Lecture 23: Spectral Meshes

COMPSCI/MATH 290-04

Chris Tralie, Duke University

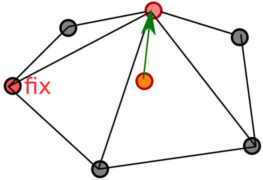
4/7/2016

Announcements

- First project milestone Monday 4/11/2016
- Final Project Rubric Up
- Group Assignment 3 Out Tomorrow or Saturday...

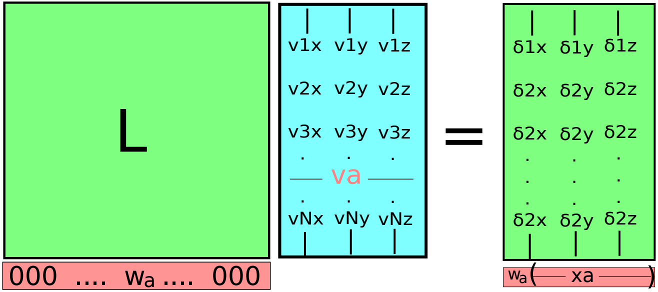
Table of Contents

Laplacian Mesh Editing: Anchors



$$\delta_i = \sum_{j \in N(i)} (v_i - v_j)$$

Delta coordinates define relative information about vertices with respect to their neighbors



A Note About Rotation Invariance

$$Lx = \delta$$

- ▷ δ is a *vector*. $||\delta|| \propto \kappa$, but it has a direction

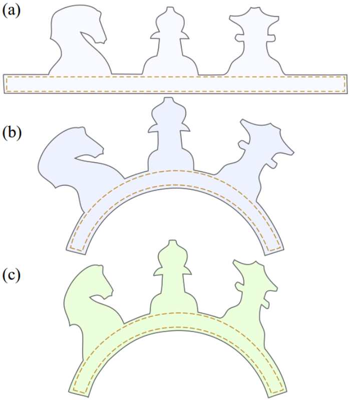
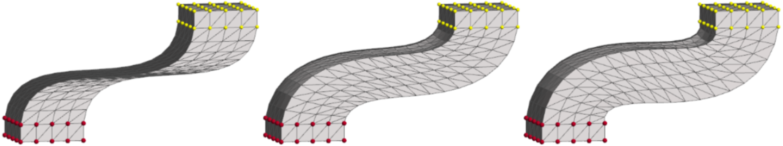


Figure: Sorkine05

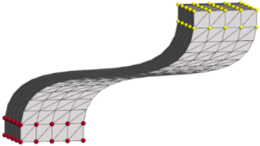
As Rigid As Possible Surface Editing



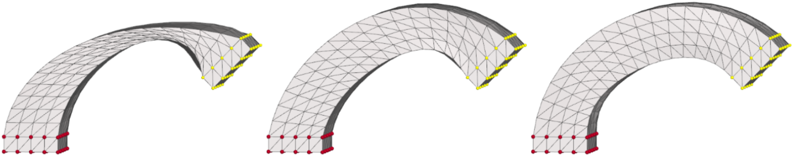
initial guess 1 iteration 2 iterations

Sorkine2007

As Rigid As Possible Surface Editing



initial guess   1 iteration   2 iterations



initial guess   1 iteration   4 iterations

Sorkine2007

Table of Contents

The 1D Second Derivative Operator

Let $f(x) = \sin(\omega x)$
What is $f'(x)$?

The 1D Second Derivative Operator

Let $f(x) = \sin(\omega x)$
What is $f'(x)$?

$$f'(x) = \omega \cos(\omega x)$$

The 1D Second Derivative Operator

Let $f(x) = \sin(\omega x)$
What is $f'(x)$?

$$f'(x) = \omega \cos(\omega x)$$

What is $f''(x)$?

The 1D Second Derivative Operator

Let $f(x) = \sin(\omega x)$
What is $f'(x)$?

$$f'(x) = \omega \cos(\omega x)$$

What is $f''(x)$?

$$f''(x) = -\omega^2 \sin(\omega x)$$

The 1D Second Derivative Operator

Let $f(x) = \sin(\omega x)$
What is $f'(x)$?

$$f'(x) = \omega \cos(\omega x)$$

What is $f''(x)$?

