

Constructing Common Base Domain by Cues from Voronoi Diagram

Tsz-Ho Kwok, Yunbo Zhang, Charlie C.L. Wang

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China

Abstract

In this paper, we propose a novel algorithm to construct common base domains for cross-parameterization constrained by anchor points. Based on the common base domains, a bijective mapping between given models can be established. Experimental results show that the distortion in a cross-parameterization generated on our common base domains is much smaller than that of a mapping on domains constructed by prior methods. Different from prior algorithms that generate domains by a heuristic of having higher priority to link the shortest paths between anchor points, we compute the surface Voronoi diagram of anchor points to find out the initial connectivity for the base domains. The final common base domains can be efficiently generated from the initial connectivity. The Voronoi diagram of the anchor points gives better cues than the heuristic of connecting shortest paths greedily, therefore resulting in an efficient and reliable algorithm for construction of common base domains that bring to low distortion in constrained cross-parameterization.

1 Introduction

A bijective mapping among different models is required in many geometry processing applications, such as texture mapping, morphing, pair-wise model editing, details transfer, model completion, shape analysis, and model database preparation. The procedure of constructing such a mapping is called parameterization, which is studied extensively in literature. Global approaches [10, 20, 4] usually map an entire model to a global domain (e.g., a sphere). However, for applications like morphing and database preparation, parameterization is usually constrained by semantic features, which can be specified as corresponding anchor points on the surfaces of input models. Consistent mesh parameterization [19] was proposed to decompose an entire mesh into several triangular domains, the vertices of which are the anchors points. Later, algorithms called *cross-parameterization* [5] or *inter-surface mapping* [21] were proposed to solve this problem by constructing consistent domains automatically, without requiring the specified domain connectivity. They trace shortest paths between anchor points and construct the domain incrementally by a trial-and-error approach. The quality of final parameterization (such as distortion) is not considered in the step of base domain construction. A relaxation based smoothing step was introduced in [5] for reducing distortion. Nevertheless, a recent study [7] shows that distortion is seriously affected by the shape and connectivity of domains, and such distortion can hardly be reduced by the smoothing step. Therefore, a method better than the heuristic of linking shortest paths greedily is needed to construct the common base domains. This motivates our research.

Given two surface meshes, a source model M_s and a target model M_t , which have the prescribed sets of corresponding anchor points G_s and G_t , linking the anchor points on both models in a consistent way can construct two triangular patch layouts P_s and P_t having the same connectivity. This layout can be represented by a pair (G, K) , where G is the anchor points and K is an ab-

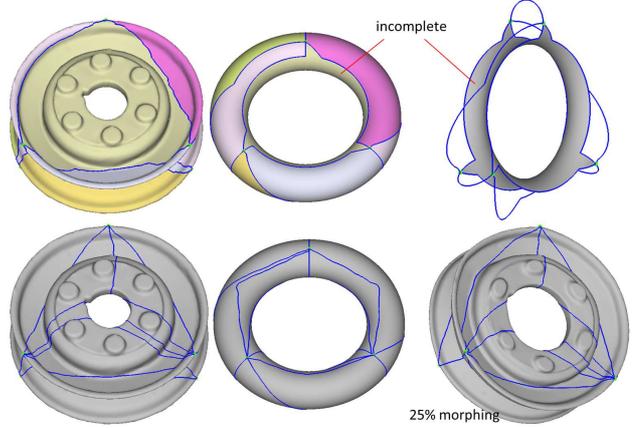


Figure 1: A comparison between the algorithm [5] and our method in construction common base domains on a pair of models with genus-one topology. Only six anchor points are specified. The algorithm [5] fails to construct a complete set of common base domains (top), as the patch at the inner tunnel loop (pointed by red lines) has not been triangulated. The patch is enclosed by all the six anchor points because a pair of anchor points cannot support multiple paths. On the contrary, our approach can build a valid set of common base domains (bottom).

stract simplicial complex that contains all the topological (i.e., adjacency) information. The cross-parameterization between M_s and M_t is found by computing the mapping between all the corresponding patches in P_s and P_t .

Basically, the mapping in cross-parameterization may not be stretch-free (i.e., isometric). However, applications such as model synthesis and shape blending wish to minimize such distortion as it causes unwanted shape distortion on the results (e.g., the mapping shown in Fig.4). Using the graph given by *Voronoi diagram* (VD) to construct base domains can reduce distortion in parameterization [2] and help having a better understanding about the shape similarity [14]. However, how to use VD to construct common base domains for constrained cross-parameterization remains unsolved. Two challenging problems are: 1) the VD centered at anchor points may not be dual to a triangulation, therefore construction of base domains is not straightforward; 2) although corresponding anchor points are given in two input models, the graph of the two VDs may not be the same – thus constructing consistent patch layouts is not guaranteed. Our approach in this paper solves these problems.

As mentioned by [5], models that need to be cross-parameterized usually have similar features. Such similarities are reflected by correspondences between anchor points. Therefore, the Voronoi diagram centered at the anchor points is a very good cue for the process, and the base domains construction based on such a global consideration is more reliable (see Fig.1). We define that a good cross-parameterization should have low distortion in terms of L^2 -stretch and angle distortion (see Section 6 for details

of metrics used).

1.1 Related work

Surface parameterization, which has been studied for many years, computes the mapping from a surface to a domain. It is an essential step in many geometry processing applications. To map genus-zero models, a spherical domain [18] is commonly used, but it fails when the topology of input surfaces is not trivial. Therefore, a more flexible framework is to use complexes based domains [2, 8, 19, 21, 5, 17], where surfaces are first segmented into simplicial complexes (called *base domains* in some approaches). The global parameterization of an entire model can be obtained by parameterizing the partitions of a mesh surface into its corresponding base domains with the same boundary condition. There are many approaches in literature related to domain construction [4, 20] and optimization [17] for single model parameterization; we refer readers to [22] for a comprehensive survey.

