

國立交通大學

多媒體工程研究所

碩士論文



根據對應歧異點集合之全域性表面對應

Global Surface Mapping Based on Corresponding Cone
Singularity Set

研究生：李紹禛

指導教授：莊榮宏 教授

黃世強 教授

中華民國 101 年 12 月

根據對應歧異點集合之全域性表面對應
Global Surface Mapping Based on Corresponding Cone Singularity Set

研究生：李紹禎

Student : Shao-Ti Lee

指導教授：莊榮宏

Advisor : Jung-Hong Chuang

黃世強

Sai-Keung Wong

國立交通大學
多媒體工程研究所
碩士論文



A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

December 2012

Hsinchu, Taiwan, Republic of China

中華民國 101 年 12 月

根據對應歧異點集合之全域性表面對應

研究生：李紹禎

指導教授：莊榮宏 博士

黃世強 博士



摘要

在電腦圖學的領域中，許多應用都需要計算一個平滑且合理的表面對應。例如網格間的屬性轉移、形狀混合，或是物體形變。其中一種計算對應的方式是將給予的表面參數化到共同的基底模型，但若將表面切割成數塊區域並分別局部參數化，區域間的分界線處會不平滑，因此必須對於不平滑處套用某些平滑演算法做事後處理。本篇論文發表了一個透過基底模型做全域性的表面對應，而不需要使用任何平滑演算法。主要概念是利用「離散瑞奇流」以及「物體間一致的保留形體之錐形歧異點集合」來計算「平坦錐形度量」，並將此度量攤平在基底模型上來做到表面對應。平坦錐形度量能保留原本表面的角度關係，並且可被想成是一種攤平在基底模型的全域參數化結果。簡言之，我們提出一個新的概念透過基底模型做全域性的表面對應。

Global Surface Mapping Based on Corresponding Cone Singularity Set

Student: Shao-Ti Lee

Advisor: Dr. Jung-Hong Chuang

Dr. Sai-Keung Wong

Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University



ABSTRACT

In computer graphics, computing a smooth and reasonable surface correspondence or inter-surface mapping is needed for applications such as mesh attributes transferring, shape blending, and object morphing. One of the methods for computing the correspondence is to parameterize the given surfaces to a common base-complex. However, the patch boundaries would not be smooth if surfaces are segmented and locally parameterized and hence some smoothing operations are normally applied as a post-processing. This thesis proposes a global inter-surface mapping based on the base-complex approach without the need of smoothing operation. The main idea is to make good use of discrete Ricci flow and the shape-preserving corresponding cone singularity set between surfaces to compute flat cone metrics, which is regarded as the common domain of the inter-surface mapping. The flat cone metric is conformal to the original surface and could be thought as a global parameterization on the common domain. To sum up, our framework provides a novel idea to map between surfaces via a common base-complex and global parameterization.

Acknowledgments

I would like to thank my advisors, Professor Jung-Hong Chuang and Professor Sai-Keung Wong, and my senior colleague, Tan-Chi Ho. They always give guidances, inspirations, and technical supports in this work. I also appreciate all my colleagues of CGGM Lab. Especially I thank Yu-Jie Chen. He helped me with many math problems. Yu-Yu Lin, Hsiu-Fan Chao, and Tsung-Shian Huang patiently discussed with me and brainstormed together on the research. It is also appreciate for Fu-Si Huang and Tse-Ching Chang. They give supports on designing textures and editing diagrams of results. Finally, I am grateful to my parents for their love and supports on living very much. Without them, I would not overcome all the pain and frustrations in my life.



Contents

1	Introduction	1
1.1	Contributions	3
1.2	Organization of the Thesis	3
2	Related Work	5
2.1	Inter-surface mapping	5
2.2	Global Parameterization	7
2.2.1	Discrete Ricci flow	7
3	Surface Mapping Based on Corresponding Cone Singularities	12
3.1	Overview	12
3.2	Common base domain	15
3.3	Global conformal parameterization	16
3.3.1	Flat cone metric and cone singularity	19
3.3.2	Discrete Ricci flow	21
3.4	Surface mapping	24
4	Experimental Result	27
4.1	Result	27
4.2	Comparison of different target Gaussian curvatures.	37

5	Conclusions	41
5.1	Summary	41
5.2	Limitation	42
5.3	Future work	42
	Bibliography	43



List of Figures

1.1	The common domain based inter-surface mapping method [KS04].	2
2.1	Circle packing metric on a triangle. v_1 , v_2 , and v_3 denote the vertices for the face. θ_1 , θ_2 , and θ_3 are the angles of the vertices in this face. γ_1 , γ_2 , and γ_3 are the circle radii. ϕ_{12} , ϕ_{23} , and ϕ_{31} represents the angles included by the radii [GY08].	8
2.2	The models of the first row are respectively conformally map to the parameterizations of the second row [GY08].	9
2.3	The genus one kitten (left) could be embedded on the Euclidean plane with no cone singularities, which means a flat metric (center). It could also deform to a flat cone metric with 8 cone singularities of Gaussian curvature $\frac{\pi}{2}$ and 8 cone singularities of Gaussian curvature $-\frac{\pi}{2}$ (right). Both metrics are conformal to the genus one kitten and with the same total Gaussian curvature [LJX ⁺ 10]. . .	11
3.1	System overview.	14
3.2	A mapping of initial features via the common base domain.	15
3.3	A simple example for the idea of using the discrete Ricci flow to obtain the flat cone metric. (a) The unit sphere. (b) The tetrahedron that the unit sphere is to be deformed to. (c) Assign the target Gaussian curvature. Four red points are set with π and others with 0. (d) The flat cone metric (global parameterization).	17

3.4	An example of finding the flat cone metric of the horse. (a) The input mesh with colored feature points mapped to the vertices of the common base domain. (b) The common domain. Red and blue vertices represent the positive and negative Gaussian curvature at vertices, respectively. (c) Assign the target Gaussian curvature to the horse model.(d) The flat cone metric (global parameterization on the common base domain).	18
3.5	A cone singularity with positive Gaussian curvature (red) and with negative Gaussian curvature (blue).	20
3.6	The feature points of both meshes are assigned with target Gaussian curvatures. Red and blue points are the assigned positive and negative target Gaussian curvature. Other points are set with zero.	20
3.7	Inversive distance circle packing [YGL ⁺ 09].	23
3.8	The flat cone metric on the horse head. (a) The red face and the yellow region of the left image are the guessed starting face and the locally flattened area of that patch respectively. (b) The parameterized faces on the common domain. (c) The parameterized result of the yellow region on the uv coordinates.	25
3.9	The final inter-surface mapping. Note that the patch lines on S and T are the intersections between the parameterized edges and the edges of the common base domain D_S and D_T respectively.	26
4.1	The texture transferring via the inter-surface mapping. Texturing target T via the 2D domain. And S gets the color from T through $f_{S \rightarrow D_S}$, $f_{D_S \rightarrow D_T}$, and $f_{T \rightarrow D_T}^{-1}$	28
4.2	Using checker board to show the inter-surface mapping from Camel to Horse. The corresponding features colored with red and blue (positive Gaussian curvature and negative Gaussian curvature) are shown on both meshes and the common base domain. Red lines on Horse are texture boundaries on the uv coordinates.	29

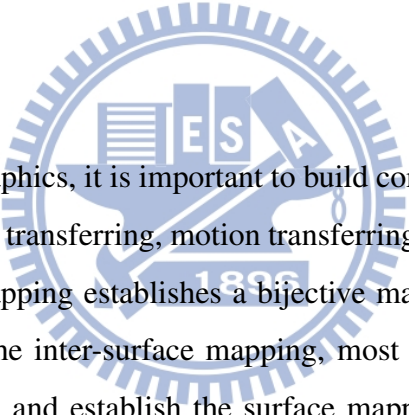
4.3	Using circle packing to show the inter-surface mapping from Camel to Horse. Red lines on Horse are texture boundaries on the uv coordinates.	30
4.4	Using colored ribbon to show the inter-surface mapping from Camel to Horse. Red lines on Horse are texture boundaries on the uv coordinates.	30
4.5	Red lines show the texture boundaries. (a) The textured cone. (b) The parameterized cone on the 2D domain.	31
4.6	Mapping between heads of Camel and Horse.	32
4.7	Mapping between bodies of Camel and Horse.	32
4.8	Mapping between legs of Camel and Horse.	33
4.9	Using checker board to show the inter-surface mapping from Wolf to Dog. . . .	34
4.10	Using circle packing to show the inter-surface mapping from Wolf to Dog. Red lines are patch lines.	34
4.11	Using colored ribbon to show the inter-surface mapping from Wolf to Dog. . .	35
4.12	Using checker board to show the inter-surface mapping between the horse models in different poses. Red and blue dots are cone singularities.	35
4.13	Using circle packing to show the inter-surface mapping between the horse models in different poses. Red lines are patch lines.	36
4.14	Using colored ribbon to show the inter-surface mapping between the horse models in different poses.	36
4.15	A comparison for different set of target Gaussian curvatures. The horse on the left column shows a conformal metric with high area distortion if the target Gaussian curvature is uniformly distributed. The mesh on the right column shows better result since the target Gaussian curvature is assigned with the Gaussian curvatures of vertices of the common base domain. L2 stretches are also shown to depict that our method lead to a quasi-isometric result (An L2 stretch will be 1 if the metric is completely isometric to the original metric). . .	38

4.16	The case with uniformly distributed target Gaussian curvature will cost more time than the one that the target Gaussian curvatures are assigned with the Gaussian curvatures of vertices on the common base domain.	39
4.17	The comparison of the conformal metric of the wolf with different set of target Gaussian curvatures. Red lines are the texture boundaries.	40
4.18	The converge speed of different set of target Gaussian curvatures.	40



CHAPTER 1

Introduction



In the field of the computer graphics, it is important to build correspondences between surfaces for applications such as texture transferring, motion transferring, object deformation, and shape blending. An inter-surface mapping establishes a bijective mapping between surfaces. Since it is difficult to directly find the inter-surface mapping, most previous methods parameterize surfaces to a common domain and establish the surface mapping by mapping points on one surface to the common domain and then inverse mapping the points on the common domain to the second surface. Figure 1.1 shows the concept of the inter-surface mapping. Given a source mesh S and a target mesh T , the inter-surface function $f : S \rightarrow T$ can be decomposed as

$$f := f_{T \rightarrow D_T}^{-1} \circ f_{D_S \rightarrow D_T} \circ f_{S \rightarrow D_S}, \quad (1.1)$$

where D_S and D_T are the common domain with the same shape and $f_{S \rightarrow D_S}$ and $f_{T \rightarrow D_T}$ are parameterizations from S to D_S and from T to D_T , respectively, and $f_{D_S \rightarrow D_T}$ is a mapping from D_S to D_T . Our algorithm is also based on this idea.

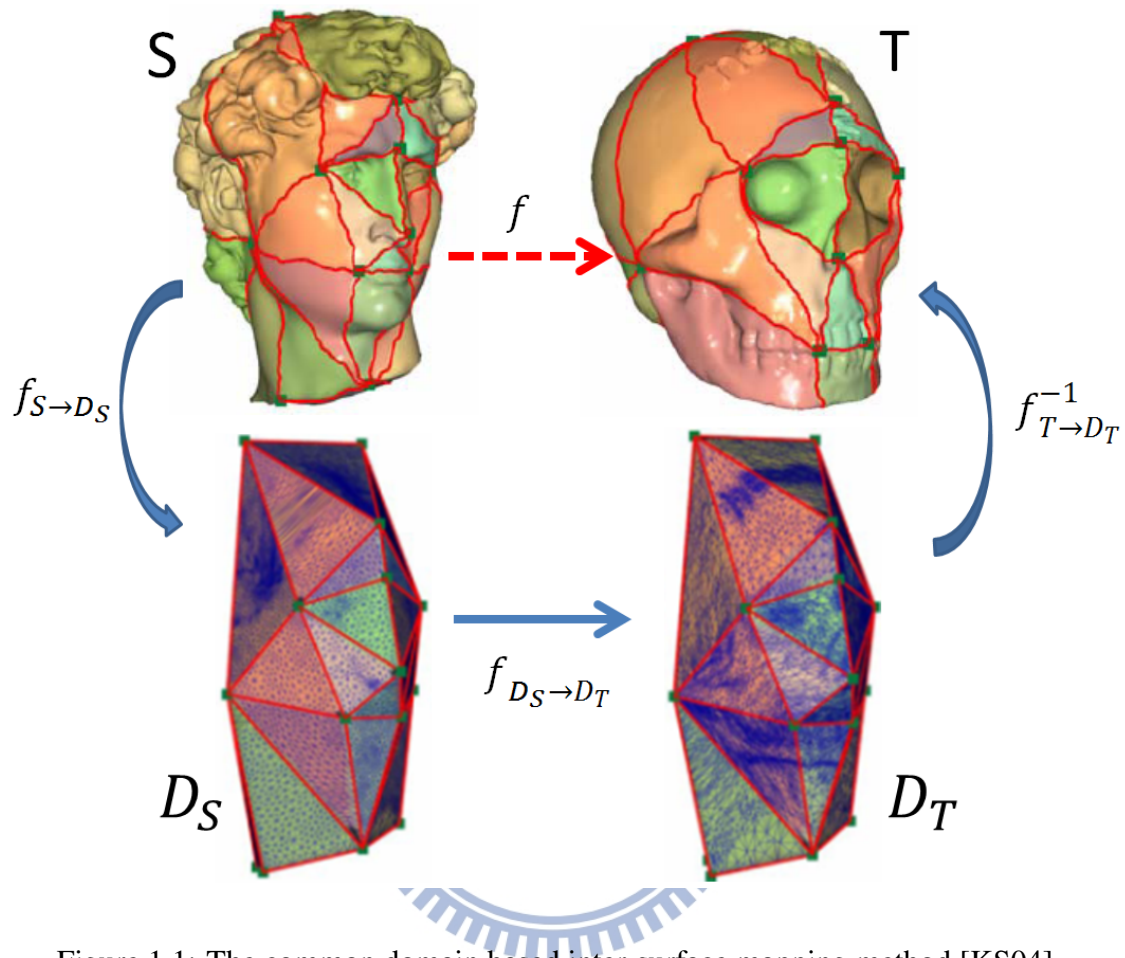


Figure 1.1: The common domain based inter-surface mapping method [KS04].

