Holistic Body Tracking for Gestural Interfaces

Christian Lange, Thomas Hermann and Helge Ritter

Neuroinformatics Group, Faculty of Technology, Bielefeld University,
Postfach 100131, D-33501 Bielefeld, Germany,
{clange,thermann,helge}@techfak.uni-bielefeld.de
http://www.TechFak.Uni-Bielefeld.DE/ags/ni/index.html

Abstract. In this paper we present an approach to track a moving body in a sequence of camera images by model adaptation. The parameters of a stick figure model are varied by using a stochastic search algorithm. The similarity of rendered model images and camera images of the user are used as quality measure. A refinement of the algorithm is introduced by using combined stereo views and relevance maps to infer responsible joint angles from the difference of successive input images. Finally, the successful application of various versions of the algorithm on sequences of synthetic images is demonstrated.

1 Introduction

In human-human communication gestures are frequently used to simplify communication or to emphasize and disambiguate verbal utterances [1]. While humans are able to interprete gestures from just seeing their communication partner, currently existing gesture recognition systems have to solve the complex problem of (i) understanding a visual scenery, (ii) analyzing the body motion for gestures and (iii) interpreting the gesture correctly in the context of other modalities like speech. Many approaches circumvent (i) by using tracking systems or markers. But sensors that need to be placed at the user are inconvenient. To enable a more natural human-computer interaction, we prefer gesture recognition by only using visual sensors. This leads to the task of extracting gestures from image sequences. An overview of vision-based gesture recognition systems can be found in [2, 3]. Most of the existing approaches use a fixed set of predefined gestures that are used like a command language. For natural human computer interaction, such restriction, however, should be avoided.

In general, gestures can be described by a sequence of body postures and each posture can be represented by a vector of joint angles of a stick figure model. From this perspective the first task of gesture recognition is to determine the vector of joint angles for the stick figure model from an image or a sequence of images.

Our system will have two or more cameras to observe the user. An internal model is used to represent and visualize the body posture which is represented by 34 angles. The model is updated about 100 times per frame by moving some

selected angles until the similarity between the rendered image and the input image reaches a (local) maximum, using a given similarity measure.

Section 2 describes details about the body model as well as our holistic tracking algorithm. Section 3 describes the environment for testing and tuning parameters of the holistic tracking algorithm. Results for different tracking approaches are presented in Section 4. The paper closes with conclusions and prospects for ongoing research.

2 Tracking by Image Adaptation

The problem of body posture recognition can be divided into two different cases, concerning the available context: (a) *single image posture estimation:* given a new image without any context and knowledge about the body posture, and (b) *tracking:* given a series of images and a valid model posture at time t, adapt the model so that it most likely describes the observations in the image at time $t+1$. The problem (a) has to be solved for instance at system start and is computationally much harder than (b), since much less prior knowledge is available. We plan to employ a hierarchical artificial neural network architecture based on work reported in [4] for learning the mapping from images to configurations. However, for a running system, tracking may offer better results in terms of an error measure in configuration space.

The system renders an image of its human model and compares it with the segmented input image. This is illustrated in the box "Adaptation loop" in the signal flow diagram (Fig. 3). To keep the angle vector of the model up to date, the system generates trials for new postures by varying individual angles by adding a Gaussian random number with zero mean and suitably scaled standard deviation. A trial is accepted, if it increases the similarity between the input and the current model image. This similarity is measured by the quality function (see Section 2.4). If performed at a sufficiently high rate, the stochastic optimization steps cause the configuration to follow/imitate the external person' s motions. A major question to be addressed to make this technique efficient for practical situations is deciding which angles should be varied and how far. This issue is taken up in the next section.

2.1 Algorithms for Model Adaptation

The adaptation algorithm changes the angles of the model and computes the difference between the input image and the rendered image of the model. Without any assumption which of the angles should be moved, the algorithm has to select one randomly. However, with such a simple strategy many trials modify "wrong" angles. The stochastic search can be accelerated by using a suitable measure that tells which angle (or angles) should be preferably changed.

One possible measure could be obtained as follows: Find out where in the image changes are located, and then infer from these local image differences to the angles whose modification is likely to cause changes in that region.

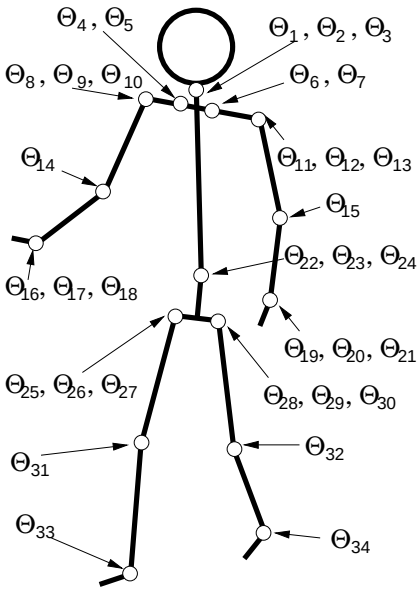
Fig. 1. Two postures of the stick figure (each rendered from two different points of view). Such images are compared to segmented input images to identify the posture of the input figure. Obviously it is helpful to have at least two views of a scene to disambiguate (self-)occlusions.