When a set of models are partitioned consistently, the base domains shared by all the models are called *common base domains*. Treating this common base domain as a bridge, we can find a mapping between different models; this mapping is called *consistent mesh parameterization* [19] or *cross-parameterization* [5]. The main difficulty of these algorithms is how to construct consistent base domains on input models. Such common base domains are the basis for the remaining procedure of computing cross-parameterization.

The work of Praun et al. [19] pioneered in this field. They use the defined connectivity of base domains on a template model as input. For each input mesh with the given anchor points specified corresponding to those on the template model, their algorithm traces the paths between anchor points based on the template’s connectivity. In order to guarantee consistency, for every path, *intersections*, *blocking* and *cyclical order* are checked before the domains are constructed.

Subsequently, more researches focus on how to generate common base domains automatically with the given models and anchor points. Kraevoy et al. [5] and Schreiner et al. [21] proposed using the heuristic of paths’ lengths to generate base domains. The shortest paths between anchor points are computed, and then the path pairs are inserted into a priority queue based on their lengths. A greedy method is used to construct the paths, and the invalid paths are eliminated in a trial-and-error manner. Finally triangular patch layouts can be obtained although there is no guarantee of success. These triangular patch layouts are the base domains we need. Nevertheless, the topology and the shape of the patch layouts generally have high distortion in the cross-parameterization. The distortion of mappings is still significant even after applying the relaxation based smoothing operator proposed by [5]. Schreiner et al. [21] tried to reduce distortion in coarse-to-fine mapping optimization, and Kwok et al. [7] developed a post-processing method for a similar purpose. However, since these optimization methods are processed after the base domains are constructed, they cannot help much if the base domains cannot be constructed properly.

Li et al. proposed a pants decomposition [11] to partition input models into pants which are genus-zero with three boundaries. After matching each corresponding pant with two regular hexagonal base domains, the bijective mapping between two models is obtained. Their method can handle models with different genus numbers while ours focuses on constructing common base domains bring low distortion in the cross-parameterization. There is some

other related work that computes the one-to-one correspondence between models via the strategy of mesh fitting (e.g., [23, 24, 25]). Wang et al. [24] extended the skin algorithm [15] to compute such a mapping. The input models for this algorithm could be mesh, polygon soup or point clouds. They grow two skins simultaneously to approximate different skeleton models, and finally obtain two models with the same connectivity. Recently, another mesh fitting based approach for generating compatible mesh surfaces was presented by Yeh et al. [25], it approximates the input geometry with a linearized biharmonic surface. This kind of template fitting based approaches [24, 25, 23] in general is slower than the cross-parameterization based remeshing approaches.

In the literature of geometric modeling, there are many approaches for the surface matching between models. The recent work focuses on how to find the point matching in an isometric (or nearly isometric) deformation. Lipman and Funkhouser used Möbius voting in [13], Clements and Zhang employed spectral analysis in [1], Ovsjanikov et al. used heat kernel in [16], and Li et al. conducted the multi-dimensional scaling (MDS) embedding domain in [9]. However, the problem to be solved here is different; we focus on constructing a general non-isometric mapping and attempt to reduce distortion in the mapping function.

1.2 Main result

We propose a new algorithm to construct the common base domains between source and target models for constrained cross-parameterization. Our algorithm gives a better result than [5] in certain aspects, such as parameterization quality, efficiency and reliability.

A novel strategy that uses *Voronoi diagram* (VD) as cues to govern the construction of common base domains brings the following advantages to our algorithm:

1. Using VD centered at anchor points to form Delaunay-like triangular layouts gives good-shaped common base domains, thus generating less distortion in cross-parameterization (see Table 2 and Fig.8).
2. Computing the shortest paths within the corresponding clusters of triangles can speed up the whole process. The complexity of our algorithm is proved to be bounded by the prior method (see Section 5.3).
3. With the help of multiple linkages, we can form a valid triangulation on a Voronoi diagram which is not dual to a triangulation (see Section 5.1).
4. The Voronoi tessellation gives directional guidance for path travelling when linking two anchor points. This prevents unwanted swirling when computing on a cylindrical shape, where prior methods need complicated operations to tackle this problem (see Fig.2 for an illustration).

The smoothness of a domain boundary is affected by mesh quality. If the quality of an input mesh is poor (e.g., having many irregular triangles and with significantly non-uniform sizes of triangles), the shape of the base domain may be poor too. We solve this problem by adding edge nodes adaptively for Dijkstra search (see Section 4). As the quality of parameterization between a set of patches does highly depend on the shape of patches, adding edge nodes can get rid of the influence of poor mesh quality and give smooth boundaries over meshes.

2 Local vs. Global Approaches

Prior methods (e.g., [5, 21]) compute common base domains by the heuristic of having higher priority to link

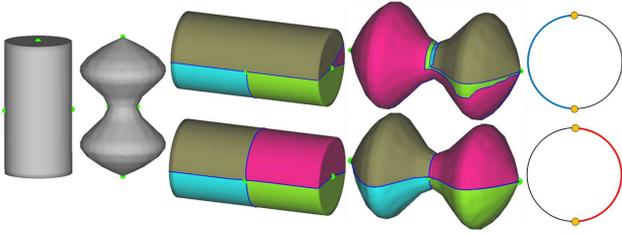


Figure 2: Unwanted swirling: (left) input models with four pairs of anchor points in green, (top-middle) patch layouts generated by [5], and (bottom-middle) patches generated by our method. The path between the two anchor points in the middle could travel in two different directions due to lack of uniqueness in shortest path based heuristic. Analogously, two anchor points are placed at antipodal points of a circle, the shortest path may pass along either the left (right-top) or right (right-bottom) path around circumference of the cylinder.

