Efficient Optimization of Common Base Domains for Cross-Parameterization

Tsz-Ho Kwok, Yunbo Zhang, and Charlie C.L. Wang, *Member, IEEE*

Abstract—Given a set of corresponding user-specified anchor points on a pair of models having similar features and topologies, the cross-parameterization technique can establish a bijective mapping constrained by the anchor points. In this paper, we present an efficient algorithm to optimize the complexes and the shape of common base domains in cross-parameterization for reducing the distortion of the bijective mapping. The optimization is also constrained by the anchor points. We investigate a new signature, *Length-Preserved Base Domain* (LPBD), for measuring the level of stretch between surface patches in cross-parameterization. This new signature well balances the accuracy of measurement and the computational speed. Based on LPBD, a set of metrics are studied and compared. The best ones are employed in our domain optimization algorithm that consists of two major operators, boundary swapping and patch merging. Experimental results show that our optimization algorithm can reduce the distortion in cross-parameterization efficiently.

Index Terms—complex domain, optimization, stretch, cross-parameterization, surface parameterization.

1 Introduction

Computations of a bijective mapping among different models are very useful for many geometry processing applications, such as texture mapping, morphing, pair-wise model editing, shape blending, details transfer, model completion, shape analysis, and model database preparation. A general solution for constructing such a mapping can be computed through the global parameterization approach (ref. [1], [2], [3]). However, for applications like morphing and database preparation, parameterization must be constrained by semantic features, which are correspondingly specified as anchor points on the surface of input models. A method called *cross-parameterization* was proposed in [4] to solve this problem by constructing consistent domains on a pair of models linking the anchor points, and Schreiner et al. [5] presented an approach named as *inter-surface mapping* for a similar purpose.

Without loss of generality, models that need to be cross-parameterized usually have similar features, and the correspondences between anchor points should respect such similarities (i.e., hands mapped to hands as mentioned in [4]). In addition, the constructed cross-parameterization should preserve the shape of the models as much as possible. Similar to the 3D-to-2D surface parameterization problem (e.g., [6]), shape preservation is usually achieved by minimizing the distortion occurring in the bijective mapping. A relaxation based smoothing step was introduced in [4] for this purpose. Nevertheless, our recent study shows that the distortion is seriously affected by the shape dissimilarity of domains between the cross-parameterized models, and such distortion can hardly be reduced by the smoothing step in [4]. This motivates our research.

Given two surface meshes, the source model M_s and the target model M_t, which have the corresponding sets of anchor points G_s and G_t prescribed, 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 (e.g., the algorithms in [4], [7]). Following [7], a polygonal mesh M is represented by a pair (V, K), where V is a set of vertices and K is an abstract simplicial complex that contains all the topological (i.e., adjacency) information. For convenience, we also treat a triangular patch layout as a special mesh with anchor points being the vertices and curved patch boundaries being the edges. Assume that each patch P^i in a patch layout can find a corresponding planar domain B^i (e.g., by [6]), we can obtain the 3D-to-2D mapping

$$\Gamma_s : P_s^i \Rightarrow B_s^i \quad \text{and} \quad \Gamma_t : P_t^i \Rightarrow B_t^i.$$

After establishing the mapping between 2D domains as $\Gamma_{st} : B_s^i \Rightarrow B_t^i$ (e.g., using mean-value coordinates [8]), we have established the cross-parameterization

$$\Gamma = \Gamma_s \cdot \Gamma_{st} \cdot \Gamma_t^{-1} \tag{1}$$

to map M_s to M_t. Basically, such a mapping is rarely stretch-free (i.e., isometric). However, applications like model synthesis and shape blending wish to minimize such distortion as it leads to unwanted shape distortion on the results – an example is shown in Fig.1. Many prior researchers have dedicated on how to minimize distortions in 3D-to-2D mappings as well as in 2D-to-2D mappings. Our recent study finds that optimizing the topology of common base domains in P_s and P_t can further reduce the stretch, and the research presented in this paper works out an efficient algorithm to optimize the topology of common base domains for reducing the distortion in cross-parameterization. The optimization is constrained by the anchor points. Specifically, every anchor point should be part of the simplicial complex of the patch layout and the positions of anchor points cannot be changed during the optimization.

After describing the overview of our domain optimization algorithm in section 2, we investigate a new signature, *Length-Preserved Base Domain* (LPBD) (in section 3), which

All authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong. Corresponding Author: Charlie C.L. Wang (Tel: (852) 2609 8052; Fax: (852) 2603 6002; E-mail: cwang@mae.cuhk.edu.hk).
Manuscript prepared - September 20, 2010;
Accepted - June 8, 2011.

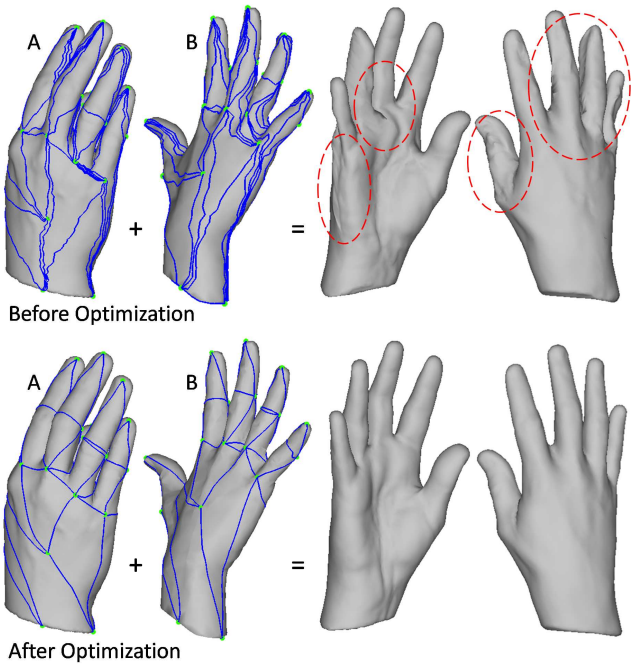

Fig. 1: An example for illustrating the influence of an imperfect cross-parameterization on the result of shape blending. (Top row) Unwanted distortions (circled by red dash lines) are generated on the hand model synthesized from half of the hand A and half of the hand B according to the cross-parameterization of [4]. (Bottom row) Such unreal distortions are vanished after applying our optimization algorithm to the common base domains generated by [4].

can indicate the level of distortion between surface patches in cross-parameterization efficiently and effectively by measuring the dissimilarity between base domains. Based on this new signature, a set of metrics are then studied (in section 4) and the best ones are selected for our domain optimization algorithm which is composed of two major operators, *boundary swapping* and *patch merging*. Before presenting the technical details, we review the previous work and brief the contributions below.

1.1 Previous Work

As an essential step for many geometry processing applications, the construction of bijective mappings between models or domains has been studied for many year. A couple of the previous approaches require a base domain in order to find such mappings. One of the most commonly used base domains is the spherical domain [9], which produces seamless and continuous parameterizations everywhere. However, it only works on genus-zero models. A more general method is to use complexes based domains as [4], [5], [7], [10], [11], [12]. The whole surface mesh is usually partitioned into simplicial complexes, which are served as base domains (or called base meshes in some approaches). Once the base domains are obtained, the original meshes can be parameterized to the base domain, and then through which the cross-parameterization is established.

Although there are many excellent approaches in literature about domain construction (e.g., [2], [3]) and optimization (e.g., [12]) for single model parameterization (see [13] for the comprehensive survey), it is not straightforward to extend them to multiple models when bijective mappings between different models are needed. The main difficulty is how to construct the consistent base domains on input models having similar features and topology. Such common base domains are the basis for the rest of the cross-parameterization.

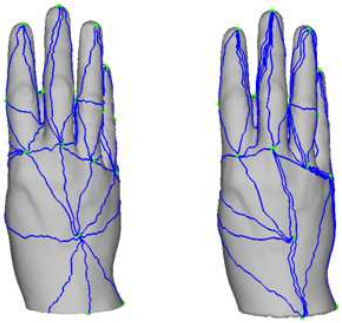

Fig. 2: The flipping path operator proposed in [4] could make the shape of base domains more irregular when the anchor points are distributed very unevenly: (left) before applying the operator and (right) after flipping paths.

Praun et al. [7] used the connectivity of a predefined template base domain and then traced the boundary of patch layouts on each of the input meshes by linking the given feature points in a consistent way as that in the template base domain. The method that links feature points in a consistent manner is based on 1) finding the shortest paths between feature points and 2) preventing the paths from intersecting, blocking and being in wrong cyclical order. Kraevoy et al. [4] and Schreiner et al. [5] further extended the idea of Praun et al. by automating the generation of the template base domain. Kraevoy et al. [4] first compute the shortest paths between all feature points, and then select the best pair of corresponding paths from a priority queue sorted by the sum of path lengths on meshes. By trial-and-error until all the patches have been triangulated, they eventually obtain the patch layout for the base domain. Nevertheless, the topology and the shape of the patch layouts are not optimized to have less distortion in the cross-parameterization. Specifically, when the distribution of anchor points are very uneven, the distortion of mappings is still significant after applying the flipping path operator and the relaxation based smoothing operator which is proposed in [4] for reducing the distortion in cross-parameterization. The main reason why their flipping path operator does not work is that their metric for governing the operator only considers the valences of base domain (i.e., the number of paths linking to an anchor point). For example, on the model shown in Fig.2, after applying the flipping paths operator, the valences at anchor points become closer to six (i.e., regular); however, the shape of domain patches becomes more irregular. The new signature, *Length-Preserved Base Domain* (LPBD) proposed in this paper, can solve this problem. The approach of Schreiner et al. [5] constructs the base domain in a manner similar to [4], and tried to reduce the distortion in the coarse-to-fine mapping optimization. However, they only change the connectivity of common base domains by adding points to resolve the domain construction problem on high genus models.

