the blending, transitions are gradual and convincing, as shown in Fig. 3 b) and c).

4. Tracking

Here we describe the tracking framework. The goal is to minimize the distance between the surface of the 3D model and the 3D reconstruction (observation) for each frame.



(a) Body model and skeleton. (b) Side view. (c) Deformed.

Figure 3. Articulated body model.

4.1. Initialization

The initialization of the model is fully automatic. The user first has to adopt an initialization pose, standing upright with his/her arms and legs spread in the “Da Vinci”-pose.

A first estimate of the user’s position can be derived from the reconstruction’s centroid. Two foot patches can be detected by using a plane sweeping algorithm near the bottom of the voxel space. The same can be done from left to right and vice-versa in order to find the hand positions and from top to bottom to find the top of the head. Then, statistics of the ratios between the different limb lengths as proposed by Dreyfuss [17] are used to complete the skeleton. During fitting for the first frame, this initial configuration is refined.

4.2. Objective Function

As we want to let the 3D model fit the 3D volumetric reconstruction as closely as possible, our objective function could be the sum of the distances of model vertices to the corresponding reconstruction voxels.

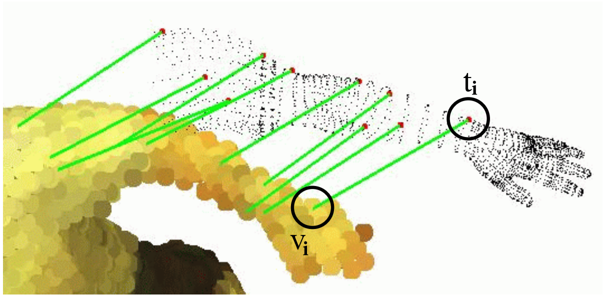


Figure 4. Illustration of the objective function of one arm.

Given the huge number of model vertices however, a subset $\mathcal{T}$ of *tracking vertices* $\boldsymbol{t_i}$ is sampled. For each $\boldsymbol{t_i}$ on the current model configuration $\boldsymbol{p_i}$, the closest voxel $\boldsymbol{v_i}$ is sought. So, the objective function is:

$$\mathcal{F}(\boldsymbol{p_i}) = \sum_{\mathcal{T}} \|\boldsymbol{t_i} - \boldsymbol{v_i}\| \tag{1}$$

Fig. 4 illustrates this, with solid lines indicating the assignment of tracking vertices (solid dots on the model) to their closest voxels.

4.3. Stochastic Meta Descent (SMD)

Stochastic Meta-Descent is a gradient descent with local step size adaptation that combines rapid convergence with excellent scalability and is built on 3 concepts:

- The use of *stochastic sampling* of the tracking vertices (rather than taking a fixed subset) at each iteration of SMD offers better convergence than standard optimization methods relying mostly on deterministic sampling (i.e. a fixed subset). Indeed, shuffling the sampling points lets SMD escape from local spurious minima more easily. Furthermore, due to the stochastic sampling, one can afford to use a smaller sample set than other standard deterministic optimizers and be faster as a consequence.

- SMD performs *two levels of optimization*: firstly, it optimizes for all parameters the individual step sizes $\boldsymbol{a}$ at each iteration by a gradient descent which is controlled by a meta step size $\boldsymbol{\mu}$. Then, the state parameters $\boldsymbol{p}$ are optimized in a gradient descent manner via $\boldsymbol{a}$.

- SMD updates parameters while taking into account the past history of step sizes, and thus is able to capture long-range effects missed by other algorithms. This dampens erratic variations and increases the method’s efficiency.

Here we give a concise overview of SMD. More detailed explanations can be found in [18]. If $\boldsymbol{g}_i$ is the gradient (of $\mathcal{F}$) at iteration step i, the parameter vector $\boldsymbol{p}_i$ is updated via

$$\boldsymbol{p}_{i+1} = \boldsymbol{p}_i \;-\; \boldsymbol{a}_i \cdot \boldsymbol{g}_i\,, \tag{2}$$

where $\cdot$ denotes the Hadamard [5] product. The vector $\boldsymbol{a}$ of local step sizes is in effect a diagonal conditioner for the gradient system.

