# Performance Capture of Interacting Characters with Handheld Kinects

Genzhi Ye[1], Yebin Liu[1], Nils Hasler[2], Xiangyang Ji[1], Qionghai Dai[1], and Christian Theobalt[2]

[1] Department of Automation, Tsinghua Univercity, Beijing, China
[2] Max-Planck Institute for Informatics, Saarbrücken, Germany

**Abstract.** We present an algorithm for marker-less performance capture of interacting humans using only three hand-held Kinect cameras. Our method reconstructs human skeletal poses, deforming surface geometry and camera poses for every time step of the depth video. Skeletal configurations and camera poses are found by solving a joint energy minimization problem which optimizes the alignment of RGBZ data from all cameras, as well as the alignment of human shape templates to the Kinect data. The energy function is based on a combination of geometric correspondence finding, implicit scene segmentation, and correspondence finding using image features. Only the combination of geometric and photometric correspondences and the integration of human pose and camera pose estimation enables reliable performance capture with only three sensors. As opposed to previous performance capture methods, our algorithm succeeds on general uncontrolled indoor scenes with potentially dynamic background, and it succeeds even if the cameras are moving.

## 1 Introduction

In recent years, the field of marker-less motion estimation has seen great progress. Two important lines of research have 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 multi-view video, e.g., [1,2,3,4,5,6]. Even though measurement accuracy has greatly increased in the recent past, these approaches are still limited to largely controlled studio settings, and rely on static frame-synchronized multi-video systems comprising 10 or more cameras. Marker-less *performance capture* approaches take one step further and not only reconstruct a skeletal motion model but also detailed dynamic surface geometry as well as detailed texture, e.g. [7,8,9,10,11,12]. Unlike most skeletal motion capture methods, these approaches can also deal with people in general wide apparel. Recently, performance capture of multiple closely interacting people was also demonstrated [13]. However, performance capture approaches typically require even more strongly controlled studio setups, and often expect an even higher number of video cameras than marker-less motion capture approaches.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part II, LNCS 7573, pp. 828–841, 2012.

At the other end of the spectrum are methods for marker-less motion capture from a single camera view. Estimation of complex poses from a single video stream is still a very challenging task, in particular if interactive or real-time frame rates are the goal [6]. If additional depth data is available, e.g., obtained by stereo reconstruction, more complex poses and simple human shape representations can be fitted [14,15]. The recent advent of so-called depth cameras, such as time-of-flight sensors [16] 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. Using some form of articulated ICP or body part detection, skeletal poses can be reconstructed from data captured with a single depth sensor [17,18,19]. With such a depth-based local pose optimization scheme, tracking errors more frequently occur due to erroneous local convergence. Monocular depth-based body tracking with improved accuracy at near-real-time is feasible by combination of pose optimization with body part detection [20]. An alternative to model-based pose estimation is real-time joint detection from depth data using a learned decision forest [21]. Recently, this approach has been augmented with a regression method to enable the algorithm to localize joints also under occlusions [22]. As opposed to the model-based pose fitting approaches, these detection methods do not deliver joint angles. To obtain the latter an additional inverse kinematics step is required. Recently it was shown that through a combination of model-based pose optimization with pose detection full joint angles of even complex poses can be captured [23], also at high real-time frame rates [24].

So far it has been difficult to capture 3D models of a complexity and detail comparable to multi-view performance capture results using just a single depth camera. Weiss et al. [25] show an approach to fit a parametric human body model to depth data from a Kinect. However, their method can only capture static models of people in skintight clothing, and the motion of the person is not captured. To capture more detailed performance models one would need more complete, ideally 360 degree, depth data showing the scene from all sides. Earlier work on multi-view image-based 3D reconstruction already hinted at this. These approaches reconstructed dynamic geometry models and skeletal motion models of good quality by fitting a body template to a sequence of shape-from-silhouette reconstructions [26,27,28]. Building up on these ideas from the video domain, Berger et al. [29] use several Kinects to capture simple skeletal body poses based on a shape template.