Wang et al. [15] extended the skin algorithm [16] by simultaneously growing two skins with identical connec-

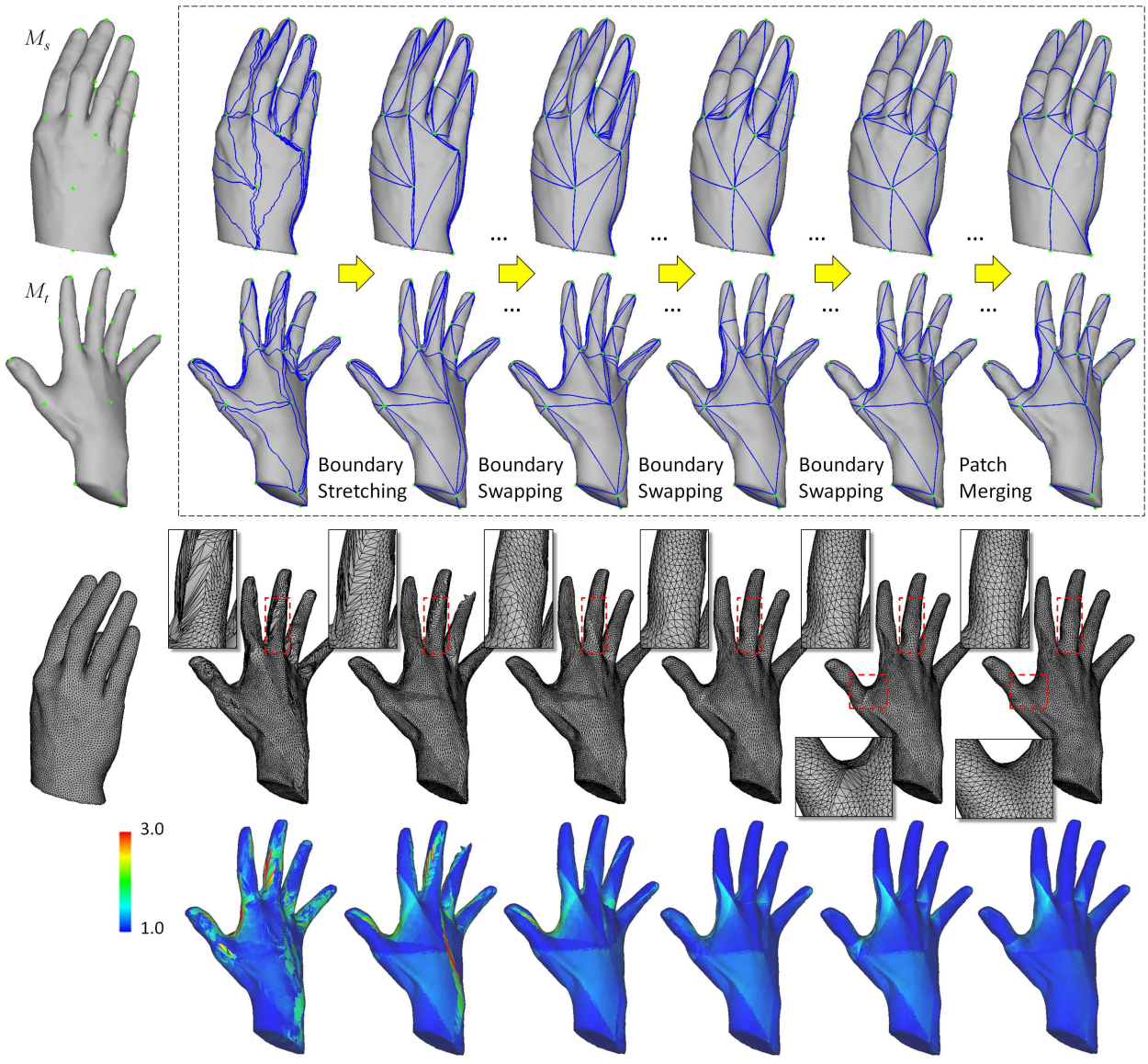

Fig. 3: An example for illustrating the overview of our domain optimization algorithm. The left most column shows the two input hand models with prescribed anchor points (in green). (First and second rows) The topology and shape of common base domains on M_s and M_t are changed during the optimization. (Third row) A uniform mesh of the source model, M_s, is transferred onto the target model, M_t, using the established cross-parameterization where the transferred mesh on the target model should be smooth if the cross-parameterization has small distortion. (Last row) The distortion of cross-parameterization can also be visualized by the distribution of L^2-stretch [14] on the transferred mesh model.

tivity over different skeleton models. This algorithm has less topological constraints on the models to be approximated, and the input models could be mesh, polygonal soap or point clouds. Another recent mesh fitting based approach for generating compatible mesh surfaces is [17], where the authors approximate the input geometry with a linearized biharmonic surface. These template fitting based approaches [15], [17], [18] in general are slower than the cross-parameterization based remeshing approaches.

Some researches focus on mapping between models with different topologies [19], [20], [21]. The work of Zhang et al. [19] decomposes a given model by branches of its reeb graph, and the approach of Bennett et al. [20] is based on an initial alignment scheme that allows users to identify topological changes. Recently, Li et al. proposed a pants decomposition in [21] to partition input models into pants which are genus-zero with three boundaries. After matching each corresponding pant with two regular hexagonal domains, the bijective mapping between two models is obtained. Their method can handle models with different genus numbers while ours focuses on solving distortion minimization problem by optimizing the common base domains.

1.2 Contributions

We develop a domain optimization algorithm for reducing the distortion in cross-parameterization, which is constrained by the prescribed anchor points. The efficiency and effectiveness of this optimization algorithm is benefited by a new signature, *Length-Preserved Base Domain* (LPBD), investigated in this paper. Based on LPBD, a set of metrics are studied and the best ones are selected for the domain

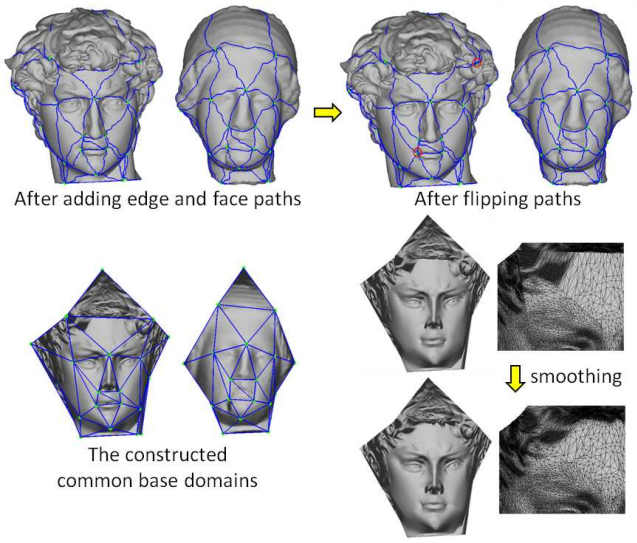

Fig. 4: Illustration of base domain construction in [4].

optimization algorithm, which manipulates the connectivity of common base domains with two operators, *boundary swapping* and *patch merging*. The major novelty comes from using the new signature, LPBD, to convey the distortion in cross-parameterization to the shape dissimilarity of base domains therefore speed up the computation.

2 Domain Optimization Algorithm

Our domain optimization algorithm starts from the common base domains constructed by the method in [4] which is briefly introduced as Pre-step, and the three major steps of our algorithm are presented after that. An example is shown in Fig.3 to illustrate the functionality of each step.

Pre-step: Initial base domain construction

The algorithm presented in [4] is implemented in this step. Firstly, the shortest paths along polygonal edges between all feature points are computed. Then, the best pairs of corresponding paths in terms of the sum of path lengths are chosen as boundaries to construct the common base domains. Again,the selection should prevent the paths from intersecting, blocking and being in wrong cyclical order (similar to [7]). Finally, flipping path and smoothing operators are applied to generate the initial common base domains (see Fig.4).

Step 1: Boundary stretching

This is a step to reduce the differences between the actual boundary shape of a surface patch in the patch layout and the shape of its corresponding planar domain. The stretching operator developed in [22] is applied to the boundary curves one by one. The principle of this curve stretching operator is to convert a curve into a geodesic curve locally by using an edge operator and a node operator to make the curve shorter dynamically – details can be found in [22]. After several iterations, the curve under stretching approximates a geodesic curve linking its two endpoints on the given mesh surface. Note that the stretched curve does not necessarily pass along the edges of given mesh models any more. The intersection between a curve under stretching and other static curves must be prevented by the intersection avoidance method presented in [7]. The resultant patch layouts should have very smooth boundaries for all patches. However, it is not difficult to find from the example shown in Fig.3 that the distortion in the cross-parameterization is still very significant although stretching the boundary of patches can reduce the distortion slightly.

Step 2: Boundary swapping

In this step of our algorithm, the patch layouts on both models, M_s and M_t, are adjusted iteratively and locally to reduce the distortion in cross-parameterization. The *boundary swapping* operator is conducted to optimize the topology of the patch layouts. Without loss of generality, for two patches sharing a common boundary curve, applying the boundary swapping operator on them will convert the patches into two new patches by replacing the boundary curve by a new one linking its opposite anchor points in the triangular patches. The new boundary curve is constructed by first finding the shortest path along the mesh edges linking the opposite anchor points and then stretching the curve by the method in [22]. Again, the intersection between the new curve and other existing curves must be prevented. The priority list of applying the boundary swapping operator is built and maintained based on the metrics of the shape similarity of base domains, and serves as a signature to indicate the level of distortion in cross-parameterization. Detailed studies about the new signature, *Length-Preserved Base Domain* (LPBD), are presented in section 3.

Step 3: Patch merging

A curved boundary shared by two patches will be removed in this step if such a removal helps reduce the distortion in the cross-parameterization. Apart from the reduction of distortion, other properties of the base domain (like the convexity and the flattenability) are also considered during patch merging to generate a valid and optimal result.

Following these main steps, the given mesh surfaces are trimmed by the boundary curves of the new patch layouts using the *Constrained Delaunay Triangulation* (CDT) [23]. The newly constructed patch layouts lead to a cross-parameterization with less distortion (see Fig.3 for an example).

3 Length-Preserved Base Domain

This section investigates a new signature, *Length-Preserved Base Domain* (LPBD), indicating the level of stretch (i.e., distortion) induced in a cross-parameterization. This signature can be efficiently evaluated on the patch layouts for the common base domains of cross-parameterization.

