Using GPLVM for Inverse Kinematics on Non-cyclic Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

We apply the Gaussian Process Latent Variable Model (GPLVM) to tackle the inverse kinematic problem in character animation during a ball catching scenario. The goal is to generate realistic upper-body movements with only the tip of the hand specified as the constraint. We ran a series of motion capture experiments to capture the body movement in a subject as he performs a ball catching task and then learn a scaled GPLVM. The ball catching scenario allows us to test the use of GPLVM to represent non-cyclic set of movements with plenty of style variation.

1 Introduction

Inverse kinematics (IK) is a very common task encountered in the field of computer animation and robotics. Its goal is to generate a solution for the degree-of-freedoms (e.g. joint angles) given a small set of constraints, such as the location of the end effector. This is almost always an under-determined problem; although it is trivial to find a solution, finding a good solution is an ongoing research topic [1, 2, 3, 4]. Specifically in character animation, we want to obtain poses that appear natural. In this paper, we present an inverse-kinematic solution base on the Gaussian Process Latent Variable Model (GPLVM) for ball catching.

To obtain natural poses in an IK, one could learn from examples of real characters. Examples of example-based IK include [5, 6, 7, 8, 1]. Due to its flexibility and good performance, we have set out to use the method by [1] to solve the IK task for a ball catching scenario. The method uses a scaled GPLVM to represent the likelihood of the poses. However, their work has been mostly tested on cyclic data set such as walking or sequences with small style variations. On the other hand, the ball catching movement involves many different styles of movement.

2 Gaussian Process Latent Variable Model

2.1 Probabilistic PCA

Probabilistic PCA (PPCA) is a latent variable model which maps the latent space variables, $\mathbf{X} = [\mathbf{x_1}\ \mathbf{x_2}\ \cdots]$ into the observation variables, $\mathbf{Y} = [\mathbf{y_1}\ \mathbf{y_2}\ \cdots]$ through

$$\mathbf{y_n} = \mathbf{W}\mathbf{x_n} + \eta\,.$$

$\mathbf{W} \in \mathcal{R}^{D\times q}$ denotes the mapping coefficient where D and q are the dimensions of the observation and latent space variables respectively. η represents the zero-mean Gaussian distributed noise term with unit covariance:

$$p(\eta_\mathbf{n}|\beta) = N(\eta_\mathbf{n}|\mathbf{0}, \beta^{-1}\mathbf{I})\,,$$

where β is the inverse variance. Assuming independence across data points, the conditional probability of the data is given by

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}, \beta) = \prod_{n=1}^{N} N(\mathbf{y_n}|\mathbf{W}\mathbf{x_n}\beta^{-1}\mathbf{I}) \,.$$

If we pick the prior for the latent variables $\mathbf{X}$ to be a zero-mean Gaussian with unit covariance, we can marginalize the latent variables $\mathbf{X}$ to get

$$p(\mathbf{Y}|\mathbf{W}, \beta) = \frac{1}{(2\pi)^{\frac{DN}{2}}|\mathbf{C}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{C}^{-1}\mathbf{Y}\mathbf{Y}^T)\right) \,,$$

where the covariance is $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \beta^{-1}\mathbf{I}$.

2.2 GPLVM as the dual of PPCA

If we marginalize over the mapping $\mathbf{W}$ instead of the latent variables $\mathbf{X}$, and assuming a Gaussian prior of the mapping, we get

$$p(\mathbf{Y}|\mathbf{X}, \beta) = \frac{1}{(2\pi)^{\frac{DN}{2}}|\mathbf{K}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)\right) \,,$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}$. Note the duality between the above two equations. We can interpret the above likelihood as a product of D independent Gaussian processes by rewriting it as

$$p(\mathbf{Y}|\mathbf{X}, \beta) = \prod_{i=1}^{D} \frac{1}{(2\pi)^{\frac{N}{2}}|\mathbf{K}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2}\mathbf{y_{:,i}}^T\mathbf{K}^{-1}\mathbf{y_{:,i}}^T\right) \,,$$

where $\mathbf{y_{:,i}}$ is the ith column of $\mathbf{Y}$. Here, the Gaussian procees creates a map from the latent space to the data space. We can learn the mapping thgough maximum likelihood with respect to $\mathbf{X}$. This is referred to as the Gaussian Process Latent Variable Model (GPLVM). We now extend this by allowing for non-linear processes by defining a non-linear kernel such as the RBF kernel

$$k(\mathbf{x}, \mathbf{x}') = \alpha \mathrm{exp}\left(-\frac{\gamma}{2}\|\mathbf{x} - \mathbf{x}'\|\right) + \delta_{\mathbf{x},\mathbf{x}'}\beta^{-1} \,,$$

here $\delta_{\mathbf{x},\mathbf{x}'}$ denotes the Kronecker delta and α, β, δ are parameters to the kernel. We can form $\mathbf{K}$ with $\mathbf{K_{i,j}} = k(\mathbf{x_i}, \mathbf{x_j})$. Note that when we jointly optimise the likelihood for the latent space variables $\mathbf{X}$ and the parameters α, β, γ, the cost function will contain many local minima as it is not unique even in the linear case.

