Consistent Segment-wise Matching with Multi-Layer Graphs

Taiwei Wang[a], David George[a], Yu-Kun Lai[b], Xianghua Xie[a], Gary K.L. Tam[a]

[a] *Department of Computer Science, Swansea University Bay Campus, Fabian Way, Swansea, SA1 8EN, UK*
[b] *School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff, CF24 3AA, UK*

Abstract

Segment-wise matching is an important research problem that supports higher-level understanding of shapes in geometry processing. Many existing segment-wise matching techniques assume perfect input segmentation, and would suffer from imperfect or over-segmented input. To handle this shortcoming, we propose multi-layer graphs (MLGs) to represent possible arrangements of partially merged segments of input shapes. We then adapt the diffusion pruning technique on the MLGs to find consistent segment-wise matching. To obtain high quality matching, we develop a voting step to find hierarchically consistent correspondences as final output. We evaluate our technique with both qualitative and quantitative experiments on both man-made and deformable shapes. Experimental results demonstrate the effectiveness of our technique when compared to two state-of-the-art methods.

Keywords: Segment-wise matching, Mesh models, Shape analysis

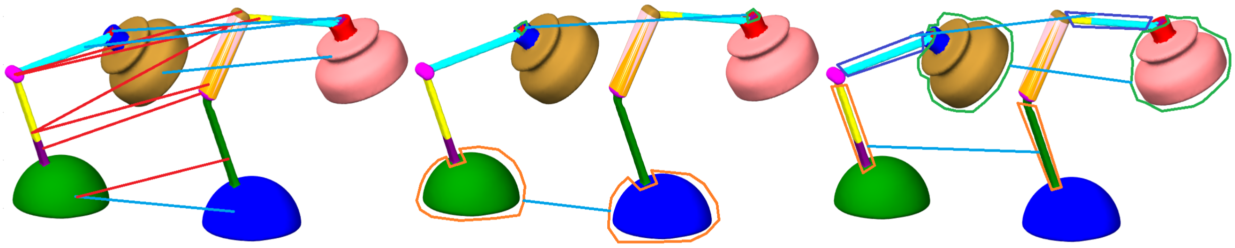
1. Introduction

Given two similar 3D meshes with pre-defined segments, 3D segment-wise matching aims to establish meaningful correspondences of segments between the two meshes. It is an important problem as it helps with higher-level and hierarchical understanding in geometry analysis Zhu et al. (2017). It further impacts many downstream applications, like defining better similarity measures between 3D models Kleiman et al. (2015); Shapira et al. (2010); Kleiman and Ovsjanikov (2017), functionality analysis van Kaick et al. (2013a), surface registration Huang et al. (2008) and structure-aware analysis Mitra et al. (2013).

A few notable techniques have been proposed in the recent literature. Many of them combine topological and geometrical information to help solve the segment-wise matching problem. Kleiman et al. (2015); Kleiman and Ovsjanikov (2017) both take input shape segments and build a component graph to capture the topological relationship of segments. Together with geometric similarity of segments, they adapt the spectral technique Leordeanu and Hebert (2005) for matching. SHED (Shape Editing Distance) Kleiman et al. (2015) innovates to consider one-to-many matching whilst Kleiman and Ovsjanikov (2017) focuses on robust matching of non-isometrically deformed segments and disambiguating symmetric segments. Alhashim et al. (2015) also takes pre-defined shape segments as input and builds a component graph to represent their topology. To solve the segment-wise matching problem, they use a deformation energy as an effective constraint to produce higher-level semantic matching results. Zhu et al. (2017) builds a component hierarchical graph using a binary partition technique. Their matching technique adopts a top-down approach and achieves good results.

We observe two problems for the methods in existing literature. First, most of these techniques rely on input with consistent segmentation Kleiman et al. (2015); Kleiman and Ovsjanikov (2017); Alhashim et al. (2015); Zhu et al. (2017). When the input segmentation is inconsistent (over-/imperfectly segmented), they often lead to incorrect correspondences. For example in Figure 1, the two lamps are inconsistently segmented

Email addresses: `688818@swansea.ac.uk` (Taiwei Wang), `654214@swansea.ac.uk` (David George), `LaiY4@cardiff.ac.uk` (Yu-Kun Lai), `x.xie@swansea.ac.uk` (Xianghua Xie), `k.l.tam@swansea.ac.uk` (Gary K.L. Tam)



(a) SHED Matching Result (b) Our Bottom Layer Matching Result (c) Our Higher Layer Matching Result

Figure 1: Example matching of inconsistently (over/imperfectly) segmented shapes. In all figures in this paper, color of segment indicates segment boundary only (not correct correspondences). Instead, we use blue lines for correct correspondences and red lines for incorrect ones (according to our user study). We further use polygons with the same color to indicate one-to-merged or merged-to-merged correspondences in our results. In this example, it is difficult to define a correct correspondence for the middle (purple) joint of the left lamp. In our results we do not force full matching but leave it as unmatched to reduce incorrect matching. Full matching techniques such as SHED produce incorrect matching between inconsistently segmented regions.

(one has more segments than the other on the joint). Kleiman et al. (2015) (Figure 1a) investigates one-to-many correspondences and further requires full matching, i.e. every segment from one shape is matched to at least one segment in another shape. Affected by the different joint composition on the right lamp, the topology (graph distance) of the underlying component graphs differs a lot. As a result Kleiman et al. (2015) returns incorrect matchings (indicated by red lines). Second, correct segment-wise matching also depends on the global shapes and functionality. For example, in Figure 1b the upper stick of the right lamp and the lower stick of the left lamp are over-segmented into two segments. Ideally, the left lamp's upper stick should be matched to all segments of upper stick on the right lamp. This requires merging of segments before a meaningful consistent segment-wise matching can be established (Figure 1c). These observations inspire us to investigate the following research questions:

- Can a technique that handles moderate topological changes in the underlying segment graphs improve matching results?
- Can merged segments help improve the accuracy of segment-wise matching with inconsistent (over-/imperfectly) segmented inputs?
- How can we develop a representation that facilitates matching of merged segments, and a technique for robust segment-wise matching?

To address these questions, we propose to construct multi-layer graphs (MLGs) to represent the input shapes with inconsistent segments. Inspired by Laga et al. (2013), an MLG is a graph consisting of nodes with input and merged segments which is built in a bottom-up manner by neighbor merging. Different from Laga et al. (2013), our merging technique uses many possible combinations based on the connectivity (if two segments share common faces/vertices) of input segments. In this way we achieve better capability with over-/imperfect input segmentation than Laga et al. (2013).

Next we find consistent matching between MLGs by adapting the diffusion pruning (DP) technique Tam et al. (2014b) and using both geometric and topological constraints. Inspired by spectral techniques, DP computes matching results by inferring global consistency from the local matching. It has been shown to be robust against moderate non-isometric deformation Tam et al. (2014b). It would allow us to handle moderate changes in graph distance due to over/imperfect input segmentation.

Further, different from existing techniques Kleiman et al. (2015); Kleiman and Ovsjanikov (2017) that apply spectral matching on component graphs built from input segments only, we apply DP on the proposed multi-layer graphs (MLGs) consisting of both input and merged segments. Compared to Kleiman et al. (2015) which innovates in one-to-many matching, our technique can offer both one-to-merged and merged-to-merged correspondences. From our experiments, our technique produces better results than Kleiman et al. (2015). The obtained matching results are also consistent across layers while existing top-down approach Zhu et al. (2017) may fail (see Section 8). In summary, our **contributions** include:

- We propose a multi-layer graph (MLG) representation to capture detailed geometric, topological and hierarchical information from the input and merged segments of shapes.
- We propose a matching technique to obtain geometrically, topologically and hierarchically consistent matching results with over/imperfectly-segmented inputs. From our experiments, it outperforms Kleiman et al. (2015) quantitatively and qualitatively in our user study.
- To the best of our knowledge, this is the first technique which can obtain meaningful merged-to-merged segment-wise correspondences. This has not been considered before in the literature.

To be consistent throughout this paper, we use the term "components" for semantic parts obtained from perfect segmentation that respect human intuition. "Segments" instead refer to regions resulted from perfect or imperfect segmentation. We discuss related work in Section 2. Section 3 provides an overview of our technique. Then we discuss the construction of MLGs from input shapes and initial matching computation in Section 4. Section 5 explains diffusion pruning and how to adapt it on MLGs. After that we vote the pruned results in Section 6. We evaluate our method in Section 7. Finally, discussions and conclusions are presented in Sections 8 and 9.

2. Related Work

Our method involves global geometry features, partial matching between shapes, and hierarchical analysis of shape topology. We summarize and discuss existing works related to ours below.

2.1. Global Geometry Features

There are many important shape features developed over the past decades. We mention some important features, and those that are particularly relevant in this section. We would like to refer readers to recent surveys Tam et al. (2013); van Kaick et al. (2011).

