Posture Reconstruction Using Kinect with a Probabilistic Model

Liuyang Zhou[1], Zhiguang Liu*[1], Howard Leung[1], and Hubert P. H. Shum[2]

[1]City University of Hong Kong, Hong Kong
[2]Northumbria University, Newcastle upon Tyne, UK

Abstract

Recent work has shown that depth image based 3D posture estimation hardware such as Kinect has made interactive applications more popular. However, it is still challenging to accurately recognize postures from a single depth camera due to the inherently noisy data derived from depth images and self-occluding action performed by the user. While previous research has shown that data-driven methods can be used to reconstruct the correct postures, they usually require a large posture database, which greatly limit the usability for systems with constrained hardware such as game console. To solve this problem, we present a new probabilistic framework to enhance the accuracy of the postures live captured by Kinect. We adopt the Gaussian Process model as a prior to leverage position data obtained from Kinect and marker-based motion capture system. We also incorporate a temporal consistency term into the optimization framework to constrain the velocity variations between successive frames. To ensure that the reconstructed posture resembles the observed input data from Kinect when its tracking result is good, we embed joint reliability into the optimization framework. Experimental results demonstrate that our system can generate high quality postures even under severe self-occlusion situations, which is beneficial for real-time posture based applications such as motion-based gaming and sport training.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity

Keywords: Posture Reconstruction, Gaussian Process, Kinect

1 Introduction

Human posture recognition is a core part of interactive applications. Traditional interaction systems such as dance training application are based on motion capture technology, where the user's movements are captured by an optical motion capture system [Chan et al. 2011]. While these applications can evaluate user performance by accurately capturing user's motions, they are not convenient since the user has to wear a tight suit with reflective markers on it. Moreover, these devices are relatively expensive and not affordable for home use.

Portable devices such as the Microsoft Kinect [Microsoft 2013] serves as an alternative to capture the human movements for interactive applications. Kinect is a controller-free hardware that infers 3D positions of human body joints from a single depth image with the help of a data driven machine learning algorithm [Shotton et al. 2011]. For example, [Morgan et al. 2014] proposed a tool based on Kinect to accelerate the implementation of natural user interface in virtual reality. It is convenient and intuitive to use Kinect for gesture based applications such as interactive virtual reality games. While Kinect can robustly track the 3D posture of the user, the captured data suffer from poor precision due to self-occlusions and insufficient information provided by Kinect sensor. Therefore, Kinect based interactive applications usually require the user to face the device so that individual body part is observable, which greatly limit the system flexibility. In addition, the user has to minimize self-occlusion postures, or Kinect would misrecognize body parts. As illustrated in Figure 1, the blue skeleton represents the tracked result by Kinect SDK [Microsoft 2013]. We can see the tracked arms are twisted when there are some self-occlusions. Therefore, it is essential to develop an effective posture reconstruction method with Kinect for interactive applications.

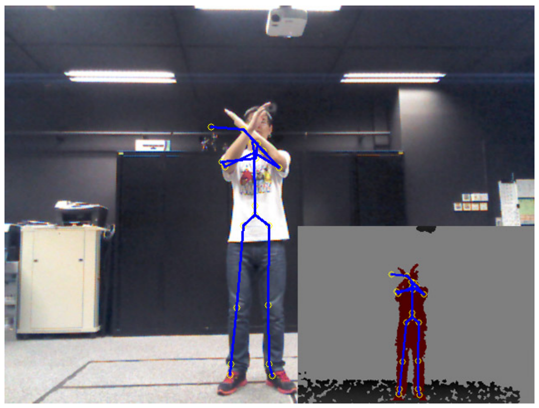


Figure 1: *Example of an inaccurately tracked pose from Kinect. The blue skeleton is the tracked result by Kinect.*

The occlusion problem and incompleteness of the tracked joints remain challenging despite the posture reconstruction research proposed in the past years. Generating poses from low dimensional signals is a potential solution for posture reconstruction [Chai and Hodgins 2005; Liu et al. 2011]. However, these methods assume the low dimensional signal to be stable and accurate, while joints tracked by Kinect are not. Hence, applying them to reconstruct Kinect postures will create unsatisfying results. Shum et al. [Shum et al. 2013] applies reliability measurement to improve the posture reconstruction process, but it still needs a large database. The result will degrade significantly if no similar posture is found.

In this paper, we propose a probabilistic model based on Gaussian Process (GP) to reconstruct postures captured from Kinect. Unlike previous systems that require a large marker-based motion database, GP based model can be robustly trained from small training sets. Moreover, the parameters of the

*zhigualiu2-c@my.cityu.edu.hk

kernel function can be optimized without relying on experimental cross validation [Rasmussen and Williams 2005]. The proposed model consists of three terms to constrain the solution space so that the reconstructed posture is accurate while maintaining the originality of the input posture from Kinect. We adopt Gaussian Process model as a spatial prior distribution to guide the optimized posture towards marker-based motion capture data, which implicitly ensures the reconstructed posture to be accurate due to the high accuracy of marker-based motion capture data. Furthermore, since reconstructing each posture independently cannot ensure the temporal smoothness of the pose sequence, we introduce a temporal consistency term to constrain the velocity variations between successive frames. Inspired by [Shum et al. 2013], we embed the reliability of each joint into the optimization framework to make sure the reconstructed posture resembles the observed input data from Kinect when the tracking result is good. The experimental results demonstrate that the proposed approach is effective to reconstruct a number of motions containing self-occlusions. For example, as illustrated in Figure 4(a), our method is able to handle postures of bending over with a number of joints occluded.

The major contributions of this paper are summarized as follows:

1. We propose a new unified framework for posture reconstruction using Kinect. The system optimizes a posture live captured by Kinect, which maintains the correctness of the posture while preserving temporal smoothness between frames. The proposed system can handle movements even when a number of joints are occluded.

2. We propose three terms to constrain the solution space. The spatial prediction term encourages the optimized posture as closely as marker-based motion capture data based on a Gaussian Process prediction model. The temporal prediction term ensures the temporal consistency between consecutive frames. Furthermore, The reliability term guides the optimized posture towards the Kinect posture, which preserves the property of the input pose from Kinect.

