Leveraging Depth Cameras and Wearable Pressure Sensors for Full-body Kinematics and Dynamics Capture

Peizhao Zhang*[1] Kristin Siu†[2] Jianjie Zhang*[1] C. Karen Liu†[2] Jinxiang Chai*[1]

[1]Texas A&M University
[2]Georgia Institute of Technology

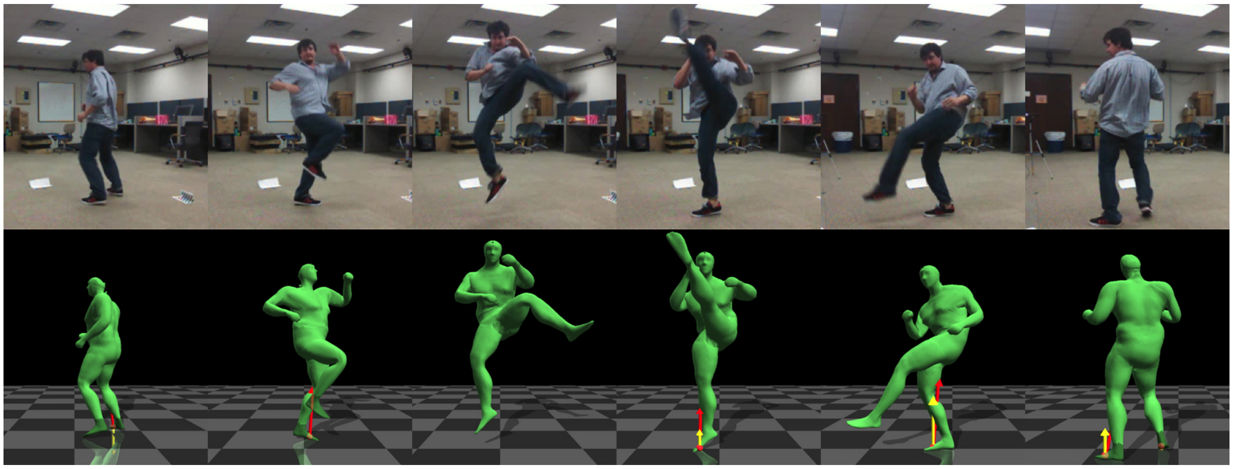

Figure 1: *Our system automatically and accurately reconstructs full-body kinematics and dynamics data using input data captured by three depth cameras and a pair of pressure-sensing shoes. (top) reference image data; (bottom) the reconstructed full-body poses and contact forces (red arrows) and torsional torques (yellow arrows) applied at the center of pressure.*

Abstract

We present a new method for full-body motion capture that uses input data captured by three depth cameras and a pair of pressure-sensing shoes. Our system is appealing because it is low-cost, non-intrusive and fully automatic, and can accurately reconstruct both full-body kinematics and dynamics data. We first introduce a novel tracking process that automatically reconstructs 3D skeletal poses using input data captured by three *Kinect* cameras and wearable pressure sensors. We formulate the problem in an optimization framework and incrementally update 3D skeletal poses with observed depth data and pressure data via iterative linear solvers. The system is highly accurate because we integrate depth data from multiple depth cameras, foot pressure data, detailed full-body geometry, and environmental contact constraints into a unified framework. In addition, we develop an efficient physics-based motion reconstruction algorithm for solving internal joint torques and contact forces in the quadratic programming framework. During reconstruction, we leverage Newtonian physics, friction cone constraints, contact pressure information, and 3D kinematic poses obtained from the kinematic tracking process to reconstruct full-body dynamics data. We demonstrate the power of our approach by capturing a wide range of human movements and achieve state-of-the-art accuracy in our comparison against alternative systems.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: motion capture, human body tracking, physics-based modeling, full body shape modeling

Links: DL PDF WEB VIDEO

1 Introduction

Motion capture technologies have revolutionized computer animation over the past decade. With detailed motion data and editing algorithms, we can directly transfer the expressive performance of a real person to a virtual character, interpolate existing data to produce new sequences, or compose simple motion clips to create a repertoire of motor skills. With appropriate computational models and machine learning algorithms, we can use motion data to create more accurate and realistic models than those based on physics laws and principles alone. Additionally, kinematic and dynamic information of human motion are extremely valuable to a wide variety of fields such as biomechanics, robotics, and health, where there continues to be a growing need for efficient, high-quality, and affordable motion capture systems.

Yet despite decades of research in computer graphics and a plethora of approaches, many existing motion capture systems still suffer from several limitations. Firstly, many systems require the subject

*e-mail: {stzpz, jjzhang10, jchai}@cse.tamu.edu
†e-mail:{kasiu, karenliu}@gatech.edu

ACM Reference Format
Zhang, P., Siu, K., Zhang, J., Liu, C., Chai, J. 2014. Leveraging Depth Cameras and Wearable Pressure Sensors for Full-body Kinematics and Dynamics Capture. ACM Trans. Graph. 33, 6, Article 221 (November 2014), 14 pages. DOI = 10.1145/2661229.2661286 http://doi.acm.org/10.1145/2661229.2661286.

DOI: http://doi.acm.org/10.1145/2661229.2661286

to wear cumbersome devices, or limit the subject's motion to a restricted area. Additionally, in order to capture high-fidelity data, the specialized hardware for these systems is often expensive and requires extensive training to operate. Finally, current motion capture technology specializes in capturing only kinematic information of the movement, rather than its underlying dynamics. Combining multiple kinds of sensory technologies in order to acquire this dynamic information is common practice in the fields of biomechanics and kinesiology. However, this data acquisition process typically involves expensive and intrusive optical motion capture systems and unmovable force platforms that can only be operated in a highly restricted environment.

Advancements in hardware technology have permitted sensory devices to become smaller and cheaper. With the advent of affordable depth cameras, image-based motion capture systems hold promise, but are still limited in the kinds of motions that can be captured. In order to find a solution to these shortcomings, we are inspired by trends in health technology, where the ubiquity of small sensors have made it possible to collect various types of data about human subjects unobtrusively. If combining small and affordable sensors has the potential to provide a powerful amount of information, the question then becomes: What is the ideal set of basic sensors required to capture both high-quality kinematic and dynamic data?

Our answer is a system consisting of a pair of low-cost, non-intrusive pressure-sensing shoes and three *Microsoft Kinect* cameras. Our solution leverages the fact that both of these two sensory technologies are inexpensive and non-intrusive. Additionally, they are complementary to each other as they capture fundamentally-different aspects of the motions. The pressure-sensing shoes provide high resolution contact timing and location information that is difficult to derive automatically from computer vision algorithms. On the other hand, depth data from the *Kinect* cameras provide kinematic information which can filter out noise in the pressure sensors, and provide global position and orientation necessary to estimate dynamic quantities such as the center of pressure. The result is that our system is easy to set up and can be used to acquire motions difficult to capture in restrictive lab settings, such as highly dynamics motions that require a large amount of space.

Our unified system integrates depth data from multiple cameras, foot pressure data, detailed full-body geometry, and environmental contact constraints. We first introduce a novel tracking process that automatically reconstructs 3D skeletal poses using input data captured by the *Kinect* cameras and pressure sensors. We formulate this problem in an optimization framework and incrementally update 3D skeletal poses with observed input data via iterative system solvers. In addition, we develop an efficient physics-based motion optimization algorithm to reconstruct full-body dynamics data, internal joint torques, and contact forces across the entire sequence. We leverage Newtonian physics, contact pressure information, and 3D kinematic poses obtained from the kinematic pose tracking process in a quadratic programming framework. By accounting for physical constraints and observed depth and pressure data simultaneously, we are ultimately able to compute both kinematic and dynamic variables more accurately.

We demonstrate our system by capturing high-quality kinematics and dynamics data for a wide range of human movements. We assess the quality of reconstructed motions by comparing them with ground truth data simultaneously captured with a full marker set in a commercial motion capture system [Vicon Systems 2014]. We show the superior performance of our system by comparing against alternative methods, including [Wei et al. 2012], [Microsoft Kinect API for Windows 2014] and full-body pose tracking using Iterative Closest Point (ICP) method (*e.g.*, [Knoop et al. 2006; Grest et al. 2007]). In addition, we evaluate the importance of each key component of our 3D motion capture system by dropping off each component in evaluation. Finally, we validate the quality of reconstructed dynamics data by comparing joint torque patterns obtained by our system against those from a *Vicon* system and force plates.

In summary, this paper makes the following contributions:

- The first system to use multiple cameras and a pair of pressure-sensing shoes for accurately reconstructing both full-body kinematics and dynamics.
- The use of a signed distance field for full-body kinematic tracking.
- The idea of incorporating depth data, pressure data, full-body geometry and environmental contact constraints into a unified framework for kinematic pose tracking.

2 Background

Various technologies have been proposed for acquiring human body movement. We use a combination of low-cost, portable devices to design a new motion capture system that automatically acquires and reconstructs full-body poses, joint torques, and contact forces all at once. To our knowledge, no single existing motion capture technology can achieve this goal. In the following section, we compare our system with existing motion capture systems popular for both research and commercial use.

One appealing solution for full-body motion capture is to use commercially available motion capture systems, including marker-based motion capture (*e.g.*, [Vicon Systems 2014]), inertial motion capture (*e.g.*, [Xsens 2014]), and magnetic motion capture (*e.g.*, [Ascension 2014]). These methods can capture full-body kinematic motion data with high accuracy and reliability. However, they are often cumbersome, expensive and intrusive. Our entire system does not require the subject to wear special suits, sensors or markers except for a pair of normal shoes. This allows us to capture performance or activities such as sports in their most natural states. More importantly, we aim for much cheaper and more accurate motion with both kinematic and dynamic information.

Image-based systems, which track 3D human poses using conventional intensity/color cameras (for more details, we refer the reader to [Moeslund et al. 2006]), offer an appealing alternative to full-body motion capture because they require no markers, no sensors, no special suits and thereby do not impede the subject's ability to perform the motion. One notable solution is to perform sequential pose tracking based on 2D image measurements (e.g., [Bregler et al. 2004]), which initializes 3D human poses at the starting frame and sequentially updates 3D poses by minimizing the inconsistency between the hypothesized poses and observed measurements. This approach, however, is often vulnerable to occlusions, cloth deformation, illumination changes, and a lack of discernible features on the human body because 2D image measurements are often not sufficient to determine high-dimensional 3D human movement.

