BIO Web of Conferences **1**, 00046 (2011)
DOI: 10.1051/bioconf/20110100046

Capturing Motion Skills with Silhouette-Based Numerical Pose Estimation

Svenja Kahn[*] Cornelius Malerczyk[‡] Holger Graf[*] Ulrich Bockholt[*]

(*) Fraunhofer IGD, Germany
(‡) University of Applied Science Giessen-Friedberg
E-mail: svenja.kahn@igd.fraunhofer.de, cornelius.malerczyk@mnd.fh-friedberg.de, holger.graf@igd.fraunhofer.de, ulrich.bockholt@igd.fraunhofer.de

Abstract

The capturing of human movements is an important step for the analysis of human skills, e.g. for sports analysis or for learning-by-demonstration tasks. In this paper we introduce a new markerless pose estimation method which estimates human poses from silhouettes. The presented numerical pose estimation algorithm adapts a non-deterministical annealing schedule for silhouette based motion capturing. The pose is estimated by numerically minimizing the differences between the silhouettes of synthesized views of a 3D avatar and the silhouettes of the real person in the camera images. The evaluation results of simulation experiments quantify the trade-off between the accuracy and the execution time of the presented algorithm.

1. Introduction

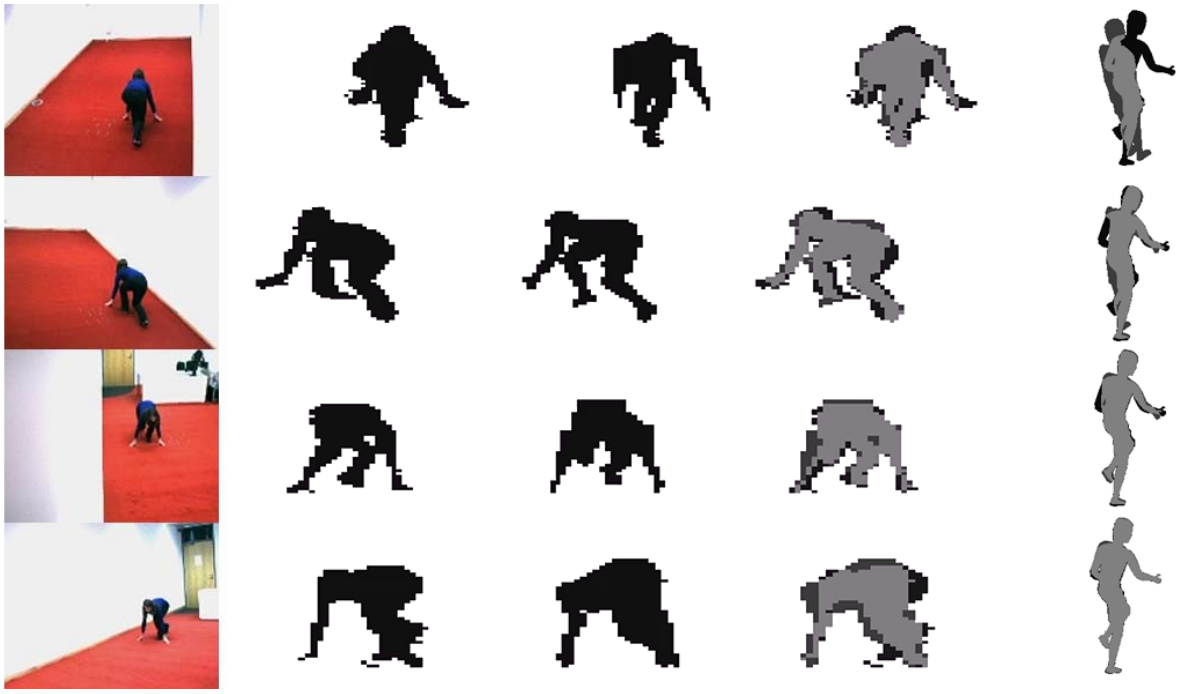
The capturing of human movements is an important step for the analysis of human skills [7]. Whereas marker-based motion capturing technologies have technologically come to maturity and are successfully used for the analysis of the movements of athletes, they have the drawback that markers need to be attached to the tracked person [8]. This requires special preparation and modeling tasks. Moreover, the attached markers can influence the natural movements of the person. Therefore there is a great need for markerless tracking technologies to overcome the limitations of marker-based pose estimation [5].

