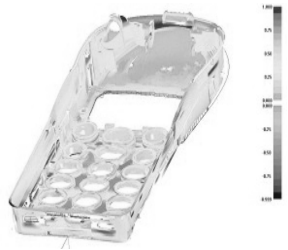
3D Photography

Obtaining 3D shape (and sometimes color)
of real-world objects

Based on slides from Szymon Rusinkiewicz

Industrial Inspection

- Determine whether manufactured parts are within tolerances

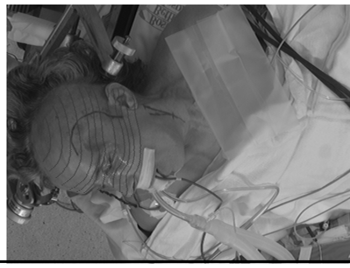


Medicine

- Plan surgery on computer model, visualize in real time

Medicine

- Plan surgery on computer model, visualize in real time

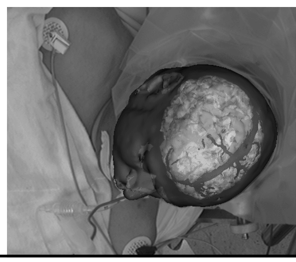


Medicine

- Plan surgery on computer model, visualize in real time

Medicine

- Plan surgery on computer model, visualize in real time

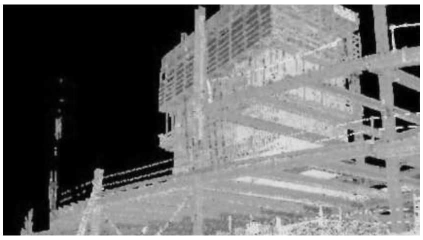


Scanning Buildings

- Quality control during building
- As-build models

Scanning Buildings

- Quality control during building
- As-build models

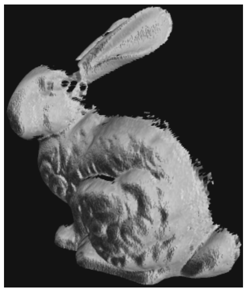


Clothing

- Scan a person, custom-fit clothing
- U.S. Army; booths in malls

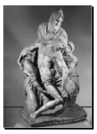
Graphics Research

- Availability of complex datasets drives research
  (you wouldn't *believe* how the poor bunny has been treated…)



Sculpture Scanning

- The Pietà Project
  IBM Research



- The Digital Michelangelo Project
  Stanford University

- The Great Buddha Project
  University of Tokyo

Why Scan Sculptures?

- Interesting geometry
- Introduce scanning to new disciplines
  - Art: studying working techniques
  - Art history
  - Cultural heritage preservation
  - Archeology
- High-visibility projects

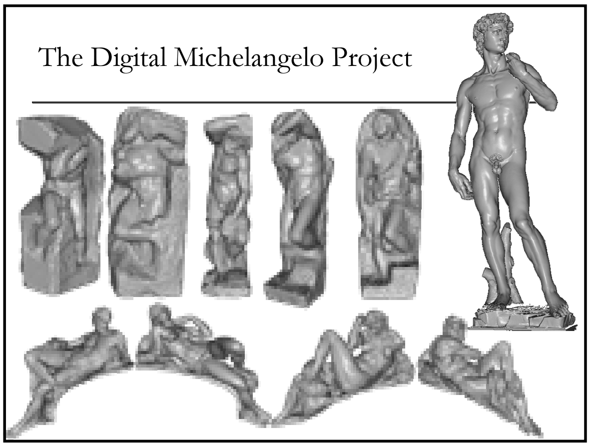
Why Scan Sculptures?

