Unsupervised Skeleton Extraction and Motion Capture from Kinect Video via 3D Deformable Matching

Quanshi Zhang[a], Xuan Song[a], Xiaowei Shao[a], Ryosuke Shibasaki[a], Huijing Zhao[b]

[a] *Center for Spatial Information Science, University of Tokyo*
[b] *Key Laboratory of Machine Perception (MoE), Peking University*

Abstract

This paper presents a novel method to extract skeletons of complex articulated objects from 3D point cloud sequences collected by the Kinect. Our approach is more robust than the traditional video-based and stereo-based approaches, as the Kinect directly provides 3D information without any markers, 2D-to-3D-transition assumptions, and feature point extraction. We track all the raw 3D points on the object, and utilize the point trajectories to determine the object skeleton. The point tracking is achieved by the 3D non-rigid matching based on the Markov Random Field (MRF) Deformation Model. To reduce the large computational cost of the non-rigid matching, a coarse-to-fine procedure is proposed. To the best of our knowledge, this is the first to extract skeletons of highly deformable objects from 3D point cloud sequences by point tracking. Experiments prove our method's good performance, and the extracted skeletons are successfully applied to the motion capture.

Keywords:
skeleton extraction, 3D point cloud sequence

1. Introduction

Unsupervised object skeleton extraction is an active research topic in computer vision, as it can potentially improve performance in various CV

Email address: zqs1022@csis.u-tokyo.ac.jp (Quanshi Zhang)

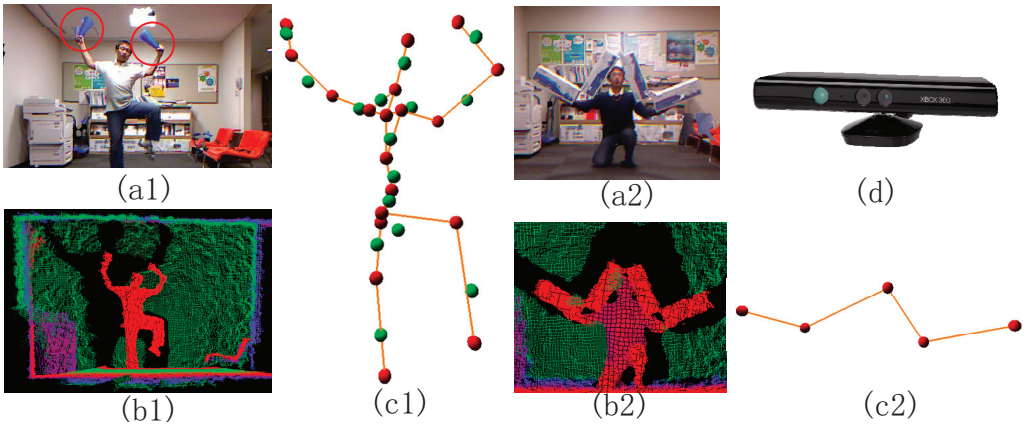

Figure 1: Skeleton extraction from a 3D point cloud sequence. The 3D point cloud sequence is collected by the Kinect. (a1–2) No prior knowledge of the object type is required. *E.g.* The object is a man holding two cones (in the circles). (b1–2) Input: a 3D point cloud sequence of the object obtained by the Kinect. (c1–2) Output: the object's specific skeleton. (d) The Kinect sensor device [14].

applications, such as 3D motion capture, 3D pose estimation, activity recognition, 3D object tracking and etc.

Previous approaches extract articulated object skeletons from videos [1] [2] [3] [4] [5] [6], motion capture data [7] [8], and static object models [9] [10] [11] [12] [13]. However, all these approaches have their intrinsic limitations, such as "difficult to reflect the 3D object motion", "need some markers on the object", "the extracted skeleton and recorded motion are inaccurate in the 3D coordinate" and etc. Nevertheless, with the fast development of 3D hardware, the Kinect as a new kind of 3D sensor has received the increasing attention for solving traditional computer vision problems. It is developed by Microsoft for the Xbox 360 video game platform, and it can collect both the RGB video stream and the depth sensing stream at a frame rate of 30Hz (as shown in Fig. 1(a1–2,b1–2)). Obviously, the Kinect provides us another choice: *why not extract object skeletons and capture motion directly from 3D point cloud sequences?* Therefore, *the purpose of this paper is to develop a novel approach that can extract articulated object skeletons and capture object motion directly from 3D point cloud sequences without any prior information, such as the object type and etc.*

Our approach mainly contains three steps, as illustrated in Fig. 2. At

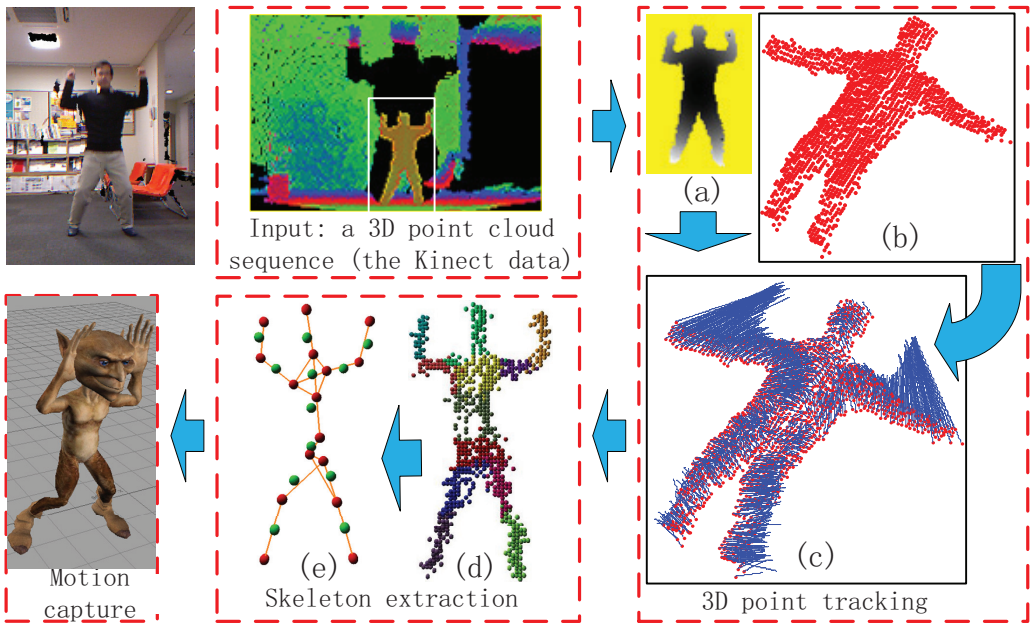

Figure 2: The whole framework. (1) The input is the 3D point cloud sequence collected by the Kinect. The RGB video data (also collected by the Kinect) are not required by our system. (2) The coarse-to-fine MRF-based 3D non-rigid matching generates the 3D point trajectories. We utilize the integral geodesic distance *(a)* as the point feature in matching. We calculate the point-by-point deformation between the 1^{st} frame *(b)* and each other frame to track 3D points *(c)*. (3) The skeleton extraction. We cluster trajectories into body segments *(d)*, and then utilize a probabilistic graphical model to determine the object skeleton *(e)*. (4) Finally, the automatically extracted skeleton is applied to the motion capture.

first, we utilize a coarse-to-fine MRF-based 3D non-rigid matching to track all the raw 3D points (as shown in Fig. 2(a,b,c)). Then, we utilize the spectral clustering to group these point trajectories into different body segments (as shown in Fig. 2(d)). Finally, we utilize a graph model to determine the connections between the body segments (as shown in Fig. 2(e)). In addition, the extracted skeleton can also be applied to the motion capture (as shown in Fig. 2).

The proposed approach has the following key features that make it advantageous over previous ones: (1) our system does not require markers in the data collection, which not only saves the human labor, but also has merits in learning the object's unknown structure, as shown in Fig. 1(a1–2). In

contrast, for the marker-based approach, people usually locate markers on some key parts of the articulated object (such as the joints and body segment centers), according to their subjective understanding. The subjective understanding can bring priori errors to the unknown structure learning. (2) Compared to the static-model-based approaches, our approach utilizes the motion information to obtain accurate object segments. Moreover, it does not require well-constructed 3D models. (3) Different from the video-based 3D reconstruction, the Kinect directly provides the object's spatial structure without any shape deformation assumptions. What's more, in order to obtain the body segments' motion, we can directly track all the raw 3D points without feature point extraction. Therefore, the 3D point tracking does not suffer from monotonous colors and illumination changes as the video-based tracking.

The main contributions of this paper can be summarized as follows: (1) To our best knowledge, this is the first work that extracts skeletons of complex articulated objects directly from 3D point cloud sequences without prior information by point-level tracking. Our algorithm provides a global segmentation of the object body segments, which is robust to small intra-segment deformation. (2) We propose an efficient coarse-to-fine framework to track the 3D points based on the MRF Deformation Model. The coarse-to-fine strategy greatly reduces the tracking's time/memory cost. To our best knowledge, this is the first work to track all the raw 3D points of a deformable object without any transformation assumptions.

