Predicting Articulated Human Motion from Spatial Processes

Søren Hauberg · Kim Steenstrup Pedersen

the date of receipt and acceptance should be inserted later

Abstract We present a probabilistic interpretation of inverse kinematics and extend it to sequential data. The resulting model is used to estimate articulated human motion in visual data. The approach allows us to express the prior temporal models in spatial limb coordinates, which is in contrast to most recent work where prior models are derived in terms of joint angles. This approach has several advantages. First of all, it allows us to construct motion models in low dimensional spaces, which makes motion estimation more robust. Secondly, as many types of motion are easily expressed in spatial coordinates, the approach allows us to construct high quality application specific motion models with little effort. Thirdly, the state space is a real vector space, which allows us to use *off-the-shelf* stochastic processes as motion models, which is rarely possible when working with joint angles. Fourthly, we avoid the problem of accumulated variance, where noise in one joint affects all joints further down the kinematic chains. All this combined allows us to more easily construct high quality motion models. In the evaluation, we show that an activity independent version of our model is superior to the corresponding state-of-the-art model. We also give examples of activity dependent models that would be hard to phrase directly in terms of joint angles.

Keywords Motion Analysis · Articulated Human Motion · Articulated Tracking · Prediction · Inverse Kinematics · Particle Filtering

S. Hauberg
Dept. of Computer Science, University of Copenhagen
E-mail: hauberg@diku.dk

K.S. Pedersen
Dept. of Computer Science, University of Copenhagen
E-mail: kimstp@diku.dk

1 Introduction

Three dimensional articulated human motion analysis is the process of estimating the configuration of body parts over time from sensor input (Poppe, 2007). One approach to this estimation is to use motion capture equipment where e.g. electromagnetic markers are attached to the body and then tracked in three dimensions. While this approach gives accurate results, it is intrusive and cannot be used outside laboratory settings. Alternatively, computer vision systems can be used for non-intrusive analysis. These systems usually perform some sort of optimisation for finding the best configuration of body parts. Such optimisation is often guided by a system for predicting future motion. This paper concerns a framework for building such predictive systems. Unlike most previous work, we build the actual predictive models in spatial coordinates, e.g. by studying hand trajectories, instead of working directly in the space of configuration parameters. This approach not only simplifies certain mathematical aspects of the modelling, but also provides a framework that is more in tune with how humans plan, think about and discuss motion.

Our approach is inspired by results from neurology (Morasso, 1981; Abend et al, 1982) that indicates that humans plan their motions in spatial coordinates. Our working hypothesis is that the best possible predictive system is one that mimics the motion plan. That is, we claim that predictions of future motion should be phrased in the same terms as the motion plan, i.e. in spatial coordinates. This is in contrast to most ongoing research in the vision community where predictions are performed in terms of the pose representation, e.g. the joint configuration of the kinematic skeleton.

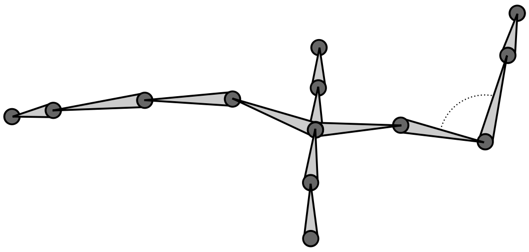


Fig. 1 A rendering of the kinematic skeleton. Each bone is computed as a rotation and a translation relative to its parent. Any subset of the circles, are collectively referred to as the *end-effectors*.

For long, researchers in computer animation have arrived at similar conclusions (Kerlow, 2003; Erleben et al, 2005). It is quite difficult to pose a figure in terms of its internal representation. For most work, animators instead pose individual bones of the figure in spatial coordinates using an *inverse kinematics* system. Such a system usually seeks a configuration of the joints that minimises the distance between a goal and an attained spatial coordinate by solving a nonlinear least-squares problem. In this paper, we recast the least-squares optimisation problem in a probabilistic setting. This then allows us to extend the model to sequential data, which in turn allows us to work with motion models in spatial coordinates rather than joint angles.

2 The Pose Representation

Before discussing the issues of human motion analysis, we pause to introduce the actual representation of the human pose. In this paper, we use the *kinematic skeleton* (see fig. 1), which, amongst others, was also used by Sidenbladh et al (2000) and Sminchisescu and Triggs (2003). The representation is a collection of connected rigid bones organised in a tree structure. Each bone can be rotated at the point of connection between the bone and its parent. We will refer to such a point of connection as a *joint*.

Since bone lengths tend to change very slowly in humans (e.g. at the time scale of biological growth), these are modelled as being constant and effectively we consider bones as being rigid. Hence, the direction of each bone constitute the only degrees of freedom in the kinematic skeleton. The direction in each joint can be parametrised with a vector of angles, noticing that different joints may have different number of degrees of freedom. We may collect all joint angle vectors into one large vector $\boldsymbol{\theta}$ representing all joint angles in the model. Since each element in this vector is an angle, $\boldsymbol{\theta}$ must be confined to the N-dimensional torus, $\mathbb{T}^N$.

2.1 Forward Kinematics

From known bone lengths and a joint angle vector $\boldsymbol{\theta}$ it is straight-forward to compute the spatial coordinates of the bones. Specifically, the purpose is to compute the spatial coordinates of the end points of each bone. This process is started at the root of the tree structure and moves recursively along the branches, which are known as the *kinematic chains*.

The root of the tree is placed at the origin of the coordinate system. The end point of the next bone along a kinematic chain is then computed by rotating the coordinate system and then translating the root along a fixed axis, i.e.

$$\mathbf{a}_l = \mathbf{R}_l \left(\mathbf{a}_{l-1} + \mathbf{t}_l \right) \quad , \tag{1}$$

where $\mathbf{a}_l$ is the l^{th} end point, and $\mathbf{R}_l$ and $\mathbf{t}_l$ denotes a rotation and a translation respectively. The rotation is parametrised by the relevant components of the pose vector $\boldsymbol{\theta}$ and the length of the translation corresponds to the known length of the bone. We can repeat this process recursively until the entire kinematic tree has been traversed. This process is known as *Forward Kinematics* (Erleben et al, 2005).

The rotation matrix $\mathbf{R}_l$ of the l^{th} bone is parametrised by parts of $\boldsymbol{\theta}$. The actual number of used parameters depends on the specific joint. For elbow, knee and angle joints, we use one parameter, while we use three parameters to control all other joints. These two different joint types are respectively known as *hinge joints* and *ball joints*.

Using forward kinematics, we can compute the spatial coordinates of the end points of the individual bones. These are collectively referred to as *end-effectors*. In fig. 1 these are drawn as circles. In most situations, we shall only be concerned with some subset of the end-effectors. Often one is only concerned with body extremities, such as the head and the hands, hence the name *end*-effectors. We will denote the spatial coordinates of these selected end-effectors by $F(\boldsymbol{\theta})$.

2.2 Joint Constraints

In the human body, bones cannot move freely at the joints. A simple example is the elbow joint, which can approximately only bend between 0 and 160 degrees. To represent this, $\boldsymbol{\theta}$ is confined to a subset $\boldsymbol{\Theta}$ of $\mathbb{T}^N$. For simplicity, this subset is often defined by confining each component of $\boldsymbol{\theta}$ to an interval, i.e.

$$\boldsymbol{\Theta} = \prod_{n=1}^{N} [l_n, u_n] \quad , \tag{2}$$

where l_n and u_n denote the lower and upper bounds of the n^{th} component. This type of constraints on the angles is often called *box constraints* (Erleben et al, 2005). More realistic joint constraints are also possible, e.g. the implicit surface models of Herda et al (2004).

3 Challenges of Motion Analysis

Much work has gone into human motion analysis. The bulk of the work is in *non-articulated* analysis, i.e. locating the position of moving humans in image sequences and classifying their actions. It is, however, beyond the scope of this paper to give a review of this work. The interested reader can consult review papers such as the one by Moeslund et al (2006).

In recent years, focus has shifted to *articulated* visual human motion analysis in three dimensions (Poppe, 2007). Here, the objective is to estimate $\boldsymbol{\theta}$ in each image in a sequence. When only using a single camera, or a narrow baseline stereo camera, motion analysis is inherently difficult due to self-occlusions and visual ambiguities. This manifests itself in that the distribution of the human pose is multi-modal with an unknown number of modes. To reliably estimate this distribution we need methods that cope well with multi-modal distributions. Currently, the best method for such problems is the particle filter (Cappé et al, 2007), which represents the distribution as a set of weighted samples. Unfortunately, the particle filter is smitten by the curse of dimensionality in that the necessary number of samples grow exponentially with the dimensionality of state space. The consequence is that the particle filter is only applicable to low dimensional state spaces. This is in direct conflict with the fact that the human body has a great number of degrees of freedom.

The most obvious solution to these problems is to introduce some activity dependent model of the motion, such that the effective degrees of freedom is lowered. Here it should be noted that the actual number of degrees of freedom in the human body (independently of representation) is inherently large. So, while this approach works it does force us into building models that does not generalise to other types of motion than the ones modelled.

3.1 Dimensionality Reduction in Motion Analysis

Motion specific models can be constructed by reducing the dimensionality of the angle space by learning a manifold in angle space to which the motion is restricted. A predictive motion model can then be learned on this manifold. Sidenbladh et al (2000) learned a low-dimensional linear subspace using Principal Component Analysis and used a linear motion model in this subspace. In the paper a model of *walking* is learned, which is a periodic motion, and will therefore be performed in a nonlinear cyclic subspace of the angle space. The choice of a linear subspace therefore seems to come from sheer practicality in order to cope with the high dimensionality of the angle space and not from a well-founded modelling perspective.

Sminchisescu and Jepson (2004) use Laplacian Eigenmaps (Belkin and Niyogi, 2003) to learn a nonlinear motion manifold. Similarly, Lu et al (2008) use a Laplacian Eigenmaps Latent Variable Model (Carreira-Perpinan and Lu, 2007) to learn a manifold. The methods used for learning the manifolds, however, assumes that data is densely sampled on the manifold. This either requires vast amounts of data or a low-dimensional manifold. In the mentioned papers, low dimensional manifolds are studied. Specifically, one-dimensional manifolds corresponding to *walking*. It remains to be seen if the approach scales to higher dimensional manifolds.

Instead of learning the manifold, Elgammal and Lee (2009) suggested learning a mapping from angle space to a *known* manifold. They choose to learn a mapping onto a two dimensional torus, which allows for analysis of both periodic and aperiodic motion. By enforcing a known topology, the learning problem becomes more tractable compared to unsupervised methods. The approach is, however, only applicable when a known topology is available.

Instead of just learning a manifold and restricting the tracking to this, it seems reasonable also to use the density of the training data on this manifold. Urtasun et al (2005) suggested to learn a prior distribution in a low dimensional latent space using a *Scaled Gaussian Process Latent Variable Model* (Grochow et al, 2004). This not only restricts the tracking to a low dimensional latent space, but also makes parts of this space more likely than others. The approach, however, ignores all temporal aspects of the training data. To remedy this, both Urtasun et al (2006) and Wang et al (2008) suggested learning a low dimensional latent space *and* a temporal model at once using a *Gaussian Process Dynamical Model.* This approach seems to provide smooth priors that are both suitable for animation and tracking.

