Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2009)
H. Prautzsch, A. Schmitt, J. Bender, M. Teschner (Editors)

Interactive Inverse Kinematics for Human Motion Estimation

M. Engell-Nørregård , S. Hauberg , J. Lapuyade , K. Erleben and K.S. Pedersen

{mort, hauberg, lapuyade, kenny, kimstp}@diku.dk,
The eScience Center, Dept. of Computer Science, University of Copenhagen

Abstract

We present an application of a fast interactive inverse kinematics method as a dimensionality reduction for monocular human motion estimation. The inverse kinematics solver deals efficiently and robustly with box constraints and does not suffer from shaking artifacts. The presented motion estimation system uses a single camera to estimate the motion of a human. The results show that inverse kinematics can significantly speed up the estimation process, while retaining a quality comparable to a full pose motion estimation system.
Our novelty lies primarily in use of inverse kinematics to significantly speed up the particle filtering. It should be stressed that the observation part of the system has not been our focus, and as such is described only from a sense of completeness.
With our approach it is possible to construct a robust and computationally efficient system for human motion estimation.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Animation—Computer Graphics [I.3.5]: Physically based modeling—Numerical Analysis [G.1.6]: Constrained Optimization—Numerical Analysis [G.1.6]: Nonlinear Programming—Image Processing and Computer Vision [I.4.6]: Scene Analysis—Probability and Statistics [G.3]: Markov processes—

Keywords: Inverse Kinematics, Nonlinear, Joint Limits, Constraints, Motion Estimation

1. Introduction

Inverse kinematics has found widespread use as an intuitive posing system for articulated figures in Computer Graphics [Fed03] and for motion planning in Robotics [SK06]. We propose a novel use of inverse kinematics as a means to reduce the dimensionality of a particle filter based tracking algorithm. Preliminary results show that this approach significantly reduces the time demands compared to existing approaches with comparable results. The method may make it possible to perform visual tracking of general human motion in an interactive way.

Three dimensional human motion analysis is the process of estimating the configuration of body parts over time from sensor input [Pop07]. Traditionally motion capture equipment has been used to track this motion. In motion capture, markers are attached to the body and then tracked in 3 dimensions. Usually this requires multiple tracking devices so motion capture is most often performed in pre–calibrated laboratory settings.

Our long term goal is to use human motion analysis as part of a physiotherapeutic rehabilitation system. In the system, the motion of a patient is tracked and analyzed during exercise sessions performed both at the clinic and at the patient's home. This application rules out the traditional motion capture approach, and results in the need for a simpler system with fewer cameras. These limitations make it necessary to formulate a better model to compensate for the lack of information from the image data. Our approach is to use inverse kinematics as a way to impose this information.

Our approach utilizes the fact that the pose of a skeleton can be deduced from the end–effector positions. While a full human skeleton may have more than 100 degrees of freedom, the end–effectors space of the same articulated figure can have e.g. only 15 degrees of freedom (positional parameters of head, hands and feet). This dimensionality reduction accounts for a significant speedup in the computational demands of the system, compared to analyzing the motion in the full pose configuration space of the skeleton. The focus

DOI: 10.2312/PE/vriphys/vriphys09/077-084

of this paper is the description of the parts of the system which relates to this dimensionality reduction,

1.1. Related Work

Estimation of human motion is an inherently high dimensional problem, since human motion is both diverse and has many degrees of freedom. The traditional approach to reducing the dimensionality has been to utilize manifold learning, i.e. to try and restrain the motions in a subspace of the full space. This approach is used in [WFH08, UFF06, LCPS08], but seems to be most suited for a constrained set of motions like walking or golf–swinging. We want to be able to process a larger set of motions, and thus need some other means of reducing the dimensionality. Inverse kinematics has been used in motion estimation before [ST03], but not as a dimensionality reduction tool. Our work differs in that it utilizes the posing abilities of the inverse kinematics system to infer the pose of the remaining joints. Thus, the estimation can be performed on the end–effector joints only.

Numerous different approaches to solving the inverse kinematics problem have been tried. *Jacobian transpose* and *Jacobian inverse* are the traditional methods [Wel93]. Cyclic coordinate descent is a simple greedy algorithm that is popular for its simplicity. However, it has weak convergence properties especially for large articulated mechanisms [Fed03]. Closed form solutions have also been developed [ML05], but they can only be used on specific low dimensional articulated figures. In practice, this makes the solutions less suitable for modeling the inter-dependency between individual limbs of the human body. [ZB94] introduced the formalism of phrasing the inverse kinematics problem using non–linear optimization.Active set methods like [ZB94] could also be chosen but they are complicated to implement and induces book keeping which our method does not.