The rest of this paper is organized as follows: The related work is briefly reviewed in the following section. Section 3 presents the coarse-to-fine 3D point tracking and Section 4 presents the skeleton extraction. The experiments and results are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Related work

Many previous approaches extract skeletons from videos (Ross *et al.* [1] [2], Yan *et al.* [3] [4], Tresadern *et al.* [5], and Ramanan *et al.* [6]). [1] [2] [3] [4] utilize the KLT tracker [15] to get feature trajectories. However, these methods require sufficient feature points on some key parts of the object for good performance, and the image-based tracking may suffer from illumination changes.

Some vision-based methods utilize multiple cameras to obtain a dense 3D point cloud sequence of the object, and then extract the object skeleton from the 3D sequence. Compared to the pure video-based methods, these methods obtain the object's 3D structure and spatial motion more directly and accurately. Cheung *et al.* [16] reconstruct a model of the kinematic structure and appearance of a person from the visual hull. However, the model reconstruction is based on the person's free motion, and they determine the joint point one by one by allowing only one body part to move in each step. Chu *et al.* [17] obtain volume sequences by multiple cameras, extract the skeleton curve from each frame, and then utilize the skeleton curves to determine the kinematic model. They do not track all the points over frames, so the system may face difficulties when the shape of some object parts (such as a round bowl) is not suitable for the principle curve extraction.

Kirk *et al.* [7] and Sturm *et al.* [8] directly get 3D point trajectories by using the motion capture system. The articulated body segments, as well as their motion, are learned from the marker trajectories.

Sturm *et al.* [18] utilize the depth data to learn the articulated models of cabinet doors and drawers with the rectangle detection.

The approaches mentioned above all extract object structure based on the motion information. [3] [5] utilize a factorization method to discover the rotation axis between two object body segments, and [8] generates a low-dimension parameter-free representation of the articulated object to extract the object structure. They assume the body segments are rigid. Therefore, these methods are not robust to the non-rigid body segments with complex intra-segment deformation. In contrast, some approaches in [7] [1] [2] [4] and ours just cluster trajectories into the body segments, which is based on the fact that the distance between two points on the same rigid body segment is constant. The clustering provides a global solution to the body segmentation, which is robust to small intra-segment deformation in theory.

Other approaches [9] [10] [11] extract skeletons from static 3D or 2D models. However, they only use the topological and geometrical information, and cannot segment the articulated object without motion cues. Aujay *et al.* [12] utilize a harmonic function to get anatomical information to improve the skeleton. Schaefer *et al.* [13] propose an example-based skeleton extraction, which requires several well-constructed 3D models in different poses.

Generally, the motion-based skeleton extraction method obtains an articulated skeleton, which represents the topological structure of the object [1] [2] [3] [4] [5] [19] [16] [7]. However, the skeleton extracted from the static

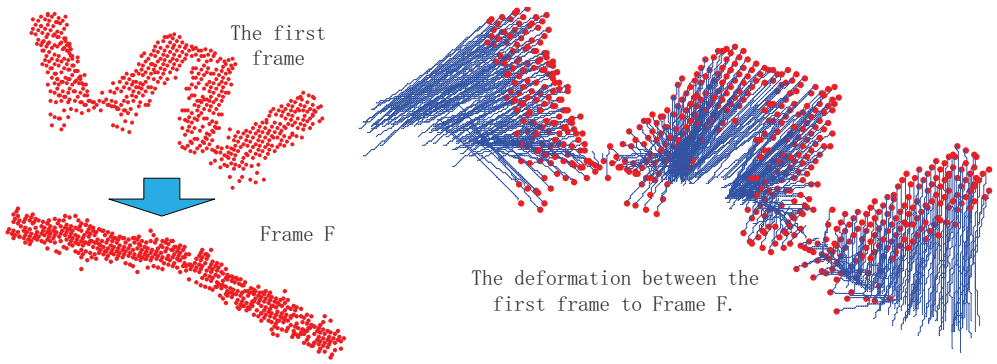

Figure 3: The 3D point tracking. The tracking is based on the non-rigid matching between the first frame to any other frame.

model is usually the medial axes of body segments [9] [10] [11].

The Kinect is a recently-developed depth sensor and can directly provide the 3D point cloud sequence of the object. More recently, researchers pay the increasing attention to this sensor and have started to apply it to various computer vision tasks. Shotton *et al.* [20] utilize the Kinect for human pose recognition, and Oikonomidis *et al.* [21] utilize the Kinect to track hand articulations.

We propose a more robust skeleton extraction method, which directly tracks all the raw 3D points collected by the Kinect, and utilizes the point trajectories to determine the skeleton. Our proposed system is more robust than the video-based and the stereo-based methods. Compared to the video-based and the stereo-based skeleton extraction approaches, our system does not have to extract sufficient feature points for enough feature trajectories or 3D reconstruction. Usually, in the unique-color area, there are not enough feature points, or the extracted feature points are not reliable. And the video-based methods should also consider the limitations of structure-from-motion algorithms in the 2D-to-3D transition.

3. Coarse-to-fine tracking 3D points

The first stage of our system is to generate the 3D point trajectories by tracking each 3D point over frames (as shown in Fig. 3). The point tracking is based on the multi-frame 3D non-rigid matching. The matching-based tracking is not achieved in a Markov process, so it avoids tracking error accumu-

lation over frames. To match two specific frames, we extend the image-based MRF Deformation Model proposed in [22] [23]. The time/memory cost to match a large number of 3D points is intolerable. Therefore, we utilize a coarse-to-fine strategy to reduce the searching range in matching.

3.1. MRF deformation model

Let F^1, F^2 denote two frames. We use a graph $G = (V, E)$ to represent the 3D points in F^1. Each node $s \in V$ stands for a point in F^1, and its spatial position is $p_s \in \mathbb{R}^2$. $(s,t) \in E$ if $\|p_s - p_t\| < \epsilon$ ($\epsilon = 10cm$, here). Each node has k candidate matching points in frame F^2. $\mathcal{L} = \{1...k\}$ is a label set. Let each node $s \in V$ be assigned a label $x_s \in \mathcal{L}$. $x = \{x_s | s \in V\}$. Different labels indicate this node's different candidate matching points.

The energy function of x is:

$$E(x|\theta) = \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \quad , \tag{1}$$

where $\theta_s(\cdot)$ is a unary data penalty function, and measures the feature dissimilarity between two matched points. $\theta_{st}(\cdot,\cdot)$ is a pairwise interaction penalty function, which measures the discontinuities in deformation.

We take the integral geodesic distance as the point feature [24]. It is a robust and discriminative feature for 3D object retrieval. The geodesic distance is the distance from point to point on a surface. For the graph $G = (V, E)$, the geodesic distance between node $s \in V$ and node $t \in V$, $g(s,t)$, is the shortest path's length from s to t in the G's network. Node s's integral geodesic distance is defined as $g(s) = \sum_{t \in V} g(s,t)$.

The unary data penalty function is defined as follows.

$$\theta_s(x_s) = \|g(s) - g(m_s(x_s))\|^2 / \sigma_g^2 \quad , \tag{2}$$

where σ_g^2 is the geodesic distance's variance. $m_s(x_s)$ is the candidate matching point in frame F^2 for node s with its label x_s.

The real deformation may be complex, but two neighboring points are expected to keep a constant distance over frames. Therefore, the pairwise interaction penalty function is defined as follows.

$$\theta_{st}(x_s, x_t) = \begin{cases} \frac{(\|p_s - p_t\| - \|p_{m_s(x_s)} - p_{m_t(x_t)}\|)^2}{\sigma_d^2} & p_{m_s(x_s)} \neq p_{m_t(x_t)} \\ \tau & p_{m_s(x_s)} = p_{m_t(x_t)} \end{cases} \tag{3}$$

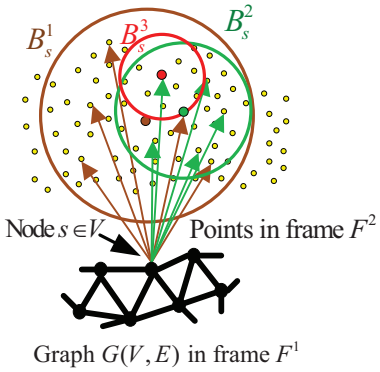

Figure 4: The N-level coarse-to-fine procedure for candidate matching point selection. $\{B_s^i\}(i = 1, 2, ..., N)$ is node s's sphere set. s's k candidate matching points are uniformly selected inside the sphere B_s^i in the i^{th} level. s's matched point calculated by the MRF deformation model is set as B_s^{i+1}'s center.

where σ_d^2 is the variance of the distance between two neighboring points, and τ is a large const to prevent two nodes from matching to the same candidate matching point. We set $\tau = \max_{u,v}(\|p_u - p_v\|^2)/(\sigma_d^2)$, here.

We utilize TRW-S algorithm [25] to the MRF Deformation Model. TRW-S is a recently developed algorithm for discrete energy minimization. Compared to belief propagation (BP), TRW-S can ensure convergence, as it has no loop.

