EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Space-Time Co-Segmentation of Articulated Point Cloud Sequences

Qing Yuan[1] Guiqing Li[1] Kai Xu[3,4] Xudong Chen[1] Hui Huang[2,3†]

[1]South China University of Technology [2]Shenzhen University [3]Shenzhen VisuCA Key Lab / SIAT [4]National University of Defense Technology

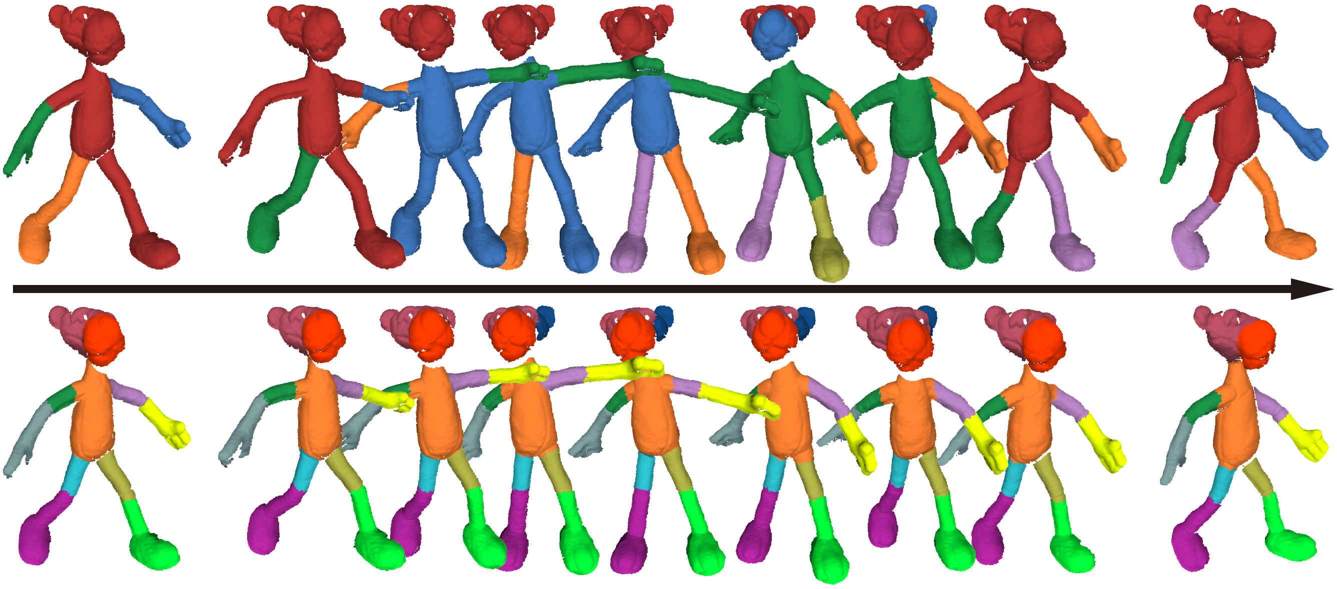


Figure 1: *Space-time co-segmentation for the Pink Panther dataset: motion-based segmentation of individual frames is shown in the top line, and the co-segmentation result is depicted at the bottom line.*

Abstract

Consistent segmentation is to the center of many applications based on dynamic geometric data. Directly segmenting a raw 3D point cloud sequence is a challenging task due to the low data quality and large inter-frame variation across the whole sequence. We propose a local-to-global *approach to* co-segment *point cloud sequences of articulated objects into near-rigid moving parts. Our method starts from a per-frame point clustering, derived from a robust voting-based trajectory analysis. The local segments are then progressively propagated to the neighboring frames with a cut propagation operation, and further merged through all frames using a novel* space-time segment grouping *technqiue, leading to a globally consistent and compact segmentation of the entire articulated point cloud sequence. Such progressive propagating and merging, in both space and time dimensions, makes our co-segmentation algorithm especially robust in handling noise, occlusions and pose/view variations that are usually associated with raw scan data.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Recent advances in the technique of real-time 3D acquisition, such as commodity RGBD cameras, have made dynamic geometric da-

† Corresponding authors: Guiqing Li and Hui Huang (ligq@scut.edu.cn, hhzhiyan@gmail.com)

ta more accessible than ever. Huge amount of dynamic geometry has been captured, and the availability of such data has stimulated many interesting research and applications, e.g., dynamic geometry reconstruction and processing [CLM*12], human pose recognition for somatosensory interaction [SFC*11], and 3D scene understanding by human activity [SCH*14, FSL*15].

Shape analysis and understanding over the input dynamic geometry is to the center of the applications mentioned above. Especially, meaningful part segmentation of articulated objects is perhaps the most fundamental and useful analysis task involved, e.g., assisting the reconstruction of dynamic range scans [CZ11] and human pose recognition [SFC*11]. For dynamic geometry, it is natural to consider a space-time analysis to achieve a globally consistent part segmentation across the whole sequence of animated geometry. Such consistent segmentation also implies a dense correspondence across all frames of geometry, which is essential for tasks such as robust tracking [HRF13] and reconstruction [NFS15].

Co-segmentation has received much attention lately in the fields of computer vision and graphics. Through co-analyzing a pair or a set of images/shapes, it is possible to enhance the segmentation through exploiting the mutually complementary information between different instances, and further to identify meaningful parts based on their correspondence [BP10,CGF09,XLZ*10,SvKK*11]. Consistent segmentation has been applied to videos through incorporating spatial and temporal constraints [DFB13]. Similarly, space-time co-segmentation has also been used to segment mesh-based animation sequences into rigidly moving components [ACH*13]. For articulated point cloud sequences, in particularly the noisy and highly incomplete ones captured by low resolution RGBD cameras, consensus skeleton extraction, frame-to-frame registration, and surface reconstruction are all challenging tasks on their own. It is therefore desirable to achieve space-time co-segmentation directly over the raw input points.

Space-time co-segmentation of point cloud sequences poses two major challenges. Spatially, the point cloud of a single frame could be rather low-quality such that local analysis based on the geometry can hardly be robust. Temporally, the point clouds of different frames can vary significantly over time, due to pose and/or view changes. Therefore, it is extremely difficult to find a coherent correspondence across the whole sequence. To address these challenges, we propose a *local-to-global* approach to *progressively* obtain a globally consistent space-time segmentation (Figure 1), where the spatial and temporal consistency benefit each other in a coupled solution. We firstly perform a local segmentation for each frame through point clustering based on local trajectory analysis. The local segmentation is then mutually propagated between every two neighboring frames to attain a consistent over-segmentation, resulting in a number of sub-sequences formed by the segments that move near-rigidly. Finally a novel space-time grouping technique, which aims to group the small segment sequences belonging to the same rigid part of the moving articulated object, is applied to achieve a consistent and compact space-time segmentation across all frames of the entire point cloud sequence.

2. Related work

We focus our review of previous works on closely related topics on co-segmentation and motion analysis.

Co-segmentation of statics meshes. Most of co-segmentation approaches consist of three steps: over-segmenting the meshes individually to produce a large set of patches, computing multiple features for each patch, and clustering the patches in feature spaces. Golovinskiy and Funkhouser [GF09] pioneer the co-segmentation of sets of 3D shapes, where the problem is formulated as graph-cut and solved with the normalized cut algorithm. Xu et al. [XLZ*10] perform co-segmentation for a set of shapes belonging to the same family via style-content separation, where the style is defined as anisotropic part proportion. Sidis et al. [SvKK*11] obtain a compatible segmentation of a set of meshes in three stages: per-object segmentation, descriptor-space spectral clustering, and refined co-segmentation. This is followed by several other approaches with various schemes of feature selection and fusion [MXLH13, HFL12, WWS*13]. Common to these co-segmentation works is that the goal is to identify functional part shared among shapes of the same family. For the animation sequence of a specific shape, it is natural to perform motion-based analysis to extract motion-wise independent parts.

