Estimation / Approximation Problems in 3D Photography

Tom Duchamp, Department of Mathematics
Werner Stuetzle, Department of Statistics
University of Washington

Previous and current members of UW 3D Photography group:

G. Arden, D. Azuma, A. Certain, B. Curless, T. DeRose, T. Duchamp, M. Eck, H. Hoppe, H. Jin, M. Lounsbery, J.A. McDonald, J. Popovic, K. Pulli, D. Salesin, S. Seitz, W. Stuetzle, D. Wood

Funded by NSF and industry contributions.
Most of the research published in a series of Siggraph papers.

Prepared for MGA Workshop III: Multiscale structures in the analysis of High-Dimensional Data, UCLA, October 25 -2 9, 2004

Outline of talk

- What is 3D Photography, and what is it good for ?
- Sensors
- Modeling 2D manifolds by subdivision surfaces
- Parametrization and multiresolution analysis of meshes
- Surface light fields
- (Smoothing on 2D manifolds)
- Conclusions

1. What is 3D Photography and what is it good for ?

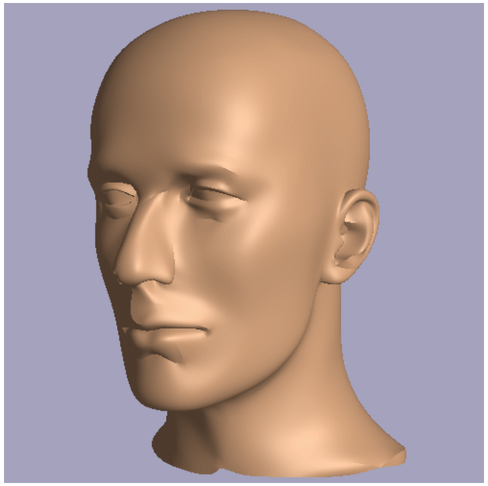
Technology aimed at

- capturing
- viewing
- manipulating

digital representations of shape and visual appearance of 3D objects.

Could have large impact because 3D photographs can be

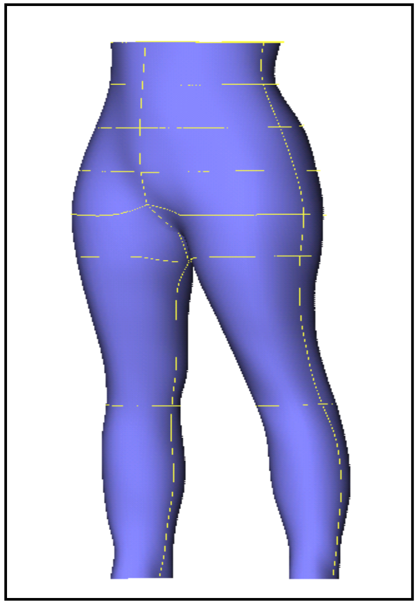
- stored and transmitted digitally,
- viewed on CRTs,
- used in computer simulations,
- manipulated and edited in software, and
- used as templates for making electronic or physical copies



Modeling humans

- Anthropometry
- Create data base of body shapes for garment sizing
- Mass customization of clothing
- Virtual dressing room
- Avatars

Scan of lower body
(Textile and Clothing Technology Corp.)



Fitted template
(Dimension curves drawn in yellow)

Full body scan
(Cyberware)

Modeling artifacts

- Archival
- Quantitative analysis
- Virtual museums

Image courtesy of Marc Levoy and the Digital Michelangelo project

Left: Photo of David's head
Right: Rendition of digital model

(1mm spatial resolution, 4 million polygons)

Modeling artifacts

Images courtesy of Marc Rioux and the Canadian National Research Council

Nicaraguan stone figurine

Painted Mallard duck

Modeling architecture

Left image: Paul Debevec, Camillo Taylor, Jitendra Malik (Berkely)

Right image: Chris Haley (Berkeley)

- Virtual walk-throughs and walk-arounds
- Real estate advertising
- Trying virtual furniture

Model of Berkeley Campanile

Model of interior with artificial lighting

Modeling environments

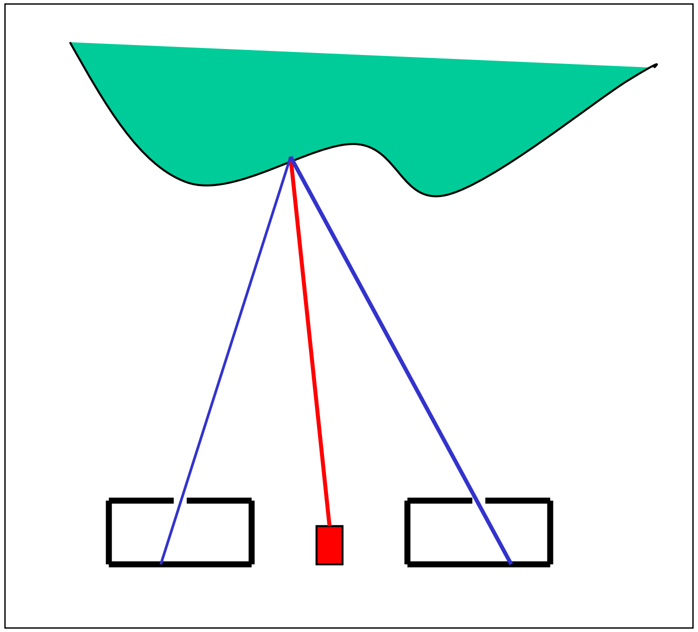
- Virtual walk-throughs and walk arounds
- Urban planning

Two renditions of model of MIT campus
(Seth Teller, MIT)

2. Sensors

Need to acquire data on shape and "color"

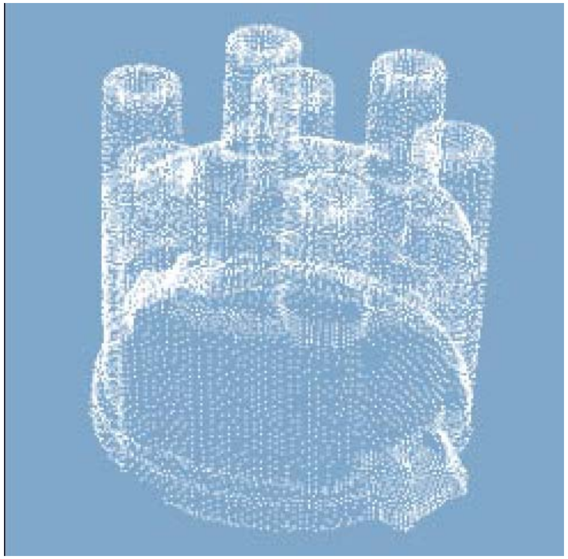
Simplest idea for shape: Active light scanner using triangulation



Laser spot on object allows matching of image points in the cameras

Cyberware scanner

Scanner output



A more substantial engineering effort:

The Cyberware Full Body Scanner

“Color” acquisition

Through digital photography

Need to register images to geometry

Watch out! “Color” can mean:

- RGB value for each surface point
- RBG value for each surface point and viewing direction
- BRDF (allows re-lighting)

Will return to this point later

Output of sensing process

- 1,000's to 1,000,000's of surface points which we assemble into triangular mesh
- Collection of ~700 images taken from different directions

Mesh generated from fish scans

Interlude: What does 3D photography have to do with this workshop?

- We estimate manifolds from data – 2D, but complex geometry and topology.
- We use multi-resolution representation of shape and “color”.
- We estimate radiance – a function on surface with values in function space. For every surface point we have function that assigns RGB values to directions.

How did we come to work on this problem?

Earlier methodological work (with Trevor Hastie) on principal curves – find a curve that “goes through the middle of a data set.”

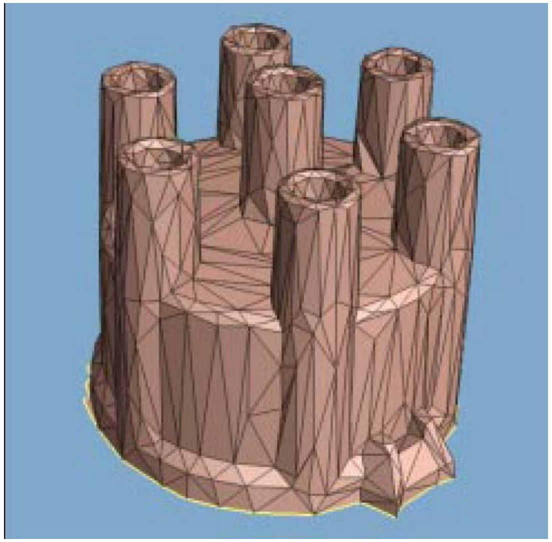
Theoretical work on principal curves and surfaces using calculus of variations.