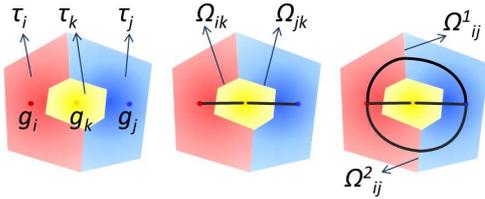


Figure 3: An illustration of multiple linkages. (Left) After clusters formation, there are two cuts between the two outermost clusters (red and blue). (Middle) Single linkage simply connects two anchor points. (Right) Multiple linkages between a pair of anchor points can be constructed through the corresponding cuts indicated by VD.

shorter paths. Patch layouts are constructed in a trial-and-error manner, which can be considered as a locally greedy search in the solution space and are therefore called *local* approaches. In contrast to these local approaches, we propose employing a Voronoi diagram centered at the anchor points as a governing field for the construction. The Voronoi diagram is actually representing the distribution of anchor points as well as the topological structure of input models. This method for constructing common base domains in a *global* manner can tackle several problems in the local approaches, which are analyzed below. A summary of comparison is listed in Table 1 and results are discussed in Section 6.

2.1 Multiple linkages

As mentioned by [5], the procedure for tracing paths can fail if there are not enough anchor points. The upper row in Fig.1 displays the failure in base domain construction by [5] with six anchor points on two genus-one models. The major reason for the failure of the algorithm [5] in this case is that there are not enough anchor points to support new paths. Specifically, the anchor points cannot be linked multiple times because the local approach cannot identify the differences between two paths with the same endpoints. [21] tried to tackle this problem by adding more anchor points automatically; however, it is not guaranteed that the newly added pair of anchor points respects the corresponding similarities correctly.

In our global approach, we allow anchor points to be connected multiple times. This can be achieved by forcing the paths to travel through the corresponding cuts where the clusters connect each other (e.g., Fig.3). Benefited by VD, more paths can be supported with the same number

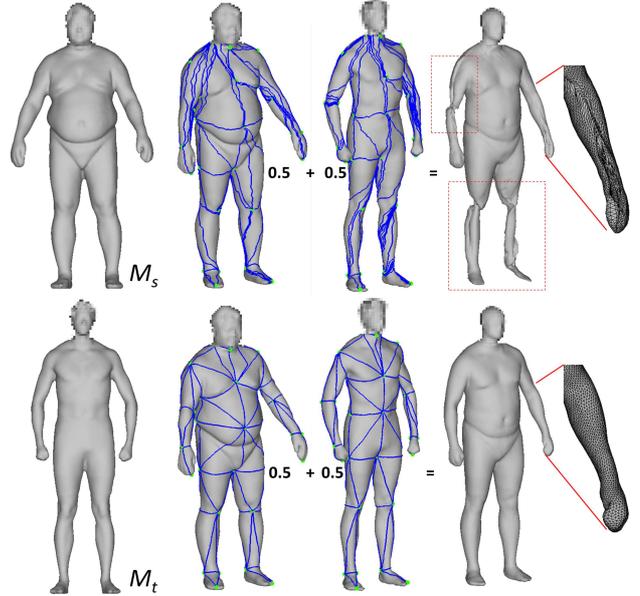


Figure 4: An example of human models (male1/2): (left) source model M_s and target model M_t , (top-right) base domains constructed by [5] and the blending result of M_s and M_t , and (bottom-right) common base domains constructed by our VD-based method and the blending result. This figure shows that the unwanted swirling can greatly affect the result of cross-parameterization (see the right arm and the legs). However, even when there is no swirling on the left arm, badly shaped base domains produce high distortion in cross-parameterization, therefore, a poor meshing result.

of anchor points. Therefore, our approach can successfully generate common base domains for the example shown in Fig.1. To address this feature, paths are identified by two anchor points as well as an index i indicating that it is the i -th path linking these two anchor points.

2.2 Unwanted swirling

Local approaches suffer from lack of uniqueness in the shortest path problem, which may have multiple solutions for the same inputs. This brings uncertainty to the local approaches for constructing base domains. An illustration is given in Fig.2 where two points on a cylindrical surface can be linked in different ways. When there is no guidance to lead the corresponding paths generated in the same direction on two models, a corresponding path may travel downward on M_s while travel upward on M_t ; therefore, swirling happens. One real example containing unwanted swirling is shown in Fig.4, where 37 anchor points are provided for cross-parameterization and the common base domains constructed by [5] lead to a highly distorted mapping caused by unwanted swirling (see the top row of Fig.4). Thanks to the features of having directional guidance, the capability of supporting multiple linkages and the global connectivity information provided by VD, our method can avoid involving complicated topological operations to handle swirling and can produce a very good result (see the bottom row of Fig.4).

2.3 Computational complexity

Local approaches (e.g., [5]) first compute the shortest paths among all pairs of anchor points, and then construct the common base domains in a trial-and-error manner. Many invalid paths are computed unnecessarily. The searching procedures of the shortest paths are conducted on the

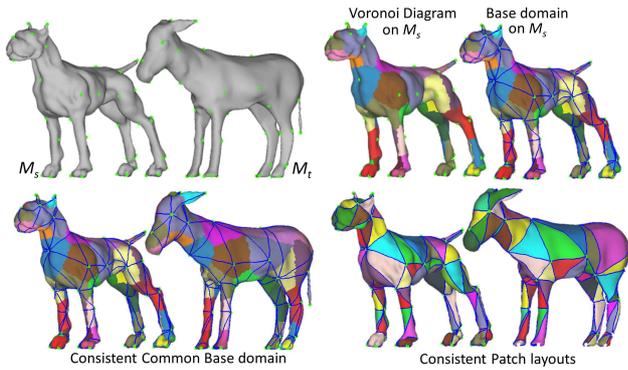


Figure 5: An algorithm overview. (Top - from left to right) The input source model M_s and the target model M_t with prescribed anchor points, computing the surface VD centered at the anchor points by face clustering, and building the base domain P_s on M_s (details can be found in Section 5.1). (Bottom - left) Constructing base domain P_t on M_t using the connectivity of P_s (in Section 5.2). (Bottom-right) Matching the corresponding patches in P_s and P_t that are shown in the same color.

