Geometric Modeling

2015

Laplace Eigenfunctions
Mesh Simplification

TU Delft

Laplace Eigenfunctions

Eigenvectors & Eigenvalues

Definition:

Let $L: V \mapsto V$ be a linear map. A scalar $\lambda \in F$ is an eigenvalue of L if there is a $\mathrm{v} \in V$ with $\mathrm{v} \neq 0$, such that

$$L\mathrm{v} = \lambda\mathrm{v}.$$

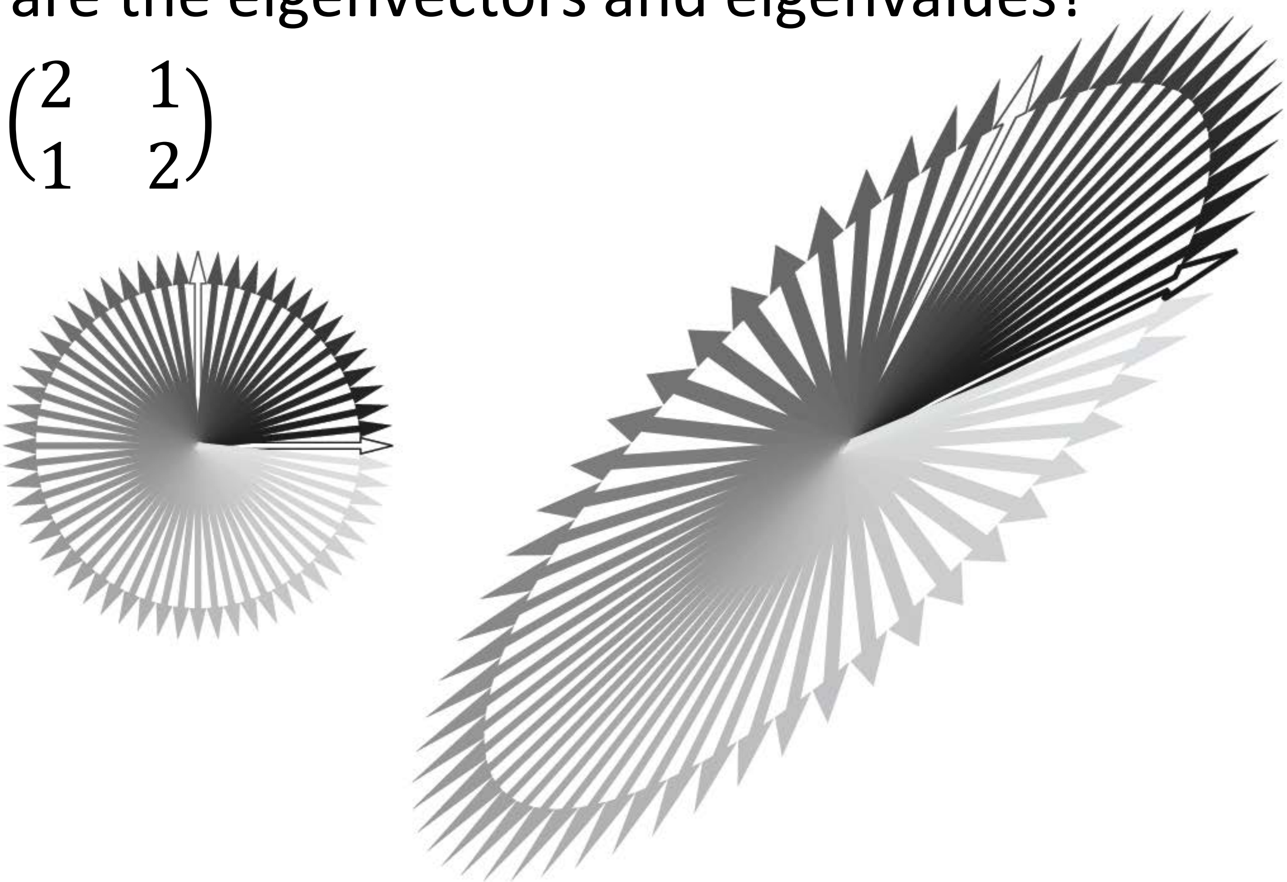
Every non-zero vector $\mathrm{v} \in V$ is with $L\mathrm{v} = \lambda\mathrm{v}$ is an eigenvector of L with eigenvalue λ.

Example

Action of a linear map

- What are the eigenvectors and eigenvalues?

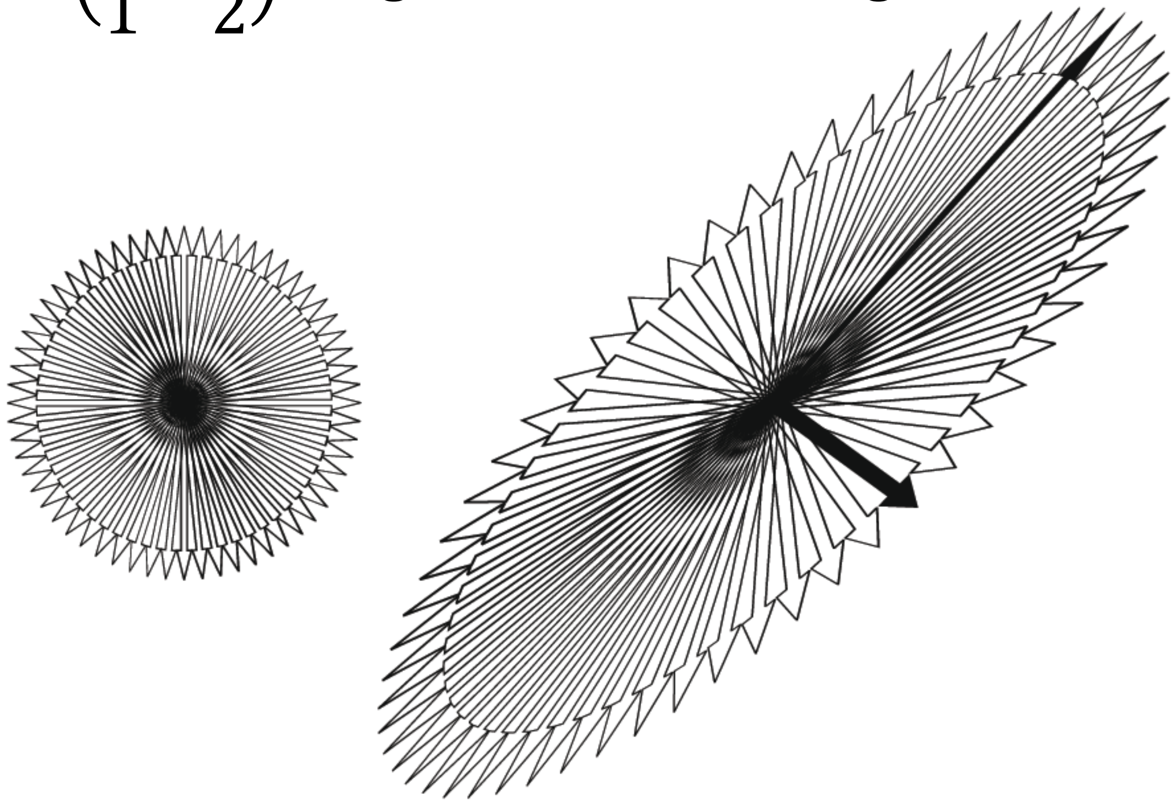
Matrix: $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$



Example

Action of a linear map

- Matrix: $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$; Eigenvalues 3, 1; Eigenvectors $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$



Eigenvectors & Eigenvalues

Diagonalization:

A linear map $L: V \mapsto V$ is called *diagonalizable* if it has dim(V) linear independent eigenvectors.

- A set of dim(V) linear independent eigenvectors forms a basis of V
- What is the matrix representation of L in this basis?

  Answer: A diagonal matrix, the diagonal entries are the corresponding eingenvalues.

Eigenvectors & Eigenvalues

Diagonalization of matrices:

In case an $n \times n$ matrix M has n linear independent eigenvectors, there exists an invertable matrix T and a diagonal matrix D such that

$$T^{-1}MT = D$$

or equivalently: $M = TDT^{-1}$.

- What is T?

Answer: Any matrix whose columns of T are linearly independent eigenvectors of M. (Multiply the equation with T from left.)

Spectral Theorem

Spectral Theorem:

Given: symmetric $n \times n$ matrix M of real numbers ($M = M^T$)

It follows: There exists an *orthogonal* set of n eigenvectors.

This implies:

Every (real) symmetric matrix can be *diagonalized*:

$M = TDT^T$ with an orthogonal matrix T, diagonal matrix D.

Examples

Illustration:

$$M = TDT^T$$

T is an orthogonal matrix, hence if det(T)=1 it is a rotation.

Remark: The property det(T)=1 can be obtained by changing the order of the basis vectors that form the columns of T.

Examples

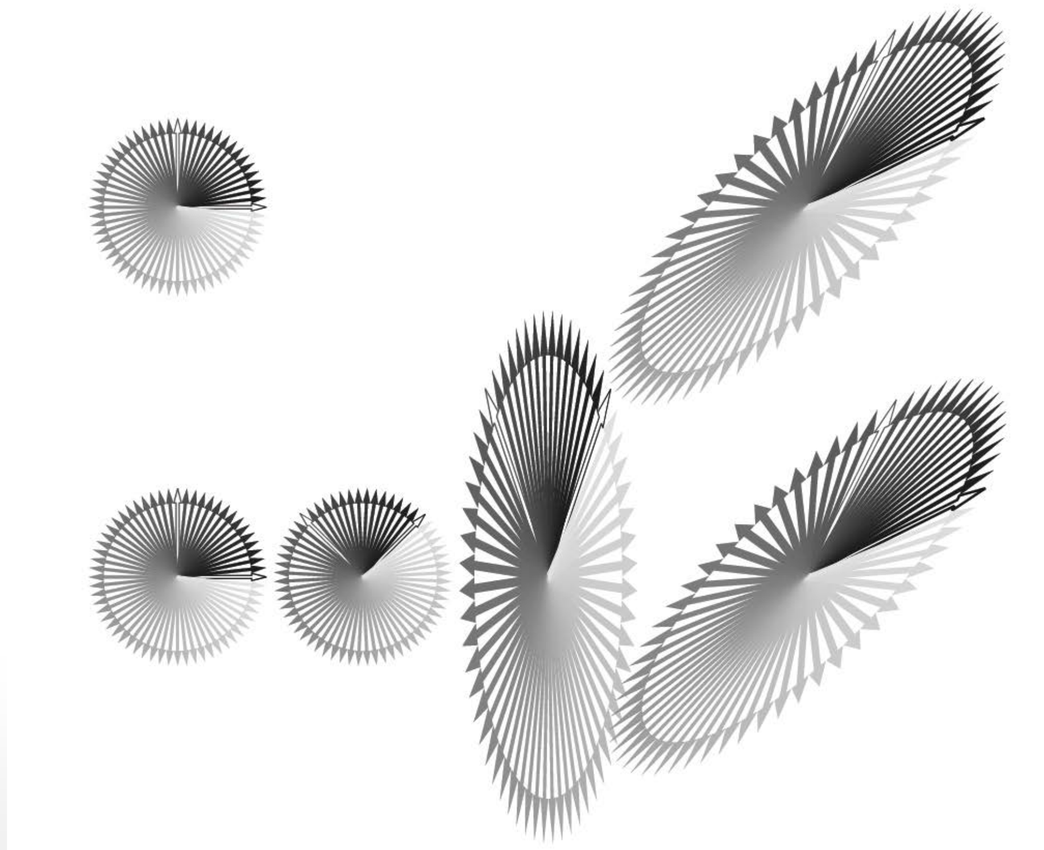
Illustration:

$$M = TDT^T$$

T is an orthogonal matrix, hence if det(T)=1 it is a rotation.

Apply M:

Rotate, apply diagonal matrix, rotate back:



Examples

Vibration modes

- Consider the space of all functions that are in $C^{\infty}([0,\pi],\mathbb{R})$ and vanish at 0 and π and the linear operator $\frac{\partial^2}{\partial x^2}$.

Any idea what could be eigenfunctions?