- Challenging
  - High detail, large areas
  - Large data sets
  - Field conditions
  - Pushing hardware, software technology
- But not too challenging
  - Simple topology
  - Possible to scan most of surface

Issues Addressed

- Resolution
- Coverage
  - Theoretical: limits of scanning technologies
  - Practical: physical access, time
- Type of data
  - High-res 3D data vs. coarse 3D + normal maps
  - Influenced by eventual application
- Intellectual Property

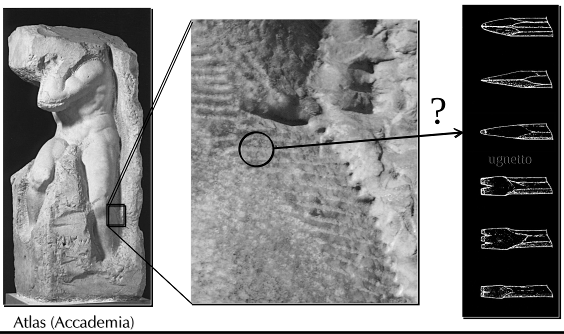
The Digital Michelangelo Project



Goals

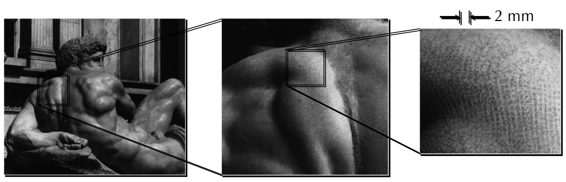
- Scan 10 sculptures by Michelangelo
- High-resolution (“quarter-millimeter”) geometry
- Side projects: architectural scanning (Accademia and Medici chapel), scanning fragments of Forma Urbis Romae

Why Capture Chisel Marks?



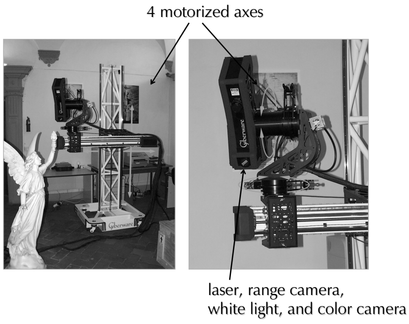
Atlas (Accademia)

Why Capture Chisel Marks?

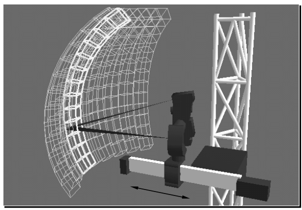


Day (Medici Chapel)

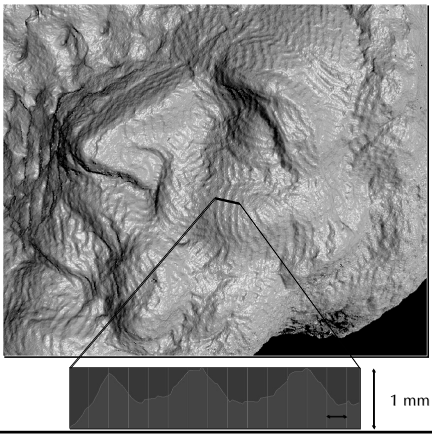
Scanner Design



Scanning a Large Object



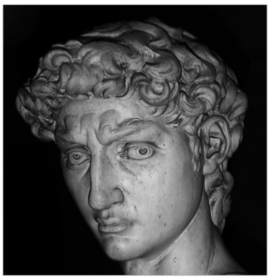
- Calibrated motions
  - pitch (yellow)
  - pan (blue)
  - horizontal translation (orange)
- Uncalibrated motions
  - vertical translation
  - rolling the gantry
  - remounting the scan head



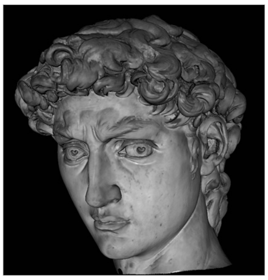
Statistics About the Scan of David

- 480 individually aimed scans
- 0.3 mm sample spacing
- 2 billion polygons
- 7,000 color images
- 32 gigabytes
- 30 nights of scanning
- 22 people

