Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Combining Marker-based Mocap and RGB-D Camera for Acquiring High-fidelity Hand Motion Data

Wenping Zhao[1] Jinxiang Chai[2] Ying-Qing Xu[3]

[1]University of Science and Technology of China [2]Texas A&M University [3]Tsinghua University

Abstract

Motion capture data has been pivotal to the success of creating realistic animation for human characters. There are a number of public full-body motion databases available, but large and heterogeneous databases for hand articulations are not available. In this paper, we introduce a novel acquisition framework for acquiring a wide range of high-fidelity hand motion data. Our key idea is to leverage marker position data recorded by a twelve-camera optical motion capture system and RGB/Depth data obtained from a single Microsoft *Kinect camera. We formulate the hand motion capture problem in a nonlinear optimization framework by maximizing consistency between the reconstructed motion and observed measurement. We introduce an efficient optimization technique to estimate the optimal hand pose that best matches observed data. We have demonstrated the power and effectiveness of our system by capturing a wide variety of delicate hand articulations, even in case of significant self-occlusion.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

One of fundamental challenges in computer graphics has been the realistic animation of the human hand. One way to address this challenge is data driven approaches, where sets of example motions are available for editing, retargeting, interpolation and composition. However, acquiring high-fidelity hand motion data is difficult because it requires modeling delicate hand articulations in a high-dimensional space (usually higher than 25 degrees of freedom).

Decades of research in computer graphics have explored a number of approaches for capturing articulated hand motion data, including marker-based motion capture, glove-based systems, and image-based systems. Despite the efforts, acquiring high-fidelity hand motion data remains a challenging task. For example, marker-based motion capture systems such as Vicon [Vic] often produce ambiguous solutions because of significant self-occlusion caused by cameras. Glove-based systems such as CyberGlove [Cyb] are occlusion-free but recorded motion data is often noisy and fails to capture delicate hand articulations with high precision. Glove-based systems are also cumbersome and unwieldy, thereby impeding the subject's ability to act out the motion. Image-based systems offer an appealing alternative to hand motion capture because they require no markers, no gloves, or no sensors. However, current image-based systems are often vulnerable to ambiguity caused by significant self-occlusion and a lack of discernible features on a hand.

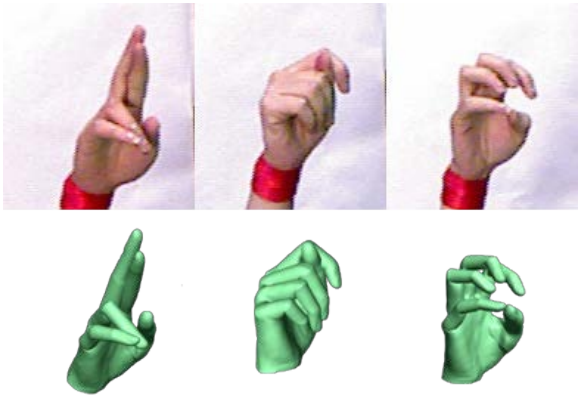


Figure 1: *Our system automatically captures high-fidelity hand motion data by combining marker position data and RGB-D image data. (top) reference image data; (bottom) the reconstructed poses.*

DOI: 10.2312/SCA/SCA12/033-042

The primary contribution of this paper is to introduce a novel acquisition framework for acquiring high-fidelity hand motion data (see Figure 1). Our key idea is to leverage high-resolution marker position data recorded by a twelve-camera optical motion capture system [Vic] and RGB/Depth (RGB-D) image data obtained by a single *Microsoft* Kinect camera. We choose the Kinect camera for hand performance acquisition because it is low-cost and portable and requires no controlled illumination. More importantly, the two capturing devices are complementary to each other as they focus on different aspects of hand performances. On one hand, marker-based motion capture systems can obtain high-resolution 3D position data at very high frame rates (up to 2000 Hz) but due to their low spatial resolution (usually 20 to 30 markers) they are often not capable of reconstructing 3D hand articulations accurately, particularly in case of significant self-occlusion. On the other end, a RGB-D camera such as Kinect can capture per-pixel information for both color and depth data. However, the data from the Kinect camera is often noisy and sampled at a much lower frame rate.

We formulate the hand mocap problem in a nonlinear optimization framework by maximizing consistency between the reconstructed motion and observed measurement, including both 3D marker position data from the marker-based system and RGB-D image data from the Kinect camera. We introduce an efficient optimization technique to estimate the optimal hand pose that best matches observed data. We have demonstrated the power and effectiveness of our system by capturing a wide range of complex hand motion data, including everyday hand gestures, American Sign Languages (ASL), and hand grasping and manipulation.

Currently there are a number of public full-body motion capture databases available, such as [cmub] and [MRC*07], which have been pivotal in the development of data-driven approaches for full-body motion modeling, synthesis and control. But this is not the case for hand. The widely popular *Columbia* grasp database [GCDA09] includes a large dataset of static grasping poses, which are created by automatic grasp planning process instead of real world measurement. One exception is the *CMU CyberGlove Database* [cmua], which contains a number of hand grasping motions captured by CyberGlove systems. However, motion data obtained from CyberGlove systems is often noisy and might not be able to capture highly detailed and accurate hand articulations. We hope our proposed work can contribute to the creation of a comprehensive high-quality hand motion capture database and can stimulate research in data-driven approaches for hand motion modeling, synthesis and control.

2. Related work

Our system combines a marker-based motion capture system and RGB-D image data for acquiring high-fidelity hand motion data. We therefore focus our discussion in related technologies for hand motion acquisition.

