IPSJ Transactions on Computer Vision and Applications Vol.6 12–24 (Mar. 2014)

[DOI: 10.2197/ipsjtcva.6.12]

Research Paper

Tree-structured Mesoscopic Surface Characterization for Kinematic Structure Estimation from 3D Video

Tomoyuki Mukasa[1,a)] Shohei Nobuhara[1] Tony Tung[1] Takashi Matsuyama[1]

Received: May 27, 2013, Accepted: December 16, 2013, Released: March 24, 2014

Abstract: This paper presents a new approach to estimate the kinematic structure underlying a sequence of 3D dynamic surfaces reconstructed from multi-view video. The key idea is a mesoscopic surface characterization with a tree-structure constraint. Combined with different levels of surface characterizations, namely macroscopic and microscopic characterizations, our mesoscopic surface characterization can cope with shape estimation errors and global topology changes of 3D surfaces from the real world to estimate kinematic structure. The macroscopic analysis focuses on global surface topology to perform temporal segmentation of 3D video sequence into topologically-coherent sub-sequences. The microscopic analysis operates at the mesh structure level to provide temporally consistent mesh structures using a surface alignment method on each of the topologically-coherent sub-sequences. Then, the mesoscopic analysis extracts rigid parts from the preprocessed 3D video segments to establish partial kinematic structures, and integrates them into a single unified kinematic model. Quantitative evaluations using synthesized and real data demonstrate the performance of the proposed algorithm for kinematic structure estimation.

Keywords: mesoscopic surface characterization, kinematic structure, motion analysis, 3D video

1. Introduction

Nowadays 3D surface capture from multi-view videos, known as 3D video [9], [12], [13], has become a popular technique in computer vision and graphics communities. 3D video consists of a temporal series of 3D surfaces reconstructed on a frame-by-frame basis from multi-view videos of real-world objects. Unlike motion-capture technique, 3D video can record the target surface geometry and texture as-is, and realizes full-3D, free-viewpoint rendering of the real scene.

In 3D video, captured surface meshes of different frames usually have different mesh structures, i.e., different number of vertices and mesh connectivity. This fact indicates that no vertex-to-vertex correspondence between different meshes is available in general. Moreover 3D video can only capture surfaces that are visible from cameras, i.e., the envelopes. That is, not only the local mesh structure but the global surface topology also can change through the entire 3D video sequence (**Fig. 1**).

On the other hand, once a time-invariant structure describing the object motion is obtained, several applications of 3D video will be possible, such as motion analysis of dance or sports activities, kinematic editing of captured data, inter-frame mesh data compression, etc. Based on these observations, this paper is aimed at estimating a kinematic structure as a time-invariant physical system underlying captured time-varying mesh structures (**Fig. 2**).

To achieve this goal, we propose a framework as follows. We first segment the entire 3D surface mesh sequence into time intervals based on a macroscopic characterization in order to keep the global topology unchanged in each interval. Next we apply a temporal surface alignment method to produce time-coherent mesh models for each interval based on a microscopic mesh char-

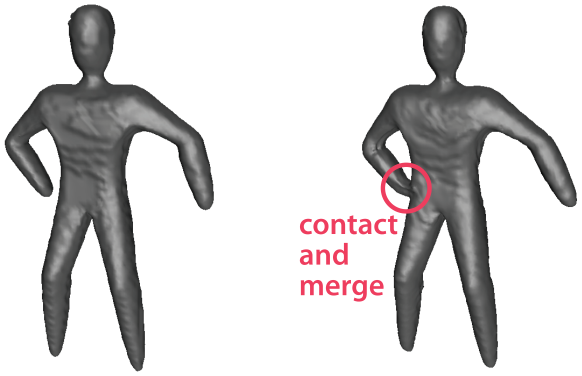

Fig. 1 The global topology change of surface. Left is a genus-0 surface while Right is a genus-1 due to the body contact.

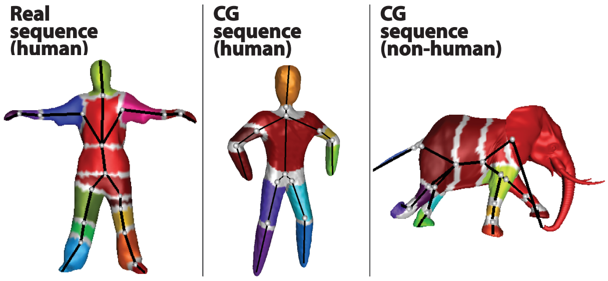

Fig. 2 Estimated kinematic structures. Colored regions indicate rigid parts. White areas denote non-rigid portions corresponding to joints (Left data is courtesy of University of Surrey and INRIA. Right data is courtesy of Computer Graphics Group at MIT).

[1] Graduate School of Informatics, Kyoto University, Kyoto 606–8501, Japan

a) mks@vision.kuee.kyoto-u.ac.jp

acterization. These two steps enable us to extract time-invariant body part candidates based on a mesoscopic mesh characterization. Moreover candidates from each interval are integrated into a unified tree-structured articulated model so as to describe the entire 3D video sequence best. Here we assume that each rigid part of the target shape can be represented by a generalized cylinder model [11], and is connected to others via non-rigid joint parts to form a single tree-structure.

The advantage of our approach is twofold. (1) It explicitly manages global topology changes. (2) It can handle any numbers of articulated parts as long as they can be modeled by a tree-structured generalized cylinders. For example, we can assume that the proposed model is valid for vertebrate animals and arthropods. This point is justified empirically by experimental evaluation using non-human data as described later.

2. Related Work

Several existing methods have been proposed to acquire kinematic structures from 3D model sequences in the literature [2], [6], [8], [19]. These methods can be categorized by their units of motion description that here we refer to as “processing units.” In Ref. [8], the processing unit is a voxel, and its motion is assumed to be relatively small compared to that of the whole body. With this modeling, it is difficult to discriminate motion derived from rigid motion and small scale non-rigid motion. Hence this approach cannot robustly estimate a kinematic structure of real-world data containing a mixture of non-rigid motions and reconstruction errors.

To solve this issue, Refs. [2], [6] and [19] proposed algorithms utilizing higher level of motion descriptions, i.e., mesoscopic surface characterizations, such as sub-surfaces and primitive volume. These descriptions acheive more robust modeling of surface motions, but final results are not guaranteed to reflect the global topology of the object. This is because their algorithms are bottom-up oriented and have no constraints on building up kinematics structure. In addition, a deforming mesh with constant mesh connectivity is given as input in Ref. [2] while our entire framework can cope with temporally varying mesh connectivity.

Thus, this paper proposes an algorithm based on a mesoscopic sub-surface modeling with an explicit top-down constraint which connects sub-surfaces to form a tree structure which is expected to be valid for regular animals including humans. The key difference compared to prior studies is the combination of top-down constraint and bottom-up sub-surface motion estimation from per-vertex surface motion flows.

3. Algorithm Overview

We introduce the following three-step approach (see **Fig. 3**) to estimate the kinematic structure from 3D surface mesh sequence:

Step I: 3D mesh sequence segmentation by macroscopic mesh characterization.

Step II: Time-coherent 3D mesh generation by microscopic mesh characterization.

Step III: Kinematic structure estimation by mesoscopic mesh characterization.

Step I is based on a global geodesic distance distribution (μ histogram, described later) to identify time intervals where the global topology of individual 3D surface mesh is unchanged.

Step II exploits this consistency to establish per-vertex microscopic surface motion flows which enable a reference mesh model to be aligned to other mesh surfaces in the interval.

Step III uses this microscopic motion information to estimate

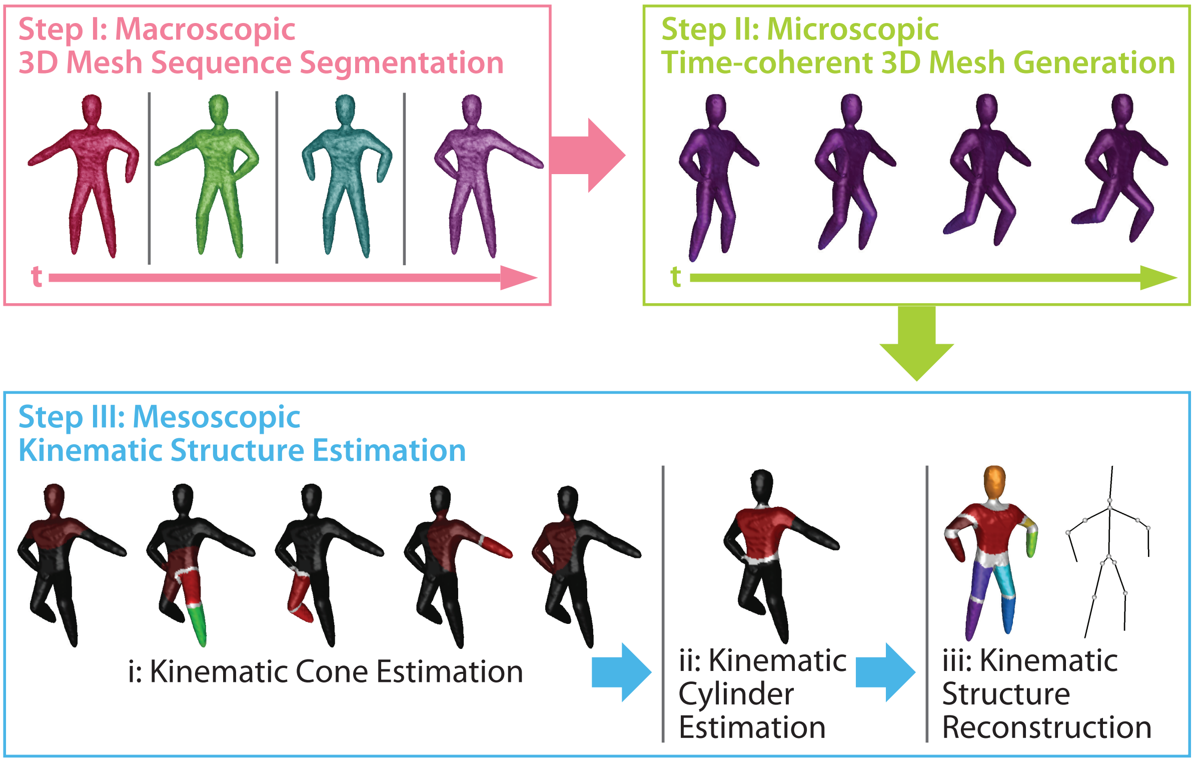
Step I: Macroscopic
3D Mesh Sequence Segmentation
t
Step II: Microscopic
Time-coherent 3D Mesh Generation
t
Step III: Mesoscopic
Kinematic Structure Estimation
i: Kinematic Cone Estimation
ii: Kinematic Cylinder Estimation
iii: Kinematic Structure Reconstruction

Fig. 3 Overview of our algorithm.