Head of Michelangelo's David

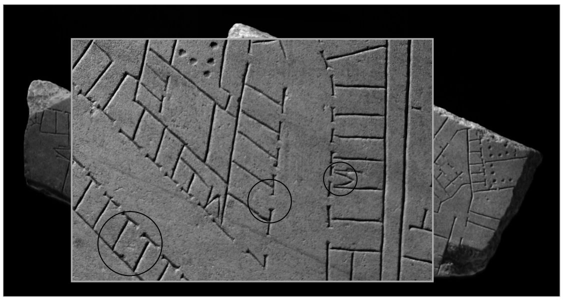


Photograph

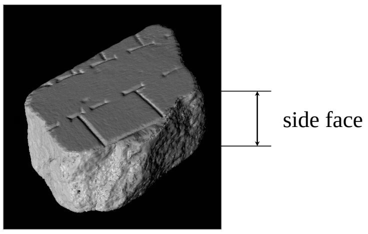


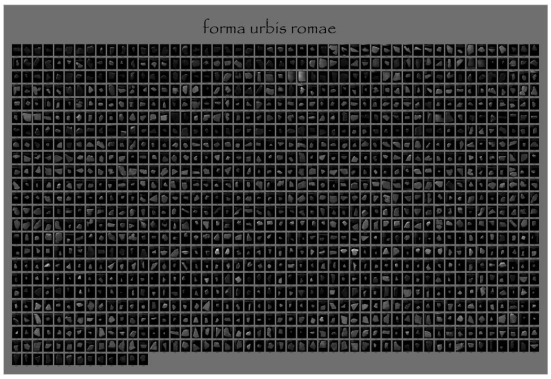
1.0 mm computer model

Side project:
The Forma Urbis Romae



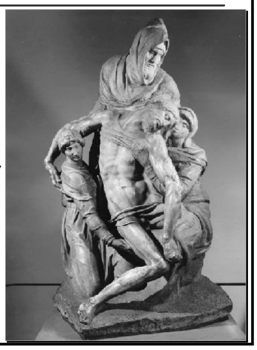
Forma Urbis Romae Fragment





IBM's Pietà Project

- Michelangelo's "Florentine Pietà"
- Late work (1550s)
- Partially destroyed by Michelangelo, recreated by his student
- Currently in the Museo dell'Opera del Duomo in Florence

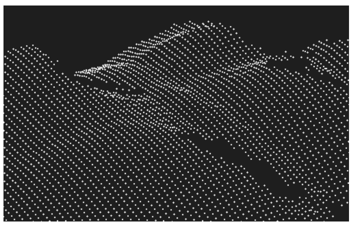


Results

The Great Buddha Project

- Great Buddha of Kamakura
- Original made of wood, completed 1243
- Covered in bronze and gold leaf, 1267
- Approx. 15 m tall
- Goal: preservation of cultural heritage

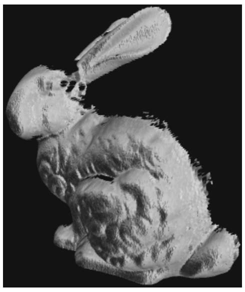
Single Laser Range Image



A single scan is a grid: Connect adjacent samples when z-difference is small.

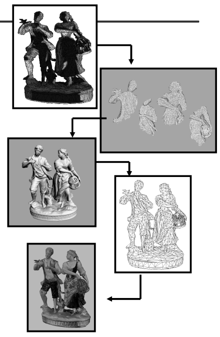
3D Scanning

The acquisition of a single **range map** is only an intermediate single step of the overall acquisition session

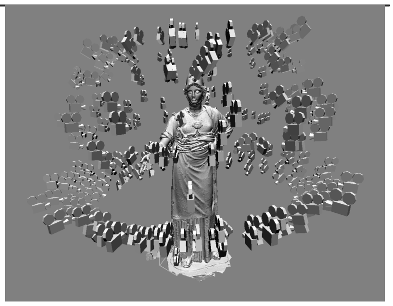


The 3D scanning pipeline

- **3D Scanning**:
  - *Acquisition planning*
  - **Acquisition** of multiple range maps
  - Range maps **Editing**
  - **Registration** of range maps
  - **Merge** of range maps
  - Mesh **Editing**
  - Geometry **simplification**
  - **Capturing appearance**
  - Archival and data conversion