Light Field Descriptor Chen et al. (2003) is one of the notable geometry descriptors. It is based on a set of 2D images of the input shape (captured from different angles) and use image-based features for measuring shape similarity. Ankerst et al. (1999) introduces a 3D shape histogram approach with sampled points on meshes to determine shape similarity. Osada et al. (2002) further extends 3D shape histograms into A3/D1/D2/D3/D4 descriptors with different random sampling based measures. Blomley et al. (2014) uses eigenvalues from PCA to determine shape distribution features (such as linearity, sphericity, omni-variance, change of curvature). These distribution-based features may be unreliable in certain cases (e.g. the left base and right cap have similar scores in Fig. 6). Heat Kernel based descriptors such as Heat Kernel Signature (HKS) Sun et al. (2009) use heat diffusion on meshes to define point-based features. Persistent-HKS Dey et al. (2010) extends HKS and can be used as a descriptor for partial matching of non-rigid shapes.

Our proposed technique mainly uses LFD as it is more robust for small segments. In general, local features can be used to obtain initial matching, but the results are likely to be globally inconsistent. Our technique aims to produce consistent segment-wise matching results.

2.2. Shape Registration and Matching

Shape registration and point-based matching is an important research area with long history Tam et al. (2013). The research challenges are to develop robust and accurate techniques to handle shapes undergoing different real-life transforms (rigid) and deformations (non-rigid), including near-/non-isometric deformations Kim et al. (2011). Finding subsets of sampled shape features can help form meaningful or semantic matching van Kaick et al. (2011). There are further many existing works, e.g. Maciel and Costeira (2003); Berg et al. (2005); Gelfand et al. (2005); Zhang et al. (2008) which rely on sampled/key points on input shapes, and then use designated objective functions to analyze alignment/distortion errors and generate matching. One of the notable techniques van Kaick et al. (2013b) uses deformation distortions to obtain semantic matching.

Compared to other techniques that require specific constraints (e.g. sphere topology Kim et al. (2011)), one of the notable matching techniques Leordeanu and Hebert (2005) uses spectral analysis and has inspired many subsequent and useful point-based matching and registration techniques e.g. Huang et al. (2008). The spectral pruning technique Huang et al. (2008) assumes near-isometric deformation using global geodesic isometry. However, when the deformation is large (becoming non-isometric deformation), the technique does not perform well. Tam et al. (2014b) proposes a diffusion pruning (DP) technique to infer global consistency from locally consistent matching. It has been shown to handle moderate non-isometric deformation well. We adapt DP on multi-layer graphs to handle moderate change in topological distances in the segment graph. A complete literature survey of shape registration and matching techniques is beyond the scope of this paper. We would like to refer readers to surveys (e.g. Tam et al. (2013) and van Kaick et al. (2011)).

2.3. Hierarchical Understanding

Some works solve the shape matching/synthesis problem using a hierarchical approach for higher-level understanding. Chaudhuri et al. (2011); Kalogerakis et al. (2012); Shapira et al. (2010) use graphs encoded with probabilistic and topological information to solve region-wise matching or shape synthesis problems. Zheng et al. (2013) converts input shapes into component relationship graphs and then combines graph subsets with designated symmetric functional arrangement for synthesizing new shapes. Alhashim et al. (2015) combines component relationship graphs and deformation energy constraints to establish meaningful segment-wise correspondences of input shapes. Binary decomposition approaches are also used to help with hierarchical understanding. Wang et al. (2011) introduces a novel shape representation in a binary hierarchical manner which cuts a shape from-whole-to-segment hierarchically. Zhu et al. (2017) finds the best binary segmentation in a top-down manner, via matching along the object hierarchy and uses recognition measures to better handle structural variations and inconsistent initial segmentation than Alhashim et al. (2015). The technique however may fail in fine-grained matching because such cases lack the support of cross-layer information (see more discussion in Section 8.) Laga et al. (2013); Pechuk et al. (2008) focus on merging shape parts to form a hierarchical graph representation of part-functionality with geometry and topological information. Inspired by all these works, we propose to build a multi-layer graph by merging adjacent nodes in a bottom up manner. We do not define specific constraints (e.g. functional constraint Zheng et al. (2013) or binary segmentation Zhu et al. (2017)). The search space we consider, compared to existing work, is arguably larger. To address this, we further develop a robust matching technique to discover meaningful segment correspondences even under inconsistent (over/imperfect) input segmentation.

2.4. Segment-Wise Matching

A few works in the literature focus on segment-wise matching which we survey here. Kleiman and Ovsjanikov (2017) relies on HKS features for pre-segmentation. It uses spectral matching to find segment-wise correspondences with a focus on symmetric/pairwise issues. However, it outputs pair-to-pair correspondences, and may lead to no matching if there are left-right symmetry issues. Alhashim et al. (2015) uses combinatorial tree search and a deformation energy constraint to establish meaningful segment-wise correspondences. One shortcoming of this method is that it may not work on fine-grained segmented shapes. Zhu et al. (2017) finds the best binary segmentation in a top-down manner, and matches along the object hierarchy. It uses recognition measures to better handle structural variations and imperfect initial segmentation than Alhashim et al. (2015). This method does not exploit matching from object hierarchies and may result in some incorrect correspondences (see also Figure 14a). SHED (Shape Editing Distance) Kleiman et al. (2015) takes shape segments and performs matching to define a better shape similarity measure. It innovates to find both one-to-one and one-to-many segment-wise correspondences, using both geometry and topology information. It forces full matching which means each input segment must have at least one correspondence to another shape, which helps resolve some ambiguities with perfect input segmentation, but when the input segmentation is inconsistent, incorrect matching may result.

To our knowledge, none of the existing techniques consider inconsistent (over-/imperfect) input segmentation. Our technique is the first work to handle this challenge. Our novel idea is to use a multi-layer graph to represent possible merging arrangement, and carry out our matching on such graphs. Together with a novel voting step, our results are shown to be geometrically, topologically and hierarchically consistent.

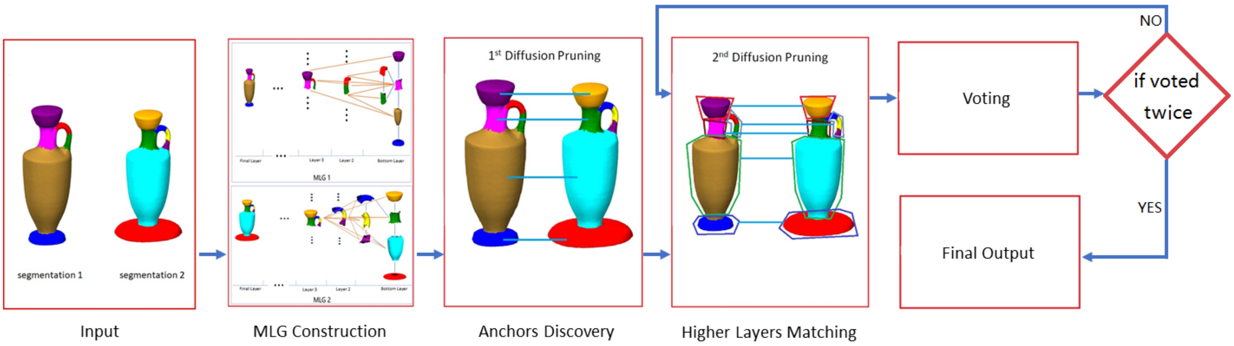


Figure 2: Method overview: our technique first builds multi-layer graphs to represent the input meshes from the pre-defined segmentation. Such pre-defined segmentation may be inconsistent between two shapes. Next we adapt diffusion pruning (DP) Tam et al. (2014b) on the bottom layer to find anchors. With the support of anchor correspondences, we apply DP again on the multi-layer graphs to obtain initial matching. A voting technique is further applied to confirm high quality segment-wise correspondences using matching from high layers.

3. Method Overview

Figure 2 shows an overview of our proposed method for segment-wise matching with inconsistent input segmentation. It involves four steps, namely **multi-layer graph construction** (Section 4), **discovery of anchor correspondences** (Section 5), **higher layer matching** (Section 5.2) and **voting** (Section 6).

