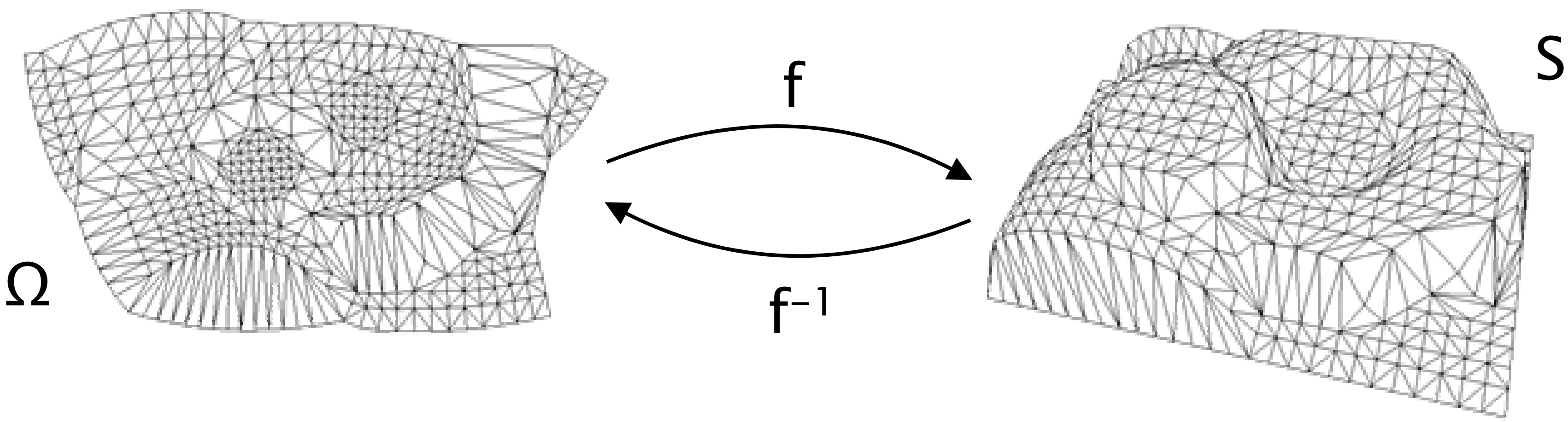
Applications of Parameterizations

Kai Hormann

TU Clausthal

A Quick Reminder

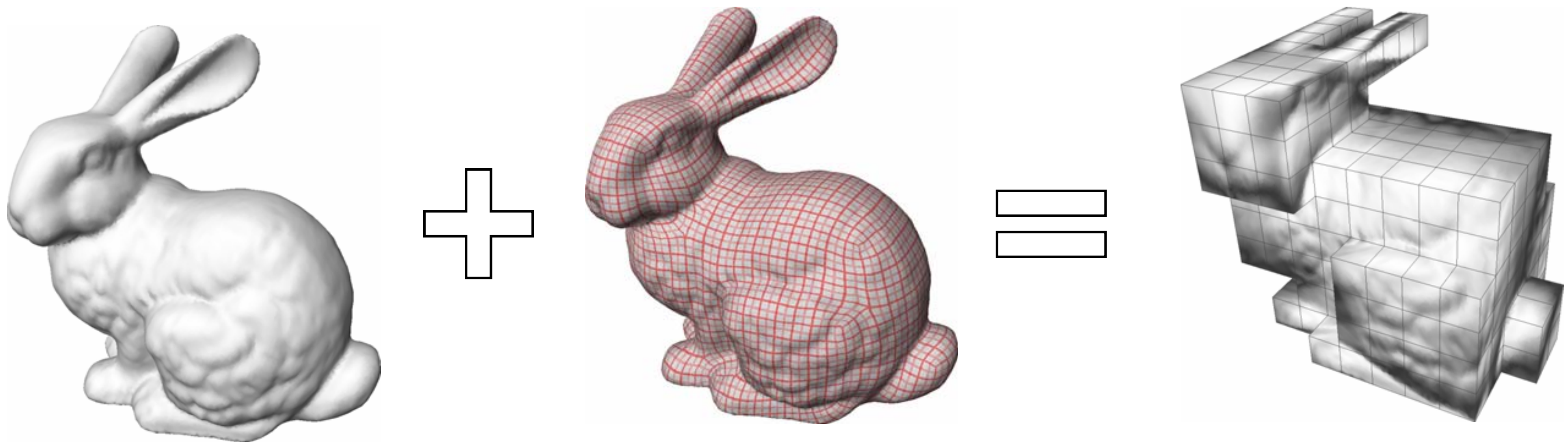
- a parameterization is a bijective mapping between a surface and a planar domain