3.2. Non-rigid matching

Coarse-to-fine Matching: To reduce the matching cost, Lee *et al.* [26] simply decompose the point searching range from a 3D space to three 1D space, but the time/memory cost remains very large. We choose a coarse-to-fine strategy for time/memory cost reduction.

The MRF Deformation Model is utilized in an N-level coarse-to-fine procedure to match two frames as shown in Fig. 4. Let's take the node s's matching for example. In the first level, we select k points uniformly inside a sphere B_s in frame F^2 as the candidate matching points. Then, we apply the MRF Deformation Model to calculate s's matched point. B_s's center is initialized as the estimated matching point in frame F^2, according to s's spatial position $p(s)$. We repeat this process in the following levels. B_s's center is set as s's matched point in the previous level's matching. We reduce B_s's

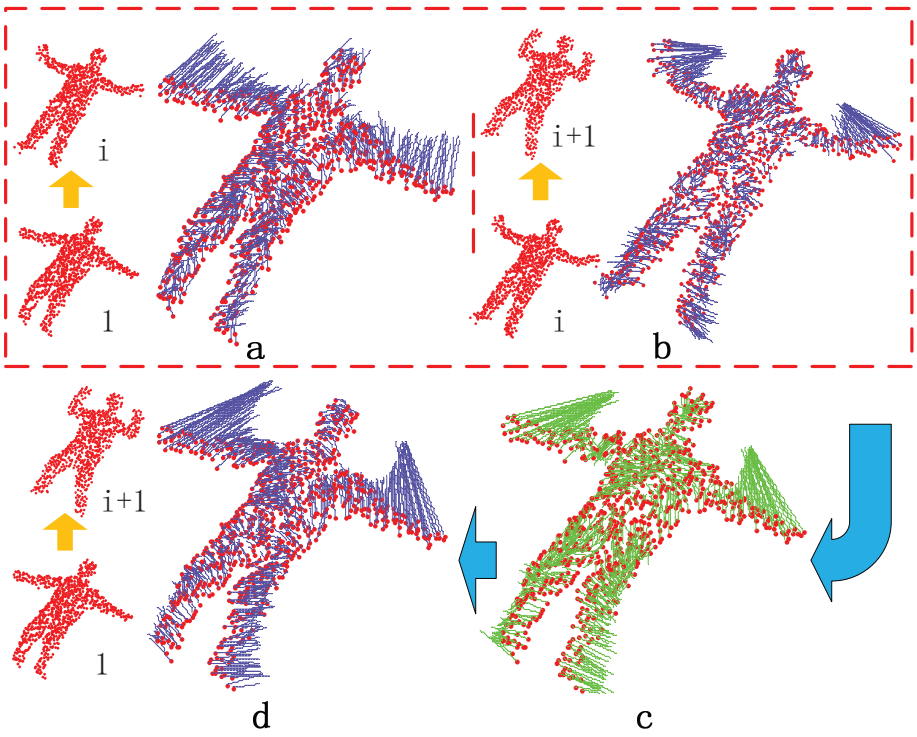

Figure 5: Matching multiple frames. The two-step non-rigid matching makes the 3D point tracking not a Markov process, and thus avoids tracking error accumulation over frames. (a) The deformation between Frame 1 and Frame i is given. (b) Match Frame i to Frame $(i+1)$. (c) The estimated deformation between Frame 1 and Frame $(i+1)$. (d) With the estimated deformation, match Frame 1 to Frame $(i+1)$ to get the accurate deformation between Frame 1 and Frame $(i+1)$. Only 50% of all the 3D points are shown for clarity.

radius level by level to get a coarse-to-fine result.

Matching multiple frames: The 3D points are tracked by the 3D non-rigid matching. To avoid matching error accumulation over frames, we match each frame to the first frame in a two-step matching framework, as shown in Fig. 5. (1) The estimation step: Given the deformation between Frame 1 to Frame i, match Frame i to Frame $(i+1)$ to estimate the deformation between Frame 1 and Frame $(i+1)$. (2) The modification step: The estimated deformation is utilized to initialize the candidate matching points. Match Frame 1 to Frame $(i+1)$ to get accurate deformation between Frame 1 and Frame $(i+1)$. The modified matching example is shown in Fig. 3.

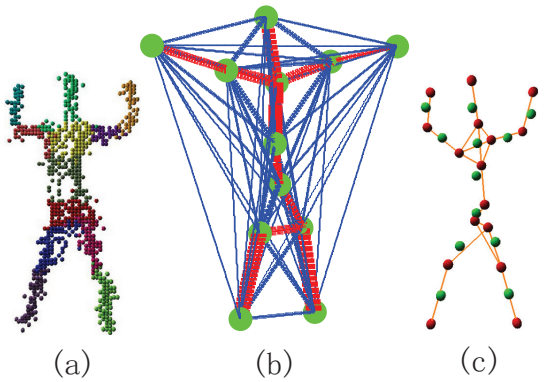

Figure 6: Skeleton extraction: (a) The body segmentation. The several-to-one matching cannot be totally avoided by the MRF Deformation Model, so the point cloud seems sparse. (b) The probabilistic graphical model of the body segments. Each node represents a body segment. The edge width is the edge weight's reciprocal. The minimum spanning tree of these nodes indicates the topological connections between body segments. (c) The skeleton with joint points and body segment central points.

4. Skeleton extraction

The second stage of our system is to extract the dynamic skeleton using 3D point trajectories generated in Section 3. The marker-based skeleton extraction approach proposed in [7] [4] is modified to adapt to the 3D point cloud data.

Body segmentation: The first step of skeleton extraction clusters the points into rigid body segments (Fig. 6(a)). The body segments reflect the body's skeletal structure. In an ideal rigid body, any two points should keep a constant distance over frames and their distance's standard deviation should be zero. Thus, we choose the standard deviation of two points' distance over frames as the distance measure as follows, and utilize spectral clustering [27] to determine body segments.

$$dist(p_1, p_2) = Var^{1/2}(\|p_1 - p_2\|) \quad , \tag{4}$$

where, p_1 and p_2 $(p_1, p_2 \in \mathbb{R}^2)$ are two points.

Joint point determination: In the second step of the skeleton extraction, we utilize a probabilistic graphical model to determine the topological connections between body segments as well as the joint points. We treat

each body segment as a node in a complete graph. The edge weight between two body segments C_1 and C_2 is defined as follows.

$$W_{C_1,C_2} = \max_{i \in F} \min_{p_{1,i} \in c_{1,i}, p_{2,i} \in c_{2,i}} \|p_{1,i} - p_{2,i}\| \quad , \tag{5}$$

where, F is the frame label set, $c_{1,i}, c_{2,i}$ are the point sets of body segments C_1 and C_2 in Frame i. $C_1 = \{c_{1,i}\}, C_2 = \{c_{2,i}\}$.

The two body segments with small edge weight have large possibility to be connected. We generate the graph's minimum spanning tree as the topological connections between body segments. Fig. 6(b) shows the body segment graph and the minimum spanning tree. We select the joint point of two connected body segments from either body segment's point set. The joint point should be the one with the shortest average distance to the other body segment over frames.

Skeleton generation: The last step of skeleton extraction draws skeleton based on joint points (Fig. 6(c)). To represent the body segment with multiple joint points, we draw a skeleton between each pair of the joint points. For the body segment with a single joint point (*e.g.* the lower leg and the head), we connect the joint point to its farthest point in this segment.

5. Experiment

To show the performance of our system, we used the Kinect to collect four 3D point cloud sequences, and extracted four skeletons from these sequences respectively. Sequence 1 is a man, Sequence 2 is a man holding two cones, Sequence 3 is a box chain and Sequence 4 is a vacuum cleaner. Meanwhile, we further collected the 3D data of a human upper body by using both the Kinect (Sequence 5) and the marker-based motion capture system (marker trajectories), in order to compare our method with the marker-based motion capture. To measure the coarse-to-fine strategy's performance in the time/memory cost reduction, we conducted five experiments on Sequence 5, and the 3D point matching with five sets of different parameters was performed in these experiments.

Body segment centers and the extracted skeleton's joint points are usually the articulated object's key points. Therefore, these automatically learned centers and joint points can take place of the markers in the motion capture, as in Experiment 1 (Section 5.2). We can extract the skeleton of any irregular articulated object or unknown object, as no prior knowledge or constrains are

required in our approach. *E.g.* the cones holding in man's hands are successfully extracted as two specific body segments (Section 5.3) and the skeleton of the box chain with a strange shape can also be extracted. (Section 5.5).

In this section, we will provide the details about these experiments and quantitative comparisons.

5.1. Kinect

The Kinect sensor is a new active depth sensor developed by Microsoft for the Xbox 360 video game platform. The Kinect contains an infrared projector and a monochrome CMOS camera. Kinect interprets depth information from continuously-projected infrared structure rays [14]. The depth estimation is based on the time-of-flight of the infrared rays.

The Kinect collects both RGB video stream and relatively accurate depth sensing stream at a frame rate of $30Hz$. The depth sensing video uses a VGA resolution of 640×480 pixels with 11-bit depth (2048 sensitivity levels).