The local step size vector $\boldsymbol{a}$ is adapted via

$$\boldsymbol{a}_i = \boldsymbol{a}_{i-1} \cdot \max(\,\tfrac{1}{2}\,,\, 1 \,+\, \boldsymbol{\mu} \cdot \boldsymbol{v}_i \cdot \boldsymbol{g}_i), \tag{3}$$

where $\boldsymbol{v}$ is an exponential average of the effect of *all* past step sizes on the new parameter values and $\boldsymbol{\mu}$ is a vector of meta step sizes. Note that in our original SMD implementation a simple scalar μ was used, which was the same for all dimensions. $\boldsymbol{v}$ is updated via:

$$\boldsymbol{v}_{i+1} = \lambda\,\boldsymbol{v}_i \;+\; \boldsymbol{a}_i \cdot (\boldsymbol{g}_i - \lambda\,\boldsymbol{H}_i\boldsymbol{v}_i)\,, \tag{4}$$

where $\boldsymbol{H}_i$ denotes the Hessian (i.e., matrix of second order derivatives), or a stochastic approximation thereof, at iteration step i. The factor $0 \le \lambda \le 1$ governs the time scale over which long-term dependencies are taken into account.

Imposing constraints

There are constraints on the parameter ranges of the different joints. Adding such constraints is most beneficial given the high-dimensional space that we have to explore. There are many ways to enforce constraints, such as penalty or barrier methods [19]. For SMD we can elegantly enforce

[5] component-wise: $\boldsymbol{a} \cdot \boldsymbol{b} = [a_1 \cdot b_1 \;\; a_2 \cdot b_2 \;...\; a_n \cdot b_n]^T$

them by gradient projection. After each update (2) a function maps the parameters back into the feasible region:

$$\boldsymbol{p}^c_{i+1} \quad = \quad \mathrm{constrain}(\boldsymbol{p}_{i+1})\,. \tag{5}$$

The constraining function $\mathrm{constrain}(\boldsymbol{p}_{i+1})$ enforces the human anatomical joint limitations by imposing a set of feasible intervals.

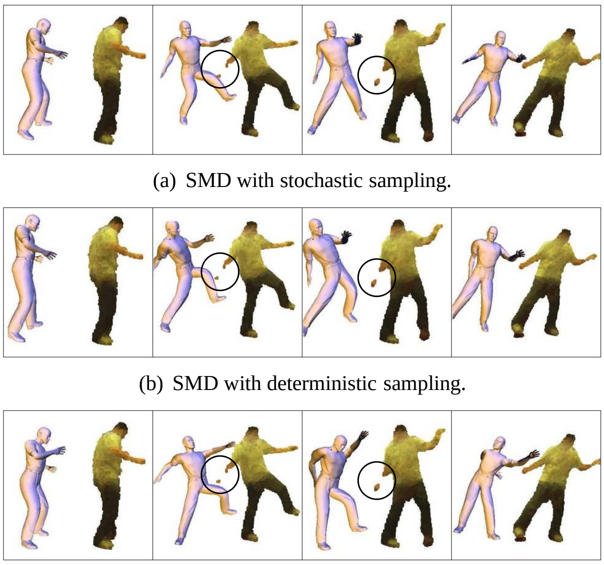
Since SMD uses the gradient not only to update the parameter vector $\boldsymbol{p}$, but also to adjust $\boldsymbol{a}$ and $\boldsymbol{v}$, we must make these adjustments also compatible with the constraints on $\boldsymbol{p}$. We do this by calculating a hypothetical 'constrained' gradient $\boldsymbol{g}^c$ which, applied in an unconstrained setting, would cause the same parameter change that we observe after application of the constraints. In other words, we require that

$$\boldsymbol{p}^c_{i+1} = \boldsymbol{p}^c_i \; - \; \boldsymbol{a}_i \cdot \boldsymbol{g}^c_i \Rightarrow \quad \boldsymbol{g}^c_i = \frac{\boldsymbol{p}_i - \boldsymbol{p}^c_{i+1}}{\boldsymbol{a}_i}. \tag{6}$$

SMD's step size adaptation machinery can then perform accurately in the constrained space by using this constrained gradient instead of the usual one in Eq. (4).

Stochastic sampling

The fact that SMD samples the tracking vertices stochastically is a key asset of this method. On each iteration, a new subset $\mathcal{T}$ of tracking vertices is chosen by a procedure that follows some rules to achieve an optimal coverage of the body, such as avoiding points close to the axil.

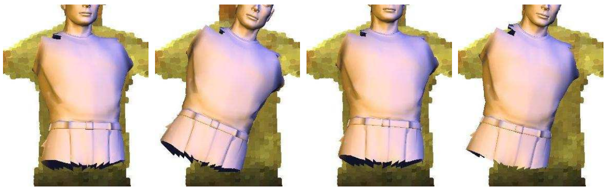


(a) SMD with stochastic sampling.

