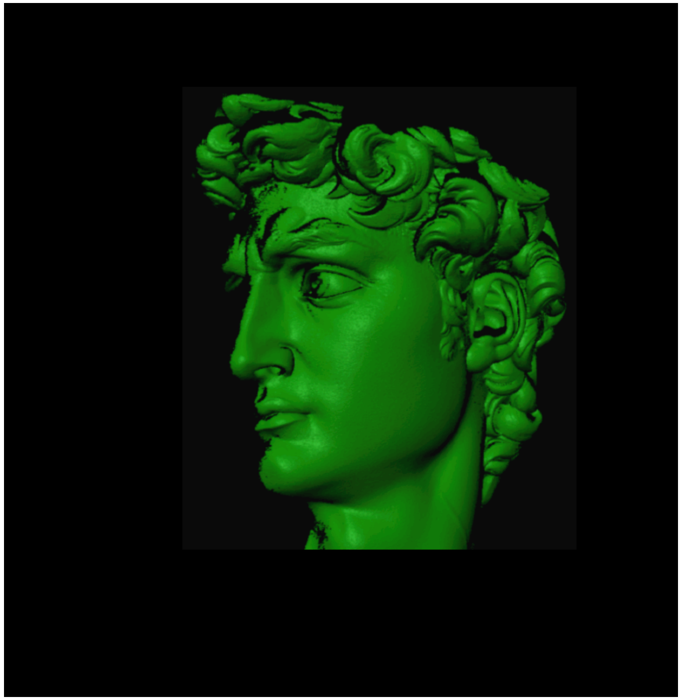
Range Processing Pipeline

Sai-Kit Yeung

ISTD, SUTD

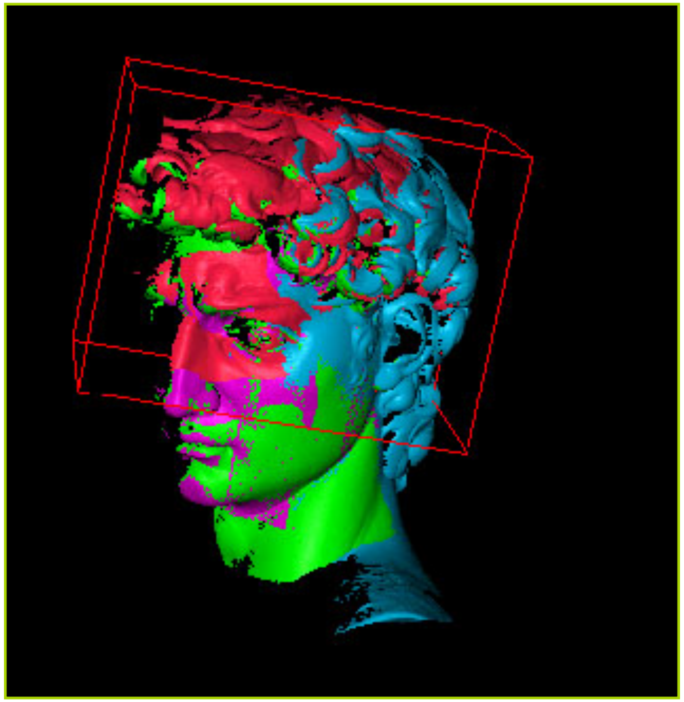
Notes courtesy by Wojciech Matusik, Some slides from Szymon Rusinkiewicz, Brian Curless, Srinivasa Narasimhan

Range Processing Pipeline



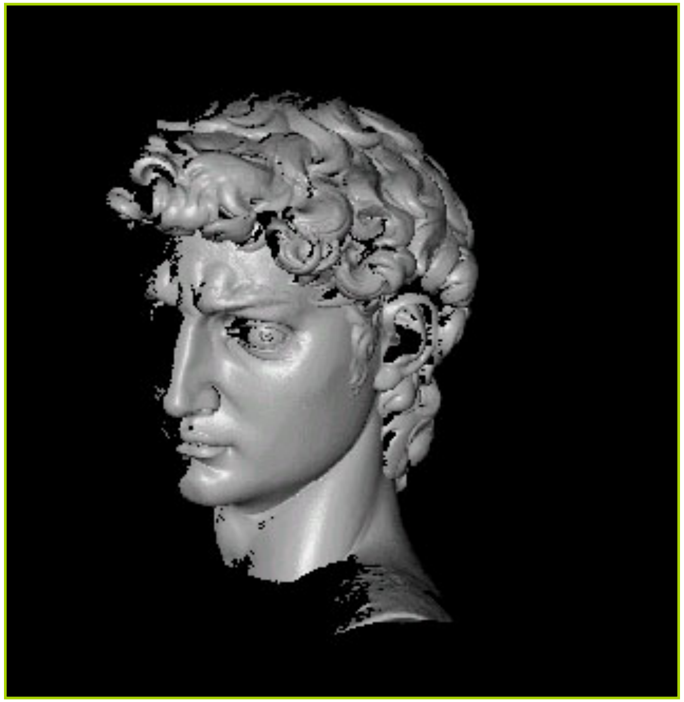
- Overview
  - Range image acquisition
    - View planning
  - Scan registration
    - Pairwise registration
    - Global registration
  - Surface reconstruction
    - Merging all scans into one surface

Range Processing Pipeline



- Overview
  - Range image acquisition
    - View planning
  - Scan registration
    - Pairwise registration
    - Global registration
  - Surface reconstruction
    - Merging all scans into one surface

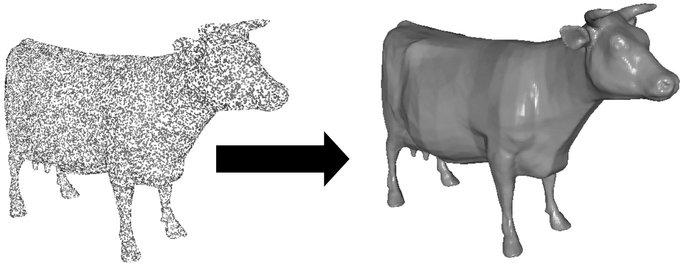
Range Processing Pipeline



- Overview
  - Range image acquisition
    - View planning
  - Scan registration
    - Pairwise registration
    - Global registration
  - Surface reconstruction
    - Merging all scans into one surface