This approach, however, gracefully ignores the topology of the angle space. Specifically, the approach treats the angle space as Euclidean, and thus ignores both the periodic nature of angles and the constraints on the joints. To deal with this issue, Urtasun et al (2008) suggested changing the inner product of the before-

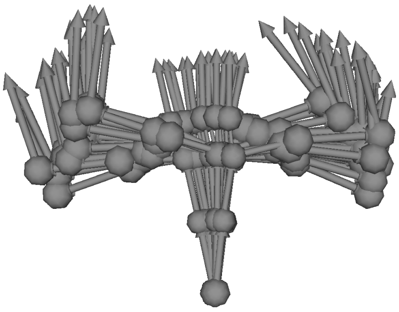


Fig. 2 One hundred random poses generated by sampling joint angles independently from a Von Mises distribution with concentration $\kappa = 500$, corresponding to a circular variance of approximately 0.001 (for details of this sampling, see sec. 7.1.1). Notice how the variance of spatial limb positions increase as the kinematic chains are traversed.

mentioned Gaussian process to incorporate joint constraints.

3.2 The Case Against Predictions in Angle Space

When representing a pose as a set of joint angles $\boldsymbol{\theta}$, it is tempting to build motion models $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ in terms of joint angles. This approach is, however, not without problems.

The first problem is simply that the space of angles is quite high dimensional. This does not rule out manual construction of models, but it can make model learning impractical.

The second problem is the relationship between spatial limb positions and limb orientations implied by forward kinematics. The spatial position of a limb is dependent on the position of its parent. Thus, if we change the direction of one bone, we change the position of all its children. Now, consider a predictive model in angle space that simply adds a little uncertainty to each angle. For this simple model, we see that the variance of the spatial position of limbs increases as we traverse the kinematic chains (see fig. 2). In other words: the position of a hand is always more uncertain than the position of the elbow. This property does not seem to come from a well-founded modelling perspective.

The third problem is that the angle space is topologically different from $\mathbb{R}^N$. If the joint angles are unconstrained, such that each angle can take on values in the circular domain $[0, 2\pi)$, they live on the N-dimensional torus. Thus, it is necessary to restrict the motion models to this manifold, which rules out models that are designed for $\mathbb{R}^N$.

If the joint angles are instead constrained, the topology of the angle space changes significantly. If e.g. box constraints are enforced the set of legal angles becomes a box in $\mathbb{R}^N$. Specifically, it becomes the product space $\prod_{n=1}^{N}[l_n, u_n]$, where l_n and u_n are the lower and upper constraints for the n^{th} angle. Again, we cannot apply motion models designed for $\mathbb{R}^N$, without taking special care to ensure that the pose adheres to the constraints.

In either case, we cannot use motion models designed for $\mathbb{R}^N$. This means that as long as we model in terms of joint angles, it is not mathematically well-defined to learn motion models using e.g. PCA. This problem can be alleviated by picking a suitable inner product for the angle space as suggested by Urtasun et al (2008). While a carefully designed inner product can solve the mathematical problems, it does not solve the rest of the above-mentioned problems.

From a modelling point of view, these problems all lead to the same fundamental question: *which space is most suitable for predicting human motion?*

3.3 Experimental Motivation

For many years, experimental neurologists have studied how people move. Morasso (1981) measured joint angles and hand positions while people moved their hand to a given target. Abend et al (1982) expanded on this experiment, and introduced obstacles that the hand had to move around. Both made the same observation: joint angles show great variation between both persons and trials, while hand positions consistently followed the same path with low variation. Recently, Ganesh (2009) made similar observations for actions like *punching* and *grasping* in a computer vision based tele-immersion system. One interpretation of these results is that the end-effector trajectories describes the *generic*, i.e. the person independent, part of the motion.

This result is a strong indication that humans plan body motion in terms of spatial limb trajectories rather than joint angles. It is our claim that we can achieve better predictions of human motion if we do so in the same space as where the motion was planned. Hence, it makes sense to build predictive models in end-effector space rather than the space of joint angles.

One can arrive at the same conclusion by taking a purely statistical point of view. The above-mentioned experiments showed less variation in end-effector positions compared to joint angles. Thus, it seems more robust to build or learn models in terms of end-effectors as this describes the generic part of the motion.

3.4 Our Approach

Inspired by the results from neurology, we turn to modelling human motion in terms of a few selected end-effectors. Which end-effectors we choose to model depends on the studied motion. Our first assumption is that we know some stochastic process that describes the motion of these end-effectors. Our goal is then to provide a framework for moving this process back into the angle space. From a practical point of view, we aim at describing the pose distribution in angle space given the end-effector positions. This problem is closely related to inverse kinematics, which seeks a joint configuration that attains the given end-effector positions.

This approach has several advantages.

1. Since we are only modelling few selected end-effectors their spatial coordinates are more low-dimensional than all angles. While the degrees of freedom in the human body remains unchanged, the modelling space becomes more low-dimensional. This makes manual model crafting more practical, but more importantly it also makes model *learning* much more robust as fewer parameters need to be estimated.
2. Many types of motion can be easily described in spatial coordinates, e.g. *move foot towards ball* is a description of kicking a ball, whereas the same motion is hard to describe in terms of joint angles. These types of motions are typically *goal oriented*. In computer animation, this line of thinking is so common, that inverse kinematics systems are integrated with practically all 3D graphics software packages.
3. The stochastic motion process is expressed in spatial coordinates, which is a real vector space, instead of angles, which is on the N-dimensional torus. This makes it easier to use *off-the-shelf* stochastic processes as the motion model, since most processes are designed for real vector spaces.

3.5 Inverse Kinematics in Tracking

The most basic fact of computer vision is that the world is inherently spatial and that we study projections of this spatial world onto the image plane. Articulated motion can be estimated as a direct optimisation in the image plane, which requires the derivative of the likelihood with respect to the pose parameters. As the image data is inherently spatial this gradient depends on a mapping from the spatial domain into the joint angle space, i.e. an inverse kinematics system. Bregler et al (2004) formalises this in terms of *twists* and *products of exponential maps* as is common in the robotics literature (Murray et al, 1994). This leads to an iterative scheme for optimising a likelihood based on the grey value constancy assumption. This assumption works well on some sequences, but fails to cope with changes in the lighting conditions. Knossow et al (2008) avoids this issue by defining the likelihood in terms of the chamfer-distance between modelled contours and observed image contours. To perform direct optimisation Knossow et al finds the derivative of the projection of the contour generator into the image plane with respect to the pose parameters – again, this requires solving an inverse kinematics problem. These direct methods works well in many scenarios, but does not allow for an immediate inclusion of a prior motion model.

In this paper, we focus on statistical models of human motion expressed in the spatial domain. This idea of modelling human motion spatially is not new. Recently, Salzmann and Urtasun (2010) showed how joint positions can be modelled, while still adhering to the constraints given by the constant limb lengths. In a similar spirit, Hauberg et al (2010) has proposed that the constraint problem can be solved by projection onto the nonlinear manifold implicitly defined by enforcing constant limb lengths. These ideas has also been explored successfully in the motion compression literature, where Tournier et al (2009) showed how spatial limb positions could be compressed using principal geodesic analysis (Fletcher et al, 2004). These approaches are different from ours as we model goal positions of a few selected end-effectors rather than considering all joints in the kinematic skeleton.

When motion is estimated from silhouettes seen from a single view-point inherent visual ambiguities creates many local minima (Sminchisescu and Triggs, 2003). Using a simple closed-form inverse kinematics system allows Sminchisescu and Triggs to enumerate possible interpretations of the input. This results in a more efficient sampling scheme for their particle filter, as it simultaneously explores many local minima, which reduces the chance of getting stuck in a single local minimum. However, their approach suffers from the problems discussed in sec. 3.2, which makes it difficult to incorporate application specific motion priors.

When modelling interactions with the environment, inverse kinematics is an essential tool as it provides a mapping from the spatial world coordinate system to the joint angle space. Rosenhahn et al (2008) uses this to model interaction with sports equipment, such as bicycles and snowboards. They use the previously mentioned formalism of twists and products of exponential maps to derive an approximate gradient descent scheme for estimating poses. In a similar spirit, Kjellström et al (2010) models interaction with a stick-like object. They solve the inverse kinematics problem using rejection sampling in joint angle space, leading to a computational expensive approach due to the high dimensionality of the joint angle space.

Previously, we (Hauberg et al, 2009; Engell-Nørregård et al, 2009) successfully used end-effector positions as the pose representation, which provided a substantial dimensionality reduction. When comparing a pose to an image, we used inverse kinematics for computing the entire pose configuration. This paper provides the full Bayesian development, analysis and interpretation of this approach.

Courty and Arnaud (2008) have previously suggested a probabilistic interpretation of the inverse kinematics problem. We share the idea of providing a probabilistic model, but where they solve the inverse kinematics problem using importance sampling, we use inverse kinematics to define the importance distribution in our particle filter.

3.6 Organisation of the Paper

The rest of the paper is organised as follows. In the next section, we derive a model, that allows us to describe the distribution of the joint angles in the kinematic skeleton given a set of end-effector goals. In sec. 5 we show the needed steps for performing inference in this model as part of an articulated motion analysis system. These two sections constitute the main technical contribution of the paper. To actually implement a system for articulated motion analysis, we need to deal with observational data. A simple system for this is described in sec. 6 and in sec. 7 we show examples of the attained results. The paper is concluded with a discussion in sec. 8.

4 Deriving the Model

The main objective of this paper is to construct a framework for making predictions of future motion. Since we are using the kinematic skeleton as the pose representation, we need to model some distribution of $\boldsymbol{\theta}$. As previously described we wish to build this distribution in terms of the spatial coordinates of selected end-effectors. To formalise this idea, we let $\mathbf{g}$ denote the spatial coordinates of the *end-effector goals*. Intuitively, this can be thought of as the position where the human wishes to place e.g. his or her hands. Our objective thus becomes to construct the distribution of $\boldsymbol{\theta}$ given the end-effector goal $\mathbf{g}$. Formally, we need to specify $p(\boldsymbol{\theta}|\mathbf{g})$.

Given joint angles, we start out by defining the likelihood of an end-effector goal $\mathbf{g}$ as a “noisy” extension of forward kinematics. Specifically, we define

$$p(\mathbf{g}|\boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{g}|F(\boldsymbol{\theta}), \mathbf{W}^{-1}\right) \ , \tag{3}$$

where $\mathbf{W}$ is the precision (inverse covariance) matrix of the distribution. Here, the stochasticity represents that one does not always reach ones goals.

We do not have any good prior information about the joint angles $\boldsymbol{\theta}$ except some limits on their values. So, we take a least-commitment approach and model all legal angles as equally probable,

$$p(\boldsymbol{\theta}) = \mathcal{U}_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \ , \tag{4}$$

where $\boldsymbol{\Theta}$ is the set of legal angles, and $\mathcal{U}_{\boldsymbol{\Theta}}$ is the uniform distribution on this set. In practice, we use box constraints, i.e. confine each component of $\boldsymbol{\theta}$ to an interval. This gives us

$$p(\boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{U}_{[l_n,u_n]}(\boldsymbol{\theta}[n]) \ , \tag{5}$$

where l_n and u_n denote the lower and upper bounds of the n^{th} component $\boldsymbol{\theta}[n]$.

Using Bayes’ theorem, we combine eq. 3 and eq. 5 into

$$p(\boldsymbol{\theta}|\mathbf{g}) = \frac{p(\mathbf{g}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{g})} \propto p(\mathbf{g}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{6}$$

$$= \mathcal{N}\left(\mathbf{g}|F(\boldsymbol{\theta}), \mathbf{W}^{-1}\right) \prod_{n=1}^{N} \mathcal{U}_{[l_n,u_n]}(\boldsymbol{\theta}[n]) \ . \tag{7}$$

As can be seen, we perform a nonlinear transformation F of the $\boldsymbol{\theta}$ variable and define its distribution in the resulting space. In other words we effectively define the distribution of $\boldsymbol{\theta}$ in the end-effector goal space rather than angle space. It should be stressed that while $p(\boldsymbol{\theta}|\mathbf{g})$ looks similar to a normal distribution, it is not, due to the nonlinear transformation F.