Our inverse kinematics system is based on the robust system described in [ENE09b] that works well with box constraint joint limits.The main difference is that our solver uses a gradient projection instead of the projected conjugate gradient.

The motion estimation system used is the one described in [HLEN*09]. While [HLEN*09] concentrated on the motion estimation, this paper focuses on the interactive inverse kinematics solver.

1.2. Organization of the Paper

The paper is organized as follows. Sec. 2 describes the inverse kinematics system used in the solution. The theoretical foundation for the inverse kinematics solver is explained along with arguments for the choice of the joint limit model. Given this inverse kinematics solver we can construct a full motion estimation system. In Sec. 3 we describe the theory of motion estimation and give an overview of how our system is constructed. In Sec. 4 we present the results of our tests and compare our approach to a traditional particle filtering method in the full pose space. We discuss our results and the benefits of the new approach. In Sec. 5 we conclude and present possible avenues for improvements, optimizations and future work.

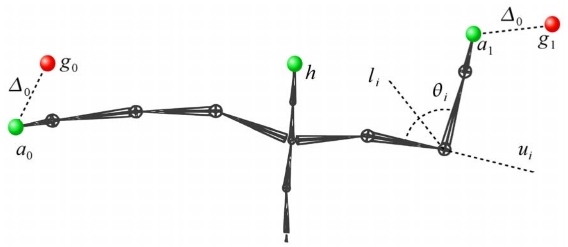


Figure 1: *An illustration of the kinematic model. End–effector positions are shown as green dots, while the desired positions (goals) are shown as red dots.*

2. Interactive Inverse Kinematics

Inverse kinematics is the problem of manipulating the pose of a skeleton, in order to achieve a desired pose disregarding inertia and forces. As shown in [ZB94], the problem can be posed as a non–linear optimization problem.

In the context of human modeling, a skeleton is often modeled as a collection of rigid bodies connected by rotational joints of 1–3 degrees of freedom. An example is shown in Fig. 1. All joints are constrained in their rotation, as exemplified by joint i in Fig. 1 with l_i and u_i showing the limits of the angle θ_i.

To compute the position and orientation of a joint in space, we perform a transformation of the bone relative to its parent joint. The transformation consists of a rotation and a translation corresponding to the shape and orientation of the joint, relative to its parent. These transformations are then nested to create chains of joints. Each chain ends in an end–effector, which can be regarded as the handle for controlling the chain. Thus, the full transformation of a joint from local space to global space can be performed.

The inverse kinematics problem can be formally stated as follows. Given the set of joint parameters θ we can change these and thus influence the position and orientation of the end–effector $a = F(\theta)$. Commonly this is known as forward kinematics. Inverse kinematics is concerned with the inverse problem: given a desired end–effector goal position, g, one seeks the value of θ such that,

$$\theta = F^{-1}(g) \quad . \tag{1}$$

Since a solution to this problem is not guarantied to exist, it is more practical to minimize the squared distance between

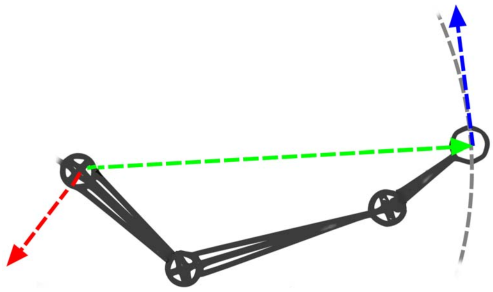


Figure 2: *Finding the rotational derivative of a joint. The derivative is found as the cross product of the rotational axis (red) and the vector from joint to end–effector (green). The resulting tangent vector is shown in blue.*

the goal and the end-effector, rather than solving the equation. The least–squares criterion makes good sense in the inverse kinematics context since it is essentially a fitting problem. By posing the problem as a least–squares fitting problem, we are given a number of possible methods to solve the problem [NW99]. Furthermore, the least squares optimization approach is more stable in or near singularities than traditional approaches like e.g. Jacobian inverse [BK05].

