Published in final edited form as:

Proc IEEE Virtual Real Conf. 2011 ; : 143–150. doi:10.1109/VR.2011.5759453.

Conformal Visualization for Partially-Immersive Platforms

Kaloian Petkov, **Charilaos Papadopoulos**, **Min Zhang**, **Arie E. Kaufman [Fellow, IEEE]**, and **Xianfeng Gu [Member, IEEE]**

Computer Science Department, Stony Brook University, Stony Brook, NY, USA

Abstract

Current immersive VR systems such as the CAVE provide an effective platform for the immersive exploration of large 3D data. A major limitation is that in most cases at least one display surface is missing due to space, access or cost constraints. This partially-immersive visualization results in a substantial loss of visual information that may be acceptable for some applications, however it becomes a major obstacle for critical tasks, such as the analysis of medical data. We propose a conformal deformation rendering pipeline for the visualization of datasets on partially-immersive platforms. The angle-preserving conformal mapping approach is used to map the 360°3D view volume to arbitrary display configurations. It has the desirable property of preserving shapes under distortion, which is important for identifying features, especially in medical data. The conformal mapping is used for rasterization, realtime raytracing and volume rendering of the datasets. Since the technique is applied during the rendering, we can construct stereoscopic images from the data, which is usually not true for image-based distortion approaches. We demonstrate the stereo conformal mapping rendering pipeline in the partially-immersive 5-wall Immersive Cabin (IC) for virtual colonoscopy and architectural review.

Keywords

Immersive Visualization; Virtual Reality; Conformal Mapping

1 Introduction

A number of visualization technologies have been developed for the immersive exploration of large-scale complex data. Prime examples are the CAVE [3] and Head-Mounted Displays (HMD) which provide a much larger fields of view in a virtual environment and also use stereo pairs of images to improve the perception of spatial relationships in the data. These technologies have found applications in a number of fields including entertainment, engineering and design, and medical imaging. While HMDs allow for arbitrary views in the virtual world, they are usually bulky, wired and easily lead to eye fatigue. At the same time, CAVEs provide a much more natural visualization without the need for a virtual avatar and allow for augmented reality applications. Building a fully enclosed CAVE however remains

kpetkov@cs.sunysb.edu
cpapadopoulo@cs.sunysb.edu
mzhang@cs.sunysb.edu
gu@cs.sunysb.edu
ari@cs.sunysb.edu

a difficult task. Although such installations exist [5], they present an engineering challenge in terms of cost, facility access and head and gesture tracking. The disadvantage of partially-immersive environments is that important visual information may be lost. While many applications may tolerate one or more missing projection screens, this loss of information becomes a critical issue in the exploration of medical data.

We have developed a visualization approach that utilizes conformal mapping to distort scene geometries or viewing rays depending on the rendering method. As a result the full virtual environment can be displayed on a partially-immersive visualization platform, for example a 5-wall CAVE. In mathematics, the conformal map is an angle preserving function that describes a mapping between two Riemannian surfaces [26]. Intuitively, it allows us to map the geometry of a 6-wall CAVE to an arbitrary configuration of display surfaces that is topologically equivalent to a disc, such as the 5-wall Immersive Cabin (IC) or a non-planar arrangement of workstation displays. In practice, the mapping is performed directly in the space of view directions, which can be mapped onto a sphere. The main advantage of using a conformal map to define the distortion is the guarantee that shapes will be preserved locally even though distances will not be. This is particularly beneficial for exploration of medical data, such as in Virtual Colonoscopy (VC) where polyps that may be cancerous are detected by the radiologist based on their shape.

The following are the specific contributions of our work:

- Conformal mapping is used to obtain a shape-preserving distortion that can map a 360°field of view to an arbitrary display configuration. We demonstrate that the shape of polyps is preserved under the distortion for Immersive Virtual Colonoscopy and compare against non-conformal distortions for the visualization of architectural data.
- The conformal distortion is applied during rendering for rasterization, direct volume rendering and ray-tracing. As a result we can construct stereo image pairs for all three rendering modalities using a unified distortion map definition.
- The performance penalty for applying the distortion map is approximately 1% and is insignificant compared to the overall rendering times. Furthermore, the required pre-computation time for the generation of the distortion cube-map is minimal.
- Our approach is verified for the visualization of medical and architectural data in the partially-immersive 5-wall IC.

2 Related Work

Immersive visualization systems allow the user to explore data in novel ways that go beyond the standard 2*D* images on a workstation. These platforms provide a superior depiction of the information via a significantly wider field of view and enhanced depth and shape perception. The first such environment, the CAVE [3] offers an immersive experience using back-projected images on 3 walls and a floor. Other display arrangements have been proposed, including the 5-wall IC [20] and the 6-wall CAVE [5]. Compared to Head-Mounted Displays (HMD), these environment provide a more natural visualization and allow users to interact with visualization-augmented physical objects, following the

Author Manuscript
Author Manuscript
Author Manuscript

paradigm of "Spatially Augmented Reality". However, building fully-immersive facilities is expensive and introduces many challenges in terms of head tracking, sound systems and even air circulation. Our conformal visualization system uses the discrete Ricci flow to compute the Riemann mapping and maps the full 360°spherical field of view to partially-immersive platforms. The main advantage is that the local shape of the scene geometry and the features in volume data are preserved under the distortion. The Ricci flow method has been applied broadly in the graphics and visualization fields, including applications to optimal surface parameterization [30], shape analysis [9] and surface matching [12].

Virtual Colonoscopy (VC) has been established as a non-invasive alternative to traditional optical colonoscopy (OC) for cancer screening [6, 10]. A VC session involves the acquisition of computed tomography (CT) scans of the patient's abdomen and the extraction and visualization of the colon surface via segmentation and volume rendering techniques. Traditional VC visualizations on a single screen covers only about 91% of the colon surface after full navigation in both the antegrade and the retrograde directions [7] and the percentage is significantly lower for a single direction. Immersive Virtual Colonoscopy (IVC) takes advantage of the wider field of view and enhanced shape and depth perception through stereo rendering to allow a faster visualization of the entire colon surface in a single navigation pass. One shortcoming of existing partially-immersive configurations is that the missing projections may hide a significant amount of information. While this may be acceptable in certain applications, it becomes a critical issue for the exploration of medical data. Our conformal distortion allows the radiologist to examine the entire surface of the colon and ensures that the shape of any colon polyps, which is crucial for the detection of colon cancer, is preserved in the partially-immersive visualization.