In the end, the goal is to be able to extract $\boldsymbol{\theta}$ from observational data such as images. We do this using a straight-forward generative model,

$$p(\mathbf{X}, \boldsymbol{\theta}, \mathbf{g}) = p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{g})p(\mathbf{g}) \ , \tag{8}$$

where $\mathbf{X}$ is the observation. This model is shown graphically in fig. 3. It should be noted that we have yet to specify $p(\mathbf{g})$; we will remedy this at a later stage in the paper.

4.1 Relation to Inverse Kinematics

Taking the logarithm of eq. 7 gives us

$$\begin{aligned} \log p(\boldsymbol{\theta}|\mathbf{g}) = &-\frac{1}{2}(\mathbf{g} - F(\boldsymbol{\theta}))^T \mathbf{W}(\mathbf{g} - F(\boldsymbol{\theta})) \\ &+ \sum_{n=1}^{N} \log \mathcal{U}_{[l_n,u_n]}(\boldsymbol{\theta}[n]) + \text{constant} \ . \end{aligned} \tag{9}$$

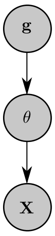


Fig. 3 Graphical representation of the *Probabilistic Inverse Kinematics* model given in eq. 8.

Maximising this corresponds to minimising

$$d^2\left(F(\boldsymbol{\theta}),\mathbf{g}\right)=\frac{1}{2}(\mathbf{g}-F(\boldsymbol{\theta}))^T\mathbf{W}(\mathbf{g}-F(\boldsymbol{\theta}))\ \ , \tag{10}$$

subject to $\mathbf{l}\leq\boldsymbol{\theta}\leq\mathbf{u}$, where $\mathbf{l}$ and $\mathbf{u}$ are the vectors containing the joint limits. This is the inverse kinematics model presented by Zhao and Badler (1994). Thus, we deem eq. 7 *Probabilistic Inverse Kinematics (PIK).*

It should be noted that due to the nonlinearity of F, this optimisation problem is nonlinear, rendering maximum a posteriori estimation difficult. Since the end-effector space is often much more low-dimensional than the angle space, the Hessian of eq. 10 does not have full rank. As a consequence the optimisation problem is not guaranteed to have a unique solution. In fact, the minima of eq. 10 often form a continuous subspace of $\boldsymbol{\Theta}$. This can be realised simply by fixing the position of a hand, and moving the rest of the body freely. Such a sequence of poses will be continuous while all poses attain the same end-effector position.

4.2 Sequential PIK

As previously mentioned we aim at building predictive distributions for sequential analysis based on the end-effector goals. To keep the model as general as possible, we assume knowledge of some stochastic process controlling the end-effector goals. That is, we assume we know $p(\mathbf{g}_t|\mathbf{g}_{1:t-1})$, where the subscript denotes time and $\mathbf{g}_{1:t-1}=\{\mathbf{g}_1,\ldots,\mathbf{g}_{t-1}\}$ is the past sequence. In sec. 7 we will show examples of such processes, but it should be noted that any continuous process designed for *real* vector spaces is applicable.

While we accept any continuous model $p(\mathbf{g}_t|\mathbf{g}_{1:t-1})$ in end-effector goal space, we do prefer smooth motion in angular space. We model this as preferring small temporal gradients $\left|\left|\frac{\partial\boldsymbol{\theta}}{\partial t}\right|\right|^2$. To avoid extending the state with the temporal gradient, we approximate it using finite differences,

$$\left|\left|\frac{\partial\boldsymbol{\theta}}{\partial t}\right|\right|^2\approx||\boldsymbol{\theta}_t-\boldsymbol{\theta}_{t-1}||^2\ \ . \tag{11}$$

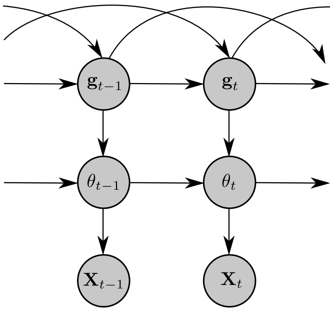


Fig. 4 Graphical representation of the *Sequential Probabilistic Inverse Kinematics* model given in eq. 13.

So, we introduce a first order Markov model in angle space and define

$$\begin{aligned}\log p(\boldsymbol{\theta}_t|\mathbf{g}_t,\boldsymbol{\theta}_{t-1})=&-\frac{1}{2}(\mathbf{g}_t-F(\boldsymbol{\theta}_t))^T\mathbf{W}(\mathbf{g}_t-F(\boldsymbol{\theta}_t))\\&-\frac{\lambda}{2}||\boldsymbol{\theta}_t-\boldsymbol{\theta}_{t-1}||^2\\&+\sum_{n=1}^{N}\log\mathcal{U}_{[l_n,u_n]}(\boldsymbol{\theta}_t[n])+\text{constant}\ \ ,\end{aligned}\tag{12}$$

where λ controls the degree of temporal smoothness. This is effectively the same as eq. 9 except poses close to the previous one, $\boldsymbol{\theta}_{t-1}$, are preferred. This slight change has the pleasant effect of isolating the modes of $p(\boldsymbol{\theta}_t|\mathbf{g}_t,\boldsymbol{\theta}_{t-1})$, which makes maximum a posteriori estimation more tractable. Consider the previously given example. A person can put his or her hands in a given position in many ways, but no continuous subset of the pose space can attain the given hand position and be closest to the previous pose at the same time.

In summary, we have the following final generative model

$$\begin{aligned}p(\mathbf{X}_{1:T},&\boldsymbol{\theta}_{1:T},\mathbf{g}_{1:T})=p(\mathbf{X}_1|\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1|\mathbf{g}_1)p(\mathbf{g}_1)\\&\prod_{t=2}^{T}p(\mathbf{X}_t|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t|\mathbf{g}_t,\theta_{t-1})p(\mathbf{g}_t|\mathbf{g}_{1:t-1})\ \ ,\end{aligned}\tag{13}$$

where $\mathbf{X}_{1:T}=\{\mathbf{X}_1,\ldots,\mathbf{X}_T\}$ denotes the sequence of observations. This model is illustrated in fig. 4.

4.3 Designing Spatial Processes

One of the advantages of the Sequential PIK model is that given a spatial model $p(\mathbf{g}_t|\mathbf{g}_{1:t-1})$ of the end-effector goals, we can work with the corresponding model in terms of joint angles. However, so far, we have avoided the question of how to define such spatial models. We will give specific examples in sec. 7, but until then, it can be instructive to consider how one might design such models.

Consider a person grasping for an object. Neurologists (Morasso, 1981; Abend et al, 1982) have shown that people most often move their hands on the straight line from the current hand position towards the object. This can easily be modelled by letting the goal position of the hand move along this line. The speed of the hand could be modelled as a constant or it could be controlled by another process. It should be stressed that the immediate goal $\mathbf{g}_t$ follows the mentioned line and hence is different from the end goal (the position of the object).

As a second example, consider a person walking. Many successful models of such motions have been derived in terms of joint angles as discussed in sec. 3.1. Most of them does, however, not consider interaction with the ground plane, which is an intrinsic part of the motion. This interaction actually makes the motion non-smooth, something most models cannot handle. By modelling the goals of the feet it is easy to ensure that one foot always is in contact with the ground plane and that the other never penetrates the ground. The actual trajectory of the moving foot could then be learned using, e.g. a Gaussian process (Rasmussen and Williams, 2006). Again, the immediate goal $\mathbf{g}_t$ would move along the curve represented by the learned process.

5 Inference

In the previous section a model of human motion was derived. This section focuses on the approximations needed to perform inference in the model. Due to the inherent multi-modality of the problem, we use a *particle filter* to perform the inference. In the next section this algorithm is described from an *importance sampling* point of view as our choice of *importance distribution* is non-conventional. Results will, however, not be proved; instead the reader is referred to the review paper by Cappé et al (2007).

We will assume that a continuous motion model $p(\mathbf{g}_t|\mathbf{g}_{1:t-1})$ is available and that we can sample from this distribution. Specific choices of this model will be made in sec. 7, but it should be stressed that the method is independent of this choice, e.g. further constraints such as smoothness may be added.

We will also assume that a system $p(\mathbf{X}_t|\boldsymbol{\theta}_t)$ for making visual measurements is available. Such a system will be described in sec. 6, but the method is also independent of the specifics of this system.

5.1 Approximate Bayesian Filtering

The aim of approximate Bayesian filtering is to estimate the filtering distribution $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \mathbf{X}_{1:t})$ by means of samples. Instead of drawing samples from this distribution, they are taken from $p(\boldsymbol{\theta}_{1:t}|\mathbf{g}_{1:t}, \mathbf{X}_{1:t})$ and then all values of the samples but $\boldsymbol{\theta}_t$ are ignored. Since the filtering distribution is unknown, we turn to importance sampling (Bishop, 2006). This means drawing M samples $\boldsymbol{\theta}_{1:t}^{(m)}$ from an *importance distribution* $q(\boldsymbol{\theta}_{1:t}|\mathbf{g}_{1:t}, \mathbf{X}_{1:t})$ after which moments of the filtering distribution can be estimated as

$$\begin{aligned}\bar{h} &= \int h(\boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t|\mathbf{g}_t, \mathbf{X}_{1:t}) \mathrm{d}\boldsymbol{\theta}_t \\ &\approx \sum_{m=1}^{M} w_t^{(m)} h\left(\boldsymbol{\theta}_t^{(m)}\right)\end{aligned} \tag{14}$$

for any function h. Here we have defined the *importance weights* as

$$w_t^{(m)} \propto \frac{p\left(\boldsymbol{\theta}_{1:t}^{(m)}|\mathbf{g}_{1:t}^{(m)}, \mathbf{X}_{1:t}\right)}{q\left(\boldsymbol{\theta}_{1:t}^{(m)}|\mathbf{g}_{1:t}^{(m)}, \mathbf{X}_{1:t}\right)} \quad , \quad \sum_{m=1}^{M} w_t^{(m)} = 1 \ . \tag{15}$$

Unsurprisingly, eq. 14 is exact when $M \to \infty$.