The common domain used in previous methods could be classified into two categories: planar domain and non-planar domain. The former is simple and segment the source surface to a set of topological disks and parameterize those disks onto some planar domains such as a unit circle or a rectangle. The latter is various, for example, a unit sphere, a base-complex of the input surface, or a convex-hull that encloses the input surface. The purpose of all methods is to simplify the domain of inputs, yet still build a reasonable correspondence since most of the features would disappear on the common domain. For a base-complex method, it still retains the shape and the part information, which is suitable for mapping surfaces with a requirement of part-awareness. However, the problem of using the base-complex method is that the param-

eterization over the boundary would not be smooth when the surface is segmented into patches and locally parameterized onto the faces of the base-complex. In this case, it is needed to apply some smoothing operations along patch boundaries.

Our work is motivated by the above observations. We seek an inter-surface mapping algorithm that is able to build a feature-based correspondence that is entirely smooth over the boundaries; in other words, the given input surfaces will not be segmented into parts and then locally parameterized respectively. Although a unit sphere is a good common domain for global parameterization, it could not preserve any feature. Discrete Ricci flow [JKLG08], a powerful global parameterization tool, could be applied to achieve our goal. By taking the advantage of discrete Ricci flow, we could not only map corresponding features assigned by users but also build a common domain that allows the input surfaces to be globally parameterized.

1.1 Contributions

The followings are our main contributions:

- An approach of computing a quasi-isometric cone singularity set is proposed. Since our method is based on the discrete Ricci flow that computes a conformal metric, if the cone singularities are referred from the base-complex of the input, the conformal metric would be more area-preserving. And thus the result would be quasi-isometric.
- After the shape of the common domain is determined, the method computes a feature-based global surface mapping function via the same shape of flat cone metrics computed by the discrete Ricci flow.

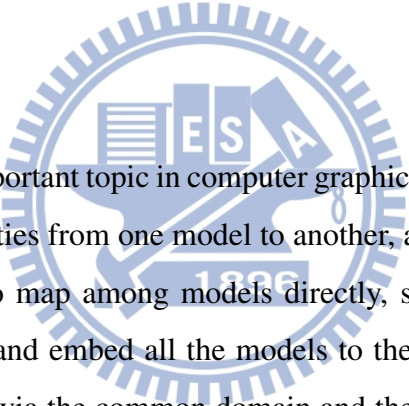
1.2 Organization of the Thesis

The following chapters are organized as follows. Chapter 2 gives a general review and previous works on the base-complex method for the inter-surface mapping and some global parameter-

ization method using the discrete Ricci flow. Chapter 3 describes how our approach works by taking the advantage of the discrete Ricci flow, including an overview, the cone singularity settings, the computation of discrete Ricci flow, and the surface mapping part. Chapter 4 demonstrates our experiment result for some pairs of data sets. And Chapter 5 summarizes our approach, describes the limitation and some future works.



Related Work



Inter-surface mapping is an important topic in computer graphics. If one wants to match, morph, or transport some mesh properties from one model to another, an inter-surface mapping should be needed. Since it is hard to map among models directly, some previous methods tried to first build a common domain and embed all the models to the common domain and then obtain the inter-surface mapping via the common domain and the parameterization. We describe existing works on inter-surface mapping in Section 2.1 and global parameterization in Section 2.2. More context of surface mapping and mesh parameterization can be found in good surveys [vKZHCO10] and [HLS07], respectively.

2.1 Inter-surface mapping

Since our algorithm is a base-complex method, we will focus on this type of surface mapping. A base-complex method for two models could be decomposed into four steps: First, find a common base domain from corresponding features of two models. Second, trace patch curves on inputs according to the connectivity of the base-complex. Third, map two models onto

the common domain respectively. In the end, the points of two models are mapped if they are mapped to the same point on the common domain. The resulting surface mapping can be further improved by some kind of relaxation driven by some energy functions until the best mapping is achieved.

Lee et al. [LDSS99] built a common domain by using MAPS [LSS⁺98] after first assigning corresponding feature points. This method does not suit for mapping shapes in greatly difference, so the later methods tried to find coarse meshes using consistent segmentation to ensure coarse meshes of the same topology such as [PSS01]. That way, the alignment and projection in [LDSS99] would no longer be used, and patch segments of the source and the target meshes could be directly matched. Praun et al. [PSS01] enable users to specify corresponding points and connectivities and based on that input meshes could be partitioned to triangular patches that are in correspondence. The methods of Schreiner et al. [SAPH04] and Kraevoy et al. [KS04] extended the method of [PSS01], which could allow users to specify a few constraint points and automatically construct the simplicial complex and patches. Kraevoy et al. [KS04] used a shape preserving method [Flo03] to map the surfaces onto patches and build the mapping from surface to the common domain. However, Schreiner et al. [SAPH04] used an alternative method that they created a continuous map between meshes directly and minimize the mapping distortion during a coarse-to-fine refinement to obtain the final mapping result. Pan et al. [PWPY07] improved the method of [KS04] by iteratively refine the common domain. For other surface mapping methods, we refer readers a good survey on shape correspondence [vKZHCO10]

Although these base-complex approaches are intuitive for surface mapping, they could not generate mapping results with globally smooth since all the mapping result of corresponding patches are computed locally. The methods usually need a post-processing step for smoothing the boundaries of patches; otherwise, the patch boundaries would have only C_0 continuity, resulting in visual artifacts. Our method is inspired by global parameterization approaches and try to obtain a globally optimized result of inter-surface mapping without the need of boundary smoothing.

2.2 Global Parameterization

Global parameterization is the result of a global minimization of energy functions. Khodakovsky et al. [KLS03] defined a system which introduced transition functions over patch boundaries such that vertices are allowed to cross patches during the relaxation step. This approach results in a global smooth parameterization with low distortion. Miao et al. [JKLG08] used the discrete Ricci flow aiming at finding a Riemannian metric [Ham82] that is of zero Gaussian curvature everywhere and conformal to the original metric. To reach this goal, they generalized the continuous Ricci flow [Ham88] to the discrete setting for the use in engineering applications. Since the algorithm could not deal well with the triangulation of low quality, Yang et al. [YGL⁺09] re-defined the circle packing metric for a better result. More details of the discrete Ricci flow will be described in Section 2.2.1. Li et al. [LBG⁺08] took the advantage of a uniformization metric, which is also a kind of Riemannian metric, to build a surface mapping scheme. We focus on the papers that are related to our method; see [HLS07] for other global parameterization papers.

2.2.1 Discrete Ricci flow

Discrete Ricci flow is proposed in [CL03]. It computes a circle packing metric (Figure 2.1) which is conformal to the original circle packing metric of an input mesh by following a set of target Gaussian curvatures. The total Gaussian curvature of the original metric and that of the computed conformal metric should be the same. Total Gaussian curvature κ could be easily calculated by the following equation:

$$\kappa = 2\pi(2 - 2g), \quad (2.1)$$

where g is the genus of the input mesh. The genus of a surface means the number of handle loops of that surface. According to Riemann Uniformization Theorem, all surfaces could be mapped into the unit sphere, the Euclidean plane, and the hyperbolic space and these three canonical shapes respectively induce constant Gaussian curvature +1, 0, and -1, as shown in Figure 2.2.

For a simpler case, we just focus on the Euclidean domain. In general, only models with genus one could be flattened on the Euclidean plane with zero Gaussian curvatures everywhere since the total Gaussian of genus one model is zero. The flattened metric is called a flat metric. For models with other genus, the flat metric does not exist since the total Gaussian curvature is not zero, and so a flat cone metric is induced. A flat cone metric is a Riemannian metric in the Euclidean space that contains Gaussian curvatures with zero everywhere except at cone singularities. It could be generated by the discrete Ricci flow.

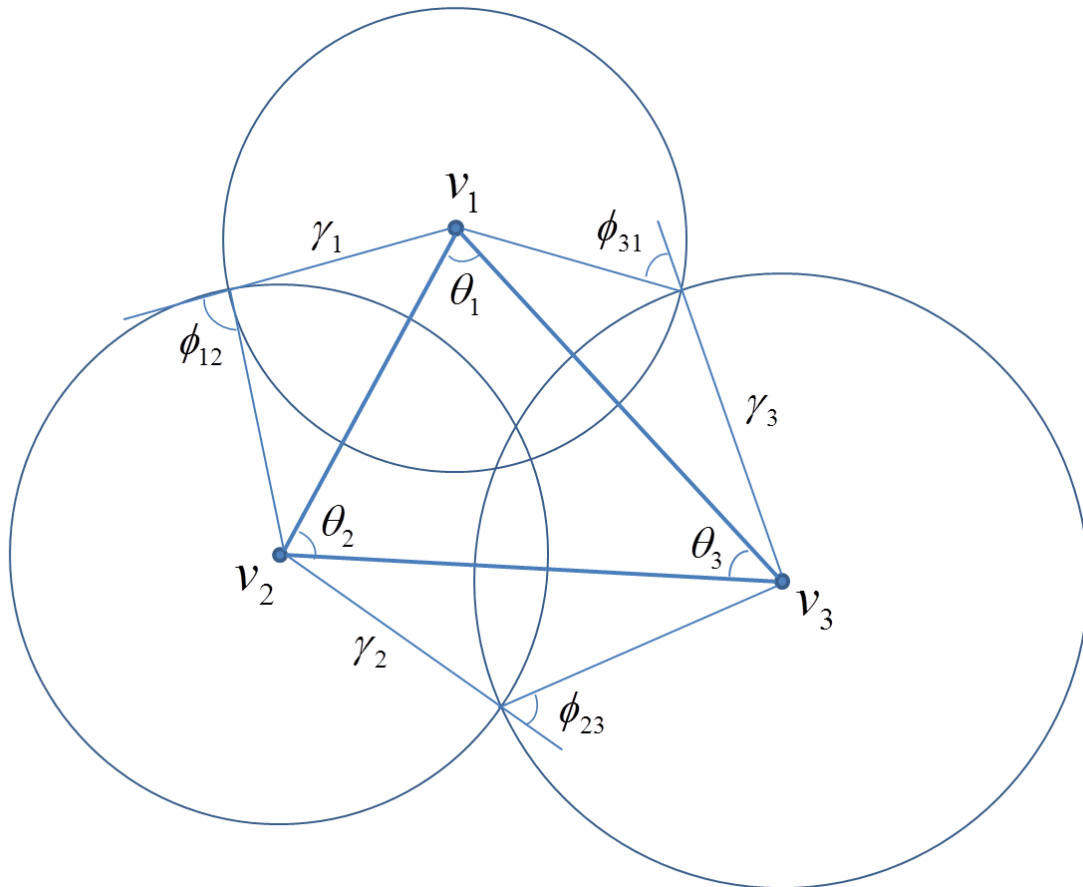


Figure 2.1: Circle packing metric on a triangle. v_1 , v_2 , and v_3 denote the vertices for the face. θ_1 , θ_2 , and θ_3 are the angles of the vertices in this face. γ_1 , γ_2 , and γ_3 are the circle radii. ϕ_{12} , ϕ_{23} , and ϕ_{31} represents the angles included by the radii [GY08].

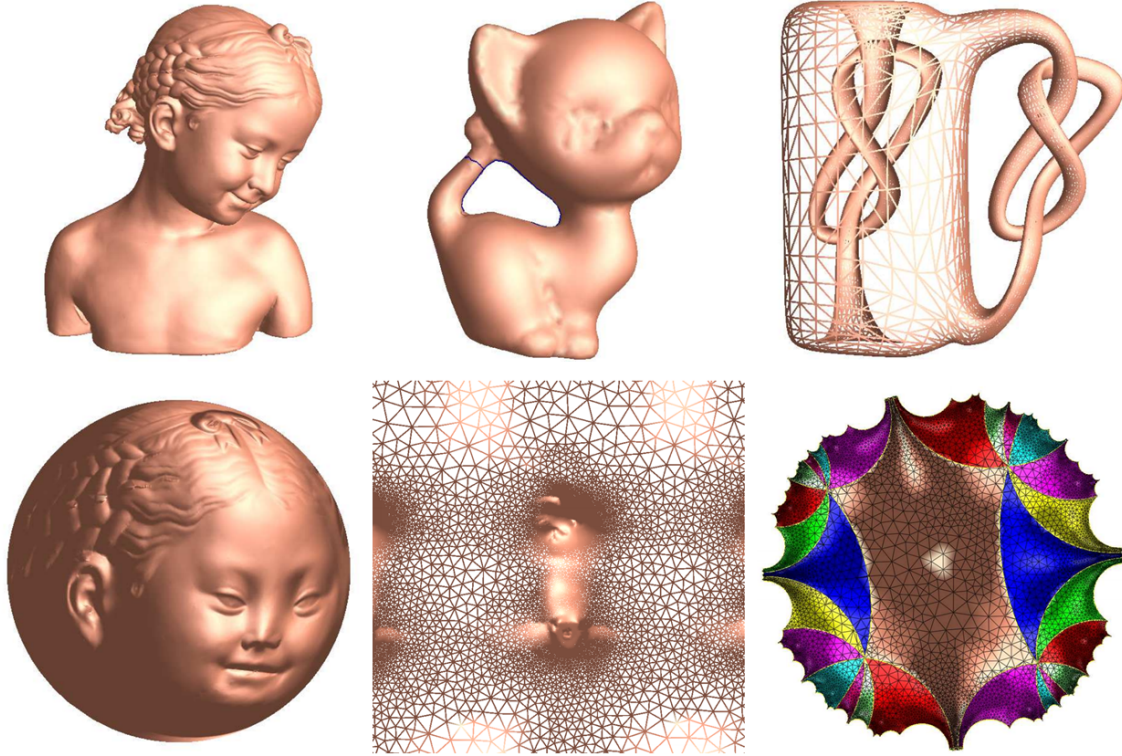


Figure 2.2: The models of the first row are respectively conformally map to the parameterizations of the second row [GY08].