Applications

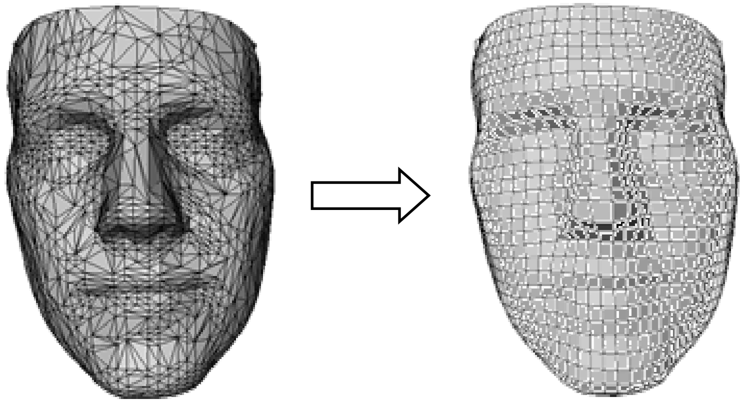
- texture mapping

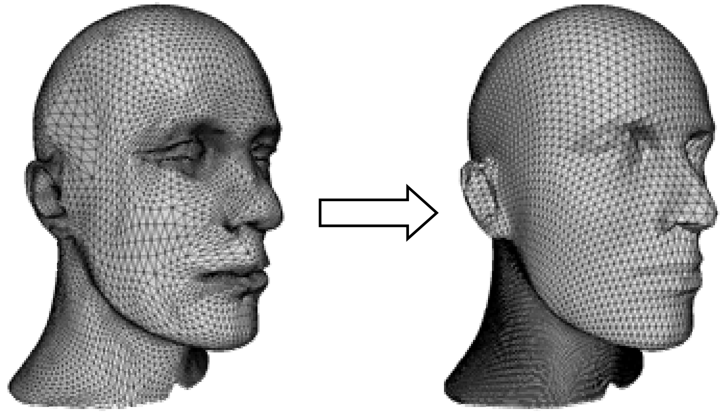
- texture synthesis



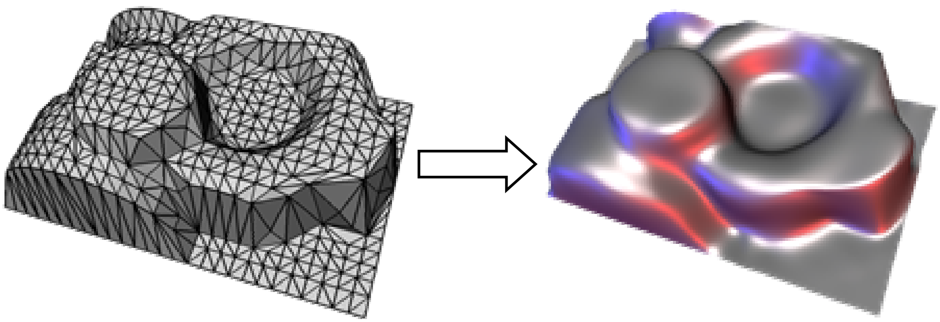
Applications

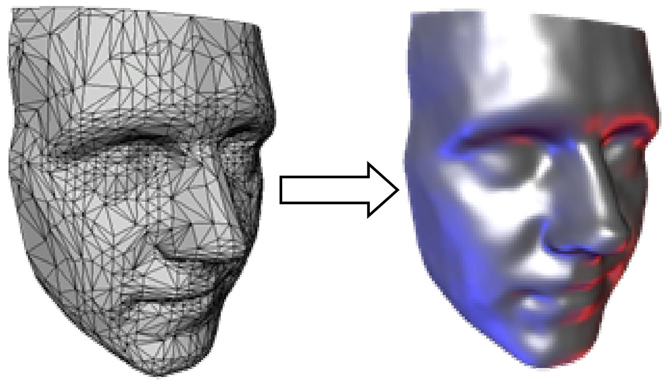
- remeshing



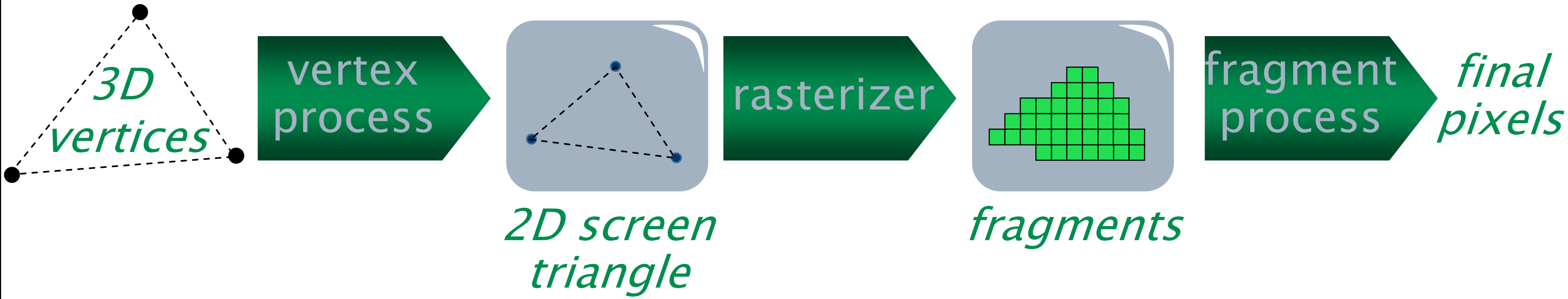


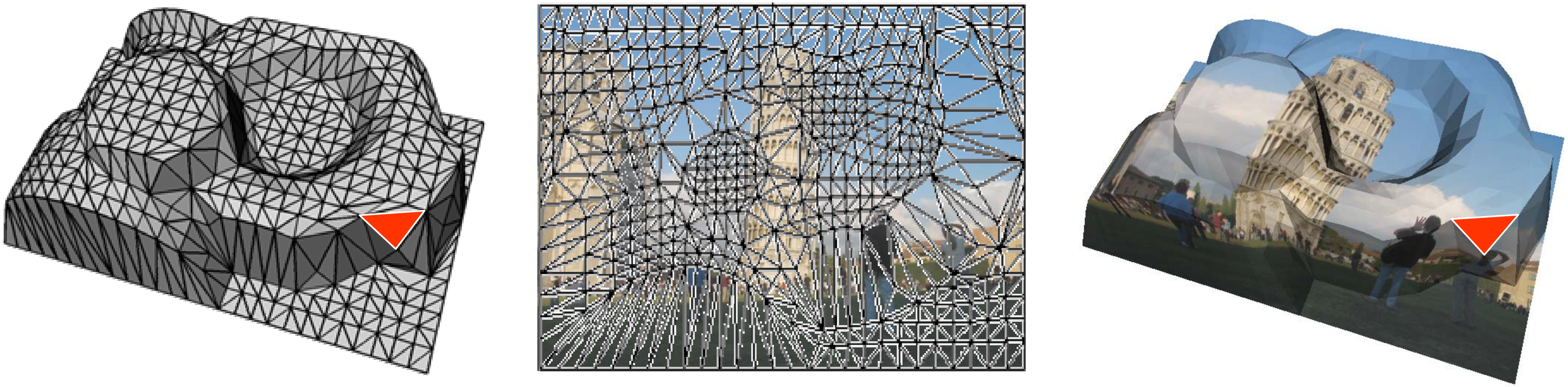
- surface reconstruction



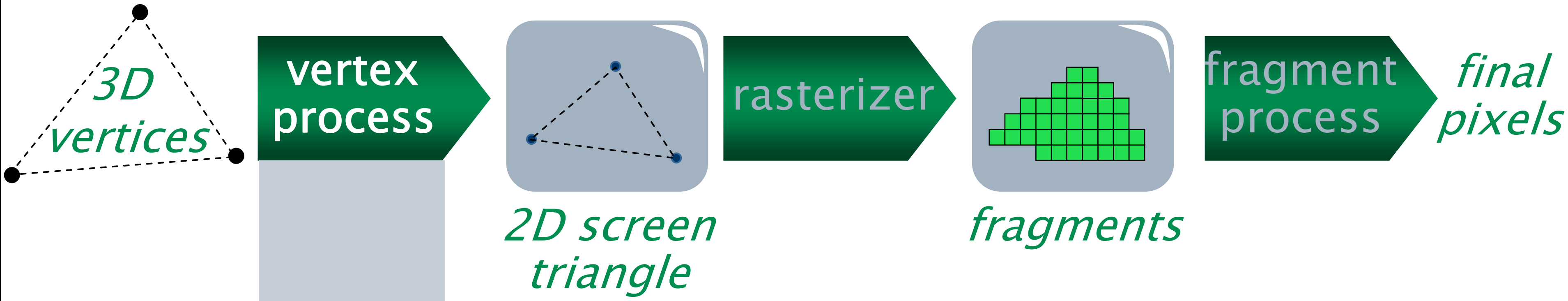


Texture Mapping



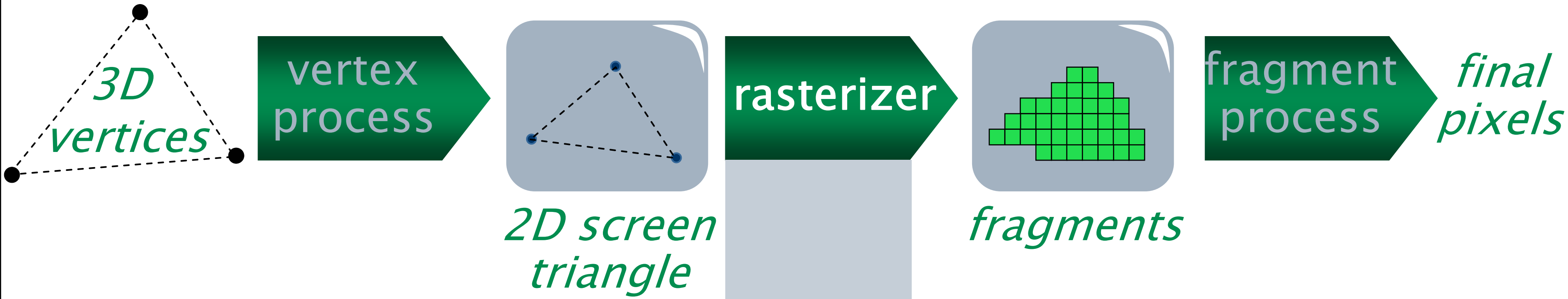


Vertex Process