Table 1: Local vs. Global Approaches

	Multi-Paths	Swirling	Complexity
Local	Not supported	Tackled by post-processing	Best case: $O(n_a n_e \log n_v)$
Global	Supported	Prevented by directional guidance	Worst case: $O(n_a n_e \log n_v)$

whole mesh surface. Using VD as the underlying governing field, we know exactly which paths need to be computed by the neighboring relationship of the diagrams. Our approach can always limit the Dijkstra searching within two diagrams (i.e., local regions) of VD centered at anchor points. Therefore, the computational complexity is greatly reduced and proved to be bounded by local approaches (detailed discussion is given in Section 5.3).

2.4 Distortion in parameterization

Last but not least, local approaches simply construct paths in a greedy way, which does not consider the shape of base domains. They just aim at building a valid common base domain. However, we found that the distortion in cross-parameterization highly relates to the shape of base domains. As shown on the left arm of a human body in Fig.4, the shape of the domain is very bad and the parameterization result is highly stretched, although swirling does not occur. Fortunately, Voronoi diagrams suggest the connectivity of the base domains so that the shape of common base domains generated by our method is much better. With these base domains, our approach gives a cross-parameterization with low distortion.

3 Algorithm Overview

The overall algorithm for the construction of common base domains on a pair of models with prescribed anchor points is presented in this section. Common base domains are constructed on a source model, M_s , and a target model, M_t , in two stages (see Fig.5 for an illustration).

Build base domains on source model: This stage finds out the connectivity of base domains constrained by anchor points on M_s . We first use geodesic distance to compute

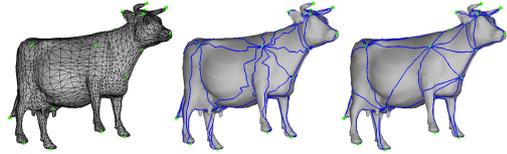


Figure 6: The shortest paths are constructed to link the anchor points on an input cow model (left). (Middle) Only edge paths and face paths are used in [5], and (right) smoother curves can be obtained by using adaptively added edge nodes.

the Voronoi diagram of the anchor points by face clustering. Next, the neighboring relationship between the diagrams (i.e., regions centered at the anchor points) is employed to establish the connectivity of base domains based on the anchor points. Finally, triangulation is performed to produce a triangular patch layout P_s , which is stimulated by the dual-graph of VD.

Construct consistent patch layout on target model:

After building the base domains on the source model M_s , we need to transform the connectivity of P_s onto the target model, M_t . We first compute the VD on M_t using geodesic distance to speed up the process. Second, a proper linking order is used to produce single-linkage paths and multiple-linkage paths correctly. Finally, a topologically consistent triangular layout P_t is constructed on M_t .

The result of this procedure is a set of common base domains in good shape. We can then compute a bijective mapping between the corresponding patches in P_s and P_t by the method in [5], and a cross-parameterization between M_s and M_t is then obtained.

4 Path Tracing

Before presenting the main part of our algorithm, we introduce a path tracing method in this section, which is a frequently used operator in our base domain construction algorithm. Edge nodes are employed to improve the smoothness of paths (see Fig.6) and a robust method (without using numerical predicates [3]) is exploited to detect intersections between paths.

The shortest path between two anchor points is found by linking vertices and edge nodes so that it can travel across faces and gives smoother boundaries for the constructed patches. The concept of adding edge nodes is similar to adding Steiner vertices [21, 6]. They add Steiner vertices to ensure success in emanating new paths, whereas we aim at having smooth boundaries. Kraevoy and Sheffer [5] add face paths after edge paths to guarantee completion of base domain; however, introducing face paths after edge paths may produce some poorly shaped patches. We adaptively add new edge nodes to ensure that there are always enough nodes to pass through. Although the computation of the shortest path is slowed down if we add too many edge nodes, our algorithm is still fast enough as we limit the computation of the shortest path in some local regions based on VD.

Initially, each edge is associated with $e = \max(b, bl/\bar{l})$ edge nodes, where b is the minimal number of edge nodes on an edge, l is the length of the current edge, and \bar{l} is the average length of the neighboring edges (i.e., adjacent edges of two end vertices of the current edge including the edge itself). This formula compares the length of current edge with its neighbors', and adaptively adds more edge nodes if the edge is much longer than its neighbors. In our implementation, $b = 3$ is used. When an edge is split by a path into two edges (as shown in the left and in the middle

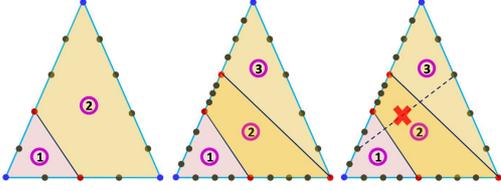


Figure 7: An illustration of adaptively adding edge nodes and robust intersection check: (left) adding a new path across the face split the face into two regions, (middle) new edge nodes have been adaptively added and a new path further subdivides the face into three regions, and (right) a proposed new path (in dash line) intersecting the existing paths is prevented as its two ending nodes are located in different regions of the face.

of Fig.7), the number of edge nodes on the new edges must be adaptively adjusted based on the above formula.