Acquisition Planning



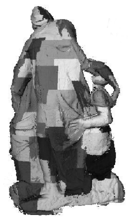
Selecting the set of views is not easy

NP-hard problem → *find good heuristics & approximate solutions*

Acquisition Planning

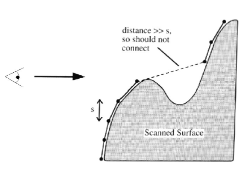
Definition of the **optimal acquisition patchwork**:

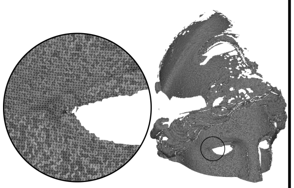
- Given: scanner & object characteristics
- Obtain an **optimal & complete** coverage (all object surface covered):
  - **Minimal number** of scans
  - Sufficient **inter-scan overlap** (registration)
  - Where each scan should be:
    - shot from a view direction **nearly orthogonal** to the surface
    - **physically feasible** (consider potential collisions with the object/environment, self-occlusions)
- NP-hard problem → find good heuristics & approximate solutions



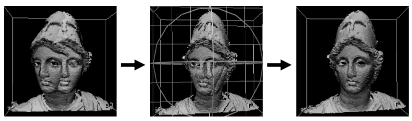
Range images

- Converting a range image into a range surface is easy.
- Use a **tessellation threshold**



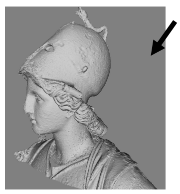


Registration and Merging



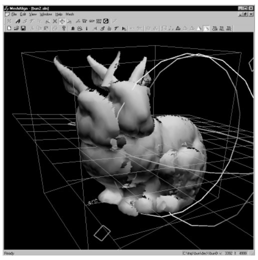
First: **Register** all range maps

Second: **Merge** in a single triangulated surface with no redundancy



Registration

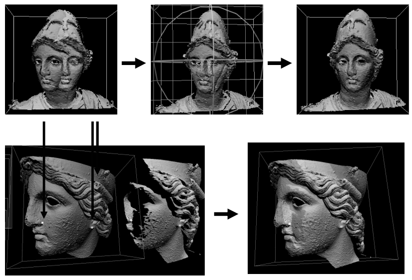
- Independent scans are defined in coordinate spaces which depend on the spatial locations of the **scanning unit** and the **object** at acquisition time. They have to be **registered** (roto-translation) to lie in the same space
- Standard approach:
  1. initial **manual** placement
  2. **Iterative Closest Point** (ICP) [Besl92,CheMed92]



Manuel Pairwise Registration

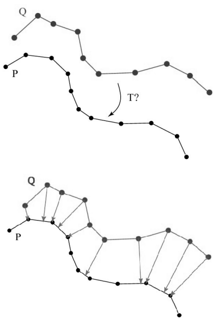
Mode 1) The user manually places a range map over another (interactive manipulation)

Mode 2) Selection of multiple pairs of matching points



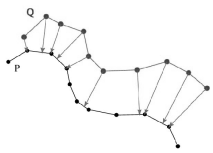
Pairwise Registration

- An approximation to the distance between range scans is:
  - $E = S \| T q_i - p_i \|^2$
- where the **qi** are samples from scan **Q** and the **pi** are the <u>*corresponding*</u> points of scan **P**.



Iterative Closet Point (ICP) [Besl+92]

- If the correspondences are known a priori, then there is a closed form solution for ***T***. However, the correspondences are not known in advance.
- **Iterative closest point** (ICP) [Besl+92]
  - Start from an approximate registration
  - Repeat
    - Identify corresponding points (minimal distance)
    - Compute and apply the optimal rigid motion ***T***
  - Until registration error **E** is small



Registration, many more issues

Pairwise Sequential vs. Global [Pulli99]

Using Color in registration [Bernardini00]

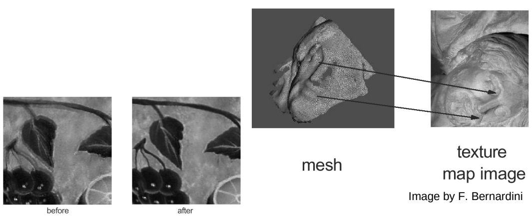
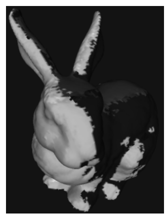