The objective function for a branched articulated mechanism can be constructed using the end–effector function. Given a skeleton containing K kinematic chains, each with exactly one end–effector, we agglomerate the K end–effector functions into one function,

$$a = \left[a_1^T \dots a_K^T\right]^T = \left[F_1(\theta)^T \dots F_K(\theta)^T\right]^T = F(\theta) \quad , \tag{2}$$

where a_j is the world coordinate position of the j^{th} end–effector and $F_j(\theta)$ is the end–effector function corresponding to the j^{th} kinematic chain. Using the agglomerated end–effector function, we create the objective function,

$$f(\theta) = (g - F(\theta))^T W (g - F(\theta)) \quad , \tag{3}$$

where $g = \left[g_1^T \quad \dots \quad g_K^T\right]^T$ is the agglomerated vector of end–effector goals, and W is some weight matrix that can be used to model the relative importance of different goals. The optimization problem is then,

$$\theta^* = \arg\min_{\theta} f(\theta) \quad \text{s.t.} \quad l \le \theta \le u \; . \tag{4}$$

Here l is a vector containing the minimum joint limits and u is a vector of the maximum joints limits.

Any constrained non-linear optimization method may be used to solve the problem. In this paper we use a simple, yet effective, gradient projection method with line search [NW99]. The motivation for this choice was the speed and robustness of this approach, and the fact that this particular approach is tailored specifically to handle box constraints.

To perform the optimization, we need to compute the gradient G of $f(\theta)$. This is given by

$$G = (g - a)J^T , \tag{5}$$

where J denotes the Jacobian matrix of $f(\theta)$. **Hauberg sez: This is wrong. The W matrix is missing, and there might be a sign error.** The entries of the Jacobian depend on the type of goal. In this paper, we will explain positional goals and orientation goals, both of which are included in the developed inverse kinematics solver. For rotational joints, these can easily be computed.

For each chain, the Jacobian matrix contains a 3 × 1 entry for positional goals. This is computed as the cross product of the rotational axis and the vector from the joint to the end–effector, as shown in Fig. 2.

Orientational goals are represented by two 3 × 1 vectors, yielding a 6 × 1 entry in the Jacobian. This can be computed as the cross product of the rotational axis of the joint and the unit vectors representing the orientation of the end–effector.

The two goal types can be combined to create further goal types such as line goal or partial orientation goals. For more details of this, see [ZB94].

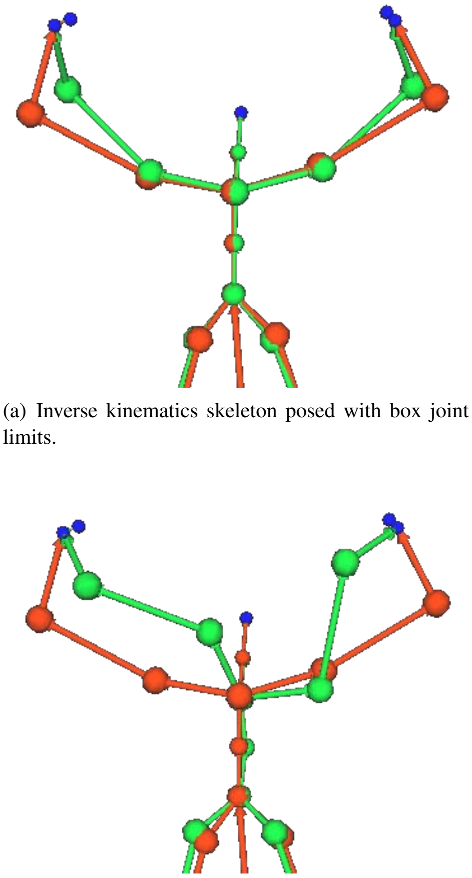
2.1. Joint Limits

In both animation and motion analysis, it is necessary to use a fast solver that can handle joint limits robustly. Using box constraints means that each limit is a simple minimum or maximum limit on a single variable. Even though these joint limits are very simple, they are the most common choice in real time simulation due to their simplicity. Fig. 3(b) shows a human upper body posed without joint limits. The green skeleton shows the inverse kinematics solution. Fig. 3(a) shows the same pose with box constraints. On both figures, the reference pose is shown in red. As can be seen, the constrained pose is not exactly the same as the reference, but it is much closer than the unconstrained solution. The small discrepancy is due to the inherent redundancy of human motion, and not the simplicity of the box constraints. It would be possible to use more detailed joint limits, but these would come at an increased computational cost. Setting up joint limits can be a time consuming process. It is quite intuitive though and methods exist to automate the process [ENE09a].

We are now ready to include the inverse kinematics solver in the visual motion estimation system.