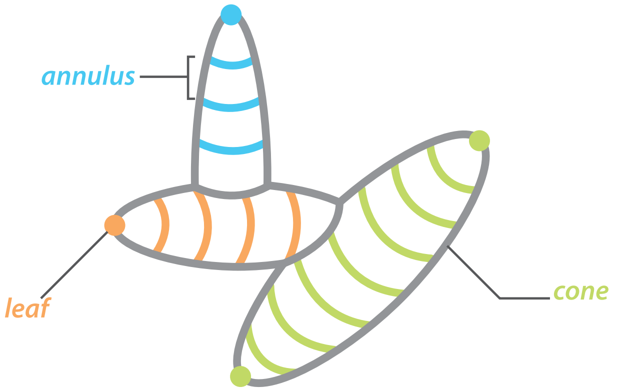

Fig. 4 Tree-structured generalized cylinders with cone and annulus representation.

the underlying kinematic structure. The key idea here is to model the object using generalized cylinders consisting of ring-shaped sub-surfaces called annulus used as mesoscopic surface characterization in our algorithm. By representing the articulated motion of the object by collections of rigid annuli motions, and enforcing them to form a tree-structure, this step models the object surface motion by tree-structured generalized cylinders motion (**Fig. 4**). Here, we call a cylinder as cone, and its apex as leaf.

In this modeling, the kinematic structure is provided as the estimated tree structure. One challenge in this strategy is the fact that the complete kinematic structure is not always estimated from a single interval. Depending on the apparent object motion, each interval will provide a different incomplete kinematic structure. For example, if the object keeps its upper body static in an interval, we have no clue about its kinematic structure from its motion obviously.

Hence the main design factors on modeling the object kinematic structure by a tree-structured generalized cylinders are twofold: (1) from which part of the tree should the estimation starts, and (2) how to integrate incomplete kinematic structures from different intervals into a unified one which can model all the surfaces in different intervals.

In the following sections, we propose a leaf-oriented approach which estimates the kinematic structure from the leaves to the root, and integrates incomplete kinematic structures by leaves-to-leaves matching. This is because leaves can be robustly estimated at extremal points of 3D mesh surfaces as a result of macroscopic and microscopic processings of **Step I** and **Step II** as shown later.

4. Step I: Macroscopic 3D Mesh Sequence Segmentation

Let us denote a 3D surface mesh sequence by $M(t) = \{V(t), E(t)\}$ where $V(t)$ and $E(t)$ denote the set of vertices and edges respectively at time t. We define a continuous function μ: $M(t) \rightarrow \mathbb{R}$ as the sum of geodesic distances from a vertex to all the others:

$$\mu(v) = \sum_{u \in V(t)} g(v, u), \tag{1}$$

where $v, u \in V(t)$, and $g(v, u)$ is the geodesic distance between v and u. If the global surface topology changes, the distribution of μ changes drastically because topological changes introduce or remove shortest paths on $M(t)$. On the other hand, since we define μ as an integral over the surface, its distribution is robust to local surface deformations caused by object motions or per-vertex reconstruction errors (e.g., holes or surface glitches). Hence the distribution of μ is only sensitive to the surface global topology (which is related to the number of critical points, according to the Morse theory [14]), and can be used as a macroscopic surface characterization.

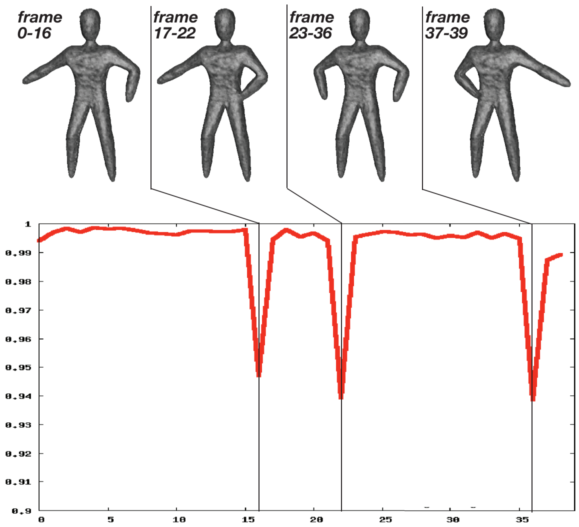

Fig. 5 Correlations of μ histograms.

Based on this observation, we introduce the following correlation coefficient index using covariance Cov and standard deviation σ to measure surface-to-surface topological difference:

$$C(H^{\mu}(t), H^{\mu}(t+1)) = \frac{\mathrm{Cov}(H^{\mu}(t), H^{\mu}(t+1))}{\sigma(H^{\mu}(t))\sigma(H^{\mu}(t+1))}, \tag{2}$$

where $H^{\mu}(t)$ denotes the histogram of μ distribution on $M(t)$. We use a fixed number of bins (100 in practice) in the histogram, and normalize it. **Figure 5** plots $C(H^{\mu}(t), H^{\mu}(t))$ and empirically proves that this index captures well topological changes at sharp drops of values. With this index, we subdivide the entire 3D mesh sequence into intervals $I_i = [t_i^{begin}, t_i^{end}]$ $(i = 1, ..., N(I))$ in each of which the global mesh topology does not change.

5. Step II: Microscopic Time-coherent 3D Mesh Generation

The second step is to estimate 3D motion flows in each I_i. We have formulated this as a mesh alignment problem [4], [5], [10], [20], [23]. In particular, we employ "tracking by deformation" approach and utilize the geodesic mapping [20] for constraint of deformation. This strategy is particularly computationally efficient as the μ distribution computed in **Step I** can be reused in the geodesic mapping.

According to the Morse theory [14], a continuous function μ defined on a surface can characterize the surface topology using its critical points. Here, we use again the sum of geodesic distance as μ to identify extremal surface points that coincide to highly concave or convex regions [7], [21], [22].

By assuming that such highly convex regions correspond to end-points of body parts, we utilize them as "leaves" of the tree-structure to be estimated (Fig. 4). In addition, we can utilize them

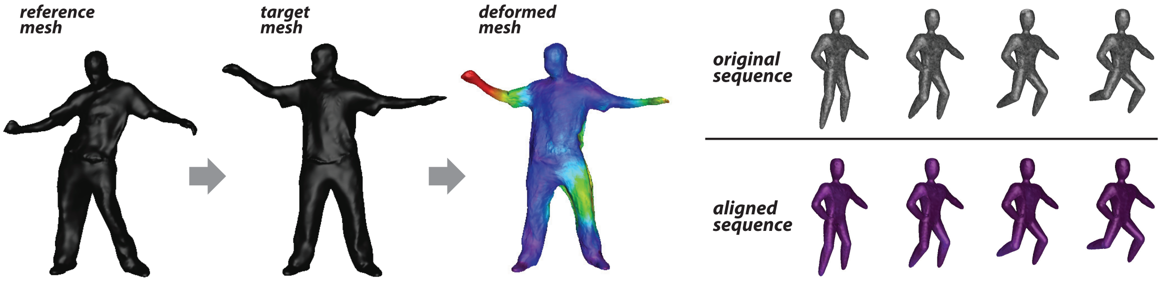

Fig. 6 Original input data and alignment results. Left: warm color indicates large deformation. Right: temporally inconsistent mesh sequence (top) and aligned sequence after surface deformation with fixed mesh geometry (bottom).

as coordinate origins of the geodesic mapping, since they are consistent in intervals where global mesh topology is kept unchanged.

In this step, we first find the leaves, i.e., the time-coherent critical point set which defines deformation invariant geodesic coordinates in each interval. Then we choose a reference mesh for each interval, and align it to other meshes in the same interval according to the vertex-to-vertex matching based on geodesic coordinates. As the result, we obtain time-coherent 3D meshes for each interval, and this is equivalent to obtain 3D motion flows.

5.1 Leaf Detection

Tips of body extremities, e.g., fingertips, appear in $\mu(v)$ distribution on the surface as local maxima. By assuming the frame rate of 3D video is high enough for capturing the object's motion, these points can be tracked based on Nearest Neighbor with Euclidean distance over time in each interval I_i.