Range Processing Pipeline

- Overview
  - Range image acquisition
    - View planning
  - Scan registration
    - Pairwise registration
    - Global registration
  - Surface reconstruction
    - Merging all scans into one surface

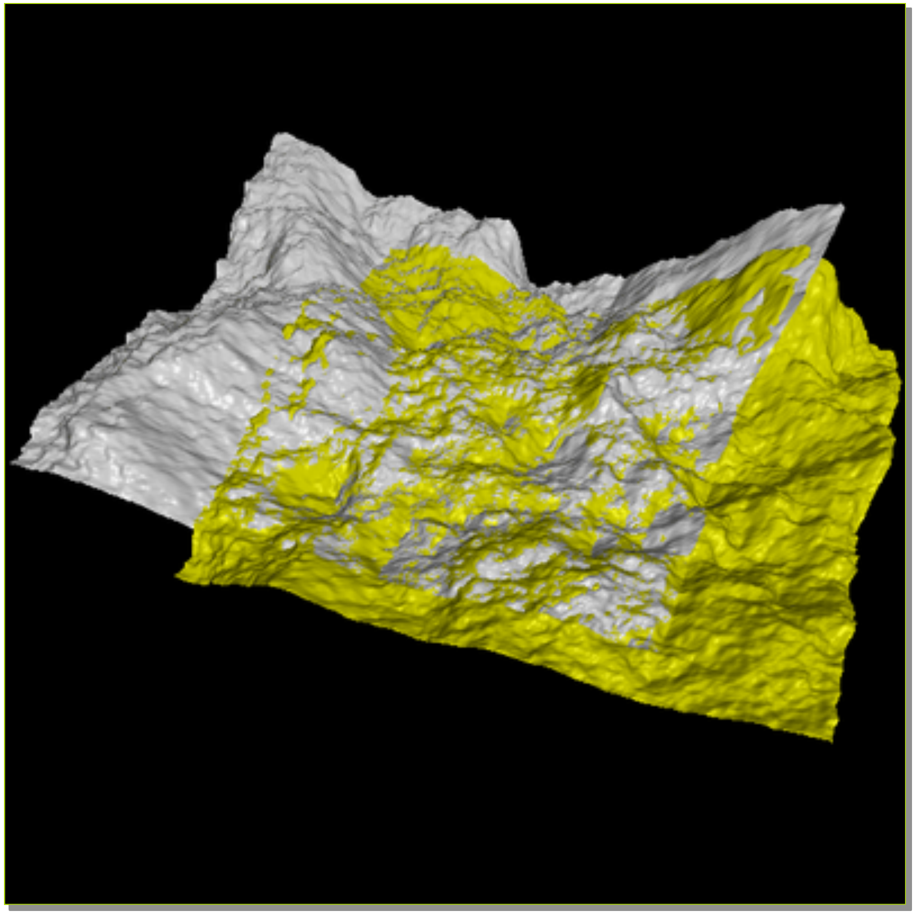


The Plan

- Scan Registration
  - Pairwise Rigid Registration
  - Global Registration
- Surface Reconstruction

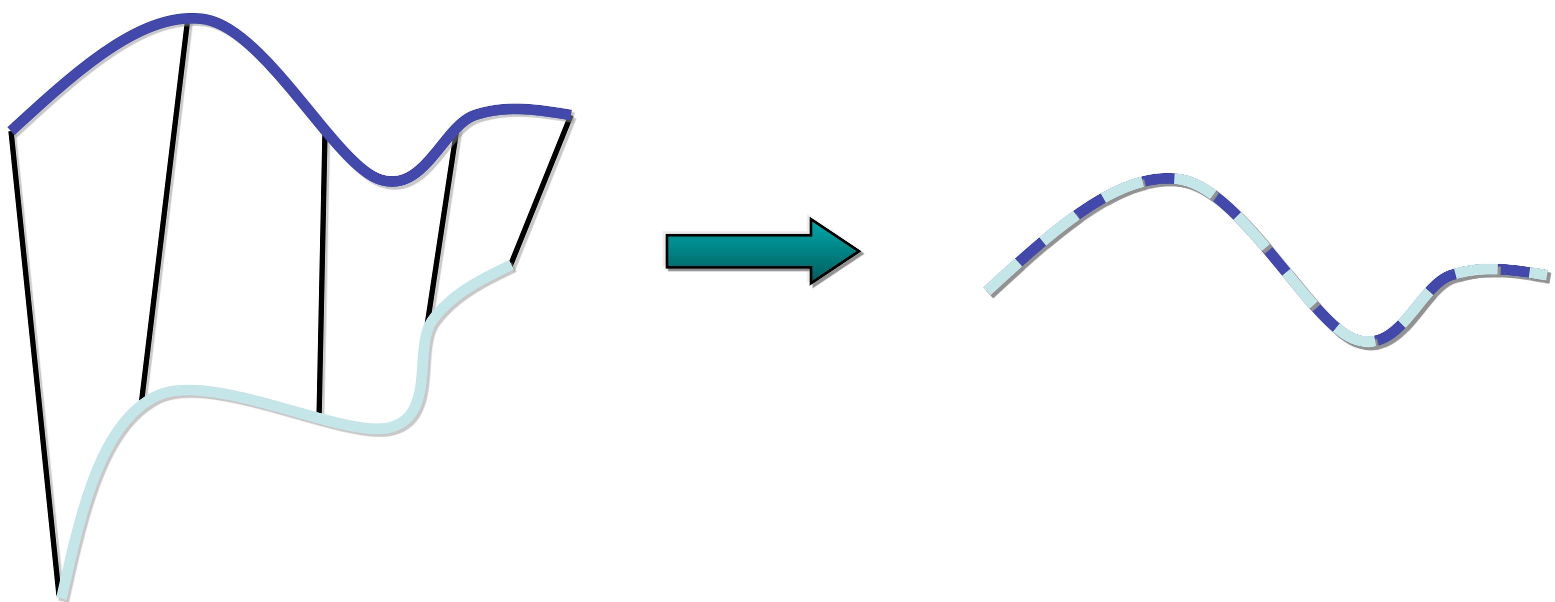
Pairwise Rigid Registration Goal

- Align two partially-overlapping meshes given initial guess for relative transform