The rest of the paper is organized as below. We first review the related works of posture reconstruction in Section 2. Section 3 explains the procedure of data acquisition and preprocessing. In section 4, we elaborate the terms of the objective function for posture reconstruction, namely spatial prediction term, temporal prediction term and reliability term. Experimental analysis and evaluation are conducted in Section 5. Finally, in Section 6, we conclude the paper, as well as discuss the limitations and future research directions.

2 Related Work

With the advancement in real-time depth camera such as Kinect sensor, human motion recognition and pose estimation has become a popular research topic in recent years. Kinect sensor consists of 3D depth sensors and combined with a RGB camera, which records depth data and video data. It is based on motion recognition technology proposed by [Shotton et al. 2011] , where they use per-pixel classification method to quickly predict 3D joint positions from a single depth image. A number of research domains have emerged based on Kinect, such as human-machine interaction [Tashev 2013], natural user interfaces [Morgan et al. 2014], and 3D reconstruction [Izadi et al. 2011] etc. A recent review on human activity analysis with Kinect can be found in [Han et al. 2013]. Bailey and Bodenheimer [Bailey and Bodenheimer 2012] investigated the perceived differences in the quality of animation generated using motion capture data and a Kinect sensor, which clearly showed that the data recorded from Kinect is with lower quality compared with motion capture data by a Vicon motion capture system. Hence, it is essential to develop an effective posture reconstruction method to obtain high quality postures from Kinect. In this section, we briefly review the related works about posture reconstruction.

2.1 Tracking based Posture Reconstruction

Human motion tracking can be considered as one way for posture reconstruction. Tracking based approaches usually require registering a 3D articulated model with depth information. [Wei et al. 2012] formulated the registration problem into a Maximum A Posteriori (MAP) framework to register a 3D articulated human body model with monocular depth via linear system solvers. They integrate depth data, silhouette information, full body geometry and temporal pose prior into a unified framework. To tackle with the problem of manual initialization and failure recovery, they combined 3D pose tracking with 3D pose detection. Although the proposed algorithm is parallel and can be implemented on GPU to accelerate the speed, it requires the calibration procedure to process skeleton of different sizes. On the contrary, our system is invariant to the location and the skeleton size of the user.

[Yasin et al. 2013] introduced a model based framework for full body reconstruction from 2D video data using motion capture database as the prior knowledge. Two dimensional features were extracted from both the input video sequence and motion capture database. Based on the obtained features, postures were reconstructed in an optimization framework, in which similar motion capture postures were retrieved through nearest neighbor searching. However, the accuracy is not robust because the 2D features projected from 3D motion induces posture ambiguity. [Iason Oikonomidis and Argyros 2011] proposed a model-based 3D tracking of hand articulations using Kinect to minimize the discrepancy between a hand model and actual hand observations. Although it is possible to solve the optimization procedure using a variant of Particle Swarm Optimization (PSO), the method lacks the ability of generalization since the system relies on a predefined hand model.

To enhance the physical validation of the reconstructed motions, dynamical models are used to constrain the solution space. [Taylor et al. 2010] introduced a probabilistic latent variable model called the Implicit Mixture of Conditional Restricted Boltzmann Machines for human pose tracking. [Vondrak et al. 2013] proposed to use a simulation-based dynamical motion prior for human motion tracking. However, these dynamic based approaches are computational costly and hence not applicable for Kinect based interactive applications.

2.2 Posture Reconstruction from Low Dimensional Signals

The full body postures can be represented by a set of low dimensional signals [Chai and Hodgins 2005]. Some research work has been proposed to reconstruct a posture with a small set of signals. [Kim et al. 2013] reconstructed human motion from 3D motion sensors on a performer using kernel CCA-based regression. Given the input data from sparse motion sensors, they retrieve similar poses from the motion capture database and transform the low dimensional signal into the pose space using an online local model. [Chai and Hodgins 2005] employed a small set of retro-reflective markers to create a performance control system, where the set of markers are used to find the best matched motion data for reconstruction. In their system, the low dimensional control signals from the user's performance were supplemented by a pre-recorded human motion database. At run time, the system automatically learned some local models from the retrieved motion capture data that were close to the marker locations recorded by the camera. Their system only needs video cameras and a small set of markers, which makes it low cost and practical for home use. However, a user cannot move freely in the space as most of the markers needed to be seen by at least one camera.

[Liu et al. 2011] used a small number of motion sensors to control a full-body human character. They construct online local dynamic models from prerecorded motion capture database and use them to construct full-body human motion in a maximum a posteriori framework, in which the system tried to find the most similar poses from database for reconstruction. [Helten et al. 2013] adaptively fused inertial and depth information in a hybrid framework for pose estimation. Although these methods can be used to reconstruct postures from low dimensional signals, there is one assumption that these low dimensional signals are reliable and stable. It is not applicable to Kinect data as poses from Kinect are not stable and consistent.

2.3 Data-Driven Posture Reconstruction

Kinect is a RGB-D sensor, which provides more information than monocular camera. [Sigalas et al. 2013] presented a data-driven model based method for 3D torso pose estimation from RGB-D image sequence. Although their method can extract the upper body pose of users without initialization phase, they did not cope with full body pose recovery nor handle the occlusion problem. [Shum and Ho 2012; Shum et al. 2013] proposed a unified framework to control physically simulated characters with live captured motion from Kinect by searching similar poses from marker-based motion database. They constructed posture space based on the retrieved similar postures. The postures were reconstructed with the posture space in an optimization framework. [Baak et al. 2011] introduced a data driven approach for full body reconstruction from a depth camera. They proposed an efficient algorithm for extracting pose features from the depth data. However, for the fast movements,the proposed system required all five extremities to be visible. [Shen et al. 2012] introduced an exemplar-based method to correct the poses from Kinect using marker-based motion data. Data-driven methods need to construct a large motion capture database as a prior knowledge. The reconstruction results highly depend on the retrieval results.