3. Visual Motion Estimation

Visual motion estimation is the process of infering the motion of a moving object from a sequence of images. In this paper we wish to infer the 3 dimensional pose of a human moving in front of a camera. In terms of Bayesian statistics, this boils down to estimating the distribution $p(s_t|I_{1:t})$ at each time step t. Here s_t denotes some representation of



(a) Inverse kinematics skeleton posed with box joint limits.

(b) Inverse kinematics skeleton posed without joint limits.

Figure 3: *A human upper body posed using inverse kinematics, with and without joint limits. The inverse kinematics skeleton is shown in green, a motion capture reference is shown in red. Notice that regardless of the joint limit model, there is some redundancy in the legal poses.*

the human joint configuration at time t, and $I_{1:t} = \{I_1, \dots, I_t\}$ denotes all images observed so far. The actual choice of representation will be discussed in Sec. 3.3.

Assuming that the current pose s_t only depends on the previous pose (i.e. the s's form a first order Markov chain) it can be proved that the distribution of the current pose can be estimated recursively as [CGM07],

$$p(s_t|I_{1:t}) \propto p(I_t|s_t) \int p(s_{t-1}|I_{1:t-1}) p(s_t|s_{t-1}) \mathrm{d}s_{t-1} \quad . \qquad (6)$$

This update equation consists of three simple terms: $p(I_t|s_t)$ measures how likely a given pose is, when compared to the current image, $p(s_{t-1}|I_{1:t-1})$ is the result from the previous iteration, and $p(s_t|s_{t-1})$ is a prediction of the current pose based solely on the previous pose. The integral can thus be interpreted as a prediction of the current pose based on a motion model and all previously observed images.

From a practical point of view, the integral in Eq. 6 is hard to evaluate, and approximations are required. Here we employ a standard Markov Chain Monte Carlo (MCMC) technique known as the *particle filter* [CGM07]. This algorithm reduces the pose estimation to performing the following steps iteratively:

1. Draw a set of random samples $s_t^{(n)}$, $n = 1, \dots, N$, from the predictive distribution $p(s_t|s_{t-1}^{(n)})$.
2. Compare each sample to the current image by computing a normalised weight $w_t^{(n)} \propto p(I_t|s_t^{(n)})$ for each sample.
3. Discard samples with low weights by sampling with replacements among the samples such that a sample is kept with probability $w_t^{(n)}$.

We do not provide the full details of the approach here, and the interested reader is referred to [HLEN*09] and [CGM07] for details.

An immediate question is, how many samples should be used in the above scheme? Unfortunately, it turns out that unless strong predictive models are available, the number of needed samples is exponential in the dimensionality of the pose representation s_t. Thus, it is of great practical importance to keep the representation as low dimensional as possible.

To perform the motion estimation we thus only need to have a system $p(s_t|s_{t-1})$ for making predictions about future poses, and a system $p(I_t|s_t)$ for comparing a pose to the current image.

3.1. Predicting Future Poses

In general it is hard to predict future poses unless the type of motion (e.g. *walking*) is known in advance. We therefore settle for a simple linear extrapolation scheme for making predictions. Specifically, we let

$$s_t^{(n)} = s_{t-1}^{(n)} + \Delta_{t-1} + \varepsilon \quad , \qquad (7)$$

where $\Delta_{t-1} = s_{t-1}^{(n)} - s_{t-2}^{(n)}$ and ε is normal distributed noise. This is a standard predictive scheme in motion estimation which assumes that the pose sequence s_t forms a second order Markov chain.

3.2. Comparing Poses to Images

To be able to compare a pose to an image, we need to define $p(I_t|s_t)$. The idea behind this system is to project the pose onto the current image, and compare each limb independently to the image. We use a simple Markov Random Field (MRF) model to describe the appearance of a limb, in which the limb appearance statistics is described by histograms of a set of descriptive features capturing texture and color information. Given a limb L, the corresponding likelihood $p(I_t|L)$, takes the form of a Gibbs distribution with an energy functional of the form

$$-\log p(I_t|L) = \alpha_T d_T^2(H^{T_m}, H^T|L) + \alpha_C d_C^2(H^{C_m}, H^C|L) + \alpha_B d_B^2(H^{B_m}, H^B|L) + \text{constant} \ , \tag{8}$$

where $d_T^2(H^{T_m}, H^T|L)$ is the distance between a texture model H^{T_m} and the observed texture H^T. $d_C^2(\cdot)$ and $d_B^2(\cdot)$ are the similar counterparts of the color and background models. The α parameters control the relative importance of the individual terms.