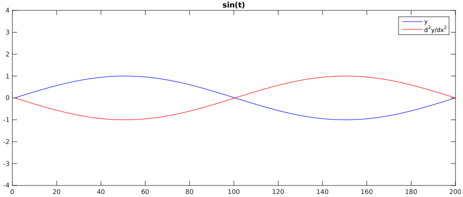
$$f''(x) = -\omega^2 \sin(\omega x)$$

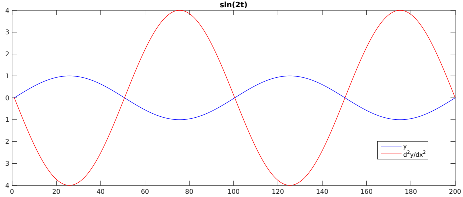
$$f''(x) = -\omega^2 f(x)$$

We say sines and cosines are *eigenfunctions* of the second derivative operator and $-\omega^2$ is the associated *eigenvalue* (just like vectors!)

1D Derivative Eigenfunctions

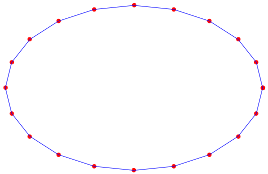
Let $f(x) = \sin(\omega x)$, $f''(x) = -\omega^2 \sin(\omega x)$





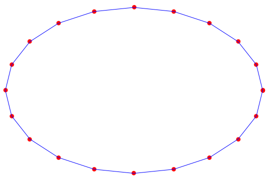
Eigenvalues tell us frequency information

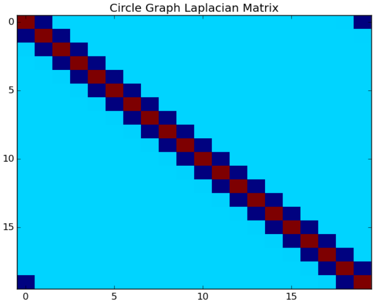
Discrete Circle Laplacian



https://github.com/COMPSCI290-S2016/NumpyDemos/blob/master/1DLaplacian.py

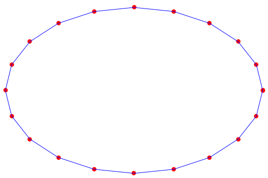
Discrete Circle Laplacian

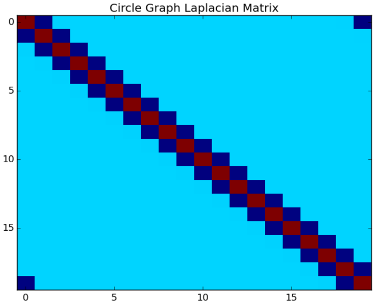


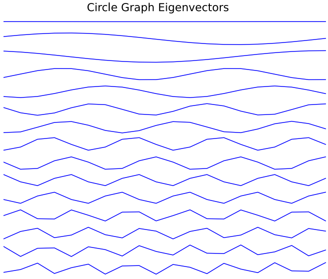


https://github.com/COMPSCI290-S2016/NumpyDemos/blob/master/1DLaplacian.py

Discrete Circle Laplacian







https://github.com/COMPSCI290-S2016/NumpyDemos/blob/master/1DLaplacian.py

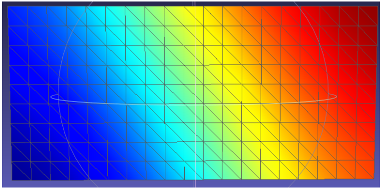
Laplacian of All 1s

$$L = D - A$$

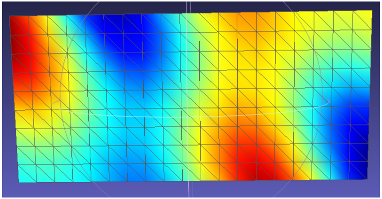
$$L \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = ?$$