Marker-based Motion Capture. A popular approach to hand motion capture is to use marker-based motion capture systems [Vic], which track a sparse set of retro-reflective markers (usually 20 to 30 markers) attached on hand and use recorded marker position data to reconstruct 3D hand poses across an entire sequence. Recent technological advances in motion capture equipment have made it possible to acquire 3D motion data with stunningly high temporal resolution (up to 2000 Hz), but due to significant self-occlusion they are often not capable of acquiring high-fidelity hand motion data consistently. Another challenge for marker-based motion capture systems is how to build temporal correspondences for all visible markers over time. Automatic marker labeling, particularly for hand capture, remains challenging while manually annotating markers over time is not only time consuming but also error prone.

Several researchers have recently explored how to simplify the problem by reducing the markers used. For example, Hovet and his colleagues [HRMO12] reconstructed 3D hand poses from a small number of markers via inverse kinematics techniques, while keeping the perceived fidelity. However, it is not clear if their technique can be applied to acquiring a wide range of high-fidelity hand motion data shown in this paper, particularly when significant self-occlusion occurs. An alternative is to use prior knowledge embedded in a prerecord motion capture database to reduce the number of markers required for articulated motion reconstruction [CH05]. However, unlike full-body motion capture, hand motion databases for detailed motions of the hand were not available.

We propose to complement a marker-based mocap system with RGB-D data obtained by a Kinect camera. This combination significantly reduces ambiguity for 3D pose reconstruction and enables us to capture hand motion even in case of significant self-occlusion. In addition, our system is fully automatic and requires no tedious manual user intervention such as marker labeling and gap filling.

Image-based Motion Capture. One way to capture 3D hand motion from images is model-based motion tracking (*e.g.* [dLGFP11]), which initializes the 3D hand pose at the starting frame and sequentially updates the 3D poses by minimizing inconsistency between the hypothesized and observed data. The approach, however, is vulnerable to ambiguities in image data caused by self-occlusion as well as a lack of discernible features on hand. In practice, image observations alone are often noisy and insufficient to capture high-fidelity 3D hand articulations.

An efficient way to reduce ambiguity is to utilize motion priors embedded in prerecorded hand motion database. Thus far, two different approaches have been taken, including generative approaches [WLH01, ZH03] and discriminative approaches [AS03, WP09, RKK10]. However, data-driven approaches can only acquire motions that are similar to train-

ing datasets, thereby significantly limiting their application to hand motion capture.

Recently, Oikonomidis and his colleagues [OKA11] sequentially estimated one-hand pose configurations by finding 3D hand poses that best match observations provided by a RGB-D camera. Combining color and depth images for hand tracking improves the accuracy of the tracking process. However, as shown in our comparison experiment, the solution might still be ambiguous when significant self-occlusion occurs. We address the challenge by complementing the RGB-D camera with a marker-based motion capture system, which enables us to accurately reconstruct hand poses even in case of significant self-occlusion.

Glove-based Systems. Thus far, one of the the most reliable options for hand capture has been glove-based devices because they are occlusion-free [Cyb, cmua]. For example, the recent wireless CyberGlove II system [Cyb] can provide up to 22 joint-angle measurements. However, glove-based systems suffer from several limitations. First and foremost, the recorded data is often noisy and fails to capture delicate hand articulations with high precision. Second, they require extra devices to capture absolute 3D positions of the hand. Lastly, glove-based systems are cumbersome and unwieldy, thereby impeding the subject's ability to perform the motion.

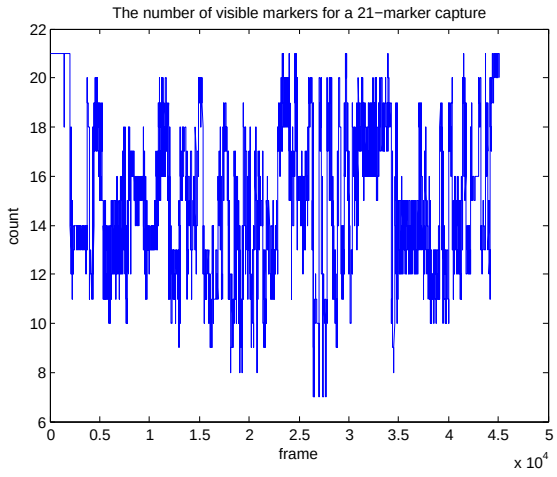


Figure 2: *Incomplete marker information caused by significant self-occlusion. The vertical axis shows the number of visible markers over time for a motion sequence of 45,092 frames (about 6.2 minutes) captured by a twelve* Vicon *camera system. On average, about 15 markers out of 21 markers are visible for each frame. To capture hand motion data, tedious work is required to manually label the markers and fill the gap for missing markers over the whole motion sequence.*

3. Motivations and Overview

This section briefly describes our idea of combing a marker-based mocap system such as *Vicon* with a single RGB-D camera such as *Kinect* for high-fidelity hand motion capture. Our idea is motivated by the current challenge in the application of marker-based motion capture systems to hand motion acquisition.

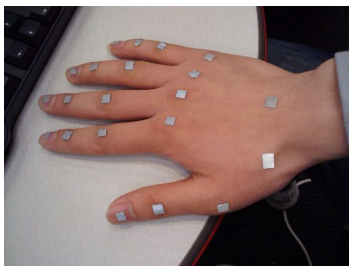


Figure 3: *Marker placement for hand motion capture. In total, 21 markers are used in our experiment.*

A major challenge for hand motion capture is incomplete marker information caused by occlusion from Vicon cameras. Figure 2 illustrates our concern about missing markers for hand motion capture. In our experiment, we attached 21 markers on hand (For details, see Figure 3) and used a twelve *Vicon* camera system to capture 3D hand performances for a wide range of hand gestures. Figure 2 shows the number of visible markers over time for a motion sequence of *45,092* frames (about *6.2* minutes). On average, about 15 markers are visible for each frame. For some frames, the number of visible markers is even lower than 8. Thus tedious work is required to manually label the markers and fill the gaps for missing markers over the whole motion sequence. In practice, it is almost impossible to obtain high-quality motion data with manual annotation and intervention because many markers might miss completely for long periods of time.