From a viewpoint of computer graphics, the discrete Ricci flow iteratively "deforms" an input mesh to a new shape with a given set of target Gaussian curvatures. The verb "deform" represents that the circle radii of the initial circle packing metric, which is computed from the input mesh, change until the target circle packing metric is obtained. The tangency of the circles would be preserved during changing, i.e., the metrics will remain conformality. Let the circle packing metric vector be defined as $\mathbf{u} = \{u_1, u_2, \dots, u_n\}$ for the input mesh with n vertices which induces Gaussian curvature vector $\mathbf{K} = \{K_1, K_2, \dots, K_n\}$ and the target circle packing metric vector be defined as $\bar{\mathbf{u}}$ which induces Gaussian curvature vector $\bar{\mathbf{K}}$. Chow and Luo [CL03] proved that the discrete Ricci flow could be formulated as the following equation:

$$\frac{du_i}{dt} = \bar{K}_i - K_i \quad (2.2)$$

where

$$u_i = \ln \gamma_i, \quad (2.3)$$

in the Euclidean space. γ_i is the circle radius of a vertex v_i . \bar{K}_i and K_i represents the target Gaussian curvature and the Gaussian curvature of a vertex v_i , respectively. They also showed that given an initial circle packing metric \mathbf{u} and a set of target Gaussian curvatures $\bar{\mathbf{K}}$ for Equation 2.2, the target circle packing metric $\bar{\mathbf{u}}$ will be obtained while the solution of the equation converges. In other words, the discrete Ricci flow is feasible for computing the target circle packing metric. Miao et al. [JKLG08] induced an alternative algorithm for computing the discrete Ricci flow by optimizing the discrete Ricci energy:

$$E(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} \sum_i (\bar{K}_i - K_i) du_i, \quad (2.4)$$

where $\mathbf{u}_0 = (0, 0, \dots, 0)$. Equation 2.2 is the negative gradient flow of the discrete Ricci energy. They used Newton's method for a fast convergence while optimizing the energy. In order to use Newton's method, the Hessian matrix of E with respect to \mathbf{u} is computed. See how the Hessian matrix is used to get the target circle packing metric in Section 3.3.2. Yang et al. [YGL⁺09] also solved the discrete Ricci energy by using Newton's method but induced inversive distance for the circle packing metric to handle the meshes with low-quality which could not be dealt with in [JKLG08].

After the target Gaussian curvature set is reached, the target metric is obtained; in other words, a conformal parameterization of the input mesh is generated. If the target Gaussian curvature set are zero for almost all the vertices, the obtained metric will be a flat cone metric. Note that the total Gaussian curvature of the target set should be the same with that of the input mesh; otherwise, the solution would not exist. Figure 2.3 shows an example that a kitten of genus one could be flattened to a flat metric or a flat cone metric. Total Gaussian curvatures are zero for both metrics.

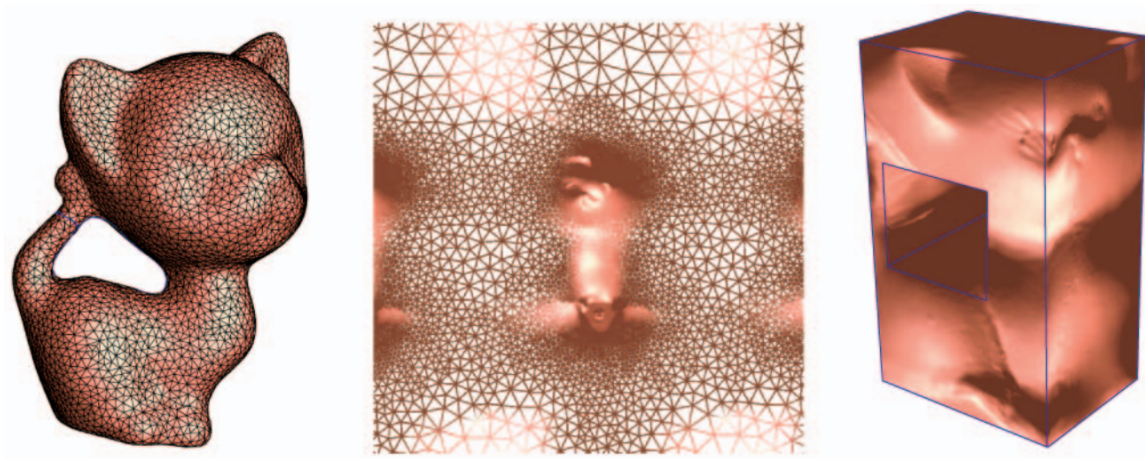
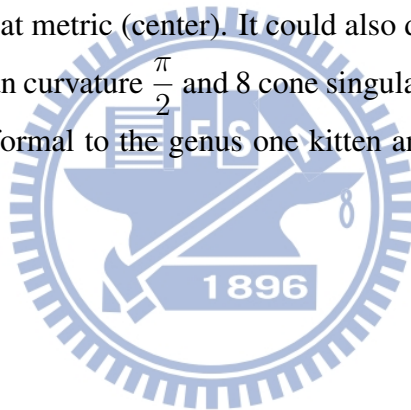
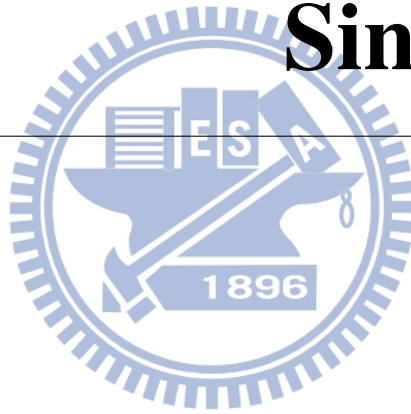


Figure 2.3: The genus one kitten (left) could be embedded on the Euclidean plane with no cone singularities, which means a flat metric (center). It could also deform to a flat cone metric with 8 cone singularities of Gaussian curvature $\frac{\pi}{2}$ and 8 cone singularities of Gaussian curvature $-\frac{\pi}{2}$ (right). Both metrics are conformal to the genus one kitten and with the same total Gaussian curvature [LJX⁺10].



Surface Mapping Based on Corresponding Cone Singularities



3.1 Overview

As we mentioned in Section 2.1, many traditional inter-surface mapping methods map a source mesh to a target mesh through a common base domain. According to the connectivity of the base domain, the method traces patch lines on both meshes to consistently segment the meshes into n corresponding patches, where n is the number of faces on the common domain. After that, some parameterization methods would be applied for each patch individually. It is necessarily that the surface mapping function f in Equation 1.1 would be represented as $f := (f_1, f_2, \dots, f_n)$, where

$$f_k := f_{T_k \rightarrow D_{T_k}}^{-1} \circ f_{D_{S_k} \rightarrow D_{T_k}} \circ f_{S_k \rightarrow D_{S_k}}, k = 1, 2, \dots, n, \quad (3.1)$$

where D_{S_K} , D_{T_K} represent the K-th face of the base domains and S_K , T_K represent the K-th patches on the source and the target mesh respectively. The kind of method needs to further improve the smoothness of the mapping result among patch boundaries since the mapping is derived individually for each corresponding patches, and the patch boundaries would have only C_0 continuity, resulting in visual artifacts. In this thesis, we aim at finding a global inter-surface mapping between two manifold meshes with the same topology, which means that the surface mapping function f would not be segmented and directly finds $f_{S \rightarrow D_S}$, $f_{T \rightarrow D_T}$, and $f_{D_S \rightarrow D_T}$.

Figure 3.1 shows the flowchart of our approach. Given two meshes S and T , users are allowed to assign a set of corresponding feature points on both meshes. The features are to be used for the mesh simplification process. Next, a single mesh S or T would be chosen to be simplified until only the assigned feature points are preserved, and that is the common base domain D . After the common base domain is constructed, our method does not trace patch lines on both input meshes as previous works did. Instead, the input meshes are globally parameterized onto the domain by using discrete Ricci flow [JKLG08] and thus $f_{S \rightarrow D_S}$ and $f_{T \rightarrow D_T}$ are obtained. In the end, since D_S and D_T are the result of the global parameterizations on D respectively, the function $f_{D_S \rightarrow D_T}$ is intuitively generated. The result of the mapping f is globally smooth everywhere except at the chosen feature points.

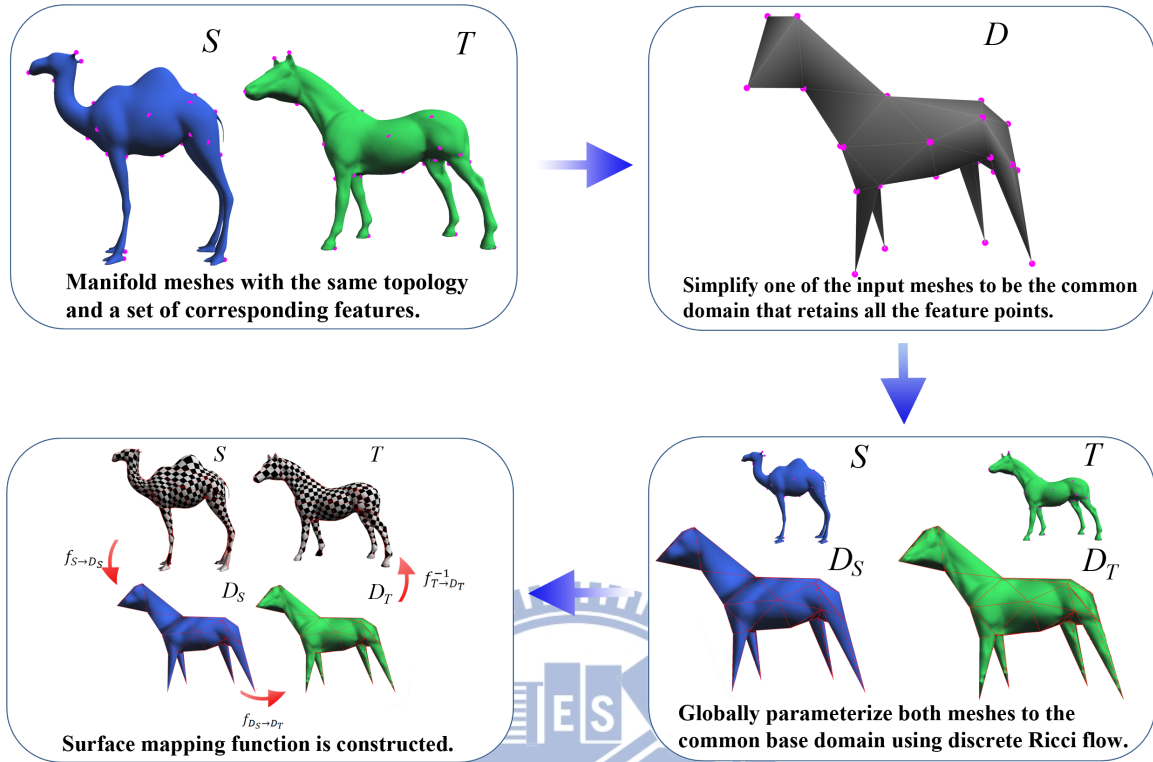


Figure 3.1: System overview.

As shown in the flowchart, before computing $f_{S \rightarrow D_S}$ and $f_{T \rightarrow D_T}$, a common base domain D needs to be built first. And the initial mapping between the given features on both input meshes and the vertices of the common base domain should be mapped respectively, as shown in Figure 3.2. The generation of the base-complex would be introduced in Section 3.2. The information of the common base domain and the mapping of initial features would be used in the process of computing flat cone metrics, i.e., global conformal parameterizations, to obtain $f_{S \rightarrow D_S}$, $f_{T \rightarrow D_T}$, and $f_{D_S \rightarrow D_T}$. The details will be described in Section 3.3.

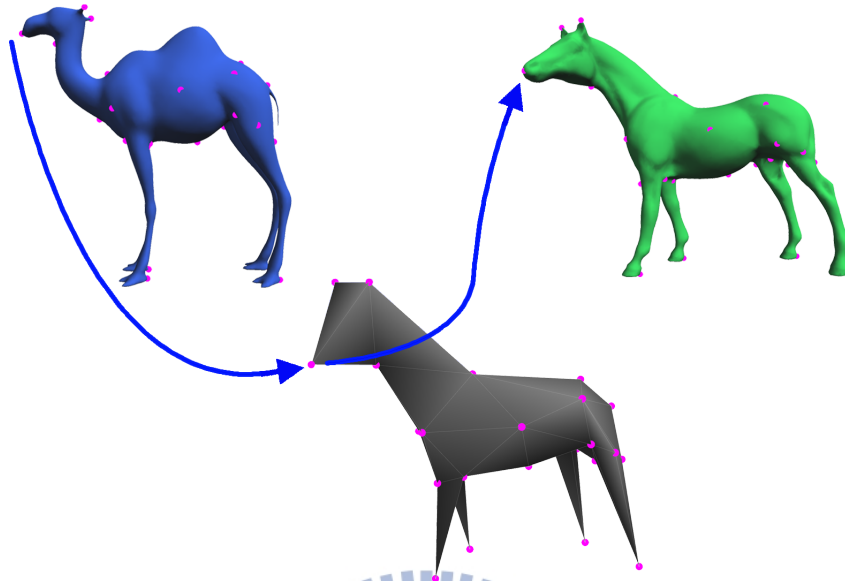


Figure 3.2: A mapping of initial features via the common base domain.

3.2 Common base domain

Given two input meshes S and T and a set of corresponding feature points assigned by users, the first step of the inter-surface mapping is to derive the common base domain D . We already have the correspondence mapping between features on S and T :

$$f(S_{s_k}) \rightarrow T_{s_k}, k = 1, 2, \dots, a, \quad (3.2)$$

where $S_s = \{S_{s_1}, \dots, S_{s_a}\}$ and $T_s = \{T_{s_1}, \dots, T_{s_a}\}$ are features on S and T , and a is the number of features on both meshes. If one of the meshes, for example, T is chosen to be simplified until only the assigned features are remained. The result is the common base domain D , and the mapping from features on T to vertices on D is also built. Since the shape-preserving common base domain is desired, the quadric error metric (QEM) [GH97] is chosen as the error metric for the mesh simplification. The process sets the assigned feature points as constraints for the edge collapsing and computes a topologically identical triangular layout, which defines the connectivity of the common base domain, for S and T .

The computation is built around the error metrics and the edges collapsing. At the beginning of the algorithm, the error metrics of vertices and edges are defined (see Section 5 in [GH97]). For each iteration, an edge with the smallest error, which means the edge being most unconcerned with the shape, would be collapsed and the vertices of that edge would be merged together, and the error of the neighbor edges of that vertex would be updated. The terminal condition used in our method is that the remaining vertices are just only the assigned vertices. If there is any other remaining vertices, the correspondence of those vertices with S and T would be undefined. As a result, we obtain a common base domain D with a vertices with the connectivity determined by the mesh simplification.

