Volumetric Focus+Context Visualization Techniques

A Thesis Presented

by

Xin Zhao

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

Stony Brook University

Dec 2013

Stony Brook University

The Graduate School

Xin Zhao

We, the thesis committee for the above candidate for the Master of Science degree, hereby recommend acceptance of this thesis.

Arie Kaufman – Thesis Advisor
Distinguished Professor, Department of Computer Science

Xianfeng Gu – Chairperson of Defense
Associate Professor, Department of Computer Science

Allen Tannenbaum
Professor, Department of Computer Science

This thesis is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Thesis

Volumetric Focus+Context Visualization Techniques

by

Xin Zhao

Master of Science

in

Computer Science

Stony Brook University

2013

This thesis introduces new techniques and applications for volumetric visualization. Focus+context visualization and interaction techniques are used to navigate and interact with objects in information spaces. They provide in-place magnification of a region of the display without consequently losing any context representation. Diverse focus+context visualization techniques are of broad use in different application domains, such as geovirtual environments, navigation and visualization of large graphs or hierarchies, as well as the volume rendering (e.g., for medical applications). However, how to accurately represent and highlight the focus objects while maximally keeping all the important context information (e.g., shape features and area size) becomes a major challenge. To overcome the limitations generated by traditional optical lenses and to effectively facilitate the data exploration and analysis (e.g., organ segmentation and cancer detection for the medical data), new focus+context methods have been proposed and used for the design of real-time volumetric visualization techniques for both 2D and 3D applications. In general, detailed views of a focus volumetric object or multiple objects are combined seamlessly with abstracted or compressed views of the context within a single rendered image. To perform the real-time display required for interactive visualization, dedicated parallel processors (GPUs) are used for computing and rendering. For this purpose, the design and implementation of appropriate computer graphics and modeling based techniques and visualization rendering pipelines are necessary. Meanwhile, effective and efficient highlighting can enable users to quickly locate and easily decode relevant information. Therefore, high-dimensional transfer functions are used as highlighting techniques for the visualization of various objects-of-interests. With the purpose of exploration and navigation of the volumetric data, there are basically three categories relevant to the scope of this thesis. First component focuses on the enhancement methods: two high dimensional transfer function systems are proposed to accurately segment ROIs in 3D medical data and provide the enhanced visualization display to allow the user to easily perceive the focus data. Second part describes and introduces the focus+context visualization techniques. Two frameworks are based on geometric theories to generate focus+context visualization styles with angle-preservation or area-preservation. The conformal magnifier, works as a novel geometric model based lens design framework to serve as the focus+context visualization for various medical applications, which provides a smooth transition between focus and context regions and optimized local shape preservation everywhere. Meanwhile, the area-preservation visualization is obtained using a novel area-preservation mapping method based on the Monge-Brenier theory based optimal mass transport technique, which is rigorous and solid in theory, efficient and parallel in computation, and general for various applications.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my thesis advisor, Professor Arie Kaufman for his guidance on research, and his help at various stages of my studies. I would like to thank Professor Xianfeng Gu for his kind advice, as well as for serving on committees. I would also like to thank my colleagues in our center of visual computing and computer science department for delightful collaborations and discussions we had together, and thank my family and friends for their support and help.

I would like to thank Stony Brook University Hospital for providing all medical related datasets. And my research was supported in part by the following grants awarded to Professor Arie Kaufman: NSF grants IIS-0916235, CNS-0959979, IIP-1069147, CCF0702699 and CNS0959979, and NIH grant R01EB7530.

Chapter 1

Introduction

This chapter introduces highlighting techniques for the 3D visualization, mainly focusing on transfer function design for volume rendering, which can help the user easily and efficiently display and detect the object or region of interest for the volumetric focus+context (F+C) visualization. Next, a brief overview of the state-of-the-art F+C visualization methods is presented. Then it discusses the potentials of geometry based mapping techniques for the F+C visualization, the core topic of this thesis, and how these techniques facilitate the data exploration and analysis for various applications. The chapter closes with problem statements and related techniques and their contributions.

1.1 Problem Statements and Solutions

The focus of this thesis is to develop efficient volumetric F+C visualization and present its applications. This section summarizes the conceptual and technical challenges as well as the contributions presented.

The application of volumetric F+C visualization is faced by a number of challenges and problems. First, the highlighting techniques should be used to accurate segment the objects of interest and direct the viewers focus of attention by facilitating the pre-attentive cognition. Second, it should efficient use of screen space to increase the amount of information visible of focus regions, while keeping all the context areas. The goal is to maintain the feature understanding in 3D space and preserve the properties (such as angle or the size of area) of spatial information after applying F+C visualization techniques. Third, it should be suitable for both 2D and 3D data, and simplify the exploration and analysis of complicated volumetric data with an automatic or semi-automatic system with minimal user interactions. A further challenge represents at generating high quality rendering results for various visualization needs. In particular this includes the support for multiple focus regions as well as smooth transitions between focus and context regions. Last technical challenge lies at the real-time enabled implementations. In order to have hardware accelerated system, suitable data structures are required to store data efficiently; and efficient algorithms are required to apply these data structures within a programmable GPU pipeline. In other words, these algorithms should be applicable to different large and complicated datasets with the real-time rendering and displaying.

Aim to solve these challahes, this thesis has the following research topics:

- Focus enhancement methods: Two high dimensional transfer function systems to accurately segment ROIs in 3D medical data and provide the enhanced visualization display to allow the user to easily perceive the focus data.

- Deformation based Focus+Context visualization: An interface scheme that allows the user to work at, and move between, focused and contextual views of a data set, using moving least squares (MLS) [6] based deformation methods.

- Angle-preservation lens design: A novel geometric model based lens design framework to serve as the Focus+Context visualization for various medical applications, which provides a smooth transition between focus and context regions and optimized local shape preservation everywhere.

- Area-preservation visualization: A novel area-preservation mapping method using the optimal mass transport technique, based on the Monge-Brenier theory, which is rigorous and solid in theory, efficient and parallel in computation, and general for various applications.

1.2 Highlighting Techniques

Highlighting functionality is an essential component of a visualization framework for direct data exploration and flexible user interactions. For the volumetric F+C visualization, highlighting techniques enable the user to easily perceive and select the targeted object. The highlighted regions typically represent features relevant for specific user tasks and facilitate effective user exploration through a volumetric dataset. With the increasing amounts of data visualized on a wide range of applications, a good system demands the visualization, especially the F+C visualization, with clear separations and differences. Here, effective and efficient highlighting methods can enable the user to quickly locate and easily decode relevant information for various applications. The highlighting effect is achieved by modifying the appearance in which an object or a region is usually depicted. Such appearance modification can be as simple as overdrawing a dominant color that can easily be distinguished from all other colors in the volume data. Modifications can be applied to an object or an area that should be highlighted (focus-based) or to be faded away (context-based), served as an alternative focus-based highlighting style.

To convey greater recognition of the objects being rendered, transfer functions are extensively used. The transfer function system modifies the way in which items are depicted or displayed using the volume rendering, in order to highlight, suppress, or contextualise them. For example, the objects of interest can be highlighted using colors or opacities, or the non-focused objects could be visually deemphasized, to draw the users attention back to the regions of interest. The transfer function design as a kind of highlighting technique, depends on the availability of the texture element or clustering information in the dataset, and can be used to enhance the volumetric F+C visualization. In this thesis, each presented system or framework has a suite of 1D, 2D and high dimensional transfer functions. The specified transfer function allows the enhancement and selection of regions and features of interest, and further interacts with 2D or 3D visual presentations of the dataset directly to display the accurate and visual-pleasure magnification results.

1.2.1 Volume Rendering

To display volumetric data in 3D, volume rendering methods have been extensively developed [7–10]. Direct volume rendering (DVR), as a rendering method in the computer graphics pipeline, allows for the volume to be directly rendered without the requirement to extract surfaces. Various rendering styles can also be implemented, which yields a 2D image that having maximum intensity projection (MIP) or appearing similar to an x-ray [11].

In order to perform DVR, an optical model should be built [12]. The emission-absorption model (in which the elements of the volume are considered as particles within a cloud which are able to both emit their own light and absorb incident light) is the most common model:

$$I(D) = I_0 T(D) + \int_0^D g(s) T'(s) ds, \tag{1.1}$$

where $I(D)$ represents the radiance reaching the camera from D. There are two term represents of this equation: first is the background illumination

(I_0) multiplied by the transparency of the cloud ($T(D)$), and second is the integration over all sample positions s, multiplying each sample's source value ($g(s)$) with the transparency between s and the eye ($T'(s)$). This volume rendering integral in the continuous domain can be further discretized into compositing for use with discretely sampled data. Two composition schemes are used: the back-to-front and the front-to-back methods. For back-to-front compositing, at each sample step, performs:

$$C_{dst} \leftarrow (1 - \alpha_{src})C_{dst} + C_{src}. \tag{1.2}$$

For front-to-back compositing, at each sample step, performs:

$$\begin{aligned} C_{dst} &\leftarrow C_{dst} + (1 - \alpha_{dst})C_{src}, \\ \alpha_{dst} &\leftarrow \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src}. \end{aligned} \tag{1.3}$$

In both equations, C_{dst} and C_{src} are the destination and source colors, while α_{dst} and α_{src} are the destination and source opacities. Note that while front-to-back compositing requires extra maintenance of the opacity term through the integration, it also allows for early termination of the composition when the opacity reaches a sufficient level (e.g., $\alpha_{dst} = 0.95$). A number of methods focus on performing the actual volume rendering. A technique has been introduced where the voxels are *splatted* onto the screen space and rendered as disks [13]. *Shear warp* has been proposed where the viewing transformation would be factored into a 3D shear parallel to the volume slices, a projection will create a distorted image, but a 2D warp is used to undistort the final image [14]. The volume as a whole can also be decomposed into individual slices which are then rendered and composited with typical 2D texture rendering in the graphics pipeline. The well-known ray casting method involves shooting rays through the volume data and sampling at regular points along the rays [15]. Ray casting typically provides the best image quality and is the preferred method of performing DVR [16]. Meanwhile, the specialized graphic hardware for volume rendering has developed and applied [17–19]. The graphics processing unit (GPU) used for gaming provides a high level of parallel performance at a relatively low cost and can allow for real-time ray casting [20]. More recently, general processing on the GPU has become popular, with NVidia's C-like CUDA language being widely adopted [21].

1.2.2 Transfer Function Design

When rendering volumes, which often consist of singular scalar density values for each voxel, it is desirable to map these scalar values to optical properties, such as color and opacity. The transfer function (TF) is used to view a certain part of the volume, by assigning RGB and alpha values for every voxel in the volume. Among all the techniques, 1D transfer function is the most commonly used because of its simple design and implementation. A 1D transfer function maps one RGBA value for every isovalue (e.g., $[0, 255]$ for each channel). However, the 1D transfer function, which is usually based on the scalar values of volume dataset, is very limited for accurate classification of complicated volume dataset. Take CT or MRI medical volumetric datasets as an example, different objects, such as tissues, muscles and bones, may have the same scalar value and need further procedure to distinct them.

In order to solve this issue, other attributes such as gradient magnitude, directional first and second derivative [22], curvature [23] and statistical measures [24] have been introduced and applied as 2D transfer functions to identify more accurate boundaries between different materials. These methods are especially effective in medical contexts, where the feature of interest is often the boundary between two materials. Revealing renderings of internal structures are possible with transfer functions based on the gradient magnitude alone [22], which can be enhanced by modulating opacity according to how orthogonal the gradient vector is to the view vector.

In order to process more complicated cases, the multi-dimensional TF design has become a fundamental and important research thrust. The multi-dimensional transfer functions (nD TF) allow multiple RGBA values to be mapped to a single isovalue. Kniss et al. [25] have introduced dual domain interaction to facilitate identification of 3D boundaries using a probe that facilitates manual segmentation of various materials. Roettger et al. [26] have used the voxel barycenter and the region variance to assist in manual specification of colors for similar features in the process of volume rendering. Canban et al.[27] have used first-, second-, and high-order local statistical texture properties to effectively assign voxels to different opacities and colors using texture-based transfer function. Maciejewski et al. [28] have proposed a novel non-parametric clustering method to design the TF. Although only 2D TF shown as examples, the clustering method can also be extended to design the nD TF. For the high dimensional computation, He et al.[29] and Marks et al.[30] have proposed a solution to the parameter selection problem, where by the user choosing the TF by browsing through many rendered images. Tzeng et al.[31] have presented a new approach to the volume classification problem, relying on an intelligent system to abstract high dimensional mapping functions from the user. The benet of using multiple dimensional transfer functions is that direct volume rendering can be an extremely expressive form of volume visualization because the image can represent such a variety of aspects of the data.

1.3 Focus+Context Visualization

With the tremendous increases in computing power, data storage, and internet bandwidth, the user can now easily store, process, and deliver over the internet very large datasets, but an inherent limitation is the real estate available to display these data. While display devices may have grown in size and resolution, a natural limit is and remains to be the human's visual field of view. At the same time, with the emergence of portable devices, such as netbooks and smart phones, there has also been a reverse trend in screen size for mobile applications. Therefore, no matter what display size is being used, a careful management of the display real estate is directly required. A natural solution to these requirements is the focus+context visualization. The expression *focus+context* is a concept of visually discriminating interesting objects (the focus), from nearby related objects (the context). F+C visualization eliminates the spatial and temporal separation by displaying the focus within the context in a single continuous view. It has been addressed in a great number of applications, including trees [32, 33], treemaps [34], [35], graphs [36], [37], tables [38], city and maps [39], nested networks [40], and 3D models [2], especially for medical data.

1.3.1 Lens based Focus+Context Visualization

The commonly used F+C techniques are lenses and magnifiers, such as fisheye [41], nonlinear magnification transformation [42], detail-in-context [43], distortion [44], multi-scale [39] and others [45, 46]. Fisheye lenses offer an effective navigation and browsing device for various applications [47]. InterRing [48] and Sunburst [49] have applied multi-focus fisheye techniques as an important feature for radial space-filling hierarchical visualizations. Keahey[35] has conceptualized a treemap as an image to show how to compound zooming with a graphical fisheye lens. The fisheye lens displays the data in a continuous manner, having an advantage in the spatial relation preservation. However, it creates noticeable distortions towards its edges and has no method to formally control the focus region as well as to preserve local features in the context region. Therefore, new approaches should focus on minimizing the distortion and formally controlling the focus region.

Various distortion lenses are further proposed to visualize important information in a detailed visualization view [42, 50, 51]. For multiple 2D layers, Bier et al. [52] have presented an interface for the user to enhance features of interest or to compress less interest regions using toolglass and magic lenses. The generalization to handle volumetric data requires extra efforts, but distortion lenses have already been applied to 3D successfully [53, 54]. LaMar et al. [55] have presented a fast, natural and intuitive magnification lens with a tessellated border region that estimates the linear compression according to the radius of lenses and the texture information. Wang et al. [56] have provided a free-form volumetric lens function to highlight, expose and nonlinearly magnify an object in the feature-adaptive or user-configurable way. All the above lenses must follow the physical properties of optical lenses which require explicit and pre-defined optical or physical models for the view distortion. Therefore, new F+C design should be able support the arbitrary model design and the flexible user control.

1.3.2 Deformation based Focus+Context Visualization

In order to overcome the limitation of optical lenses, many F+C visualization methods have been proposed and implemented, such as cutaway views [57], transfer functions [58], ray defectors [59], ray casting [60], real-time rendering [61] and artificial intelligent training and learning [62]. Most of these approaches provide the user a direct and active magnification view, but only support the indirect control of magnification/distortion types using global or data-dependent parameters. Unlike those methods with only active viewing operations, various deformation techniques [63, 64], providing direct operations of both viewing and handling to distort the data according to the user's requirements, are widely used for the F+C visualization. McGuffin et al. [65] have provided a deformation method, interactively allowing the user to open up and peel away the outer layers to reveal the hidden structures. But this method is suffered from the undesired aliasing and undersampling effects. Discontinuous displacement maps, proposed by Correa et al. [64], allow arbitrary deformations and cuts to be applied to the volumetric objects. However, at the cost of computing displacement maps, it is very difficult to create and place in 3D space to obtain a desired deformation. Then, they have provided an extended deformation method for volumetric datasets using scattered data interpolation and radial basis functions on the 2D images or 3D volume datasets to visualize internal features [66]. However, instead of the direct voxel transformation, the user has to take extra efforts to deform a desirable 2D rendered image and then extrude the same deformed type into a 3D displacement map. They have also proposed the illustrative deformation for 3D data exploration by using the combination of geometric or optical illustration operators and examining the best means to prevent the deformed context from being misperceived [67]. The active combination of viewing and handling operations is extremely helpful for exploring a complex or unfamiliar object [68], but may seriously distort context structures.

Mesh-based Deformation. Mesh-based deformation is an alternative method for the F+C visualization based applications. Westermann et al. [69] have defined the deformation as the transformation of nodes located in a proxy mesh generated from the boundary of volume data. Wang et al. [2] have presented an interactive F+C method to visualize large surface models without perceivable distortions based on the energy optimization model. However, the optimal energy models are passive during the deformation: the user fails to have the required non-optimal energy deformation because of the violation of energy stability. In order to eliminate distortions of energy models in the focus region, they [70] have further implemented a volumetric F+C visualization using the global optimization to minimize visual artifacts of salient features.

1.4 Surface Mapping and Flattening

With the fast generation of large and complicated data nowadays, it is desirable to develop new frameworks aiming at generating a visualization of the entire data needed for the navigation, detection, exploration and a global understanding of selected objects or regions of interest (ROIs). Complex geometric structures are often better visualized and analyzed by mapping the surface properties, such as normal map, angle, or area, to a simple canonical domain, such as a rectangle or a sphere. Surface flattening and texture mapping offer a good way of visualizing a surface section by enabling the visualization of all surface parts within a single planar image. In general, surface flattening unavoidably introduces distortions. There are two types of distortions, angle distortion and area distortion. A mapping, which is both angle preservation and area preservation, must be isometric. Therefore, the surface must have zero Gaussian curvature everywhere, namely a developable surface or a ruled surface. For general surfaces, one can only choose either angle-preservation mapping or area-preservation mapping, but never both of them simultaneously.

By comparison, conformal mapping is very applicable for visualization needs because it is angle, and thus shape, preserving [71]. The use of conformal geometry for mapping triangular meshes, where local angles are preserved, has been well established in the field of computer graphics, especially in the creation of texture maps [72] and computer aided detection [5, 73]. However, a conformal method usually substantially distorts area, thus failing to display accurate size of area, including height, width, thickness or diameter of ROIs [74]. Therefore, on the other side, area preservation is also important for many medical applications, especially organ measurement and cancer detection and diagnosis. By comparison, area-preservation mapping can generate accurate and information lossless mapping results, which is a key objective for many medical imaging applications, with the ability to carry out measurements for detecting anatomic abnormalities, such as brain fold detection [75] or colon polyps detection and diagnosis [76, 77].

1.4.1 Angle-Preservation Mapping

Conformal (angle-preservation) mapping has its specially valuable properties, which are extremely suitable for diverse applications:

Angle Preserving - Conformal mappings are angle-preserving. The most common examples of conformal mappings are univalent analytical functions in complex analysis. A more general definition is given in differential geometry [78]. Intuitively, suppose $f : S_1 \to S_2$ is a mapping between two surfaces S_1, S_2, and $\gamma_1, \gamma_2 \subset S_1$ are two arbitrary intersecting curves on S_1, with the intersection point as $p = \gamma_1 \cap \gamma_2$. Then they are mapping to intersecting curves on S_2, $f(p) = f(\gamma_1) \cap f(\gamma_2)$. Suppose at the intersection point p, the intersection angle between two tangent vectors $d\gamma_1, d\gamma_2$ is θ. f is conformal, if and only if the intersection angle between the tangent vectors $df(\gamma_1)$ and $df(\gamma_2)$ is also θ. A formal definition is as follows: $f : (S_1, \mathbf{g_1}) \to (S_2, \mathbf{g}_2)$ is conformal, where $\mathbf{g}_k$ is the Riemannian metric on S_k, $k = 1, 2$, if and only if

$$f^*\mathbf{g}_2 = e^{2\lambda}\mathbf{g}_1.$$

where $\lambda : S_1 \to \mathbb{R}$ is a function, $f^*\mathbf{g}_2$ is the pull back metric induced by f on S_1. Namely, locally a conformal mapping is a scaling transformation, $e^{2\lambda}$ is the scaling factor, therefore it is *shape preserving*.

Intrinsic - Conformal parameterization of a surface is solely determined by its Riemannian metric and does not require its embedding in $\mathbb{R}^3$. For example, one can change a magnifier model by rotation, translation, folding and bending without stretching, the conformal parameterization is invariant.

Stable and Practical - Computing conformal parameterization is equivalent to solving an elliptic geometric PDE [79], which is stable and insensitive to the noise and the resolution of the data. Therefore, a low-resolution magnifier (around 3K vertices) is good enough for most cases. It also effectively accelerates the computation of conformal mapping.

Discrete Ricci flow is a popular method of computing conformal maps of structures [80]. This process acts similar to heat diffusion, with the deformation of the edge lengths being driven by the discrete Gaussian curvature. A circle packing metric is used to approximate the discrete conformal deformation. The final mapping result can be affected by setting the target Gaussian curvature for each vertex in the mesh. There are many applications where discrete Ricci flow is useful, such as in surface parameterization for texture mapping and construction of geometric structures [81], and the optimal surface parameterizations using inverse curvature maps [82]. Applications of Ricci flow will typically use a target Gaussian curvature of zero everywhere, or constrain the boundary to a canonical shape, such as a disc. Discrete Ricci flow has also been used for 3D shape analysis [83]. Based on the circle packing, the Ricci flow method has typically required the triangular meshes to contain triangles that are not obtuse. The discrete Ricci flow method has been generalized using inversive distance circle packing to remove this requirement for the triangulation to lack skinny triangles [84].

1.4.2 Area-Preservation Mapping

Texture mapping on arbitrary surfaces with minimal distortion can preserve the local and global structure of texture [85]. Therefore, the area preservation mapping is important in many medical based research fields, such as the image segmentation, registration, and cancer recognition and diagnosis applications. Most popular solution for area-preservation mapping is inspired by the similarity between a mapping problem and optimal mass transport problem [86], which concerns determining the optimal method, with minimal transportation cost, to move a pile of soil from one place to another. Zhu et al. [87] have combined conformal mapping and area-preservation mapping for flattening branched physiological surfaces, such as vessels. The optimal transport map is carried using the minimal flow approach. Similar method has been applied for image morphing [88]. Rehman et al. [89] have applied the minimizing flow approach for the optimal mass transport with applications to non-rigid 3D image registration. The implementation also employs multi-grid and parallel methodologies on a consumer graphics processing unit (GPU) for fast computation. Dominitz and Tannanbaum [75] have proposed a method to compose conformal mapping with area-preservation mapping, using the technique of OMT, based on Monge-Kantorovich theory [90], which accurately preserves the area element and also maximally preserves the angle. However, the methods based on Monge-Kantorovich's theory and approaches, require n^2 variables. For example, in image registration applications, a 1024×1024 image would result in 2^{40} variables, the storage cost is thus very high, and the computation is extremely expensive.