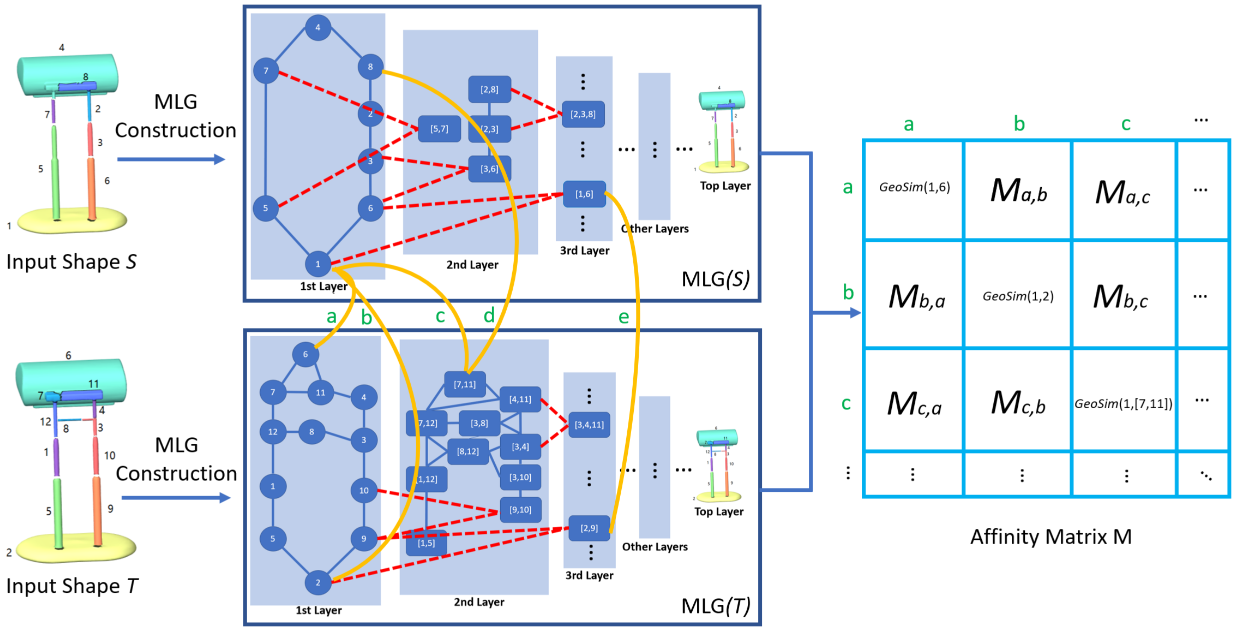
Given two shapes with inconsistent segments, we build two hierarchical segment graphs (referred to as multi-layer graphs, MLGs) to represent the original shapes. Each input segment in a shape is assigned a graph node. All input segment nodes are grouped into one layer, denoted as the **bottom layer**. A merging stage is then applied to the nodes in the bottom layer to construct the MLG. It generates new nodes and new layers and is applied recursively until all nodes are merged into one — the original shape. After we have built two MLGs, we compute geometry similarities between nodes in the two MLGs for initial matching. Next, we adapt the diffusion pruning technique to compute good matching. There are two stages: the first pruning stage involves only the bottom layer in both MLGs. This is inspired by Kleiman et al. (2015) as SHED provides reasonable results with perfect segmentation. Only strong results are used as anchors for the second pruning stage. For inconsistent input with large topological/geometrical variation however, using only nodes in the bottom layer alone often does not provide acceptable results. The second pruning further uses these anchors and involves more layers than previous pruning computation. Finally, we apply our voting technique to extract and confirm highly confident segment matching, using correspondences in higher layers.

4. Multi-Layer Graph and Initial Matching

Given a shape with predefined segments, we define the multi-layer graph (MLG) as a hierarchical representation. It covers possible merging arrangements of segments that are adjacent in a shape. An MLG consists of nodes and edges. Nodes are further grouped into layers. Bottom layer (layer 1) consists of input segment nodes whilst higher layers consist of nodes due to merging of two adjacent nodes in a lower layer. Nodes in internal layers are further connected by edges indicating their adjacent connections (within layer) and where the nodes are merged from (across this and lower layer). The highest layer consists of only one node. It represents the entire shape where all segments are merged. We first define the construction of multi-layer graph equipped with a specific volume constraint, and then discuss the initial correspondences.

4.1. Multi Layer Graph

Node Construction with Volume Constraint. Precisely, let $\mathcal{S} = (V, E)$ be a 3D shape with sets of vertices V, edges E and pre-defined input segments $\{S_1, S_2, S_3, ...\}$ where $\mathcal{S} = \bigcup S_i$ is the union of vertices $\subset V$ and edges $\subset E$ in S_i. Denote by $\bar{N}_k^{[l]}$ the k^{th} node in the l^{th} layer of a source shape MLG($\mathcal{S}$). We



	a	b	c	...
a	GeoSim(1,6)	$M_{a,b}$	$M_{a,c}$	...
b	$M_{b,a}$	GeoSim(1,2)	$M_{b,c}$	...
c	$M_{c,a}$	$M_{c,b}$	GeoSim(1,[7,11])	...
⋮	⋮	⋮	⋮	⋱

Figure 3: Example of MLG construction. We use blue edges to indicate adjacent nodes in the same layer. The red dotted lines indicate cross-layer edges. Note that merged nodes may not reside in the next layer, but a layer that satisfies the volume constraint. We use curved yellow lines to indicate the initial matching. The affinity matrix M is computed by Equation 3.

construct the nodes of MLG($\mathcal{S}$) recursively in a bottom-up manner:

$$\bar{N}_k^{[l]} = \begin{cases} S_k & \text{if } l = 1 \\ \bar{N}_i^{[m]} \cup \bar{N}_j^{[m]} & \text{if } \bar{N}_i^{[m]} \cap \bar{N}_j^{[m]} \neq \emptyset,\ i \neq j,\ \mathcal{C}_{vol}^{l-1} \leq VOL(\bar{N}_k^{[l]}) < \mathcal{C}_{vol}^{l},\ \ m < l \end{cases} \tag{1}$$

In this way every input segment S_i is assigned a node $\bar{N}_i^{[1]} = S_i$ and are grouped to form the bottom layer ($l = 1$). Higher-layer nodes are created by merging all vertices and edges in lower-layer nodes (in the same layer) only if they are adjacent. Two nodes are adjacent if they share some vertices $\subset V$, edges $\subset E$ in $\mathcal{S}$ such that $\bar{N}_i^{[l]} \cap \bar{N}_j^{[l]} \neq \emptyset$. Simply merging adjacent nodes would lead to exponential growth in number of merged nodes. We thus define a volume constraint $\mathcal{C}_{vol}^{l-1} < VOL(\bar{N}_k^{[l]}) < \mathcal{C}_{vol}^{l}$ to restrict the volume of a node in each layer. We define the upper bound $C_{vol}^{l} = \frac{l}{L} VOL(S)$ for each layer l, where L is the maximum number (a user defined parameter) of layers in MLG($\mathcal{S}$) and VOL($\mathcal{S}$) is the total volume of shape $\mathcal{S}$.

Edge Construction. Next, we define the edges of MLG($\mathcal{S}$). For every pair of nodes $\bar{N}_i^{[l]}, \bar{N}_j^{[l]}$ in the same layer l with shared vertices/edges (i.e $\bar{N}_i^{[l]} \cap \bar{N}_j^{[l]} \neq \emptyset$), a within-layer or "adjacency" edge $(\bar{N}_i^{[l]}, \bar{N}_j^{[l]})$ is established between them. Let $\bar{N}_k^{[l]} = \bar{N}_i^{[m]} \cup \bar{N}_j^{[m]}$ be an internal node which is merged from two nodes $\bar{N}_i^{[m]}$ and $\bar{N}_j^{[m]}$, where $m < l$. We establish two cross-layer or "part-of" edges $(\bar{N}_i^{[m]}, \bar{N}_k^{[l]})$ and $(\bar{N}_j^{[m]}, \bar{N}_k^{[l]})$ between them. That is, the edge $e \in E_{MLG(\mathcal{S})}$, the edge set of MLG($\mathcal{S}$), is defined as:

$$e = \begin{cases} (\bar{N}_i^{[m]}, \bar{N}_j^{[l]}) & \text{if } m = l,\ i \neq j,\ \bar{N}_i^{[m]} \cap \bar{N}_j^{[l]} \neq \emptyset \\ (\bar{N}_f^{[m]}, \bar{N}_k^{[l]}) & \text{if } m < l,\ i \neq j,\ f \in \{i, j\},\ \text{s.t. } \bar{N}_k^{[l]} = \bar{N}_i^{[m]} \cup \bar{N}_j^{[m]} \end{cases} \tag{2}$$

We have tried different weights for within-layer and cross-layer edges, and found empirically that setting all edge weights to 1 can produce good results. We therefore use this for all subsequent experiments due to simplicity. An example of the construction of nodes and edges in MLG is shown in Figure 3.

4.2. Initial Matching

Next, we compute the geometric similarity score and generate initial correspondences. We have tried several techniques and found that LFD Chen et al. (2003) similarity scores perform well (even for our non-rigid experiments as individual segments are relatively small and close to rigid). We will use LFD similarity throughout this paper. We pre-compute MLG($\mathcal{S}$) and MLG($\mathcal{T}$) for two input shapes $\mathcal{S}$ and $\mathcal{T}$. For each node $\bar{N}_i^{[u]}$ in MLG($\mathcal{S}$) we pre-compute the K best matching (in terms of LFD similarity scores) of node $\tilde{N}_j^{[v]}$ in MLG($\mathcal{T}$), as its initial matching (shown as the yellow lines in Figure 3).

5. Diffusion Pruning with Anchors

Once the initial matching has been pre-computed, we adapt and apply diffusion pruning to obtain consistent matching results. We equip our technique with two pruning stages (Figure 2). The first stage considers input matching between nodes in the bottom layers of the two MLGs only (i.e. correspondences between nodes $\bar{N}_i^{[1]}$ and $\tilde{N}_j^{[1]}$). We treat these first-stage matching results, with high confident scores as anchors. In the second stage, we consider higher layers matching (i.e. correspondences between nodes $\bar{N}_i^{[u]}$ and $\tilde{N}_j^{[v]}$) in the MLG hierarchy. There are often a large number of nodes in the MLG. The first-stage anchors offer good constraints to the second-stage matching results.

