

Chapter 5

Human Performance Capture Using Multiple Handheld Kinects

Yebin Liu, Genzhi Ye, Yangang Wang, Qionghai Dai
and Christian Theobalt

Abstract Capturing real performances of human actors has been an important topic in the fields of computer graphics and computer vision in the last few decades. The reconstructed 3D performance can be used for character animation and free-viewpoint video. While most of the available performance capture approaches rely on a 3D video studio with tens of RGB cameras, this chapter presents a method for marker-less performance capture of single or multiple human characters using only three handheld Kinects. Compared with the RGB camera approaches, the proposed method is more convenient with respect to data acquisition, allowing for much fewer cameras and carry-on camera capture. The method introduced in this chapter reconstructs human skeletal poses, deforming surface geometry and camera poses for every time step of the depth video. It succeeds on general uncontrolled indoor scenes with potentially dynamic background, and it succeeds even for reconstruction of multiple closely interacting characters.

5.1 Introduction

In recent years, the field of marker-less motion estimation has seen great progress. Two important lines of research have recently emerged in this domain. On the one side, there are multi-view motion capture approaches that reconstruct skeleton motion and possibly simple body shape of people in skintight clothing from a set of video recordings that were taken from around the scene, e.g., [1–6]. Human performance capture approaches take one step further and not only reconstruct a motion model (like a skeleton) but also detailed dynamic surface geometry as well as detailed texture, e.g., [7–12]. However, these approaches are still limited to mostly controlled studio settings and static frame-synchronized multi-video systems which often feature 10 or more cameras.

Y. Liu (✉) · G. Ye · Y. Wang · Q. Dai · C. Theobalt
Tsinghua University, Beijing, China
e-mail: liuyebin@tsinghua.edu.cn

On the other end of the spectrum are methods for marker-less motion capture from a single camera view at interactive or near real-time frame rates. Estimation of complex poses from a single video stream is still a very challenging task. The recent advent of depth cameras, such as time-of-flight sensors [13] and the Microsoft Kinect, has opened up new possibilities. These cameras measure 2.5D depth information at real-time frame rates and, as for the Kinect, video as well. This makes them ideal sensors for pose estimation, but they suffer from significant noise and have, at best, moderate resolution. Therefore, (using a single depth camera) it has been difficult to capture 3D models of a complexity and detail comparable to multi-view performance capture results.

This chapter introduces a method to do full performance capture of moving humans using just three handheld, thus potentially moving, Kinect cameras. It reconstructs detailed time-varying surface geometry of humans in general apparel, as well as the motion of the underlying skeleton without any markers in the scene. It can handle fast and complex motion with many self-occlusions. By resorting to depth sensors, it can be applied to more general uncontrolled indoor scenes and is not limited to studios with controlled lighting and many stationary cameras. Also, it requires only three handheld sensors to produce results that rival reconstructions obtained with recent state-of-the-art multi-view performance capture methods [14, 15].

The key technology in this method is a tracking algorithm that tracks the motion of the handheld cameras and aligns the RGB-D data and that simultaneously aligns the surface and skeleton of each tracked performer to the captured RGB-D data. The algorithm also needs to be robust against the sensor noise, as well as missing data due to multi-Kinect interference and occlusions in the scene. It therefore introduces an efficient geometric 3D point-to-vertex assignment strategy to match the Kinect RGB-D data points to the geometric model of each performer. The assignment criterion is stable under missing data due to interference and occlusions between persons. Based on this criterion, a segmentation of the scene into performers, ground plane, and background is implicitly achieved. As a second correspondence criterion, it detects and tracks SIFT features in the background part of each video frame. Based on these model-to-data assignment criteria, the pose parameters of the performers and the poses and orientations of the Kinects are jointly estimated in a combined optimization framework. Finally, the nonlinear objective function can be linearized and effectively minimized through a quasi-Newton method.

The experimental section shows results on several challenging single- and multi-person motions including dance and martial arts. Also, quantitative proof of the accuracy of the reconstruction method is given by comparing it to an available video-based performance capture approach [12].

The technique presented in this chapter is a revision of former published papers [16, 17].¹

¹ [2014] IEEE. Reprinted, with permission, from [Genzhi Ye, Yebin Liu, Yue Deng, Nils Hasler, Xiangyang Ji, Qionghai Dai, Christian Theobalt, Free-viewpoint Video of Human Actors using Multiple Handheld Kinects, IEEE Trans. Cybernetics, 43(5), pp 1370–1382, 2013].

5.2 Related Works

Most human performance capture approaches reconstruct human skeletal motion in controlled studios with a large number of cameras [3, 18, 19]. They usually employ a template skeleton with simple shape primitives [2, 5, 20] to fit the image data by optimizing an energy function parameterized by the template skeleton. This energy function usually exploits motion cues, such as silhouette, edge, and salient features. Typically, they are solved by local [2] or global optimization [20, 21], or the combine of the two [12].

Some advanced methods further reconstruct a detailed 3D deforming surface [7, 12, 22] of people in more general clothing. All these methods take advantage of a more elaborate shape and skeleton template, to improve the tracking accuracy while enforcing some surface refinement [12] or shading refinement [23] for better geometry reconstruction, or possibly reflectance reconstruction [24]. Recently, Wu et al. [22] proposed the integration of shading cues and develop the local linear optimization for more reliable skeletal pose tracking from indoor multi-view video.

The recent trend of marker-less motion capture aims to simplify the capture setup, e.g., by reducing the number of cameras, capturing in outdoor scenes, or using handheld devices. Outdoor marker-less motion capture with handheld cameras is studied in the pioneering work by Hasler et al. [25], but the accuracy of their method is restricted by the limited silhouette cues. Under the assumption of fixed distant global illumination, Wu et al. [26] employed a handheld stereo rig for performance capture in front of a general background, which significantly broadens the operation range of marker-less motion capture. But their local optimization and fixed global illumination model is only demonstrated for relatively restricted camera motion. However, only a frontal view of depth information from a binocular camera cannot guarantee a robust tracking under fast motion and serious occlusions. Wei et al. [27] further studied motion tracking from a monocular video. This challenging problem requires intensive manual interventions.

