CGI2014 manuscript No.
(will be inserted by the editor)

Data-Driven Motion Capture Using a Single Smartphone

Haegwang Eom · Byungkuk Choi · Junyong Noh

Abstract Generating a visually appealing human motion sequence using low-dimensional control signals is a major line of study in the motion research area in computer graphics. We propose a novel approach that allows to reconstruct full body human locomotion using a single smartphone. Smartphones are among the most widely used devices and include inertial sensors such as an accelerometer or a gyroscope. To find mapping between a full body pose and smartphone sensor data, we perform low dimensional embedding of full body motion capture data, based on a Gaussian Process Latent Variable Model. Our system ensures temporal coherence between the reconstructed poses by using a state decomposition model for automatic phase segmentation. Finally, application of our nonlinear regression algorithm finds a proper mapping between the latent space and the sensor data. Our framework effectively reconstructs plausible 3D locomotion sequences. We compare the generated animation to ground truth data that are obtained using a commercial motion capture system.

Keywords Motion Reconstruction · Motion Capture · Smartphone · Data-Driven Method

1 Introduction

Motion capture systems have been widely used in computer animation fields to create natural human motion easily and accurately. However, commercial motion capture systems require special equipment that is often too costly for general usage. In this paper, we present a novel method for reconstructing a user's motion using a single smartphone, one of the most widely used, low-cost, and portable devices.

Our approach combines a couple of machine learning algorithms to address the challenge of mapping directly between all joint channels and single sensor data. Our low-dimensional motion model controls the human pose in response to new sensor data. Low dimensional embedding is performed through the Gaussian Process Latent Variable Model(GPLVM). Reconstructing motions pose by pose may cause ambiguity and temporal incoherence. To avoid these problems, we employ a Hidden Markov Model(HMM) for phase segmentation of new input data from smartphone sensors.

2 Related Work

Many studies have focused on reconstructing full body motion from low dimensional sensor inputs. Badler et al. [1] applied real time inverse kinematics to control a standing character using the data captured by four magnetic sensors. Chai and Hodgins [2] presented an online control system that employed a couple of synchronized cameras and several markers. Their online local model utilized various pre-recorded captured data. Pons-Moll et al. [6] presented a hybrid tracker that combined a video with a small number of inertial sensors. Tautges et al. [8] recently suggested a method for high-quality full body motion reconstruction using four accelerometers attached to the end of each limb.

Generating motion in real time sometimes requires a reduction of degree of freedom. To overcome limitations caused by the use of linear regression, Lawrence [4] proposed a nonlinear dimension reduction approach based on GPLVM. This technique has been applied to human

Haegwang Eom · Byungkuk Choi · Junyong Noh
GSCT, KAIST, Daejeon, Korea
E-mail: junyongnoh@kaist.ac.kr

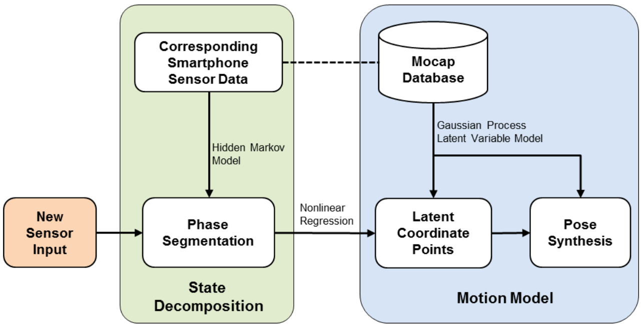

Fig. 1: System overview

motion data and extended to continuous character control [3] [9] [5].

Our approach is similar with previous methods in that the goal is to generate high-quality full body locomotion. Different from Tautges et al. [8], however, we utilize a single smartphone that is equipped with inertial sensors. In addition, unlike Chai and Hodgins [2], we do not require markers or cameras and eliminate spatial constraints during the capture step. Finally, our approach does not rely on search and playback of data from a database, as Slyper and Hodgins [7] does. Therefore, our method provides flexibility against speed variation and produces reasonably interpolated results.