5.2. Experiment 1: A man

Data: We utilized the Kinect to collect the 3D point sequence of a man, as shown in Fig. 7(a1–a4). The Kinect collected both the 3D point sequence and the RGB video, and our approach only required the 3D point sequence. The 3D point sequence had 22 frames with the frame rate of 5 fps. The depth image's resolution in each frame was 160×120. In each frame, there were 1526 points on the object on average.

Tracking 3D points: Some body segments moved fast, such as the upper arms and the lower legs. Sometimes, their translations between two neighboring frames reached $50cm$. Thus, we had to search each point's candidate matching points within a radius of $50cm$ in the next frame. The total number of candidate matching points reached more than 400 in some frames. The non-rigid matching problem was an MRF optimization problem as shown in Section 3.1. Without the coarse-to-fine procedure in matching, each node would have at most 400 label choices, which cost too much computation time and memory.

Therefore, in the estimation step (Section 3.2), we applied an 8-level coarse-to-fine procedure to match two neighboring frames empirically. In each level, at most 20 candidate matching points were selected. The sphere radiuses in the 8 levels decreased from $50cm$ to $4cm$ exponentially. (The sphere is the range to select candidate matching points for a node; see Section 3.2). Inside the smallest sphere (the radius is $4cm$), there were at most

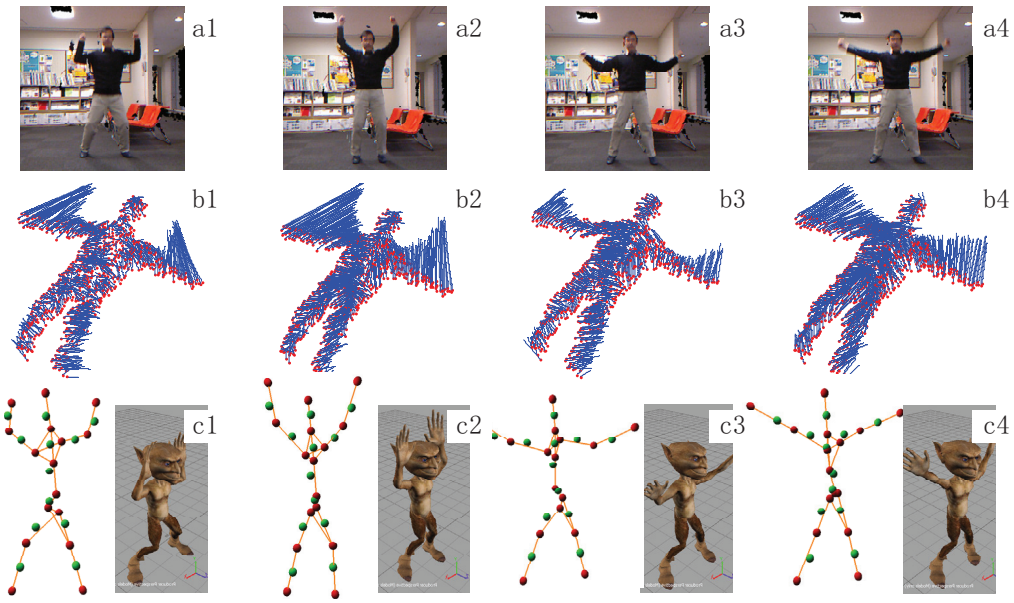

Figure 7: Experimental result of "a man". (a1–a4) show the human's poses in 4 video frames. Lines in (b1–b4) show the deformations from the first frame to the other frames. Only 50% of the 3D points are shown for clarity. (c1–c4) The left figure shows the extracted skeleton with joint points and body segment central points. The right figure shows the corresponding motion capture result.

11 (< 20) points to select. Thus, each node could be matched to any point within $50cm$ in theory.

In the modification step of multiple-frame matching (Section 3.2), we matched the first frame to each frame based on the estimated deformation. Therefore, the candidate matching point's searching range was not so large as in the estimation step. We matched the first frame to each frame in a 6-level coarse-to-fine procedure. In each level, we also selected 20 candidate matching points, and the sphere radiuses decreased from $25cm$ to $4cm$ exponentially.

Fig. 7(b1–b4) shows the modified deformation between the first frame to the other frames.

The skeleton and the motion capture: The man's extracted skeleton had 12 body segments: the head, the chest, the waist, the buttocks, two upper arms, two lower arms, two upper legs and two lower legs. This body segmentation agreed with the common sense. The joint points and body

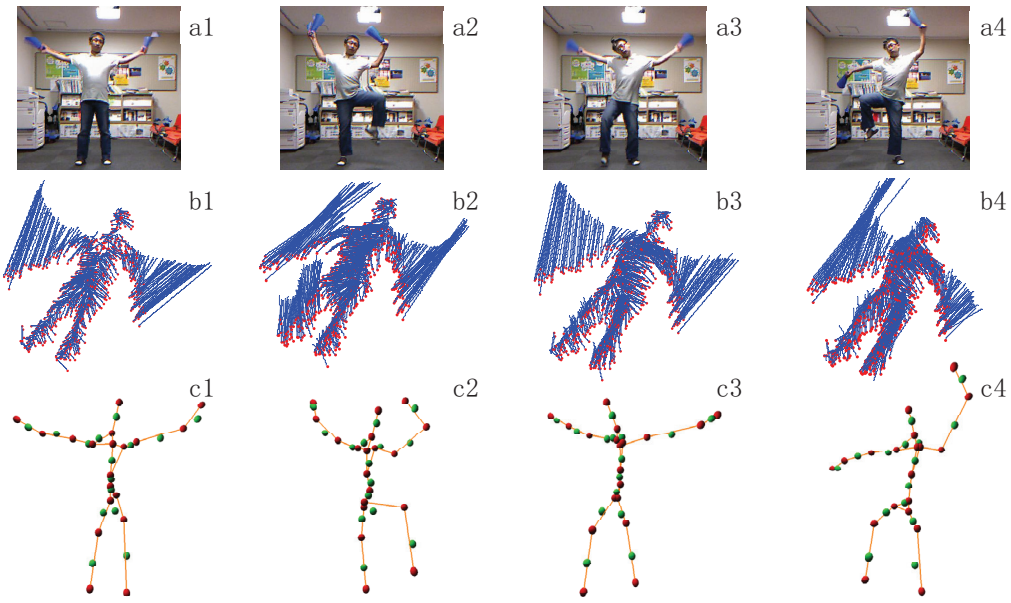

Figure 8: Experimental result of "a man holding two cones". (a1–a4) show the object's poses in 4 video frames. Lines in (b1–b4) show the deformations from the first frame to the other frames. Only 50% of the 3D points are shown for clarity. (c1–c4) show the extracted skeleton with joint points and body segment central points. Note that the two cones are successfully modeled as two individual body segments of the skeleton.

segment centers were important for the human body's structure. Therefore, these points could be utilized as the "markers" in the motion capture. Fig. 7(c1–c4) shows the extracted skeleton and the motion capture result based on the extracted skeleton.

5.3. Experiment 2: A man holding two cones

Data: We utilized the Kinect to collect the 3D point sequence of a man holding two cones, as shown in Fig. 8(a1–a4). The RGB video was not required by our approach. The 3D point sequence had 19 frames with the frame rate of 5 fps. The depth image's resolution was 160×120. In each frame, there were 1292 points on the object on average.

Tracking 3D points: The average 3D point number of the "a man" data and the "a man holding two cones" data in each frame was similar—1526 and 1292 respectively. Their point resolutions were also similar. Therefore, we set the same parameters in the non-rigid matching: The 8-level coarse-to-

fine matching procedure in the estimation step, and the 6-level coarse-to-fine matching procedure in the modification step.

Fig. 8(b1–b4) shows the modified deformation between the first frame to other frames.

The skeleton: The extracted skeleton had 15 body segments: two cones, the head, the right shoulder, the chest, the waist, the buttocks, two upper arms, two lower arms, two upper legs and two lower legs, as shown in Fig. 8(c1–c4). The cones were successfully modeled as two individual body segments. The skeleton contained a right shoulder but no left shoulder. The asymmetric skeleton was due to the asymmetric motion in the data.

5.4. Experiment 3: A box chain

Data: We utilized the Kinect to collect the 3D point sequence of a box chain, as shown in Fig. 9(a1–a8). The RGB video was not required by our approach. The 3D point sequence had 49 frames with the frame rate of 5 fps. The depth image's resolution was 160×120. We utilized the depth information to subtract the man behind the box chain. Thus, in each frame, there were 627 points on the object on average.

Tracking 3D points: The box chain's translations between two neighboring frames were usually less than $10cm$. Therefore, we applied an 4-level coarse-to-fine matching procedure in both the estimation step and the modification step empirically (Section 3.2). In each level, at most 20 candidate matching points were selected. The sphere radiuses in the 4 levels decreased from $10cm$ to $4cm$ exponentially. Larger searching spheres and more levels (as in Experiment 1 and 2) could be utilized in the coarse-to-fine matching procedure, but it increased the computation cost.