Image by F. Bernardini

So far…

- Mainly engineering problems, planning, scanning and registration.
- Now, once registered, all scans have to be fused in a single, continuous, hole-free mesh
- In other words – surface reconstruction…
- Or, surface consolidation



Consolidation

Desirable properties for surface reconstruction:

- No restriction on topological type
- Representation of range uncertainty
- Utilization of all range data (integrate over overlapped regions)
- Incremental and order independent updating
- Time and space efficiency
- **Robustness** (to noise)
- Ability to **fill holes** in the reconstruction

Methods that construct **triangle meshes directly**:

Reconstruction from point clouds

- *Local Delaunay triangulations [Boissonat84]*
- *Alpha shapes [Edelsbrunner+92]*
- *Crust algorithm [Amenta+98]*
- *Delaunay-based sculpturing [Attene+00]*
- *Ball Pivoting [Bernardini+99]*
- *Localized Delaunay [Gopi+00]*

Reconstruction from range maps

- *Re-triangulation in projection plane [Soucy+92]*
- *Zippering in 3D [Turk+94]*

Zippering [Turk+94]

- Redundancy removal and zippering

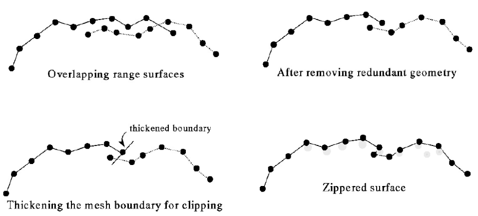
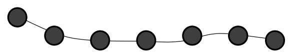


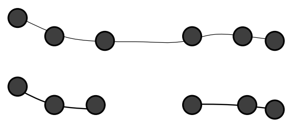
Image by Brian Curless, Sig'2000 Course Notes

Sampling quality and reconstruction issues

Ideal Sampling

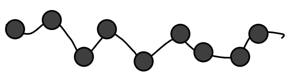


Uneven Sampling or holes?

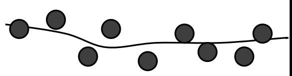


Sampling quality and reconstruction issues

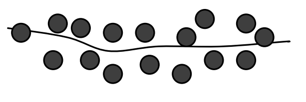
Interpolating?



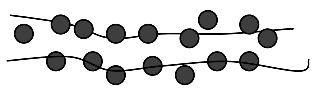
Or noisy sampling?



Sampling quality and reconstruction issues



Solid Object with thin section?



Sampling quality and reconstruction issues

Solid Object with small features?

Sampling quality and reconstruction issues

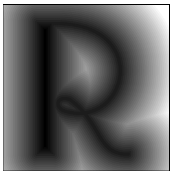
Smooth or Sharp features?

Implicit Surface Representation

Represented as a function f(x,y,z) = 0

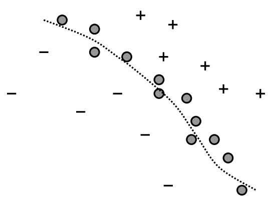
Volumetric representation

One important implicit function is the distance function



Distance Field

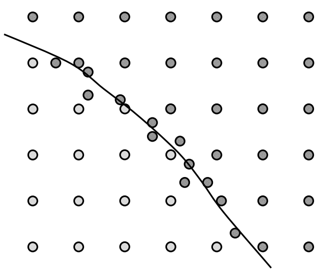
- Define an implicit function D(p) = distance to the surface at point p



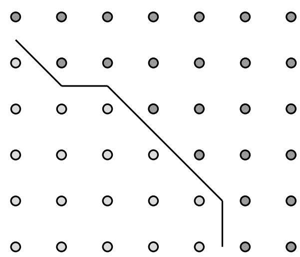
> 0 outside the surface
< 0 inside the surface
= 0 on the surface