In contrast, using Monge-Brenier's approach [91] to compute the optimal transport map, a discrete algorithm is solidly based on the variational principle [92]. Gu et al. [92] first compute an area-preserving map from a metric surface

$(S, \mathbf{g})$, where $\mathbf{g}$ is the Riemannian metric, to the planar disk $\mathbb{D}$. The method starts with an angle preserving $\phi : S \to \mathbb{D}$, which introduces area distortions on the disk; the area distortion factor is used to define a measure on the disk, denoted as μ. Then, an optimal mass transport map is computed between the disk with this measure and the disk with the Euclidean measure $dxdy$, $\psi : (\mathbb{D}, \mu) \to (\mathbb{D}, dxdy)$. The composition $\psi \circ \phi : S \to \mathbb{D}$ gives the area-preserving map. Basically, the method only discretizes the target space Y, and finds a convex function whose gradient gives the optimal transport map. Finding the OMT is equivalent to optimizing a convex energy, which can be efficiently achieved using Newton's method. The whole computation requires only n variables. Therefore, this method greatly reduces the computation cost and improves the efficiency.

Chapter 2

Highlighting using Transfer Function Design

2.1 Motivation

Techniques for highlighting focus objects or areas for various volumetric applications facilitate the pre-attentive perception of the user for the F+C visualization. Meanwhile, due to the complexity of volumetric data, it is extremely time consuming for the user to select the region of interest effectively. Thus, transfer functions are fundamental to focus oriented direct rendering because their role is essentially to make the data of interest visible: by assigning optical properties (e.g., color and opacity) to voxels, highlight or further separate the focus and context regions.

High Dimensional Transfer Function Design. Transfer function design, as an important classification method, has been proposed to produce images that display, highlight and even select the region of interest in the volumetric datasets. Good transfer functions should be able to accurately reveal the important structures in the data without any unrelated context. Some significant experience has been accumulated on how to identify the accurate boundaries between different materials, such as 1D scalar values based TF and various 2D TFs with respect to gradient magnitude, directional first and second derivative [22], curvature [23] and statistical measures [24]. However, the specification of a transfer function to accurately identify different objects in a complex volumetric dataset is still a challenging task. Then a powerful nD parameters visualization and data analysis tool, parallel coordinates plots (PCP), is used as an alternative TF design style. This chapter introduces a novel parallel coordinates based multi-dimensional (nD) transfer function design method, termed as parallel coordinates based transfer function (PCbTF) design method.

Dimension Reduction based Transfer Function Design. On the other side, dimension reduction is an alternative solution for the high dimensional TF design. Takanashi et al. [93] have used independent component analysis (ICA) for nD parameter reduction. Rezk-Salama et al. [94] have created models from several training datasets by principle component analysis (PCA) to reveal the desired structures. Pinto and Freitas [95] have applied self-organizing maps (SOMs) and radial basis functions (RBFs) to simplify the design of nD TF to achieve the accurate classification. The local linear embedding (LLE) method [96], a good nonlinear high dimension reduction method, can be used to reduce the dimension, effectively simplifying the complicated polyline analysis. LLE maps its inputs into a single global coordinate system of lower dimensionality, and thus its optimizations do not involve local minima, which leads to the ability of learning the global structure of nonlinear manifolds (details in Section 2.3).

The pipeline of entire system is shown as Figure 4.2. First, for each voxel of input dataset, various high dimensional parameters are calculated. Next, parameters are selected according to the patterns of corresponding polylines drawn in PCP. For the high dimensional TF design, the user can choose to either interactively design special widgets on the coordinates directly or automatically project all the attribute parameters to the 2D space by the LLE technique as dimension reduction, and then assign colors and opacities to the classes calculated by a k-mean algorithm in the 2D space. A simple but effective user interface has also been developed to assist in the nD TF design. Through the framework, a carefully designed nD TF can emphasize details which are difficult to visualize by other approaches.

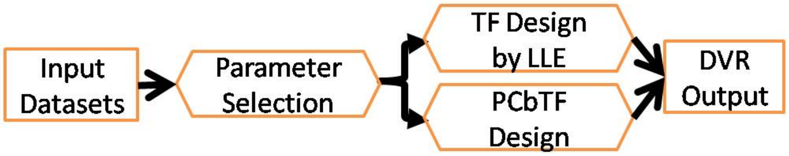


Figure 2.1: An overview of the transfer function design pipeline.

2.2 Parallel Coordinates based Transfer Function Design

A new design procedure is proposed for the presented PCbTF framework. After the data preparation and parameter extraction, the system applies a simulated annealing (SA) method [97] to find the best sorting order of all the parameters in the PCP with respect to energy minimization. Then, the correlated parameters are removed according to mathematically defined patterns of polylines. Next, several novel widgets are proposed to help the user to design PCbTF.

2.2.1 Data Preparation

For most volumetric datasets, simply removing the background voxels does not influence the information of the feature of interest, but will significantly decrease the computational time and operation complexity. Therefore, the region growing method is used to remove the background voxels around the major objects. Next, sixteen statistical attributes (angular second moment, contrast, correlation, variance, inverse difference moment, individual entropy, sum average, sum variance, sum entropy, skewness, kurtosis, correlation information measurements, intensity, gradient and second order derivative) are extracted and drawn as coordinates in the PCP (following the feature equations defined in [27] and [98]). The concept of an outlier [99] is applied to remove noise. Randomly distributed noises are simply removed by erasing the outlier polylines.

2.2.2 Visualization of Parameter Sorting and Selection

For the visual clustering in the PCP, various methods have been published such as blending [100] and scatter plot matrix [101]. The system imbeds the brush function to select, highlight and erase polylines using the XmdvTool library [102]. One of the most important motivations of PCP is to best reveal the relationship/correlation of neighboring coordinates. However, the PCP sorting problem, as NP-complete problem, has no efficient algorithm (with running time O(n!)). Peng et al. [103] have provided random swapping, nearest neighbor and greedy methods to solve this problem. However, random swapping suffers from serious repetition problem, while nearest neighbor and greedy algorithms are easily stuck at the local optimization. Thus, the SA is modified to find a fast optimization solution (with running time $O(n^p 2^n)$). First, new equations are defined as the clutter measurement to describe the internal energy $f(h)$ between the neighboring coordinates: $f(h) = \frac{1}{\sum_{i=1}^{N}(p_i - \frac{1}{N})^2}$, where $N = N_1 * N_2$, N_1 and N_2 are the pre-defined bin number of adjacent coordinates; p_i is the 2D joint histogram distribution probability, $p_i = \frac{l_i}{\sum_{i=1}^{N} l_i}$, while l_i is the total polyline number in the i_{th} bin.

This formula describes the basic idea: the more polyline aggregation in a

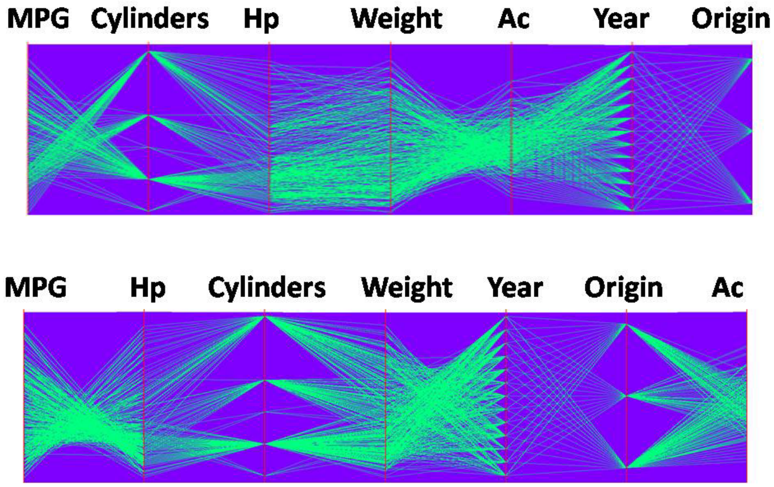
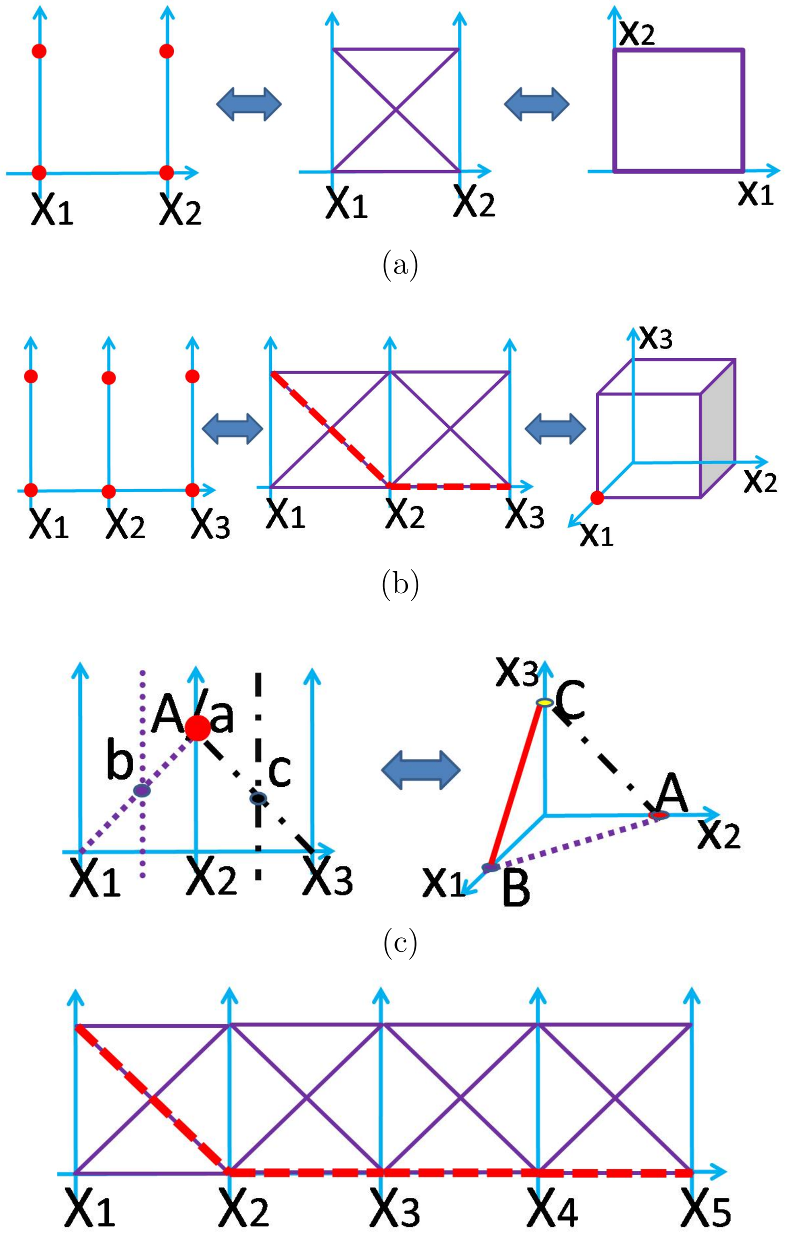


Figure 2.2: Sorting results of the car dataset (7 dimensions, 392 items). Up: Random sorting sequence. Bottom: The same optimal sorting result generated by brutal search (T=20s) and the presented method (T=1.6s) with different computational time.

single bin, the less internal energy. Algorithm 1 lists the details and conditions during the implementation. This method can quickly reach the global optimal solution using appropriate parameters, as shown in Figure 2.2. The best sorting order makes it easy to select important parameters: various correlated patterns can be identified and removed according to precise definitions by Inselberg [104].

2.2.3 Widgets Design

PCP preserves properties of a hypersurface by polyline patterns, which makes the TF design easy. Figure 2.3 shows the corresponding patterns between traditional orthogonal space (2D, 3D and nD) and the parallel coordinates. In addition to these basic patterns, various primitives are extended for TF design. Figures 2.4a-f show the design ideas in PCP and the corresponding modifications in the 2D or 3D space. All the designed patterns or primitives can be easily extended to nD by setting constraints between the neighboring coordinates: $x_2 = f_1(x_1), x_3 = f_2(x_2), ..., x_n = f_{n-1}(x_{n-1})$, where the functions are pre-defined by the user. Therefore, the adjustment of x_1 will automatically control the selection of other parameters. Through using the widgets, the complicated design of high dimensional TF, which is impossible to be drawn on the screen by traditional methods, is easily implemented in the PCP.



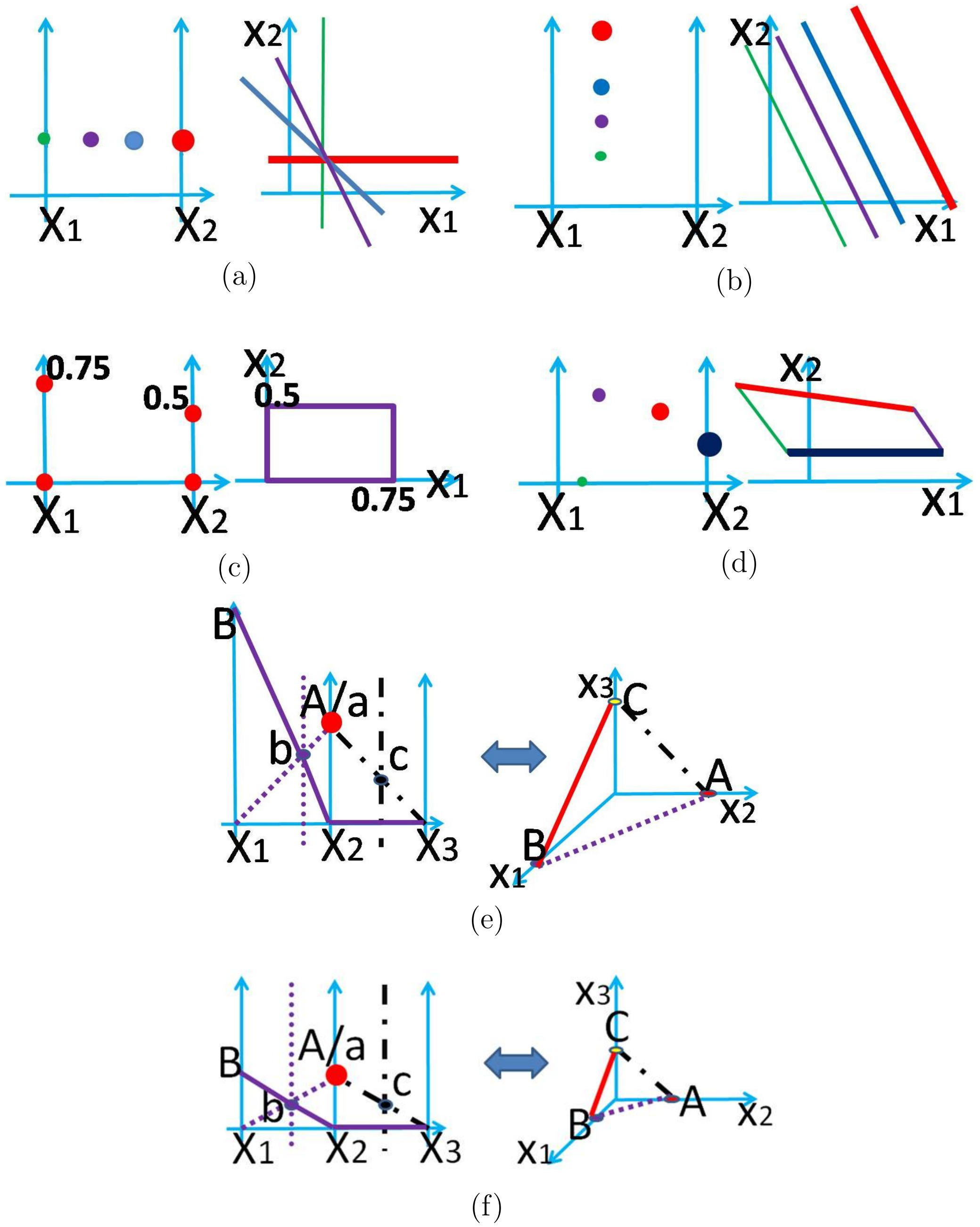
(a)

(b)

(c)

(d)

Figure 2.3: Matching patterns between PC and orthogonal space. (a) 2D rectangle or square. (b) 3D cube, the dashed polyline is corresponding to the corner point of cube, with position (1,0,0). (c) 3D plane, the points of intersection of one dashed-line type in the PC is matching with the same dashed-line type in the 3D space. (d) 5D manifold, which is impossible to be displayed on the screen by the orthogonal coordinates. The dashed polyline is corresponding to (1,0,0,0,0).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.4: PCbTF design widgets. For a 2D line: y=a*x+b, the adjustment of (a) the slope b, and (b) the rate a. For a 2D plane (Figure 2.3a), (c) 2D rectangle and (d) 2D any type polygon designed from changes of control points. For a 3D plane (Figure 2.3c), (e) update the vertical dashed line (point b as the point of intersection) in the PC, matching with the adjustment of point B in the 3D space, and (f) proportionately scale of the 3D plane (scale=0.5).

Algorithm 1 Simulated Annealing Algorithm

```
$T \leftarrow T_0$; $X \leftarrow X_0$ {Initial temperature, sequence}
$T_i \leftarrow T_0$; $X_i \leftarrow X_0$ {Initial the best solution $X_i$}
while $T_i > T_{min}$ do
  $i : (w_1, \ldots, w_k, w_{k+1}, \ldots, w_m, w_{m+1}, \ldots) \leftarrow j : (w_1, \ldots, w_m, w_{k+1}, \ldots, w_k, w_{m+1}, \ldots)$ {For the random values k and m, setting $k < m$}
  $df \leftarrow (f(X_j) - f(X_i))$ {Calculate energy difference}
  if $df < 0$ then
    $X_i \leftarrow X_j$
  else if $df > 0$ and $\exp^{\frac{-df}{T_i}} > random()$ then
    $X_i \leftarrow X_j$
  end if
  $T_i \leftarrow T_i - T_d$ {Decrease temperature}
end while
Return $X_i$
```

2.3 Dimension Reduction based Transfer Function Design

Although the PCP can directly assist the design of high dimensional TF, the major limitations are the facts that large datasets or parameter axes cause difficulty in the interpretation for the accurate classification, and relationships are only preserved between adjacent coordinates. Resorting the coordinates is an extremely time-consuming task especially for very high dimensions. Therefore, dimension reduction is motivated for the design of nD TF. We apply local linear embedding method (LLE) [96], an unsupervised learning algorithm that computes the low-dimensional, neighborhood-preserving embedding of high-dimensional inputs. LLE has several predominances: (1) It eliminates the need to estimate pairwise distances between the widely separated data points; (2) It maintains the global nonlinear structure from locally linear fits. By comparison with PCA [105] and metric MDS [106], LLE is especially good at identifying the underlying complicated manifold structure. Therefore, LLE is implemented as a dimension reduction based TF design method. The algorithm is briefly described as follows: Suppose the data consists of N real-valued vectors $\overrightarrow{X_i}$, the LLE algorithm, is based on simple geometric intuitions with two cost functions:

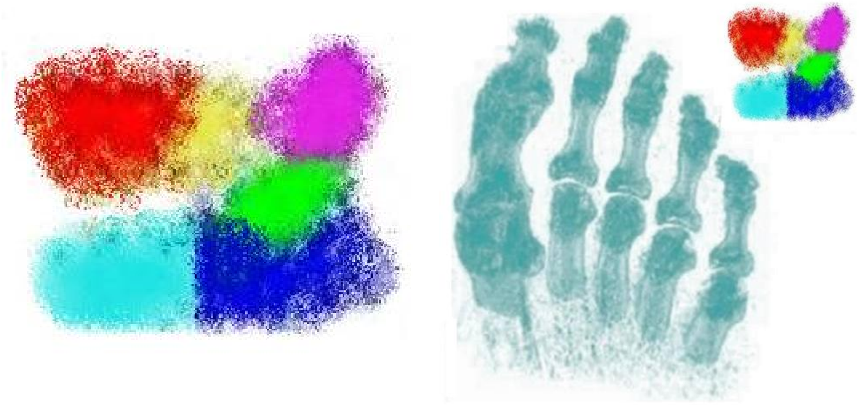
$$\varepsilon(W) = \sum_i |\overrightarrow{X_i} - \sum_j W_{ij} \overrightarrow{X_i}|^2 \tag{2.1}$$

and

$$\phi(Y) = \sum_i |\overrightarrow{Y_i} - \sum_j W_{ij} \overrightarrow{Y_i}|^2 \tag{2.2}$$

First, compute the neighbor of each data point, $\overrightarrow{X_i}$. Then compute the weights W_{ij} from the neighbor points of $\overrightarrow{X_i}$ by minimizing the cost in Equation 2.1. Last, compute the best reconstruction by weights W_{ij}, through minimizing Equation 2.2 using its lowest nonzero eigenvectors.

The user can control two parameters of LLE, the number of neighbors and the dimension of the embedding space (default is 2D). After applying LLE, the embedded space tends to be very abstract and its meaning unclear. Fortunately, the high dimensional relationships among voxels can be perfectly preserved and can be easily classified by k-mean algorithm in 2D space, as shown in Figure 2.5a. Thus, the user just needs to assign color and opacity for each class to obtain the final rendering result, as shown in Figure 2.5b. Due to the loss and distortion of information usually caused by dimension reduction, LLE cannot provide quantitative information as accurate as PCbTF does. However, this approach is well suitable for revealing qualitative aspects such as shape of structures and clear dissimilarity between regions.



(a) (b)

Figure 2.5: TF design scheme using LLE. (a) The k-mean classes in the embedded 2D space for the CT foot dataset with five parameters: intensity, gradient, variance, entropy and angular second moment. (b) The rendering result by assigning high opacity to the classes of bone structure shown as dark and light blue regions.

2.4 Implementation

The parameter extraction and dimension reduction is used as the off-line processes, but to maximize system interactivity, the system accelerates the volume rendering and TF specification using graphics hardware. Given a parameter representation of each voxel, the task of designing PCbTF becomes specifying a mapping from the nD vector to a color and opacity value. The classification of each voxel forms a 3D texture stored in the frame buffer. New texture will be recomputed and redrawn on the screen by the GPU whenever TF changes. We calculate another 3D RGB texture to store colors and opacities picked by the user. The interface used to display polylines and design TFs is implemented using OpenGL and FLTK libraries. Rendering uses Cg on a desktop: Intel Xeon CPU 3.60GHz, 3GB memory and Nvidia GeForce GTX 285 graphic card. This approach is fast enough for the purposes of real time interaction to modify the design of TF and update the volume rendering results.