One of the matching problems with inconsistent input segmentation is that the underlying connectivity graph often shows non-isometric inconsistency in term of topological distances. The diffusion pruning technique Tam et al. (2014b) has been shown useful to obtain good point-wise correspondences under moderate non-isometric shape deformation. We thus adapt it to our use for segment-graph hierarchical matching. Given some initial correspondences, we construct an affinity matrix to encode both geometry similarity and topological consistency of initial matching. We then adapt the diffusion framework to generate confidence scores. Based on the scores, inconsistent correspondences are pruned in a greedy manner. We would refer readers to Tam et al. (2014b) for the mathematical and implementation details. Here, we focus on the adaptation for our segment-wise matching task.

5.1. Affinity Matrix Computation

Given some segment-wise correspondences C, we build an affinity matrix M of size $|C| \times |C|$. M encodes both topological (MLG distance) and geometry (LFD) information. As shown in Figure 3 each element in $M(a,b)$ indicates the compatibility of two segment-wise correspondences $a = (\bar{N}_i^{[u]}, \tilde{N}_j^{[v]})$ and $b = (\bar{N}_x^{[n]}, \tilde{N}_y^{[m]})$ $(a, b \in C)$.

Using local isometry to infer global consistency is a key concept in diffusion pruning Tam et al. (2014b). For a pair of nodes $\bar{N}_i^{[u]}$ and $\bar{N}_x^{[n]}$ in the same MLG, we define the MLG distance $d(\bar{N}_i^{[u]}, \bar{N}_x^{[n]})$ as the number of edges in the shortest path between them. The distance models the topological (both adjacent and part-of) relationship between segments within the MLG hierarchy. A local topological MLG region can be further defined around a node $\bar{N}_i^{[u]} \in MLG(\mathcal{S})$ (similarly for nodes $\tilde{N}_j^{[v]} \in MLG(\mathcal{T})$) in the MLG hierarchy as $R^{\delta}_{\bar{N}_i^{[u]}} = \{x | d(\bar{N}_i^{[u]}, x) \leq \delta D\}$ where $\delta \in [0, 1]$ is a user defined threshold and D is the largest MLG distance in an MLG. Given this, we can compute the element of matrix M. Let $m_{a,b}$ be the distance compatibility for two segment-wise correspondences $a, b \in C$. We follow the normalization procedure in Huang et al. (2008); Tam et al. (2014b) to obtain $M_{a,b}$ as follows:

$$M_{a,b} = \begin{cases} \frac{m_{a,b} - c_0}{1 - c_0}, & a \neq b, \;\; m_{a,b} \geq c_0, \;\; 0 \leq c_0 \leq 1 \\ GeoSim(\bar{N}_i^{[u]}, \tilde{N}_j^{[v]}), & \text{otherwise}, \end{cases}$$

$$m_{a,b} = \min\left(\frac{d(\bar{N}_i^{[u]}, \bar{N}_x^{[n]})}{d(\tilde{N}_j^{[v]}, \tilde{N}_y^{[m]})}, \frac{d(\tilde{N}_j^{[v]}, \tilde{N}_y^{[m]})}{d(\bar{N}_i^{[u]}, \bar{N}_x^{[n]})} \right),$$

$$\bar{N}_x^{[n]} \in R^{\delta}_{\bar{N}_i^{[u]}} \;\; \text{and} \;\; \tilde{N}_y^{[m]} \in R^{\delta}_{\tilde{N}_j^{[v]}}$$

$$GeoSim(\bar{N}_i^{[u]}, \tilde{N}_j^{[v]}) = |\mathrm{LFD}(\bar{N}_i^{[u]}) - \mathrm{LFD}(\tilde{N}_j^{[v]})| \tag{3}$$

$M_{a,b}$ will take into account only segment-wise correspondences a and b with end-point nodes fall into respective local topological MLG regions $R^{\delta}_{\bar{N}_i^{[u]}}$ and $R^{\delta}_{\tilde{N}_j^{[v]}}$ Tam et al. (2014b). It further ensures $c_0 \leq m_{a,b} \leq 1$, i.e., $m_{a,b}$ be at least c_0 isometrically consistent Huang et al. (2008), and sparsifies M if it does not. Different from Tam et al. (2014b), we further encode geometric similarity in the diagonal entries $M_{a,a}$ where $GeoSim(\bar{N}_i^{[u]}, \tilde{N}_j^{[v]})$ is the dissimilarity score of their LFD features.

5.2. Diffusion Framework and Pruning

Matrix M encodes both local geometric similarity and local topological isometric consistency information. The matrix is then normalized to a Markov probability matrix to model the Markov random walk for diffusion analysis. After this, the stationary distribution π is computed as the confidence score $\pi(a)$ for a correspondence a. The normalisation step is essential to infer the global consistency from local topological isometric compatibility in MLGs. This framework is supported by the spectral graph theory Tam et al. (2014b). (Due to the page limits, readers are referred to Section 4.3 in Tam et al. (2014b) for detailed explanation.) We sort all initial matchings with descending confidence scores and examine each of them in a greedy manner Leordeanu and Hebert (2005); Huang et al. (2008); Tam et al. (2014b).

In our algorithm, we apply diffusion pruning twice. In the first run, we only use bottom layers to obtain good correspondences (anchors). In the second run, we involve more layers in the two MLG($\mathcal{S}$) and MLG($\mathcal{T}$). During the second pruning stage, we first accept anchors into result correspondences, and then greedily add new consistent correspondences from higher layers. The idea is supported by two observations. First, SHED Kleiman et al. (2015) produces reasonable results if the input contains perfect segments or there are some segments with high distinctive geometric scores. Our technique is similar to SHED that uses spectral analysis and shows similar behavior. Second, given imperfect input segmentation, our technique can better handle moderate non-isometric differences because of diffusion pruning. It can often find good and consistent matching based on local regions using just bottom layer. Given these good anchors, we can further constrain consistent outputs in the higher layers.

6. Voting and Final Output

Most of the results obtained in the previous step are useful. Still, some incorrect matching may still be present due to the greedy pruning procedure. There are two further reasons. First, our simple topological distance incorporates both adjacency and part-of relationships as one measure and does not differentiate the two relationships. Second, nodes in higher layers often have similar shorter MLG distances, which easily lead to ambiguous matching. In our final step, we would like to further confirm that the pruned segment-wise correspondences are consistent throughout the MLG hierarchy. For example, a consistent segment-wise correspondence should appear as "part of" some merged-to-merged segment-wise correspondences in a higher layer. To confirm lower-layer correspondences using higher-layer ones, we develop a voting-prune procedure which is discussed below.

Let C_{dp} be a set of segment-wise correspondences (e.g. Figure 4) obtained from our adapted diffusion pruning step (Section 5.2). We first go through each correspondence $a = (\bar{N}_i, \tilde{N}_j) \in C_{dp}$ and check against another correspondence $b = (\bar{N}_x, \tilde{N}_y) \in C_{dp}$ where $a \neq b$. If both $\bar{N}_i \subset \bar{N}_x$ and $\tilde{N}_j \subset \tilde{N}_y$, we increment a vote Vote(a) for a. A correspondence a from lower layers which are consistent with higher layer correspondences will accumulate more votes. Next, we sort all $a \in C_{dp}$ in descending order of Vote(a) and use higher confidence score $\pi(a)$ from DP to break the tie if possible. Figure 4 shows example values of Vote(a) and $\pi(a)$ of each correspondence at the top left and right corners of each subfigure respectively.

Our greedy hierarchical pruning step is then carried out using the sorted list. We first accept the first $a \in C_{dp}$ with the highest Vote(a) into the C_{vote}, and remove a from C_{dp}. For each subsequent $b = (\bar{N}_x, \tilde{N}_y) \in C_{dp}$, we check $\forall a = (\bar{N}_i, \tilde{N}_j) \in C_{vote}$ if b satisfies either:

$$\bar{N}_i \subset \bar{N}_x \text{ and } \tilde{N}_j \subset \tilde{N}_y \qquad \text{or} \qquad \bar{N}_i \not\subset \bar{N}_x \text{ and } \tilde{N}_j \not\subset \tilde{N}_y$$

Algorithm 1: Voting Algorithm

```
Input: $C_{dp}$
Output: $C_{vote}$
procedure VOTING($C_{dp}$)
    for each $a = (\tilde{N}_i, \tilde{N}_j) \in C_{dp}$
        Vote($a$) $\leftarrow 0$
        for each $b = (\tilde{N}_x, \tilde{N}_y) \in C_{dp} \setminus a$
            if $\tilde{N}_i \subset \tilde{N}_x$ and $\tilde{N}_i \subset \tilde{N}_y$ then
                Vote($a$) $\leftarrow$ Vote($a$) + 1
            end if
        end for
    end for
    $C_{vote} \leftarrow \emptyset$
    while $C_{dp} \neq \emptyset$ do
        $b = (\tilde{N}_x, \tilde{N}_y) \leftarrow \mathrm{argmax}_{a \in C_{dp}}$ Vote($a$)
        if $\forall a = (\tilde{N}_i, \tilde{N}_j) \in C_{vote}$
            $\tilde{N}_i \subset \tilde{N}_x$ and $\tilde{N}_j \subset \tilde{N}_y$ $\vee$
            $\tilde{N}_i \not\subset \tilde{N}_x$ and $\tilde{N}_i \not\subset \tilde{N}_y$ then
            $C_{vote} \leftarrow C_{vote} \cup b$
        end if
        $C_{dp} \leftarrow C_{dp} \setminus b$
    end while
    return $C_{vote}$
end procedure
```