Implementing the conformal distortion for real-time visualization requires the rendering of a non-linear, non-pinhole projection, which is not currently supported on the GPU. A number of image-based and geometry-based techniques have been proposed. Image-space approaches [29] suffer from linear sampling artifacts and provide no guarantees with regard to shape preservation on the distorted image, which is imperative for screening procedures like VC.

Focus and Context (F+C) techniques such as magic lenses [18, 28] are designed for a single projection surface and do not translate directly to immersive environments with multiple and possibly non-planar displays. Illustrative deformations for data exploration have been proposed [2] which could be used to warp data that lies around the CAVE volume, however they also do not provide shape preservation. The technique presented by Lorenz and Döllner [14] handles piecewise approximation of non-planar perspective projections on the GPU. Non-planar projections can be used to define a projection surface that "wraps" around the CAVE, including part of the ceiling and thus recovering the non-visualized part of the data. The technique can be applied in either image-space or geometry-space, however in the first case the sampling artifacts can impact visual quality significantly [27], while the geometry approach does not scale well with mesh density [14].

The geometry-based approach is not directly applicable to ray-tracing or direct volume rendering, which are frequently used for architectural visualization and medical imaging respectively. In contrast, we use a simple vertex-based distortion for well-tessellated geometry, similar to the approach by Spindler et al. [24], and viewing ray distortions for volume rendering of medical data and ray-tracing of non-uniformly tessellated architectural scenes. Moreover, a unified conformal transformation is defined for all three rendering pipelines. A number of GPU-based ray-tracing algorithms have been proposed that can achieve interactive frame-rates even for large geometric models [13, 17, 19, 31]. Our implementation is based on the NVidia OptiX engine [17] which is used as a full replacement for the OpenGL traverser in the scene graph.

3 Theoretical Background

According to Riemann mapping theorem, simply connected surfaces with a single boundary can be mapped onto the planar disk, such that the mapping is *angle-preserving*. Locally, the mapping is only a scaling and therefore it is also shape-preserving. Conformal mapping has advantages for our application, it maps the 6 walls to the 5 walls, and preserves local shapes. Our computational algorithm is based on the surface Ricci flow theorem [1].

3.1 Conformal Mapping

Let S_1 and S_2 be two surfaces with Riemannian metrics $\mathbf{g}_1$ and $\mathbf{g}_2$, and let $\varphi : (S_1,\mathbf{g}_1) \to (S_2,\mathbf{g}_2)$ be a homeomorphism between them. We say φ is *conformal*, if the pull back metric tensor induced by φ on the source differs from the original metric by a scalar,

$$\phi^*\mathbf{g}_2=e^{2\lambda}\mathbf{g}_1,$$

where $\lambda:S_1 \to \mathbb{R}$, is a function. The following Riemann mapping theorem plays a fundamental role in the current work:

Theorem 3.1 (Riemann Mapping) Suppose a surface S is simply connected with a Riemannian metric. There exists conformal mappings $\phi:S \to \mathbb{D}$, where D is the unit disk on the complex plane and all such mappings differ by a Möbius transformation.

A Möbius transformation is given by

$$z \to e^{i\theta}\frac{z-z_0}{1-\bar{z}_0 z},$$

where $\{\theta, z_0\}$ are constants.

In order to compute the mapping φ, one can compute the pull back metric first, which can be achieved by the surface Ricci flow.

3.2 Ricci Flow

A surface Ricci flow is the process to deform the Riemannian metric of the surface. The deformation is proportional to Gaussian curvatures so that the curvature evolves in a manner similar to heat diffusion. It has been considered a powerful tool for finding a Riemannian metric satisfying the prescribed Gaussian curvature in mathematics. Chow and Luo [1] laid down the theoretic foundation for the discrete Ricci flow on surfaces, and Jin et al. [8] developed the computational algorithm.

Let Σ be a triangular mesh, embedded in the three dimensional $\mathbb{R}^3$. A *discrete Riemannian metric* on a mesh Σ s a piecewise constant metric with cone singularities at the vertices. The edge lengths are sufficient to define a discrete Riemannian metric,

$$l{:}E \to \mathbb{R}^+, \quad (1)$$

as long as, for each face [v_i, vj, v_k], the edge lengths satisfy the triangle inequality: $l_{ij} + l_{jk} > l_{ki}$.

For simplicity, we use e_i to denote the edge against the vertex v_i, namely e_i = [vj, v_k], and l_i is the edge length of e_i. The cosine laws are given by:

$$l_k^2 = l_i^2 + l_j^2 - 2 l_i l_j \cos\theta_k \quad (2)$$

The *discrete Gaussian curvature* K_i on a vertex v_i ε Σ can be computed as the angle deficit,

$$K_i = \begin{cases} 2\pi - \Sigma_{[v_i,v_j,v_k]\in\Sigma}\theta_i^{jk}, & v_i \notin \partial\Sigma \\ \pi - \Sigma_{[v_i,v_j,v_k]\in\Sigma}\theta_i^{jk}, & v_i \in \partial\Sigma \end{cases} \quad (3)$$

where θ_i^{jk} represents the corner angle attached to vertex v_i in the face [v_i, v_j, v_k], and ∂ Σ represents the boundary of the mesh.

The circle packing metric was introduced [25] and [26] to approximate the conformal deformation of metrics. Fig. 2 illustrates the circle packing metric. Each vertex v_i has a circle whose radius is r_i. On each edge e_{ij}, an intersection angle φ_{ij} is defined by two circles of v_i and v_j, which intersect with or are tangent to each other.

Let $u{:}V \to \mathbb{R}$ be the *discrete conformal factor*, which measures the local area distortion $u_i = \log\gamma_i$ for each vertex.Then, the discrete Ricci flow is defined as follows:

$$\frac{du_i(t)}{dt} = \left(\bar{K}_i - K_i\right), \quad (4)$$

Discrete Ricci flow can be formulated in the variational setting, namely, it is a negative gradient flow of some special energy form:

$$f(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} \sum_{i=1}^{n} \left(\bar{K}_i - K_i\right) du_i, \quad (5)$$

where $\mathbf{u_0}$ is an arbitrary initial metric. The integration above is well-defined, and it is called the Ricci energy. The discrete Ricci flow is the negative gradient flow of the discrete Ricci energy. The discrete metric which induces $\bar{k}$ is the minimizer of the energy.

Computing the desired metric with prescribed curvature $\bar{K}$ is equivalent to minimizing the discrete Ricci energy, which is strictly convex (namely, its Hessian is positive definite). The global minimum uniquely exists, corresponding to the metric $\bar{u}$, which induces $\bar{k}$. The discrete Ricci flow converges to this global minimum [1].