Co-segmentation of animated meshes. Given an animated sequence of meshes sharing the same connectivity, Lee et al. [LWC06] conduct co-segmentation by analyzing triangular face trajectories and performing dual graph segmentation. By measuring the local deformation degree with the change of the dihedral angle between adjacent faces, Wuhrer and Brunton [WB10] cast the co-segmentation problem as the d-partition of a dual graph of mesh faces with the edges weighted by the maximal change of dihedral angles. Group-valued regularization [RBB*13,RBBK12,RBB*12] formulates motion-based surface segmentation as a piecewise-smooth regularization problem for transformations between poses. This approach attempts to find a rigid transformation at each surface point such that the overall transformation is described by a relatively sparse set of such transformations, each corresponding to a rigid part of the object.

Ghosh et al. [GSLB12] segment a sequence of meshes of an articulated object in various poses into rigid parts. They introduce a modified distance dependent Chinese restaurant process to allow nonparametric segmentation. Assuming vertices ongoing a similar rigid/scaling transformation during motion have similar local geometric attributes, Liao et al. [LXL*12] achieve segmentation by clustering all vertices in an attribute space. This method can deal with animated meshes with different connectivity. Vasilakis and Fudos [VF14] develop a method for varying level-of-detail segmentation of arbitrary animated objects. This is achieved by first performing a partitioning-aware over-segmentation over the animated meshes and then conducting a simplification over the segments based on rigidity-preserving criteria.

There is a significant amount of work for co-segmenting animated 3D shapes. However, most of them deal with only clean meshes. For dynamic range scans, Chang et al. [CZ11] perform motion-based segmentation and global registration simultaneously

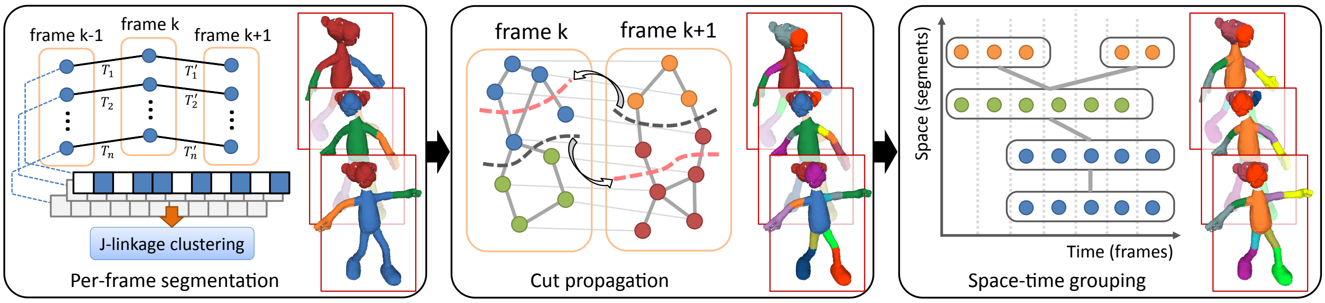


Figure 2: *Pipeline of our space-time co-segmentation: given an input sequence of point clouds, our method starts by performing a local per-frame segmentation through motion-based point clustering (left). The local segmentation are then mutually propagated between every two adjacent frames, obtaining a number of sub-sequences formed by the segments with near-rigid motion (middle). Finally, we conduct a space-time grouping over the sub-sequences to form a globally consistent and compact segmentation over the whole 4D sequence (right).*

for space-time surface reconstruction. Zheng et al. [ZST*10] introduce the notion of consensus skeletons for non-rigid space-time registration of a deforming point-based shape. However, the extension from the consensus skeletons to the motion consistent co-segmentation is not trivial, especially when dealing with the joint areas. Our method is devised to deal with low-quality raw point cloud sequences without explicit skeleton extraction.

Trajectory analysis for motion structures. For a video sequence, the feature points of an object can be traced to form a set of motion trajectories. It has been a long-standing problem in computer vision to reconstruct motion structures of the objects in the video out of these trajectories [TK92]. Akhter et al. [ASKK11] propose a dual approach to describe the evolving 3D structure in a trajectory space with a linear combination of basis trajectories independent of objects in motion. Zaheer et al. [ZAB*11] present a factorization approach for 3D reconstruction from multiple static cameras based on the compact trajectory subspace representation. Fradet et al. [FRP09] formulate a labeling problem to solve the motion pattern recognition among a set of trajectories with variety of life-spans. It randomly generates a number of motion models and groups the trajectories to vote the best models using an agglomerative clustering strategy. We extend this method to deal with more challenging input of point cloud sequences, through progressively propagating the local trajectory analysis to form a globally consistent motion-guided segmentation.

3. Algorithm Overview

Problem description. Given a point cloud sequence $S = \{C^1, C^2, \cdots, C^F\}$, capturing F frames of a performance, our goal is to consistently segment all frames in the sequence into parts with distinct rigid motions. The implication of such space-time co-segmentation are two-fold. Firstly, the point cloud of each frame is segmented into a set of disjoint parts, each of which admits an independent rigid motion, denoted by a set of segments $C^f = \cup_{k=1}^{n_f} C_k^f$. Secondly, we establish correspondence for the segments of adjacent frames based on their rigid motions, such that two segments in correspondence, e.g., C_k^f and $C_{\pi_f(k)}^{f+1}$, represent a consecutive rigid motion, where π_f denotes the correspondence map. We seek the optimal co-segmentation, which can explain the input sequence with the minimal number of rigidly moving segments. To make the problem tractable, we assume that two adjacent frames C^i and C^{i+1} have moderate time interval so that the difference between the two point clouds can be discriminated but still support a valid correspondence (e.g., $>$ 5 frames per second for general human motions).

We propose a local-to-global approach to progressively achieve the space-time co-segmentation. Our method consists of three major phases, as shown in Figure 2, including motion-based clustering for per-frame local segmentation, cut propagation for consistent over-segmentation, and space-time grouping of the fine segments for the final co-segmentation.

Per-frame segmentation. While the point clouds vary moderately between adjacent frames, they may change significantly across the whole sequence due to pose/view change and occlusion. Therefore, we start our space-time segmentation from each single frame through analyzing trajectories of all its points with the help of the neighboring frames. Given a point at the current frame, we build its local temporal trajectory through finding its corresponding points in the neighboring frames, using a simple deformable 3D shape registration method [PB11]. We randomly choose a number of trajectory triplets, each of which determines a rigid motion model. These models are then used to perform an agglomerative hierarchical clustering for all point trajectories, leading to a rigid motion based local segmentation of the current frame (Figure 2(left)).

Cut propagation. The per-frame segmentation, based on local trajectory analysis, may be inconsistent even for neighboring frames. In order to achieve consistent segmentation across as-long-as-possible sub-sequences, our next step is to mutually propagate the cutting seams between every two neighboring frames, based on the point-to-point correspondence between them; see Figure 2(middle). Although this may lead to an over-segmentation for most of the frames, it guarantees that neighboring frames are consistently segmented into rigid parts.