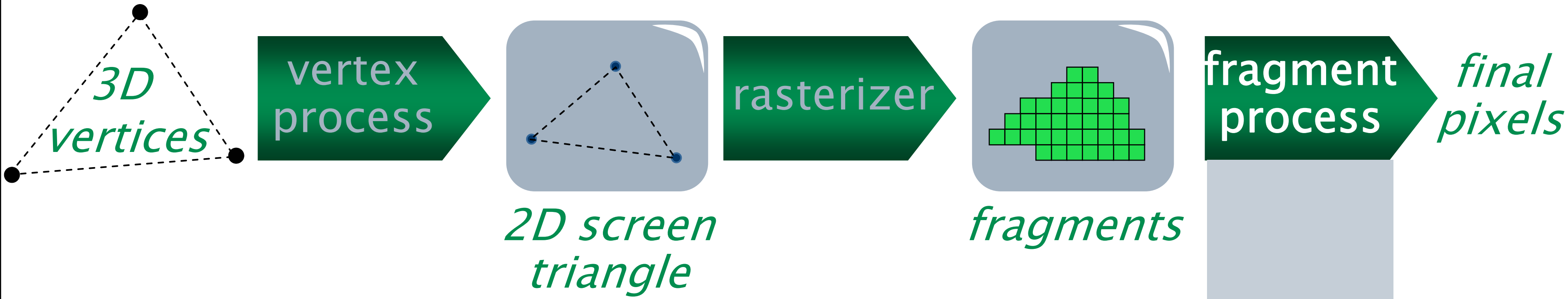
- project vertices
- define vertex attributes
- for example, texture coordinates
- etc.

Rasterizer



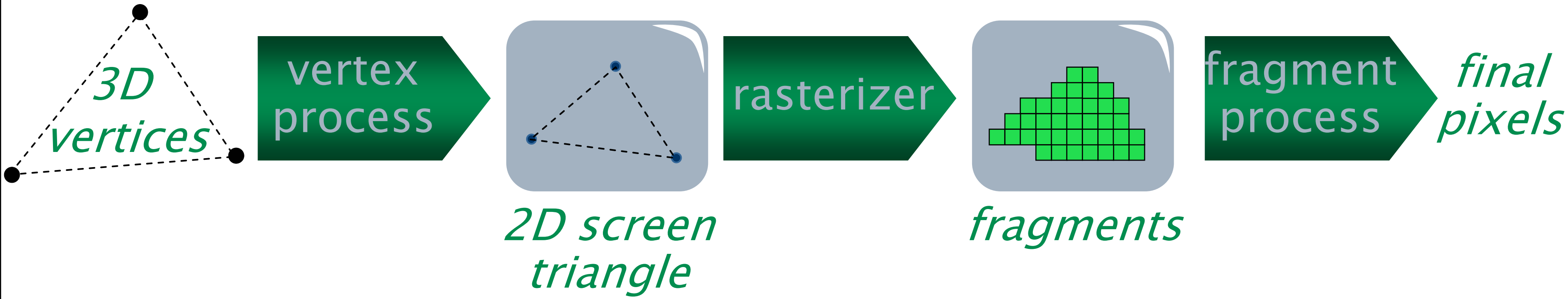
- determine fragments to be covered
- interpolate attributes per fragment
- for example, texture coordinates
- etc.

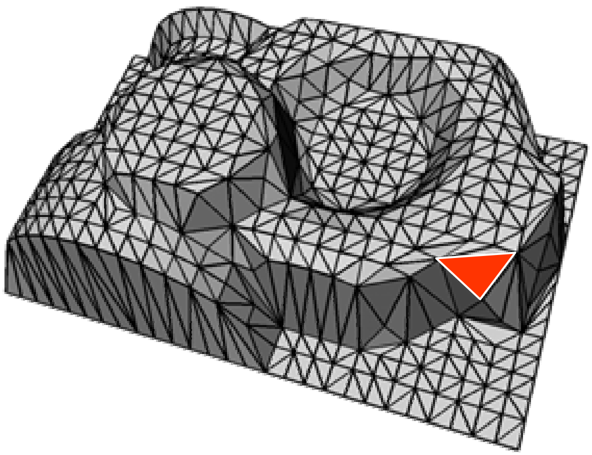
Fragment Process

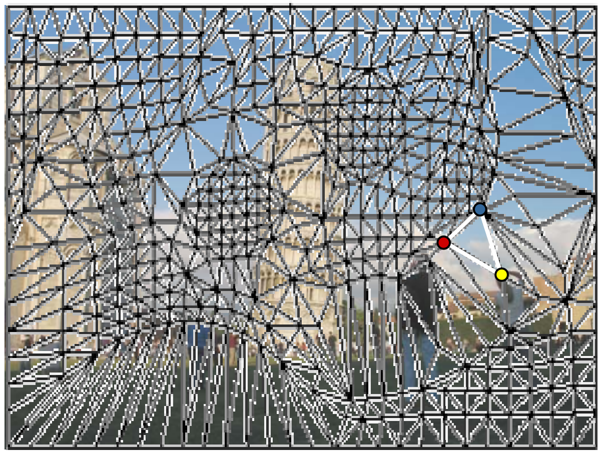


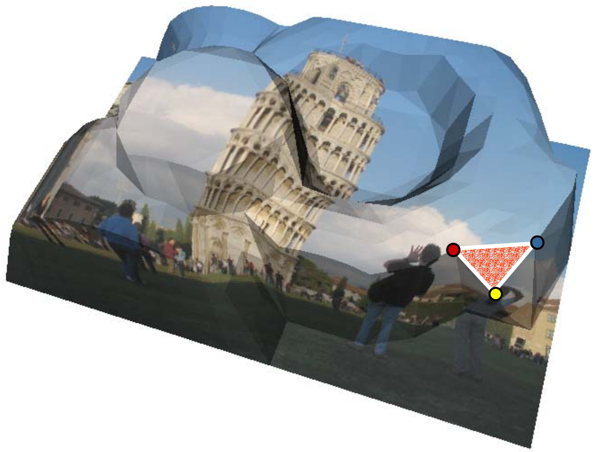
- depth test
- shading
- texture accesses
- etc.