The prevalence of consumer depth cameras, such as the Microsoft Kinect camera, opens up new opportunities to solve fundamental problems in computer vision [28] and especially in human pose estimation [29, 30] and motion tracking [31]. Shotton et al. [32] and Ganapathi et al. [33] developed real-time motion estimation systems for indoor scenes captured recorded by a depth camera, which enables enormous applications in human-computer interaction and gaming. Typically, to achieve real-time performance, discriminative methods [34, 35] are applied to predict skeleton pose directly from the depth map by a trained regression model. Some recent works [31, 36, 37] further improved these discriminative methods with local optimization-based motion tracking. However, pose tracking with a single depth camera is far from robust and accurate and is challenged by motions with serious self-occlusions.

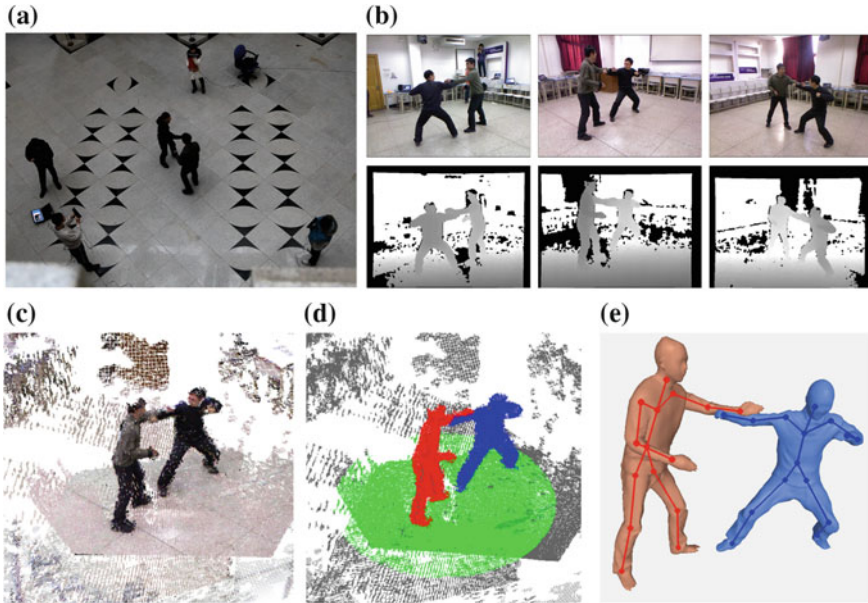


Fig. 5.1 Overview of the processing pipeline. **a** Overhead view of typical recording setup: three camera operators (*circled in red*) film the moving people in the center (*blue*); **b** input to the algorithm – RGB images and the depth images from three views; **c** Registered RGB-D point cloud from all cameras; **d** segmented RGB-D point cloud—color labels correspond to background, ground plane (*green*), and interacting humans (*red, blue*); **e** reconstructed surface models and skeletons

5.3 Data Capture and Scene Representation

For the data capture, one or more moving humans are recorded by $C = 3$ individuals (camera operators) that stand around the scene. Each of the operators holds a Kinect camera and points it toward the center of the recording volume. They are free to move the cameras during recording. The performance capture method can handle a certain amount of moving scene elements in the background. To improve the practicability of the whole system, we rely on simple and convenient software synchronization since hardware synchronization of multiple Kinects so far is not possible. Each Kinect is connected to a notebook computer, and all recording notebooks are connected through WiFi. One computer serves as a master that sends a *start recording* signal to all other computers. The cameras are set to a frame rate of 30fps, and with the software solution, the captured data of all cameras are frame-synchronized with at most 10ms temporal difference.

Each Kinect captures a 640×480 video frame and an aligned depth frame at every time step t , Fig. 5.1b, which yields a combined RGB-D point cloud. The intrinsics of both the depth and the video cameras are calibrated off-line using a checkerboard [38]. Depth and color data are aligned with each other using the OpenNI API [39].

For each RGB-D point p , we store a triplet of values $p = \{x_p, n_p, l_p\}$. Here, x_p is the 3D position of the point, n_p is the local 3D normal, and l_p is a RGB color triplet. The normal orientations are found by PCA-based plane fitting to local 3D point neighborhoods. Note that the 3D point locations are given with respect to each camera's local coordinate system. For performance capture, the points from all cameras are required to be aligned into a global system. Since the Kinects are allowed to move in our setting, the extrinsic camera parameters Λ_c^t (position and orientation) of each Kinect c at every time step of video t , i.e., the combined extrinsic set $\Lambda^t = \{\Lambda_c^t\}_{c=1}^C$, need to be solved for. Fig. 5.1c shows the merged point set at time t after solving for the extrinsics using the method later described in this chapter. Also, due to occlusions in the scene and interference between several Kinects, 3D points corresponding to some Kinect camera pixels cannot reliably be reconstructed. The joint camera tracking and performance capture method thus need to be robust against such missing measurements.