Rect Eigenvectors

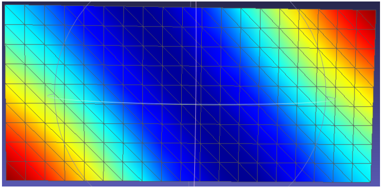
1



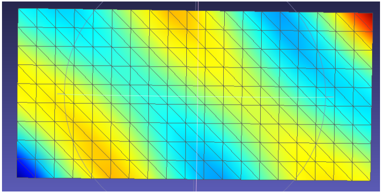
7



2



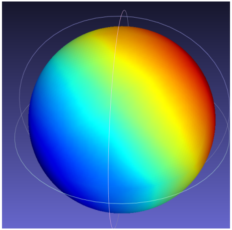
8



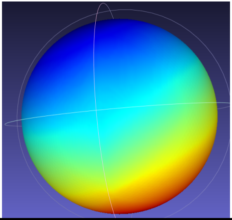
Sphere Eigenvectors

Sphere Eigenvectors

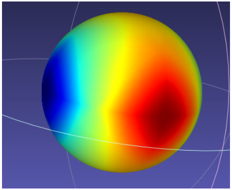
1



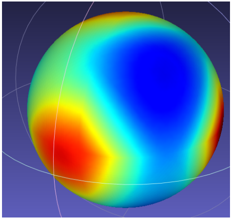
2



7



8



Homer Modes

0

1

2

Homer Modes

3

4

5

Homer Modes

6

7

8

Homer Modes

9 10 11

Homer Modes

12

13

14

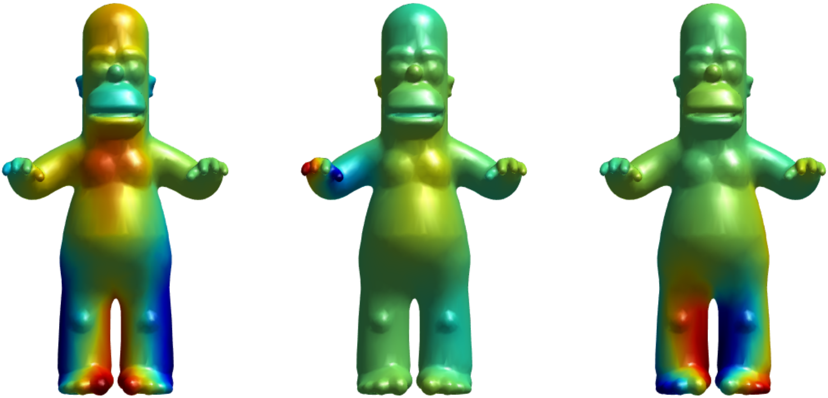
Homer Modes

15

16

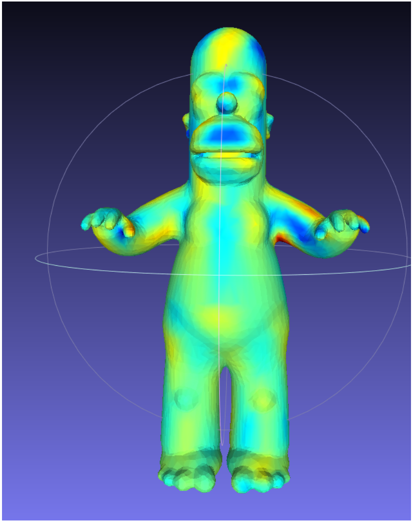
17

Homer Modes



18 19 20

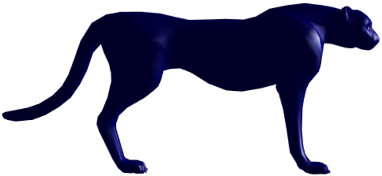
Homer Modes



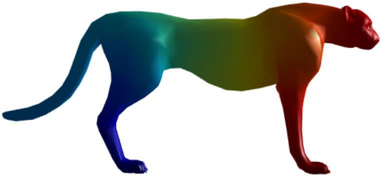
200

Cheetah Modes

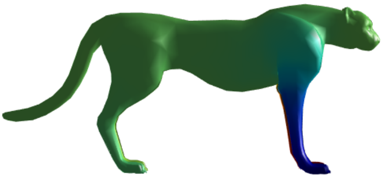
0



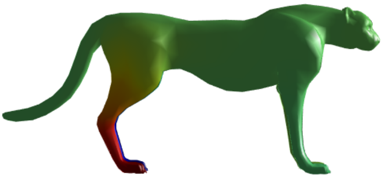