Space-time grouping. To compute a compact segmentation for all frames, we perform a space-time grouping step to merge the segments resulted in the over-segmentation of the previous step;

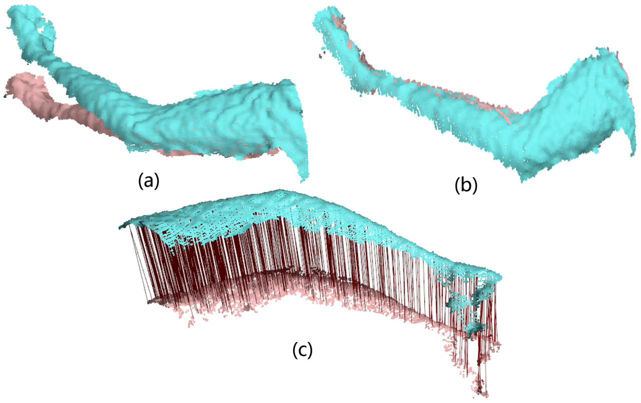


Figure 3: *Illustration of local similarity registration: (a) two original frames; (b) the blue frame (11581 points) is iteratively deformed to register with the red frame (12648 points); (c) correspondence illustration of partial points using red lines.*

see Figure 2(right). This is achieved by formulating a space-time graph-cut to guarantee spatial and temporal compactness of the final segments. Spatially, we merge the segments of each frame indicating the same rigidly moving part. Temporally, we group the segment trajectories, which represent a consecutive rigid motion.

4. Motion-based per-frame segmentation

For the sake of efficiency, we down-sample the input point cloud of each frame using the simple octree-based method in [RL01] with a sampling rate of 10%. All operations in this section are performed on the simplified data.

4.1. Registration of adjacent frames

3D shape correspondence is a fundamental problem in geometry processing and many approaches have been proposed for 3D point clouds [vKZHCO11]. We adapt the method in [PB11], a variant of ICP incorporating local similarity transformation, into our problem setting. Specifically, to handle relatively large inter-frame point cloud variation, we improve the method by introducing a correspondence optimization procedure based on local smoothness.

Given the point clouds of two frames C^s and C^t, the method interleaves between correspondence and deformation to iteratively improve the registration. In the correspondence stage, we first match each point $p \in C^s$ to its closest point in $q \in C^t$ as in [PB11]. We then optimize p's matching point by searching over q's k-nearest neighbors such that the matching error around the local neighborhood is as smooth as possible. The rationale of doing this is, assuming the underlying surfaces of the point clouds are smooth, the matching error should vary smoothly across the surface for a good correspondence. Formally, the corresponding phase seeks a mapping $\tau_{s\to t} : C^s \to C^t$ by minimizing a Laplacian smoothness energy of the residual field $\{\tau_{s\to t}(p) - p : p \in C^s\}$:

$$\mathcal{E}_{\text{smooth}}(\tau_{s\to t}) = \sum_{p\in C^s} \|\Delta(p)\|^2, \qquad (1)$$

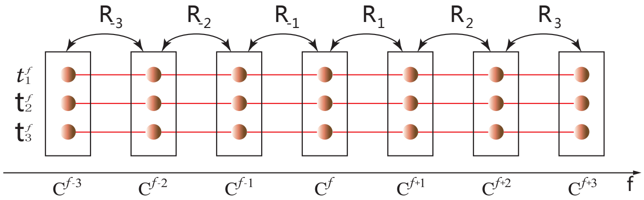


Figure 4: *Three trajectories t_1,t_2,t_3 determine a motion model $M = (\mathbf{R}_{-r},\cdots,\mathbf{R}_{-1},\mathbf{R}_1,\cdots,\mathbf{R}_r)$. Here the trajectories are of length 7, namely, $r = 3$.*

where the residual Laplacian $\Delta(p)$ is defined as:

$$\Delta(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{q\in\mathcal{N}(p)} [(\tau_{s\to t}(p) - p) - (\tau_{s\to t}(q) - q)].$$

Here $\mathcal{N}(p)$ is the set of the k-nearest neighbors of point p. In the deformation phase, we estimate a similarity transformation, denoted by $(s,\mathbf{R},\mathbf{t})$, a triplet of scaling factor, rotation matrix and translation vector, to minimize a fitting error between the local neighborhood of $p \in C^s$ and its matched counterpart:

$$\mathcal{E}_{\text{fitting}}(s,\mathbf{R},\mathbf{t}) = \sum_{q\in\mathcal{N}(p)} \|\tau_{s\to t}(q) - (s\mathbf{R}q + \mathbf{t})\|^2. \qquad (2)$$

This minimization has a closed form solution which can be computed with SVD decomposition [SMW06, YPG01]. Figure 3 demonstrates the two phases in aligning two poses of an arm (a). The blue pose is iteratively deformed to align with the red one (b) while a dense correspondence is established by the improved method (c).

4.2. Trajectory-based clustering

Toldo and Fusiello propose J-linkage clustering. It utilizes agglomerative clustering to extract multiple models in RANSAC where the affinity of classes is measured by the Jaccard distance [TF08]. This method has been widely adopted to detect various kinds of models in computer vision [FRP09, FSB10]. Different from the existing applications, we apply this method to motion-based segmentation of 3D point cloud sequences via analyzing temporal trajectories.

Local temporal trajectories. When dealing with raw scan data, it is often impossible to track a full temporal trajectory across the whole sequence since some areas of the object seen before may disappear in some later frames due to pose/view occlusion. Even if there exist full trajectories for some points, the correspondence (tracking) error along them may accumulate severely, making the analysis based on them rather unreliable. Therefore, we instead rely on short-range, local trajectories to obtain motion-based point clustering. Specifically, we construct the local trajectory for a given point $p \in C^f$ for a window of $2r+1$ frames, covering frames from C^{f-r} to C^{f+r}. The trajectory is created with the help of the correspondence computed in the previous step. r is a user-specified parameter and is set to 2 for all experiments described in Section 6.

Trajectory analysis. Given a frame of point cloud C^f, we perform clustering over the points based on their local trajectories. Following the work of [FRP09], we start by randomly selecting a number of triplets of local trajectories to generate a set of initial rigid motion models, denoted by $M^f = \{M_m\}_{m=1,\cdots,|M^f|}$, where the model count is set to $|M^f| = |C^f|$ in all our experiments. The random selection of the triplets does not rely on any prior, making the method simple and practically robust. Each motion model M_m is represented as a sequence of $2r$ rigid transformations corresponding to a triplet of local trajectories, as illustrated in Figure 4.

Each transformation maps the three points of a frame to their corresponding points of the adjacent frame.

Next, we evaluate how well a given motion model fits a known trajectory. Specifically, for a given trajectory $t = (p^{f-r}, \cdots, p^f, \cdots, p^{f+r})$ and its corresponding rigid motion model $M = (\mathbf{R}_{-r}, \cdots, \mathbf{R}_{-1}, \mathbf{R}_1, \cdots, \mathbf{R}_r)$ (See Figure 4), we compute 3D *position* residual between a trajectory t and a motion model M:

$$\mathcal{R}_{\text{pos}}(t,M) = \sum_{-r\leq i\leq r, i\neq 0} \|\mathbf{R}_i p^{f+i-sign(i)} - p^{f+i}\|^2. \qquad (3)$$

Following that, we create a binary matrix $V \in \mathbf{R}^{|C^f|\times|M^f|}$ to use the trajectories to vote the models. The entries of V are defined as:

$$V(k,l) = \begin{cases} 1 & \mathcal{R}_{pos}(t_k, M_l) < \varepsilon, \\ 0 & \mathcal{R}_{pos}(t_k, M_l) \geq \varepsilon, \end{cases}$$

where $t_k \in T^f$, $M_l \in M^f$, and the threshold ε is assigned with the $\rho|C^f||M^f|$-th smallest one among all residuals $\mathcal{R}_{pos}(t_k, M_l)$. The percentage parameter ρ is set to 55% by default in our experiments if not specified. We define the *preference set* of a trajectory t_k, denoted by $P(t_k) = \{j : V(k,j) = 1\}$, as the intersection of the preference sets of its all elements.