(b) SMD with deterministic sampling.

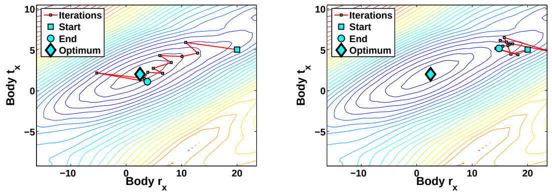
(c) LM.

Figure 5. Comparison between deterministic and stochastic sampling in presence of missing data.



(a) Optimum (b) Start (c) SMD (d) LM

Figure 6. Tracking comparison between LM and SMD with few tracking vertices.



(a) SMD (b) LM

Figure 7. Comparison of SMD and LM convergence.

We demonstrate the benefit of stochastic sampling by running two trackers, both based on SMD, with 5 samples on each limb and on the torso, and with a maximum of 20 iterations per frame. The results at four time instances are shown in the first and second rows of Fig. 5. As can be seen, the deterministic version looses track at the point where a large chunk of the lower right arm goes missing in the reconstruction (solid circle). One would have to increase the number of samples to make sure that some fall on the hand, which remains available in the reconstruction throughout. Stochastic sampling survives, because chances of sampling there at some iterations for a frame increase dramatically. Moreover, by repeatedly changing the position of the samples the shape of the objective function is changed each time. Spurious local optima will tend to move around, while the global optimum will typically stay put, as it will be consistent with all the samples within and across the sample sets.

We have also compared our SMD optimization against the popular Levenberg-Marquardt (LM) approach. Fig. 5 c) shows the same 4 key frames for the same sequence, under the same conditions. Again, LM with its deterministic sampling clearly fails to track the right arm, which gets stuck to the torso to which some of the lower arm vertices got erroneously assigned. Note that for the deterministic sampling experiments, the initial choice of the tracking vertices is stochastic, whereafter all points are kept fixed. Thus, the points selected for the deterministic experiments are also within the optimal coverage area.

Fig. 6 shows another comparison between SMD and LM. The torso was moved away (b) from its optimal position (a) by a rotation over 20 degrees in the plane of the figure and a translation to the left of 5cm. Both optimizers had to maneuver the torso back into the correct state through the 6D space of rotations and translations. The torso model was sampled with 5 points by both optimizers. SMD (c) is seen to move back into the desired position, whereas LM (d) got stuck in a local minimum. Fig. 7 shows a plot with isolines of the objective function for hyperplanes through the 6D parameter space, parallel to the axes corresponding to the changed parameters. The squares represent the starting positions and the circles the end positions of each optimizer. The global optimum is drawn as a diamond. The left plot a) reveals that SMD directly iterates towards the global minimum. The right plot b) shows that LM was stuck in a local minimum.

The experiments demonstrated the advantages of stochastic sampling over deterministic sampling. Stochastic sampling does not only provide robustness against incomplete data, it also lowers the number of required sampling points and consequently increases the tracker's speed.

4.4. Color-Based 3D Segmentation

The objective function $\mathcal{F}$ presented in Section 4.2 is not robust against body parts coming close to each other. Limbs may keep stuck to the wrong part of the data upon their actual departure. To remedy this flaw, $\mathcal{F}$ is extended with the texture information computed along with the reconstruction (Section 2.2). Colors of the model should match those of the reconstruction.

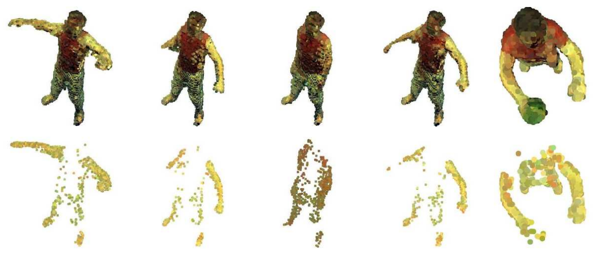


Figure 8. 3D color segmentation for a vertex on the lower right arm.