The key to making this strategy work is to choose an importance distribution, which ensures that the resulting algorithm is recursive. With this in mind, it is chosen that the importance distribution should be factorised as

$$\begin{aligned}q(\boldsymbol{\theta}_{1:t}|\mathbf{g}_{1:t}, \mathbf{X}_{1:t}) = \ & q(\boldsymbol{\theta}_{1:t-1}|\mathbf{g}_{1:t-1}, \mathbf{X}_{1:t-1}) \\ & q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t) \ .\end{aligned} \tag{16}$$

With this choice, one can sample from $q(\boldsymbol{\theta}_{1:t}|\mathbf{g}_{1:t}, \mathbf{X}_{1:t})$ recursively by extending the previous sample $\boldsymbol{\theta}_{1:t-1}^{(m)}$ with a new sample $\boldsymbol{\theta}_t^{(m)}$ from $q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t)$. The weights $w_{t-1}^{(m)}$ can also be recursively updated by means of

$$w_t^{(m)} \propto w_{t-1}^{(m)} \ p\left(\mathbf{X}_t|\boldsymbol{\theta}_t^{(m)}\right) r^{(m)} \tag{17}$$

with

$$r^{(m)} = \frac{p\left(\boldsymbol{\theta}_t^{(m)}|\mathbf{g}_t^{(m)}, \boldsymbol{\theta}_{t-1}^{(m)}\right)}{q\left(\boldsymbol{\theta}_t^{(m)}|\mathbf{g}_t^{(m)}, \boldsymbol{\theta}_{t-1}^{(m)}, \mathbf{X}_t\right)} \ . \tag{18}$$

When extending the previous sample, we need to draw a sample from $q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t)$. This, however, assumes that the true value of $\boldsymbol{\theta}_{t-1}$ is known, which is not the case. Several strategies can be used to approximate this value. *Sequential Importance Sampling* assumes that the previous sample positions were the true value, i.e. $\boldsymbol{\theta}_{t-1} = \boldsymbol{\theta}_{t-1}^{(m)}$. This is usually not stable, since small differences between the sample and the true value accumulates over time. The *particle filter* approximates the distribution of $\boldsymbol{\theta}_{t-1}$ with the weighted samples from the previous iteration, i.e.

$$p(\boldsymbol{\theta}_{t-1}|\mathbf{g}_{t-1}, \mathbf{X}_{1:t-1}) \approx \sum_{m=1}^{M} w_{t-1}^{(m)} \delta\left(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-1}^{(m)}\right) . \quad (19)$$

The value of $\boldsymbol{\theta}_{t-1}$ is then approximated by sampling from this distribution. This simply corresponds to a resampling of the previous samples, where samples with large weights have a high probability of surviving. Since these samples are assumed to come from the true distribution $p(\boldsymbol{\theta}_{t-1}|\mathbf{g}_{t-1}, \mathbf{X}_{1:t-1})$, the associated weights have to be reset, i.e. $w_{t-1}^{(m)} = 1/M$ for all m.

We have still to choose the importance distribution $q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t)$. The most common choice is inspired by eq. 18. Here, we note that $r^{(m)} = 1$ if we set $q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t) = p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$, which simplifies the weight update. With this choice, the resulting filter is called the *Bootstrap filter*. This cannot, however, be applied for the model described in this paper, as we cannot sample directly from $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$. A different importance distribution will be presented next.

5.2 The Importance Distribution

Inspired by the Bootstrap filter, we drop the observation $\mathbf{X}_t$ from the importance distribution, such that $q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t) = q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$. It would seem tempting to use $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$ as the importance distribution. It is, however, not straightforward to draw samples from this distribution, so this choice does not seem viable. Instead, we seek a distribution that locally behaves similarly to $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$.

In sec. 4.2 we noted that $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$ has isolated modes. Thus, it seems reasonable to locally approximate this distribution using a Laplace approximation (Bishop, 2006), which is a second order Taylor approximation of the true distribution. This boils down to fitting a normal distribution around the local mode $\boldsymbol{\theta}_t^*$, using the Hessian of $-\log p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$ as an approximation of the precision matrix. Assuming that $F(\boldsymbol{\theta}_t^*) = \mathbf{g}_t$, i.e. the located pose actually reaches the given end-effector goal, this Hessian matrix attains the simple form

$$\mathbf{H} = -(\mathbf{g}_t - F(\boldsymbol{\theta}_t^*))^T \mathbf{W} \frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}} + \mathbf{J}^T \mathbf{W} \mathbf{J} + \lambda \mathbf{I} \quad (20)$$
$$= \mathbf{J}^T \mathbf{W} \mathbf{J} + \lambda \mathbf{I} \ , \quad (21)$$

where $\mathbf{J}$ is the Jacobian of F at $\boldsymbol{\theta}_t^*$. This Jacobian consists of a row for each component of $\boldsymbol{\theta}_t$. Each such row can be computed in a straightforward manner (Zhao and Badler, 1994). If $\mathbf{r}$ is the unit-length rotational axis of the n^{th} angle and $\Delta_{\mathbf{g}}$ is the vector from the joint to the end-effector, then the row is computed as $\frac{\partial F}{\partial \boldsymbol{\theta}_t[n]} = \mathbf{r} \times \Delta_{\mathbf{g}}$. This is merely the tangent of the circle formed by the end-effector when rotating the joint in question as is illustrated in fig. 5.

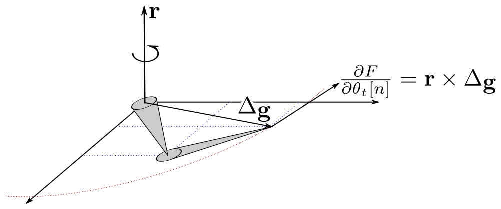

r
$\frac{\partial F}{\partial \theta_t[n]} = \mathbf{r} \times \Delta\mathbf{g}$
$\Delta\mathbf{g}$

Fig. 5 The derivative of a joint is found as the cross product of the rotational axis $\mathbf{r}$ and the vector from the joint to the end-effector $\mathbf{g}$.

We, thus, have an easy way of computing the Laplace approximation of $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$, which we use as the importance distribution. That is, we pick

$$q(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1}, \mathbf{X}_t)$$
$$= \mathcal{N}\left(\boldsymbol{\theta}_t \middle| \boldsymbol{\theta}_t^*, \left(\mathbf{J}^T \mathbf{W} \mathbf{J} + \lambda \mathbf{I}\right)^{-1}\right) \mathcal{U}_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_t) \ . \quad (22)$$

Hence, we are using a local second order approximation of the Bootstrap filter, while adhering to the joint constraints. It should be noted that we can easily sample from this distribution using *rejection sampling* (Bishop, 2006). When using box constraints, this rejection sampling can be performed one joint angle at a time and as such does not impact performance in any measureable way.

5.2.1 Solving the Nonlinear Least-Squares Problem

One assumption made in the previous section was that we can compute the mode $\boldsymbol{\theta}_t^*$ of $p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$. Since the modes of this distribution are isolated, we can easily locate a mode, using a nonlinear constrained least-squares solver. In this paper, we are using a simple, yet effective, gradient projection method with line search (Nocedal and Wright, 1999).

To perform the optimisation, we need to compute the gradient of $\log p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})$. This is given by

$$\frac{\partial \log p(\boldsymbol{\theta}_t|\mathbf{g}_t, \boldsymbol{\theta}_{t-1})}{\partial \boldsymbol{\theta}_t} = (\mathbf{g}_t - F(\boldsymbol{\theta}_t))^T \mathbf{W} \mathbf{J} - \lambda(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \ . \quad (23)$$

When solving the nonlinear least-squares problem, we start the search in $\boldsymbol{\theta}_{t-1}$, which usually ensures convergence in a few iterations.

6 Visual Measurements

In this section we define $p(\mathbf{X}_t|\boldsymbol{\theta}_t)$, i.e. we describe how we compare an observation with a pose hypothesis. This allows us to compute weights for the particle filter, which then optimises the posterior. Since this paper is focused on the prediction aspect of a tracker, we deliberately keep this part as simple as possible.

6.1 General Idea

To keep the necessary image processing to a minimum, we use a small baseline consumer stereo camera from Point Grey[1]. At each time-step, this camera provides a set of three dimensional points as seen from a single view point (see fig. 6). The objective of $p(\mathbf{X}_t|\boldsymbol{\theta}_t)$ is then essentially to measure how well $\boldsymbol{\theta}_t$ fits with these points. Due to the small baseline, visual ambiguities will occur, which leads to local maxima in the likelihood. This is one reason for using a particle filter for performing inference.

Let $\mathbf{X}_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ denote the set of three dimensional points provided by the stereo camera. Our first simplifying assumption is that these are independent and identically distributed, i.e.

$$p(\mathbf{X}_t|\boldsymbol{\theta}_t) = \prod_{k=1}^{K} p(\mathbf{x}_k|\boldsymbol{\theta}_t) \ . \tag{24}$$

We then define the likelihood of an individual point as

$$p(\mathbf{x}_k|\boldsymbol{\theta}_t) \propto \exp\left(-\frac{D^2(\boldsymbol{\theta}_t, \mathbf{x}_k)}{2\sigma^2}\right) \ , \tag{25}$$

where $D^2(\boldsymbol{\theta}_t, \mathbf{x}_k)$ denotes the squared distance between the point $\mathbf{x}_k$ and the surface of the pose parametrised by $\boldsymbol{\theta}_t$.

We, thus, need a definition of the surface of a pose, and a suitable metric.

6.2 The Pose Surface

The pose $\boldsymbol{\theta}_t$ corresponds to a connected set of L bones, each of which have a start and an end point. We, respectively, denote these $\mathbf{a}_l$ and $\mathbf{b}_l$ for the l^{th} bone; we can compute these points using forward kinematics. We then construct a capsule with radius r_l that follows the line segment from $\mathbf{a}_l$ to $\mathbf{b}_l$. The surface of the l^{th} bone is then defined as the part of this capsule that is visible from the current view point. This surface model of a bone is illustrated in fig. 7a. The entire pose surface is then defined as the union of these bone surfaces. This is essentially the same surface model as was suggested by Sidenbladh et al (2000), except they used cylinders instead of capsules. In general, this type of surface models does not describe the human body particularly well. The capsule skin can, however, be replaced with more descriptive skin models, such as the articulated implicit surfaces suggested by Horaud et al (2009).

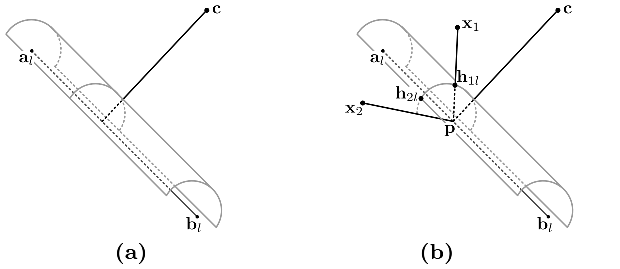


Fig. 7 (a) An illustration of the surface of a bone. Here $\mathbf{a}_l$ and $\mathbf{b}_l$ denotes the end points of the bone, while $\mathbf{c}$ denotes the camera position. For illustration purposes, we only show the cylindric part of the capsules. (b) An illustration of the computation of the point on the bone surface from a data point. To keep the figure simple, we show data points $\mathbf{x}_1$ and $\mathbf{x}_2$ that share the same nearest point $\mathbf{p} = \mathbf{p}_{1l} = \mathbf{p}_{2l}$ on the line segment between $\mathbf{a}_l$ and $\mathbf{b}_l$. The vectors $\mathbf{w}_{1l}$ and $\mathbf{w}_{2l}$ are the vectors from $\mathbf{p}$ pointing towards $\mathbf{h}_{1l}$ and $\mathbf{h}_{2l}$.

In the end, our objective is to compute the distance between a data point and the surface. We do this by first finding the nearest point on the surface and then compute the distance between this point and the data point. Since we define the pose surface bone by bone, we can compute this distance as the distance to the nearest bone surface, i.e.

$$D^2(\boldsymbol{\theta}_t, \mathbf{x}_k) = \min_l \left(d^2(\mathbf{x}_k, \mathbf{h}_{kl})\right) \ , \tag{26}$$

where $\mathbf{h}_{kl}$ is the nearest point (in the Euclidean sense) on the l^{th} bone and $d^2(\mathbf{x}_k, \mathbf{h}_{kl})$ is the squared distance between $\mathbf{x}_k$ and $\mathbf{h}_{kl}$. Note that the minimisation in eq. 26 can be trivially performed by iterating over all L bones.

