15: Smoothing

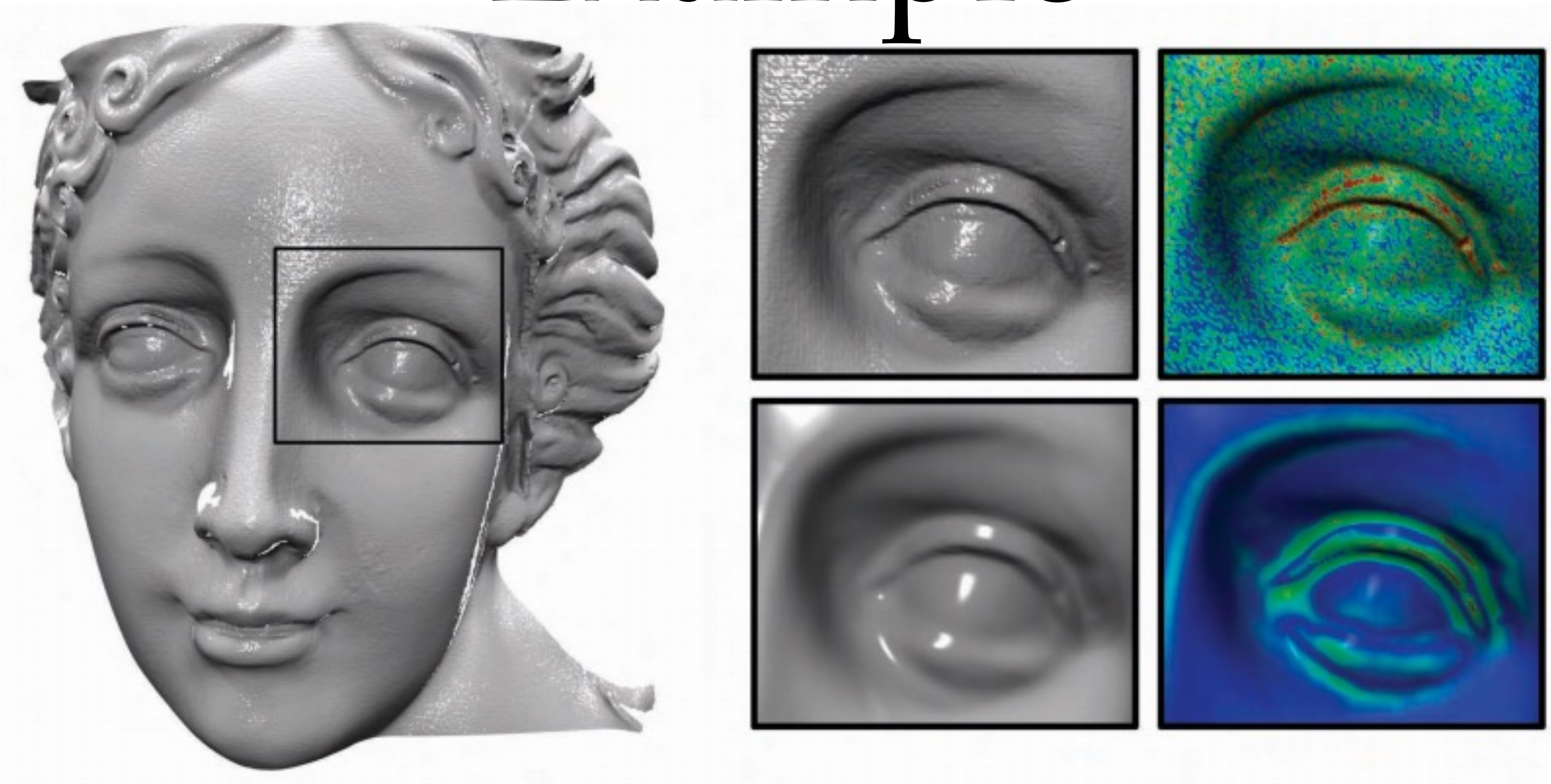
Smoothing Surfaces

- Many techniques lead to rough surfaces
- We want the equivalent of image filters
  - Blur
- Or surfaces that are too smooth already
  - Sharpen
  - Or add detail (bump maps, &c.)
- So what does it mean to smooth a surface?

