Visualizing PML

David Dumas
University of Illinois at Chicago

July 1, 2016

The PML Visualization Project

dumas.io/PML

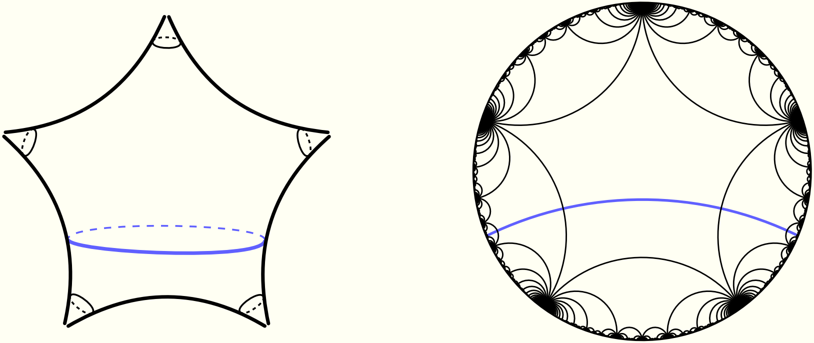
Joint work with François Guéritaud (Univ. Lille)

I will also demonstrate 3D graphics software developed by UIC undergraduate researchers Galen Ballew and Alexander Gilbert.

Mathematical Computing Laboratory

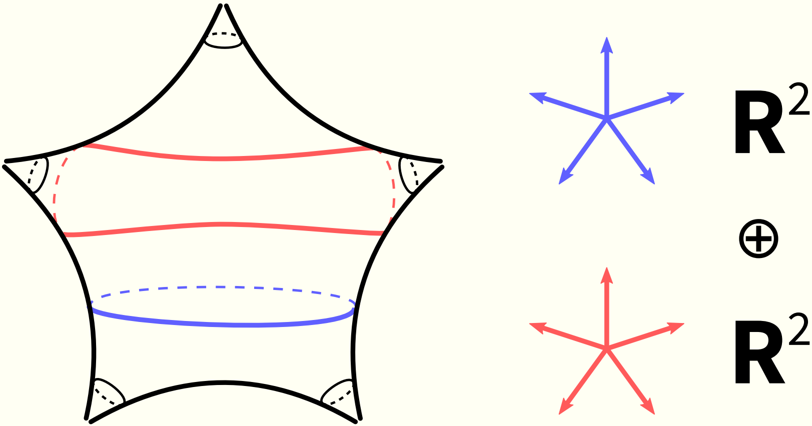
UIC

Five-punctured sphere



$S_{0,5}$

Earthquake basis



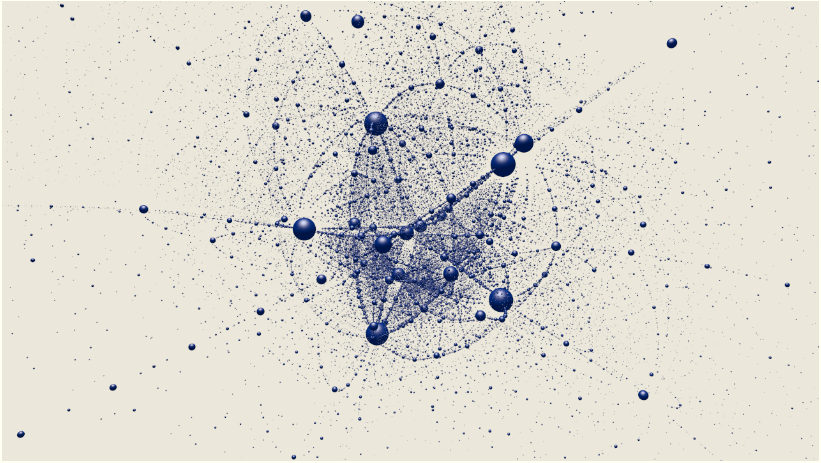
Observations

Already apparent:

- Features related to short curves dominate
- Lots of “filaments”; all have corners

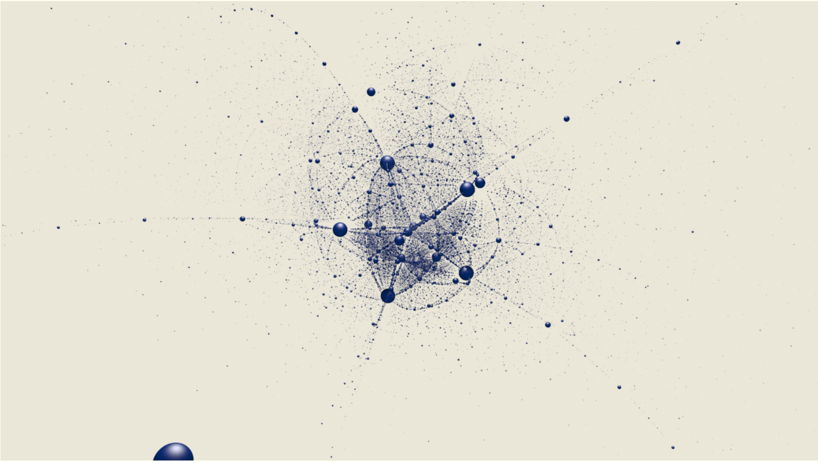
Exploring variations and alternatives, we also found:

- Several choices for simple curve cutoffs give visually indistinguishable results
- “First person” perspective from the antipode is theoretically natural, but feels too limiting in pre-rendered animations



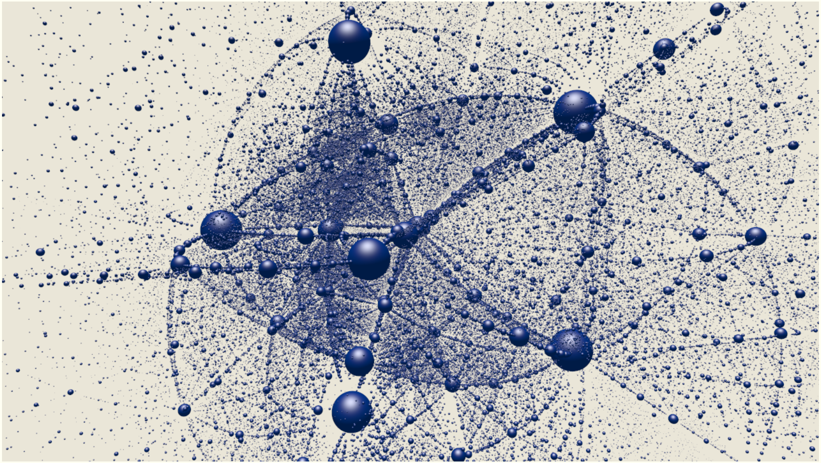
pmls05-001

Rotating the pole



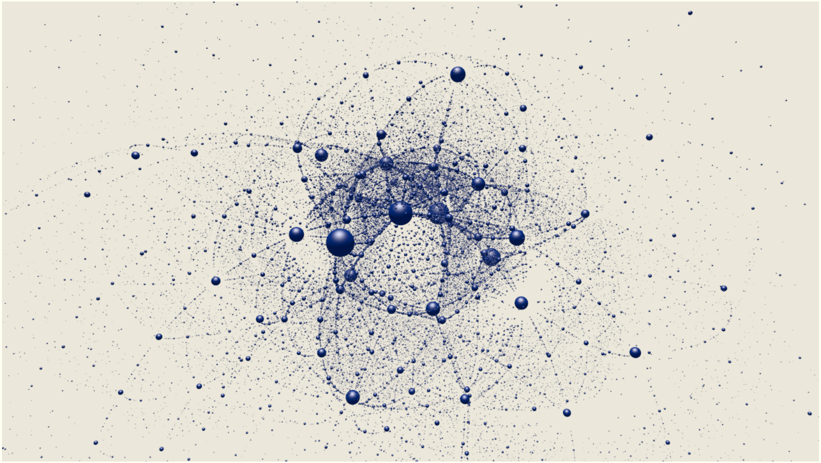
pmls05-010

Closer?