One way to reduce the reconstruction ambiguity is to use multiple color cameras to capture full-body performances [Vlasic et al. 2008; de Aguiar et al. 2008]. Another possibility is to learn kinematic motion priors from pre-captured motion data, using generative approaches (*e.g.*, [Pavlović et al. 2000; Urtasun et al. 2005]) or discriminative models (*e.g.*, [Rosales and Sclaroff 2000; Elgammal and Lee 2004]). While the use of learned kinematic models clearly reduces ambiguities in pose estimation and tracking, the 3D motions estimated by these methods are often physically implausible, therefore displaying unpleasant visual artifacts such as out-of-plane rotation, foot sliding, ground penetration, and motion jerkiness.

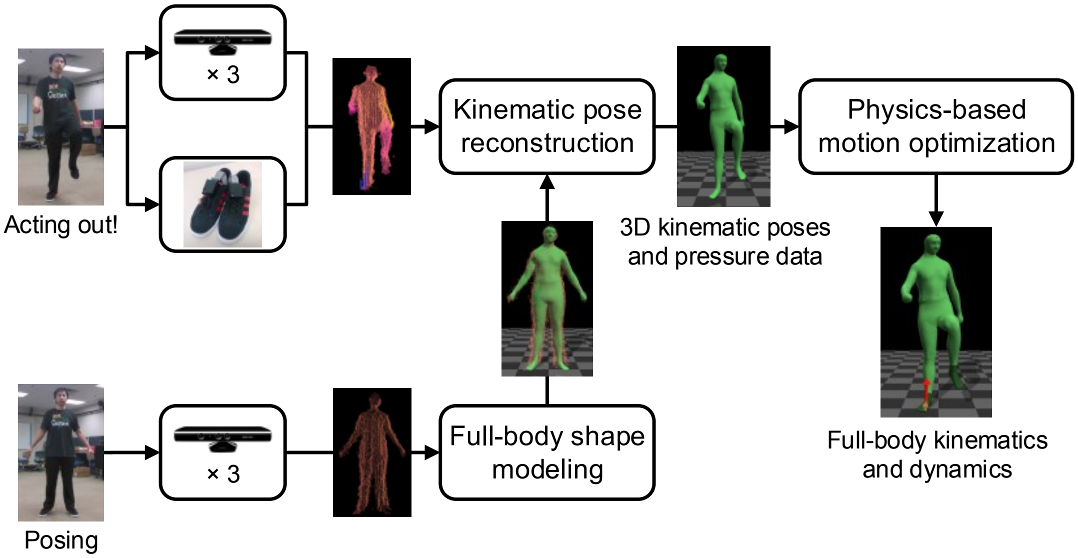
× 3
Kinematic pose reconstruction
Physics-based motion optimization
Acting out!
3D kinematic poses and pressure data
Full-body kinematics and dynamics
× 3
Full-body shape modeling
Posing

Figure 2: *System overview.*

Our work is closely related to a rapidly growing body of recent literature on 3D pose tracking and detection with depth data (*e.g.*, [Plagemann et al. 2010; Shotton et al. 2011; Baak et al. 2011; Ye et al. 2011; Wei et al. 2012]). These approaches are appealing for human motion capture because current commercial depth cameras are low-cost and can record per-pixel 3D depth information at a high frame rate. However, the use of a single depth camera for online motion capture often produces poor results due to sensor noise and inference ambiguity caused by significant occlusions. Among these, our work is most comparable to [Wei et al. 2012]. Both systems build upon the full-body motion tracking process that sequentially updates 3D skeletal poses using observed depth image data. However, our kinematic motion tracking process produces much more accurate results because we integrate depth data from three *Kinect* cameras, foot pressure information, detailed full-body geometry, and environmental contact constraints to reconstruct the full-body kinematic poses. In addition, our goal differs that we aim to reconstruct both full-body kinematics and dynamics data while their work is focused only on 3D kinematic pose reconstruction.

Our idea of leveraging Newtonian physics, contact pressure information, and depth image data to reconstruct kinematic and dynamic information is motivated by recent efforts in combining physical constraints and image data for human motion tracking (*e.g.*, [Brubaker and Fleet 2008; Vondrak et al. 2008; Wei and Chai 2010; Vondrak et al. 2012]). The use of physics for human motion tracking has been shown to be effective for tracking 2D low-body walking motion [Brubaker and Fleet 2008] or normal walking and jogging motion [Vondrak et al. 2008] in a recursive Bayesian tracking framework. Notably, Vondrak and his colleagues [2012] proposed a video-based motion capture framework that optimizes both the control structure and parameters to best match the resulting simulated motion with input observation data. Our system is different because we optimize kinematic poses, internal joint torques, and contact forces based on observed data. In addition, our idea of combining depth images and pressure data significantly reduces the ambiguity of physics-based motion modeling.

In computer animation, pressure and contact information has been used to reconstruct and synthesize human motion using devices such pressure-sensing mats [Yin and Pai 2003] and Wii balance boards [Ha et al. 2011]. Unfortunately, these systems do not permit the capture of highly dynamic motions, due to the static restrictions of the pressure-sensing hardware. Meanwhile, in biomechanics and health technology, there are a number of systems that have been used to acquire dynamic information, such as the center of pressure [Adelsberger and Tröster 2013], which embed sensor technology directly into wireless footwear. However, the novelty of our system is that instead of collecting these dynamic values for separate analysis, we use this information immediately to assist in our kinematic and dynamic reconstruction.

Among all existing systems, our work is most similar to Wei and Chai [2010], where a physics-based model was applied for reconstructing physically-realistic motion from monocular video sequences. Both systems aim to reconstruct full-body kinematic poses, internal joint torques, and contact forces across the entire motion sequence. However, their system relies on manual specification of pose keyframes, and intermittent 2D pose tracking in the image plane, to define the objective for the optimization. In addition, they rely on manual specification of contacts or foot placement constraints to reduce the ambiguity of physics-based motion modeling. By contrast, our system is fully automatic. We also complement depth image data with pressure sensor data to obtain more accurate kinematic and dynamic information.

3 Overview

Our full-body kinematics and dynamics data acquisition framework automatically reconstructs 3D body shape, 3D kinematic poses, internal joint torques, and contact forces as well as contact locations and timings using three *Kinect* cameras and a pair of pressure-sensing shoes. The algorithm consists of three main components summarized as follows (see Figure 2):

Kinematic pose tracking. We introduce a novel tracking process that sequentially reconstructs 3D skeletal poses over time using input data captured by the *Kinect* cameras and wearable pressure sensors. We formulate an optimization problem which minimizes the inconsistency between the reconstructed poses and the observed depth and pressure data. We propose a new metric (*signed distance field* term) to evaluate how well the reconstructed poses match the observed depth data. The results are highly accurate because our system leverages depth data from multiple cameras, foot pressure data, detailed full-body geometry, and environmental contact con-

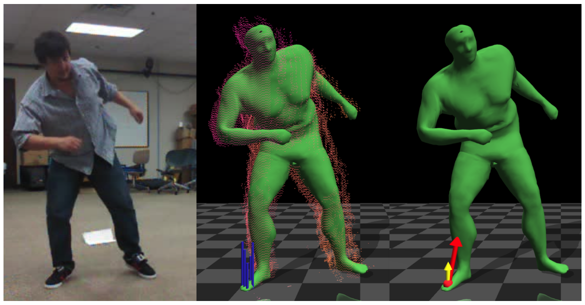

Figure 3: *Full-body kinematic and dynamics data acquisition. (left) reference image; (middle) the reconstructed 3D kinematic pose superimposed on observed depth and pressure data (blue lines); (right) the reconstructed pose, contact force (red arrows) and torsional torque (yellow arrows) applied at the center of pressure (red spheres).*

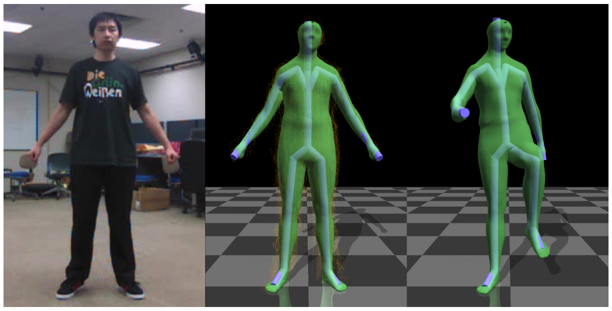

Figure 4: *Full-body shape modeling. (left) "A"-pose of the subject; (middle) the subject's 3D body shape reconstructed from observed depth data; (right) the reconstructed body shape under a new pose obtained from our motion acquisition system.*

straints. Figure 3 (middle) shows the reconstructed kinematic pose that matches both observed depth data and foot pressure data.

Physics-based motion optimization. Acquiring full-body dynamics data requires computing both contact forces and joint torques across the entire motion sequence. To achieve this goal, we introduce an efficient physics-based motion reconstruction algorithm that solves contact forces and joint torques as a quadratic programming problem. During reconstruction, we leverage Newtonian physics, friction cone constraints, contact pressure information, and 3D kinematic poses to reconstruct contact forces and joint torques over time. Figure 3 (right) shows the reconstructed contact forces and torsional torques applied at the center of pressure.

Full-body shape modeling. Reconstructing the body shape of the subject is important to our task because our kinematic tracking process relies on the full-body geometry to measure how well the reconstructed skeletal poses match the observed depth data. Furthermore, incorporating physical constraints into the reconstruction process requires the shape of the human subject to estimate the moment of inertia of each body segment. To address this challenge, we automatically construct a skinned full-body mesh model from the depth data obtained by three *Kinect* cameras so that the full-body mesh model can be deformed according to pose changes of an underlying articulated skeleton using Skeleton Subspace Deformation (SSD). Each user needs to perform the shape modeling step only once (Figure 4).

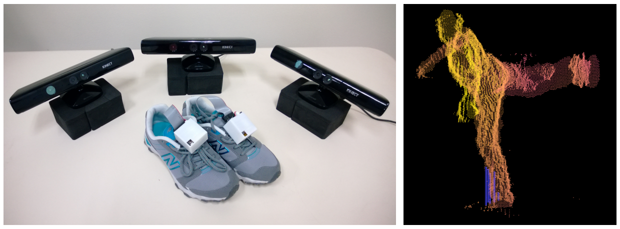

Figure 5: *Data acquisition. (left) three* Kinect *cameras and a pair of pressure-sensing shoes; (right) input data to our motion capture system includes the point cloud obtained by three cameras and pressure data (blue lines) recorded by pressure sensors.*

4 Data Acquisition and Preprocessing

Our system captures full-body kinematics and dynamics data using three synchronized depth cameras and a pair of pressure-sensing shoes (Figure 5). In our experiment, *Kinect* cameras are used for motion capture but other commercially available depth cameras could be used as well.