6.2.1 Finding Nearest Point on the Bone Surface

We thus set out to find the point on a bone surface that is nearest to the data point $\mathbf{x}_k$. We start by finding the nearest point on the capsule with radius r_l around the line segment from $\mathbf{a}_l$ to $\mathbf{b}_l$. We let $\mathbf{p}_{kl}$ denote the point on the line segment that is nearest to $\mathbf{x}_k$, and then the nearest point on the capsule can be found as $\mathbf{p} + r_l \frac{\mathbf{x}_k - \mathbf{p}_{kl}}{||\mathbf{x}_k - \mathbf{p}_{kl}||}$.

We now turn our attention to the part of the capsule that can be seen from the camera. Points on this part of the capsule can be described as the points where the angle between the vectors from $\mathbf{p}_{kl}$ to the camera and

[1] `http://www.ptgrey.com/products/bumblebee2/`

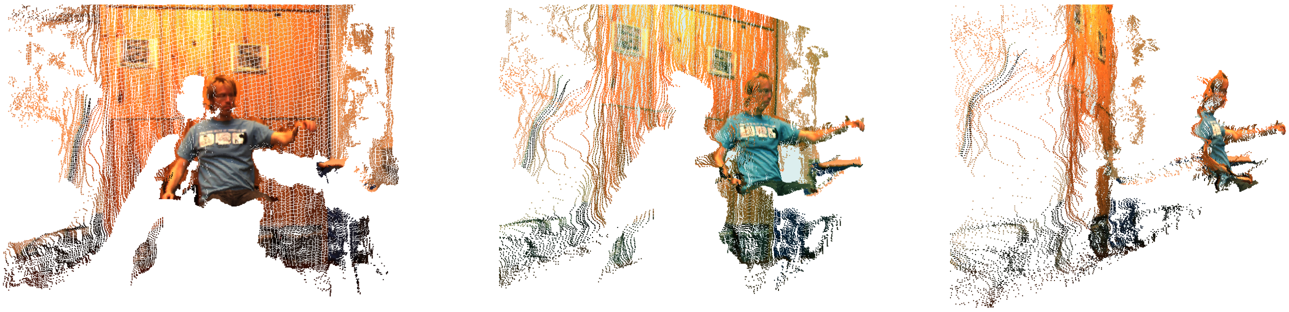


Fig. 6 A rendering of the data from the stereo camera from different views.

to the point on the surface should be no greater than 90 degrees. This is formalised by requiring $(\mathbf{x}_k - \mathbf{p}_{kl})^T(\mathbf{c} - \mathbf{p}_{kl}) \geq 0$, where $\mathbf{c}$ denotes the position of the camera.

If the nearest point is not on the visible part, then it is on the border of the surface. Hence, the vector from $\mathbf{p}_{kl}$ to the nearest point must be orthogonal to the line segment formed by $\mathbf{a}_l$ and $\mathbf{b}_l$, and the vector from $\mathbf{p}_{kl}$ to $\mathbf{c}$. In other words, the nearest point on the surface can then be computed as

$$\mathbf{h}_{kl} = \mathbf{p}_{kl} + r_l \frac{\mathbf{w}_{kl}}{||\mathbf{w}_{kl}||} \quad , \tag{27}$$

where

$$\mathbf{w}_{kl} = \begin{cases} \mathbf{x}_k - \mathbf{p}_{kl}, & (\mathbf{x}_k - \mathbf{p}_{kl})^T(\mathbf{c} - \mathbf{p}_{kl}) \geq 0 \\ \operatorname{sgn}\left(\mathbf{x}_k^T \mathbf{v}\right) \mathbf{v}, & \text{otherwise} \quad , \end{cases} \tag{28}$$

with $\mathbf{v} = (\mathbf{c} - \mathbf{p}_{kl}) \times (\mathbf{b}_l - \mathbf{a}_l)$. The geometry behind these computations is illustrated in fig. 7b.

6.3 Robust Metric

We are now able to find the point on the surface of a bone that is nearest to a data point $\mathbf{x}_k$. Thus, we are only missing a suitable way of computing the squared distance between the data point and the nearest point on the bone surface. The most straight-forward approach is to use the squared Euclidean distance. This is, however, not robust with respect to outliers. Looking at fig. 6, we see that the data from the stereo camera contains many outliers; some due to mismatches and some due to other objects in the scene (i.e. the background).

To cope with these outliers, we could use any robust metric. Here, we choose to use a simple thresholded squared Euclidean distance, i.e.

$$d^2(\mathbf{x}_k, \mathbf{h}_{kl}) = \min\left(||\mathbf{x}_k - \mathbf{h}_{kl}||^2, \tau\right) \quad . \tag{29}$$

7 Results

We now have a complete articulated tracker, and so move on to the evaluation. First, we compare our predictive model to a linear model in angle space. Then, we show how the model can be extended to model interactions between the person and objects in the environment. Finally, we provide an example of an activity dependent model from the world of physiotherapy.

In each frame, the particle filter provides us with a set of weighted hypotheses. To reduce this set to a single hypothesis in each frame, we compute the weighted average, i.e.

$$\hat{\boldsymbol{\theta}}_t = \sum_{m=1}^{M} w_t^{(m)} \boldsymbol{\theta}_t^{(m)} \quad , \tag{30}$$

which we use as an estimate of the current pose. This simple choice seems to work well enough in practice.

7.1 Linear Extrapolation in Different Spaces

We start out by studying an image sequence in which a test subject keeps his legs fixed while waving a stick with both hands in a fairly arbitrary way. This sequence allows us to work with both complex and highly articulated upper body motions, without having to model translations of the entire body. A few frames from this sequence is available in fig. 8. This sequence poses several problems. Due to the style of the motion, limbs are often occluded by each other, e.g. one arm is often occluded by the torso. As the sequence is recorded at approximately 15 frames per second we also see motion blur. This reduces the quality of the stereo reconstruction, which produces ambiguities for the tracker.

7.1.1 Angular Motion Model

For comparative purposes we build an articulated tracker in which the predictive model is phrased in terms of

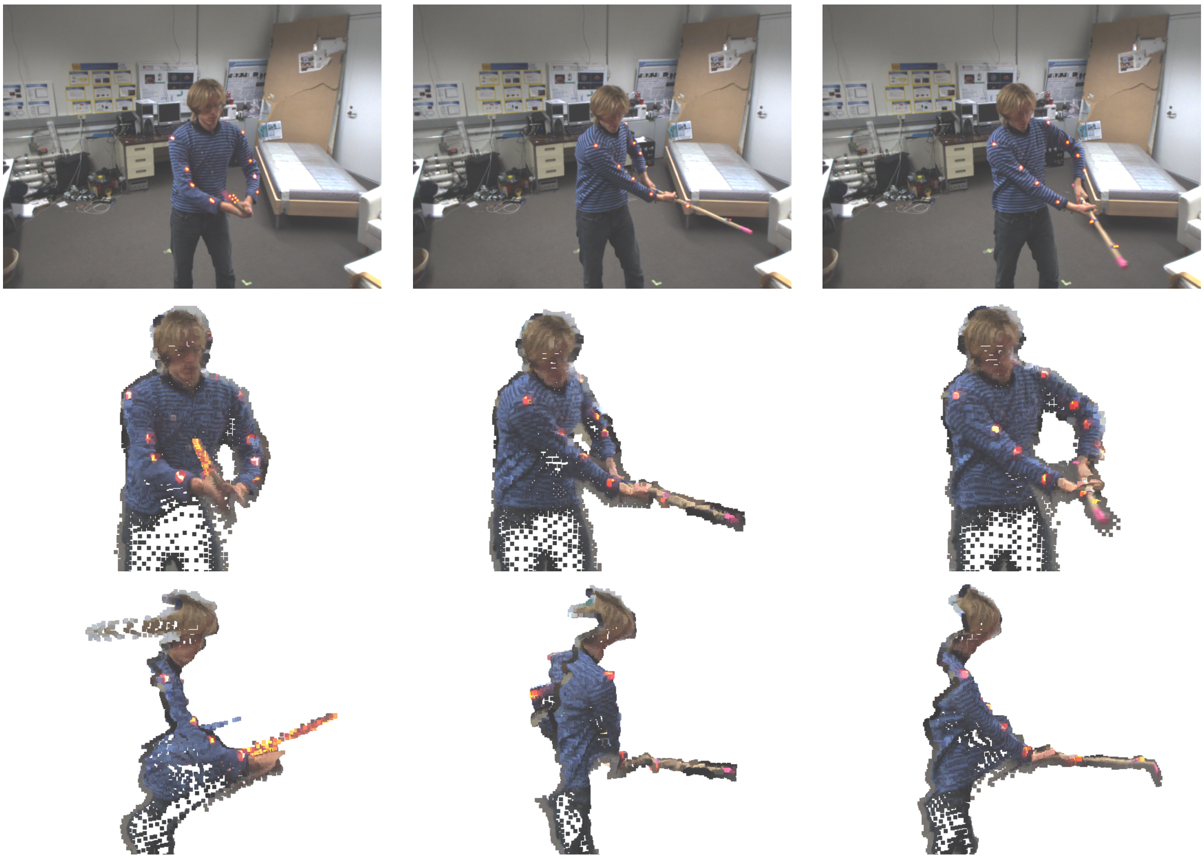


Fig. 8 Frames 91, 159 and 268 from the first test sequence. On the top is one image from the camera; in the middle and in the bottom is a rendering of the stereo data from two different view points. The final objective is to extract $\boldsymbol{\theta}_t$ from such data.

independent joint angles. Specifically, we linear extrapolate the joint angles, i.e. we define the mean of the predictive distribution as

$$\bar{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\theta}_t + (\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \ . \tag{31}$$

The predictive distribution is then defined as a Von Mises distribution (Bishop, 2006) with the above mean, which is constrained to respect the joint limits. Precisely, the predictive distribution is defined as

$$\begin{aligned} &p\big(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}\big) \\ &\quad \propto \mathcal{U}_{\Theta}(\boldsymbol{\theta}_{t+1}) \prod_{n=1}^{N} \exp\left(\kappa_n \cos(\boldsymbol{\theta}_{t+1}[n] - \bar{\boldsymbol{\theta}}_{t+1}[n])\right) \end{aligned} \tag{32}$$

This model is conceptually the one proposed by Poon and Fleet (2002), except our noise model is a Von Mises distribution whereas a Normal distribution was previously applied.

7.1.2 End-effector Motion Model

We compare the linear predictor in angle space to a linear predictor in the space of end-effector goals. Specifically, we focus on the spatial positions of the head and the hands, such that $\mathbf{g}_t$ denotes the goal of these. We then define their motion as

$$p(\mathbf{g}_{t+1}|\mathbf{g}_t, \mathbf{g}_{t-1}) = \mathcal{N}(\mathbf{g}_{t+1}|\mathbf{g}_t + (\mathbf{g}_t - \mathbf{g}_{t-1}), \sigma^2\mathbf{I}). \tag{33}$$

The predictive distribution in angle space is then created as described in sec. 4.

7.1.3 Experimental Setup

To evaluate the quality of the attained results we position motion capture markers on the arms of the test subject. We then measure the average distance between the motion capture markers and the capsule skin of the estimated pose. This measure is then averaged across frames, such that the error measure becomes

$$\mathcal{E}(\boldsymbol{\theta}_{1:T}) = \frac{1}{TM} \sum_{t=1}^{T} \sum_{m=1}^{M} D(\boldsymbol{\theta}_t, \mathbf{v}_{mt}) \ , \tag{34}$$

where $D(\boldsymbol{\theta}_t, \mathbf{v}_{mt})$ is the shortest Euclidean distance between the m^{th} motion capture marker and the skin at time t.