Leaves are defined as microscopic (local) surface feature points $v_n^{leaf}(t) \in L(t)$ at local maxima of $\mu(v)$ at each time frame t, and $L(t)$ $(n = 1, ..., N(L(t)))$ as the set of leaves of tree shaped kinematic structure to be estimated. Here we introduce a function $\epsilon(v, L)$ which returns a leaf $v_n^{leaf} \in L$ nearest to v in Euclidean space. We then find a mapping $F_{t,t+1}$ between $L(t)$ and $L(t+1)$ as follows:

$$\begin{aligned} F_{t,t+1} = \{ \ < v_n^{leaf} &\in L(t), v_m^{leaf} \in L(t+1) > | \\ v_n^{leaf} &= \epsilon(v_m^{leaf}, L(t)), \\ v_m^{leaf} &= \epsilon(v_n^{leaf}, L(t+1)), \\ \mathbf{n}(v_n^{leaf}(t)) &\cdot \mathbf{n}(v_m^{leaf}(t+1)) > 0\}, \end{aligned} \tag{3}$$

where $\mathbf{n}(v)$ stands for the normal vector of v. We start the above mapping from the first frame in I_i, and propagate the result frame by frame to the last frame in I_i. As the result, we find fixed number of leaves L_i in each interval I_i, and establish their correspondences across frames in I_i.

5.2 Definition of Geodesic Coordinates by Leaves

As we have made correspondences between leaves v_n^{leaf} over frames in each interval I_i in the previous section, we can define interval specific geodesic coordinate $\boldsymbol{x}_i^{geo}(v(t))$ whose element is the geodesic distances to the leaves as follows:

$$\boldsymbol{x}_i^{geo}(v(t)) = (g(v(t), v_1^{leaf}(t)), ..., g(v(t), v_{N(L)}^{leaf}(t))). \tag{4}$$

Since the global topology does not change in I_i, the distance between each vertex $v(t) \in V(t)$ and each of leaves $v_n^{leaf}(t)$ does not change through I_i. Therefore geodesic coordinate is invariant against mesh deformations in each interval I_i [20].

5.3 Time-coherent 3D Mesh Generation Based on Geodesic Mapping

We choose a 3D video frame at the middle of each interval as the reference mesh M_i^{ref} because it is likely to represent an average posture of the object in the interval and hence is likely to minimize the deformation artifacts. We then obtain a mesh sequence $M'_{t \in I_i}$ with time invariant surface mesh connectivity by deforming M_i^{ref} to fit to all the other 3D mesh $M_{t \in I_i}$ in the same interval.

We first establish per-vertex correspondence between successive frames $f_{t \to t+1}^{geo} : v(t) \in V(t) \to v(t+1) \in V(t+1)$ by finding a point $v(t+1)$ such that:

$$v(t+1) = \underset{v' \in V(t+1)}{\arg\min}(d(v(t), v')), \tag{5}$$

where $d(v(t), v')$ stands for the geodesic distance between vertices in different frames in the same interval as follows:

$$d(v \in V(t1 \in I_i), v' \in V(t2 \in I_i)) = \|\boldsymbol{x}_i^{geo}(v) - \boldsymbol{x}_i^{geo}(v')\|. \tag{6}$$

If we move each vertex of $M_{t \in I_i}$ according to $f_{t \to t+1}^{geo}$, the local mesh structure on $M_{t \in I_i}$ will not be preserved because Eq. (5) returns the geodesically nearest vertex without considering connectivities among vertices.

To preserve the local mesh structures, we apply as-rigid-as-possible (ARAP) deformation method [16] to M_i^{ref} using $f_{i \to j}^{geo}$ as soft constraint. ARAP deformation preserves surface details by keeping rigidities of each local area around vertex while the whole mesh is deformed so as to satisfy the soft constraint. With the combination of geodesic mapping and ARAP deformation, we can align 3D mesh sequence $M_{t \in I_i}$ by $M'_{t \in I_i}$ in which mesh topology and connectivity are guaranteed to be equal to the reference mesh M_i^{ref} (**Fig. 6**).

6. Step III: Mesoscopic Kinematic Structure Estimation

Up to this point, we have segmented the sequence into intervals using macroscopic surface characteristics, and found the per-vertex surface motion flows in each interval using microscopic surface features. We then extract the kinematic structure by the following processes utilizing a mesoscopic mesh characterization.

Step III.i Estimation of kinematic cones on mesh surface
Step III.ii Recursive estimation of kinematic cylinders on mesh surface
Step III.iii Reconstruction of kinematic structure

In **Step III.i**, we find cones on the 3D mesh such that each of which includes a leaf, and segment each of them (Fig. 4) into rigid or non-rigid areas. We call the segmented cone as "kinematic cone."

In **Step III.ii**, we cut off the kinematic cones from the mesh surface and define "kinematic cylinders" on the remaining area until the entire mesh surface are segmented into rigid or non-rigid areas.

In **Step III.iii**, we obtain partial kinematic structures from kinematic cones and cylinders, then integrate them into a unified kinematic structure valid through the entire sequence.

6.1 Step III.i: Kinematic Cone Estimation

We estimate the kinematic cone for each leaf by iterating the following process:

Step III.i.a Mesoscopic surface characterization by cone and annulus representation.
Step III.i.b Clustering annuli into kinematic cones.
Step III.i.c Temporal integration of kinematic cones.

Step III.i.a introduces a mesoscopic surface characterization: cone and annulus representation of 3D surface. The annulus representation is aimed at robustly managing small per-vertex motions and noises acquired in **Step II**.

Step III.i.b examines the rigidity of annuli, and cluster them into sub-meshes corresponding to rigid body parts or joints that form kinematic cones.

Step III.i.c matches and transfers non-rigid labels in kinematic cones found independently in each interval.

6.1.1 Step III.i.a: Mesoscopic Surface Characterization

By assuming that the object has several body extremities each of which can be represented by a generalized cylinder model constrained to form a tree structure, we introduce a mesoscopic surface characterization with annulus and cone (Fig. 4). In **Step II**, we have computed leaves that are local maxima of $\mu(v)$ corresponding to tips of body extremities. We define a set of annuli for each leaf. An annulus A_n^k is the ring-shaped sub-surface where the geodesic distance from a leaf v_n^{leaf} is within a range r^k. Borders between annuli corresponds to a *disc* in the generalized cylinder model. A cone consists of a set of contiguous annuli and corresponds to a generalized cylinder.

We partition the reference mesh M_i^{ref} in each time interval into cones as a connected annuli. The partition of M_i^{ref} is transferred to other 3D surface meshes $\{M'\}_i$ in the interval I_i automatically as the mesh geometry is consistent in each interval.

After obtaining a consistent set of cones in each interval, we match them across intervals and integrate them into a unique set of cones consistent through the entire sequence.

6.1.1.1 Cone and Annulus Representation

For each leaf v_n^{leaf} of M_i^{ref} ($n^{leaf} = 1, ..., N(L)$), we define a cone S_n^{cone} consisting of annuli A_n^k independently. An annulus A_n^k consists of connected vertices whose geodesic distances from a leaf v_n^{leaf} are within the range $r^k = [\delta_g(k), \delta_g(k+1))$, where δ_g is an heuristic geodesic width [*1]. A cone and its annuli are obtained by the following processes simultaneously:

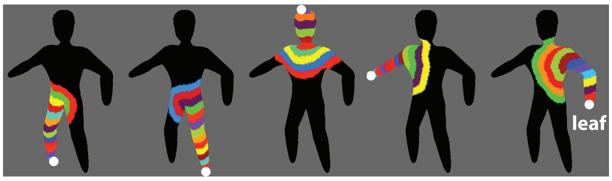

Fig. 7 Cone and annulus partition. Black colored areas are not included in a cone, each of other colors represents different annulus, and white circles indicate leaves of each cone.

(1) Partition M^{ref} into sub-surfaces $\{B_n^k\}$ where the geodesic distance from a leaf v_n^{leaf} is within the range r^k.
(2) In each B_n^k, we merge connected vertices into sub-surfaces.
(3) Let S_n^{cone} consist of B_n^0.
(4) Grow S_n^{cone} by adding the adjacent sub-surface B_n^k until B_n^{k+1} is disjoint.

At the end of the above process, we consider each of sub-surfaces B_n^k in S_n^{cone} as an annulus A_n^k. By applying this for each annulus, we obtain cones corresponding to all the body extremities (**Fig. 7**).

Note that a sub-surface B_n^k is not guaranteed to be a connected sub-surface. If B_n^{k+1} is disjoint, it suggests that it partially covers at least two body extremities, and B_n^k is located at their conjugation area. Therefore, as shown in Fig. 7, cones are overlapping when the object is tree-shaped.

6.1.1.2 Cone Matching Across Intervals

We have assumed that the object consists of body extremities that can be described as cones, and form a time invariant tree shaped global structure. On the other hand, the global mesh topology changes when the body extremities touch each other, e.g., when the subject puts his/her hand on the hip (Fig. 1).

The change of the global mesh topology affects whether each body extremity appear as cone or not. That is, a cone found in an interval can disappear in other intervals due to body contacts. Hence, we collect cones appearing constantly through the entire sequence, and map them to the mesh of other intervals in which the cones are not observed.