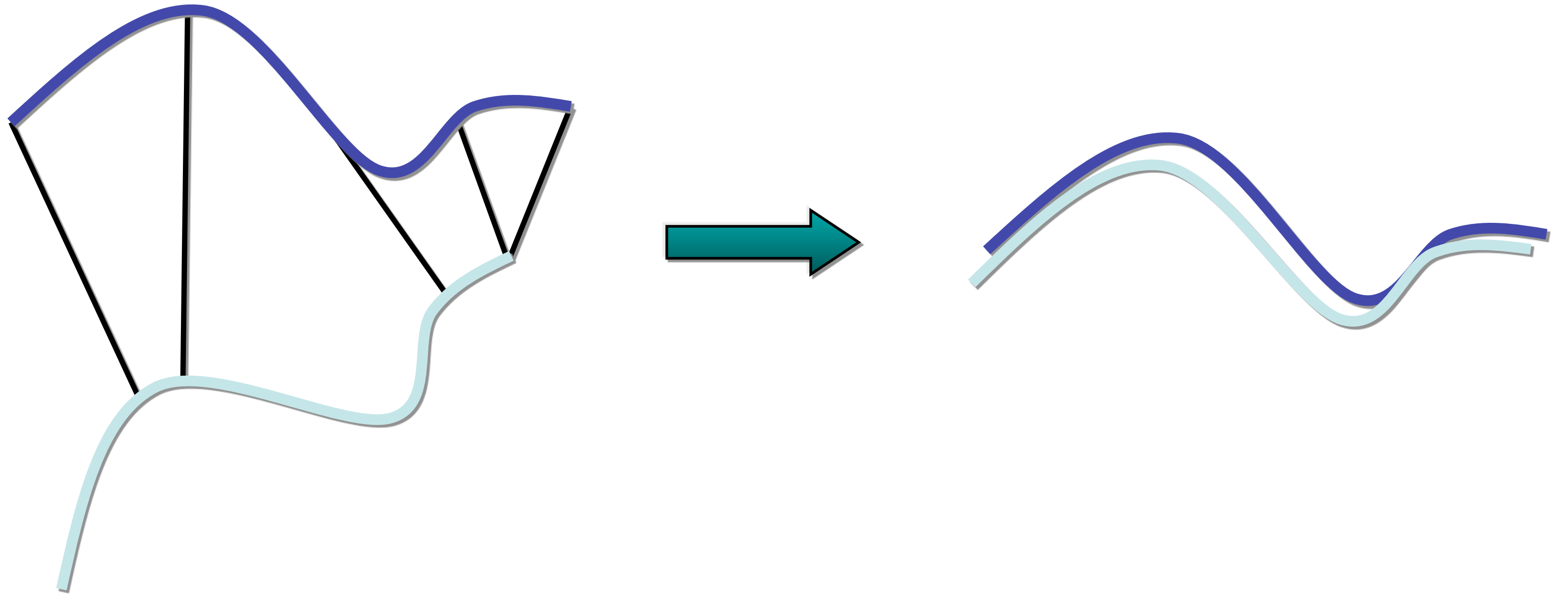
Aligning 3D Data

- If correct correspondences are known, can find correct relative rotation/translation (that minimizes pairwise distance)



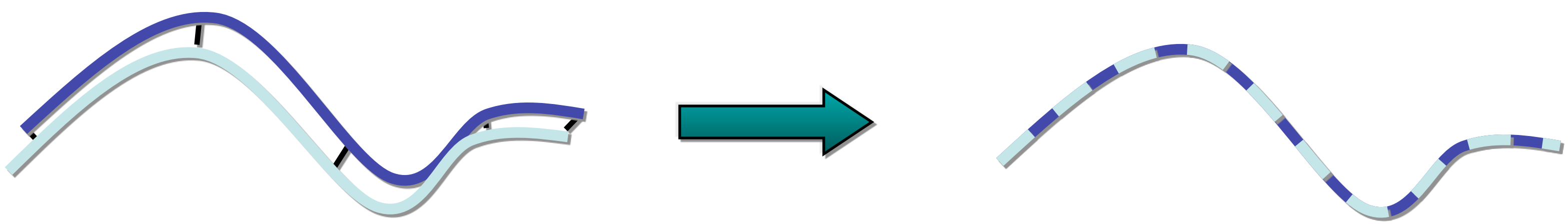
Aligning 3D Data

- How to find correspondences: User input? Feature detection? Signatures?
- Alternative: assume closest points correspond



Aligning 3D Data

- … and iterate to find alignment
  - Iterative Closest Points (ICP) [Besl & McKay 92]
- Converges if starting position “close enough“



Basic ICP

- Select e.g. 1000 random points
- Match each to closest point on other scan, using data structure such as *k*-d tree
- Reject pairs with distance > k times median
- Construct error function:

$$E = \sum \left| Rp_i + t - q_i \right|^2$$

- Solve for the pose (R, t)
  - Minimize (closed form solution in [Horn 87])

ICP Variants

- Variants on the following stages of ICP have been proposed:

1. Selecting source points (from one or both meshes)
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs
5. Assigning an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation

Performance of Variants

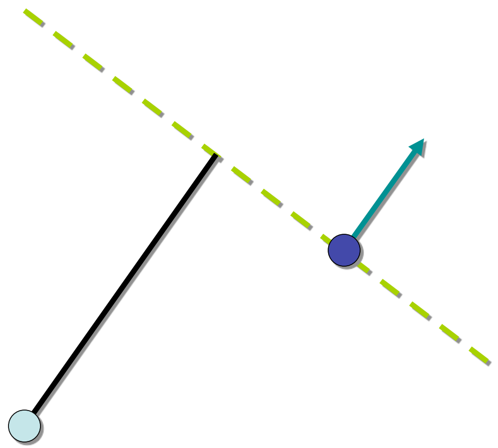
- Can analyze various aspects of performance:
  - Speed
  - Stability
  - Tolerance of noise and/or outliers
  - Maximum initial misalignment
- Comparisons of many variants in
  [Rusinkiewicz & Levoy, 3DIM 2001]

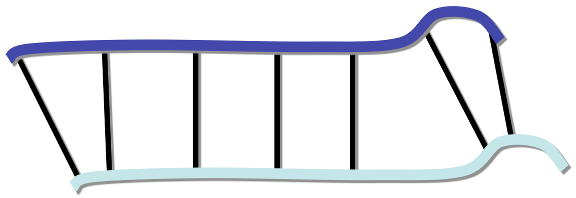
ICP Variants

1. Selecting source points (from one or both meshes)
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs
5. Assigning an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation

Point-to-Plane Error Metric

- Using point-to-plane distance instead of point-to-point lets flat regions slide along each other [Chen & Medioni 91]





point-to-point prevent the scans from "sliding"

ICP Variants

1. Selecting source points (from one or both meshes)
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs
5. Assigning an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation

Closest Compatible Point

- Closest point often a bad approximation to corresponding point
- Can improve matching effectiveness by restricting match to compatible points
  - Compatibility of colors [Godin et al. 94]
  - Compatibility of normals [Pulli 99]
  - Other possibilities: curvatures, higher-order derivatives, and other local features

Need to think of their weightings

ICP Variants

1. Selecting source points (from one or both meshes)
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs
5. Assigning an error metric to the current transform
6. Minimizing the error metric w.r.t. transformation

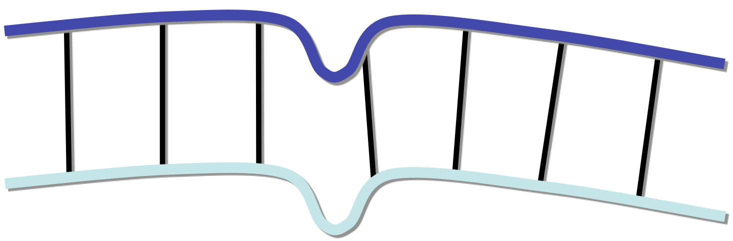
Selecting Source Points

- Use all points
- Uniform subsampling
- Random sampling
- Stable sampling [Gelfand et al. 2003]

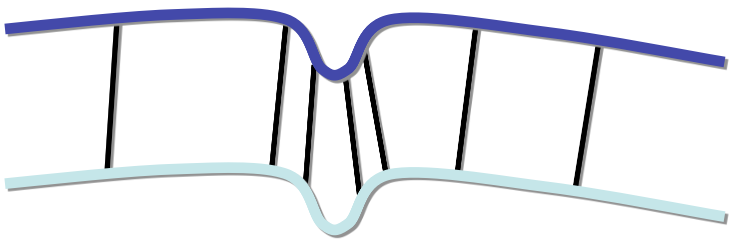
Selecting Source Points

- Use all points
- Uniform subsampling
- Random sampling
- Stable sampling [Gelfand et al. 2003]
  - Select samples that constrain all degrees of freedom of the rigid-body transformation

Stable Sampling



Uniform Sampling



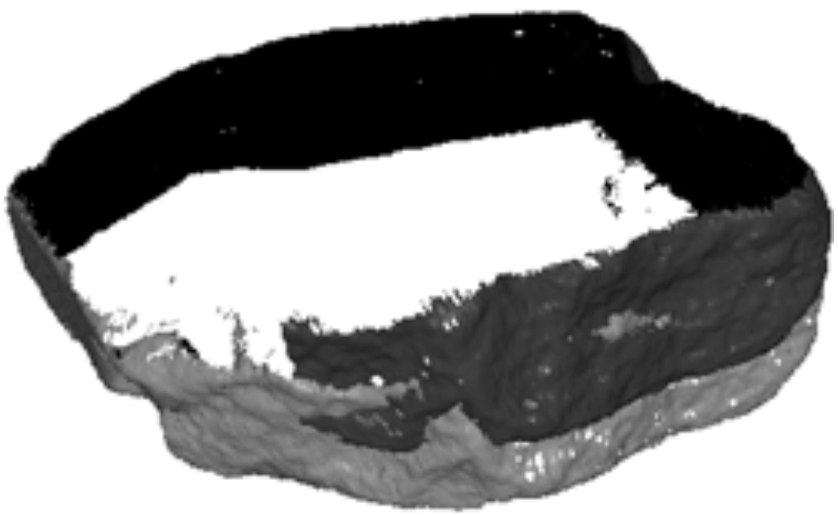
Stable Sampling

The Plan

- Scan Registration
  - Pairwise Rigid Registration
  - Global Registration
- Surface Reconstruction

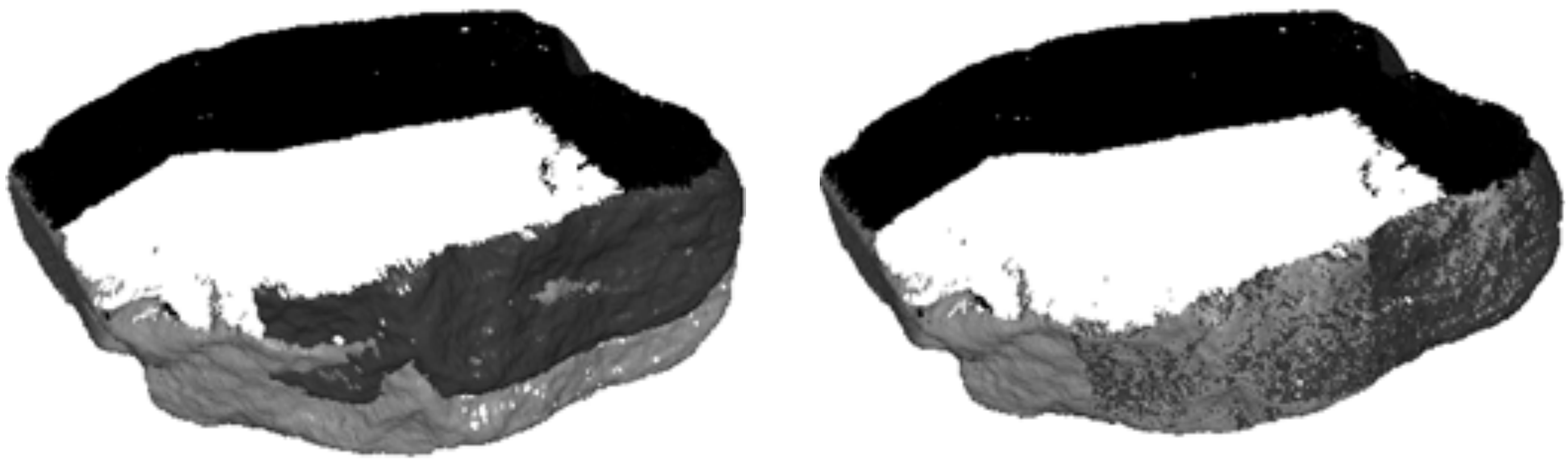
Global Registration Goal

- Given: *n* scans around an object
- Goal: align them all
- First attempt: ICP each scan to one other