Where might principal surfaces be useful??

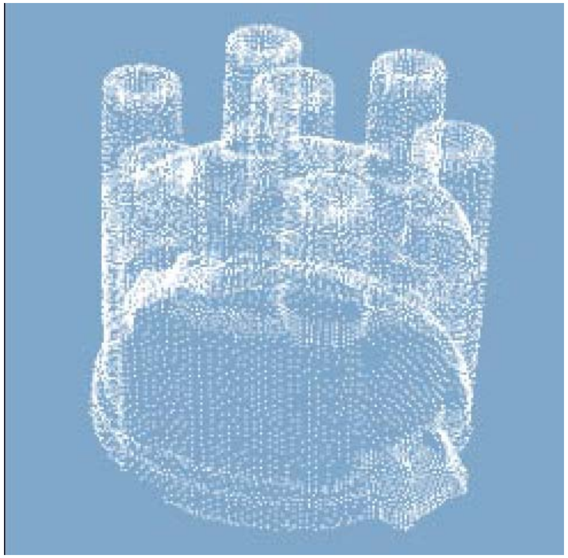
3. Modeling shape

Why not stick with meshes ?

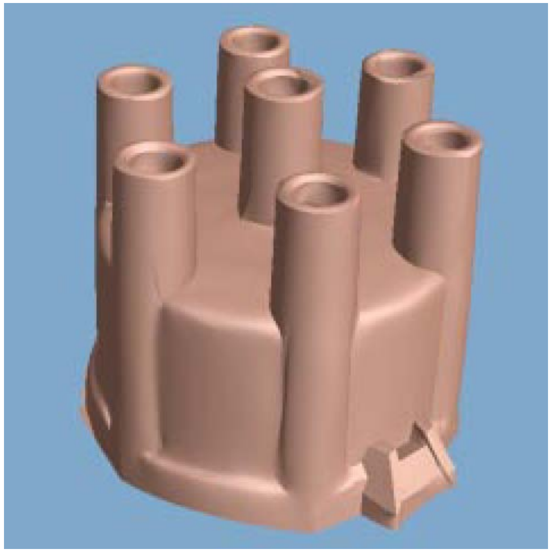
- Real world objects are often smooth or piecewise smooth
- Modeling a smooth object by a mesh requires lots of small faces
- Want more parsimonious representation



Fitted mesh



Sensor data



Fitted subdivision surface

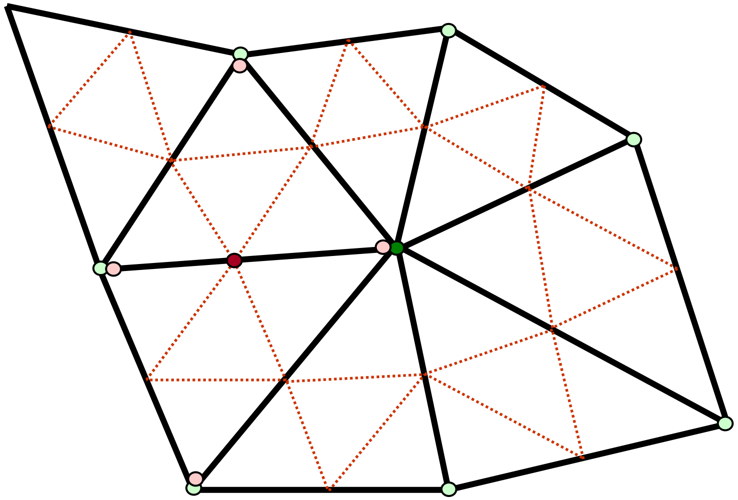
Subdivision surfaces
(Catmull – Clark, Loop)

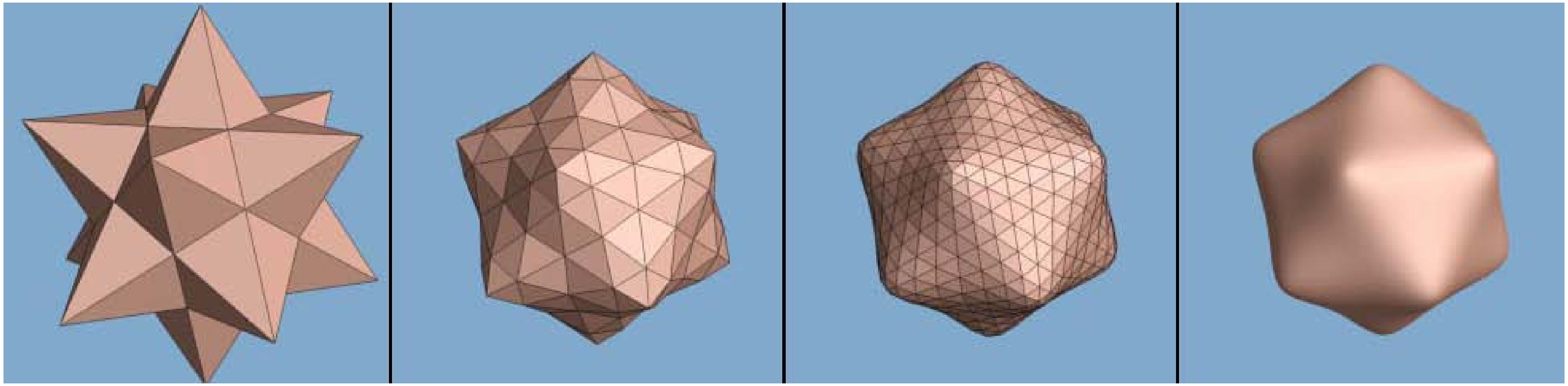
Defined by limiting process, starting with control mesh (bottom left)

Split each face into four (right)