1



2

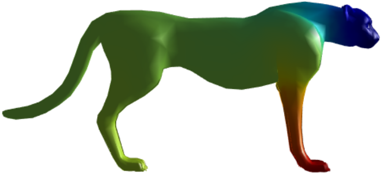


3

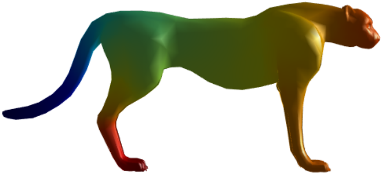


Cheetah Modes

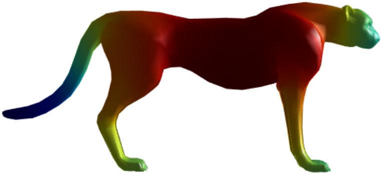
4



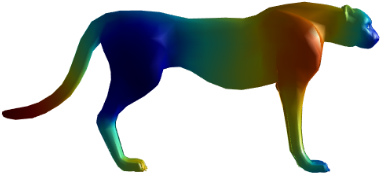
6



6

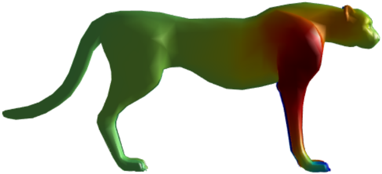


7

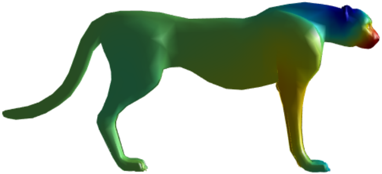


Cheetah Modes

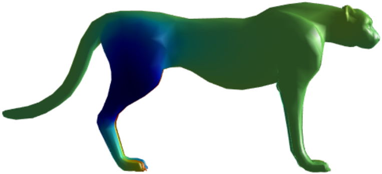
8



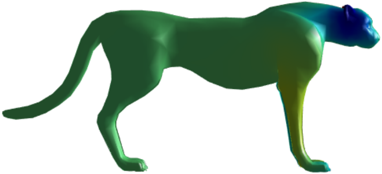
9



10

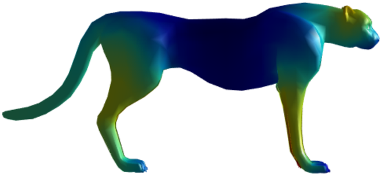


11

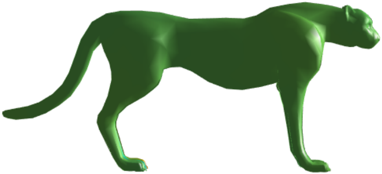


Cheetah Modes

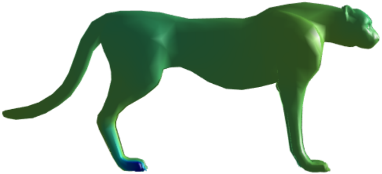
12



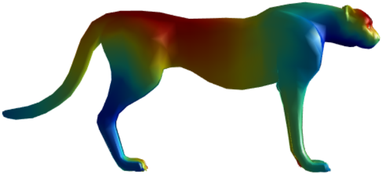
13



14

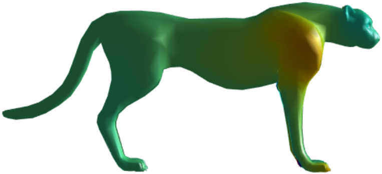


15

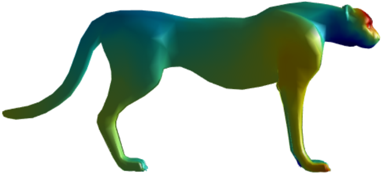


Cheetah Modes

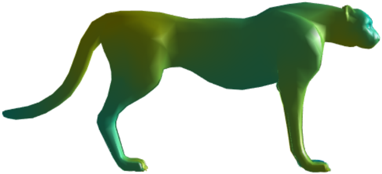
16



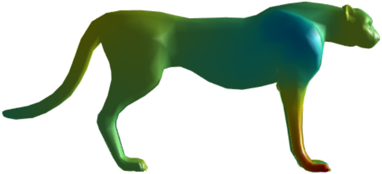