In this work we introduce a new markerless pose estimation method for multi-camera setups which estimates human poses from silhouettes. The problem setting is explained in section 2 and section 3 describes our numerical pose estimation algorithm which solves this pose estimation task. It adapts a simulated annealing schedule such that the differences between the real silhouettes and the silhouettes of a 3D avatar are iteratively minimized in a probabilistic manner. Due to the fact that the pose estimation algorithm is based on numerical optimization, there is a trade-off between the estimation accuracy and the required processing time. This trade-off is quantified by the results of the simulation experiments which are presented in section 4.

2. Silhouette-Based Pose Estimation

Our markerless motion capturing algorithm uses the silhouette images from a calibrated multi-camera setup for the pose estimation task. To calculate the silhouettes, background images are captured before the user enters the scene. Then the silhouettes of the person are calculated with a kernel density estimation based background substraction [3]. The first two columns of figure 1 visualize a set of color images as well as the silhouette images. Please note that the silhouette images were downscaled from $640 \cdot 480$ pixel to $160 \cdot 120$ pixel to speed up the pose estimation process.

Most previous approaches first reconstruct the 3D shape of the person from the silhouettes and use the reconstructed 3D shape for pose estimation [1][2]. However, in this paper we estimate the pose directly from the silhouettes, thereby avoiding the costly 3D reconstruction and 3D residual evaluation steps. We use an articulated 3D model of a human for the pose estimation and seek a configuration of joint angles such that the silhouette of the projected 3D avatar corresponds as well as possible to the real silhouettes. The third column of figure 1 visualizes a set of projections of the 3D avatar onto the camera images. Our avatar model has 22 degrees of freedom, each of which is a joint angle which needs to be estimated. An exhaustive search is not possible due to the enormous size of the feasible configuration space: If n is the number of discrete



(a) (b) (c) (d) (e)

Figure 1: (a) Color images (b) Silhouette images (resolution: $160 \cdot 120$ pixel) (c) Projected silhouettes of the virtual avatar (resolution: $160 \cdot 120$ pixel) (d) Silhouette residual between the real silhouettes and the projected avatar (resolution: $160 \cdot 120$ pixel) (e) Silhouette residual of simulation (resolution: $640 \cdot 480$ pixel)

values each joint angle can be set to (e.g. 180 if a rotation from $0°$ to $180°$ is modeled in $1°$ steps), the search space consists of n^{22} different joint rotations. Therefore we use a numerical optimization method with the goal to find a good approximation of the body pose within a reasonable amount of processing time.

3. Numerical pose estimation

Many numerical optimization methods such as downhill simplex methods or direction-set methods have been proposed for multidimensional search spaces. However, human pose estimation from camera images is a problem with many local minima (e.g. sets of silhouettes from different poses which look similar although the poses are quite different). Most numerical optimiziation methods are prone to get stuck in local minima rather than finding a solution which is close to the global optimum. However, annealing methods are able to find global extrema even in the presence of many local extrema [6]. They were developed in analogy to natural thermodynamical cooling processes, e.g. the cooling of metal: At the beginning the atoms move very quickly and their mobility decreases steadily during the cooling process. When metal cools slowly, the atoms are redistributed during the cooling process such that they line up in a state of minimal energy. This is not the case if the metal is cooled too quickly. Annealing is thus the process of a slow decrease of the energy state of a system in order to find a global optimum.

As a numerical optimization method, annealing minimizes an objective function E(X) where X is the state vector of the method E. In each step, a value of the state vector is randomly changed. This change can either increase or decrease the evaluated function value E(X). Whereas many other numerical optimization algorithms accept only ameliorations, annealing algorithms accept a change from state E_i to E_{i+1} with the probability $p = exp[-(E_{i+1} - E_i)/kT]$. If $E_{i+1} < E_i$, p is set to 1. This means that ameliorations are always accepted. However, demeliorations are also accepted with a certain probability. This probability is set by the temperature T (which is steadily decreased during the annealing process) and by the Boltzmann constant k which relates temperature to energy. This probabilistic acceptance function with its analogy to natural annealing is the core of annealing algorithms as it helps to avoid getting stuck in local minima. It was first introduced by Metropolis et al. [4] and is based on the thermodynamic Boltzmann probability distribution.