J-linkage clustering starts with each single trajectory of T^f as a class and iteratively merges two nearest classes until the convergence conditions are satisfied. Therefore, we need to measure the similarity between two trajectory sets. Let T_a and T_b be two clusters of trajectories with their preference sets denoted by $P(T_a)$ and $P(T_b)$ respectively. The preference set of a trajectory set is simply the union of that of its members. We measure the dissimilarity between trajectory sets using the Jaccard distance following [FRP09]:

$$d_T(T_a, T_b) = 1 - \frac{|T_a \cap T_b|}{|T_a \cup T_b|}. \qquad (4)$$

It should be noted that it does not work to perform traditional clustering methods, such as k-means and spectral clustering, directly in the space of trajectories, which necessitates our motion model based clustering. Figure 5(a) shows a girl raising a hand while lifting a leg. The segmentation results demonstrate that our method (d) can correctly extract the motion components while the other two fail as shown in (b) and (c).

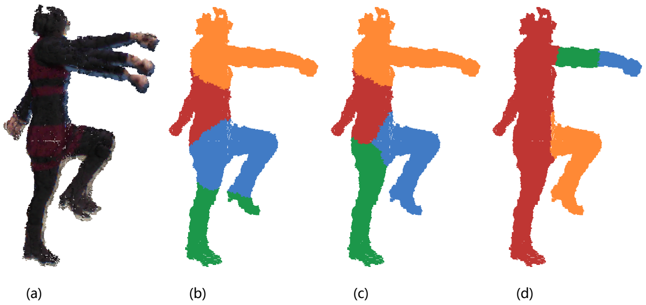


Figure 5: *Trajectories clustering comparison: (a) three frames of the input sequence; (b-d) segmentation by K-means clustering, spectral clustering and our method, respectively.*

4.3. Smoothing of segment boundaries

The segmentation quality for individual frames is important to the subsequent processing; imperfect segmentation may have adverse effect on inter-frame segment correspondence. We improve the segmentation boundaries through solving a relabeling problem. The goal is to produce a new segmentation, with the same segment count, with smoother boundaries between adjacent segments. Suppose the current segmentation is denoted by $\{C^f_k : k = 1, 2, \cdots, n_f\}$ of C^f. The labeling map is $l : C^f \rightarrow \{1, 2, \cdots, n_f\}$ with label k indicating the class determined by segment C^f_k. We minimize the following objective function:

$$\mathcal{E}^{\text{b}}(l, C^f) = \sum_{p\in C^f} \mathcal{E}^{\text{b}}_{\text{data}}(l, p) + \lambda \sum_{(p_1,p_2)\in G(C^f)} \mathcal{E}^{\text{b}}_{\text{reg}}(l, p_1, p_2), \qquad (5)$$

where $G(C^f)$ is the set of edges of the KNN graph generated for the points of C^f, and $\mathcal{E}^{\text{b}}_{\text{data}}$ and $\mathcal{E}^{\text{b}}_{\text{reg}}$ are data and regularity terms respectively (see below). Without specification, we always set the element number of KNN sets to 20. λ is a blending weight and is fixed to 0.1 for all experiments in Section 6.

We generate a local temporal segment sequence for each C^f_k and then fit a motion model M^f_k (a geometric transformation sequence) using RANSAC which separately transforms C^f_k to the other segments of C^f. For point $p^f \in C^f$, we define its data item as the deviation of its local trajectory $(p^{f-r}, \cdots, p^f, \cdots, p^{f+r})$ from its rigid motion predicted by an arbitrary motion model M^f_k:

$$\mathcal{E}^{\text{b}}_{\text{data}}(l, p^f) = \mathcal{R}_{\text{pos}}(p^f, M^f_k).$$

For simplicity, we define $\mathcal{E}^{\text{b}}_{\text{data}}(l, p^f) = \infty$ if p^f is neither in $C^f_{l(p^f)}$ nor in a segment neighboring to $C^f_{l(p^f)}$.

The regularity term is defined with the similarity of two points $p_1, p_2 \in C^f$, measured using positions and normals. Specifically, assigning two neighboring points with different labels is penalized:

$$\mathcal{E}^{\text{b}}_{\text{reg}}(l, p_1, p_2) = \begin{cases} 0, & l(p_1) = l(p_2), \\ e^{-|p_1-p_2|} e^{\mathbf{n}(p_1)\cdot\mathbf{n}(p_2)}, & l(p_1) \neq l(p_2), \end{cases}$$

where $\mathbf{n}(p)$ is the normal of point p.

Figure 6 demonstrates the segmentation results over an example with and without boundary smoothing.

5. Space-time co-segmentation

The segmentation of individual frames in previous section only relies on local temporal trajectories. In order to achieve a globally

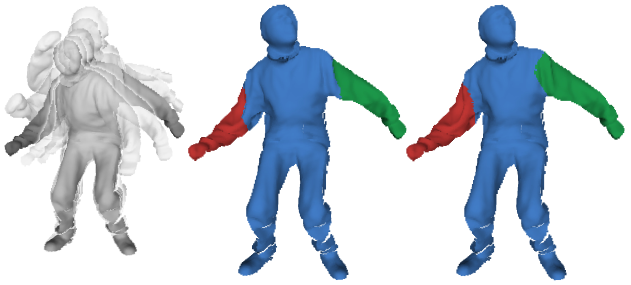


Figure 6: *Smoothing the boundaries between the segments generated by J-linkage clustering (middle) leads to a segmentation with more desirable cutting seams (right).*

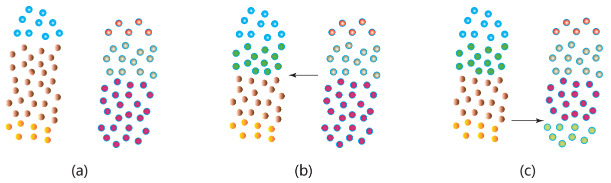


Figure 7: *Propagate the segmentation between two frames (a); (b) cut the middle part of the left frame into two under the guidance of the boundary between middle and bottom parts of the right; (c) transfer the cut in the left frame to the right one.*

consistent segmentation, we associate the per-frame segmentations through a space-time analysis. The problem is challenging due to (1) the large point cloud variation across the sequence and (2) the incompatible segmentation boundaries between adjacent frames. To address these issues, we propose to first propagate the initial segmentation cuts aggressively between every two adjacent frames, to attain a compatible over-segmentation, and then merge the segments based on motion using a space-time graph-cut.

5.1. Cut propagation between adjacent frames

Taking a sequence of segmentation $C^f = \{C^f_k : k = 1, \cdots, n_f\}$ for individual frames as input, we propagate the segmentation cuts across the whole sequence, in an incremental manner. We start from C^1 and C^2 and perform mutual cut propagation. Supposing $C^1, \cdots, C^f$ having been processed, we first propagate the cuts from C^f to C^{f+1} and then propagate those of C^{f+1} backward to all previous frames. To facilitate the propagation, we maintain a segment graph G^f for each frame C^f with its segments as nodes and segment adjacency as edges. During the propagation, the nodes of G^f would be split into multiple ones, if it matches to multiple segments of the adjacent frames.

To propagate cuts of C^f to C^{f+1}, we traverse the edges of G^f and find for each edge the segments in C^{f+1} which should be partitioned. The partition is performed one by one and a segment is split into two each time, as shown in Figure 7. We update G^{f+1} accordingly after cut propagation.