Reposition vertices by local averaging

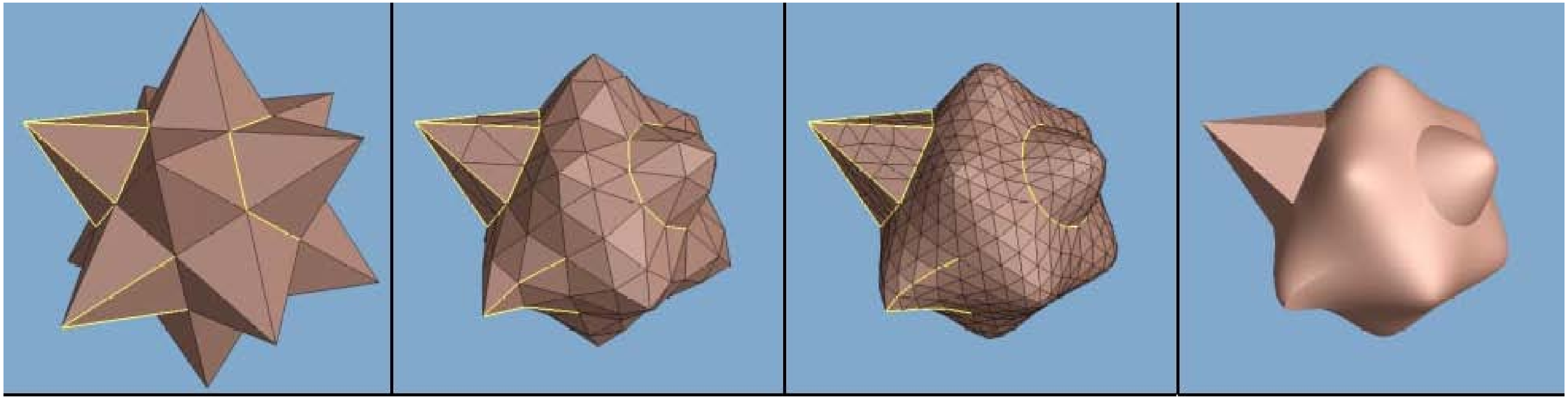
Repeat the process

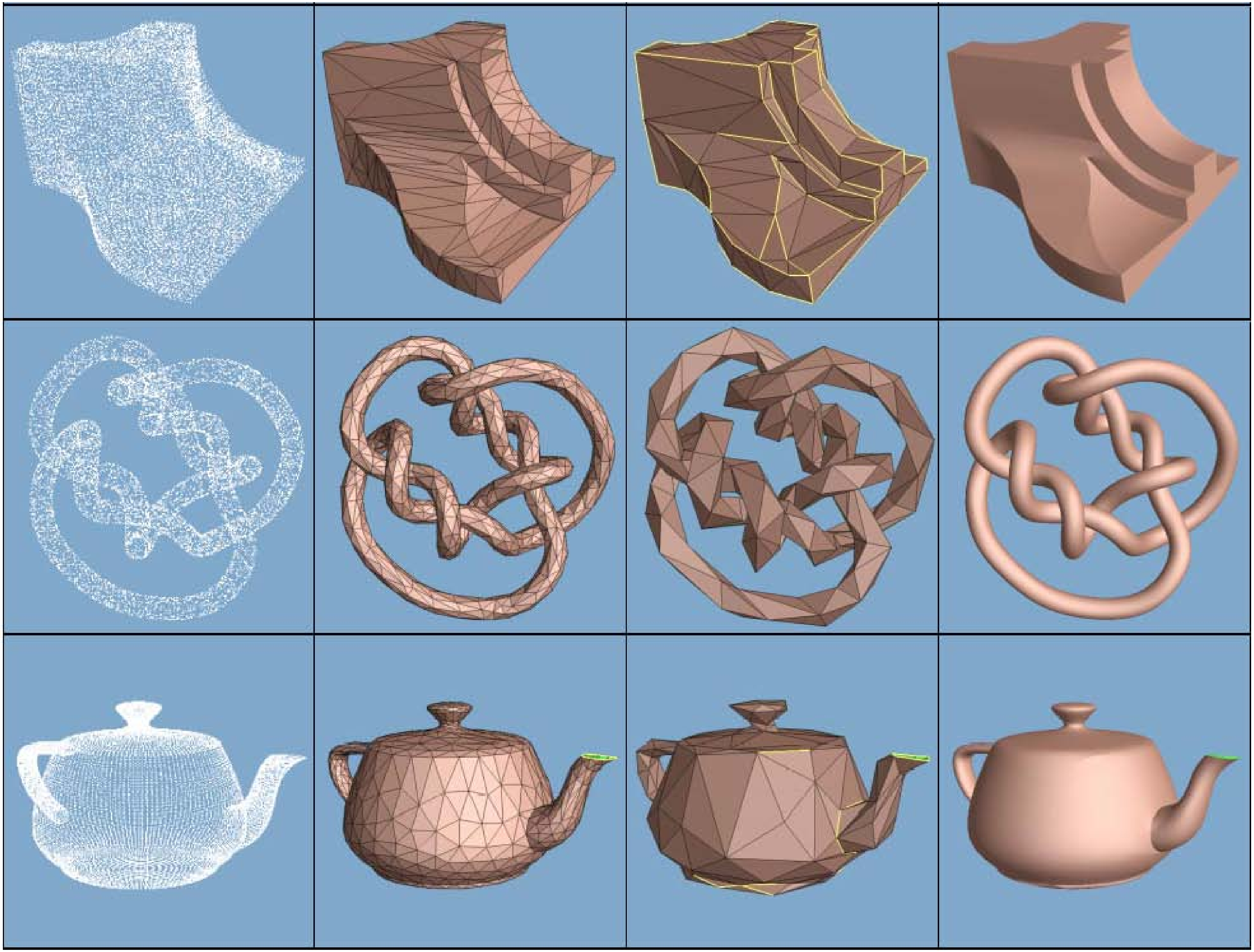




Remarks

- Limiting position of each vertex is weighted mean of control vertices.
- Important question: what choices of weights produce smooth limiting surface ?
- Averaging rules can be modified to allow for sharp edges, creases, and corners (below)
- Fitting subdivision surface to data requires solving nonlinear least squares problem.

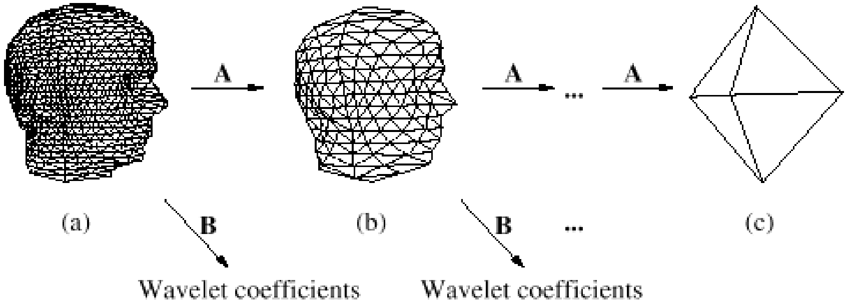




4. Parametrization and multiresolution analysis of meshes

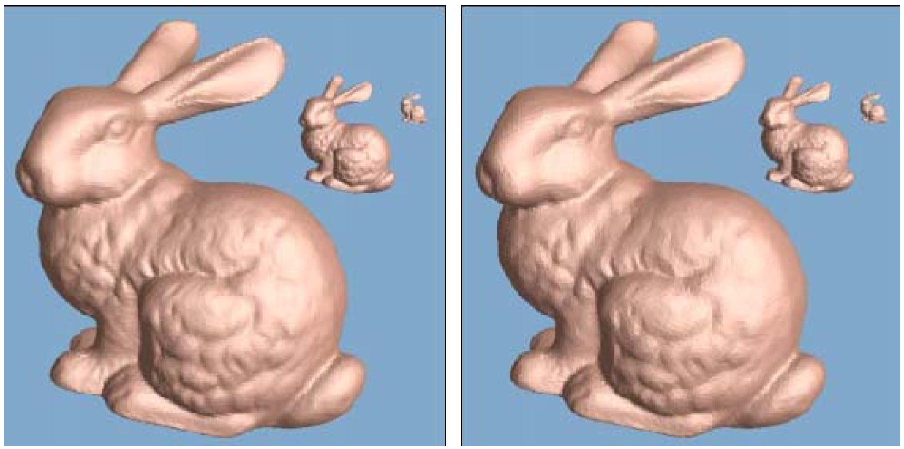
Idea:

Decompose mesh into simple “base mesh” (few faces) and sequence of correction terms of decreasing magnitude



Motivation:

- Compression
- Progressive transmission
- Level-of-detail control
  - Rendering time ~ number of triangles
  - No need to render detail if screen area is small