In this paper, we show a method to do full performance capture of moving humans using just three hand-held, and thus potentially moving, Kinect cameras. Without resorting to any markers in the scene, it reconstructs detailed time-varying surface geometry of humans in general apparel, as well as the motion of the underlying skeleton. It can handle fast and complex motion with many self-occlusions and also captures non-rigid surface deformation, e.g., due to cloth motion. By resorting to depth sensors, our algorithm can be applied to more general uncontrolled indoor scenes and is not limited to studios with controlled

lighting and many stationary cameras. Also, our method requires only three hand-held sensors to produce results that rival reconstructions obtained with video-based performance capture methods that require a lot more cameras [13]. To enable this, we developed a method that **a)** tracks the motion of the hand-held cameras and aligns the RGBZ data, and **b)** that simultaneously aligns the surface and skeleton of each tracked performer to the captured RGBZ data. Our algorithm succeeds despite notable sensor noise, and is designed to be insensitive to multi-Kinect interference and occlusions in the scene. Our goal is thus related to the approach by Hasler et al. [30] which enables skeletal motion capture from hand-held video cameras. However, it goes beyond their method by being able to capture motion *and* non-rigid dynamic shape of more than one closely interacting performer from only three camera views.

Similar to the approach by Gall et al. [12] our algorithm deforms a template made of a skeleton and a deforming surface mesh for each performer into the captured RGBZ data. Our algorithm succeeds because of the interplay of several algorithmic contributions: We propose an efficient geometric 3D point-to-vertex assignment strategy to match the Kinect RGBZ 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, we detect and track SIFT features in the background part of each video frame. Based on these model-to-data assignment criteria, we jointly estimate the pose parameters of the performers and the poses and orientations of the Kinects in a combined optimization framework. Our non-linear objective function can be linearized and effectively minimized through a quasi-Newton method. As we will show, the integration of geometry-based and video-based correspondence estimation is crucial in our setting, as only in this way combined reconstruction of human animation models and camera poses is feasible even in scenes with occlusions, notable sensor noise and moving background. We show results on several challenging single and multi- person motions including dance and martial arts. We also quantitatively prove the accuracy of our reconstruction method by comparing it to video-based performance capture.

## 2 Data Capture with Handheld Kinects

Our algorithm enables the reconstruction of detailed skeleton models of humans, as well as their deforming surface geometry, in general indoor environments. The method does not expect controlled, i.e., it can handle non-static background and general non-studio lighting. In a typical recording situation, 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 towards the center of the recording volume, Fig. 1(a). Thee operators are free to move the cameras during recording, for instance to get a better view of the action with less occlusions. Our performance capture method can also handle a certain amount of moving scene elements in the background.

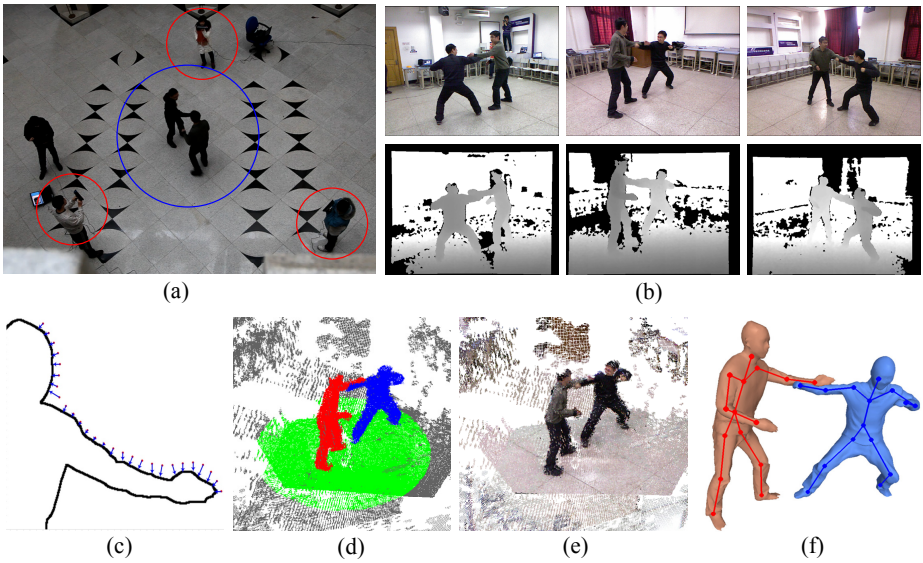


**Fig. 1.** Overview of the processing pipeline. (a) overhead view of typical recording setup: three camera operators (circled in red) film the moving people in the center (blue); (b) input to the algorithm - RGB images and the depth images from three views; (c) 2D illustration of geometric matching of Kinect points to model vertices (Sect. 4.2); (d) segmented RGBZ point cloud - color labels correspond to background, ground plane (green) and interacting humans (red,blue); (e) Registered RGBZ point cloud from all cameras; (f) reconstructed surface models and skeletons.