pmls05-020

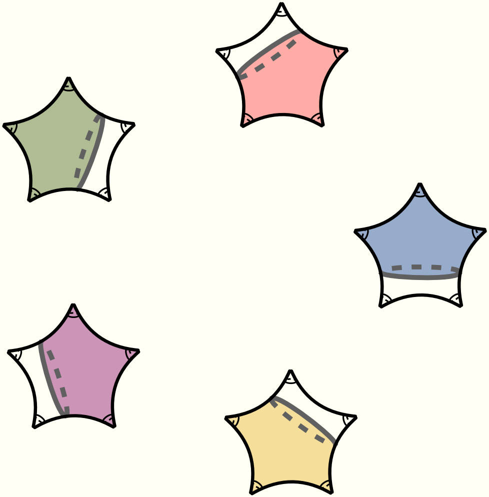
Clifford flow



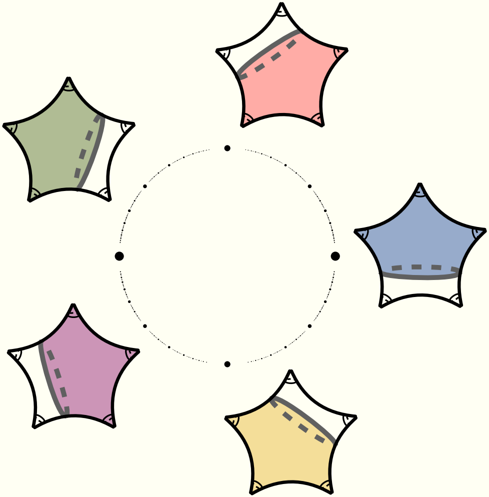
pmls05-030

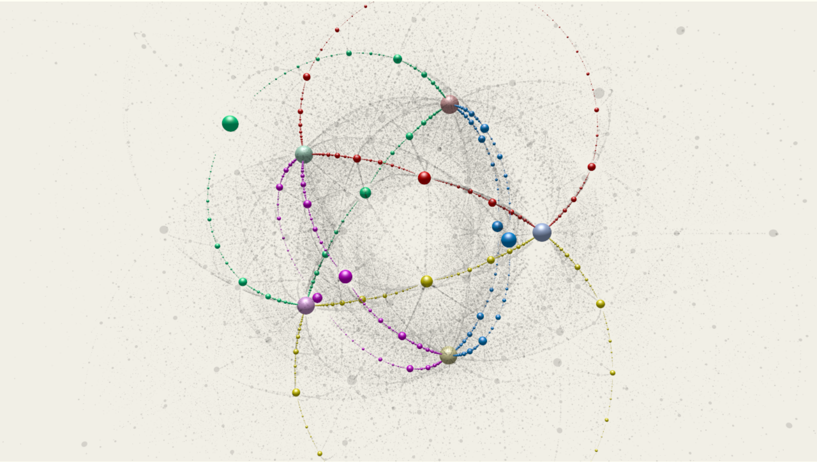
Rings

Rings



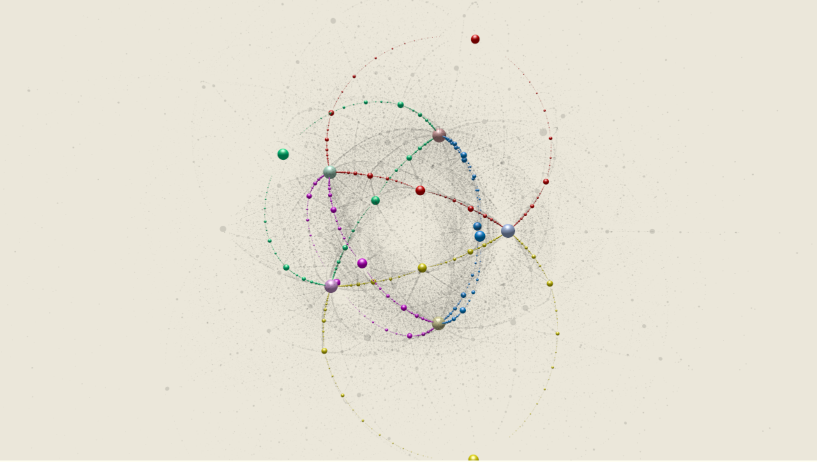
Rings





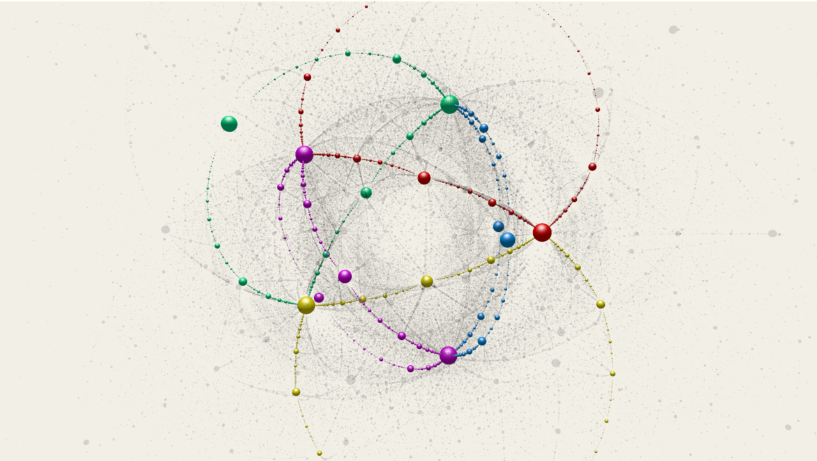
pmls05-071

Rotating the pole



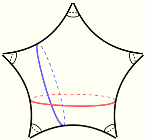
pmls05-041

Rotating the pole II