We solve the problem by complementing the marker-based mocap system with RGB-D image data obtained from a single Kinect camera. We choose the *Microsoft* Kinect camera for hand performance acquisition because it is low-cost, portable and non-intrusive. The Kinect camera allows us to simultaneously capture depth data with a resolution of 320×240 and color image data with a resolution of 640×480 at 30 frames per second based on infrared projection. We hypothesize that the information lost by missing markers can be filled by per-pixel color and depth information obtained from the Kinect camera. We develop a robust acquisition framework that seamlessly combines marker position data and RGB-D image data for 3D hand capture. Our system is fully automatic and does not require tedious marker labeling and gap filling. In addition, our method can acquire high-fidelity hand articulation data in the same frame rate as the marker-based motion capture system.

We formulate the motion capture problem in a nonlinear optimization framework by minimizing inconsistency be-

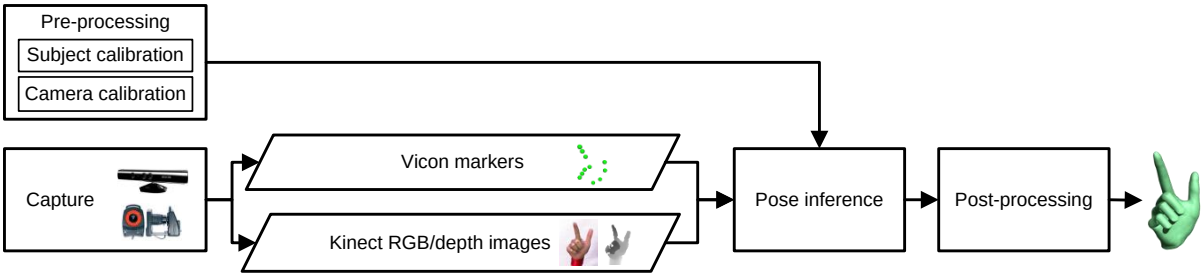


Figure 4: *System overview.*

tween the reconstructed motion M and the observed data O:

$$\arg\min_{M} E(M,O) \tag{1}$$

where the reconstructed motion $M = [\mathbf{q}_0,...,\mathbf{q}_T]$ specifies hand poses $\mathbf{q}_t$ over time. The observed data $O = [\mathbf{o}_0,...,\mathbf{o}_T]$ includes both marker position data from *Vicon* and RGB-D image data obtained by the *Kinect* camera.

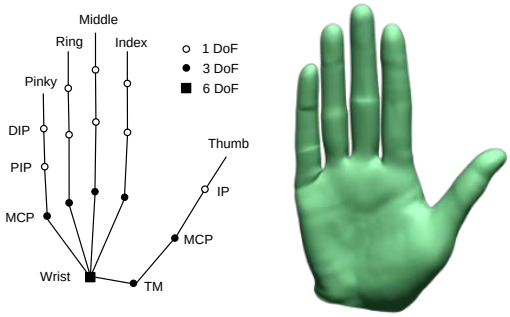
An overview of our method is shown in Figure 4. We begin with the calibration process, including subject calibration and camera calibration (Section 4). Subject calibration estimates both hand skeletal sizes and local coordinates of each marker attached on hand. Subject calibration ensures that our motion capture process is robust to hands of different skeletal sizes and to variations in marker placement. Camera calibration estimates both intrinsic and extrinsic parameters of the Kinect camera, which allows us to transform marker position data from the marker-based mocap system into the Kinect system. Both calibration processes are done offline. At run time, the system records both 3D marker position data and RGB-D image data of hand performances. We formulate the 3D hand pose reconstruction as a nonlinear optimization problem and reconstruct 3D poses via an efficient optimization process (Section 5). Finally we run a post-processing step to enhance the reconstructed motion to match the frame rate of the marker-based mocap system (Section 6).

We demonstrate the effectiveness of our method by capturing a wide range of sophisticated hand articulations. In addition, we compare against pose inferences using only marker position data or RGB-D image data to show the necessity of combination (Section 7).

4. Pre-processing

Our pre-processing step consists of two components: subject calibration and camera calibration, which will be described separately in the following.

Subject calibration. We describe a hand pose using a set of independent joint coordinates $\mathbf{q} \in R^{33}$, including absolute root position and orientation as well as the relative joint angles of individual joints. Figure 5(a) shows our articulated hand skeletal model. For Distal Interphalangeal (DIP) and Proximal Interphalangeal(PIP) joint, we use 1 Degree of Freedom (DoF) to describe their movement. We choose to model Metacarpophalangeal (MCP) joints using a ball and socket joint, which has 3 DoF instead of the 2 DoF in most literature (for example, [OKA11] and [WP09]). Thus each finger has 5 DoF except the thumb. For the thumb, we use 1 DoF for Interphalangeal (IP) joint, 3 DoF for MCP, and 3 DoF for Trapeziometacarpal (TM) joint. So a 3D hand pose consists of 33 DoF in total.



(a) Skeleton model (b) Mesh model

Figure 5: *Our hand pose consists of* 33 *degrees of freedom, including absolute root position and orientation (6) and the relative joint angles of individual joints (27). (left) skeletal hand model; (right) skinned mesh model for hand.*

We represent the skeletal size of a hand using a long vector $\mathbf{s} = [s_1,...,s_{20}]^T$, where $s_b, b = 1,...,20$ is the length of the b-th bone segment. We attached 21 markers on hand to capture 3D hand performances (see Figure 3). Specifically, there are 4 markers on each of five fingers. For the thumb, the 4 markers are placed at fingertip, IP, MCP and TM, respectively. For the other fingers, the 4 markers are placed at fingertip, DIP, PIP and MCP. Additional 1 marker is placed at the wrist. We denote the local coordinates of each marker as $\mathbf{p}_i, i = 1,...,21$. The goal of subject calibration is to esti-

mate the hand's skeleton model ($\mathbf{s}$) and local coordinates of each marker attached on hand ($\mathbf{p}_i, i = 1,...,21$).

