Conformal Visualization for Partially-Immersive Platforms

Kaloian Petkov‡ Charilaos Papadopoulos† Min Zhang‡ Arie E. Kaufman§ Xianfeng Gu‡

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

ABSTRACT

Current 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, such as a ceiling or a back wall, 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 that need to utilize the users’ entire field of vision. We have developed a conformal rendering pipeline for the visualization of datasets on partially-immersive platforms. The angle-preserving conformal mapping approach is used to render the 360° view directly onto arbitrary display configurations. It has the desirable property of preserving shapes locally, which is important for identifying shape-based features in the data. Our conformal visualization technique is applicable to rasterization, volume rendering and real-time raytracing, and in contrast to image-based retargeting approaches, it constructs accurate, artifact-free stereoscopic images. We demonstrate our conformal visualization pipeline in the 5-sided CAVE for Immersive Virtual Colonoscopy, as well as architectural and abstract data exploration. Our user study shows that on the visual polyp detection task, conformal visualization leads to improved sensitivity at comparable examination times against the traditional rendering approach.


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 field of view into a virtual environment compared to traditional desktop systems, and also use stereoscopic pairs of images to improve the perception of spatial relationships. 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 Mixed Reality applications. Building a fully enclosed CAVE, however, remains a difficult task. Although such installations exist [5], they present an engineering challenge in terms of cost, facility access, as well as head and gesture tracking.

Immersion in the virtual data is a function of different factors, such as sensory perceptions, interaction techniques and realism of the visualization. We focus on visual immersion, defined by the field of view coverage afforded by the physical visualization platform, as well as the coverage of the virtual scene provided by the visualization software. The disadvantage of partially-immersive environments, such as CAVEs with at least one missing display surface, is that important visual information may be lost. While many applications may tolerate one or more missing projection screens, this partial loss of visual context adversely affects the general navigation capabilities of the user and becomes a critical issue in the exploration of medical data.

We have developed a visualization approach that utilizes conformal mapping to modify scene geometries or viewing rays at runtime. As a result, the full virtual environment can be displayed on a partially-immersive visualization platform, for example a 5-sided CAVE, without the artifacts typically associated with image-based retargeting approaches. In mathematics, the conformal map is an angle preserving function that describes a mapping between two Riemannian surfaces [25]. Intuitively, it allows us to map the geometry of a 6-sided CAVE to an arbitrary configuration of display surfaces that is topologically equivalent to a disk, such as a 5-sided CAVE or a non-planar arrangement of workstation displays. This mapping is then used to transform the viewing directions during rendering. The main advantage of using a conformal map to define the transformation is the guarantee that shapes will be preserved locally even though distances will not be. This is particularly beneficial for the exploration of medical data, such as in Virtual Colonoscopy (VC) where potentially cancerous polyps 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 transformation that can map a 360° field of view to an arbitrary display configuration. In medical visualization, we demonstrate that the shape of the polyps is preserved under the distortion for Immersive Virtual Colonoscopy.

- The conformal distortion is applied during rendering for rasterization, direct volume rendering and ray-tracing. As a result, we construct stereo image pairs for all three rendering modalities using a unified transformation definition.

- The performance penalty for applying the transformation is approximately 1% and requires minimal pre-computation.

- Our approach is applied to the visualization of medical, architectural and abstract data in a 5-sided CAVE. An informal user study shows an improvement in the task of visual polyp detection in phantom colonoscopy datasets.

2 RELATED WORK

Immersive visualization systems allow the user to explore data in novel ways that go beyond the standard 2D 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 front projection on the floor. Other display arrangements have been proposed, including the 5-sided Immersive Cabin (IC) [20] and the 6-sided 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 paradigm of Augmented Virtuality [15], or more generally, Mixed 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 [8] to compute the Riemann mapping and maps the full 360° spheic field of view to partially-immersive platforms. The main advantage is that the local shape of the projected scene geometry and the features in volume data are preserved under the distortion. The Ricci flow method has found a broad range of applications in the graphics and visualization fields, including optimal surface parameterization [30, 28], shape analysis [9] and surface matching [11].