Global Registration Goal

- Want method for distributing accumulated error among all scans



Approach #1: Avoid the Problem

- In some cases have 1 scan that covers large part of surface (e.g., cylindrical scan)
- Align all other scans to this “anchor”
- Disadvantage: not always practical to obtain anchor scan

Approach #2: The Greedy Solution

- Align each new scan to the union of all previous scans [Masuda 96]
- Disadvantages:
  - Order dependent
  - Doesn't spread out error

Approach #3: The Brute-Force Solution

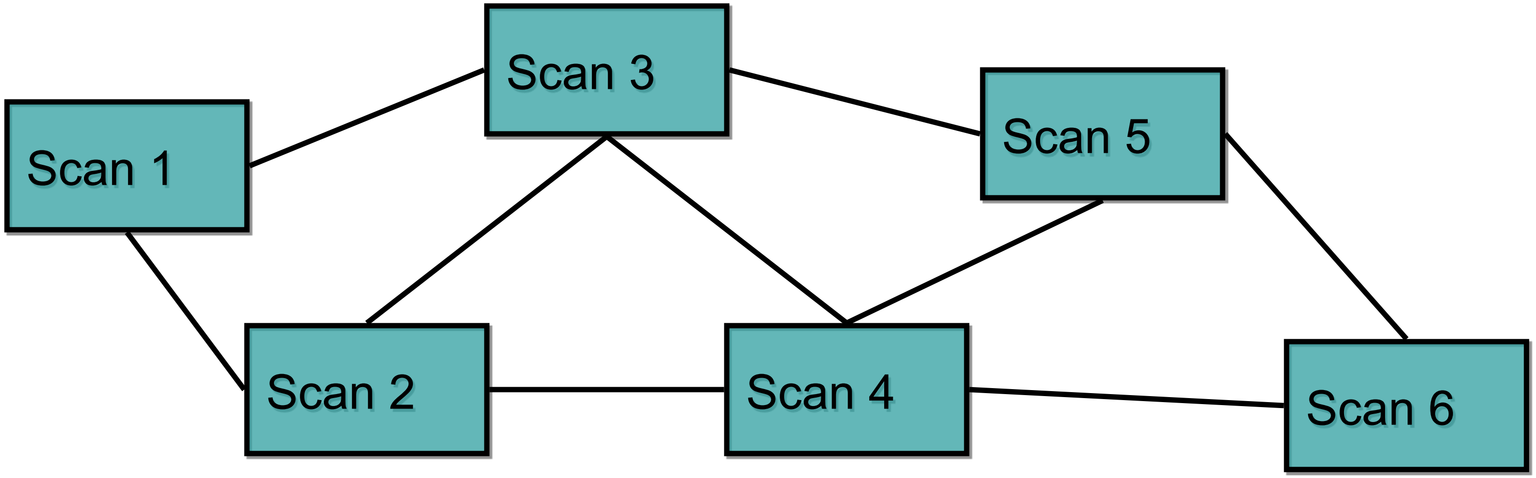
- While not converged:
  - For each scan:
    - For each point:
      - For every other scan
        - Find closest point
  - Minimize error w.r.t. transforms of all scans
- Disadvantage:
  - Solve $(6n)\times(6n)$ matrix equation, where n is number of scans

Approach #3a: Slightly Less Brute-Force

- While not converged:
  - For each scan:
    - For each point:
      - For every other scan
        - » Find closest point
    - Minimize error w.r.t. transform of this scan
- Faster than previous method (matrices are 6×6) [Bergevin 96, Benjemaa 97]

Graph Methods

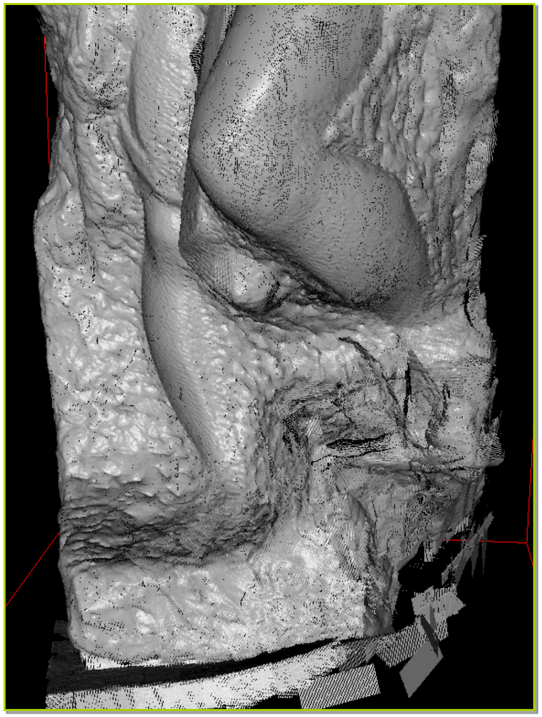
- Many global registration algorithms create a graph of pairwise alignments between scans
  - Compute pairwise alignments for all graph edges
  - Solve for a set of global transformations as consistent as possible with all pairwise alignments



Bad ICP in Globalreg

- One bad ICP can throw off the entire model!