17



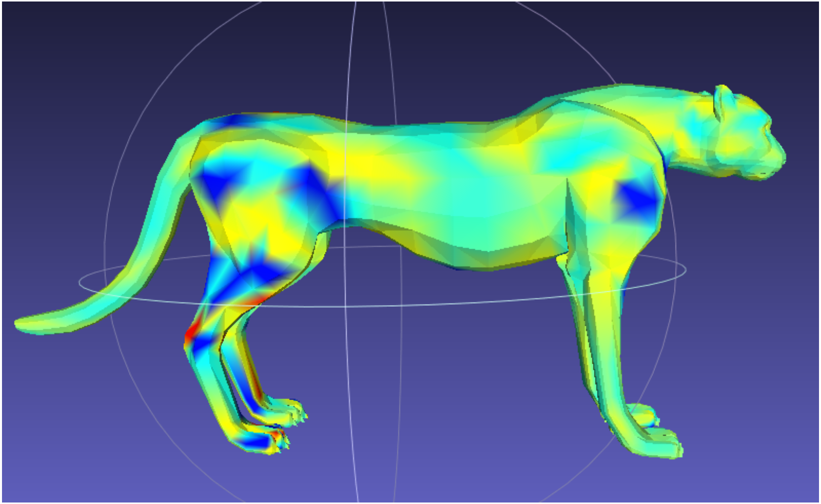
18



19



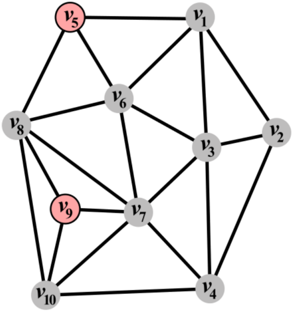
Cheetah Modes



400

Normalize Each Row By Degree?

Why not $\hat{L} = D^{-1}A$?

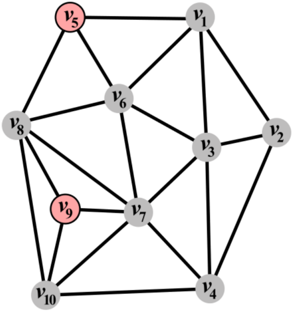


$$
\begin{array}{c}
1/4 \\ 1/3 \\ 1/5 \\ 1/4 \\ \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \\
\end{array}
\begin{bmatrix}
4 & -1 & -1 & & -1 & -1 & & & & \\
-1 & 3 & -1 & -1 & & & & & & \\
-1 & -1 & 5 & -1 & & -1 & -1 & & & \\
 & -1 & -1 & 4 & & & -1 & & & -1 \\
-1 & & & & 3 & -1 & & -1 & & \\
-1 & & -1 & & & 4 & -1 & -1 & & \\
 & & -1 & -1 & & -1 & 6 & -1 & -1 & -1 \\
 & & & & -1 & -1 & -1 & 6 & -1 & -1 \\
 & & & & & & -1 & -1 & 3 & -1 \\
 & & & -1 & & & -1 & -1 & -1 & 4
\end{bmatrix}
$$

Sorkine 05

Normalize Each Row By Degree?