Depth data acquisition. Current commercial depth cameras are often low-cost and can record 3D depth data at a high frame rate. A number of options exist; our system uses three *Microsoft Kinect* cameras, which cost roughly around five hundred dollars. Each camera returns 320 by 240 depth images at 30 frames per second (fps) with a depth resolution of a few centimeters. The three cameras are arranged uniformly in a circle with a radius of about 3 *m*, pointing to the center of the circle. The camera height is about 1 *m*. We found that this camera configuration yields the best trade-off between capture volume and depth data accuracy. We also found that in this configuration, interference of structured lights between cameras is not a major issue because each camera receives very little infrared (IR) light from other cameras. Most of the IR light is reflected back by the subject and most of the remaining IR light does not reach other cameras due to the large angle (120 degrees) and large distance (2.6 *m*) between cameras.

Pressure data acquisition. The subject wears a pair of shoes during data acquisition. The insole of each shoe is equipped with eight highly accurate Tekscan [2014] Flexiforce® sensors (the accuracy is linear within ±3% of full scale) that correspond to eight points on the feet as shown in Figure 6. These sensors act as force-sensing resistors which are connected to a small microprocessor board enclosed and attached to the top of the shoe. Data is transmitted via a wireless Bluetooth connection at 120 *fps*.

Data synchronization. We connect each depth camera to a different computer and connect the pressure shoes to one of them. We synchronize each computer's system time using Network Time Protocol (NTP). Data from different devices is synchronized by aligning timestamps to the timeline of the first camera. The Network Time Protocol provides very high accuracy synchronization in the local network, usually 5 – 10 *ms* in our experiments. This accuracy is sufficient for synchronization between *Kinect* sensors since the time interval between *Kinect* frames is about 33.3 *ms*. The pressure-sensing shoes are running at a much higher frame rate (120 *fps*), hence picking the frame with the closest timestamp for alignment usually gives satisfactory results.

Depth camera calibration. Reconstructing 3D body poses using multiple depth cameras requires computing the relative positions and orientations of each depth camera. For depth camera calibration, we use a large calibration box to find the rigid transformations between three cameras by aligning visible faces of the box

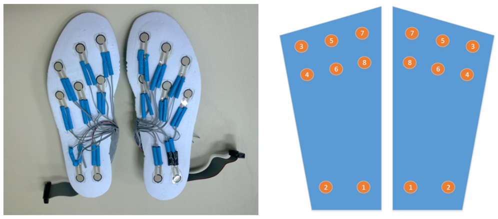

Figure 6: *Insoles of our pressure-sensing shoes. (left) Tekscan [2014] Flexiforce® pressure sensors on the insole of the shoes; (right) corresponding assignments for the sensors.*

and their intersection points. The calibration box is color coded so that each face/plane can be easily identified in RGBD images. Briefly, we first detect each plane of the box by using color detection and RANSAC techniques. We then extract the intersection point of three neighboring faces (or two neighboring faces and the ground plane) which are visible to the same camera. We align the intersection points from different cameras based on the known geometry of the calibration box. We move the box around in the scene to get a sufficient number of constraints to solve for the transformation matrices.

Depth data filtering. Given the calibrated camera parameters and the timestamps of each camera, we align the depth data from the three cameras to obtain a point cloud of the subject at each frame using the rigid transformations obtained from the calibration step (see Figure 5 (right)). We introduce a simple yet effective filtering technique to reduce noise in point cloud data. Specifically, we first build a neighbor graph, each node of which represents a point from the point cloud. We connect two nodes if their distance is smaller than a threshold. We obtain the filtered point cloud by extracting the largest connected components from the neighbor graph. This process usually does not discard noisy points close to the body, but we have found that these points do not affect the accuracy of our full-body tracking process. Combining depth data with pressure data for kinematics and dynamics data capture also requires enforcing ground contact constraints. To this end, we extract the 3D ground plane by applying RANSAC technique [Fischler and Bolles 1981] to the observed depth data.

Full-body pose representation. We use a skinned mesh model to approximate full-body geometry of human subjects (see Section 7). This mesh is driven by an articulated skeleton model using Skeleton Subspace Deformation (SSD). The skinned mesh model contains 6449 vertices and 12894 faces; and our skeleton model contains 24 bone segments. We describe a full-body pose using a set of independent joint coordinates $\mathbf{q} \in R^{36}$, including absolute root position and orientation as well as the relative joint angles of individual joints. These bones are head (1 Dof), neck (2 Dof), lower back (3 Dof), and left and right shoulders (2 Dof), arms (3 Dof), forearms (1 Dof), upper legs (3 Dof), lower legs (1 Dof), and feet (2 Dof).

5 Kinematic Pose Tracking

We now describe our kinematic pose tracking algorithm that sequentially reconstructs 3D human poses from observed point cloud and pressure sensor data. We formulate the sequential tracking problem in an efficient optimization framework and iteratively register a 3D skinned mesh model with observed data via linear system solvers. In the following section, we explain how to incorporate point cloud, pressure data, full-body geometry, contact constraints and pose priors into our tracking framework.

Let O_i be the point cloud obtained from *Kinect* cameras and S_i be the readings from pressure sensors at the current frame i. We want to estimate from O_i and S_i the skeletal poses $\mathbf{q}_i$ for the current frame given previously reconstructed poses $\mathbf{q}_{i-1},...,\mathbf{q}_{i-M}$. Dropping the index i for notational brevity, we aim to estimate the optimal skeletal poses $\mathbf{q}^*$ that best match observed data O and S.

We estimate the full-body kinematic poses by minimizing an objective function consisting of five terms:

$$\min_{\mathbf{q}} \ \lambda_1 E_{SDF} + \lambda_2 E_{Boundary} + \lambda_3 E_{PD} + \lambda_4 E_{GP} + \lambda_5 E_{Prior}, \qquad (1)$$

where E_{SDF}, $E_{Boundary}$, E_{PD}, E_{GP} and E_{Prior} represent the *signed distance field* term, *boundary* term, *pressure data* term, *ground penetration* term and *prior* term, respectively. The weights $\lambda_1,...,\lambda_5$ control the importance of each term and are experimentally set to 2, 2, 100, 100, and 0.1, respectively. We describe details of each term in the following subsections.

5.1 Signed Distance Field Term

We adopt an analysis-by-synthesis strategy to evaluate how well the hypothesized pose $\mathbf{q}$ matches the observed point cloud O. Specifically, given a hypothesized joint angle pose $\mathbf{q}$, we first apply the corresponding transformation $T_{\mathbf{q}}$ obtained by forward kinematics to each vertex of the skinned mesh model to synthesize 3D geometric model of the human body. Given the calibrated camera parameters, we can further project the posed 3D mesh model onto the image plane and render the hypothesized depth images from each viewpoint. The hypothesized point cloud is formed by aligning the rendered depth images from each viewpoint.

So how can we evaluate the distance between the observed and hypothesized point clouds? This often requires identifying the correspondences between the two sets of depth points. Previous approaches (e.g., [Knoop et al. 2006; Grest et al. 2007]) often apply Iterative Closest Points (ICP) method to find the correspondences between the two data sets. However, ICP techniques often produce poor results for human pose registration (for details, see our evaluation in Section 8.2). To address this challenge, we propose to compute signed distance fields from the two point clouds and register the hypothesized and observed signed distance fields via 3D image registration techniques, thereby avoiding building explicit correspondences between the hypothesized and observed point clouds.

A signed distance field (SDF) [Curless and Levoy 1996] is often represented as a grid sampling of the closest distance to the surface of an object described as a polygonal model. SDFs are widely applied in computer graphics and have been used for collision detection in cloth animation [Bridson et al. 2003], multi-body dynamics [Guendelman et al. 2003], and deformable objects [Fisher and Lin 2001]. In our application, we compute SDFs from the point clouds and apply them to iteratively register the hypothesized joint angle pose $\mathbf{q}$ with the observed point cloud O.

We define the SDF on a $50 \times 50 \times 50$ regular grid in three dimensional space. We define the voxel values of the signed distance field V from a point cloud C as follows:

$$V(\mathbf{p}_i) = f_s(\mathbf{p}_i) \cdot \min_{\mathbf{r} \in C} \|\mathbf{p}_i - \mathbf{r}\|^2, \qquad (2)$$

where $\mathbf{p}_i$ is the coordinates of the center of the ith voxel V^i, $\mathbf{r}$ is a point in the point cloud C, and

$$f_s(\mathbf{p}) = \begin{cases} -1, & \text{if } \mathbf{p} \text{ inside;} \\ 1, & \text{if } \mathbf{p} \text{ outside.} \end{cases} \qquad (3)$$

That is, for a volume V, each voxel V^i represents its smallest signed distance to the point cloud C.

We compute the SDF of the observed point cloud in two steps. We first obtain the value of each voxel by searching the closest points in the point cloud. The sign of the voxel value is determined by projecting the voxel V^i onto each of the depth images and comparing the projected depth value d_{proj} with the corresponding depth value d_o in each of the observed depth images. We set the sign to be negative if $d_{proj} > d_o$ for all three images. The sign is set to be positive if d_o does not exist or $d_{proj} < d_o$ for any image. The SDF of the hypothesized point cloud is computed in a similar way.

Once we compute the SDFs for the hypothesized and observed point clouds, we can use them to evaluate the following term in the objective function:

$$E_{SDF}(\mathbf{q}) = \sum_{i \in S_{SDF}} \|V_R^i(\mathbf{q}) - V_O^i\|^2, \tag{4}$$

where $V_R^i(\mathbf{q})$ is the value of the ith voxel of the hypothesized SDF and it depends on the hypothesized skeletal pose $\mathbf{q}$. V_O^i is the voxel value of the ith voxel of the observed SDF, and S_{SDF} includes the indices of all the voxels used for evaluating E_{SDF}. Note that not all the voxels are included for evaluation. In our implementation, we exclude voxels with zero gradients because they do not contribute to the pose updates. To speed up the tracking system, we also ignore the voxels that are far away from the surface of the rendered skinned mesh model as they provide little guidance on the tracking process.