Meaning of Smoothing

- To reduce high-frequency information
  - I.e. rapid changes in surface
  - Reduce the curvature (aha!)
  - Related to simplification
  - But retains the triangle count

Example



- Laser scan has measurement noise
- We want to get rid of it
- Colour-coding is mean curvature

A Warning

- Sometimes you *want* sharp edges
- Eg machined surfaces
- There are many techniques for retaining them
  - NURBS in particular
  - Some variants of subdivision surfaces
  - But we will ignore this

Image Filtering

- We introduced Fourier analysis for filtering
- Breaks a function into sums of sines / cosines
  - High frequency – sharp detail
  - Low frequency – blurring, large features
- We did it for 1D & 2D
- Of course it works for 3D (volumetric)
- But we now have a 2-manifold surface

Spherical Harmonics

- Define a set of radial patterns on a sphere
- Different frequencies & amplitudes
- Break the surface function into them
- Works best for roughly spherical shapes
- Less well for arbitrary manifolds

Manifold Harmonics

- Based on observation about Fourier analysis
- Sines / cosines are *eigenfunctions*
  - Of the Laplace operator of the function
- So we use the Laplace-Beltrami of the position
  - It's a symmetric positive semi-definite matrix
  - Find it's eigenvectors
  - Use these to decompose the function

Manifold Harmonics

- Basis functions emphasise different regions

Laplace-Beltrami Matrix

- Given a mesh M

- And a vector $f = \begin{bmatrix} f(v_1) \\ \cdots \\ f(v_n) \end{bmatrix}$ on it's vertices

- Find the Laplace(-Beltrami) matrix $\boldsymbol{L}$ such that

$$\Delta f = \boldsymbol{L} f$$

L-B Development

$$\Delta f(v_i) \quad = \sum_{v_j \in N_1(v_i)} \quad w_{ij} \quad \left(f(v_j) - f(v_i)\right)$$

Vertex value (position) Sum on 1-ring Weight Value difference (edge vector)

Notice that we subtract $-w_{ij} f(v_i)$ per neighbour

And add $w_{ij} f(v_j)$ for each neighbor

So we collect them together as coefficients in a matrix

L-B Development

$$\Delta f(v_i) = \sum_{v_j \in N_1(v_i)} w_{ij} \left(f(v_j) - f(v_i) \right)$$

$$= \left(\sum_{v_j \in N_1(v_j)} L_{ij} \right) f$$

$$L_{ij} = \begin{array}{cc} \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ 0 & -w_{ij} & 0 & w_{ij} & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ & col\ i & & col\ j & \end{bmatrix} & \begin{array}{l} \\ row\ i \\ \\ \\ \end{array} \end{array}$$

L-B Development

$$\Delta f(v_i) \quad = \quad L \quad f$$

where $$l_{ij} = \begin{cases} -w_i \displaystyle\sum_{v_k \in N_1(v_i)} w_{ik} & i = j \\ w_{ij} & v_j \in N_1(v_i) \\ 0 & otherwise \end{cases}$$

And l_{ii} will often work out to -1
(or $-\delta(v_i)$, the degree)

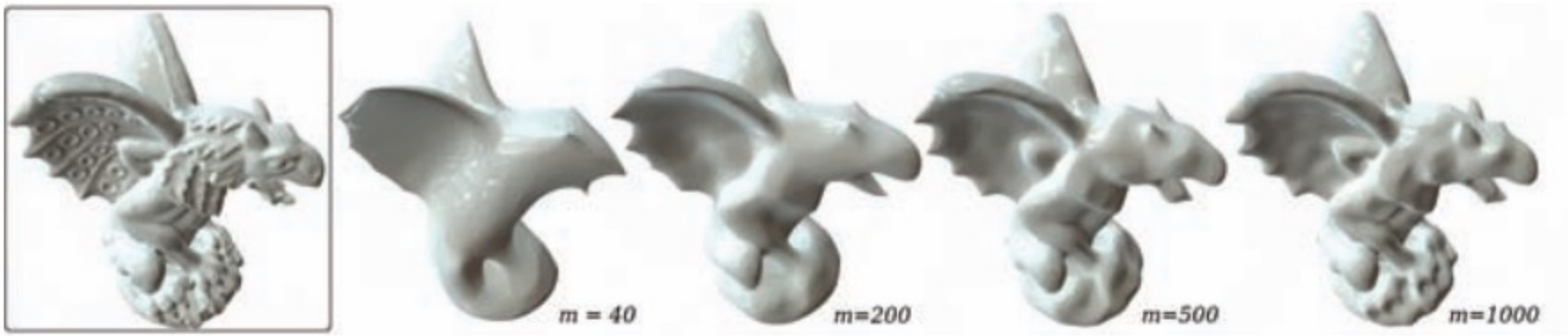
Constraints

- L needs to be symmetric positive semi-definite
  - Symmetric: $w_{ij} = w_{ji}$
    - Can't use area- or angle- weighting
    - Either $w_{ij} = 1$ (simple 1-ring)
    - Or $w_{ij} = cot\alpha_{ij} + cot\beta_{ij}$
    - AND don't divide through by 1-ring size