3.1 LPBD as a Signature for Distortion

From the problem description presented at the beginning of this paper, we know that a cross-parameterization is composed of three mapping functions: Γ_s, Γ_{st} and Γ_t. Different from the domain optimization approach for a single model [12] that always employs regular triangles as the planar base domain in the mapping, we allow the shape of the base domain to vary in 2D. Specifically, the planar domain B^i for a 3D patch P^i on the given model is constructed by

1) preserving the length of boundary curves on P^i (i.e., having invariant length of boundaries),
2) straightening each of the 2D boundary curves,
3) trying to mimic the shape of P^i in 2D,

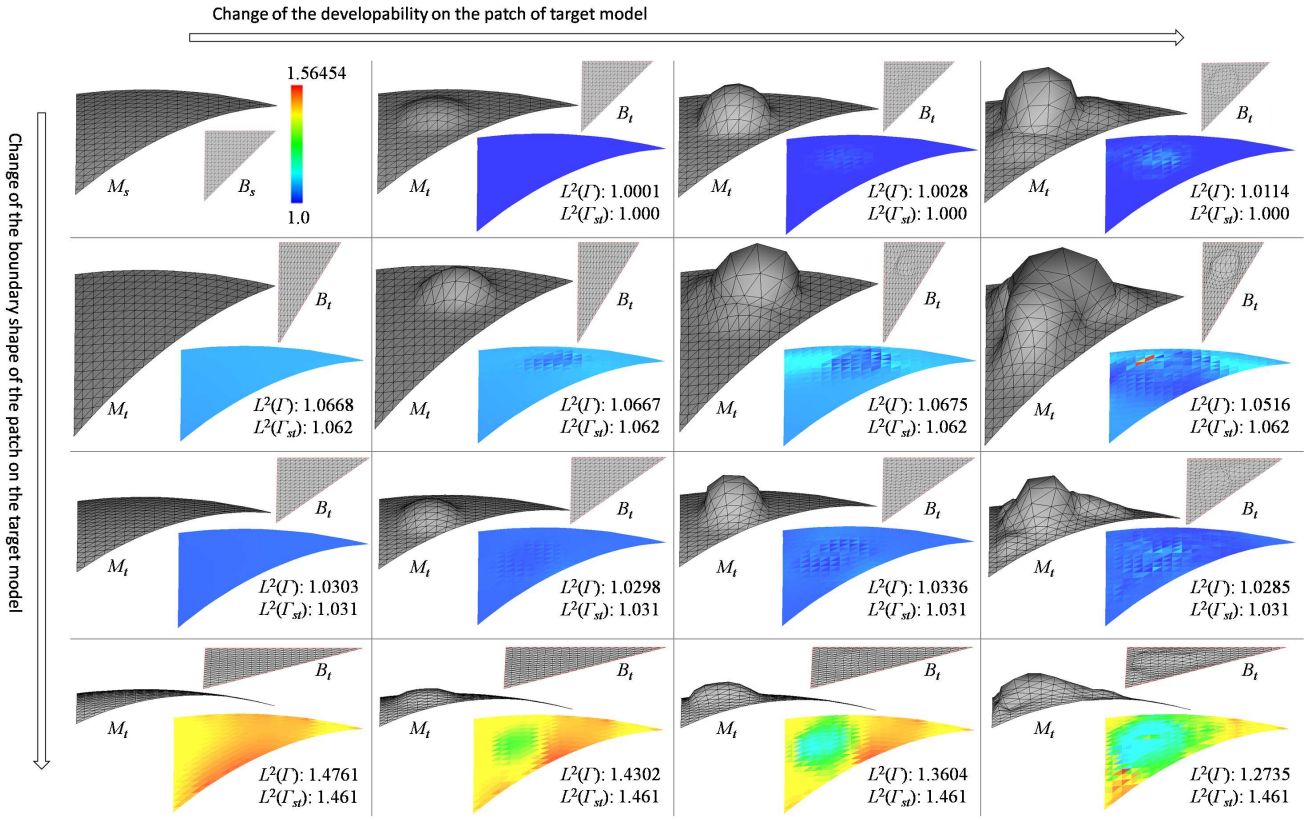

Fig. 5: A study of the robustness of using the dissimilarity on LPBDs to measure the distortion in cross-parameterization. For a given source model M_s shown in the upper-left corner of the figure, we test the distortions between it and various of target models M_t, where the developability of patches keeps decreasing from left to right for patches in the same row and the boundaries of patches are changed for the patches in the same column. The L^2-stretches [14] in the cross-parameterization, $L^2(\Gamma)$, and the L^2-stretches on the planar base domains, $L^2(\Gamma_{st})$, are also listed in the figure. Note that only the big triangles linking the anchor points used to evaluate planar domain distortion, $L^2(\Gamma_{st})$. The color maps are used to visualize the distribution of L^2 stretches generated by the cross-parameterization.

where the whole boundary of a patch is subdivided into curves by the anchor points. Obviously, the planar base domains generated in this way, named as *Length-Preserved Base Domain* (LPBD), have different shapes and sizes. The dissimilarity between the planar base domains B_s^i and B_t^i is a good signature to indicate the cross-parameterization from P_s^i to P_t^i.

By retaining the length of boundary curves, the mapping from P^i to B^i is stretch-free along the boundaries. If the boundary curves on P^i are geodesic curves, flattening them into straight planar line segments leads to a stretch-free mapping around/across the lines. If P_s^i and P_t^i are developable surfaces, the mappings Γ_s and Γ_t from P_s^i and P_t^i to the LPBD, B_s^i and B_t^i, are stretch-free too. As the cross-parameterization Γ from P_s^i to P_t^i is series-wound mappings, $\Gamma_s \cdot \Gamma_{st} \cdot \Gamma_t^{-1}$, the distortion of Γ is only introduced by the 2D-to-2D mapping $\Gamma_{st} : B_s^i \Rightarrow B_t^i$ in this scenario. In other words, the dissimilarity between B_s^i and B_t^i measures the distortion of cross-parameterization Γ.

A more interesting study relates to how significantly the distortion in Γ_{st} contributes to the overall distortion in the cross-parameterization Γ when the patches P_s^i and P_t^i are not developable. With a given planar shape B^i, we can assume that the distortion in the 3D-to-2D mapping from P^i to B^i has been minimized – in our implementation, the mean-value coordinates based parameterization [8] is used for this purpose. The mapping Γ_{st} is again established by mean-value coordinates [8]. We now study how the developability of a 3D patch affects the robustness of using the signature, LPBD, to measure the stretch in cross-parameterization. As shown in Fig.5, for an unchanged source model M_s, when the target model M_t becomes more and more nondevelopable while still keeping the same boundary, the stretch error of the cross-parameterization does not show a significant change. However, when keeping the geometric details (i.e., the level of developability) of patches but changing their boundaries (i.e., the shape of LPBD), the stretch errors of the cross-parameterization change significantly (see the changes in Fig.5 by columns). In summary, the dissimilarity between the LPBDs, B_s^i and B_t^i, is a robust signature to indicate the stretch in the overall cross-parameterization. Different from the original cross-parameterization approach [4] that constructs base domains by linking anchor points with straight lines in 3D, mapping geodesic curves of the patch boundaries into planar straight lines conveys more surface dissimilarity to the shape of 2D domains, B_s^i and B_t^i. This enhances the robustness of the LPBD signature.

3.2 LPBD Computing

The *Length-Preserved Base Domain* (LPBD), B^i, of a patch $P^i \in \Re^3$ on the given model can be easily computed

by a constrained optimization framework similar to the computation of *Length-Preserved Free Boundary* (LPFB) in [24]. Assume that the boundary of P^i is determined by linking n anchor points with approximate geodesic curves, we compute the optimal planar angles α_i on the i-th anchor point by solving a constrained optimization problem as

$$\begin{array}{ll} & \min \sum_{i=1}^{n} \frac{1}{2}(\alpha_i - \theta_i)^2 \\ s.t. & n\pi - \sum_{i=1}^{n} \alpha_i \equiv 2\pi \\ & \sum_{i=1}^{n} l_i \cos\phi_i \equiv 0 \\ & \sum_{i=1}^{n} l_i \sin\phi_i \equiv 0 \end{array} \quad (2)$$

where θ_i is the surface angle of the patch at the i-th anchor point, and l_i denotes the length of the boundary curve between the i-th and the $(i+1)$-th anchor points on the patch. The computation is taken in the angle space by setting the angle variations as objective function to generate a planar domain with a shape similar to the patch's boundary in 3D. The first hard constraint is derived from the closed-path theorem to prevent self-intersection, and the latter two hard constraints are employed to ensure the generation of a closed boundary loop with

$$\phi_i = i\pi - \sum_{b=1}^{i} \alpha_i.$$

Once the 2D angles on anchor points are determined, we can easily locate the i-th anchor point of LPBD at

$$(\sum_{b=1}^{i} l_b \cos\phi_b, \sum_{b=1}^{i} l_b \sin\phi_b).$$

The LPBD can be computed very efficiently because for a patch circled by n boundary curves, only n variables plus three constraints are involved in the computation (i.e., a $(n+3)\times(n+3)$ linear equation system). This can be easily solved by the Quasi-Newton's method. Notice that we can directly compute the planar shape of a base domain by the lengths of its boundaries if the domain is triangular.

4 Boundary Swapping

Edge swapping (or called edge-flip) is a widely used operator to improve the quality of triangular meshes in mesh optimization (e.g., [25], [26], [27]); we employ a similar concept to optimize the shape similarity of common base domains. Different from the edge swapping operator, performing swapping on the curved boundaries of base domains is more difficult because of the following four reasons:

1) The shapes of domains on both the source and the target models need to be considered.
2) The shapes of 2D domains after swapping a curved boundary cannot be predicted by that of 2D domains before swapping.
3) The shape of a 2D domain after swapping could be degenerated, where the length of one boundary curve is longer than the sum of the other two so that the computed planar triangle could be an invalid LPBD.
4) The computation of measuring the distortion in cross-parameterization based on dense meshes is time-consuming.

Our target here is to exploit an effective metric that can be efficiently evaluated.

Before going to the study of finding a good metric, let's assume that such a metric for boundary swapping has been well defined on every boundary curve $\widetilde{v^a v^b}$, as $\Upsilon(\widetilde{v^a v^b})$. For two patches $\widetilde{\Delta} v^a v^b v^l$ and $\widetilde{\Delta} v^a v^b v^r$ defined on four anchor points v^a, v^b, v^l and v^r and sharing the common boundary curve $\widetilde{v^a v^b}$, applying the boundary swapping operator on them converts them into two new patches $\widetilde{\Delta} v^a v^r v^l$ and $\widetilde{\Delta} v^b v^l v^r$ (see Fig.6). The metric $\Upsilon(\widetilde{v^a v^b})$ returns a value to indicate the level of distortion reduction in cross-parameterization by swapping $\widetilde{v^a v^b}$ to $\widetilde{v^l v^r}$, the value of which is the greater the better. Based on this metric, a greedy algorithm can be developed to conduct boundary swapping by using a maximum heap keyed on $\Upsilon(\cdots)$. Pseudo-code of the basic algorithm is given in the Algorithm *GreedyBndSwapping*. Notice that, when considering the new boundary $\widetilde{v^l v^r}$, we need to prevent the paths from intersecting, blocking and being in wrong cyclical order. Moreover, we need to verify the validity of LPBD. If invalid LPBDs will be generated after swapping, the operation must be prevented. On the other hand, if a boundary swapping can convert invalid LPBDs into valid LPBD, it must have the highest priority to perform.

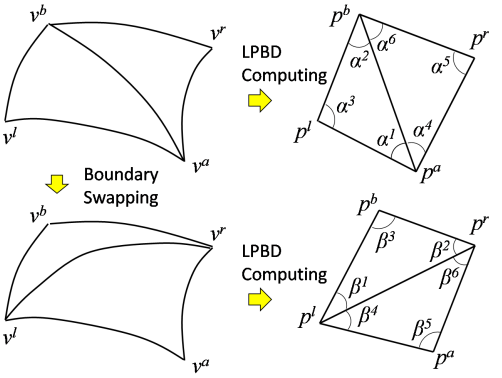

Fig. 6: The *boundary swapping* operator applied on a curved boundary edge $\widetilde{v^a v^b}$ shared by two patches $\widetilde{\Delta} v^a v^b v^l$ and $\widetilde{\Delta} v^b v^a v^r$ replaces $\widetilde{v^a v^b}$ by a new curve $\widetilde{v^l v^r}$ and therefore forms two new triangular patches $\widetilde{\Delta} v^a v^r v^l$ and $\widetilde{\Delta} v^b v^l v^r$ on the surface.