3.3 Global conformal parameterization

The main purpose of this step is to use the information of the corresponding features between S and D and between T and D , and a vertices of D to globally parameterize the input meshes onto the common base domain and thus obtain the global surface mapping. A global conformal parameterization problem could be formalized as *finding a flat cone metric that is conformal to the original metric by using the discrete Ricci flow*. The set of edge lengths of an input mesh could be seen as an original metric and the global conformal parameterization on the base domain could be seen as a flat cone metric. In other words, we aim at finding a process for the input meshes to conformally deform to the shape of the common base domain. The discrete Ricci flow could be used to achieve this goal.

As mentioned in Section 2.2.1, given a set of target Gaussian curvature on an original metric, the discrete Ricci flow could conformally deform the metric to a new metric that is with the assigned Gaussian curvature. For example, a mesh of the unit sphere shown in Figure 3.3(a) is to be deformed to a regular tetrahedron (Figure 3.3(b)). It is known that the regular tetrahedron has four vertices and each has Gaussian curvature π . If one assigns Gaussian curvature π to the four points that correspond to the four vertices on the tetrahedron and 0 to all other points on the unit sphere (Figure 3.3(c)), the unit sphere would conformally deform to the regular tetrahedron

(Figure 3.3(d)) by using discrete Ricci flow. The deformation process can be considered as a global parameterization.

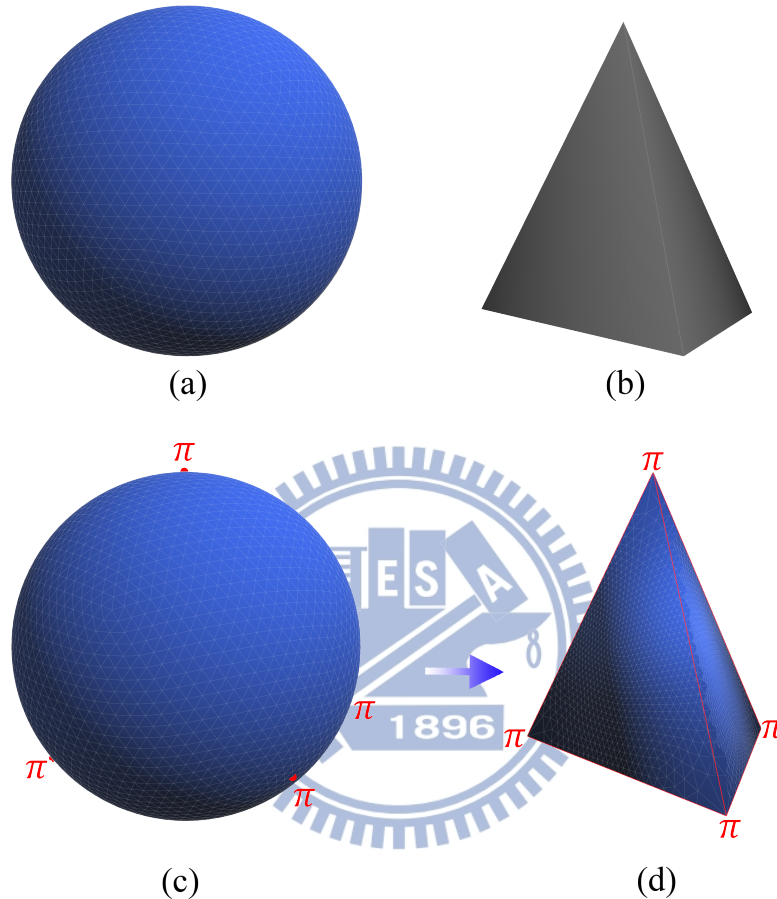


Figure 3.3: A simple example for the idea of using the discrete Ricci flow to obtain the flat cone metric. (a) The unit sphere. (b) The tetrahedron that the unit sphere is to be deformed to. (c) Assign the target Gaussian curvature. Four red points are set with π and others with 0. (d) The flat cone metric (global parameterization).

Our purpose is to globally and conformally deform S and T to the shape of D . If we assign the Gaussian curvature of the vertices in D to the corresponding features on S and T , S and T could deform to the shape of D by using the discrete Ricci flow. Figure 3.4 shows the idea applied to the horse.

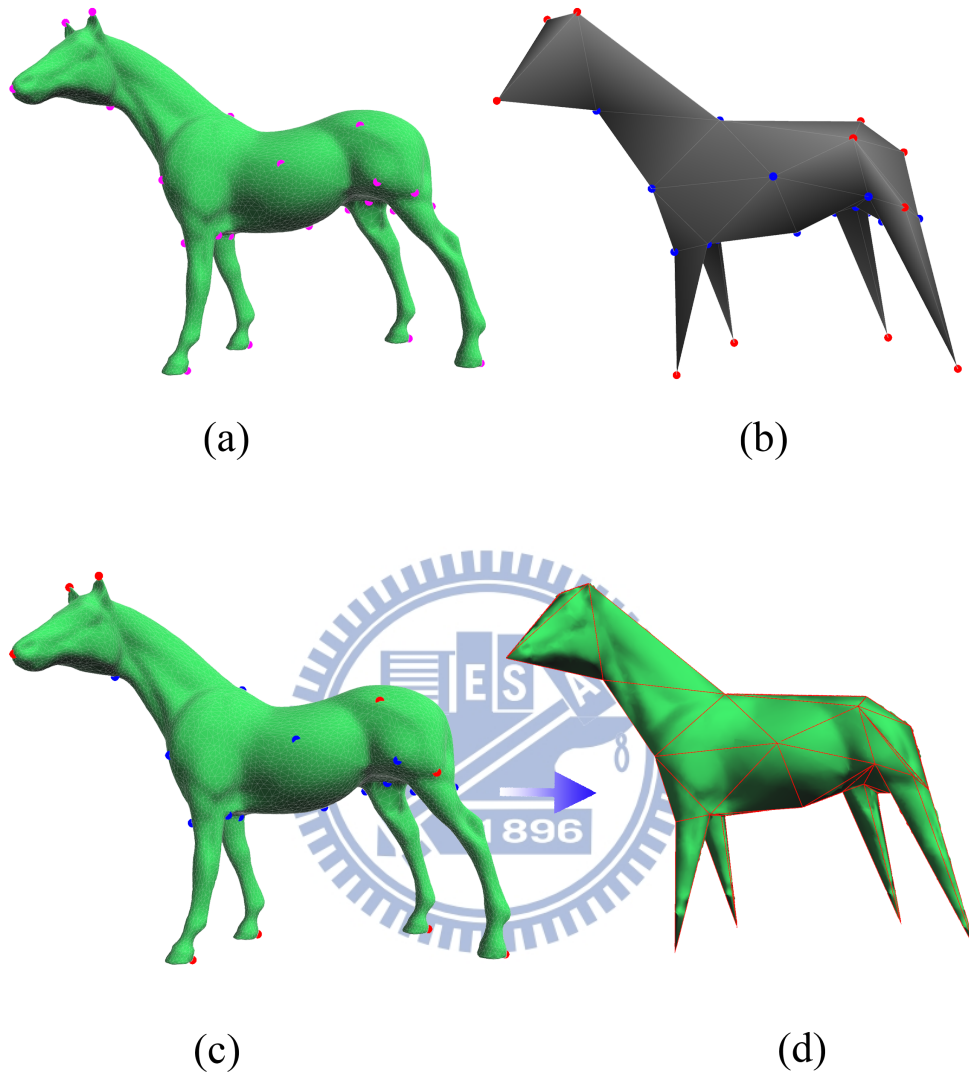


Figure 3.4: An example of finding the flat cone metric of the horse. (a) The input mesh with colored feature points mapped to the vertices of the common base domain. (b) The common domain. Red and blue vertices represent the positive and negative Gaussian curvature at vertices, respectively. (c) Assign the target Gaussian curvature to the horse model. (d) The flat cone metric (global parameterization on the common base domain).

In order to deform meshes S and T to the shape of the common base domain D using the discrete Ricci flow, the information of the vertices of D should be used. In fact, if D is

considered to be the shape of the flat cone metric, then the vertices of D could be considered as the cone singularities of the flat cone metric. In the next section, we would show the relationship between the cone singularities of the flat cone metric and the vertices of common base domain D . After introducing the terms, the computation of the discrete Ricci flow and the algorithm of global parameterization will be described.

3.3.1 Flat cone metric and cone singularity

As described in Section 2.2.1, a flat cone metric is a metric such that the Gaussian curvatures are zero everywhere except at some points, which are called cone singularities. It is definitely that a cone singularity should be with either positive Gaussian curvature or negative Gaussian curvature. For a triangular mesh M , the Gaussian curvature K_{s_i} of a cone singularity s_i could be estimated by the equation:

$$K_{s_i} = 2\pi - \sum_{j,k} \theta_{s_i}^{jk}, s_i \notin \partial M, \quad (3.3)$$

where j, k denotes the vertices of neighbor faces, $\theta_{s_i}^{jk}$ denotes the angle of $\angle j s_i k$, and ∂M denotes the boundary of M . We only focus on the mesh with no boundaries since the models we deal with are of zero genus. Figure 3.5 shows the idea of the equation. If the summation of neighbor angles is close to 2π , the Gaussian curvature would be close to zero, which means the point is nearly flat. It is also easy to find that a cone singularity with positive Gaussian curvature and negative Gaussian curvature represent a local convex point and a local concave point, respectively.

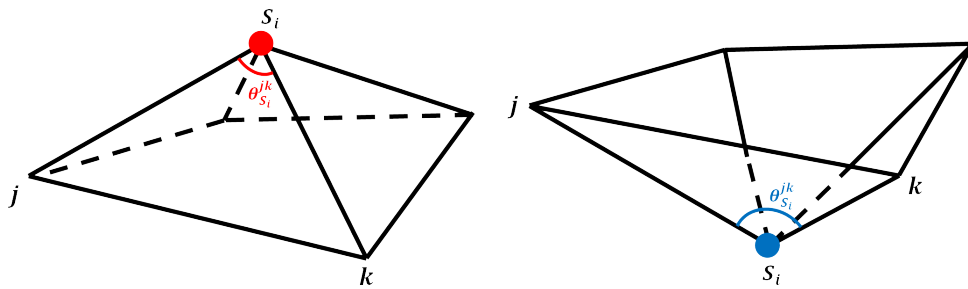


Figure 3.5: A cone singularity with positive Gaussian curvature (red) and with negative Gaussian curvature (blue).

Since D could be considered as the shape of a flat cone metric and the vertices of D could be thus considered as the cone singularities of the flat cone metric, we can compute the Gaussian curvature of the vertices and then propagate them to the corresponding points on the source S and the target T . Figure 3.6 shows the propagated target Gaussian curvature for the input meshes S and T . The discrete Ricci flow makes use of these target Gaussian curvatures to flow the original metric until the Gaussian curvatures on the surfaces are spread to the assigned value and then the flat cone metrics would be obtained.

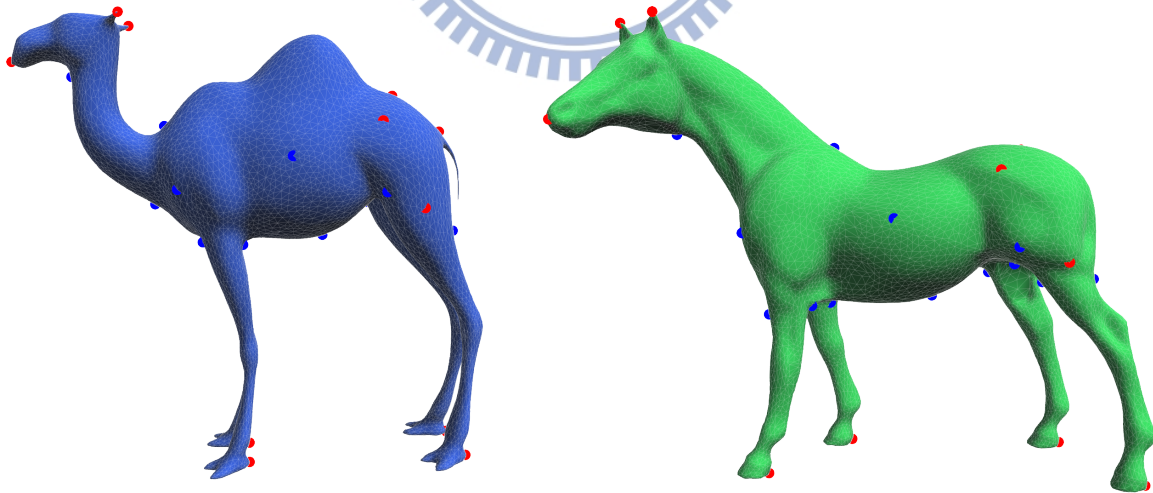


Figure 3.6: The feature points of both meshes are assigned with target Gaussian curvatures. Red and blue points are the assigned positive and negative target Gaussian curvature. Other points are set with zero.

In the next section, we could use the information of the target Gaussian curvature on the input meshes to apply the discrete Ricci flow, and the flat cone metrics could be obtained.

3.3.2 Discrete Ricci flow

In this section, the input meshes would be deformed to the shape of the common base domain since the target Gaussian curvatures assigned to the point of S and T are the Gaussian curvature of the base domain. As mentioned in Section 2.2.1, the algorithm in [YGL⁺09] finds the target metric by optimizing the discrete Ricci energy using Newton's method. According to Section 12.4 in [GY08], the discrete Ricci energy is independent of the choice of the integration path, so the gradient of the Equation 2.4 is

$$\nabla E = \left(\frac{\partial E}{\partial u_1}, \frac{\partial E}{\partial u_2}, \dots, \frac{\partial E}{\partial u_n} \right) = (K_1, K_2, \dots, K_n). \quad (3.4)$$

If $[v_i, v_j]$ is an edge of a mesh M , the element of Hessian matrix will be

$$\frac{\partial^2 E(\mathbf{u})}{\partial u_i \partial u_j} = \frac{\partial K}{\partial u_j} = -w_{ij}. \quad (3.5)$$

Otherwise, the element will be

$$\frac{\partial^2 E(\mathbf{u})}{\partial u_i \partial u_j} = 0. \quad (3.6)$$

And the element of vertex v_i is

$$\frac{\partial^2 E(\mathbf{u})}{\partial u_i^2} = \frac{\partial K}{\partial u_i} = \sum_{[v_i, v_j] \in M} w_{ij}. \quad (3.7)$$

The w_{ij} are defined in Equation 3.17. The Hessian matrix will be computed in each iteration of the optimization of the discrete Ricci energy to achieve the global optima.