Without loss of generality, suppose that we are given an edge in G^f connecting neighboring segments C^f_1 and C^f_2. We map the boundary points between the two segments to C^{f+1} and find a segment containing the image points. Let us denote this segment by C^{f+1}_k. We then cast the problem to a labeling one with two labels (corresponding to two split segments) over the KNN graph $G(C^{f+1}_k)$ of the points of C^{f+1}_k. Let l be the labeling map $l : C^f_k \to \{1,2\}$. We define the following segment splitting energy:

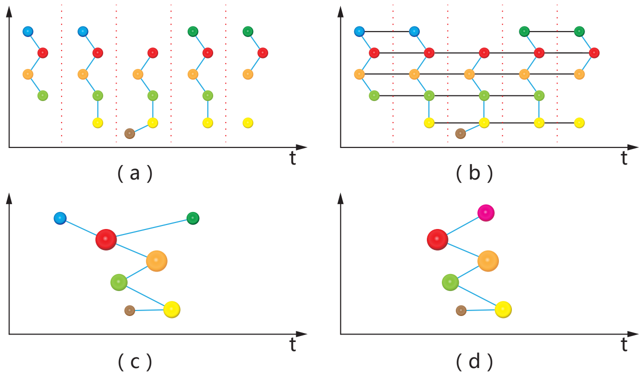


Figure 8: *Generation of graph G for grouping segment sequences: (a) graphs for individual frames, where the same color is assigned to nodes in different graph to indicate segments belonging to the same motion segment sequence; (b) connecting the same color nodes in different graph to form a global graph; (c) collapsing nodes into one; (d) merging nodes which share at least one node and associate with segment sequences with overlapping in the time axis. Note that here the size of nodes indicates the length of its associated segment sequence.*

$$\mathcal{E}^{\mathrm{s}} = \sum_{p \in C^{f+1}_k} \mathcal{E}^{\mathrm{s}}_{\mathrm{data}}(p,l) + \mu \sum_{(p_1,p_2) \in G(C^{f+1}_k)} \mathcal{E}^{\mathrm{s}}_{\mathrm{reg}}(p_1,p_2,l), \quad (6)$$

where μ is a weight used to tune the importance of the two terms.

To define the data term, a natural idea is to utilize the correspondence information. It encourages that the labeling is compatible with the correspondence. So for a point $p \in C^{f+1}_k$, we define:

$$\mathcal{E}^{\mathrm{s}}_{\mathrm{data}}(p,l) = e^{-\frac{x_{l(p)}}{x_1+x_2}} e^{d(p,\tau_{f\to f+1}(\tau_{f+1\to f}(p)))},$$

where x_1 (x_2) is the number of points in the intersection of C^f_1 (C^f_2) and the KNN of $\tau_{f+1\to f}(p)$, $x_{l(p)}$ is one of x_1 and x_2, and $d(p_1,p_2)$ denotes the Euclidean distance of p_1 and p_2. In case that p has correspondence neither C^f_1 nor C^f_2, we always defined $\mathcal{E}^{\mathrm{s}}_{\mathrm{data}}(p,l)$ as a constant regardless what the label of p is.

The regularity term is defined based on the point connectivity in the KNN graph of C^{f+1}_k:

$$\mathcal{E}^{\mathrm{s}}_{\mathrm{reg}}(p_1,p_2,l) = \begin{cases} 0, & l(p_1) = l(p_2), \\ 1, & l(p_1) \neq l(p_2). \end{cases}$$

5.2. Space-time grouping of segment sequences

After the cut propagation, we obtain an over-segmentation for all frames in terms of both space and time. Spatially, a rigid part

Table 1: *Description of tested datasets.*

Datasets	#Fra.	#Ave. of pts.	Data sources	#Fig.
Cabinet	30	92126	Kinect	9(t)
Fan	50	19807	Kinect	9(b)
One girl	20	1134	Kinect	12
Two girls	175	5478	Kinect	10
Panther	40	30684	Range	1
Hand	130	36237	Range	11
Horse	32	31891	Virtual	13

Table 2: *Average computation time (in second) per sequence.*

Datasets	#Samp. pts.	Reg.	Ind. seg.	Co-seg.	Tot.
Cabinet	2688	38	158	6	202
Fan	1966	18	73	6	97
One girl	1134	5	13	6	24
Two girls	1601	12	34	14	60
Panther	1037	5	13	18	36
Hand	2001	19	56	16	91
Horse	2040	20	79	25	124

in a frame could be over-split into more than one segments which should be merged. In temporal direction, there is the case that a rigid part is missing in some frames due to occlusion, leading to broken sequences of some rigid motion. It is desirable to concatenate these sequences into a longer trajectory, if the missing frames do not take a long time interval. We propose to address the above two issues in a *unified* space-time grouping framework, which is again formulated as a graph-cut problem.

Note that the segmentation for each frame has been updated after cut propagation, for which we still use G^f to refer the segment graph of C^f. Each node of G^f generally has a corresponding node in both G^{f-1} and G^{f+1} representing the same rigid motion.

To perform space-time segmentation, a global graph G of subsequences is built for the whole sequence based on the graphs $\{G^f : f = 1,2,\cdots,F\}$; see Figure 8(a) for an illustration. Noting that a set of segment sequences have been generated during cut propagation, we connect all node pairs from adjacent frames if they are two successive segments in one segment sequence (Figure 8(b)). All nodes associated with the same segment sequence are then collapsed into one node (c). Furthermore, we traverse the nodes of the current graph (c) to detect node pairs which potentially represent the motion of the same part of the object. For each node, its adjacent nodes are checked to find node pairs where the two nodes represent the same region of the object but their corresponding segment sequences do not overlapping temporally. Collapsing all connected nodes yields the graph of sub-sequences (d).

We now turn to evaluate the graph edge weights that measure the similarity between two segment trajectories. Let $e = (u,v)$ be an edge of G, where u and v are two neighboring sub-sequences. Suppose that $M_u = (R_u^{s(u)},\ldots,R_u^{e(u)})$ and $M_v = (R_v^{s(v)},\ldots,R_v^{e(v)})$ are the motion models associated with u and v, and $s(u)$ and $e(u)$ represent the start and end time for sub-sequences u (the same goes for v), respectively. We then define the similarity between u and v as:

$$\mathcal{S}(u,v) = e^{-\frac{y(u,v)}{y_{max}} - \frac{z(u,v)}{z_{max}}},$$

where $y(u,v) = max\{||R_t^u - R_t^v||_F : max\{s_u,s_v\} \leq t \leq min\{e_u,e_v\}\}$ with $||\cdot||_F$ as Frobenius norm. y_{max} is the maximum value of $y(u,v)$. $z(u,v)$ is the average point number of segments associated with u and v, with z_{max} being the maximum value.

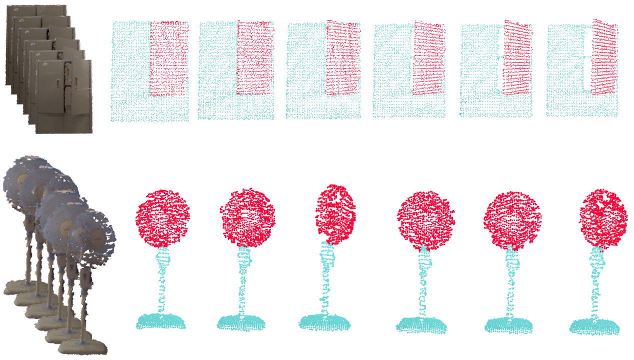


Figure 9: *Applying our algorithm on man-made objects with mechanical motion (the input data are shown with the overlapped sequence to the left). Top row: a cabinet with an opening door. Bottom row: a stand fan with a shaking head.*

6. Experimental results and applications

Our space-time co-segmentation algorithm was implemented in C++ and tested on an Intel(R) Core(TM) i5-2300CPU@2.80GHz with 12GB RAM. The result shown in Figure 1 demonstrates the capability of our method in handling raw inputs of point cloud sequence. Note that the point cloud in each frame is a single view scan. Our method can also robustly extract some tiny parts (e.g., mouth and ears) with small motion. In this section, we present more examples (Figures 9-14), including both real-world and synthetic datasets to qualitatively demonstrate and quantitatively evaluate the effectiveness and robustness of the proposed method. Following that, the limitations are discussed and showcased with Figure 15.