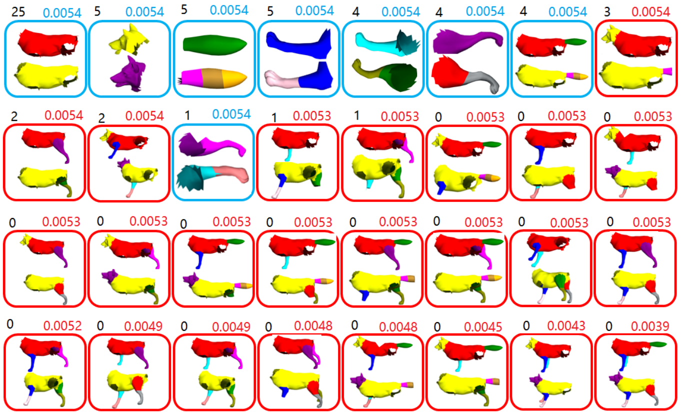
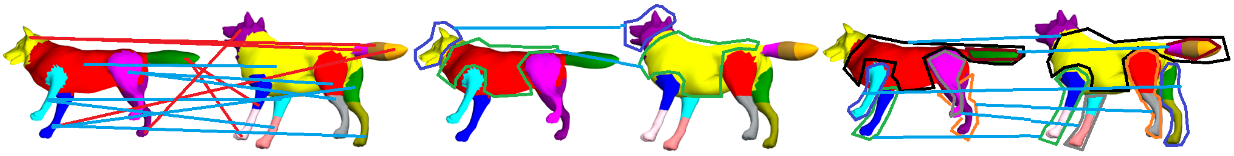


Figure 4: Voting of pruned results. The red ones are removed by voting. Top left numbers are votes and top right numbers are diffusion pruning confident scores.



(a) SHED Matching Result (b) Our Bottom Layer Matching Result (c) Our Higher Layers Matching Result

Figure 5: Refined matching by voting mechanism. All diffusion pruning results have been visualized in Figure 4. There are incorrect matchings, for example, at the top right corner *head-body* is matched to *body-tail*. After voting, these incorrect matchings are pruned (b-c). As a comparison, the SHED result is shown in (a).

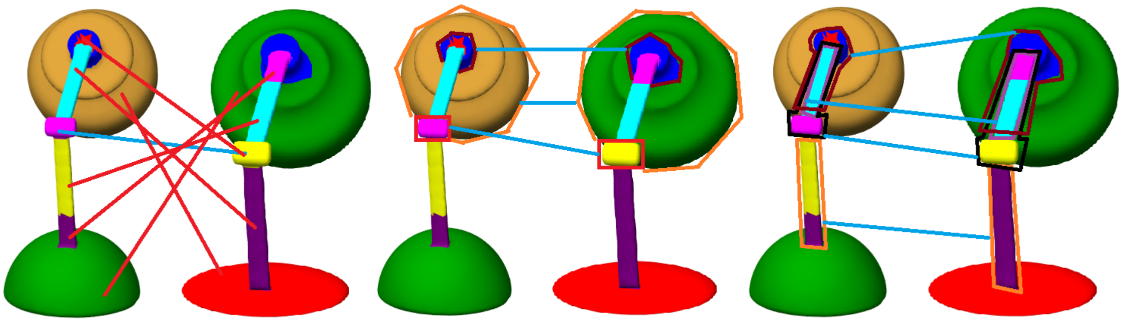
This step requires that the new segment-wise correspondence b is consistent with all accepted $a \in C_{vote}$ or b is not seen before. We then move b from C_{dp} to C_{vote}. If b violates both constraints, it means that b is an inconsistent correspondence. We simply prune it from C_{dp}. Matchings highlighted in blue round boxes in Figure 4 are all accepted correspondences C_{vote}. Matchings highlighted in red are inconsistent correspondences that are pruned. In our implementation, we further use C_{vote} as anchors for DP (which sometimes improves the greedy results), and run the voting-prune step again to obtain C'_{vote} as the final output (Figure 5(b)-(c)).

This voting step, together with diffusion pruning (Section 5.2), ensures that the accepted correspondences are topologically and hierarchically consistent within MLGs, and their endpoint nodes are geometrically similar. The algorithm is detailed in Algorithm 1.

7. Evaluation

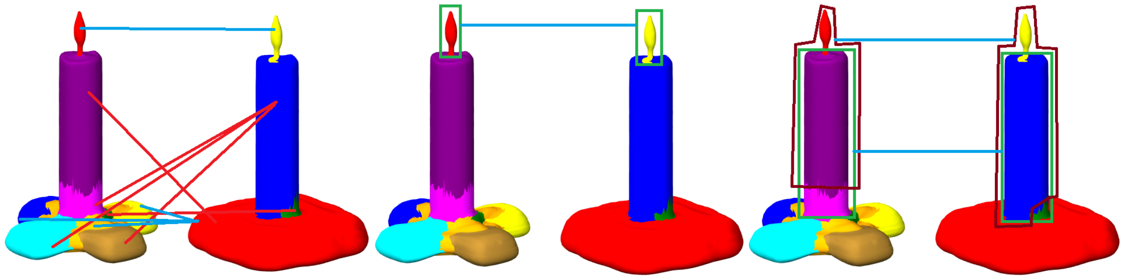
We evaluate our method on both rigid (man-made) and non-rigid shapes. The rigid data set is downloaded from the SHED's project page, which consists of four subsets, namely vases, airplanes, lamps and candles. All rigid shapes are segmented by a weakly-convex segmentation technique as mentioned in Kleiman et al. (2015). Our non-rigid set consists of wolf, human, horse, and centaur. We use the consistent segmentation results from Huang et al. (2011) and further manually over-segment those shapes to provide initial inconsistent segmentations for our evaluation. With these inputs, we use SHED Kleiman et al. (2015) and our proposed technique to compute segment-wise correspondences, and evaluate both techniques qualitatively (visual examples) and quantitatively (precision).

To our knowledge, there is no existing ground-truth dataset for segment-wise matching. For high-level matching, there is a certain degree of human subjectivity involved. For example in Figure 1b, the purple joint on the left lamp has only one segment, but there would be many possible correct matching segments (e.g. all or one of the unmatched segments) on the right lamp. Even a no matching as shown in Figure 1b-1c can



(a) SHED Matching Result (b) Our Bottom Layer Matching Result (c) Our Higher Layer Matching Result

Figure 6: Near-consistent segmentation matching result. Our method outputs meaningful matching at upper and lower sticks. There is a large variation between the two bases and our method does not match them.



(a) SHED Matching Result (b) Our Bottom Layer Matching Result (c) Our Higher Layer Matching Result

Figure 7: Comparison results of candles with inconsistent segmentation. As shown in green polygons our method can match body segments in a meaningful way.

be a correct choice. To provide a fair evaluation, we recruited three volunteers (one sculptor, two musicians) from non-computer science background to carry out the annotations. We informed all volunteers that their annotations should be based on their own intuition of meaningful/reasonable correspondences with respect to the shape and segments. In this way each correspondence produced by Kleiman et al. (2015) and our technique is given a correct or wrong label. We use a majority vote in cases where there is a discrepancy. These are used to compute the precision and to indicate correct or incorrect matchings in all figures. For all visualized figures, segment colors are only used to show distinct boundaries of segments, rather than matching correctness. Blue (red) lines indicate correct (incorrect) segment-wise correspondences. We further use colored polygonal lines to indicate our one-to-merged / merged-to-merged segment-wise correspondence results.

7.1. Rigid Shapes

We have tried HKS Sun et al. (2009) and persistent HKS Dey et al. (2010) but they cannot produce distinct similarity scores for MLG nodes. Similarly, PCA Blomley et al. (2014) and D1/D2 Osada et al. (2002) distributions occasionally produce incorrect scores. In this paper we use LFD Chen et al. (2003) to generate geometry similarity scores for segments, since it performs well in our experiments.

In this section we evaluate our method on rigid shapes. We first test our method on shapes with near consistent input segmentation, and then with inconsistent input segmentation. Our evaluation mainly focuses on inconsistent segmentation which is the focus of this paper.

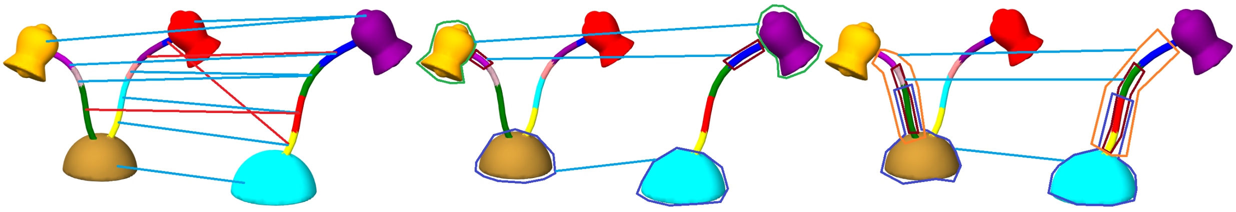
7.1.1. Near-consistent Input Segmentation

In Figure 6 two lamps have similar input segmentation except some small over-segmented pieces in the stand and cap joint. Figure 6a shows that SHED mismatches the base of the left lamp to the right lamp's cap.