Examples

Vibration modes

On the space of all functions that are in $C^{\infty}([0,\pi],\mathbb{R})$ and vanish at 0 and π, the functions

$$\{\sin(x),\sin(2x),\sin(3x)\ ,\ldots\}$$

are eigenfunctions of the linear operator $\frac{\partial^2}{\partial x^2}$ with eigenvalues

$$\{-1,-4,-9,\ldots\}.$$

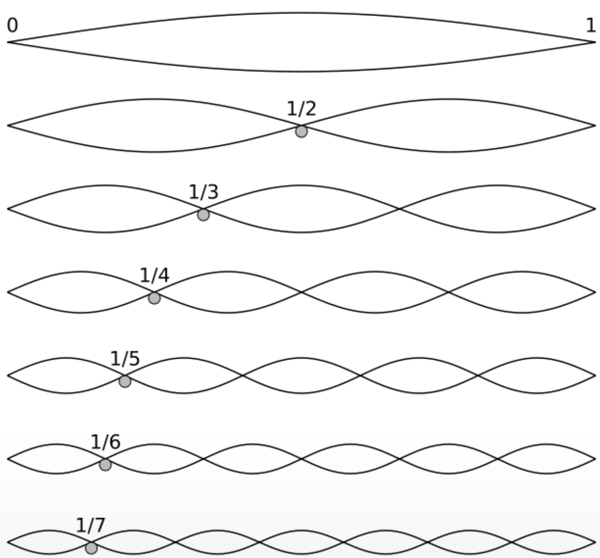
A simple calculation shows:

$$\frac{\partial^2}{\partial x^2}\sin(n\,x) = -n^2\,\sin(n\,x)$$

Linear operator

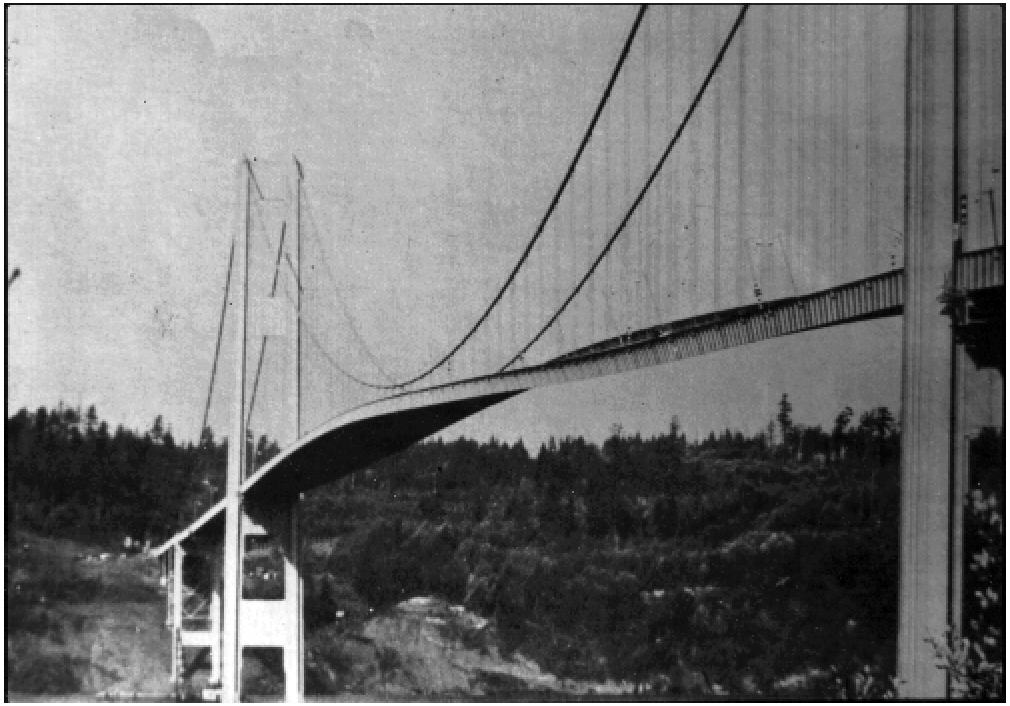
Function/ vector

Eigenvalue



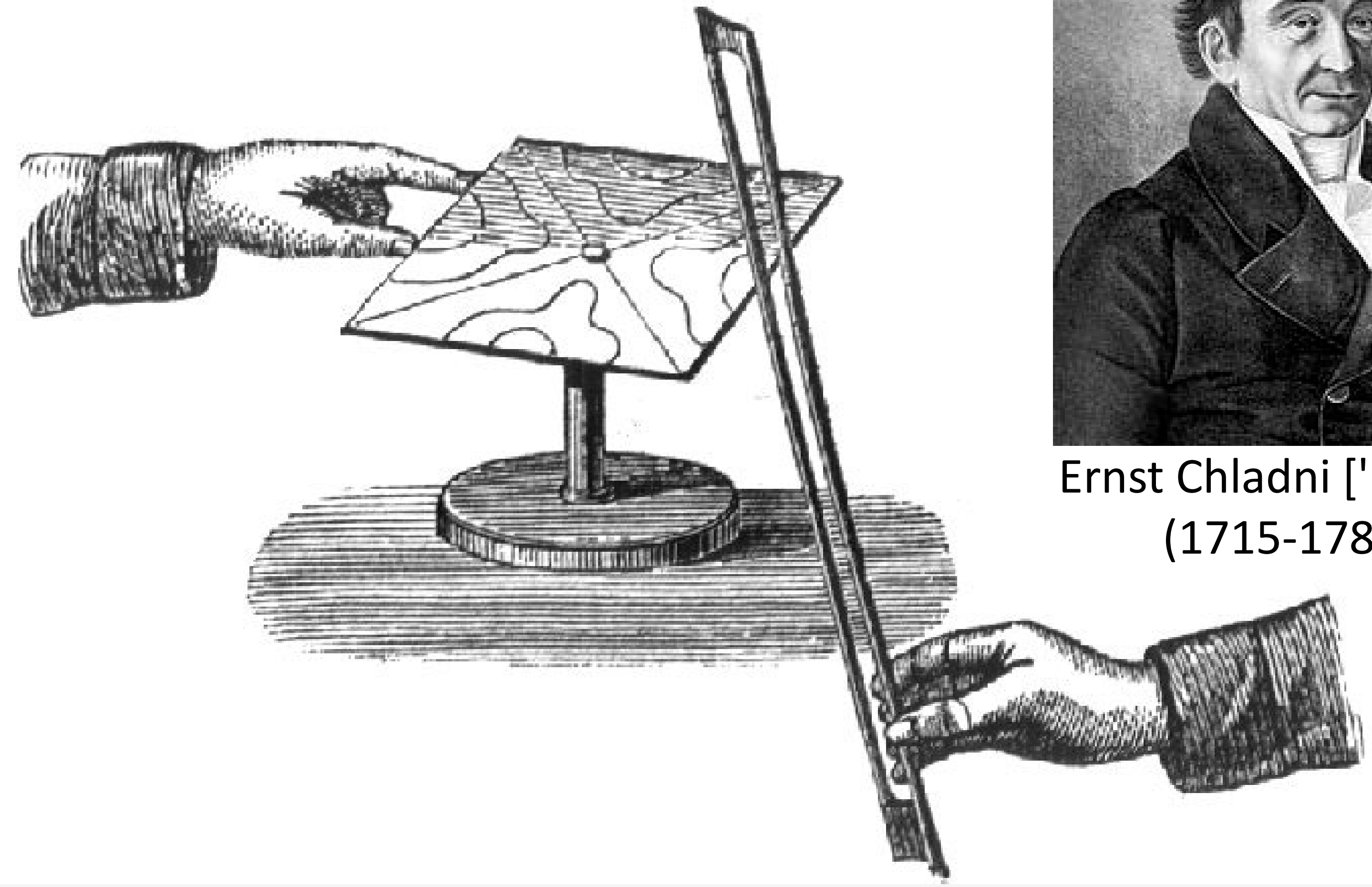
Vibrating string

Vibration modes



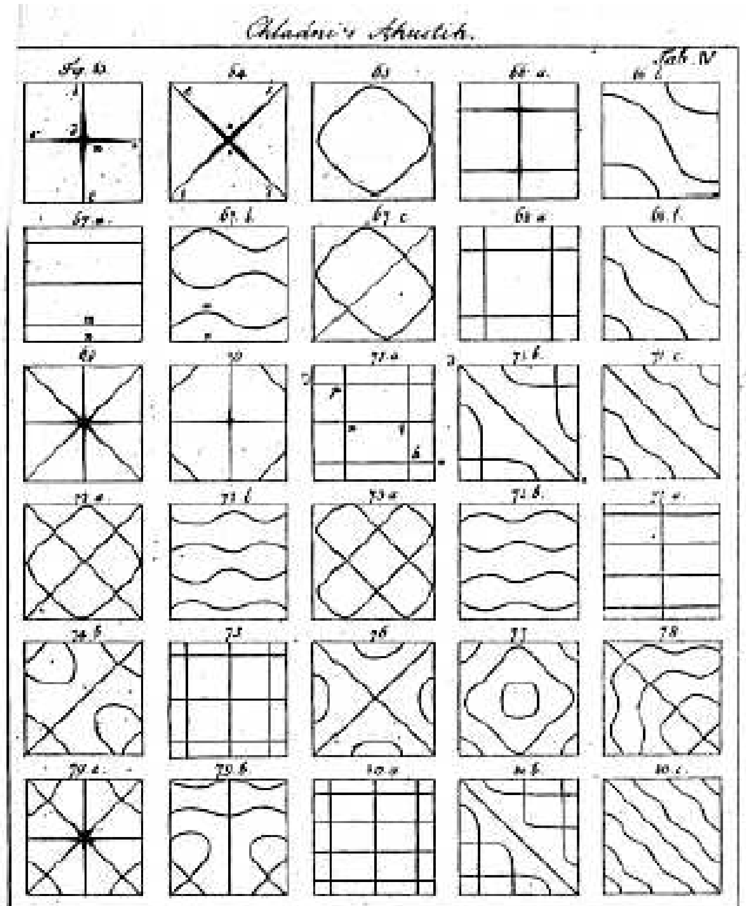
Tacoma Narrows Bridge on Nov. 7, 1940

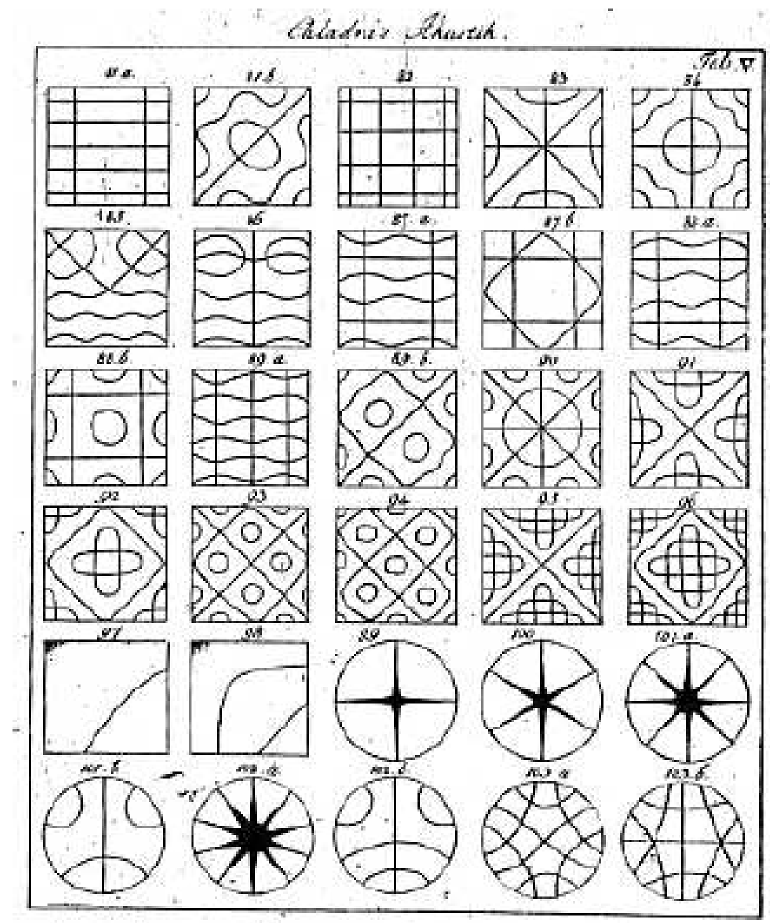
Chladni Plates



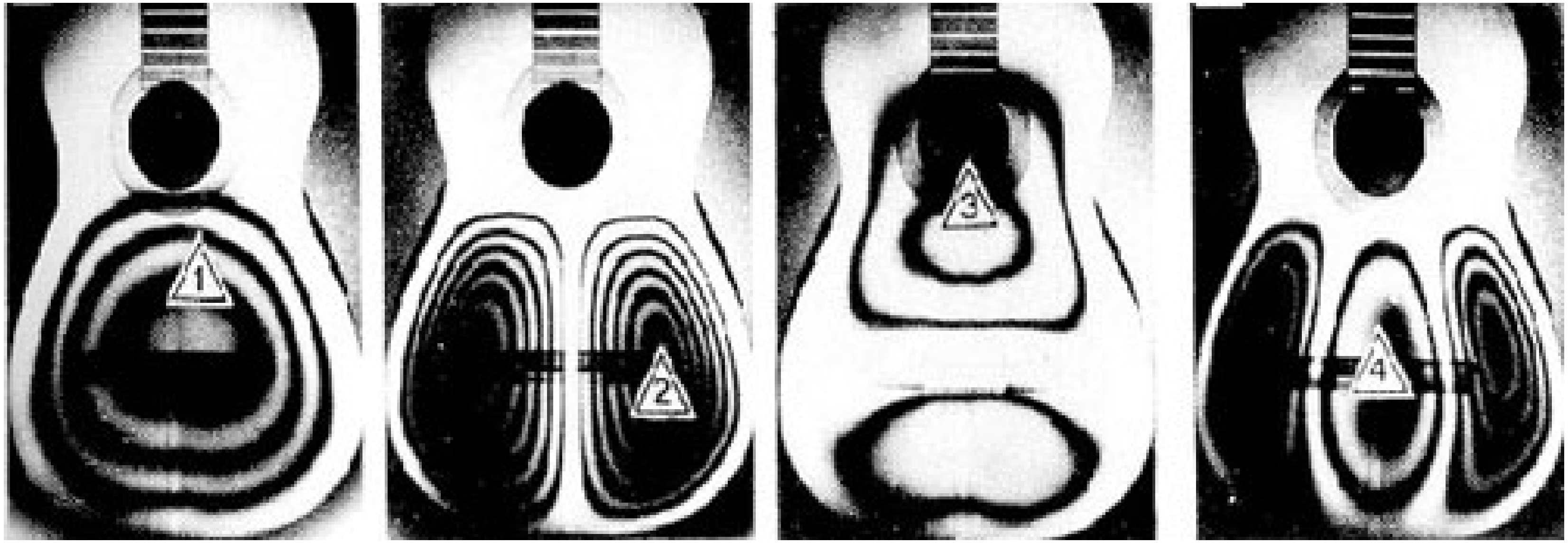
Ernst Chladni ['kladnı]
(1715-1782)