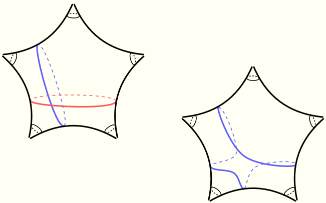


pmls05-061

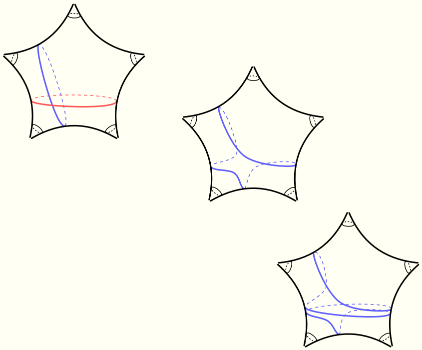
Twists



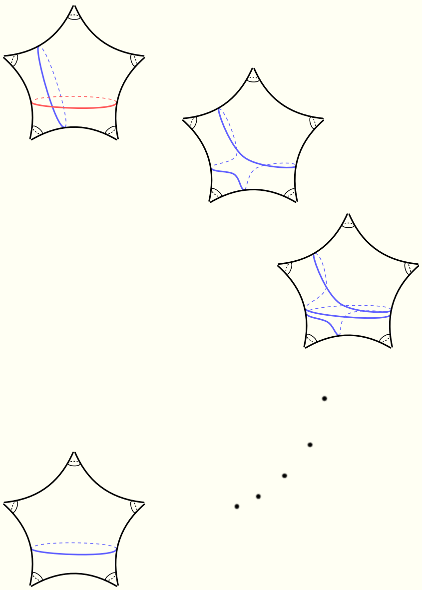
Twists



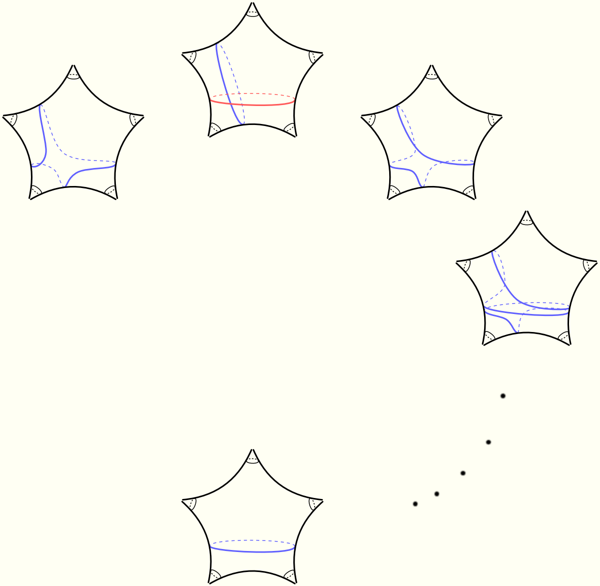
Twists



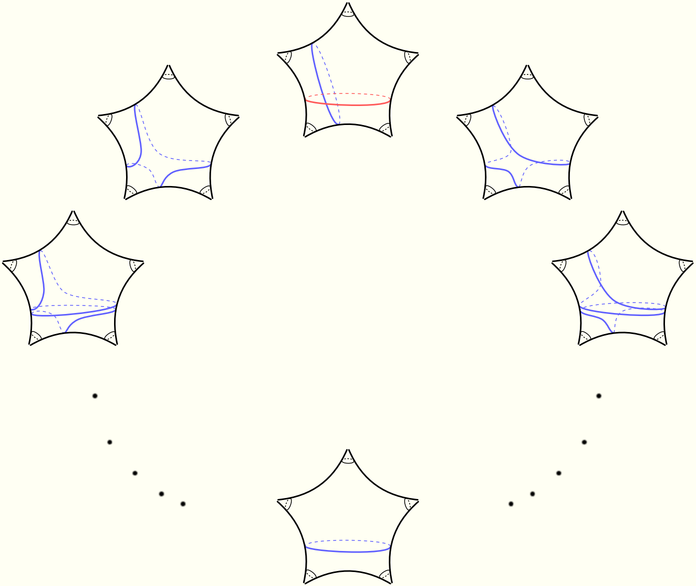
Twists

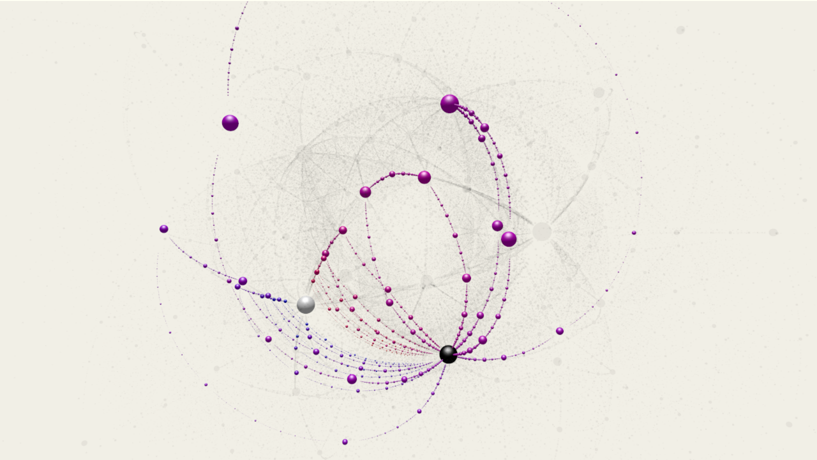


Twists



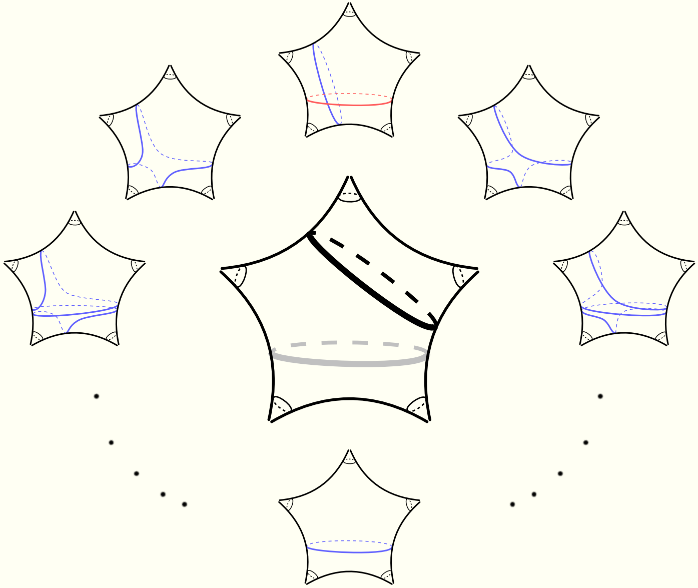
Twists



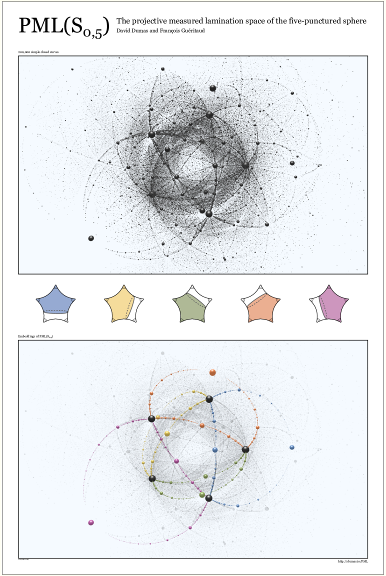