3 Inverse Kinematics with GPLVM

Following [1], we can leverage the dimensionality reduction provided by GPLVM to project the character data into a low-dimensional latent space while learning the probability distribution function of the poses in the latent space. This distribution can then be used to select a good pose for the IK problem.

3.1 Character Model

The character pose is determined by a feature vector containing joint angles and the kinematic chain root position. We parameterize the joint angles using quaternions to avoid singularity at the cost of adding an exra dimension (per joint) to the representation [9]. The trade-off is made as such since it is difficult to avoid singularities just by performing a space rotation [1] for an unpredictable and varied arm movement as encountered during the catching task.

3.2 Model Learning

We apply a slight modification to the GPLVM model presented earlier by adding a scaling term to each of the observed variable $\mathbf{y}$ [1]. This can be modeled by using different kernel function

$k(\mathbf{x}, \mathbf{x}')/s_k^2$ for each observation dimension. Alternatively, one could use different covariance function $\mathbf{K}$ for each dimension [10]. This takes into account the different variance in each of its dimension - necessitated by the different characteristic.

To learn the GPLVM representation of the training data $\{\mathbf{y_i}\}$, we use the following priors for the unknown latent variable and parameters: $p(\mathbf{x}) = N(0, \mathbf{I})$ and $p(\alpha, \beta, \gamma) = \alpha^{-1}\beta^{-1}\gamma^{-1}$. We can then maximize the posterior

$$p(\{\mathbf{x_i}\}, \alpha, \beta, \gamma, \{s_k\}|\{\mathbf{y_i}\}) = p(\{\mathbf{y_i}\}|\{\mathbf{x_i}\}, \alpha, \beta, \gamma, \{s_k\}) \left(\prod_i p(\mathbf{x_i})\right) p(\alpha, \beta, \gamma)$$

by minimizing the objective function corresponding to the negative log posterior

$$L_{GP} = \frac{D}{2}\ln|\mathbf{K}| + \frac{1}{2}\sum_k s_k^2 \mathbf{Y_k}^T \mathbf{K}^{-1} \mathbf{Y_k} + \frac{1}{2}\sum_i \|\mathbf{x_i}\|^2 + \ln\frac{\alpha\beta\gamma}{\prod_k s_k^N}$$

with respect to the unknowns.

3.3 Pose Synthesis

Once the scaled GPLVM has been learned from training data, we can use this model to calculate the posterior probability of a new pose as per Gaussian process regression. We can maximize this posterior by minimizing the objective function

$$L_{IK}(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{S}(\mathbf{y} - \mathbf{f}(\mathbf{x}))\|^2}{2\sigma^2(\mathbf{x})} + \frac{1}{2}\ln\sigma^2(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}\|^2$$

where

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mu + \overline{\mathbf{Y}}^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \\ \sigma^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x})^T \mathbf{K}^{-1} k(\mathbf{x})\,. \end{aligned}$$

Here, $\overline{\mathbf{Y}}$ is the mean (μ) subtracted matrix of observed variables and $k(\mathbf{x})_i = k(\mathbf{x}, \mathbf{x_i})$.

The inverse kinematics problem of solving for a pose given some constraints can be solved by optimizing L_{IK} with respect to $\mathbf{x}$ and $\mathbf{y}$ with some constraint $C(\mathbf{y}) = 0$. More formally, we want to solve for:

$$\begin{aligned} &\arg\max_{\mathbf{x}, \mathbf{y}} L_{IK}(\mathbf{x}, \mathbf{y}) \\ &\text{s.t.}\, C(\mathbf{y}) = 0\,. \end{aligned}$$

4 Experiments and Results

4.1 Motion Capture Experiment

We motion captured a sitted subject upper body movement (excluding the left arm) using a set of 21 markers and instructed the subject to catch the ball thrown towards him with his right hand. Each trial starts with the subject taking a pre-defined 'home' pose. We process the motion capture data into a feature vector $\mathbf{y}$ containing the upper torso, head, and arm joint angles as well as the position of the upper torso. The data is downsampled from 100 Hz to 25 Hz.

4.2 Learning the scaled GPLVM

We use the Scaled Conjugate Gradient method [11] to optimize the L_{GP} for learning the scaled GPLVM representation. In our experiment, the dataset was not sufficiently large enough to warrant the use of fast approximations through the information vector machine [1, 12] or other sparse Gaussian Process representations [13, 14].