A color model is assigned to every model vertex during the initialization phase. After each frame, these color models are updated with the color of the last assigned voxel. The color model is represented by a Gaussian distribution in YUV space. Given the time constraints, we only update the mean, not the covariance matrix:

$$\mu_i^+ = (1-\alpha)\mu_i + \alpha c_i \qquad (7)$$

based on the newly incoming value c_i for channel i, and where α denotes a learning rate in the range of $[0..1]$, controlling the adaptation speed. During tracking of a particular model vertex, only voxels with a sufficiently high probability according to the vertex color model are considered. The closest voxel will be searched among only these. Fig. 8 shows examples for a skin vertex on the right lower arm of a person. The first row shows the complete reconstructions, while the second row shows the voxels selected based on their color. The last column is particularly interesting, as the person holds a green cube which extends the length of the 'arm'. Color selection eliminates it. Such color-based voxel selection can also significantly speed up the tracking.

Here again, stochastic sampling yields an important advantage, as it will greatly increase the chance of hitting colors that help distinguish body parts. For instance, when a person is wearing a shirt with long sleeves and homogeneous color, SMD has a good chance of sampling the skin colored hand. This allows the method to deal with the arm touching the body and leaving again, whereas without the hand sample, the arm will tend to get glued to the torso.

Stochastic sampling also has a drawback, however. Color models get updated only in a very piecemeal fashion, namely for those vertices which are used at the last iteration of a frame, when assignments of voxels are considered correct. When now changing the sample set, the color models of the new vertices have probably not been updated for a long time, if at all. Therefore, we cluster vertex colors and represent each cluster by a single color model. Instead of updating the color model of a particular vertex, that of the corresponding cluster is updated according to the sampled vertices.

The final algorithm is summarized in Fig. 9. At this stage, we should also mention two further refinements. On the one hand, we follow a hierarchical tracking approach by first fixing the torso and then tracking the limbs. Also, the replay of the motion by the model is based on smoothed trajectories, using a simple 5-wide averaging filter.

```
1. Compute textured reconstruction
2. Iterate SMD until convergence:
    • Compute the gradient $\boldsymbol{g}_i$ of $\mathcal{F}$
    • Update gradient step sizes (3)
    • Update model parameters (2)
    • Apply joint constraints (5) (6)
    • Update SMD's $\boldsymbol{v}$ vector (4)
3. Update color models (7)
```

Figure 9. Final algorithm for one frame.

5. Results

All experiments have been performed on 640x480 images with a single PIV 3GHz. 4 cameras are used for the results shown in Fig. 10, 11 and 12. Fig. 10 demonstrates the advantages of using color segmentation. Indeed, the tracked user holds his arms against his body and the tracker is able to split arms and torso again when the arms are raised. Fig. 11 presents tracking results while the user bends the knees whereas in Fig. 12, the user handles a green cube. At some point the cube is handed over from one arm to the other. The trackers nicely handles this configuration. Finally, Fig. 13 demonstrates the flexibility of our framework in a setting with 11 cameras. The user performs a 360 degrees turn while articulating all limbs.

For the presented experiments, between 5 and 15 tracking points have been chosen on each limb and all results have been obtained with the same SMD parameter settings: $\mu = 100$ for translation parameters, $\mu = 2500$ for the limb axis angles and $\mu = 10000$ for the twists. All numbers have been evaluated experimentally in order to strike an optimal balance between tracking speed and quality. Note that in the examples shown here $\lambda = 0$, which eliminates the need for second order derivatives (Eq. (4)). This simplification of SMD proved effective for our application, and yields a gain in speed.

Tab. 2 gives an overview of the number of cameras used and the average tracking time needed for one frame. Due to SMD, our method combines robustness and tracking speed below 1 second per frame in a 24-dimensional search space.

Experiment	Fig. 10	Fig. 11	Fig. 12	Fig. 13
Nb. of Cameras	4	4	4	11
Sec. per Frame	0.961	0.666	0.686	0.806

Table 2. Average tracking processing times.

6. Summary and Conclusions

In this paper, we proposed a method for the fast and robust tracking of full human body pose. A novel approach was proposed, which features efficient textured 3D reconstruction, SMD with stochastic sampling, and automatic color model updates. The approach was shown to be benefit from the stochastic sampling strategy. The use of color further increases the robustness. Tracking in less than 1 second per frame is achieved, in a 24D search space. Results for several challenging sequences with four and eleven cameras have been shown.

For future research, it is planned to incorporate 2D features to refine the tracking result and to improve the accuracy of our method. As we apply a hierarchical tracking, parallelization can be used to increase tracking speed.

7. Acknowledgments

The authors would like to thank Stefaan De Roeck from the University of Leuven, Belgium for his assistance on the segmentation code and Tomas Svoboda from the Czech Technical University, Prague for his valuable support during the first phase of this project. This work is carried out in the context of the blue-c-II project, funded by ETH grant No. 0-21020-04 as an internal poly-project.