Least square conformal maps (LSCM) is an alternative heuristic technique for computing a conformal mapping [14]. Unlike Ricci flow, which automatically produces seam-free global conformal parameterizations, LSCM and other computational approaches may require special algorithms or heuristic inputs when dealing with certain topologies [8]. The main disadvantage of LSCM is that it cannot control the boundary of the region. In our case, we require a mapping between the visibility boundary of the CA VE and the unit circle. Therefore, LSCM is not a viable technique for our application. Also, unlike the approach by Springborn et al. [23], the Ricci flow is a stable algorithm with a theoretical assurance of convergence.

Raskar et al. have used the LSCM to display the output of an ad-hoc cluster of projectors onto an arbitrary set of surfaces [21]. Our conformal visualization solves a different problem: given a set of display surfaces and a visualization system that produces the correct projections (e.g., partially-immersive CA VE or an arrangement of LCD displays), we want to compute a mapping between the full visibility sphere and the visibility area provided by the display system. The user is able to visualize the entire data at once on the target platform, while the properties of the mapping guarantee that shapes are locally preserved. Our application of conformal mapping is fundamentally different and requires the use of advanced rendering techniques beyond texture projection.

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 medical screening procedures such as VC.

Focus and Context (F+C) techniques such as magic lenses [18, 27] 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 CA VE volume, however they also do not provide shape preservation. The technique presented by Lorenz and Döllner [13] 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 CA VE, 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 [26], while the geometry approach does not scale well with mesh density [13].

Our conformal visualization is applicable to a number of visualization tasks, however in the user study of its effectiveness we focus primarily on medical visualization. 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 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. One shortcoming of existing partially-immersive configurations for Immersive Virtual Colonoscopy (IVC) 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 approach allows the radiologist to examine the entire surface of the colon in a single pass and ensures that the shape of any colon polyps, which is crucial for the detection of colon cancer, is preserved in the conformal visualization.

3 Theoretical Background

According to the 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. This is advantageous in our rendering technique as it maps the viewing directions from the 6 original projections in the fully-enclosed CA VE to the 5-sided CA VE while preserving the 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 $g_1$ and $g_2$, and let $\phi: (S_1, g_1) \to (S_2, g_2)$ be a homeomorphism between them. We say $\phi$ is conformal, if the pull back metric tensor induced by $\phi$ on the source differs from the original metric by a scalar:

$$\phi^*(g_2) = e^{2\lambda} g_1,$$

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 exist conformal mappings $\phi: S \to D$, where $D$ is the unit disk on the complex plane and all such mappings differ by a Möbius transformation.

A Möbius transformation of a point $z$ on the complex plane is given by

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

where $\theta$ and $z_0 \in \mathbb{C}$ are constants.

In order to compute the mapping $\phi$, 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 used 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. In mathematics, it is a powerful tool for finding a Riemannian metric satisfying the prescribed Gaussian curvature. Chow and Luo have described the theoretic foundation for the discrete Ricci flow on surfaces [1], and Jin et al. have developed an efficient computational algorithm [8].

Let $\Sigma = (V, E, F)$ be a triangular mesh embedded in $\mathbb{R}^3$, where $V$, $E$ and $F$ are respectively the sets of vertices, edges, and faces. A discrete Riemannian metric on $\Sigma$ is a piecewise constant metric with cone singularities at the vertices. The edge lengths are sufficient to define a discrete Riemannian metric,

$$I: E \to \mathbb{R}^+,$$

as long as, for each face $f_{ijk} = [v_i, v_j, v_k]$, $f_{ijk} \in F$, 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 $e_i = [v_j, v_k]$, and $l_i$ is the edge length of $e_i$ in triangle $f_{ijk}$. The cosine laws are given by:

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

The discrete Gaussian curvature $K_i$ on a vertex $v_i \in \Sigma$ can be computed as the angle deficit,

$$K_i = \frac{2\pi - \sum_{f_{ijk} \in F} \theta_{jk}}{\pi - \sum_{f_{ijk} \in F} \theta_{jk}}, \quad v_i \not\in \partial\Sigma$$  (3)

where $\theta_{jk}$ represents the corner angle attached to vertex $v_i$ in all the faces $f_{ijk} \in F$ that share vertex $v_i$, and $\partial\Sigma$ represents the boundary of the mesh $\Sigma$.