Using a 2 dimensional latent space, we obtain the latent space representation shown in Figure 1. The intensity of the background indicates the uncertainty of the mapping. A lighter pixel indicates lower uncertainty (higher precision). The different colored traces represents the different latent space trajectory of each catch trial. Due to the non-cyclic nature of the character pose sequence, we do not get a cyclic trajectory in the latent space as often seen in various latent space modeling of human movement examples [1, 15, 10]. We can see that all the trajectories originate from the bottom-left corner of the latent space; this corresponds to the 'home' pose at the start of the experiment.

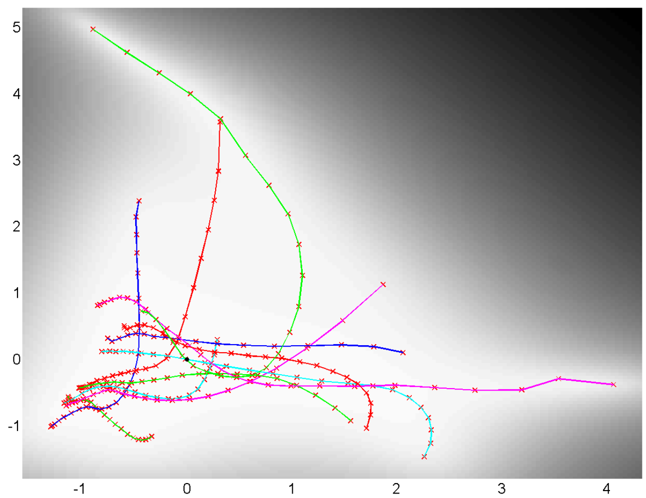


Figure 1: Learned latent space for 11 ball catching sequences. Lines indicate latent space trajectory of each sequence. Red crosses are the actual training points.

4.3 Solving for the Inverse Kinematics

We proceed to use the learned model to perform inverse kinematics as described in subsection 3.3. In the experiment, we set constraints on the position of the tip of the hand. We use the Sequential Quadratic Programming method as implemented in Matlab's `fmincon` function to solve the non-linear constrained optimization problem. Note that both the constraints and the objective function are nonlinear. The nonlinearity of the constraints comes from the rigid transformations needed to transform the observation variable $\mathbf{y}$ into the hand position. We initialize the optimizer by picking a point in the vicinity of the 'home' pose of the latent space and at every iteration, the initialization is updated to use the result of the previous optimization step. This has the benefit of giving some amount of temporal coherence which is important to achieve smooth character motion.

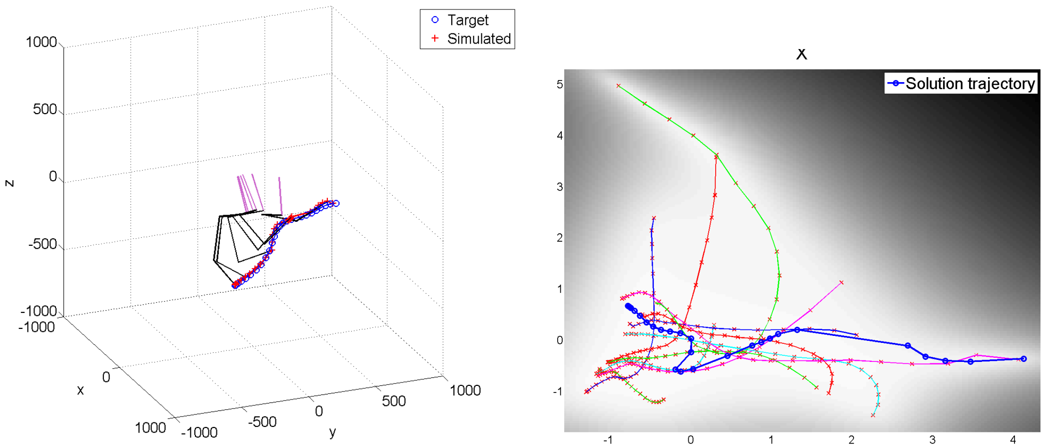
We first ran the inverse kinematic solver on one of the training data set. The result is shown in Figure 2. We can see that the inverse kinematic solution manages to generate poses which gives hand position close to the specified (constraint) trajectory.

A different set of IK tests are performed whereby the constraint trajectories do not come from the training set. An example of the result is shown in Figure 3. Here, the generated poses fail to meet the constraints specified. Furthermore, the latent space trajectory (not shown) snakes haphazardly through the latent space.

5 Discussion

The inability of the method to solve the IK problem in a ball catching scenario comes as a surprise to us in light of the success by [1] in using this method to solve the IK problem. We thus suspect that scaled GPLVM is not suitable for modeling non-cyclic and diverse movements.