The texture and color models are based on the same principle, which boils down to computing a normalized histogram of a descriptive feature within the limb and comparing that with a normalized model histogram. When comparing the histograms we are using the earth mover's distance [RTG98], which makes the energy functional invariant to small perturbations and shifts of the histograms.

The background model is based on a simple thresholding of the absolute difference between a background image and the current image. This binary image is then compared to a simple binary rendering of the pose. We then define $d_B^2(H^{B_m}, H^B|L)$ as the number of pixels where the two binary images do not agree.

Using Eq. 8 we can compute the log-likelihood of observing a given limb L in the current image. To compute the log-likelihood of the entire pose we assume independence of the limbs and simply sum up the contributions of all limbs, i.e.

$$-\log p(I_t|s_t) = -\sum_k \log p(I_t|L_k) \ . \tag{9}$$

For more details of this scheme, see [HLEN*09].

3.3. Representing a Pose

One final question that needs to be answered before the motion estimation system is complete, is how the pose s_t should be represented. The straight-forward approach is to represent the pose as the vector of joint angles in the kinematic skeleton, i.e. $s_t \equiv \theta_t$. We will denote the resulting system a *full-pose tracker*. Such an approach is for instance taken in [SBF00].

The problem with the full-pose tracker is the dimensionality of the representation. As mentioned in Sec. 3, the number of samples needed in the particle filter is exponential in the dimensionality of the representation. Even for simple kinematic skeletons, the number of joint angles easily exceeds 30, which renders the particle filter computationally very expensive. This in turn makes real-time motion estimation intractable.

As an alternative we propose to use a set of end-effectors to represent the pose. Specifically, we use the 3 dimensional coordinates of the head and the hands to represent an upper body pose. That is, we define

$$s_t \equiv [x_{hand_0}, x_{hand_1}, x_{head}] \ , \tag{10}$$

where x_{hand_0} is the 3 dimensional coordinate of one of the hands and so forth.

When comparing such a pose to the current image we need the position of the individual limbs, which is not readily available in this representation. For this, we use inverse kinematics to compute the joint angles using the head and hands as end-effectors. This is possible due to the robustness of the inverse kinematics system. We will denote this system an *end-effector tracker*. It should be noted that the inverse kinematics problem often has several solutions, which unless handled, may lead to some loss of accuracy.

4. Results and Discussion

To verify the quality and to measure the time improvement we performed some simple tests. The test consisted of estimating the motion of a person sitting on a chair moving his arms about. The resulting skeleton had 3 end–effectors , the hands and the head. The skeleton was fixed at the hip and the legs were not modeled. The method can handle a full skeleton but for this preliminary tests we chose a simple skeleton. The original video clip used was app. 45 seconds long, at 15 fps.

The purpose of the tests was to compare the time expenditure and quality of our method to a traditional method.

We performed tests and timing of the system on a Lenovo T400 Thinkpad®with an Intel ®core$^{\text{TM}}$2 duo 2.40 Ghz.

Fig. 4, 5 and 6 show selected frames from an image sequence with the pose estimation results superimposed. The images in each of the figures correspond to the 32nd, 94th, 126th and the 196th frame of the sequence. A video version of the test results are available at: `http://humim.org/vriphys2009/`

The first test was a traditional full pose motion estimation without inverse kinematics using 100 particles. Fig. 4 shows the result. Here the system quickly looses track of one arm and produces a large amount of shaking in the motion estimation. The computations took approximately five minutes on the test hardware. We then increased the number of particles to 5000, which resulted in a successful motion estimation with only little shaking. Unfortunately, this required more than 10 hours of computation time. The results of this test can be seen in Fig. 5.

The final experiment was run using the inverse kinematics system for pose calculation, making it possible to track in only 9 dimensions. Only 25 particles were used and the running time was approximately 5 minutes. This is a speedup factor of approximately 120 compared to the 5000 particle run and a comparable quality, while the 100 particle run is comparable in time spent, but in this case, the quality of the inverse kinematics tracker is much better.

The results show that motion estimation in end–effector space is possible and that large speedups can be achieved using this approach. Our long term goal is to create a motion estimation system for use in a physiotherapeutic rehabilitation program and here it is essential to have real–time performance, in order to provide feedback to both patients and therapists. Estimation in end–effector space makes this requirement more feasible compared to estimation in full–pose space.

Using only a single camera gives no depth information which means that the system has difficulty in placing the goals correctly in this direction. This problem might be solved by using a camera type which can infer some depth information such as a stereo camera or a time of flight camera. Weighting the depth parameter of the goals with a small weight might also help since this would give the inverse kinematics solver more freedom in the placement with regards to the depth. It could however, also result in poses which would fit badly with the visual data so it could not stand alone.