Algorithm 1 GreedyBndSwapping

```
Initialize an empty maximum heap H;
for every pair of v_s^a v_s^b ∈ M_s and v_t^a v_t^b ∈ M_t do
    Evaluate the metric Υ(v^a v^b) on them;
    Insert the curve v^a v^b into H when Υ(v^a v^b) > 0;
end for
while H is NOT empty do
    Remove a curved boundary edge v^a v^b from H;
    Apply the swapping operator on v^a v^b;
    for any of v^a v^l, v^a v^r, v^b v^l and v^b v^r do
        Evaluate the metric Υ(···) on the boundary curve;
        if the boundary curve is in H then
            Update its position in H;
        else
            Insert the curve into H when Υ(···) > 0;
        end if
    end for
end while
```

4.1 Metrics Based on Dense Meshes

A kind of obvious metrics for $\Upsilon(\widetilde{v^a v^b})$ is the ones measured by the deformation between the dense meshes

of $\widetilde{\Delta}v_s^a v_s^b v_s^l$ and $\widetilde{\Delta}v_s^a v_s^r v_s^b$ and their transformed shapes $\Gamma(\widetilde{\Delta}v_s^a v_s^b v_s^l)$ and $\Gamma(\widetilde{\Delta}v_s^a v_s^r v_s^b)$, which are defined by the cross-parameterization Γ. The metrics based on angle distortion or L^2-stretch on these dense meshes are presented below.

Angle Distortion

Borrowing the idea of Sheffer et al. from [28], for any triangle $f_i \in \widetilde{\Delta}v_s^a v_s^b v_s^l$ or $\widetilde{\Delta}v_s^a v_s^r v_s^b$, we define that angle distortion measures the change of its three angles $a_{i,1}$, $a_{i,2}$ and $a_{i,3}$ on the source model compared with the corresponding angles $a'_{i,1}$, $a'_{i,2}$ and $a'_{i,3}$ after applying the mapping Γ. If there are $|f|$ triangles in these two patches, the angle distortion on the patches can be calculated by

$$E_{ang}(\widetilde{v^a v^b}) = \frac{1}{3|f|}\sum_{i=1}^{|f|}\sum_{j=1}^{3}(\frac{a_{i,j}-a'_{i,j}}{a_{i,j}})^2 \quad (3)$$

which has a lower bound zero on examples without distortions. The boundary swapping metric based on the angle distortion is then defined by the difference between the angle distortions $E_{ang}(\widetilde{v^a v^b})$ and $E_{ang}(\widetilde{v^l v^r})$.

$$\Upsilon_{AD}(\widetilde{v^a v^b}) = E_{ang}(\widetilde{v^a v^b}) - E_{ang}(\widetilde{v^l v^r}) \quad (4)$$

The angle distortion for the whole model, $E_{ang}(M_s)$, can be evaluated in a way similar to Eq.(3) by including all triangles on M_s.

L^2-Stretch

Another widely used method for measuring the distortion of surface parameterization is the L^2-stretch introduced by Sander et al. in [14]. For a triangle $f_s \in M_s$ and its corresponding shape defined by the mapping Γ as $\Gamma(f_s)$, the L^2-stretch is defined by computing the Eigen values of the Jacobian formed by partial derivatives of a unique affine mapping between f_s and $\Gamma(f_s)$. To ease the computation, for f_s with three vertices $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ in $\Re^3$, we define a local planar frame on $\Gamma(f_s)$ to obtain the planar coordinates of its three vertices as (s_k, t_k) $(k = 1, 2, 3)$. Then, the L^2-norm defined on the triangle f_s based on the cross-parameterization is

$$L^2(f_s) = \sqrt{(\tau_s \cdot \tau_s + \tau_t \cdot \tau_t)/2} \quad (5)$$

with

$$\tau_s = ((t_2 - t_3)\mathbf{p}_1 + (t_3 - t_1)\mathbf{p}_2 + (t_1 - t_2)\mathbf{p}_3)/(2A(\Gamma(f_s)))$$
$$\tau_t = ((s_2 - s_3)\mathbf{p}_1 + (s_3 - s_1)\mathbf{p}_2 + (s_1 - s_2)\mathbf{p}_3)/(2A(\Gamma(f_s)))$$

and $A(\cdots)$ defining the area of a triangle. Therefore, the normalized L^2-stretch of all triangles in a particular region Ω is

$$E_{L^2}(\Omega) = \sqrt{\frac{\sum_{f_i\in\Omega} A(\Gamma(f_i))\sum_{f_i\in\Omega}((L^2(f_i))^2 A(f_i))}{(\sum_{f_i\in\Omega} A(f_i))^2}}, \quad (6)$$

which has a lower bound of 1.0. The boundary swapping metric based on the L^2-stretch on dense meshes is then defined on all triangles $\widetilde{\Delta}v_s^a v_s^b v_s^l$, $\widetilde{\Delta}v_s^a v_s^r v_s^b$, $\widetilde{\Delta}v^a v^r v^l$ and $\widetilde{\Delta}v^b v^l v^r$ as

$$\Upsilon_{L^2}(\widetilde{v^a v^b}) = E_{L^2}(\widetilde{v^a v^b}) - E_{L^2}(\widetilde{v^l v^r}). \quad (7)$$

Similarly, the L^2-stretch of the whole model, $E_{L^2}(M_s)$, can also be evaluated by including all triangles on M_s in the computation of Eq.(6).

Both of these two metrics, $E_{ang}(M_s)$ and $E_{L^2}(M_s)$, reflect the distortion of a cross-parameterization in their own aspects. An ideally optimized cross-parameterization should reduce both of them.

4.2 Metrics Based on Shape of Domains

The main problem of directly applying these metrics to the domain optimization for cross-parameterization is that the evaluation of them is generally time-consuming (especially when the given models have very dense triangular meshes). Therefore, we have studied some other metrics which are defined directly on the LPBD signature and wish to find one that well balances the quality of results and the speed of computation. As the metrics, $\Upsilon_{AD}(\widetilde{v^a v^b})$ and $\Upsilon_{L^2}(\widetilde{v^a v^b})$, are tightly coupled with the quality of cross-parameterization, the results of applying them in the Algorithm *GreedyBndSwapping* are served as benchmarks for the selection of a good metric below.

Area Similarity

The first tested metric is stimulated by the analysis given in [24] that a surface parameterization with small distortion usually has small area variations (e.g., flattening a developable mesh surface). Therefore, an area similarity metric is defined as

$$\Upsilon_{area}(\widetilde{v^a v^b}) = E_{area}(\widetilde{v^l v^r}) - E_{area}(\widetilde{v^a v^b}) \quad (8)$$

where

$$E_{area}(\widetilde{v^c v^d}) = (\psi(P_s^l) - \psi(P_t^l))^2 + (\psi(P_s^r) - \psi(P_t^r))^2$$

with P^l and P^r denoting the surface patches on the left and the right of the curved boundary $\widetilde{v^c v^d}$ in the patch layout. The function $\psi(P^i)$ returns a value based on the area difference between a surface patch, P^i, and its LPBD, B^i.

$$\psi(P^i) = \begin{cases} A(P^i)/A(B^i) & (A(P^i) \geq A(B^i)) \\ -A(B^i)/A(P^i) & (A(P^i) < A(B^i)) \end{cases}$$

$A(B^i)$ is the area of a base domain, which can be directly calculated by *Heron's formula* using the lengths of boundary curves on P^i. However, it should be noted that the Heron's formula which is base on lengths may fail if the length of a boundary curve is larger than the sum of other two boundary curves' lengths – i.e., a degenerated case happens. For a degenerated case, we simply define the area of B^i with a very small value (e.g., 10^{-8}).

Domain Angle Distortion

In order to measure the similarity of triangular domains, one of the methods is to calculate the difference of corresponding angles in the planar domains, B_s^i and B_t^i. Therefore, using the notation illustrated in Fig.6 as the angles in LPBDs, we can define the following metric to govern the swapping operator.

$$\Upsilon_{DAD}(\widetilde{v^a v^b}) = \sum_{k=1}^{6}(\frac{\alpha_s^k - \alpha_t^k}{\alpha_s^k})^2 - \sum_{k=1}^{6}(\frac{\beta_s^k - \beta_t^k}{\beta_s^k})^2 \quad (9)$$

Again, a curved boundary edge $\widetilde{v^a v^b}$ is inserted into the priority list only when $\Upsilon_{DAD}(\widetilde{v^a v^b}) > 0$.

Domain L^2-Stretch

Another method that measures the similarity between the triangular domains is through the L^2-stretch which

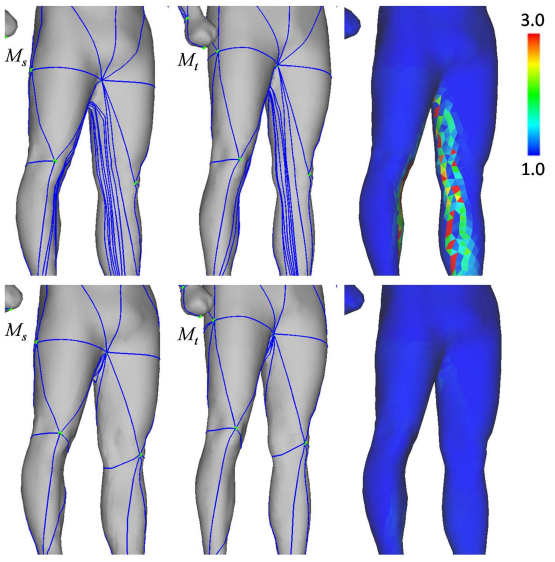

Fig. 7: An example to illustrate the fact that highly irregular shapes in LPBDs will also lead to high distortion in cross-parameterization (top row). The color map shows the distribution of L^2-stretch. The distortion can be reduced by using domains with more regular shapes (bottom row).

has been used in section 4.1. Differently, the L^2-stretch here measures the difference between a planar triangular domain of the source model and its corresponding domain of the target model. For example, $L^2(\Delta p^i p^j p^k)$ gives the L^2-stretch between $\Delta p_s^i p_s^j p_s^k$ and $\Delta p_t^i p_t^j p_t^k$. For triangular domains adjacent to a curved boundary edge (as illustrated in Fig.6), the metric of domain L^2-Stretch is defined as

$$\Upsilon_{DL^2}(\widetilde{v^a v^b}) = E_{L^2}(\widetilde{v^a v^b}) - E_{L^2}(\widetilde{v^l v^r}) \tag{10}$$

with

$$E_{L^2}(\widetilde{v^a v^b}) = L^2(\Delta p^a p^b p^l) + L^2(\Delta p^b p^a p^r), \tag{11}$$

$$E_{L^2}(\widetilde{v^l v^r}) = L^2(\Delta p^a p^l p^r) + L^2(\Delta p^b p^r p^l). \tag{12}$$