The circle packing metric (see Fig. 1) was introduced to approximate the conformal deformation of metrics [24, 25]. The function $\Gamma: V \rightarrow \mathbb{R}^+$ assigns a radius $\gamma$ to each vertex $v_i \in V$. Similarly, the weight function $\Phi: E \rightarrow [0, \frac{\pi}{2}]$ associates the acute angle $\Theta_{ij}$, with each edge $e_{ij} \in E$, where $\Theta_{ij}$ is defined as the angle of intersection of the circles at vertices $v_i$ and $v_j$. The circle packing metric for $\Sigma$ is the pair $(\Gamma, \Phi)$.

Let $u: V \rightarrow \mathbb{R}$ be the discrete conformal factor, which measures the local area distortion, where $u_i = \log \gamma_i$ for each vertex $v_i \in V$. Then, the discrete Ricci flow is described by the following:

$$\frac{du_i(t)}{dt} = (K_i - K_i)$$  (4)

where $K_i$ is the prescribed curvature at vertex $v_i$. The discrete Ricci flow can also be formulated in the variational setting, namely, it is the negative gradient flow of some special energy form:

$$f(u) = \int_{u_0}^{u} \sum_{i=1}^{n} (K_i - K_i)du_i,$$  (5)

where $u_0$ is an arbitrary initial metric. The integration above is well-defined, and it is called the Ricci energy. Then, the discrete Ricci flow is the negative gradient flow of the discrete Ricci energy and the discrete metric which induces $K = (K_1, K_2, \ldots, K_n)^T$ is the minimizer of that energy.

Computing the desired metric with prescribed curvature $K$ is equivalent to minimizing the discrete Ricci energy, which is strictly convex (namely, its Hessian is positive definite). The global minimum exists uniquely, corresponding to the metric $u$, which induces $K$. The discrete Ricci flow converges to this global minimum [1] and the global conformal parameterization for $\Sigma$ can be computed in an automated and robust fashion.

### 4 Computational Algorithm

The pre-computation step of our visualization approach processes a pair of finely-tessellated meshes that represent the original 6-sided CAVE and the target display configuration. The discrete Ricci flow is used to map each mesh to the unit disk and the maps are aligned using a 3-point correspondence on the boundaries. By performing this computation for a specific pair of closed boundaries, we obtain a one-to-one mapping between the viewing rays in the two configurations, which is then stored in a pair of cube-map textures (one for the forward and one for the backward correspondences). The technique is general and can be applied to a number of different display configurations, however, in the following discussion we focus mainly on the 5-sided CAVE as the target platform.

#### 4.1 Discrete Ricci flow

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

We compute the curvature at each vertex $v_i$ and adjust the conformal factor $u_i$ in proportion to the difference between the target curvature $K_i$ and the current curvature $K_i$. 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. Alg. 1 summarizes the computational steps and more details can be found in the work of Jin et al. [8].

**Algorithm 1 Discrete Ricci Flow**

**Require:** Triangular mesh $\Sigma$, target curvature for each vertex $K_i$, error threshold $\varepsilon$

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

1. $u = [u_i], v = [v_i]$ for $u, v \leftarrow 0$
2. while true do
3. Compute edge length $l_{ij}$ for $v_i, v_j$:
4. Compute the corner angle $\theta_{ij}$ in triangle $v_i, v_j, v_k$:
5. Compute the curvature $K_i$ at $v_i$:
6. if $|\hat{K}_i - K_i| < \varepsilon$ then
7. return the discrete metric $l_{ij}$
8. end if
9. Update $u$:
10. end while

#### 4.2 Riemann Mapping

Fig. 2 illustrates the algorithm for computing the Riemann mapping. We remove a face from the mesh $\Sigma$ to convert it to a topological cylinder (Fig. 2(a)), resulting in the creation of a new boundary $\gamma_i$. The target curvature for both the interior and the boundary vertices is set to zero. The Ricci flow described in Alg. 1 produces
a flat cylinder that is periodically embedded in the complex plane (Fig. 2(b)). Each period of the embedding is a rectangle and the original boundaries $\gamma_1$ along the cut face and $\gamma_2$ are aligned with the imaginary axis in the complex plane. The cylinder is then mapped to the unit disk with the hole in the center of the image by the exponential map $e^z$ (Fig. 2(c)). Finally, the central hole is filled to obtain the final mapping (Fig. 2(d)).