Exact hardware synchronization of multiple Kinects is impossible, and we thus resort to software synchronization. We connect each Kinect 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 our software solution the captured data of all cameras are frame-synchronized with at most 10ms temporal difference.

The Kinect features two sensors in the same housing, a RGB video camera and an active stereo system to capture depth. The intrinsics of both the depth and the video cameras are calibrated off-line using a checkerboard [31]. Depth and color data are aligned with each other using the OpenNI API[1]

At every time step of video $t$, each Kinect captures a $640 \times 480$ video frame and an aligned depth frame, Fig. 1(b), which yields a combined RGBZ point cloud. For each such RGBZ 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

[1] `http://www.openni.org/`.

respect to each camera's local coordinate system. For performance capture, we need to align the points from all cameras into a global system. Since the Kinects are allowed to move in our setting, we also need to solve for the extrinsic camera parameters $\Lambda_c^t$ (position and orientation) of each Kinect $c$ at every time step of video $t$, i.e., solve for the combined extrinsic set $\Lambda^t = \{\Lambda_c^t\}_{c=1}^C$. Fig.1(e) shows the merged point set at time $t$ after solving for the extrinsics using the method later described in this paper. 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 needs to be robust against such missing measurements.

## 3 Scene Models

For each of the $k = 1, \ldots, K$ performers in the scene, a template model is defined. Similar to [12,13] a template model comprises a surface mesh $M_k$ with an embedded kinematic bone skeleton (see Fig. 1(f)). Similar to other model-based performance capture work, we use a laser scanner 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. We remesh the surface models, such that they have around $N_k = 5000$ vertices. Each vertex is also assigned a color that can change over time as described in Sec. 4.5. 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, we also explicitly model the ground plane as a planar mesh $V_0^t$ with circular boundary. The ground plane model has a fixed radius of 3m and during initialization is centered below the combined center of gravity of the human models (see Fig.1(d)). 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 $n = 31$ degrees of freedom is manually placed into each human mesh and surface skinning weighs are computed using a similar process as [12,13]. Skeleton poses $\chi^t = (\xi^t, \Theta^t) = \left(\theta_0\hat{\xi}, \theta_1, ..., \theta_n\right)$ are parameterized using the twist and exponential maps parameterization [2,12]. $\theta_0\hat{\xi}$ is the twist for the global rigid body transform of the skeleton and $\Theta^t$ is the vector of the remaining joint angles. Using linear blend skinning, the configuration of a vertex of human mesh $M_k$ in skeleton pose $\chi_k^t$ is then determined by

$$v_i\left(\chi_k^t\right) = \sum_{m=1}^{n} \left( w_i^m \prod_{j=0}^{j_m} \exp\left(\theta_{\psi_m(j)}\hat{\xi}_{\psi_m(j)}\right) \right) v_i. \tag{1}$$

Here, $w_i^m$ is the skinning weight of vertex $i$ with respect to bone $m$. Further on, $j_m$ is the number of joints in the kinematic chain that influence bone $b_m$, and $\psi_m(j)$ determines the index of the $jth$ of these joints in the overall skeleton

configuration. In addition to $\Lambda^t = \{\Lambda_c^t\}_{c=1}^C$, our performance capture approach thus needs to solve for the joint parameters of all persons at each time step, $\mathrm{X^t} = \{\chi_k^t\}_{k=1}^K$.

## 4 Simultaneous Performance Capture and Kinect Tracking

Performance capture from 3D point data is only feasible if the RGBZ data from all Kinects are correctly registered. In the beginning, for each time step the correct extrinsics $\Lambda_t$ are unknown. A traditional approach to track camera extrinsics is structure-from-motion (SfM) performed on the background of the sequence [30]. However, in our recording setting, the moving subjects fill most of the visible area in each video frame. Thus a different approach has to be used. In our 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 RGBZ data. At the same time, we exploit feature correspondences in the background similarly to SfM since they provide addtional evidence for correct reconstruction. We therefore simultaneously solve for camera and body poses, and regularize the solution to additional feature correspondences found in the video frame.