Timing and Performance. The main time-consuming part of the pipeline is the parameter extraction, which is related to dataset size, window size and selected parameters. Table 2.1 shows some statistics for all the test datasets. The table shows that the volume size and window size are major timing factors - time increases as the volume size grows while the window size decreases. Meanwhile, comparing with PCbTF, the design of LLE basd TF provides an easy operation and classification method in the 2D space, although some extra mapping time is needed. From the timing point view, the bottleneck is the interactive design of PCbTF. Therefore, total design time of LLE basd TF is small, which is a good choice when no specified objects are wanted by the user.

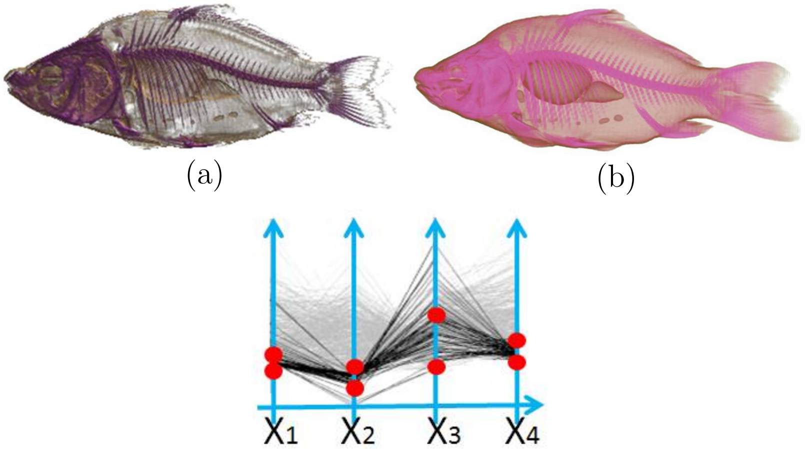
Table 2.1: Statistics of various test datasets (WS: window size, T.PE: parameter extraction time and T.LLE: local linear embedding time in seconds).

Model	Datasize	WS	T.PE	T.LLE
CT Carp	256^2x512	9^3	43.54	–
CT Bladder	256^2x24	7^2x3	14.37	–
MRI Prostate	128^2x16	5^2x3	8.76	–
CT Foot	256^3	7^3	75.11	60.95
CT Engine	256^3	9^3	22.35	17.30
CT Bonsai	256^3	7^3	76.90	64.72

2.5 Experimental Results and Discussion

Various CT and MRI datasets are used to demonstrate the ability of the presented framework to deal with challenging tasks.

Applications of Parallel Coordinates based Transfer Function Design Figure 2.6 shows the rendering result of the CT carp dataset generated by the presented technique. Through the interface, the user can design a PCbTF as shown in Figure 2.6c to highlight the internal region of interest. Figure 2.6b shows the obviously visual enhancement of the swimming bladder by comparison with the result generated by 2D TF (shown in Figure 2.6a).



(a) (b)

(c)

Figure 2.6: Volume rendering results of the CT carp dataset generated by (a) 2D TF and (b) PCbTF. (c) The design of PCbTF with four parameters ($X_1, \ldots, X_4$): intensity, gradient, sum variance and sum average.

Figure 2.7 shows the PCbTF results by comparison with other TFs for the CT bladder dataset. Figures 2.7a and b show that the 1D/2D TF will fail to distinguish accurate features from bladder and other tissues. We first apply the nD TF without any spatial information and receive a clear segment of bone structure, as shown in Figure 2.7c. However, for the bladder, as shown in Figure 2.7d, many "noise" points surround the bladder because these voxels are close to the class of bladder in the parameter space but far away in the spatial space, as shown in Figures 2.7e and f. We further refine the classification using the prior position information. Figure 2.7g shows the final rendering result using PCbTF designed as depicted in Figure 2.7h. The bladder is clearly separated from the surrounding tissues.

Another case study is the classification of MRI prostate dataset, an extremely challenging task for the common TF, as shown in Figure 2.8b. From

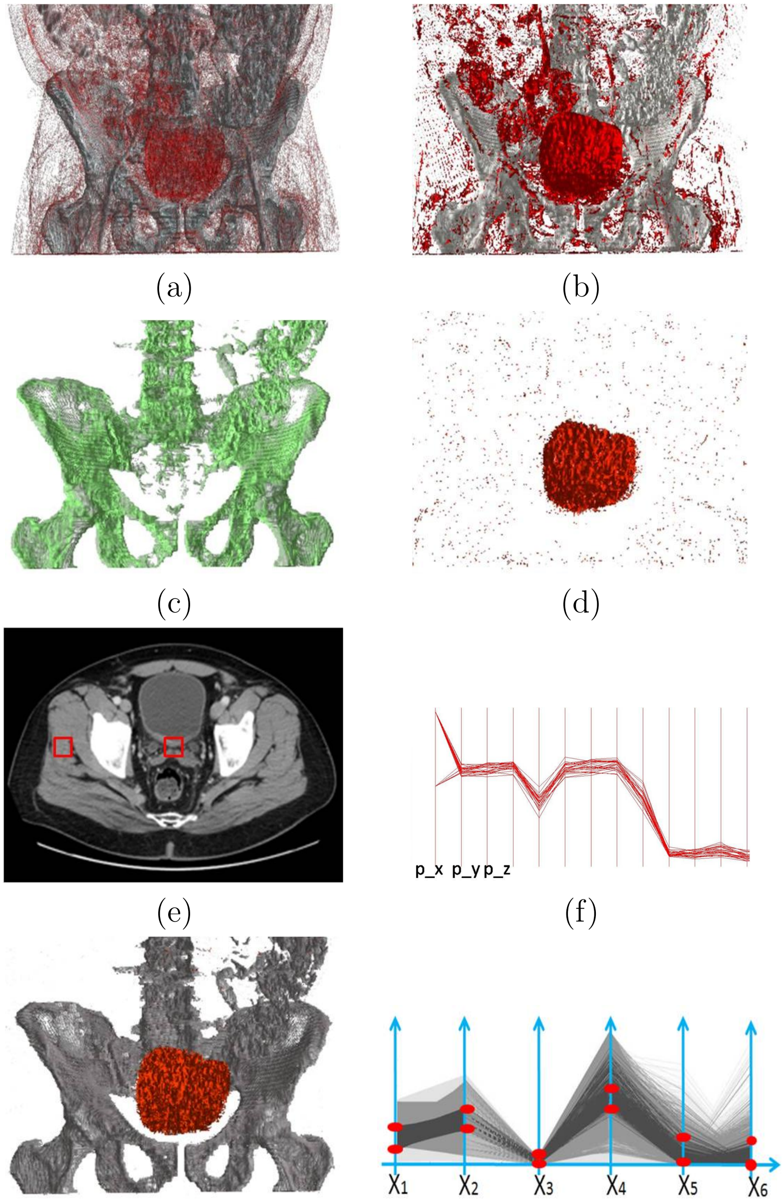
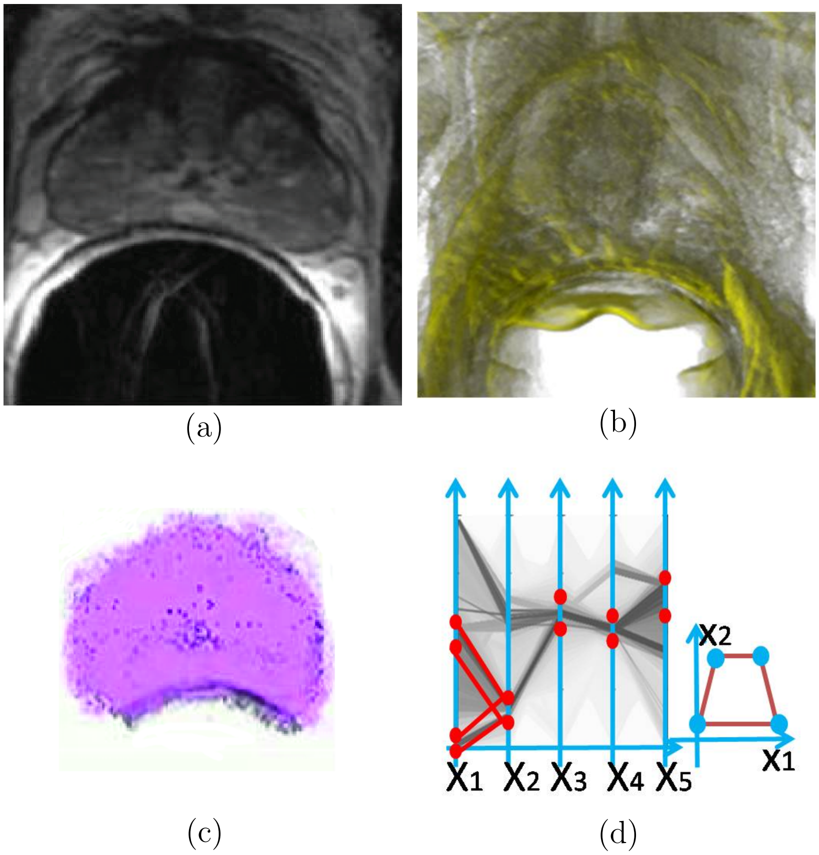

(a) (b) (c) (d) (e) (f) (g) (h)

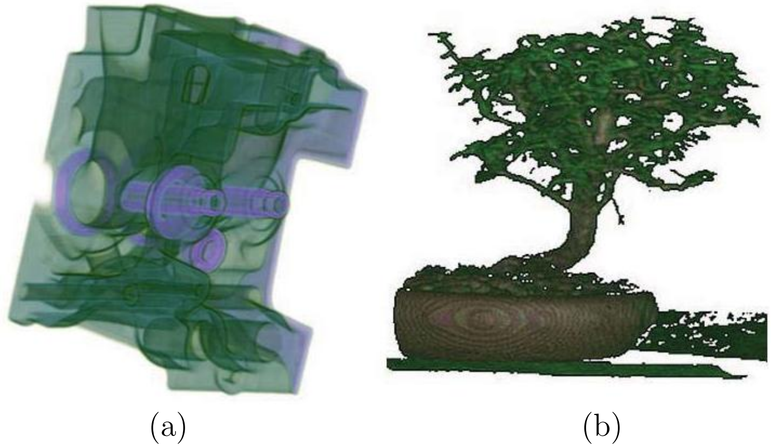
Figure 2.7: Volume rendering results of the CT bladder dataset generated by (a) 1D TF and (b) 2D TF. Volume segment of (c) bone structure and (d) bladder generated by PCbTF in parameter space. Two similar parameter values (in boxes) in (e) a CT bladder slice but with different spatial values (pos_x) in the PCP (f). (g) the result generated by PCbTF combined with spatial information. (h) The design of PCbTF with six parameters ($X_1, \ldots, X_6$): position (x,y,z), intensity, gradient and entropy.


(a) (b)

(c) (d)

Figure 2.8: Volume rendering results of the MRI prostate dataset generated by various TFs. (a) MRI axial slice of the prostate. (b) The result generated by 1D TF. (c) The result generated by PCbTF combined with anatomical knowledge and spatial information. (d) The design of PCbTF with five parameters ($X_1, \dots, X_5$): intensity, second order derivative, kurtosis, contrast and variance. Points are the boundary of the TF for each coordinate and small figure (right) shows the matching 2D pattern between X_1 and X_2.

the image, a 1D TF cannot truly separate the prostate out from its surrounding tissues due to the very similar parameter characteristics. Luckily, the medical information of the prostate such as location, size and shape can be easily found. We use the anatomical and spatial information to improve the result. First, detect the colon, an organ with obvious features (large dark region as shown in Figure 2.8a). According to the anatomical knowledge, the colon is the nearest organ to the prostate along the posterior end, and thus the system can estimate the possible position of prostate. The position information becomes an important parameter combined with the other selected parameters (shown in Figure 2.8d) for the design of PCbTF to identify the prostate. Figure 2.8c shows the final result, with high opacity for the prostate and zero opacity for the surrounding tissues and organs. The above cases support the ability of PCbTF to clearly and accurately distinguish objects from surrounding tissues especially for the challenging datasets that tractional TFs fail.


(a)

(b)

Figure 2.9: Volume rendering results using LLE as dimension deduction for (a) the CT engine and (b) the CT bonsai tree datasets.

Applications of Dimension Reduction based Transfer Function Design For some datasets, more parameters are picked as independent parameters. For example, the CT Engine dataset has five parameters: intensity, gradient, entropy, correlation information and sum average, while the CT bonsai tree dataset contains six parameters: intensity, gradient, difference moment, skewness, variance and sum entropy. In order to simplify the design process, the system projects all the parameters to the embedded 2D space by LLE, then implement k-mean classification and design a 2D TF. Although there is information loss during the dimension reduction, Figures 2.9a and b still show good classification results: the user can easily recognize the piston rings and piston rods for the engine dataset, and distinguish the soil, trunk and leaves for the bonsai tree dataset by assigning them different colors.

Chapter 3

Conformal Mapping based Focus+Context Visualization

3.1 Motivation

Many optical lenses have been available for centuries and humans have become very familiar with them. Computers can easily simulate lenses' effects, while at the same time providing great opportunity to overcome limitations of lenses. The presented framework addresses a specific limitation that optical lenses as well as their many digital counterparts have: their local distortion. Although current lenses are "magic" in the sense what they can reveal, they are less so when it comes to overcoming serious distortion effects or artifacts. The specific need is to control local distortion, to preserve local detail undistorted and thus enable the user to reliably read and decode accurate information. In addition, a good F+C method must support continuity when transitioning from the magnified to the minified areas. Only then can the user perform effective visual searches at these multiple levels of scales. Thus, the framework also considers the smooth transaction between focus and context regions. The presented lens uses the concept of *conformal mapping* as a novel F+C technique to capture the region of interest (ROI) into a single view while providing a smooth transition between the focus and context regions. Instead of only using lenses with regular circle or square shape, arbitrary shape models are embedded in the system to magnify ROIs with different shapes. The *conformal magnifier* minimizes global angle distortions and preserves local angular relationships which, in turn, preserve important shapes and features of objects during the deformation.

The local angle of objects is an important structure-based factor/descriptor for the visual cognition [107]. This property plays a crucial role during the

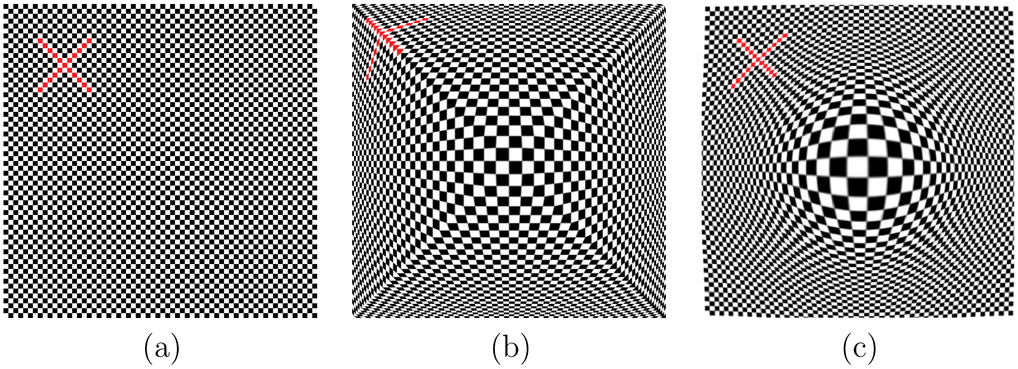


Figure 3.1: Basic merits of the conformal mapping compared with the traditional lens for symbols located in the transition region between the focus and context regions. (a) A checker board image with a red X-like symbol. Magnification results using (b) the fisheye lens and (c) the conformal magnifier.

magnification, especially in the transition region between the focus and context areas, as shown in Figure 3.1. After the magnification, the test symbol is hard to be identified and recognized visually by the traditional fisheye lens because of its local angle and area distortion in the transition area, while the conformal magnifier well-preserves the local orthogonal features of the symbol. The presented method is advantageous over previous approaches, as the application of the conformal magnifier well-preserves the local shape feature, keeps the global structure, and builds a smooth transition field, leading to reliable zooming results. The main contributions of the presented system are:

- Arbitrary shape models, used as magnifiers to satisfy different application needs;
- A conformal magnifier with local shape preservation and smooth transition;
- The general applicability of the framework for diverse graphics and visualization areas.

To the best of the knowledge, no previous work has used conformal mapping theory as an F+C technique for visualization. In this respect, the conformal magnifier, representing an ideal continuous multi-focus F+C technique with several unique features, is a novel non-linear magnification method. The well-defined conformal function is numerically well behaved: in theory, conformal mapping does not have any local angle distortion. Namely, everywhere, there is neither angle nor shape distortion. In the discrete setting of triangular meshing structure, the approximation of conformal mapping globally minimizes the angle distortion. Therefore, conformal mapping is able to well preserve both local and global shapes with minimal distortion, robustly supporting the solution of several challenging cases.

3.2 Conformal Mapping Theory

This section the merits of conformal mapping and briefly introduce the basic theory background of conformal geometry, necessary for the discussion in this work. For more details, read more detials in [108] for Riemann surface theory and [78] for differential geometry.

3.2.1 Conformal Structure

Conformal structure and its properties are important mathematical foundations used to support the solid conformal mapping theory. Thus, this section briefly introduces the necessary background knowledge of conformal geometry.

Suppose S is a surface embedded in $\mathbb{R}^3$, therefore S has the induced Euclidean metric $\mathbf{g}$. Let $U_\alpha \subset S$ be an open set on S, with local parameterization $\phi_\alpha : U_\alpha \to \mathbb{C}$, such that the metric has local representation

$$\mathbf{g} = e^{2\lambda(p)} dz d\bar{z}, \quad p \in U_\alpha$$

where $\lambda : U_\alpha \to \mathbb{R}$ is called a *conformal factor* function, $z \in \mathbb{C}$ is parameter coordinates, d denotes the exterior derivative. Then (U_α, ϕ_α) is called an *isothermal coordinate chart*. The whole surface can be covered by a collection of isothermal coordinate charts. All the isothermal coordinate charts form a *conformal structure* of the surface. The surface with a conformal structure is a *Riemann surface* [108]. Suppose S_1 and S_2 are two Riemannian surfaces. Suppose (U_α, ϕ_α) is a local chart of S_1, (V_β, ψ_β) is a local chart of S_2. $\phi : S_1 \to S_2$ is a *conformal map* if and only if

$$f = \psi_\beta \circ \phi \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha) \to \psi_\beta(V_\beta)$$

is bi-holomorphic, i.e., it satisfies the Cauchy-Riemann equation $\frac{\partial f}{\partial \bar{z}} = 0$. For simplicity, the same ϕ is used to denote its local representation. Then a conformal map ϕ satisfies $\frac{\partial \phi}{\partial \bar{z}} = 0$.

3.2.2 Conformal Mapping by Surface Ricci Flow

Let S be a surface embedded in $\mathbb{R}^3$ with the induced Euclidean metric $\mathbf{g}$. Name that another Riemannian metric $\bar{\mathbf{g}}$ is *conformal* to $\mathbf{g}$, if there is a scalar function $u : S \to \mathbb{R}$, such that $\bar{\mathbf{g}} = e^{2u}\mathbf{g}$. The Gaussian curvature induced by $\bar{\mathbf{g}}$ is

$$\bar{K} = e^{-2u}(-\Delta_{\mathbf{g}} u + K),$$

where $\Delta_{\mathbf{g}} = e^{-2\lambda}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$ is the Laplacian-Beltrami operator under the original metric $\mathbf{g}$, K the original Gaussian curvature under $\mathbf{g}$, $\bar{K}$ the induced Gaussian curvature under $\bar{\mathbf{g}}$. The above equation is called the *Yamabe equation*. By solving the Yamabe equation, one can design a conformal metric $e^{2u}\mathbf{g}$ by a prescribed target curvature $\bar{K}$.

Ricci flow can be used to solve Yamabe equation. It is a powerful tool which has been used for proving the Poincaré conjecture. Ricci flow behaves like a heat diffusion process in the following form:

$$\frac{dg_{ij}(t)}{dt} = 2(\bar{K} - K(t))g_{ij}(t) \tag{3.1}$$

where t is the time parameter. If $\bar{K} \equiv 0$, Ricci flow deforms the Riemannian metric $\mathbf{g}$ to the *uniformization metric* $\bar{\mathbf{g}}$ by evolving the Gaussian curvature K, such that the Gaussian curvature becomes constant everywhere, according to the surface uniformization theorem in [108]. The convergency of Ricci flow to the uniformization metric has been proved in [109] and [110]. Here the system deals with quadrilateral surfaces with Euclidean background geometry, and maps them to a planar rectangle parametric domain. Thus, the algorithm chooses $\bar{K} = 0$ at interior points and $\bar{K} = \pi/2$ at four boundary corners in such cases.

3.2.3 Discrete Surface Ricci Flow

This part describes several major concepts for computing Ricci flow on discrete surfaces. In practice, most surfaces are approximated by simplicial complexes, namely triangular meshes. Suppose M is a triangle mesh, V, E, F are vertex, edge and face set, respectively. Use v_i to denote the i-th vertex; $[v_i, v_j]$ the edge from v_i to v_j; $[v_i, v_j, v_k]$ the face, where the vertices are sorted counter-clock-wisely.

Discrete Metric and Curvature. A *discrete metric* on a mesh M is a function $l : E \to \mathbb{R}^+$, such that on each face $[v_i, v_j, v_k]$, the triangle inequality holds, $l_{jk} + l_{ki} > l_{ij}$. If all faces of M are Euclidean, then the mesh is with Euclidean *background geometry*, denoted as $\mathbb{E}^2$. The discrete metric represents a configuration of edge lengths and determines the corner angles on each face by cosine law,

$$\theta_i^{jk} = \cos^{-1} \frac{l_{ki}^2 + l_{ij}^2 - l_{jk}^2}{2l_{ki}l_{ij}},$$

where θ_i^{jk} is the angle at v_i opposite to edge $[v_j, v_k]$ in the face. The *discrete Gaussian curvature* of v_i is defined as an angle deficit at v_i, considering all the corner angles surrounding a vertex v_i,

$$K_i = \begin{cases} 2\pi - \sum_{jk} \theta_i^{jk} & v_i \notin \partial M \\ \pi - \sum_{jk} \theta_i^{jk} & v_i \in \partial M \end{cases} .$$

Circle Packing Metric. The discrete Ricci flow can be carried out through the circle packing metric, which is a discretization of conformality and was introduced by Thurston [111]. Each vertex v_i is associated with a circle with radius r_i. Two circles at the end vertices of an edge $[v_i, v_j]$ intersect at an angle θ_{ij}, then the edge length l_{ij} is given by

$$l_{ij}^2 = r_i^2 + r_j^2 + 2r_i r_j \cos\theta_{ij}.$$

A conformal deformation maps infinitesimal circles to infinitesimal circles and preserves the intersection angles among the infinitesimal circles. As shown in Figure 3.2a, the circle radius centered at each vertex deforms while not changing the intersection angles among circles $\theta'_{ij} = \theta_{ij}, \theta'_{ik} = \theta_{ik}$. The circle packing metric can be defined as $\mathbf{u} = \{u_i\}$, where $u_i = \log r_i$, r_i is the circle radius of v_i. The variation of the circle packing metric under Ricci flow generates the desired metric.