4 Computational Algorithm

4.1 Discrete Ricci flow

Suppose Σ is a triangle mesh embedded in $\mathbb{R}^3$. We associate each vertex v_i with a circle (v_i, γ_i) where γ_i equals the minimal length of any edge in the immediate neighborhood of v_i. Then we compute the intersection angle φ_{ij}, such that the circle packing metric is as close to the induced Euclidean metric as possible.

We compute the curvature at each vertex K_i and adjust the conformal factor u_i in proportion to the difference between the target curvature and the current curvature $\bar{K}_i - K_i$. Then we update the metric, recompute the curvature, and repeat this procedure until the difference between the target curvature and the current curvature is less than the given threshold. More details can be found in the work of [8].

4.2 Riemann Mapping

Fig. 3 illustrates our algorithm for computing the Riemann mapping. Given a simply connected triangular mesh Σ, we first punch a hole on the mesh. The hole in Fig. 3(a) is intentionally large for illustration purposes, while in practice, it can be very tiny. The boundary of the mesh is given by

$$\partial\Sigma = \gamma_1 - \gamma_2.$$

We run the Ricci flow algorithm as described in Alg. 1 by setting all vertex target curvatures to be zero, including the boundary vertices. Then we find two boundary vertices, $v_1 \varepsilon \gamma_1$ and $v_2 \varepsilon \gamma_2$, and a shortest path τ connecting them. We slice Σ along τ to get a simply connected mesh $\bar{\Sigma}$. By using the metric obtained by the Ricci flow, we flatten $\bar{\Sigma}$ isometrically onto the plane, where γ_1 and γ_2 are mapped to parallel straight lines. By a planar rigid motion, γ_1 and γ_2 are mapped to vertical lines, as shown in Fig. 3(b).

We use an exponential map to map the planar strip to the annulus in Fig. 3(c) given by

$$z \longrightarrow e^{2\pi \frac{z}{h}},$$

where h is the length of γ_1 on the plane. In practice, the center hole consists of a single triangle. Then, we fill the hole to get the mapping for the entire surface. The mapping is conformal, which can be verified by checkerboard texture mapping, as shown in Fig. 3(d). All the right corners of the checkerboard are well-preserved.

4.3 Mapping from 5 Walls to 6 Walls in the IC

Fig. 5 demonstrates the mapping between the 5-wall CAVE and the 6-walls CAVE. We cut the closed cube (the 6-wall CAVE) along a cross slit at the top and map the cube to a sphere by the *direction map*

$$p \to \frac{p-c}{|p-c|},$$

where p is a point on the cube and c is the center of the cube. Then the 6-walls CAVE is mapped to a sphere with slits, shown in Fig. 5(c). We then conformally map the sphere with the slits to the unit disk, which is illustrated in Fig. 5(d). Similarly, we map the 5-wall CAVE to an open sphere using the same direction map, then map the sphere to the unit disk using Ricci flow.

The Möbius transformation is used to align the two disk images. We choose three corresponding boundary vertices on both the 5-wall CAVE and the 6-wall CAVE, and use special Möbius transformations to map them to 1, i, −1 on the unit circle, which aligns the corresponding markers.

Suppose $\{p, q, r\}$ are three markers on the unit circle and

$$\eta_1(z) = \frac{z-p}{z-q}\frac{r-q}{r-p}$$

maps them to $\{0, \infty, -1\}$. Let

$$\eta_2(z) = \frac{1+i}{2}\frac{z-1}{z-i},$$

then $\eta_2^{-1} \circ \eta_1$ is the desired Möbius transformation, which maps $\{p, q, r\}$ to $\{1, i, -1\}$. Fig. 5(b) and Fig. 5(d) show the result after the alignment.

Fig. 4 illustrates the intuition for the application of the discrete Ricci flow. We consider the view directions in the IC mapped onto a unit sphere and two curves. The first is an infinitesimal cut at the top of the sphere, which corresponds to the center of the missing projection wall (Fig. 4(b)). The second is the union of 4 curves that correspond to the top edges of the side screens in the IC projected onto the sphere (Fig. 4(d)) and they define the visibility boundary $\partial\Sigma$ for the central IC position. Although we can compute a conformal mapping between these curves, the resulting distortion would be very large near the top edges. Instead we cut the sphere along the curves between the cusps in the visibility boundary and the top curve on the sphere. The four edges shown in Fig. 4(b) are then mapped to $\partial\Sigma$, reducing the perceived distortion in the image. Using this mapping, the view directions are then projected onto the geometry of the 5-wall IC and encoded into a cube map (Fig. 4(e)).

5 Implementation Details

The 6-to-5 wall conformal map is applicable to a number of rendering modalities, however in practice we focus on mesh rendering, single-pass raycasting for volume data and real-time raytracing on the GPU. Although the conformal map is defined in the spherical space of view directions in the IC, we encode the distortion map in a cube-texture, which is natively supported on the GPU and can be sampled in both vertex shader and pixel shaders. The cube-map is generated in a preprocessing step, where for each pixel we reconstruct a view direction and sample the analytical representation of the conformal map. The normalized directions are stored in the $RGB32F$ format which provides 32-bit floating point precision and it is the highest precision format that is supported natively on the GPU. The resolution of the cube-map is dependent on the distortion, and therefore on the display configuration of the target visualization platform. We have determined experimentally that a resolution of 1024^2 per cube-map face is sufficient for a mapping to th 5-wall IC and even 512^2 provides acceptable results if GPU memory is limited.

We first apply the conformal distortion to a mesh-based rendering pipeline. Intuitively, every vertex in the scene is distorted so that triangles that are projected on the top screen are instead projected on the 4 side screens. This transformation is defined by the following relationship:

$$\mathbf{p}'_{\mathbf{w}} = WCi * \left(T_{5\to 6}\left(WCit * norm\left(\mathbf{p}_{\mathbf{w}} - WCi * [0,0,0,1]^T\right)\right)|WC * \mathbf{p}_{\mathbf{w}}|\right) \quad (6)$$

In this equation, $\mathbf{p}_{\mathbf{w}}$ is the vertex position in world-space and $\mathbf{p}'_{\mathbf{w}}$ is the transformed position also in world-space. WC is the world-space to CAVE-space transformation matrix and the WCi and $WCit$ are its inverse and inverse-transpose respectively. $T_{5\to 6}$ is the conformal mapping from a 5-wall to a 6-wall configuration, which is the inverse of the mapping discussed in Sec. 3. The computation is performed in a custom vertex-shader that is bound to the state-set of every primitive in the scene graph. In our framework, the head node for the visualization cluster emits its camera information to all the rendering clients. The view matrix V associated with that camera is the world-space to CAVE-space transformation matrix. Each visualization node then computes a final view-matrix based on the target projection surface (e.g., front, left, etc.).