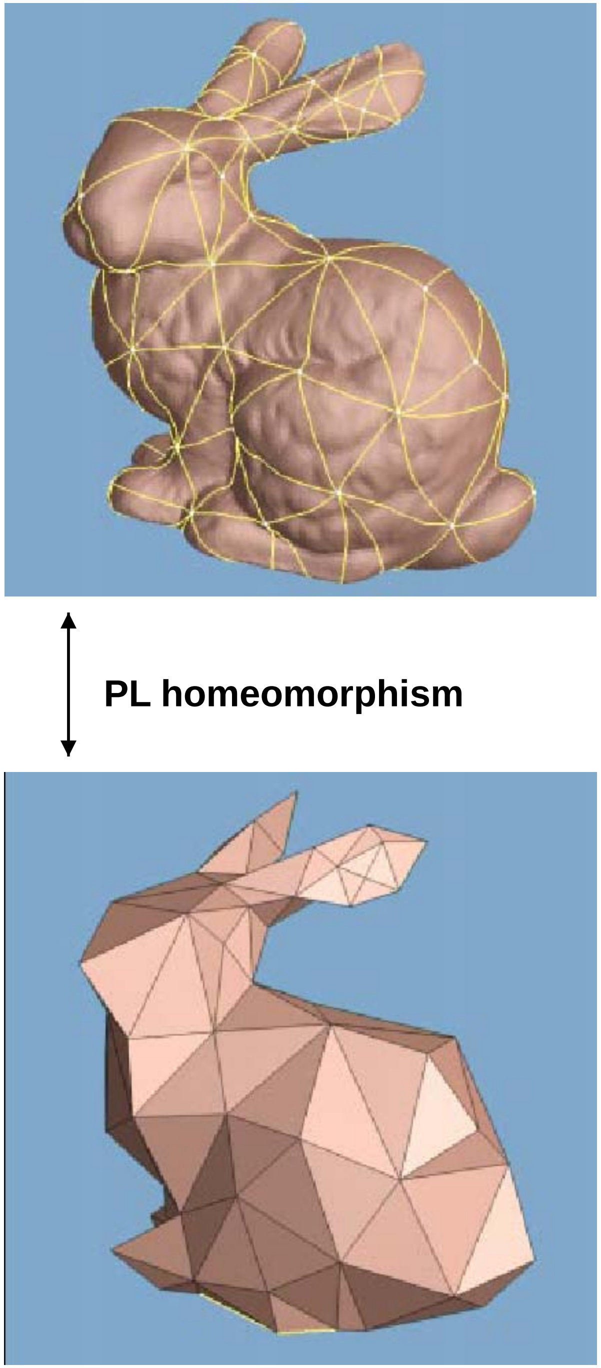


Full resolution
70K faces

LoD control
38K - 4.5K - 1.9K faces

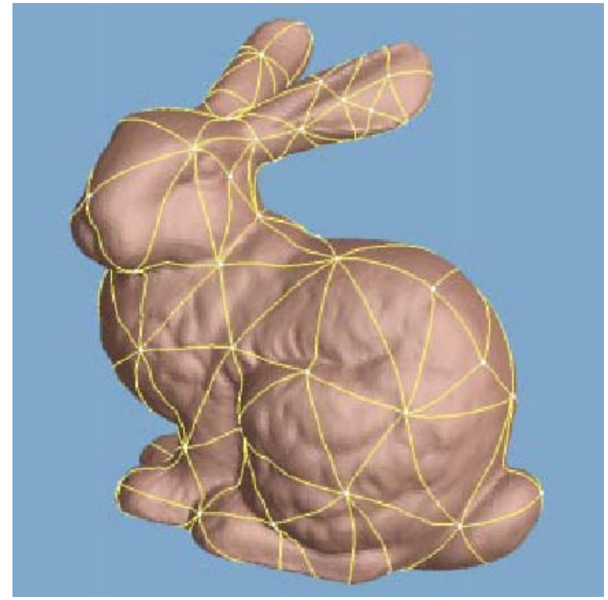
Procedure
(“computational differential geometry”)

- Partition mesh into triangular regions, each homeomorphic to a disk
- Create a triangular “base mesh”, associating a triangle with each of the regions
- Construct a piecewise linear homeomorphism from each region to the corresponding base mesh face
- Now we have representation of original as vector-valued function over the base mesh
- Natural multi-resolution sequence of spaces of PL functions on base mesh induced by 1-to-4 splits of triangles.
- (Lot of work…)

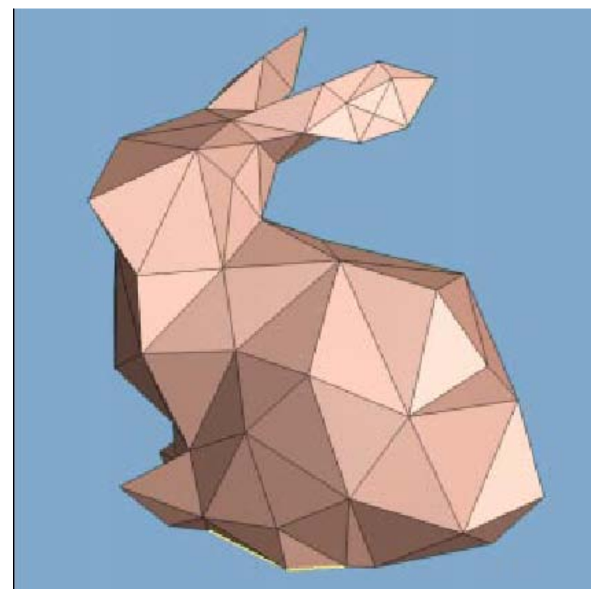


Texture mapping

- Homeomorphism allows us to transfer color from original mesh to base mesh
- This in turn allows us to efficiently color low resolution approximations (using texture mapping hardware)
- Texture can cover up imperfections in geometry



PL homeomorphism



Mesh doesn’t much look like face, but…

What would it look like without texture ?

5. Modeling of surface light fields

Motivation

- Real objects don't look the same from all directions (specularity, anisotropy)
- Ignoring these effects makes everything look like plastic.

Goal

Generate compact model that can be rendered in real time.

What we would see if we walked around the object

Model

Appearance of object under fixed lighting is captured by surface light field (SLF)

$$L : M \to C(S^2, R^3) : p \mapsto L_p \,,$$

which assigns a function from the sphere into RGB to each surface point.

Need to estimate function on manifold from scattered data.

From surface light field can synthesize image of object from any view point.

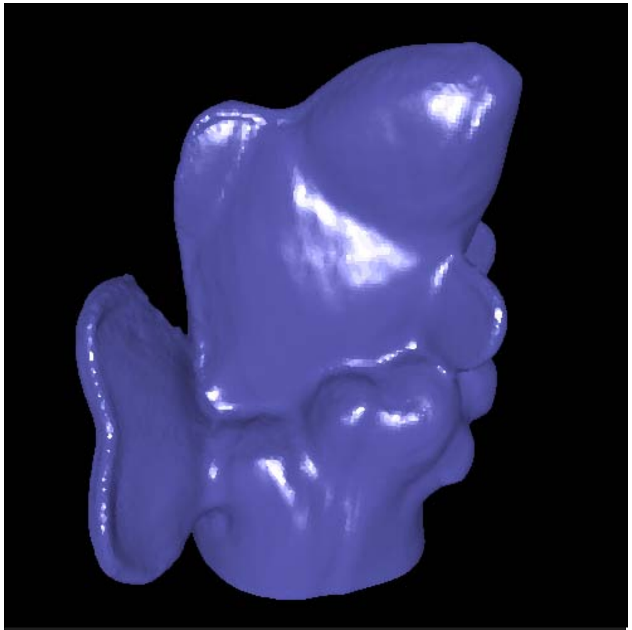
Data

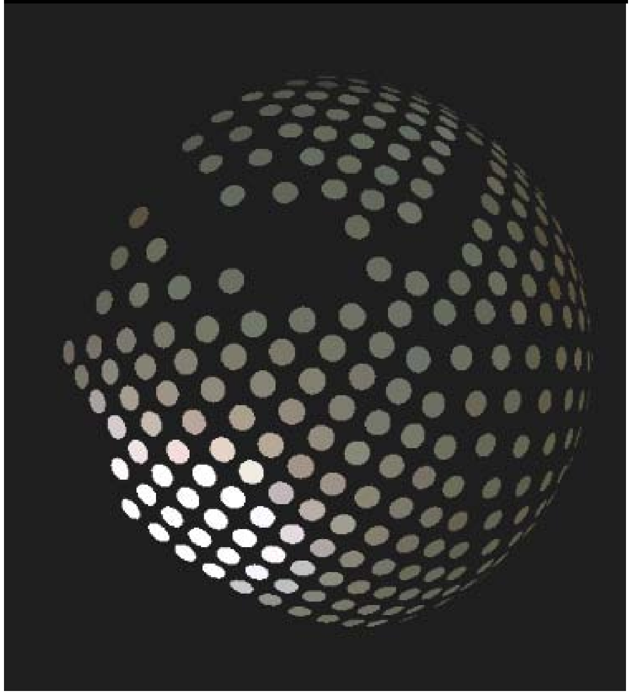
- Mesh representing geometry.
- Images taken from many different viewpoints

Let $\underline{x}_i$ be the vertices of the mesh.

Associate each vertex with "data lumisphere" L_i.

Data lumispheres $\approx$ 500MB





Problems:

- Choose computational representation of SLF;
- Fit surface light field to data lumispheres.

Computational representation of SLF

Represent lumispheres by functions in $C_{PL}(S^2, R^3) \subset C(S^2, R^3)$, the space of piecewise linear functions on a 3-fold subdivided octahedron (dim $= 258$).

Represent SLF by a piecewise linear function $L : M \to C_{PL}(S^2, R^3)$.

Fitting the SLF

Naive idea: For each vertex $\underline{x}_i$, fit lumisphere $\hat{L}_i \in C_{PL}(S^2, R^3)$ to data lumisphere L_i.

Problem: Huge model size.

Fitting the SLF (II)

Better idea

Represent the lumispheres $\hat{L}_i$ at the vertices by convex combinations of a small number of prototypes $F = (F_1, \ldots, F_p)$:

$$\hat{L}_i = \sum_j c_{ij}\, F_j$$

Define figure of merit $E(F)$:

$$E(F) = \sum_i \left(\text{argmin}_c \| L_i - \sum_j c_{ij}\, F_j \|^2 \right) + \text{smoothness penalty for } F\,.$$

Find $\hat{F} = \text{argmin}_F E(F)$ by alternating optimization.