Fig. 9(b1–b8) shows the modified deformation between the first frame to other frames.

The skeleton: The extracted skeleton had 4 body segments: the left segment, the left-center segment, the right-center segment and the right segment, as shown in Fig. 9(c1–c8).

5.5. Experiment 4: A vacuum cleaner

Data: We utilized the Kinect to collect the 3D point sequence of a vacuum cleaner, as shown in Fig. 10(a1–a4). The RGB video was not required by our approach. The 3D point sequence had 37 frames with the frame rate of 5 fps. The depth image's resolution was 160×120. In each frame, there were 724 points on the object on average.

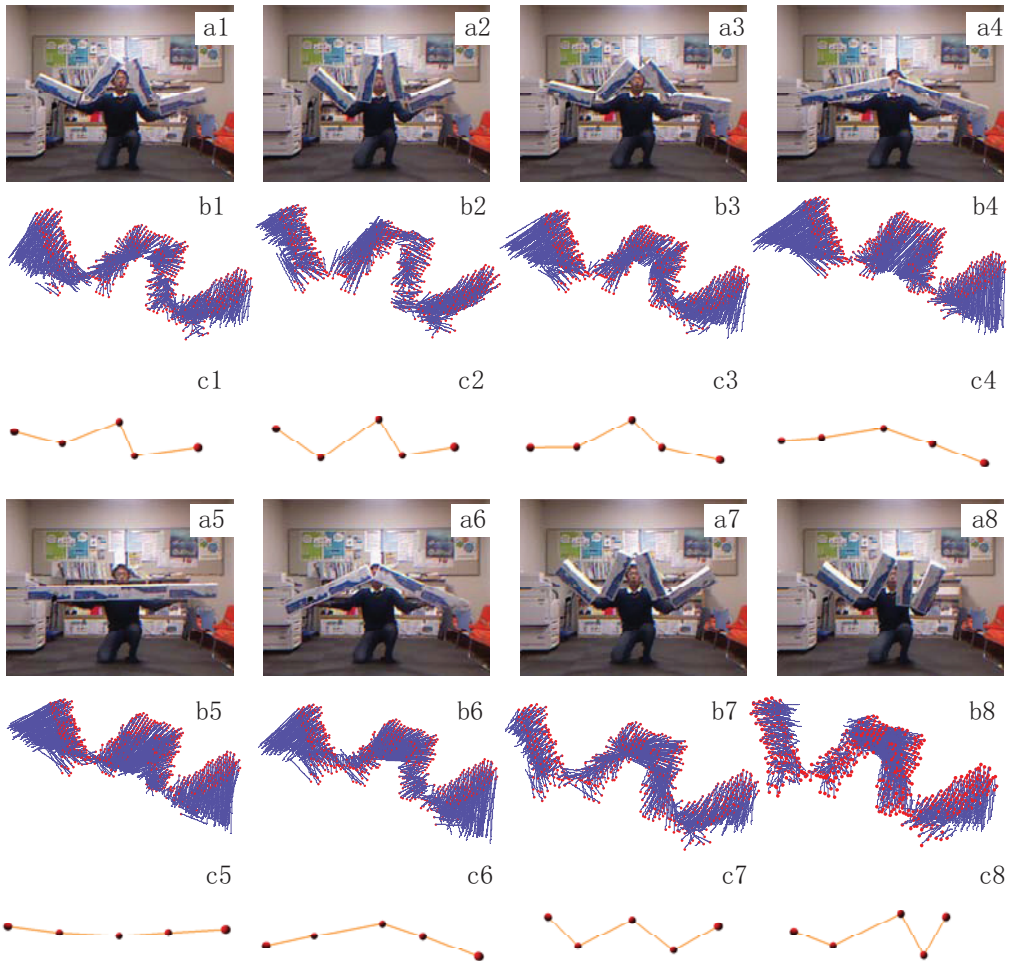

Figure 9: Experimental result of "a box chain". (a1–a8) show the object's poses in 8 video frames. Lines in (b1–b8) show the deformations from the first frame to the other frames. Only 50% of the 3D points are shown for clarity. (c1–c8) show the extracted skeleton with joint points.

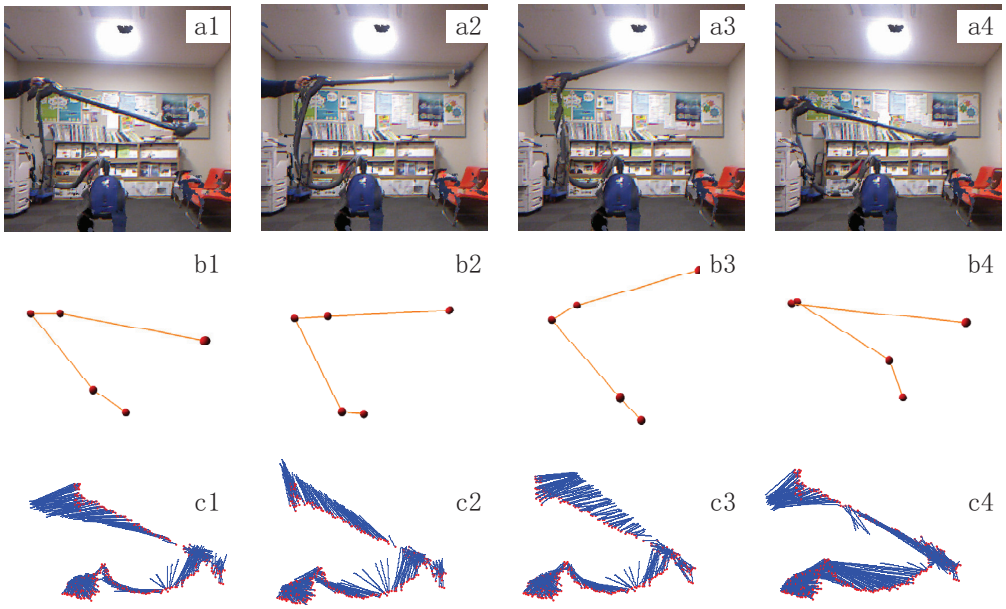

Figure 10: Experimental result of "a vacuum cleaner". (a1–a4) show the object's poses in 4 video frames. Lines in (b1–b4) show the deformations from the first frame to the other frames. Only 50% of the 3D points are shown for clarity. Some matching errors exist in the pipe and the stick, because the pipe and the stick are very thin and black, and thus, the point number is very small in these parts in some frames. (c1–c4) show the extracted skeleton with joint points.

Tracking 3D points: The vacuum cleaner's translations between two neighboring frames were usually less than $14cm$. Therefore, we applied an 5-level coarse-to-fine matching procedure in both the estimation step and the modification step empirically (Section 3.2). In each level, at most 20 candidate matching points were selected. The sphere radiuses in the 5 levels decreased from $14cm$ to $4cm$ exponentially. Larger searching spheres and more levels (as in Experiment 1 and 2) could be utilized in the coarse-to-fine matching procedure, but it increased the computation cost.

Fig. 10(b1–b4) shows the modified deformation between the first frame to other frames.

The skeleton: The extracted skeleton had 4 body segments, including the stick and the pipe, as shown in Fig. 10(c1–c4).