Shorter Diagonal Length

During our experimental tests, we find that the distortion in cross-parameterization will still be high if the shapes of LPBDs of two models are similar (not exactly same) to each other but are far from regular triangles. See Fig.7 for an extreme example to demonstrate this observation. Therefore, the fourth metric tries to make the common base domains on both the source and the target models become 'regular' triangles. A rule employed in the mesh optimization to generate regular triangles is that a diagonal edge will be swapped if the swapped one is shorter. The metric Υ_{SDL} below is introduced for the same purpose.

$$\Upsilon_{SDL}(\widetilde{v^a v^b}) = 2 - \left(\frac{\|\widetilde{v_s^l v_s^r}\|}{\|\widetilde{v_s^a v_s^b}\|} + \frac{\|\widetilde{v_t^l v_t^r}\|}{\|\widetilde{v_t^a v_t^b}\|}\right), \tag{13}$$

where $\|\widetilde{v^a v^b}\|$ denotes the length of the curve linking the anchor points v^a and v^b, while $\|\widetilde{v^l v^r}\|$ denotes the length of the curve linking the anchor points v^l and v^r. In this metric, the length changes of the boundary curves on both the source and the target models are considered. Swapping is prohibited if either side has become much worse after swapping – e.g., when $\|\widetilde{v_s^l v_s^r}\| < \|\widetilde{v_s^a v_s^b}\|$ but $\|\widetilde{v_t^l v_t^r}\| > 2\|\widetilde{v_t^a v_t^b}\|$. Similarly if the curves are elongated on both M_s and M_t, the swapping is also prevented since the value of Υ_{SDL} is negative. When the patches P_s^i and P_t^i for the common based domains on both models approaches regular triangles, the shapes of their corresponding LPBDs, B_s^i and B_t^i, should be similar to each other as well (i.e., the distortion in the mapping Γ_{st} is reduced).

Greatest Angle Reduction

In mesh optimization, another widely used rule which detects the possibility of applying the edge swapping operator is based on checking whether the greatest angle in two triangles adjacent to the edge is reduced. Assuming that α^k and β^k ($k = 1, \ldots, 6$) are the six angles in the two LPBD triangles adjacent to the edge $\widetilde{v^a v^b}$ and the swapped one $\widetilde{v^l v^r}$ respectively – see Fig.6 for an illustration, we define the greatest angle reduction metric as

$$\Upsilon_{GAR}(\widetilde{v^a v^b}) = \max_k\{\alpha_s^k, \alpha_t^k\} - \max_k\{\beta_s^k, \beta_t^k\}. \tag{14}$$

The swapping is applied only when $\Upsilon_{GAR}(\widetilde{v^a v^b}) > 0$. Similar to using Υ_{SDL} to govern boundary swapping, the metric Υ_{GAR} also tends to make the shapes of based domains on both models approach regular triangles. Therefore, it can reduce the distortion in cross-parameterization. It should be remarked that, all the angles of LPBDs in 2D are calculated by the length of the boundary curves directly using the *Law of Cosine*. For the degenerated cases (e.g., the length of one side is longer than the sum of the other two), the computed value ς for arccos by the *Law of Cosine* does not fall in the range $[-1, 1]$. For these cases, we simply assign the angle as $-\varsigma\pi$ ($\forall\, \varsigma < -1$) or $(1 - \varsigma)\pi$ ($\forall\, \varsigma > 1$).

Analysis and discussion of all metrics

In order to select the 'best' metric for the *GreedyBndSwapping* algorithm whose purpose is to reduce the distortion in cross-parameterization through optimizing the shape of common base domains, we apply the algorithm to a variety of models by using all above metrics. The distortion in cross-parameterization based on the resultant base domains is measured by warping a source model M_s to the shape of the target model M_t and calculating the angle distortion $E_{ang}(M_s)$ and the L^2-stretch $E_{L^2}(M_s)$. Figure 8 shows the bar charts for the statistics of $E_{ang}(M_s)$ and $E_{L^2}(M_s)$ on different models.

When analyzing the results, several interesting phenomena can be observed.

- Firstly, the evaluation results obtained from using $E_{ang}(M_s)$ and $E_{L^2}(M_s)$ to measure the distortion in cross-parameterization are not consistent with each other. Take David-egea models as an example, after using the angle distortion metric Υ_{AD} based on dense meshes to optimize the common base domains, the L^2-stretch, $E_{L^2}(M_s)$, becomes even worse than the cross-parameterization before domain optimization. Another example is the models of men. Using the L^2-stretch metric Υ_{L^2} to govern the boundary swapping turns out have enlarged the global angle distortion, $E_{ang}(M_s)$.
- Secondly, the results obtained from using the L^2-stretch metric Υ_{L^2} are better than that obtained from using the metric Υ_{ang} in terms of the L^2-stretch error, $E_{L^2}(M_s)$. However, using Υ_{ang} does not always give the result

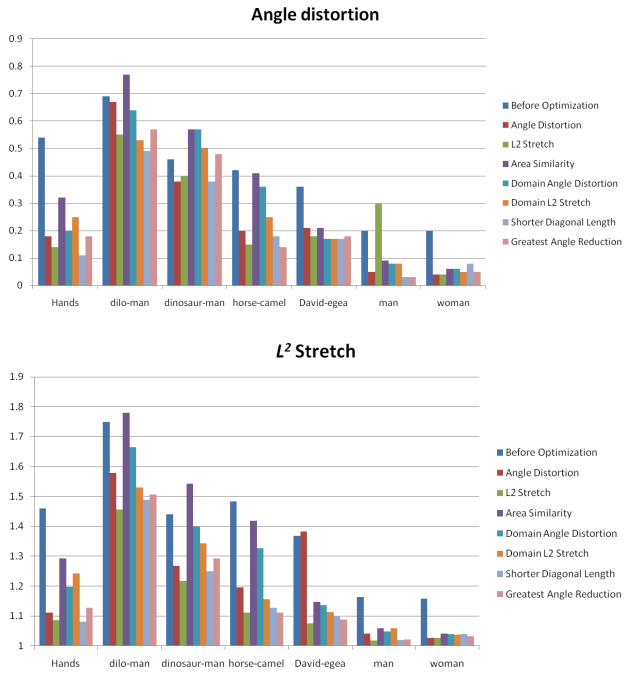

Fig. 8: The statistics of using different metrics in the greedy algorithm when applying the *boundary swapping* operator to improve the shape similarity of common base domains and thus reduce the distortion in cross-parameterization. The top bar chart shows the angle distortion $E_{ang}(M_s)$, and the bottom one gives the results of $E_{L^2}(M_s)$.

that a smaller value is on $E_{ang}(M_s)$ than the results obtained by using Υ_{L^2} in the optimization.

- Thirdly, when using the metrics defined on the dissimilarity of LPBDs to govern optimization, Υ_{DL^2} results in cross-parameterizations with less distortion than Υ_{area} and Υ_{DAD} – based on the evaluations of both $E_{L^2}(M_s)$ and $E_{ang}(M_s)$.
- Lastly, after considering the regularity on the shape of base domains, the optimizations using metrics Υ_{SDL} and Υ_{GAR} always give better results than one using the metrics, Υ_{area}, Υ_{DAD} and Υ_{DL^2}, that consider only the dissimilarity. Among them, using Υ_{SDL} generates better results in most cases.

Based on these observations, we can conclude that the metric Υ_{SDL} is the best choice (based on the results of our experimental tests) to reduce the distortion in cross-parameterization under the greedy boundary swapping algorithm. Furthermore, the metric Υ_{DL^2} gives better results in measuring the similarity of base domains than Υ_{DAD}.

4.3 Balanced Algorithm

Using the metrics defined on the shape of LPBDs instead of dense meshes can gain some speedup in the optimization procedure; however, we find that a more significant speedup can be obtained if some of the swapping operations on boundary curves are skipped. Specifically, we skip the operations on those 'similar-enough' common base domains. In detail, the balanced algorithm for boundary curve swapping consists of three major steps.

- **Step 1)** The domain L^2-stretch, $E_{DL^2}(\widetilde{v^a v^b})$, defined on every curved boundary edge $\widetilde{v^a v^b}$ is first evaluated by Eq.(11).
- **Step 2)** Among all curved boundaries, only those having $E_{DL^2} > 2.0 + (1-\varphi)\max\{E_{DL^2} - 2.0\}$ are defined as active boundary curves. The golden ratio, i.e., 0.618, is selected for φ.
- **Step 3)** Applying the greedy boundary swapping algorithm only on the active boundary curves and the degenerated cases to optimize the common base domains by using the metric Υ_{SDL}.

This balanced algorithm gives a good trade-off between the quality and the speed. Around 5 to 15 times speedup can be gained – more discussion on the results can be found in section 6.

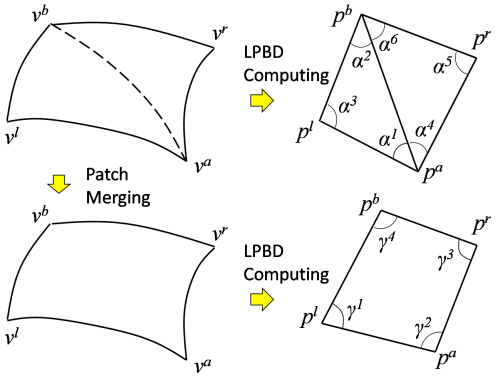

Fig. 9: The *patch merging* operator applied on a curved boundary edge $\widetilde{v^a v^b}$ (the one shown in dash line) adjacent to two triangular patches.