Chladni Plates





Chladni Plates



Laplace Eigenfunctions on Meshes

Eigenfunction

- The eigenvalues and –functions of the discrete Laplace operator on a triangle mesh are solutions of the generalized eigenproblem

$$S\Phi_i = \lambda_i M \Phi_i$$

Some Properties

- The eigenvalues are non-negative
- The eigenfunctions Φ_i are M-orthogonal:

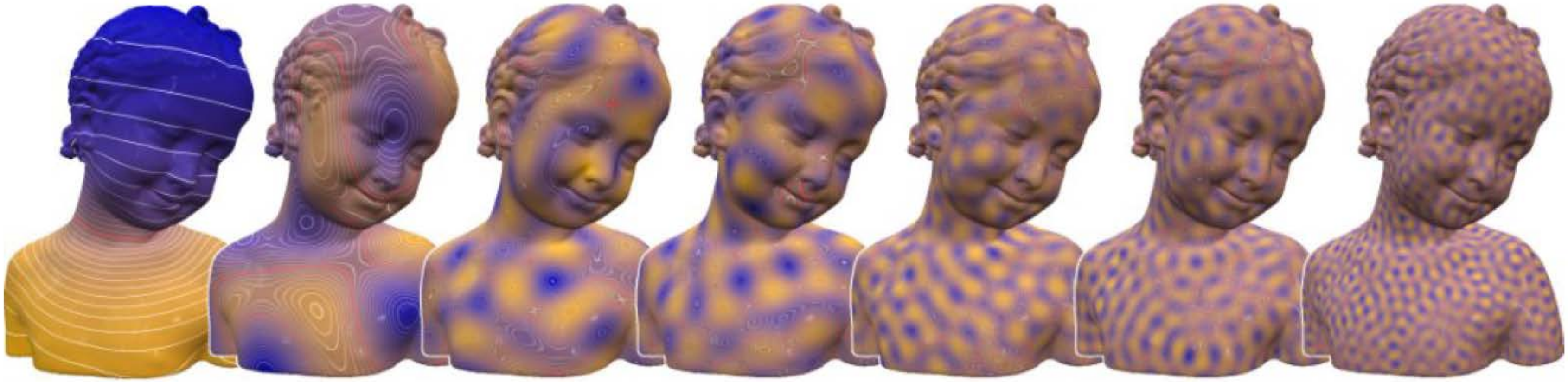
$$\Phi_i{}^T M \Phi_j = \delta_{ij}$$

- The eigenvalues and -functions are invariant under isometric deformations of the surface

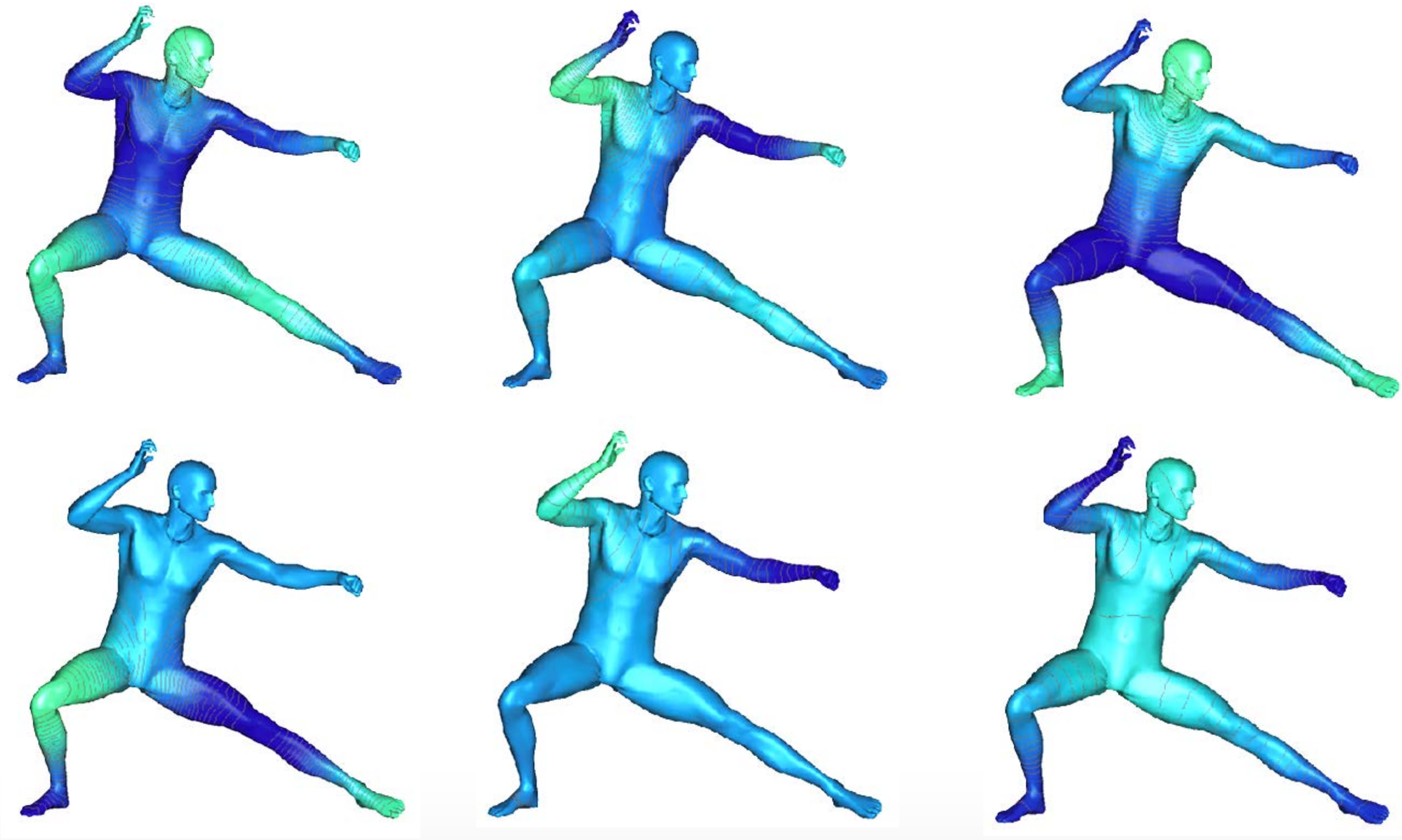
Computation:

- Sparse eigensolver (e.g. Arpack)

Examples

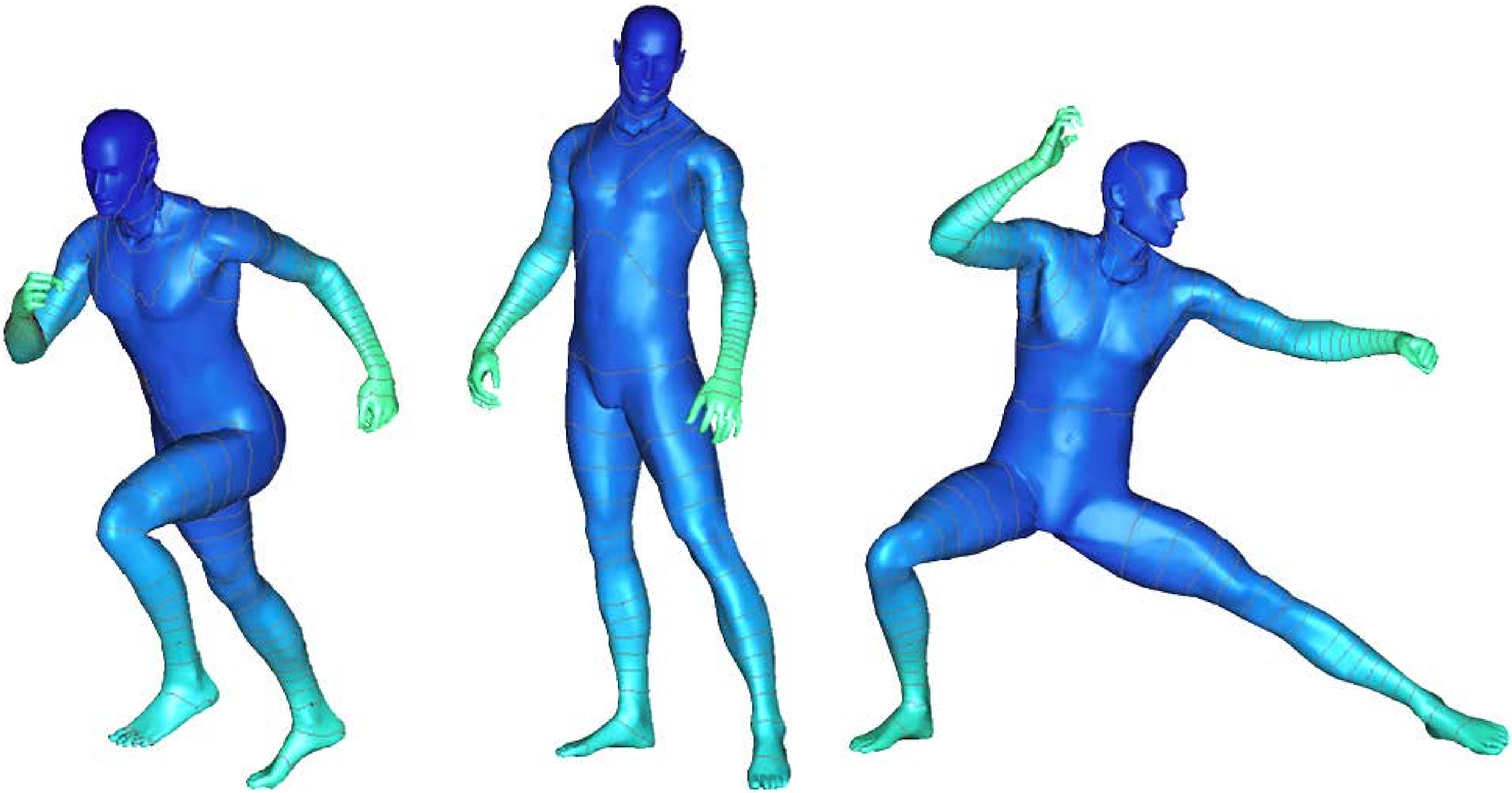


Examples



The first eigenfunctions of the Laplace-Beltrami operator

Invariance to Isometries



An eigenfunction of the Laplace-Beltrami operator computed on different deformations of the shape, showing the invariance of the Laplace-Beltrami operator to isometries