3 Method

3.1 Sensor Recording

We use a Samsung Galaxy S4 smartphone for sensor data recording, although any smartphone equipped with an accelerometer and gyroscope would be equally adequate for the present purpose. As we focus on the reconstruction of locomotion, we attach the smartphone near the ankle using an armband. Calibrated data are specified with the unit of $\mathrm{m/s^2}$ for the accelerometer and rad/s for the gyroscope. All sensor readings are recorded with respect to the local coordinate system of the sensor. Post-processing of the captured data from the smartphone sensors is necessary because the data contain abundant noise. Sensor data are not always reliable and the time unit of recording is not constant. To overcome this problem, we sample the data in a fixed time interval(e.g. 30fps) to obtain feature vectors and linearly interpolate the data for the remaining portions. We apply a 1-D Gaussian filter to each dimension to clean up the data.

3.2 Motion Model

Character motion generally requires a high dimensional description and it is difficult to match the motions directly with low dimensional data obtained from smartphone sensors. Therefore, we create a new low-dimensional space that represents high quality training motion examples. Use of low-dimensional embedding similar to Levine et al. [5] serves this purpose. The pose vector y is composed of each channel value of joint angles including the root. In order to define a mapping from low-dimensional latent variables x to pose vectors y, we use a Gaussian Process(GP) model. After training the system with corresponding (x, y) data, the GP model tries to predict the likelihood of a new vector y for a new input vector x. To account for a different range of variance for each joint, the pose vector's channels are scaled up by a diagonal matrix $\mathrm{W} = \mathrm{diag}\left(w_1, \ldots, w_{d_y}\right)$, where d_y is the dimension of y [3]. To consider correlation among data, we choose a radial basis function(RBF) kernel with the parameter $\overrightarrow{\alpha}$.The log likelihood term about the pose is represented by $\ln p\left(\mathrm{Y}|\mathrm{X},\mathrm{W},\overrightarrow{\alpha}\right)$, which is proportional to the following equation.

$$\mathrm{L_Y} = -\frac{1}{2}\mathrm{tr}\left(\mathrm{K_Y^{-1}YW^2Y^T}\right) - \frac{\mathrm{d_y}}{2}\ln|\mathrm{K_Y}| + \mathrm{N}\ln|\mathrm{W}| \quad (1)$$

where N is the number of training data, K is the kernel matrix, $\mathrm{Y} = [\mathrm{y_1}, \ldots, \mathrm{y_N}]^\mathrm{T}$ and $\mathrm{X} = [\mathrm{x_1}, \ldots, \mathrm{x_N}]^\mathrm{T}$.

In the learning process, a setup is required for parameters such as latent variables X, hyperparameters for kernel functions $\overrightarrow{\alpha}, \overrightarrow{\beta}$, and scaling matrices $\mathrm{W},\mathrm{W_{\dot{Y}}}$. Maximizing the log posterior that includes input pose data Y and its velocity $\dot{\mathrm{Y}}$ achieves this.

$$\begin{aligned}&\ln p\left(\mathrm{X}, \overrightarrow{\alpha}, \overrightarrow{\beta}, \mathrm{W}, \mathrm{W_{\dot{Y}}}|\mathrm{Y}, \dot{\mathrm{Y}}\right) \propto \\ &\mathrm{L_Y} + \mathrm{L_{\dot{Y}}} + \ln p\left(\overrightarrow{\alpha}\right) + \ln p(\overrightarrow{\beta})\end{aligned} \quad (2)$$

where $\mathrm{L_{\dot{Y}}}$ is the velocity GP term for considering temporal coherence. The log posterior is maximized by applying the LBFGS algorithm. The gradients of each likelihood and the priors for the parameters are also calculated in this step [9].

When new sensor input come, a learned model $\Gamma = \{\mathrm{X}, \mathrm{Y}, \overrightarrow{\alpha}, \overrightarrow{\beta}, \mathrm{W}\}$ can predict new pose data $\mathrm{Y_{new}}$ that corresponds to the latent coordinate points by means of a Gaussian distribution.

$$\mathrm{y_{new}} = \mathrm{WY^TK_Y^{-1}k}\left(\mathrm{x_{new}}\right) + bias \quad (3)$$

where k(x) is an $\mathrm{N} \times 1$ vector with an ith element of $\mathrm{k_{rbf}}(\mathrm{x_{new}}, \mathrm{x_i})$. The bias term is a $\mathrm{d_y} \times 1$ vector whose value is determined as the row-wise mean of the entire Y vectors used for the training.