The subject calibration process is conceptually similar to skeletal calibration process required for full-body motion capture using the *Vicon* system. Briefly, the subject was instructed to perform some simple hand gestures, denoted as $\mathbf{q}_n, n = 1,...,N$, in front of Vicon cameras. In our experiment, we instructed the subject to bend each finger separately so that all the markers are visible to at least two Vicon cameras and their 3D positions can be accurately recorded. The 3D marker positions of each frame are then used as the input to simultaneously estimate the skeletal sizes and local coordinates of each marker.

Specifically, we formulate the subject calibration process as the following nonlinear optimization problem:

$$\arg\min_{\mathbf{s},\{\mathbf{p}_i\},\{\mathbf{q}_n\}} \sum_n \sum_i \|\mathbf{f}(\mathbf{p}_i,\mathbf{s},\mathbf{q}_n) - \mathbf{t}_n^i\| + \lambda_1\|\mathbf{s}-\bar{\mathbf{s}}\| + \lambda_2 \sum_i \|\mathbf{p}_i - \bar{\mathbf{p}}_i\| \tag{2}$$

where the vector-valued function $\mathbf{f}$ is the forward kinematics function that maps the local coordinates of the i-th marker ($\mathbf{p}_i$), under the hand skeletal size ($\mathbf{s}$) and joint angle pose ($\mathbf{q}_n$), to its global coordinates. The vector $\mathbf{t}_n^i$ represents the observed global coordinates of the i-th marker at calibration frame n.

Intuitively, the first term measures how well the hypothesized marker positions match the observed marker positions. The second and third terms are regularization terms used to reduce the solution space for skeletal lengths ($\mathbf{s}$) and local coordinates of each marker ($\mathbf{p}_i, i = 1,...,21$). This is because in general the solution is not unique because of the uncertainty of local coordinate systems. The default values of local coordinates of each marker, denoted as $\bar{\mathbf{p}}_i, i = 1,...,21$, were simply set to zeros because they are approximately attached to joints. The default values for skeletal lengths, denoted as $\bar{\mathbf{s}}$, were obtained by computing the Euclidean distances between inboard and outboard markers. The weights λ_1 and λ_2 balance the importance of each term.

We optimize the cost function using the Levenberg-Marquardt (LM) algorithm [BSS93]. The optimization simultaneously computes skeletal lengths of the hand $\mathbf{s}$ and local coordinates of each marker $\mathbf{p}_i, i = 1,...,21$, as well as each calibration pose $\mathbf{q}_n, n = 1,...,N$. Based on the result of the subject calibration, we can further build a skinned mesh model for the hand so that our hand mesh model can be deformed according to pose changes of an underlying articulated skeleton using Skeleton Subspace Deformation (Figure5(b)), which will later be used in pose inference.

Camera calibration. Camera calibration is required for combining 3D marker position data from the *Vicon* system and RGB-D image data from the *Kinect* camera. Here we focus our discussion on calibrating the *Kinect* camera since calibrations of *Vicon* cameras are automatically achieved by *Vicon* software.

The intrinsic parameters of the *Kinect* camera are computed using the method in [Bou04]. To compute the extrinsic parameters, *i.e.* the rigid transformations between the *Kinect* camera coordinate system and the *Vicon* camera coordinate system, we adopted the following procedure. We attached several markers on a calibration box and captured several frames of the calibration box under different poses with both the *Vicon* cameras and *Kinect* camera. The marker locations in *Kinect* images can be easily extracted because of high intensity values caused by retro-reflective marker. To illuminate the retro-reflective markers, we placed a photography light near the *Kinect* camera during the process of camera calibration. Their 3D locations in the *Kinect* camera coordinate frame can further be obtained via the recorded depth maps. Given a set of 3D corresponding points in the *Vicon* coordinate system and the *Kinect* coordinate system, the rigid transformation between the two coordinate systems can be estimated via least square techniques.

5. 3D Hand Pose Reconstruction

We formulate the 3D hand pose reconstruction problem in an optimization framework by minimizing the discrepancies between the reconstructed motion and observed data. Section 5.1 introduces a technique for automatically extracting silhouettes of hand from observed Kinect images. In Section 5.2, we define a cost function to measure the consistency between the reconstructed motion and observed image data. Section 5.3 presents an efficient optimization technique to find the 3D hand poses that best match observed data.

5.1. Hand Silhouette Extraction

Silhouette maps are important for our combined acquisition method, as they are not captured in marker-based systems. We extract silhouettes of hand based on color images only. This is because depth images obtained from the Kinect camera, particularly around the contour, are often very noisy, thereby failing to produce clean silhouette maps.

Given a Kinect RGB image, we first do simple background subtraction to locate an approximate region of the hand. Graph cut technique [BJ01] is then applied to refine the results and obtain accurate segmentation of the hand from the background image. Graph cut formulates segmentation as a binary graph labeling problem, which labels each pixel to be foreground or background. The optimal labeling is achieved by minimizing an energy function containing two terms: *data* term and *edge* term. The *data* term evaluates the likelihood of pixels belonging to foreground and background. In our experiment, we model the probability density functions of foreground/background pixels using Gaussian Mixture Models (GMM). The *edge* term constrains the extracted contour to align with edges. This optimization problem can be solved efficiently with the max-flow algorithm. Note that we instructed the subject to wear a "red" wristband to ensure a clean segmentation of the hand from the arm.