Datasets, parameters and performance. The *datasets* we tested include three categories: low quality motion data acquired using Kinect, animated range scan data captured by rather high quality 3D sensors, and synthetic data generated by virtual scanning. Table 1 summarizes several important statistics about the input data we have tested, including frame count (#Fra), the average number of points per frame (#Ave. of pts), and the acquisition method. There are tunable *parameters* designed in our algorithm, including the sampling rate (section 4), the length $(2r+1)$ of local temporal trajectories (subsection 4.2), the threshold ε in trajectory analysis (subsection 4.2), the smoothing regularity weight λ (subsection 4.3), and the propagation regularity weight μ (subsection 5.1). We use their default values throughout all examples presented here. The computation *performance*, which mainly consists of three stages: registration (Reg.), individual frame segmentation (Ind. seg.) and co-segmentation (Co-seg), is recorded in Table 2 as well as the total time (Tot.) spent. The average number of sampling

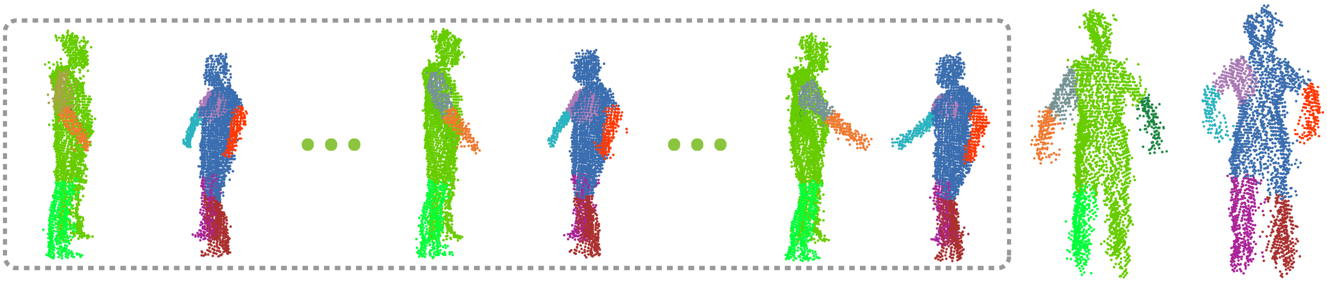


Figure 10: *Space-time co-segmentation over a sequence with multiple moving objects. Our method not only separates the two girls but also accurately identifies their body parts based on the articulated motion from highly noisy, sparse and incomplete data.*

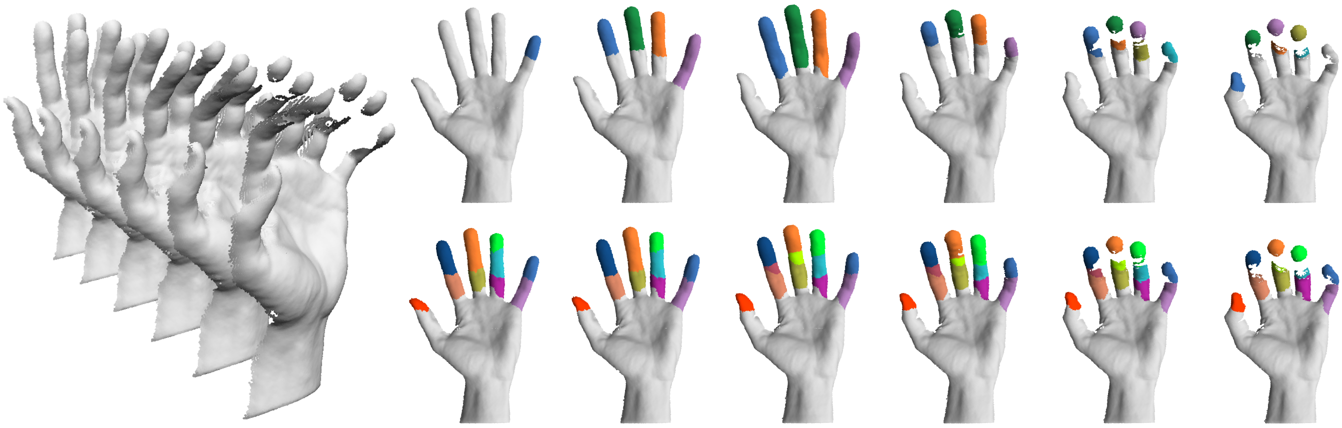


Figure 11: *Space-time co-segmentation of a clenching hand over the motion data shown as the overlapped scan sequence to the left. Top row: per-frame motion-based segmentation. Bottom row: our co-segmentation result.*

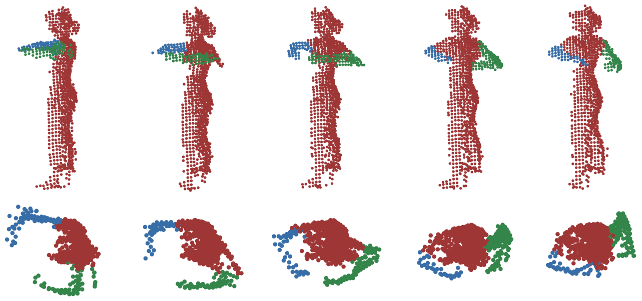


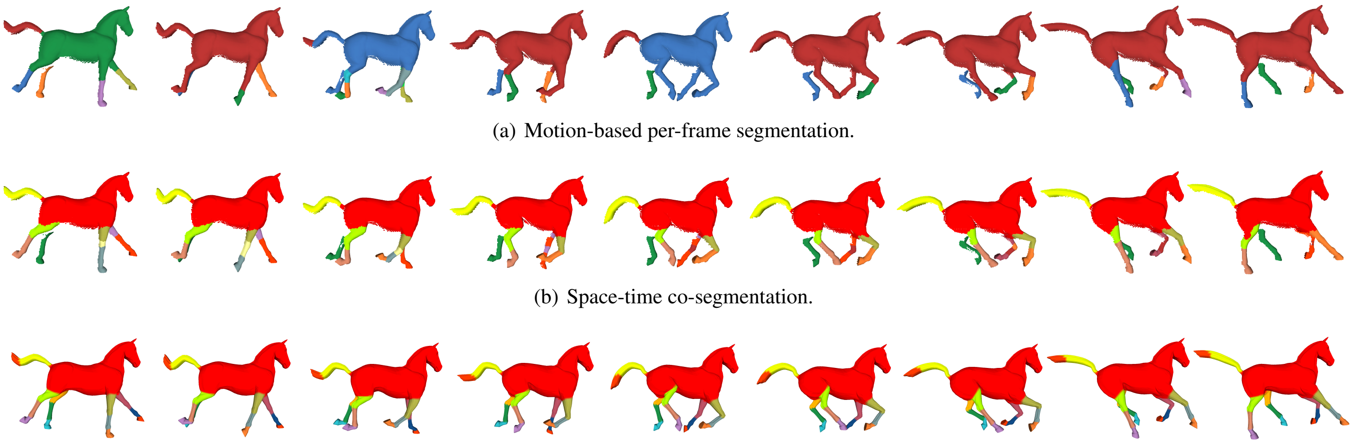
Figure 12: *Co-segmentation of a sequence with topology changes in the captured motion: top and bottom rows render segmented raw point clouds from two different views, respectively. Clearly, our method is able to accurately and consistently identify the girl's two swing arms, which move fast and occasionally stick to the body.*

points for each frame (#Samp. pts.) is also depicted. Note that the sampling rate varies for different datasets, but we use a 5-level octree to perform the sampling for all experiments.