In this paper, we use Gaussian Process (GP) to model the prior distribution of postures from Kinect with marker-based motion data. Our approach is effective even with small training data as GP based model can robustly learn from small training sets. Moreover, our method requires no manual intervention such as marker labeling in model based works.

3 Data Acquisition and Preprocessing

For brevity, in this paper we will use *MOCAP* to represent human motion data captured by an optical motion capture system. The postures obtained from Kinect are noisy and incomplete, whereas MOCAP is accurate and stable. Hence, we can use MOCAP to recover postures from Kinect.

In this paper, we model the relationship between Kinect data and MOCAP with Gaussian Process. Specifically, we capture motions with Kinect and optical motion capture system at the same time to identify the correspondence between them. The setup of this capturing procedure is shown in Figure 2. The pose of Kinect at time t is denoted as $X_t = (x_t^1, x_t^2, ..., x_t^i), x_t^i \in R^3$, where x_t^i represents the 3D joint position of joint i over time t. There are 20 joints based on the skeleton definition of Kinect, i.e. $i = 20$. The corresponding MOCAP of X_t is denoted as $M_t = (m_t^1, m_t^2, ..., m_t^i), m_t^i \in R^3$. To enhance the robustness

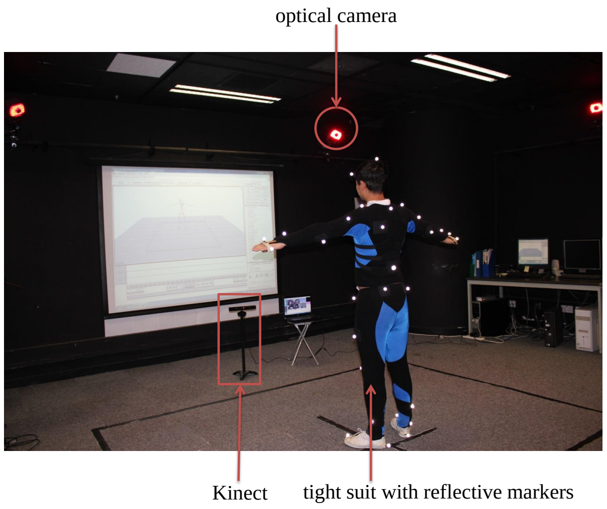


Figure 2: *Human motion capture with Kinect and an optical motion capture system.*

of the spatial prediction model (Section 4.1) and to make the system invariant to different subjects, we normalize the postures for prediction. Specifically, the prediction model should output the same results given the same posture performed by different subjects when they are facing different directions. Here, we use the normalized data of X_t as the input of the predictor, that is $\tilde{X}_t = F(X_t)$, where $F(\cdot)$ represents the posture normalization procedure and retargeting of the user's skeleton into a fixed skeleton size. The posture normalization procedure consists of two steps: removing the rotation along the vertical axis and the global 3-D translation. The retargeting procedure ensures the system to be invariant to the skeleton size of the user. We follow [Shum et al. 2013] to conduct the normalization and retargeting as it is simple yet effective.

4 Posture Reconstruction

To ensure that the reconstructed posture is accurate and resembles the input data from Kinect, we formulate the posture reconstruction as a optimization problem by minimizing an energy function. The energy function consists of a set of three energy terms to constrain the solution space, which are spatial prediction term, temporal prediction term, and reliability constraint. In the following, we will elaborate the definition and purpose of each term.

4.1 Spatial Prediction

Assuming that the MOCAP posture M_t is the corrected pose of the Kinect posture X_t, we design a spatial prediction term to evaluate how well the reconstructed posture fits with the MOCAP data, which implicitly favors solutions that are more similar to the correct posture.

Due to self-occlusions, there will be residual offset between X_t and M_t, which is calculated by $Y_t = M_t - X_t$, where $Y_t = (y_t^1, y_t^2, ..., y_t^i), y_t^i \in R^3$. During run-time, the objective is to predict the residual offset Y_t so that we can obtain the reconstructed pose M_t by appending Y_t to X_t. Obtaining Y_t is regarded as a prediction problem when given X_t. In this paper, we adopt the non-parametric method Gaussian Process (GP) as the predictor. As

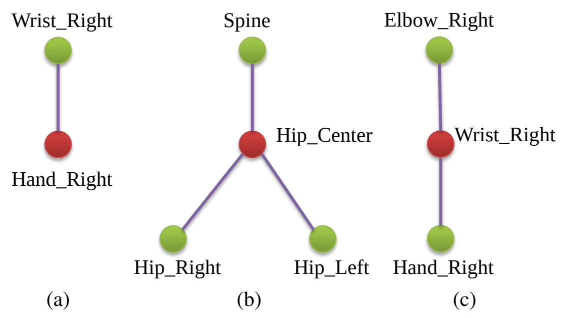


Figure 3: *Three cases of neighboring joints. The red dot represents the joint for prediction, the green dots represent its neighboring joints. a) Right hand joint; b) Hip center joint; c) Right wrist joint.*

GP based models can be robustly learned from small training sets, the usability of our algorithm is enhanced. GP is based on the assumption that adjacent observations should convey information about each other, and the coupling between observations takes place by means of the covariance matrix. More formally, let $\mathbf{A} = [a_1, \ldots, a_N]^T$ be a matrix representing the input data, which is constructed by concatenating the extracted source features. Let $\mathbf{B} = [b_1, \ldots, b_N]^T$ denote the output values. b_i is the corresponding output of the input a_i, $b_i = f(a_i) + \epsilon$ where $\epsilon \sim N(0, \beta^{-1})$ is a noise variable, which is independent for each data point. The shape of function f is determined by the selection of covariance function, in this work we use the following covariance function:

$$k(a_i, a_j) = \theta_0 \exp\left(-\frac{\theta_1}{2}||a_i - a_j||^2\right) + \theta_2 + \beta^{-1}\delta_{ij} \quad (1)$$

where δ_{ij} is Kronecker's delta function and $\Phi = \{\theta_0, \theta_1, \theta_2, \beta\}$ are the hyper-parameters. For more details about GP, we refer the readers to [Rasmussen and Williams 2005].