For each of the $k = 1, \dots, K$ performers in the scene, a template model is defined. Similar to [12, 14], a template model comprises a surface mesh M_k with an embedded kinematic bone skeleton (see Fig. 5.1e). A laser scanner is used to get a static surface mesh of the person. Alternatively, image-based reconstruction methods could be used or the mesh could be reconstructed from the aligned Kinect data [40–42]. The surface models are remeshed to have around $N_k = 5,000$ vertices. Each vertex is also assigned a color that can change over time, as described in Sect. 5.4.3. Henceforth, the 3D positions of vertices of mesh k with attached colors at time t are denoted by the set $V_k^t = \{v_{k,i}^t\}_{i=1}^{N_k}$. To stabilize simultaneous 3D human shape and Kinect position tracking, the ground plane is also explicitly modeled as a planar mesh V_0^t with circular boundary. The ground plane model has a fixed radius of 3m, and during initialization, it is centered below the combined center of gravity of the human models (see Fig. 5.1d). In total, this yields a combined set of vertex positions $V^t = \{\{V_k^t\}_{k=0}^K\}$ that need to be reconstructed at each time step. This excludes the ground plane vertices as their position is fixed in world space. Its apparent motion is modeled by moving the cameras.

A kinematic skeleton with $n = 31^\circ$ of freedom (DoFs) is manually placed into each human mesh, and surface skinning weights are computed using a similar process as [12, 14]. Skeleton poses $\chi^t = (\xi^t, \Theta^t) = (\theta_0 \hat{\xi}, \theta_1, \dots, \theta_n)$ are parameterized using the twist and exponential maps parameterizations [2, 12]. $\theta_0 \hat{\xi}$ is the twist for the global rigid body transform of the skeleton, and Θ^t is the vector of the remaining angles. Using linear blend skinning, the configuration of a vertex of human mesh M_k in skeleton pose χ_k^t is then determined by

$$v_i(\chi_k^t) = \sum_{m=1}^n \left(w_i^m \prod_{j \in \text{Parent}(m)} \exp(\theta_j \hat{\xi}_j) \right) v_i. \quad (5.1)$$

Here, w_i^m is the skinning weight of vertex i with respect to the m -th DoF. Further on, $\text{Parent}(m)$ is the set of all the DoFs in the kinematic chain that influences

the m -th DoF, i.e., all the parents of m -th DoF. In addition to $\Lambda^t = \{\Lambda_c^t\}_{c=1}^C$, the performance capture approach thus needs to solve for the joint parameters of all persons at each time step, $X^t = \{\chi_k^t\}_{k=1}^K$.

5.4 Human Performance Capture

Performance capture from 3D point data is only feasible if the RGB-D data from all Kinects are correctly registered. In the beginning, for each time step, the correct extrinsics Λ_t are unknown. A traditional approach to track camera extrinsics is structure from motion (SfM) performed on the background of the sequence [25]. However, in our multiple Kinect recording setting, the moving subjects fill most of the visible area in each video frame and thus, a different approach has to be used. In such a setting, human pose capture and camera pose estimation are performed simultaneously, leading to more robust results. In other words, the optimization tries to mutually align all point clouds and fit the poses of the actors to the RGB-D data. At the same time, feature correspondences in the background are exploited similarly to SfM, since they provide additional evidence for correct reconstruction. Camera and body poses are therefore simultaneously computed, and the solution is regularized to additional feature correspondences found in the video frame.

In the first frame of multi-view RGB-D video, camera extrinsics are initialized interactively and the template models are fitted to each person’s depth map. The initialization pose in the data sequence is guaranteed to be close to the scanned pose. Thereafter, the algorithm runs in a frame-by-frame manner applying the processing pipeline from Fig. 5.1c–e to each time step. For a time step t , the steps are as follows: The Kinect RGB-D point clouds are first aligned according to the extrinsics Λ^{t-1} from the previous frame. Starting with the pose parameters X^{t-1} and resulting mesh configurations and vertex colors from the previous frame, a matching algorithm is introduced to match the Kinect point data to the model vertices. During this matching, the RGB-D data are also implicitly segmented into classes for *ground plane*, *background*, and one class for each *person*, Sect. 5.4.1 (Fig. 5.1d). Thereafter, a second set of 3D correspondences is found by matching points from the *ground plane* and the *background* via SIFT features, Sect. 5.4.1.

Based on these correspondences, Sect. 5.4.2 simultaneously solves for Kinect and skeleton poses for the current frame. Correspondence finding and reconstruction are iterated several times, and the model poses and point cloud alignments are continuously updated (Fig. 5.1c). Non-rigid deformations of the human surface, e.g., due to cloth deformation, are not explained by skeleton-driven deformation alone. In a final step, the meshes M_k are thus non-rigidly deformed into the aligned point clouds via Laplacian deformation and the vertex colors of the mesh model(s) are updated (Fig. 5.1e). The following section explains each step in the algorithm for a specific time t , and it omits the index t for legibility.

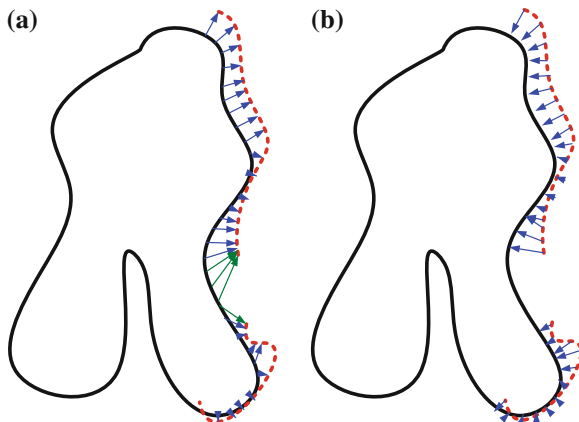


Fig. 5.2 Comparison of forward matching and inverse matching when Kinect data occlusion happens. The red points are Kinect data, and the black lines are model surface. The blue arrows and green arrows are matching. **a** forward matching. Green matching are error correspondences. **b** inverse matching

5.4.1 Point Data Correspondence and Labeling

The reconstructed 3D human models should be matched to the RGB-D point clouds. Therefore, an error function that measures the alignment of the RGB-D point clouds with the 3D human models is minimized for finding the correct camera and body configurations, Sect. 5.4.2. To evaluate this error, for all scene model vertices V , plausible correspondences to the RGB-D points P need to be defined. With these correspondences, the alignment error can be evaluated, as it was also used in video-based performance capture to measure alignment in the image domain [12].