4.3 Conformal Mapping for the CAVE

Fig. 3 demonstrates the mapping between the 5-sided CAVE and the 6-sided CAVE. We cut the closed cube (the 6-sided CAVE) along a cross slit at the top and map the cube to a sphere by the direction map

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

where $p$ is a point on the cube and $c$ is the center of the cube. The 6-sided CAVE is mapped to a sphere with slits (Fig. 3(c)), and then it is conformally mapped to the unit disk using Ricci flow (Fig. 3(d)). Similarly, we map the 5-sided CAVE to an open sphere using the same direction map, and then map the sphere to the unit disk.

The Möbius transformation is used to align the two disk images. We choose three corresponding boundary vertices on both the 5-sided CAVE and the 6-sided CAVE geometries, 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

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

then $\eta_2^{-1} \circ \eta_1$ is the desired Möbius transformation, which maps $\{p, q, r\}$ to $\{1, i, -1\}$. Fig. 3(b) and Fig. 3(d) show the result after the alignment.

5 IMPLEMENTING CONFORMAL VISUALIZATION

The algorithm described in Sec. 4 is implemented as a preprocessing step to the visualization. The output of that stage consists of the following:

- Two $8192 \times 8192$ textures in the RGB32F format, containing the normalized viewing directions on the unit disk. These directions are computed directly from the conformal mappings between the 5-sided and 6-sided CAVE meshes and the unit disk.
- The tessellated meshes for the 5-sided and 6-sided CAVE configurations. The aligned conformal parameterizations computed in Sec. 4.3 are stored as UV texture coordinates, referencing the unit disk.

Although the mapping we have computed is defined in the spherical space of view directions in the CAVE, we encode it in a cube
texture, which is natively supported on the GPU and can be sampled efficiently in both vertex shaders and pixel shaders. The cube-map is generated by placing a virtual camera at the center of the 5-sided CAVE mesh and rendering the texture-mapped faces to 6 32-bit precision render targets, one for each CAVE wall. The buffers are then composited into a single cube-map texture in the vertical-cross format. At this point, we use the disk texture computed from the 6-sided mesh and we refer to the cube-map as $T_{\text{ray}}$ (see Fig. 4(e)). Similarly, we compute the reverse mapping by rendering the mesh for the 6-sided configuration with the correct parameterization for the disk texture from the 5-sided mesh, and label the resulting cube-map as $T_{\text{geom}}$. We determined experimentally that $1024 \times 1024$ cube-map resolution with 32-bit floating-point precision per component is sufficient so that no banding or artifacts are visible in the final visualization results.

The conformal maps stored in the two cube-map textures are suitable for both forward and backward rendering pipelines. We demonstrate applications to mesh rendering with OpenGL, single-pass raycasting for Direct Volume Rendering (DVR) and real-time raytracing on the GPU.

5.1 Mesh Rendering

We first apply the conformal distortion to a mesh-based rendering pipeline. We use an efficient vertex-based distortion for rendering well-tessellated geometry with OpenGL, similar to the approach by Spindler et al. [22]. Intuitively, every vertex in the scene is transformed so that triangles that are projected on the top screen in the 6-sided CAVE configuration are instead projected on the 4 side screens in the 5-sided configuration. This transformation is defined by the following:

\[
\begin{align*}
\mathbf{r}_w &= (\mathbf{M}_{\text{wc}}^{-1})^T \cdot \mathbf{n} \left( \mathbf{p}_w - \mathbf{M}_{\text{wc}}^{-1} \cdot [0, 0, 0, 1]^T \right) \\
\mathbf{p}_w' &= \mathbf{M}_{\text{wc}}^{-1} \cdot (T_{\text{geom}}(\mathbf{r}_c) \cdot [\mathbf{M}_{\text{wc}} \cdot \mathbf{p}_w])
\end{align*}
\]

where $\mathbf{p}_w$ is the vertex position in world-space, $\mathbf{r}_c$ is the normalized view direction in CAVE space and $\mathbf{M}_{\text{wc}}$ is the world-space to CAVE-space transformation matrix. In our rendering framework, the head node for the visualization cluster emits the camera information to all the rendering clients and the view matrix $V$ associated with that camera is the world-space to CAVE-space transformation matrix. Each visualization node then computes the final view and projection matrices based on the target projection surface (e.g., front, left, etc.). The geometry transformation is performed in a custom vertex shader that is bound to every primitive in the scene.