We first match leaves and its corresponding cones across intervals in the same manner as Section 5.1. Suppose we have found cones $\{S_n^{cone}\}_i$ $(n = 1, ..., n_i^{\max}))$ in I_i, $\{S_m^{cone}\}_{i+1}$ $(m = 1, ...m_{i+1}^{\max})$ in I_{i+1}, where I_i and I_{i+1} are contiguous each other and their bounding time frames are t_i^{end} and t_{i+1}^{begin} respectively. When $n_i^{\max}$ and $m_{i+1}^{\max}$ differs, here we suppose $n_i^{\max} > m_{i+1}^{\max}$ for convenience, some of cones $\{S_n^{cone}\}_i^{surplus}$ do not have corresponding cones in $\{S_m^{cone}\}_{i+1}$. We map $\{S_n^{cone}\}_i^{surplus}$ on the 3D mesh $M_{t_{i+1}^{begin}}$ as in the following Section 6.1.1.4.

6.1.1.3 Erroneous Cone Rejection

Even if a cone S_n^{cone} is matched to a cone S_m^{cone}, the matching is not reliable when the distance between their leaf positions is longer than a threshold θ:

$$||v_n^{leaf}(t_i^{end}) - v_m^{leaf}(t_{i+1}^{begin})|| > \theta. \tag{7}$$

In this case, S_n^{cone} or S_m^{cone} should be an erroneous cone which often appears when touching body parts bend at joints and form

[*1] We employed $1/25th$ of $\max(g(v(t^{ref}), v_n^{leaf}(t^{ref})))$ in practice.

a loop in global mesh topology (see center of **Fig. 8**). Notice that the parameter θ should be determined based on the assumption on the object motion, i.e., how far the body leaves can move in a frame.

To reject the erroneous cone from S_n^{cone} and S_m^{cone}, we introduce an assumption in which each rigid body has a bone inside along its long axis direction because it is natural for the tree-shaped object, e.g., most of animals (**Fig. 9**). Based on this assumption, an erroneous cone must have less number of annuli inside. We count the number of annuli in S_n^{cone} and S_m^{cone}, and reject the one with less annuli. If two cones have a same number of annuli inside, then we select the one in which μ ranges more widely. If the ranges are equivalent, then we randomly select either of them.

By introducing this erroneous cone rejection mechanism, we can be free from false sub-trees not corresponding to any body extremities, and construct unique kinematic structure as we discuss later.

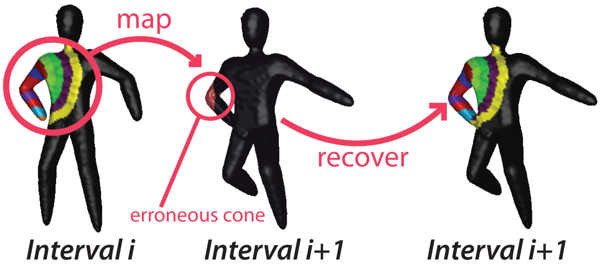

Fig. 8 Processing of erroneous cone. The erroneous cone consists of only one annulus while the corresponding cone in the interval i has 12 annuli.

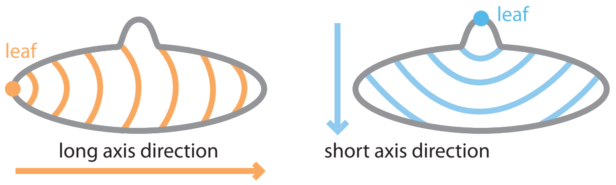

Fig. 9 Annuli with long axis and short axis.

6.1.1.4 Cone Consistency

As the result of the above two steps, we have obtained reliable cones. However, some of them are not observed in intervals in which global mesh topology is affected by contact of body extremities. For each body extremity $S_n^{cone} = (V_n^{cone}, E_n^{cone})$ in I_i such that it cannot be matched to any other cones in a contiguous interval I_{i+1}, we find the corresponding area on the mesh in I_{i+1} as follows:

(1) make a mapping between $U(t_i^{end}) = \{V(t_i^{end}) \setminus V_n^{cone}(t_i^{end})\}$ and $V(t_{i+1}^{begin})$:

$$\{< v_{j'} \in V(t_{i+1}^{begin}), v_j \in U_i > | \epsilon(v_{j'}, U_i), \mathbf{n}(v_{j'}) \cdot \mathbf{n}(v_j) > 0\}, \quad (8)$$

(2) make $v_{j'}$ part of $A_n^k \in S_n^{cone}$ if $v_{j'}$ is mapped onto v_j in $A_n^k \in S_n^{cone}$ (Fig. 8),

where $\epsilon(v, U)$ returns a vertex $v_j \in U$ nearest to v in Euclidean space.

We obtain temporally consistent numbers of cones through the entire sequence as the result of this step.

6.1.2 Step III.i.b: Annulus Clustering

We cluster annuli in each cone into rigid areas $\{S_p^{rigid}\}$ or non-rigid areas $\{S_q^{non-rigid}\}$ to compose kinematic cone (**Fig. 10**). The clustering starts from A_n^0 which contains the leaf v_n^{leaf} and relies on two criteria: "self-error" and "cast-error."

6.1.2.1 Self-error

For each annulus $A_n^k = (V_n^k(t), E_n^k)$, we can compute the rigid motion $D_{t\to t+1}(A_n^k) = \{R_{t\to t+1}(A_n^k), T_{t\to t+1}(A_n^k)\}$ where $R_{t\to t+1}(A_n^k)$ and $T_{t\to t+1}(A_n^k)$ denote the rotation and translation of A_n^k by minimizing residual error e:

$$\begin{aligned} &e(V_n^k(t), V_n^k(t+1)) \\ &\quad = \sum_{v \in V_n^k(t)} \|(R_{t\to t+1}(A_n^k)v(t) + T_{t\to t+1}(A_n^k)) - v(t+1)\|. \end{aligned} \quad (9)$$

We call the sum of this as "self-error" and denote it by $e^{self}(A_n^k(t))$ which shows the apparent non-rigidity of A_n^k.

6.1.2.2 Cast-error

The rigid motion of a sub-surface (an annulus or clustered an-

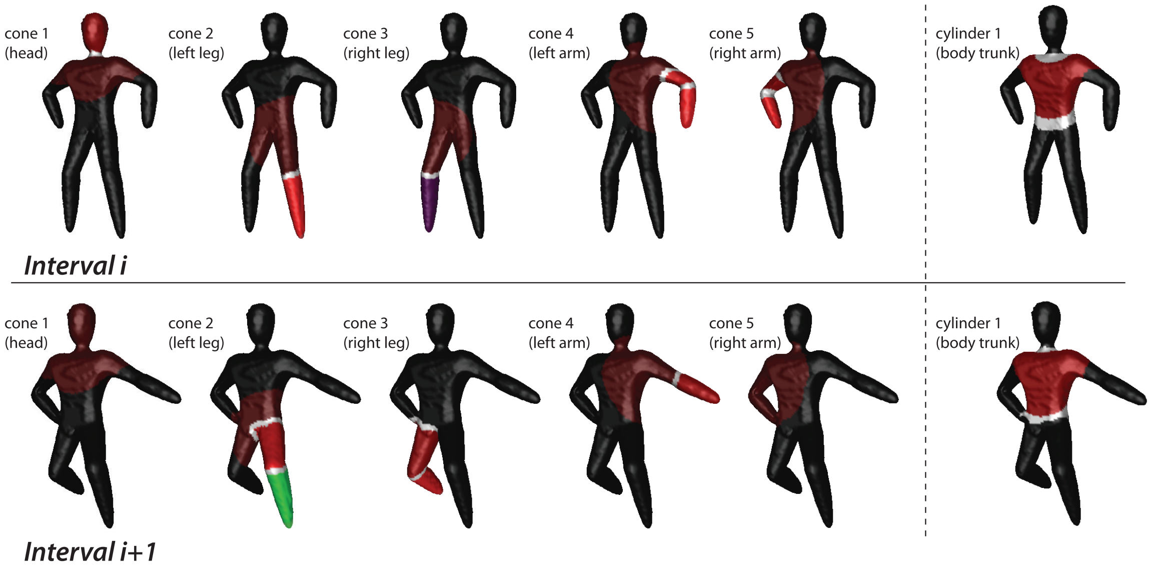

Fig. 10 Kinematic cone and cylinder for each body extremity and body trunk. Colored and white regions indicate rigid and non-rigid areas respectively. Black regions are out of range of cones.

nuli) $S = (V_n^S(t), E_n^S)$ adjacent to A_n^k is given by $D_{t\to t+1}(S) = \{R_{t\to t+1}(S), T_{t\to t+1}(S)\}$. We compute the residual error e for an annulus A_n^k by assuming that A_n^k follows $D_{t\to t+1}(S)$ and each vertex $v \in V_n^S$ is moved to $v' \in V'^S_n$:

$$e(V_n^k, V'^k_n) = \sum_{v\in V_n^S} \|(R_{t\to t+1}(S)v + T_{t\to t+1}(S)) - v'\|. \qquad (10)$$

We call the sum of the residual errors as "cast-error" and denote it by $e^{cast}_{t\to t+1}(A_n^k)$:

6.1.2.3 Annulus Clustering Based on Merge-error

We cluster the annuli by examining the similarity between adjacent sub-surfaces (an annulus or clustered annuli) based on the self-error and the cast-error. We start the clustering from each annulus A_n^0 which contains the leaf v_n^{leaf} in each cone S_n^{cone} independently with assuming that A_n^0 is rigid.