At the beginning of the algorithm, an initial circle packing metric is needed as a starting metric of the discrete Ricci flow. Given a mesh, the radius γ_i of the circle c_i at vertex v_i is computed as follows.

$$\gamma_i = \min_{jk} \gamma_i^{jk}. \quad (3.8)$$

where

$$\gamma_i^{jk} = \frac{l_{ij} + l_{ki} - l_{jk}}{2} \quad (3.9)$$

and l_{ij} is the edge length of vertex v_i to its adjacent vertex v_j . The initial inversive circle packing metric I_{ij} of edge $[v_i, v_j]$ is computed by

$$I_{ij} = \frac{l_{ij}^2 - \gamma_i^2 - \gamma_j^2}{2\gamma_i\gamma_j}. \quad (3.10)$$

After the initial inversive circle packing metric is obtained, the optimization of the discrete Ricci energy starts. \mathbf{u} is initialized to be $(0, 0, \dots, 0)$, as described in Section 2.2.1. The conformal metric which induces the target curvature vector $\bar{\mathbf{K}}$ is computed by the following steps.

1. For each edge $[v_i, v_j]$, which is using current radii γ_i, γ_j and inversive distance I_{ij} , the edge length l_{ij} of the current metric is computed by

$$l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2I_{ij}\gamma_i\gamma_j}. \quad (3.11)$$

2. For each face $[v_i, v_j, v_k]$, the current corner angles θ_i, θ_j , and θ_k is computed by

$$\theta_i = \arccos \frac{l_{ij}^2 + l_{ki}^2 - l_{jk}^2}{2l_{ij}l_{ki}} \quad (3.12)$$

using the edge length from the previous step.

3. Since the current angles change, the current Gaussian curvature K_i for each v_i is updated by

$$K_i = 2\pi - \sum_{j,k} \theta_i^{jk}. \quad (3.13)$$

We only consider the mesh of genus zero, so the equation for boundaries is unnecessary.

4. For each face $[v_i, v_j, v_k]$, compute $\frac{\partial \theta_i}{\partial u_j}$ and $\frac{\partial \theta_i}{\partial u_i}$ by

$$\frac{\partial \theta_i}{\partial u_j} = \frac{\partial \theta_j}{\partial u_i} = \frac{h_k}{l_k} \quad (3.14)$$

and

$$\frac{\partial \theta_i}{\partial u_i} = -\frac{\partial \theta_i}{\partial u_j} - \frac{\partial \theta_i}{\partial u_k}, \quad (3.15)$$

where l_k denotes the length of edge which is opposite the vertex k and h_k represents the distance from the center of the red circle to l_k . The red circle in Figure 3.7 is orthogonal to the three circles of v_i, v_j , and v_k . The two equations are for the Hessian matrix in the next step. See the Appendix in [YGL⁺09] for proofs.

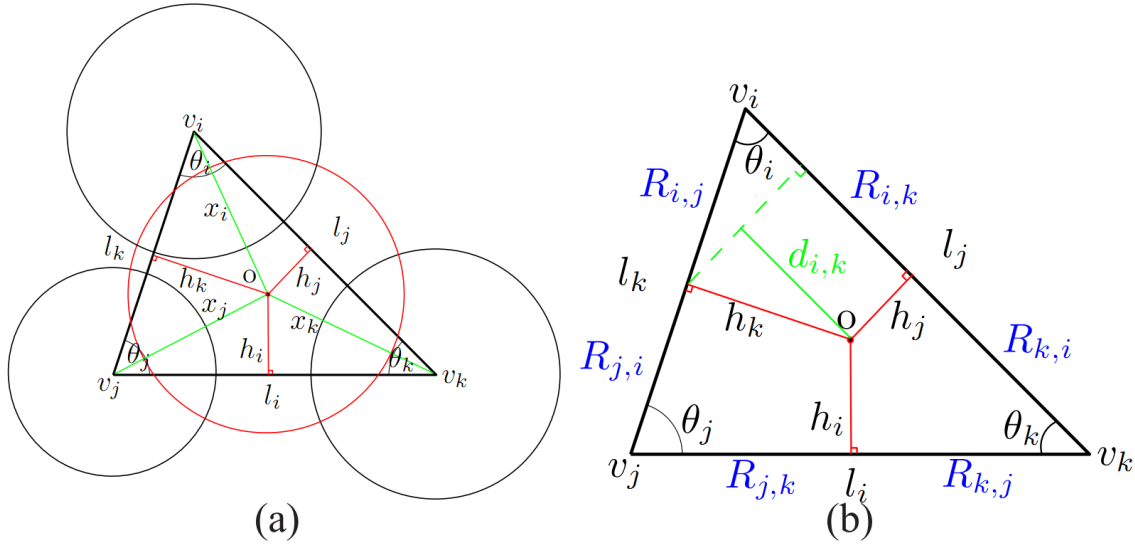


Figure 3.7: Inversive distance circle packing [YGL⁺09].

5. The Hessian matrix Δ could be formulated to solve Equation 2.4 as described in the beginning of this section. The Hessian matrix has an explicit form $\Delta = (h_{ij})$:

$$h_{ij} = \begin{cases} -w_{ij} & i \neq j, [v_i, v_j] \in E \\ \sum_k w_{ik} & i = j, [v_i, v_k] \in E \\ 0 & i \neq j, [v_i, v_j] \notin E \end{cases} \quad (3.16)$$

where

$$w_{ij} = \frac{\partial \theta_i^{jk}}{\partial u_j} + \frac{\partial \theta_i^{jl}}{\partial u_j}. \quad (3.17)$$

6. After solving the following linear system

$$\Delta \mu = \bar{K}_i - K_i, \quad (3.18)$$

the obtained μ could be taken to update the current u_i by

$$u_i \leftarrow u_i + \mu_i. \quad (3.19)$$

7. The γ_i could be updated by

$$\gamma_i = e^{u_i} \quad (3.20)$$

and then back to Equation 3.11 to compute l_{ij} for the next iteration.

8. The process executes until the terminal condition

$$\max_i |\bar{K}_i - K_i| < \varepsilon, \quad (3.21)$$

where ε is set as 10^{-3} in our experiment, is true.

Since we only shows the algorithm of the discrete Ricci flow, if more statements or proofs of each equation is needed, it is recommended to refer [JKLG08] and [YGL⁺09]. In the next section, the obtained flat cone metric would be flattened on the common domain, which could construct the correspondence between two input triangle meshes.

3.4 Surface mapping

In the previous section, the flat cone metrics (global parameterizations) of both input meshes are obtained. The goal of this step is to flatten the global parameterizations onto the common base domain and return the final result of the inter-surface mapping. Since the flat cone metric only represents the length of edges, while we flatten the metric patch by patch, a region growing method like Step 4. in [JKLG08] is needed. The following is our flattening algorithm: First, a patch of the common base domain is chosen. Since the feature points of the input mesh and the vertices on the common domain are already in correspondence, it is easy to find three mapped feature points on the mesh. Second, we choose a geometric center to be a seed point of the region growing method and start to flatten. The difference between the flattening algorithm of [JKLG08] and ours is that our algorithm stops while the chosen three feature points are

flattened, since we only need a local region which is enclosed by three feature points in the parameterization domain. Figure 3.8 shows a completely flattened result on patches of the horse head.

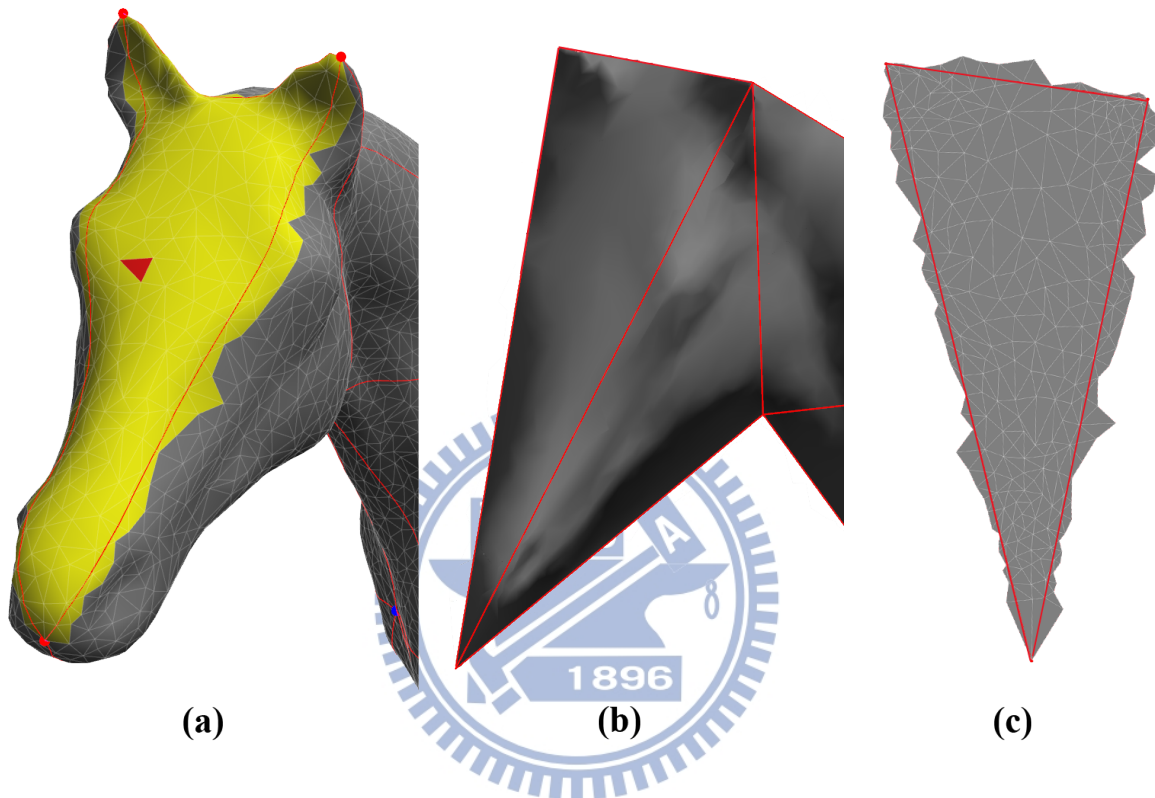


Figure 3.8: The flat cone metric on the horse head. (a) The red face and the yellow region of the left image are the guessed starting face and the locally flattened area of that patch respectively. (b) The parameterized faces on the common domain. (c) The parameterized result of the yellow region on the uv coordinates.

Next, we repeatedly apply region growing flattening algorithm for each patch until all patches are done. After the two input meshes are applied with the same algorithm, we could get $f_{S \rightarrow D_S}$ and $f_{T \rightarrow D_T}$ mentioned in Section 3.1. Since D_S and D_T are with same shape from D , the function $f_{D_S \rightarrow D_T}$ is also generated intuitively. A surface mapping $f : S \rightarrow T$ is thus completed. (Figure 3.9) Since our initial feature mapping with this method is based on user-assigned, the result of correspondence looks acceptable. If any relaxation is needed, one could

design an energy function like some previous methods of cross parameterization and relax the vertices on this kind of common base domain. Note that the flat cone metrics generated by the discrete Ricci flow does not exist patch boundaries; therefore, in theory, vertices could move from patch to patch to get a global optima of the energy function while relaxing.

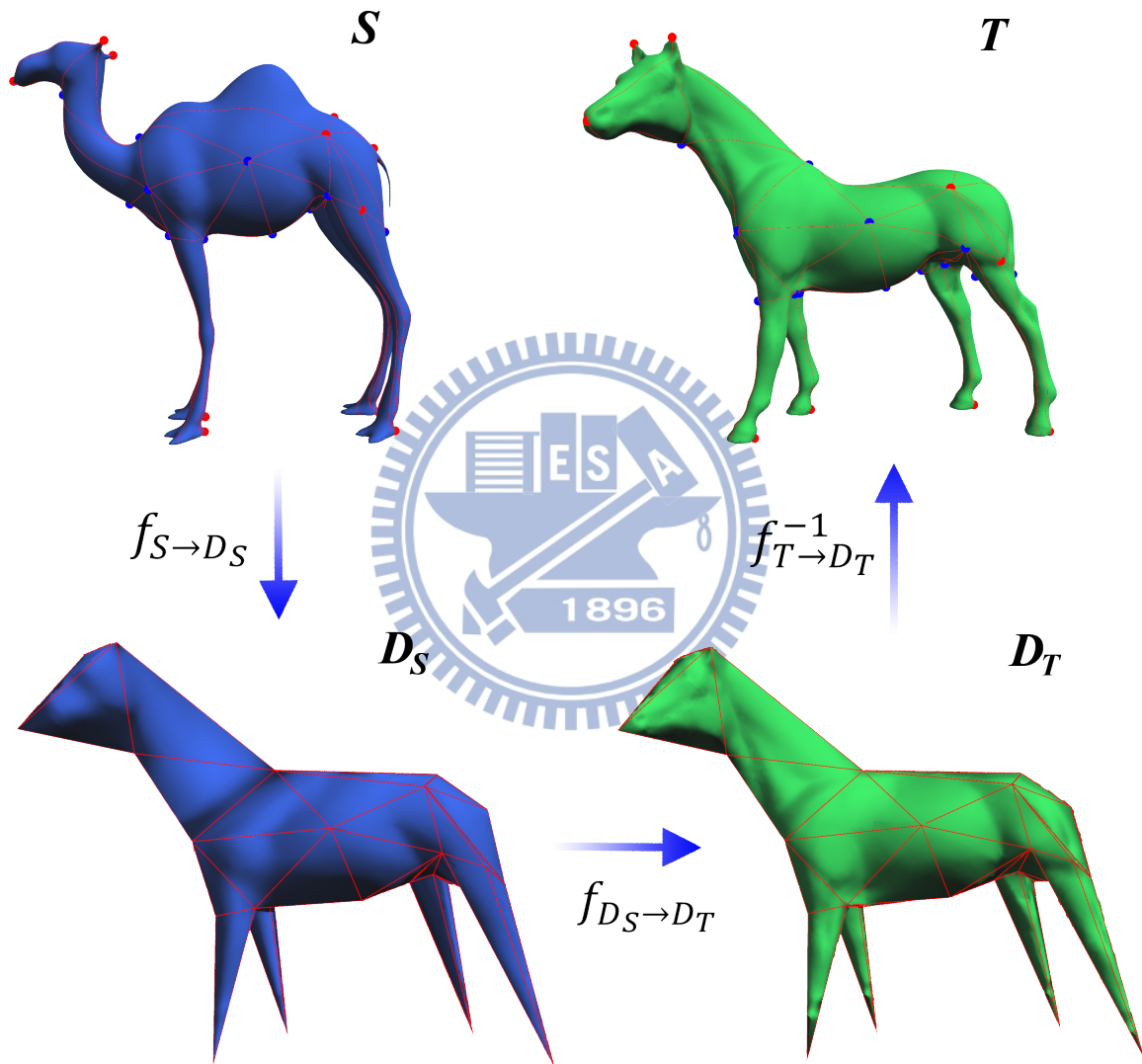


Figure 3.9: The final inter-surface mapping. Note that the patch lines on S and T are the intersections between the parameterized edges and the edges of the common base domain D_S and D_T respectively.