Texture Mapping





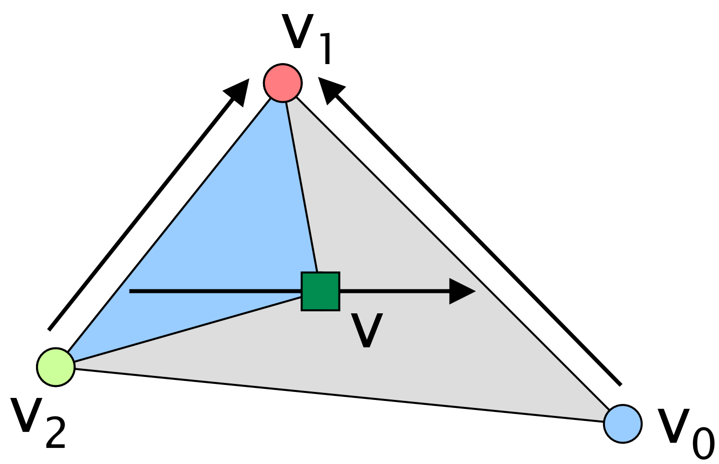




Interpolating Vertex Attributes

- barycentric coordinates

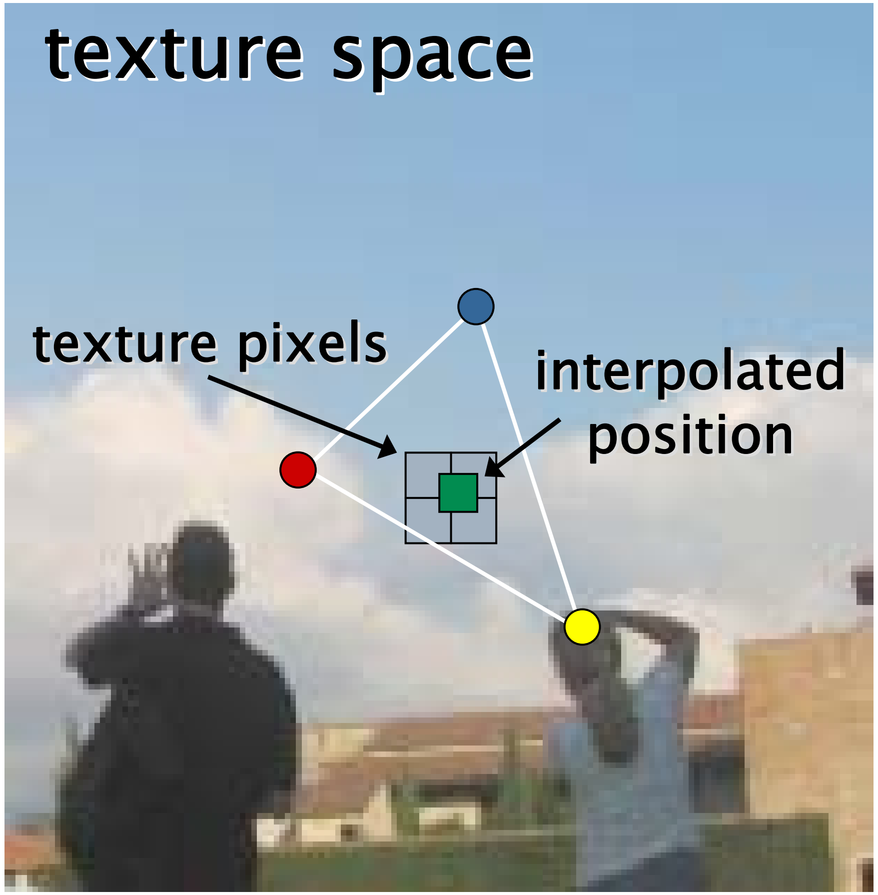
$$\lambda_i(v) = \frac{A(v,\, v_{i+1}, v_{i+2})}{A(v_0,\, v_1, v_2)}$$

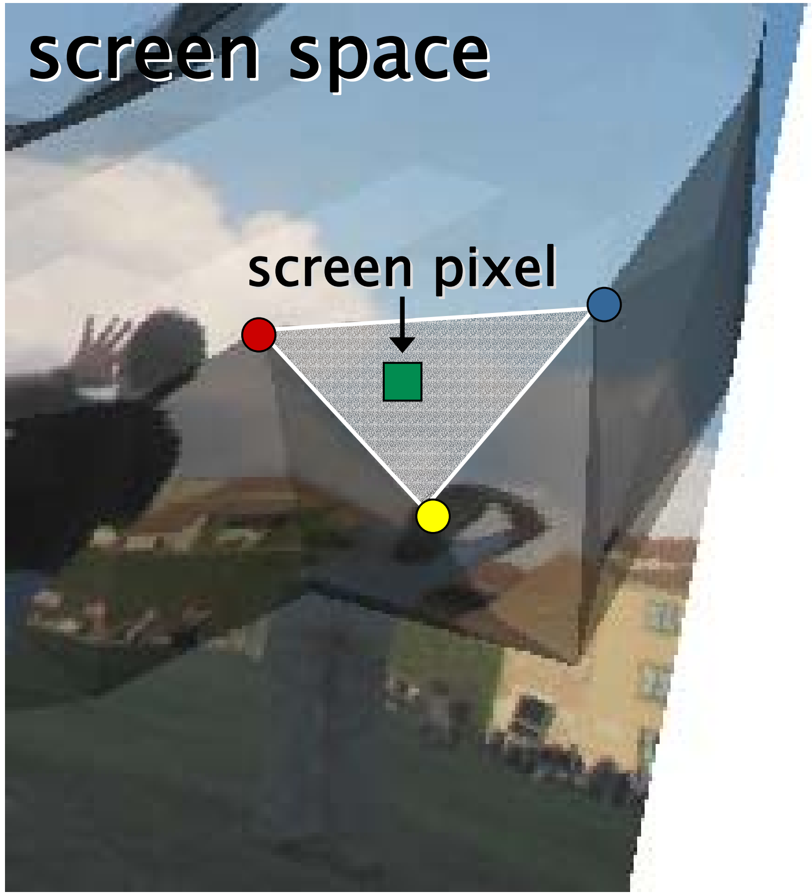


- weights for linear interpolation

$$a(v) = \sum_{i=0}^{2} \lambda_i(v)\, a_i$$

Texture Access





- bilinear filtering of texture pixels

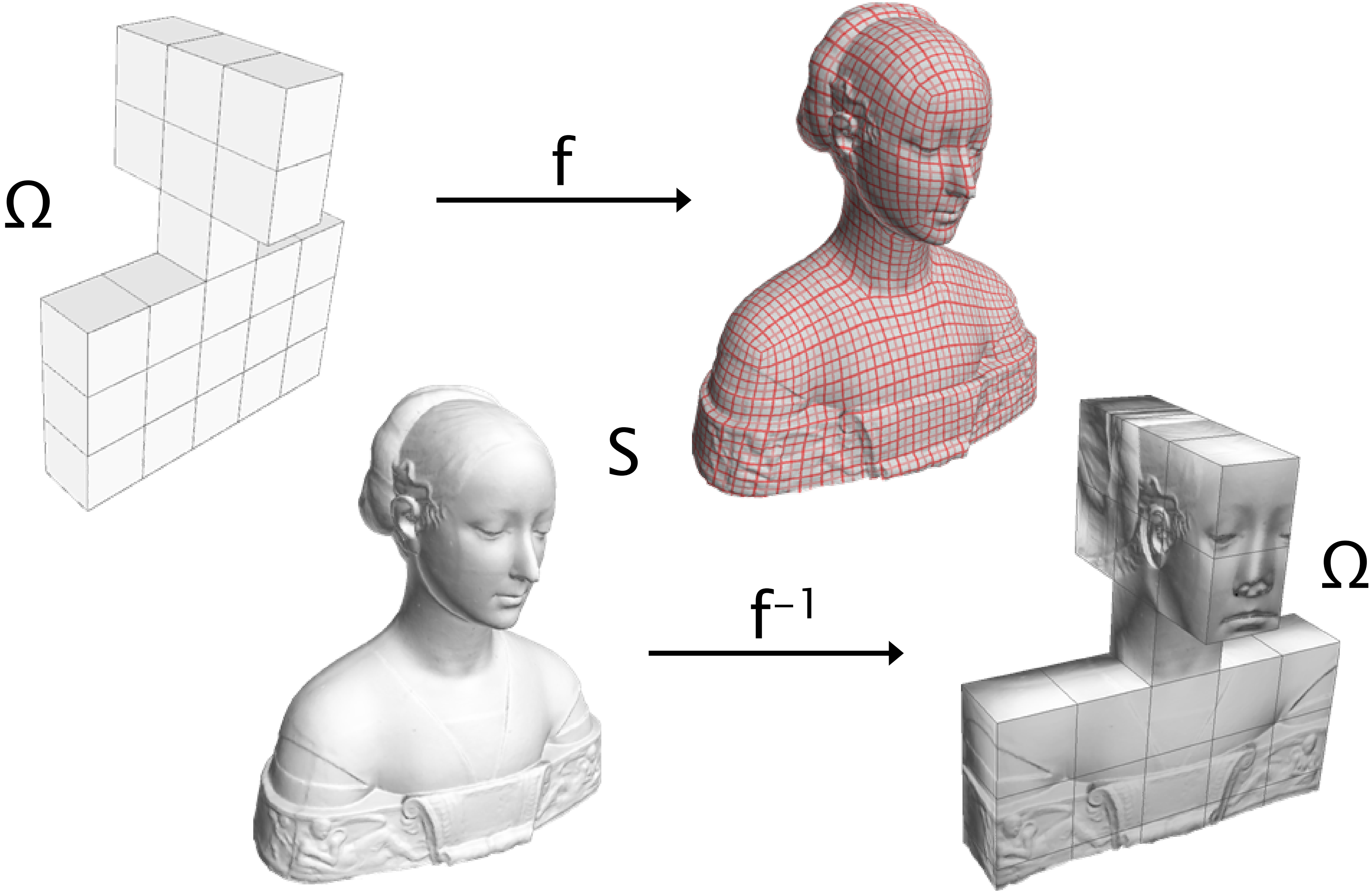
Parameterization Requirements

- image should not be distorted too much
  - area distortion
  - angle distortion
- parameterization should be as isometric as possible
- use pointwise measure that
  - measures isometric distortion, or
  - some combination of angle and area distortion