Even if an annulus A_n^k follows the rigid motion of adjacent sub-surface, its cast error will be augmented as its self error increases. To elicit the effect of merging A_n^k to the adjacent sub-surface from $e^{cast}_{t\to t+1}(A_n^k)$ affected by $e^{self}_{t\to t+1}(A_n^k)$, we employ "merge-error" which is the ratio of cast-error to self-error:

$$e^{merge}_{t\to t+1}(A_n^k) = e^{cast}_{t\to t+1}(A_n^k)/e^{self}_{t\to t+1}(A_n^k), \qquad (11)$$

The closer $e^{merge}_{t\to t+1}(A_n^k)$ to 1, the more the motion of A_n^k and its adjacent sub-surface is assumed to be similar (See discussions in Section 7.3 below). In other words, we merge an annulus to the cluster in question if its non-rigidity is comparable to that of the cluster. We merge A_n^k to its adjacent sub-surface only if the merge-error $\overline{e^{merge}}(A_n^k)$ averaged over I_i is smaller than a threshold θ_r:

$$\overline{e^{merge}}(A_n^k) = \frac{1}{t_i^{end} - t_i^{begin} + 1} \sum_{t\in I_i} e^{merge}_{t\to t+1}(A_n^k) < \theta_r. \qquad (12)$$

If the average is greater than θ_r, we do not merge the annulus A_n^k to any other annuli and restart the clustering from the adjacent annulus A_n^{k+1}. Hence, annuli in non-rigid motion are not merged to any others. Note that clustered sub-surfaces are not labeled as rigid or non-rigid at this stage.

Even if an annulus A_n^k has low self-error value and seems to follow its own rigid motion, we regard A_n^k as a part of a joint if its motion is significantly different from each adjacent annulus A_n^{k-1} and A_n^{k+1}. Based on this point of view, we examine the number of annuli in each clustered sub-surfaces after the above clustering, If the number of annuli in a clustered sub-surface is only one, we label the sub-surface as a non-rigid area $S_q^{non-rigid}$ with other contiguous non-rigid sub-surfaces, otherwise we label it as a rigid area S_p^{rigid} (**Fig. 11**). Now each cone is converted to kinematic cone C_n^{kinema} consisting rigid and non-rigid areas (Fig. 10).

6.1.3 Step III.i.c: Temporal Integration of Kinematic Cones

Since we defined kinematic cones for each interval independently, the number of kinematic cones and the partitions of them are not consistent between intervals. In this section, we match kinematic cones across intervals and integrate them to a consistent set of kinematic cones.

Now we have kinematic cones $\{C_n^{kinema}\}$ of each body extremity found in each interval. However, they are only reflecting the motion in each interval, i.e., non-rigid areas (joints) can be found only in an interval in which the joints are in motion. Thus it is not guaranteed that all kinematic cones share a same non-rigid and rigid areas. We transfer non-rigid labels of annuli among corresponding kinematic cones in different intervals to obtain a unique kinematic structure K in the later section.

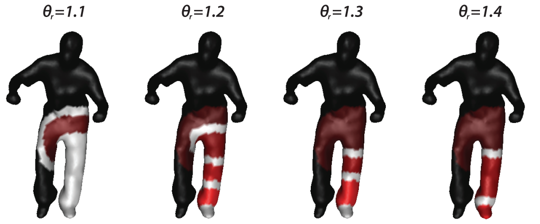

Fig. 11 Clustering results at different thresholds: θ_r is empirically defined so as to maximize the number of clustered sub-surfaces. In the above example, we chose 1.2 for the threshold. (Data is courtesy of University of Surrey.)

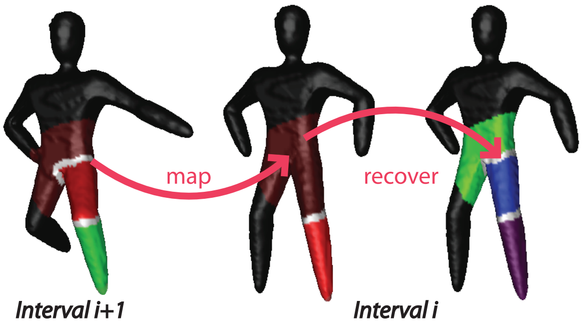

Fig. 12 Non-rigid label transfer in kinematic cones.

As we have already matched cones across intervals in Section 6.1.1.4, each annulus in cone has corresponding annulus in all the other intervals. For each set of matched cones, we transfer all non-rigid labels of A_n^k in an interval to the annuli having index k in all the other intervals (**Fig. 12**). This is because we assume the object approximated by articulated rigid bodies in which lengths and connectivities of bones are fixed.

This process returns the minimal but sufficient rigid and non-rigid areas in kinematic cones (the upper half of **Fig. 13**), which can describe the entire motions of body extremities through a sequence, as opposed to techniques which rely on prior knowledge on the kinematic structure [23] which can be inaccurate or overdetermined.

6.2 Step III.ii: Kinematic Cylinder Estimation

6.2.1 Undefined Area

The kinematic cones corresponding to body extremities in different intervals are made to share same rigid and non-rigid areas in the previous steps. However, some rigid areas in the cones can partially overlap (as described in Section 6.1.1.1), and some areas on surface can remain without belonging to any cones (remaining area) at this step. We call the union of such overlapping and remaining area as "undefined area."

As we mentioned before, we assume that the object can be modeled by a tree-shaped articulated rigid body. Therefore, overlapping areas indicate that branches of the kinematic structures exist there, and the trunk of the tree-shaped object (e.g., body trunk of human) is remaining to be defined as a set of bone or

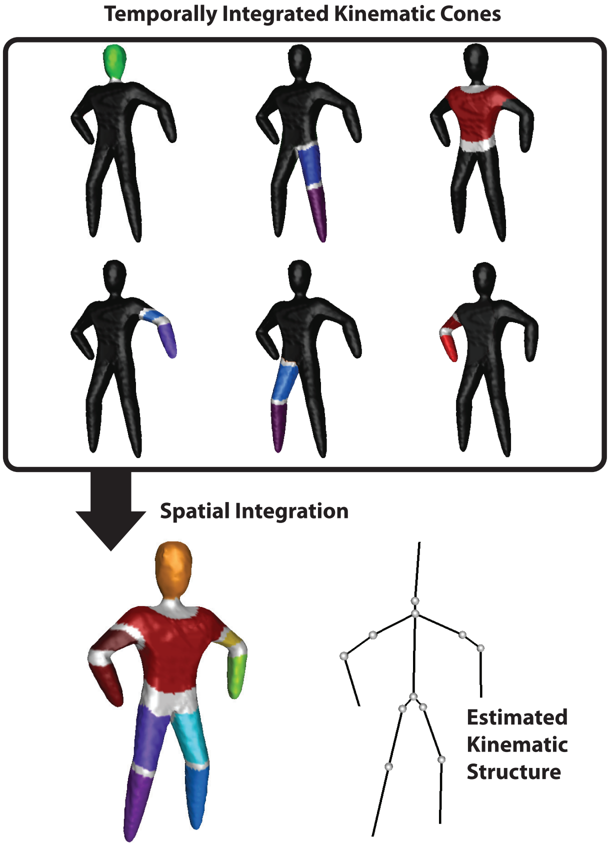
Temporally Integrated Kinematic Cones
Spatial Integration
Estimated Kinematic Structure

Fig. 13 Unique kinematic structure for entire 3D mesh sequence.

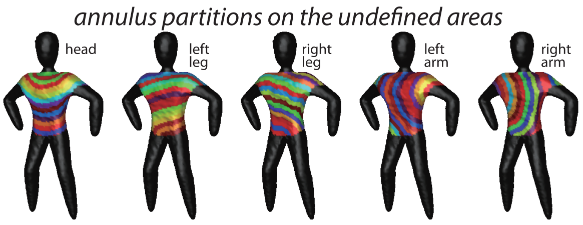

Fig. 14 Extensions of kinematic cones.

non-rigid area in the undefined area.

6.2.2 Cylinder and Annuli Representation

For each kinematic cone, we first cut off the overlapping rigid area which are the most geodesically distant from each leaf of the kinematic cone, because the area has any possibility to be the part of a specific cone up to this point. We define a set of annuli in the undefined area by the same manner as in previous steps (Section 6.1.1.1 and Fig. 7) as the extention of each kineamtic cone (**Fig. 14**).

Also considering that the most of animals has their trunk parts as an extension of a body extremity, we take these extensions as the candidates of partition on the undefined area.

6.2.3 Kinematic Cylinder and its Temporal Integration

Here, we again employ the assumption for reliable annulus partition which we introduced in Section 6.1.1.3. We choose the extension of the kinematic cone which has most wide-ranging geodesic values inside, because it provide the longest, and therefore probable bone inside. We then cluster contiguous and connected annuli of the chosen extension in the same manner as Section 6.1.1 and 6.1.2 (Fig. 10), and call it kinematic cylinder. Clustering results are transferred among intervals same as in Section 6.1.3, and finally we obtain a kinematic cylinder consistent through the entire sequence.

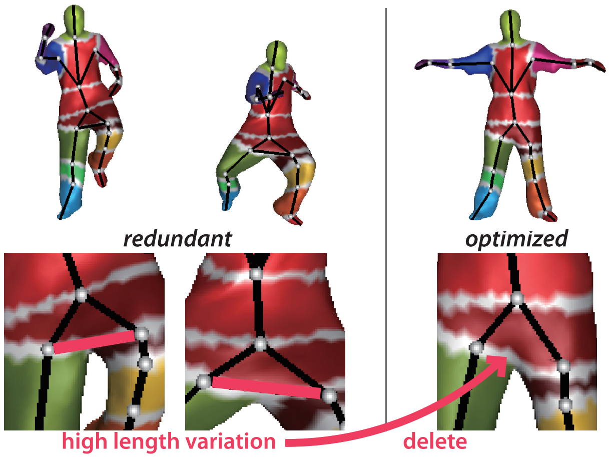

Fig. 15 Redundant bone rejection by minimum spanning tree.