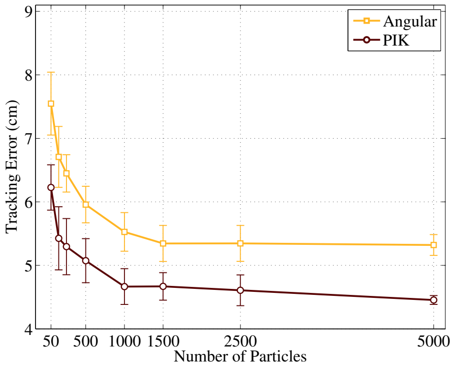


Fig. 13 The error measure $\mathcal{E}(\boldsymbol{\theta}_{1:T})$ plotted as a function of the number of particles. The shown results are averaged over several trials; the error bars correspond to one standard deviation.

If the observation density $p(\boldsymbol{\theta}_t|\mathbf{X}_t)$ is noisy, then the motion model tends to act as a smoothing filter. This can be of particular importance when observations are missing, e.g. during self-occlusions. When evaluating the quality of a motion model it, thus, can be helpful to look at the smoothness of the attained pose sequence. To measure this, we simply compute the average size of the temporal gradient. We approximate this gradient using finite differences, and hence use

$$\mathcal{S}(\boldsymbol{\theta}_{1:T}) = \frac{1}{TL}\sum_{t=1}^{T}\sum_{l=1}^{L}||\mathbf{a}_{lt} - \mathbf{a}_{l,t-1}|| \tag{35}$$

as a measure of smoothness.

7.1.4 Evaluation

To see how the two motion models compare we apply them several times to the same sequence with a variable number of particles. The tracker is manually initialised in the first frame. Visualisations of the attained results for a few frames are available in fig. 9–12. Movies with the same results are also available on-line[2]. Due to the fast motions and poor stereo reconstructions, both models have difficulties tracking the subject in all frames. However, visually, it is evident that the end-effector motion model provides more accurate tracking and more smooth motion trajectories compared to the angular motion model.

In order to quantify these visual results, we compute the error measure presented in eq. 34 for results attained using different number of particles. Fig. 13 shows this. Here the results for each number of particles has been averaged over several trials. It is worth noticing that our model consistently outperforms the model in angle space.

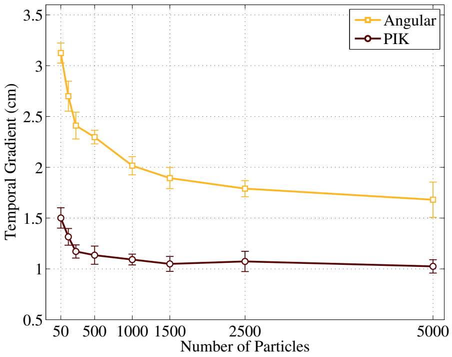


Fig. 14 The smoothness measure $\mathcal{S}(\boldsymbol{\theta}_{1:T})$ plotted as a function of the number of particles. Low values indicate smooth trajectories. The shown results are averaged over several trials; the error bars correspond to one standard deviation.

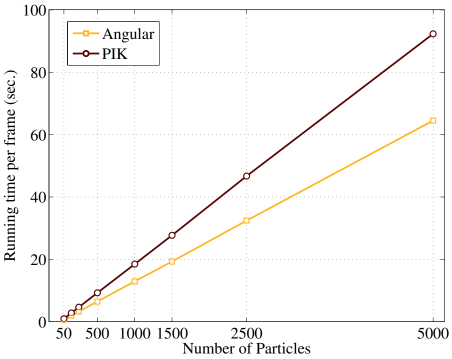
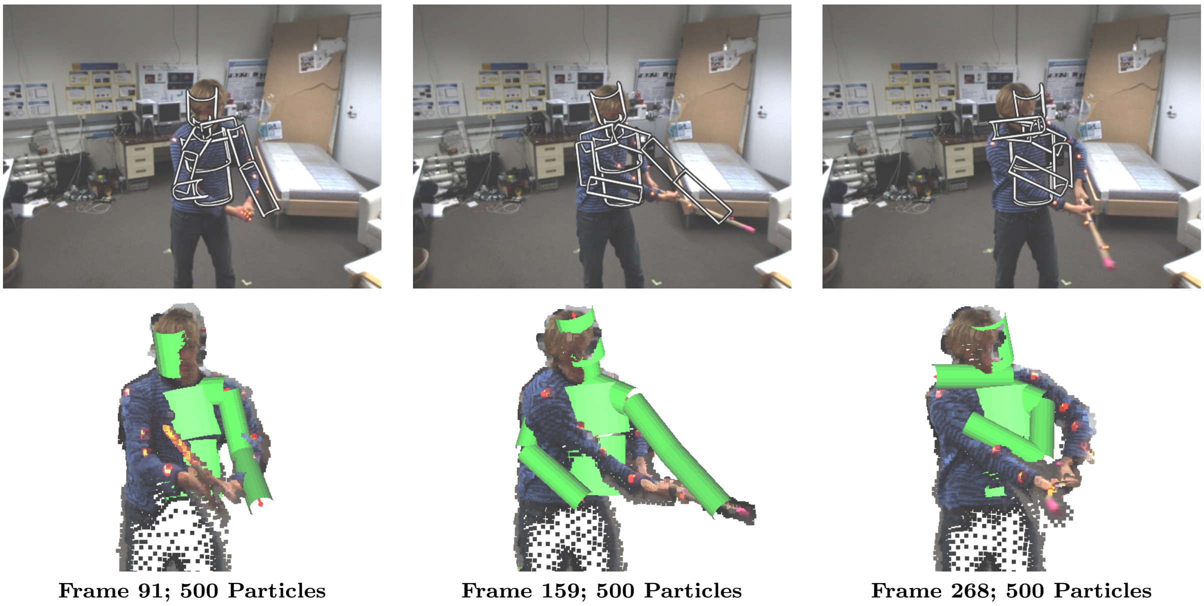


Fig. 15 The running time per frame of the tracking system when using different motion models. The end-effector based model approximately requires 1.4 times longer per frame.

We also measure the smoothness $\mathcal{S}$ of the attained pose sequences as a function of the number of particles. This is plotted in fig. 14. As can be seen, our model is consistently more smooth compared to the linear model in angle space. This result is an indication that our model is less sensitive to noise in the observational data.

In summary, we see that our model allows for improved motion estimation using fewer particles compared to a linear model in angle space. This, however, comes at the cost of a computationally more expensive prediction. One might then ask, when this model improves the efficiency of the entire program. The answer to this question depends on the computation time required by the visual measurement system as this most often is the computationally most demanding part of tracking systems. For our system, the running time per frame is plotted in fig. 15. Comparing this with fig. 13, we see that our model produces superior results for fixed computational resources.

[2] `http://humim.org/pik-tracker/`

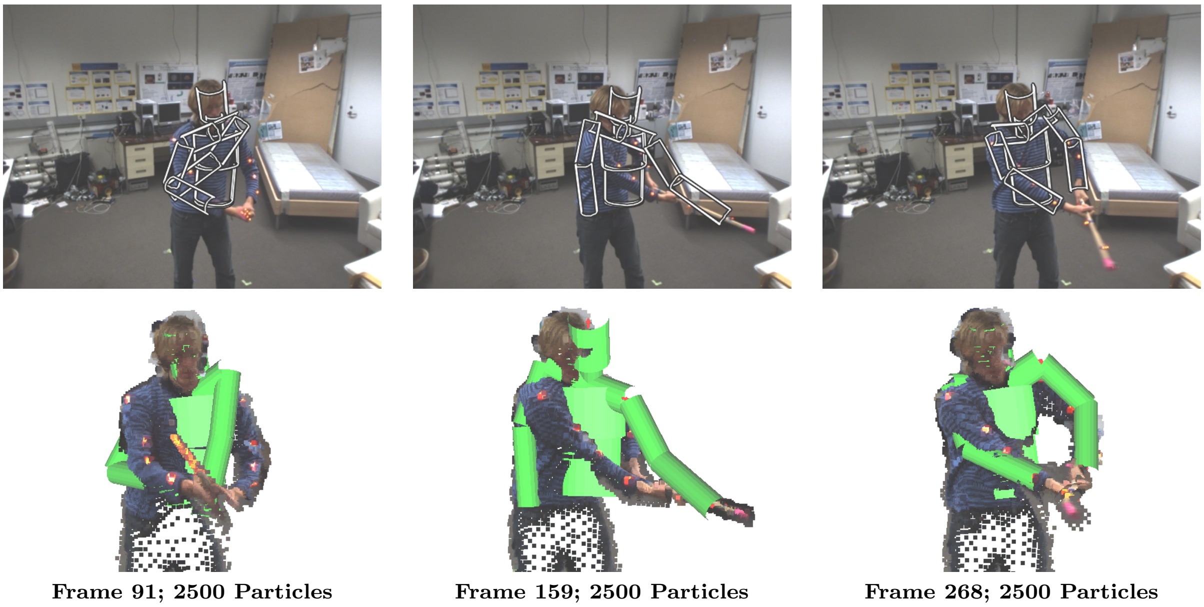


Frame 91; 500 Particles

Frame 159; 500 Particles

Frame 268; 500 Particles

Fig. 9 Results in frame 91, 159 and 268 using the angular model with 500 particles.



Frame 91; 2500 Particles

Frame 159; 2500 Particles

Frame 268; 2500 Particles

Fig. 10 Results in frame 91, 159 and 268 using the angular model with 2500 particles.

7.2 Human–Object Interaction

In the previous section we saw that the activity independent end-effector model improved the results of the angular model. However, results were still not perfect due to the poor data. Inspired by the work of Kjellström et al (2010) we now specialise the end-effector model to include knowledge of the stick position. Assume we know the person is holding on to the stick and that we know the end points of the stick. As the stick is linear, we can write the hand positions as a linear combination of the stick end points.