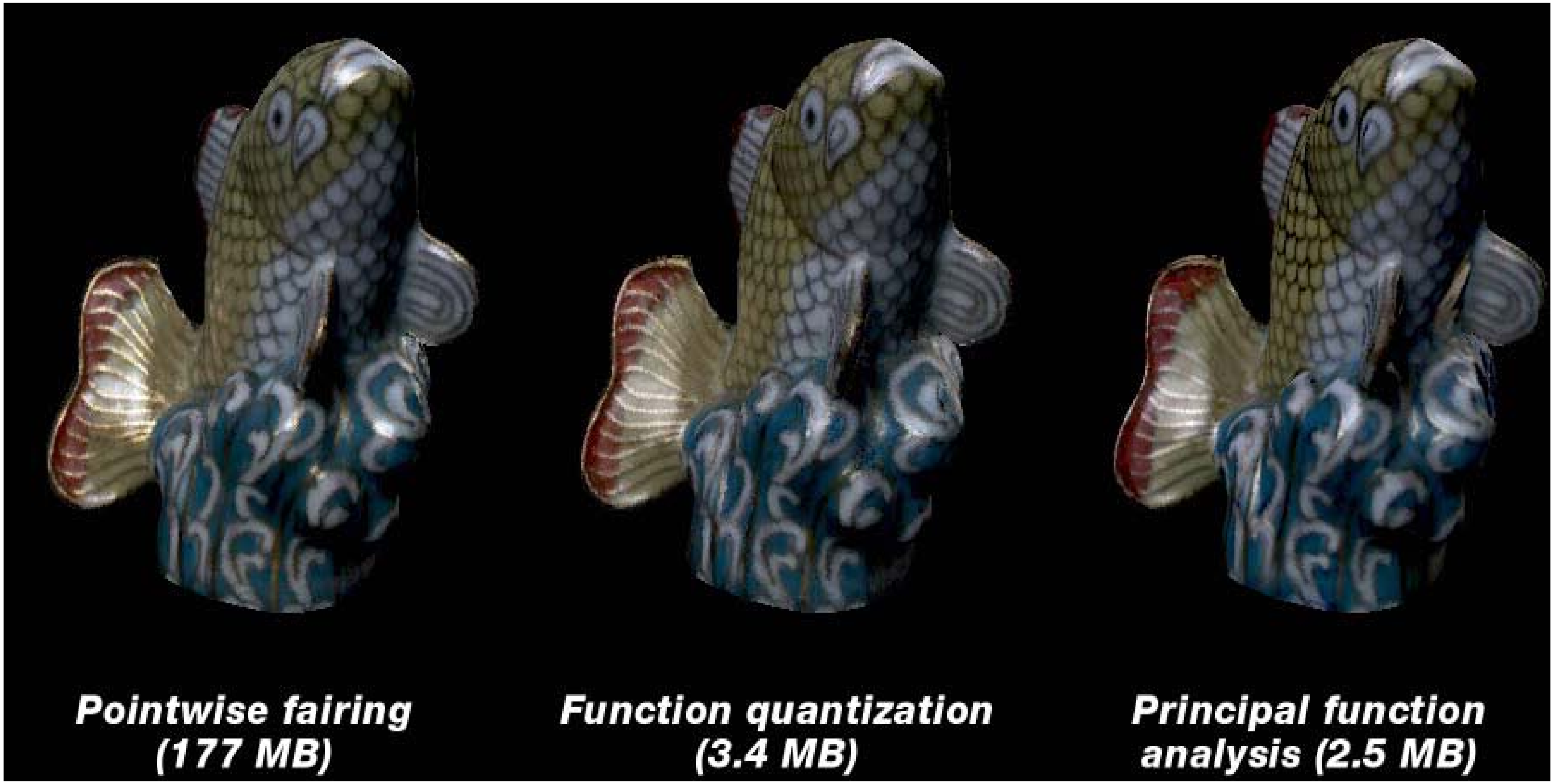
Note: Same idea as principal component analysis, but for irregular sampling with missing data.

Results (i)

Real vs synthesized image.

Results (ii)

Uncompressed vs compressed.



Comments and Conclusions

Principal component analysis results in huge compression without preceptible loss of quality.

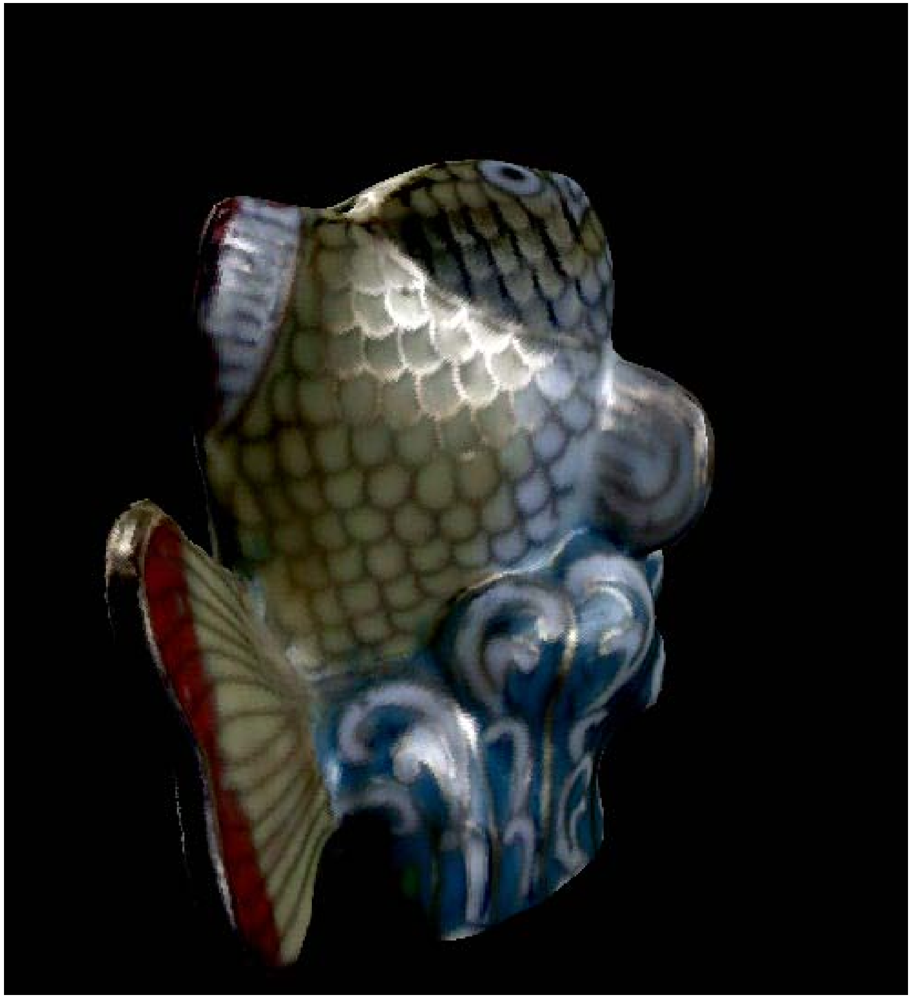
Glossed over some important steps in compression, for example:

- Before compressing lumispheres, subtract out diffuse colors.
- Transformed data using physical properties about reflectance.

Without these steps, which are motivated by the specific problem, performance degrades signficantly.