Robust Detection of Intersections: Intersections between paths on a face must be prevented. However, using numerical predicates to detect such intersections suffers from robustness problems [3]. Such intersections can be effectively and robustly detected by topological information in our algorithm. As illustrated in Fig.7, adding a new path in a face subdivides the face into two regions. We can easily assign the edges nodes on the left of the newly added path to one region and the edge nodes on the right side to another region. The same assignment is performed on the adaptively added edge nodes. Another path can be further added if its two end-nodes are located in the same region of the face (see the middle of Fig.7). The face is further subdivided into three regions by this new path, and the relevant edges nodes are assigned to the new regions. A path intersects with other existing paths on a face when its two end-nodes fall in different regions of the face (see the proposed third path in the right of Fig.7). By this way, intersection prevention can be robustly conducted in our path tracing operator.

5 VD-based Construction of Base Domains

Prior methods use the heuristic of always linking the shortest paths first, in which the shapes of base domains are not taken into account. In this section, our VD-based construction algorithm of common base domains is presented in detail. The shapes of common base domains are considered as the connectivity for the patch layout obtained from the cues of Voronoi diagram, the dual graph of which is a Delaunay triangulation presenting triangular patches in very good quality.

5.1 Base domain on source model

In the first stage, we construct base domains on a source model, M_s . There are mainly three steps: 1) computing the approximate Voronoi diagram centered at the anchor points, 2) linking anchor points by paths passing through the adaptively added edge nodes, and 3) triangulating the regions that not forming triangular patches. Note that, the Voronoi diagram (VD) computed here is an approximation of the exact VD as it is based on clustering triangular faces into different regions (i.e., approximate cells of VD) centered at anchor points.

Given the anchor points $\{g_1, \dots, g_s\}$ on M_s , we first randomly select one face adjacent to each of these anchor points as the sites of the approximate VD, $S = \{f_1, \dots, f_s\}$. The triangular faces are then clustered into s regions by the multi-source Dijkstra's algorithm using geodesic distance. As a result, every anchor point g_i is

surrounded by a set of faces which are closer to g_i than to other anchor points. The region formed by these faces is the approximation of a Voronoi cell (called *cluster*, τ_i , in this paper). Different from the approach of Eck et al. [2], the anchor points of VD in our algorithm are hard constraints and cannot be moved. Therefore, we have the following observations.

Observation 1 When the sites (anchor points) are the hard constraints, the Voronoi diagram centered at them may not always be dual to a triangulation.

A *cut* on the clustered triangular mesh is a contiguous set of edges along which the same pair of clusters touch.

Observation 2 There are pairs of clusters may sharing more than one *cut*.

Therefore, constructing the triangulation linking the anchor points as a dual-graph of VD is not straightforward. Sometime a two-manifold base domain cannot be constructed from the VD here. However, base domain does not have to present a fully two-manifold topological structure. For example, a base domain can have multiple different edges sharing same endpoints. This is often misunderstood as an ill-defined situation, because base domain is normally considered as a two-manifold mesh. In the section below, we get rid of this misunderstanding and propose an approach that allows multiple linkages between two clusters to form a triangulation here.

Once the clusters centered at the anchor points are generated, we can find out the connections between clusters as well as the paths linking anchor points by the following method.

- **Initial connectivity:** For each cluster τ_i , we first find out its neighboring clusters, $\tau_j \in N(\tau_i)$, as well as the cut Ω_{ij} between τ_i and τ_j . When τ_i and τ_j share more than one cut, an index η is assigned to each of these cuts, where Ω_{ij}^η denotes the η -th cut shared by τ_i and τ_j (see Fig.3 for an illustration).
- **Constructing paths:** After obtaining the initial connectivity of base domains, we construct the boundary of base domains on M_s by tracing paths between anchor points. For example, if the clusters τ_i and τ_j share a cut Ω_{ij} , the shortest path across Ω_{ij} linking g_i and g_j is computed. Note that the search of the shortest path by the Dijkstra's algorithm is narrowed to the regions occupied by τ_i and τ_j , which speeds up the computation significantly. For two clusters sharing more than one cut (i.e., $\eta > 1$), the anchor points must be linked by multiple paths (see Fig.3 for an example). We can enforce a path to travel through a particular cut Ω_{ij}^η by removing the nodes and edge nodes (as well as arcs linking to them) located on other cuts (i.e., Ω_{ij}^ζ with $\zeta \neq \eta$).
- **Patch triangulation:** The paths generated in the above step sometimes do not subdivide the whole surface into triangular surface patches. In this case, we create triangular patches simply by adding paths between anchor points that are not adjacent to each other. This is similar to the triangulation method stated by [5]. Noted that, the search of the shortest path linking two anchor points by Dijkstra's algorithm is conducted locally on a non-triangular patch holding these two anchor points, and the computation is efficient.

This procedure ends with the base domains, P_s , in the form of triangular patch layout on M_s .

5.2 Common base domain on target model

Having the connectivity of base domains P_s defined on M_s , we need to construct the base domains P_t on the target model M_t , which should be topologically identical to the connectivity in P_s . Similar to [19], intersection, cyclical order and blocking need to be checked in order to guarantee that the connectivity of P_s can be consistently transformed to P_t . Different from [19], which randomly picks paths to construct, we sort out a proper order for linking anchor points multiple times, otherwise multiple paths cannot be constructed correctly.

First, clusters, $\{\tau_1^*, \dots, \tau_t^*\}$, for the anchor points, $G_t = \{g_1^*, \dots, g_t^*\} \in M_t$, are generated in the same way as that mentioned in the above section. Second, paths corresponding to the ones in P_s are constructed one by one. Note that the cyclical order of paths between anchor points must be enforced. In addition, once a candidate path is going to form a cycle, non-blocking check is performed to ensure that this cycle encloses the same set of anchor points on both M_s and M_t (details can be found in Section 3 of [5]). The paths are classified into different groups to be constructed in a proper order based on the configurations of underlying Voronoi diagram. The simple paths linking two anchor points, the clusters of which only share one cut, are constructed first. Then, the paths between two anchor points located in non-neighboring clusters are generated. Lastly, the multiple paths linking the same pair of anchor points are traced on the surface of M_t . Pseudo-code of this algorithm is shown in Procedure 1.