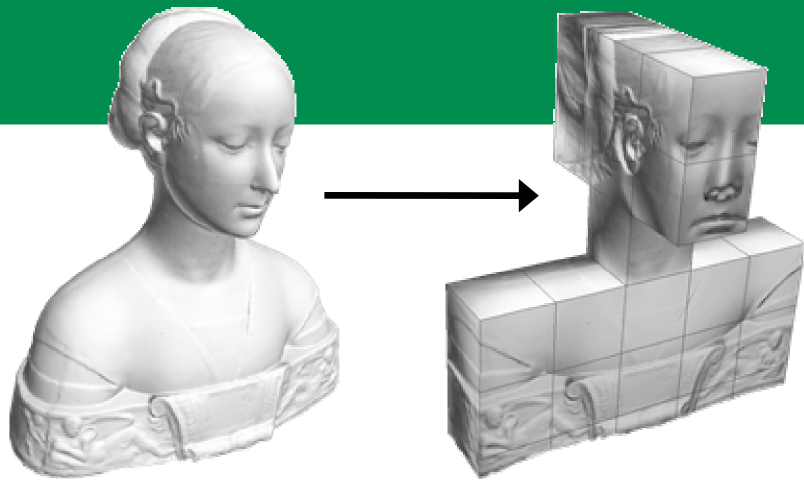
Texture Synthesis

- so far: “forward parameterization”
  - map colour from image Ω to surface S
- “backward parameterization”
  - colour or other signal given on S
  - map it to Ω
  - store it as a bitmap
  - use standard texture mapping to map it back during rendering

Texture Synthesis



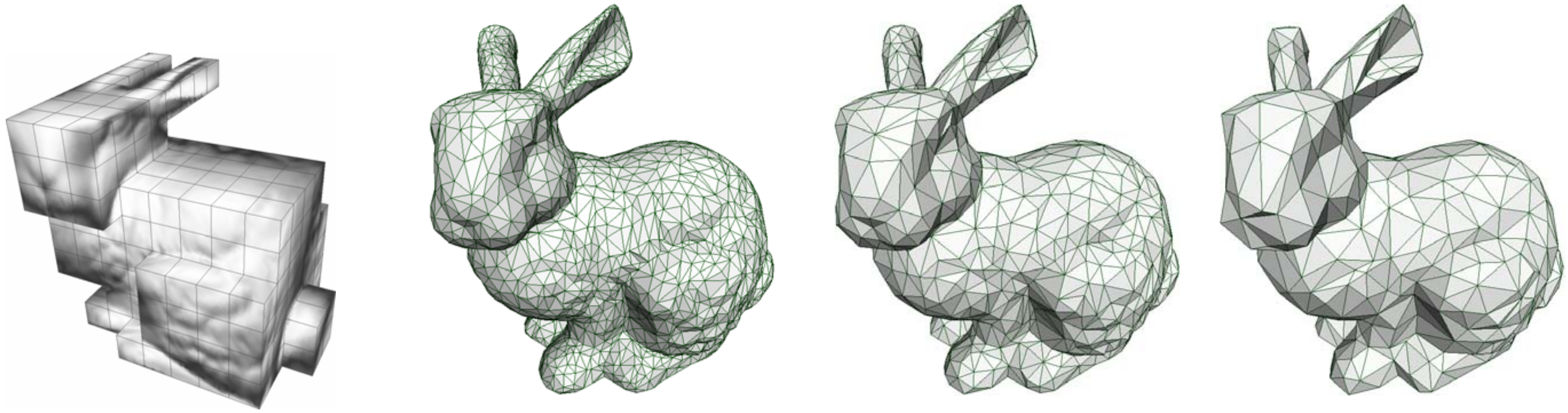
Texture Synthesis



- two approaches
  to generate texture
- splatting
  - sample triangles T' in S uniformly
  - map sample into Ω, using f^{-1}
  - store colour in affected pixel(s)
- direct look-up
  - map each pixel in Ω onto S, using f
  - compute and store colour

Applications

- interactive rendering of still scenes with high quality shading
  - from global radiosity computation
  - from ray-tracing
- high quality rendering of simplified meshes

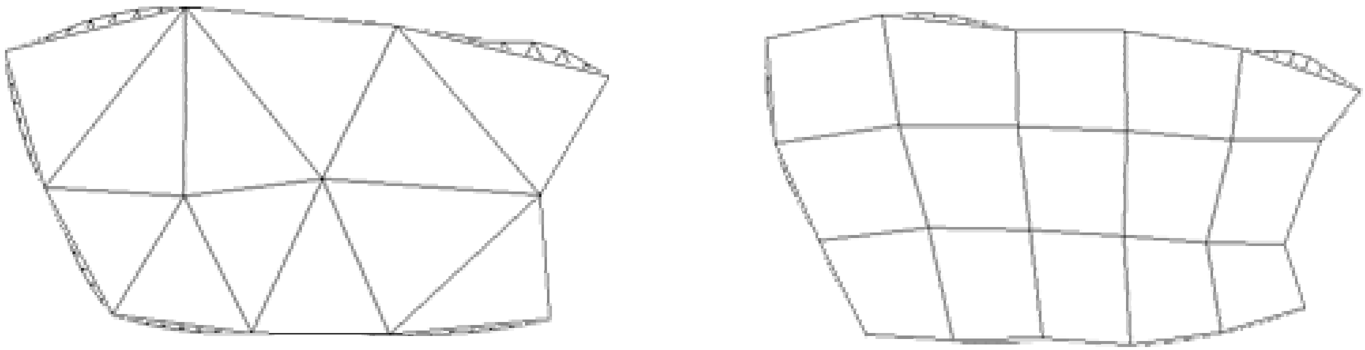


Parameterization Requirements

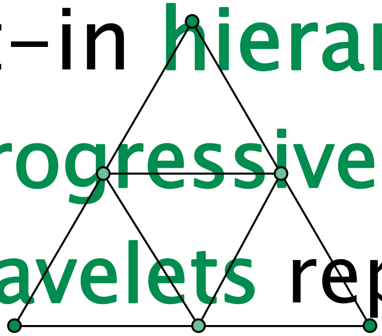
- distortion does not matter in principle
  - signal is distorted upon construction
  - but undistorted when mapped back
- should maintain uniform signal density
  - for any two patches on S with the same size, the signal should be stored in the same number of pixels
- equiareal mappings
  - e.g. stretch metric [Sander et al. 2001]