pmls05-081

Twists

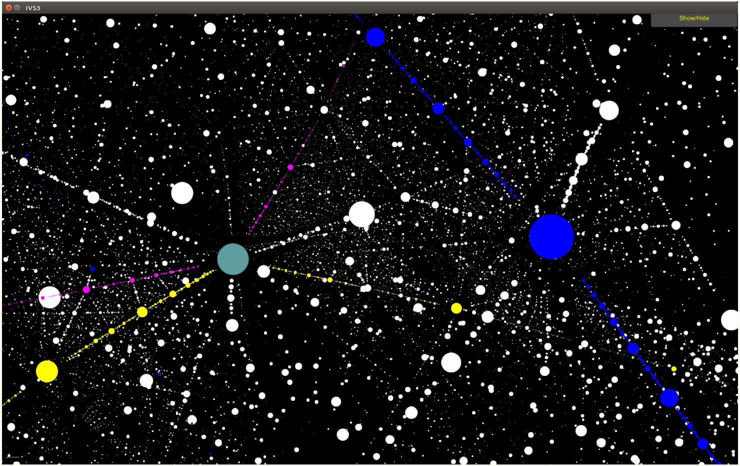


Rings poster



PDF for full-size printing at: dumas.io/PML/

Unity 3D Demo



By Galen Ballew and Alexander Gilbert, undergraduate researchers in UIC's Mathematical Computing Laboratory.

Toolchain

python

Toolchain

Toolchain

python

POV-Ray

Toolchain

python

POV-Ray

Linux, Emacs, GNU Parallel, ffmpeg, ...

Toolchain

python

POV-Ray

Linux, Emacs, GNU Parallel, ffmpeg, ...

Unity 3D, Oculus Rift, WebGL, ...

Process

1. Fuchsian representation
2. Cocycle basis
3. Enumerate simple closed curves
4. Covectors
5. Spheres
6. Ray-tracing
7. Encoding / post-processing

Fuchsian representation

A description of the base hyperbolic structure X in a form that allows computation of lengths.