5.2. Cost Function

We formulate the hand motion capture process in a model-based registration framework and adopt the "analysis-by-synthesis" strategy to sequentially register the skinned hand mesh model to observed data. Specifically, we define the following objective function to evaluate how well a pose hypothesis $\mathbf{q}_t$ matches an observation $\mathbf{o}_t$:

$$E(\mathbf{q}_t, \mathbf{o}_t) = \lambda_m E_m + \lambda_s E_s + \lambda_d E_d + \lambda_e E_e + \lambda_c E_c \quad (3)$$

where E_m measures the distance between hypothesized and observed marker positions. E_s ensures the hypothesized silhouette map is consistent with the extracted silhouette map. E_d and E_e measure the depth and edge discrepancies between the hypothesized and observed RGB-D data. E_c is a smoothness term, which penalizes sudden changes of reconstructed poses. The weights λ_m, λ_d, λ_s, λ_e, and λ_c control the importance of each term. In our experiment, λ_m, λ_s, λ_d, λ_e, and λ_c are set to 1.5, 22, 0.8, 0.4, and 0.16, respectively.

The marker term, E_m, measures the discrepancy between the hypothesized marker positions, denoted as $r_m(i), i = 1,...,N$, and the observed marker positions, denoted as $o_m(j), j = 1,...,M$, obtained from the Vicon cameras. We have

$$E_m = \frac{1}{N}\sum_{i}^{N} \min(\min_j \|r_m(i) - o_m(j)\|, T) \quad (4)$$

Specifically, for each hypothesized marker $r_m(i), i = 1,...,N$, we find the corresponding observed marker by searching the nearest neighbor in all the observed markers $o_m(j), j = 1,...,M$. In addition, we define a cut-off threshold T for the return error because we want to ensure the cost function is robust to outliers caused by missing markers. In our experiment, the threshold T is set to $8mm$.

The silhouette term, E_s, ensures that silhouette maps of synthesized images match those extracted from observed images. A silhouette map is encoded as a binary image whose foreground and background pixels are set to one and zero, respectively. The system automatically extracts silhouette maps of the hand, denoted as o_s, in observed Kinect images using the method described in Section 5.1. In contrast, the silhouette map of a rendered image, denoted as r_s, is automatically generated by the rendering process. The silhouette term is defined as follows:

$$E_s = \frac{\sum_i o_s(i)(1 - r_s(i))}{\sum_i o_s(i)} + \frac{\sum_i r_s(i)(1 - o_s(i))}{\sum_i r_s(i)} \quad (5)$$

where sums are computed over every pixel of the entire rendered or observed image. Intuitively, the silhouette term minimizes the area of the non-overlapping region between the synthesized and observed silhouette maps, thereby maximizing the area of their overlapping region.

The depth term, E_d, measures the depth differences between the observed and synthesized depth maps. This term is only evaluated within the overlapping region between the rendered and observed silhouette map, *i.e.* $o_s \wedge r_s$. We have

$$E_d = \frac{1}{\sum o_s \wedge r_s} \sum_{i \in o_s \wedge r_s} |o_d(i) - r_d(i)| \quad (6)$$

where r_d is the depth map generated by the hypothesized pose $\mathbf{q}_t$ and o_d is the observed depth map. Again o_s and r_s are the observed and rendered silhouette map, respectively.

The edge term, E_e, measures the discrepancies of edge maps between the rendered and observed images. In our experiment, we use a binary image to encode an edge map, where edge and non-edge pixels are set to 1 and 0, respectively. In our implementation, we apply Canny edge detectors [Can86] to extract edge maps of observed images, denoted as I_e^o. We extract the edge map of the rendered image, denoted as I_e^r, using the method described in [Goo03]. We have

$$E_e = \frac{\sum_i I_d^o(i) \cdot I_e^r(i)}{\sum_i I_e^r(i)} + \frac{\sum_i I_d^r(i) \cdot I_e^o(i)}{\sum_i I_e^o(i)} \quad (7)$$

where I_e^r and I_e^o are the rendered and observed binary edge maps for the hand. I_d^r and I_d^o are distance transform images of the rendered and observed edge images. The pixel values in a distance transform image indicate the distance of the pixels to the closest edge pixels in the corresponding edge image. As a result, distance transform images provide a more robust and smooth measurement for edge images.

The smoothness term, E_c, ensures that the reconstructed motion is temporally smooth. Specifically, it penalizes the sudden changes of poses between two consecutive frames. We have

$$E_c = \|\mathbf{q}_t - \tilde{\mathbf{q}}_{t-1}\| \quad (8)$$

where $\mathbf{q}_t$ and $\tilde{\mathbf{q}}_{t-1}$ are the hypothesized pose at the current frame and the reconstructed pose at the previous frame, respectively.

5.3. Optimization

Our hand motion capture process sequentially reconstructs 3D poses of the hand by minimizing the cost function defined in Equation (3). We apply particle swarm optimization (PSO) techniques [CK02] to sequentially estimate the pose over time. PSO is a computational method that optimizes a cost function by iteratively improving a candidate solution with regard to a given measure of quality.

Briefly, PSO optimizes a problem by having a population of candidate solutions, termed "particles", and moving these particles around in the search-space according to simple mathematical formulae over the particle's position x and velocity v. Each particle's movement is influenced by its local best known position P and is also guided toward the best known positions G in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. In our

experiment, the solution at frame $t-1$ is used to generate the initial population at frame t by Gaussian perturbation.