### 4.1 Overview

In the first frame of multi-view RGBZ 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. 1(c-f) to each time step. For a time step $t$ the steps are as follows: we first align the Kinect RGBZ point clouds according to the the extrinsics $\Lambda^{t-1}$ 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 RGBZ data are also implicitly segmented into classes for *ground plane*, *background* and one class for each *person*, Sect. 4.2 (Fig. 1(d)). Thereafter, a second set of 3D correspondences is found by matching points from the *ground plane* and the *background* via SIFT features, Sect. 4.3. Based on these correspondences, we simultaneously solve for Kinect and skeleton poses for the current frame, Sect. 4.4. Correspondence finding and reconstruction is iterated several times and the model poses and point cloud alignments are continuously updated (Fig. 1(e)). 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 we thus non-rigidly deform the meshes $M_k$ into the aligned point clouds via Laplacian deformation and update the vertex colors of the mesh model(s) (Fig. 1(f)). In the following, we explain each step for a specific time $t$ and omit the index $t$ for legibility.

### 4.2 3D Point Correspondences and Implicit RGBZ Point Cloud Segmentation

As stated above, for finding correct camera and body configurations, we will minimize an error function that measures the alignment of the RGBZ point clouds with the 3D human models, Sect. 4.4. To evaluate this error, for all scene model vertices $V$, plausible correspondences to the RGBZ points $P$ need to be defined. With these correspondences, we would be able to evaluate an alignment error, as it was also used in video-based performance capture to measure alignment in the image domain [12].

Our matching term ensures that as much as possible of each 3D human template is explained by the point data. Unfortunately, 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. In contrast, 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 RGBZ points are proposed as matches. Our results show that using this approach tracking is robust even with two performers in the scene.

To this end, we define a distance measure $F$ between Kinect points $p$ and model vertices $v$ that simultaneously measures a color distance and a geometric distance as follows:

$$F(v,p) = \Delta\left(\|l_v - l_p\|, \theta_l\right) \Delta\left(\|x_v - x_p\|, \theta_x\right) \max\left(n_v n_p, 0\right) \tag{2}$$

where

$$\Delta(x,\theta) = \max\left(1 - \frac{x}{\theta}, 0\right) \tag{3}$$

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 color term enforces color similarity between the mesh vertex and the corresponding Kinect point, the geometry term only matches RGBZ points and vertices that are spatially close and have similar normal orientation. We experimentally choose the maximum color difference $\theta_l = 100$ and the maximum distance a mesh vertex is allowed to move $\theta_x = 100mm$.

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

### 4.3 Feature-Based Background Correspondences

As stated in the beginning, our reconstruction error is also based on feature correspondences in the scene background, similar to classical structure-from-motion

approaches. The method from Sect. 4.2 provides a classification of background RGBZ points, and thus corresponding RGB pixels in each Kinect video image. We detect SIFT features on the background regions of the RGB images from $t-1$ and $t$, and convert them into 3D correspondences $Z_s = \{(p', p)\} \mid p' \in P^{t-1}, p \in P^t, (p', p)$ 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 our error function additionally measures point-to-model correspondences in the foreground. Fig.2(b) shows that alignment based on SIFT features in the background alone will not suffice.

### 4.4 Optimization of Skeleton and Camera Poses

Given correspondence sets $Z_{pv}$, $Z_{pg}$, and $Z_s$ we can define a geometric error function that we minimize in the space of skeleton pose X and camera extrinsics $\Lambda$:

$$E(\mathrm{X}, \Lambda) = \underset{\mathrm{X},\Lambda}{\arg\min} \left\{ \sum_{(p,v)\in Z_{pv}} \frac{\|p(\Lambda) - v(\mathrm{X})\|^2}{\|Z_{pv}\|} + \sum_{(p,v)\in Z_{pg}} \frac{\|p(\Lambda) - v\|^2}{\|Z_{pg}\|} + \sum_{(p,p')\in Z_s} \frac{\left\|p(\Lambda) - p'\right\|^2}{\|Z_s\|} \right\} \quad (4)$$

Here $||Z||$ is the number of elements in set $Z$. We solve this function through linearization within an iterative quasi-Newton minimization. Using Taylor expansion of the exponential map, the transformation of $\Lambda$ on point $p$ leads to

$$p(\Lambda) = Rp + T = e^{\theta\hat{\xi}} p \approx \left(I + \theta\hat{\xi}\right) p \quad (5)$$

For the body pose a similar expansion can be performed.