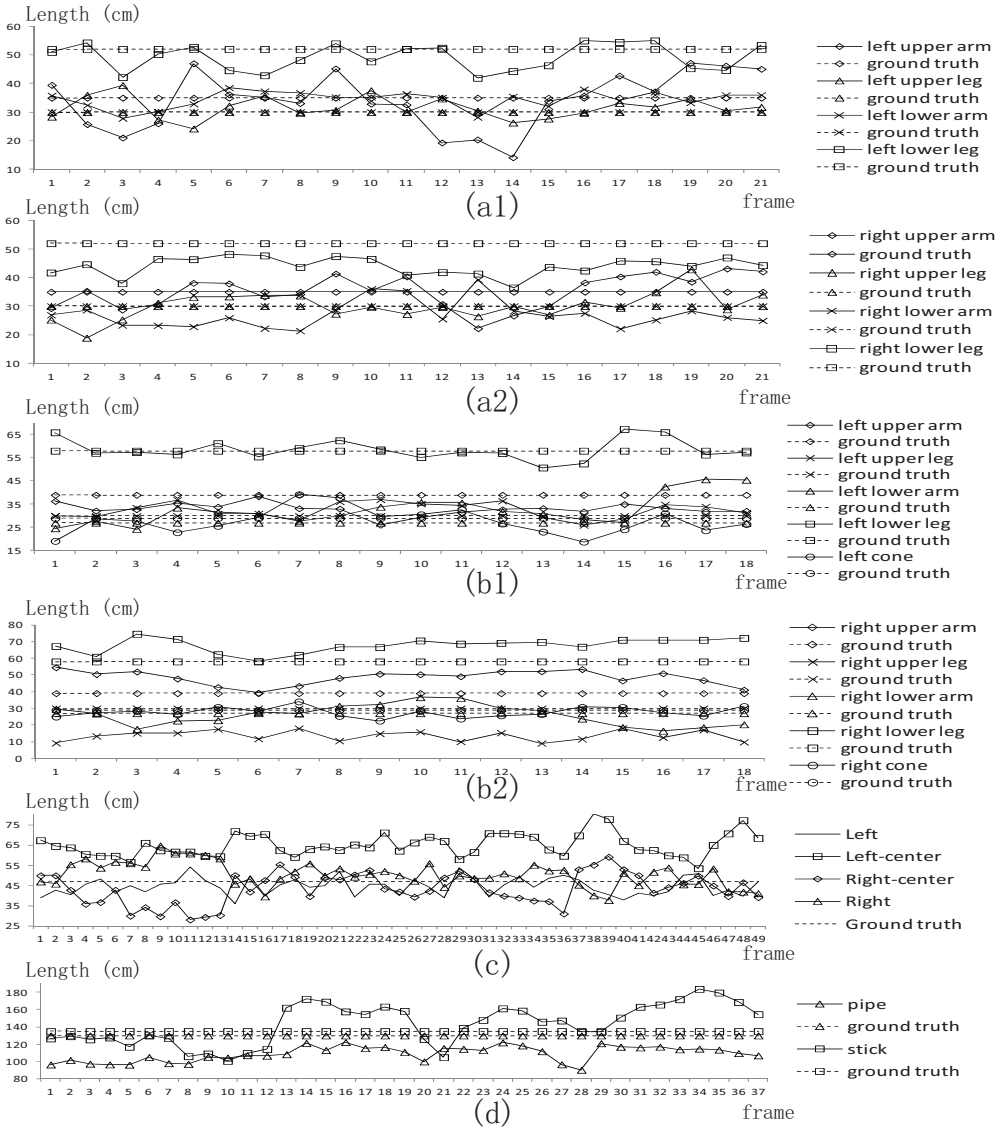

Figure 11: The inaccurate one-frame skeleton: The solid lines show the limbs' length of "a man" in Experiment 1 (a1,a2); the limbs' and the cones' length of "a man holding two cones" in Experiment 2 (b1,b2); the four segments' length of "a box chain" in Experiment 3 (c); the stick's and pipe's length of "a vacuum cleaner" in Experiment 4 (d). The dashed lines show the ground truth measured manually. The segments' length vibrates due to joint points' tracking errors.

Segment name	Extracted length	Ground length	Accuracy
A man			
left upper arm	34.1	35.0	97.45%
left upper leg	31.5	31.1	98.61%
left lower arm	34.5	28.9	80.74%
left lower leg	49.2	47.8	97.11%
right upper arm	35.2	35	99.58%
right upper leg	30.1	31.1	96.91%
right lower arm	27.1	28.9	93.60%
right lower leg	44.0	47.8	92.08%
A man holding two cones			
left upper arm	32.9	39.1	84.10%
left upper leg	32.4	29.8	91.15%
left lower arm	32.6	27.1	79.41%
left lower leg	58.5	57.0	97.33%
left cone	27.6	29.4	94.03%
right upper arm	48.4	39.1	76.27%
right upper leg	13.6	29.8	45.58%
right lower arm	25.9	27.1	95.64%
right lower leg	67.8	57.0	81.10%
right cone	27.6	29.4	94.00%
A box chain			
left	44.6	47.0	94.85%
left-center	65.0	47.0	61.76%
right-center	43.3	47.0	92.18%
right	50.5	47.0	92.57%
A vacuum cleaner			
pipe	109.0	129.3	84.3%
stick	143.2	135.2	94.11%

Table 1: Evaluation of segments' length

5.6. Results and evaluation

To evaluate the extracted skeleton's accuracy, we selected some semantically correct body segments from the skeleton, and measured their real length manually as the ground truth. We took the segment's average length in all frames as the extracted length, and compared the extracted length with the ground truth. The reason was that one-frame skeletons were inaccurate. The segments' length in these skeletons were greatly affected by the joint points' tracking errors (as shown in Fig. 11).

However, there is no measurement to evaluate the correctness of a complex skeletal structure (such as the skeleton of human beings) for the following two reasons: (1) one object can be subjectively divided into different number of body segments. *E.g.* A person's trunk can be considered as consisted of the chest, the waist and the buttocks, or just as one whole body segment. (2) We can use different skeletal structures to represent the same body segment. *E.g.* The person's trunk can be represented as a rectangle, a stick, or a "X" shape. Intuitively, only the limbs and the cones in hands have only one skeletal structure hypothesis—a stick.

In Experiment 1–2, our approach could not detect the axial rotations and tiny deformation on the wrist and the ankle, due to the data's resolution and noise. Thus, in the extracted skeleton, the hand was a part of the upper arm, and the foot (as well as the shoe) was a part of the lower leg. Therefore, the upper arm's length was set as the distance from the elbow to the palm. The upper leg's length was set as the distance from the crotch to the knee along the inner thigh. The lower arm's length was set as the distance from the axilla (the sleeve's crotch) to the elbow. The lower leg's length was set as the distance from the knee to the arch. Note that all the measurement was done with the clothes, as the data utilized in the experiment was with clothes on. Therefore, for accuracy, we counted the wear's thickness in the measurement. The shoe elongated the lower leg. The clothes made the body fatter, which shortened the lower arm. The trousers made the crotch lower and thus shortened the upper leg.

We admit that the anatomical length is a good measurement to evaluate the skeleton of human beings. However, compared to the human-model-based pose estimation, the unsupervised skeleton extraction's potential application is to discover the "unknown" dynamic skeleton of any "unfamiliar" object without any prior knowledge. Thus, we evaluate our algorithm by how much the extracted skeleton objectively reflect the actual object deformation and

the actual segment length in appearance. Therefore, the anatomical length is not utilized.

In Experiment 3, the box chain was successfully divided into four segments. The segments' extracted length and the real length were compared for evaluation. In Experiment 4, the vacuum cleaner was divided into four segments, the pipe segment, the stick segment, the handle segment and the box segment. As the handle segment's length was greatly affected by the arm's motion, and the box segment did not cover the whole body of the vacuum cleaner after background subtraction, we just selected the pipe segment and the stick segment as reliable segments for evaluation.

The segments' extracted length in Experiment 1–4 is evaluated in Table 1.

Generally, the unsupervised learned skeleton successfully reflects the object's true articulated structure. However, the automatically extracted skeletal pose cannot be as accurate as the model-based pose estimation, and the reasons are summarized as follows: (1) The small intra-segment deformation (such as the changes of clothes' wrinkles) brings errors to the body segmentation. (2) The clothes smooth the sharp limbs motion, which makes it difficult to determine the joints' positions. *E.g.* Crotches of trousers and sleeves smooth the motion of arms and legs. As a result, sometimes, the upper arms contain small parts of the chest, and the upper legs are shorter than usual. (3) Noise causes some small errors in the 3D non-rigid matching.

Although noise brings some error into the 3D point tracking, the error is not accumulated over frames. Besides, our tracking is not robust to large occlusion. Only the 3D point cloud sequence without other information (such as the RGB color) is not sensitive enough to detect the column-shape segment's axial rotation (*E.g.* arm's axial rotation). We assume two points on the same body segment should keep similar distances over frames, but the assumption fails when the expansion and contraction of the object size exist. *E.g.* A man is blowing a balloon.

5.7. Quantitative comparisons and evaluation

In this section, we utilized Sequence 5 to compare our method with the marker-based motion capture system. Moreover, we conducted the 3D point matching with five sets of different parameters to evaluate the coarse-to-fine strategy's performance in the time/memory cost reduction.

Experimental settings: We utilized the Kinect to collect the 3D point sequence of the upper body (Sequence 5), as shown in Fig. 13(a1–a5). The RGB video was not required by our approach. The 3D point sequence was

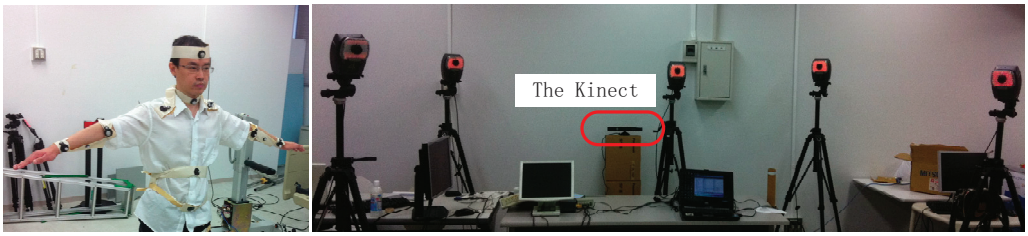

Figure 12: The marker-based motion capture system, the Kinect and markers on the body. Five hark cameras are used.

collected with the frame rate of 6.67 fps. The resolution of the depth image was 160×120. In each frame, there were 1221 points on the object on average. Meanwhile, the marker-based motion capture system tracked 10 markers on the upper body with the frame rate of 100 fps. These markers were fixed on the forehead, the neck, the shoulders, the elbows, the wrists, the chest center, and the abdomen center. Fig. 12 illustrates our experimental settings.