The update equations from generation k to $k+1$ are defined as

$$v_{k+1} = w(v_k + c_1 r_1 (P_k - x_k) + c_2 r_2 (G_k - x_k)) \quad (9)$$

and

$$x_{k+1} = x_k + v_{k+1} \quad (10)$$

c_1 and c_2 are cognitive component and social component, which control the importances of local and social term. w is the constriction factor, and r_1, r_2 are uniformly distributed random numbers in the range $[0,1]$. As suggested in [CK02], we fix the behavioral parameters to $c_1 = 2.8$, $c_2 = 1.3$ and

$$w = 2/\|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\| \quad (11)$$

with $\varphi = c_1 + c_2$. In our experiment, we found 48 particles and 200 generations for one frame are sufficient to generate good results for all of our testing examples.

Initialization. Our motion reconstruction process runs in a sequential mode and therefore requires initialization of hand poses at the first frame. We automatically initialize the pose at the first frame by instructing the subject to start at a default hand pose $\mathbf{q}_0$. Given the default hand pose, we apply iterative closest point (ICP) techniques to estimate the rigid transformations of the hand based on the observed and hypothesized marker positions. We further refine the initial pose $\mathbf{q}_0$ via inverse kinematics techniques, as there might be a slight difference between the performed default pose and actual one. We denote the refined pose at the first frame as $\tilde{\mathbf{q}}_0$.

Data Synchronization. A remaining issue is how to synchronize RGB-D image data from the kinect camera with marker position data obtained by Vicon as the two systems run in different frame rates. Our idea is to use the initial pose $\tilde{\mathbf{q}}_0$ reconstructed from marker position data to search the corresponding frame in Kinect data. This is achieved by matching the initial pose $\tilde{\mathbf{q}}_0$ against the observed RGB-D images at the first several frames of the Kinect data. We evaluate the matching error based on the Equation (3) except that we exclude the marker term and smoothness term from evaluation. We pick the frame with the smallest matching error as the corresponding frame. The synchronization of subsequent frames between the two systems can be easily achieved as the frame rate of the Vicon system is several (four in our experiment) times faster than the Kinect system.

6. Post-processing

The goal of our post-processing step is to enhance the temporal resolution of the reconstructed motion to match the frame rate of the marker-based motion capture system. This is because the marker-based motion capture systems often run at much higher frame rates than RGB-D cameras. However, the motion reconstruction process described in Section 5 can only reconstruct hand motion data in the same frame rate as the Kinect system. To benefit from the high frame rate of the *Vicon* system, we need to estimate 3D hand poses for the remaining *Vicon* frames.

Assume that we have obtained hand poses for *Vicon* frame t and $t+k$, denoted as $\tilde{\mathbf{q}}_t$ and $\tilde{\mathbf{q}}_{t+k}$, respectively. Our goal here is to reconstruct the poses for in-between frames, $\mathbf{q}_{t+i}, i = 1,...,k-1$, based on observed marker position data from Vicon, denoted as $\mathbf{m}_{t+i}, i = 1,...,k-1$. Simply applying inverse kinematics techniques might not be sufficient to determine the 3D hand poses for intermediate frames because of missing markers caused by occlusion.

To address this challenge, we linearly interpolate poses between frame t and $t+k$ using $\tilde{\mathbf{q}}_t$ and $\tilde{\mathbf{q}}_{t+k}$ and refine the interpolated pose $\hat{\mathbf{q}}_{t+i}$ by fitting it to the visible marker positions $\mathbf{m}_{t+i}$. This is an optimization problem and the objective function is

$$\min_{\mathbf{q}_{t+i}} E_m(\mathbf{q}_{t+i}, \mathbf{m}_{t+i}) + \lambda ||\mathbf{q}_{t+i} - \hat{\mathbf{q}}_{t+i}||, \quad i = 1, \cdots, k-1. \quad (12)$$

The first term is the same as the marker term defined in Equation (4), while the second term measures the difference between the solution $\mathbf{q}_{t+i}$ and the linearly interpolated pose $\hat{\mathbf{q}}_{t+i}$. The weight λ controls the relative importance of the two terms.

We solve the optimization using PSO. Once we obtain the solution for intermediate frames, we can use them to find the corresponding information between the hypothesized markers and observed markers, which allows us to refine the solution via inverse kinematics techniques. To ensure the interpolated motion is temporally smooth, we employ *per-frame inverse kinematics plus filtering* (PFIK+F) [Gle01] for post-processing. PFIK+F consists of two components: per-frame IK and filtering. Per-frame IK applies inverse kinematics to each frame separately, to enforce the spatial constrains. However, changes resulting from this step may introduce high frequency components to the motion such as spikes. The filtering step enforces the smoothness constraint by only accepting low frequency changes. They are interleaved in an iterative processing mode, and can result in accurate while smooth results.

7. Results

We have evaluated the performance of the system by constructing a large and heterogeneous hand motion database (about 30 minutes mocap data), including everyday hand gestures, American sign language (ASL), and hand gestures for object grasping and manipulation. Our results show that the system can capture a wide range of high-quality hand motion data even in case of significant self-occlusion, a capability that has not been demonstrated in previous hand motion capture systems. Our results are best seen in the accompanying video although we show sample frames of a few reconstructed hand poses in the paper (see Figure 8).