Due to mutual occlusions, the Kinect point cloud P will not always sample every part of the body surfaces. Additionally, interference between several Kinects renders some 3D points unreliable. That is, in this scenario, matching model vertices V to Kinect point clouds P tends to be unstable. Also, the matching term should ensure that each 3D human template is explained by the point data. Therefore, as shown in Fig. 5.2, reverse matching is much more robust since all the foreground points physically exist and, in theory, can all be explained by the model surface, although there is noise and outliers in the captured data. Thus, the closest mesh vertices for all RGB-D points are proposed as matches.

Here, we define the metric F for the searching of a model vertex v to a given 3D point p . Such metric simultaneously measures the color distance and a geometric distance as follows:

$$F(v, p) = \Delta(\|l_v - l_p\|, \theta_l) \Delta(\|x_v - x_p\|, \theta_x) \max(n_v n_p, 0) \quad (5.2)$$

where

$$\Delta(x, \theta) = \max\left(1 - \frac{x}{\theta}, 0\right) \quad (5.3)$$

Here, x_p, l_p, n_p and x_v, l_v, n_v denote the position, color, and normal of a Kinect point and a mesh vertex, respectively. The first part in F is a color term enforces color similarity between the mesh vertex and the corresponding Kinect point, with the maximum color difference θ_l is experimentally set to 100. The second part in F is a geometry term, and it only matches RGB-D points and vertices that are spatially close and have similar normal orientation. The maximum distance a mesh vertex is allowed to move θ_x is also experimentally set to 100 mm.

Based on F , we first select the points corresponding to the persons and the ground plane as:

$$Z_V^k = \left\{ (v, p) \mid v = \underset{v}{\operatorname{argmax}} F(v, p), F(v, p) > 0, v \in M_k \right\} \quad (5.4)$$

and

$$Z_G = \left\{ (v, p) \mid v = \underset{v}{\operatorname{argmax}} F(v, p), F(v, p) > 0, v \in V_0 \right\}. \quad (5.5)$$

For each point p , the vertex v is first selected from V to maximize F . If the maximum $F > 0$, according to the label of v , the correspondence (p, v) is classified into a person correspondence set Z_V^k of person k , or into the ground plane correspondence set Z_G . After the correspondences $Z_V = \{Z_V^k\}_{k=1}^K$ and Z_G are established, the RGB-D point cloud is thus implicitly segmented into one class for each *person*, *ground plane*, and *background* for all RGB-D points that were not assigned a corresponding point in V .

As stated in the beginning, the reconstruction error is also based on feature correspondences in the scene background, similar to classical structure-from-motion approaches. The method from Sect. 5.4.1 provides a classification of background RGB-D points and thus corresponding RGB pixels in each Kinect video image. SIFT features are detected on the background regions of the RGB images from $t - 1$ and t and then converted into 3D correspondences $Z_S = \{(p', p) \mid p' \in P^{t-1}, p \in P^t, (p', p) \text{ matched via SIFT}\}$ through the available depth. As stated earlier, background correspondences are not always fully reliable. Measurement accuracy decreases with increasing distance from the camera, and moving objects in the background lead to erroneous correspondences. Thus, the error function additionally measures point-to-model correspondences in the foreground. Fig. 5.3b shows that alignment based on SIFT features in the background alone will not suffice.

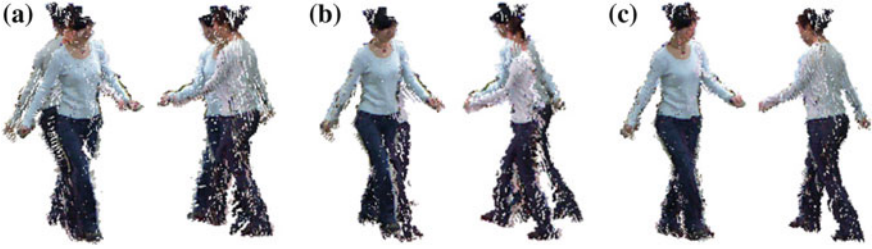


Fig. 5.3 Comparison of RGB-D point data fusion at frame t before and after joint skeleton and Kinect optimization. **a** Fusion using extrinsics from the former time. **b** Fusion based on SIFT features alone fails. **c** Fusion using extrinsics solved by the combined human and camera pose optimization produces much better results

5.4.2 Joint Skeleton and Kinect Tracking

After the computing of correspondence sets Z_V , Z_G , and Z_S , a geometric error function can be defined and minimized in the space of skeleton pose X and camera extrinsics Λ :

$$E(X, \Lambda) = \arg \min_{X, \Lambda} \left\{ \sum_{(p,v) \in Z_S} \frac{\|p(\Lambda) - v(X)\|^2}{\|Z_V\|} + \sum_{(p,v) \in Z_G} \frac{\|p(\Lambda) - v\|^2}{\|Z_G\|} + \sum_{(p,p') \in Z_S} \frac{\|p(\Lambda) - p'\|^2}{\|Z_S\|} \right\} \quad (5.6)$$

Here, $\|Z\|$ is the number of elements in set Z . This function is solved through linearization within an iterative quasi-Newton minimization. Using Taylor expansion of the exponential map, the transformation of Λ on point p leads to a linear formulation

$$p(\Lambda) = Rp + T = e^{\theta \hat{\xi}} p \approx (I + \theta \hat{\xi}) p \quad (5.7)$$

For the body pose, we can perform the similar expansion. Specifically, Eq. (5.1) can be linearized as

$$v(X) = \left(I + \theta_0 \hat{\xi}_0 + \sum_{m=1}^n \left(\sum_{j \in \text{Children}(m)} w_j \right) \theta_m \hat{\xi}_m \right) v, \quad (5.8)$$

where, $\text{Children}(m)$ is the set of DoFs corresponding to the children DoFs of the m -th DoF.