Discrete Conformal Metric Deformation. The discrete Ricci flow method is applied to conformally map the surfaces onto planar domains $\phi : M \to D$. In all configurations, the discrete Ricci flow is defined as follows:

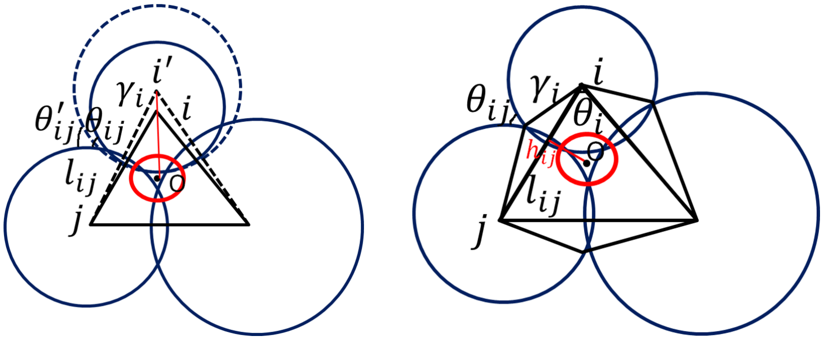
$$\frac{du_i(t)}{dt} = \bar{K}_i - K_i, \tag{3.2}$$

where $\bar{K}_i$ is the user prescribed target curvature and K_i is the curvature induced by the current metric. The discrete Ricci flow has exactly the same form as the smooth Ricci flow, which conformally deforms the discrete metric according to the Gaussian curvature.

The discrete Ricci flow can be formulated in the variational setting, namely, it is a negative gradient flow of a special energy form, called *Ricci energy*, which is given by

$$f(\mathbf{u}) = \int_{\mathbf{u_0}}^{\mathbf{u}} \sum_{i=1}^{n} (\bar{K}_i - K_i) du_i, \tag{3.3}$$

where $\mathbf{u_0}$ is an arbitrary initial metric. Computing the desired metric with user-defined curvature $\{\bar{K}_i\}$ is equivalent to minimizing the discrete Ricci energy.



(a) (b)

Figure 3.2: Geometric interpretation of discrete conformal metric deformation. (a) Conformal circle packing metric deformation and (b) the radial circle (in red) of a triangle.

The Hessian matrix for discrete Ricci energy is positive definite for the Euclidean case (with normalization constraint $\sum_i u_i = 0$). Therefore, the energy is convex and can be optimized using Newton's method. The Hessian matrix is computed on the circle packing metric [111]. As shown in Figure 3.2b, the *radial circle* of a triangle is unique and perpendicular to each vertex circle. For all configurations with Euclidean metric, suppose the distance from the radial circle center to edge $[v_i, v_j]$ is d_{ij}, then $\frac{\partial \theta_i}{\partial u_j} = \frac{d_{ij}}{l_{ij}}$, furthermore, $\frac{\partial \theta_j}{\partial u_i} = \frac{\partial \theta_i}{\partial u_j}$, $\frac{\partial \theta_i}{\partial u_i} = -\frac{\partial \theta_i}{\partial u_j} - \frac{\partial \theta_i}{\partial u_k}$. Define the edge weight w_{ij} for edge $[v_i, v_j]$ which is adjacent to triangles $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$ as

$$w_{ij} = \frac{d_{ij}^k + d_{ij}^l}{l_{ij}}.$$

The Hessian matrix $\mathbf{H} = (h_{ij})$ is given by the discrete Laplace form,

$$h_{ij} = \left\{ \begin{array}{cc} 0 & [v_i, v_j] \notin E \\ -w_{ij} & i \neq j \\ \sum_k w_{ik} & i = j \end{array} \right. .$$

According to the Gauss-Bonnet theory [78], the total curvature must be $2\pi\chi(M)$, where χ is the Euler characteristic number of M. In this application, M is a topological quadrilateral, so $\chi(M) = 1$. Set the target curvature of the four boundary corners to be $\pi/2$, other boundary vertices and interior vertices to be 0. Then the toplogical quadralteral is mapped to a rectangle. The convergency of discrete Ricci flow has been proved by [112]. Detaials about discrete analogue for general Ricci flow can be found in [113] and [114].

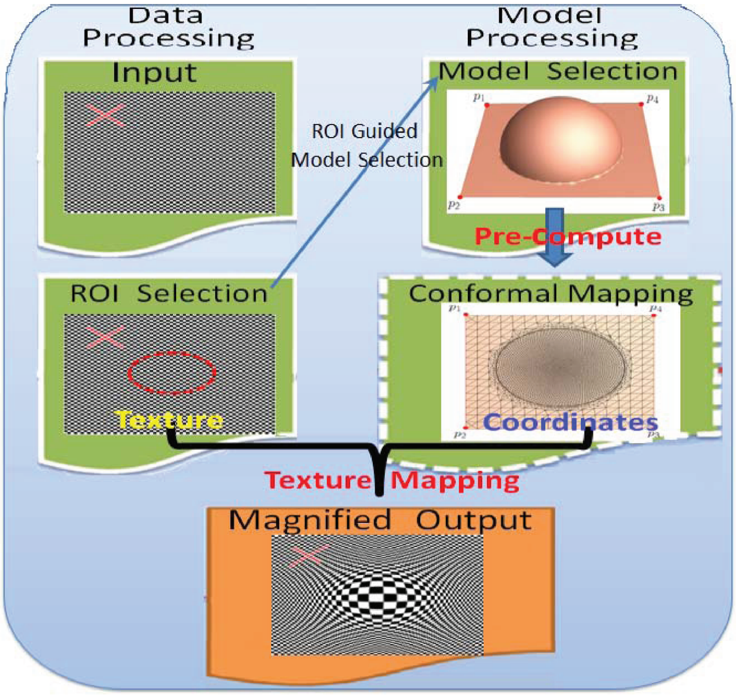


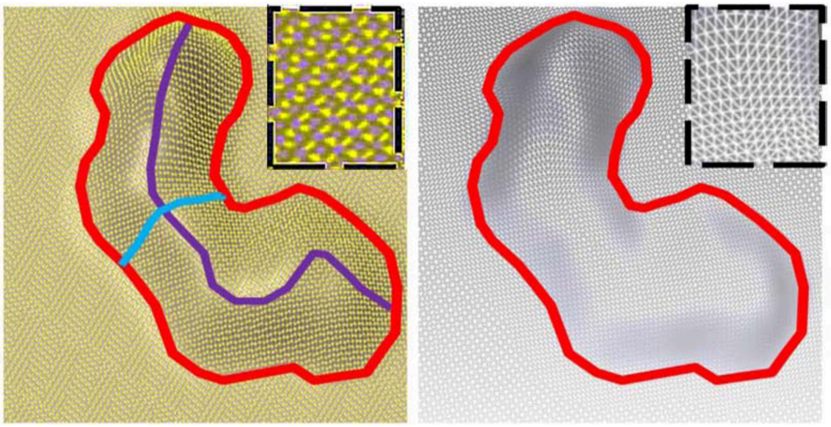
Figure 3.3: A schematic diagram of the conformal magnifier pipeline using a simple 2D checker board image as input.

3.3 Conformal Magnifier

The conformal magnifier is built upon the conformal mapping theory. Figure 3.3 shows the pipeline of the framework. There are two pre-computation steps: magnifier mesh model design and conformal mapping. An arbitrary mesh model as the magnifier can either be automatically generated based on mathematical definitions or be manually drawn through the user interface. Then, the system pre-calculates the conformal map of each magnifier model (parameterization of each vertex of mesh model). For any input, including both 2D map and 3D volumetric datasets, with the user defined ROI and magnifier models, the system can automatically display magnification results in real time using texture mapping or volume rendering. This section first introduces the magnifier mesh model design, and then briefly describes the theory and algorithm of the conformal mapping.

3.3.1 Magnifier Model Design

Two steps are required to efficiently design a surface model: point cloud construction and mesh generation from point cloud, as shown in Figure 3.4.

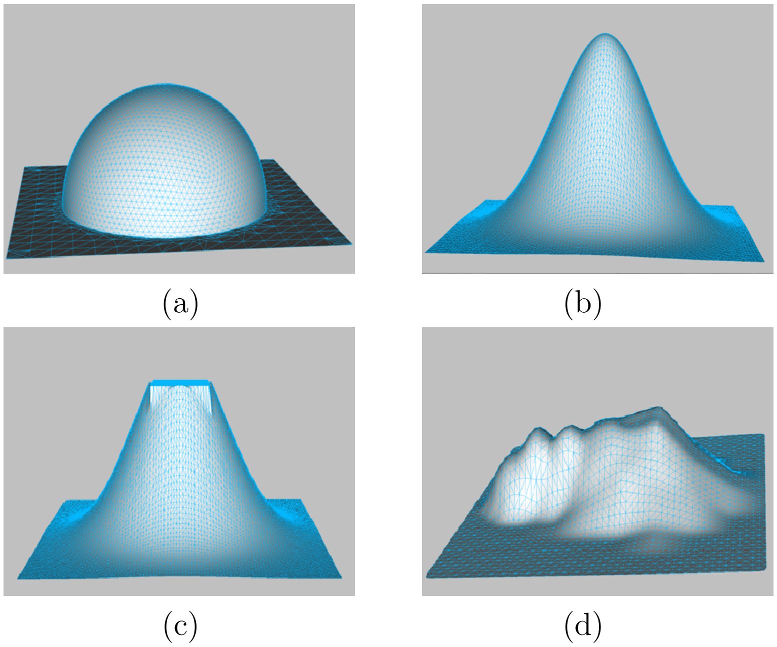


(a) (b)

Figure 3.4: Two steps in designing a magnifier model with an arbitrary shape: (a) Point cloud generation with respect to the user defined (red) boundary and (purple) centerline. The cyan line is a quadratic curve for generating 3D interpolation points between a boundary sample point and its nearest centerline sample point. (b) Mesh model generation using the point cloud generated in (a). The black dashed boxes show the magnified details.

Point Cloud Construction. The specified plane point cloud is constructed by the boundary curve and centerline drawn by the user to highlight the ROI, as shown in Figure 3.4a. The user can adjust the magnification ratio by adjusting the height of the centerline along z-direction position (in R^3 with P_{xy} as a 2D plane). Then, the system automatically discretizes both boundary curve and centerline based on the sampling rate. For each boundary sample point, the L2-nearest centerline sample point is found to form a pair. For each pair, a pre-defined quadratic function (e.g., Gaussian function) is set with the current pair as start and end points, and then the interpolated positions of new cloud points are calculated and generated based on it, which results in a model (for the entire plane) with a continuous and smooth transition. The sampling rate can be interactively set by the user. In general, a high sampling rate generates a large number of points and fits the transition area smoothly, but takes long to compute (e.g., 14.36 sec for a test point cloud with 20,152 points).

Mesh Generation from Point Cloud. The system generates smooth surface meshes from the point cloud based on Delaunay refinement [115], as shown in Figure 3.4b, and further refines the mesh by inserting additional vertices if necessary. The manifold extraction is also implemented to have a regular smooth surface using the ball pivoting method [116]. In general,



(a) (b)

(c) (d)

Figure 3.5: Various specified mesh models. Regular shapes using (a) a hemisphere and (b) a Gaussian function. Arbitrary shapes using (c) a square plane with a smooth Gaussian transition, and (d) a random shape. The height of each model is non-linearly proportional to its magnification ratio.

more points in high curvature features produce a better fitting surface but requires longer computing time. Meanwhile, the computation speed of the mesh generation is also affected by the shape and topology of the point cloud (e.g., 24.77 sec for the same test point cloud with 20,152 points).

With this design method, the user can interactively define various regular or arbitrary-specified magnifier models (Figure 3.5) to magnify the ROI. Supported by conformal mapping theory, the *truly* arbitrary magnifier model design is the first major advantage of the framework.

3.3.2 Conformal Mapping Algorithm

The discrete Ricci energy can be optimized using Newton's method to achieve the unique global optimal metric with the prescribed curvature. The implementation detail is listed in Algorithm 2.

The system designs the magnifier model as a 2.5-dimensional mesh with coordinates (x, y, z), where the model can be projected to the 2D plane with coordinates (x, y), z describes the height from the plane. According to the conformal mapping theory, the local magnification ratio of the conformal mag-

Algorithm 2 Newton's Method of Discrete Ricci Flow

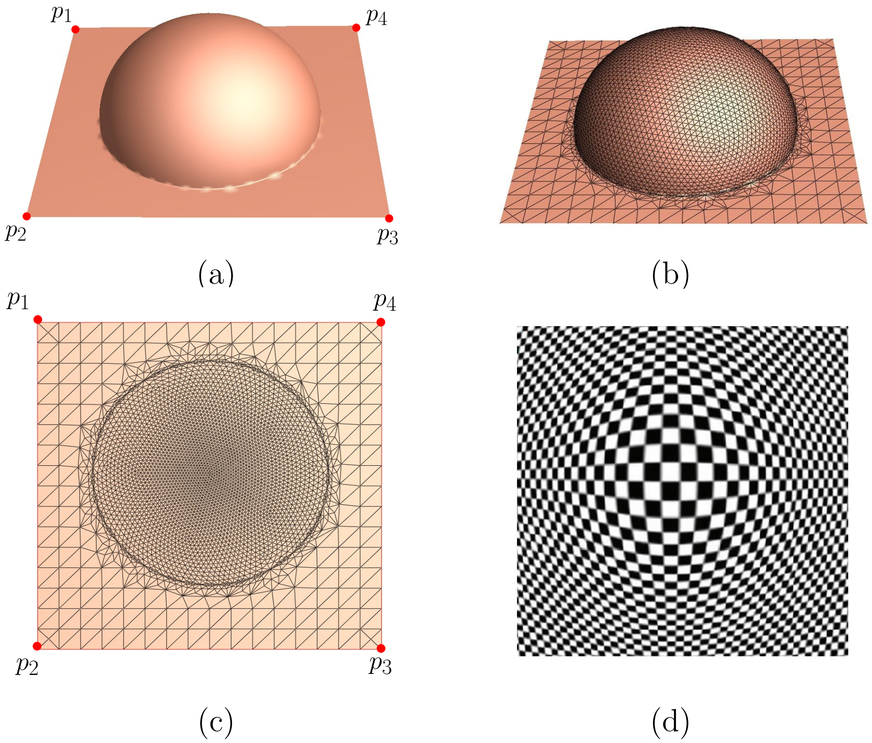
```
*Input:* a 3D mesh $M = (V, E, F)$, target curvature $\bar{\mathbf{K}} = \{\bar{K}_i\}$, curvature error threshold $\varepsilon$
*Output:* corresponding 2D parameterization positions $u$ of mesh vertices, used as coordinates for texture mapping or volume rendering
Initial the parameterization position $u_0$ and curvature $\mathbf{K} = \{K_i\}$
while $\max|K_i - \overline{K}_i| > \varepsilon$ do
  for all edges $e = [v_i, v_j] \in E$ do
    $l_{ij} \leftarrow \sqrt{\gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\theta_{ij}}$ {Compute the edge length by radii $\gamma_i, \gamma_j$ centered at $v_i, v_j$}
  end for
  for all corner angles $\theta_i \in [v_i, v_j, v_k]$ do
    $\theta_i \leftarrow \cos^{-1} \frac{l_{ij}^2 + l_{ki}^2 - l_{jk}^2}{2l_{ij}l_{ik}}$ {Compute the corner angle}
  end for
  for all edges $e = [v_i, v_j] \in E$ do
    $w_{ij} \leftarrow \frac{d_{ij}^k + d_{ij}^l}{l_{ij}}$ {Compute the edge weight}
  end for
  for all vertices $v_i \in V$ do
    $h_{ij} \leftarrow -w_{ij}, [v_i, v_j] \in E$;
    $h_{ii} \leftarrow \sum_k w_{ik}$ {Compute the Hessian matrix $\mathbf{H}$}
  end for
  $d\mathbf{u} \leftarrow \mathbf{H}^{-1}(\mathbf{K} - \bar{\mathbf{K}})$ {Minimize the discrete Ricci energy}
  for all vertices $v_i \in V$ do
    $u_i \leftarrow u_i + du_i$ {Update the circle packing metric for the calculation of each corresponding parameter position}
  end for
  for all vertices $v_i \in V$ do
    $K_i \leftarrow 2\pi - \sum_{jk} \theta_i^{jk}, v_i \notin \partial V$; or
    $K_i \leftarrow \pi - \sum_{jk} \theta_i^{jk}, v_i \in \partial V$ {Update the Gaussian curvature}
  end for
end while
```

Table 3.1: Statistics of test magnifier models.

Model	No. of Vertices	No. of Faces	Time (s)
Sphere	3446	6826	3.24
Square	3125	6018	3.08
Elongated Model	3691	7063	3.76
Low Gaussian	2192	4830	2.19
High Gaussian	4246	8926	3.97
Curved Model	2639	5328	2.65

nifier is non-linearly proportional to the height z at the corresponding vertex $v = (x, y, z)$. Therefore, the global/local magnification ratio is controllable by adjusting the heights of all/some of vertices. The entire computational process for a magnifier model which is a topological quadrilateral is illustrated in Figure 3.6. The original magnifier mesh model is shown in Figure 3.6a, with four corner vertices noted as p_1, p_2, p_3, p_4. For solving Equation 3.1, first set the target curvature for each vertex in the triangle mesh (Figure 3.6b). Then the four corner vertices are assigned $\pi/2$, while other vertices are 0. In Figure 3.6c, the Ricci flow conformally maps the mesh model onto a planar rectangle, with the corner vertices mapped to the rectangle corners. Thus, the system generates the conformal map, with the corresponding 2D parameterization position of each vertex in the selected mesh model, used as coordinates for texture mapping or volume rendering. Giving rise to the conformally bijective mapping from checker-board texture to this model based conformal map in the parameter domain, the ROI is magnified with a smooth transition, as shown in Figure 3.6d.

Computing conformal mapping by Ricci flow equals to solving a non-linear geometric PDE, which is stable and robust to the resolution of models. With an accelerated CPU/GPU solution, the conformal mapping of magnifier models is on second level for most cases (note that the conformal mapping for the designed modelscan is computed *offline*, separated from the F+C magnifying procedure in practice). The *real-time* performance of F+C conformal magnifiers depends on the efficiency of rendering or ray-tracing technique design, which could be easily satisfied on various devices.



(a) (b)

(c) (d)

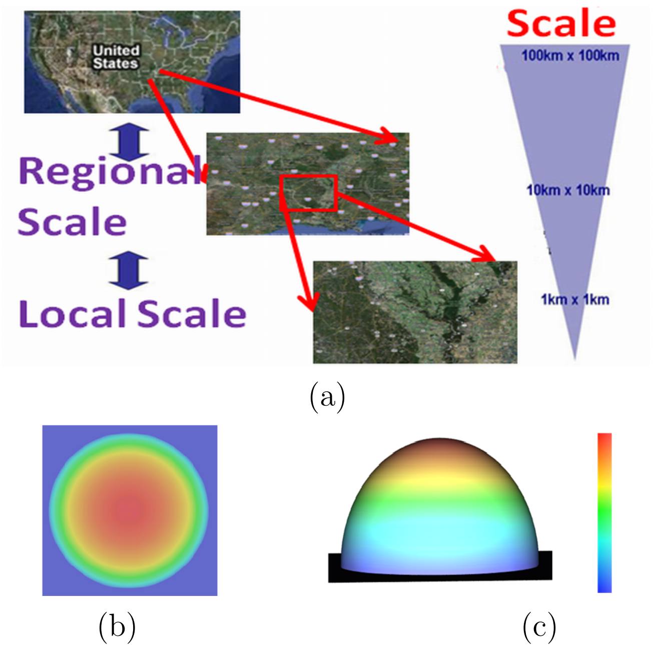
Figure 3.6: Conformal mapping for a topological quadrilateral surface: (a) The original surface. (b) The corresponding triangular mesh magnifier model. (c) The image of the conformal map, which is a rectangle. (d) A checker board texture mapping through the rectangular conformal map, demonstrating that the local angles are correctly preserved.

3.4 Implementation

A general framework is implemented for the conformal mapping based F+C visualization. The system is built using a two-tier architecture. The front-end user interface and interactive operations are based on a small number of menu bars, check boxes and pointer interactions using OpenGL and Glut libraries. With the objective to optimize the computing speed, the combination of CPU and GPU is applied.

3.4.1 Pre-computation

For efficiency purposes, the core algorithms of two pre-computation steps: mesh generation and conformal mapping, are implemented on the CPU. The computation time of conformal parameterization is proportional to the number of vertices and is slightly affected by the topology shape of each magnifier



(a)

(b) (c)

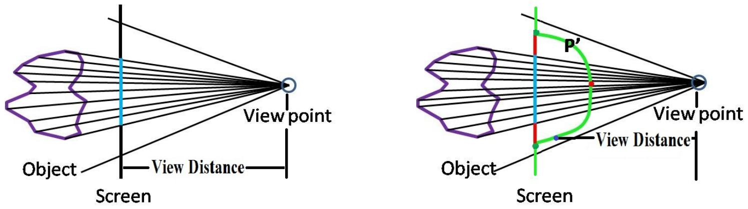
Figure 3.7: Implementation of multi-scale magnification using the conformal magnifier with a hemisphere model. (a) Multi-scale satellite images of the United States. The ROI contains more details/pixels as the scale decreases (the magnification ratio increases). (b) The top view and (c) the side view of the continuous magnification ratios of the conformal magnifier calculated by conformal mapping. The colorbar shows the scale of the magnification ratio: from large (red) to small (purple).

model. Table 4.1 shows the model shape, vertex and face count, and precomputation time of conformal parameterization for various magnifier models on a Dell desktop precision PWS670 with Intel Xeon CPU 3.60GHz, 3GB Memory and Nvidia GeForce GTX 285 graphic card. Once a magnifier model is parameterized, there is no need to do any parameterization modification for different input datasets. This is the key to the real-time performance of the system.

3.4.2 Real-time Performance

The transfer function specification, texture mapping, and volume rendering techniques are implemented using GPU acceleration.

Texture Mapping. For the 2D texture image, the system directly calls the texture mapping functions provided by the OpenGL library. Each input image is processed as an 8bit per-channel texture and directly mapped to the



(a) (b)

Figure 3.8: The new raycasting scheme: each ray is calculated based on (b) the designed magnifier plane (green curve marked as P') instead of (a) the traditional 2D plane.