We iterate robust correspondence finding and skeleton-camera optimization 20 times. After each iteration, the position and normal of each point $p$ is updated according to the new $\Lambda$, while the skeletons and model vertices are updated according to X. Fig.2 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 [32].

### 4.5 Non-rigid Mesh Deformation and Vertex Color Update

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:

$$\arg\min_{\mathbf{v}} \{\|L\mathbf{v} - \delta\|_2^2 + \|\mathbf{C}\mathbf{v} - \mathbf{p}\|_2^2\} \quad (6)$$

Here, $\mathbf{v}$ denotes the vector of vertices on human body mesh $M_k$. $L$ is the cotangent Laplacian matrix and $\delta$ is the differential coordinates of the current mesh

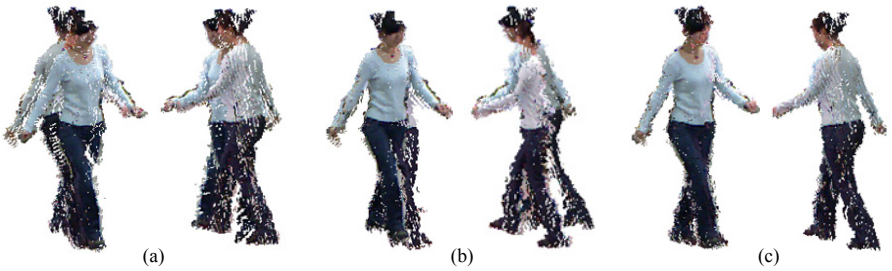


**Fig. 2.** Comparison of RGBZ point data fusion at frame $t$ before and after joint skeleton and Kinect optimization. (a) Fusion using extrinsics from the former time; (b) Fusion based on SIFT features alone fails; (c) Fusion using extrinsics solved by the combined human and camera pose optimization produces much better results.

vertices. $C$ is a diagonal matrix with non-zero entries $c_{jj} = \alpha$ ($\alpha$=0.1) for vertices in correspondence set $Z_{pv}^k$. $\mathbf{p}$ is the vector with non-zero position entries for those $p$ in $Z_{pv}^k$. 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

$$l_v = \frac{t}{t+1}\widehat{l}_v + \frac{1}{t+1}l_{nn} \tag{7}$$

where $\hat{l}_v$ is the color of $v$ before the update and $l_{nn}$ is the color of the nearest RGBZ neighbor point of $v$.

## 5 Results

We recorded 8 test sequences consisting of over 2500 frames. The data was recorded with 3 moving Kinects at a resolution of 640 × 480 pixels and at a framerate of 30fps. The sequences consist of a wide range of different motions, including dancing, fighting and jumping, see Fig. 5 and the accompanying video. The motions were performed by 5 different persons wearing casual clothing. We also recorded two evaluation sequences where the performer was simultaneously tracked by a multi-view video system. A quantitative evaluation using these data sets is performed in Sect. 5.1. Table 1 shows the capture properties of each sequence.

### 5.1 Comparison with Multi-view Video Based Tracking

We recorded two sequences captured in a multi-view video studio with 10 calibrated cameras (15fps, 1024 × 768) and background green screen. 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 sub-frame accuracy,

**Table 1.** Description of the capture sequences

| Sequence | Frame rate | Number of performers ($K$) | Number of Kinects ($C$) | Kinect status | Comparison with multi-view video |
|---|---|---|---|---|---|
| Dancing-walk | 30 | 1 | 3 | Moving | No |
| Kungfu | 30 | 1 | 3 | Moving | No |
| Couple-dance | 30 | 2 | 3 | Moving | No |
| Fight | 30 | 2 | 3 | Moving | No |
| Hug | 30 | 2 | 3 | Moving | No |
| Arm-crossing | 30 | 1 | 3 | Static | No |
| Rolling | 15 | 1 | 3 | Static | Yes |
| Jump | 15 | 1 | 3 | Static | Yes |

the evaluation of the difference between the two results will still give us a conservative performance evaluation of the proposed algorithm.