3.1. Adapting Simulated Annealing for Silhouette-Based Pose Optimization

We adapted the Simulated Annealing algorithm specified in [6] for silhouette-based pose estimation. To use this algorithm for pose estimation, the following algorithmic components were defined:

1. The possible system configurations of the state vector X: For pose estimation, these are the feasible degrees of freedom of the joint rotations (in our case the 22 rotation angles).

2. A generator of random changes: To test a new pose, the algorithm either sets one of the joints to a random value or slightly changes one of the joint rotations.

3. An objective function E(X) which is minimized by the Annealing algorithm: We seek to minimize the differences between the real and the artificial silhouettes. The real and the synthetic images are compared pixel by pixel to calculate the silhouette residual defined in equation 1. For a real silhouette image r and a synthesized silhouette image s we define the silhouette residual by:

$$\frac{\sum_{x,y}(r_{x,y} \cdot s_{x,y} == 0)}{\sum_{x,y}(r_{x,y} \cdot s_{x,y} == 0) + \sum_{x,y}(r_{x,y} \cdot s_{x,y} == 1)} \quad (1)$$

where $r_{x,y}$ is the binary value of the pixel at position (x,y) in the real silhouette image and $s_{x,y}$ is the binary value of the pixel at position (x,y) in the synthesized silhouette image. The binary value of a pixel is 1 if it is a silhouette pixel and its value is 0 if the pixel is a background pixel. The last two columns of figure 1 visualize the silhouette residual: The residual (which consists of the dark gray and black pixels) is minimized to maximize the (bright gray) overlapping area of both silhouettes.

4. An annealing schedule which tells how the temperature decreases and how many possible poses are tested per temperature step: In each temperature step the temperature is decreased by the factor 0.9. Furthermore we use two parameters (n_{over} and n_{limit}) to trade off between accuracy and calculation time: In each temperature step a maximum of $n_{over} \cdot 22$ poses are evaluated (there are 22 estimated joint angles and thus 22 degrees of freedom). The parameter n_{limit} sets the convergence criterion for the Simulated Annealing algorithm: If more than $n_{limit} \cdot 22$ changes are accepted within a temperature step, the algorithm continues with the next temperature step. If less than $0.2 \cdot n_{limit} \cdot 22$ changes are accepted in a temperature step, the algorithm finished (this is the convergence criterion).

4. Results

The experiments were conducted on an 3.07 GHz Intel Core i7 with an NVIDIA GeForce GTX 470. Our camera setup consists of four Firewire cameras which have a resolution of $640 \cdot 480$ pixel. The cameras are located in the corners of a room which measures $8m \cdot 4m$. To quantitatively evaluate the pose estimation algorithms with ground truth data we used the calibrated extrinsic and intrinsic parameters of the real cameras to generate synthesized camera images for the evaluation. We then compared the positions of the head, shoulders, elbows, hands, hip, knees and feet of the reference and the calculated body poses (the positional residual).

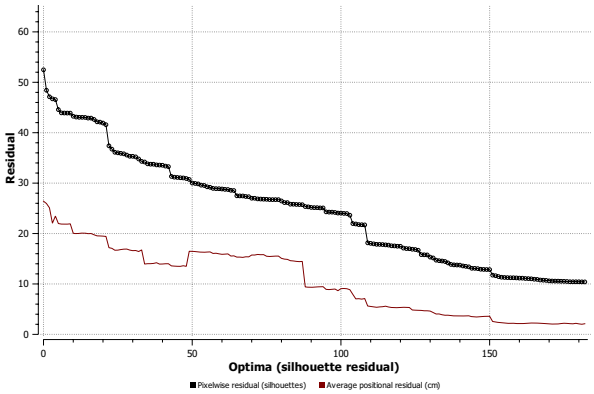


Figure 2: Silhouette residuals (black) and positional residuals (red) of the Simulated Annealing algorithm. Silhouette residual: y-value = residual as defined in equation 1. Positional residual: y-value = cm.