The mismatch is caused by a good geometry similarity score due to D1/D2/volume computation between the base (left lamp) and cap (right lamp). Further, both segments are located at the endpoints of their respective component graphs with similar topological distances to the rest of the nodes. As both geometric and topological information is very similar, SHED outputs an upside down matching. The volunteers consider the result as a mismatch. Our technique shows reasonable correspondences, with many one-to-merged segment-wise correspondences (Figures 6b-6c). For example in the left lamp, the small red piece above the cap joint, and the small purple piece in the lower stand are merged with respective larger piece in the matching results. Further, our technique is able to solve the upside down ambiguity because the one-to-merged segment-wise correspondences offer better geometric, topological and hierarchical consistency. The base is not matched because their geometry (LFD features) differs a lot.

In Figure 7, we show another example where the upper candlestick is near-consistently segmented, but the lower base is highly over-segmented. It is a challenging case because the base contains six inter-connected segments making the component graph very complex. Though SHED is able to obtain a one-to-many base-to-base matching, it also badly mismatches both candlesticks to the bases. Our technique is able to discover candlestick matching in a reasonable manner without any incorrect matching. It does not discover the one-to-merged base matching because it requires merging of all six segments to form the base which is beyond the number of layers we consider for the example (see Section 7.3).

7.1.2. Inconsistent Input Segmentation



(a) SHED Matching Results (b) Our Bottom Layer Matching Results (c) Our Higher Layers Matching Results

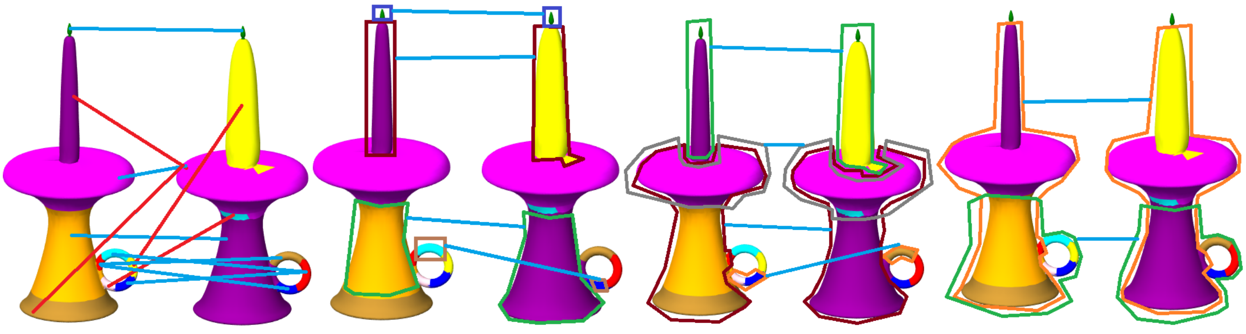
Figure 8: Lamp matching results with large topological variation. Our method can find consistent matching with no mismatched correspondences.

Matching with large difference in number of nodes

Figure 8 shows an example with large topological variation. The number of segments in the left lamp is almost two times more than that of right lamp. SHED's one-to-many results are mostly good, but mismatches still appear. For example, the lower stick in the left lamp is adjacent to the base, but it is mismatched to a node in the right lamp which is not adjacent to the base. Our volunteers consider the matching incorrect. Our technique considers merged nodes in higher layers. It finds consistent matching on the left branch of the left lamp. Caps and bases are matched with two one-to-one correspondences, whilst the main stick is matched with a merged-to-merged correspondence. In this way we match all stick segments consistently, and avoid incorrect matching. Our technique does not offer one-to-many matching and thus no matching is obtained for the right stick (which is plausible). Our method may be extended to produce matching to the right lamp by first removing matched nodes and re-applying our technique (as demonstrated in Tam et al. (2014a) for discovering point-wise correspondences of multiple parts).

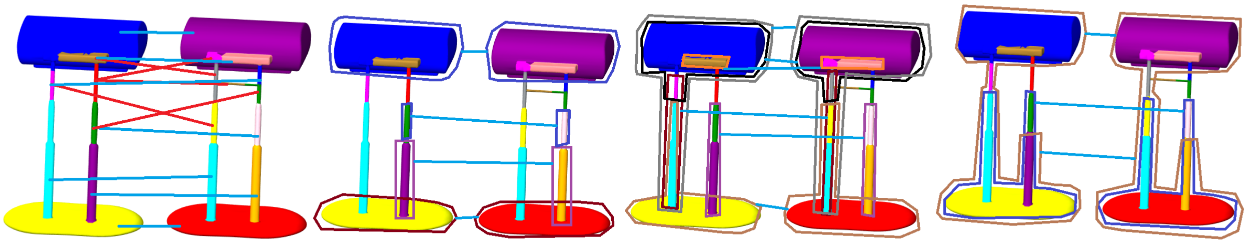
Matching with inconsistent input segments and loops

Figure 9 shows another challenging candle example with inconsistent over-segments and loops. SHED matches many segments incorrectly. These incorrect matchings are largely influenced by the topologically-adjacent correct matchings. However, by using geometric and topological information alone, it is not sufficient to find good matching. Our technique discovers many reasonable matchings with merged nodes in higher layers which are consistent with human intuition. The loop handle is very challenging as it consists of many small pieces. Note that both SHED and our technique cannot resolve symmetry issue. Therefore, both SHED and our technique have some matchings that are controversial. For example, SHED returns many one-to-many matchings in the loop handle (Figure 9a). Our technique obtains a matching from the



(a) SHED Matching Result (b) Our Bottom Layer Result (c) Our Higher Layer Result (d) Our Higher Layers Result

Figure 9: Matching results of candles with inconsistent segmentation. For challenging segments our method matches them in higher layers to avoid incorrect correspondences. We show all higher layers results in two sub-figures (c) and (d) for clarity purpose.



(a) SHED Matching Result (b) Our Bottom Layer Result (c) Our Middle Layers Result (d) Our Higher Layers Result

Figure 10: Matching result comparison for shapes with large topological variation and loops. We show all higher layers results in two sub-figures (c) and (d) for clarity purpose.

lower piece of the loop to the upper piece of the loop handle (in Figure 9b, and similarly upper piece to lower piece matching in the loop in Figure 9c). Our volunteers independently consider them (both SHED's and ours results) correct because they are part of the handle (due to functionality). Having said that, our technique discovers the loop pieces in a upside down, but consistent manner.

Matching with multiple loop structures

Next, we focus on a more challenging example. Figure 10 shows the matching between two lamps with highly inconsistent input segmentation. In particular, the crossbeam and T-shaped segment (adjacent to the crossbeam) exist only in the right lamp. SHED tries to find one-to-many matchings for all segments. Though it can find some good matchings, it also returns many incorrect ones (Figure 10a). Note that in the left stand (left lamp), the upper segment is inconsistently matched to the left and right stand (right lamp). The results can be explained by the segment graphs in Figure 11 as both segment graphs contain cycles. The crossbeam acts as a shortcut edge and creates another shorter cycle. This shorter path significantly distorts the topological distance on the segment graphs, leading to the inconsistent matchings in SHED.

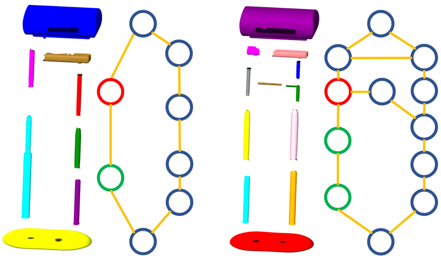
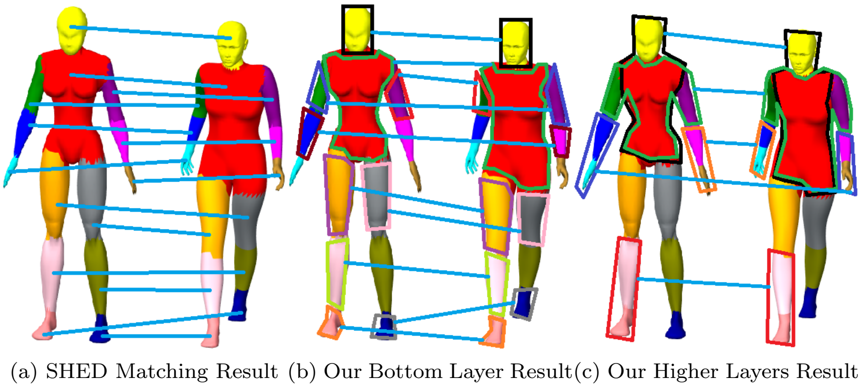


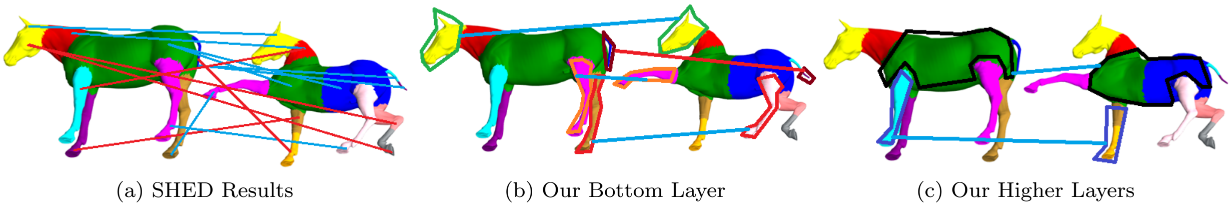
Figure 11: Segment graphs of two lamps.