Chladni Plates - Simulations

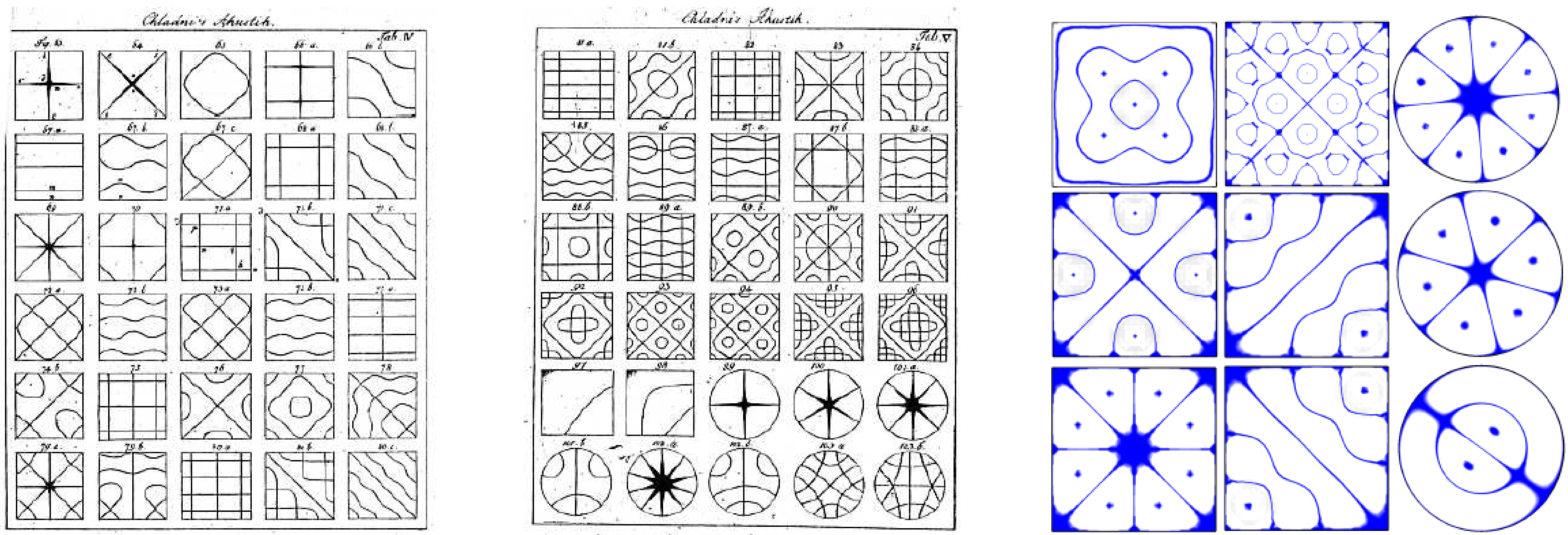


Figure 10: Left: In the late 1700's, the physicist Ernst Chladni was amazed by the patterns formed by sand on vibrating metal plates. Right: numerical simulations obtained with a discretized Laplacian.

Function Filtering

Eigenbasis

- A discrete function x can be written in the eigenbasis

$$x = \sum_{i=0}^{n} (x^T M \, \Phi_i) \Phi_i$$

Filtering

- Assign a weight w_i to every eigenfunction, then the filtered function is

$$\hat{x} = \sum_{i=0}^{n} w_i (x^T M \Phi_i) \Phi_i$$

- Usually only the k lowest eigenfunctions are used

Mesh Filtering

Mesh filtering

- The filter can be applied to the embedding of the surface

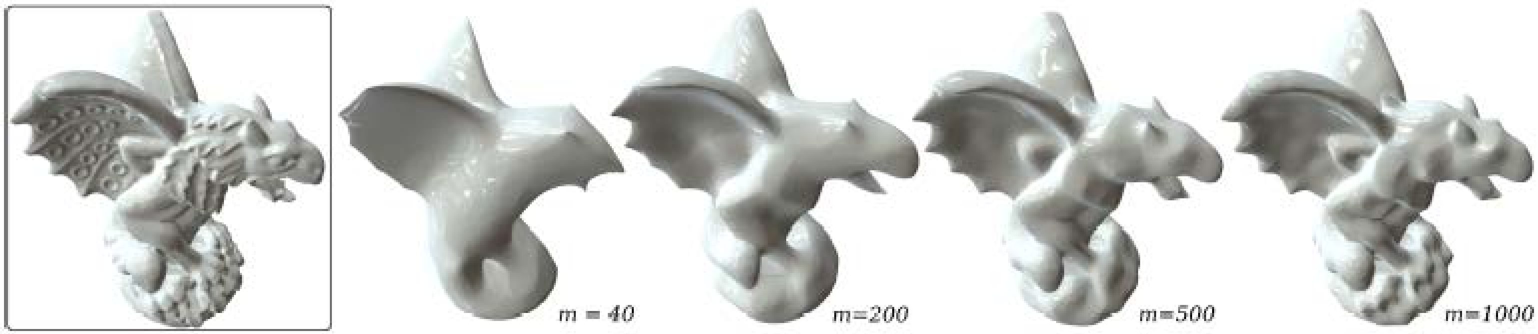
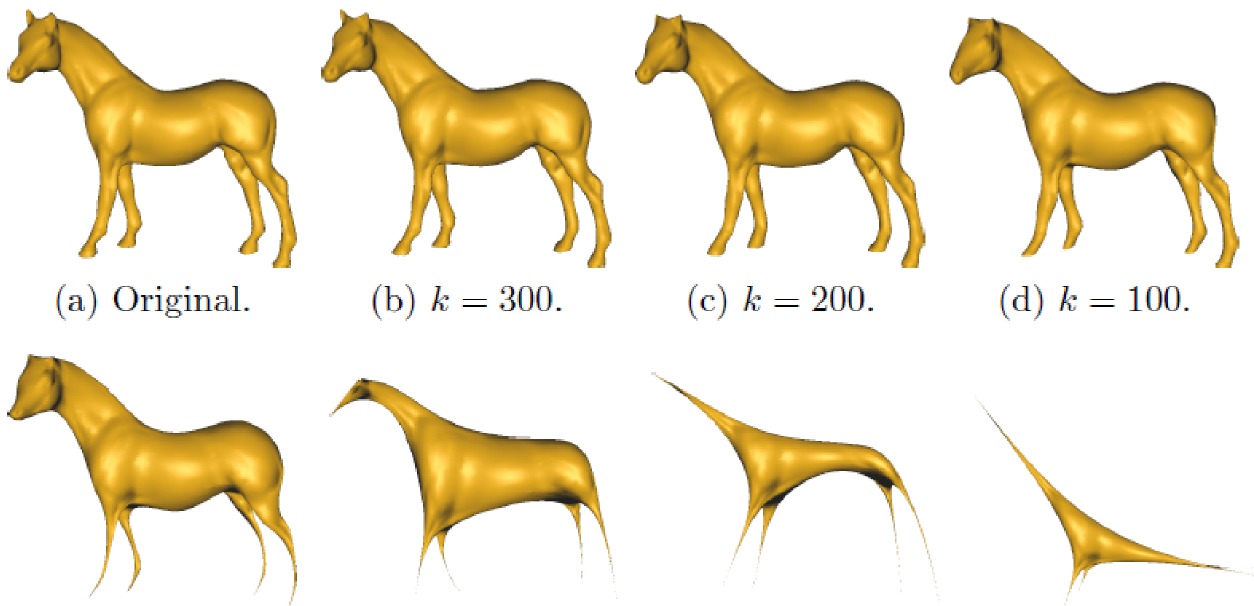


Figure 15: Reconstructions obtained with an increasing number of eigenfunctions.

Mesh Filtering

Mesh filtering

- The filter can be applied to the embedding of the surface

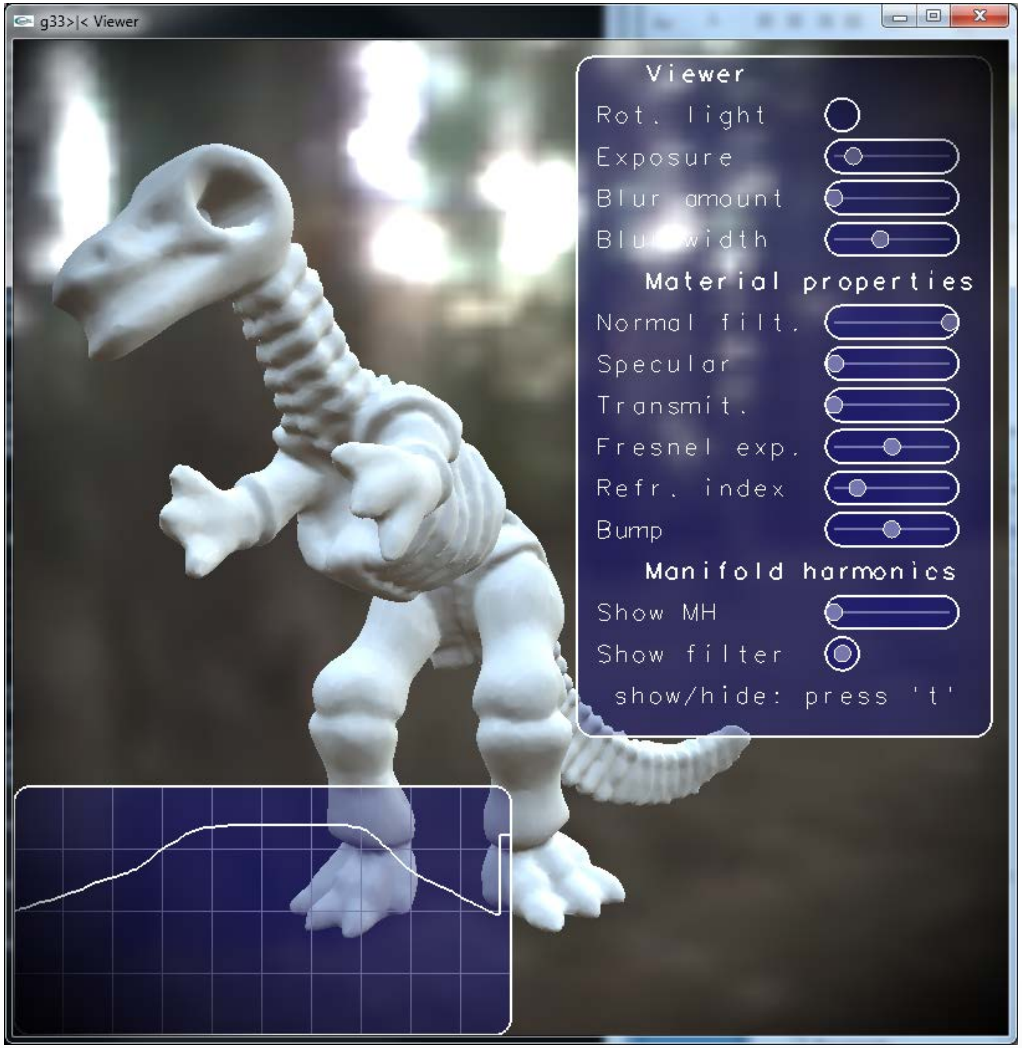


(a) Original. (b) $k = 300$. (c) $k = 200$. (d) $k = 100$.

(e) $k = 50$. (f) $k = 10$. (g) $k = 5$. (h) $k = 3$.

Figure 8: Shape reconstruction based on spectral analysis using a typical mesh Laplace operator, where k is the number of eigenvectors or spectral coefficients used. The original model has 7,502 vertices and 15,000 faces.

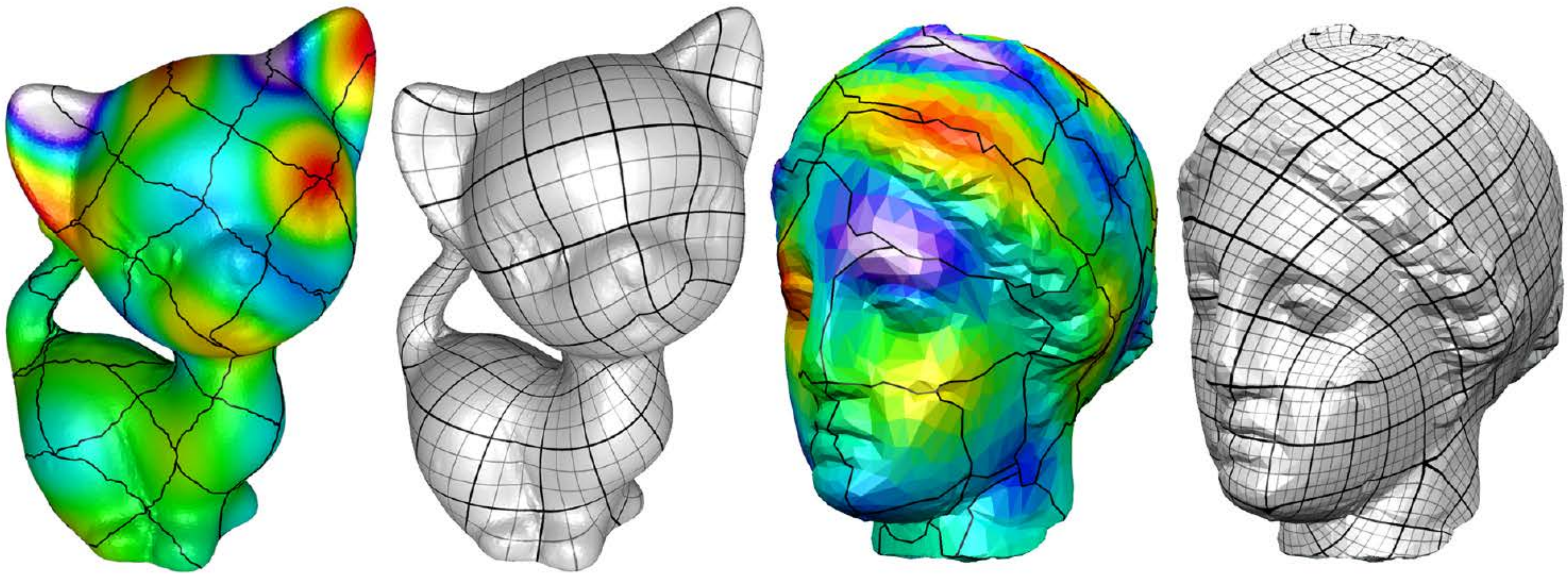
Mesh Filtering



Meshing

Quadrangulation

- Morse-Smale complex of an eigenfunction is used for quad-meshing



Global Point Signature (GPS)

Goal

- Find a good, isometry-invariant shape descriptor

Idea

- For every point p define the Global Point Signature

$$GPS(\mathbf{p}) = \left(\frac{1}{\sqrt{\lambda_1}}\phi_1(\mathbf{p}), \frac{1}{\sqrt{\lambda_2}}\phi_2(\mathbf{p}), \frac{1}{\sqrt{\lambda_3}}\phi_3(\mathbf{p}) \ldots \right)$$

- GPS is a mapping of the surface onto an infinite dimensional space. Each point gets a signature.