Robust correspondence finding and pose optimization are iterated for 20 times. After each iteration, the normals of the fused point cloud points n_p are updated. Fig. 5.3 shows the comparison of the fused data before pose optimization (a) and after pose optimization (c). Please note that even using state-of-the-art techniques, direct fusion of the point data without the aid of a 3D model is extremely difficult and error prone because of the small overlap region between the different Kinects [43].

5.4.3 Surface Geometry Optimization

After tracking and skinned deformation of the skeleton-driven model of each character, mesh deformation is performed to refine the surface geometry of the performers and capture non-rigid deformation effects, such as cloth motion. Similar to [8], for each person k , surface deformation is formulated as:

$$\arg \min_{\mathbf{v}} \{ \|\mathbf{L}\mathbf{v} - \delta\|_2^2 + \|\mathbf{C}\mathbf{v} - \mathbf{p}\|_2^2 \} \quad (5.9)$$

Here, \mathbf{v} denotes the vector of vertices on human body mesh M_k . L is the discrete Laplace operator, and δ is the differential coordinates of the current mesh vertices. The Laplace operator is defined as [44]

$$L = \begin{cases} d_i & i = j \\ -1 & (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (5.10)$$

and δ is usually defined as

$$\delta_i = \frac{1}{d_i} \sum_{j \in N(i)} (v_i - v_j). \quad (5.11)$$

Here, $(i, j) \in E$ means vertex v_i and v_j are on the same edge and d_i is the number of neighbor vertices of v_i . The definition of δ is upgraded to cotangent weights in this work, please refer to [44] for detail. In (5.9), \mathbf{C} is a diagonal matrix with nonzero entries $c_{jj} = \alpha$ ($\alpha=0.1$) for vertices in correspondence set Z_{pv}^k . \mathbf{p} is the vector with nonzero position entries for those p in Z_V^k .

After non-rigid mesh deformation, the color of each vertex is updated according to a linear interpolation between the previous color and the current color using

$$l_v = \frac{t}{t+1} l_v + \frac{1}{t+1} l_{nn} \quad (5.12)$$

where l_{nn} is the color of the nearest RGB-D neighbor point of v .

5.5 Experimental Results

This section verifies the performance capture system from both the perspectives of qualitative analysis and quantitative evaluations. The data were recorded with three moving Kinects at a resolution of 640×480 pixels and at a frame rate of 30fps. The sequence is consisted of a wide range of different motions, including



Fig. 5.4 Performance capture results on a variety of sequences: one of the input image, layered geometry, reconstructed geometry, and skeleton. Each row shows two results

dancing, fighting, and jumping, see Fig. 5.4, and accompanying video.² The motions were performed by five different persons wearing casual clothing. There are also two evaluation sequences where the performer was simultaneously tracked by a multi-view video system and also three evaluation sequences where one of the two human actors is wearing a marker suit for simultaneous optical motion capture. The configurations of the acquisition setups for all these sequences are shown in Table 5.1.

5.5.1 Qualitative Evaluation

Figure 5.4 shows several results produced by the system. The approach enables fully automatic reconstruction of skeletal pose and shape of two persons, even if they are as closely interacting as in martial arts fight, hug, or while dancing, see Fig. 5.4 and the accompanying video. Despite notable noise in the captured depth maps, the method successfully captures pose and deforming surface geometry of persons in

² The accompanying video is available at: www.media.au.tsinghua.edu.cn/kinectfvv.mp4.

Table 5.1 Description of the capture sequences

Sequence	Frame rate	Number of performers (K)	Number of Kinects (C)	Number of frames	Kinect status	Comparison
Dancing walk	30	1	3	300	Moving	No
Kungfu	30	1	3	300	Moving	No
Couple dance	30	2	3	300	Moving	No
Fight	30	2	3	300	Moving	No
Hug	30	2	3	250	Moving	No
Arm crossing	30	1	3	400	Static	No
Rolling	15	1	3	200	Static	Multi-view Video
Jump	15	1	3	200	Static	Multi-view Video
Exercise1	30	1	3	450	Static	Marker based
Exercise2	30	1	3	450	Moving	Marker based
Exercise3	30	2	3	450	Moving	Marker based

loose apparel. With a capturing frame rate of only 30fps, the introduced approach can also handle very fast motions, see the jump and kicking motions in Fig. 5.4.

5.5.2 Comparison to Vision-Based Motion Capture

To compare against a vision-based motion capture system, two sequences are captured in a multi-view video studio with 10 calibrated cameras (15fps, 1024×768) and a green screen in the background. The Kinect data were temporally aligned to the multi-view video data at frame-level accuracy using event synchronization. Although the synchronization of the video camera system and the Kinect system is not guaranteed at subframe accuracy, the evaluation of the difference between the two results still presents a conservative performance evaluation of the proposed algorithm.

Since the multi-view video system runs at 15fps, a sequence “rolling” is captured with slow motion and a sequence “jump” with fast motion. The Kinect system runs at 30fps, so the frames from the multiple Kinect system are subsampled by factor two and compared to the performance captured with multi-view video-based tracking (MVT) [12]. Figure 5.5 visually demonstrates the results of the two systems on the basis of four frames selected at regular equidistant intervals from the “rolling” sequence. Since the MVT requires a green screen for clean background subtraction and it, thus, does not work with extra camera operators in the scene background, the three Kinects are fixed in the MVT studio during data capture. With these fixed Kinects, the introduced algorithm can be validated by comparing the optimized Kinect extrinsics in the later frames with those of the first frame. The average distance from the Kinect center in the first frame to the Kinect center of other frames (both “rolling” and “jump”) for each of the Kinects are 10.66 mm, 7.28 mm and