5.2 Direct Volume Rendering

The distortion is applied in a similar fashion for volume rendering. The visualization algorithm is based on single-pass ray-casting over 3D textures with support for advanced lighting and shadowing, as well as pre-integrated transfer functions [4]. Our framework integrates volume rendering tightly into the scene graph and we render out the volume-space positions of the front and back faces of the volume bounding box, modified by the depth of other scene geometry. One possible approach for incorporating the distortion map is to tessellate the bounding box and apply the $T_{\text{geom}}$ transformation as described in Sec. 5.1. However, our target application is the exploration of the virtual colonoscopy data, in which case the camera is often within the volume and the starting positions of the rays are defined on the near clipping plane. It is more accurate to transform the positions on the near clipping plane and the back face of the bounding volume to world-space and then to apply the following transformation:

\[
\begin{align*}
\mathbf{r}_c &= (\mathbf{M}_{\text{wc}}^{-1})^T \cdot \mathbf{n} \left( \mathbf{p}_w - \mathbf{M}_{\text{wc}}^{-1} \cdot [0, 0, 0, 1]^T \right) \\
\mathbf{p}_w' &= \mathbf{M}_{\text{wc}}^{-1} \cdot (T_{\text{ray}}(\mathbf{r}_c) \cdot [\mathbf{M}_{\text{wc}} \cdot \mathbf{p}_w])
\end{align*}
\]

where, again, $\mathbf{p}_w$ is the vertex position in world-space, $\mathbf{r}_c$ is the normalized view direction in CAVE space and $\mathbf{M}_{\text{wc}}$ is the world-space to CAVE-space transformation matrix. The viewing vector is then constructed from the modified starting and ending positions. Note that this transformation is very similar to Eq. 6. The difference is that since this is a backward rendering pipeline, we are transforming the view directions directly and the $T_{\text{ray}}$ cube-map should be used.

5.3 GPU Raytracing

A major limitation of the geometry transformation 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 and non-uniformly tessellated meshes. A number of GPU-based ray-tracing algorithms have been proposed that can achieve interactive frame-rates even for large geometric models [12, 17, 19, 31].

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. The programs are then intelligently scheduled on all CUDA-enabled GPUs in the system. Its features are comparable to other real-time
ray-tracing approaches [19, 31]. For our conformal visualization, the ray transformation is applied at the ray-generation level, which is separate from the scene-graph and therefore much simpler to re-implement. The distortion computation is also simplified:

\[
d_{w} \rightarrow M_{wv}^{T} \cdot \left( T_{ray} \left( (M_{wc}^{-1})^{T} \cdot d_{w} \right) \right) \tag{8}
\]

Similarly to the volume rendering approach, we transform the world-space ray direction \(d_{w}\) to CAVE-space and fetch a new ray direction from the conformal map. CUDA (and by extension OptiX) does not support cube textures natively, therefore indexing the conformal map is implemented in the ray-generation program. In addition to simplifying the application of the conformal map, the ray-tracing enables a number of effects that are particularly suitable for architectural visualization, such as dynamic shadows, global illumination, reflections and refractions.

6 Results

We first demonstrate the visual properties of our approach on a unit sphere with a checkerboard texture. In Fig. 5, the virtual camera is positioned at the center of the sphere so that the two poles of the texture are shown on the left and the right displays. Fig. 5(a) shows the results of our conformal visualization and demonstrates how the visual information originally shown on the top screen is instead rendered on the left, front, right and back screens. For Fig. 5(c) we apply the \(T_{ray}\) transformation a second time rotated by 90\(^\circ\) to map the viewing directions for the back screen to the remaining 4 walls. This particular transformation illustrates a preliminary application of our technique to a 4-sided CAVE, however, it results in a 8\% loss in the field of view since the second pass does not account for the missing top screen. The transformation for the 4-sided CAVE can be generated directly with the algorithm described in Sec. 4, in which case no such loss will occur.