We have tried using higher dimensional latent space model to improve the result to no avail. As mentioned before, the objective function is non-convex and thus the optimization process might be stuck in local minima. We have tried adding the model smoothing process used in [1] which involves



(a) Character pose

(b) Corresponding latent space trajectory

Figure 2: Inverse kinematic result on a trajectory in the training set

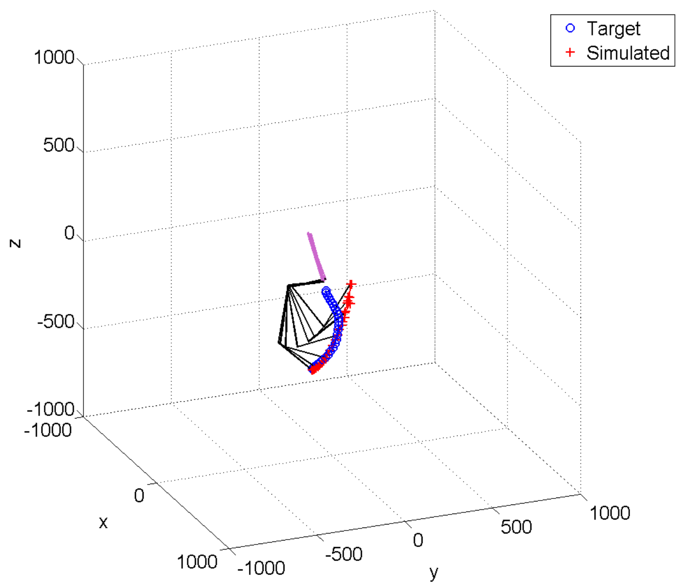


Figure 3: Inverse kinematic result on a trajectory in the test set.

creating smoother version of the model to be used as a starting model to search in the optimization process. This however, does not seem to improve the result obtained.

We suspect that the main reason for the failure of GPLVM to generate a usable latent space stems from the lack of a smooth mapping from the observed space to the latent space even though the converse is true. This causes points close together in data space to not necessarily be close in latent space. We can clearly see this behavior in Figure 1 – the ‘home’ locations for the different sequences in the latent space can be fairly far apart even though they are very similar in the data (character pose) space. This is a commonly observed problem in GPLVM and often causes jumps in the latent space trajectory [10, 15, 16]. In [1], velocity variables are augmented to the observation variable possibly with the intent of alleviating this problem. However when we tried this, we found no significant change to the latent space.

There are several modifications proposed to the GPLVM that can enforce this smoothness. One such proposal is the use of back constraints [17]. Back constraints constraint the latent space to be a smooth function of the observation space. Alternatively, one could incorporate a non-linear dynamic model in the latent space using the Gaussian Process Dynamical Models (GPDM) [16, 15]. Both the back constraint and GPDM has been shown to improve the quality of the learned latent space representation. Further extensions to the GPLVM that can possibly improve the result includes adding topological constraints to improve handling of motion style variations [18].

References

[1] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 522–531, New York, NY, USA, 2004. ACM.

[2] Sethu Vijayakumar and Stefan Schaal. Learning inverse kinematics. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, pages 298–303, 2001.

[3] Ferdinando Mussa-Ivaldi Massachusetts, Ferdinando A. Mussa-ivaldi, and Neville Hogan. Integrable solutions of kinematic redundancy via impedance control. *The International journal of robotics research*, 10(5):481, 1991.

[4] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton : CRC Press, 1994.

[5] George ElKoura and Karan Singh. Handrix: animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 110–119, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[6] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transaction on Graphics*, 21:473–482, July 2002.

[7] C. Rose, M.F. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. *Computer Graphics and Applications, IEEE*, 18(5):32 –40, sep/oct 1998.

[8] D.J. Wiley and J.K. Hahn. Interpolation synthesis of articulated figure motion. *Computer Graphics and Applications, IEEE*, 17(6):39 –45, nov/dec 1997.

[9] Andrew J. Hanson. Visualizing quaternions. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[10] Neil D. Lawrence. The gaussian process latent variable model, 2006.

[11] Martin Fodslette and Mller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525 – 533, 1993.

[12] Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003.

[13] Neil D. Lawrence. Learning for larger datasets with the gaussian process latent variable model. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

[14] Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, December 2005.

[15] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283 –298, feb. 2008.

[16] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *In NIPS*, pages 1441–1448. MIT Press, 2006.

[17] Neil D. Lawrence and Joaquin Quiñonero Candela. Local distance preservation in the gp-lvm through back constraints. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 513–520, New York, NY, USA, 2006. ACM.

[18] Raquel Urtasun, David J. Fleet, Andreas Geiger, Jovan Popović, Trevor J. Darrell, and Neil D. Lawrence. Topologically-constrained latent variable models. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 1080–1087, New York, NY, USA, 2008. ACM.