5. Conclusion and Future Work

In this paper we presented an application for interactive inverse kinematics as dimensionality reduction for sequential Bayesian motion estimation. We have described how inverse kinematics can be used to drastically cut down the computational demands of a human motion estimation system. The motion estimation method used, is particle filtering which is a highly parallel process. Thus, it is viable to bring the motion estimation into the real time domain by combining the presented method with a parallelized particle filter. The tracker used in this paper does not utilize parallelization since the focus has been on measuring the speed improvement obtained from the inverse kinematics system. For this reason we have also chosen to concentrate on the inverse kinematics based dimensionality reduction and its impact on the performance. Thus, focus has not been on the observation part of the motion estimation.

The inverse kinematics approach used is the simplest possible with equally weighted positional goals for the hands and head. This can be improved by weighting the parameters such that the depth of limbs are weighted differently from the rest to reflect the difficulty in estimating depth due to using only one camera. Inclusion of a prioritized inverse kinematics scheme could improve the positioning of intermediate goals by utilizing the redundancy of the solution. A more detailed joint limit model might also prove to be an interesting addition. Also, a more detailed experimental validation, e.g. by comparing the tracker with motion capture data acquired simultaneously with the image data, is planned.

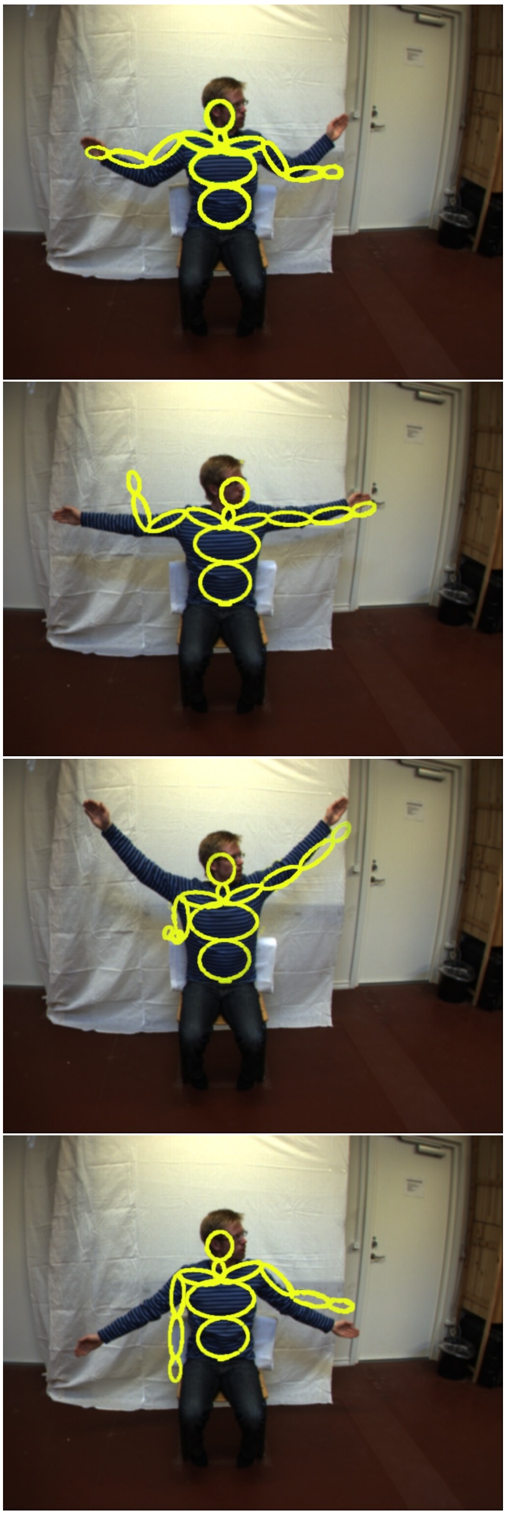


Figure 4: *Traditional motion estimation in full pose space using 100 particles. Notice that the superimposed figure looses track of the person in the image.*