Motion-based space-time co-segmentation. The rigid motion assumption of our method makes it especially suitable for analyzing man-made objects with mechanical motions. Figure 9(top) shows an example in which the door of a cabinet is gradually opened. The input data (Figure 9(top-left)) was captured using Kinect, and our method can correctly detect the moving and still parts (Figure 9(top-right)). Figure 9(bottom-left) depicts a stand fan rotates around the axis of its support strut, also acquired using Kinect. The still strut and the moving head is correctly separated (Figure 9(bottom-right)) by our co-segmentation algorithm.

Figure 10 demonstrates an example of multiple moving objects with mutual interactions. Specially, it shows a motion scene with two girls approaching to each other and shaking hands. From the result, our method can not only separate different humans but also identify each body part with nearly-rigid motion. Moreover, note how sparse and noisy the raw input data is due to the low resolution of the Kinect depth maps.

Figure 11(left) presents an example of a clenched hand, which contains a large near-rigid part (the palm) and a set of small motion parts (finger). A hand contains tens of joints, making the co-segmentation very challenging. Especially for the last few frames to the right, where the bent fingers introduce severe occlusion. The result (right) demonstrates that the proposed co-segmentation method



(a) Motion-based per-frame segmentation.

(b) Space-time co-segmentation.

(c) Segmented and animated clean mesh sequence (ground truth).

Figure 13: *Space-time co-segmentation for the dataset of a galloping horse. Compare our individual per-frame segmentation (a) and co-segmentation (b) with the ground truth (c), which is a pre-segmented and animated mesh sequence.*

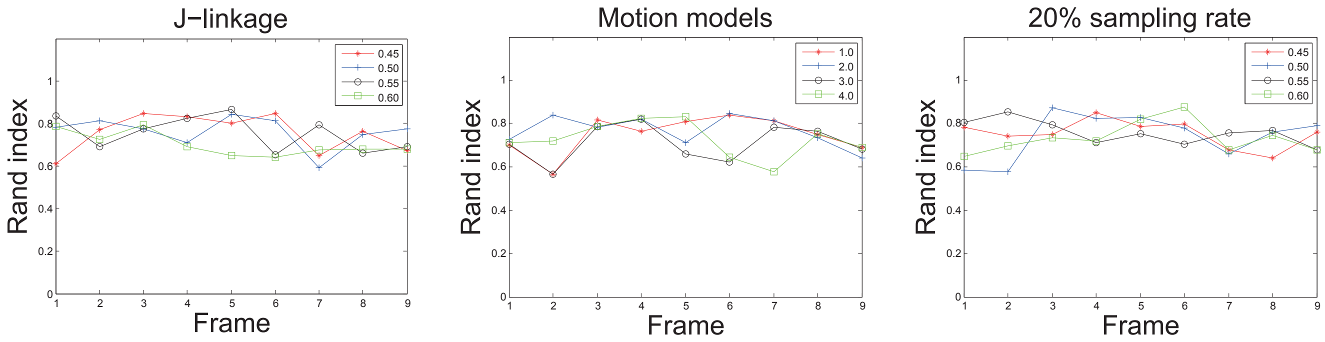


Figure 14: *Influence of three key parameters on motion co-segmentation results. Left: the percentage ρ for determining the J-linkage threshold. Middle: the number of initial motion models (transformation matrices). Right: the sampling rate. The rand indices are computed at all cases to measure the approximation degree of our co-segmentation to the co-segmentation ground truth (Figure 13(c)).*

can establish correct correspondences between the segments of different frames, even for the frames with occluded parts.

Figure 12 depicts an example with topology changes in the captured motion sequence. The performer swings her two arms back and force, which move fast and occasionally stick to the body then separate. Our method accurately segments two arms out from the body at each pose frame, performing robustly to topology changes.

Quantitative evaluation against ground-truth. To quantitatively evaluate the segmentation quality of our method, we compare our result against a co-segmentation ground-truth derived from a segmented and animated mesh sequence. Specifically, given a horse mesh, we manually segment it into rigidly moving parts and then generate a dynamic sequence by animating its segmented parts with consistent correspondences. The co-segmented mesh sequence can therefore serve as the ground truth; see Figure 13(c). We then virtually scan the mesh sequence from only two views to generate an incomplete point cloud sequence for testing our motion-based per-frame segmentation (Figure 13(a)) and space-time co-segmentation (Figure 13(b)). We evaluate the effectiveness of our segmentation technique using the Rand index criterion proposed in [CGF09]. For the co-segmented nine frames presented in Figure 13(b), the corresponding Rand index values from the left to right are 0.91, 0.92, 0.93, 0.93, 0.93, 0.92, 0.93, 0.94, 0.89, respectively. All values are relatively high and verify that our algorithm produces good segmentation results for all dynamic frames, behaving robustly.

Furthermore, we evaluate the influence of three key parameters on our motion co-segmentation results in Figure 14, which include the percentage ρ for determining the threshold ε of J-linkage (Section 4.2) at left, the number of initial motion models (Section 4.2) in the middle and the sampling rate (Section 4) at right. We change one parameter while fixing others to generate the co-segmentation on the motion sequence, and then estimate the Rand index of each our result against to the co-segmentation ground truth as shown

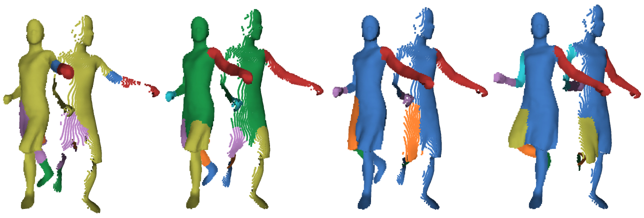


Figure 15: *Our method fails for sequences which are seriously violating the rigidity assumption, e.g., a clip of dancing performance with non-rigid skirt movement.*

Figure 13(c). Four parametric values are depicted in all three subplots of Figure 14. This experiment verifies the robustness of our algorithm against the parameter tuning. All results evaluated sustain a high Rand index.

Limitations The effectiveness of the proposed approach depends on the preciseness of the correspondence built between adjacent frames. Therefore, it may fail for those datasets in which adjacent frames exhibit substantial movements. It often results in mismatching between adjacent frames in such cases. In addition, though our approach is robust to the performances with rigid parts of the object being added with small perturbation, it can not work out a good segmentation for those performances with large-scale non-rigid movements as shown in Figure 15. In this example, the skirt of the dancing girl flutters around, making our method hard to obtain a suitable co-segmentation of the soft skirt regions.

7. Conclusions

We have described a local-to-global approach to co-segment dynamic point clouds. All frames of the point clouds are consistently decomposed into nearly-rigid parts implied by the motion. Our method first conducts per-frame segmentation via local trajectories analysis. Propagating the segmentation cuts of individual frames between every two adjacent frames yields a globally consistent over-segmentation. Finally, we devise a space-time grouping algorithm over a graph of temporal segment sequences to achieve space-time co-segmentation.

Our algorithm can handle the raw input of low quality motion data, which often exhibits noise, outliers, significant missing regions, and geometric variations due to occlusions and view changes. We have tested the proposed technique on a variety of datasets, consisting of object motions in real world acquired by low-cost Kinect sensor, dynamic geometries captured by 3D scanners, and synthetic data generated by virtual cameras. Experimental results clearly demonstrate that our method can produce plausible space-time co-segmentation for most of the motion datasets and is robust against to heavy noise and high incompleteness of point cloud sequences.