The distortion is applied in a similar fashion for volume rendering. Our framework integrates volume rendering tightly into the scene graph and therefore we render out the volume-space positions of the front and back faces of the volume bounding box. One possible approach for incorporating the distortion map is to tessellate the bounding box and apply the geometry-based approach described in this section. However, our target application is the exploration of the virtual colonoscopy data, in which case the starting positions of the rays are often defined on the camera's near clipping plane. It is more accurate to directly distort the ray directions per-pixel by transforming the positions on the near clipping plane and the back faces of the bounding volume to world-space and applying the following transformation:

$$\mathbf{p}_{\mathbf{w}}^{'}=WCi*\left(T_{6\rightarrow 5}\left(WCit*norm\left(\mathbf{p}_{\mathbf{w}}-WCi*[0,0,0,1]^{T}\right)\right)|WC*\mathbf{p}_{\mathbf{w}}|\right) \quad (7)$$

Note that this transformation is very similar to Eq. 6. The difference is that since we are transforming the view directions directly, the 6-to-5 wall conformal map has to be used. Our framework varies the step size for ray integration based on the voxel projection size on the image plane, similar to the differential sampling approach introduced by Knoll et al. [11]. We extend the idea further by taking into account the distortion profile of the conformal map along the IC *y*-axis and lowering the sampling density in areas of larger compressive distortions. Elaborate acceleration structures for empty space skipping can further improve the rendering speed, however for the particular task of navigating the colon dataset, such techniques are not necessary since most of the time the user is close to the colon wall. Instead we employ early ray termination [15] to stop the ray integration after finding an iso-surface or accumulating density above a user-specified threshold. We also use stochastic jittering of the starting positions of the rays [21] to reduce sampling artifacts and a binary search for the iso-surface [23] to improve the reconstruction quality. During interaction with the system, we generally sample the volume data using a trilinear filter with a reduced step size in order to improve the response time. However, for a static camera position we render the images using higher order filtering on the GPU [22, 4] for both the density data and the on-the-fly gradient computations.

A major limitation of the geometry distortion approach is that existing shaders in the scene need to be modified to compute Eq. 6. This is not an issue for the virtual colonoscopy mesh dataset which contains a single shader, however large architectural scenes may contain dozens of materials. We integrate a ray-tracing renderer with our scene graph based on the NVidia OptiX engine [17]. OptiX accelerates ray-tracing on the GPU by defining ray-generation, ray-scene intersections and shading programs in the CUDA [16] language which access traditional acceleration structures that are also stored on the GPU. Its features are comparable to other real-time ray-tracing approaches [19, 31]. The main advantage is that the distortion is applied at the ray-generation level, which is separate from the scene-graph and therefore much simpler to override. The distortion computation is also simplified:

$$\mathbf{d}_{\mathbf{w}}^{'}=WCt*\left(T_{6\rightarrow 5}\left(WCit*\mathbf{d}_{\mathbf{w}}\right)\right) \quad (8)$$

Similarly to the volume rendering approach, we transform the ray direction in world-space $\mathbf{d}_{\mathbf{w}}$ to CAVE-space and fetch a new ray direction from the conformal map. OptiX does not support cube textures natively, therefore indexing the conformal map is implemented in the ray-generation shader. In addition to simplifying the application of the conformal map, the raytracing introduces a number of effects that are particularly suitable for architectural visualization, such as accurate lighting, dynamic shadows, reflections and refractions.

6 Experimental Results

Our conformal mapping is used for the real-time rendering of stereo image pairs in the IC and the distortion is computed and applied in our existing rendering framework. For the evaluation of the approach, all experiments are first run on a single workstation with 2 Intel Xeon E5430 processors and an NVidia Quadro FX 5800 GPU with a 1024^2 viewport. The production cluster environment consists of 5 workstations with an Intel Xeon processor and dual NVidia Quadro FX 4600 GPUs with 10 1400 × 1500 viewports.

We first evaluate the geometry-based conformal distortion on a surface model from a virtual colonoscopy dataset. The triangular mesh is uniformly well-tessellated and as a result distorting the individual vertices as described in Sec. 5 yield acceptable rendering results. Fig. 6 demonstrates the conformal distortion near a known polyp. The two images on the left illustrate the original front and top projections in the IC. Since our platform is only partially-immersive, the polyp would not be visible. After the distortion, the polyp is displayed on the front screen in the IC and its shape is preserved under the conformal mapping. Other sections of the viewing space would similarly be transformed to the side and the back screens. The mesh contains 4M vertices and the performance decreases by approximately 1% when the distortion is applied on modern GPUs with unified shader units. The performance hit is significantly larger on older hardware with dedicated vertex shaders. Also, such GPUs have very limited texturing capability in the vertex processing pipeline, and as a result vertex shaders may even be emulated on the CPU leading to non-interactive rendering.

We also implement the conformal distortion for a single-pass ray-casting volume rendering. Fig. 7 illustrates the result of conformally mapping the top projection onto the front screen. The pair of images on the left are the original IC-correct perspective projection for the front and top screens. The image on the right is the result of the conformal distortion described in Sec. 5 for the front IC screen. Note that unlike in image-based approaches, the final result is rendered directly, and the original images are included only for demonstration purposes. We evaluate the volume rendering on a 512 × 512 × 451 16-bit CT scan of a patient's abdomen after digital cleansing of the colon. The rendering uses the Cook-Torrance lighting model with distance-based light attenuation and subtle specular highlights to assist in the perception of small shapes. A suspicious area is visible on the top screen, which would normally be missed in the IC due to the angle of the approach and the missing top screen. After the conformal distortion, this area is projected onto the front IC screen and the shape of the bumps on the colon wall is preserved. Although conformal mapping is not distance-preserving and sizes are distorted, we provide tools to measure the distance in voxel-space. Since the conformal distortion map is computed in a pre-processing step, the cost of the initial transform of the ray direction is negligible compared to the ray integration. In practice the performance drop is measurable but it is less than 1% on average.