Extensions

- Better methods for principal functions analysis;
- Improve estimated object geometry using image information (Faugeras, Osher)
- 3D Photoshop



Thanks for your interest

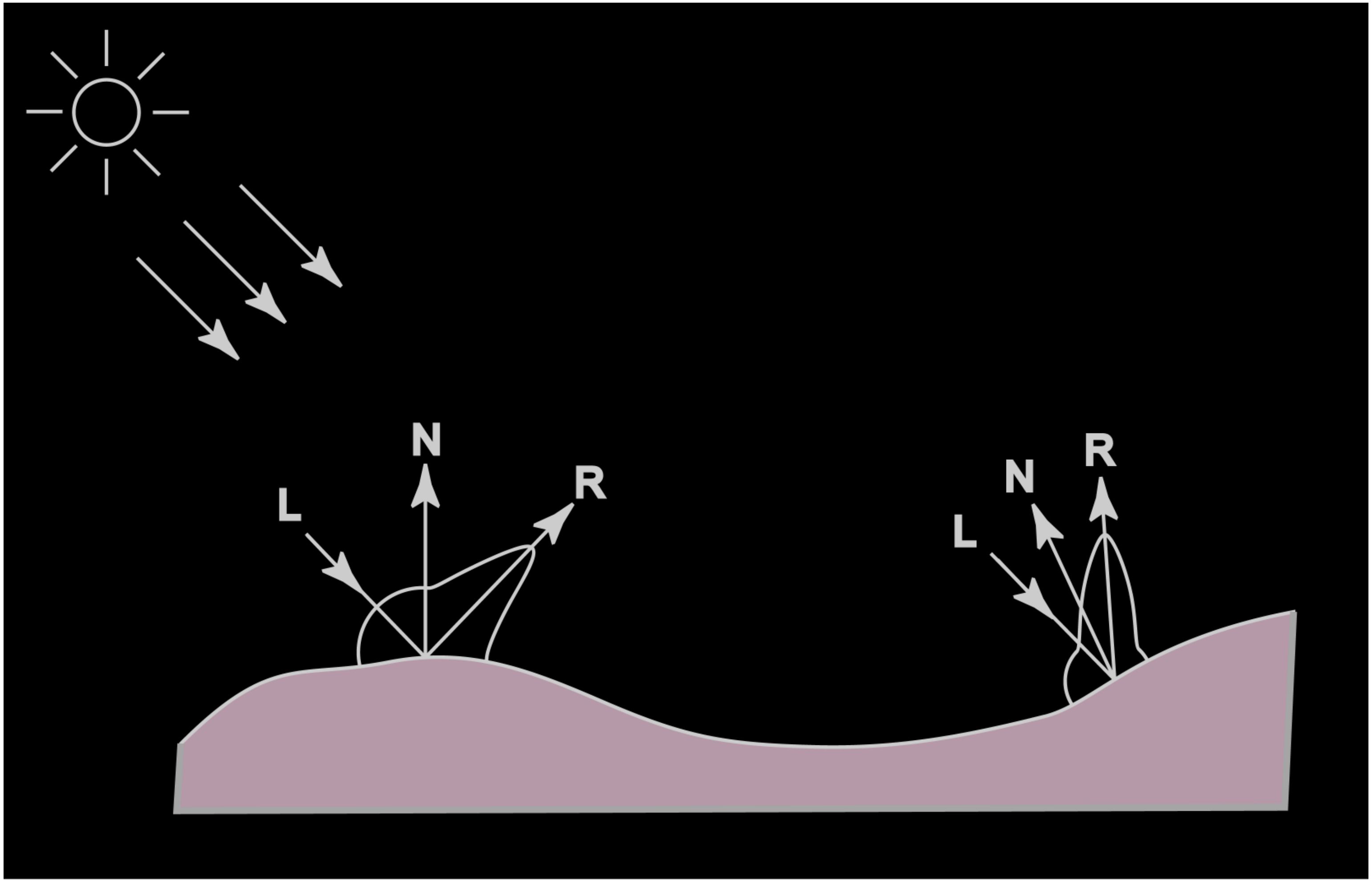
Another example

Note automatic imputation of missing values

Naïve idea: Associate color with direction of reflected light

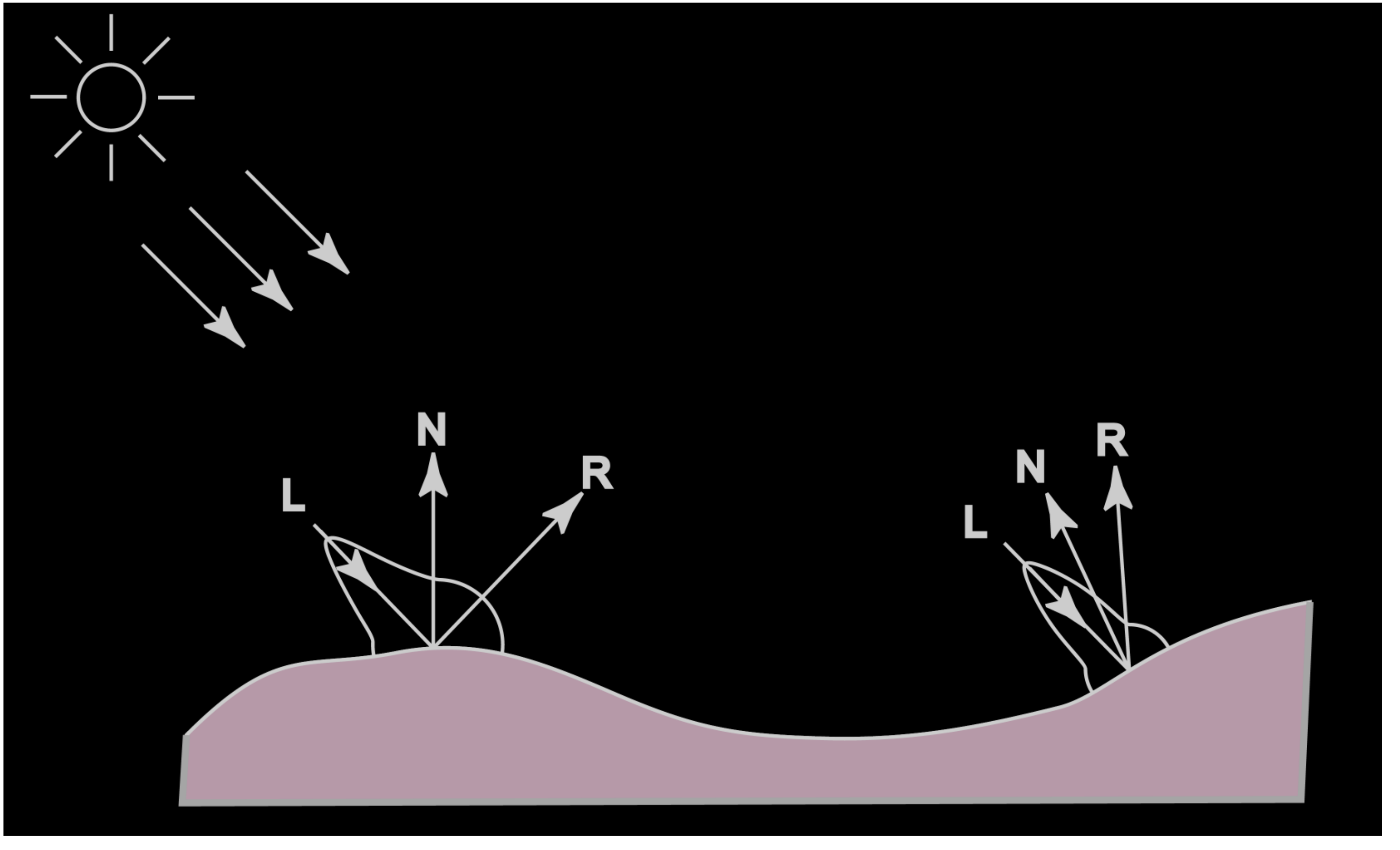
Better idea: Associate color with direction of incoming light.

- Higher coherence between points on surface
- Lumisphere can be easily obtained by reflecting around normal.