6.2.4 Recursive Kinematic Cylinder Estimation

We run this kinematic cylinder acquisition process recursively until all the area on the surface are defined with bone or non-rigid area. Finally, we have a unique tree-structured set of kinematic cones and cylinders $\{C_n^{kinema}\}$ reflecting the underlying kinematic motion through the entire sequence (the upper half of Fig. 13).

6.3 Step III.iii: Kinematic Structure Reconstruction

We reconstruct partial kinematic structures K^{cone} and $K^{cylinder}$ corresponding to each kinematic cone and cylinder in $\{C_n^{kinema}\}$ respectively. We then integrate them by defining bones among them under the constraint of tree-structure.

6.3.1 Partial Kinematic Structure Reconstruction

In each kinematic cone, we first estimate joint positions by computing the centroids of the non-rigid areas in each frame. Then, we define bones as links between pair of joints whose corresponding non-rigid areas are adjacent to the rigid area. Consequently, we have partial kinematic structure for each kinematic cone and cylinder.

6.3.2 Integration of Partial Kinematic Structures

For each partial kinematic structure $K^{cylinder}$ of kinematic cylinder, we connect partial kinematic structure K^{cone} which corresponds to kinematic cone or cylinder is adjacent to the cylinder. We first simply define bones between the joints $p^{cylinder}$ in $K^{cylinder}$ and joints p^{cone} in K^{cone} if their corresponding non-rigid areas are adjacent to each other or a same rigid area.

As a result, we can observe redundant bones at branch points (left of **Fig. 15**). This is against the assumption that the object can be approximated by tree-shaped articulated rigid body. We formulate the deletion problem of the redundant bones as a minimal spanning tree problem. We take the current kinematic structure as a graph in which nodes and edges are corresponding to joints and bones respectively. For each bone, we compute the length variation through the sequence, and assign the value to corresponding edge as its weight. We find optimal topology by finding a minimal spanning tree from the graph.

We then add bones between each of leaves and a joint of which corresponding non-rigid area which is the geodesically nearest

Table 1 Average computation time per frame for various sized datasets. Even though macroscopic and microscopic steps are not needed for *free* sequence as it has fixed mesh geometry, we note their computation time for comparison in brackets.

sequence	vertices	edges	macroscopic	microscopic	mesoscopic	total
free	4,284	8,564	(3.0sec)	(18.3sec)	1.2sec	22.5sec
human	around 13,000	around 27,000	60.7sec	125.1sec	4.0sec	190.0sec
head	around 68,500	around 137,000	23.5min	47.6min	0.3min	71.4min

to the leaf. The length of the bones can vary in each frame at this stage. We enforce fixed bone length constraints to the entire skeleton through the sequence based on an existing method [1]. Finally we obtain a unique tree-shaped kinematic structure from the entire sequence (Fig. 13).

7. Experiments

We evaluated the proposed method using synthesized and real data (Fig. 2). The synthesized data are the *human* sequence and the *elephant-gallop* sequence [18], and the real data are the *free*, the *lock*, *head* [5], [17] and the *crane* sequence [23] representing a real human performance.

The algorithm was implemented in C++ using an Intel Core-i7 2.3 GHz computer. The computation times for datasets of different number of vertices are given in **Table 1**. These results indicates the running time is roughly proportional to the square of the number of vertices.

7.1 Quantitative Evaluation

We evaluated the proposed method from the quantitative point of view using the *human* sequence. The *human* sequence contains 36 frames, and is reconstructed from multi-view video virtually capturing a skeleton-based synthetic animation in which a human model is waving its body extremities. Each of the reconstructed 3D meshes contains around 13,000 vertices and small noises on the surfaces.

We then compared the ground truth and kinematic structure extracted from the *human* sequence by the proposed method. The distance between the estimated and the ground truth positions of each joint was less than 3% of the object's height (1,500 mm) with a small range of deviation: e.g., for the right knee, the average error was 27.2 mm (**Fig. 16**, **Table 2**). These results show that our kinematic structure estimation is reasonably stable and accurate.

In the Fig. 16, we can observe the all errors increasing around the 12th frame for all joints. This is caused by the residual error of the deformation based on geodesic mapping (Section 5.3). After the 12th frame, the right hand of *human* touches the body, and the tip cannot be found in the mesh model hereafter. As the result, geodesic mapping is destabilized as the dimension of the geodesic coordinates is reduced, and the residual error of the deformation has increased.

7.2 Qualitative Evaluation

We performed qualitative evaluations of our mesoscopic characterization for kinematic structure estimation using the *elephant-gallop* (synthesized public dataset [18]), as well as, the *free*, the *lock*, the *crane* and *head* sequences (real-world public datasets [17], [23]).

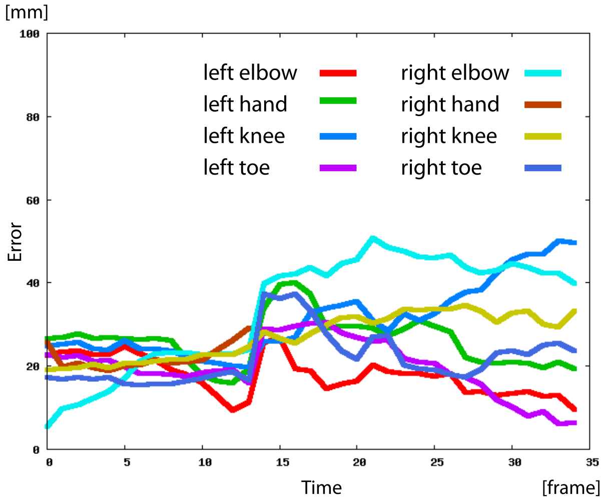

Fig. 16 Estimation errors on synthesized data compared to ground truth.

Table 2 Averaged estimation errors and their deviations.

	average (mm)	standard deviation (mm)		average (mm)	standard deviation (mm)
🟥 left elbow	17.8	1.4	🟦 right elbow	33.7	1.1
🟩 left hand	26.0	1.1	🟧 right hand	22.1	1.9
🟦 left knee	31.2	3.1	🟨 right knee	27.2	1.0
🟪 left toe	19.9	2.3	🟦 right toe	21.8	0.3

In the *elephant-gallop* sequence containing 49 frames, a synthesized model of an elephant gallops. Every mesh shares a same number of vertices (42,321), and their connectivities. The surfaces are relatively smooth but has folds at non-rigid areas. The *free* and the *lock* sequence contains 500 frames and 250 frames respectively, and consist of a subject wearing loose clothes and performing a break dance. Both sequences are pre-processed by Ref. [5] for purpose of comparison to Ref. [6] which is based on Ref. [5], and have fixed mesh geometries. The *free* sequence consists of 4,284 vertices and the *lock* sequence 5,301 vertices. The *crane* sequence contains 175 frames, and consists of a subject walking with his arms up and down. The data was a reconstruction result of Ref. [23] and has a fixed mesh geometry with 10,002 vertices. The *head* sequence contains 30 frames, and consists of a subject performing a break dance. The mesh geometry is not temporally coherent and has a varing number of vertices around 68,500 vertices. Surfaces of *free*, *lock*, *crane* and *head* are also very smooth but has noises caused by both the surface reconstruction step and the tracking by deformation method.

Figures 17, **18** shows that we can obtain anatomically consistent results for the *free*, *lock*, *crane*, *head* and the *elephant-gallop* sequences. In the results for *elephant-gallop*, the hones are recognized as erroneous cones based on the assumption we showed in

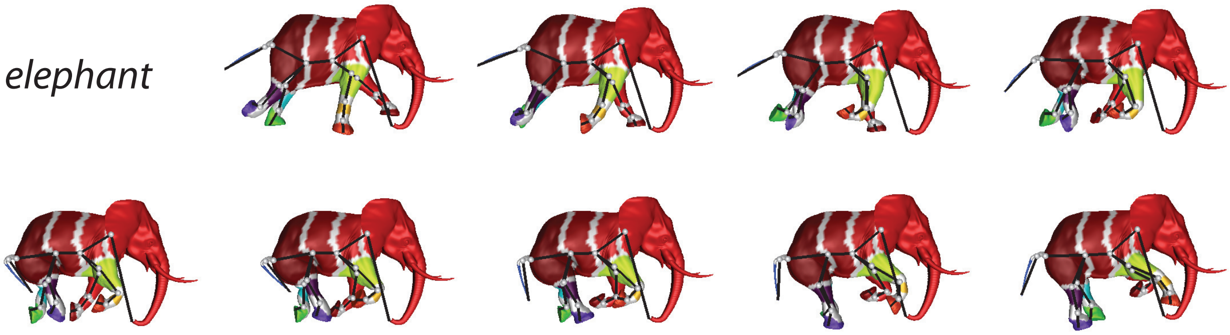

Fig. 17 Temporally coherent rigid/non-rigid surface segmentation and estimated kinematic structure of a synthesized public dataset.

Section 6.1.1.3, and merged to the nose and the head as they are sharing same rigid motion. The results for *free*, *lock*, *crane head* sequences show that our approach is robust to noisy real world data, The other one for *elephant-gallop* sequence shows that proposed method can be applied not only to human figure but also to non-human objects that can be approximated by tree-shaped articulated rigid bodies.