Human body joints are highly coordinated and it is important to take into account the relationship between them. Here, given one joint, we use its neighboring joints for prediction. Specifically, given joint i at time t, x_t^i, its neighboring joints $N(x_t^i)$ are defined as the set of joints that are directly connected with the same bone segment as joint i. Figure 3 shows three examples, the joints in green color are the neighboring joints of those in red color. For example, Figure 3(a) illustrates the neighboring joint of the right hand is right wrist. With such neighbor relationship extraction, the prediction model implicitly models the relationship between joints. Therefore, the input feature for obtaining y_t^i of joint i is the union set of $\tilde{x}_t^i$ and $N(\tilde{x}_t^i)$. $N(\tilde{x}_t^i)$ is the normalized data of the neighboring joints for joint i. Therefore, the input data is $\tilde{X}_t$ and $N(\tilde{X}_t)$, which correspond to $\mathbf{A}$. The output data is Y_t that corresponds to $\mathbf{B}$ of the prediction model. At training stage, with the obtained training data from Kinect and MOCAP, we can learn the hyper-parameters of the GP model.

With the learned model, we formulate the above prediction for Y_t as a conditional probability distribution, yielding the spatial prediction energy term as defined below:

$$\begin{aligned} E_S &= \ln p(Y_t|\tilde{X}_t, N(\tilde{X}_t)) \\ &= \frac{||Y_t - \mu(\tilde{X}_t, N(\tilde{X}_t))||^2}{2\sigma^2(\tilde{X}_t, N(\tilde{X}_t))} \end{aligned} \quad (2)$$

where μ and σ are the predicted mean and covariance functions from GP prediction respectively. The term E_S ensures that the reconstructed pose are close to the correct pose as much as possible. We use the residual offset as the output of the predictor because the system could explore more in the data space by predicting offset rather than the pose data directly, e.g. M_t. There are several publicly available implementations of Gaussian Process. In this paper, we used the library developed by Lawrence [Lawrence 2009]. To improve the performance of Gaussian Process prediction, we adopted the sparse approximation strategy proposed by Candela and Rasmussen [Quiñonero Candela and Rasmussen 2005].

4.2 Temporal Prediction

The above spatial prediction considers each posture independently. To ensure the temporal smoothness between consecutive frames, the relationship between frames is modeled as a second order temporal model, which has been verified to be effective to preserve temporal smoothness [Sidenbladh and Black 2001]. Specifically, we adopt a constant velocity variation to smooth velocity, which is formulated as below:

$$E_T = \ln p(M_t|M_{t-1}, M_{t-2}) \quad (3)$$

M_t, M_{t-1}, and M_{t-2} are the reconstructed postures at time slices t, $t-1$, and $t-2$. We have the following relationship between the reconstructed posture, input posture and the residual offset:

$$M_t = Y_t + X_t \quad (4)$$

Therefore, we can rewrite Equation 3 as :

$$\begin{aligned} E_T &= \ln p(Y_t + X_t|M_{t-1}, M_{t-2}) \\ &= ||(M_t - M_{t-1}) - (M_{t-1} - M_{t-2})||^2 \\ &= ||M_t - 2M_{t-1} + M_{t-2}||^2 \\ &= ||Y_t - (-X_t + 2M_{t-1} - M_{t-2})||^2 \end{aligned} \quad (5)$$

which facilitates the continuity in the reconstructed motions.

4.3 Reliability Embedding

The accuracy of each tracked joint is different depending on the degree of occlusion. These incorrectly tracked joints from Kinect will incorrectly guide the system to infer the joint positions. The residual offset of these correctly tracked joints should be smaller as they are closer to the corrected posture, namely M_t. Thus, it is essential to consider the reliability of each joint to constrain the residual offsets of these joints with higher confidence during the prediction of Y_t. We use a reliability term E_R to penalize the residual offset of each joint based on its reliability, which implicitly ensures that the reconstructed pose resembles the input pose from Kinect as much as possible. More specifically, the residual offset value y_t^i of joint i should be smaller if the corresponding joint is with higher reliability.

We adopt the strategy proposed by [Shum et al. 2013] to evaluate the reliability of the tracked joints from Kinect. They evaluate the reliability in three aspects: behavior reliability, kinematics reliability, and tracking state reliability. The behavior reliability refers to abnormal behavior of a tracked joint, which is calculated by the cosine similarity between two consecutive displacement vectors of one joint. The kinematics reliability represents the kinematic correctness of the tracked joints, which measures the change of bone length that connected with that joint. The tracking state reliability tells if a joint is tracked, inferred or not tracked when it is completely occluded. More details about the calculation of the reliability of each joint can be found in [Shum et al. 2013]. As a result, the reliability rate of each joint is a value between 0.0 and 1.0 (inclusive). We embed the reliability of each joint into the optimization framework and come up with the following reliability term:

$$E_R = ||RY_t||_F^2 \quad (6)$$

$|| \cdot ||_F$ is the Frobenius norm. The entry of R is the reliability of each joint, which ensures the reconstructed posture does not deviate from the input pose from Kinect. Intuitively, while minimizing the objective function, the value of y_t^i tends to be small when its reliability value is large.

4.4 Energy Minimization Function

Equipped with the terms defined in the above sections, the posture reconstruction problem is formulated as the following optimization function:

$$E = \arg\min_{Y_t} \{w_s E_s + w_T E_T + w_R E_R\} \tag{7}$$

where w_S, w_T, and w_R are the weights of the energy terms. In our implementation, they are empirically set to be 0.6, 0.2, and 0.2, respectively. Our posture reconstruction system is a frame-based framework. The initial posture for optimization at each frame is defined as previous reconstructed posture, which makes the system has more chance to find the optimized posture. The optimization procedure stops when an optimal solution is found or the number of iterations reaches a predefined threshold.