A major benefit of the signed distance field term is that it merges all the observation information from depth cameras, including both depth and boundary information. This significantly reduces ambiguity for 3D pose reconstruction. In our experiment, we have found that using a coarse resolution SDF is often sufficient for tracking 3D poses since it provides us a large number of constraints, even more than using the point clouds itself, due to the use of information inside and outside the point clouds. Another benefit of the SDF term is that the function $E_{SDF}(\mathbf{q})$ is continuous and smooth, which makes the gradient differentiable everywhere with respect to the hypothesized pose $\mathbf{q}$. This property is particularly appealing to our pose tracking solver because we apply gradient-based optimization to do the pose tracking. As shown in our results, our method produces more accurate results than alternative solutions such as ICP (*e.g.*, [Knoop et al. 2006; Grest et al. 2007]) and model-based depth flow [Wei et al. 2012].

5.2 Boundary Term

In practice, even with ground truth poses, the hypothesized point cloud might not precisely match the observed point cloud due to camera noise, cloth deformation, calibration errors and blurry depth images caused by fast body movements. Therefore, the signed distance field term alone is often not sufficient to produce satisfactory results, particularly when significant occlusions occur. This motivates us to introduce the boundary term to further improve the tracking accuracy.

Intuitively, the boundary term minimizes the size of non-overlapping regions between the hypothesized and observed point clouds. To be specific, we penalize the distances between the hypothesized points $\mathbf{p}(\mathbf{q})$ in the non-overlapping region and their closest points $\mathbf{p}^*$ from the observed point cloud. We have

$$E_{Boundary}(\mathbf{q}) = \sum_{\mathbf{p} \in S_B} \|\mathbf{p}(\mathbf{q}) - \mathbf{p}^*\|^2. \tag{5}$$

A critical issue for the boundary term evaluation is to determine which points in the hypothesized point cloud should be included for evaluation (*i.e.*, S_B). Our evaluation considers all the points in non-overlapping regions of the hypothesized and observed depth images from each camera viewpoint. This ensures that the hypothesized point cloud moves towards the observed point cloud to reduce the size of non-overlapping regions as quickly as possible.

In our implementation, we search the closest points based on a bidirectional distance measurement in order to ensure one-to-one correspondences. For observed depth points in non-overlapping region, we first find the closest points in the hypothesized point cloud. Then for the hypothesized depth points who have multiple correspondences, we pick the one with the largest distance to ensure a one-to-one correspondence. Correspondences for hypothesized depth points are determined similarly.

5.3 Pressure Data Term

Depth data alone is often not sufficient to accurately reconstruct the movement of both feet because the observed depth data is often very noisy. The most visible artifact in the reconstructed motion is footskate, which can be corrected by existing methods if the footplants are annotated [Kovar et al. 2002]. However, footplant constraints are extremely hard to derive from noisy depth image data. To address this challenge, we complement depth data with pressure data obtained from a pair of pressure-sensing shoes. When a pressure sensor is "on", we can enforce the corresponding footplant constraints on pose reconstruction.

Under the assumption that the only contact the feet have is with the ground plane, we define the pressure data term as follows:

$$E_{PD}(\mathbf{q}) = \sum_m b_m dist(\mathbf{p}_m(\mathbf{q}), G_F), \tag{6}$$

where the function $dist$ measures the distance between the global coordinates of the mth pressure sensor $\mathbf{p}_m(\mathbf{q})$ and the 3D ground plane G_F. Here the local coordinates of each pressure sensor are known in advance so that we can apply forward kinematics to map the local coordinates of the mth pressure sensor to its global 3D coordinates $\mathbf{p}_m(\mathbf{q})$ under the current pose $\mathbf{q}$. In our implementation, we use a binary variable b_m to indicate whether the mth pressure sensor is "on". This variable provides a means to exclude erroneous non-zero pressure data that can be received even when airborne. Such readings can occur because the sensors are attached to the insole of the shoe rather than the exterior of the shoe sole.

We adopt a simple yet effective rule to determine if a particular pressure sensor is "on" or "off". At each iteration of kinematic pose optimization, we evaluate whether the pressure sensor is "off" based on the following two criteria: (1) we consider all the pressure sensors from a foot as "off' if the sum of pressure values is smaller than a threshold ε_1 and (2) we consider a particular pressure sensor is "off" if its vertical position in the previous iteration of kinematic pose optimization is above the ground plane and its distance to the ground plane is larger than a threshold ε_2. We experimentally set ε_1 and ε_2 to 0.008 and 0.05 m, respectively.

5.4 Ground Penetration Term

The pressure data term alone often cannot avoid foot-ground penetration. This is because we model each foot using a detailed mesh model and therefore a small number of contact points are often not sufficient to avoid ground penetration. We introduce the ground penetration term to address this issue.

We sample a set of points $n = 1, ...N$ on each foot and prevent them from penetrating into the ground. In particular, we penalize the penetration between the foot and the ground G_F, resulting in the following objective term:

$$E_{GP}(\mathbf{q}) = \sum_n \|f_P(\mathbf{p}_n(\mathbf{q}), G_F)\|^2, \tag{7}$$

$$f_P(\mathbf{p}_n(\mathbf{q}), G_F) = \begin{cases} 0, & \text{if no penetration;} \\ dist(\mathbf{p}_n(\mathbf{q}), G_F), & \text{otherwise.} \end{cases} \tag{8}$$

where $\mathbf{p}_n(\mathbf{q})$ is the global coordinates of the nth contact point on the foot. Like the pressure data term, the function *dist* measures the distance between the global coordinates of the nth contact point $\mathbf{p}_n(\mathbf{q})$ and the 3D ground plane G_F.

5.5 Prior Term

We incorporate the prior term into our tracking process for two reasons. First, the depth data is sometimes ambiguous because of significant occlusions, camera noise, cloth deformation or blurry depth images caused by fast body movements. Second, the reconstructed joint angle poses may violate the joint limits. We utilize subspace pose priors embedded in a highly varied motion capture database to solve this problem.

We construct separate PCA models for the pose of each body part (arms, shoulders, spines, legs and feet). The training data we use is from the CMU mocap database, which includes 4.6 hours of highly varied motions. We use the constructed PCA models to constrain the solution space of kinematic tracking. In our implementation, we enforce the subspace constraints as soft constraints, resulting in the following objective term:

$$E_{Prior}(\mathbf{q}) = \|P_k^T(P_k(\mathbf{q} - \mu)) + \mu - \mathbf{q}\|^2, \tag{9}$$

where P_k is the first k principal components of the PCA model and μ is the mean vector of the PCA model. The numbers of dimension of the PCA models (k) are automatically determined by keeping 95% of original variations.

We have found that enforcing such weak PCA priors allow us to achieve similar results as the joint limit constraints while still enabling us to optimize the pose using iterative linear solvers.

5.6 Kinematic Pose Reconstruction

Solving the objective function described in Equation (1) requires minimizing a sum of squares of non-linear functions. We apply a Gauss-Newton gradient descent optimization algorithm to solve this problem. Given a known, current estimate of $\mathbf{q}$, we iteratively solve for increments to the parameters $\delta\mathbf{q}$ using linear system solvers. Note that our kinematic pose tracking process is fully automatic as we initialize the pose at the first frame using *Microsoft Kinect* for Windows [2014].

For each subsequent time step, we initialize the current pose using the previously estimated pose and iteratively perform the following steps until the change of the pose is smaller than a specified threshold:

- Step 1: Given the current pose $\mathbf{q}$ and the full-body skinned mesh model, we render the depth images $D_R(\mathbf{q})$ from each camera viewpoint. For a point $\mathbf{p} \in R$ in the rendered depth image, we use OpenGL's selection buffer to determine which bone segments the point is associated with as well as the local coordinates of the corresponding surface point. This step is necessary for evaluating the partial derivatives $\partial\mathbf{p}/\partial\mathbf{q}$ because the global coordinates of surface points are dependent on both the local coordinates and associated bone segments.
- Step 2: We compute the hypothesized and observed signed distance fields V_R and V_O based on the point clouds C_R and C_O obtained from the hypothesized and observed depth images $D_R(\mathbf{q})$ and $D_O(\mathbf{q})$ (see Equation (2)).
- Step 3: We calculate the gradients of the hypothesized signed distance field and other partial derivatives in Equations (4), (5), (6), (7) and (9) to form linear equations (for details, see Appendix A).
- Step 4: We compute the optimal increment $\delta\mathbf{q}$ using linear system solvers and update the current pose: $\mathbf{q} = \mathbf{q} + \delta\mathbf{q}$.

The algorithm usually converges within 10 iterations as we initialize the solution using previous reconstructed poses. The output of the kinematic tracking process includes kinematic pose $\mathbf{q}$ at current frame as well as contact states (b_m) and global 3D coordinates ($\mathbf{p}_m(\mathbf{q})$) of each pressure sensor.

6 Physics-based Motion Optimization

In this section, we describe how to reconstruct full-body dynamics data using both observed pressure data and reconstructed kinematic motion data obtained from Section 5. We formulate this as a quadratic programming problem and seek optimal values for internal joint torques and contact forces that best match observed pressure data and reconstructed kinematic poses as well as contact states. Similar to the kinematic tracking process, we solve the full-body dynamics reconstruction process in a sequential manner.

Full-body dynamics. The Newtonian dynamics equations for full-body movement can be defined as follows:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + h(\mathbf{q}) = \mathbf{u} + \mathbf{J}^T\mathbf{f}, \tag{10}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ represent the joint angle poses, velocities and accelerations, respectively. The quantities $M(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $h(\mathbf{q})$ are the joint space inertia matrix, centrifugal/Coriolis and gravitational forces, respectively. The vectors $\mathbf{u}$ and $\mathbf{f}$ are joint torques and contact forces respectively. The contact force Jacobian matrix $\mathbf{J}$ maps joint velocities to world space cartesian velocities at the contact points. Human muscles generate torques about each joint, leaving global position and orientation of the body as unactuated joint coordinates. The movement of global position and orientation is controlled by contact forces $\mathbf{f}$. Modifying those coordinates requires contact forces $\mathbf{f}$ from the environment.

Enforcing Newtonian dynamics constraints requires computing the mass and moment of inertia of each body segment. To achieve this goal, we first reconstruct a full-body skinned mesh model to approximate the whole-body geometry of the subject (see Section 7). We then voxelize the reconstructed skinned mesh model. For each voxel, we compute its geodesic distance to all bone segments and associate it with a particular bone segment that is closest to the voxel. Assuming the weight of the subject is known, we can estimate the density of a subject's body and use it to compute the physical quantities of each bone segment, including mass and moment of inertia.