Global Point Signature (GPS)

Properties of GPS:

- If $p \neq q, GPS(p) \neq GPS(q)$.
- GPS is isometry invariant (since Laplace-Beltrami is)
- Given all eigenfunctions and eigenvalues, one can recover the shape up to isometry (not true if only eigenvalues are known).
- Euclidean distances in the GPS embedding are meaningful

Global Point Signature (GPS)

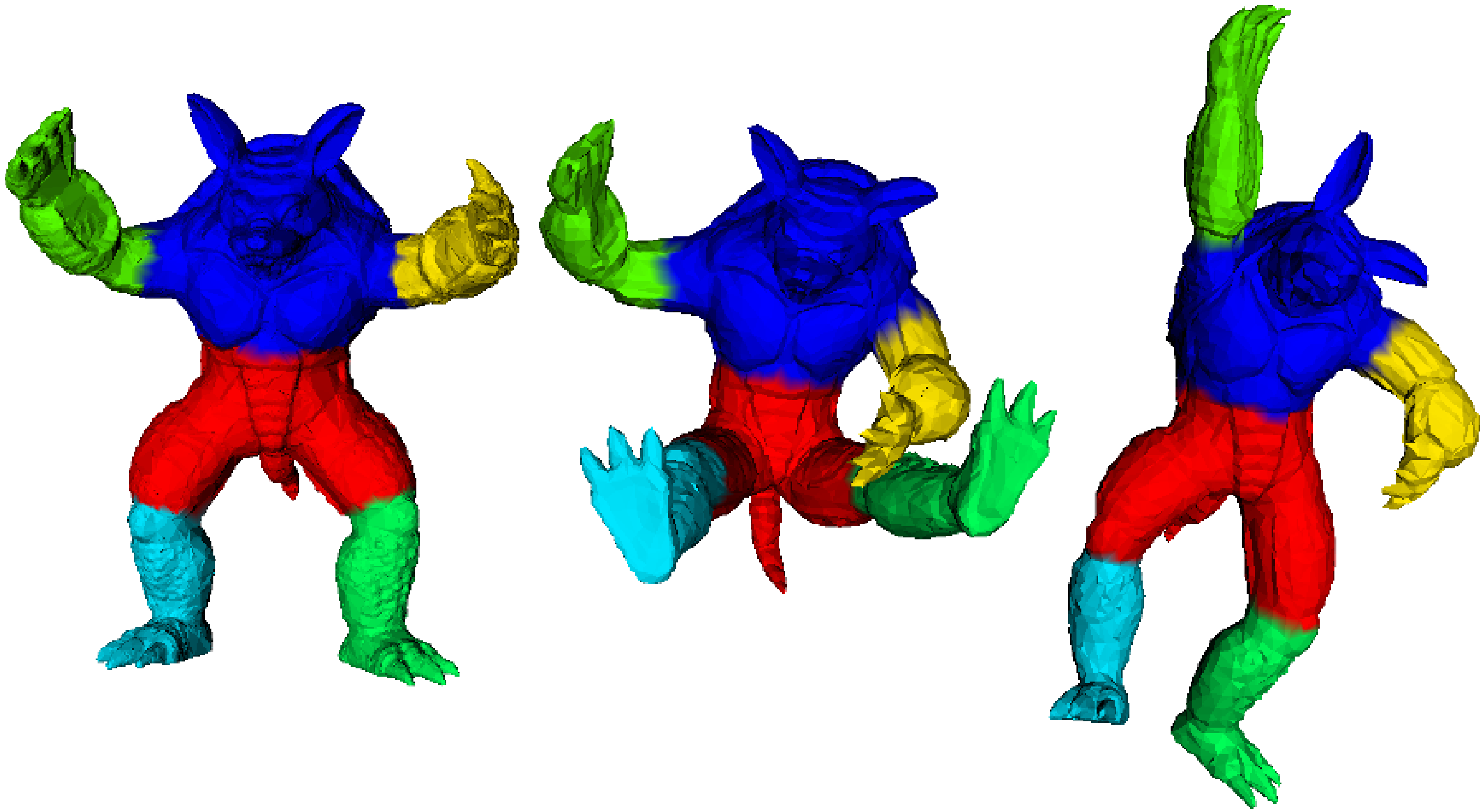
Properties of GPS:

- If $p \neq q, GPS(p) \neq GPS(q)$.
- GPS is isometry invariant (since Laplace-Beltrami is)
- Given all eigenfunctions and eigenvalues, one can recover the shape up to isometry (not true if only eigenvalues are known).
- Euclidean distances in the GPS embedding are meaningful

Global Point Signature (GPS)

Euclidean distances in the GPS embedding

- K-means done on the embedding provides a segmentation



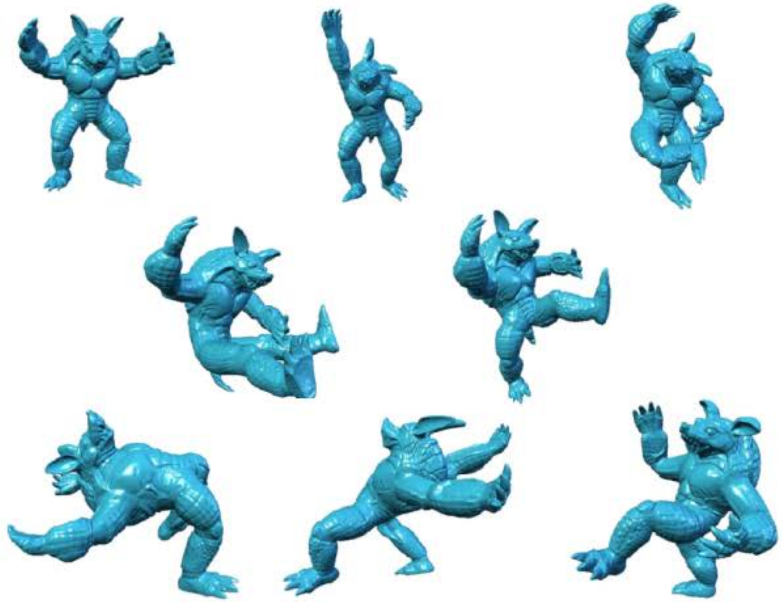
Global Point Signature (GPS)

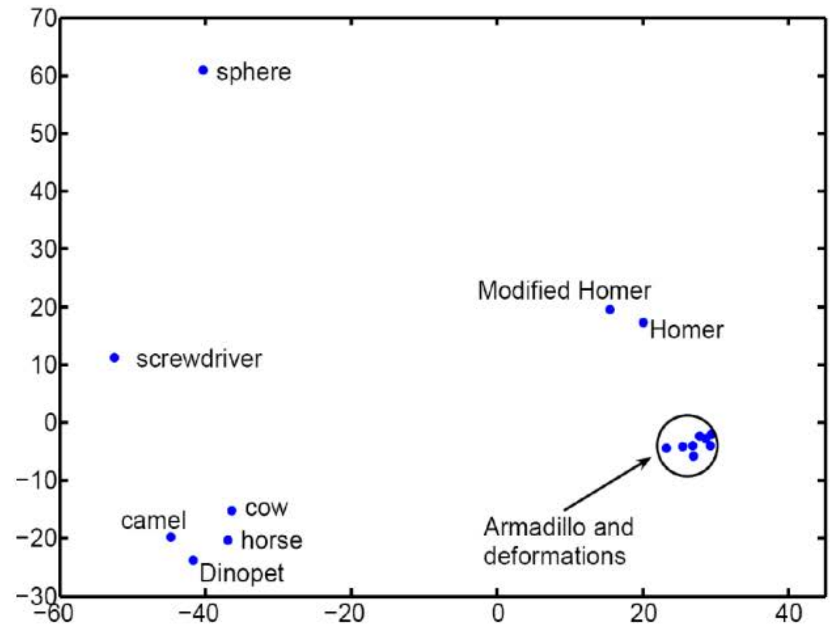
Comparing GPS

- Given a shape, determine its GPS embedding
- Construct a histogram of pairwise GPS distances (note that GPS is defined up to sign flips, distances are preserved)
- For any 2 shapes, compute the difference between their histograms
- For refined comparisons use more than one histogram

Global Point Signature (GPS)