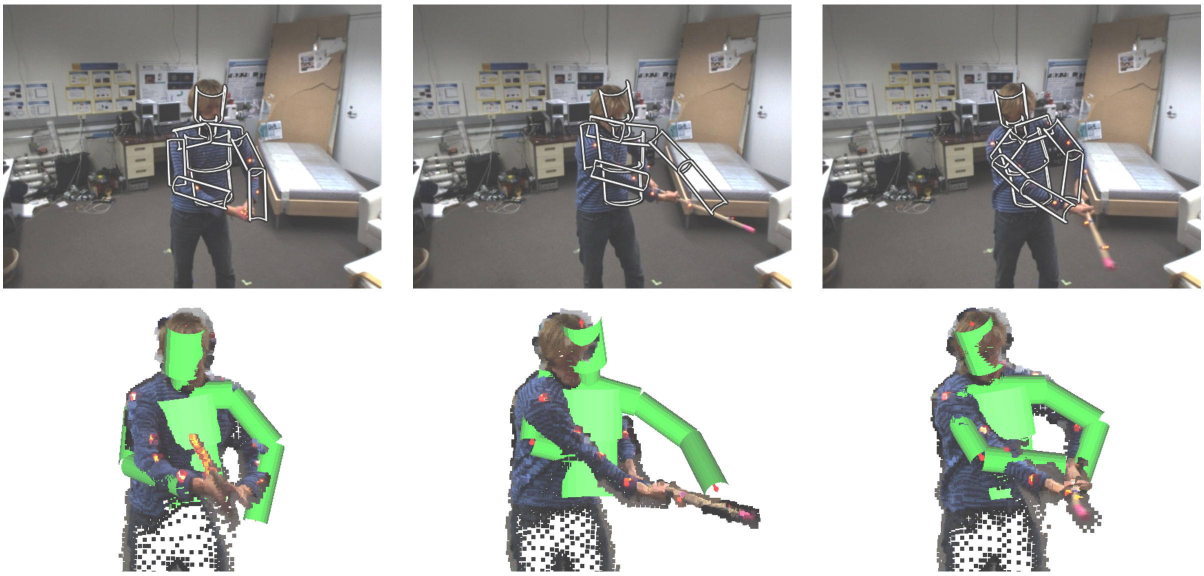
We now model the goal positions of the hands as such a linear combination, i.e. we let

$$\mathbf{g}_t = \mathbf{s}_t^{(1)}\gamma_t + \mathbf{s}_t^{(2)}(1-\gamma_t) \ , \tag{36}$$

where $\mathbf{s}_t^{(i)}$ denotes the end points of the stick. Here we let γ_t follow a normal distribution confined to the unit interval, i.e.

$$p(\gamma_t|\gamma_{t-1}) \propto \mathcal{N}\left(\gamma_t|\gamma_{t-1},\sigma^2\right)\mathcal{U}_{[0,1]}(\gamma_t) \ . \tag{37}$$

We now have a motion model that describes how the person is interacting with the stick. To apply this model

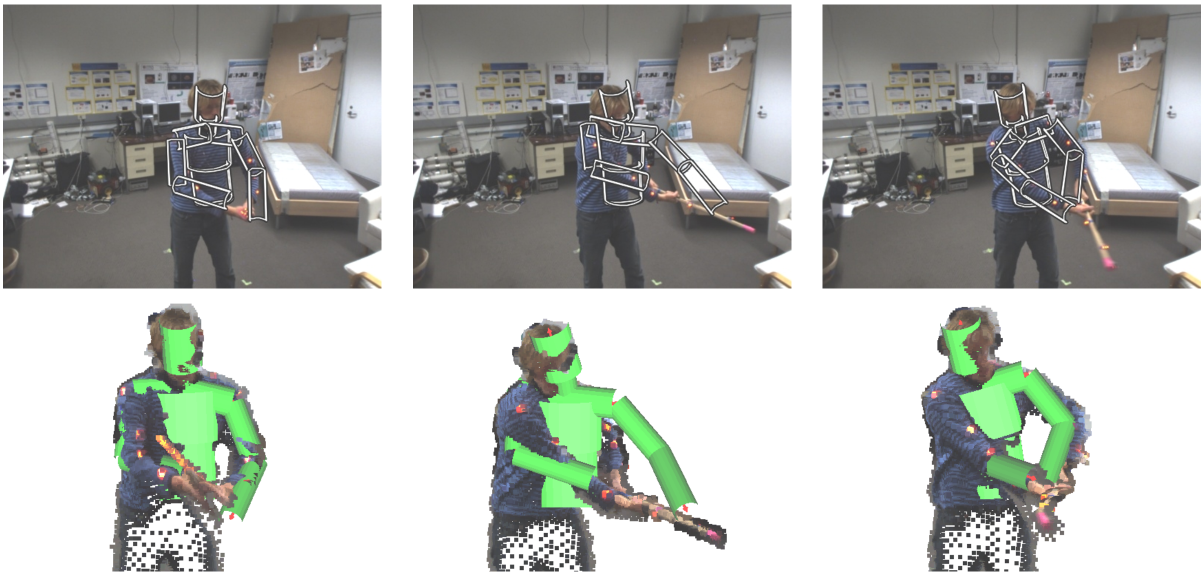


Frame 91; 500 Particles

Frame 159; 500 Particles

Frame 268; 500 Particles

Fig. 11 Results in frame 91, 159 and 268 using the end-effector model with 500 particles.



Frame 91; 2500 Particles

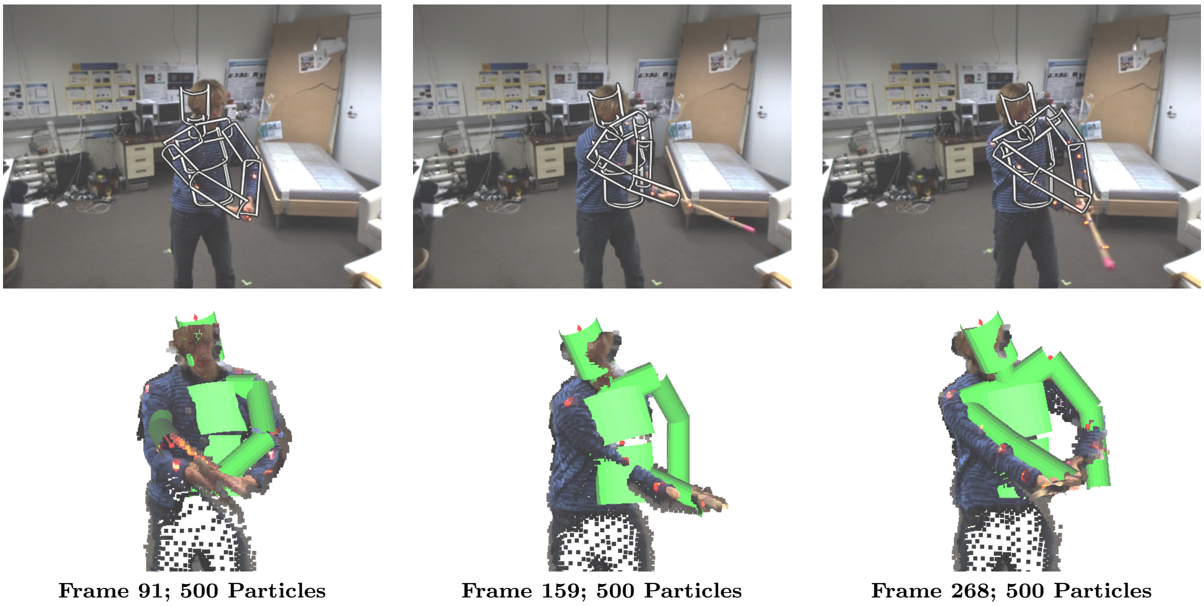
Frame 159; 2500 Particles

Frame 268; 2500 Particles

Fig. 12 Results in frame 91, 159 and 268 using the end-effector model with 2500 particles.

we need to know the end points of the stick at each frame. Here, we simply attain these by placing markers from the optical motion capture system on the stick. In practical scenarios, one would often only have the two dimensional image positions of the stick end points available. The model can be extended to handle this by restricting the hand goals to the plane spanned by the lines starting at the optical centre going through the stick end points in the image plane (Hauberg and Pedersen, 2011).

We apply this object interaction model to the same sequence as before. In fig. 16 we show the attained results using 500 particles on the same frames as before. As can be seen, the results are better than any of the previously attained results even if we are using fewer particles. This is also evident in fig. 17, where the tracking error is shown. In general, it should be of no surprise that results can be improved by incorporating more activity dependent knowledge; the interesting part is the ease with which the knowledge could be incorporated into the model.


Frame 91; 500 Particles

Frame 159; 500 Particles

Frame 268; 500 Particles

Fig. 16 Results in frame 91, 159 and 268 using the object interaction model.

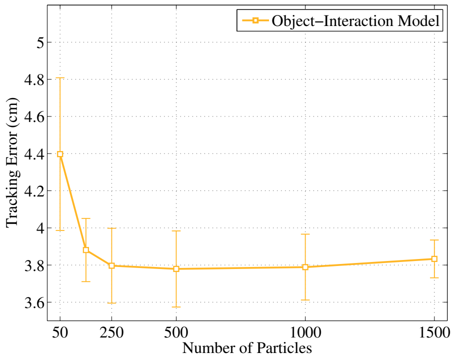


Fig. 17 The error measure $\mathcal{E}(\boldsymbol{\theta}_{1:T})$ when using the object interaction model. The shown results are averaged over several trials; the error bars correspond to one standard deviation.

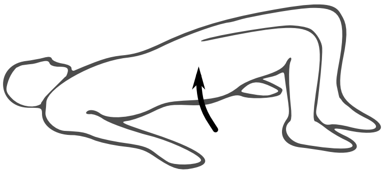


Fig. 18 An illustration of the *pelvic lift* exercise. The patient lies on a flat surface with head, feet and hands fixed. The pelvic region is lifted and lowered repeatedly.

7.3 The Pelvic Lift

We have seen that the suggested modelling framework can be useful to create activity independent motion models and to model interactions with objects. The main motivation for creating the framework is, however, to be able to easily model physiotherapeutic exercises. As a demonstration we will create such a model for the *the pelvic lift* exercise. The simple exercise is illustrated in fig. 18. The patient lies on the floor with bend knees. He or she must then repeatedly lift and lower the pelvic region.

To model this motion we focus on the position of the feet, the hands and the pelvic region. We fix the goals of the feet and the hands, such that they always aim at the position in which the tracking was initialised, i.e.

$$\mathbf{g}_{t+1} = \mathbf{g}_1 \qquad \text{for hand and feet.} \tag{38}$$

The root of the kinematic skeleton is placed at the pelvic region. We model the motion of the root as moving mostly up or downwards, by letting

$$\mathbf{root}_{t+1} = \mathbf{root}_t + \boldsymbol{\eta} \ , \tag{39}$$

where $\boldsymbol{\eta}$ is from a zero-mean Normal distribution with covariance $\mathrm{diag}(\sigma^2, \sigma^2, 16\sigma^2)$. Here, the factor 16 ensures large variation in the up and downwards directions. We further add the constraint that $\mathbf{root}_{t+1}$ must have a positive z-value, i.e. the root must be above the ground plane. This model illustrates the ease with which we can include knowledge of both the environment and the motion in the predictive process.

To illustrate the predictions given by this model, we sample 10 particles from the model. These are drawn in fig. 19. As can be seen, the position of the head, hands and feet show small variation, whereas the position of both the pelvic region and the knees shows more variation. It should be noted that the knees vary as we did

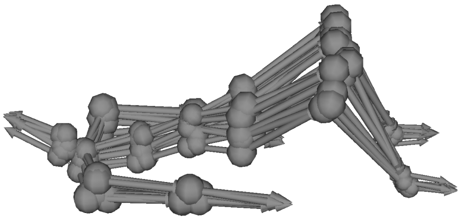


Fig. 19 Ten samples from the tracking of the pelvic lift exercise. Notice how the spine and pelvic region shows large variation in the vertical direction, the knees shows large variation in all directions, while other limbs show low variation. The precision matrix of the importance distribution used to generate one of these samples is shown in fig. 20.

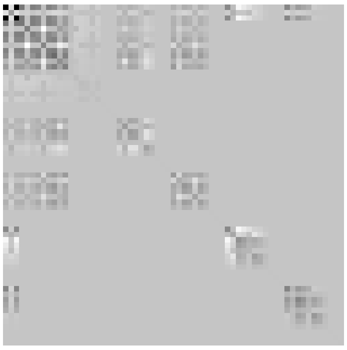


Fig. 20 Precision matrix of the importance distribution used for predicting one of the samples shown in fig. 19. Dark entries correspond to positive values, light entries correspond to negative values while the light grey covering most entries correspond to zero.

not model their motion. To gain further insight into the prediction, we plot the precision matrix of the importance distribution in fig. 20. It can be seen that this matrix has a block-structure indicating that clusters of joints are correlated.