Compared with the state-of-the-art [6] (**Fig. 19**), our method robustly estimate the kinematic structure embedded inside the mesh while the result by Ref. [6] includes artifacts due to occasional mis-fitting of rigid sub-surfaces on the leg as described in Ref. [6] (See top-rihgt in Fig. 19). Moreover, our method explicitly estimates non-rigid areas which allow us to reconstruct the kinematic structure as shown in Fig. 18 while Ref. [6] remains at the level of surface segmentation.

We also tested the proposed method with a challenging data, i.e., the *head* sequence, in which global toporogy vary (**Fig. 20**). The result proves that our method can estimate a consistent body-parts segmentation throughout frames in comparison with another state-of-the-art method [3].

7.3 Discussions

Our algorithm is based on the distribution of μ which is the sum of geodesic distance on the surface. Due to this integral operation over the surface, it is robust to random deformations of the surface due to noise [22]. **Figure 21** shows results using the ground truth sequence (left) and a synthesized sequence by introducing Gaussian noise of $\sigma = 1$ and zero mean into the vertices of the ground truth (right). While the surface on the right is highly noisy, the estimated body-parts are fairly equivalent to the left. Hence we can conclude that our method is robust against noise on the 3D surfaces.

However, the distribution of μ can be largely affected once a hole on the surface which changes the topological structure is introduced during the 3D shape reconstruction process, because it changes the geodesic distance on the surface drastically. Since such holes can typically introduced by pixel-level misdetections of the multi-view silhouettes for the shape-from-silhouette process, we can eliminate such failure cases by applying hole-fillings to the silhouettes in practice. This point is justified by the fact that our method could estimate reasonable kinematic structures for real datasets (Fig. 18).

In some of the real world data such as *free* and *lock* sequence, as the result of the subject wearing loose cloth, there exist the extra bone (Fig. 18, the left leg of *free* and *lock*). This is because the cloth is sliding along the long axis direction of a bone inside, and such freely-drifting portion appeared as an extra apparent rigid part, since it could not be described by the original leg motion. In the result of *free* sequence, we can see extra bones, one around the right knee and another near by the left hip joint. From this observation, it can be said that our method prefer the surface with less sliding along the long axis direction of embed bones as input mesh even though the algorithm is robust to non-rigid motion and noise. Otherwise, if the 3D shape capture is accurate enough and the wrinkles and drifts of clothing are well modeled in the mesh structure, we can expect the geodesic distance is well preserved among different frames, and hence our method will work.

In the annulus clustering step, we assumed that the closer $\overline{e^{merge}}(A_n^k)$ to 1, the more the motion of A_n^k and its adjacent sub-surface is similar. In fact having comparable self and cast errors is a necessary but not a sufficient condition to conclude that A_n^k and S follow a similar rigid motion, since completely different rigid motions can return a similar error in some degenerated cases. While merging A_n^k to S in such cases results in making the joint between them to disappear, we here simply assume such degenerated cases are less likely to happen, and can be ignored. Our evaluations demonstrate this assumption is valid in bibpractice. Also, we used an empirically-determined threshold θ_r. Learning or estimating an optimal θ_r automatically is a part of our future work.

Although we have introduced a lost cone recovering mechanism in Section 6.1.1.4, we still have localization ambiguity in the kinematic structure when a non-rigid area or a tip is invisible in the reconstructed 3D mesh. In **Fig. 22**, a leaf corresponding to the right hand is found in interval I_i, but has no corresponding vertex on surface mesh in interval I_{i+1}. In this case, we localize every part of the kinematic structure in interval I_i, but cannot fix the posture of right hand in interval I_{i+1}.

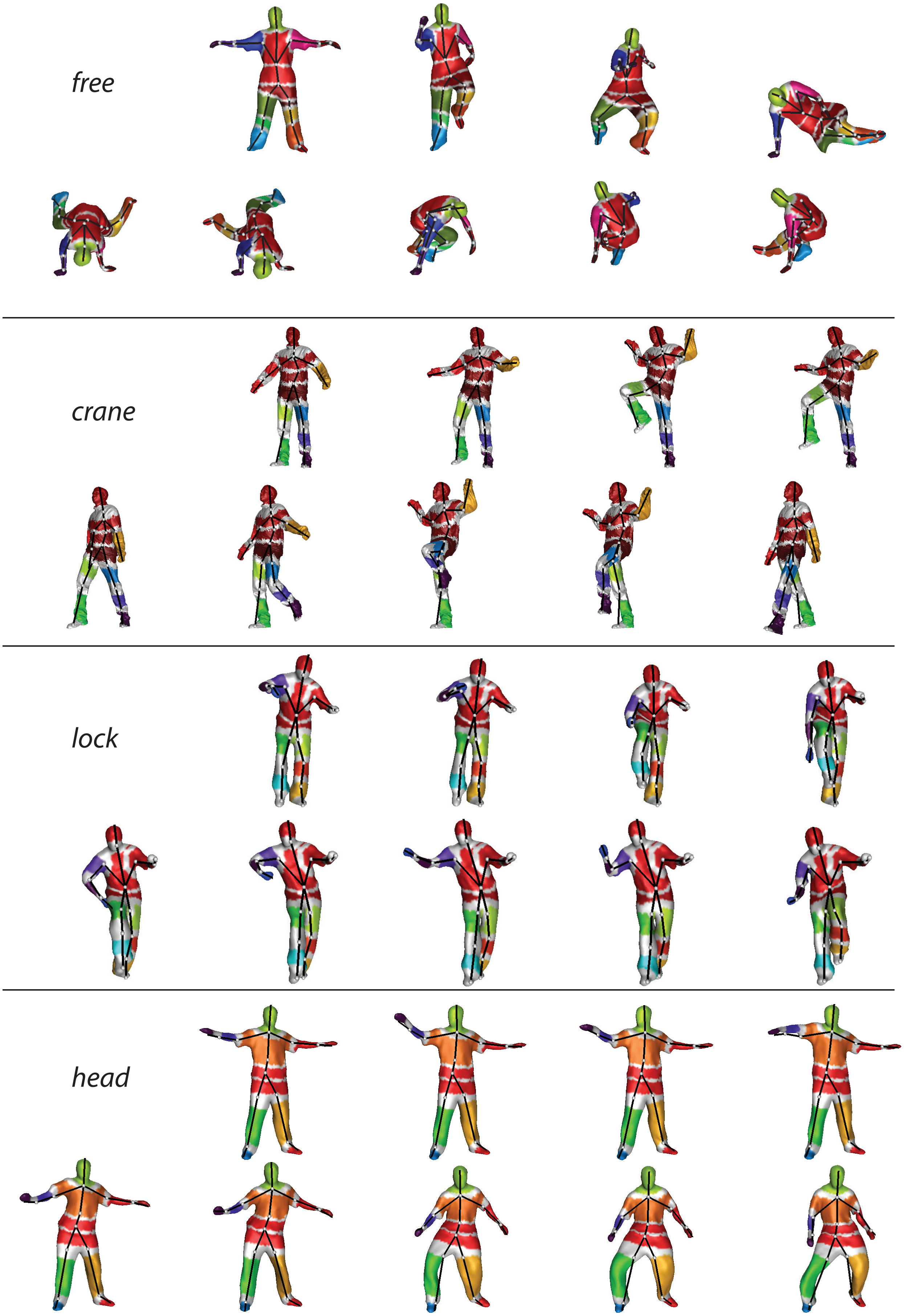

Fig. 18 Temporally coherent rigid/non-rigid surface segmentation and estimated kinematic structure of real-world public datasets.

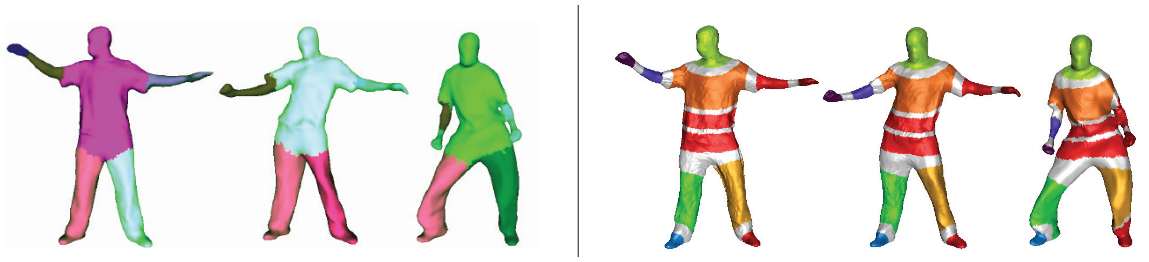

Fig. 20 Segmentation results for *head* sequence by an existing method (left, from Arcila et al. [3], ©2013 Elsevier B.V.) and the proposed method (right). Notice that the estimated kinematic structure is not shown in our results, in order to make the images are comparable with the ones in Ref. [3]. Please refer to Fig. 18 for our results with the kinematic structure.

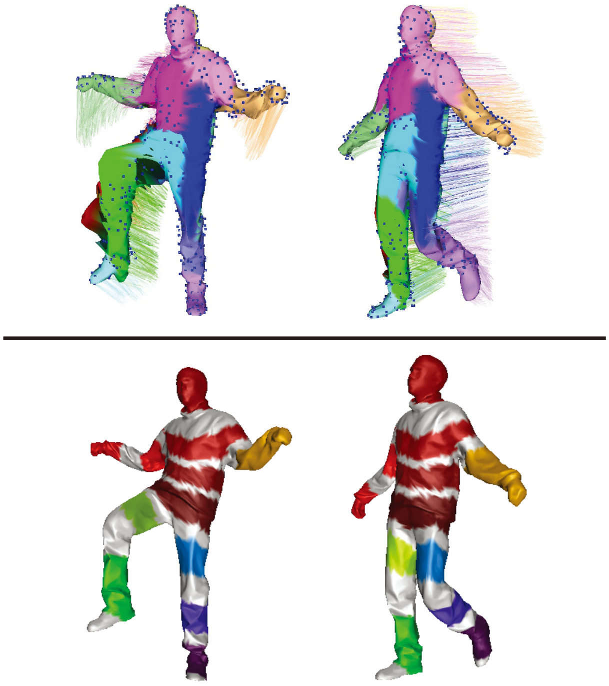

Fig. 19 Segmentation results for *crane* sequence by an existing method (top, from Franco and Boyer [6], ©2011 IEEE) and the proposed method (bottom). Notice that the estimated kinematic structure is not shown in our results, in order to make the images are comparable with the ones in Ref. [6]. Please refer to Fig. 18 for our results with the kinematic structure.