Correct Globalreg



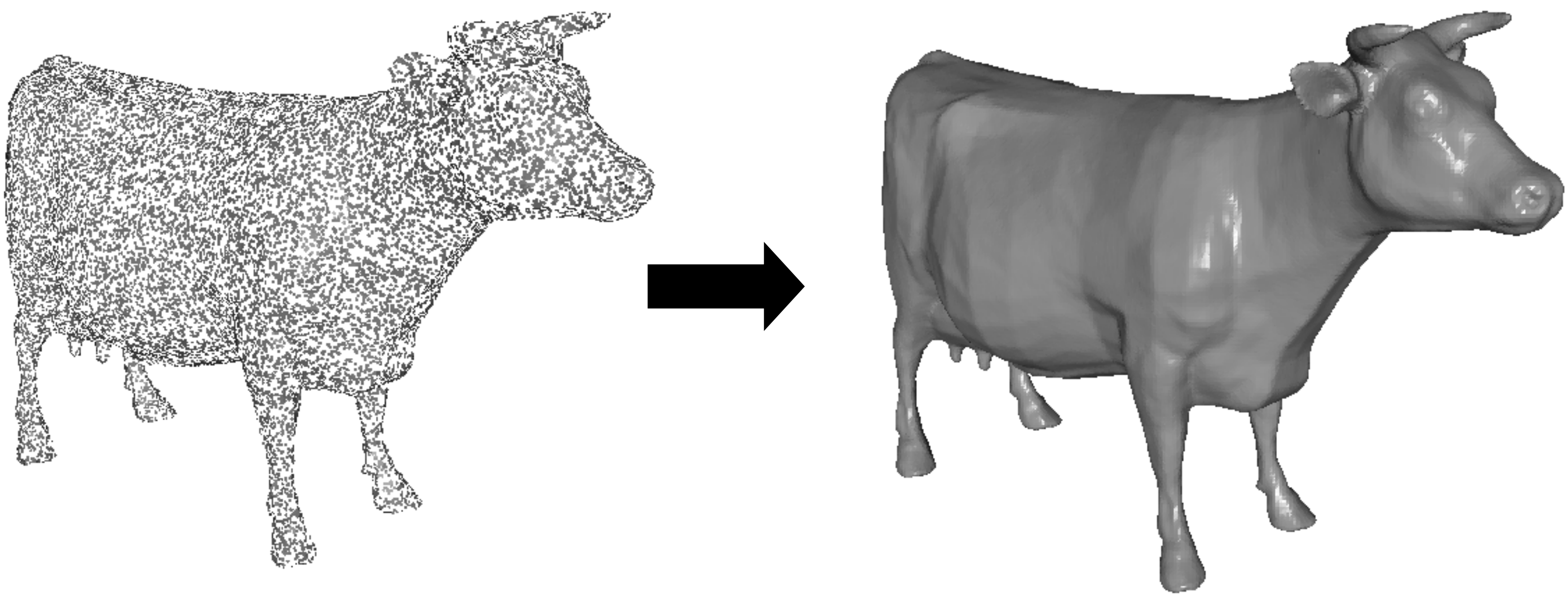
Globalreg Including Bad ICP

The Plan

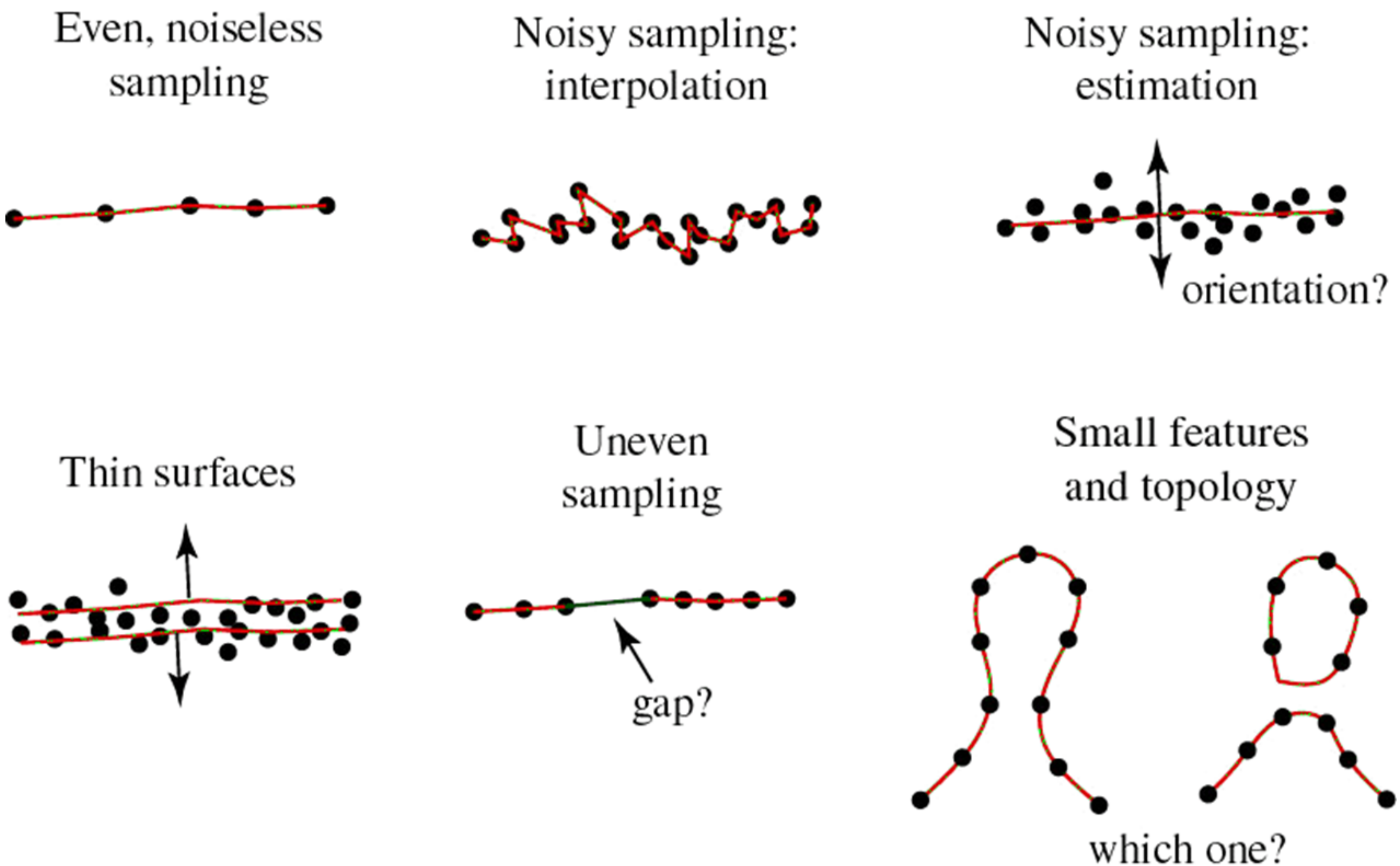
- Scan Registration
  - Pairwise Rigid Registration
  - Global Registration
- Surface Reconstruction