Ray-tracing is developed as a traverser object for the scene-graph and therefore it is a drop-in replacement for the OpenGL renderer. As described in 5, the application of the distortion map is simplified in terms of both computation and integration with the scene. Fig. 8 compares the projection images of the front and top screens (left pairs of images) to the conformally distorted front view. As before, the original images are only use for illustrative purposes and the final result is rendered directly. The performance of the renderer scales with the size of the viewport as opposed to scene complexity in the rasterization pipeline. As a result the ray-tracing performance for the colon mesh dataset is comparable to the OpenGL rendering. Although that dataset contains geometry with uniformly-high mesh density, the same is not true for the architectural scene. The ray-tracing approach makes no assumptions regarding the tessellation of the data and provides high-quality, conformally distorted results, even in areas of low polygonal density. While the performance for the architectural scene is lower than the rasterization pipeline, it allows for interactive frame-rates and special effects such as glass reflections and shadows. Again, the cost of applying the distortion maps is negligible compared to the overall rendering and its effect on the frame-rate is below 1%.

In Fig. 9 we compare the conformal distortion against a non-conformal mapping of the viewing directions in the ray-tracer. Here, the view directions are linearly scaled along the longitude lines on the sphere. Although this approach also allows missing data from the top projection to be shown on the side screens, Fig. 9(b) shows that the spatial relationships are lost, particularly in areas of highly-detailed geometry. In contrast, the distances in Fig. 9(b) are distorted, however the local spatial structures are preserved.

Our conformal distortion is evaluated in the 5-wall IC with stereo rendering (Fig. 1) for both medical and architectural visualization of surface and volume data. As expected, the conformal distortion produces slightly unnatural images in the areas of larger compressive distortions. However, the benefit is that depth perception is maintained in the distorted areas and the user can examine the entire scene even in a partially-immersive visualization platform. During the navigation of the virtual colonoscopy scene, the comments from the radiologist indicated that this type of distortion is acceptable for immersive exploration of the colon since the shape of potential polyps is preserved while allowing the entire colon surface to be examined. Fig. 10 illustrates the conformal distortion of the IC projections for a snapshot of the Immersive Virtual Colonoscopy. Fig. 11 shows the effect of the distortion on the left, front and right IC projections for the architecture scene.

7 Conclusions And Future Work

We have presented a distortion-based visualization technique for partially immersive visualization platforms that allow the user to visually explore the full virtual environment. Our approach is based on conformal mapping, which is an angle-preserving function that guarantees the consistency of local shapes under deformation.

We have demonstrated our approach with rasterization, volume rendering and GPU-based ray-tracing for medical and architectural visualization. We avoid many of the challenges related to image-based approaches by rendering the final results directly, which allows for the generation of accurate stereo image pairs. We primarily describe an application to Immersive Virtual Colonoscopy in the 5-wall IC in which missing visual information for the top screen is rendered onto the front, back and side screens. In addition, the technique is not limited to medical visualization and can be applied to arbitrary display configurations that are topologically equivalent to a disc.

Currently, the conformal distortion is defined for a specific cut in the space of viewing directions. While this provides a useful visualization for medical and architectural scenes, the cut is not optimal for off-center viewing positions in the CAVE. One limitation is that the conformal mapping is currently performed on the CPU. An NVIDIA CUDA-based implementation may provide adequate performance for the dynamic generation of the distortion map. In addition, the generated maps can be re-sampled in a pixel shader in order to control the distortion so that areas of interest are better preserved.

References

[1]. Chow B. The ricci flow on the 2-sphere. Journal of Differential Geometry. 1991; 2(33):325–334.

[2]. Correa C, Silver D, Chen M. Illustrative deformation for data exploration. Transactions on Visualization and Computer Graphics. 2007; 13(6):1320–1327. [PubMed: 17968080]

[3]. Cruz-Neira C, Sandin DJ, DeFanti TA, Kenyon RV, Hart JC. The cave: Audio visual experience automatic virtual environment. Communications of the ACM. 1992; 35(6):64–72.

[4]. Hadwiger M, Sigg C, Scharsach H, Bhler K, Gross M. Real-time ray-casting and advanced shading of discrete isosurfaces. Computer Graphics Forum. 2005; 24(3):303–312.

[5]. Hogue, A.; Robinson, M.; Jenkin, MR.; Allison, RS. A vision-based head tracking system for fully immersive displays. Proceedings of the workshop on Virtual environments; 2003. p. 179-187.

[6]. Hong, L.; Muraki, S.; Kaufman, A.; Bartz, D.; He, T. Virtual voyage: interactive navigation in the human colon. SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques; 1997. p. 27-34.

[7]. Hong W, Wang J, Qiu F, Kaufman A, Anderson J. Colonoscopy simulation. Proc. SPIE Medical Imaging. 2007:6511.

[8]. Jin M, Kim J, Luo F, Gu X. Discrete surface ricci flow. IEEE Transactions on Visualization and Computer Graphics. 2008; 14(5):1030–1043. [PubMed: 18599915]

[9]. Jin M, Zeng W, Luo F, Gu X. Computing *teichmüller* space. IEEE Transactions on Visualization and Computer Graphics. 2009; 15(3):504–517. [PubMed: 19282555]

[10]. Johnson CD, Dachman AH. Ct colonography: The next colon screening examination? Radiology. 2000; 216(2):331–341. [PubMed: 10924550]

[11]. Knoll A, Hijazi Y, Westerteiger R, Schott M, Hansen C, Hagen H. Volume ray casting with peak finding and differential sampling. IEEE Transactions on Visualization and Computer Graphics. 2009; 15(6):1571–1578. [PubMed: 19834235]

[12]. Li X, Gu X, Qin H. Surface matching using consistent pants decomposition. ACM Solid and Physical Modeling Symposium. 2008:125–136.

[13]. Liu B, Wei L-Y, Yang X, Xu Y-Q, Guo B. Nonlinear beam tracing on a gpu. Technical Report MSR-TR-2007-34, Mcrosoft Research. 2007

[14]. Lorenz, H.; Döllner, J. Real-time piecewise perspective projections. GRAPP 2009 - International Conference on Computer Graphics Theory and Applications; 2009. p. 147-155.

[15]. Neubauer A, Wolfsberger S, Forster M-T, Mroz L, Wegenkittl R, Buhler K. Steps - an application for simulation of transsphenoidal endonasal pituitary surgery. IEEE Visualization. 2004:513–520.