Why not $\hat{L} = D^{-1}A$?



$$
\begin{array}{c}
1/4 \\ 1/3 \\ 1/5 \\ 1/4 \\ \\ . \\ . \\ . \\ . \\ \\
\end{array}
\begin{bmatrix}
4 & -1 & -1 & & -1 & -1 & & & & \\
-1 & 3 & -1 & -1 & & & & & & \\
-1 & -1 & 5 & -1 & & -1 & -1 & & & \\
 & -1 & -1 & 4 & & & -1 & & & -1 \\
-1 & & & & 3 & -1 & & -1 & & \\
-1 & & -1 & & & 4 & -1 & -1 & & \\
 & & -1 & -1 & & -1 & 6 & -1 & -1 & -1 \\
 & & & & -1 & -1 & -1 & 6 & -1 & -1 \\
 & & & & & & -1 & -1 & 3 & -1 \\
 & & & -1 & & & -1 & -1 & -1 & 4
\end{bmatrix}
$$

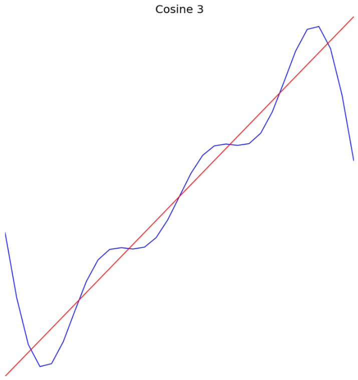
Sorkine 05

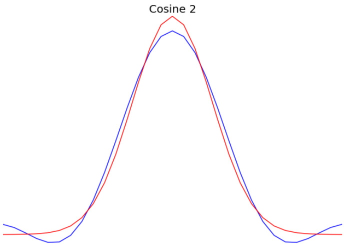
$\hat{L} \neq \hat{L}^T$ symmetry is broken..can't get real orthogonal eigenvectors

Table of Contents

Recall: Sinusoidal Decomposition

Project a function onto sinusoids, approximate with lower order sinusoids





This is “lowpass filtering”

Laplacian Eigenbasis Decomposition

$$L = D - A = USU^T$$

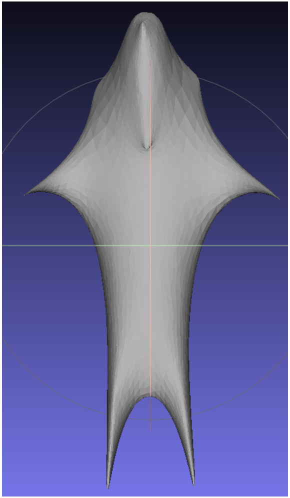
Let's project the coordinates of the mesh onto the truncated Laplacian eigen basis

$$\begin{bmatrix} | & | & | \\ x' & y' & z' \\ | & | & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_K \\ | & | & & | \end{bmatrix} \begin{bmatrix} - & u_1 & - \\ - & u_2 & - \\ \vdots & \vdots & \vdots \\ - & u_K & - \end{bmatrix} \begin{bmatrix} | & | & | \\ x & y & z \\ | & | & | \end{bmatrix}$$