Surface Reconstruction

- Generate a mesh from a set of surface samples

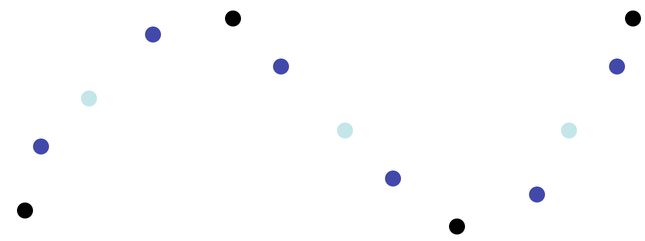


Challenges for Surface Reconstruction



Computational Geometry Approaches

- Computational geometry approaches figure out how to connect up “nearby” points
- Need sufficiently dense sampling, little noise

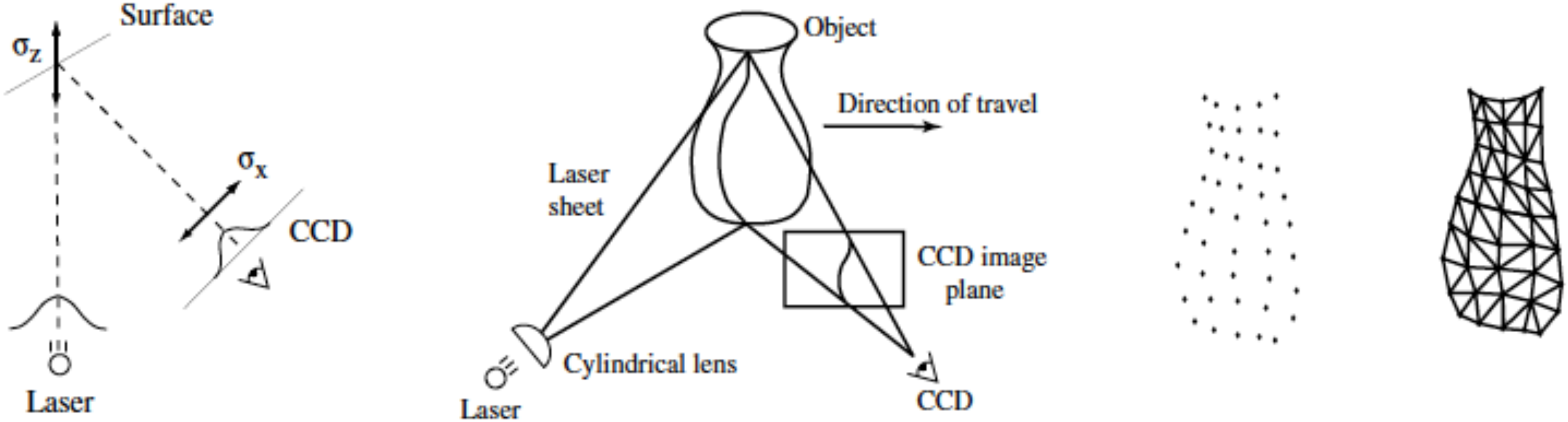


- Delaunay triangulation: connect nearest points

Surface Reconstruction From Range Images

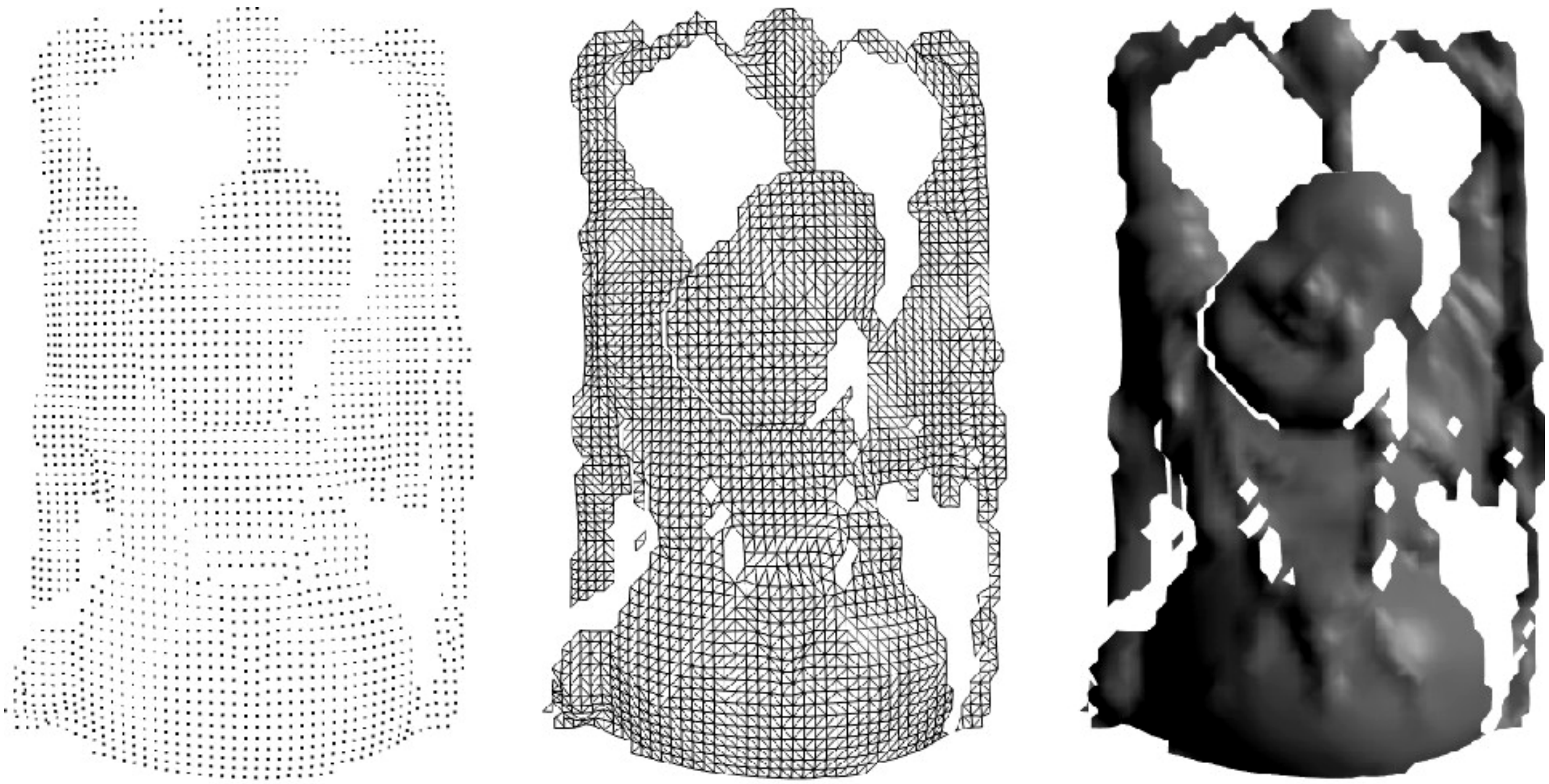
- Often an easier problem than reconstruction from arbitrary point clouds
  - Implicit information about adjacency, connectivity
  - Roughly uniform spacing
- Algorithm
  - Construct surface from each range image
  - Merge resulting surfaces
    - Obtain average surface in overlapping regions
    - Control point density

