**Immersive Conformal Visualization**

Kaloian Petkov and Arie Kaufman

Center for Visual Computing, Stony Brook University

### Mesh Templates

**Define source and target visibility meshes**
- Source mesh is fully enclosed
- Target mesh matches the topology of the display device
- Source mesh is cut to match the boundary of the target mesh

### Conformal Mapping

- Map the mesh template to a sphere
- Remove a face from the mesh
- Cut the geometry along \( \tau \)
- Flatten to the 2D plane
- Map to the annulus
- Map the source mesh to the disc
- Project the 2D mapping back to the mesh

### Conformal Visualization

**Visual information from the missing walls is recovered**
- Conformal map is angle-preserving, therefore shapes are preserved locally
- Size of objects changes near the boundary

### Dynamic Visibility Manipulation

Reference point is moved during rendering of the conformal map
- Computed entirely on the GPU, real-time performance
- Implemented as a Focus+Context user interface

### Rasterization

**GPU tessellation of all scene geometry**
- Smooths the results of conformal mapping
- Adaptive tessellation based on projection sizes

**Conformal map applied during rendering**
- Performance scales with scene complexity
- Computed in a custom geometry shader
- All vertices are translated based on the \( T_{\text{geom}} \) transformation

### Volume Rendering and Raytracing

**Viewing rays are transformed directly**
- Using the \( T_{\text{map}} \) conformal transformation
- Performance scales with image resolution

**Direct Volume Rendering (DVR)**
- Same view with Conformal Visualization

**GPU Raytracing**
- Same view with Conformal Visualization