The ideal algorithm, however, should minimize the distance in configuration space from measuring distances in image space, because the distance in angle space indicates whether the posture of the internal model matches the observed person and the image space is the available input for the system.

2.2 Human Model

For an internal representation of body postures, it is necessary to agree upon a common body model. The more a model respects physical constraints of the human example, the more likely it will behave similar. The simplest model might characterize a human by its hand, shoulder and head coordinates in 3d space. Such a model is computationally not far from what a skin-color-based pattern detection algorithm can provide as its output, but it still is vulnerable to the possibility of representing impossible configurations, for instance a hand-head distance longer than the length of an arm. A much better description can be given by a skeleton model or stick figure model. A posture then is characterized by a vector of joint angles, and physical limitations can be incorporated into angle ranges. Our approach adopts this stick figure model, using length and angle range parameters as described by Badler (see [5] for details). The structure is shown in Figure 2. In practical situations, two different time scales for model adaptation have to be considered: (i) the model dynamics covers the joint angles and time-variant changes of the posture, while (ii) the model parameters cover person specific details like relative arm length or the person's total size. For reason of simplicity, model parameters are considered fixed for the following discussion.

Given a vector $\vec{\Theta}$ of 34 values for the bodies angles (see Fig. 2 for a list), a 3d rendering module based on the GL-library renders a corresponding image of the stick figure. For the further algorithms we binarized those images (see Fig. 1 for some examples). The size of the images was varied between 50×50 and 100×100 during the experiments. First results indicate that the adaptation is less accurate for smaller images, so a trade-off between efficiency and quality must be found. For most of the experiments two different views of the stick figure are used, since this can overcome problems with degeneracy from special single views and therefore leads to better adaptation than a single image. The



	name	lower lim. hard Θ^{LH}	lower lim. conv. Θ^{LC}	upper lim. conv. Θ^{UC}	upper lim. hard Θ^{UH}
Θ_1	neck z	-100	-30	30	100
Θ_2	neck y	-70	-20	30	70
Θ_3	neck x	-64	-20	20	64
Θ_4,Θ_6	clavicle x	-15	-10	10	30
Θ_5,Θ_7	clavicle y	-10	0	10	45
Θ_8,Θ_{11}	shoulder x	-63	-5	60	162
Θ_9,Θ_{12}	shoulder z	-127	-15	80	97
Θ_{10},Θ_{13}	shoulder y	-83	-15	60	211
Θ_{14},Θ_{15}	elbow y	0	0	140	159
Θ_{16},Θ_{19}	wrist z	-216	-90	0	26
Θ_{17},Θ_{20}	wrist x	-48	-10	10	37
Θ_{18},Θ_{21}	wrist y	-78	-45	30	95
Θ_{22}	waist y	0	0	10	60
Θ_{23}	waist x	-50	-10	10	50
Θ_{24}	waist z	-75	-20	20	75
Θ_{25},Θ_{28}	hip y	-10	-5	80	148
Θ_{26},Θ_{29}	hip x	-51	-5	5	54
Θ_{27},Θ_{30}	hip z	-56	-15	15	51
Θ_{31},Θ_{32}	knee y	0	0	80	146
Θ_{33},Θ_{34}	ankle y	-80	-45	0	20

Fig. 2. Angles and limits of the human model (angles of the left hand side have the same ranges as their counterpart on the right side). The larger ranges of hard limits will never be exceeded by a human, whereas the tighter range of convenience limits can.

displacement of the viewpoints is chosen as if the cameras are placed beside a projection wall or a big screen, because the real world scenario will be a user in front of such a display as described in Nölker [6].

2.3 Convenience of a Posture

The space of possible configurations is only very sparsely filled with realized gestures for two reasons: firstly angle ranges restrict the postures to a 34-dimensional hypercube. Secondly, even when assuming only 3 different, discrete settings per angle, the resulting 34-dimensional state grid would offer 3^{34} or more than 10^{16} configuration points, of which only a tiny fraction can be generated within human lifetime. Thus the actually occurring configurations are likely to live only on a limited submanifold of much lower dimension.

Our system has some mechanisms to avoid unnatural postures. Firstly, for all angles there are hard limits (Θ^{LH} and Θ^{UH}) that average people can't exceed. Secondly, tighter "convenience limits" (Θ^{LC} and Θ^{UC}) are introduced that most people won't exceed for a long time. The values are listed in Figure 2.