There are some principles to tune the values of the weights. The weight of the spatial prediction term should be set the largest, since this term drags the reconstructed posture to the corrected pose as closely as possible. Second, the temporal prediction term ensures the temporal stability of the pose sequences. The details of the movements will be smoothed out as the term is based on a second order temporal prediction, which explicitly constrains the speed variation between consecutive frames. The reliability term makes sure the reconstructed posture is as similar as the Kinect posture, since the primary purpose of the system is to reconstruct Kinect postures. We will evaluate how these terms affect the accuracy of the system in Section 5.4.

As a summary of the proposed frame work for posture reconstruction. At offline stage, we learn a spatial prediction model using Gaussian Process with pairwise Kinect data and marker-based motion capture data. It ensures the reconstructed posture as accurate as MOCAP data. At online stage, the system obtains an optimized posture with live captured data from Kinect, which ensures the reconstructed posture resembles the input pose from Kinect while maintaining the temporal smoothness between previous frames. Although we use pose data from Kinect, the overall framework is applicable to other system with different design of equations.

5 Experimental Results

In this section, we will show the experimental results and present the comparisons between alternative approaches such as the postures estimated by Kinect SDK [Microsoft 2013] and the work proposed by [Shen et al. 2012]. We first show results of posture reconstruction by our approach, where the postures are with severe self-occlusions. Qualitative and quantitative analysis are conducted to evaluate the accuracy between the proposed method and other alternatives.

The experimental results were conducted on a desktop computer with Intel Core 2 Duo 3.17 GHZ processor. The input data of our system are the postures estimated from Kinect. We obtain the 3D position of each joint through the latest official SDK (i.e. v1.8) provided by Microsoft [Microsoft 2013]. An optical motion capture system with 7 cameras was used to capture MOCAP data in our own laboratory. The infrared emitters from the optical motion capture system will interfere with the Kinect. Here, we consider the Kinect device as one additional reflective marker of the optical motion capture system to eliminate the interference between Kinect and the optical motion capture system. The setup environment of Kinect and optical motion capture system is shown in Figure 2. On average, the system can reconstruct postures at 22 frame per second which is enough for realtime interactive applications.

5.1 Posture Reconstruction

The proposed approach works for users with different body sizes and proportions, because we use the normalized postures from Kinect as the input of the prediction model, where the rotation along the vertical axis and global 3D translation of each posture has been removed. We evaluate our system on a wide range of human motions, including sports activity such as Tai Chi, bending, golf swinging, and daily actions such as crossing arms, waving right hand, clapping hands, rolling hands up and down, rolling hands forward and backward. The details of the motion database used in our method and [Shen et al. 2012] are reported in Table 1, which shows that our system can use around 75% less data in the database.

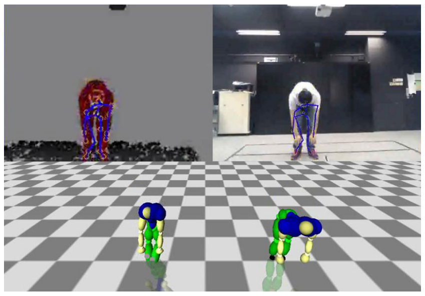
Table 1: *The detailed number of frames for each type of motion database used in our method and Shen et al. 2012.*

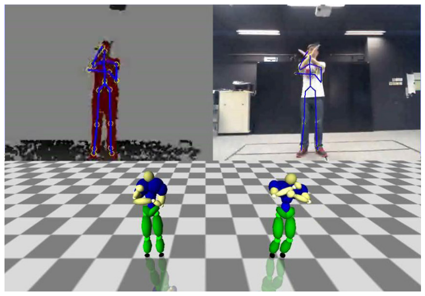
Motions	Our method	Shen et al. 2012
Tai Chi	650	2320
Bending	320	1580
Golf Swinging	460	1765
Crossing Arms	380	1685
Waving Right Hand	350	1650
Clapping Hands	420	1720
Rolling Hand Up and Down	480	1840
Rolling Hand Forward and Backward	475	2050
Bending Leg	385	1890

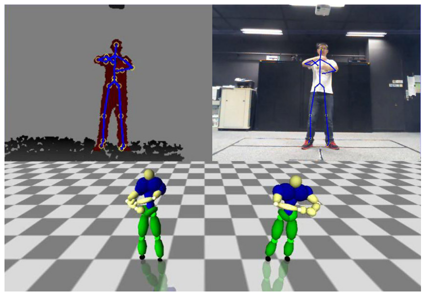
We choose these motions because all these motions contain severe self-occlusions, which are not well tracked by the Kinect system. However, the proposed method can well reconstruct these inaccurate poses even if a number of joints cannot be tracked by the Kinect sensor. For example, for the crossing hands motion, the joints of the body are occluded by the hands and the joints of the hands are occluded by each other. For a bending motion, the upper body joints are occluded by the arms. Figure 4 showed several frames of our results, the right avatar in front represents the postures reconstructed by our method and the left avatar in front corresponds to the estimated posture by Kinect SDK [Microsoft 2013]. The top two pictures are the RGB image and depth image respectively. We can observe that certain parts of the postures from Kinect SDK are twisted when there exist occlusions whereas our method can reconstruct the postures very well. The readers are referred to the accompanying video for better visualization of the posture reconstruction results.