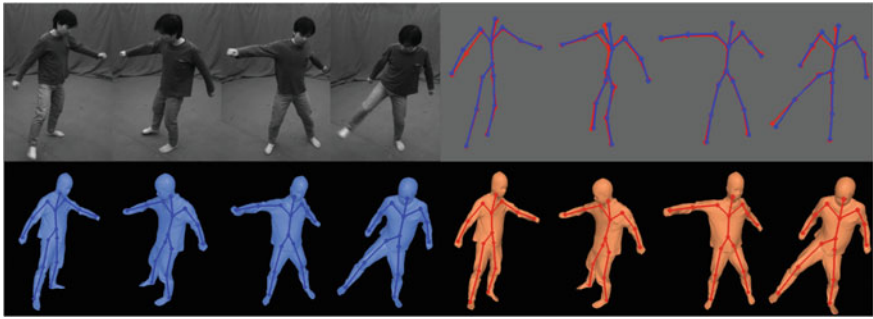


Fig. 5.5 Comparison with multi-view video-based tracking (MVT) approach on the “rolling” sequence. The *top left* are four input images of the multi-view video sequence. The *top right* shows the close overlap of the two skeletons tracked with MVT (*blue*) and the multiple Kinect-based approach (*red*). The *bottom left* is the reconstructed surface with the skeleton using MVT, and the *bottom right* is the results from the multiple Kinect-based approach. Quantitative and visual comparisons show that MVT-based and Kinect-based reconstructions are very similar



Fig. 5.6 Comparison with multi-view video-based tracking (MVT) approach on the “jump” sequence. The *left three* and the *right three* are input image, result of MVT, and result of the Kinect-based approach. On this fast motion, Kinect-based tracking succeeds, while MVT fails to capture the arm motion

6.67 mm, respectively. For the slow motion sequence “rolling”, the result from the multiple Kinect system closely matches the input images and the result of the MVT system, see Fig. 5.5. In addition, the differences on the joint centers of these results from the two systems are computed. The average distance between the corresponding joint positions across all 200 frames of the sequence is 21.42 mm with a standard deviation of 27.49 mm. This distance also includes the synchronization differences between the two systems. For the fast motion sequences, the MVT even fails despite a much higher number of cameras, while the Kinect-based tracking is able to track the whole sequence, see Fig. 5.6.

5.5.3 Comparison to Marker-Based Motion Capture

Most of the commercial motion capture systems apply marker-based techniques since they provide comparably robust and accurate performance. In this chapter, a quantitative evaluation on the accuracy of simultaneous skeleton motion capture and

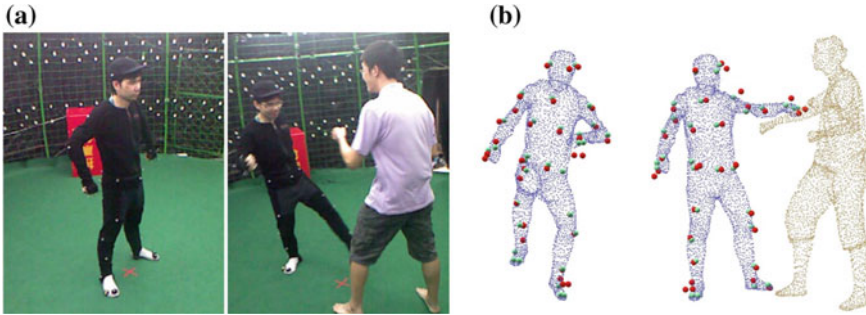


Fig. 5.7 The capturing environment with the OptiTrack motion capture system **a** and comparison between this marker-based Mocap system and our algorithm **b**. *Red* dots represent the ground truth marker position and the *green* dots are their corresponding vertice on the body model. The distance between them is evaluated

geometry reconstruction with the proposed system against a marker-based system is conducted. One sample image showing the capturing environment of the marker system is provided in Fig. 5.7a. In our setting, OptiTrack marker-based motion capture system [45] is adopted for comparison. Besides three handheld Kinects in the capture environment, 34 optical markers were attached to one of the persons, whose motions were captured with the marker-based system. Since it is impossible to synchronize the two systems, both of the two systems run at 30fps and the start frames are manually aligned. The synchronization error is then within $1/30$ s.

The error metric is defined as the average distance between the markers and their corresponding vertices on the template model across all the frames of the sequence. The corresponding vertices are found in the first frame when the markers and the 3D model template is well aligned. The error metric not only accounts for the performance of skeleton tracking, but also the accuracy of the geometry reconstruction. Figure 5.7b and the accompany video show the ground truth marker positions and their corresponding vertices on the recovered models. Our experiments show that the multiple Kinect-based system produces reasonable and similar tracking performances as the marker system. In cases of occlusions between two persons, the tracking result is also robust and accurate. Quantitative evaluations are also performed, as shown in Fig. 5.8. The average distance and the standard derivations of the distance between each marker and their corresponding vertices on the 3D model are calculated. The results are shown in Fig. 5.8. The average error between the two systems is about 38 mm. Considering that the two systems are not strictly synchronized, this geometry accuracy is plausible. Compared with available single depth camera-based motion capture systems such as Ganapathi et al. [33], which report an average tracking error about 200 mm, the multiple Kinect-based system provides much lower tracking errors. This improvement is achieved through the combination of multiple depth cameras and the delicate algorithm design. From the quantitative evaluation on “Exercise 1”(static cameras) and “Exercise 2”(handheld cameras), it

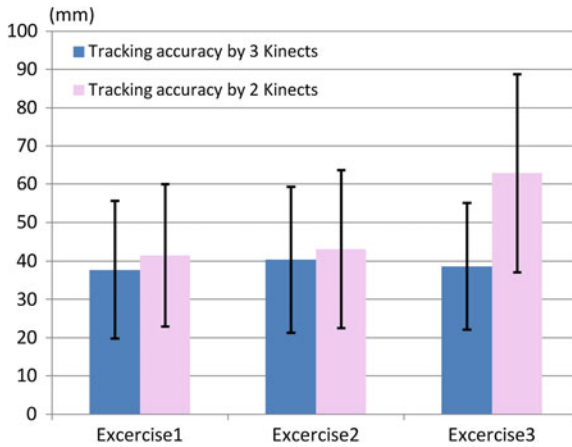


Fig. 5.8 The quantitative comparison of the Kinect-based tracking accuracy with the marker-based result (*ground truth*). The *blue* bars and *pink* bars show the accuracy and standard derivations (*std*) with three Kinects and two Kinects, respectively

is also interesting to note that, compared with static capture, handheld capture does not obviously damage system performance.