Since our multi-view video system runs at 15fps, we captured a sequence "Rolling" with slow motion and a sequence "Jump" with fast motion. The Kinect system runs at 30fps, so we subsample the frames for the Kinect based tracking by factor two and compare the performance with multi-view video based tracking (MVT) [12]. We visually compared the results of the two systems by evenly select 4 frames from the "Rolling" sequence (see Fig. 3). Since the MVT tracking requires green screen for clean background subtraction and it thus does not work with extra camera operators in the scene background, we fix the three Kinects in the MVT studio during data capture. With these fixed Kinects, we can validate the proposed algorithm by comparing the optimized Kinect extrinsics in the later frames with that 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 10.66mm, 7.28mm and 6.67mm, respectively. For the slow motion sequence "Rolling", our result closely matches the input images and the result of the MVT system, see Fig. 3. In addition, we quantitatively compare the two results by measuring the differences on the joint centers. The average distance between the corresponding joint positions across all 200 frames of the sequence is 21.42mm with a standard deviation of 27.49mm. 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.4.

### 5.2 Qualitative Evaluation

Our approach enables us to fully-automatically reconstruct 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 and the accompanying video. Despite notable noise in the captured depth maps, our method successfully captures pose and deforming surface geometry of persons in loose apparel. With a cap-

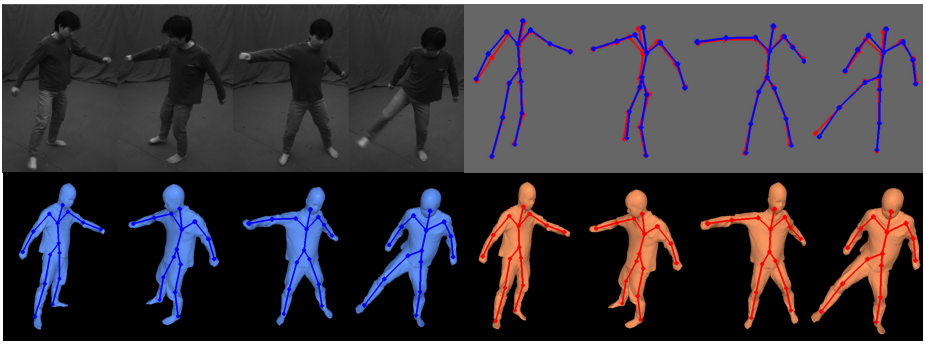

**Fig. 3.** Comparison with multi-view video tracking (MVT) approach on the "Rolling" sequence. The top left are four input images of the multi-view video sequence. The top right shows the close overlap of the two skeletons tracked with MVT (blue) and our Kinect-based approach (red). The bottom left is the reconstructed surface with the skeleton using MVT and the bottom right is the results from our approach. Quantitative and visual comparisons show that MVT-based and Kinect-based reconstructions are very similar.

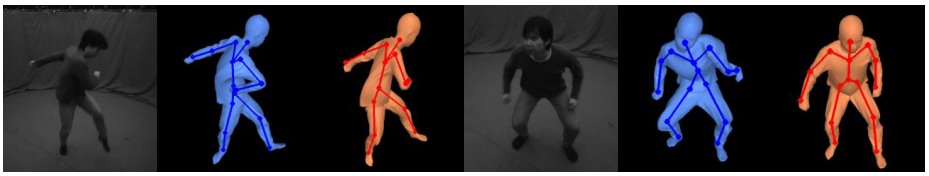

**Fig. 4.** Comparison with multi-view video tracking (MVT) approach on the "Jump" sequence. The left three and the right three are: input image, result of MVT, result of our Kinect-based approach. On this fast motion Kinect-based tracking succeeds while MVT fails to capture the arm motion.

turing frame rate of only 30fps, the proposed approach can also handle very fast motions, see the jump and kicking motions in Fig. 5. Our system uses local optimization and the complexity of the system mainly depends on the number of captured points. Computational complexity does not starkly depend on the number of subjects in the scene. It takes about 10 seconds for single person tracking of a frame and 12 seconds for the two person tracking on a standard PC using unoptimized code.