[16]. NVidia. Compute Unified Device Architecture Programming Guide (Version 2.3). 2010http://developer.nvidia.com/object/gpucomputing.html

[17]. NVidia. NVidia OptiX raytracing engine (Version 2.2b). 2010http://developer.nvidia.com/object/optix-beta.html

[18]. Pietriga E, Bau O, Appert C. Representation-independent in-place magnification with sigma lenses. Transactions on Visualization and Computer Graphics. 2010; 16(3):455–467. [PubMed: 20224140]

[19]. Popov S, Gnther J, Seidel H-P, Slusallek P. Stackless kd-tree traversal for high performance gpu ray tracing. Computer Graphics Forum. 2007; 26(3):415–424.

[20]. Qiu, F.; Zhang, B.; Petkov, K.; Chong, L.; Kaufman, A.; Mueller, K.; Gu, XD. Enclosed five-wall immersive cabin. ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing; 2008. p. 891-900.

[21]. Scharsach H, Hadwiger M, Neubauer A, Wolfsberger S, Bhler K. Perspective isosurface and direct volume rendering for virtual endoscopy applications. IEEE-VGTC Symposium on Visualization. 2006:315–322.

[22]. Sigg, C.; Hadwiger, M. Fast third-order texture filtering. In: Pharr, M.; Fernando, R., editors. GPU gems 2 : programming techniques for high-performance graphics and general-purpose computation. Addison-Wesley; 2004. p. 313-330.

[23]. Sigg, C.; Hadwiger, M.; Gross, M.; Bhler, K. Technical report. VRVis Research Center; ETH Zrich: 2004. Real-time high-quality rendering of isosurfaces.

[24]. Spindler, M.; Bubke, M.; Germer, T.; Strothotte, T. Camera textures. GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and South-east Asia; 2006. p. 295-302.

[25]. Stephenson, K. Introduction To Circle Packing. Cambridge University Press; 2005.

[26]. Thurston WP. Geometry and topology of three-manifolds. Princeton Lecture Notes. 1976

[27]. Trapp, M.; Dllner, J. Advances in Computer Graphics and Computer Vision (VISIGRAPP), Communications in Computer and Information Science (CCIS). Springer; 2008. Generalization of single-center projections using projection tile screens.

[28]. Wang L, Zhao Y, Mueller K, Kaufman A. The magic volume lens: An interactive focus + context technique for volume rendering. Proc. IEEE Visualization. 2005:367–374.

[29]. Yang Y, Chen JX, Beheshti M. Nonlinear perspective projections and magic lenses: 3d view deformation. Computer Graphics. 2005; 25(1):76–84.

[30]. Yang Y-L, Kim J, Luo F, Hu S-M, Gu X. Optimal surface parameterization using inverse curvature map. Transactions on Visualization and Computer Graphics. 2008; 14(5):1054–1066. [PubMed: 18599917]

[31]. Zlatuska M, Havran V. Ray tracing on a gpu with cuda – comparative study of three algorithms. WSCG 2010. 2010

(a) Immersive Virtual Colonoscopy in the IC

(b) Left, front and right projections after conformal distortion to the IC

Fig. 1.

Shape-preserving conformal distortion is used for Immersive Virtual Colonoscopy. The areas that would be projected onto the missing ceiling in the Immersive Cabin (IC) (a) are clearly visible on the front screen under the deformation (b).

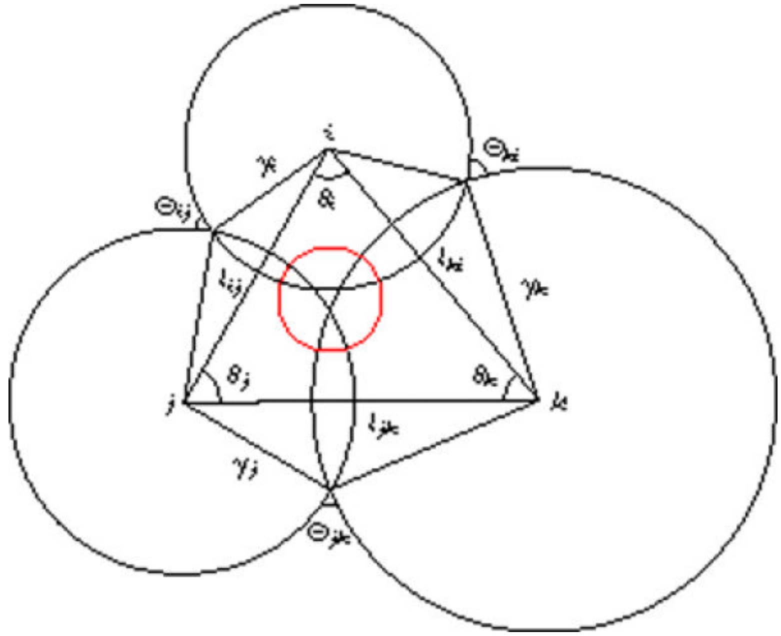


Fig. 2.
Circle Packing Metric.

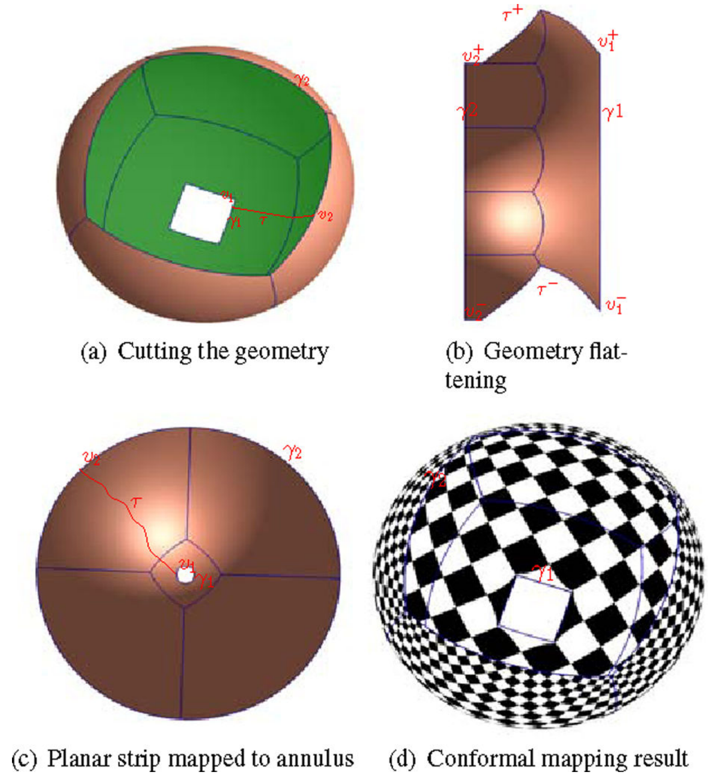


Fig. 3.