Friction cone constraints. During ground contact, the feet can only push, not pull on the ground, contact forces should not require an unreasonable amount of friction, and the center of pressure must fall within the support polygon of the feet. We use Coulomb's friction model to compute the forces caused by the friction between the character and environment. A friction cone is defined to be the range of possible forces satisfying Coulomb's function model for an object at rest. We ensure the contact forces stay within a basis that approximate the cone with nonnegative basis coefficients. We model the contact between the foot and ground using eight contact points (see Figure 6), which are consistent with the locations of pressure sensors. This allows us to represent the contact forces $\mathbf{f}$ as a linear function of nonnegative basis coefficients:

$$\mathbf{f}(\mathbf{w}_1,...,\mathbf{w}_8) = \sum_{m=1}^{8} \mathbf{B}_m \mathbf{w}_m \text{ subject to } \mathbf{w}_m \geq \mathbf{0}, \tag{11}$$

where the matrix $\mathbf{B}_m$ is a 3×4 matrix consisting of 4 basis vectors that approximately span the friction cone for the m-th contact force. The 4×1 vector $\mathbf{w}_m$ represents the nonnegative basis weights for the m-th contact force.

Pressure data. Each pressure sensor records an analog resistance reading proportional to the applied pressure, which is then converted to a digital value. The relationship between the analog resistance reading R_m and the digital pressure force value P_m returned is defined as follows:

$$P_m = k_m / R_m, \tag{12}$$

where k_m is a scaling parameter for each sensor and assumed to be unknown.

Full-body dynamics reconstruction. We formulate full-body dynamics reconstruction in a quadratic programming framework. Given observed pressure data R_m and reconstructed kinematic poses $\mathbf{q}$ and contact states b_m obtained from the tracking process, the optimization simultaneously computes joint torques $\mathbf{u}$, contact forces $\mathbf{f}(\mathbf{w})$ and pressure sensors coefficients $\mathbf{k} = [k_1,...,k_8]^T$ that maximize the performance of the following multiobjective function:

$$\begin{array}{c} \arg\min_{\mathbf{u},\mathbf{w},\mathbf{k}} E_{pressure}(\mathbf{w},\mathbf{k}) + \lambda_1 E_{reg}(\mathbf{k}) + \lambda_2 E_{torque}(\mathbf{u}) \\ \text{subject to } \; M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}}) + h(\mathbf{q}) = \mathbf{u} + \mathbf{J}^T\mathbf{f}(\mathbf{w}), \\ \mathbf{w} \geq \mathbf{0}. \end{array} \tag{13}$$

In the above, the first term $E_{pressure}$ evaluates the consistency between the reconstructed contact forces and observed pressure forces. Specifically, the pressure term is defined as follows:

$$E_{pressure} = \sum b_m ||f_{m,\perp} - k_m/R_m||^2, \tag{14}$$

where $f_{m,\perp}$ is the vertical component of the reconstructed contact force at the mth sensor. And R_m and k_m are the reading and scale of the mth pressure sensor.

The second term E_{reg} is a regularization term that ensures the scaling parameters of all the pressure sensors are as close as possible. This is achieved by minimizing the variance of the scale parameters for all the "on" pressure sensors:

$$E_{reg} = \frac{1}{(\sum b_m - 1)} \sum b_m (k_m - \frac{\sum b_m k_m}{\sum b_m})^2. \tag{15}$$

The third term E_{torque} minimizes the sum of squared torques at the current frame. The optimization is also subject to the discretization of Newtonian dynamics equations determined by a finite difference scheme and friction cone constraints $\mathbf{w} \geq \mathbf{0}$.

In our implementation, we use the backward difference approximation to compute joint velocities and use the central difference approximation to compute joint accelerations with δt set to $1/30$ s. We solve the optimization problem using quadratic programming.

7 Full-body Shape Modeling

This section describes how to reconstruct full-body mesh models of human subjects using a small number of depth images captured by three *Kinect* cameras. We model full-body geometry of human subjects as a skinned mesh model. We introduce an efficient full-body shape modeling technique that automatically reconstructs a detailed skinned mesh model of a subject using the depth data obtained from three *Kinect* cameras. Each user needs to perform this step only once. Note that the user should not wear overly loose clothing like skirt for modeling, as it will mislead the system and produce an inaccurate shape model for estimating physical quantities of human bodies.

7.1 Shape Representation

Our human body model is based on statistical analysis of a database of pre-registered 3D full-body scans [Allen et al. 2003]. In particular, we apply PCA to hundreds of aligned body scans [Hasler et al. 2009] to construct a low-dimensional parametric model for human body representation. We represent human body geometry using a mean mesh model $\mathbf{A}$ and a weighted combination of eigen mesh basis P:

$$\mathbf{M}(\mathbf{X}) = P\mathbf{X} + \mathbf{A}, \tag{16}$$

where $\mathbf{M} = [x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_n, y_n, z_n]$ is a long vector stacking all the vertices of the mesh model and $\mathbf{X}$ is the low-dimensional shape parameter to represent a full-body geometric model.

We further build a skinned mesh model for the registered mesh model so that the mesh model can be deformed according to pose changes of an underlying articulated skeleton using Skeleton Subspace Deformation (SSD).

7.2 Shape Reconstruction

To reconstruct a full-body skinned mesh model for the subject, we instruct the user to perform a reference pose ("A" -pose, see Figure 4) for about one second. As a result, we obtain three sequences of depth images. Our goal herein is to reconstruct both full-body poses and full-body geometry from the recorded depth image sequences. We formulate the problem as an optimization and seek to find the optimal shape parameter $\mathbf{X}$ and skeletal pose $\mathbf{q}$ that best fit the observed point cloud C:

$$\mathbf{X}^*, \mathbf{q}^* = \arg\min_{\mathbf{X},\mathbf{q}} \sum_i ||\mathbf{p}_i(P\mathbf{X}+\mathbf{A}) \bigoplus T_{\mathbf{q}} - \mathbf{p}_i^*||^2, \tag{17}$$

where $\mathbf{p}_i(\mathbf{M})$ is 3D coordinates of the ith vertex of the parametric mesh model $\mathbf{M}$ and $\mathbf{p}_i^*$ is the 3D coordinates of the closest point of $\mathbf{p}_i$ in C. The operator $\bigoplus$ applies the corresponding transformation $T_{\mathbf{q}}$ to each vertex of the surface mesh model $\mathbf{p}_i(\mathbf{M})$ to obtain 3D full-body geometric model under the pose $\mathbf{q}$.

We have found that direct optimization of the cost function is not efficient and the optimization is prone to falling into local minima. To address this issue, we introduce an iterative optimization algorithm to decompose the large optimization problem into two smaller problems that can be solved efficiently. We initialize the pose using the "A" -pose. In each iteration, we keep one group of the unknowns unchanged and search for an optimal update for the other group of unknowns.

Non-rigid shape estimation. In this step, we estimate the shape parameter $\mathbf{X}$ from the observed point cloud while keeping the pose $\mathbf{q}^*$ constant. This requires solving the following optimization problem:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_i ||\mathbf{p}_i(P\mathbf{X}+\mathbf{A}) \bigoplus T_{\mathbf{q}^*} - \mathbf{p}_i^*||^2. \tag{18}$$

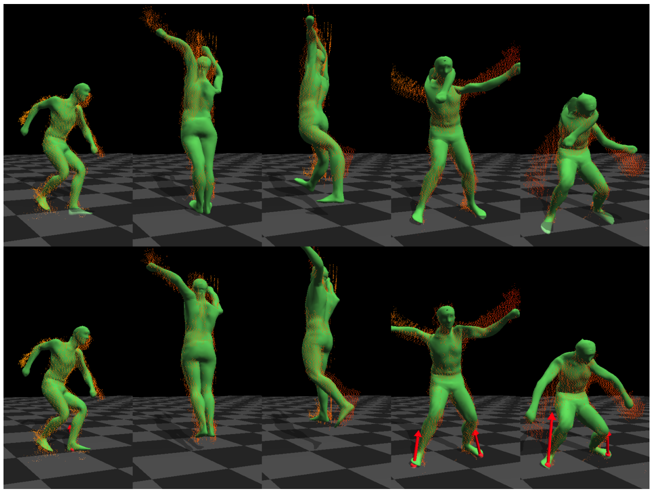

Figure 7: *Comparison against [Wei et al. 2012]. (top) results obtained from [Wei et al 2012]; (bottom) our results.*

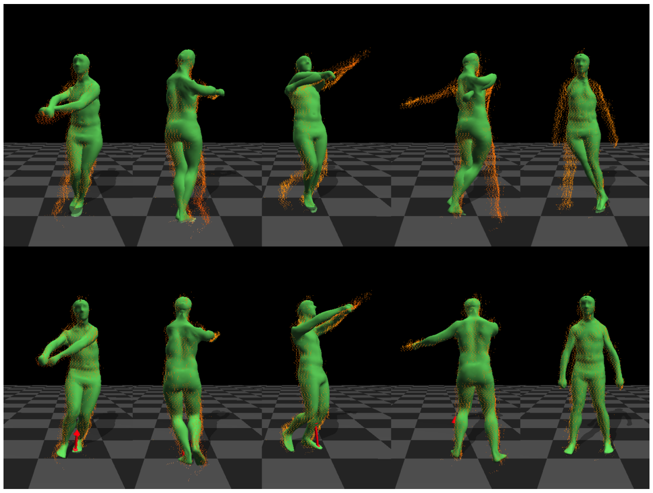

Figure 8: *Comparison against ICP algorithms. (top) results from ICP algorithm; (bottom) our results.*

We extend iterative closest points (ICP) techniques to iteratively estimate the shape parameter $\mathbf{X}$. Briefly, we search the closest points for each vertex of the current mesh model $\mathbf{M}(\mathbf{X})$ on the observed point cloud and use them to update the shape parameter $\mathbf{X}$ with least-square fitting techniques.

Skeletal pose update. We fix the shape parameter $\mathbf{X}$ and use it to update the skeletal pose $\mathbf{q}$ based on the observed point cloud. This problem can be solved efficiently using the kinematic tracking algorithm described in Section 5.

8 Results

In this section, we demonstrate the power and effectiveness of our system by capturing a wide range of human movements using our proposed system (Section 8.1). Our comparison against alternative methods shows the system achieves state-of-the-art accuracy (Section 8.2 and 8.3). We assess the performance of our kinematic tracking process by dropping off each term in the cost function (Section 8.4). We validate the quality of dynamics data obtained from our system by comparing joint torques patterns obtained from our system against those reconstructed from the *Vicon* system and force plates (Section 8.5). Our results are best seen in the accompanying video.