The convenience $\text{conv}(\vec{\Theta})$ of a posture $\vec{\Theta}$ is defined to be 1 if all angles are within their convenience limits, between 0 and 1 if at least one angle is out of

its convenience limits and 0 if at least one angle is beyond its hard limits:

$$\mathrm{conv}(\vec{\Theta}) = \prod_{i=1}^{34} \mathrm{conv}_i(\Theta_i) \tag{1}$$

$$\mathrm{conv}_i(\Theta) = \begin{cases} 0 & \forall \quad \Theta \notin [\Theta_i^{LH}, \Theta_i^{UH}] \\ \dfrac{\Theta - \Theta_i^{LH}}{\Theta_i^{LC} - \Theta_i^{LH}} & \forall \quad \Theta \in [\Theta_i^{LH}, \Theta_i^{LC}[\\ 1 & \forall \quad \Theta \in [\Theta_i^{LC}, \Theta_i^{UC}] \\ \dfrac{\Theta_i^{UH} - \Theta}{\Theta_i^{UH} - \Theta_i^{UC}} & \forall \quad \Theta \in]\Theta_i^{UC}, \Theta_i^{UH}] \end{cases} . \tag{2}$$

Thus the convenience measure gives higher values for postures that are closer to the normal center position.

2.4 Quality Measure

To decide whether an adaptation step has improved the model posture, the similarity of the two images must be computed. There are different possibilities of quality measures. Since images can be seen as large vectors of pixel values the euclidian distance is one way to compute their difference. Afterwards the reciprocal is computed, because the term "quality" implies to have higher values for more similar images. In case of two views the distances are added before computing the reciprocal:

$$Q = \frac{1}{||\vec{I}_{in}^{l} - \vec{I}_{model}^{l}|| + ||\vec{I}_{in}^{r} - \vec{I}_{model}^{r}||}. \tag{3}$$

The nonlinear transformation/mapping of the functions doesn't bother the adaptation algorithm because it just checks whether a trial increases the quality.

2.5 Adaptation Step Generator

To adapt the internal model to the input, the adaptation step generator randomly selects an angle j, and adds a Gaussian increment to it:

$$\Theta_j := \Theta_j + \mathcal{N}(0, \sigma) \cdot (\Theta_j^{UC} - \Theta_j^{LC}). \tag{4}$$

Then a model image is rendered and the quality of the new posture is measured. If the quality has increased, the step is accepted, otherwise it is rejected and taken back. Steps that lead to a posture with convenience less than 0.8 are also taken back. After trying a fixed number of N_{adapt} steps, the generator stops and should have reached a good approximation to the input image.

The parameters in this algorithm are the number of adaptation steps N_{adapt}, the variance for the Gaussian increment of those steps and the size of the images.

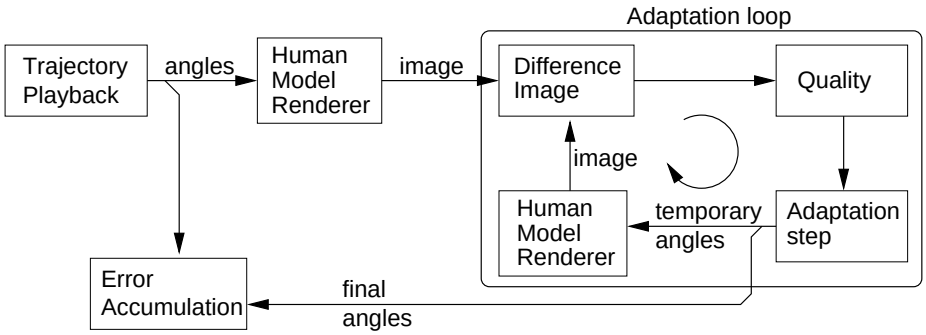


Fig. 3. Signal flow diagram to illustrate the testing environment. For each step of the presented trajectory, the "Adaptation loop"-algorithm tries to adapt the internal model to the input by analyzing the difference image.

3 Testing Environment

To test and optimize the algorithm we use rendered images as input instead of camera images. So we can easily compute the difference between the given angle vector and the adapted one in the model (see Fig. 3). Within this framework, we tried several adaptation algorithms. The distinctive features between the algorithms were the following: the computation of the difference between the images, the decision which angles are changed, the kind of image preprocessing and the size (variance) and number of the random steps.

3.1 Trajectory Generator

The trajectory generator creates a random sequence of postures. For each step it selects some angles at random. Then all of those angles are moved by small Gaussian steps with random size and direction. Steps that would lead to a posture with convenience less than 0.85 are rejected and not recorded.

To compare the different adaptation algorithms, we once generated a trajectory of $N_{traj} = 1000$ steps as a test dataset. The examined algorithm successively gets the images as input and tries to adapt the model to them.