Results

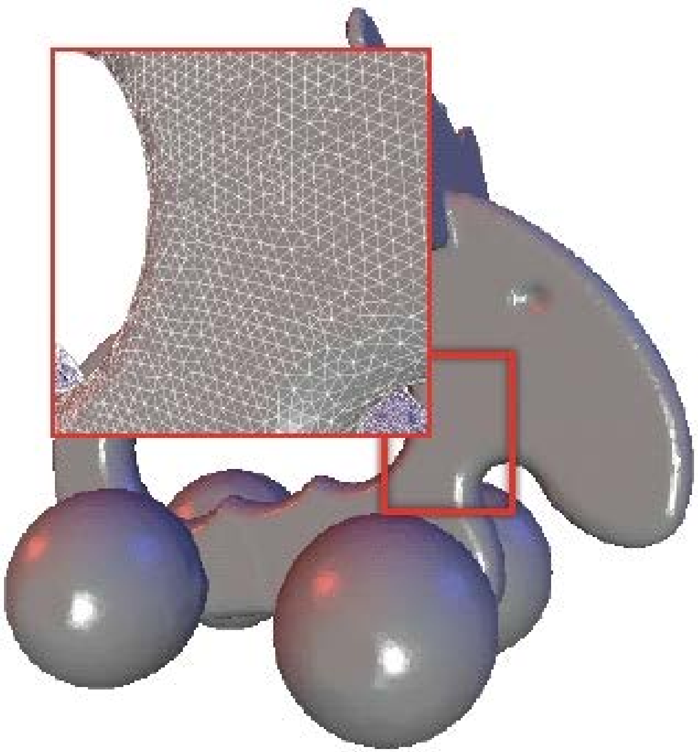




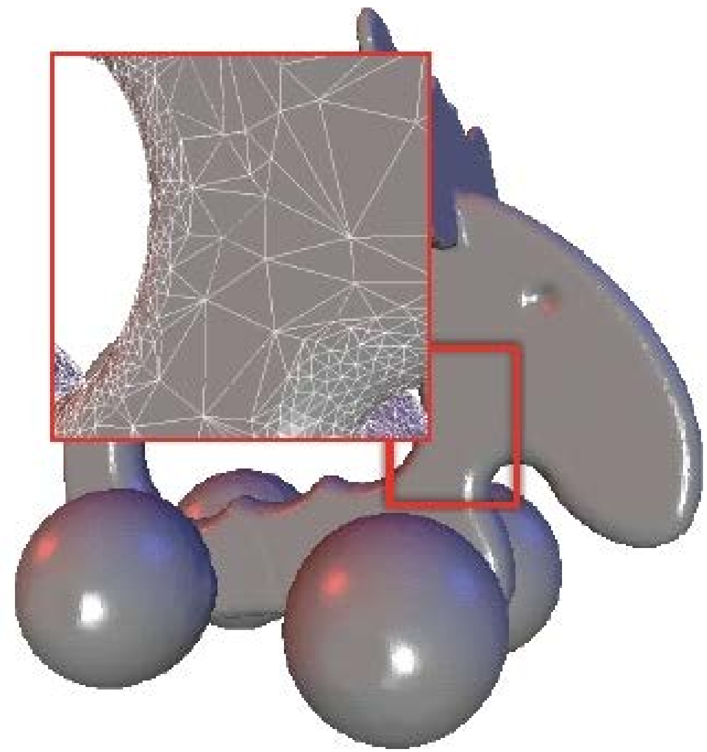
Mesh Simplification

Motivation

Oversampled 3D scan data



~150k triangles

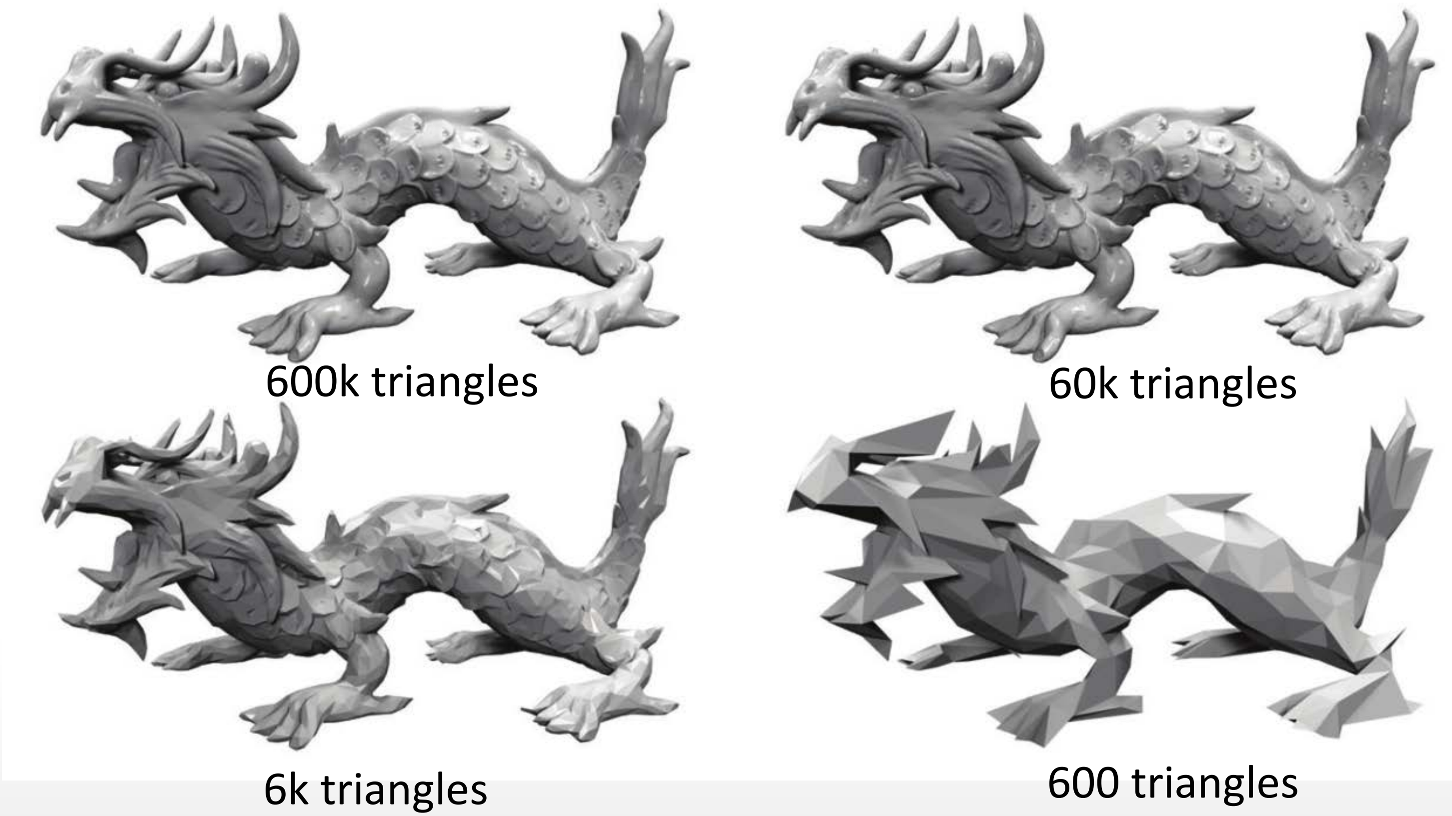


~80k triangles

Motivation

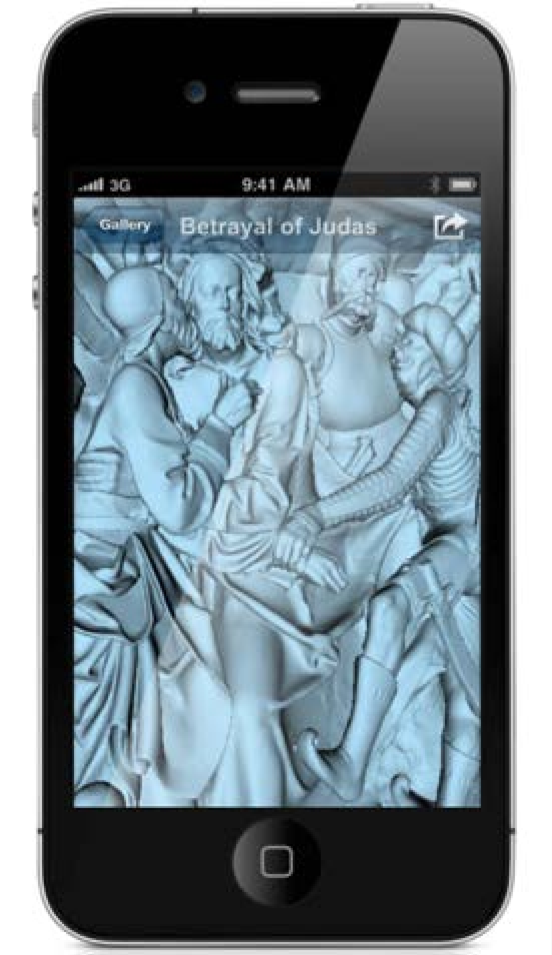
Multiresolution hierarchy

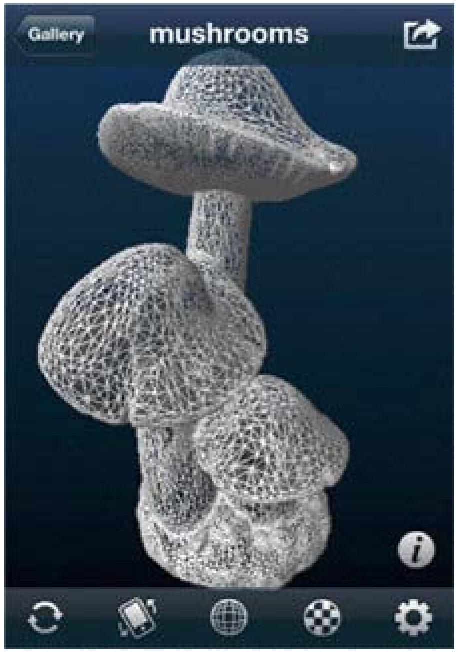
- Efficient goemetry processing
- Level-of-detail rendering



Motivation

Adaptation to hardware capabilities

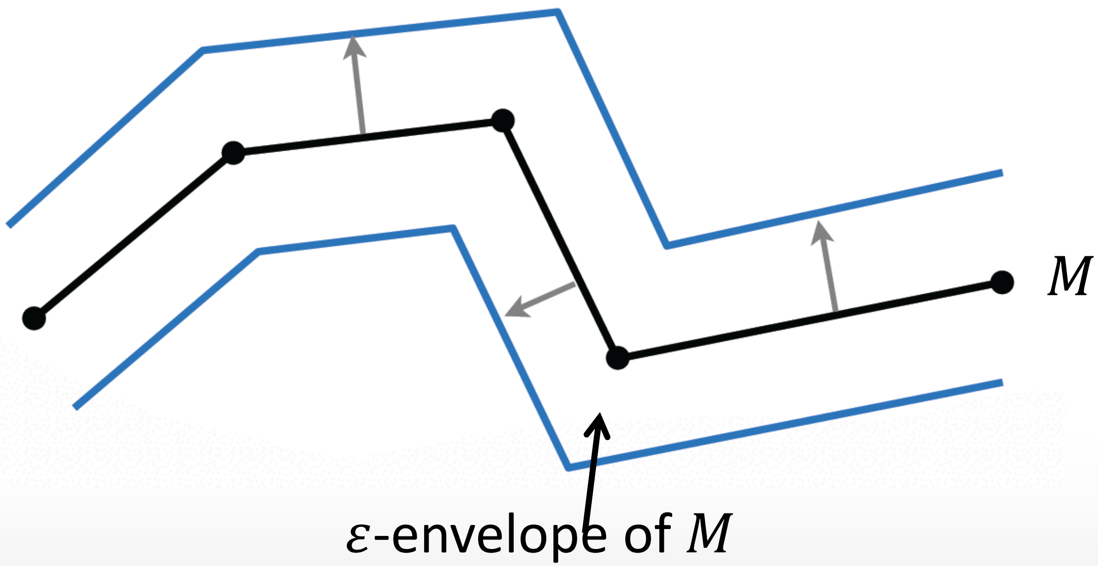




Envelope

Problem Statement

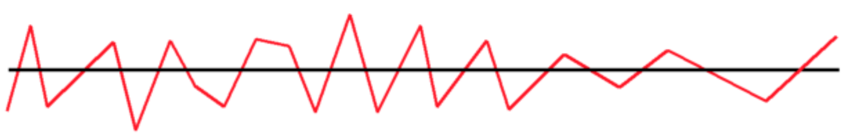
- Input: mesh $M = (V, F)$ and $\varepsilon \in \mathbb{R}_+$
- Output: $\widetilde{M} = (\tilde{V}, \tilde{F})$ such that $|\tilde{V}|$ is minimal and $\widetilde{M}$ is in the ε-envelope of M



Mesh Simplification

Approximation algorithms:

- Polynomial time approximation algorithms with strict error guarantees are known, but they are too slow for practical applications
- Does not take derivatives into account



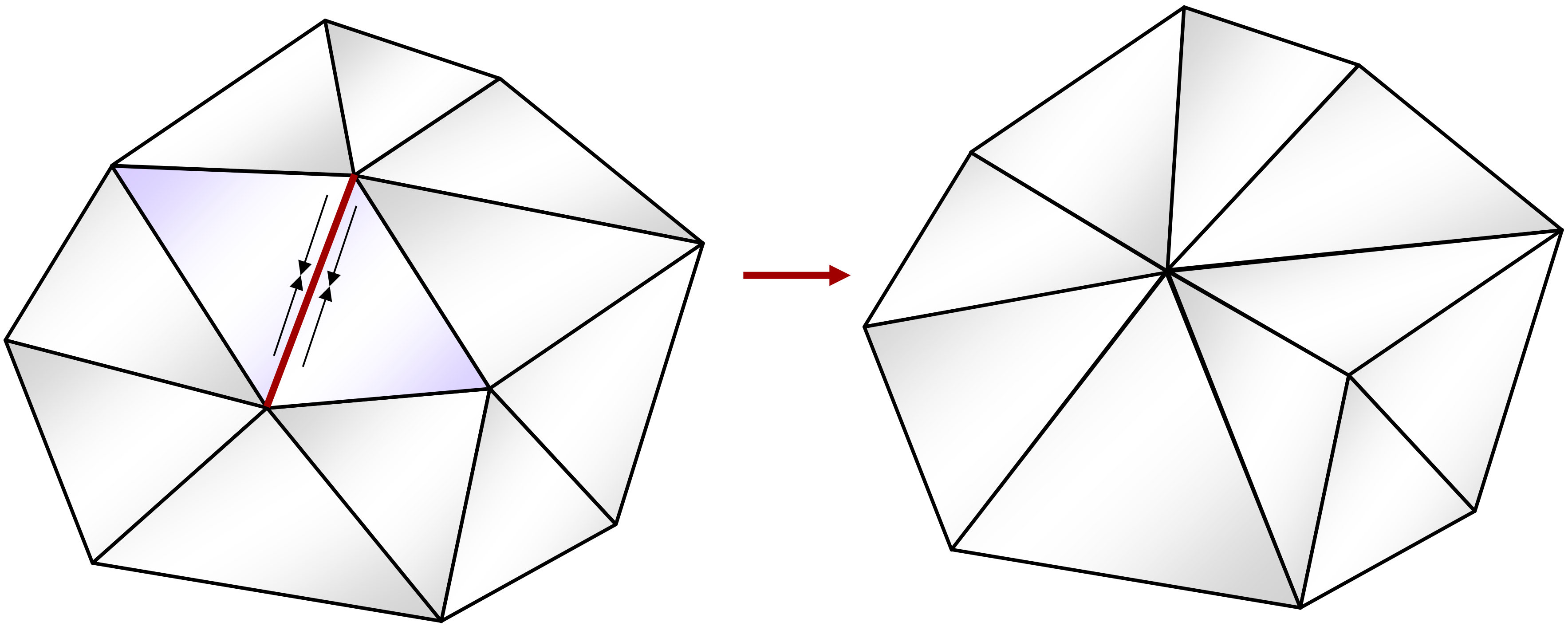
Agenda:

- Heuristics for the rescue!
- Practical performance is still good

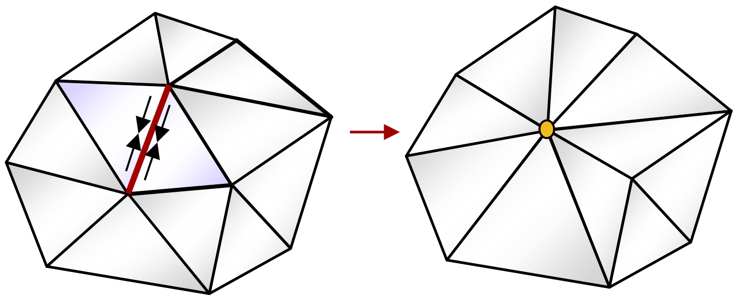
Michelangelo's St. Matthew
386,488,573 triangles
[Stanford Digital Michelangelo Project]

Simplification Algorithm

Edge contraction:



Edge Contraction



Edge contraction algorithm:

- Questions:
  - Which edges can be contracted?
  - Edges contract into points –
    where should we place the new point?
  - What is the best order for edge contractions?
- Standard algorithm:
  - Greedy algorithm
  - Put edges in priority queue
  - Pick the “cheapest” edge and remove it
  - Recompute costs

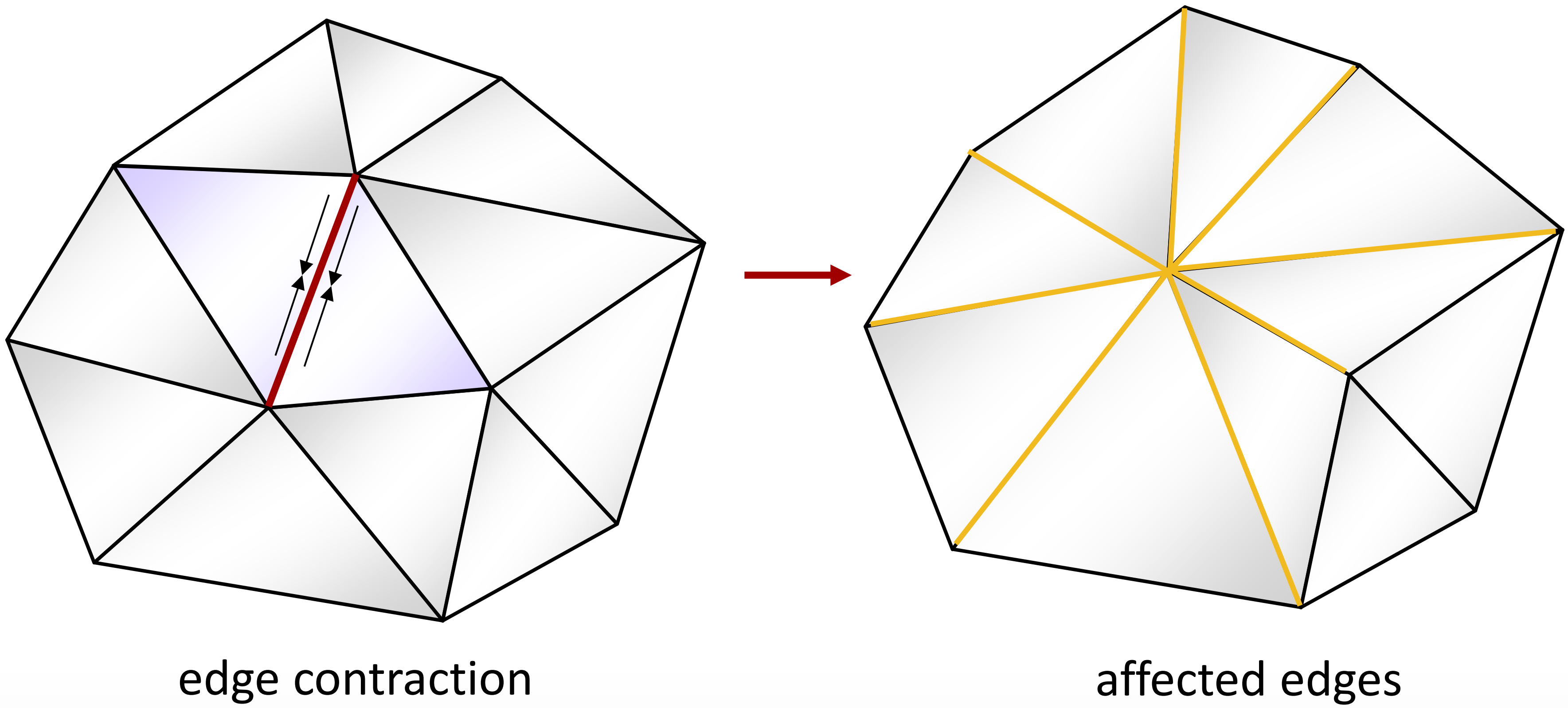
Edge Contraction

Algorithm:

- For each edge in the mesh, compute the cost of contracting the edge
  - If an edge contraction changes the topology, set costs to $+\infty$
  - Put all (finite cost) edges in priority queue sorted by cost
- **While** queue not empty **and** result not simple enough
  - Remove min-cost edge
  - Contract the edge
  - Recompute costs of all affected edges (incl. topology check)
  - Update the priority queue accordingly

Edge Contraction

Affected edges:



Components

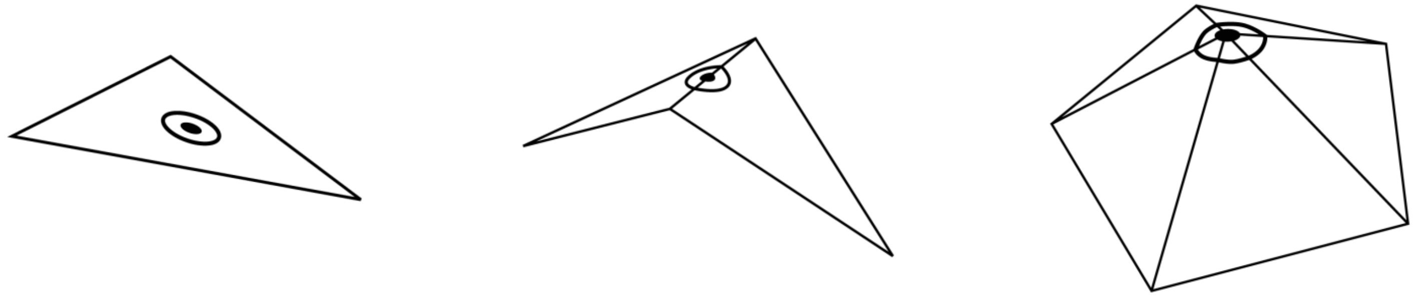
The algorithm needs the following components:

- Topology check (mostly fixed)
- Costs for edge contractions (lots of choices)
- Placement of new vertices (lots of choices)

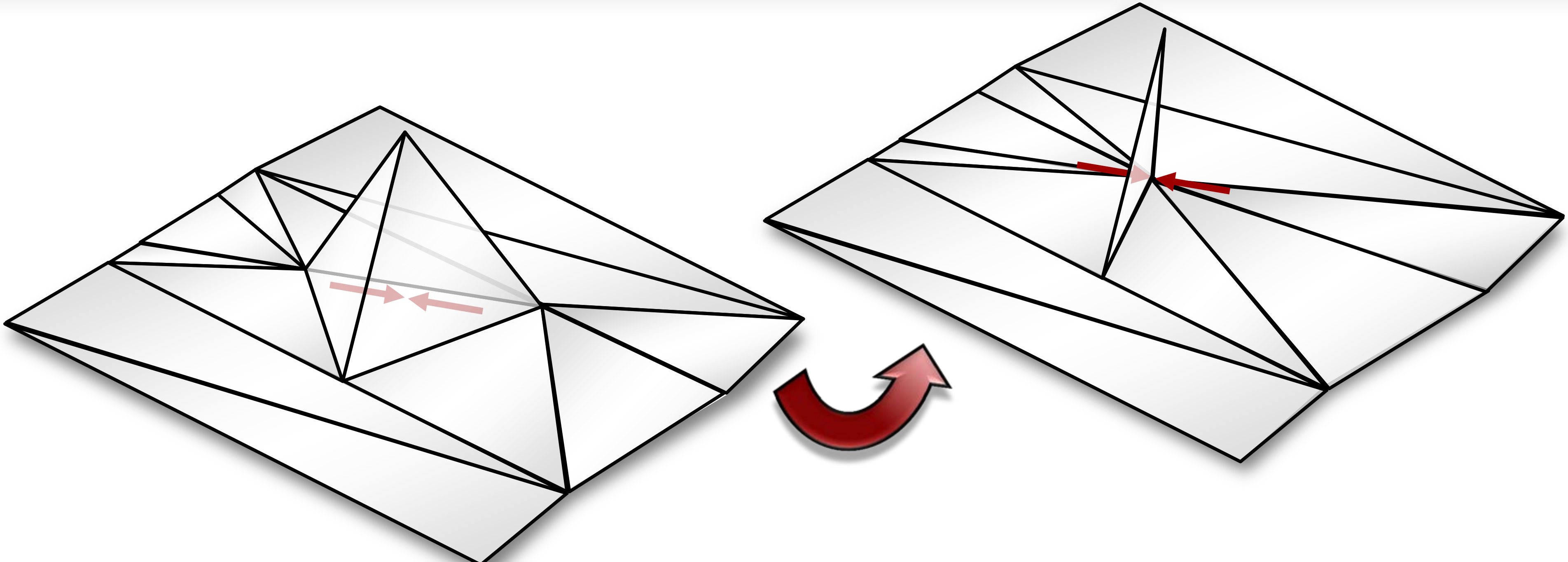
Topology Check

We do not want to change the topology of the mesh

- Input is a triangulated two-manifold, probably with boundary
- This means:
  - All points are locally disks or on a boundary



Problem #1: Folds



Problem #1:

- Edge contraction can cause topological folds in meshes
- We need a criterion to prevent this

Valid Edge Contractions

Valid Edge Contraction

An edge contraction on a triangular surface mesh is valid if after contraction the mesh still describes a surface

Criterion:

Contracting an edge (p, q) is a valid operation if and only if the following two criteria hold:

- If both p and q are boundary vertices, then the edge (p, q) has to be a boundary edge.
- For all vertices r incident to both p and q there has to be a triangle (p, q, r). In other words, the intersection of the one-rings of p and q consists of vertices opposite the edge (p, q) only.