(a) (b) (c) (d) (e) (f)

Figure 6: *With/without RGB-D image data. (a) the input image superimposed by all markers visible to the Kinect camera; (b) all visible markers in 3D space; (c) result using marker position data only; (d) an overlay view of the result of (c); (e) result of using both marker position data and RGB-D data; (f) an overlay view of the result of (e).*

(a) (b) (c) (d) (e)

Figure 7: *With/without marker position data. (a) the input RGB image; (b) result of using RGB-D image data only; (c) an overlay view of the result of (b); (d) result of using both marker position data and RGB-D image data; (e) an overlay view of the result of (d).*

Without RGB-D Image Data. To demonstrate the importance of RGB-D image data to our hand motion capture process, we compared the motion capture results with and without RGB-D image data. Specifically, we compared the reconstruction results by dropping off the *silhouette*, *depth* and *edge* terms against the reconstruction results obtained by optimizing all the terms defined in Equation 3. We adopted the same optimization process and used the same number of particles and generations for 3D pose optimization. The accompanying video provides a side-by-side comparison between the two. Figure 6 shows one sample result for with and without RGB-D image data. Due to significant self-occlusion, there were only *13* markers visible, out of 21 in total. In particular, the "fingertip" and "IP" markers of the thumb and the "fingertip" and "DIP" of the ring finger had been missed for a long time. Therefore the pose for both the thumb and ring finger can not be reconstructed accurately using marker position data alone (see Figure 6 (b), (c) and (d)). But the RGB-D data can provide clear information about the thumb and the ring finger so we can get an accurate pose by complementing marker position data with RGB-D image data (see Figure 6 (e) and (f)).

Without Marker Position Data. We have also evaluated the importance of marker position data to our hand motion capture process. We obtained the reconstruction results for "without marker position data" by simply dropping off the marker term in the objective function defined in Equation 3. Figure 7 shows a side-by-side comparison of reconstructed poses using Kinect data only and using both types of data. Without marker position data, the reconstruction pose for both the middle finger and ring finger is completely wrong (Figure 7(b) and (c)). This is because the RGB-D data alone might not be sufficient to accurately determine the solution, as the data obtained from the Kinect camera is often very noisy and low-resolution and is also prone to occlusion. Figure 7(c) shows that the incorrect pose also fits the Kinect data well because of the confusion between the middle finger and the ring finger from the current camera viewpoint. So in these cases the results of using Kinect data only can not be guaranteed. By complementing RGB-D image data with marker position data, we can significantly reduce ambiguity and thereby obtain an accurate reconstruction pose shown in Figure 7(d) and (e). More comparison results can be viewed in Figure 8 as well as the companying video.

Computational time. The current system runs in about 1 frame per second without code optimization. All of our experiments were tested on an Intel Xeon GPU E5430, 3GB RAM, NVIDA GeForce 8800 GTX.

8. Conclusion

We have demonstrated a robust method for acquiring high-fidelity hand articulation data. By complementing a marker-based motion capture system with RGB-D image data obtained from a single Kinect camera, the effect of missing markers is significantly reduced and high-quality hand motion can be reconstructed. In addition, the system is fully