Procedure 1 Base domain construction on M_t

Input: P_s, M_t and G_t

Output: P_t

- 1: Construct $\{\tau_1^*, \dots, \tau_t^*\}$ on M_t by G_t .
 - 2: Obtain the neighboring relationship $\Omega_{ij}^{*\eta}$ on M_t .
 - 3: Construct paths ξ_{ij}^* for (g_i^*, g_j^*) if (τ_i^*, τ_j^*) share only one cut and only one path ξ_{ij}^* linking (g_i, g_j) in P_s .
 - 4: Construct paths ξ_{ij}^* if (g_i^*, g_j^*) share no cuts but there is one path ξ_{ij}^* linking (g_i, g_j) in P_s .
 - 5: Construct multiple paths $\xi_{ij}^{*\eta}$ in their corresponding cuts $\Omega_{ij}^{*\eta}$ enforced by cyclical order defined in P_s . If the multiple paths are in the same sector at this moment, the construction of these paths are postponed.
 - 6: Repeat step 5 if any paths are postponed.
-

5.3 Complexity

Let n_v be the number of nodes on the mesh, n_e be the number of edges, n_f be the number of faces, and n_a be the number of anchor points, the multi-source Dijkstra's algorithm to generate VD has complexity $O(n_e \log n_f)$ as the faces serving as 'nodes' on the graph and the neighboring faces having an 'edge' linking them on the graph. For a closed triangular mesh surface, $n_f \simeq 2n_v$; therefore, the complexity of the step generating VD is also $O(n_e \log n_v)$.

When we compute the shortest path between two anchor points, the computational complexity is $O(n_e \log n_v)$. In practice, as the computation takes place in local regions (e.g., two clusters holding two anchor points to be linked), the complexity can be reduced to $O(\frac{2n_e}{n_a} \log \frac{n_v}{n_a})$. On a successfully constructed layout of triangular patches, the number of paths n_p is about $3n_a$. Therefore, constructing n_p paths leads to a computation with $O(\frac{2n_p n_e}{n_a} \log \frac{n_v}{n_a}) \simeq O(n_e \log \frac{n_v}{n_a})$. In fact, this is the optimal case. In the worst case, the number of edges and nodes are non-uniformly distributed in the clusters, so every path needs the computa-

tion in the complexity of $O(n_e \log n_v)$. Therefore, the overall computation has complexity $O(n_a(n_e \log n_v))$. Since the number of edge nodes on each edge is a small constant, it is neglected in the analysis of computational complexity.

To summarize, the upper bound of the computational complexity of our algorithm is $O(n_a(n_e \log n_v))$ (which in fact rarely happens) while the lower bound is only $O(n_e \log n_v)$.

Considering the methods based on the heuristic of having higher priority to link shortest paths (e.g., [5, 21]), they need to first compute the length of all the possible paths linking all anchor points, which leads to a computation with complexity $O(n_a(n_e \log n_v))$. The subsequent steps to add new paths in a trial-and-error manner are also taken on the whole mesh surface (i.e., may have an overall order of $O(n_a(n_e \log n_v))$ for a few runs). Although there is no theoretical analysis to find out the complexity of trial path generation, the best case of their approach gives the complexity $O(n_a(n_e \log n_v))$.

Remark The speed of our method is much higher on mesh surfaces with relatively uniform anchor points.

When more anchor points are given on the models, a more significant speedup can be achieved in our algorithm compared with prior methods.

6 Results

We have implemented the proposed algorithm in C++ plus OpenGL. All the experimental tests shown in this paper are run on a standard PC with Intel Core 2 Duo CPU E6750 at 2.67GHz plus 2GB RAM.

The approach proposed in this paper can construct good common base domains in an efficient way. We compare our approach with the prior cross-parameterization method [5]. We transfer the triangles of M_s onto M_t by the mapping, Γ , of established cross-parameterization to obtain a new mesh surface M_{st} , which has the same connectivity as M_s but with a shape similar to M_t . To conduct a fair comparison, when implementing [5], we fully apply the post-processing steps, flipping paths and smoothing, to improve the quality of mapping. Moreover, we test our approach by using only one edge node, which is similar to the face path used in [5]. Two error metrics: angle distortion (E_{ang}) and L^2 -stretch (E_{L^2}) in [7] are employed to measure the distortion in the mapping of cross-parameterization.

We have tested our approach on several examples and compared them with the results generated by [5]. The computational statistics and their corresponding charts are shown in Table 2 and Fig.8. In these examples, our method outperforms theirs on both the quality of cross-parameterization (with less distortion) and the computing time. Example models are shown in Figs.5, 9, 10, and 11. The experimental tests verify the analysis in Section 5.3. For the example of armadillo/woman pair, when 45 anchor points are used, our method is 18 times faster than [5]. After increasing the number of anchor points to 85, our method is 25.4 times faster. The computational statistics also show that using more edge nodes could further reduce the distortion. Although the processing time is increased, it is still much faster compared with prior method.