Comparison with the marker-based motion capture: We conducted the quantitative comparisons between our system and marker-based motion capture system. For the extracted skeleton in each frame, the head length, upper arm length, lower arm length and arm total length were calculated. Because the length of body segments in the marker-based motion capture system only measured the distance between two markers (not the body segment's real length), we used the standard deviation of the body segment length to evaluate the stability of the extracted skeleton. Then, we compared the animation performances in the marker-based motion capture and in our extracted-skeleton-based motion capture.

The average 3D point number of the "a man" data and the "the upper body" data in each frame was similar—1526 and 1221 respectively, and their point resolutions were also similar. Hence, we set the same parameters in the non-rigid matching: The 8-level coarse-to-fine matching procedure in the estimation step, and the 6-level coarse-to-fine matching procedure in the modification step. The sphere radiuses decreased from $50cm$ to $4cm$ exponentially. Fig. 13(b1–b5) shows the modified deformation between the first frame to other frames, and Fig. 13(c1–c5) shows the extracted skeleton.

To evaluate the stability of the extracted skeleton, we compared the segment length's standard deviation in our extracted skeleton with the length variation in the marker-based motion capture. Table 2 shows the standard

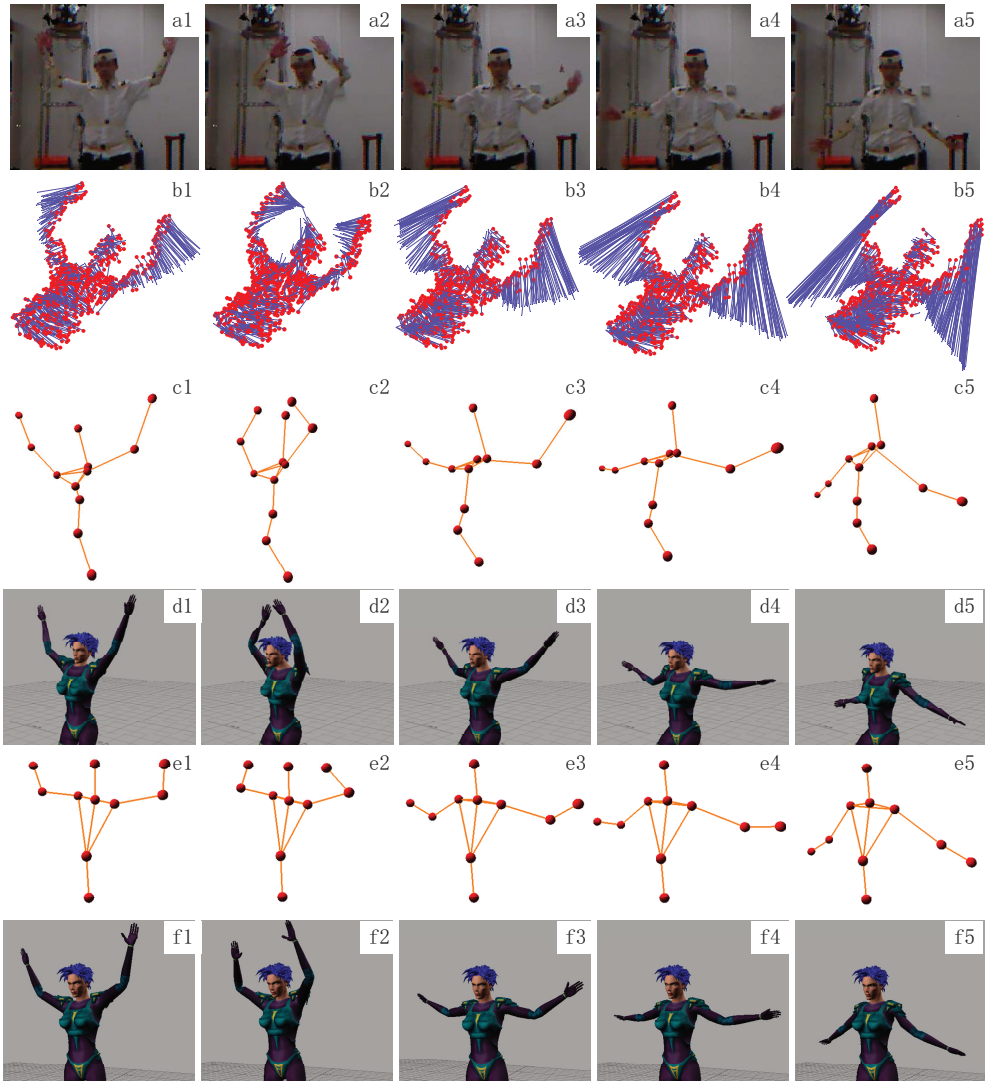

Figure 13: Comparison with marker-based motion capture. (a1–a5) show the object's poses in 5 video frames. Lines in (b1–b5) show the deformations from the first frame to the other frames. (c1–c5) show the extracted skeleton with joint points. (d1–d5) show the motion capture result based on the extracted skeleton. (e1–e5) show the skeleton obtained from marker-based motion capture system. (f1–f5) show the marker-based motion capture result.

	Extracted skeleton from the Kinect data (cm)	Marker-based motion capture (cm)
Left upper arm	2.54	4.68
Left lower arm	3.56	0.75
Right upper arm	6.48	3.66
Right lower arm	4.85	1.21
Head	2.41	0.28

Table 2: The body segment length's standard deviation in the extracted skeleton and the marker-based motion capture

deviations of body segment lengths in our extracted skeleton and in the marker-based motion capture system. The markers on the shoulder were fixed on the T-shirt (not directly fixed on the body), which made a relatively large standard deviation of the upper arm's length in the marker-based motion capture.

Time/memory cost analysis: The 3D point tracking takes most of the memory space and computational time. We propose a coarse-to-fine 3D point matching method to reduce its time/memory cost. To evaluate the performance of the coarse-to-fine strategy, the 3D point matching with five sets of different parameters was utilized to matching points between two neighboring frames, as follows:

1. The single-level matching: For each point, at most 90 candidate matching points were uniformly selected inside the sphere with its radius of $50cm$.

2. 6-level 20-candidate coarse-to-fine matching: In each level, at most 20 candidate matching points were selected. The sphere radius decreased from $50cm$ to $4cm$ exponentially.

3. 3-level 20-candidate coarse-to-fine matching: In each level, at most 20 candidate matching points were selected. The sphere radius decreased from $50cm$ to $4cm$ exponentially.

4. 6-level 5-candidate coarse-to-fine matching: In each level, at most 5 candidate matching points were selected. The sphere radius decreased from $50cm$ to $4cm$ exponentially.

5. 3-level 5-candidate coarse-to-fine matching: In each level, at most 5 candidate matching points were selected. The sphere radius decreased from $50cm$ to $4cm$ exponentially.

The single-level matching is the 3D point matching without the coarse-

to-fine strategy, and the other four kinds of matching are the coarse-to-fine 3D point matching. The candidate matching point number was set to 90 in the single-level matching due to the memory limitation. Only 1/4 of the 3D points in Sequence 5 (304 points per frame on average) were utilized in the five kinds of 3D point matching to ensure that all the points inside the $50cm$ matching sphere could be selected as the candidate matching points in the single-level matching.

We utilized MATLAB R2008a to realize our system. We tested the real time/memory cost on a computer with Intel(R) Core(TM) i7 CPU M640 2.80GHz. The coarse-to-fine point matching process took most of the computational time and memory in the whole skeleton extraction framework.

We utilized the coarse-to-fine strategy to reduce the large time/memory cost of the point matching process. In each level of the matching, let each node have at most k candidate matching points in total. The variable number for the energy minimization problem is $|V|k$, where $|V|$ is the average node (point) number in the MRF. The matching time cost for each pair of frames is $O(NL|E|k^2)$, where N is the level number, L is the iteration number of the TRW-S to minimize the total energy of the MRF, and $|E|$ is the edge number in the MRF. The total time cost for the 3D point tracking is $O(|F|(N_e + N_m)L|E|k^2)$, where N_e is the level number in the estimation step of the tracking, N_m is the level number in the modification step of the tracking, $|F|$ is the total frame number. Note that we set a energy threshold (0.00001) for the MRF, and the growth of the variable number (k) increased the iteration number L for TRW-S to meet the energy threshold. We stored the state transition matrices, so the memory cost for matching is $O(|E|k^2)$.

The real average time/memory cost for the 3D point matching with the five sets of different parameters are shown in Table 3.