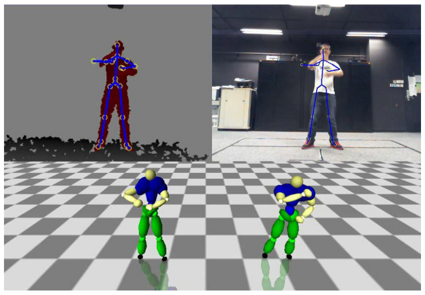
5.2 Qualitative Analysis

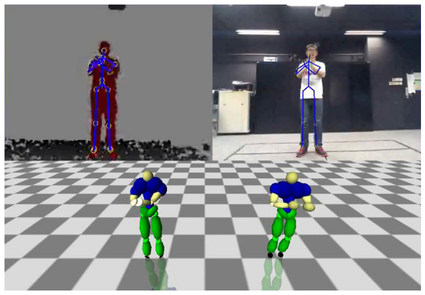
In this section, we evaluate the perceptual accuracy of the postures reconstructed by our method, postures from Kinect, postures by the method proposed by [Shen et al. 2012] and postures captured by an optical motion capture system. We use the Kinect SDK to obtain the 3D position of each joint as the Kinect data. [Shen et al. 2012] proposed to correct postures from Kinect with a random forest regression based approach without considering the reliability of each tracked joint from Kinect. The skeleton structure in the optical motion capture system and Kinect are different, and we need to carefully select these joints so that they are with the same skeleton structure definition. In this paper, we use the skeleton definition of Kinect as a reference skeleton structure. We measure the perceptual

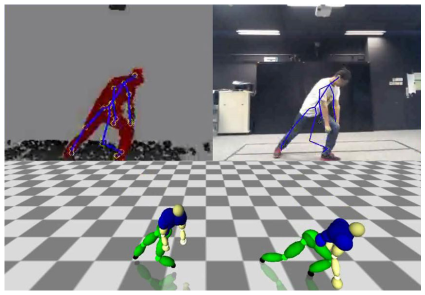

(a)

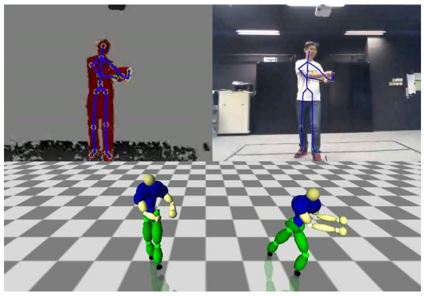

(b)

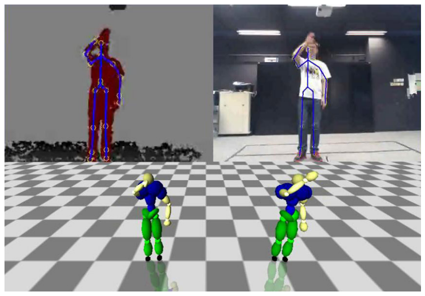

(c)


(d)


(e)


(f)


(g)


(h)

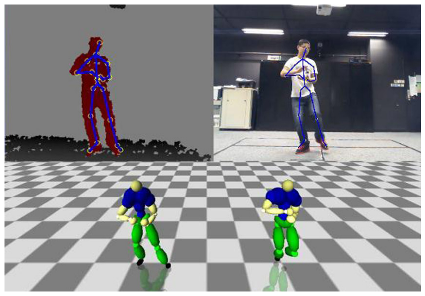

(i)

Figure 4: *Postures from Kinect and their corresponding reconstructed poses. The top two pictures are the depth and RGB image, in which the blue skeleton is the tracked results from Kinect. The left avatar in front represents the posture data from Kinect, and the right avatar in front corresponding to the postures reconstructed by our method. a) Bending over; b) Crossing arms; c) Rolling hands forward and backward; d) Rolling hands up and down; e) Clapping hands; f) Bending leg; g) Golf swinging; h) Waving right hand; i) Taichi motion.*

correctness for the postures of each method with a survey based evaluation [Hoyet et al. 2012], and it is also used in [Shum et al. 2013].

A total of 15 participants were invited to conduct this experiment. All of them had little or no experience about motion capture and 3D animation. The purpose of this experiment is to assess the relative correctness of the obtained postures from these four methods. We create a set of posture sequences with these four methods together with the RGB video so that the participants know what the actual actions are. Participants were asked to give a score for each motion based on its correctness without knowing the displayed motion come from which method. The score ranges from 1 to 10 (inclusive), where 1 means the most incorrect, and 10 means the most correct.

The score distribution for Kinect SDK [Microsoft 2013], [Shen et al. 2012], our method and MOCAP is shown in Figure 5. The overall average scores of these four methods are 5.20, 6.42, 7.51, and 9.16 respectively, and the standard deviations are 0.84, 0.73, 0.71, and 0.57. It is expected that MOCAP data achieve the best scores. However, we can observe from the results that our method greatly outperforms Kinect and [Shen et al. 2012], which verifies our approach can improve the correctness of the postures

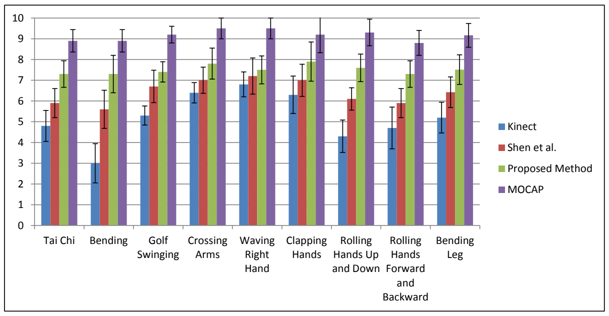


Figure 5: *The score distribution based on the correctness of the postures from Kinect, Shen et al., proposed method and an optical motion capture system.*

output from Kinect sensor. We can see that for these motions with more occlusions, our method performs much better than Kinect and [Shen et al. 2012] such as bending and rolling hands. The reason is that we embed the reliability term into our optimization framework, which implicitly ensures the system tends to recover these joints more than other joints with higher reliability.

5.3 Quantitative Analysis

In this section, we quantitatively analyze the correctness of the proposed method. We can capture the postures with both Kinect and the optical motion capture system at the same time. We assume the data from optical motion capture system is the ground truth data. To evaluate the accuracy of the reconstructed postures, we define an error function to measure the distance between reconstructed postures and ground truth postures:

$$E(F_1, F_2) = \frac{1}{J}\sum_{j=1}^{J} E_j(F_1, F_2) \tag{8}$$

where F_1 and F_2 are the two sets of postures, J is the total number of joints. E_j is the reconstruction error of joint j between two set of postures, which is defined as:

$$E_j(F_1, F_2) = \sum_{t=1}^{T} D(F_{1t}^j, F_{2t}^j) \tag{9}$$

where T is the total number of postures and F_{1t}^j is the j-th joint of the posture at time t from F_1 . D is the Euclidean distance between two joints of two postures:

$$D(P_1^j, P_2^j) = \sqrt{(P_{1x}^j - P_{2x}^j)^2 + (P_{1y}^j - P_{2y}^j)^2 + (P_{1z}^j - P_{2z}^j)^2} \tag{10}$$

With the error function defined in Equation 8, we compare the Kinect raw data, postures reconstructed by [Shen et al. 2012], and postures reconstructed by our method with MOCAP data (unit: cm). Here we choose 5 types of motion for evaluation: clapping hands, crossing arms, bending, Tai Chi, and waving right hand. The results are shown in Table 2.