One interesting observation from Fig.8 is that the quality of results obtained from using one edge node is similar to that of using an adaptive number of edge nodes as proposed in Section 4. So one may ask: why do we not simply use one edge node? The answer lies clearly in the example shown in Fig.12 which demonstrates the function of using an adaptive number of edge nodes, where the input mesh of bear has highly non-uniform triangles (see the region near

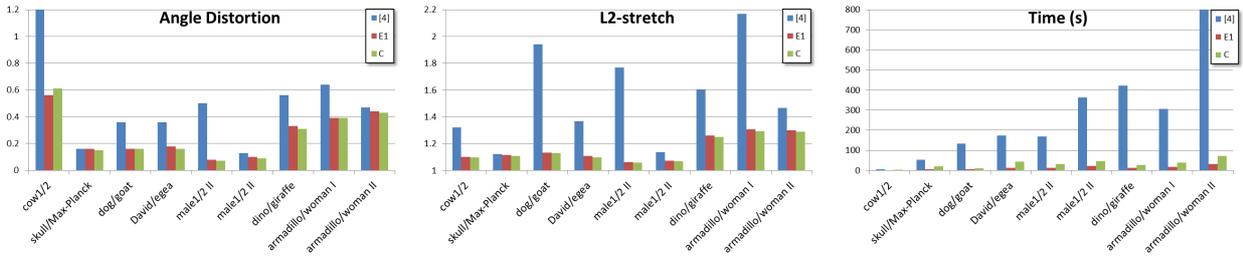


Figure 8: Comparison charts with [5] in terms of angle distortion, the L^2 -stretch and the computing time.

Table 2: Computational Statistics of Cross-Parameterization

Models (M_s/M_t)	Sizes (#v)	n_a	E_{ang}			E_{L^2}			Time (s)		
			[5]	E1	C	[5]	E1	C	[5]	E1	C
cow1/2	2904/4315	27	1.33	0.56	0.61	1.320	1.103	1.097	6	1	3
skull/Max-Planck	23908/7399	24	0.16	0.16	0.15	1.122	1.116	1.107	53	6	20
dog/goat	8020/10436	59	0.36	0.16	0.16	1.943	1.132	1.131	134	5	12
David/egesa	49493/8268	24	0.36	0.18	0.16	1.368	1.110	1.097	174	13	44
male1/2	38438/8950	37	0.50	0.08	0.07	1.769	1.063	1.060	169	14	31
male1/2	38438/8950	75	0.13	0.10	0.09	1.138	1.074	1.068	362	23	47
dino/giraffe	23982/9239	67	0.56	0.33	0.31	1.605	1.260	1.251	422	13	28
armadillo/woman	43245/5691	45	0.64	0.39	0.39	2.170	1.307	1.292	306	17	40
armadillo/woman	43245/5691	85	0.47	0.44	0.43	1.465	1.299	1.289	814	32	72

E1 stands for our algorithm using only one edge node, and C is our algorithm using the number of edge nodes stated in Section 4.

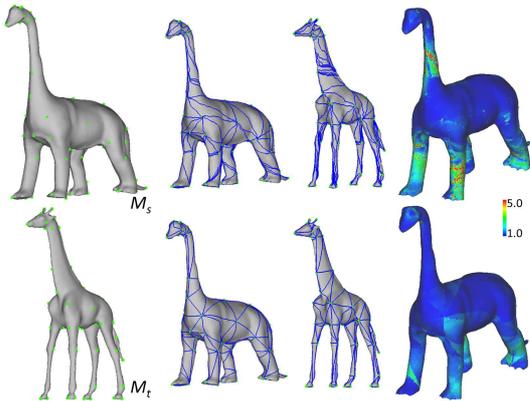


Figure 9: An example of cross-parameterization between dinosaur and giraffe models: (top row) the result produced by [5] and (bottom row) the result produced by our approach. The color maps show the distortions in terms of L^2 -stretch.

its heart). When only one edge node is used, the mapping quality in that region is very bad (as shown by the color map of L^2 -stretch). With the help of adaptive edge nodes, our algorithm can give a much better result even in such an extreme case.

Lastly, we demonstrate that our approach can successfully construct common base domains constrained by anchor points that are distributed quite differently on M_s and M_t . As shown in Fig.13, the anchor points are relatively uniform on the human model but highly non-uniform on the Moai model. However, valid base domains in good shape can still be effectively constructed.

7 Conclusion and Discussion

We propose a novel method to generate common base domains constrained by anchor points for cross-parameterization in a global manner. The Voronoi di-

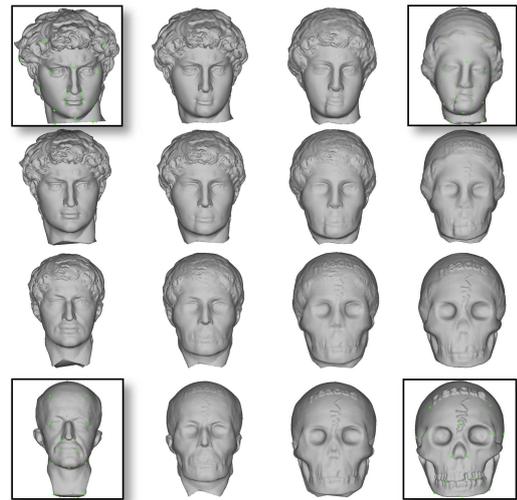


Figure 10: Heads of David (top-left), egea (top-right), Max-Planck (bottom-left) and skull (bottom-right): shape interpolation among four head models at the corners can be generated by the cross-parameterization established on the resultant base domains from our approach.

agrams centered at anchor points are used as governing fields to provide information to construct base domains more reliably. The base domains generated by our approach produce low distortion cross-parameterization. The tracing of the shortest path is conducted locally in two diagrams, which can greatly speed up the computation of base domain construction.