Experimental Result

In this chapter, we present the results of the proposed inter-surface mapping algorithm in Section 4.1. We also show that the target Gaussian curvature defined by our method is quasi-isometric in Section 4.2.



4.1 Result

Our inter-surface mapping scheme is based on the base-complex domain which can be derived by using a mesh simplification. Users are allowed to pick some corresponding feature points on source and target meshes, as described in Section 3.1. These points will be cone singularities, as the input of the discrete Ricci flow. The feature points are enforced to be the constraint points in the QEM simplification; in other words, the feature points will be retained during simplification. We observed that the common base domain of 50-70 faces is enough for preserving shape properties, which means that about 33 feature points should be assigned by users. Since the common base domain have the same topology with two input models, they have the same total Gaussian curvature according to Euler characteristic, and we could directly propagate Gaussian

curvatures for all vertices of the common domain to the corresponding feature points on both models. Next, the discrete Ricci flow is applied to source and target models to generate the flat cone metric of two models. After the discrete Ricci flow step the flat cone metric could be directly flatten (global parameterize) onto the common domain and the inter-surface mapping could be derived since the correspondence is constructed.

We demonstrate the correspondence by transferring texture from the target T to the source S , since the mapping function is from source S to target T . The texture for T is to flatten the flat cone metric, which means global parameterize, to the 2D domain and then map the texture to the target T , as shown in Figure 4.1. So it is naturally to have some texture boundaries on T and also on S . Note that the global parameterization of T on the 2D domain and on the common base domain are certainly the same.

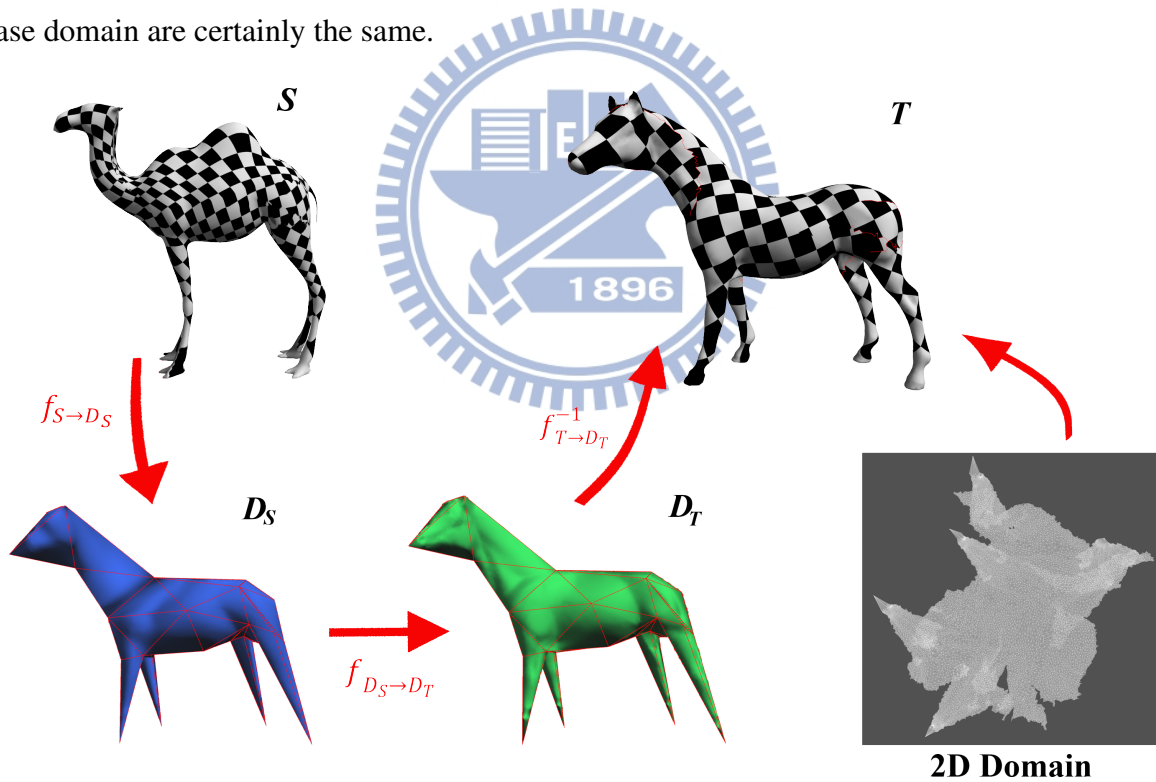


Figure 4.1: The texture transferring via the inter-surface mapping. Texturing target T via the 2D domain. And S gets the color from T through $f_{S \rightarrow D_S}$, $f_{D_S \rightarrow D_T}$, and $f_{T \rightarrow D_T}^{-1}$.

Figure 4.2, Figure 4.3, and Figure 4.4 show the inter-surface mapping from Camel to Horse

with different textures. Since the texture coordinates of Horse (T in this case) are from the 2D domain (Figure 4.1), it is inevitable to show texture boundaries (red lines on Horse). It could also be considered as "cuts" for unfolding a closed meshes to the 2D domain. Figure 4.5 is a simple example to show the texture boundary that is visually discontinuous but in fact it is a global parameterization, which means sides between texture boundaries are continuous.



Figure 4.2: Using checker board to show the inter-surface mapping from Camel to Horse. The corresponding features colored with red and blue (positive Gaussian curvature and negative Gaussian curvature) are shown on both meshes and the common base domain. Red lines on Horse are texture boundaries on the uv coordinates.

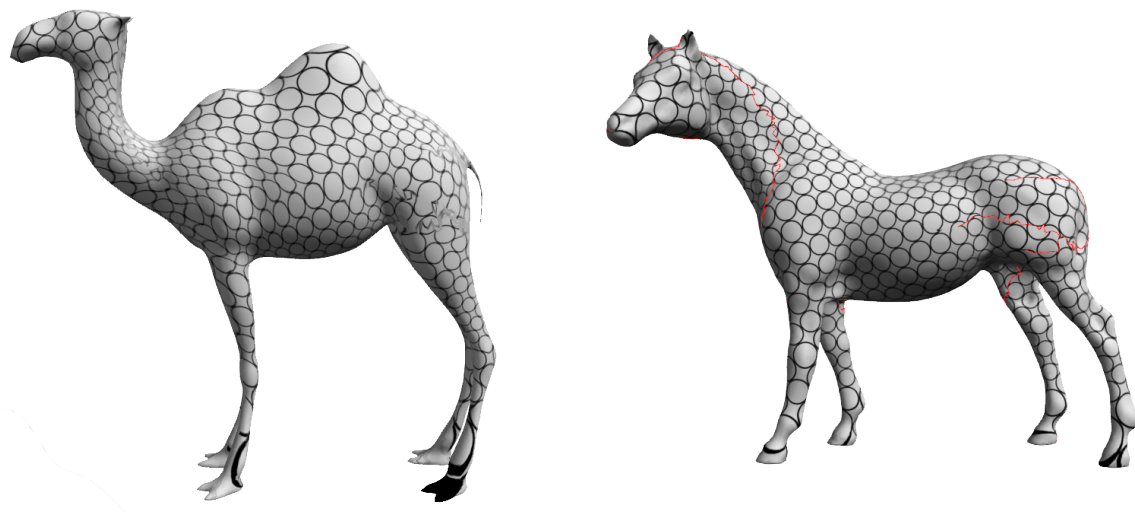


Figure 4.3: Using circle packing to show the inter-surface mapping from Camel to Horse. Red lines on Horse are texture boundaries on the uv coordinates.



Figure 4.4: Using colored ribbon to show the inter-surface mapping from Camel to Horse. Red lines on Horse are texture boundaries on the uv coordinates.

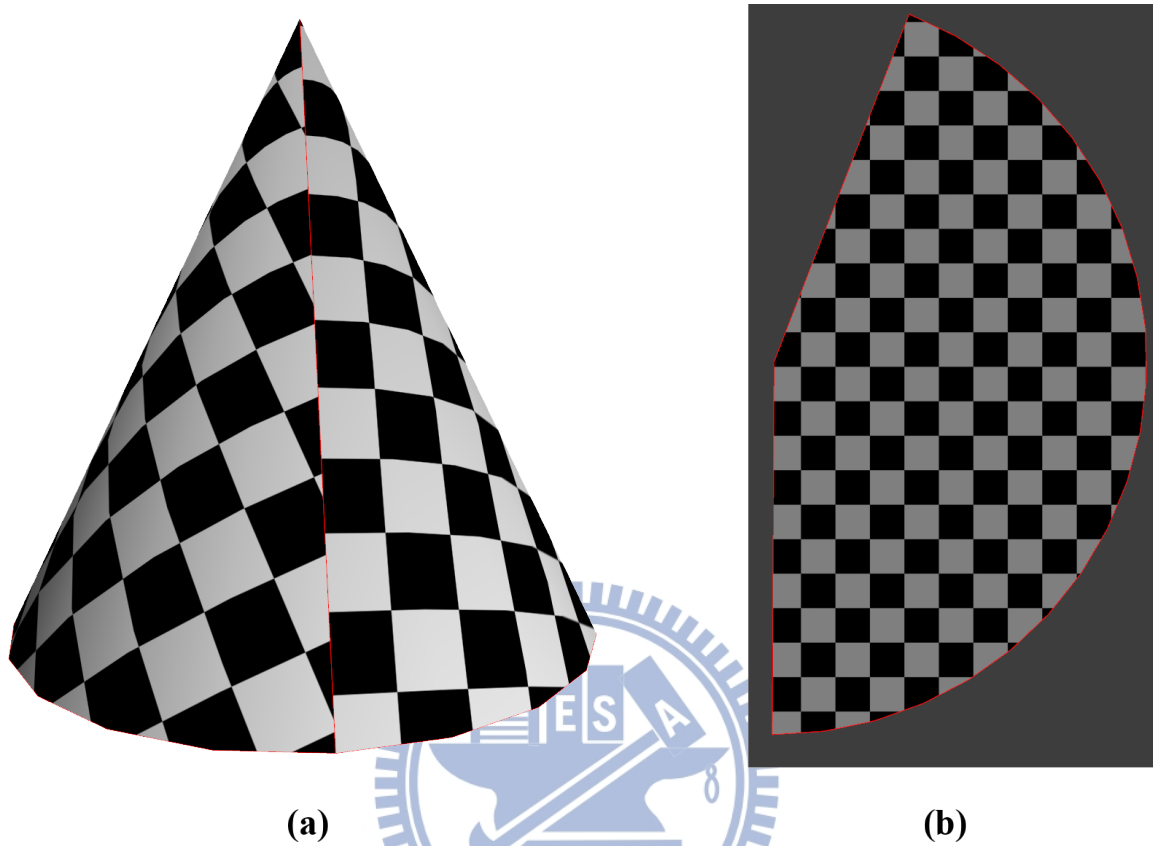


Figure 4.5: Red lines show the texture boundaries. (a) The textured cone. (b) The parameterized cone on the 2D domain.

Although the correspondence is acceptable on head and body of both models (Figure 4.6 and 4.7), one might discover that the mapping between legs is poor (Figure 4.8). The reason is that the discrete Ricci flow does not converge to the assigned Gaussian curvatures at legs. If the discrete Ricci flow completely deforms the original mesh to the metric with zero Gaussian curvatures everywhere except at cone singularities, the patch lines at two sides of gray parts of Camel's left leg should fit together. If more corresponding features are assigned at leg parts, the result may be improved.

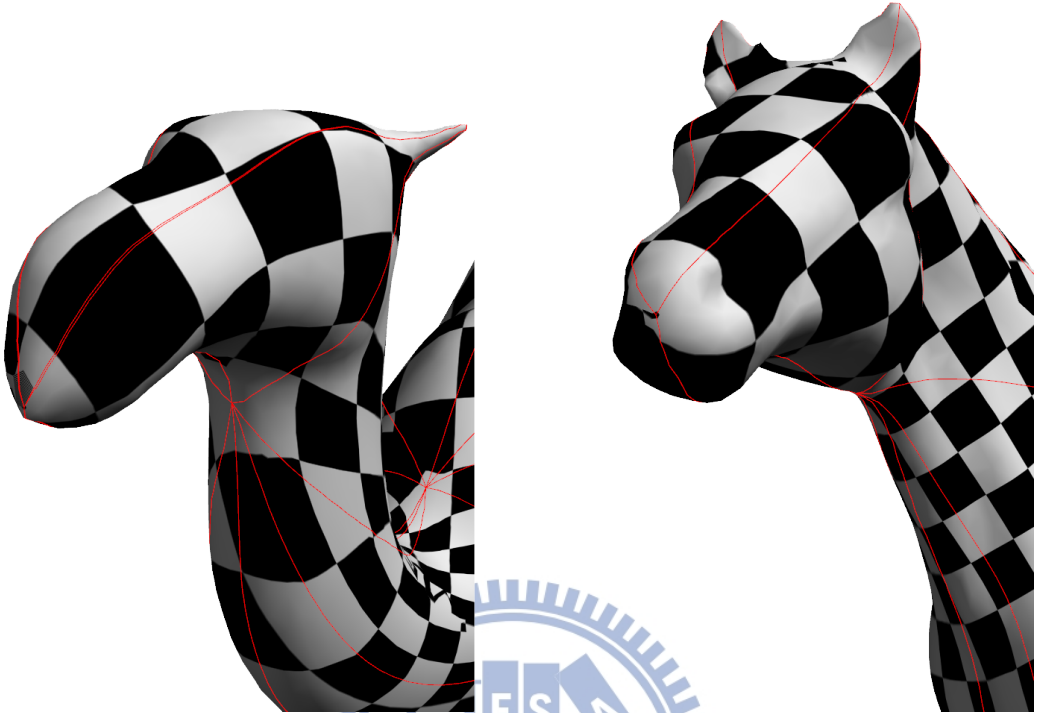


Figure 4.6: Mapping between heads of Camel and Horse.