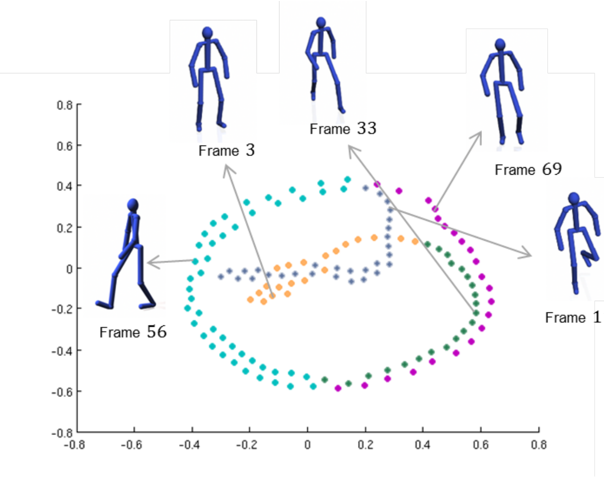

Fig. 2: 2 dimensional GPLVM latent space learned from a two step walking motion clip. Colors represent each phase; orange(from start until taking the left foot off the ground), green(from the left foot moving until taking the right foot off(in the starting part)), cyan(from the right foot moving until taking the left foot off(during walking), purple(the opposite of cyan), gray(from taking the left foot off the ground to stop)

3.3 Hidden Markov Model

Given sensor data S, where S is an N × k matrix with k sensor data dimensions, and its corresponding latent variable X from our motion model, the mapping function from S to X needs to be trained in semantic time pieces to ensure temporal coherence. Figure 2 illustrates the state decomposition in the latent space from GPLVM, where each color represents a different phase. The phase separation prevents possible ambiguity by imposing phase information in the pose reconstruction step.

Assuming that we have the total number of m different phases, the HMM must be trained m times for each phase. In the process of applying HMM for each phase, the discretized matrix S is rearranged based on the classification of the phase and produces the total number of m $\mathrm{c_i}$ block matrices, where each row of $\mathrm{c_i}$ represents a sensor data vector that is classified as the $\mathrm{i}th$ phase.

New input data are cleaned up by means of linear interpolation and a 1-D Gaussian filter, as described in Section 3.1. The result is saved as $\mathrm{S_{new}}$. Before finding the corresponding new latent points $\mathrm{X_{new}}$ from $\mathrm{S_{new}}$, phase classification of each frame is essential. We discretize values of $\mathrm{S_{new}}$ as an input sequence to the HMM built in Section 3.2. We use a window of size v $\mathrm{U} = [\mathrm{u_1}, \ldots, \mathrm{u_{N-v+1}}]^{\mathrm{T}}$. U moves frame by frame, and is applied to each dimension of the sensor data. For every window, the trained HMM determines the probability $\mathrm{P}\left(\mathrm{O}|U\right)$. The phase of U is determined as that with the highest value. Each frame is set to the phase through majority voting by the windows covering that frame.

3.4 Mapping with Sensor Data

After optimizing the low-dimensional latent variable X that represents the entire example motions Y, a mapping between the smartphone sensor data and the latent coordinate points is defined. Even in a low dimensional space, a linear mapping generally leads to poor results. We apply a non-linear neural network $\mathrm{P_{mlp}} = [\mathrm{p_1}, \ldots, p_m]^{\mathrm{T}}$, where $\mathrm{p_i}$ is a MLP model trained by each element of C and its corresponding elements in X in Section 3.2.

4 Results

To verify the effectiveness of our system, we trained the motion model using around 100 frames for each motion type. We trained the HMM with a clip with around 400 frames that present a combination of 4 similar motions. We set the dimension to 5 for latent coordinates for the creation of the motion model. In the HMM, the size of the window was set to 10-20, and we used 2-3 discretization levels.

Figure 3 shows results of the capture for various motions. It is clear that there no special environmental constraints were imposed during the capture. There are differences in proportions between the simple stick figure and the human. Therefore, the angles of the joint may not match exactly. However, the overall characteristics of the motion are very similar between the two. In the clockwise direction from the top left, the images indicate straight walking, running, hopping, and jumping, respectively. The generated motions verifiy that our method is robust against various types of locomotion and actor differences.