Algorithm 2 GreedyPatchMerging

```
Initialize an empty maximum heap H_m;
for every pair of ṽ_s^a ṽ_s^b ∈ M_s and ṽ_t^a ṽ_t^b ∈ M_t do
    Evaluate the metric Υ_merge(ṽ^a ṽ^b) on them;
    Insert ṽ^a ṽ^b into H_m when Υ_merge(ṽ^a ṽ^b) > 0;
end for
while H is NOT empty do
    Remove a curved edge ṽ^a ṽ^b from H_m;
    if Υ_merge(ṽ^a ṽ^b) > 0 then
        Apply the patch merging operator on ṽ^a ṽ^b;
        Remove the edges ṽ^a ṽ^l, ṽ^a ṽ^r, ṽ^b ṽ^l and ṽ^b ṽ^r from
        H_m if any of them has already been in H_m;
    end if
end while
```

5 Patch Merging

The patch merging operator can be applied to further reduce the distortion in cross-parameterization. Without loss of generality, when applying the patch merging operator to a curved boundary edge $\widetilde{v^a v^b}$ adjacent to two triangular patches $\widetilde{\Delta} v^a v^b v^l$ and $\widetilde{\Delta} v^b v^a v^r$, the triangular patches are merged into a quadrilateral patch $\widetilde{\Diamond} v^a v^r v^b v^l$. Although merging patches into n-sided patches ($n > 4$) could possibly further reduce the distortion in cross-parameterization, a polygon with $n > 4$ is easier to have concave corners – using which as the 2D domain may violate the correctness

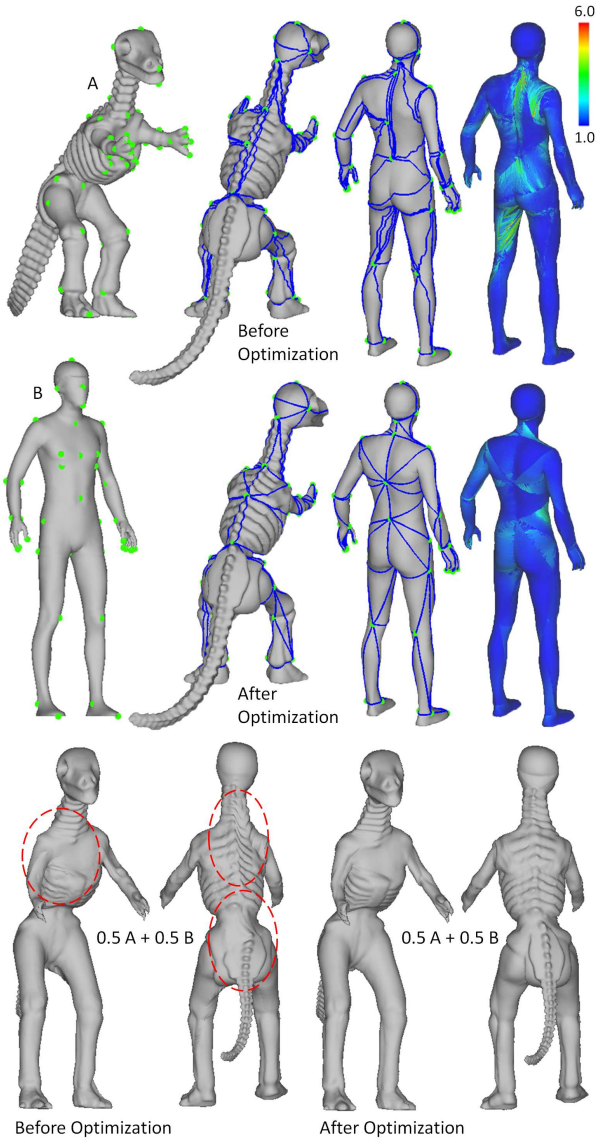

Fig. 10: The shape of a blended model of a dinosaur and a man can be improved by the optimized cross-parameterization. (Top row) The initial common base domains generated by [4], and (Middle row) the optimized base domains – the L^2-stretches are shown by the color map. (Bottom row) The blending results obtained by the cross-parameterizations before vs. after domain optimization, where the red dash lines circle the unreal distortions caused by the large stretch in cross-parameterization.

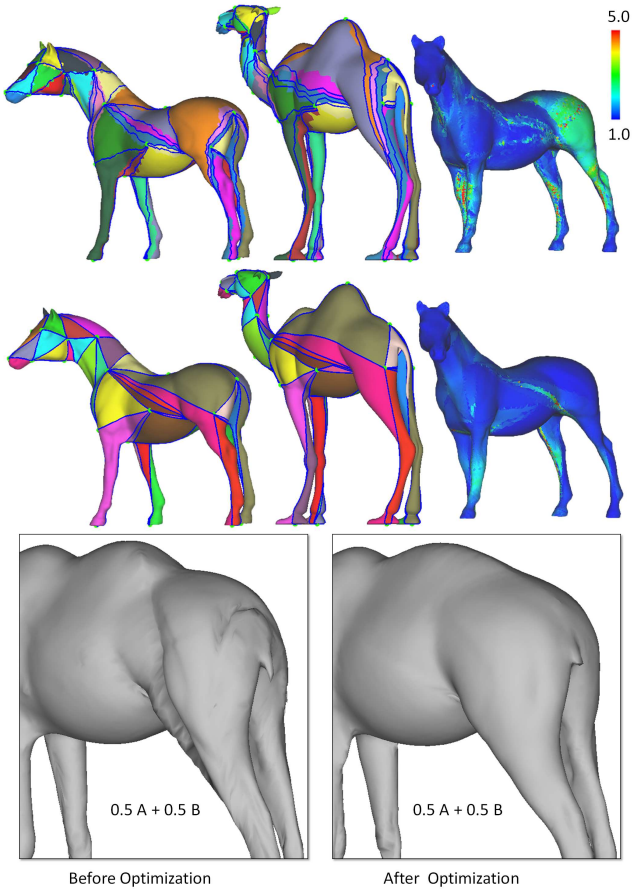

Fig. 11: The base domains of horse-camel models before (top row) vs. after (middle row) optimization, and their corresponding L^2-stretches in the color map. The triangles belonging to different domains are displayed in different colors. Note that, the smoothing technique presented in [4] has already been applied to reduce the distortion. Bottom row shows the shape interpolation results before (left) and after (right) domain optimization.

of cross-parameterization (see the discussion in detail below). Thus, we choose not to do that in order to ease the implementation and shorten the computing time.

The metric for governing the order of patch merging is defined based on the shape similarity of the LPBDs measured by angle distortions.

$$\Upsilon_{merge}(\widetilde{v^a v^b}) = \frac{1}{6}\sum_{k=1}^{6}|\alpha_s^k - \alpha_t^k| - \frac{1}{4}\sum_{k=1}^{4}|\gamma_s^k - \gamma_t^k|, \quad (15)$$

where α^k and γ^k are angles on the LPBDs before and after merging respectively (as illustrated in Fig.9). More than that, the metric is set to be $-\infty$ or removed from heap to prevent patch merging operation when any of the following cases occurs.

- Any of the patches adjacent to the curved boundary $\widetilde{v^a v^b}$ is a quadrilateral patch.
- There is one edge of the to-be-merged quadrilateral patch longer than the sum of the other three – this will lead to a degenerated result when computing the LPBD of the patch.
- $\exists \gamma^k > \pi$ among $k = 1, \ldots, 4$. This case is prevented since any angle greater than π will make LPBD a concave polygon, which will generate self-overlaps on the 3D-to-2D parameterization such that the bijective mapping for the cross-parameterization cannot be established. Detailed discussion about self-overlapping in mesh parameterizations can be found in [29].

Again, we use the greedy strategy with the metric $\Upsilon_{merge}(\widetilde{v^a v^b})$ to perform patch merging. See the pseudo-code in Algorithm *GreedyPatchMerging* for more information.

6 Results and Discussion

We have implemented the proposed algorithm in C++ plus OpenGL. All the experimental tests shown in this paper are

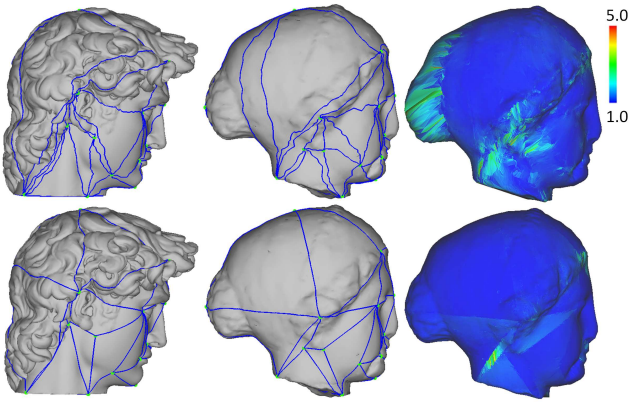

Fig. 12: An example of head models – David and Egea. (Top) The cross-parameterization by [4]. (Bottom) The result after optimizing the common base domains by our approach.

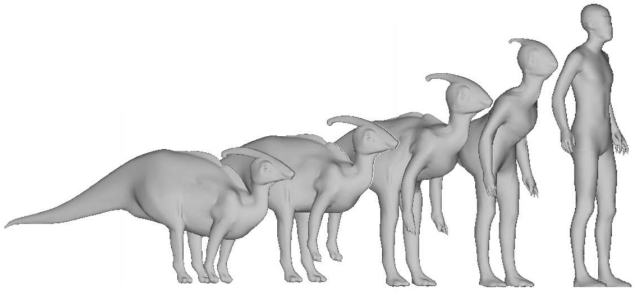

Fig. 13: Morphing between a dilo model and a mannequin can be established by the cross-parameterization with optimized common base domains.

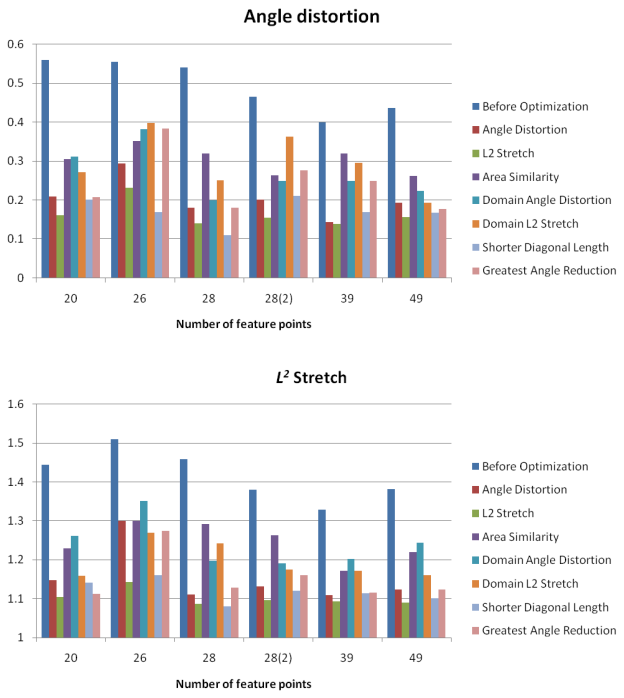

Fig. 14: The statistics of results obtained by using different metrics in the *GreedyBndSwapping* algorithm on the hand example shown in Fig.3 but having different sets of feature points. The top bar chart shows the angle distortion $E_{ang}(M_s)$, and the bottom one gives the results of $E_{L^2}(M_s)$.

run on a standard PC with Intel Core 2 Duo CPU E6750 at 2.67GHz plus 2GB RAM.

The first example is a pair of hand models. From the blending result shown in Fig.1, we can see that the unwanted distortion in the blending can be eliminated after optimizing the common base domains by our method. More examples, a dinosaur-man pair, a horse-camel pair, a pair of head models – David and Egea, and a dilo-man pair, are shown in Figs.10-13. The results are quite encouraging. The distortions in cross-parameterization (measured by the L^2-stretch [14] and displayed by the color map) on all pairs of models are significantly reduced after applying the proposed domain optimization approach. Note that, all results shown in Figs.10-13 are generated by the balanced boundary swapping algorithm in optimization. The initial base domains are generated by our implementation of approach in [4].