![5-sided CAVE](a) ![6-sided CAVE](b) ![4-sided CAVE](c)

Figure 5: Raytracing of a checkerboard sphere in (b) a simulated 6-sided CAVE, (a) 5-sided CAVE with our conformal visualization and (c) preliminary results for a 4-sided CAVE. The layout is in the standard vertical cross format and the color coding is defined for the original walls in the 6-sided CAVE configuration (green for front/back, blue for left/right, red for top/bottom).

User interactions with the virtual scenes often involve navigation in 3D space. We demonstrate that aspect of our conformal visualization with a checkerboard tunnel. Fig. 6 illustrates the camera panning down during the navigation. Note that compared to the straightforward front projection, the conformal rendering preserves the context of the tunnel’s direction even when the camera is pointed downward. Also, the visual information corresponding to the original front projection is presented with minimal distortion, as also shown in Fig. 5(a).

Interactive ray-tracing is supported in our visualization software as a complete replacement for the OpenGL renderer. As described in Sec. 5.3, the application of the distortion map is simplified in terms of both computation and integration with the 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 1024×1024 viewport. The production cluster environment consists of 5 workstations with an Intel Xeon processor and dual NVIDIA Quadro FX 4600 GPUs with 10 1400×1050 viewports.

We first evaluate the conformal visualization for the rendering of volumetric data in IVC. The pair of images on the left in Fig. 7(a) are the original perspective projections for the front and top screens in a fully-enclosed CAVE. The image on the right is the result of the conformal visualization described in Sec. 5.2 for the front screen in the 5-sided CAVE. Note that unlike in image-based retargeting approaches, the final result is rendered directly, and the original images are included here only for demonstration purposes. The source of the volume data is a 512×512×451 16-bit CT scan of a patient’s abdomen after digital cleansing of the colon. A suspicious area is visible on the top screen, which would normally be missed in the CAVE due to the angle of the approach and the lack of a top screen. After the conformal distortion, this area is projected onto the front CAVE 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 actual 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.

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 attending the session 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.

Interactive ray-tracing is supported in our visualization software as a complete replacement for the OpenGL renderer. As described in Sec. 5.3, the application of the distortion map is simplified in terms of both computation and integration with the 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 1024×1024 viewport. The production cluster environment consists of 5 workstations with an Intel Xeon processor and dual NVIDIA Quadro FX 4600 GPUs with 10 1400×1050 viewports.

We first evaluate the conformal visualization for the rendering of volumetric data in IVC. The pair of images on the left in Fig. 7(a) are the original perspective projections for the front and top screens in a fully-enclosed CAVE. The image on the right is the result of the conformal visualization described in Sec. 5.2 for the front screen in the 5-sided CAVE. Note that unlike in image-based retargeting approaches, the final result is rendered directly, and the original images are included here only for demonstration purposes. The source of the volume data is a 512×512×451 16-bit CT scan of a patient’s abdomen after digital cleansing of the colon. A suspicious area is visible on the top screen, which would normally be missed in the CAVE due to the angle of the approach and the lack of a top screen. After the conformal distortion, this area is projected onto the front CAVE 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 actual 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.

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 attending the session 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.
purposes and the final result is rendered directly. The performance of the renderer scales with the size of the viewport as opposed to the scene complexity in the rasterization pipeline. Although our colon mesh contains geometry with uniformly-high triangle 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 conformal visualization results even in areas of low polygonal density. While the performance for the architectural scene is lower than with the rasterization pipeline, it allows for interactive frame-rates and special effects such as accurate glass reflections and shadows. Again, the cost of applying the transformation map is negligible compared to the overall rendering and its effect on the frame-rate is below 1%.