Riemann Mapping Algorithm.

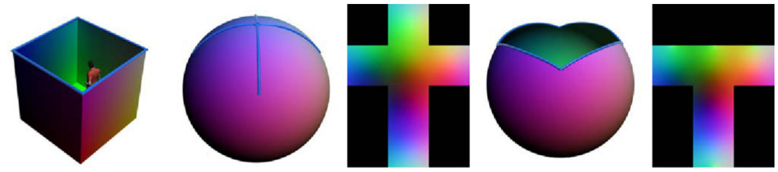
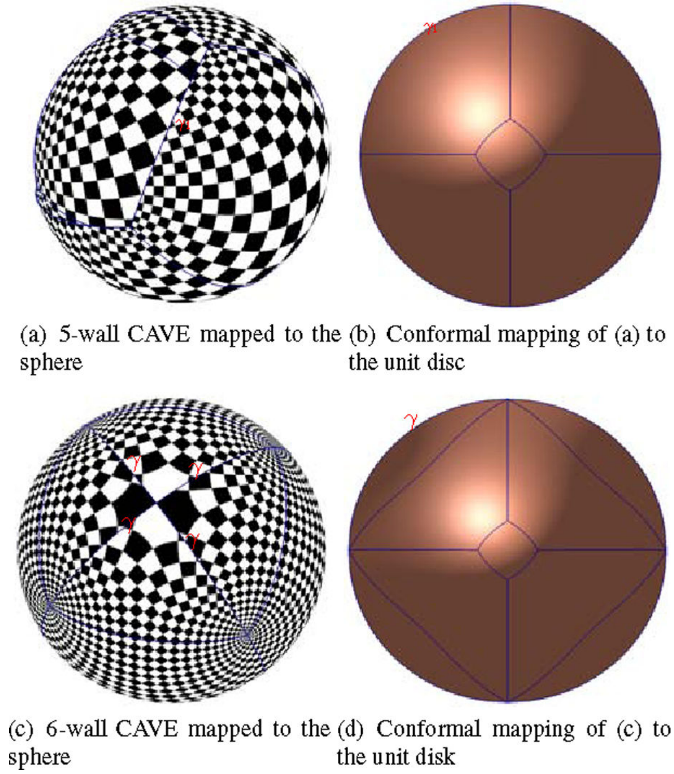


Fig. 4.
View directions in the IC are mapped to a sphere. A small cut near the top (b) is mapped to the visibility boundary on the sphere (d) and this mapping is used to compute the conformal distortion cube-map (e).



(a) 5-wall CAVE mapped to the sphere

(b) Conformal mapping of (a) to the unit disc

(c) 6-wall CAVE mapped to the sphere

(d) Conformal mapping of (c) to the unit disk

Fig. 5.

Algorithm for conformal mapping between a 5-wall CAVE and 6-wall CAVE.

Author Manuscript
Author Manuscript
Author Manuscript

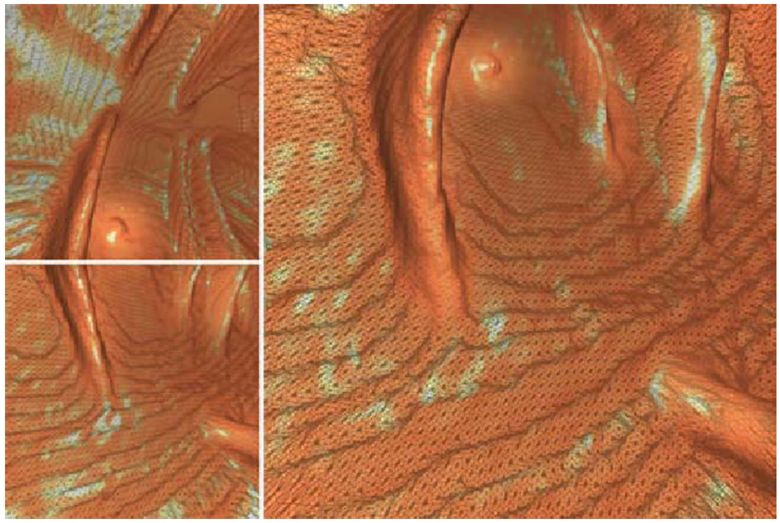


Fig. 6.
Visualizing the mesh model of a patient's colon during Virtual Colonoscopy. The left pairs of images show the front and top views, and the image on the right shows the front view after conformal distortion. The shape of the polyp is preserved under the conformal geometry deformation.

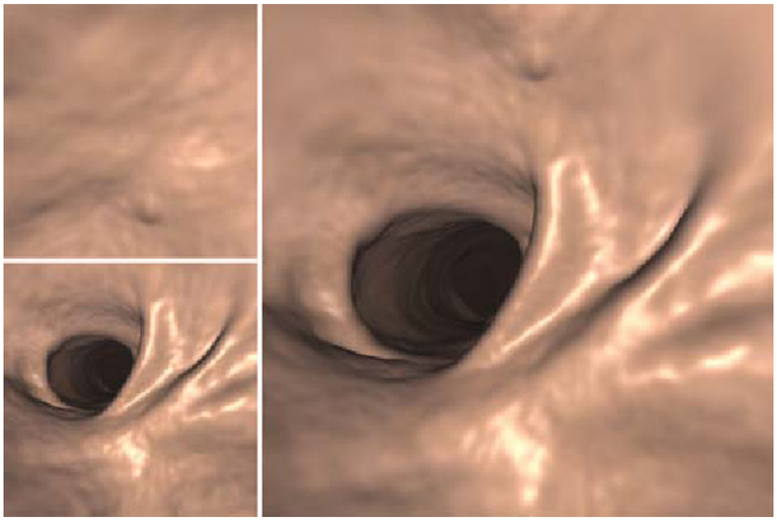


Fig. 7.

Direct visualization of the raw slices from a CT scan of a patient's abdomen. The left pair of images shows the front and top views, and the image on the right shows the front view after the conformal distortion. The shape of the suspicious area on the top view is preserved under the volumetric conformal distortion.