Use interpolation to compute distance at an arbitrary point

Distance Field



Marching Cube



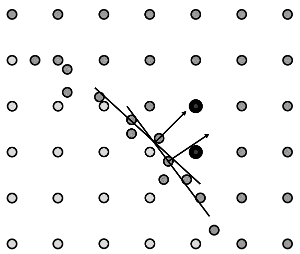
Tangent plane and signed distance estimation

- Compute n_i that minimizes

$$\sum_{p_j} (p_j - x_i) \cdot n_i$$

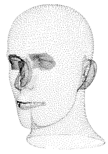
Where

$$p_j \in Nbhd(x_i)$$

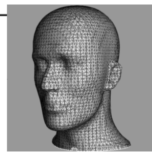


Assume that if points are close then normals are nearly parallel

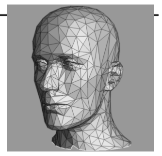
Results of Three Phases[Hoppe92]



Points

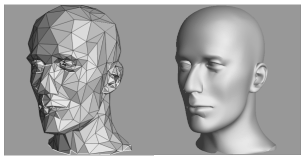


Phase 1



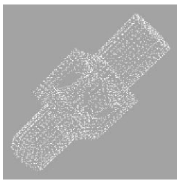
Phase 2

Phase 3

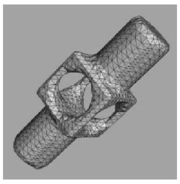


Phase 1: Initial Surface Estimation

- From points to mesh:
  - Inferring topological type
  - Creating geometric approximation

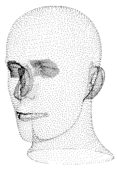


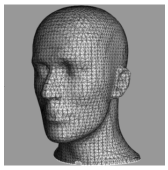
Points (4,102)



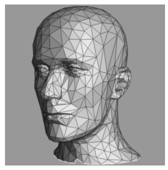
Initial Mesh (886 vertices)

Phase 2: Mesh optimization





Phase 1 (886 vertices)

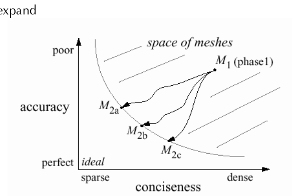


Phase2 (163 vertices)

- **Input**: data points *P*, initial mesh
- **Output:** optimized mesh *M*

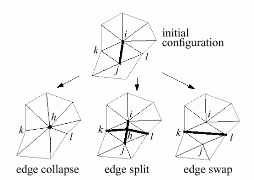
Phase 2: Mesh Optimization

- Optimize the energy function:
  Optimization rules use edge collapse and expand
- Sum of square distances
  - (accurate)
- Number of vertices
  - (sparse)
- Regulation term
  - make vertices with equidistance

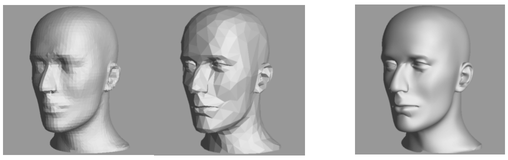


Optimization steps

```
Repeat
   (K',V')=GenerateLegalMove(K,V)
   V' = OptVertexPosition(K',V')
   if E(K',V') < E(K,V)
       (K,V) = (K', V')
   endif
Until converges
```



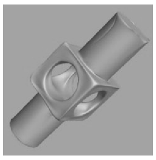
Phase 3: Piecewise Smooth Surface



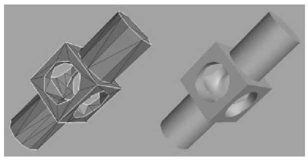
piecewise planar $\Rightarrow$ piecewise smooth surface

Piecewise smooth

- Not everywhere smooth, but piecewise smooth!