Figure 10b shows that our technique obtains more reliable one-to-one matchings in the right stand. For the left stand, nodes are merged in the higher layers in the MLG graph (green circles in Figure 11). One-to-merged and merged-to-merged segment-wise matchings are resulted (see also the brown, blue and purple polygons in Figure 10c-10d). Since our technique looks for geometrically, topologically and hierarchically consistent matching, the crossbeam is not matched. The volunteers find our result reasonable.



(a) SHED Matching Result (b) Our Bottom Layer Result (c) Our Higher Layers Result

Figure 12: Non-rigid matching comparison with consistent segmentation.



(a) SHED Results (b) Our Bottom Layer (c) Our Higher Layers

Figure 13: Non-rigid matching comparison with inconsistent segmentation.

7.2. Non-Rigid Shapes

In the literature, some segment-wise matching techniques do not support non-rigid shapes (e.g. Alhashim et al. (2015); Zhu et al. (2017)). We further evaluate if our technique can support them. We have tried some geometric features designed for non-rigid shapes (e.g. HKS Sun et al. (2009), Persistent HKS Dey et al. (2010)) but they do not provide distinctive geometric measures. Therefore, we use LFD in these experiments. Both SHED and our technique are built on top of component/segment graphs, and are not designed to handle symmetry issue — both cannot differentiate left or right. In non-rigid shapes symmetry is common. We thus consider matching say, left arm to right arm (or vice versa) as correct, as long as the whole arm (every segments in the arm) is consistently matched. Such symmetry issues could be addressed by incorporating a symmetry detection technique to resolve ambiguities.

Figure 12 demonstrates one human example with consistent input segmentation. In our technique, the non-rigidly deformed hands are not matched due to no initial correspondences (low LFD scores). LFD is defined mostly for rigid shapes only. In our result, hand and arm merged-to-merged matching can be obtained in higher layers because initial correspondences are available (merging hand and arm offer good LFD scores). We do not obtain matching for lower legs because of the volume constraint defined in the MLG (see section 4) where the leg (for the left human) is moved into higher layer for one of the shapes. It can be easily solved by relaxing the topological consistency thresholds c_0. We argue that our technique still performs reasonably well in this example despite of the LFD issue.

Figure 13 shows a horse example with inconsistent input segmentation. Our technique is able to obtain accurate matching under inconsistent input segmentation in legs and body. Note that under symmetry, front legs to back legs matching in both SHED and our techniques are considered correct. In Figure 13b our method outputs 1 incorrect result between tails. This is caused by highly similar LFD scores. However, SHED often mismatches leg to tail or head. The volunteers consider them incorrect.

7.3. Quantitative Evaluation

We further evaluate our technique on large rigid and non-rigid datasets. Our method outputs matching of higher layers. There is no ground truth dataset, so volunteers have to manually examine each output

Rigid	MLG	SHED	pairs	layers
lamps	85.2%	76.7%	30	8
vases	86.0%	63.0%	20	4
candles	86.2%	71.3%	11	4
planes	83.5%	60.8%	11	8
average	85.3%(1.2)	69.6%(7.4)		

Table 1: Precision (std. dev.) on rigid (man-made) set.

Non-Rigid	MLG	SHED	pairs	layers
wolf	97.2%	59.0%	3	8
human	83.3%	62.7%	7	8
horse	85.3%	81.6%	6	8
centaur	90.9%	67.5%	4	8
average	87.5%(6.2)	68.8%(9.9)		

Table 2: Precision (std. dev.) on non-rigid set.

matching to compute precision rate — it is a time consuming process. It is also not possible for us to enumerate all higher layer matching. For example, a shape with 14 segments can lead to 500+ internal nodes in the MLG depending on their topology. It is simply too laborious and time-consuming to annotate all of them. Therefore we do not evaluate on recall rate. Following Kim et al. (2011) we randomly select pairs of shapes from each set and annotate the output. The whole annotation process takes several weeks to finish among all three unpaid volunteers. In our experiments, we use fixed parameters for all pairs in a set (similar to Kleiman et al. (2015)).

For the rigid set, we use the following parameters for the adapted diffusion pruning to compute anchors: local distance $\delta_1 = 0.2$ and LFD threshold is 0.8. The second run of diffusion pruning uses $\delta_2 = 0.8$ and LFD threshold $= 0.8$. For both runs, the number of initial matching for each node K is set to 7; the threshold in diffusion pruning is set as default $c_0 = 0.7$ (Tam et al. (2014b)). For the non-rigid set, the values of $\delta_1 = 0.8$ and $\delta_2 = 0.2$ and other parameters stay the same as the rigid set.

The only parameter we adjust is the number of layers in MLG construction. We use eight layers in lamp and plane sets, and four layers for vase and candle sets. The reason is that there are too many internal nodes in the constructed MLGs with eight layers. Reducing the number of layers to four still provides reasonable results. All shapes in non-rigid sets have eight layers.

All quantitative results are shown in Tables 1 and 2, and are based on 72 pairs of rigid shapes and 20 pairs of non-rigid shapes. Our method outperforms SHED in all cases. For both rigid (man-made) and non-rigid sets, our technique outperforms SHED in precision with lower standard deviation. The lower standard deviation further shows the stability and robustness of our technique.

Our annotation focuses on the outputs of the two techniques. We plan to release the annotation results and codes to the research community, for inspection, comparison and downstream applications.

8. Discussion

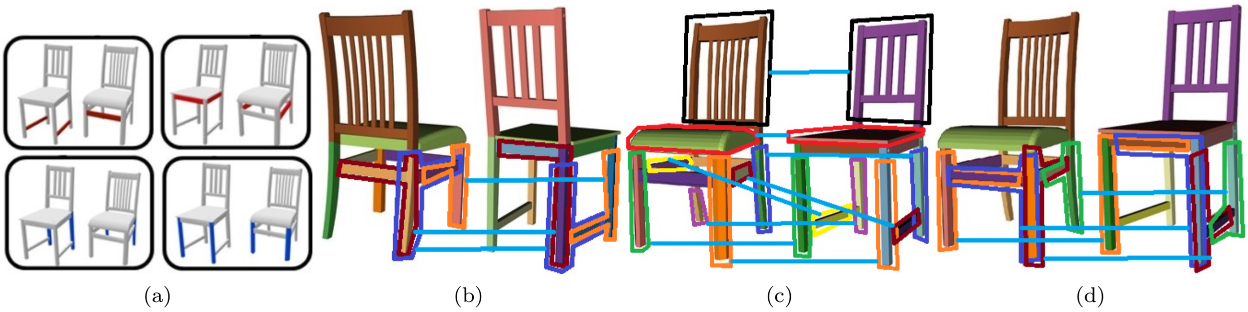
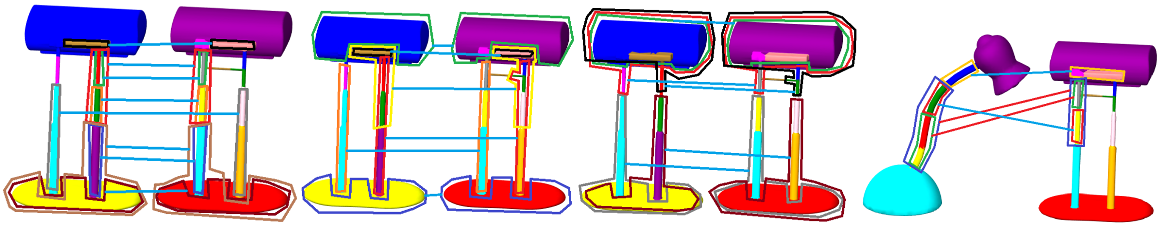


Figure 14: (a) image courtesy of Zhu et al. (2017). (b)(c)(d) are our method matching results.

Here, we further provide a brief comparison of our technique with the state-of-the-art Zhu et al. (2017). Figure 14a shows the matching result of two chairs (image courtesy of Zhu et al. (2017)). In the figure, the red side panels are mismatched to the front panels between chairs. The technique proposed in Zhu et al. (2017) is a top-down matching technique assuming perfect input segmentations. The technique seeks the best split along the component tree. To our knowledge, it does not use higher layer matching to support lower layer matching which would have solved the mismatch.