All the experiments discussed above are implemented with three Kinect cameras. To verify the robustness of the multiple Kinect-based system, the number of cameras is decreased to show how the final results are affected by the number of Kinects. The accuracy achieved with two Kinects is shown with the pink bars in Fig. 5.8. This shows that the accuracy obtained with two Kinects and three Kinects is very similar for one body tracking (for “Exercise 1” and “Exercise 2”). However, the accuracy decreases significantly by using two cameras for the “Exercise 3.” This is because in “Exercise 3,” occlusion between the two persons is more serious. Using only two cameras could not sufficiently capture the whole scenario and therefore results in a relatively low tracking quality.

5.6 Discussion

Interference: Currently, multiple Kinects aiming at the same scene may introduce interferences between each other and degrade the depth quality. In our case, the Kinects are very sparse with any of the two Kinects covering a viewing angle of about 120° . Such a baseline with such a large angle causes only about 10% of the whole pixels suffers interference for a camera in the experiments, compared with the case of single Kinect. Moreover, the Kinect will not return depth values for pixels with interference, so interference will not produce erroneous 3D point data to degrade

the tracking accuracy. To adapt for more Kinects working together, multiple depth sensors may share the same lighting source to infer depth information while reducing interference.

Complexity: Computational complexity does not starkly depend on the number of subjects in the scene. It takes about 10 s for single person tracking of a frame and 12 s for the two persons tracking on a standard PC using unoptimized code. The system uses local optimization, and the run time of the system mainly depends on the number of captured points. The number of captured points decides the time complexity of both correspondence establishment and the pose estimation. The correspondence establishment takes more than half of the total time and could be optimized using more efficient data structures, like octrees, to decrease the searching number of matching vertices to speed up the search. Real-time tracking is possible with code optimization and parallel computing.

Alternative Optimization Approach: Since the method introduced is based on local optimization, the tracking may fail when serious occlusions happen or when the motion is too fast. Alternatively, a global optimization approach [12] can be employed based on the local optimization results, similar to the method proposed for multi-view video-based tracking. Such global optimization is, for instance, based on analysis by synthesis, that is, sampling on the skeleton and camera pose space and retrieving the one that best matched to the input data in a particle swarm optimization (PSO) [46] or interacting simulated annealing (ISA) [21] optimization. Theoretically, the tracking results will be comparably more robust than local optimization approach; however, the computation complexity is greatly increased and the temporal smoothness of the tracking results will be degraded.

5.7 Conclusions

This chapter introduces a method for human performance capture using several hand-held Kinects. The method adopts a local optimization approach to simultaneously solve for the skeleton parameters and camera pose by driving them to fit to the input point data from the Kinects. The tracking approach is based on iterating robust matching of the tracked 3D models and the input Kinect data and a quasi-Newton optimization on Kinect poses and skeleton poses. This joint optimization enables us to reliably and accurately capture shape and pose of multiple performers. In summary, the proposed technique removes the common constraint in traditional multi-view motion capture systems that cameras have to be static and scenes need to be filmed in controlled studio settings. Instead, the introduced system allows users to hold the Kinects for motion capture and 3D reconstruction of performers. This enriches the practical application, especially when considering the anticipated introduction of depth cameras in consumer devices like tablets.