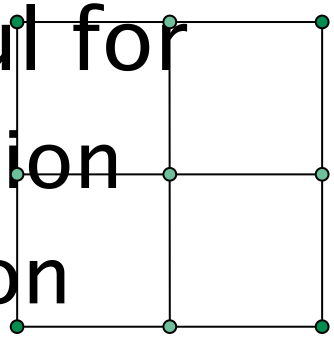
Regular Meshes

- successive refinement of a base mesh



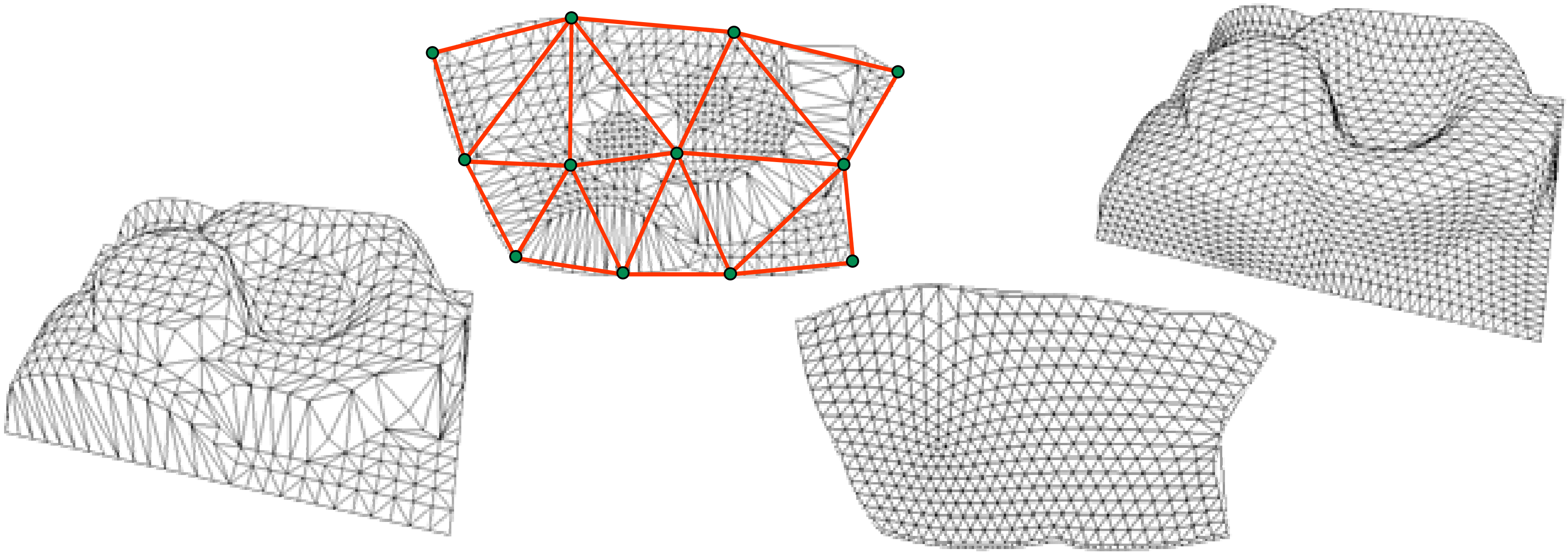
- built-in hierarchy useful for
  - progressive transmission
  - wavelets representation
  - hierarchical modelling



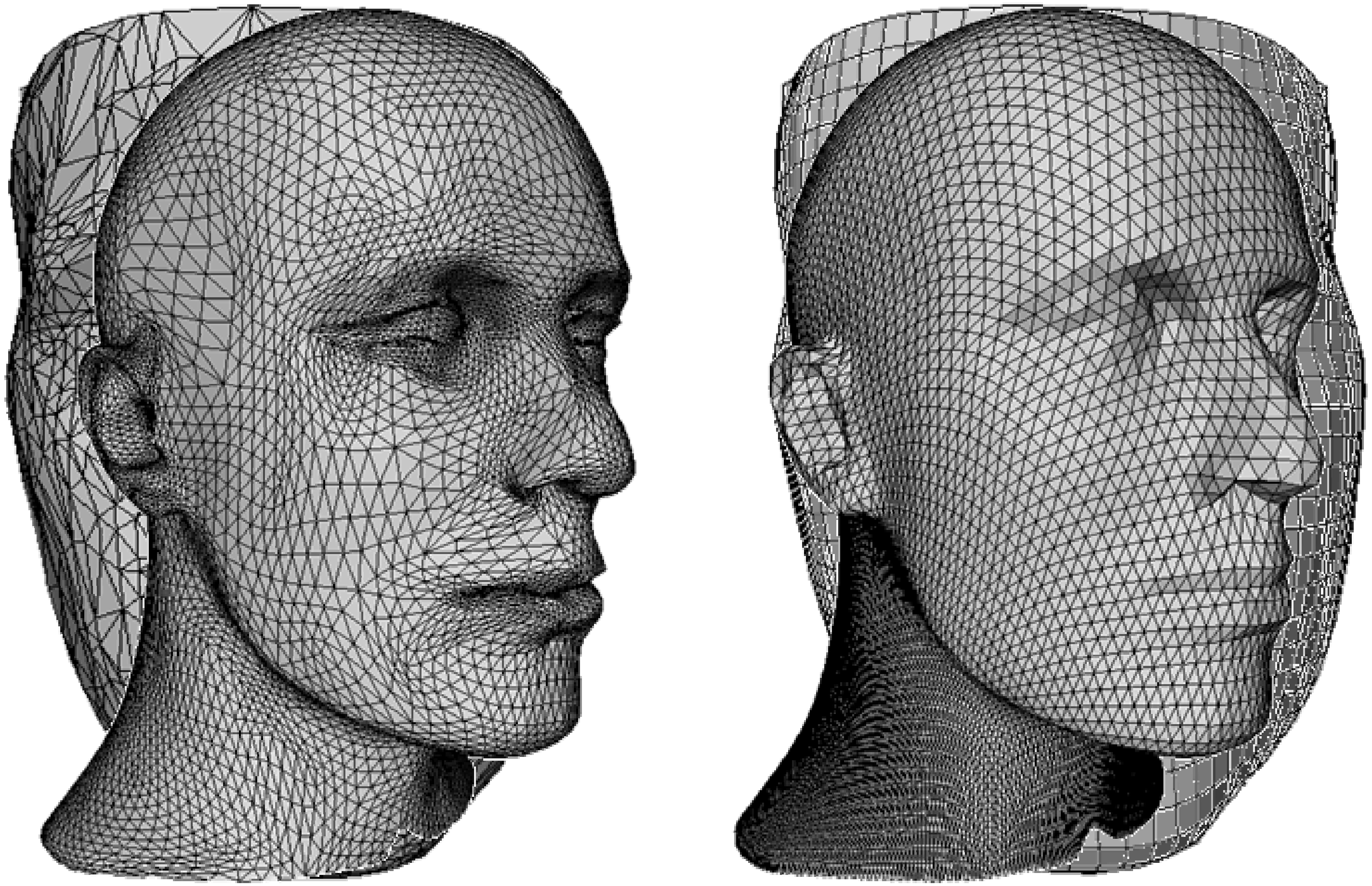


Remeshing

- replace arbitrary mesh with a regular one
- parameterization (3D → 2D)
- remeshing in 2D
- lift the regular mesh (2D → 3D)