Computational times. For the current implementation, our kinematic tracking and physics-based optimization process run at 6 *fps*. It takes three seconds to complete the offline full-body shape modeling process.

8.1 Test on Real Data

We have tested our system on a wide variety of human actions, including walking, running, jumping, dancing, and sport activities such as basketball, baseball and boxing. The accompanying video shows the performance of our system on a large number of complex and fast motions that a single camera could not capture, such as jumping with a 360 degree rotation and kicking while rotating. We also demonstrate the robustness of our system on several long sequences like boxing, stealing and dancing.

8.2 Comparisons against Alternative Methods

We have evaluated the effectiveness of our kinematic tracking system by comparing against alternative full-body tracking methods. It is worth pointing out that our whole motion capture system can automatically and accurately capture internal joint torques and contact forces, as well as contact locations and timings, across the entire sequence, a capability that has not been demonstrated in alternative tracking systems.

Comparison against [Wei et al. 2012]. We compare our system against the state-of-the-art in full-body motion capture using a single depth camera [Wei et al. 2012]. For a fair comparison, we first extend their tracking algorithm to multiple *Kinect* cameras by combining all the information obtained from three depth cameras. The accompanying video highlights a side-by-side comparison between the two systems. Figure 7 shows the advantage of our system.

Comparison against ICP techniques. We compare our 3D kinematic tracking process described in Section 5 against Iterative Closest Point (ICP) techniques [Knoop et al. 2006; Grest et al. 2007]. Specifically, we apply ICP to minimize the distances between the observed point cloud obtained from three depth cameras and the hypothesized point cloud rendered from the skinned mesh model by iteratively finding the closest correspondences between them. We start both methods with the same initial pose. The accompanying video clearly shows that our tracking process is much more robust and accurate than the ICP algorithm. In the jumping example shown in Figure 8, our tracking process successfully tracks the entire motion sequence while ICP fails to track most of frames. This is because ICP is often very sensitive to initial poses and prone to local minima, particularly when tracking high-dimensional human body poses from noisy depth data.

Comparison against *Vicon* [2014]. In this experiment, we quantitatively assess the quality of the captured motion by comparing against motion data captured with a full marker set in a twelve-camera *Vicon* system [2014]. The average reconstruction error, which is computed as the average 3D joint position discrepancy between the estimated poses and the ground truth mocap poses, is about 3.8 *cm* per joint per frame. Figure 9 shows a side-by-side comparison between our result and the result obtained by *Vicon*.

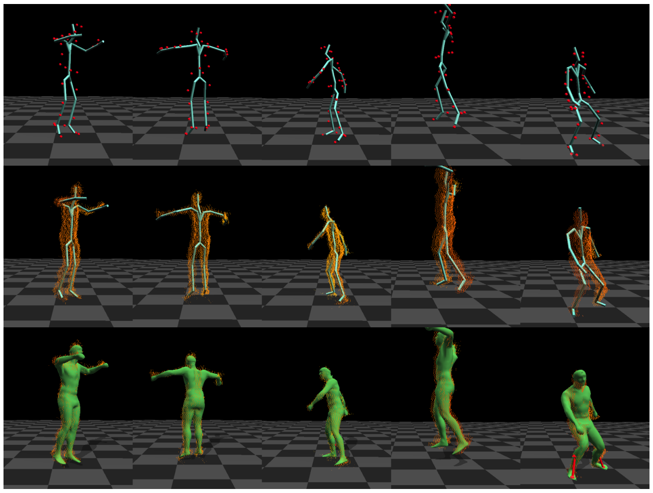

Figure 9: *Comparison against* Vicon *[2013]. (top) results from a twelve-cameras* Vicon *system [2013] with a full set of markers; (middle) our results with a skeleton model; (bottom) our results with a skinned mesh model.*

8.3 Quantitative Evaluation

We quantitatively evaluate the reconstruction accuracy and robustness of our system by comparing against four alternative methods, including our algorithm without pressure data, [Wei et al. 2012], [Microsoft Kinect API for Windows 2014] and ICP, on six different actions. The ground truth data is obtained by motion data captured with a twelve-camera *Vicon* system in a full marker set. For a fair comparison, we include the prior term in all alternative methods except [Microsoft Kinect API for Windows 2014]. For [Microsoft Kinect API for Windows 2014], we obtain separate poses from three depth cameras at each frame and choose the pose closest to the ground truth data as the output.

Reconstruction accuracy evaluation. To evaluate the reconstruction accuracy, we compute average joint position errors and variances for each method by comparing against ground truth poses obtained from the *Vicon* system (Figure 10). The evaluation shows that our system produces a much lower error and variance (3.8 ± 1.3 *cm*) than [Wei et al. 2012] (5.0 ± 2.2 *cm*) and *Kinect* (7.7 ± 2.5 *cm*). Among all the methods, ICP produces largest errors for all the test data. The evaluation also shows that complementing depth data with pressure data improves the accuracy from 4.1 ± 1.3 *cm* to 3.8 ± 1.3 *cm*. Figure 11 compares average reconstruction errors of each joint for our method, [Wei et al. 2012] and *Kinect* system. Our system produces more accurate reconstruction results than two alternative methods for all the joints.

Robustness evaluation. To evaluate the system robustness, we compute the percentage of failure frames for each motion. Here we define a reconstructed frame as "failure" if the average joint position discrepancy is larger than 6 *cm*. Figure 12 shows that our system produces a much lower failure rate (5.9%) than alternative methods (14.9% for [Wei et al. 2012] and 68.3% for *Kinect* [2014]).

8.4 Evaluation of Kinematic Pose Tracking Process

We have evaluated the importance of key components of our kinematic tracking process by dropping off each term in Equation (1).

Importance of the boundary term. We evaluate the importance of the boundary term by comparing the results with and without this term. Figure 13 clearly shows the importance of the boundary term.

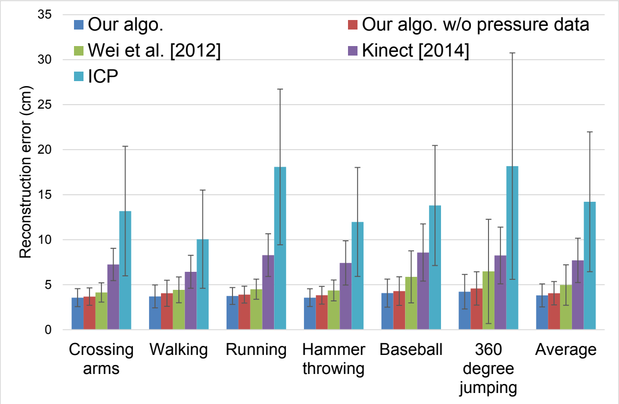

Figure 10: *Evaluation of reconstruction accuracy (average joint position errors and variances) for five methods on six test actions.*

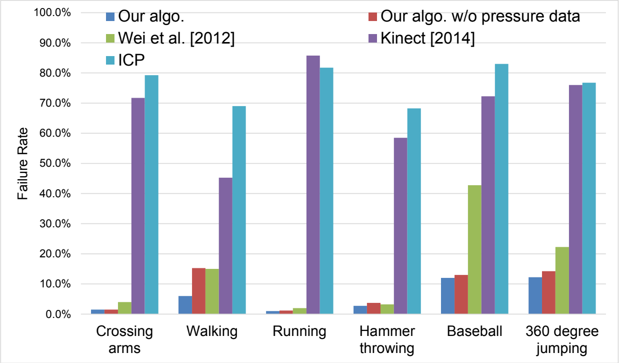

Figure 12: *Evaluation of system robustness (percentage of frames whose average reconstruction error is larger than* 6 *cm) on six test sequences.*

Importance of the pressure data/ground penetration term. Figure 14 shows a side-by-side comparison with and without the pressure data/ground penetration term. The term is critical to our system for two reasons. First, it enables us to remove foot skating artifacts and avoid the ground penetration issue in the reconstructed kinematic motion. Second, it significantly reduces the reconstruction ambiguity of full-body dynamics.

Importance of the prior term. Figure 15 shows a side-by-side comparison with and without the prior term. The use of the prior term improves the reconstruction accuracy of full-body poses, particularly the torso part in this example.

8.5 Comparison against Vicon and Force Plates

We have validated the effectiveness of our dynamic data capture process by comparing the reconstructed internal torques with those obtained from a twelve-camera *Vicon* system in a full marker set and force plates. We capture 120 walking sequences using the *Vicon* system and force plates and reconstruct the internal joint torques based on the recorded force data from force plates and the full-body kinematic motion data obtained from the *Vicon* system via inverse dynamics technique. Figure 16 (a) plots internal joint torques of the left knee from 120 walking sequences (blue curve). We repeat the captured motion five times and extract the joint torque patterns of

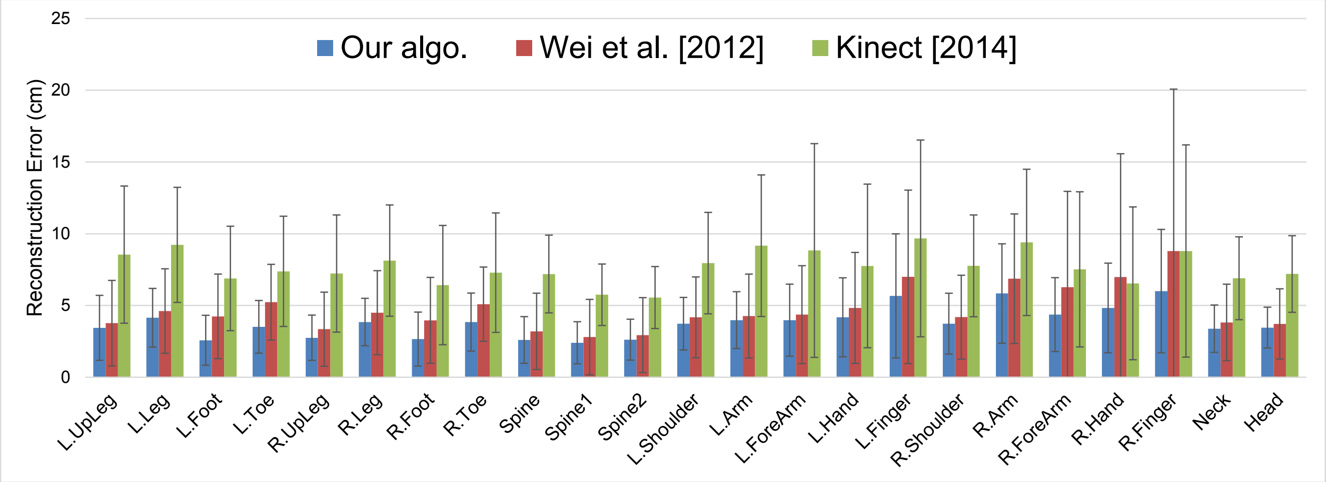

Figure 11: *Average joint reconstruction errors and variances on six action sequences.*