Table 2: *Reconstruction errors of Kinect, Shen et al. and the proposed method.*

Name	Number of Frames	Kinect (cm)	Shen et al. (cm)	Proposed Method (cm)
Crossing Arms	2052	12.5	9.8	7.2
Bending	1835	13.7	9.5	8.4
Tai Chi	2885	14.5	10.2	7.5
Waving Right Hand	1568	12.5	8.8	6.5

As expected, the errors from Kinect were large in general. Our method outperforms [Shen et al. 2012] as we take into account the reliability of each joint as the inaccurately tracked joint will guide the system to incorrectly infer the postures. For all classes of motions, our method consistently outperforms the other two, which verify the effectiveness of the proposed method in terms of reconstruction accuracy.

To better observe the relationship between reconstruction error and reconstructed postures. We use the error function defined in Equation 10 for each joint and plot the reconstruction error across frames. Figure 6 shows the joint errors of two example motions. We can observe that both [Shen et al. 2012] and our method can achieve lower reconstruction errors compared with Kinect SDK. Another observation is that the trend of the error curves of [Shen et al. 2012] and our work tend to be similar, because both of these two works tried to reconstruct postures output from Kinect.

5.4 Performance Analysis

In this section, we analyze the reconstruction accuracy by examining the effectiveness of different terms in the objective function of Equation 7, respectively. The temporal prior term E_T ensures the temporal consistency between successive frames, which ensures the temporal smoothness between frames. The reliability term E_R encodes the reliability of each joint, which implicitly ensures the reconstructed posture resembles the Kinect observation as closely as possible. We computed the reconstruction error by dropping off the temporal term E_T and the reliability term respectively. We used Tai Chi motion, bending over and crossing arms for evaluation because of their complicated movement features. The results are reported in Table 3. We found

Table 3: *Reconstruction error of the proposed framework with different constraint terms.*

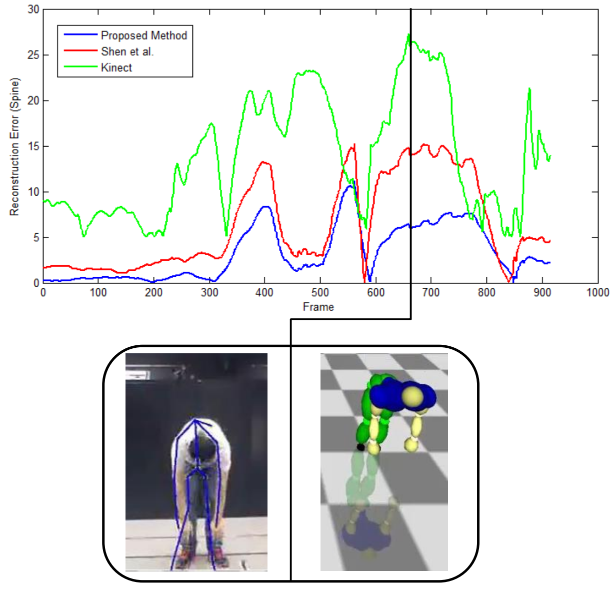
Setup	Terms Used	Reconstruction Error (cm)
(a)	E_S	11.7
(b)	E_S, E_T	10.2
(c)	E_S, E_R	9.4
(d)	E_S, E_T, E_R	7.7

that both the temporal prior term and the reliability term improve the reconstruction accuracy, especially for the movements with severe self-occlusions. Although setup (c) achieves better results than setup(b), the obtained movements are jerky, because setup (c) predicts postures independently without considering relationship between consecutive frames.

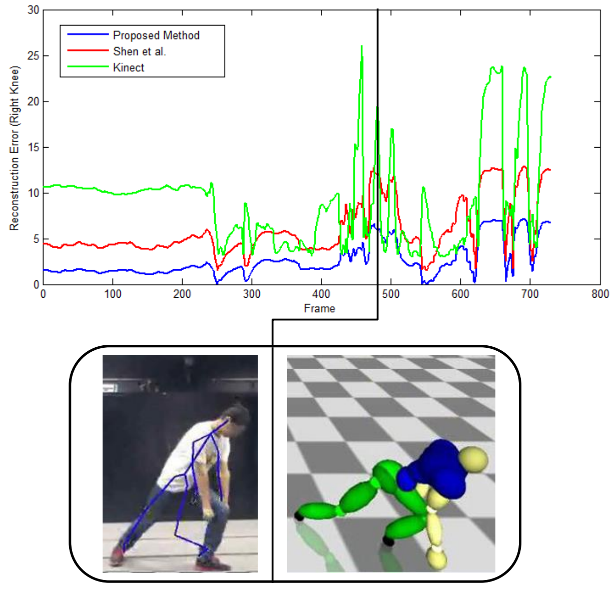
6 Conclusions and Discussions

In this paper, we present a probabilistic framework to reconstruct live captured postures from Kinect. To overcome the incorrectly tracked and missing joints by Kinect, we adopt Gaussian Process (GP) model as a spatial prior to leverage position data obtained from Kinect and an optical motion capture system. Specifically, we model the residual offset between postures obtained from Kinect and mocap system instead of pairwise posture relationship. The residual offset modeling enables the system to cover more of the motion space. GP is advantageous here, in that makes our system work even with small training data sets. We introduce a temporal consistency term into the optimization framework by minimizing the discrepancy between current posture and previous postures. The reliability of the tracked joints from Kinect is different due to self-occlusions. To guide the optimized posture towards the input posture from Kinect, we incorporate the reliability of each tracked joint to constrain the residual offset between posture from Kinect and MOCAP data. Intuitively, the system tends to preserve the joint position value from Kinect as much as possible when the reliability of this joint is high. The experimental results demonstrate that our system can achieve high quality poses even under severe self-occlusion situations and obtain higher accuracy compared with other alternatives. Especially, the system can handle cases even when the motions involve a large number of occluded joints.