References

[1] J. Deutscher, A. Blake, I. Reid, “Articulated Body Motion Capture by Annealed Particle Filtering,” *CVPR*, pp. 126-133, 2000.

[2] C. Sminchisescu, B. Triggs, “Covariance Scaled Sampling for Monocular 3D Body Tracking,” *CVPR*, pp. 447-454, 2001.

[3] H. Sidenbladh, M. J. Black, D. J. Fleet, “Stochastic Tracking of 3D Human figures using 2D Image Motion,” *ECCV*, pp. 702-718, 2000.

[4] C. Bregler, J. Malik, “Tracking People with Twists and Exponential Maps,” *CVPR*, pp. 8-15, 1998.

[5] I.A. Kakadiaris, D. Metaxas, “Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection,” *CVPR*, pp. 81-87, 1996.

[6] D.M. Gavrila, L.S. Davis, “3D model-based tracking of humans in action: a multi-view approach,” *CVPR*, pp. 73-80, 1996.

[7] G. K. M. Cheung, S. Baker, T. Kanade, “Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture,” *ACM SIGGRAPH”*, pp. 77-84, 2003.

[8] J. Carranza, C. Theobalt. M. Magnor, H.P. Seidel, “Free-Viewpoint Video of Human Actors,” *ACM SIGGRAPH*, pp. 569-577, 2003.

[9] C. Theobalt, J. Carranza, M. Magnor, J. Lang, H. Seidel, ”Enhancing silhouette-based human motion capture with 3D motion fields,” *Pacific Graphics 2003*, pp. 185-193, 2003.

[10] T. Svoboda, D. Martinec, T. Pajdla, “A convenient multi-camera self-calibration for virtual environments,” *PRESENCE: Teleoperators and Virtual Environments, 14(4)*, August 2005. To appear.

[11] R. Mester, T. Aach, L. Dümbgen, “Illumination-Invariant Change Detection Using a Statistical Colinearity Criterion,” *DAGM*, Vol. 23, pp. 170-177, 2001.

[12] S.M. Seitz, C.R. Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring,” *International Journal of Computer Vision*, 35(2), pp. 151-173, 1999

[13] R. Szeliski “Rapid octree construction from image sequences,” *Computer Vision, Graphics and Image Processing*, 58(1), July 1993.

[14] J.-M. Hasenfratz, M. Lapierre, J.-D. Gascuel, E. Boyer, ”Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors,” *Vision, Video and Graphics*, 2003.

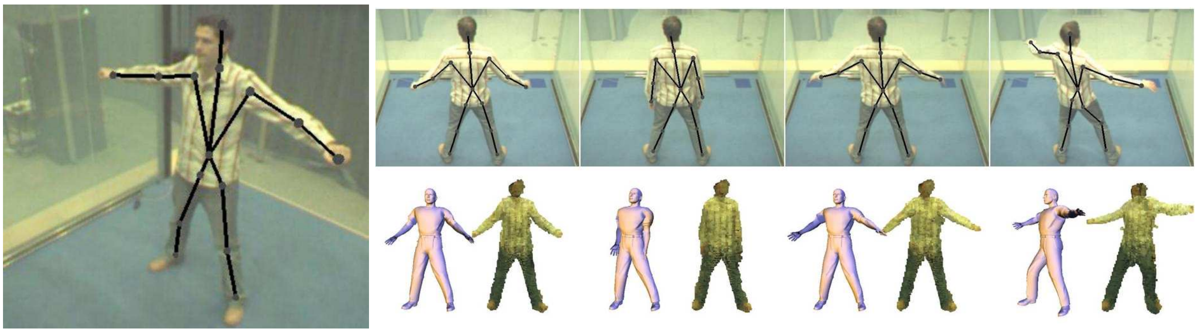
[15] R. Hartley, A. Zisserman, ”Multiple View Geometry in Computer Vision,” *Cambridge University Press*, 2000.

[16] K. Komatsu, “Human Skin Model Capable of Natural Shape Variation,” *The Visual Computer*, Vol. 4, No. 3, pp. 265-271, 1988.

[17] H.Dreyfuss, “The Measure of Man: human factors in design,” *Whitney Library of Design*, New York, 1959-1967.

[18] M. Bray, E. Koller-Meier, P. Müller, L. Van Gool, “3D Hand Tracking by Rapid Stochastic Gradient Descent using a Skinning Model,” *CVMP*, pp. 59-68, 2004.