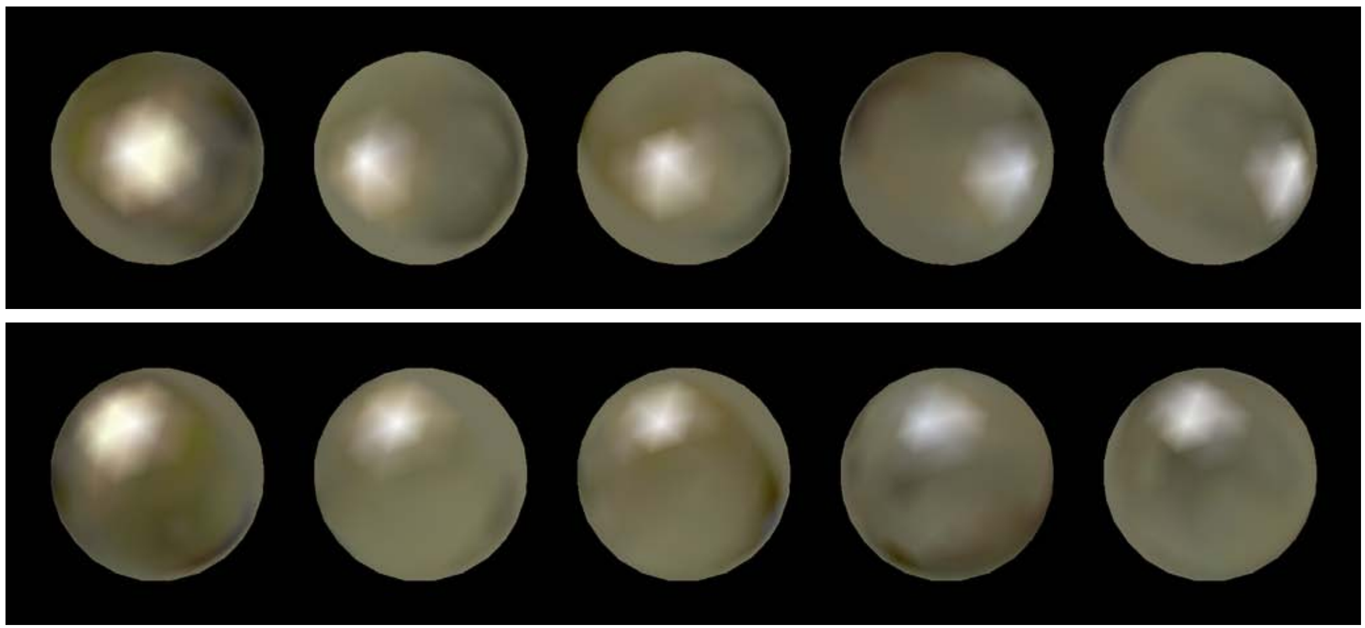
Naïve idea: Associate color with direction of reflected light

Better idea: Associate color with direction of incoming light.
Higher coherence between points on surface
Lumisphere can be easily obtained by reflecting around normal.



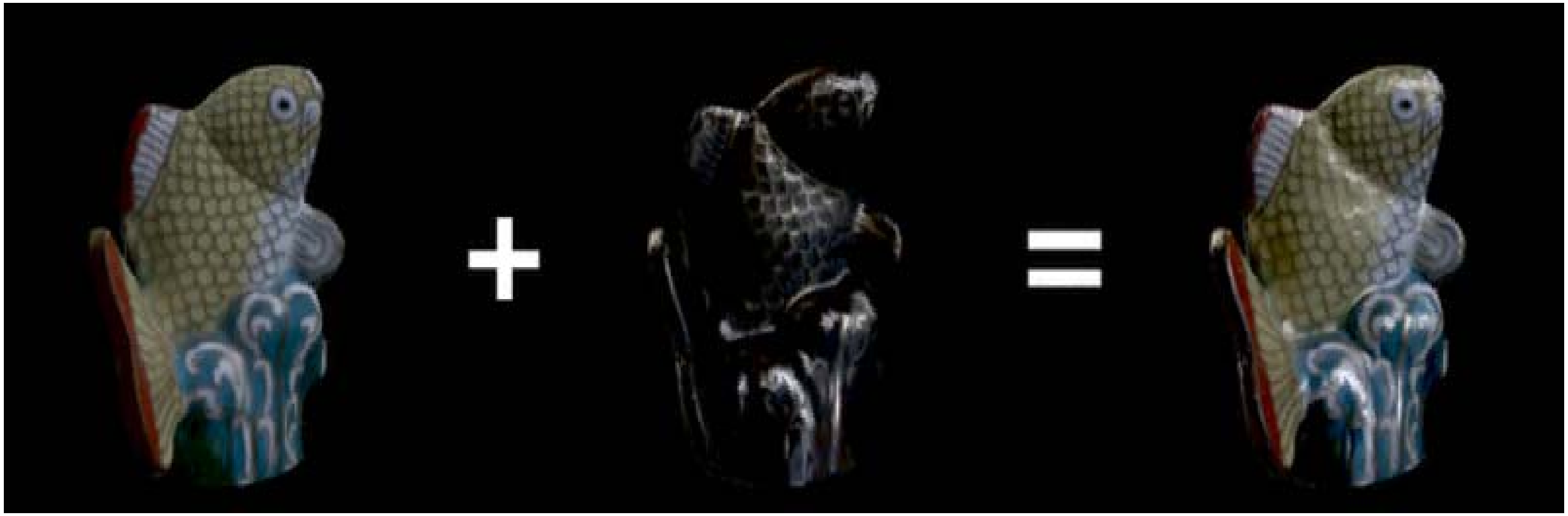
Reflected reparameterization

Before



After

Median removal



Median values Specular Result

Geometry (fish)

Reconstruction: 129,000 faces

Memory for reconstruction: 2.5 MB

Base mesh: 199 faces

Re-mesh (4x subdivided): 51,000 faces

Memory for re-mesh: 1 MB

Memory with view-dependence: 7.5 MB

Compression (fish)

Pointwise faired:

Memory = 177 MB RMS error = 9

FQ (2000 codewords)

Memory = 3.4 MB RMS error = 23

PFA (dimension 3)

Memory = 2.5 MB RMS error = 24

PFA (dimension 5)

Memory = 2.9 MB RMS error = ?

Breakdown and rendering (fish)