References

1. Deutscher J, Blake A, Reid I (2000) Articulated body motion capture by annealed particle filtering. In: IEEE conference on computer vision pattern recognition, pp 1144–1149
2. Bregler C, Malik J, Pullen K (2004) Twist based acquisition and tracking of animal and human kinematics. *IJCV* 56:179–194
3. Sigal L, Black M (2006) Humaneva: synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University
4. Balan A, Sigal L, Black M, Davis J, Haussecker H (2007) Detailed human shape and pose from images. In: IEEE conference on computer vision pattern recognition, pp 1–8
5. Stoll C, Hasler N, Gall J, Seidel HP, Theobalt C (2011) Fast articulated motion tracking using a sums of gaussians body model. In: IEEE international conference on computer vision, pp 951–958
6. Poppe R (2007) Vision-based human motion analysis: an overview. *CVIU* 108:4–18
7. Vlasic D, Baran I, Matusik W, Popović J (2008) Articulated mesh animation from multi-view silhouettes. *ACM Trans Graph* 27:1–9
8. De Aguiar E, Stoll C, Theobalt C, Ahmed N, Seidel H, Thrun S (2008) Performance capture from sparse multi-view video. In: *ACM Transactions on Graphics (TOG)*. vol 27, p 98
9. Ballan L, Cortelazzo G (2008) Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: *3DPVT*, vol 37
10. Cagniard C, Boyer E, Ilic S (2010) Free-form mesh tracking: a patch-based approach. In: IEEE conference on computer vision pattern recognition, pp 1339–1346
11. Starck J, Hilton A (2007) Surface capture for performance based animation. *IEEE Comput Graph Appl* 27(3):21–31
12. Gall J, Stoll C, De Aguiar E, Theobalt C, Rosenhahn B, Seidel H (2009) Motion capture using joint skeleton tracking and surface estimation. In: IEEE conference on computer vision pattern recognition, pp 1746–1753
13. Kolb A, Barth E, Koch R, Larsen R (2010) Time-of-flight cameras in computer graphics. *Comput Graph Forum* 29:141–159
14. Liu Y, Stoll C, Gall J, Seidel HP, Theobalt C (2011) Markerless motion capture of interacting characters using multi-view image segmentation. In: IEEE conference on computer vision pattern recognition, pp 1249–1256
15. Liu Y, Gall J, Stoll C, Dai Q, Seidel HP, Theobalt C (2013) Markerless motion capture of multiple characters using multiview image segmentation. *IEEE Trans Pattern Anal Mach Intell* 35:2720–2735
16. Ye G, Liu Y, Hasler N, Ji X, Dai Q, Theobalt C (2012) Performance capture of interacting characters with handheld kinects. In: IEEE conference on computer vision ECCV. Springer, Berlin, pp 828–841
17. Ye G, Liu Y, Deng Y, Hasler N, Ji X, Dai Q, Theobalt C (2013) Free-viewpoint video of human actors using multiple handheld kinects. *IEEE T Cybern* 43:1370–1382
18. Moeslund TB, Hilton A, Krüger V (2006) A survey of advances in vision-based human motion capture and analysis. *Comput vis image underst* 104:90–126
19. Poppe R (2007) Vision-based human motion analysis: an overview. *Comput vis image underst* 108:4–18
20. Deutscher J, Blake A, Reid I (200) Articulated body motion capture by annealed particle filtering. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*, Vol 2, pp 126–133
21. Gall J, Rosenhahn B, Brox T, Seidel HP (2010) Optimization and filtering for human motion capture. *Int j comput vis* 87:75–92
22. Wu C, Varanasi K, Theobalt C (2012) Full body performance capture under uncontrolled and varying illumination: a shading-based approach. Springer, New York, pp 757–770
23. Wu C, Varanasi K, Liu Y, Seidel HP, Theobalt C (2011) Shading-based dynamic shape refinement from multi-view video under general illumination. In: IEEE international conference on computer vision (ICCV), pp 1108–1115

24. Li G, Wu C, Stoll C, Liu Y, Varanasi K, Dai Q, Theobalt C (2013) Capturing relightable human performances under general uncontrolled illumination. *Comput Graph Forum* 32:275–284
25. Hasler N, Rosenhahn B, Thormählen T, Wand M, Gall J, Seidel HP (2009) Markerless motion capture with unsynchronized moving cameras. In: *IEEE international conference on computer vision pattern recognition*, pp 224–231
26. Wu C, Stoll C, Valgaerts L, Theobalt C (2013) On-set performance capture of multiple actors with a stereo camera. *ACM Trans Graph (TOG)* 32:161
27. Wei X, Chai J (2010) Videomocap: modeling physically realistic human motion from monocular video sequences. *ACM Trans Graph (TOG)* 29:42
28. Han J, Shao L, Xu D, Shotton J (2013) Enhanced computer vision with microsoft kinect sensor: a review. *IEEE T Cybernet* 43:1318–1334
29. Shum HPH, Ho ESL, Jiang Y, Takagi S (2013) Real-time posture reconstruction for microsoft kinect. *IEEE T Cybernet* 43:1357–1369
30. Ni B, Pei Y, Moulin P, Yan S (2013) Multilevel depth and image fusion for human activity detection. *IEEE T Cybernet* 43:1383–1394
31. Baak A, Müller M, Bharaj G, Seidel HP, Theobalt C (2013) A data-driven approach for real-time full body pose reconstruction from a depth camera. In: *Consumer depth cameras for computer vision*. Springer, New York, pp 71–98
32. Shotton J, Fitzgibbon AW, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: *IEEE international conference on computer vision pattern recognition*, pp 1297–1304
33. Ganapathi V, Plagemann C, Koller D, Thrun S (2010) Real time motion capture using a single time-of-flight camera. In: *IEEE international conference on computer vision pattern recognition*, pp 755–762
34. Agarwal A, Triggs B (2004) 3d human pose from silhouettes by relevance vector regression. In: *Proceedings of the IEEE computer society conference on computer vision and Pattern Recognition*, vol 2, p 882
35. Ye M, Wang X, Yang R, Ren L, Pollefeys M (2011) Accurate 3d pose estimation from a single depth image. In: *IEEE international conference on computer vision*, pp 731–738
36. Taylor J, Shotton J, Sharp T, Fitzgibbon A (2012) The vitruvian manifold: inferring dense correspondences for one-shot human pose estimation. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, pp 103–110
37. Wei X, Zhang P, Chai J (2012) Accurate realtime full-body motion capture using a single depth camera. *ACM Trans Graph (TOG)* 31:188
38. Bouguet JY (2004) Camera calibration toolbox for matlab
39. OpenNI: (<http://www.openni.org/>)
40. Barmpoutis A (2013) Tensor body: real-time reconstruction of the human body and avatar synthesis from rgb-d. *IEEE T Cybernet* 43:1347–1356
41. Tong J, Zhou J, Liu L, Pan Z, Yan H (2012) Scanning 3d full human bodies using kinects. *IEEE Trans Vis Comput Graph* 18:643–650
42. Li H, Vouga E, Gudym A, Luo L, Barron JT, Gusev G (2013) 3d self-portraits. *ACM Trans Graph* 32:187
43. Aiger D, Mitra NJ, Cohen-Or D (2008) 4-points congruent sets for robust surface registration. *ACM Trans Graph* 27(85):1–10
44. Sorkine O (2006) Differential representations for mesh processing. *Comput Graph Forum* 25:789–807
45. OptiTrack: (<http://www.naturalpoint.com/optitrack/>)
46. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: *IEEE international conference on BMVC*, pp 1–11