[19] R. Fletcher, “Practical Methods of optimization,” *a Wiley-Interscience Publication*, Great Britain, 1987.



(a) Closeup and skeleton.

(b) Tracking result.

Figure 10. Sequence of the user holding his arms on the body.

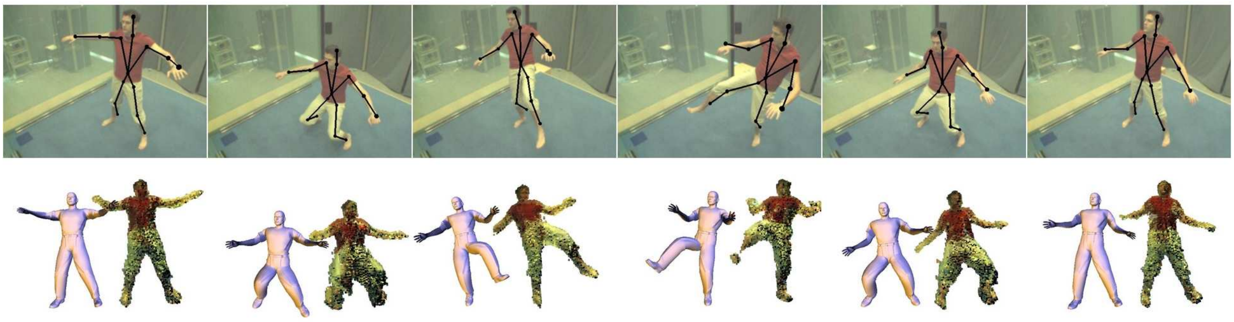


Figure 11. User bending knees and performing kicks.

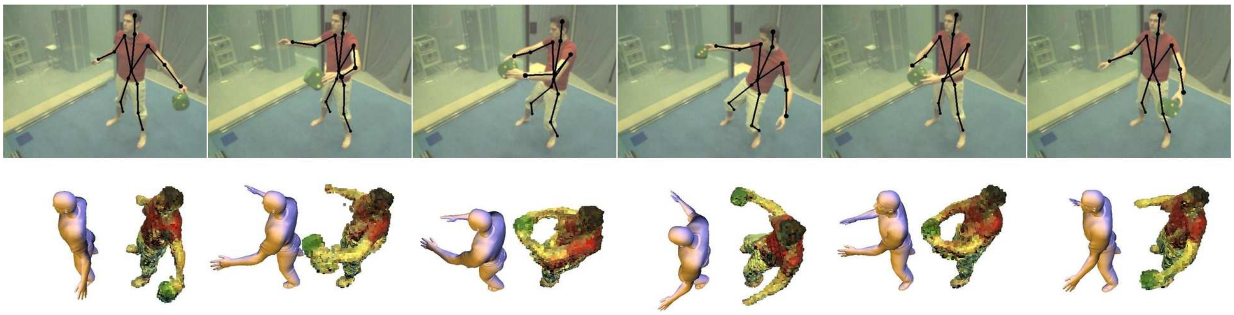


Figure 12. User handling a green cube.

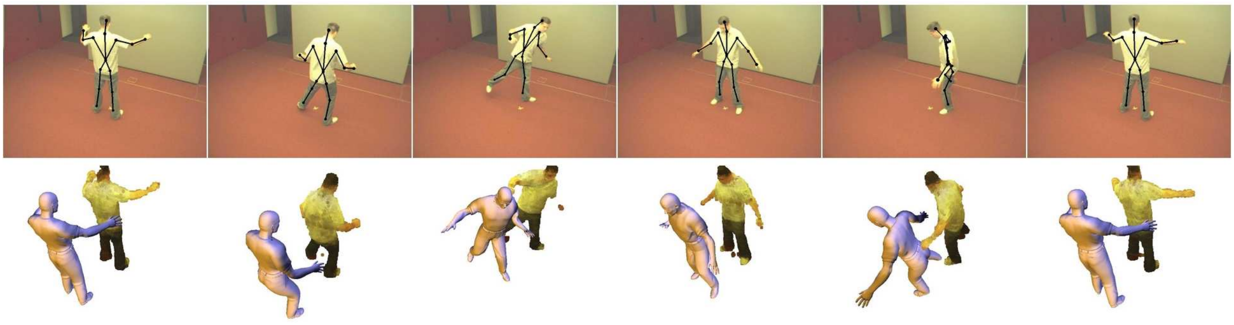


Figure 13. Full articulation with a 360 degrees turn.