$$X' = U_K U_K^T X$$

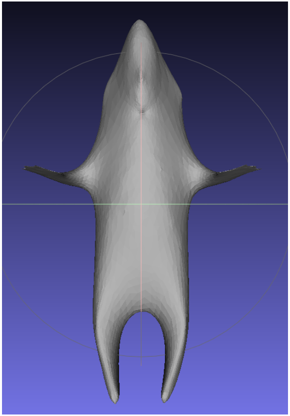
Homer: 5000 Vertices

10 Eigenvectors



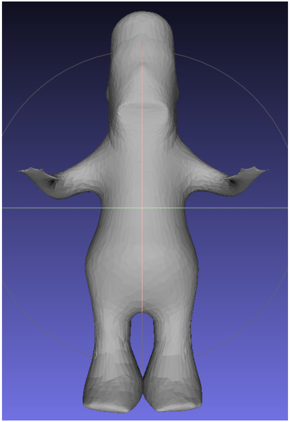
Homer: 5000 Vertices

20 Eigenvectors



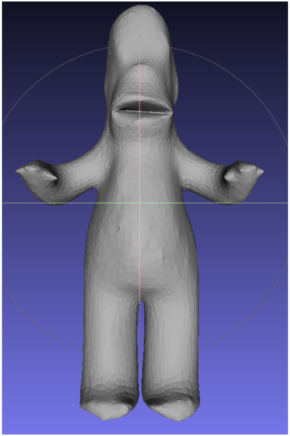
Homer: 5000 Vertices

50 Eigenvectors



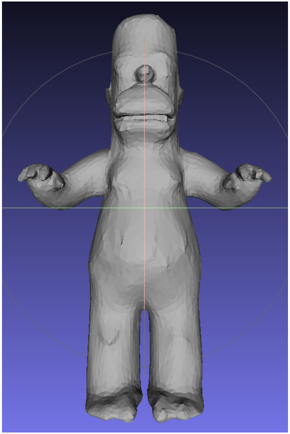
Homer: 5000 Vertices

100 Eigenvectors



Homer: 5000 Vertices

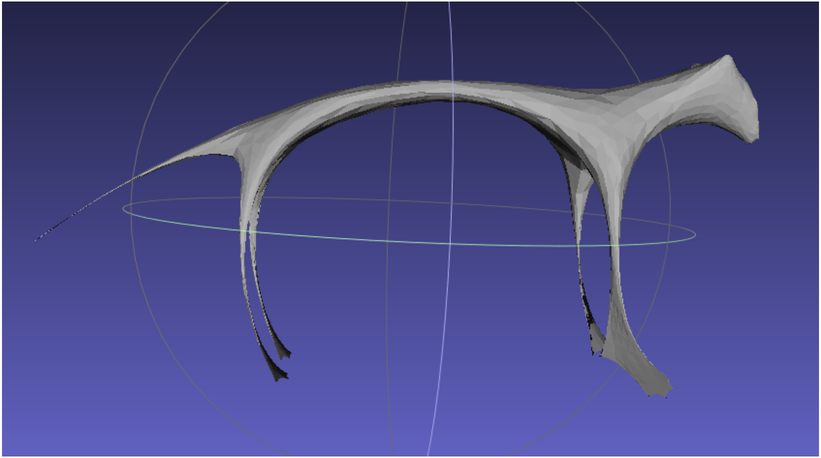
500 Eigenvectors



10x compression ratio for vertices!