Examples

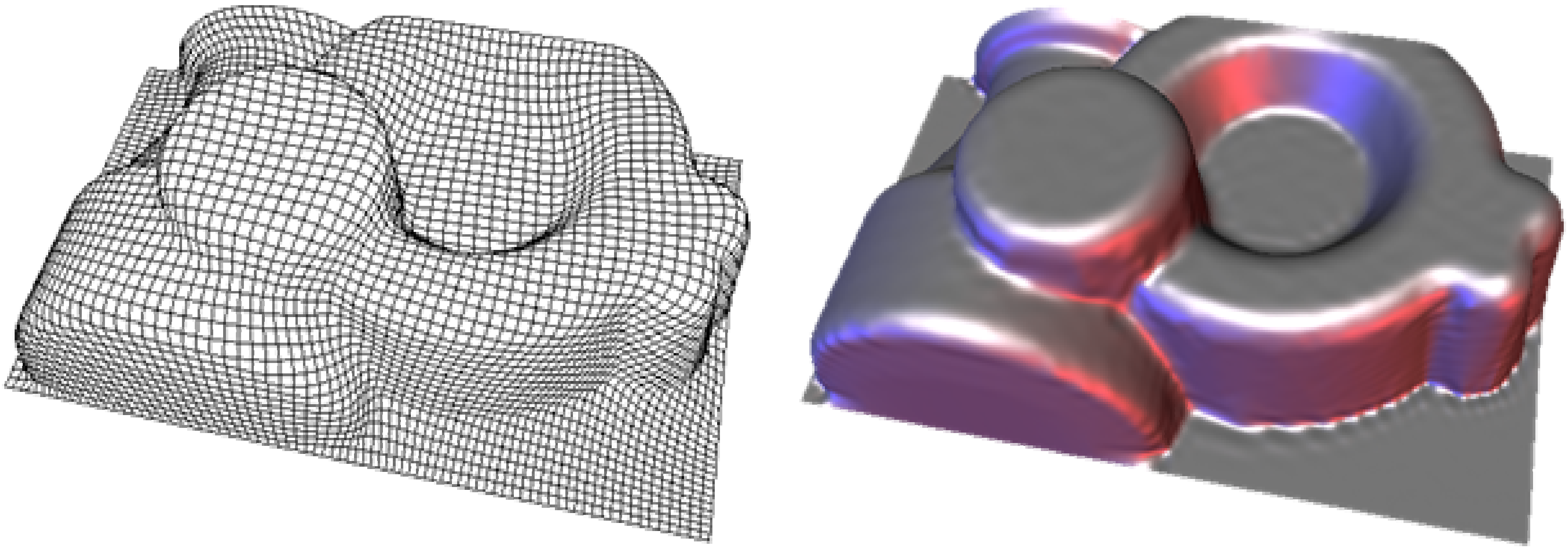


Parameterization Requirements

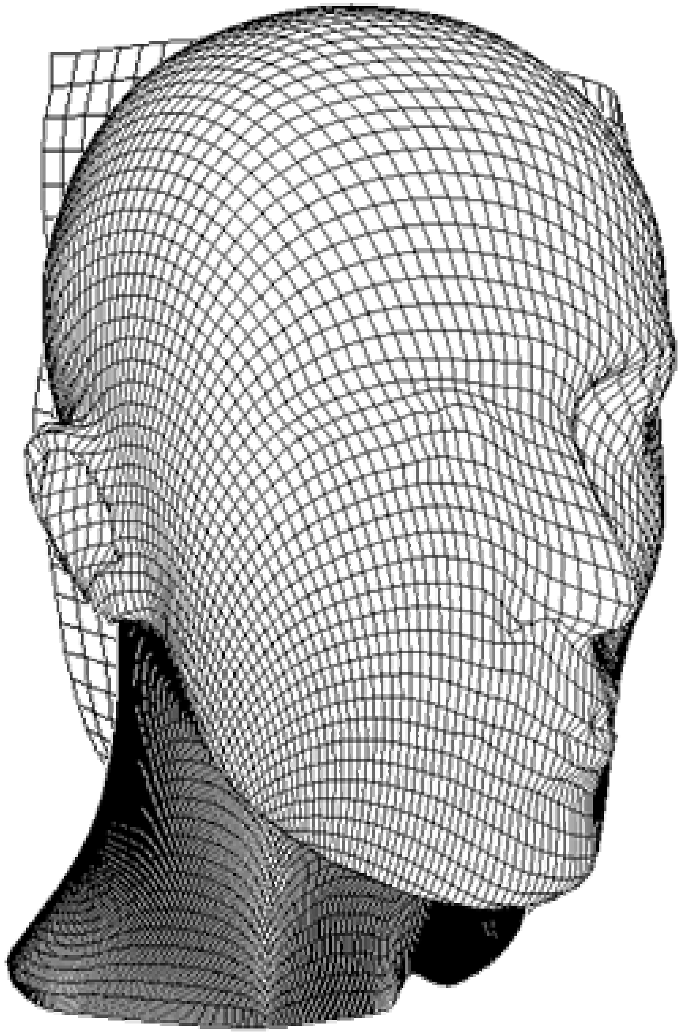
- remesh should have uniform faces
- conformal maps ⇒ uniform shape
- equiareal maps ⇒ uniform size
- ideally, parameterization should be isometric
- "correct" pointwise measure unknown

Interpolation of Regular Grids

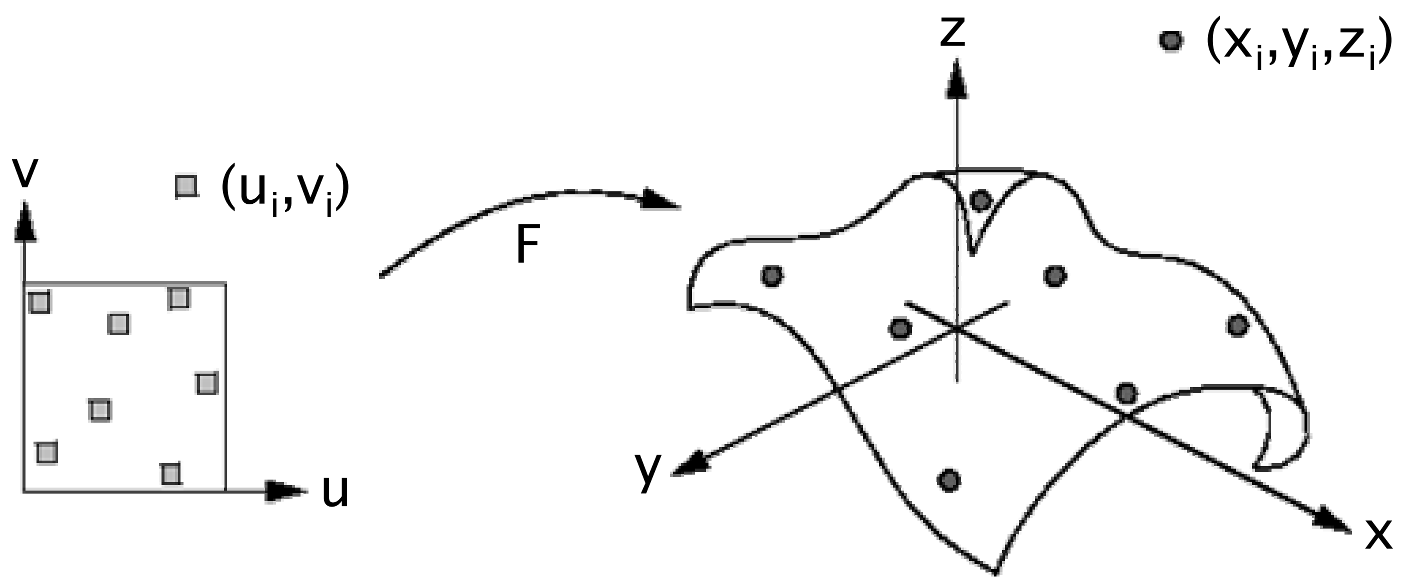
- regularity allows for simple interpolation
- bicubic tensor-product B-splines
- problem reduces to curve interpolation
- tri-diagonal linear systems