The global angle distortion $E_{ang}(M_s)$ and the L^2-stretch $E_{L^2}(M_s)$ on the resultant models are listed in Table 1 and compared between the original cross-parameterization results of [4] and the optimized ones by the metrics defined on the dense meshes, Υ_{ang} and Υ_{L^2}. The balanced boundary swapping algorithm followed by the *GreedyPatchMerging* algorithm generates results that are similar to that of the optimization which is based on the 'real' parameterization (i.e., the metrics evaluated on the dense meshes). However, the speed of the balanced algorithm is about 5 to 15 times faster (see the statistics shown in Table 1). In short, the LPBD signature newly introduced in this paper and the proposed balanced algorithm can efficiently improve the quality of cross-parameterization results by optimizing the shape of common base domains, which are constrained by the anchor points.

In the boundary swapping algorithm, we do not explicit prevent letting the boundary curves pass through the highly stretched area; however, as it will enlarge the dissimilarity on the LPBD signatures, such cases will be avoided automatically by the metrics according to LPBDs.

Another interesting study is about how the positions of feature points (constraints) affect the distortion in cross-parameterization. See the statistics in Fig.14 on the same pair of hand models shown in Fig.3 before, the level of distortion could change significantly when different sets of feature points are specified. Such variation occurs even if the same number of feature points are used. To compare with the results generated by [4], we apply our approach on a pair of head models provided by authors of [4] on their webpage (see Fig.15). We tried several sets of feature points on this pair of models, and our method can always further reduce the overall L^2-stretch, $E_{L^2}(M_S)$, from 1.164 to 1.043, 1.037 and 1.035 respectively. The color maps for illustrating the value of L^2-stretch are also shown in Fig.15. In these tests, as the input models are not provided by the authors of [4], we can only work on the remeshed models (i.e., using the output of [4] as our input). Although this is not a direct comparison to the original implementation in [4], it still somewhat proves the performance of our approach.

In order to further verify the performance of our optimization algorithm, we change the way to construct initial common base domains in [4], where the query is based

TABLE 1: Comparisons between different optimization methods.

Models (M_s/M_t)		Hand1/2	Dilo/Man	Dinosaur/Man	Horse/Camel	David/Egea	Male1/2	Female1/2
Sizes (#v)		14k / 1,515	27k / 11k	56k / 11k	20k / 9,770	50k / 8,268	3,856 / 8,950	11k/2230
Feature points #		28	52	53	29	24	37	50
Original [4]	$E_{L^2}(M_s)$	1.459	1.750	1.441	1.483	1.368	1.163	1.158
	$E_{ang}(M_s)$	0.54	0.69	0.46	0.42	0.36	0.20	0.20
By Υ_{AD}	$E_{L^2}(M_s)$	1.111	1.579	1.267	1.195	1.383	1.042	1.027
	$E_{ang}(M_s)$	0.18	0.67	0.38	0.20	0.21	0.05	0.04
	Time (sec.)	159	200	285	183	162	180	96
By Υ_{L^2}	$E_{L^2}(M_s)$	1.087	1.456	1.217	1.112	1.076	1.018	1.026
	$E_{ang}(M_s)$	0.14	0.55	0.40	0.15	0.18	0.3	0.04
	Time (sec.)	149	286	345	157	208	158	90
Balanced	$E_{L^2}(M_s)$	1.142	1.522	1.298	1.229	1.131	1.039	1.040
	$E_{ang}(M_s)$	0.23	0.47	0.41	0.34	0.19	0.05	0.07
	Time (sec.)	17	52	51	9	23	24	7
+ Merging	$E_{L^2}(M_s)$	1.136	1.484	1.287	1.196	1.130	1.039	1.038
	$E_{ang}(M_s)$	0.23	0.46	0.4	0.28	0.18	0.05	0.08
	Time (sec.)	19	58	61	11	25	26	7

* [4] is the original cross-parameterization before re-meshing. The reported time is the processing time of domain optimization.

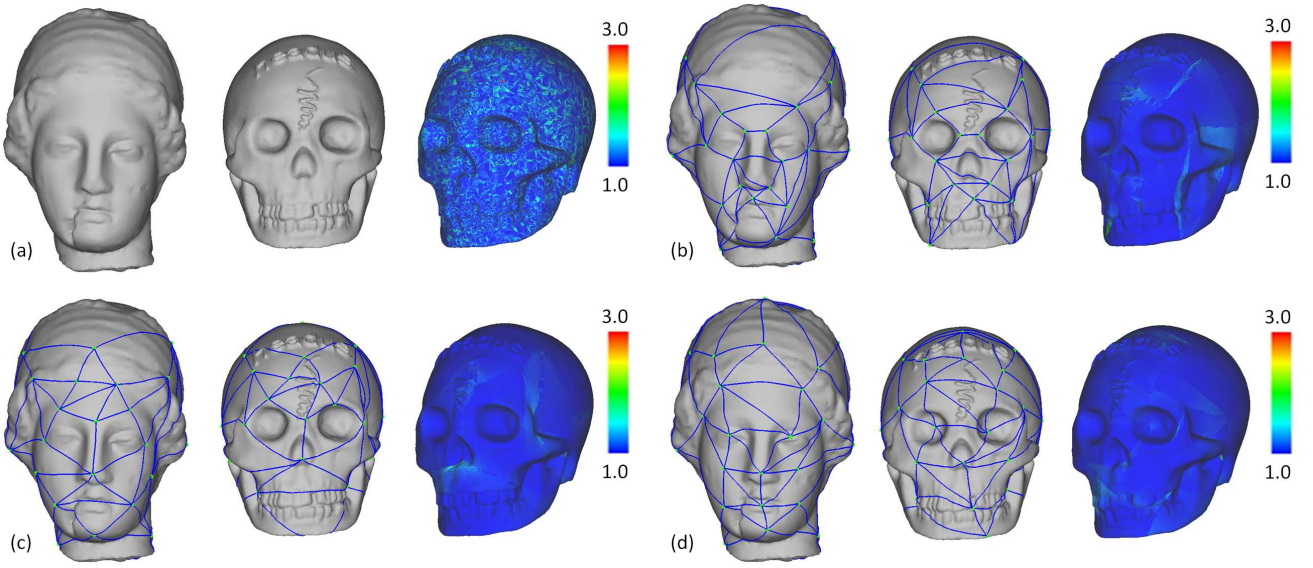

Fig. 15: Applying our approach on the result of cross-parameterization generated by [4], the L^2-stretch can still be further reduced: (a) the result of cross-parameterization downloaded from the webpage of authors [4], (b) our result generated on the set of feature points shown in the left - the L^2-stretch is $E_{L^2}(M_S) = 1.043$, (c) the result on another set of feature points with $E_{L^2}(M_S) = 1.037$, and (d) the result with 1.035 being the value of L^2-stretch.

on computing the shortest path between feature points by the Dijkstra's algorithm. Here, the priority query is keyed according to the sum of Geodesic distances between feature points (see Fig.16 for an example), and we compute the Geodesic distances approximately by the method in [22]. It is found that using Geodesic distances can somewhat improve the shape of initial common base domains thus also the cross-parameterization. However, our algorithm can still further reduce the distortion by optimizing the topology of base domains. Table 2 gives the related statistics.

Lastly, we test the results of cross-parameterization constructed by our method in two applications. The first application shown in Fig.17 is to establish a human body database from the scanned 3D models. After reconstructing a mesh surface from the scanned point cloud by [30], we wish to generate consistent connectivity for all human models stored in the database. More than that, the semantic features predefined on the template model must be able to be easily transfered to all the models by the consistent connectivity. As shown in Fig.17, if large distortions are embedded in the cross-parameterization, the semantic feature curves will be highly distorted, which is intolerable for downstream applications (e.g., mannequin fabrication). The second application is to apply the optimized cross-parameterization in design automation of apparel products. After constructing correspondences between the triangles on the source model M_s and the target model M_t, we can use some deformation transformation [18] techniques or spatial warping techniques [31], [32] to deform an apparel product designed for the model M_s to a new shape fit for the new model M_t. However, if large distortions are embedded in the cross-parameterization, the resultant model of the apparel product will have highly uneven distortions – i.e., the shape of the new product is unacceptable. From Fig.18, we

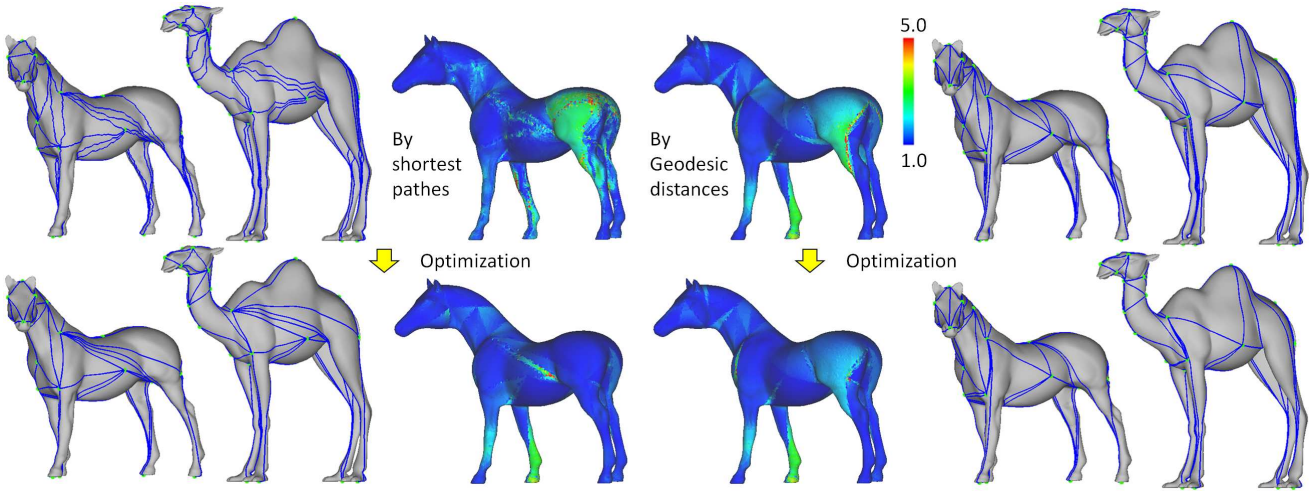

Fig. 16: The initial common base domains constructed according to Geodesic distances instead of the lengths of shortest paths (as [4]) can still be improved by our optimization approach, which can be seen from the color map for L^2-stretches in cross-parameterization. The statistic data has been listed in Table 2.