Processing

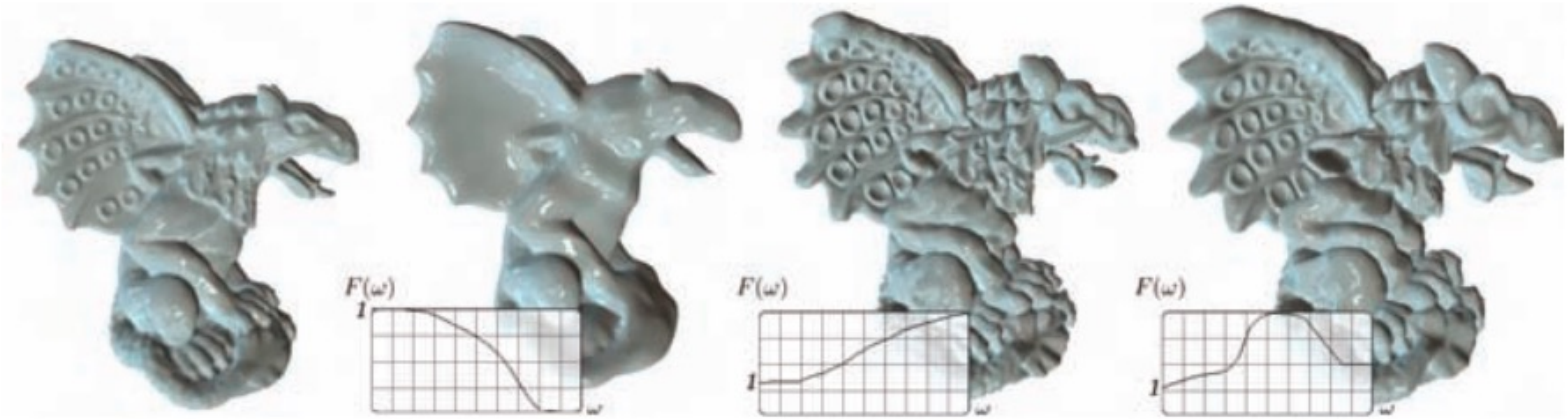
- Once we've done this, find eigenvectors
- Often referred to as "natural vibrations"
- Sorted by size of eigenvalue
- Low pass filter:
  - Use the first m eigenvectors
  - And apply to position function $f = \boldsymbol{x}$
- But this is expensive

First M Harmonics



- First harmonics define shape
- Later harmonics define details
- But you can have a *lot* of them

Results



Model Low Pass High Pass Enhancement

- General filters weight all harmonics
- Preserving at least some detail from each

Diffusion Flow

- Also known as *heat maps*
- Many processes such as heat mix over time
  - i.e. gradually diffuse outwards
  - So we use the term "heat equation"
  - $\frac{\partial f(\boldsymbol{x},t)}{\partial t} = \lambda \Delta f(\boldsymbol{x},t)$
    - λ is the spatial diffusion coefficient (speed)
    - $\Delta f(\boldsymbol{x},t)$ is the spatial Laplacian

Discretisation & Integration

- Discretise this into a vector:
  - $\frac{\partial f(v_i,t)}{\partial t} = \lambda \Delta f(v_i, t)$
- Then apply explicit Euler integration
  - $f(t+h) = f(t) + h\frac{\partial f(t)}{\partial t}$
  - $= f(t) + h\lambda \mathbf{L}\mathrm{f}(\mathrm{t})$
  - For a small time step h