The conformal visualization preserves shapes locally, however scenes with expansive straight edges present certain challenges. In particular, straight edges projected onto the missing ceiling surface are bent toward the top center of the front, side and back screens. The Ricci flow allows for direct control of the boundary during the mapping with the result that the distortion is minimized for areas of the scenes that were projected on the original 5 surfaces. Fig. 7(b) illustrates this - the lower railings and the column are visually similar to the original projections onto the front screen. The goal of conformal visualization in this application is to present additional visual context for the navigation while preserving the relationships between local structures. We provide a simple control over the pre-generated conformal maps that allows the user to specify how much of the ceiling projection is visualized onto the screens. The resulting choice is between visual distortions and visual coverage, and the selection can be optimized for the particular application or dataset. This control is implemented in the ray-generation kernel and can be modified at runtime without re-computing the conformal map or the cube texture. For illustrative purposes, Fig. 7(b) presents a worst-case distortion scenario with approximately 95% ceiling coverage.

In addition to architectural scenes, we have performed visualization experiments with large connected data structures. The conformal visualization is particularly suitable for scenes containing a large number of elements when the exact shape of the connections is not relevant to understanding the data, but the shape of the elements may be. The expanded field of view and the resulting visual context can be beneficial for exploration tasks in information visualization applications. Fig. 7(c) presents our test scene, which contains an abstract binary tree with 10,000 data elements and color-coded connections.

7 Evaluation

Our conformal visualization is evaluated informally in the 5-sided CAVE for the visual detection of polyps in phantom colon datasets. The data is generated from a section of the colon centerline of a real patient who had undergone Virtual Colonoscopy. There are 4 datasets similar to the one in Fig. 8 with between 25 and 29 hemispherical polyps. The control scheme in the CAVE uses an analog gamepad with the following mapping: view control with the left analog nub, forward/backward motion with the right analog nub, camera roll with the shoulder pads. The initial camera roll is selected randomly at the beginning of each experiment. The users are given time to study the navigation scheme for both regular and conformal visualization, after which they are asked to find as many of the hemispheres as possible. Each user examines the 4 datasets with a pre-set rendering modality for each run. The examiner records the number of detected polyps, the number of false positives and the time for each fly-through.

Our informal study had 12 participants (11 male, 1 female) and on average, the examination time was consistent between the regular and conformal visualization. The conformal visualization improved the non-biased detection sensitivity for polyps from 91% to 93% overall with individual differences in the range $[-2\%,+10\%]$. On average this translates to the detection of up to 0.9 additional polyps over the baseline per dataset with 29 polyps. Stereoscopic vision remained effective for the majority of users in both the low- and high-distortion sections of the images. If we isolate the polyps that were projected only to the missing ceiling, the detection rates were close to the average for the dataset with conformal visualization, compared to 0% for the regular volume rendering. In both cases, the false positive rate was about 0.1%.
8 Conclusions and Future Work

We have presented a visualization technique based on conformal mapping for partially immersive visualization platforms that allow the user to visually explore the full virtual environment. The discrete Ricci flow is used to compute the mapping and the theoretical foundation for our approach guarantees the consistency of local shapes under the deformation.

We have demonstrated applications of our approach to rasterization, volume rendering and GPU-based ray-tracing pipelines for medical, architectural and abstract data visualization in the 5-sided CAVE. We avoid many of the challenges related to image-based re-targeting approaches by rendering the final results directly onto the output surfaces, which allows for the generation of accurate stereoscopic image pairs. Our user study simulates the task of finding cancerous polyps in Virtual Colonoscopy and we demonstrate improved sensitivity compared to the traditional visualization at similar examination times and false positive rates. In addition, the discrete Ricci flow can also be used to compare the conformal maps for arbitrary display configurations that are topologically equivalent to a disk and we show preliminary results for the 4-sided CAVE. This is a generalization of the technique that we plan to explore in detail.

Currently, the conformal mapping 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. A GPU-based implementation may provide adequate performance for the dynamic generation of the distortion map. In addition, the generation of the conformal maps can be augmented with information from scene or rendering features so that the distortion can be controlled around areas of interest. Certain distortion controls can also be achieved by resampling the cube map at runtime without the need to recompute the underlying conformal map.

Acknowledgements

This work has been supported by NSF grants CCF-0448399, CCF-0702699, CNS-0959979, CNS-1016829, IIS-0916235, IIS-0916286, NIH grant R01EB7530 and ONR grant N0001409110228. The volumetric colon datasets have been obtained through the NIH, courtesy of Dr. Richard Choi, Walter Reed Army Medical Center.

References