For a proof, see "Geometry and Topology of Mesh Generation" H. Edelsbrunner, 2001, Cambridge University Press (pdf available on the authors webpage)

Illustration

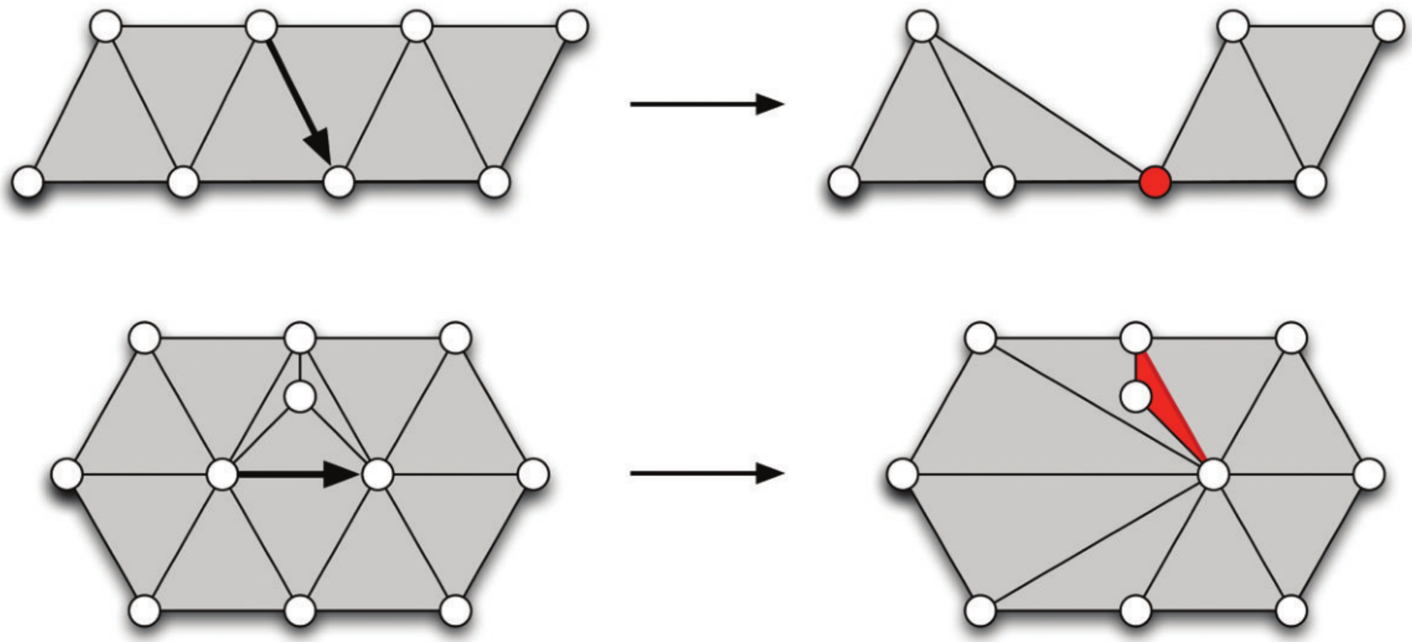
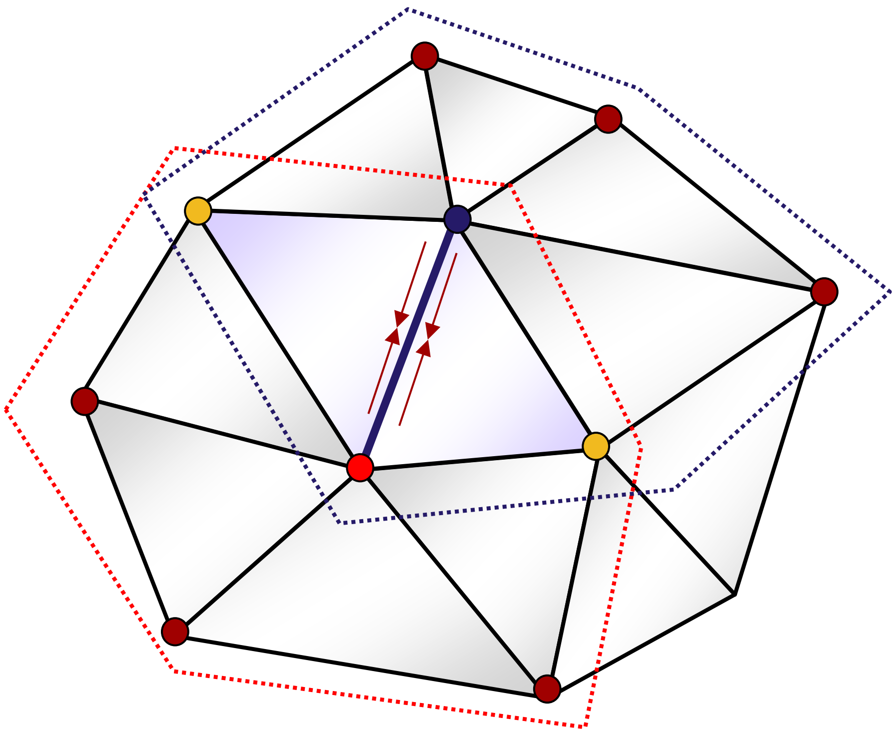


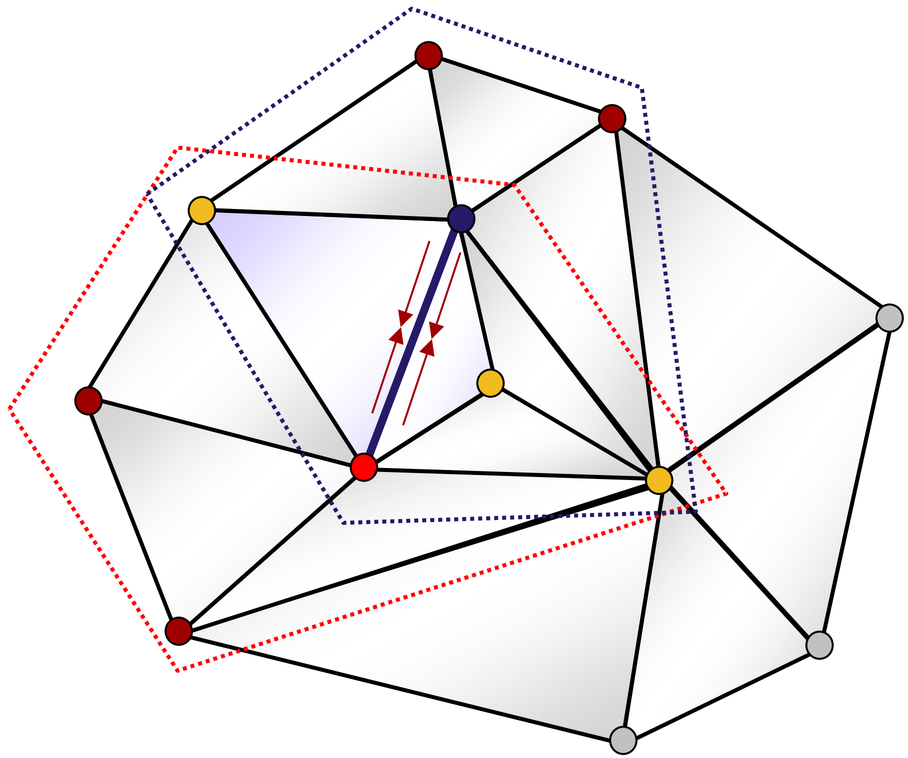
Figure 7.4. Two examples for topologically illegal (half-)edge collapses $\mathbf{p} \to \mathbf{q}$. Collapsing two boundary vertices through the interior leads to a non-manifold pinched vertex (top). The one-rings of $\mathbf{p}$ and $\mathbf{q}$ intersect in more than two vertices, which after collapsing results in a duplicate fold-over triangle and a non-manifold edge (bottom).

Botsch et al., Polygonal Mesh Processing

Illustration



this works



this folds

Components

The algorithm needs the following components:

- Topology check (mostly fixed) ✓
- Costs for edge contractions (lots of choices)
- Placement of new vertices (lots of choices)

Quadric Error Metric

Quadric error metric: [Garland and Heckbert 1997]

- Very efficient solution to the error quantification problem

Idea:

- Measure distance to planes, rather than original triangles
- Collapsed edge results in a point minimizing the error
- The error is represented as a quadric function

Quadric Error Metric

Implicit plane equation:

$$\langle N, p - q \rangle = 0$$

Quadratic error function:

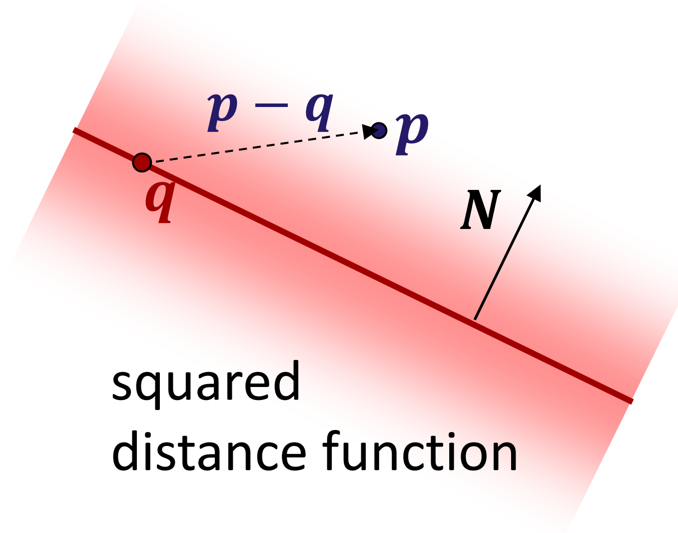
$$\langle N, \boxed{p} - q \rangle^2$$

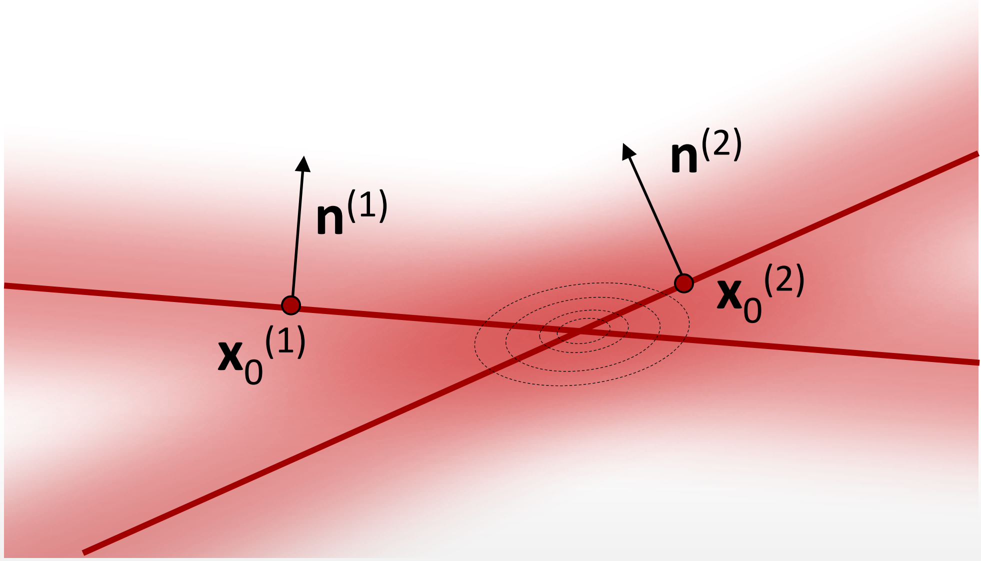
variable

Minimum distance to several planes:

$$\sum_i \langle N_i, \boxed{p} - q_i \rangle^2$$

variable





Quadric Error Metrics

Use in mesh simplification:

- Assign an initial error quadric to each vertex
  - Summing up the plane error functions of all adjacent triangles
  - Weight components by triangle area
  - Error will be zero for the vertex itself
    - Intersection of all planes
- For each possible edge contraction:
  - Just add the error quadrics of both vertices involved
  - The new, contracted vertex should approximate the planes of all triangles involved so far

Quadric Error Metrics

Use in mesh simplification:

- For each possible edge contraction:
  - Compute the optimum vertex position
    - According to the summed error metric
  - Evaluate the quadric to determine the error
  - This is the candidate move (error, position)
    - Stored in the edge contraction queue
- When an edge contraction occurs:
  - Use the computed position
  - Recompute neighborhood error quadrics
    - Add error matrix of the new vertex to each neighboring vertex

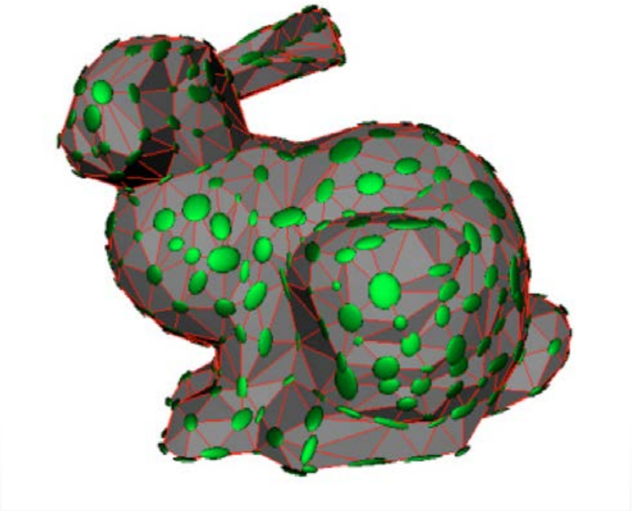
Quadrics

Eigenvectors

- The eigenvectors of the quadrics

$$\sum_i N_i {N_i}^T$$

provide information about curvature directions and the normal direction of the surface



Extension

Meshes also have attributes, such as:

- Color
- Texture coordinates

This can be handled using quadric error metrics as well:

- Just store additional columns in the x-vectors
- Treat color values (etc.) as additional dimensions of the vertex position, weighted by relative importance to preserve them

How well does this work?

Advantage:

- Very fast: Evaluating the error metric and finding a new vertex position is O(1)

Disadvantage:

- For noisy meshes, the error approximation is bad:

- Possible solutions:
  - Mesh smoothing (normals from larger neighborhoods)
  - Reset quadrics after a few computation steps

Components

The algorithm needs the following components:

- Topology check (mostly fixed) ✓
- Error metric (lots of choices) ✓
- Placement of new vertices (lots of choices) ✓

Conclusion:

- Quadric error metrics are a very popular choice due to their simplicity and performance.
- Alternatives exist (at higher costs).