Examples



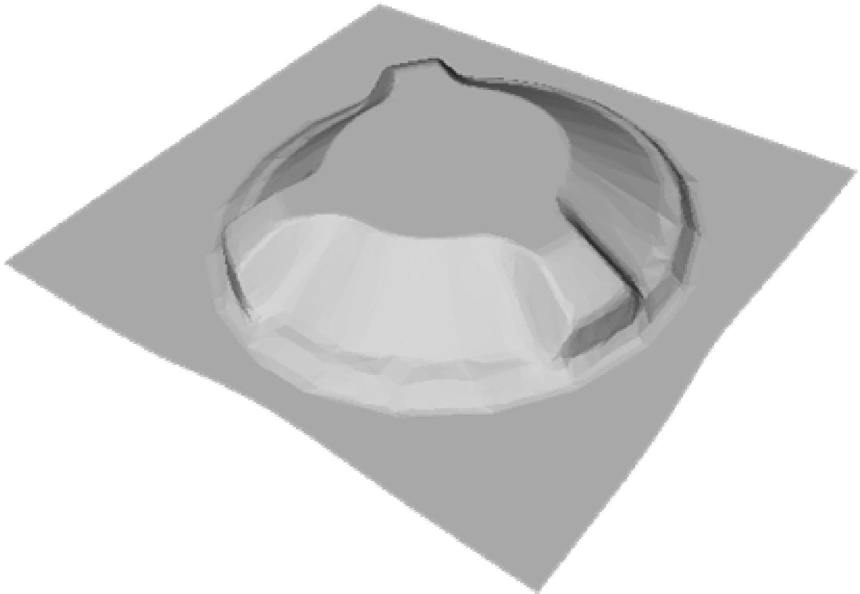
Approximation of Scattered Data

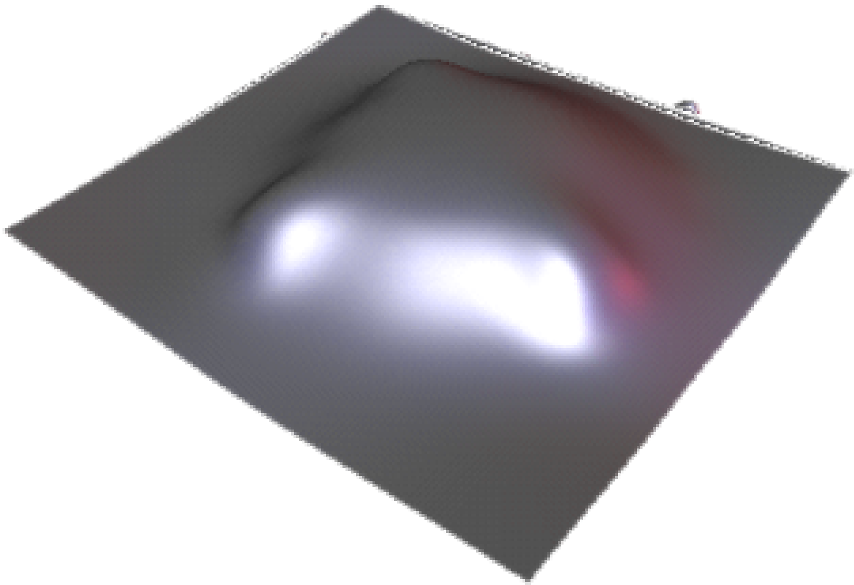


- **bicubic tensor-product B-splines**
  - numerically stable and efficient
  - standard surfaces in CAGD

Approximation Methods

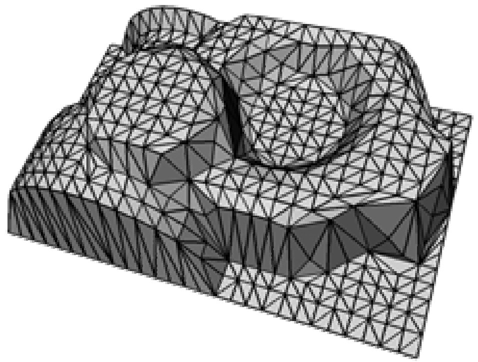
- classical approach: **least squares approx.**
- solving a **linear system**
- **stabilization** by smoothing functionals

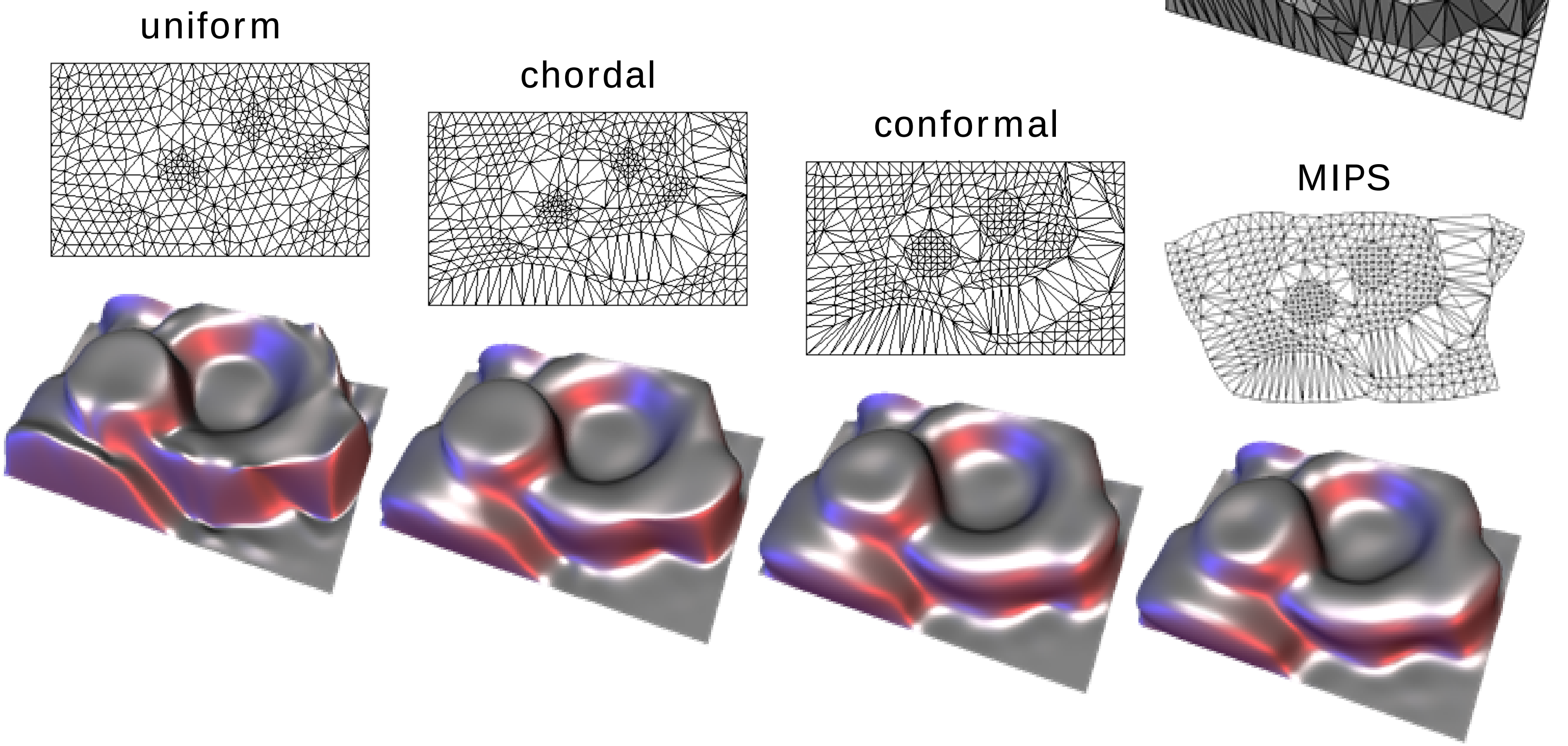




Approximation of Triangle Meshes

- effect of the parameterization





Parameterization Requirements

- empirical observations
  - conformal maps give good results
  - big area and angle distortions lead to oscillations
- not well understood
- not even in the curve case