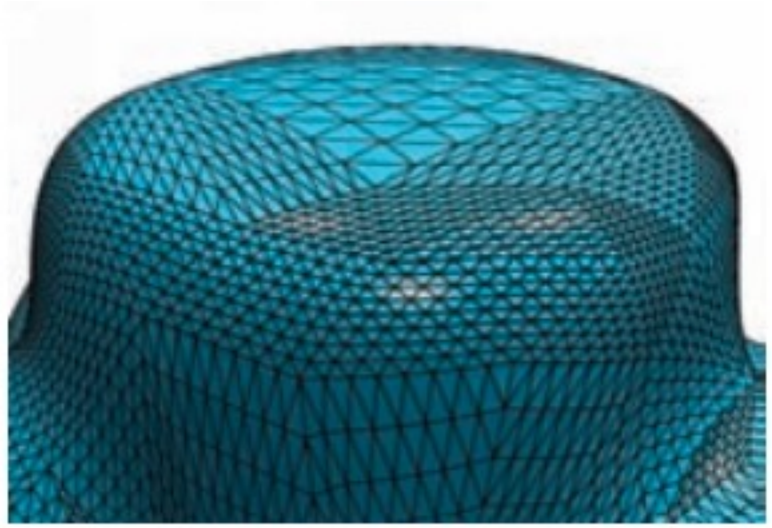
Spatial Position

- This has assumed a general function f
  - $f(t+h) = f(t) + h\lambda \mathbf{L}\mathrm{f}(\mathrm{t})$
- Substitute in our usual $f = \boldsymbol{x}$, and we get:
  - $x_i \leftarrow x_i + h\lambda\Delta x_i$
- The last term is just h times mean curvature!
- So we are smoothing out rough patches

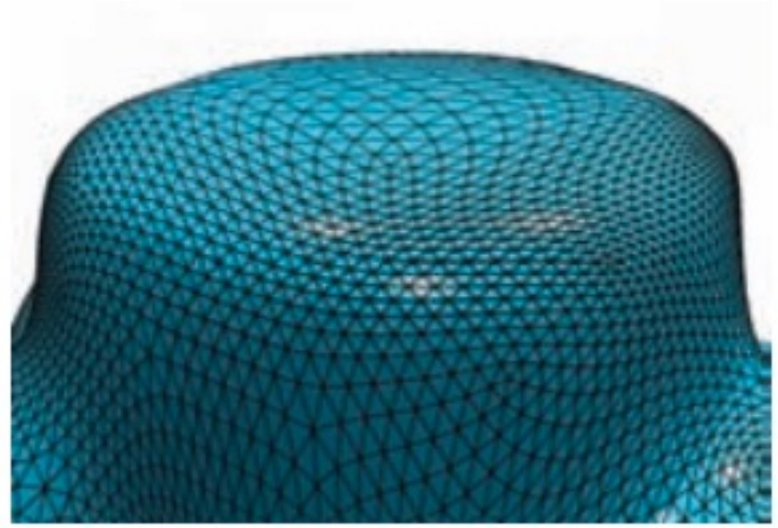
Curvature Flow Smoothing

- After 0, 10 and 100 iterations

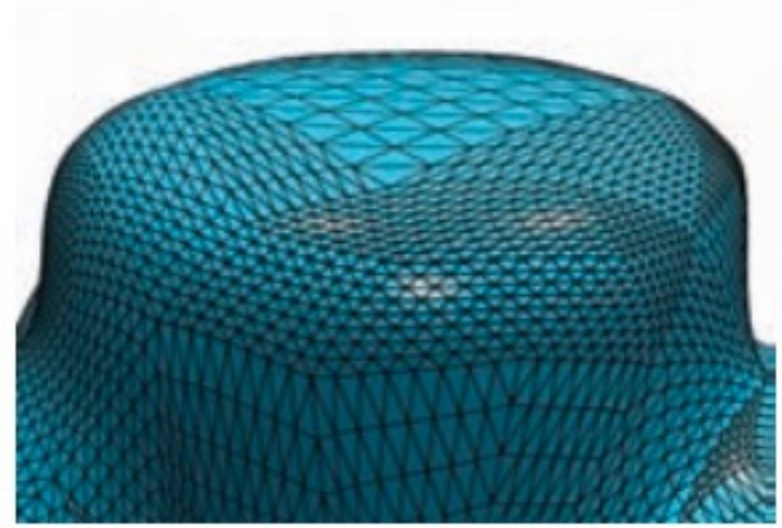
Variations



Original



Uniform: 10



Cotangent: 10

- If you weight the 1-ring uniformly
  - Vertex moves towards neighbours' centroid
  - And it tends to regularize the triangles
- Cotangent weights preserve triangle shapes