We tried our technique on the same set of chairs by manually labeling the left (right) chair into 10 (12) segments, according to the initial segmentation as shown in Figure 14a. We then apply our technique using



(a) Results of changed δ (b) Results of changed c_2 (c) Large difference in shapes

Figure 15: Results from adjusting different parameters, compared to Figure 10. The $\delta 1$ used in (a) is 0.5 (anchor stage) and 0.7 (final stage) - note the symmetric issue, (b) uses the same δ as (a) and further reduce threshold c_2 to 0.2. (c_2 is a threshold used in diffusion pruning for greedy pruning.) Here, relaxing c_2 leads to only higher layer matchings. (c) shows our method performs poorly when the inputs have large topological and geometrical difference.

our two-stage diffusion pruning (DP) with a 2-layer MLG for each chair. We only use 2 layers because the chair is highly complex with high connectivity for each segment. If we use 3 or more layers, the number of internal nodes grows to 1000+ which is too slow to compute. Due to the lack of high-layer nodes, we cannot apply our voting step. However, simply using the proposed two-stage DP step yields perfect matching result (Figures 14b-14d). This answers our research question that considering merged nodes in the MLG hierarchy can improve matching results. As the source code and data for Zhu et al. (2017) are not available, further comparison is not possible. Having said that Zhu et al. (2017) cannot support non-rigid shapes, and assume consistent input segmentations. Our technique is comparatively more flexible. It can handle non-rigid shapes and inconsistent input segmentations.

There are limitations in our technique however. One issue is the sensitivity to the chosen parameters. Figure 15 compares the results in Figure 10 with different parameters. In Figure 15a, we tighten the δ_1 threshold (i.e., use smaller local isometric disk). Though the volunteers consider the results correct, it leads to more local matching and cannot avoid the symmetry issue. In Figure 15b we further reduce c_2 (a threshold used in Tam et al. (2014b) for the last greedy pruning step), the matching results all shift to higher layers, with no bottom-layer one-to-one correspondences found. Figure 15c further shows that our technique does not perform well when the input shapes have large difference in topology and/or geometry.

Our current unoptimised code is too slow to handle shapes with a large number of input segments. There is an exponential growth in the number of possible internal nodes in MLG, with respect to the number of input segments. We constrain the MLG using volume, but it can sometimes miss some matchings (e.g. the leg in Figure 12). In the future, we hope to develop a more robust hierarchical representation than MLG to reduce the search space. Another direction is to incorporate our bottom-up idea into a top-down approach Zhu et al. (2017). Further, our technique consists of quite a few parameters. Although most of them are fixed to default settings, we plan to develop a more robust technique and make it more generic to a large variety of input shapes and inconsistent segmentations.

In the future, we are going to use better geometrical features to enhance our matching techniques. We also need to condense the size of MLG, which means a better merging technique is necessary. Based on the simplified MLG we can further investigate the convergence property of our technique.

9. Conclusion

In this paper, we propose a novel segment-wise matching technique that can handle shapes with inconsistent (over-/imperfect) input segmentation. Our idea is to greedily optimize matchings that are geometrically, topologically and hierarchically consistent. To do so, we develop a multi-layer graph (MLG) representation to store the possible merging arrangement of segments. Apart from geometric and topological consistency, we explicitly seek consistency in the hierarchical segment merging space. Experimental results demonstrate the effectiveness of our technique when compared to two state-of-the-art methods.

Acknowledgements

We would like to thank the authors of Kleiman et al. (2015) for making their codes publicly available.

References

Alhashim, I., Xu, K., Zhuang, Y., Cao, J., Simari, P., Zhang, H., 2015. Deformation-driven topology-varying 3D shape correspondence. ACM Trans. Graph. 34, 236:1–236:13.

Ankerst, M., Kastenmüller, G., Kriegel, H.P., Seidl, T., 1999. 3D shape histograms for similarity search and classification in spatial databases, in: Proceedings of the 6th International Symposium on Advances in Spatial Databases, Springer-Verlag. pp. 207–226.

Berg, A.C., Berg, T.L., Malik, J., 2005. Shape matching and object recognition using low distortion correspondences, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, IEEE Computer Society. pp. 26–33.

Blomley, R., Weinmann, M., Leitloff, J., Jutzi, B., 2014. Shape distribution features for point cloud analysis &ndash; a geometric histogram approach on multiple scales. ISPRS annals II-3, 916. ISPRS Technical Commission III Symposium, Zürich, CH, September 5-7, 2014.

Chaudhuri, S., Kalogerakis, E., Guibas, L., Koltun, V., 2011. Probabilistic reasoning for assembly-based 3D modeling. ACM Trans. Graph. 30, 35:1–35:10.

Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M., 2003. On Visual Similarity Based 3D Model Retrieval. Computer Graphics Forum .

Dey, T., Li, K., Luo, C., Ranjan, P., Safa, I., Wang, Y., 2010. Persistent heat signature for poseoblivious matching of incomplete models, in: Computer Graphics Forum, pp. 1545 – 1554.

Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H., 2005. Robust global registration, in: Proceedings of the Third Eurographics Symposium on Geometry Processing, Eurographics Association.

Huang, Q., Koltun, V., Guibas, L., 2011. Joint shape segmentation with linear programming. ACM Trans. Graph. 30, 125:1–125:12.

Huang, Q.X., Adams, B., Wicke, M., Guibas, L.J., 2008. Non-rigid registration under isometric deformations, in: Proceedings of the Symposium on Geometry Processing, Eurographics Association. pp. 1449–1457.

van Kaick, O., Xu, K., Zhang, H., Wang, Y., Sun, S., Shamir, A., Cohen-Or, D., 2013a. Co-hierarchical analysis of shape structures. ACM Trans. Graph. 32, 69:1–69:10.

van Kaick, O., Zhang, H., Hamarneh, G., 2013b. Bilateral maps for partial matching. Comput. Graph. Forum 32, 189–200.

van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. A survey on shape correspondence 30, 1681–1707.

Kalogerakis, E., Chaudhuri, S., Koller, D., Koltun, V., 2012. A probabilistic model for component-based shape synthesis. ACM Trans. Graph. 31, 55:1–55:11.

Kim, V.G., Lipman, Y., Funkhouser, T., 2011. Blended intrinsic maps. ACM Trans. Graph. 30, 79:1–79:12.

Kleiman, Y., van Kaick, O., Sorkine-Hornung, O., Cohen-Or, D., 2015. SHED: Shape edit distance for fine-grained shape similarity.

Kleiman, Y., Ovsjanikov, M., 2017. Robust structure-based shape correspondence. CoRR abs/1710.05592.

Laga, H., Mortara, M., Spagnuolo, M., 2013. Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. ACM Trans. Graph. 32, 150:1–150:16.

Leordeanu, M., Hebert, M., 2005. A spectral technique for correspondence problems using pairwise constraints, in: Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2, IEEE Computer Society, Washington, DC, USA. pp. 1482–1489.

Maciel, J.a., Costeira, J.a.P., 2003. A global solution to sparse correspondence problems. IEEE Trans. Pattern Anal. Mach. Intell. 25, 187–199.

Mitra, N., Wand, M., Zhang, H.R., Cohen-Or, D., Kim, V., Huang, Q.X., 2013. Structure-aware shape processing, in: SIGGRAPH Asia 2013 Courses, ACM, New York, NY, USA. pp. 1:1–1:20.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. ACM Trans. Graph. 21, 807–832.

Pechuk, M., Soldea, O., Rivlin, E., 2008. Learning function-based object classification from 3D imagery. Comput. Vis. Image Underst. 110, 173–191.

Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., Zhang, H., 2010. Contextual part analogies in 3D objects. International Journal of Computer Vision 89, 309–326.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion, in: Proceedings of the Symposium on Geometry Processing, Eurographics Association. pp. 1383–1392.

Tam, G.K., Martin, R.R., Rosin, P.L., Lai, Y.K., 2014a. An efficient approach to correspondences between multiple non-rigid parts. Computer Graphics Forum 33, 137–146.

Tam, G.K.L., Cheng, Z., Lai, Y., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X., Rosin, P.L., 2013. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. IEEE Transactions on Visualization and Computer Graphics 19, 1199–1217.

Tam, G.K.L., Martin, R.R., Rosin, P.L., Lai, Y.K., 2014b. Diffusion pruning for rapidly and robustly selecting global correspondences using local isometry. ACM Trans. Graph. 33, 4:1–4:17.

Wang, Y., Xu, K., Li, J., Zhang, H., Shamir, A., Liu, L., Cheng, Z., Xiong, Y., 2011. Symmetry hierarchy of man-made objects. Computer Graphics Forum Eurographics 2011 30, 287–296.

Zhang, H., Sheffer, A., Cohen-or, D., Zhou, Q., Kaick, O.V., Tagliasacchi, A., 2008. Deformation-driven shape correspondence.

Zheng, Y., Cohen-Or, D., Mitra, N.J., 2013. Smart variations: Functional substructures for part compatibility. Computer Graphics Forum (Eurographics) 32, 195–204.

Zhu, C., Yi, R., Lira, W., Alhashim, I., Xu, K., Zhang, H., 2017. Deformation-driven shape correspondence via shape recognition. ACM Trans. Graph. 36, 51:1–51:12.