Typical (e.g. $S_{0,5}$):

2×2 matrix generators for the Fuchsian group

Alternative (e.g. $S_{1,1}$):

Sufficiently many traces of elements to determine the Fuchsian representation up to conjugacy

Cocycle basis

A basis for $T_x\mathcal{T}(S)$ represented in same form as the base hyperbolic structure.

Write a family of representations

$$\rho_t : \pi_1 S \to \mathrm{SL}_2\mathbf{R}$$

as

$$\rho_t(\gamma) = \left(\mathrm{Id}_{2\times 2} + t\,u(\gamma) + O(t^2)\right)\rho_0(\gamma).$$

Then $u : \pi_1 S \to \mathrm{Mat}_{2\times 2}\mathbf{R}$ is a cocycle representing the tangent vector $\frac{d}{dt}\rho_t\big|_{t=0}$ to $\mathcal{T}(S)$.

Simple closed curves

Homotopy classes of closed curves are conjugacy classes in the group $\pi_1(S)$.

Of these, we only want the simple ones.

Procedure:

- Start with a few “seed” words (known to be simple)
- Generate more curves by applying mapping classes
- Repeat until a stopping condition attained, e.g.
  - Max word length
  - Max hyperbolic length
  - Max depth in Mod(S)

Covectors

Hyperbolic translation length ℓ of an element $A \in \mathrm{SL}_2\mathbf{R}$:

$$\ell = 2\operatorname{arccosh}(\frac{1}{2}\mathrm{tr}(A))$$

For each word w representing a simple curve α and for a basis of cocycles u_i:

- Compute length of w at X and at $X + \epsilon u_i$
- Difference quotient approximates

  $$\frac{d\mathrm{length}(\alpha)}{du_i}$$

  i.e. component i of the d(length) covector.
- Divide by length at X to get d(log(length))

Spheres

In $S_{0,5}$ case we now have a list of tuples

$$(w, \ell, \frac{d\ell}{du_1}, \frac{d\ell}{du_2}, \frac{d\ell}{du_3}, \frac{d\ell}{du_4})$$

which in practice might look like:

```
acADaCbcd 22.5373 -0.6807 0.6506 -0.8551 0.3537
```

Stereographic projection of the 4-vector gives the center and a negative power of ℓ gives the radius.

Generate a POV-Ray sphere primitive:

```
sphere { <-1.001967,-1.154298,0.477426>, 0.014278 }
```

Ray-tracing and encoding

A POV-Ray scene file sets background, lighting, camera parameters and imports the list of spheres generated from the covectors.

For animations: Iterate over a list of parameter values for stereographic projection, camera position, etc. to make a series of frame images.

Compress/encode frame images to h.264/mp4 video with ffmpeg.

Ray-tracing and encoding

Along the way, we made a ffmpeg frontend for encoding video from a series of frame images.

Features:

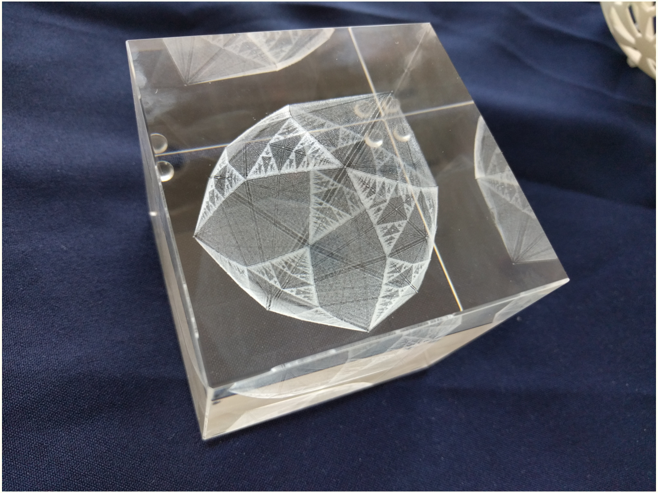
- Read image file names from a “manifest” file
- Simplified option syntax

http://github.com/daviddumas/ddencode/

PML rendering demo

Code at http://github.com/daviddumas/pmls05-demo/

Glass cube



Laser engraving with technical assistance from Bathsheba Grossman

3-punctured projective plane

$N_{1,3}$ = Non-orientable surface with 1 crosscap and 3 punctures.