### 5.3 Discussion

We presented a method to capture shape and skeletal motion of one or two characters in fast motion using three handheld depth cameras. Reconstruction accuracy and quality is comparable to multi-view video-based approaches, but our method comes with the additional benefit that it applies also to less controlled

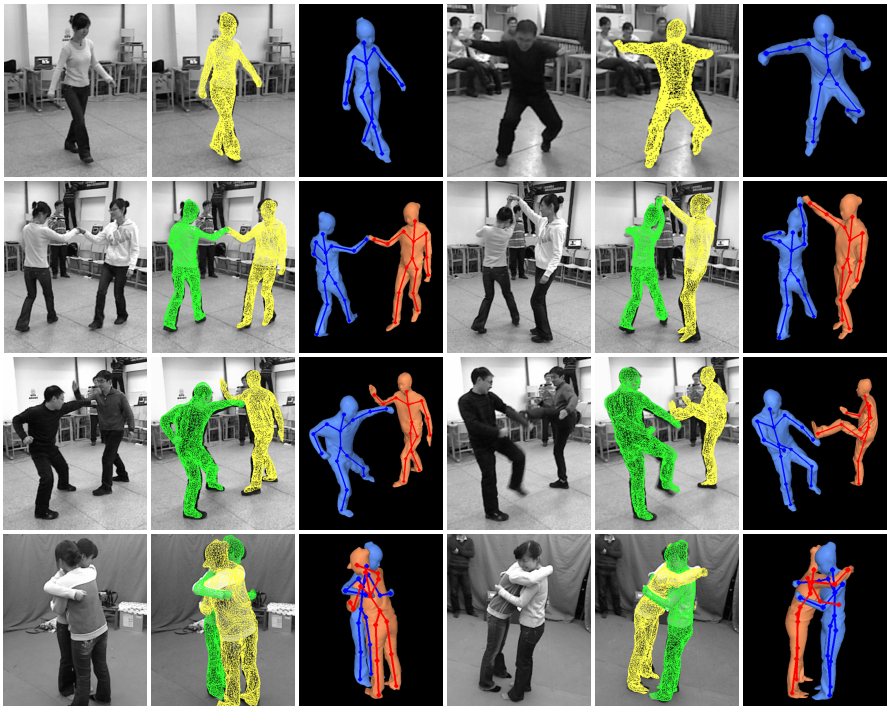

**Fig. 5.** Performance capture results on a variety of sequences: input image, deformed mesh overlay, and 3D model with estimated skeleton respectively

indoor environments. A key element to the method's success is the integrated estimation of camera and model poses based on geometric and photometric correspondences (see also Fig. 2).

Currently, our approach is designed for setup with multiple synchronized Kinects. An extension to un-synchronized handheld Kinects is feasible and will be investigated in the future. Currently, we do not explicity filter sensor noise and we do not explicitly model multi-Kinect interferences. We will further investigate this in the future. Since the Kinect data is noisy, the mesh deformation is not as effective as the one used in multi-view video systems for the reconstruction of surface details. Noise in the data may thus transfer into the capture meshes. The correspondence finding is implemented as a dense matching between measured points and vertices. More robust deformation in the presence of noise may be achieved by a noise-aware sparse matching algorithm. Similar to multi-view video systems, the tracking of joints on the shoulder may sometimes go error for fast and complex hand motions (e.g., "couple-dancing" sequence in the accompany video). This problem can be solved by adding physical constraints on the shoulders. Since skeleton and Kinect pose optimization can be linearized efficiently, it is also promising to investigate realtime applications in the future. We believe that with the advancement of depth camera techniques, improvement of sensor

resolution, sensor quality, portability, and compatibility (between Kinects) will be achieved to allow the efficient production of 3D content in everyday life.

## 6 Conclusion

In this paper, we proposed a method for simultaneously marker-less performance capture and camera pose estimation with several hand-held 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. 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, we allow users to hold the Kinects for motion capture and 3D reconstruction of performers. This improves the consumer experience especially with respect to the anticipated introduction of depth cameras in consumer devices like tablets. To our knowledge, this is the first method to fully-automatically perform multi-person performance capture using moving Kinects.

**Acknowledgments.** This work was supported by the National Basic Research Project (No.2010CB731800) and the Project of NSFC (No.60933006 and No.61073072).