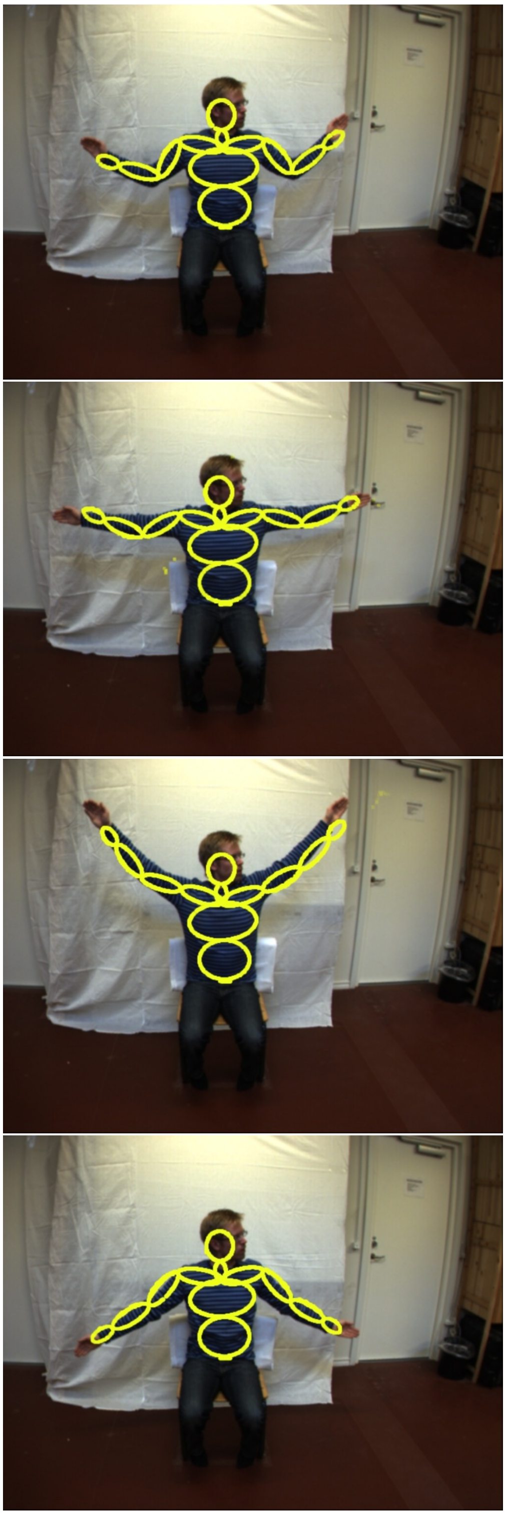


Figure 5: *Traditional motion estimation in full pose space using 5000 particles. The results of this motion analysis are satisfactory but the computation took more than 10 hours.*

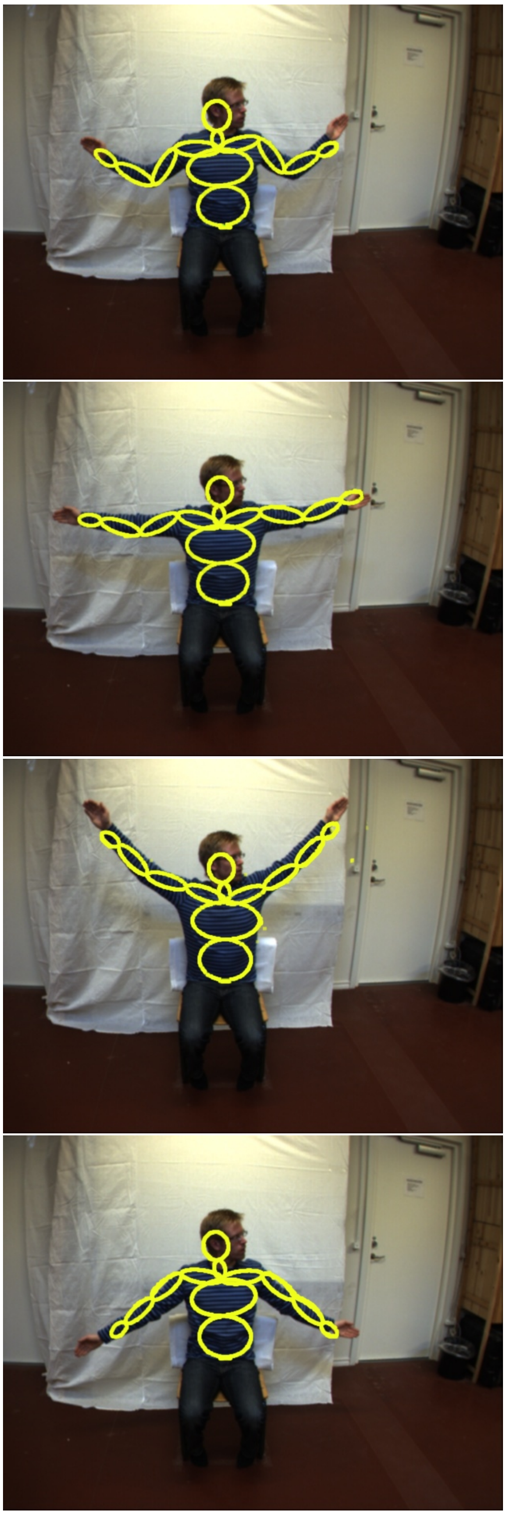


Figure 6: *Motion estimation using inverse kinematics with 25 particles. As it can be seen the results are very similar to Fig. 5. However the time spent was only 5 minutes.*