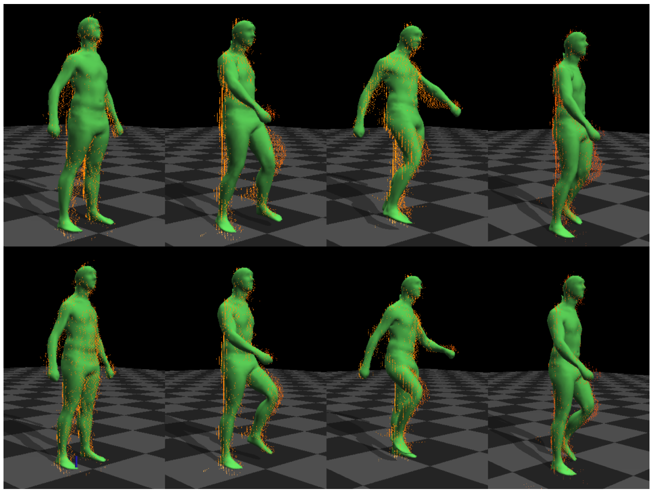

Figure 13: *Importance of the boundary term. (top) result without the boundary term; (bottom) result with the boundary term.*

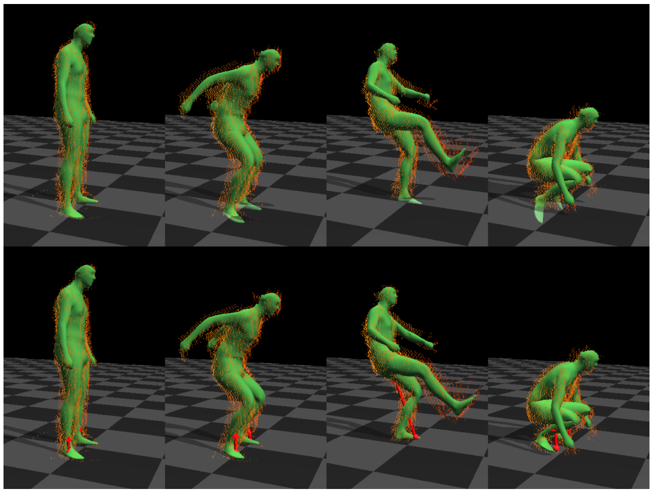

Figure 14: *Importance of the pressure data/ground penetration term. (top) result without the pressure data/ground penetration term; (bottom) result with the pressure data/ground penetration term.*

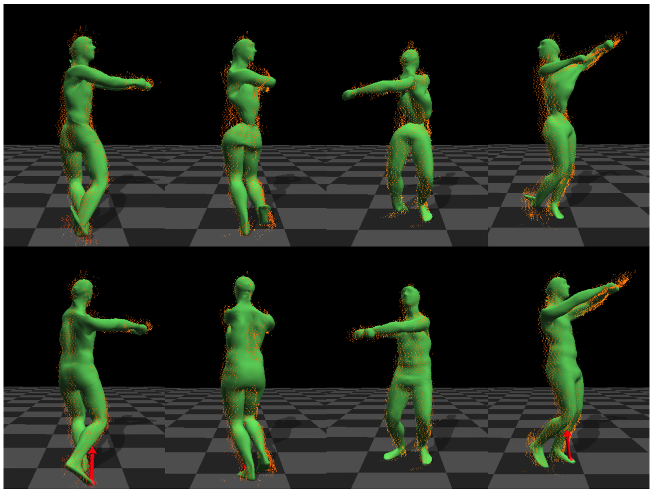

Figure 15: *Importance of the prior term. (top) result without the prior term; (bottom) result with the prior term.*

the left knee by temporally aligning and averaging 120 sequences (red curve in Figure 16 (a)).

We capture a walking sequence of a different subject using our full-body kinematics and dynamics capture system. Figure 16 (b) shows a plot of internal joint torque of the left knee for a single walking cycle of the reconstructed dynamic data (blue curve). The figure shows that our reconstruction data (blue curve) has very similar patterns as those (red curve) obtained from the *Vicon* system and force plates.

9 Conclusion

In this paper, we have developed an end-to-end full-body motion capture system using input data captured by three depth cameras and a pair of pressure-sensing shoes. Our system is appealing because it is low-cost and fully automatic, and can accurately reconstruct full-body kinematics and dynamics data. The system is also non-intrusive and easy to set up because it requires no markers and no special suits. We have demonstrated the power of our approach by capturing a wide range of complex human movements. The system achieves state-of-the-art accuracy in our comparison against

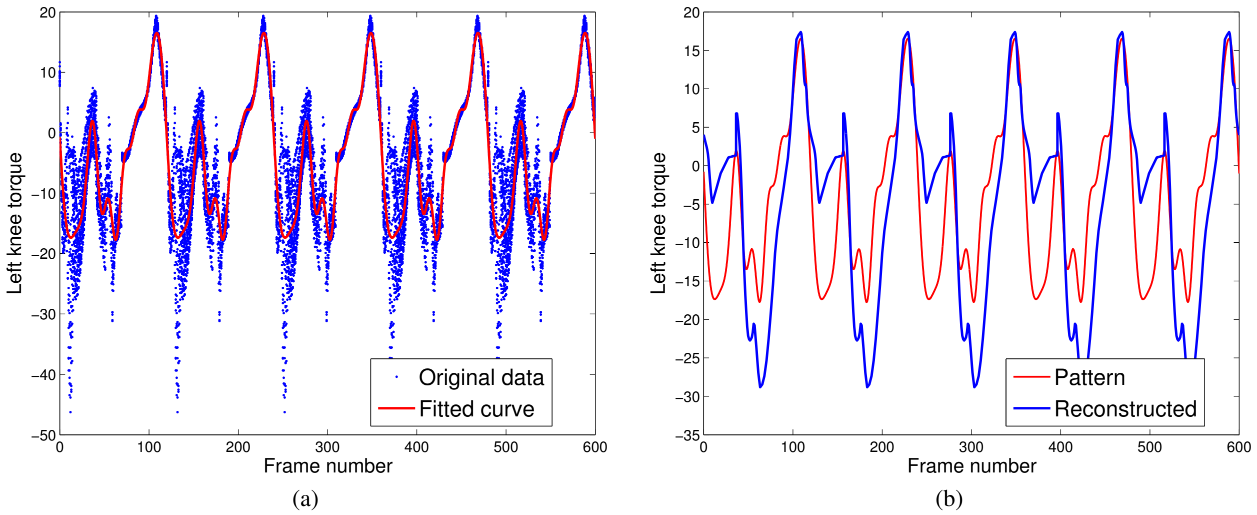

Figure 16: *Validation of reconstructed dynamic data. (a) internal torque patterns (red curve) obtained by temporally aligning and averaging 120 walking sequences captured by the* Vicon *system and force plates; (b) internal joint torques patterns (blue curve) from our result superimposed on internal torque patterns (red curve) obtained from the* Vicon *system and force plates.*

alternative methods.

Complementing depth data with pressure data not only improves the accuracy and robustness of the kinematic tracking process but also enables us to automatically capture and reconstruct full-body poses, joint torques, and contact forces all at once. The current system is based on three depth cameras and our own version of prototype pressure sensors. Our framework, however, is flexible and is not limited to particular types of sensors. For example, any pressure sensor commercially available (*e.g.*, Tekscan F-Scan [2014]) could be plugged into our system. We could also replace three depth cameras with a single consumer-level video camera to acquire motions difficult to capture in the lab, such as a run on the beach or a boxing match.

We choose to reconstruct human body kinematic data and dynamic data in a sequential manner because we are focused on online applications. An alternative solution is to use batch-based optimization [Wei and Chai 2010] to reconstruct kinematics and dynamics data for a certain period of time. For our application, however, batch-based optimization is very time consuming and memory-intensive because it requires solving a complex non-linear optimization with a huge number of constraints. We have also chosen to sequentially reconstruct kinematics and dynamics data because kinematic motion data obtained from the tracking process are often highly accurate and often sufficient to reconstruct the dynamics data. If the kinematic motion data are not reliable, a better solution is to use all the observed data, along with physical constraints, to simultaneously optimize kinematic and dynamic variables. This inevitably requires solving a more challenging optimization problem and certainly will slow down the entire reconstruction process.

Our full-body shape modeling process enables our system to work for human subjects of different body sizes and proportions. In the future, we would like to include more body scans into the training data sets to improve the generalization ability of our parametric model, as the current training data sets are still not sufficient to model shape variations across all the human subjects. Another way to improve the accuracy and robustness of the system is to combine depth data with color image data. We are particularly interested in incorporating color and texture information obtained from a video camera into the current tracking framework.

Our system often fails to produce good results when a large portion of depth data is missing (*e.g.*, when a large part of the body is out of the camera range) or when significant occlusions occur (*e.g.*, when the hands is extremely close to the torso that it cannot be distinguished from the subject's torso). Another limitation of the current system is that it can only capture contact phenomena between feet and the ground. The current system is not suitable to capture motion with complex contact phenomena such as falling down to the ground and rolling on one's back. In the future, we wish to explore how to capture full-body kinematics and dynamics data for these kinds of motions. We are also interested in extending the current system to capture interaction between multiple subjects.

We believe the new type of data captured by our system will provide insights into designing controllers for simulated virtual humans and biped robots, as well as extending our current biomechanics knowledge in motor control. In particular, the captured kinematics and dynamics data could be leveraged for many applications in human motion processing, analysis and synthesis, such as motion filtering, motion editing, motion registration, and physics-based motion control and optimization. For example, the motion can be cleaned to remove noise at the level of the driving signal (joint torques), it can be more accurately edited to meet new constraints, it would allow us to register the motion more accurately using both kinematics and dynamics data, it can serve as a basis for development of control algorithms for human movement, and it can be used to build much more precise models to predict how human takes a compensatory step to maintain the balance. One of the immediate directions for future work is, therefore, to investigate the applications of the captured data to human motion analysis, synthesis and control.