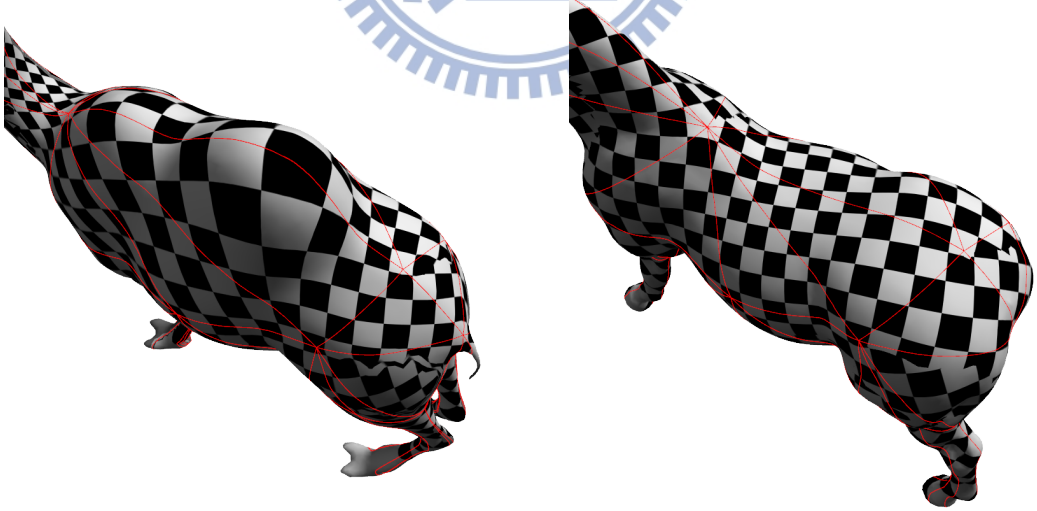


Figure 4.7: Mapping between bodies of Camel and Horse.



Figure 4.8: Mapping between legs of Camel and Horse.

We also include other pairs of models in our testing. Figure 4.9, Figure 4.10, and Figure 4.11 show the result of inter-surface mapping from Wolf to Dog models. Since the discrete Ricci flow converges well in this case, the parameterization does not show any split region like the result of Camel and Horse. However, the correspondence between legs of the models is still poor. The reason is that we do not utilize any mapping information except the assigned corresponding features, although the shape-preserving property of the mesh simplification is enforced. Since the target Gaussian curvatures at the end point of legs are usually sharp, the conformality at those regions could not be maintained well.

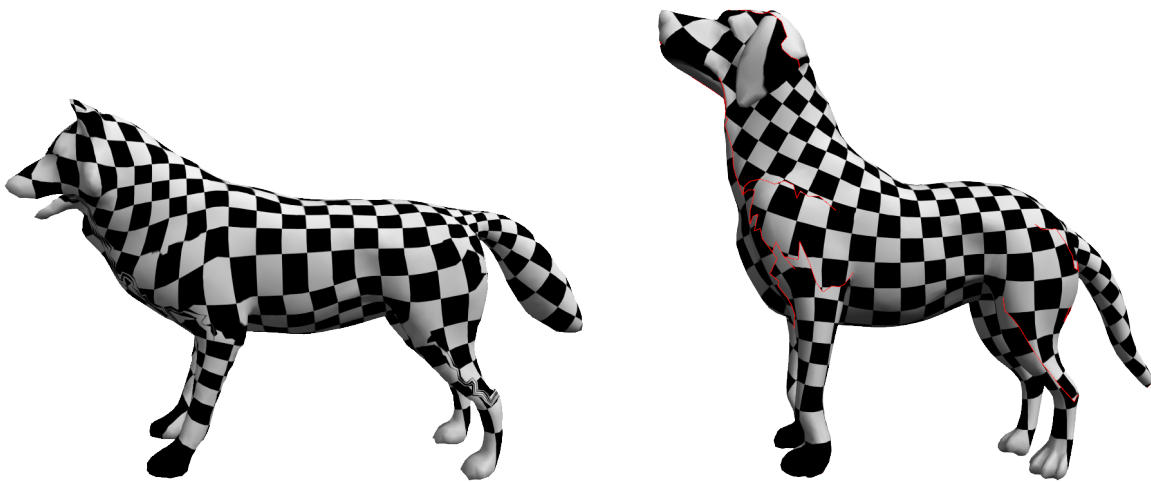


Figure 4.9: Using checker board to show the inter-surface mapping from Wolf to Dog.

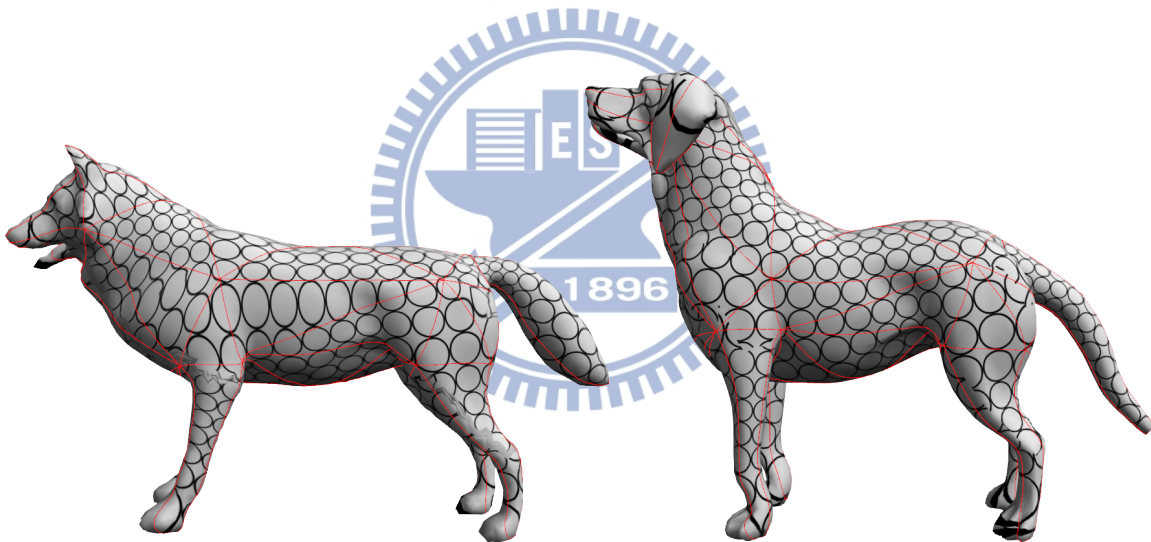


Figure 4.10: Using circle packing to show the inter-surface mapping from Wolf to Dog. Red lines are patch lines.



Figure 4.11: Using colored ribbon to show the inter-surface mapping from Wolf to Dog.

Our algorithm is also suitable for a same model with different poses. Figure 4.12, Figure 4.13, and Figure 4.14 shows the results which show better results on leg parts. The reason is that if the shapes are similar, and the discrete Ricci flow would generate similar flat cone metrics.

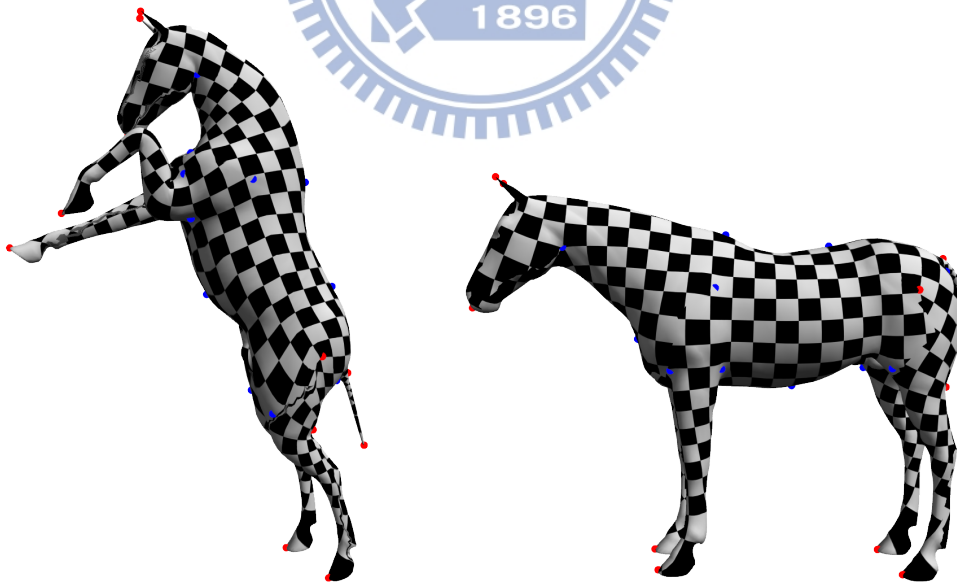


Figure 4.12: Using checker board to show the inter-surface mapping between the horse models in different poses. Red and blue dots are cone singularities.

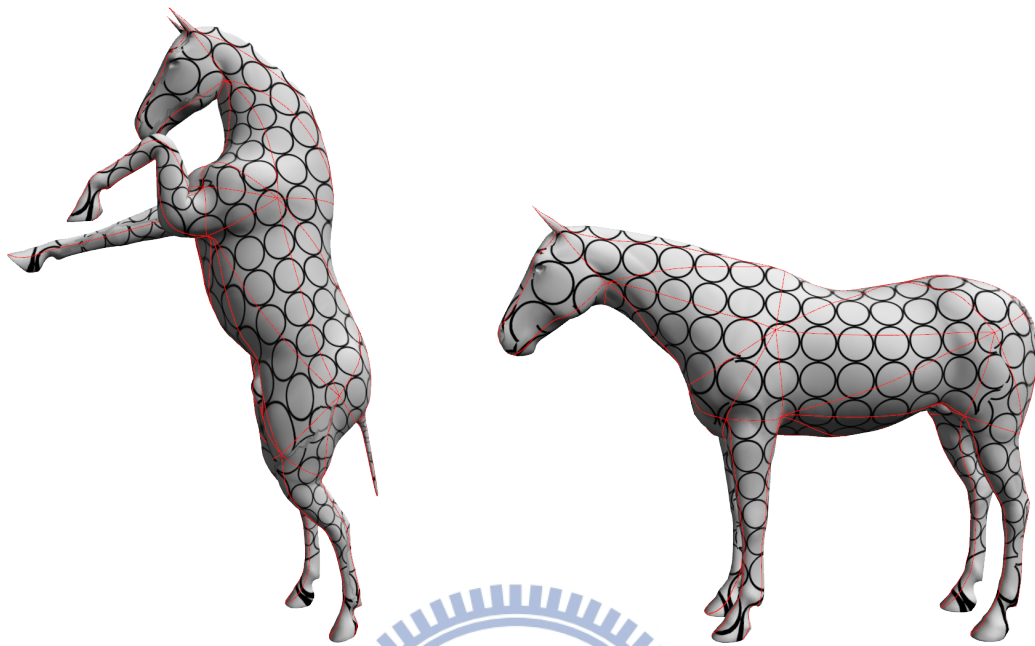


Figure 4.13: Using circle packing to show the inter-surface mapping between the horse models in different poses. Red lines are patch lines.

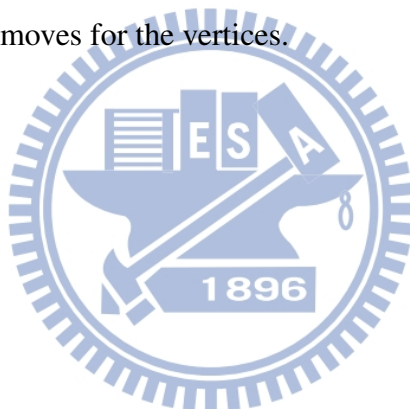


Figure 4.14: Using colored ribbon to show the inter-surface mapping between the horse models in different poses.

4.2 Comparison of different target Gaussian curvatures.

As we mentioned in Section 1.1, the cone singularity set we defined is quasi-isometric. Since the discrete Ricci flow computes conformal metrics, if the target Gaussian curvature set is not assigned carefully, the result of the metric would have high area distortion.

Figure 4.15 shows the results for models with the same distribution of cone singularities but with different sets of target Gaussian curvatures. It is observed that the uniform target Gaussian curvatures lead to high area distortion since the vertices move largely during the discrete Ricci flow. However, the way we assign the target Gaussian curvature results in a quasi-isometric result. The reason is that the target shape is close to the original shape, so the generated conformal metric does not result in great moves for the vertices.