Table 3.2: Time statistics of texture mapping or volume rendering for test datasets. (second)

Catalog	Name	Size	Model	Time (ms)
Information	Symbol	512^2	Hemisphere	2.6
Route+city	SF city	800×750	Gaussian	3.0
Route+city	Expressway	1080×680	Elongated	3.8
Route+city	NYC	1024^2	Gaussian	4.8
Surface	Foot	vertex:$21.3K$	Gaussian	8.2
Medical	Colon polyps	$512^2 \times 96$	Gaussian	16.3
Medical	Skull	256^3	Gaussian	14.7
Volume	Smoke	64^3	Hemisphere	0.5

selected magnifier model according to the pre-calculated conformal parameters. Unlike traditional piecewise blending methods in the image domain, supported by the conformal mapping theory, the system directly provides a continuous blending function in the parameter space. Figures 3.7b and 3.7c show the continuous magnification ratios of the hemisphere magnifier model using a color map. They also reveal two properties of conformal mapping: the model height is non-linearly proportional to the magnification ratio; and the smooth transition of the magnification ratios results from the numerical continuity in the parameter space, which theoretically supports the texture mapping of continuous-scale images. Therefore, the conformally bijective texture mapping between the parameterization value and the image pixel is syntactically and semantically trivial. In order to accelerate the search efficiency, the system has tree structures for the multi-scale images with the use of pre-marked feature points. Take Figure 3.7a as an example, the root of a tree structure is a pixel at the largest scale (100km×100km). Its direct children are right from the next small scale (10km×10km), and leaves are at the smallest scale (1km×1km). The smaller scales and leaves contain more details to reveal the local information. For the optimal speed and space, only tree structures in the ROI is stored.

Volume Rendering. In order to extend the conformal magnifier as a 3D exploration tool, different from the camera texture [117], which locally changes the 2D perspective plane of camera space, the 2D camera plane is directly replaced with the specified 3D magnifier model, as shown in Figure 3.8. Fixing a view point, the surface of the 3D hemisphere magnifier model forms a continuous view with the following fact: the closer to the view point, the smaller the view distance will be and the larger the object of interest will be. The fragment program is adapted for volume rendering proposed by Stegmaier et al. [20], considering the model shape and several factors including depth, view angle, and camera position. The framework generates results with the nonlinear magnification of 3D views in real-time with less performance degradation. Notice that this method directly works in 3D, which is more realistic and interactive than the direct deformation on the 2D rendered image.

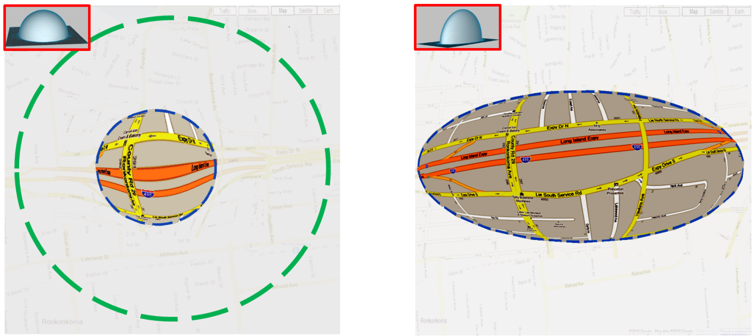
The texture mapping or volume rendering needs to be re-computed whenever the focus or the magnifier model changes, but with real-time performance. Table 3.2 shows the texture mapping or volume rendering time for all experimental cases on the same desktop mentioned above. Table 3.2 demonstrates that the design strategy is fast enough for various requirements.

3.5 Experimental Results and Discussion

This section applies the conformal magnifier to various applications and demonstrate the merits of the presented framework.

3.5.1 Route and Map Visualization

City and route F+C visualization is another popular application. For the virtual city, Qu et al. [118] have described an F+C route zoom and information overlay method for 3D urban environments. Trapp et al. [119] have proposed a 3D generalization lens for the interactive F+C visualization and applied it to virtual city models with different levels of structural abstraction. For the viewing of routes, Ziegler et al. [120] have presented an automated system for generating context-preserving route maps. It depicts navigation and orientation routes as a path between nodes and edges of a topographic network. Recently, Karnick et al. [121] have presented a novel multifocus technique to generate a printable version of a route map that shows the overview and detail



(a) (b)

Figure 3.9: Efficient route view using the conformal magnifier. Instead of using (a) hemisphere models with a small (blue dashed circle) and a large radius (green dashed circle), (b) an elongated model can magnify the entire route of interest without taking any extra movement or magnifying any non-interest area. It also has no widening artifacts of the routes of interest.

views of the route within a single, consistent visual frame. These methods, however, may fail to preserve either the local features or the overall context of the surrounding map constituents, such as nearby cities, forests, and other useful information. The presented method magnifies the target routes with local shape preservation while keeping all context information, which improves the magnification function for routes and maps.

The presented method can enable the user to directly design arbitrary magnifier models based on the shape of ROI on the route map. This design has effective merits for the route view: cover the entire ROI without the need of moving the magnifier around, and only cover the ROI without any non-interest region. For example, an elongated magnifier model in Figure 3.9b covers the entire route of interest (highlighted by a blue dashed circle), while a regular hemisphere magnifier model in Figure 3.9a only covers a small part of it. A parameter Λ is defined to describe the ROI magnification efficiency as: $\Lambda = \frac{Area_{ROI}}{Area_{ROM}}$, where ROM is the region of magnification. As shown in Figures 3.9a (ROM is highlighted by a blue dashed circle) and 3.9b, although both magnification results have the high magnification efficiency ($\Lambda \approx 1.0$), Figure 3.9a fails to reveal the entire ROI. If increasing the radius of the hemisphere magnifier model in Figure 3.9a (a new ROM is highlighted by a green dashed circle) to cover the entire ROI, many non-interest regions will be magnified as well, sharply decreasing *the magnification efficiency factor* Λ down to 0.63. The system successfully provides a good magnification scheme for the route view as it presents general route trends and specific spots simultaneously.

In order to further demonstrate the merit of the conformal magnifier for

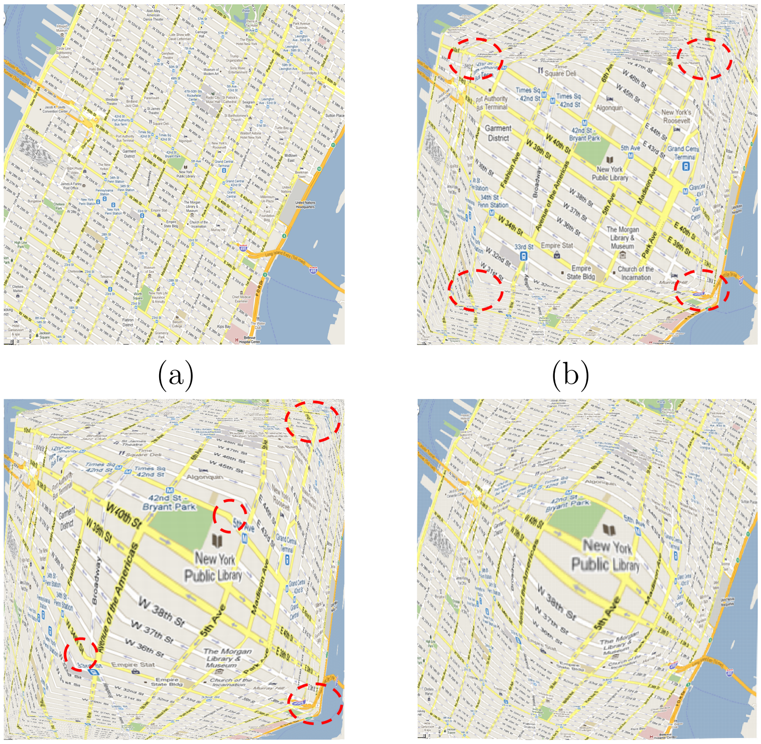

(a) (b) (c) (d)

Figure 3.10: Magnification results using different lenses for the NYC street map. (a) Original NYC map. Magnification results using (b) the bifocal lens, (c) the fisheye lens, and (d) the conformal magnifier. The red circles highlight the seriously distorted areas.

the route and city visualization, popular lenses, including bifocal [122] and fisheye lenses [56], have been implemented as comparison. Figure 3.10 shows the route/map magnification results using different lenses. By comparison, the conformal magnifier enlarges the small landmark/roads along the route of interest for a detail view with the best local shape preservation, and keeps the entire context region through a smoothest transition.

3.5.2 Volumetric Data Visualization

Our conformal magnifier can be easily applied to various volumetric datasets for real-time navigation.

Medical Datasets. Angle preserving method is important, especially for the computer aided detection (CAD), because the 3D geometric features are carried by the mapping with high fidelity. For example, Figure 3.11d

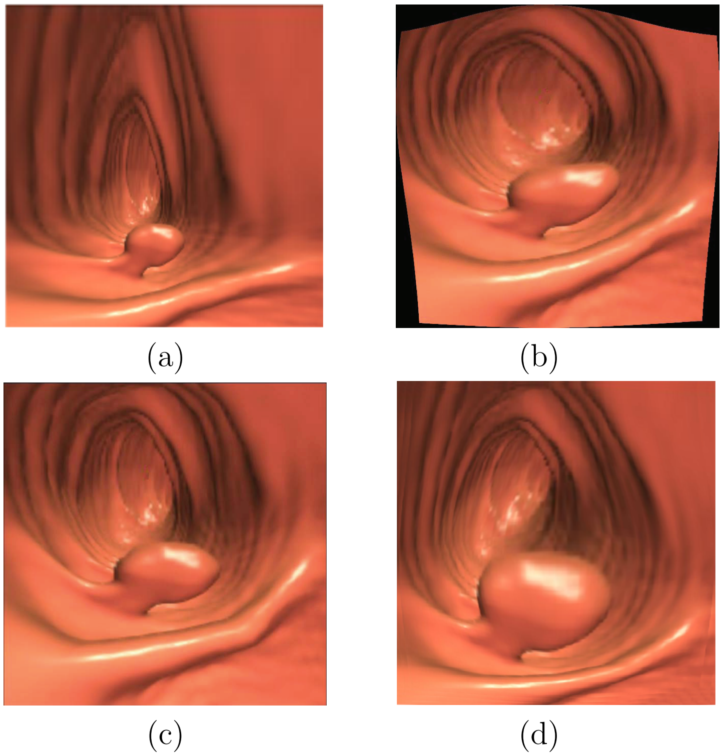


Figure 3.11: Magnification results using different lenses for the volumetric colon dataset. (a) Original colon dataset. Magnification results using the polyfocus lens (b) without and (c) with fixed boundary, following the Carpendale's approach [1]. (d) The magnification result using the conformal magnifier with a Gaussian model: the local shape/features of the interior surface and the polyp are well-preserved with the smoothest transition region.

illustrates a pre-marked polyp on a colon surface both in original 3D view and in display view of the system. It is obvious that the shape of the polyp is well-preserved after the magnification. During the diagnosis, radiologists identify colon polyps mainly based on shape information. By comparison with traditional lenses (Figures 3.11b and c), the conformal magnifier well-preserves both global and local shapes of the colon polyps for easy and accurate detection and diagnosis.

As an alternative F+C method, although the presented method is not distortion-free in the focus area, it has a solid mathematical foundation to preserve both local shape and global structure simultaneously and to produce a smooth transition area without any serious distortion. Both properties are important merits for the accurately visual cognition. Figure 3.12 shows another magnification comparison results using two prominent methods with the

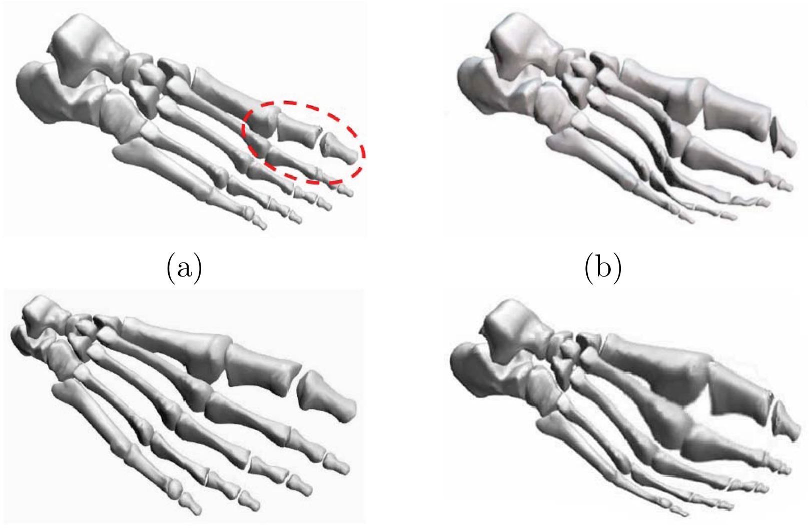
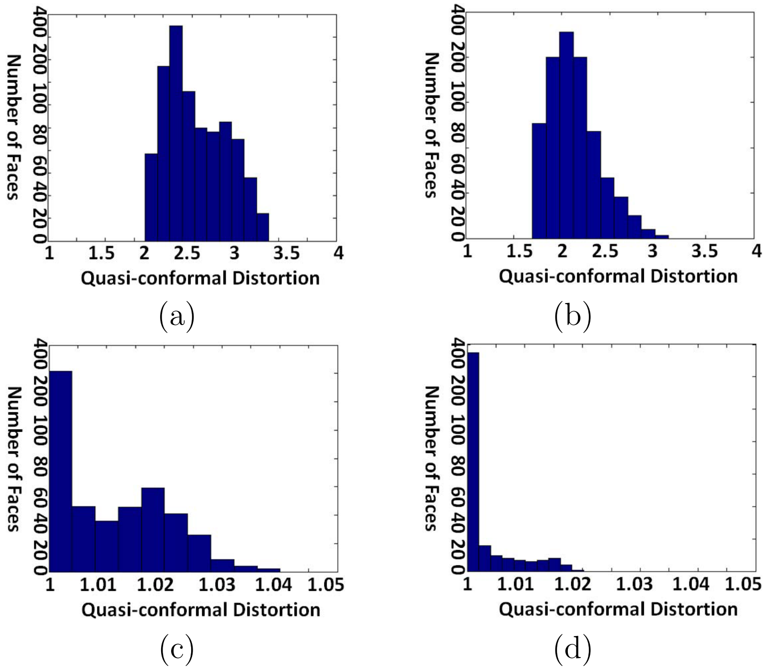


Figure 3.12: Magnification results using different lenses for the volumetric foot dataset. (a) Original foot surface mesh dataset with a predefined focal area (red circle). Magnification results using (b) the polyfocus lens following the Carpendale's approach [1], and (c) the energy based distortion minimization method [2] (courtesy to Wang et al. [2]). (d) The magnification result using the conformal magnifier with a Gaussian model: both local and global shape/features are well preserved.

volumetric foot dataset. From the perspective of local shape preservation, as shown in Figure 3.12b, Carpendale's method [1] seriously deforms the surrounding transition area (two toe bones near the ROI) without preserving the original shape features. By comparison with the original input dataset as shown in Figure 3.12a, the magnification results generated by Carpendale's [1] method (Figure 3.12b) and the presented technique (Figure 3.12d), respect the global shape/structure of foot dataset without any obvious shape confusion for the accurate object recognition. However, as shown in Figure 3.12c, after setting the ROI (two foot toes), the magnification result generated by Wang's method [2] well-preserves the local structure/shape in the focus area, but seriously affects the context region and introduces visual artifacts for the global structure, such as the extra extension of the left three toes and the overall foot width (the surrounding regions are enforced to expand because the cubes are connected [2]). Wang et al. [2] have also mentioned this problem as one of their major limitations.



(a) (b) (c) (d)

Figure 3.13: Histograms show the distribution of the quasi-conformal distortion using (a) the fisheye lens, (b) the polyfocus lens following [1], and the conformal magnifiers with (c) a hemisphere model and (d) a Gaussian model.

3.5.3 Distortion Measurement

In order to quantitatively measure the angle distortions of various lenses, *the distortion of conformality* is computed as: the ratio of the larger to the smaller eigenvalue of the Jacobian matrix generated using the parameterization value with respect to lens definition functions or magnifier models (details in [78, 108]). The ideal conformality is 1.0, which is the most conformal. But due to the numerical problem, it cannot be exactly 1.0. The region with a lower distortion value means that it is more similar to its original shape, while the region with a higher distortion value is naturally stretched. Histograms are used to show the distribution of *the distortion of conformality* for different lenses. As shown in Figures 3.13c and 3.13d, the mapping of the conformal magnifier is close to conformal everywhere: the maximal errors of the two predefined magnifier models are less than 4% and 2%, respectively. By comparison with the fisheye lens (Figure 3.13a) and the Carpendale's polyfocus lens [1] (Figure 3.13b), the presented method is statistically significant for the angle preservation (local shape preservation).

Chapter 4

Area-Preservation based Focus+Context Visualization

4.1 Motivation

Angle-preservation (conformal) mapping flattening preserves local shapes, and thus has been broadly used in many feature oriented applications in visualization and medical imaging. However, conformal method usually substantially distorts area, which fails to display accurate size of area, including height, width, thickness or diameter of ROIs. Unfortunately, these distorted area parameters are extremely important in many medical image recognition and auto diagnosis applications, such as brain fold detection [75] or colon polyps detection and diagnosis [77, 123]. Moreover, it is well known that conformal mapping induces severe area distortions for surfaces with long tube shapes, such as the elongated lion head model, as shown in Fig. 4.1. This disadvantage derives from the fundamental obstacle of conformal mapping theory and it can not be easily overcome. Imagine a cylinder $\mathbf{r}(\theta, z) = (\cos\theta, \sin\theta, z)$, a conformal mapping $\phi(\theta, z) = e^{-z}(\cos\theta, \sin\theta)$ maps it to the unit disk, the area distortion factor e^{-2z} is exponential with respect to the height z, and in practice easily exceeds the machine precision.

By comparison, area-preservation mapping can generate accurate and information lossless mapping results, which is a key objective for many medical imaging applications, with the ability to carry out measurements for detecting anatomic abnormalities. For example, in virtual colonoscopy, the physician may want to measure and compare different sizes of polyps, to determine disease conditions and cancer risks [76]. A special case of this problem also occurs in any application where volume or area measurement is critical (e.g., brain data in [75, 124, 125]). From human cognition perspective, area-preservation

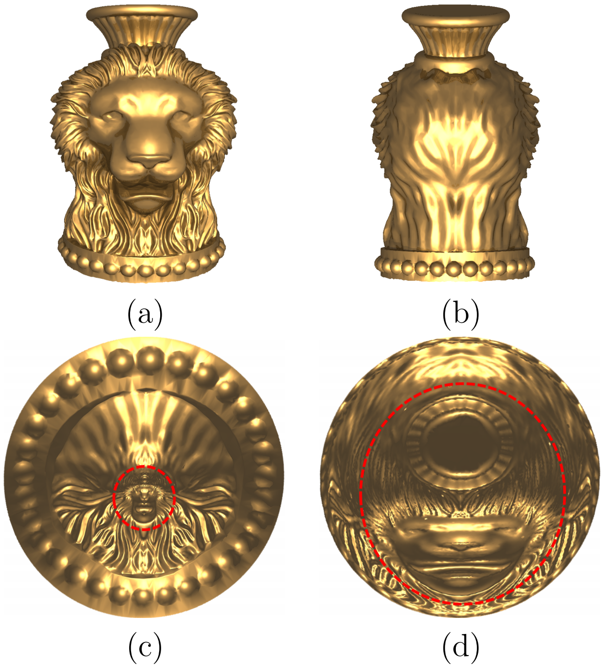


Figure 4.1: Disadvantages of conformal mapping for elongated shapes. (a) Front view and (b) back view of the elongated lion head surface model. Surface flattening results induced by (c) conformal mapping and by (d) the area-preservation mapping. Conformal mapping generates major area distortions for both the lion face and the vase regions, while the presented method can preserve them accurately for clear view without losing any information (highlighted by the red circles).

mapping and flattening can also enhance the viewer's ability to easily recognize the component-aware patches or long branch parts distribution of models, and consequently understand the local feature with the knowledge of a global structure (Figure 4.1). Therefore, area-preservation mapping has vast potentials to be applied to many related visualization and graphics applications.

To simultaneously tackle the above challenges, a flattening framework is developed, which provides a global view of the surface with a minimum of area distortions, while still, at the same time, maximally preserving local angle/shape features on the flattened surface. This work introduces a solution to this problem using the technique of optimal mass transport (OMT), based on Monge-Brenier theory [126].

4.1.1 Optimal Mass Transport

The presented solution is inspired by the similarity between a mapping problem and *optimal mass transport problem*. Monge [86] raised the classical "optimal mass transport problem" that concerns determining the optimal way, with minimal transportation cost, to move a pile of soil from one place to another. Formally, spaces X and Y are with measures μ and ν, respectively, the transportation cost for moving from $x \in X$ to $y \in Y$ is $c(x, y)$. The optimal transport map $T : X \to Y$ is measure preserving, namely for any $B \subset Y$, $\nu(B) = \mu(T^{-1}(B))$, and minimizes the total transportation cost $\int_X c(x, T(x))\mu(x)dx$. From OMT perspective, the surface mapping/flattening can be viewed as an optimal transport map T, and the size of area can be viewed as the preserved measure μ and ν.

The solution of the OMT problem lies at the following theory. Kantorovich [90] has proved the existence and uniqueness of the optimal transport plan using a linear programming method on n^2 variables. This Monge-Kantorovich optimization has been used in numerous fields from physics, econometrics to computer science, including data compression and image processing [127]. Recently, researchers have realized that optimal transport could provide a powerful tool in image processing, if one could reduce its high computational cost [75, 89]. However, it has one fundamental disadvantage that the number of variables is $O(n^2)$, which is unacceptable to visualization and graphics applications since a high resolution 3D/volume dataset normally includes more than 10^6 vertices.

An alternative Monge-Brenier optimization can significantly reduce the number of variables. Brenier [91] has developed a different approach for special optimal transport problem, where the cost function $c(x, y)$ is a quadratic distance $c(x, y) = \|x - y\|^2$. Brenier's theory proves that there is a convex function $u : X \to \mathbb{R}$, the unique optimal transport map is given by its gradient map, $x \to gradu(x)$. The Monge-Brenier's approach reduces the unknown variables from n^2 to n, which greatly reduces the computational cost, and improves the efficiency. Here, Monge-Brenier's approach is followed.

4.1.2 Comparisons and Contributions

Comparison. The presented method mainly follows the Monge-Brenier approach, based on the variational principle [92]. Comparing to the state-of-the-art techniques, it has many merits as follows:

- Comparing to the Monge-Kantorovich approaches [89, 90, 128], the presented method only requires $O(n)$ variables instead of $O(n^2)$ variables.