APPENDIX

In this section, we show how to linearize the non-linear expressions in Equation (4), (5), (6), (7) and (9) so that the non-linear least-square problem can be iteratively solved via linear system solvers.

Signed distance field term. This term can be linearized by using first-order Taylor expansion. Suppose we have the pose for previous frame $\mathbf{q}_{i-1}$, we can get the pose for current frame $\mathbf{q}_i$ by computing a $\delta\mathbf{q}$ using an optical flow-like algorithm.

By assuming the constancy of the signed distance value for the voxels in the volume, we get

$$V(\mathbf{p},t) = V(\mathbf{p}+\delta\mathbf{q}, t+\delta t), \tag{19}$$

where $V(\mathbf{p},t)$ represents the signed distance value for the voxel at position $\mathbf{p}$ at time t.

For a pose $\mathbf{q}$, we can compute the world coordinate of any point $\mathbf{p}(\mathbf{q})$ on the mesh model by forward kinematics and skeleton subspace deformation. Therefore, for any point on the model, we have

$$V(\mathbf{p}(\mathbf{q}),t) = V(\mathbf{p}(\mathbf{q}+\delta\mathbf{q}), t+\delta t). \tag{20}$$

By expanding $P(q+\Delta q)$ and $V(P(q+\Delta q), t+\Delta t)$ using Taylor expansion, we get

$$\mathbf{p}(\mathbf{q}+\delta\mathbf{q}) = \mathbf{p}(\mathbf{q}) + \frac{\partial \mathbf{p}}{\partial \mathbf{q}}\delta\mathbf{q}, \tag{21}$$

$$\begin{aligned} V(\mathbf{p}(\mathbf{q}),t) &= V(\mathbf{p}(\mathbf{q}) + \frac{\partial \mathbf{p}}{\partial \mathbf{q}}\delta\mathbf{q}, t+\delta t) \\ &= V(\mathbf{p}(\mathbf{q}),t) + \frac{\partial V}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}\delta\mathbf{q} + \frac{\partial V}{\partial t}. \end{aligned} \tag{22}$$

Hence, we have

$$\frac{\partial V}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}\delta\mathbf{q} = -(V_{t+1} - V_t), \tag{23}$$

$$A_{SDF}\,\delta\mathbf{q} = B_{SDF}, \tag{24}$$

where $A_{SDF} = \frac{\partial V}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}$, $B_{SDF} = -(V_{t+1} - V_t)$, $\frac{\partial V}{\partial \mathbf{p}} = [\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z}]$ is the gradient of the signed distance field, and $\frac{\partial \mathbf{p}}{\partial \mathbf{q}}$ is the Jacobian matrix for the point $\mathbf{p}$ with respect to $\mathbf{q}$.

Boundary, pressure data and ground penetration term. These terms can be linearized in a similar way. For the corresponding point pairs $\mathbf{p}_i(\mathbf{q})$ and $\mathbf{p}_i^*$, where $\mathbf{p}_i(\mathbf{q})$ is the ith point on the model for pose $\mathbf{q}$, $\mathbf{p}_i^*$ is the target position of $\mathbf{p}_i(\mathbf{q})$, we have

$$\mathbf{p}_i(\mathbf{q}+\delta\mathbf{q}) = \mathbf{p}_i^*. \tag{25}$$

We can linearized the left part of the equation and get

$$\mathbf{p}_i(\mathbf{q}+\delta\mathbf{q}) = \mathbf{p}_i(\mathbf{q}) + \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}}\delta\mathbf{q} = \mathbf{p}_i^*. \tag{26}$$

Hence we have

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}}\delta\mathbf{q} = \mathbf{p}_i^* - \mathbf{p}_i(\mathbf{q}), \tag{27}$$

$$A_{IK}\,\delta\mathbf{q} = B_{IK}, \tag{28}$$

where $A_{IK} = \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}}$ and $B_{IK} = \mathbf{p}_i^* - \mathbf{p}_i(\mathbf{q})$.

Prior term. For the prior term $E_{Prior}(\mathbf{q})$, we have

$$\begin{aligned} E(q) &= P_k^T(P_k(\mathbf{q}-\mu)) + \mu - \mathbf{q} \\ &= (P_k^T P_k - I)(\mathbf{q}-\mu) \end{aligned} \tag{29}$$

and

$$\begin{aligned} E(\mathbf{q}+\delta\mathbf{q}) &= (P_k^T P_k - I)(\mathbf{q}+\delta\mathbf{q}-\mu) \\ &= A_{Prior}\delta\mathbf{q} - A_{Prior}(\mu-\mathbf{q}), \end{aligned} \tag{30}$$

where $A_{Prior} = P_k^T P_k - I$, I is an identity matrix.

$E(\mathbf{q}+\delta\mathbf{q})$ can be solved by $\frac{\partial E(\mathbf{q}+\delta\mathbf{q})}{\partial \delta\mathbf{q}} = 0$, thus we have

$$\frac{\partial E(\mathbf{q}+\delta\mathbf{q})}{\partial \delta\mathbf{q}} = 2E(\mathbf{q}+\delta\mathbf{q})^T \frac{\partial E(\mathbf{q}+\delta\mathbf{q})}{\partial \delta\mathbf{q}} = 0, \tag{31}$$

$$(A_{Prior}\delta\mathbf{q} - A_{Prior}(\mu-\mathbf{q}))^T A_{Prior} = 0, \tag{32}$$

$$A_{Prior}^T A_{Prior}\delta\mathbf{q} = A_{Prior}^T B_{Prior}, \tag{33}$$

where $A_{Prior} = P_k^T P_k - I$ is the Jacobian matrix for the term and $B_{Prior} = (P_k^T P_k - I)(\mu - \mathbf{q})$.

Acknowledgements

We thank Peter Presti for his assistance in designing and developing the pressure-sensing shoes. We also thank Xiaolei Lv for providing us joint torque patterns captured by the Vicon system and force plates. This work is partially supported by the National Science Foundation under Grants No. IIS-1055046, IIS-1065384 and IIS-1064983, and NIH R01 NS069655-05.

References

ADELSBERGER, R., AND TRÖSTER, G. 2013. Pimu: A wireless pressure-sensing imu. In *ISSNIP*, IEEE Proceedings of the 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph. 22*, 3 (July), 587–594.

ASCENSION, 2014. http://www.ascension-tech.com/.

BAAK, A., MÜLLER, 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 *IEEE 13th International Conference on Computer Vision (ICCV)*, 1092–1099.

BREGLER, C., MALIK, J., AND K PULLEN. 2004. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*. 56(3):179–194.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '03, 28–36.

BRUBAKER, M. A., AND FLEET, D. J. 2008. The Kneed Walker for human pose tracking. In *Proceedings of IEEE CVPR*.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '96, 303–312.

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Trans. Graph. 27*, 3 (Aug.), 98:1–98:10.

ELGAMMAL, A., AND LEE, C. 2004. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2: 681–688.

FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM 24*, 6, 381–395.

FISHER, S., AND LIN, M. C. 2001. Deformed distance fields for simulation of non-penetrating flexible bodies. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 99–111.

GREST, D., KRUGER, V., AND KOCH, R. 2007. Single view motion tracking by depth and silhouette information. In *Proceedings of the 15th Scandinavian Conference on Image Analysis (SCIA)*, 719–729.

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Nonconvex rigid bodies with stacking. *ACM Trans. Graph. 22*, 3 (July), 871–878.

HA, S., BAI, Y., AND LIU, C. K. 2011. Human motion reconstruction from force sensors. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 129–138.

HASLER, N., STOLL, C., SUNKEL, M., ROSENHAHN, B., AND SEIDEL, H.-P. 2009. A statistical model of human pose and body shape. In *Computer Graphics Forum (Proc. Eurographics 2008)*, P. Dutr'e and M. Stamminger, Eds., vol. 2.

KNOOP, S., VACEK, S., AND DILLMANN, R. 2006. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1686–1691.

KOVAR, L., SCHREINER, J., AND GLEICHER, M. 2002. Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '02, 97–104.

MICROSOFT KINECT API FOR WINDOWS, 2014. http://www.microsoft.com/en-us/kinectforwindows/.

MOESLUND, T. B., HILTON, A., AND KR*ü*GER, V. 2006. A survey of advances in vision-based human motion capture and analysis. *Journal of Computer Vision and Image Understanding*. 104:90–126.

PAVLOVIĆ, V., REHG, J. M., AND MACCORMICK, J. 2000. Learning Switching Linear Models of Human Motion. In *Advances in Neural Information Processing Systems 13*, 981–987.

PLAGEMANN, C., GANAPATHI, V., KOLLER, D., AND THRUN, S. 2010. Realtime identification and localization of body parts from depth images. In *Proceedings of International Conferences on Robotics and Automation (ICRA 2010)*, 3108–3113.

ROSALES, R., AND SCLAROFF, S. 2000. Specialized mappings and the estimation of human body pose from a single image. In *Proceedings of the Workshop on Human Motion*. 19-24.

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 a single depth image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1297–1304.

TEKSCAN, 2014. http://www.tekscan.com/.

URTASUN, R., FLEET, D. J., HERTZMANN, A., AND FUA., P. 2005. Priors for people tracking from small training sets. In *IEEE International Conference on Computer Vision*, 403–C410.

VICON SYSTEMS, 2014. http://www.vicon.com.

VLASIC, D., BARAN, I., MATUSIK, W., AND POPOVIĆ, J. 2008. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph. 27*, 3 (Aug.), 97:1–97:9.

VONDRAK, M., SIGAL, L., AND JENKINS, O. C. 2008. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

VONDRAK, M., SIGAL, L., HODGINS, J., AND JENKINS, O. 2012. Video-based 3d motion capture through biped control. *ACM Trans. Graph. 31*, 4 (July), 27:1–27:12.

WEI, X. K., AND CHAI, J. 2010. Videomocap: Modeling physically realistic human motion from monocular video sequences. *ACM Transactions on Graphics*. 29(4): Article No. 42.

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.

XSENS, 2014. http://www.xsens.com.

YE, M., WANG, X., YANG, R., REN, L., AND POLLEFEYS, M. 2011. Accurate 3d pose estimation from a single depth image. In *Proceedings of IEEE 13th International Conference on Computer Vision*, 731–738.

YIN, K., AND PAI, D. K. 2003. Footsee: An interactive animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '03, 329–338.