We have applied this predictive model to a sequence where a test subject performs the pelvic lift exercise. The exercise is repeated 6 times and is successfully tracked using 100 particles. A few selected frames are available in fig. 21 and a movie is available on-line[2].

8 Conclusion

In this paper we have presented a probabilistic extension of inverse kinematics. With this, we are able to build predictive models for articulated human motion estimation from processes in the spatial domain. The advantages of this approach are many.

First, we have empirically demonstrated that our spatial motion models improve the tracking using far less particles, while they at the same time provide more smooth motion trajectories compared to simple models in the space of joint angles. In our experience, the traditional motion models in joint angle space actually provide little to no predictive power. The basic issue is that the spatial variance of limb coordinates tends to accumulate with these models as the kinematic chains are traversed. From a practical point of view this means that limbs which are far down the kinematic chains are rarely predicted correctly, meaning many particles are required. Our model does not suffer from this issue as we control the end positions of the chains. We believe this is the main cause of the models efficiency.

Secondly, we saw that the model allows us easily to take the environment into account. We saw this with the stick example, where we could trivially incorporate the stick position into the model. We also saw this with the pelvic lift example, where we modelled the ground plane.

Thirdly, our approach makes it easier to construct high quality models of a large class of motions. Specifically, *goal oriented* motions are usually easy to describe in spatial coordinates. This was demonstrated on the *pelvic lift* exercise, which is trivial to describe spatially, but complicated when expressed directly in terms of joint angles.

Fourthly, our models mostly works in the low dimensional space of end-effector goals, which is simply an ordinary Euclidean space. This makes it more practical to build motion models as we do not need to deal with the topology of the space of joint angles.

We feel there is great need for predictive motion models that are expressed spatially as this seems to mimic human motion plans more closely. It is, however, not clear if our approach of modelling end-effector *goals* is the best way to achieve spatial motion priors. It could be argued that it would be better to model the actual end-effector positions rather than their goals. Our strategy do have the advantage that the resulting optimisation problems are computationally feasible. It also allows us to study stochastic processes in ordinary Euclidean spaces. Had we instead chosen to model the actual end-effector positions, we would be forced to restrict our motion models to the reachable parts of the spatial domain, making the models more involved.

At the heart of our predictive models lies an inverse kinematics solver that computes the mode of eq. 12. In the future this solver should be extended with a collision detection system, such that self-penetrations would be disallowed. We are also actively working on determining more restrictive joint limit models (Engell-Nørregård et al, 2010). Both extensions would reduce the space of possible poses, which would allow us to reduce the number of particles even further.

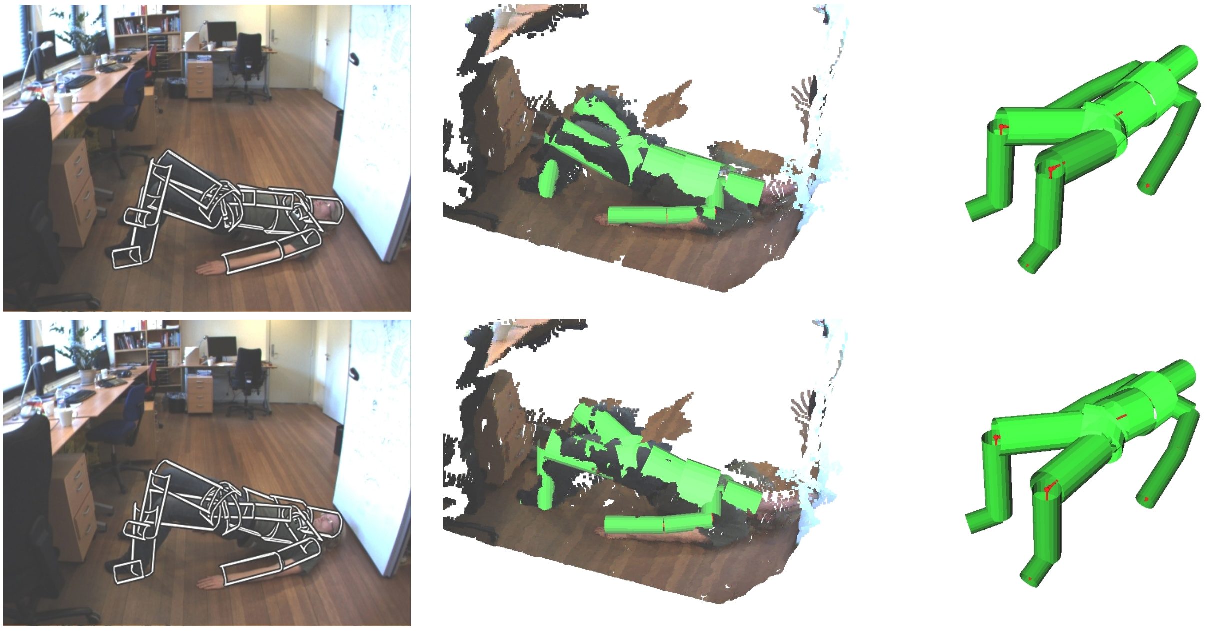


Fig. 21 Frames 71 and 159 from a sequence with a pelvic lift exercise. The exercise is repeated 6 times during approximately 200 frames. The video was recorded at approximately 10 frames per second.

References

Abend W, Bizzi E, Morasso P (1982) Human arm trajectory formation. Brain 105(2):331–348

Belkin M, Niyogi P (2003) Laplacian Eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6):1373–1396

Bishop CM (2006) Pattern Recognition and Machine Learning. Springer

Bregler C, Malik J, Pullen K (2004) Twist based acquisition and tracking of animal and human kinematics. International Journal of Computer Vision 56:179–194

Cappé O, Godsill SJ, Moulines E (2007) An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE 95(5):899–924

Carreira-Perpinan MA, Lu Z (2007) The Laplacian Eigenmaps Latent Variable Model. JMLR W&P 2:59–66

Courty N, Arnaud E (2008) Inverse kinematics using sequential monte carlo methods. In: Articulated Motion and Deformable Objects: 5th International Conference, Springer-Verlag New York Inc, pp 1–10

Elgammal AM, Lee CS (2009) Tracking People on a Torus. IEEE Transaction on Pattern Analysis and Machine Intelligence 31(3):520–538

Engell-Nørregård M, Hauberg S, Lapuyade J, Erleben K, Pedersen KS (2009) Interactive inverse kinematics for monocular motion estimation. In: Proceedings of VRIPHYS'09

Engell-Nørregård M, Niebe S, Erleben K (2010) Local joint–limits using distance field cones in euler angle space. In: Computer Graphics International

Erleben K, Sporring J, Henriksen K, Dohlmann H (2005) Physics Based Animation. Charles River Media

Fletcher TP, Lu C, Pizer SM, Joshi S (2004) Principal geodesic analysis for the study of nonlinear statistics of shape. Trans on Medical Imaging 23(8):995–1005

Ganesh S (2009) Analysis of goal-directed human actions using optimal control models. PhD thesis, EECS Dept., University of California, Berkeley

Grochow K, Martin SL, Hertzmann A, Popović Z (2004) Style-based inverse kinematics. ACM Transaction on Graphics 23(3):522–531

Hauberg S, Pedersen KS (2011) Stick it! articulated tracking using spatial rigid object priors. In: Kimmel R, Klette R, Sugimoto A (eds) ACCV 2010, Springer, Heidelberg, Lecture Notes in Computer Science, vol 6494, pp 758–769

Hauberg S, Lapuyade J, Engell-Nørregård M, Erleben K, Pedersen KS (2009) Three Dimensional Monocular Human Motion Analysis in End-Effector Space. In: Cremers D, et al (eds) Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, LNCS, pp 235–248

Hauberg S, Sommer S, Pedersen KS (2010) Gaussian-like spatial priors for articulated tracking. In: Daniilidis K, Maragos P, , Paragios N (eds) ECCV 2010, Springer, LNCS, vol 6311, pp 425–437

Herda L, Urtasun R, Fua P (2004) Hierarchial implicit surface joint limits to constrain video-based motion capture. In: Pajdla T, Matas J (eds) Computer Vision - ECCV 2004, LCNS, vol 3022, Springer, pp 405–418

Horaud R, Niskanen M, Dewaele G, Boyer E (2009) Human motion tracking by registering an articulated surface to 3d points and normals. IEEE Transactions on Pattern Analysis and Machine Intelligence 31:158–163

Kerlow IV (2003) Art of 3D Computer Animation and Effects, 3rd edn. John Wiley & Sons

Kjellström H, Kragić D, Black MJ (2010) Tracking people interacting with objects. In: CVPR '10: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition

Knossow D, Ronfard R, Horaud R (2008) Human motion tracking with a kinematic parameterization of extremal contours. International Journal of Computer Vision 79(2):247–269

Lu Z, Carreira-Perpinan M, Sminchisescu C (2008) People Tracking with the Laplacian Eigenmaps Latent Variable Model. In: Platt JC, Koller D, Singer Y, Roweis S (eds) Advances in Neural Information Processing Systems 20, MIT Press, pp 1705–1712

Moeslund TB, Hilton A, Krüger V (2006) A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104(2):90–126

Morasso P (1981) Spatial control of arm movements. Experimental Brain Research 42(2):223–227

Murray RM, Li Z, Sastry SS (1994) A Mathematical Introduction to Robotic Manipulation. CRC Press

Nocedal J, Wright SJ (1999) Numerical optimization. Springer Series in Operations Research, Springer

Poon E, Fleet DJ (2002) Hybrid monte carlo filtering: Edge-based people tracking. IEEE Workshop on Motion and Video Computing 0:151

Poppe R (2007) Vision-based human motion analysis: An overview. Computer Vision and Image Understanding 108(1-2):4–18

Rasmussen CE, Williams C (2006) Gaussian Processes for Machine Learning. MIT Press

Rosenhahn B, Schmaltz C, Brox T, Weickert J, Cremers D, Seidel HP (2008) Markerless motion capture of man-machine interaction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 0:1–8

Salzmann M, Urtasun R (2010) Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In: Proceedings of CVPR'10

Sidenbladh H, Black MJ, Fleet DJ (2000) Stochastic tracking of 3d human figures using 2d image motion. In: Proceedings of ECCV'00, Springer, Lecture Notes in Computer Science 1843, vol II, pp 702–718

Sminchisescu C, Jepson A (2004) Generative modeling for continuous non-linearly embedded visual inference. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning, ACM, pp 759–766

Sminchisescu C, Triggs B (2003) Kinematic Jump Processes for Monocular 3D Human Tracking. In: In IEEE International Conference on Computer Vision and Pattern Recognition, pp 69–76

Tournier M, Wu X, Courty N, Arnaud E, Reveret L (2009) Motion compression using principal geodesics analysis. Computer Graphics Forum 28(2):355–364

Urtasun R, Fleet DJ, Hertzmann A, Fua P (2005) Priors for people tracking from small training sets. In: Tenth IEEE International Conference on Computer Vision, vol 1, pp 403–410

Urtasun R, Fleet DJ, Fua P (2006) 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, pp 238–245

Urtasun R, Fleet DJ, Geiger A, Popović J, Darrell TJ, Lawrence ND (2008) Topologically-constrained latent variable models. In: ICML '08: Proceedings of the 25th international conference on Machine learning, ACM, pp 1080–1087

Wang JM, Fleet DJ, Hertzmann A (2008) Gaussian Process Dynamical Models for Human Motion. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2):283–298

Zhao J, Badler NI (1994) Inverse kinematics positioning using nonlinear programming for highly articulated figures. ACM Transaction on Graphics 13(4):313–336