Acknowledgments

This research was done as part of the HUMIM project under the eScience Centre, Faculty of Science, University of Copenhagen.

The research was partly funded by the EU Marie Curie Research Training Network VISIONTRAIN MRTN-CT-2004-005439 and an ERCIM Post Doc scholarship.

References

[BK05] BUSS S. R., KIM J.-S.: Selectively damped least squares for inverse kinematics. *journal of graphics tools 10*, 3 (2005), 37–49. 3

[CGM07] CAPPÉ O., GODSILL S. J., MOULINES E.: An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE 95*, 5 (2007), 899–924. 4

[ENE09a] ENGELL-NØRREGÅRD M., ERLEBEN K.: Estimation of Joint types and Joint Limits from Motion capture data. In *WSCG 2009: 17-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2009). 3

[ENE09b] ENGELL-NØRREGÅRD M., ERLEBEN K.: A Projected Non-linear Conjugate Gradient Method for Interactive Inverse Kinematics. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling* (2009). 2

[Fed03] FEDOR M.: Application of inverse kinematics for skeleton manipulation in real-time. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics* (New York, NY, USA, 2003), ACM Press, pp. 203–212. 1, 2

[HLEN*09] HAUBERG S., LAPUYADE J., ENGELL-NØRREGÅRD M., ERLEBEN K., PEDERSEN K. S.: Three dimensional monocular human motion analysis in end-effector space. In *Energy Minimization Methods in Computer Vision and Pattern Recognition* (August 2009), Cremers D. et al., (Eds.), Lecture Notes in Computer Science, Springer, pp. 235–248. 2, 4, 5

[LCPS08] LU Z., CARREIRA-PERPINAN M., SMINCHISESCU C.: People Tracking with the Laplacian Eigenmaps Latent Variable Model. In *Advances in Neural Information Processing Systems 20*, Platt J., Koller D., Singer Y., Roweis S., (Eds.). MIT Press, Cambridge, MA, 2008, pp. 1705–1712. 2

[ML05] MORADI H., LEE S.: Joint limit analysis and elbow movement minimization for redundant manipulators using closed form method. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part II* (2005), Huang D.-S., Zhang X.-P., Huang G.-B., (Eds.), vol. 3645 of *Lecture Notes in Computer Science*, Springer, pp. 423–432. 2

[NW99] NOCEDAL J., WRIGHT S. J.: *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999. 3

[Pop07] POPPE R.: Vision-based human motion analysis: An overview. *Computer Vision Image Understing 108*, 1-2 (2007), 4–18. 1

[RTG98] RUBNER Y., TOMASI C., GUIBAS L.: A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision, 1998* (1998), pp. 59–66. 5

[SBF00] SIDENBLADH H., BLACK M. J., FLEET D. J.: Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of ECCV'00* (2000), vol. II of *LNCS 1843*, Springer, pp. 702–718. 5

[SK06] SENTIS L., KHATIB O.: A whole-body control framework for humanoids operating in human environments. In *ICRA* (2006), IEEE, pp. 2641–2648. 1

[ST03] SMINCHISESCU C., TRIGGS B.: Kinematic Jump Processes for Monocular 3D Human Tracking. In *In IEEE International Conference on Computer Vision and Pattern Recognition* (2003), pp. 69–76. 2

[UFF06] URTASUN R., FLEET D. J., FUA P.: 3D People Tracking with Gaussian Process Dynamical Models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society , Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 238–245. 2

[Wel93] WELLMAN C.: *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation*. Master's thesis, Simon Fraser University, 1993. 2

[WFH08] WANG J. M., FLEET D. J., HERTZMANN A.: Gaussian Process Dynamical Models for Human Motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*, 2 (2008), 283–298. 2

[ZB94] ZHAO J., BADLER N. I.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Trans. Graph. 13*, 4 (1994), 313–336. 2, 3