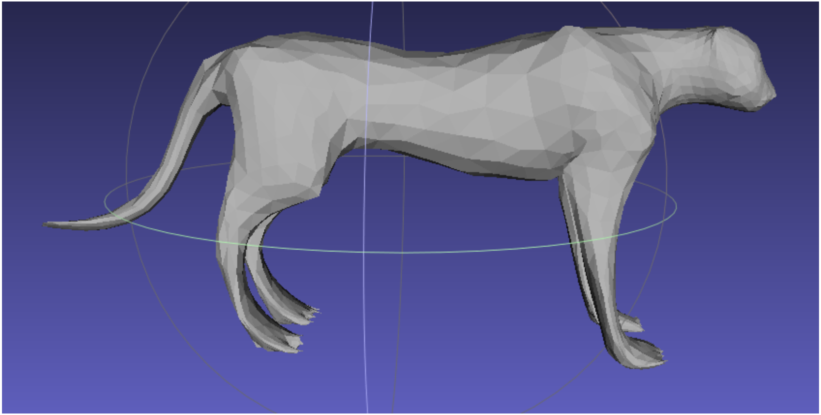
Cheetah: 2000 Vertices

20 Eigenvectors



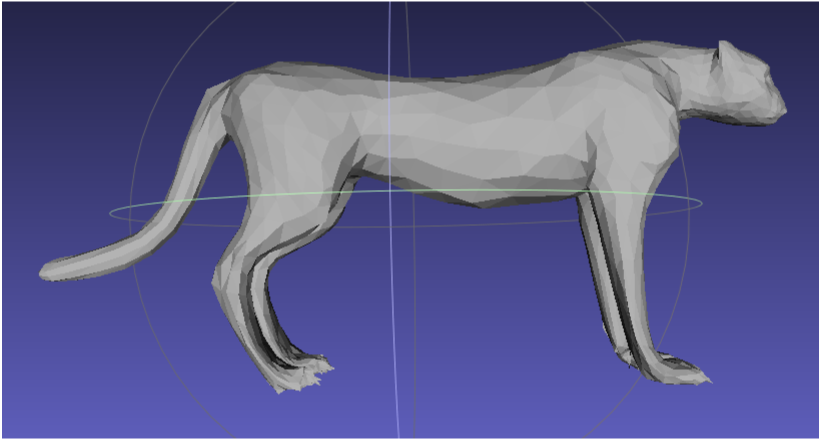
Cheetah: 2000 Vertices

100 Eigenvectors



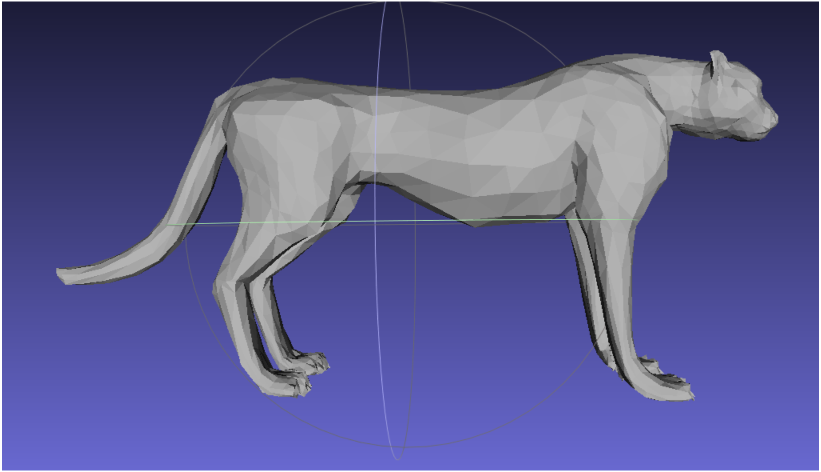
Cheetah: 2000 Vertices

200 Eigenvectors

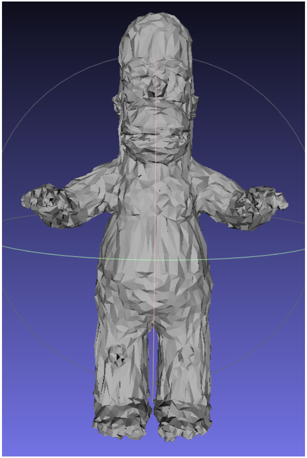


Cheetah: 2000 Vertices

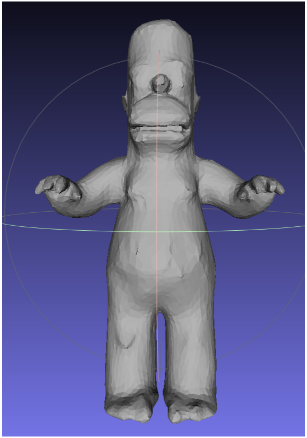
400 Eigenvectors



Can Also Use to Denoise

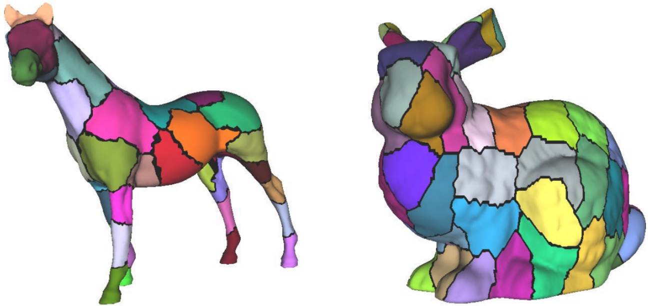


Noisy Homer



Projected Onto First 500 Eigenvectors

Compress Surface Patchwise

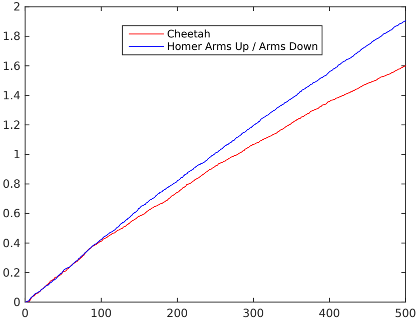


Karni 2000

- ▷ Avoids numerical stability issues, better picking up on local geometry
- ▷ Like DCT for JPEG

Shape DNA

Histogram of eigenvalues, sorted in ascending order



- ▷ Invariant to *isometries*
- ▷ Our first nonrigid shape descriptor!