There is still room to improve the proposed reconstruction system. The performance of Kinect drops significantly for movement such as extremely fast movement, large object interaction and body rotation. In such cases, the amount of correct data present is so little that our system cannot produce very good result. One future direction would be using multiple Kinects to capture postures



(a)



(b)

Figure 6: *Examples of the reconstruction error of one joint across frames. The blue curve corresponding to our method, red curve is the reconstruction error of Shen et al. and the green curve is the reconstruction error of Kinect. a) Bending over motion; b) Bending leg motion.*

from different directions. The assigned weights for the terms in the objective function are empirically set to be fixed in the proposed system. However, the weighs can be different for different types of motion to obtain optimal reconstructed postures. One possible solution would be to formulate the weight as a function of the residual offset, which is used to measure the importance of each term. Therefore, the weights can be adaptively determined according to the type of motion. The incorporation of physical constraints into the proposed framework is another interesting direction as the reconstructed postures in this work are not physically plausible. One possible way would be modeling the physical attributes (i.e. force field) between the Kinect data and mocap data as a prior distribution, and embed it to the optimization framework to generate physically valid postures. Last but not least, integrating our system with other simple yet stable devices such as motion sensor would be an interesting topic, because Kinect can only detect limited range of movements while motion sensor can be used as a complement, e.g. the user's back side.

Acknowledgements

We thank all the anonymous reviewers for their insightful comments and constructive suggestions. The work described in this paper was supported by a grant from City University of Hong Kong (Project No. 7004045).

References

BAAK, A., MULLER, M., BHARAJ, G., SEIDEL, H.-P., AND THEOBALT, C. 2011. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 1092–1099.

BAILEY, S. W., AND BODENHEIMER, B. 2012. A comparison of motion capture data recorded from a vicon system and a microsoft kinect sensor. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '12, 121–121.

CHAI, J., AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 686–696.

CHAN, J., LEUNG, H., TANG, J., AND KOMURA, T. 2011. A virtual reality dance training system using motion capture technology. *Learning Technologies, IEEE Transactions on 4*, 2 (April), 187–195.

HAN, J., SHAO, L., XU, D., AND SHOTTON, J. 2013. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on 43*, 5 (Oct), 1318–1334.

HELTEN, T., MULLER, M., SEIDEL, H.-P., AND THEOBALT, C. 2013. Real-time body tracking with one depth camera and inertial sensors. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 1105–1112.

HOYET, L., MCDONNELL, R., AND O'SULLIVAN, C. 2012. Push it real: Perceiving causality in virtual interactions. *ACM Trans. Graph. 31*, 4 (July), 90:1–90:9.

IASON OIKONOMIDIS, N. K., AND ARGYROS, A. 2011. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, BMVA Press, 101.1–101.11.

IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., AND FITZGIBBON, A. 2011. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 559–568.

KIM, J., SEOL, Y., AND LEE, J. 2013. Human motion reconstruction from sparse 3d motion sensors using kernel cca-based regression. *Computer Animation and Virtual Worlds 24*, 6.

LAWRENCE, N. 2009. Gaussian process library, http://www.cs.manchester.ac.uk/neill/gp/.

LIU, H., WEI, X., CHAI, J., HA, I., AND RHEE, T. 2011. Realtime human motion control with a small number of inertial sensors. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, 133–140.

MICROSOFT, C. 2013. Kinect for windows sdk programming guide version 1.8.

MORGAN, T., JARRELL, D., AND VANCE, J. 2014. Poster: Rapid development of natural user interaction using kinect sensors and vrpn. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, 163–164.

QUIÑONERO CANDELA, J., AND RASMUSSEN, C. E. 2005. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res. 6* (Dec.), 1939–1959.

RASMUSSEN, C. E., AND WILLIAMS, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

SHEN, W., DENG, K., BAI, X., LEYVAND, T., GUO, B., AND TU, Z. 2012. Exemplar-based human action pose correction and tagging. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 1784–1791.

SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. 2011. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 1297–1304.

SHUM, H., AND HO, E. S. 2012. Real-time physical modelling of character movements with microsoft kinect. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, VRST '12, 17–24.

SHUM, H. P. H., HO, E. S. L., JIANG, Y., AND TAKAGI, S. 2013. Real-time posture reconstruction for microsoft kinect. *IEEE Transactions on Cybernetics 43*, 5, 1357–1369.

SIDENBLADH, H., AND BLACK, M. 2001. Learning image statistics for bayesian tracking. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, 709–716 vol.2.

SIGALAS, M., PATERAKI, M., OIKONOMIDIS, I., AND TRAHANIAS, P. 2013. Robust model-based 3d torso pose estimation in rgb-d sequences. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 315–322.

TASHEV, I. 2013. Kinect development kit: A toolkit for gesture- and speech-based human-machine interaction [best of the web]. *Signal Processing Magazine, IEEE 30*, 5 (Sept), 129–131.

TAYLOR, G., SIGAL, L., FLEET, D., AND HINTON, G. 2010. Dynamical binary latent variable models for 3d human pose tracking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 631–638.

VONDRAK, M., SIGAL, L., AND JENKINS, O. 2013. Dynamical simulation priors for human motion tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35*, 1 (Jan), 52–65.

WEI, X., ZHANG, P., AND CHAI, J. 2012. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph. 31*, 6 (Nov.), 188:1–188:12.

YASIN, H., KRÜGER, B., AND WEBER, A. 2013. Model based full body human motion reconstruction from video data. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, MIRAGE '13, 1:1–1:8.