However, our method has some limitations similar to that in prior methods, for example, we can only handle a pair of models having the same topology. The method to process models with different genus numbers is still under research. Li et al. [12] proposed a pants decomposition and showed the possibility of solving this prob-

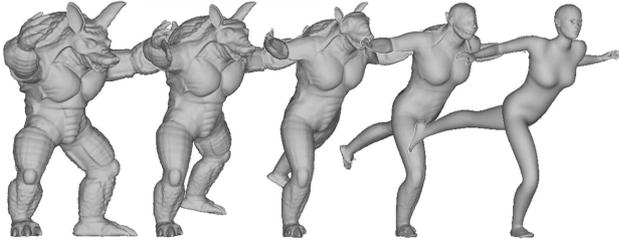


Figure 11: Morphing between armadillo and woman model.

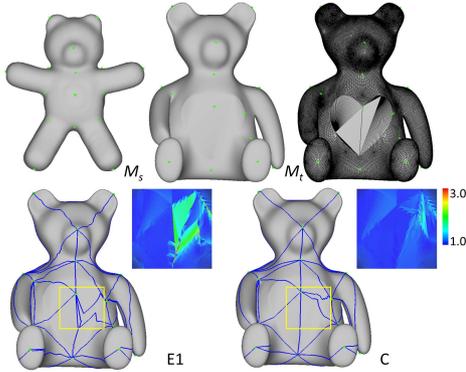


Figure 12: An illustration demonstrating the function of an adaptive number of edge nodes: when an input model is badly meshed (top row), using only one edge node is not enough (bottom-left). The proposed adaptive edge nodes insertion method gives a better result (bottom-right). The color map shows the L^2 -stretch.

lem. One possible future work is try to take advantages of cross-parameterization and pants decomposition to find a high quality parameterization for models having different topologies.

Acknowledgments

The research presented in this paper was partially supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF): CUHK/417508 and CUHK/417109.

References

- [1] A. Clements and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *Proc. of Shape Modeling International*, pages 118–129, 2006.
- [2] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95: ACM SIGGRAPH 1995 Papers*, pages 173–182, 1995.
- [3] C. Hoffmann. *Geometric and Solid Modeling, Chapter 4: Robust and Error-Free Geometric Operations*.
- [4] A. Khodakovsky, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 350–357, 2003.
- [5] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. *ACM Trans. Graph.*, 23(3):861–869, 2004.
- [6] V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: constructing constrained texture maps. *ACM Trans. Graph.*, 22(3):326–333, 2003.
- [7] T.-H. Kwok, Y. Zhang, and C. C. L. Wang. Efficient optimization of common base domains for cross-

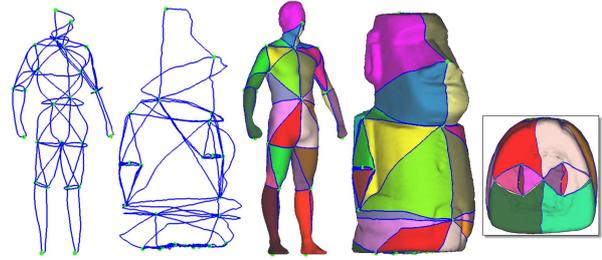


Figure 13: Our approach can successfully generate common base domains constrained by non-uniformly distributed anchor points. The two legs of a human model are mapped to the bottom surface of a Moai model.

parameterization. *IEEE Trans. on Visualization and Computer Graphics*, accepted.

- [8] A. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: multiresolution adaptive parameterization of surfaces. In *SIGGRAPH '98: ACM SIGGRAPH 1998 Papers*, pages 95–104, 1998.
- [9] S. S.-M. Li, C. C. L. Wang, and K.-C. Hui. Bending-invariant correspondence matching on 3d human bodies for feature point extraction. *IEEE Transactions on Automation Science and Engineering*, 8(4):805–814, 2011.
- [10] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, and H. Qin. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Trans. on Visualization and Computer Graphics*, 14(4):805–819, 2008.
- [11] X. Li, X. Gu, and H. Qin. Surface mapping using consistent pants decomposition. *IEEE Trans. on Vis. and Comp. Graph.*, 15(4):558–571, 2009.
- [12] X. Li, X. Gu, and H. Qin. Surface mapping using consistent pants decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):558–571, 2009.
- [13] Y. Lipman and T. Funkhouser. Möius voting for surface correspondence. *ACM Trans. Graph.*, 28:72:1–72:12, 2009.
- [14] Y.-J. Liu, Z. Chen, and K. Tang. Construction of isocontours, bisectors, and voronoi diagrams on triangulated surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1502–1517, 2011.
- [15] L. Markosian, J. Cohen, T. Crulli, and J. Hughes. Skin: a constructive approach to modeling free-form shapes. In *SIGGRAPH '99: ACM SIGGRAPH 1999 Papers*, pages 393–400, 1999.
- [16] M. Ovsjanikov, Q. Merigot, F. Memoli, and L. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.
- [17] N. Pietroni, M. Tarini, and P. Cignoni. Almost isometric mesh parameterization through abstract domains. *IEEE Trans. on Vis. and Comp. Graph.*, 16(4):621–635, 2010.
- [18] E. Praun and H. Hoppe. Spherical parametrization and remeshing. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 340–349, 2003.
- [19] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *SIGGRAPH '01: ACM SIGGRAPH 2001 Papers*, pages 179–184, 2001.
- [20] N. Ray, W. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, 2006.
- [21] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *ACM Trans. Graph.*, 23(3):870–877, 2004.
- [22] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.
- [23] R. Sumner and J. Popović. Deformation transfer for triangle meshes. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 399–405, 2004.

- [24] Y. Wang, C. Wang, and M. Yuen. Duplicate-skins for compatible mesh modelling. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 207–217, 2006.
- [25] I.-C. Yeh, C.-H. Lin, O. Sorkine, and T.-Y. Lee. Template-based 3d model fitting using dual-domain relaxation. *IEEE Trans. on Vis. and Comp. Graph.*, 17(8):1178–1190, 2011.