The system greatly reduces the storage complexity, and it is thus much more efficient.

- Comparing to the Monge-Brenier based approaches [129–131], all the existing methods are for image processing tasks. The presented method however focuses on surfaces. For image processing, the samples are relatively uniform, and therefore, the computation is relatively stable. In these cases, the sample points are produced by the conformal mapping, the sample density is highly non-uniform, and thus conventional methods are very vulnerable and error-prone for the large area distortions induced by the conformal mapping. All experiments indicate that conventional methods are not robust enough. Therefore, the *step length control* algorithm (Section 4.2) is used to improve the robustness.

- Comparing to the minizing flow methods [75, 88, 132], the solution of latter is equivalent to a gradient descend method for optimizing the transportation cost. In contrast, the presented method is equivalent to the Newton's method to optimize a convex energy, which has a higher order convergence rate. Therefore, the presented method is more efficient.

The key contribution of this work is the introduction of a novel area-preservation mapping/flattening algorithm using the optimal transport technique, based on Monge-Brenier theory. The new method has the following merits:

- It reduces the number of variables from n^2 to n, greatly reducing the complexity.

- It converts the optimal mass transport problem to a convex optimization problem, and can be solved using Newton's method, greatly improving the efficiency.

- The algorithm gives users full control of the size of area everywhere. Users can design and manipulate the area of each triangular component freely, improving the flexibility.

4.2 Theoretical Foundation

In this section presents the theoretical foundation.

4.2.1 Optimal Mass Transport

Monge's Problem. The problem of finding a map T minimizing Equation 4.1 (such that $\nu = T_{\#}\mu$), has been first studied by Monge [126] in the 18th century. Let X and Y be two metric spaces with probability measures (volumes or areas) μ and ν, respectively. Assume X and Y have equal total measures (volumes or areas):

$$\int_X \mu = \int_Y \nu.$$

A map $T : X \to Y$ is *measure preserving* (volume or area preservation) if for any measurable set $B \subset Y$, such that:

$$\mu(T^{-1}(B)) = \nu(B).$$

Denote by $c(x, y)$ the transportation cost for sending $x \in X$ to $y \in Y$, then, the total *transportation cost* is given by:

$$\int_X c(x, T(x))d\mu(x). \tag{4.1}$$

If this condition is satisfied, ν is said to be the push forward of μ by T, and then write $\nu = T_{\#}\mu$.

In the 1940s, Kantorovich [90] has introduced the relaxation of Monge's problem and solved it using linear programming. At the end of 1980's, Brenier [91] has proved that there is a convex function $f : X \to \mathbb{R}$, and the optimal mass transport map is given by the gradient map $x \to \nabla f(x)$.

4.2.2 Discrete Optimal Mass Transport

Suppose μ has compact support on X, define:

$$\Omega = supp\ \mu = \{x \in X | \mu(x) > 0\},$$

and assume Ω is a convex domain in X. The space Y is discretized to $Y = \{y_1, y_2, \cdots, y_n\}$ with Dirac measure $\nu = \sum_{j=1}^{n} \nu_j \delta(y - y_j)$.

Define a *height vector* $\mathbf{h} = (h_1, h_2, \cdots, h_n) \in \mathbb{R}^n$, consisting of n real numbers. For each $y_i \in Y$, construct a hyperplane defined on X:

$$\pi_i(\mathbf{h}) : \langle x, y_i \rangle + h_i = 0, \tag{4.2}$$

where $\langle , \rangle$ is the inner product in $\mathbb{R}^n$.

Define a function:

$$u_{\mathbf{h}}(x) = \max_{1 \leq i \leq n} \{\langle x, y_i \rangle + h_i\}, \tag{4.3}$$

then $f(\mathbf{h}, x)$ is a convex function. Next denote its graph by $G(\mathbf{h})$, which is an infinite convex polyhedron with supporting planes $\pi_i(\mathbf{h})$. The projection of $G(\mathbf{h})$ induces a polygonal partition of Ω,

$$\Omega = \bigcup_{i=1}^{n} W_i(\mathbf{h}), W_i(\mathbf{h}) = \{x \in X | u_{\mathbf{h}}(x) = \langle x, y_i \rangle + h_i\} \cap \Omega. \tag{4.4}$$

Each cell $W_i(\mathbf{h})$ is the projection of a facet of the convex polyhedron $G(\mathbf{h})$ onto Ω. The convex function $u_{\mathbf{h}}$ on each cell $W_i(\mathbf{h})$ is a linear function $\pi_i(\mathbf{h})$, therefore, the gradient map

$$grad\ u_{\mathbf{h}} : W_i(\mathbf{h}) \to y_i, i = 1, 2, \cdots, n. \tag{4.5}$$

maps each $W_i(\mathbf{h})$ to a single point y_i.

The following theorem plays a fundamental role here:

Theorem 4.2.1 *For any given measure ν, such that*

$$\sum_{j=1}^{n} \nu_j = \int_{\Omega} \mu, \nu_j > 0,$$

there must exists a height vector $\mathbf{h}$ *unique up to adding a constant vector* $(c, c, \cdots, c)$, *the convex function Equation 4.3 induces the cell decomposition of* Ω, *Equation 4.4, such that the following* area-preserving constraints *are satisfied for all cells,*

$$\int_{W_i(\mathbf{h})} = \nu_i, i = 1, 2, \cdots, n. \tag{4.6}$$

Furthermore, the gradient map $grad u_{\mathbf{h}}$ optimizes the following transportation cost

$$E(T) := \int_{\Omega} |x - T(x)|^2 \mu(x) dx. \tag{4.7}$$

The existence and uniqueness have been first proven by Alexandrov [133] using a topological method. The existence has been also proven by Argmstrong [134], and the uniqueness and optimality have been proven by Brenier [91].

Recently, Gu et al. [92] have given a novel proof for the existence and uniqueness based on variational principle. This framework follows their approach. First, define the admissible space of the height vectors:

$$H_0 := \{\mathbf{h} | \int_{W_i(\mathbf{h})} \mu > 0, \sum_i h_i = 0\}.$$

Then, define the energy $E(\mathbf{h})$ as the volume of the convex polyhedron bounded by the graph $G(\mathbf{h})$ and the cylinder through Ω minus a linear term,

$$E(\mathbf{h}) = \int_\Omega u_{\mathbf{h}}(x)\mu(x)dx - \sum_{i=1}^{n} \nu_i h_i. \tag{4.8}$$

The gradient of the energy is given by:

$$\nabla E(\mathbf{h}) = \left(\int_{W_i(\mathbf{h})} \mu - \mu_i \right), \tag{4.9}$$

Suppose the cells $W_i(\mathbf{h})$ and $W_j(\mathbf{h})$ intersects at an edge $e_{ij} = W_i(\mathbf{h}) \cap W_j(\mathbf{h}) \cap \Omega$, then, the Hessian of $E(\mathbf{h})$ is given by:

$$\frac{\partial^2 E(\mathbf{h})}{\partial h_i \partial h_j} = \begin{cases} \int_{e_{ij}} \mu & W_i(\mathbf{h}) \cap W_j(\mathbf{h}) \cap \Omega \neq \emptyset \\ 0 & otherwise \end{cases} \tag{4.10}$$

In Gu et al. [92], it is proven that H_0 is convex, and the Hessian is positive definite on H_0, this implies the convexity of the energy in Equation 4.8. Furthermore, the global unique minimum $\mathbf{h}$ is an interior point of H_0. At the minimum point, $\nabla E(\mathbf{h}) = 0$, this implies the gradient map $gradu_{\mathbf{h}}$ meets the measure-preserving constraints in Equation 4.6. Furthermore, this gradient map is the optimal mass transportation map.

Due to the convexity of the volume energy (Equation 4.8), the global minimum can be obtained efficiently using Newton's method. Comparing to Kantorovich's approach, where there are n^2 unknowns, this approach has only n unknowns.

4.2.3 Conformal Mapping

Current work first applies conformal mapping [135] to map a topological surface onto the planar domain. Suppose $(S, \mathbf{g})$ is a surface embedded in $\mathbb{R}^3$, with the induced Euclidean metric $\mathbf{g}$. Let the mapping $\phi : (S, \mathbf{g}) \to (\mathbb{D}, dx^2 + dy^2)$ transforms the surface to the planar unit disk $\mathbb{D}$, where $dx^2 + dy^2$ is the planar Euclidean metric. Therefore, ϕ is a conformal mapping, or angle-preservation mapping, if ϕ is a diffeomorphism, such that:

$$g(x, y) = e^{2\lambda(x,y)}(dx^2 + dy^2),$$

where $\lambda : S \to \mathbb{R}$ is a smooth function defined on the surface, the so called *conformal factor*.

Theorem 4.2.2 (Riemann Mapping) *Suppose $(S, \mathbf{g})$ is an oriented metric surface, which is of genus zero with a single boundary. Given an interior point $p \in S$ and a boundary point $q \in \partial S$, there is a unique conformal mapping $\phi : S \to \mathbb{D}$, satisfying $\phi(p) = 0$ and $\phi(q) = 1$.*

This work follows the approach proposed by Dominitz et al. in [75], which gives an area-preservation mapping with shape preservation. First, map the surface $(S, \mathbf{g})$ onto the planar disk using a Riemann mapping $\phi : S \to \mathbb{D}$, then the conformal factor defines a measure on $\mathbb{D}$, $e^{2\lambda(x,y)}dxdy$. Then construct an optimal mass transport map $\tau : (\mathbb{D}, e^{2\lambda(x,y)}dxdy) \to (\Omega, dxdy)$, where Ω is a planar convex domain, the composition $\tau \circ \phi : (S, \mathbf{g}) \to (\Omega, dxdy)$ is an area-preservation mapping.

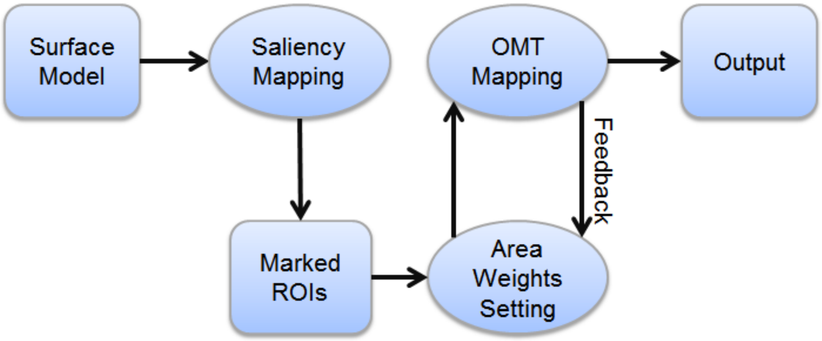


Figure 4.2: The pipeline of the OMT based area-preserving framework.

4.3 Optimal Mass Transport Map

This section gives the detailed algorithms for the optimal mass transport map generation. Figure 4.2 shows the pipeline of the OMT based area-preserving framework. The input surface is approximated by a triangular mesh M, with vertex sets V, face set F and a convex planar domain Ω, represented as a convex polygon. The goal is to compute an area-preserving map from the mesh M to the planar domain Ω. The discrete algorithm is based on the Monge-Briener theory and utilizes the variational principle to solve the optimization problem. For the input, aiming to get ROIs with arbitrary shape (e.g., irregular shape of the brain folds), the saliency map [3] is utilized to for the ROI detection. Once users specify local area weight $\overline{w_i}$ everywhere, the system will iteratively solve the OMT map and refine the area-preserving result that yields strict equality of weighted sizes of area between the input surface and flattened plane. The area weighting parameter $\overline{w_i}$ is defined as weighted areas in 2D or weighted volumes in 3D. After the generation of the OMT map (bijectively area-preservation mapping), the system directly applies the ROI guided texture mapping to obtain the output.

Merits of Saliency Map. The application of saliency map can (1) accurately detect arbitrary ROI shape to obtain the accurate area preservation; and (2) provide hierarchical resolution of surface models, supporting the reduction of triangles in the context area, while preserving high resolution ones in ROIs, for the purpose of fast computation. Take the brain model as an example, instead of using the original model with 100K faces, with the saliency guided ROI detection, the face account can be significantly reduced to 10K or less (data oriented).

4.3.1 Initialization

The system algorithm uses the conformal mapping result (angle preservation) to set up the initial position for each vertex v_i. First normalize the mesh such that its total area equals to the area of the planar domain Ω. Then initialize a discrete conformal mapping $\phi : M \to \mathbb{D}$. This framework utilizes the discrete Ricci flow method [135] to achieve this step. Then, after assigning each vertex a target area $\bar{w}_i$, the algorithm defines for each vertex $v_i \in V$ the Dirac measure associated with it, as one third of the total area of faces adjacent to it,

$$\mu(v_i) = \frac{1}{3} \sum_{[v_i, v_j, v_k] \in F} Area([v_i, v_j, v_k]),$$

where $[v_i, v_j, v_k]$ represents the triangle formed by vertices v_i, v_j and v_k.

Use the images of all the vertices as the sample points of the unit disk $\mathbb{D}$, $Y = \{\phi(v_i) | v_i \in V\}$, each sample $\phi(v_i)$ is associated with the Dirac measure $\mu(v_i)$. By translation and scaling, further transform Y to be contained by Ω.

4.3.2 Optimal Mass Transport Mapping

According to the Monge-Briener theory, we need to find the height vector $\mathbf{h} = (h_1, h_2, \cdots, h_n)$. Fix a height vector, the support planes are given by

Algorithm 3 Area-Preservation Mapping

```
Input: Input triangular mesh $M$, total area $\pi$ and area difference threshold $\delta w$.
Output: A unique diffeomorphic area-preservation mapping $f : M \to \mathbb{D}$, where $\mathbb{D}$ is a unit disk. The area $w_i$ of each cell $W_i \in \mathbb{D}$ is close to the target area $\overline{w_i}$.
1. Run conformal mapping by discrete Ricci flow method [135] $\phi : M \to \mathbb{D}$, where $\mathbb{D}$ is a unit disk. Assign each site $\phi(v_i) \in \mathbb{D}$ with zero power weight, and target area $\overline{w_i} = \mu(v_i)$ defined above. Translate and scale all sites so that they are in the unit disk.
2. Compute the power diagram and calculate the area $w_i$ of each cell $W_i$.
3. Compute the dual power Delaunay triangulation, and compute the lengths of edges in the diagram and triangulation to form the Hessian matrix of the convex energy in Equation 4.7 .
4. Update the power $\mathbf{h} \leftarrow \mathbf{h} + H^{-1}(\bar{\mathbf{w}} - \mathbf{w})$.
5. Repeat step 2 through step 4, until $\|w_i - \overline{w_i}\|$ of each cell is less than $\delta w$.
6. Compute the centroid of cell $W_i$, denoted as $c_i$. Then the area-preservation mapping is given by $\tau^{-1} \circ \phi(v_i) = c_i$, where $\tau$ is the Brénier map $\tau : W_i \to \phi(v_i)$.
```

$\pi_i(\mathbf{h}) : \langle x, y_i \rangle + h_i$, the convex function is $u_{\mathbf{h}}(x) = \max_i \langle x, y_i \rangle + h_i$, and its graph $G(\mathbf{h})$ can be computed as upper envelope of the supporting plane $\pi_i(\mathbf{h})$. The projection of $G(\mathbf{h})$ onto Ω forms a polygonal partition $\Omega = \bigcup_i W_i(\mathbf{h})$.

The implementation details are listed in Alg. 3. In order to preserve the area of cell W_i, the system need to iteratively update the virtual variable for each vertex with height vector $\mathbf{h} = (h_1, h_2, \cdots, h_n)$. Thus, in each iteration, first compute the power diagram, using each vertex as a point and the weighted radius as the power in the diagram. Then, in step 3, compute the dual triangulation of this calculated power diagram. Then record every edge length in both the power diagram and its dual triangulation in this step to form the Hessian matrix. In step 4 (the last step of each iteration), the system uses the Newton's method to solve the gradient energy equation (Equation 4.9) and to update the height vector $\mathbf{h}$ until it satisfies that $\|w_i - \overline{w_i}\|$ of each cell is less than δw. Finally, in last step, the system updates the vertex position as the center of the power Voronoi diagram to obtain the area-preserving parameterization result.

Initial Height Vector. At the initial stage, the algorithm scales and transforms a point set Y to ensure they are contained in Ω, and then computes the Voronoi diagram with zero power weights, or namely, with initial heights

$h_i = -1/2\|y_i\|^2$, where $\|y_i\|^2$ is the point position in the planar domain. This guarantees that all the cells are non-empty.

Power Diagram. The OMT based area-preserving computation for the partition of Ω is equivalent to the classical power diagram in computational geometry [92]. Given a point set $Y = \{y_1, y_2, \cdots, y_n\}$, each point y_i associated with the weight w_i as its power, the power distance from any point x to y_i is defined as:

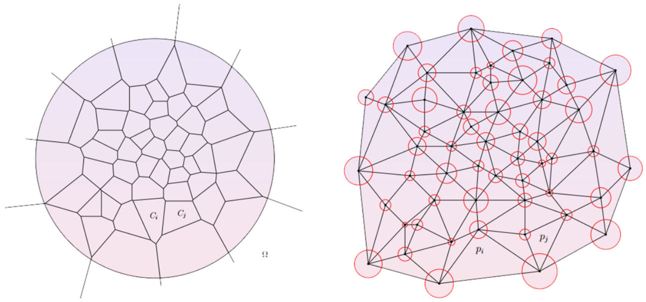
$$Pow(x, y_i) = \frac{1}{2}\|x - y_i\|^2 - \frac{1}{2}w_i,$$

Then, the power diagram is the Voronoi diagram when using the power distance instead of the standard Euclidean distance.

In this method, the partition induced by the convex function $u_{\mathbf{h}}$ in Equation 4.3 is equivalent to the power diagram with the power weight:

$$w_i = 2h_i + \langle y_i, y_i \rangle.$$

Therefore, the computation can be carried out using power diagram functionalities in standard computational geometry library, such as CGAL [136]. The construction of the power Voronoi diagram and the power Delaunay triangulation are illustrated in Figure 4.3.



(a) (b)

Figure 4.3: Construction of (a) the power Voronoi diagram and (b) the power Delaunay triangulation.

Hessian Matrix. The algorithm represents the gradient of energy $\nabla E(\mathbf{h})$ in Equation 4.9 as the area changes of cell $(\bar{\mathbf{w}} - \mathbf{w})$, where $\bar{\mathbf{w}}$ and $\mathbf{w}$ as weighting values. Then, compute the dual triangulation and the cell areas to form the gradient, as in Equation 4.9,

$$\nabla E(\mathbf{h}) = (Area(W_i(\mathbf{h}) \cap \Omega))$$

Following the theory proposed by Gu et al. [92], in order to form the Hessian as in Equation 4.10, first compute all edge lengths e_{ij} and the dual edge lengths $\bar{e}_{ij}$ from the power diagram and its dual triangulation (Figure 4.3). Then use the following matrix: $H(\mathbf{h}) = (h_{ij}(\mathbf{h}))$, where

$$h_{ij}(\mathbf{h}) = \begin{cases} -|e_{ij}|/|\bar{e}_{ij}| & i \neq j, W_i \cap W_j \cap \Omega \neq \emptyset \\ -\sum_{k\neq i} h_{ik} & i = j \\ 0 & otherwise, \end{cases}$$

h_{ij} is the (i, j) entry of a matrix, $(i \neq j)$ is the off diagonal entry, and the diagonal entry is defined as $h_{ii} = -\sum_{k\neq i} h_{ik}$ (namely, h_{ii} is equal to the sum of all off diagonal entries).

Then, use Newton's method to update the height vector

$$\mathbf{h} \leftarrow \mathbf{h} + \epsilon H(\mathbf{h})^{-1} \nabla E(\mathbf{h}),$$

where ϵ is the step length.

Step Length Control. During the computation, it is crucial to ensure that all the cells $W_i(\mathbf{h}) \cap \Omega$ are non-empty. Suppose at step k all the cells are non-empty, then update $\mathbf{h_k} \leftarrow \mathbf{h_k} + \epsilon H(\mathbf{h}_k)^{-1} \nabla E(\mathbf{h_k})$. If some cells are empty in the power diagram induced by $\mathbf{h}_{k+1}$, the system will return to $\mathbf{h}_l$, shrink the step length ϵ to be $1/2\epsilon$, and try again. If some cells are still degenerated, further shrink the step length iteratively, until all the power cells are non-empty.

4.3.3 ROIs Guided Texture Mapping

After computing the bijective area-preservation surface mapping between the 3D surface model and the flattened 2D disk or rectangular parameter domain, the texture mapping is straightforward with the ROIs guided alignment. With respect to user predefined mapping criteria (e.g., fix the disk boundary or fix the four corner points of a rectangular domain with the alignment of ROIs), the bijective texture mapping between parameterizations and image pixels is syntactically and semantically trivial. Directly call texture mapping functions provided by OpenGL with bilinear interpolation, which is fast and easy to implement. Moreover, the system considers the model shape and rendering factors, such as depth, view angle, and camera position to obtain reality style visualization, especially for medical data. The pixel color and alpha can be adjusted by the user defined transfer functions.

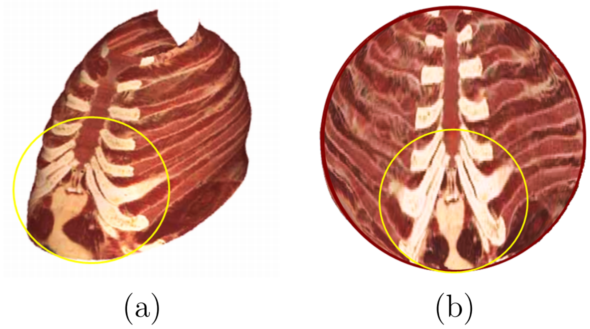


Figure 4.4: Surface flattening of a chest model using the area-preservation mapping for direct display and accurate measurement. The yellow circles highlight the corresponding ROIs between (a) the 3D surface model and (b) the 2D flattening plane.

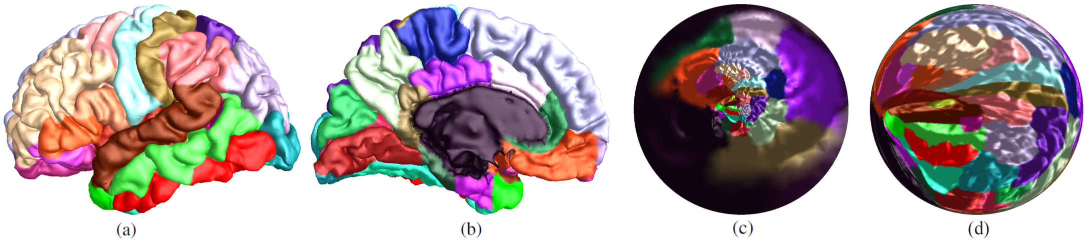


Figure 4.5: Surface flattening and area manipulation using a brain surface model. A brain surface extracted from the MRI data with color coded components with (a) a lateral view and (b) a medial view. The major brain folds are color coded for easy recognition. (c) Conformal mapping result, and (d) the area-preservation mapping result. By comparison, the presented method accurately preserves the size of area for each fold component, while conformal mapping leads to severely area distortions (severely shrinking some brain folds while enlarging the others).