Due to the complexity and defect of dynamic motion data, co-segmenting them into meaningful parts with both spatial and temporal coherence is a very challenging task. In the future, we plan to combine the analysis of both geometric and semantic information to improve the accuracy of the segmentation boundaries. Non-rigid motion guided co-segmentation is another interesting direction with high potential. We would also like to explore applications that may benefit from our co-segmentation results, such as the global registration among consecutive 3D frames and 4D consensus skeleton representation.

Acknowledgments

We would like to thank all the reviewers for their valuable comments and constructive suggestions. This work was supported in part by NSFC (61572202, 61522213, 61572507), 973 Program (2015CB352501, 2014CB360503), Guangdong Science and Technology Program (2015A030312015, 2014B050502009, 2014TX01X033, S2013020012795), Shenzhen Innovation Program (JCYJ20151015151249564, CXB201104220029A) and Research Funds for the Central Universities (F020501).

References

[ACH*13] ARCILA R., CAGNIART C., HETROY F., BOYER E., DUPONT F.: Segmentation of temporal mesh sequences into rigidly moving components. *Graphical Models 75*, 1 (2013), 10–22. 2

[ASKK11] AKHTER I., SHEIKH Y., KHAN S., KANADE T.: Trajectory space: a dual representation for nonrigid structure from motion. *IEEE Trans. Pattern Analysis & Machine Intelligence 33*, 7 (2011), 1442–1456. 3

[BP10] BACH F., PONCE J.: Discriminative clustering for image co-segmentation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2010), pp. 143–195. 2

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 8*, 3 (2009), 73:1–73:12. 2, 9

[CLM*12] CHANG W., LI H., MITRA N. J., PAULY M., WAND M.: Dynamic geometry processing. In *Eurographics 2012: Tutorial Notes* (2012). 2

[CZ11] CHANG W., ZWICKER M.: Global registration of dynamic range scans for articulated model reconstruction. *ACM Trans. on Graphics 30*, 3 (2011), 26:1–26:15. 2

[DFB13] DJELOUAH A., FRANCO J.-S., BOYER E.: Multi-view object segmentation in space and time. In *Proc. Int. Conf. on Computer Vision* (2013), pp. 2640–2647. 2

[FRP09] FRADET M., ROBERT P., PEREZ P.: Clustering point trajectories with various life-spans. In *Proc. Conf. Visual Media Production* (2009), pp. 7–14. 3, 4, 5

[FSB10] FOUHEY D. F., SCHARSTEIN D., BRIGGS A. J.: Multiple plane detection in image pairs using j-linkage. In *Proc. IEEE Int. Conf. on Pattern Recognition* (2010), pp. 336–339. 4

[FSL*15] FISHER M., SAVVA M., LI Y., HANRAHAN P., NIESSNER M.: Activity-centric scene synthesis for functional 3D scene modeling. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 34*, 6 (2015). 2

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers & Graphics 33*, 3 (2009), 262–269. 2

[GSLB12] GHOSH S., SUDDERTH E. B., LOPER M., BLACK M. J.: From deformations to parts: motion-based segmentation of 3D objects. In *Proc. Neural Information Processing Systems* (2012), pp. 2006–2014. 2

[HFL12] HU R., FAN L., LIU L.: Co-segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum (Proc. of SGP) 31*, 5 (2012), 1703–1713. 2

[HRF13] HERBST E., REN X., FOX D.: RGB-D flow: Dense 3D motion estimation using color and depth. In *Proc. IEEE Int. Conf. on Robotics & Automation* (2013), pp. 2276–2282. 2

[LWC06] Lee T. Y., Wang Y. S., Chen T. G.: Segmenting a deforming mesh into near-rigid components. *The Visual Computer 22*, 9 (2006), 729–739. 2

[LXL*12] Liao B., Xiao C., Liu M., Dong Z., Peng Q.: Fast hierarchical animated object decomposition using approximately invariant signature. *The Visual Computer 28*, 4 (2012), 387–399. 2

[MXLH13] Meng M., Xia J., Luo J., He Y.: Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. *Computer-Aided Design 45*, 2 (2013), 312–320. 2

[NFS15] Newcombe R., Fox D., Seitz S.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2015), pp. 343–352. 2

[PB11] Papazov C., Burschka D.: Deformable 3D shape registration based on local similarity transforms. *Computer Graphics Forum 30*, 5 (2011), 1493–1502. 3, 4

[RBB*12] Rosman G., Bronstein A. M., Bronstein M. M., Tai X. C., Kimmel R.: Group-valued regularization for analysis of articulated motion. In *Proc. Workshop of Euro. Conf. on Computer Vision* (2012), pp. 52–62. 2

[RBB*13] Rosman G., Bronstein M. M., Bronstein A. M., Wolf A., Kimmel R.: Group-valued regularization for motion segmentation of articulated shapes. In *Innovations for Shape Analysis: Models and Algorithms* (2013), pp. 263–281. 2

[RBBK12] Rosman G., Bronstein A. M., Bronstein M. M., Kimmel R.: Articulated motion segmentation of point clouds by group-valued regularization. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2012), pp. 77–84. 2

[RL01] Rusinkiewicz S., Levoy M.: Efficient variants of the icp algorithm. In *Proc. Int. Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152. 4

[SCH*14] Savva M., Chang A. X., Hanrahan P., Fisher M., Niessner M.: Scenegrok: Inferring action maps in 3D environments. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 33*, 6 (2014), 212:1–212:10. 2

[SFC*11] Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., Blake A.: Real-time human pose recognition in parts from a single depth image. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2011), pp. 1297–1304. 2

[SMW06] Schaefer S., McPhail T., Warren J.: Image deformation using moving least squares. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 25*, 3 (2006), 533–540. 4

[SvKK*11] Sidi O., van Kaick O., Kleiman Y., Zhang H., Cohen-Or D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 30*, 6 (2011), 126. 2

[TF08] Toldo R., Fusiello A.: Robust multiple structures estimation with j-linkage. In *Proc. Euro. Conf. on Computer Vision* (2008), pp. 537–547. 4

[TK92] Tomasi C., Kanade T.: Shape and motion from image streams: a factorization method full report on the orthographic case. *Int. J. Computer Vision 9*, 2 (1992), 137–154. 3

[VF14] Vasilakis A., Fudos I.: Pose partitioning for multi-resolution segmentation of arbitrary mesh animations. *Computer Graphics Forum (Proc. of Eurographics) 33*, 2 (2014), 293–302. 2

[vKZHCO11] van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 4

[WB10] Wuhrer S., Brunton A.: Segmenting animated objects into near-rigid components. *The Visual Computer 26*, 2 (2010), 147–155. 2

[WWS*13] Wu Z., Wang Y., Shou R., Chen B., Liu X.: Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering. *Computers & Graphics 37*, 6 (2013), 627–637. 2

[XLZ*10] Xu K., Li H., Zhang H., Cohen-Or D., Xiong Y., Cheng Z.: Style-content separation by anisotropic part scales. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 29*, 5 (2010), 184:1–184:10. 2

[YPG01] Yee H., Pattanaik S., Greenberg D. P.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. on Graphics 20*, 1 (2001), 39–65. 4

[ZAB*11] Zaheer A., Akhter I., Baig M. H., Marzban S., Khan S.: Multiview structure from motion in trajectory space. In *Proc. Int. Conf. on Computer Vision* (2011), pp. 2447–2453. 3

[ZST*10] Zheng Q., Sharf A., Tagliasacchi A., Chen B., Zhang H., Sheffer A., Cohen-Or D.: Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum (Proc. of Eurographics) 29*, 2 (2010), 635–644. 3