The matching performances with the five sets of different parameters as shown in Fig. 14. Some nodes (points in the MRF) in the single-level matching ($k = 90$, $N = 1$) were wrong matched. We set a large pairwise interaction penalty τ to prevent two nodes from matching to the same candidate matching point (in Equation 3). Thus, if there were some many-to-one matching conditions in some local areas, some nodes would be wrong matched. However, in the coarse-to-fine 3D point matching ($N > 1$), the sphere for the candidate-matching-point selection was large in low matching levels, and there were a large number of points inside the sphere, so we avoided the wrong-matching problem by selecting different candidate-matching-point sets from the sphere for different nodes. In high matching levels, this problem

	Candidate matching point number	Level number	Iteration number per level	Time per iteration (second)	Total time (second)	Memory cost (KB)
1	90	1	3813.3	0.09316	355.2	118062.7
2	20	6	1473.9	0.00652	57.67	5834.3
3	20	3	1185.3	0.00665	23.7	
4	5	6	503.3	0.00099	3.0	408.9
5	5	3	578.9	0.00096	1.6	

Table 3: The average time/memory cost of matching two frames

might occur, but the matching errors were reduced within a small range by the previous-level matching. On the contrary, in the single-level matching ($N = 1$), all the points inside the sphere were selected as the candidate matching points, and the neighboring nodes shared their most candidate matching points, so the local many-to-one matching could not be avoided. Therefore, the single-level matching produced some large-range matching errors.

Besides, the computation complexity of the integral-geodesic-distance feature for all the points in each frame is $O(|V|^3)$, where $|V|$ is the node number in the MRF. It took $0.35s$ on average to calculate the integral-geodesic-distance feature for each frame of Sequence 5 ($|V| = 304$, here). The computation complexity of the body segmentation and the joint point determination is $O(|F||V|^2)$. The computation complexity of the skeleton generation is $O(|F||V|)$, where $|F|$ is the total frame number. It took $0.18s$ on average for the body segmentation, the joint point determination and the skeleton generation.

6. Conclusion and discussion

The paper presents a novel skeleton extraction algorithm by using the 3D point sequence, and the extracted skeletons can be easily used for the motion capture. Generally, the unsupervised learned skeleton successfully reflects the object's true articulated structure. Though the unsupervisedly-extracted skeleton cannot be as accurate as the model-based pose estimation, our proposed system can discovery the the dynamic topological structure of the "unknown" object.

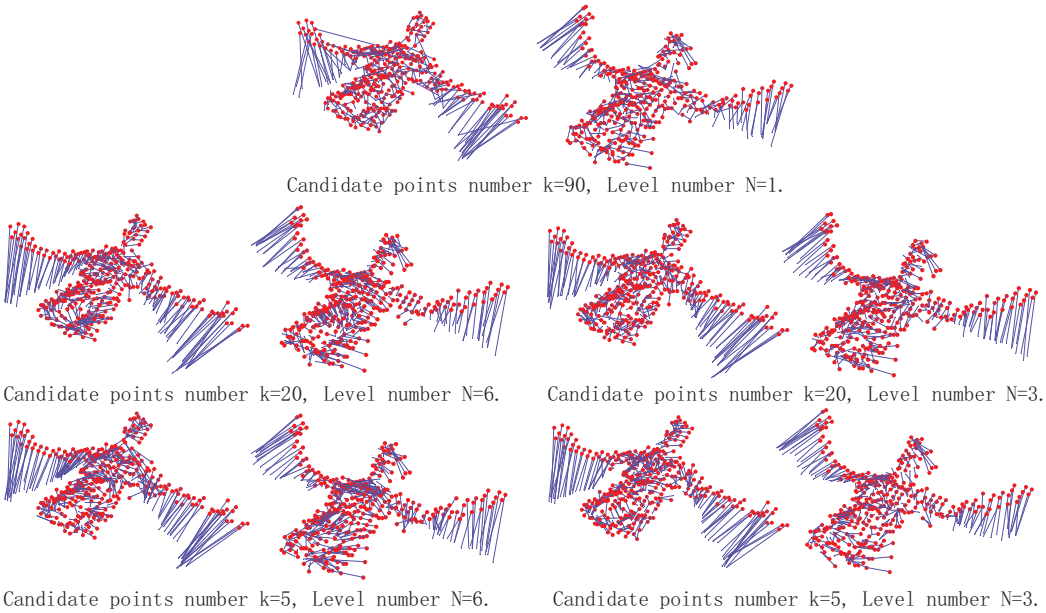

Figure 14: The matching performances with five sets of different parameters. The cause of the matching error in the singe-level matching ($N = 1$) is analyzed in the text.

The integral geodesic distance is utilized as the point feature in the non-rigid matching, and it is a good feature for 3D object retrieval. However, the integral geodesic distance is sensitive to changes of the object's topological structure. *E.g.* If a man folds his hands together, the integral geodesic distance of his hands will decrease greatly. One possible solution is to utilize some 3D local features, but 3D local features have their own problems: (1) Usually, reliable 3D local features only exist in "edges" and "corners", so they are not suitable to deal with smooth-surface objects. (2) Compared to well-constructed 3D models, the Kinect 3D point cloud cannot provide many reliable local features due to noise. Another possible solution is to combine the integral geodesic distance with the RGB color in tracking.

7. Acknowledgement

Thank Prof. Hajime Asama, Mr. Yuki Ishikawa, and Mr. Qi An in the Asama Laboratory, University of Tokyo, for their help in data collection. This work was supported by a Grant-in-Aid for Young Scientists (23700192)

and Strategic Project to Support the Formation of Research Bases at Private Universities :Matching Fund Subsidy from MEXT (Ministry of Education,Culture,Sports,Science and Technology),2008-2012. This work was also partially funded by Microsoft Research.

References

[1] D. A. Ross, D. Tarlow, R. S. Zemel, Unsupervised learning of skeletons from motion, In *Europeon Conference on Computer Vision* (2008) 560–573.

[2] D. A. Ross, D. Tarlow, R. S. Zemel, Learning articulated structure and motion, In *International Journal of Computer Vision* 88 (2) (2010) 214–237.

[3] J. Yan, M. Pollefeys, A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (5) (2008) 865–877.

[4] J. Yan, M. Pollefeys, Automatic kinematic chain building from feature trajectories of articulated objects, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2006) 712–719.

[5] P. Tresadern, I. Reid, Articulated structure from motion by factorization, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2005) 1110–1115.

[6] D. Ramanan, D. Forsyth, K. Barnard, Building models of animals from video, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (8) (2006) 1319–1334.

[7] A. G. Kirk, J. F. O'Brien, D. A. Forsyth, Skeletal parameter estimation from optical motion capture data, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2005) 782–788.

[8] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, W. Burgard, Learning kinematic models for articulated objects, In *In Proc. of the International Conference on Artifical Intelligence* (2009) 1851–1856.

[9] D. Attali, J.-O. Lachaud, Delaunay conforming iso-surface, skeleton extraction and noise removal, In *Computational Geometry* 19 (2001) 175–189.

[10] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve skeleton extraction from incomplete point cloud, In *In Proceedings of ACM SIGGRAPH* 28 (3) (2009) Article 71, 9 pages.

[11] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, T.-Y. Lee, Skeleton extraction by mesh contraction, In *In Proceedings of ACM SIGGRAPH* (2008) 1–44.

[12] G. Aujay, F. Hétroy, F. Lazarus, C. Depraz, Harmonic skeleton for realistic character animation, In *ACM SIGGRAPH Symposium on Computer Animation* (2007) 151–160.

[13] S. Schaefer, C. Yuksel, Example-based skeleton extraction, In *Eurographics Symposium on Geometry Processing* (2007) 153–162.

[14] Wikipedia. Kinect—Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Kinect, 2011. [Online; accessed 19–September–2011].

[15] J. Shi, C. Tomasi, Good features to track, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1994) 593–600.

[16] G. K. Cheung, S. Baker, T. Kanade, Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2003) 77–83.

[17] C.-W. Chu, O. C. Jenkins, M. J. Matarić, Marker-less kinematic model and motion capture from volume sequence, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2003) 475–482.

[18] J. Sturm, V. Pradeep, C. Stachniss, Vision-based detection for learning articulation models of cabinet doors and drawers in household environment, In *IEEE International Conference on Robotics and Automation* (2010) 362–368.

[19] Y. Song, L. Goncalves, P. Perona, Unsupervised learning of human motion, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (7) (2003) 814–827.

[20] J. Shotton, A. Fitzgibbon, M. Cook, A. Blake, Real-time human pose recognition in parts from single depth images, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011) 1297–1304.

[21] I. Oikonomidis, N. Kyriazis, A. A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect, In *British Machine Vision Conference* (2011) 1–11.

[22] A. Shekhovtsov, I. Kovtun, V. Hlaváč, Efficient mrf deformation model for non-rigid image matching, In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007) 1–6.

[23] A. Shekhovtsov, I. Kovtun, V. Hlaváč, Efficient mrf deformation model for non-rigid image matching, In *Computer Vision and Image Understanding* 112 (2008) 91–99.

[24] L. Torresani, V. Kolmogorov, C. Rother, Feature correspondence via graph matching: Models and global optimization, In *Europeon Conference on Computer Vision* (2) (2008) 596–609.

[25] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (10) (2006) 1568–1583.

[26] K. J. Lee, D. Kwon, I. D. Yun, S. U. Lee, Deformable 3d volume registration using efficient mrfs model with decomposed nodes, In *British Machine Vision Conference* (2008) 1–10.

[27] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, In *Advances in Neural Information Processing Systems 14 (NIPS)* (2002) 849–856, Cambridge, MA,.