3.2 Comparative Runs

Input trajectory and internal model both are initialized to a central rest position. After every 100 trajectory steps, the model is readjusted to the actual angle values of the input stick figure. Thus each algorithm has up to 10 trials to follow the trajectory. This adjustment is done because we want to compare the capability of following a given trajectory when starting close to it. After a sequence of 100 steps some algorithms are not close to the input anymore. After each step the difference in angle space between the given figure and the model

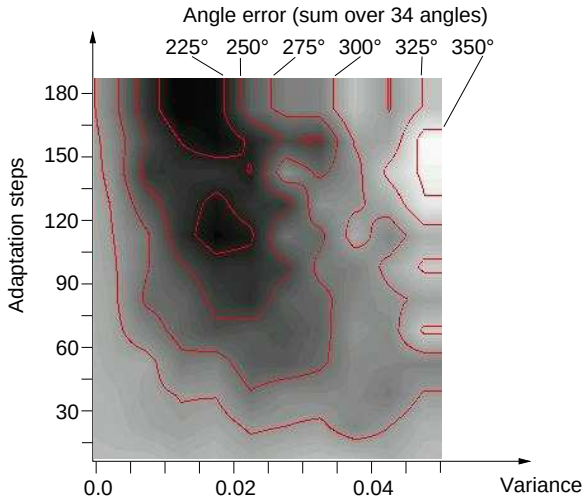


Fig. 4. Error surface with variance and number of adaptation steps as parameters shows that the default algorithm (two views, each 100×100 and unconstrained random selection of varied angle) causes smallest errors when having more than 100 adaptation steps and variance between 0.01 and 0.02.

is recorded as the sum of the deviations of all angles. We denote this error after step i by $E_i = ||\vec{\Theta}_{desired} - \vec{\Theta}_{model}||_1$. The mean error for the dataset

$$\overline{E} = \frac{1}{N_{traj}} \sum_{i=1}^{N_{traj}} E_i \tag{5}$$

is used to compare the different algorithms and parameters.

4 Results

The qualitative observation of the images shows that the adaptation of the model works fine most of the time.

The adaptation algorithm using equidistributed random choices to select an angle is the most simple method, because it needs no heuristic. It will be used as reference to rate the other algorithms. Applying it on several values for jumping variances and different numbers of adaptation-steps gives an error surface (Fig. 4).

The figure shows that the ideal variance is between 0.01 and 0.02 and the algorithm should use more than 100 adaptation steps. These results are obtained by using a stick figure renderer creating images from two views of the person of 100×100 pixels each. Using smaller images reduces the processing time but results in similar shaped error surfaces with a higher total error value. The two-views algorithm is superior to a one-image variant, even when using a higher image resolution such that the processing times become equivalent.

One starting-point for further enhancement should be the quality function. The current version tends to vary very strongly even if the input images differ only slightly. The attempt to smooth the images before computing the difference doesn't solve the problem, but leads to higher errors.

The testing environment not only accumulates the error but also counts how many of the final model steps reduce the error measured as the difference in configuration space. The algorithm using two views of 100×100 pixel each, did 70% steps towards a lower error. All other versions achieved more than 50%.

5 Conclusion

This paper presents an approach for tracking a human model by using an adaptive stochastic search in model space. For investigating and optimizing the performance of the algorithm, we generated test images by using the model itself. The results shed light on important aspects to be addressed in stochastic tracking, namely the trade-off between acceptance rate and search radius (represented by covariance structure of the transition distribution). Optimal values for jumping covariance and number of steps were experimentally derived for different similarity measures and search algorithms.

The next step will be to replace the currently used synthetically rendered input images by camera images. Necessary prerequisite is a segmentation of camera pictures into foreground and background. For the process of image segmentation, we intend to incorporate both knowledge of the model, skin color and the distribution patterns for image regions.

References

1. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press, Chicago (1992)
2. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding: CVIU **81** (2001) 231–268
3. Lenman, S., Bretzner, L., Thuresson, B.: Computer vision based recognition of hand gestures for human-computer interaction. Technical Report TRITA-NA-D0209, University of Stockholm, Department of Numerical Analysis and Computer Science, CID, Centre for User Oriented IT Design (2002)
4. Nölker, C., Ritter, H.: Visual recognition of continuous hand postures. IEEE Transactions on Neural Networks, Special Issue Multimedia **13** (2002) 983–994
5. Grosso, M., Quach, R., Otani, E., Zhao, J., Wei, S., Ho, P., Lu, J., Badler, N.: Anthropometry for computer graphics human figures. Technical Report MS-CIS-87-71, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, PA (1987)
6. Nölker, C., Ritter, H.: Illumination independent recognition of deictic arm postures. In: Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Aachen. (1998) 2006–2011