The Simulated Annealing algorithm can not directly evaluate the positional residuals of the body because these cannot be observed directly (except during a simulation). Thus it has to rely on the pixelwise silhouette residuals to estimate the accuracy of the calculated pose. Figure 2 visualizes the residuals of the body pose (red) during the minimization of the silhouette residuals (black) for a single run of the parameter set ($640 \cdot 480$, $n_{over} = 10$, $n_{limit} = 5$). Whereas an improvement of the silhouette residual does not always improve the real pose (e.g. at the 48th found silhouette residual the average body pose residual increases from 13.8cm to 16.4cm), finally the body pose converges to a good approximation of the real pose. The right column of figure 1 visualizes (from top to bottom) the silhouette residuals number 0, 60, 120 and 180 of figure 2.

Table 1 shows the performance of the presented algorithm in terms of accuracy and execution time. For each parameter set (image size, n_{over} and n_{limit}) the execution time was calculated by averaging ten execu-

Table 1: Simulated Annealing: Average execution time and pose estimation accuracy

Image size	n_{over}	n_{limit}	Max. tests per temp. step	Execution time	Temperature steps	Silhouette residual	Positional residual
$640 \cdot 480$	2	1	44	15.55s	32	0.2180	11.53 cm
$320 \cdot 240$	2	1	44	6.79s	28	0.2368	11.92 cm
$160 \cdot 120$	2	1	44	6.26s	22	0.2582	14.06 cm
$80 \cdot 60$	2	1	44	5.50s	21	0.2833	16.15 cm
$640 \cdot 480$	5	2	110	42.79s	31	0.1724	7.32 cm
$320 \cdot 240$	5	2	110	15.24s	30	0.1878	7.20 cm
$160 \cdot 120$	5	2	110	11.83s	22	0.2113	9.46 cm
$80 \cdot 60$	5	2	110	9.62s	13	0.2407	13.56 cm
$640 \cdot 480$	10	5	220	69.17s	25	0.1322	3.54 cm
$320 \cdot 240$	10	5	220	31.28s	22	0.1635	5.96 cm
$160 \cdot 120$	10	5	220	24.68s	18	0.1873	7.67 cm
$80 \cdot 60$	10	5	220	17.13s	14	0.2369	12.17 cm
$640 \cdot 480$	50	5	1100	1022.20s	74	0.0941	2.30 cm
$320 \cdot 240$	50	5	1100	567.65s	74	0.1105	2.78 cm
$160 \cdot 120$	50	5	1100	430.81s	74	0.1458	3.74 cm
$80 \cdot 60$	50	5	1100	311.35s	74	0.1974	9.80 cm

tions of the algorithm with the specified parameters. In each temperature step a maximum of $n_{over} \cdot 22$ poses were evaluated (there are 22 estimated joint angles and thus 22 degrees of freedom). If more than $n_{limit} \cdot 22$ changes were accepted within a temperature step, the algorithm continued with the next temperature step. If less than $0.2 \cdot n_{limit} \cdot 22$ changes were accepted in a temperature step, the algorithm finished (convergence criterion). The column "temperature steps" shows after how many temperature steps this convergence criterion was reached. This value is identical for the last four parameter sets because for these sets the convergence criterion was only reached as soon as the temperature had decreased to 0. The last column shows the positional residual. The accuracy of the estimated pose increases with the number of tested poses, so the parameters for the pose estimation can be chosen such that they trade off between the processing time and the required accuracy.

References

[1] S. Corazza, L. Mündermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. Andriacchi. A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Annals of Biomedical Engineering*, 34:1019–1029, 2006.

[2] J. Gall, B. Rosenhahn, and H.-P. Seidel. Clustered stochastic optimization for object recognition and pose estimation. In F. Hamprecht, C. Schnörr, and B. Jähne, editors, *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 32–41. Springer, 2007.

[3] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1186–1197, 2008.

[4] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[5] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104:90–126, 2006.

[6] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, 1992.

[7] L. Unzueta. Skills capture and transfer: Human motion capture. Tutorial in Robotics: Science and Systems Conference (RSS 2008).

[8] Vicon, 2011. http://www.vicon.com/.