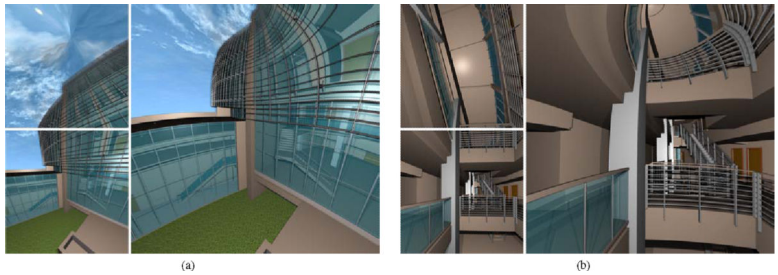


Fig. 8.

Snapshots of an architectural fly-through using conformal mapping. For both (a) and (b), the left pairs of images show the front and top views, and the image on the right shows the front view after conformal mapping. Approximately $\frac{1}{4}$ of the visual data on the missing top projection in the 5-wall configuration is presented on the front screen.

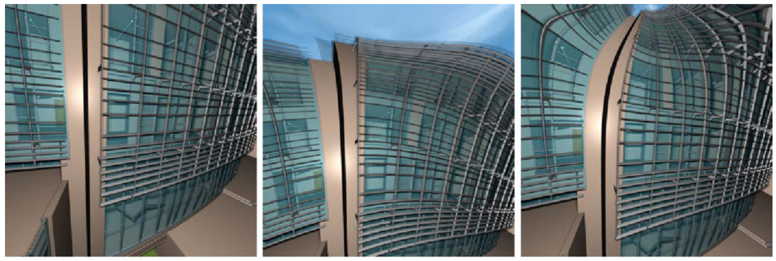
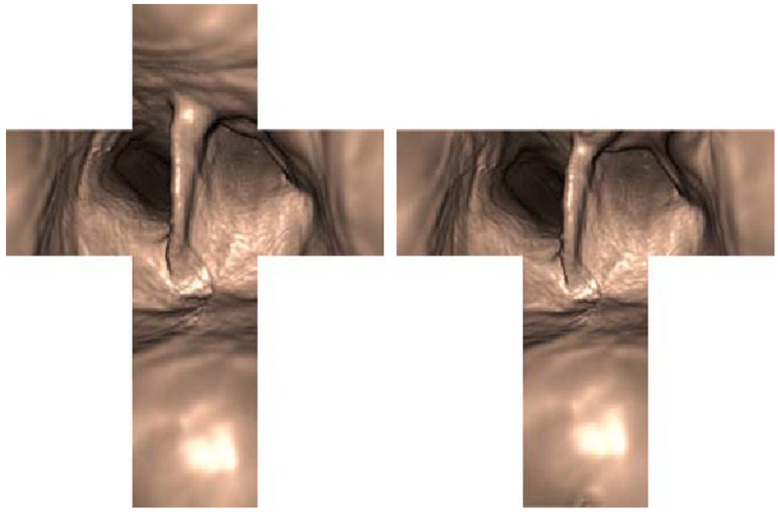


Fig. 9.
Comparison between (b) a non-conformal and the (c) conformal distortion. The conformal distortion preserves the local structure of the original image (a) around the building's sun screen, while visualizing the missing information from the top projection.

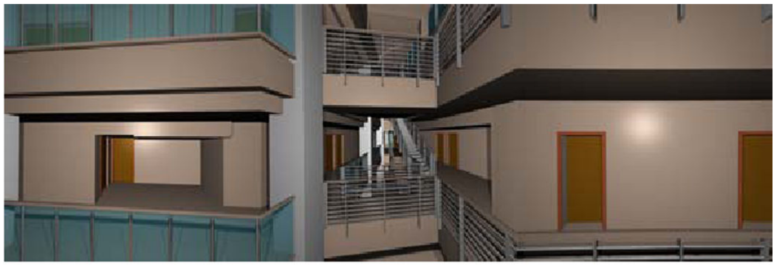


(a) Original IC projections

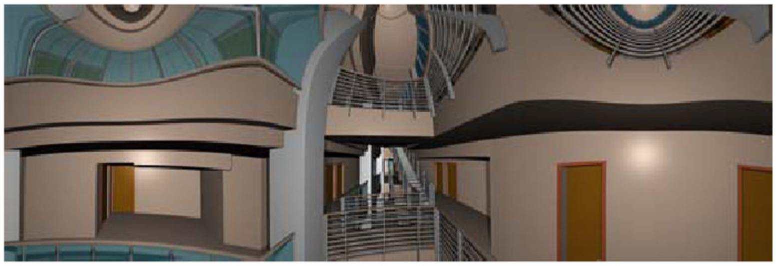
(b) Projections after conformal distortion to the 5-wall IC

Fig. 10.

The effect of the conformal distortion is show for all projections during the Immersive Virtual Colonoscopy. All visual information from (a) is presented in the 5-wall IC (b) with local shape preservation.



(a) Left, front and right projections for a 6-wall CAVE



(b) Left, front and right projections after conformal distortion to the 5-wall IC

Fig. 11.

The effect of the conformal distortion is show for the left, front and right projections during the architectural fly-through. Shapes in (a) are locally preserved in the global conformal distortion (b).

Algorithm 1

Discrete Ricci Flow

Require: Triangular Mesh Σ, Target curvature for each vertex $\bar{K}i$. Error threshold ε.

Ensure: Discrete metric (edge lengths) satisfying the target curvature.

u=[u_i],**v**=[v_i] for **u,v** ←0

while true **do**

Compute edge length l_{ij} for edge [v_i;v_j]: $l_{ij} = e^{u_i} + e^{u_j} + 2\cos\phi_{ij} e^{u_i + u_j}$

Compute the corner angle θ_i^{jk} in triangle [v_i;v_j;v_k]: $\theta_i^{jk} = \cos^{-1}\frac{l_{ij}^2 + l_{ki}^2 - l_{jk}^2}{2l_{ij}l_{ki}}$

Compute the curvature K_i at v_i:$K_i = \begin{cases} 2\pi - \Sigma_{[v_i,v_j,v_k] \in \Sigma}\theta_i^{jk}, & v_i \notin \partial\Sigma \\ \pi - \Sigma_{[v_i,v_j,v_k] \in \Sigma}\theta_i^{jk}, & v_i \notin \partial\Sigma \end{cases}$ $\mathbf{K} = [K_i]$

if max $|\bar{K}i - K_i| < \varepsilon$ **then**

return the discrete metric l_{ij}

end if

Update **u**: Compute the Hessian Matrix H,H_{ij}=$\frac{\partial K_i}{\partial u_j}$ $\mathbf{u} \leftarrow \mathbf{u} - H^{-1}(\bar{\mathbf{K}} - \mathbf{K})$

end while