For PFA dimension 3…

Direction mesh: 11 KB

Normal maps: 680 KB

Median maps: 680 KB

Index maps: 455 KB

Weight maps: 680 KB

Codebook: 3 KB

Geometry w/o view dependence: <1 MB

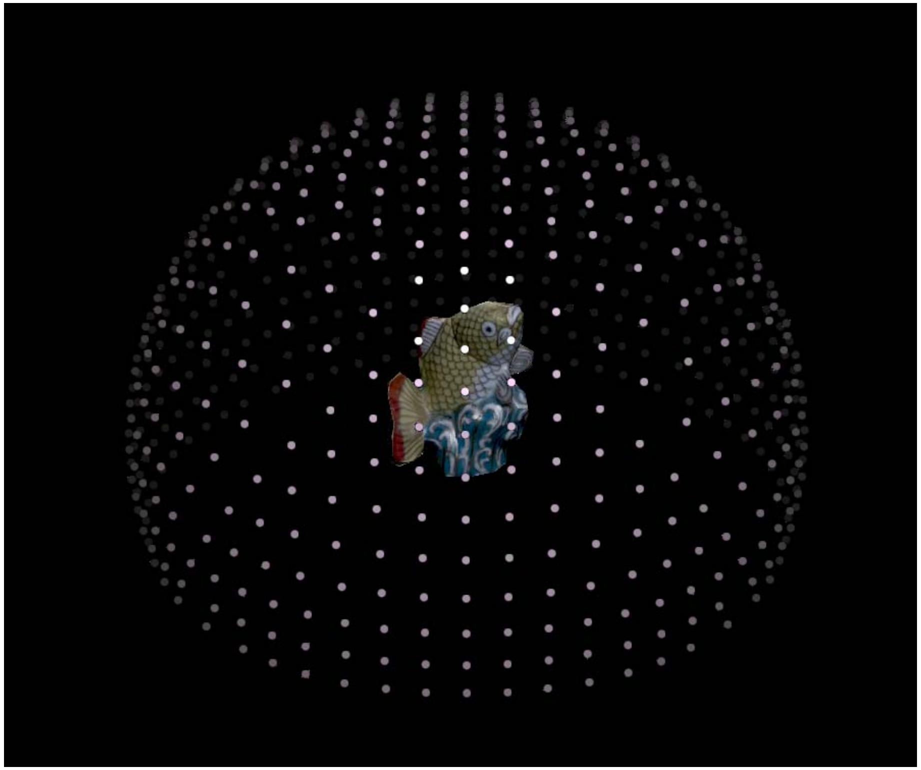
Geometry w/ view dependence: 7.5 MB

Rendering platform: 550 MHz PIII, linux, Mesa

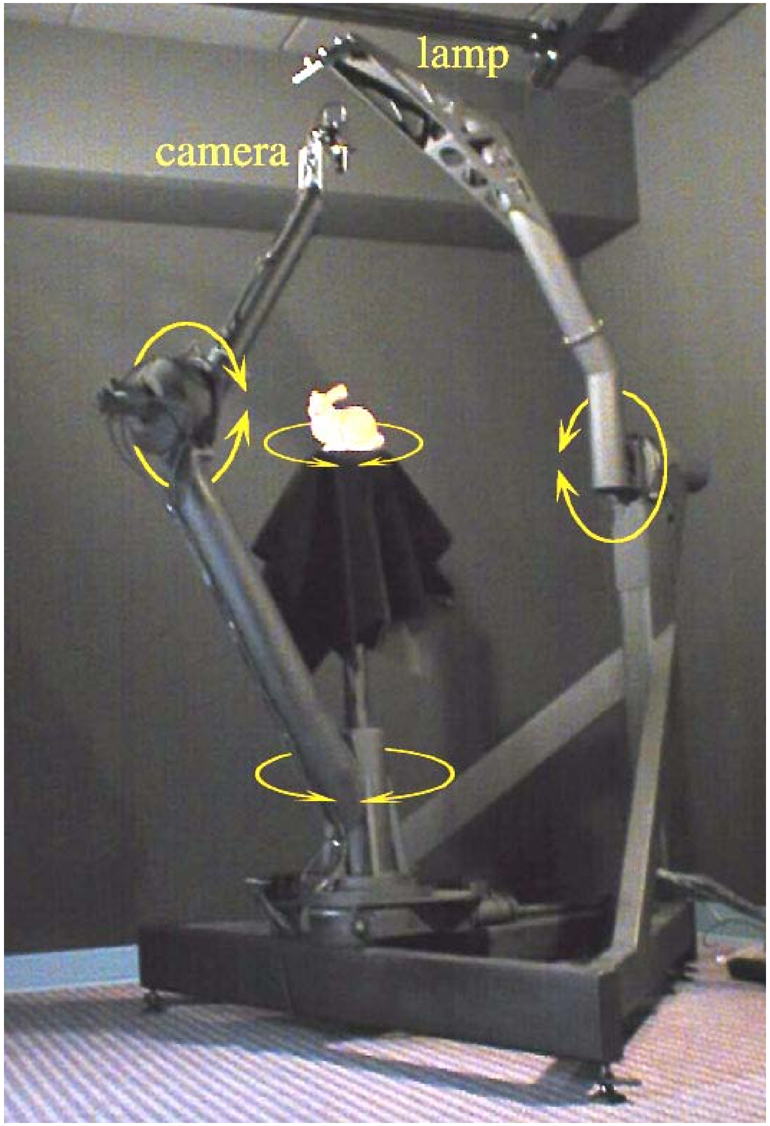
Rendering performance: 6-7 fps (typical)

Data acquisition (ii)

Take photographs



Camera positions



Stanford Spherical Gantry

6. Smoothing on 2D manifolds

Given: Training data $(\underline{x}_1, y_1), \ldots, (\underline{x}_n, y_n)$ with $\underline{x}_i$ in some domain M and $y_i \in \mathbf{R}$.

Assumption: $y_i = f_{true}(\underline{x}_i) + \epsilon_i$.

Goal: Estimate f_{true}.

Well established method for $M = \mathbf{R}$: **spline smoothing**

Smoothing spline f_λ minimizes

$$E[f] = \frac{1}{n} \sum (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

in the Sobolev space of functions with square integrable second derivative.

Spline smoothing on 2D manifold M

Replace f'' by Laplace-Beltrami operator $\Delta_M f$.

Find function f_λ minimizing spline functional

$$E[f] = \frac{1}{n}\sum (y_i - f(x_i))^2 + \lambda \int_M (\Delta_M f)^2 dA$$

in the Sobolev space $W_2(M)$ of functions with square integrable second derivative.

No closed form solution except in special cases (line, sphere, torus)

$\Rightarrow$ use finite elements.

Approximation of smoothing splines by finite elements

Suppose we have multi-resolution sequence of finite-dimensional function spaces

$$V^0 \subset V^1 \subset V^2 \subset \cdots \subset W_2(M)$$

whose union is dense in $W_2(M)$.

Can then approximate f_λ by choosing resolution level J and minimizing $E[f]$ over V^J.

Nested spaces of subdivision functions

Suppose M is a subdivision surface parametrized over a polyhedron K with triangular faces.

To define a resolution level 0 subdivision function, start with a function f^0_{PL} that is piecewise linear on $K^0 = K$, with values f_α at the vertices.

The function f^J_{PL} is piecewise linear on K^J (obtained by J 4-1 splits of K^0).

The values of f^{J+1}_{PL} at the vertices of K^{J+1} are obtained by

- Up-sampling f^J_{PL} to the vertices of K^{J+1};
- Local averaging.

The subdivision function defined by f^0_{PL} is the limit of this process.

The resolution level 0 subdivision functions form a vector space V^0 with dimension = number of vertices of K.

Resolution level J subdivision functions are obtained by

- fixing the values of a piecewise linear function at the vertices of K^J
- running the subdivision process.

Averaging rules have to be carefully crafted to make subdivision functions "smooth".

If we embed K into R^3 using subdivision functions, the resulting surface is smooth (essentially C^2).

Approximate calculation of smoothing splines

To find an approximate minimum for $E[f]$, choose a resolution level J and express $f(x)$ as a finite sum

$$f(x) = \sum_{\alpha} f_\alpha \phi_\alpha^J(x)\,,$$

where α ranges over the basis functions.

Substituting into the formula for the spline functional $E[f]$ gives

$$E[f] = \frac{1}{n}\sum_i \left(y_i - \sum_\alpha f_\alpha \phi_\alpha^J(x_i) \right)^2 + \lambda \sum_{\alpha,\beta} f_\alpha f_\beta B_{\alpha,\beta}\,,$$

with

$$B_{\alpha,\beta} = \int_M \Delta_M \phi_\alpha^J \, \Delta_M \phi_\beta^J \, dA\,.$$

We solve the resulting linear algebra problem using preconditioned conjugate gradients.

Numerical experiment

Spline smoothing on the sphere $\Rightarrow$
Know exact solution $\Rightarrow$
Can assess accuracy of finite element approximation.

1. Generate test function

- Generate 100 points uniformly over sphere
- Simulate 100 standard Gaussian function values
- Test function f_{true} = interpolating spline

2. Generate data

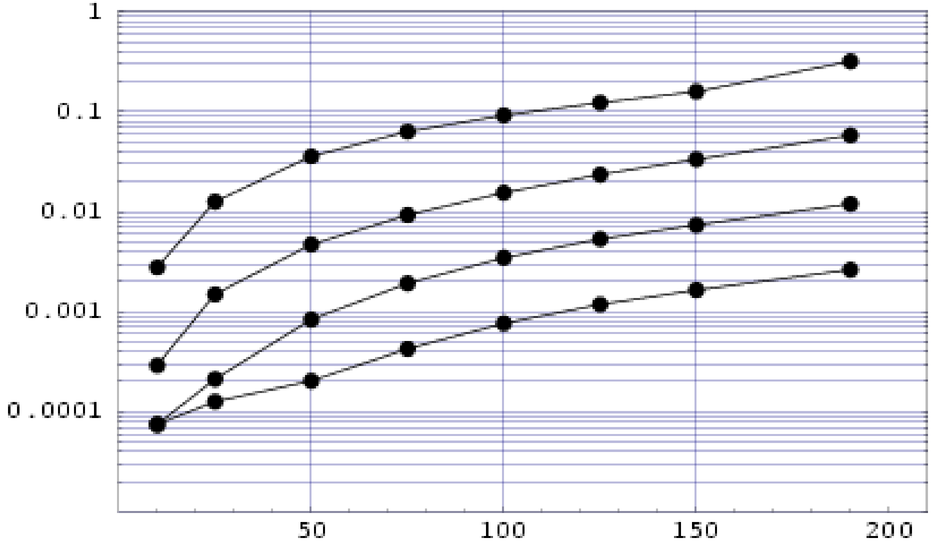
- $\underline{x}_1, \ldots, \underline{x}_{200}$ uniform over sphere
- $y_i = f_{true}(\underline{x}_i)$

3. Find exact smoothing splines for range of values of λ.

4. Approximate sphere by subdivision surface.

5. Find approximate smoothing splines by finite elements.

Figure shows relative approximation error $\frac{\|f_\lambda - f_{\lambda,J}\|}{\|f_\lambda\|}$ as a function of λ and subdivision levels $J = 3, 4, 5, 6$



For fixed λ error decreases exponentially with subdivision level (in agreement with theoretical result by G. Arden (2001)

For fixed subdivision level error increases for decreasing λ (moving right on horizontal axis).