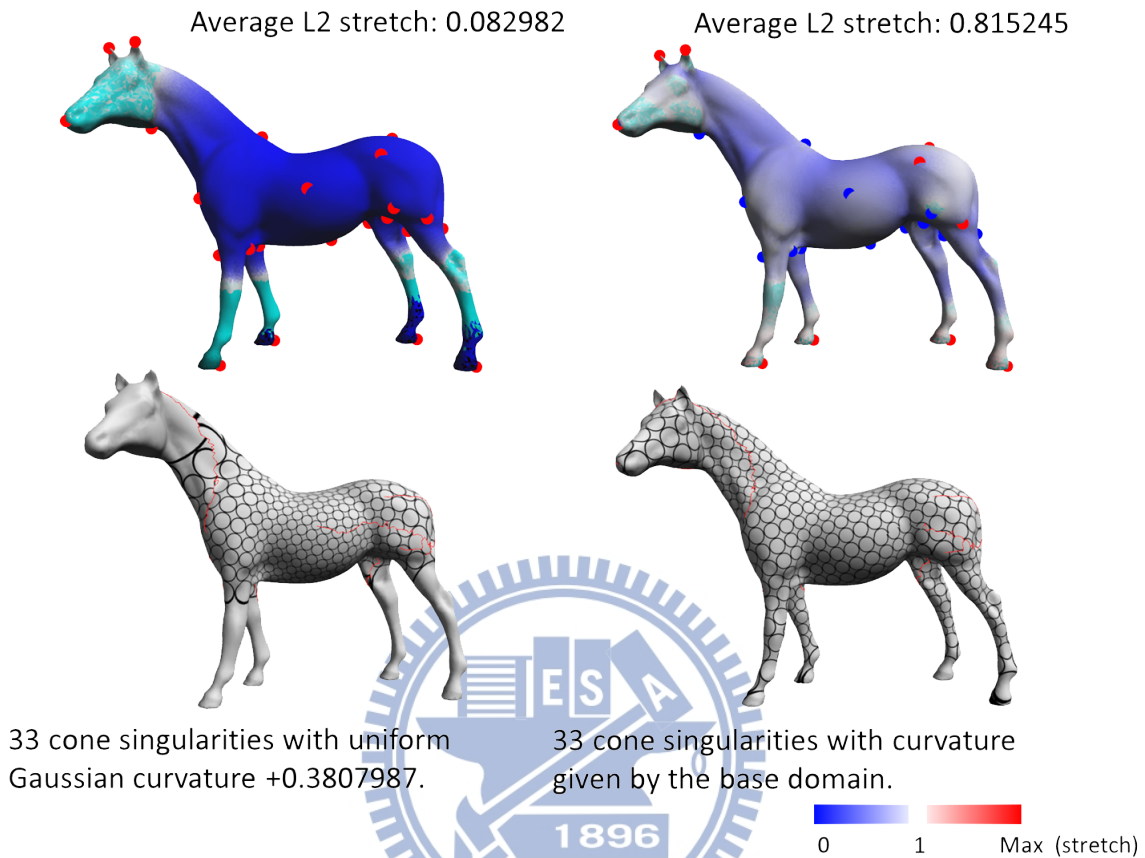


Figure 4.15: A comparison for different set of target Gaussian curvatures. The horse on the left column shows a conformal metric with high area distortion if the target Gaussian curvature is uniformly distributed. The mesh on the right column shows better result since the target Gaussian curvature is assigned with the Gaussian curvatures of vertices of the common base domain. L2 stretches are also shown to depict that our method lead to a quasi-isometric result (An L2 stretch will be 1 if the metric is completely isometric to the original metric).

On the other hand, since our cone singularity set are assigned with the target metric that is close to the original metric, the iterations of the discrete Ricci flow would be also smaller; in other words, the converging is faster since the shape of the common base domain is close to the shape of the original surface. Figure 4.16 shows the speed of the convergence between two meshes with different sets of target Gaussian curvatures.

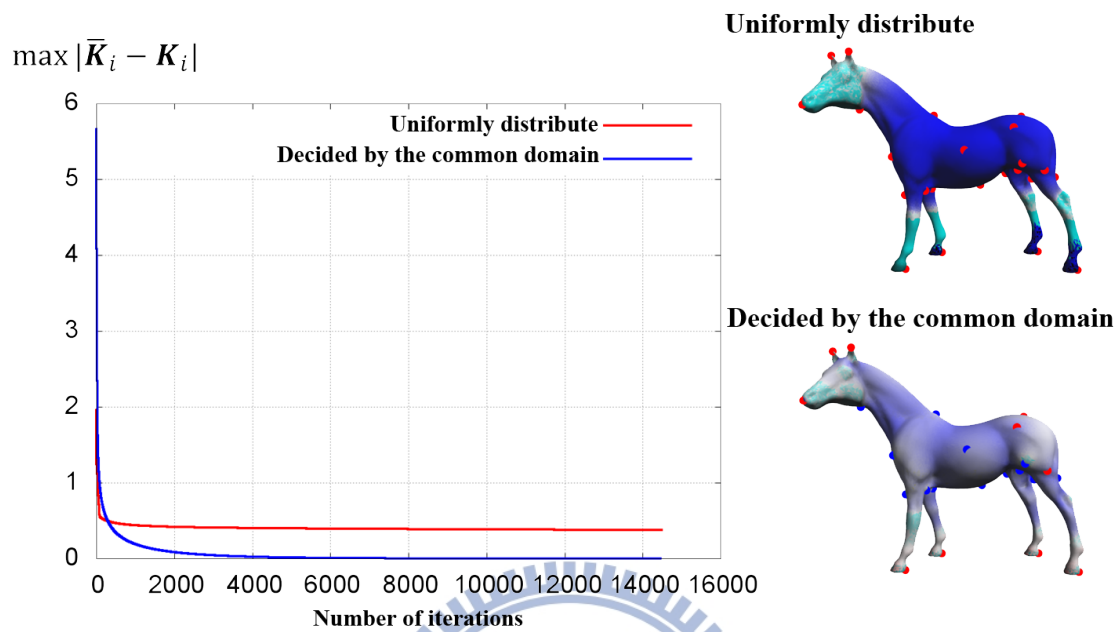


Figure 4.16: The case with uniformly distributed target Gaussian curvature will cost more time than the one that the target Gaussian curvatures are assigned with the Gaussian curvatures of vertices on the common base domain.

Figure 4.17 and 4.18 are the results of the wolf models with different poses. Same conclusion can be concluded.

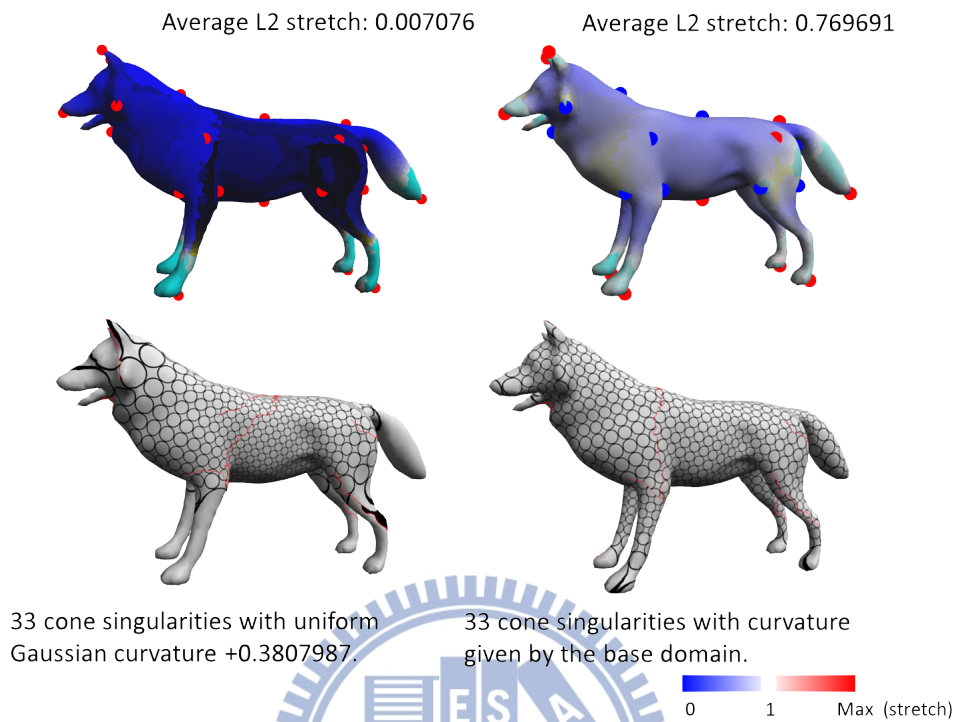


Figure 4.17: The comparison of the conformal metric of the wolf with different set of target Gaussian curvatures. Red lines are the texture boundaries.

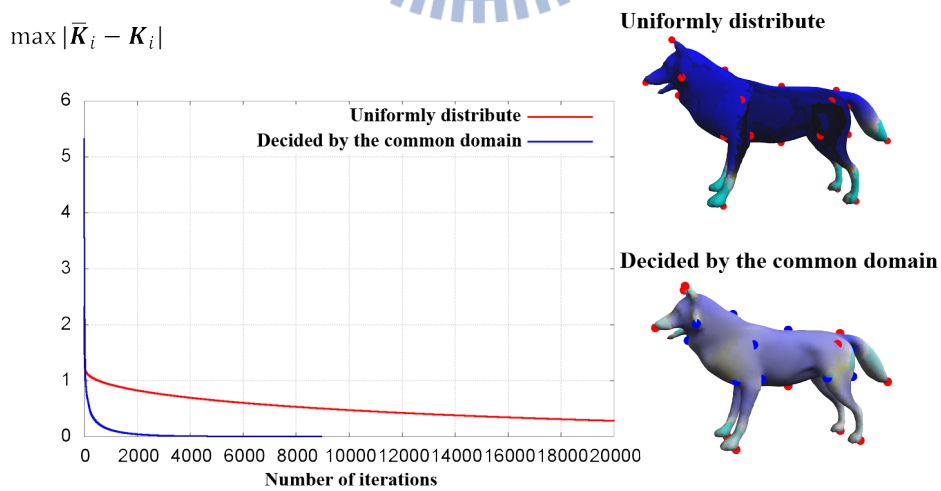


Figure 4.18: The converge speed of different set of target Gaussian curvatures.

Conclusions

In this chapter, we give a summary of our approach, describe the limitation, and some future works.



5.1 Summary

We have proposed a novel algorithm to compute the shape correspondence between meshes via the common domain and the flat cone metrics derived by using the discrete Ricci flow. Previous base-complex approaches locally parameterize the segmented patch after the common domain is derived and then apply smoothing operation to ensure the continuity along patch boundaries. Our approach globally parameterize the whole models and flatten the parameterization on the common domain; hence smoothing operation is not required.

The proposed method needs a set of corresponding features assigned by users. After the common base domain is constructed, it gives a shape-preserving cone singularity set for input models. The set makes the input models deform to a shape conformal to the common domain by the discrete Ricci flow. Since the flow returns a flat cone metric, we could flatten the global

parameterization result on any Euclidean domain. For texturing, flat cone metrics are flattened simply to the 2D domain. For inter-surface mapping, flat cone metrics are flattened to the common base domain. Since the inter-surface mapping process does not have any mapping information except feature points assigned by users, if the conformality could not be controlled, or the discrete Ricci flow does not converge well, the result of the correspondence might be poor; otherwise, the result would be acceptable.

5.2 Limitation

Our framework does not suit for the inter-surface mapping shapes with different topologies since it requires input shapes with the same cone singularity settings in order to apply the discrete Ricci flow. In other words, the input shapes cannot have different total Gaussian curvatures. For some meshes with the same topology but highly different shapes the discrete Ricci flow cannot converge well. When the discrete Ricci flow could not reach the global optima successfully with a cone singularity set decided by the common domain, one way to resolve the problem is to rebuild the base-complex since the decided target curvature space is not always admissible for both inputs (see "Admissible Curvature Space" in Section IV of [JKLG08]). We can also smooth the input mesh if the quality of the input mesh is too bad such that the discrete Ricci flow fails even using the method of Section 5.4 in [YGL⁺09].

5.3 Future work

Since our approach needs many corresponding features to be assigned by users, it is desirable to apply a simplification algorithm to derive the common base domain. Our algorithm does not take any correspondence information, except the corresponding feature points, into account during the inter-surface mapping. For a better inter-surface mapping, it is desirable to apply a relaxation step after the initial inter-surface mapping is constructed. In our framework, the global parameterizations are derived and are continuous from patch to patch while the initial

surface mapping is built, we could take the advantage of this property to design a novel energy function that allows vertices to cross patches while relaxing. Another direction is to apply our algorithm to models of higher genus by using some other topology-preserving mesh simplification method. A remeshing process could be also applied to our results, and compare the shape using PSNR (Peak signal-to-noise ratio). Since the texturing in the tests shows discontinuity along texture boundaries, some other rendering approaches, such as texture with the global flow, may be used to depict a better quality of the results. Besides, the discrete Ricci flow for meshes with genus zero is slower than that with high genus. Improvement on computation and efficiency will definitely make the proposed method more practical.



Bibliography

- [CL03] B. Chow and F. Luo. Combinatorial Ricci flows on surfaces. *Journal Differential Geometry*, 63(1):97–129, 2003.
- [Flo03] M. S. Floater. Mean value coordinates. *Comput. Aided Geom. Des.*, 20:19–27, 2003.
- [GH97] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *ACM SIGGRAPH 97 Conference Proceedings*, pages 209–216, 1997.
- [GY08] X. Gu and S. Yau. *Computational Conformal Geometry*. International Press, 2008.
- [Ham82] R. S. Hamilton. Three manifolds with positive Ricci curvature. In *Journal of Differential Geometry*, volume 17, pages 255–306, 1982.
- [Ham88] R. S. Hamilton. The Ricci flow on surfaces. In *Mathematics and general relativity (Santa Cruz, CA, 1986)*, *Contemp. Math. Amer.Math.Soc. Providence, RI*, volume 71, 1988.
- [HLS07] K. Hormann, B. Lévy, and A. Sheffer. Mesh parameterization: theory and practice. In *ACM SIGGRAPH 2007 courses*, 2007.
- [JKLG08] M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface Ricci flow. *IEEE Trans. on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.

- [KLS03] A. Khodakovsky, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. *ACM Trans. on Graphics*, 22(3):350–357, 2003.
- [KS04] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 861–869, 2004.
- [LBG⁺08] 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.
- [LDSS99] A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proceedings of SIGGRAPH 99*, pages 343–350, 1999.
- [LJX⁺10] Y. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S. Hu, and X. Gu. Metric-driven rosy field design and remeshing. *IEEE Trans. on Visualization and Computer Graphics*, 16(1):95–107, 2010.
- [LSS⁺98] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 95–104, 1998.
- [PSS01] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 179–184, 2001.
- [PWPY07] J. Pan, H. Wu, C. Pan, and Q. Yang. A novel scheme for efficient cross-parameterization. In *Proc. Conf. Computer Graphics Int'l (CGI)*, 2007.
- [SAPH04] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 870–877, 2004.

- [vKZHCO10] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Proc. of Eurographics State-of-the-art Report*, 2010.
- [YGL⁺09] Y. Yang, R. Guo, F. Luo, S. Hu, and X. Gu. Generalized discrete Ricci flow. *Computer Graphics Forum, Special issue of Pacific Graphics 2009*, 28(7):2005–2014, 2009.