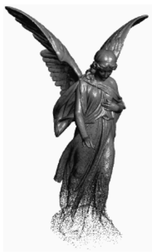
Smooth surface



Piecewise smooth surface

Point Set Surfaces[Alexa+01]

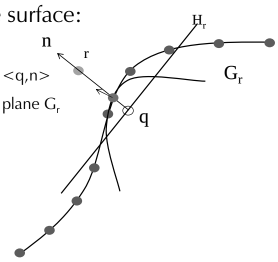
- Smooth surface manifold representation of point sets
- Up-sample
- Down-sample
- Noise reduction
- Interactive rendering



The moving least squares (MLS)[Levin]

Constructive definition of the surface:

- Input point r
- Compute a local reference plane H_r=<q,n>
- Compute a local polynomial over the plane G_r
- Project point $r'=G_r(0)$



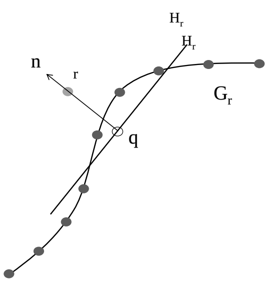
Local reference plane

- Plane equation: H_r=<q,n>
- $H_r = \min \Sigma_i (d(x_i, H_r)\theta(||q - x_i||)$
  Non-linear optimization

Where

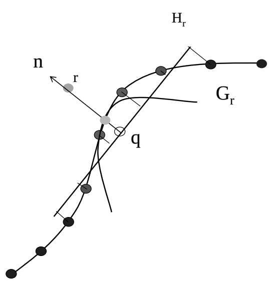
- $\{x_i\}$ neighbor points
- $\theta(x)=\exp(-x^2/h^2)$
- h reconstruction parameter, depends on the spacing of points

- Note: q != r



Projecting the point

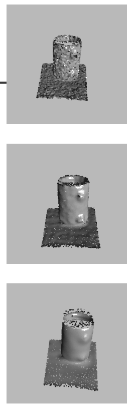
- $G_r = \min \Sigma_i |G(x_{i,x}) - x_{i,y}| \theta(||q - x_i||)$
  Linear optimization
- $r' = G_r(0)$



Noise Reduction

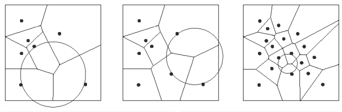
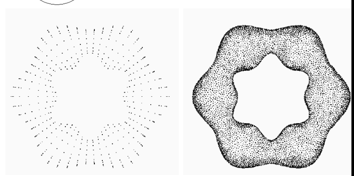
- Using high values of h
  - Smoother surfaces
  - Less detail





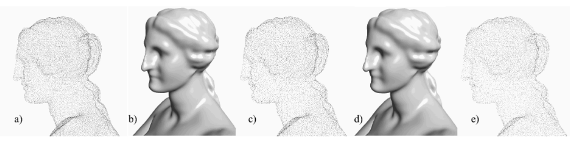
Up sample

- Create a local Voronoi diagram
- Add points at Voronoi vertex with the largest radius
- Stop when spacing is small enough

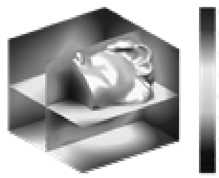
Downsample

- S_p- Surface
- Remove Point p_i if $|S_p - S_{P-pi}| < \varepsilon$
- Speedup: Evaluate $|p_i - \text{Project}(p_i, S_{P-pi})|$



Reconstruction by Radial Bases Function

- Scattered data interpolation scheme
- Input: set of points on the surface
- Output: Implicit function that interpolates the points in a nice way
- Nice way:



$$E(f) = \int_{s\in\Omega} f_{xx}^2(s) + 2f_{xy}^2(s) + f_{yy}^2(s)\,ds$$

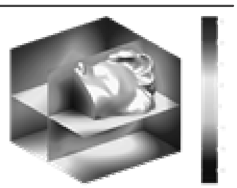
$$E(f) = \int_{s\in\Omega} f_{xx}^2(s) + f_{yy}^2(s) + f_{zz}^2(s) + 2f_{xy}^2(s)ds + 2f_{xz}^2(s)ds + 2f_{yz}^2(s)ds$$

Form of solution

- The solution is of the form:

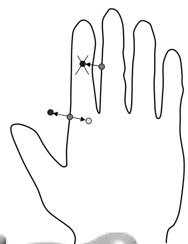
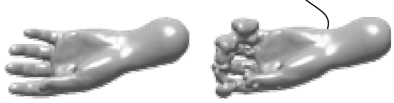
$$f(x) = \sum_{i=1}^{n} w_i \Phi(\|x - x_i\|)$$