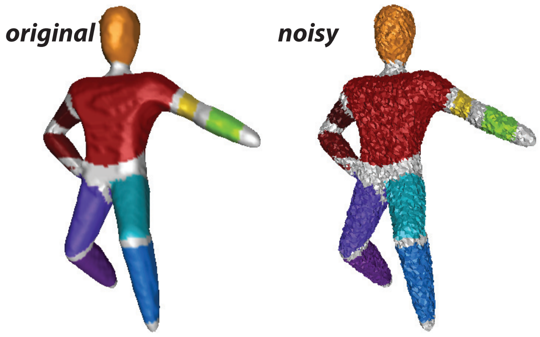

Fig. 21 Segmentation results for original data and one with artificial noise.

8. Conclusion

In this paper, we proposed a new method for acquiring kinematic structure of 3D dynamic surface using tree-structured mesoscopic surface characterization based on cones and annuli modeling. We proved that our method can estimate the kinematic structure even for real human data with loose cloth deforming non-rigidity.

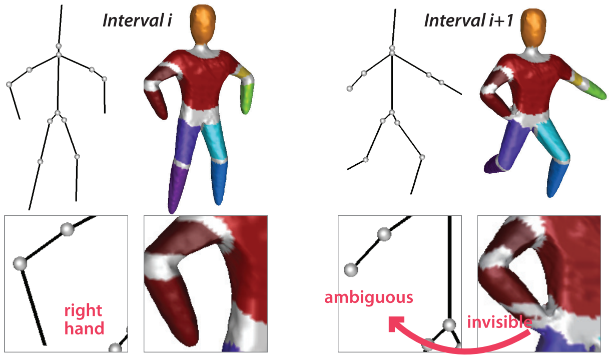

Fig. 22 Localization ambiguity in the kinematic structure in some intervals.

As we discussed in the last section, the obtained kinematic structure can have localization ambiguities in some intervals. To eliminate the ambiguities, we can apply existing tracking by deformation methods [15], [23] using the obtained kinematic structure as the constraint in future work.

Acknowledgments This research is partially supported by JSPS Bilateral Joint Research Project "Visual Gesture Perception".

References

[1] Aguiar, E., Theobalt, C. and Seidel, H.P.: Automatic Learning of Articulated Skeletons from 3D Marker Trajectories, *Proc. ISCV 2006*, pp.485–494 (2006).

[2] Aguiar, E., Theobalt, C., Thrun, S. and Seidel, H.P.: Automatic Conversion of Mesh Animations into Skeleton-based Animations, *Proc. Eurographics* Vol.27, No.2, pp.389–397 (2008).

[3] Arcila, R., Cagniart, C., Hetroy, F., Boyer, E. and Dupont, F.: Segmentaion of temporal mesh sequences into rigidly movin components, *Graphical Models*, Vol.75, No.1, pp.10–22 (2012).

[4] Bronstein, A.M., Bronstein, M.M. and Kimmel, R.: Calculus of non-rigid surfaces for geometry and texture manipulation, *IEEE Trans. Visualization and Computer Graphics*, Vol.13, No.5, pp.902–913 (2007).

[5] Cagniart, C., Boyer, E. and Ilic, S.: Probabilistic Deformable Surface Tracking From Multiple Videos, *Proc. ECCV, Part IV*, Lecture Notes in Computer Science, Vol.6314, pp.326–339 (2010).

[6] Franco, J. and Boyer, E.: Learning Temporally Consistent Rigidities, *Proc. CVPR*, pp.1241–1248 (2011).

[7] Hilaga, M., Shinagawa, Y., Kohmura, T. and Kunii, T.L.: Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes, *Proc. SIGGRAPH*, pp.203–212 (2001).

[8] Iiyama, M., Kameda, Y. and Minoh, M.: Estimation of the Location of Joint Points of Human Body from Successive Volume Data, *Proc. ICPR 2000*, Vol.3, pp.699–702 (2002).

[9] Kanade, T., Rander, P. and Narayanan, P.J.: Virtualized Reality: Con-

structing VirtualWorlds from Real Scenes, *IEEE Multimedia, Immersive Telepresence*, Vol.4, No.1, pp.34–47 (1997).

[10] Letouzey, A. and Boyer, E.: Progressive Shape Models, *Proc. CVPR*, pp.190–197 (2012).

[11] Marr, D.: Vision: *A Computational Investigation into the Human Representation and Processing of Visual Information*, Freeman (1982).

[12] Matsuyama, T., Nobuhara, S., Takai, T. and Tung, T.: *3D Video and Its Applications*, Springer (2012).

[13] Moezzi, S., Tai, L. and Gerard, P.: Virtual View Generation for 3D Digital Video, *IEEE Multimedia*, Vol.4, No.1, pp.18–26 (1997).

[14] Morse, M.: The calculus of variations in the large, *American mathematical Society, Colloquium Publication*, Vol.18 (1934).

[15] Mukasa, T., Miyamoto, A., Nobuhara, S., Maki, A. and Matsuyama, T.: Complex Human Motion Estimation Using Visibility, *Proc. FG*, pp.1–6 (2008).

[16] Sorkine, O. and Alexa, M.: As-Rigid-As-Possible Surface Modeling, *Proc. Eurographics Symposium on Geometry Processing*, pp.109–116 (2007).

[17] Starck, J., Maki, A., Nobuhara, S., Hilton, A. and Matsuyama, T.: The Multiple-Camera 3-D Production Studio, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.19, No.6, pp.856–869 (2009).

[18] Sumner, R.W. and Popovic, J.: Deformation Transfer for Triangle Meshes, *ACM Trans. Gr.*, Vol.23, No.3, pp.399–405 (2004).

[19] Theobalt, C., de Aguiar, E., Magnor, M.A., Theisel, H. and Seidel, H.P.: Marker-free Kinematic Skeleton Estimation from Sequence of Volume Data, *Proc. VRST*, pp.57–64 (2004).

[20] Tung, T. and Matsuyama, T.: Dynamic Surface Matching by Geodesic Mapping for 3D Animation Transfer, *Proc. CVPR* pp.1402–1409 (2010).

[21] Tung, T. and Matsuyama, T.: Topology Dictionary for 3D Video Understanding, *IEEE Trans. PAMI*, Vol.34, No.8, pp.1645–1657 (2012).

[22] Tung, T. and Schmitt, F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes, *International Journal of Shape Modeling*, Vol.11, No.1, pp.91–120 (2005).

[23] Vlasic, D., Baran, I., Matusiky, W. and Popovic, J.: Articulated Mesh Animation from Multi-view Silhouettes, *ACM Trans. Gr.*, Vol.28, Issue.3, No.97 (2008).

Tomoyuki Mukasa received his B.Sc. in Engineering and M.Sc. in Informatics from Kyoto University in 2004 and 2006. Since 2012, he has been a Ph.D. candidate at Kyoto University. His research interests include computer vision, 3D video and their application in augmented reality, e.g., pre-visualization in film making.

Shohei Nobuhara received his B.Sc. in Engineering, M.Sc. and Ph.D. in Informatics from Kyoto University, Japan, in 2000, 2002, and 2005 respectively. He is currently a senior lecturer in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. His research interests include computer vision and 3D video. He is a member of IPSJ, IEICE, and IEEE.

Tony Tung received his M.Sc. degree in Physics and Computer Science from the Ecole Nationale Superieure de Physique, France, with a double degree in Photonics and Image Processing in 2000, and Ph.D. degree in Signal and Image processing from the Ecole Nationale Supérieure des Telecommunications de Paris in 2005. Currently, he is an assistant professor at Kyoto University, working jointly at the Department of Intelligence Science and Technology, Graduate School of Informatics, and at the Academic Center for Computing and Media Studies. His research interests include computer vision, pattern recognition, shape modeling, and multimodal interaction. He was awarded Fellowships from the Japan Society for the Promotion of Science in 2005 and 2008, and Grant-in-Aid for Young Scientists in 2011.

Takashi Matsuyama received his B.Eng., M.Eng., and D.Eng. degrees in Electrical Engineering from Kyoto University, Japan, in 1974, 1976, and 1980, respectively. He is currently a professor in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. His research interests include knowledge-based image understanding, computer vision, 3D Video, and human-computer interaction. He wrote about 100 papers and books including two research monographs, A Structural Analysis of Complex Aerial Photographs, PLENUM, 1980 and SIGMA: A Knowledge-Based Aerial Image Understanding System, PLENUM, 1990. He won nine best paper awards from Japanese and international academic societies including the Marr Prize at ICCV95. He is on the editorial board of the Pattern Recognition Journal. He was awarded Fellowships from the International Association for Pattern Recognition, IPSJ and IEICE.

(Communicated by *Tien-Tsin Wong*)