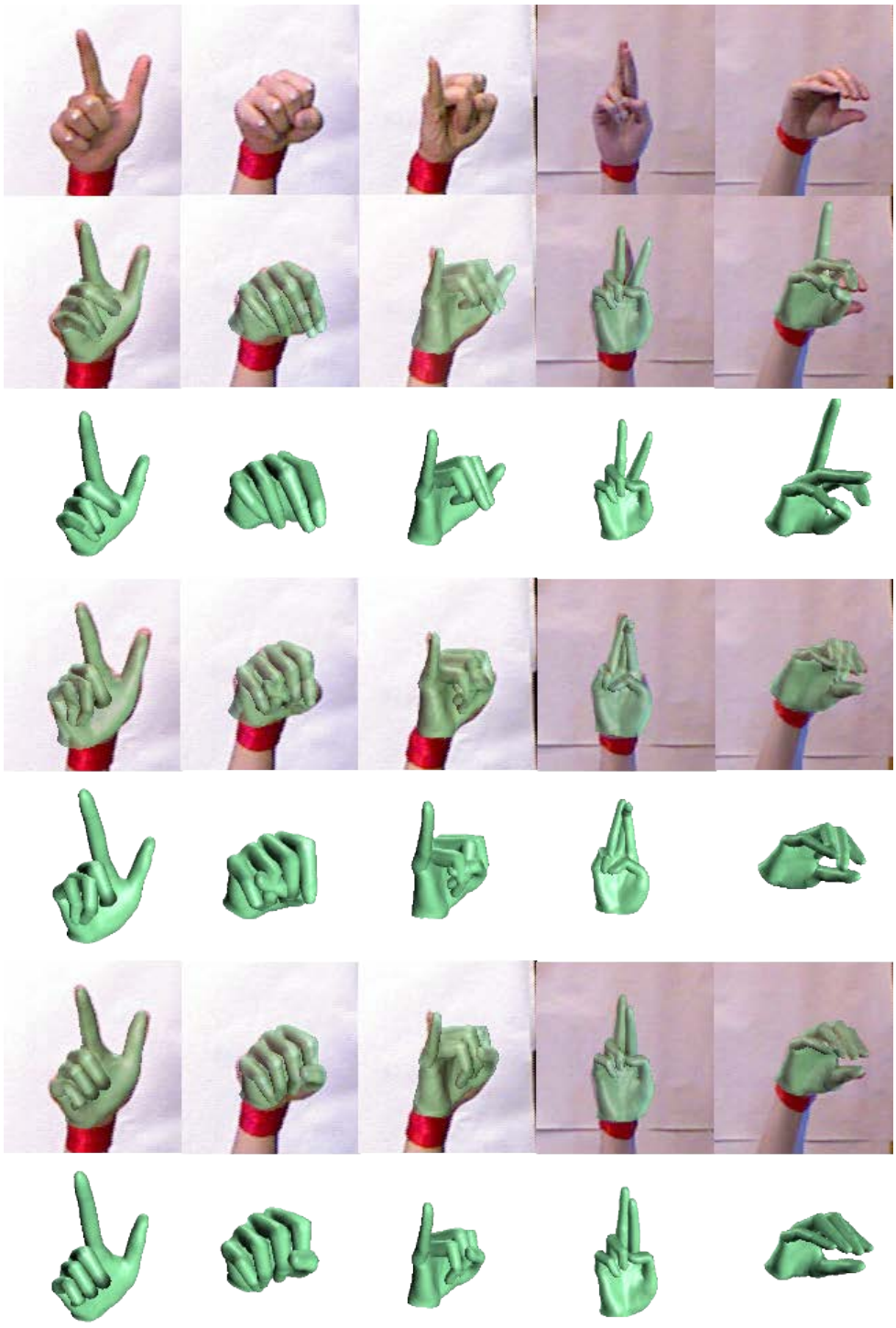


Figure 8: *Comparisons among different methods. Each column is an example. The first row gives the reference color images captured by the* Kinect *camera. The third, fifth and seventh rows give the reconstructed poses using marker position data only, RGB-D data only, and both kinds of data (our method). The second, forth and sixth rows show the corresponding overlay views.*

automatic and requires no manual user intervention such as marker labeling and gap filling.

Experiments have demonstrated that the system can accurately capture articulated hand movement, even in case of significant self-occlusion. However, we have not yet attempted to rigorously assess when the system will break down. Currently we use only a single RGB-D camera to complement marker-based motion capture systems. We plan to add more RGB-D cameras into the system and make the system more robust to occlusion and sensor noise.

The current system does not run in real time. We believe most components of the current system can be GPU-accelerated. For example, particle swarm optimization (PSO) is easy to parallelize and can be implemented entirely on a GPU [ZT09, OKA11]. One of the immediate directions for future work is, therefore, to speed up the system with GPU implementations.

In the future, we would also like to test our system on capturing more hand motion data. This would enable us to build a comprehensive and detailed hand motion database required for data-driven hand modeling and synthesis. We also plan to extend our system to acquiring interaction between hand and object such as hand grasping and manipulation. This requires modeling both articulated hand movement and object movement as well as subtle contact phenomena between hand and object (*e.g.* grasping the cup's handle). Another direction of future work is the investigation of new algorithms for modifying and reusing the captured hand articulation data to achieve new tasks such as motion transformation, editing, interpolation and composition.

9. Acknowledgement

We are thankful to Yiming Qian for the help with motion capture. This material is based upon work partially supported by National Basic Research Program of China (No. 2012CB725300) and the National Science Foundation CAREER award IIS-1055046.

References

[AS03] ATHITSOS V., SCLAROFF S.: Estimating 3d hand pose from a cluttered image. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition* (2003), CVPR '03, pp. 432–439. 2

[BJ01] BOYKOV Y., JOLLY M.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (2001), vol. 1, pp. 105–112. 5

[Bou04] BOUGUET J.: Camera calibration toolbox for matlab. 5

[BSS93] BAZARAA M. S., SHERALI H. D., SHETTY C.: *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons Ltd, 1993. 2nd Edition. 5

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* (1986). 8(6): 679–698. 6

[CH05] CHAI J., HODGINS J.: Performance animation from low-dimensional control signals. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 686–696. 2

[CK02] CLERC M., KENNEDY J.: The particle swarm–explosion, stability, and convergence in a multidimensional complex space. In *IEEE Transactions on Evolutionary Computation* (2002). 6(1):58–73. 6, 7

[cmua] Carnegie mellon university (cmu) cyberglove database. http://graphics.cs.cmu.edu/data/grasp_posture/database.html.bak/. 2, 3

[cmub] Carnegie mellon university (cmu) mocap database. http://mocap.cs.cmu.edu/. 2

[Cyb] Cyberglove systems. http://www.cyberglovesystems.com/. 1, 3

[dLGFP11] DE LA GORCE M., FLEET D. J., PARAGIOS N.: Model-based 3d hand pose estimation from monocular video. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011), vol. 33, pp. 1793–1805. 2

[GCDA09] GOLDFEDER C., CIOCARLIE M., DANG H., ALLEN P.: The columbia grasp database. In *IEEE International Conference on Robotics and Automation (ICRA)* (2009), pp. 1710–1716. 2

[Gle01] GLEICHER M.: Comparing constraint-based motion editing methods. *Graphical models 63*, 2 (2001), 107–134. 7

[Goo03] GOOCH B.: Silhouette extraction. *Course Notes for Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems* (2003). 6

[HRMO12] HOYET L., RYALL K., MCDONNELL R., O'SULLIVAN C.: Sleight of hand: perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2012), pp. 79–86. 2

[MRC*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation Mocap Database HDM05*. Tech. Rep. CG-2007-2, Universität Bonn, June 2007. 2

[OKA11] OIKONOMIDIS I., KYRIAZIS N., ARGYROS A.: Efficient model-based 3d tracking of hand articulations using kinect. *Proceedings of The 22nd British Machine Vision Conference (BMVC)* (2011). 3, 4, 10

[RKK10] ROMERO J., KJELLSTROM H., KRAGIC D.: Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation (ICRA)* (2010), pp. 458–463. 2

[Vic] Vicon motion capture systems. http://www.vicon.com/. 1, 2

[WLH01] WU Y., LIN J., HUANG T. S.: Capturing natural hand articulation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (2001), pp. 426–432. 2

[WP09] WANG R. Y., POPOVIĆ J.: Real-time hand-tracking with a color glove. *ACM Trans. Graph. 28*, 3 (2009), 63:1–63:8. 2, 4

[ZH03] ZHOU H., HUANG T. S.: Tracking articulated hand motion with eigen dynamics analysis. In *Proceedings of the Ninth IEEE International Conference on Computer Vision* (2003), ICCV '03, pp. 1102–1109. 2

[ZT09] ZHOU Y., TAN Y.: Gpu-based parallel particle swarm optimization. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation* (2009), CEC'09, pp. 1493–1500. 10