TABLE 2: Statistics of distortions on the initial base domains constructed by Geodesic distances and the optimization results.

Models (M_s/M_t)		Hand1/2	Horse/Camel
Feature points #		28	29
Geodesic distances	$E_{L^2}(M_s)$	1.202	1.305
	$E_{ang}(M_s)$	0.28	0.40
	Time† (sec.)	728	956
By Υ_{AD}	$E_{L^2}(M_s)$	1.112	1.160
	$E_{ang}(M_s)$	0.17	0.21
	Time (sec.)	125	174
By Υ_{L^2}	$E_{L^2}(M_s)$	1.090	1.109
	$E_{ang}(M_s)$	0.13	0.15
	Time (sec.)	102	147
Balanced	$E_{L^2}(M_s)$	1.150	1.238
	$E_{ang}(M_s)$	0.20	0.34
	Time (sec.)	13	11
+ Merging	$E_{L^2}(M_s)$	1.144	1.212
	$E_{ang}(M_s)$	0.19	0.33
	Time (sec.)	14	13

†Construction according to the Geodesic distances takes a very long time to obtain the initial base domains.

can find that the shape of an automatically designed dress for a new client (i.e., M_t) has been improved to an acceptable level by using the optimized cross-parameterization.

6.1 Limitation

The main limitation of the approach proposed in this paper is that, like other greedy algorithm based mesh optimization techniques, the result of this approach may be only a local optimum as it depends heavily on how good the given common base domains (generated by [4]) are. Some topological obstacles may prevent the algorithm from generating a further optimized common base domains. We plan to develop a new common base domain construction method to generate common base domains which have better shape similarities from the beginning of domain construction. This will be our near future work.

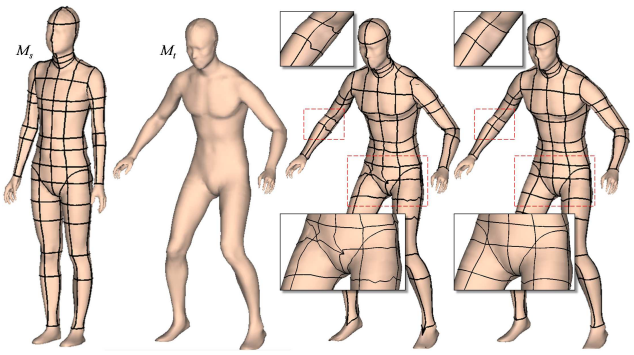

Fig. 17: An application of data preparation for establishing human body database. From left to right, the given source model M_s served as a template model, the input mesh surface as a target model M_t, the result based on the original cross-parameterization [4] (with unwanted distortion on the semantic feature curves), and the result based on optimized domains.

7 Conclusion

We present a new signature, *Length-Preserved Base Domain* (LPBD), in this paper for measuring the level of stretch between surface patches in cross-parameterization. This new signature well balances the accuracy of measurement and the computational speed. A set of metrics have been studied and compared to find a good criterion to govern the optimization of common base domains. A greedy optimization algorithm is adopted to reduce the distortion in cross-parameterization by repeatedly applying two operators, boundary swapping and patch merging, to the input base domains. A variety of models have been tested to demonstrate the functionality of our approach. Experimental tests prove that our method can efficiently and effectively improve the quality of cross-parameterization through optimizing the connectivity of the common base domains, which are constrained by the anchor points.

Acknowledgments

The authors would like to thank the valuable comments given by the reviewers. The work presented in this paper is

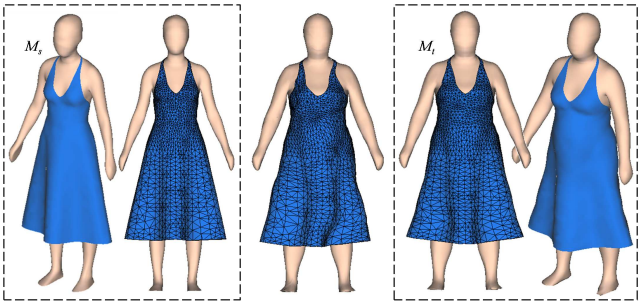

Fig. 18: An application of the design automation for apparel products. (Left) The source model M_s wearing a well designed dress. (Middle) A new model M_t and the dress warped based on the cross-parameterization between M_s and M_t – uneven distortions are generated on the warped dress which are caused by the large distortion in the cross-parameterization. (Right) The distortion becomes smooth on the deformed dress based on the optimized cross-parameterization.

partially supported by the HKSAR Research Grants Council (RGC) General Research Fund (GRF) – CUHK/417508 and CUHK/417109. Thanks Chen et al. [33], AIM@SHAPE Shape Repository, and Image-based 3D Models Archive for sharing some 3D models used in this paper.

REFERENCES

[1] 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*, vol. 14, no. 4, pp. 805–819, 2008.

[2] N. Ray, W.C. Li, B. Lévy, A. Sheffer, and P. Alliez, "Periodic global parameterization," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1460–1485, 2006.

[3] A. Khodakovsky, N. Litke, and P. Schröder, "Globally smooth parameterizations with low distortion," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, New York, NY, USA, 2003, pp. 350–357, ACM.

[4] V. Kraevoy and A. Sheffer, "Cross-parameterization and compatible remeshing of 3d models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 861–869, 2004.

[5] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe, "Inter-surface mapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 870–877, 2004.

[6] M.S. Floater, "Parametrization and smooth approximation of surface triangulations," *Comput. Aided Geom. Des.*, vol. 14, no. 3, pp. 231–250, 1997.

[7] E. Praun, W. Sweldens, and P. Schröder, "Consistent mesh parameterizations," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2001, pp. 179–184, ACM.

[8] M.S. Floater, "Mean value coordinates," *Comput. Aided Geom. Des.*, vol. 20, no. 1, pp. 19–27, 2003.

[9] E. Praun and H. Hoppe, "Spherical parametrization and remeshing," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, New York, NY, USA, 2003, pp. 340–349, ACM.

[10] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1995, pp. 173–182, ACM.

[11] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, "MAPS: multiresolution adaptive parameterization of surfaces," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1998, pp. 95–104, ACM.

[12] N. Pietroni, M. Tarini, and P. Cignoni, "Almost isometric mesh parameterization through abstract domains," *IEEE Trans. on Vis. and Comp. Graph.*, vol. 16, no. 4, pp. 621–635, 2010.

[13] A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 2, pp. 105–171, 2006.

[14] P.V. Sander, J. Snyder, S.J. Gortler, and H. Hoppe, "Texture mapping progressive meshes," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2001, pp. 409–416, ACM.

[15] Y. Wang, C.C.L. Wang, and M.M.F. Yuen, "Duplicate-skins for compatible mesh modelling," in *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, New York, NY, USA, 2006, pp. 207–217, ACM.

[16] L. Markosian, J.M. Cohen, T. Crulli, and J. Hughes, "Skin: a constructive approach to modeling free-form shapes," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1999, pp. 393–400, ACM Press/Addison-Wesley Publishing Co.

[17] 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.*, 2010, accepted.

[18] R.W. Sumner and J. Popović, "Deformation transfer for triangle meshes," in *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, pp. 399–405, ACM.

[19] E. Zhang, K. Mischaikow, and G. Turk, "Feature-based surface parameterization and texture mapping," *ACM Trans. Graph.*, vol. 24, no. 1, pp. 1–27, 2005.

[20] J. Bennett, V. Pascucci, and K. Joy, "A genus oblivious approach to cross parameterization," *Comput. Aided Geom. Des.*, vol. 25, no. 8, pp. 592–606, 2008.

[21] X. Li, X. Gu, and H. Qin, "Surface mapping using consistent pants decomposition," *IEEE Trans. on Vis. and Comp. Graph.*, vol. 15, no. 4, pp. 558–571, 2009.

[22] C.C.L. Wang, "Cybertape: an interactive measurement tool on polyhedral surface," *Computers & Graphics*, vol. 28, no. 5, pp. 731–745, 2004.

[23] W.-C. Li, B. Levy, and J.-C. Paul, "Mesh editing with an embedded network of curves," in *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*, Washington, DC, USA, 2005, pp. 62–71, IEEE Computer Society.

[24] C.C.L. Wang, "Computing length-preserved free boundary for quasi-developable mesh segmentation," *IEEE Trans. on Vis. and Comp. Graph.*, vol. 14, no. 1, pp. 25–36, 2008.

[25] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1993, pp. 19–26, ACM.

[26] V. Surazhsky and C. Gotsman, "Explicit surface remeshing," in *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, 2003, pp. 20–30, Eurographics Association.

[27] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, New York, NY, USA, 2004, pp. 185–192, ACM.

[28] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, "ABF++: fast and robust angle based flattening," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 311–330, 2005.

[29] Z. Karni, C. Gotsman, and S.J. Gortler, "Free-boundary linear parameterization of 3d meshes in the presence of constraints," in *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*, Washington, DC, USA, 2005, pp. 268–277, IEEE Computer Society.

[30] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, 2006, pp. 61–70, Eurographics Association.

[31] C.C.L. Wang, K.-C. Hui, and K.-M. Tong, "Volume parameterization for design automation of customized free-form products," *IEEE Trans. on Auto. Sci. and Eng.*, vol. 4, no. 1, pp. 11–21, 2007.

[32] C.C.L. Wang, Y. Wang, and M.M.F. Yuen, "Design automation for customized apparel products," *Comput. Aided Des.*, vol. 37, no. 7, pp. 675–691, 2005.

[33] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.

Tsz-Ho Kwok received his B.Eng. degree in Automation and Computer-Aided Engineering from the Chinese University of Hong Kong (2009). He is currently a PhD student in Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong. His research interests include image processing, geometric and solid modeling, and visual Computing.

Yunbo Zhang is a Ph.D Student of Department of Mechanical and Automation Engineering at The Chinese University of Hong Kong. His research interests include CAD&CAM, physical based geometric modeling, and computer graphics. Zhang received his Mphil in digital design and manufacture from National Engineering Research Center for CAD, Huazhong University of Science and Technology.

Charlie C.L. Wang is currently an Associate Professor at the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, where he began his academic career in 2003. He gained his B.Eng. (1998) in Mechatronics Engineering from Huazhong University of Science and Technology, M.Phil. (2000) and Ph.D. (2002) in Mechanical Engineering from the Hong Kong University of Science and Technology. He is a member of IEEE and ASME, and an executive committee member of Technical Committee on Computer-Aided Product and Process Development (CAPPD) of ASME. Dr. Wang has received a few awards including the ASME CIE Young Engineer Award (2009), the CUHK Young Researcher Award (2009), the CUHK Vice-Chancellor's Exemplary Teaching Award (2008), and the Best Paper Awards of ASME CIE Conferences (in 2008 and 2001). His current research interests include geometric modeling in computer-aided design and manufacturing, biomedical engineering and computer graphics, as well as computational physics in virtual reality.