4.1 Limitations

Because our system tries to solve a very challenging regression problem and smartphone sensors do not produce very accurate data, it was not easy to generate motion details. Although we utilized velocity GP to produce improved accuracy and root translation, perfect turning motion was not generated as the horizontal root velocity could not be reconstructed faithfully. Because of the attached sensor location, our reconstruction is focused on lower body motion. Therefore, upper

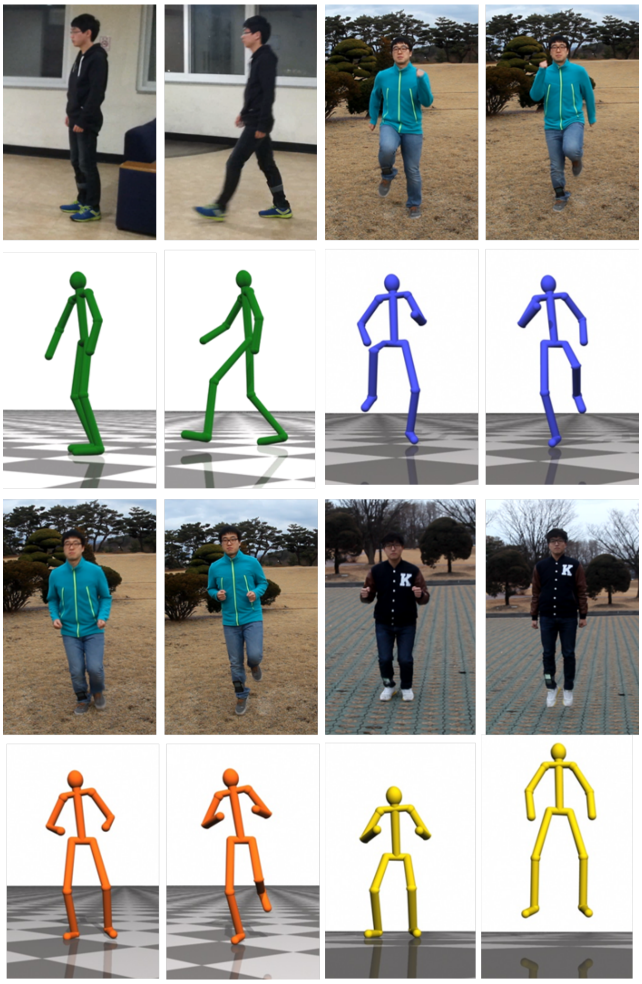

Fig. 3: Results of the capture for various motions.

body motions such as punching or shaking hands are difficult to reconstruct.

Similar to other data-driven methods, motions that are not included in the motion capture database are impossible to recreate. Meanwhile, training with a large data set of motion can cause more ambiguity and result in bad poses. Reconstruction can be performed in less than a millisecond. However, the runtime process cannot be utilized online as the phase segmentation stage should be performed over entire sequences of sensor data.

5 Conclusion and Future Work

We present a novel approach to reconstruct full-body human locomotion using a single smartphone. Our method successfully maps data from inertial sensors to full body 3D poses through GPLVM and nonlinear regression. To consider temporal coherence, we applied HMM and identified the phase of each frame automatically for new sensor input. As shown by various results, our method can reconstruct motions that are comparable to ground truth data.

Our technique can be further developed to capture more general motions such as turning and dancing. We will explore ways to operate our method online to handle new sensor input, via communication between a computer and a smartphone through Bluetooth or wireless internet. This real-time motion capture application will be useful for game control or crowd capture. Combining physics-based refinement is another possible way to create more natural motion.

6 Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(2014R1A2A1A01002871)

References

1. Badler, N.I., Hollick, M.J., Granieri, J.P.: Real-time control of a virtual human using minimal sensors (1993)
2. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. In: ACM Transactions on Graphics (TOG), vol. 24, pp. 686–696. ACM (2005)
3. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. In: ACM Transactions on Graphics (TOG), vol. 23, pp. 522–531. ACM (2004)
4. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. Advances in neural information processing systems **16**(329-336), 3 (2004)
5. Levine, S., Wang, J.M., Haraux, A., Popović, Z., Koltun, V.: Continuous character control with low-dimensional embeddings. ACM Transactions on Graphics (TOG) **31**(4), 28 (2012)
6. Pons-Moll, G., Baak, A., Helten, T., Muller, M., Seidel, H.P., Rosenhahn, B.: Multisensor-fusion for 3d full-body human motion capture. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 663–670. IEEE (2010)
7. Slyper, R., Hodgins, J.K.: Action capture with accelerometers. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 193–199. Eurographics Association (2008)
8. Tautges, J., Zinke, A., Krüger, B., Baumann, J., Weber, A., Helten, T., Müller, M., Seidel, H.P., Eberhardt, B.: Motion reconstruction using sparse accelerometer data. ACM Transactions on Graphics (TOG) **30**(3), 18 (2011)
9. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(2), 283–298 (2008)