- Where P is a polynomial
- Φ is a radially symmetric function
- The trivial solution is $w_i=0$

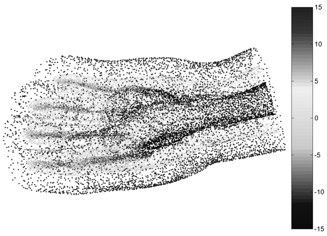


Forming a signed-distance function Off surface points

- For every point, add two *off-surface* points, one inside and one outside the surface in the direction of the normal
- Add a point only if it is closest to its source
- $N \approx 3n$ points

Forming a signed-distance function



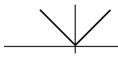
Outward normal points

On-surface points

Inward normal points

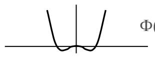
- Off-surface points are projected along surface normals

Radial basis functions



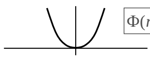
$\Phi(r) = |r|$

Minimizes 2nd derivative in 3D



$\Phi(r) = r^2 \log r$

Minimizes 2nd derivative in 2D



$\Phi(r) = r^3$

Minimizes 3rd derivative in 3D

Computing the weights

- Input : $\{x_i\}$, $\{f_i\}$. Compute $\{w_i\}$ $\quad f(x) = \sum_{i=1}^{n} w_i \Phi(\|x - x_i\|)$

Unknowns to compute

$$\left(A_{NxN}\right)\left(W\right) = \left(f\right)$$

← Function values

Matrix dependent on the locations of the data points

Computing - example

$$\begin{pmatrix} \Phi(\|x_1 - x_1\|) & \Phi(\|x_2 - x_1\|) & \cdots & \Phi(\|x_N - x_1\|) \\ \Phi(\|x_1 - x_2\|) & \Phi(\|x_2 - x_2\|) & & \Phi(\|x_N - x_2\|) \\ \vdots & & & \vdots \\ \Phi(\|x_1 - x_N\|) & \Phi(\|x_2 - x_N\|) & \cdots & \Phi(\|x_N - x_N\|) \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}$$

- Symmetric positive matrix

Complexity

Straight-forward method:

- Storage $O(N^2)$
- Solving the W_i $O(N^3)$
- Evaluating f(x) O(N)

Fast method [Carr01]

- Storage O(N)
- Solving the W_i O(NlogN)
- Evaluating f(x) O(1) + O(NlogN) setup

Carr 01

Simplification (center reduction)

- Reduce the number of centers (points)
- Greedy algorithm, reduce points as long as the surface is close enough

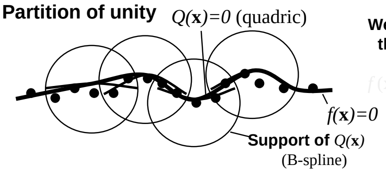
Carr 01

Why use implicit functions

- Well-defined interior and exterior
- Topology changes easy
- Constructive Solid Geometry
- Shape Interpolation

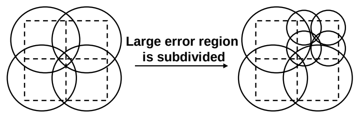
Multi-level Partition of Unity

Partition of unity

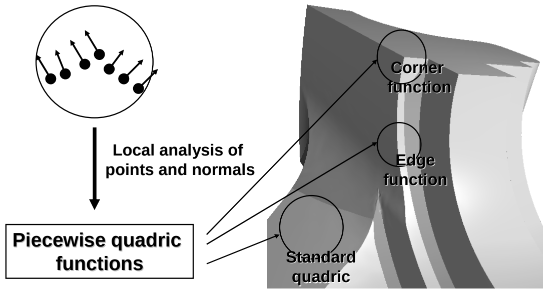


Weighted average of the local functions

$$f(\mathbf{x}) = \frac{\sum w_i(\mathbf{x}) Q_i(\mathbf{x})}{\sum w_i(\mathbf{x})}$$



Sharp Features



Ray-traced $f=0$