## References

1. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR, pp. 1144–1149 (2000)
2. Bregler, C., Malik, J., Pullen, K.: Twist based acquisition and tracking of animal and human kinematics. IJCV 56, 179–194 (2004)
3. Sigal, L., Black, M.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University (2006)
4. Balan, A., Sigal, L., Black, M., Davis, J., Haussecker, H.: Detailed human shape and pose from images. In: CVPR (2007)
5. Stoll, C., Hasler, N., Gall, J., Seidel, H.P., Theobalt, C.: Fast articulated motion tracking using a sums of gaussians body model. In: ICCV, pp. 951–958 (2011)
6. Poppe, R.: Vision-based human motion analysis: An overview. CVIU 108, 4–18 (2007)
7. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. ACM Trans. Graph. 27, 1–9 (2008)
8. De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H., Thrun, S.: Performance capture from sparse multi-view video. In: ACM Transactions on Graphics (TOG), vol. 27, Article 98. ACM (2008)
9. Ballan, L., Cortelazzo, G.: Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: 3DPVT (2008)
10. Cagniart, C., Boyer, E., Ilic, S.: Free-form mesh tracking: A patch-based approach. In: CVPR, pp. 1339–1346 (2010)

11. Starck, J., Hilton, A.: Surface capture for performance based animation. IEEE Computer Graphics and Applications 27(3), 21–31 (2007)
12. Gall, J., Stoll, C., De Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.: Motion capture using joint skeleton tracking and surface estimation. In: CVPR, pp. 1746–1753 (2009)
13. Liu, Y., Stoll, C., Gall, J., Seidel, H.P., Theobalt, C.: Markerless motion capture of interacting characters using multi-view image segmentation. In: CVPR, pp. 1249–1256 (2011)
14. Friborg, R., Hauberg, S., Erleben, K.: GPU accelerated likelihoods for stereo-based articulated tracking. In: ECCV Workshops, CVGPU (2010)
15. Plankers, R., Fua, P.: Articulated soft objects for multiview shape and motion capture. TPAMI 25, 1182–1187 (2003)
16. Kolb, A., Barth, E., Koch, R., Larsen, R.: Time-of-flight cameras in computer graphics. Comput. Graph. Forum 29, 141–159 (2010)
17. Knoop, S., Vacek, S., Dillmann, R.: Fusion of 2D and 3D sensor data for articulated body tracking. Robotics and Autonomous Systems 57, 321–329 (2009)
18. Zhu, Y., Dariush, B., Fujimura, K.: Kinematic self retargeting: A framework for human pose estimation. CVIU 114, 1362–1375 (2010)
19. Pekelny, Y., Gotsman, C.: Articulated object reconstruction and markerless motion capture from depth video. CGF 27, 399–408 (2008)
20. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real time motion capture using a single time-of-flight camera. In: CVPR, pp. 755–762 (2010)
21. Shotton, J., Fitzgibbon, A.W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR, pp. 1297–1304 (2011)
22. Girshick, R., Shotton, A., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. In: ICCV (2011)
23. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3d pose estimation from a single depth image. In: ICCV, pp. 731–738 (2011)
24. Baak, A., Müller, M., Bharaj, G., Seidel, H.P., Theobalt, C.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: ICCV, pp. 1092–1099 (2011)
25. Weiss, A., Hirshberg, D., Black, M.J.: Home 3d body scans from noisy image and range data. In: ICCV, pp. 1951–1958 (2011)
26. Cheung, K., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3D voxel reconstruction of human motions. In: CVPR, pp. 714–720 (2000)
27. Horaud, R., Niskanen, M., Dewaele, G., Boyer, E.: Human motion tracking by registering an articulated surface to 3d points and normals. TPAMI 31, 158–163 (2009)
28. Corazza, S., Mündermann, L., Gambaretto, E., Ferrigno, G., Andriacchi, T.P.: Markerless motion capture through visual hull, articulated icp and subject specific model generation. IJCV 87, 156–169 (2010)
29. Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A., Magnor, M.A.: Markerless motion capture using multiple color-depth sensors. In: VMV, pp. 317–324 (2011)
30. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR, pp. 224–231 (2009)
31. Bouguet, J.Y.: (Camera calibration toolbox for matlab)
32. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust surface registration. ACM Transactions on Graphics 27, #85, 1–10 (2008)