4.4 Area-Preservation Mapping Applications

To demonstrate the merits of the presented area-preservation mapping method, various visualization applications and a distortion measurement analysis is presented.

4.4.1 Medical Applications

The framework is tested using various medical data. Figure 4.4 shows a major advantage of the area-preservation mapping and flattening method. The mapping framework can bijectively project the 3D surface model into a unit

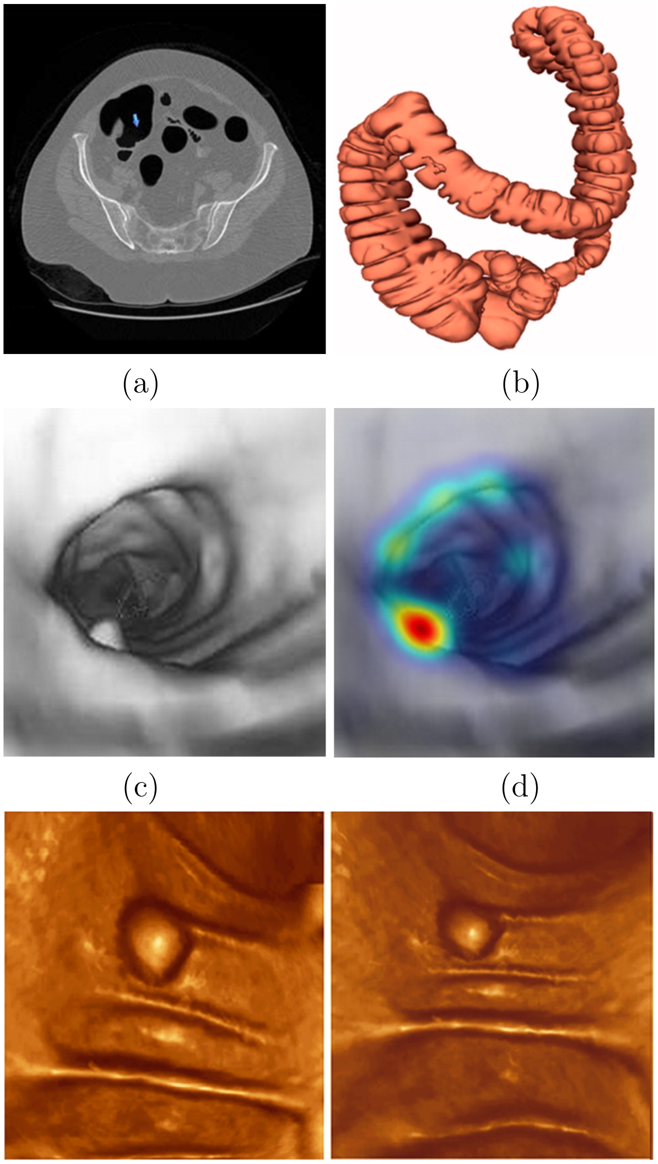


Figure 4.6: Saliency map guided area-preservation mapping using a colon model. (a) A slice of CT colon images. (b) A colon surface, extracted from (a). (c-d) Possible polyps detected using the saliency map [3]. Surface flattening results using (e) the area-preservation mapping and (f) the conformal mapping. By comparison, the final result generates the accurate polyp size for area measurement (verified by the doctor marked area measurement of the polyp as ground truth) without any severe angle distortion.

2D disk, so the doctor can directly and accurately visualize and measure the size of the entire ROI area without repeatedly rotating and scaling.

Figure 4.6 shows the saliency map [3] guided area-preservation mapping for diverse computer aided detection (CAD) applications. After extracting the surface model from CT colon data (Figure 4.6a), the electronic biopsy [107] is working as saliency map, for the polyp detection (Figure 4.6b). The area-preservation flattening framework is only applied in the detected ROIs, providing an area-preserving view of polyps for the accurate measurement of the diameter and the size of area (verified by the doctor marked measurement as ground truth in the original 3D surface model). By comparison with the conformal mapping method, the framework still preserves major shape characteristics of the colon surface (e.g., colon folds) without any obviously visual distortion (Figure 4.6c).

Flattening the brain surface with area preservation is important to visualize and study neural activity or to detect diseases/disfunctions [137]. For the easy recognition of different brain folds, colors are used to mark different folds as the ROIs (Figure 4.5a and Figure 4.5b). In contrast to the conformal mapping result (Figure 4.5c), Figure 4.5d shows the area-preservation mapping result using the MRI brain dataset, which accurately displays accurate sizes of brain folds without severely compressing or stretching. Moreover, users can set different weight coefficients in ROIs to flexibly adjust sizes of different ROI areas (default 1X: equal area).

4.4.2 Informatics Applications

With the general application property of parameterization and texture mapping, the framework can be easily implemented for various informatics applications including earth map, city map and graph.

The fundamental challenge for earth visualization lies at mapping the sphere earth model to a planar domain with maximal information preserved. Direct projection only projects the half sphere, and then causes severe information lost (Figure 4.7b). The state-of-the-art method, such as conformal mapping (Figure 4.7c), preserves the whole sphere with angle preservation, but severely compresses some continents while inappropriately enlarging others without any control. By comparison, the presented method (Figure 4.7d) can keep the original areas for all major continents, providing the accurate size and area impression for users.

The system also provides a direct multiresolution display, functioned as a “magic-lens” to reveal additional details in the ROIs. This method makes the multiscale alignment accurate but easy without the need of any predefined

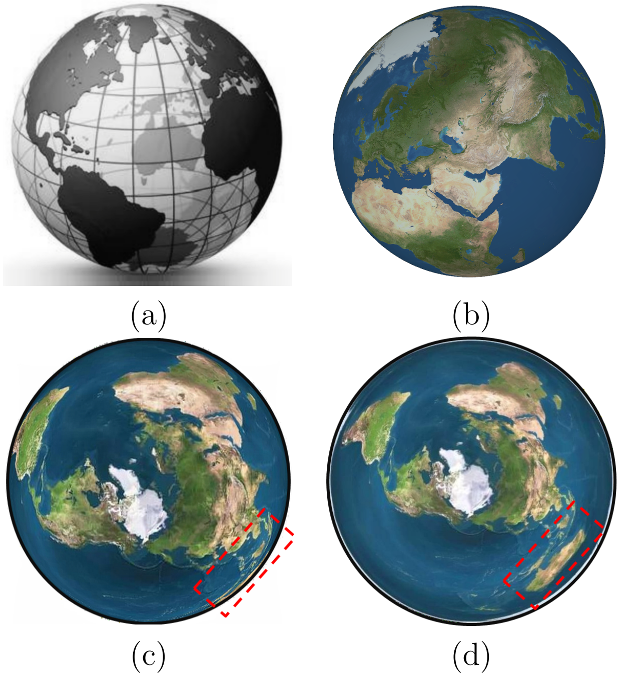


Figure 4.7: Different mapping results and comparisons using an earth surface model. (a) A 3D earth model. (b) Direct projection mapping with large information loss. (c) Conformal mapping result is with large area distortions, while (d) the area-preservation mapping result is with accurate area preservation and small angle distortion (highlighted by the red frames).

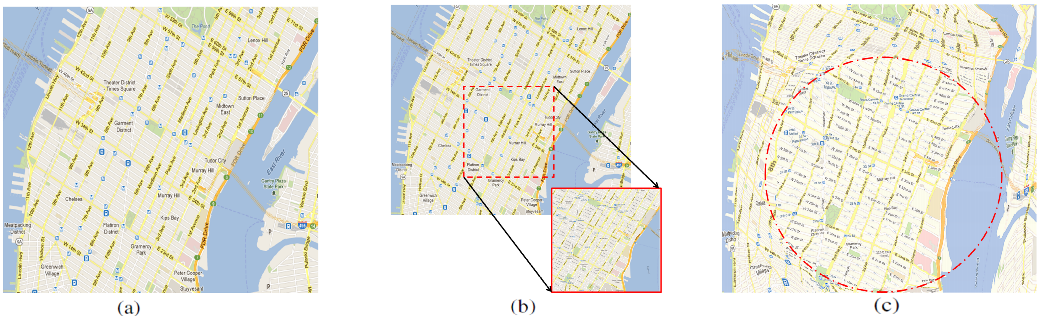
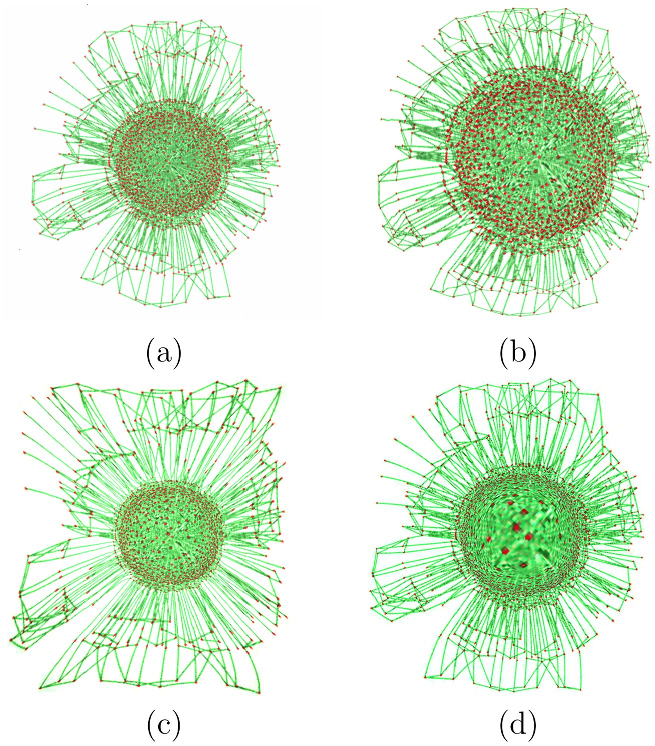


Figure 4.8: Multiresolution view without any predefined landmarks. (a) The original New York city (NYC) map. (b) NYC map with multiresolution texture images. The red frames highlight the corresponding multiresolution texture maps in the ROI. (c) Area manipulation result with a detailed view to show additional street information. The high resolution detail view can be easily aligned/merged into the low scale map without using any landmark due to the accurate area preservation.



(a) (b)

(c) (d)

Figure 4.9: Mapping comparisons using the network graph. (a) Original graph layout [4]. Magnification results with (b) the central nodes as the ROI, and with (c) the surrounding exterior nodes as the ROI, using the presented framework. (d) Magnification result using conformal magnifier [5]. By comparison, the presented method has flexible area control to generate various views.

landmark, due to the accurate area preservation. As shown in Figure 4.8, the presented method generates multiresolution texture mapping to reveal additional street information of the city map. The result demonstrates that the presented method can well magnify the ROI without causing any obvious distortion.

The presented system can generate various visual displays for the graph visualization to satisfy diverse user requirements, due to the flexible weight settings. Figure 4.9a showcases its merit using a network visualization example from the AT&T graph library [4]. Each graph node stands for a network station, while each straight line depicts direct connection between two neighbor nodes. Figure 4.9b enlarges the radius of the central core to increase the node separation, while compressing exterior nodes to further reduce the potential attention. Figure 4.9c shows another area manipulation style: compressing central nodes while enlarging exterior nodes for further separation. There is

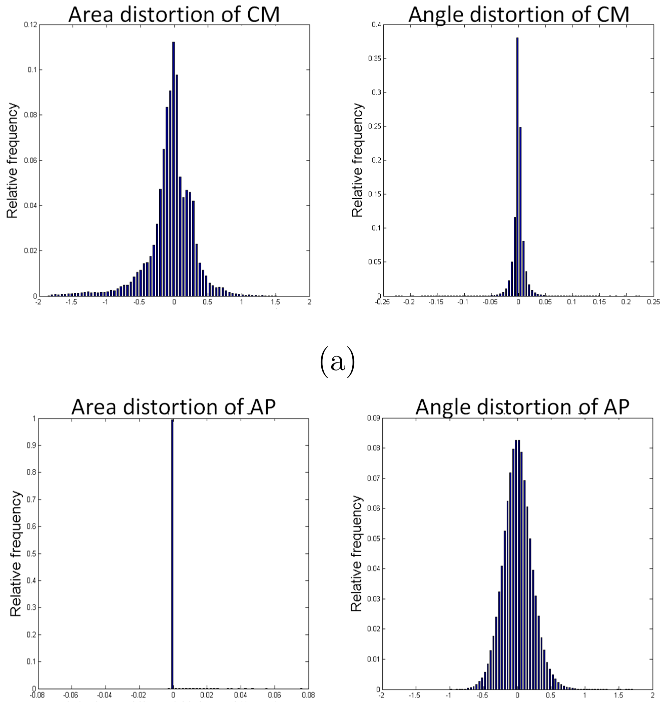
Table 4.1: Computing time for all experimental cases using the presented framework. N: the number of vertices, F: the number of faces, AP: area-preserving parametrization (bijective mapping) and T: texture mapping time in ms.

Model	Texture	N	F	AP	T
Chest	Chest	1528	2999	97	42
Brain	MRI brain	14499	29662	1254	108
Colon	CT colon	12762	24953	1096	97
Sphere	Earth	3456	6846	265	63
Square	City	5252	10471	474	62
Sphere	Graph	3456	6846	265	71
Gaussian	Graph	10201	20006	812	45

no efficient way to generate a similar result using either geometry methods (e.g., conformal magnifier [5]) or deformation methods. Taking a close look at Figure 4.9d, the conformal magnifier fails to flexibly control magnification ratios in both focus and context regions. It excessively enlarges the central core area, while compressing exterior nodes without any control. By comparison, the presented system can easily manipulate the size of area everywhere to generate a user preferred view with appropriate node distributions.

4.4.3 Implementation

In order to support fast visual display, the presented framework combines both CPU and GPU for computing optimization, parameterization, texture mapping and volume rendering using C++ with OpenGL library. The algorithm and solution of the optimal transport map is easy to implement robustly to have interactive-rate computation for all experimental cases, overcoming a major limitation of the OMT problem - computation inefficiency. By comparison with other optimization algorithms, such as [75] and [125], comparative experiments show that the presented system provides a significant speedup, empirically at least 3-5 times faster (more comparison technique analysis details are listed in Section 2). For the application of large volumetric data (e.g., brain dataset with size $256^2 \times 142$), the system can easily obtain full resolution results in an interactive-rate using various surface models with texture mapping or volume rendering. To further increase the flexibility of system control, the presented framework embeds mesh editing tools, allowing users to interactively choose a tradeoff between quality and computing speed.



(a)

(b)

Figure 4.10: Comparison histograms of mapping distortions using the lion head model of Figure 4.1. (a) Conformal mapping (CM) and (b) the area-preservation mapping (AP). Left column: area distortion. Right column: angle distortion. By comparison, the presented framework generates good mapping results with accurate area preservation and small angle distortions.

All experiments have been carried out on an Intel Core2Duo 2.2GHz laptop with 4GB memory and Windows 7 as the operating system. Generally, the cost linearly depends on the property of the surface models (the vertex and face counts) and the number of iterations needed for the desired accuracy. Table 4.1 presents detailed performance of the presented method, which shows that it is significantly fast and suitable for real-time/interactive operations, even for large datasets with high resolution requirements. In theory, finer surface models increase in resolution to support finer rendering but having longer computation time. Fortunately, the combination of saliency map and hierarchical mesh design (high resolution in the ROI, while low resolution in the context) allows the presented framework to work accurately and effectively.

4.4.4 Quantitative Analysis for Area-Preservation Mapping

The main challenge of a good area-preservation mapping is yielding strict equality of area elements between the original surface and the flattened result at its final state. Both the area distortion and the quasi-conformal distortion per face over the mesh are examined as the quantitative analysis. Define γ_{max} and γ_{min} as the larger and smaller eigenvalues of the Jacobian of the affine transformation that maps the domain triangles to the original surface, normalized with the hypothesis that the total area of the surface equals that of the domain. Then, the area distortion metric Υ is computed as $\Upsilon = log(\gamma_{max}\gamma_{min})$, while the quasi-conformal distortion metric Λ is computed using $\Lambda = log(\frac{\gamma_{max}}{\gamma_{min}})$. In both cases, zero indicates that there is no distortion and a larger value means a larger distortion.

Figure 4.10 shows the histograms of area distortion and quasi-conformal (angle) distortion for the conformal mapping [135] and the area-preserving approach, using the lion head surface model (Figure 4.1). By comparison, as shown in Figure 4.10, the presented system can generate great area-preservation results with extremely small area distortions (error $< 2\%$) even for the complicated surface model (in theory, more complicated models typically cause larger area distortion errors). Therefore, the presented framework generates accurately mapping and rendering results.

Chapter 5

Conclusions and Future Work

This chapter first briefly summarizes and reviews the presented techniques and their advantages for the volumetric F+C visualization. Then, it outlines the possible future related research directions and topics.

5.1 Summary

In conclusion, the presented techniques have the following advantages: apply a more effective selection for the focus regions, and improve ways for the 3D exploration and navigation of the complicated volume data, which overcome a number of limitations and drawbacks originating from using traditional methods for F+C visualization.

In particular, the first presented system highlights the volumetric region or object of interest by designing high dimensional transfer functions using parallel coordinates and dimension reduction techniques. Then, new focus+context design extends geometric models serving as lenses with a smooth transition between the focus and context regions and optimizes the local shape preservation everywhere; or accurately and efficiently preserves the size of area after mapping, using the optimal mass transport technique based on the Monge-Brenier theory.

Furthermore, all the F+C related techniques are capable of using multiple foci, i.e., magnifying multiple focus regions or volumes with the arbitrary shape, which maximally facilitate the effective use of available screen space. Meanwhile, the presented techniques, algorithms, and data structures are especially designed for the combination of CPU and GPU implementations to have the optimal computing speed. The flexibility of pipelines enables the efficient implementation of the visualization techniques suitable for volumetric data, especially for complicated medical datasets. Based on the analysis of existing related and previous work in the area of transfer function design and F+C visualization, this thesis proposal has presented the following four related technological components with specified contributions and applications for the volumetric F+C visualization. With the support of experimental results and comparisons within various task scenarios, all proposed framework and system show that, served as novel volumetric visualization techniques, they have great applications in the visualization and volume graphics research areas.

Highlighting Techniques - Transfer Function Design. In order to highlight the focused region or object of interest, high dimensional transfer function design is essential because the multivariate classification and its visualization could reduce the complexity of datasets and provide a vital connection between the dataset and the analyst. Thus, a parallel coordinates based transfer function and a local linear embedding technique based dimension reduction scheme are used to construct high dimensional transfer functions. A simple but effective interface is provided for the user to interactively design the high dimensional transfer function to facilitate the discovery of the best classification scheme for complicated datasets in the parallel coordinates. Meanwhile, the dimension reduction method can simplify the high dimensional design problems, although it may cause some information loss.

Conformal Mapping based F+C Visualization. The conformal magnifier, a conformal mapping based non-linear spatial distortion magnifier, serves as a novel geometry based F+C visualization technique to overcome the limitation of the screen real estate. The conformal magnifier focuses on the angle distortion minimization and visual continuity, producing optimally visual results: magnify the ROI with minimal local angle distortion and contain a continuous transition region. Moreover, different from traditional lenses designed in the spatial domain, the conformal magnifier is defined in the parameter domain, which can easily extend into various graphics or visualization frameworks, including the route, map, surface model and 3D volumetric data.

Area-Preservation based F+C Visualization. Different from conformal magnifier with optimal angle preservation, a computationally efficient numerical scheme is proposed to achieve the area-preservation mapping, served as an alternative F+C visualization technique. The presented framework, using the optimal mass transport method, is implemented in a novel manner and used for various visualization and graphic applications. The entire system is built based on geometric parameterization techniques, where an area-preserving map is generated with the minimal area distortion. The use of parameterization is compatible with various volume applications, easily and efficiently supporting the handling of various complicated data [138]. With the combination of CPU and GPU, the area-preservation system becomes practical for the use on large 3D datasets in terms of both speed and accuracy.

5.2 Future Work

The presented techniques form a basis for development of more advanced visualization concepts and applications for the volumetric F+C visualization. There are several available directions for future work including accurate transfer function design for feature enhance, new optimization algorithm for feature oriented F+C visualization, formal user studies for the analysis and evaluation, and hardware accelerated algorithms for real-time display and operations.

5.2.1 Highlighting Techniques

One unavoidable drawback to using multi-dimensional transfer functions is the increased memory consumption needed to store all the transfer function variables at each voxel sample point. Future work can expand the dataset size by using parallel hardware for computing and rendering, such as the effective scheme of assigning opacity, and the dynamic rendering via hierarchical fuzzy clustering. Meanwhile, the automatic transfer function design based on visual clustering algorithms [139–141] is also a promising direction.

5.2.2 Volumetric Focus+Context Visualization Techniques

Aim to improving the volumetric F+C visualization approaches, the following research directions are discussed, including the energy optimization, surface mesh smoothing, hardware acceleration and formal user evaluations and studies.

Smooth Surface Model Design. For the geometric model based techniques, a major limitation is how to generate smooth and grid-unified mesh models, which is a key factor to produce F+C visualizations with the minimal distortion (for both angle and area distortions) and the smooth transition. Therefore, some specified filters or blending algorithms, inspired by [142] are required in the future to further smooth the transition region of each model to obtain the high quality results.

Hardware Accelerated Algorithms. The adaptation of the existing implementations to the ongoing development of graphics accelerators is promising to overcome the required preprocessing steps and enable the processing of complicated large data. The major challenges comprise new theory and algorithm development by exploiting GPU features as well as the integration into existing visualization and rendering pipelines. The future work includes the following components:

- With the development of the graphic hardware, how to further accelerate the implementation of various non-trivial volumetric deformations that respect the material boundaries (e.g., use the as-rigid-as-possible deformation for bones, while using the as-similar-as-possible deformation for organs).

- The computational speed of conformal mapping, although not slow, still needs to be improved for interactive magnification operations. Further accelerate the calculation of conformal mapping and to incorporate new rendering methods for the high performance and resolution using GPU.

- Inspired by Sandhu et al. [143] and Wang et al. [144], the area-preservation mapping framework can be further extended to volumetric mapping, which will significantly increase the computing time. Luckily, due to the application of Voronoi diagram and its dual Delaunay triangulation for the mapping algorithm, inspired by Rong et al. [145], GPU can be easily used to reach real-time performance.