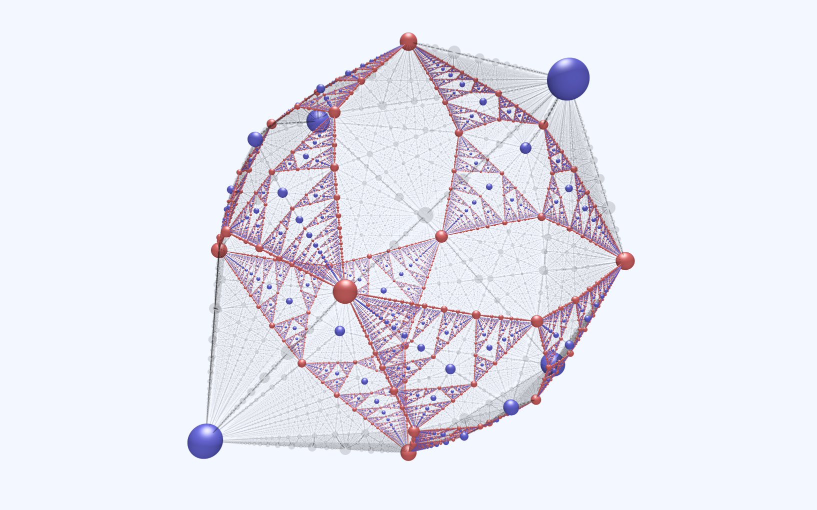
Teichmüller space has dimension 3, so PML $\simeq \mathbf{S}^2$!

Has one-sided and two-sided simple curves.

Scharlemann:

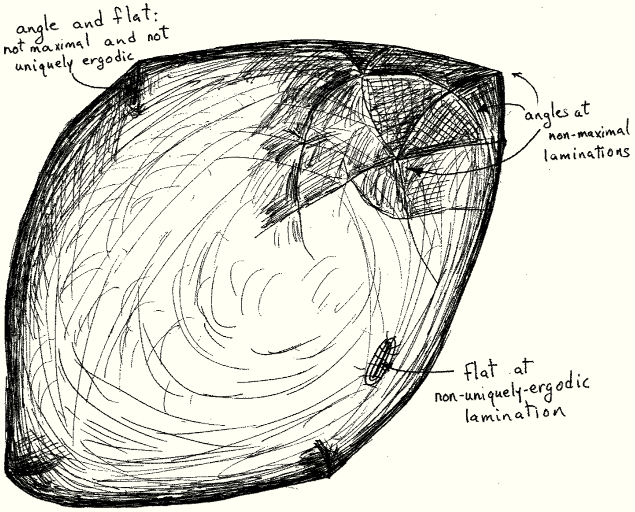
- One-sided curves are isolated points of the image of $\mathcal{C}$
- Two-sided curves are dense in a gasket, which is also the limit set of the one-sided curves

Open problem: Compute Hausdorff dimension of this gasket in PL coordinates or in the Thurston embedding.



n13-010

Thurston’s drawing of PML



From “Minimal stretch maps between hyperbolic surfaces”, preprint, 1986.

Added in proof (after the lecture):

- There were questions about minimal but non-uniquely ergodic laminations. None of the pictures show these directly. Such laminations exist on $S_{0,5}$ but I do not know whether they exist on $N_{1,3}$. I suspect not.

David Dumas
david@dumas.io