User Evaluation and Studies. Although some of the presented techniques are included in software system used and tested by users, an important aspect to focus in future work represents the conduction of qualitatively and quantitatively formal user studies to further improve the visualizations techniques. The feedback collection from professional users targeting to specific applications and systems, would significantly improve the usability of the presented F+C techniques.

Bibliography

[1] M. Sheelagh T. Carpendale, David J. Cowperthwaite, and F. David Fracchia. Extending distortion viewing from 2D to 3D. *IEEE Comput. Graph. Appl.*, 17(4):42–51, 1997.

[2] Yu-Shuen Wang, Tong-Yee Lee, and Chiew-Lan Tai. Focus+Context visualization with distortion minimization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1731–1738, 2008.

[3] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, (4):219–227, 1985.

[4] AT&T graph library. *http://www.graphdrawing.org*, 2009.

[5] Xin Zhao, Wei Zeng, Xianfeng David Gu, Arie E. Kaufman, Wei Xu, and Klaus Mueller. Conformal magnifier: A focus+context technique with local shape preservation. *IEEE Transactions on Visualization and Computer Graphics*, 18(11):1928–1941, 2012. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.70.

[6] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, 2006.

[7] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21 (4):163–169, August 1987.

[8] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *SIGGRAPH*, pages 65–74, 1988.

[9] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[10] B. Lichtenbelt, R. Crane, and S. Naqvi. *Introduction to Volume Rendering.* Hewlett-Packard Professional Books, Prentice Hall, 1998.

[11] Lukas Mroz, Helwig Hauser, and Eduard Gröller. Interactive high-quality maximum intensity projection. *Computer Graphics Forum*, 19 (3):341–350, 2000.

[12] N. Max. Optical models for direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[13] Lee Westover. Footprint evaluation for volume rendering. *SIGGRAPH*, 24:367–376, Sept. 1990.

[14] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Proc. of ACM SIGGRAPH*, pages 451–458, July 1994.

[15] Marc Levoy. Efficient ray tracing of volume data. *ACM Trans. on Graphics*, 9(3):245–261, July 1990.

[16] Mikhail Smelyanskiy, David Holmes, Jatin Chhugani, Alan Larson, Douglas M. Carmean, Dennis Hanson, Pradeep Dubey, Kurt Augustine, Daehyun Kim, Alan Kyker, Victor W. Lee, Anthony D. Nguyen, Larry Seiler, and Richard Robb. Mapping high-fidelity volume rendering for medical imaging to CPU, GPU and many-core architectures. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1563–1570, November/December 2009.

[17] Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler. The VolumePro real-time ray-casting system. *SIGGRAPH*, pages 251–260, 1999.

[18] Hanspeter Pfister and Arie Kaufman. Cube-4 - A scalable architecture for real-time volume rendering. *Proc. of Symposium on Volume Visualization*, pages 47–54, 1996.

[19] K. Engel, M. Hadwiger, J.M. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. AK Peters, 2006.

[20] Simon Stegmaier, Magnus Strengert, Thomas Klein, and Thomas Ertl. A simple and flexible volume rendering framework for graphics-hardware-based raycasting. *Proc. Volume Graphics*, pages 187–195, 2005.

[21] CUDA. http://www.nvidia.com/object/cuda_home_new.html.

[22] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. *Volume Visualization*, pages 79–86, 1998.

[23] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moler. Curvature-based transfer functions for direct volume rendering: Methods and applications. *IEEE Visualization*, pages 513–520, 2003.

[24] S. Tenginakai, J. Lee, and R. Machiraju. Salient isosurface detection with model-independent statistical signatures. *IEEE Visualization*, pages 231–238, 2001.

[25] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. on Visualization and Computer Graphics.*, 8(3):270–285, 2002.

[26] S. Roettger, B.M. M, and M. Stamminger. Spatialized transfer functions. *Eurographics IEEE VGTC Symposium on Visualization*, pages 271–278, 2005.

[27] Jesus J Caban and Penny Rheingans. Texture-based transfer functions for direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1364–1371, 2008.

[28] R. Maciejewski, I. Woo, W Chen, and D.S. Ebert. Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics.*, 15 (6):1473–1480, 2009.

[29] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search techniques. *IEEE Visualization*, pages 227–234, 1996.

[30] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. ”Design galleries”: A general approach to setting parameters for computer graphics and animation. *SIGGRAPH*, pages 389–400, 1997.

[31] Fan-Yin Tzeng, Eric B Lum, and Kwan-Liu Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Trans. on Visualization and Computer Graphics.*, 11(3):273–284, 2005.

[32] Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, 22:453–462, 2003.

[33] James Slack and Tamara Munzner. Composite rectilinear deformation for stretch and squish navigation. *IEEE Transactions on Visualization and Computer Graphics*, 12:901–908, 2006. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2006.127.

[34] Jean Daniel Fekete and Catherine Plaisant. Interactive information visualization of a million items. *IEEE Symposium on Information Visualization*, pages 117–124, 2002.

[35] T. Alan Keahey. Getting along: Composition of visualization paradigms. *IEEE Symposium on Information Visualization*, pages 37–40, 2001.

[36] Emden Gansner, Yehuda Koren, and Stephen North. Topological fisheye views for visualizing large graphs. *IEEE Symposium on Information Visualization*, pages 175–182, 2004. ISSN 1522-404X. doi: http://doi.ieeecomputersociety.org/10.1109/INFVIS.2004.66.

[37] T. Munzner. H3: laying out large directed graphs in 3D hyperbolic space. *IEEE Symposium on Information Visualization*, pages 2–8, 1997. doi: http://doi.ieeecomputersociety.org/10.1109/INFVIS.1997.636718.

[38] T. Tenev and R. Rao. Managing multiple focal levels in table lens. *IEEE Symposium on Information Visualization*, pages 59–66, 1997. doi: http://doi.ieeecomputersociety.org/10.1109/INFVIS.1997.636787.

[39] M. Sheelagh T. Carpendale, M. Sheelagh, T. Carpendale, David J. Cowperthwaite, and F. David Fracchia. Multi-scale viewing. *SIGGRAPH*, pages 149–152, 1996.

[40] Masashi Toyoda and Etsuya Shibayama. Hyper mochi sheet: a predictive focusing interface for navigating and editing nested networks through a multi-focus distortion-oriented view. *Conference on Human Factors in Computing Systems*, pages 504–511, 1999. doi: http://doi.acm.org/10.1145/302979.303145.

[41] Arno Formella and J Keller. Generalized fisheye views of graphs. *Proceedings Graph Drawing, Lecture Notes in Computer Science, LNCS 1027*, pages 242–253, 1995.

[42] T. Alan Keahey and Edward L. Robertson. Nonlinear magnification fields. *IEEE Symposium on Information Visualization*, pages 51–58, 1997.

[43] T. Alan Keahey. The generalized detail-in-context problem. *IEEE Symposium on Information Visualization*, pages 44–51, 1998.

[44] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

[45] X. Zhao, B. Li, L. Wang, and A. Kaufman. Focus+context volumetric visualization using 3d texture-guided moving least squares. *Proceedings of the Computer Graphics International*, 2011.

[46] B. Li, X. Zhao, and H. Qin. Four dimensional geometry lens: A novel volumetric magnification approach. *Computer Graphics Forum*, page preprint, 2013.

[47] Kang Shi, Pourang Irani, and Ben Li. An evaluation of content browsing techniques for hierarchical space-filling visualizations. *IEEE Symposium on Information Visualization*, pages 11–18, 2005. doi: http://dx.doi.org/10.1109/INFOVIS.2005.4.

[48] Jing Yang, Matthew O. Ward, Elke A. Rundensteiner, and Anilkumar Patro. Interring: a visual interface for navigating and manipulating hierarchies. *IEEE Symposium on Information Visualization*, 2(1):16–30, 2003. ISSN 1473-8716. doi: http://dx.doi.org/10.1057/palgrave.ivs.9500035.

[49] John Stasko and Eugene Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *IEEE Symposium on Information Visualization*, pages 57–64, 2000. ISSN 1522-404X. doi: http://doi.ieeecomputersociety.org/10.1109/INFVIS.2000.885091.

[50] T. Alan Keahey and Edward L. Robertson. Techniques for non-linear magnification transformations. *IEEE Symposium on Information Visualization*, pages 38–45, 1996.

[51] John Lamping, Ramana Rao, and Peter Pirolli. A Focus+Context technique based on hyperbolic geometry for visualizing large hierarchies. *SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408, 1995. doi: http://doi.acm.org/10.1145/223904.223956.

[52] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. Derose. Toolglass and magic lenses: The see-through interface. *Computer Graphics*, pages 73–80, 1993.

[53] M. Cohen and K. Brodlie. Focus and context for volume visualization. *Theory and Practice of Computer Graphics*, pages 32–39, 2004.

[54] M. Ikits and C.D. Hansen. A focus and context interface for interactive volume rendering. *http://www.cs.utah.edu/ ikits*, 2004.

[55] Eric LaMar, Bernd Hamann, and Kenneth I. Joy. A magnification lens for interactive volume visualization. *Pacific Graphics*, pages 223–232, 2001.

[56] Lujin Wang, Ye Zhao, Klaus Mueller, and Arie Kaufman. The magic volume lens: An interactive Focus+Context technique for volume rendering. *IEEE Visualization*, pages 367–374, 2005.

[57] Ivan Viola, Armin Kanitsar, and Meister Eduard Groller. Importance-driven volume rendering. *IEEE Visualization*, pages 139–145, 2004.

[58] M. Chen, D. Silver, A. S. Winter, V. Singh, and N. Cornea. Spatial transfer functions: a unified approach to specifying deformation in volume modeling and animation. *Eurographics/IEEE VGTC Workshop on Volume Graphics*, pages 35–44, 2003.

[59] Kurzion Yair and Yagel Roni. Space deformation using ray deflectors. *Rendering Techniques*, pages 21–30, 1995.

[60] Jens Kruger, Jens Schneider, and Rudiger Westermann. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12:941–948, 2006.

[61] Jianlong Zhou, Manfred Hinz, and Klaus D.Tonnies. Focal region-guided feature-based volume rendering. *International Symposium on 3D Data Processing Visualization and Transmission*, pages 87–90, 2002. doi: http://doi.ieeecomputersociety.org/10.1109/TDPVT.2002.1024047.

[62] Cheng-Kai Chen, Russell Thomason, and Kwan-Liu Ma. Intelligent focus+context volume visualization. *International Conference on Intelligent Systems Design and Applications*, 1:368–374, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/ISDA.2008.232.

[63] M Chen, C Correa, S Islam, M. W Jones, P.-Y Shen, D Silver, S. J Walton, and P. J. Willis. Manipulating, deforming and animating sampled object representations. *Computer Graphics Forum*, 26(4):824–852, 2007.

[64] Carlos D. Correa, Deborah Silver, and Min Chen. Discontinuous displacement mapping for volume graphics. *Fifth Eurographics / IEEE VGTC Workshop on Volume Graphics*, pages 9–16, 2006.

[65] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. *IEEE Visualization*, pages 401–408, 2003. doi: http://dx.doi.org/10.1109/VISUAL.2003.1250400.

[66] Carlos D. Correa, Deborah Silver, and Min Chen. Volume deformation via scattered data interpolation. *Sixth Eurographics / IEEE VGTC Workshop on Volume Graphics*, pages 91–98, 2007.

[67] Carlos D. Correa, Deborah Silver, and Min Chen. Constrained illustrative volume deformation. *Comput. Graph.*, 34(4):370–377, 2010.

[68] Carlos Correa, Debora Silver, and Mi Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13:1320–1327, 2007.

[69] Rudiger Westermann and Christof Rezk-Salama. Real-time volume deformations. *Computer Graphics Forum*, 20(3):443–451, 2001. doi: http://doi.acm.org/10.1145/1141911.1141920.

[70] Yu-Shuen Wang, Chaoli Wang, Tong-Yee Lee, and Kwan-Liu. Ma. Feature-preserving volume data reduction and focus+context visualization. *IEEE Transactions on Visualization and Computer Graphics*, preprint online, 2011.

[71] Xianfeng David Gu and Shing-Tung Yau. *Computational Conformal Geometry*. International Press of Boston, 2008.

[72] Boris Springborn, Peter Schröder, and Ulrich Pinkall. Conformal equivalence of triangle meshes. *ACM Trans. on Graphics*, 27(3):77:1–77:11, August 2008.

[73] Wei Zeng, Joseph Marino, Krishna Chaitanya Gurijala, Xianfeng Gu, and Arie Kaufman. Supine and prone colon registration using quasi-conformal mapping. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1348–1357, Nov. 2010.

[74] X. Zhao, Z. Su, X. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo. Area preservation using optimal mass transport. *IEEE Transactions on Visualization and Computer Graphics*, page preprint, 2013.

[75] A. Dominitz and A. Tannenbaum. Texture mapping via optimal mass transport. *IEEE Transactions on Visualization and Computer Graphics*, 16(13):419–432, 2010.

[76] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3D visualization of colon CT images. *IEEE Trans. Med. Imag.*, 19(7):665–670, 2000.

[77] Wei Zeng, Joseph Marino, Arie Kaufman, and Xianfeng Gu. Volumetric colon wall flattening using harmonic differentials. *Computers and Graphics*, 35:726–732, 2011.

[78] Heinrich W. Guggenheimer. *Differential Geometry*. Dover Publications, 1977.

[79] R. Schoen and S.-T. Yau. Lectures on harmonic maps. *International Press.*, pages 187–241, 1997.

[80] M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface Ricci flow. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.

[81] Miao Jin, Junho Kim, and Xianfeng David Gu. Discrete surface Ricci flow: Theory and applications. *Proc. of IMA Conf. on the Mathematics of Surfaces*, pages 209–232, 2007.

[82] Yong-Liang Yang, Junho Kim, Feng Luo, Shi-Min Hu, and Xianfeng Gu. Optimal surface parameterization using inverse curvature map. *IEEE Trans. on Visualization and Computer Graphics*, 14(5):1054–1066, Sept. 2008.

[83] Wei Zeng, Dimitris Samaras, and Xianfeng David Gu. Ricci flow for 3D shape analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(4):662–677, April 2010.

[84] Yong-Liang Yang, Ren Guo, Feng Luo, Shi-Min Hu, and Xianfeng Gu. Generalized discrete Ricci flow. *Computer Graphics Forum*, 28(7):2005–2014, Oct. 2009.

[85] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.

[86] G. Monge. Mémoire sur la théorie des déblais et de remblais. *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la meme année,*, pages 666–704, 1781.

[87] L. Zhu, S. Haker, and A. Tannenbaum. Flattening maps for the visualization of multibranched vessels. *IEEE Trans. Med. Imag.*, 24(2): 191–198, 2005.

[88] L. Zhu, Y. Yang, S. Haker, and A. Tannenbaum. An image morphing technique based on optimal mass preserving mapping. *IEEE Trans. Med. Imag.*, 16(6):1481–1495, 2007.

[89] T.Rehman, E.Haber, G.Pryor, J.Melonakos, and A.Tannenbaum. 3D nonrigid registration via optimal mass transport on the GPU. *Medical Image Analysis*, 13:931–40, 2009.

[90] L. Kantorovich. On a problem of Monge. *Uspekhi Mat. Nauk.*, 3:225–226, 1948.

[91] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Comm. Pure Appl. Math.*, 44(4):375–417, 1991.

[92] Xianfeng Gu, Feng Luo, Jian Sun, and S.-T. Yau. Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampere equations. *arXiv:1302.5472 [math.GT]*, pages 1–13, 2013.

[93] I. Takanashi, E. B. Lum, and S. Muraki. Ispace: Interactive volume data classification techniques using independent component analysis. *The Pacific Conference on Computer Graphics and Applications*, pages 366–374, 2002.

[94] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics.*, 12(5):1021–1028, 2006.

[95] Francisco de Moura Pinto and Carla M. D. S. Freitas. Design of multi-dimensional transfer functions using dimensional reduction. *Eurographics IEEE-VGTC Symposium on Visualization*, pages 131–138, 2007.

[96] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science.*, 290(5500):2323–2326, 2000.

[97] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science.*, 220:671–680, 1983.

[98] R Haralick, K Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3: 610–621, 1973.

[99] Matej Novotny and Helwig Hauser. Outlier-preserving Focus+ Context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics.*, 12(5):893–900, 2006. ISSN 1077-2626. doi: http://dx.doi.org/10.1109/TVCG.2006.170.

[100] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure within clustered parallel coordinates displays. *IEEE Symposium on Information Visualization*, pages 125–132, 2005.

[101] Mark Gahegan, Masa Takatsuka, and Xiping Dai. An exploration into the definition, operationalization and evaluation of geographical categories. *International Conference on GeoComputation*, pages 66–75, 2001.

[102] The homepage of xmdvtool–multivariate data visualization tool. http://www.davis.wpi.edu/ xmdv/index.html.

[103] Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. *IEEE Symposium on Information Visualization*, pages 89–96, 2004.

[104] Alfred Inselberg. *Parallel coordinates: visual multidimensional geometry and its applications*. Springer, New York, NY, USA, 2009.

[105] I.T. Jolliffe. *Principal Component Analysis.* Springer Verlag, New York, 1989.

[106] T Cox and M. Cox. *Multidimensional Scaling.* Chapman Hall, London, 1994.

[107] Wei Hong, Feng Qiu, and Arie kaufman. A pipeline for computer aided polyp detection. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):861–868, 2006. ISSN 1077-2626. doi: http://dx.doi.org/10.1109/TVCG.2006.112.

[108] Hershel M. Farkas and Irwin Kra. *Riemann Surfaces.* Springer, 2004.

[109] Bennett Chow. The ricci flow on the 2-sphere. *J. Differential Geom.*, 33 (2):325–334, 1991.

[110] Richard S. Hamilton. Three manifolds with positive Ricci curvature. *Journal of Differential Geometry*, 17:255–306, 1982.

[111] W. P. Thurston. Geometry and topology of three-manifolds. *Lecture notes at Princeton university*, 1980.

[112] B. Chow and F. Luo. Combinatorial ricci flows on surfaces. *Journal Differential Geometry*, 63(1):97–129, 2003.

[113] X. Gu, W. Zeng, F. Luo, and S.-T. Yau. Numerical computation of surface conformal mappings. *Computational Methods and Functional Theory (CMFT)*, 11(2):747–787, 2011.

[114] M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface ricci flow. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.

[115] Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):548–585, 1995.

[116] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5: 349–359, 1999.

[117] M. Spindler, M. Bubke, T. Germer, and T. Strothotte. Camera textures. *SIGGRAPH*, pages 295–302, 2006.

[118] Huamin Qu, Haomian Wang, Weiwei Cui, Yingcai Wu, and Ming-Yuen Chan. Focus+context route zooming and information overlay in 3D urban environments. *IEEE Transactions on Visualization and Computer Graphics*, 15:1547–1554, 2009.

[119] Matthias Trapp, Tassilo Glander, Henrik Buchholz, and Jurgen Dollner. 3D generalization lenses for interactive focus+context visualization of virtual city models. *International Conference on Information Visualisation*, pages 356–361, 2008.

[120] Hartmut Ziegler and Daniel A. Keim. Copernicus: Context-preserving engine for route navigation with interactive user-modifiable scaling. *Comput. Graph. Forum*, 27(3):927–934, 2008. URL http://dblp.uni-trier.de/db/journals/cgf/cgf27.html#ZieglerK08.

[121] Pushpak Karnick, David Cline, Stefan Jeschke, Anshuman Razdan, and Peter Wonka. Route visualization using detail lenses. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):235–247,

2010. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2009.65.

[122] Robert Spence and Mark Apperley. Bifocal display. http://www.interaction-design.org/encyclopedia/bifocal-display.html.

[123] S. Halier, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3D visualization of colon CT images. *IEEE Transactions on Medical Imaging*, 19(7):665–670, 2000.

[124] X. Gu, Yalin Wang, T.F. Chan, P.M. Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(8):949–958, 2004. ISSN 0278-0062.

[125] G. Zou, J. Hu, X. Gu, and J. Hua. Area-preserving surface flattening using Lie advection. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 14:335–342, 2011.

[126] Nicolas Bonnotte. From Knothe's rearrangement to Brenier's optimal transport map. *arXiv:1205.1099 [math.OC]*, pages 1–29, 2012.

[127] S. Rachev and L. Ruschendorf. *Mass Transportation Problems*, volume I-II. Springer, New York, 1998.

[128] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich. Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph. (SIGGRAPH Asia)*, 30:158:1–158:12, 2011.

[129] F. de Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun. Blue noise through optimal transport. *ACM Trans. Graph. (SIGGRAPH Asia)*, 31: 1–10, 2012.

[130] F. de Goes adn D.Cohen-Steiner, P. Alliez, and M. Desbrun. An optimal transport approach to robust reconstruction and simplification of 2D shapes. *Eurographics Sym. on Geometry Processing*, 30(5):1593–1602, 2011.

[131] Q. Merigot. A multiscale approach to optimal transport. *Comput. Graph. Forum*, 30(5):1583–1592.

[132] Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *International Journal on Computer Vision*, 60:225–240, 2004.

[133] A. D. Alexandrov. Convex polyhedra. *Springer Monographs in Mathematics. Springer-Verlag, Berlin*, 2005.

[134] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal of Computing*, 16(1):78–96, 1987.

[135] Y. Wang, J. Shi, X. Yin, X. Gu, T. F. Chan, S. T. Yau, A. W. Toga, and P. M. Thompson. Brain surface conformal parameterization with the Ricci flow. *IEEE Trans Med Imaging*, 31(2):251–264, Feb 2012.

[136] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[137] G. Zou, J. Hu, X. Gu, and J. Hua. Authalic parameterization of general surfaces using Lie advection. *IEEE Transactions on Visualization Computer Graphics*, 17(12):2005–2014, 2011.

[138] B. Li and H. Qin. Feature-aware reconstruction of volume data via trivariate splines. *Pacific Graphics*, pages 49–54, 2011.

[139] A. O. Artero, M. C. F. De oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. *IEEE Symp. on Information Visualization*, pages 81–88, 2004.

[140] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo. Splatting the lines in parallel coordinates. *Computer Graphics Forum*, 28(3):759–766, 2009.

[141] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.

[142] B. Li, X. Li, K. Wang, and H. Qin. Surface mesh to volumetric spline conversion with generalized polycube. *IEEE Transactions on Visualization and Computer Graphics*, page preprint, 2013.

[143] Romeil Sandhu, Ayelet Dominitz, Yi Gao, and Allen Tannenbaum. Volumetric mapping of genus zero objects via mass preservation. *CoRR*, abs/1205.1225:1–12, 2012.

[144] K. Wang, X. Li, B. Li, H. Xu, and H. Qing. Restricted trivariate polycube splines for volumetric data modeling. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):703–716, 2012.

[145] Guodong Rong, Yang Liu, Wenping Wang, Xiaotian Yin, XianFeng Gu, and Xiaohu Guo. GPU-assisted computation of centroidal voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):345–356, 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2010.53.