Surface Reconstruction From Range Images



Step 1: From Range Images to Range Surfaces

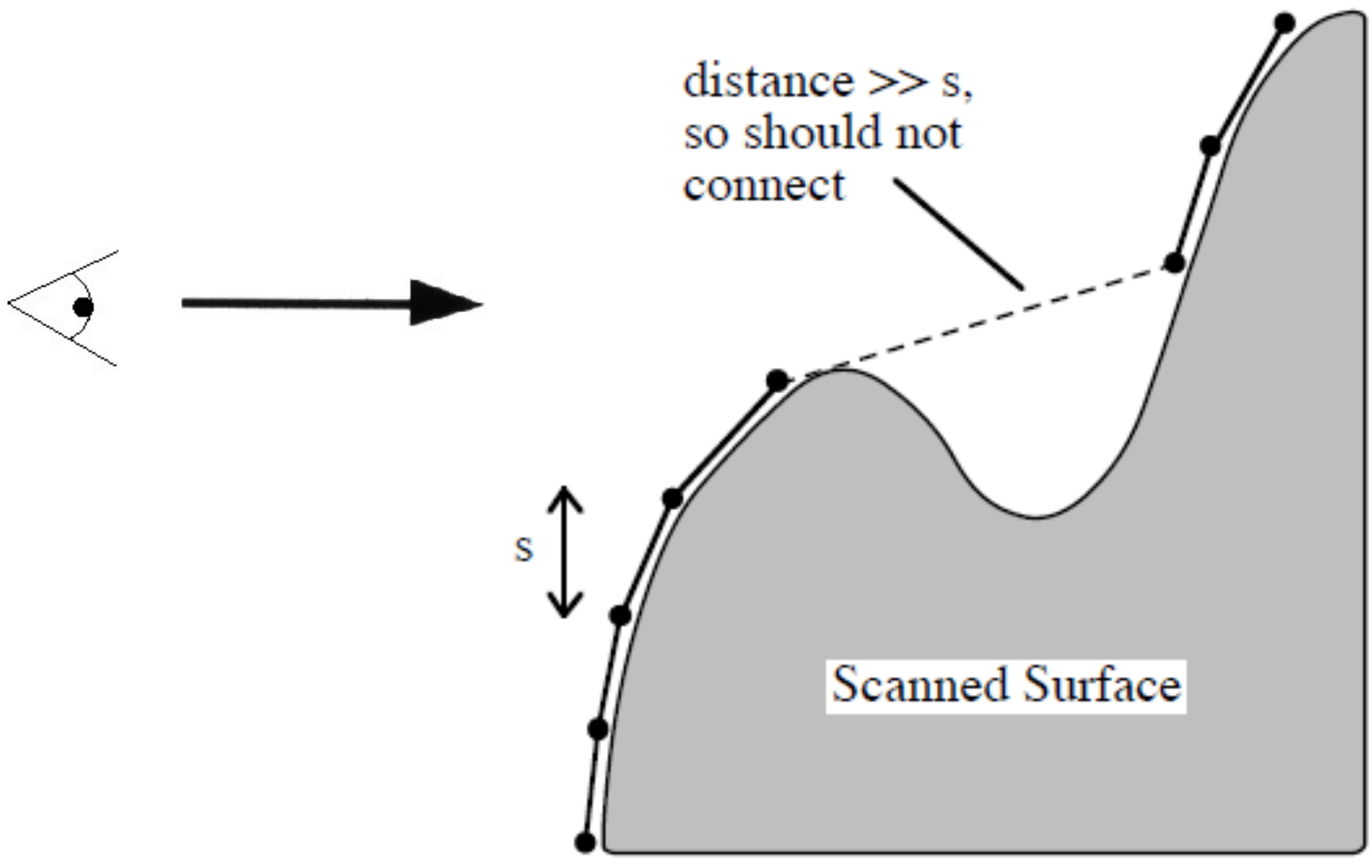
- Given a range image, connect up the neighbors



Range image Tesellation Range surface

Range Image Tessellation

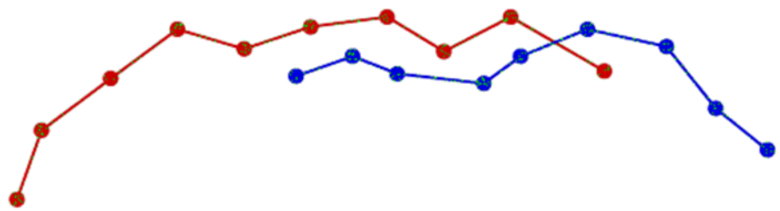
- To avoid "prematurely aggressive" reconstruction, a tessellation threshold should be employed



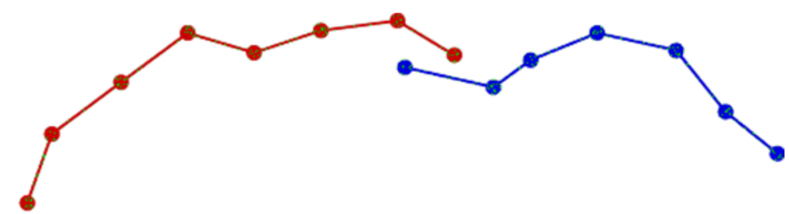
Step 2: Scan Merging

- Zippering, Turk & Levoy, 1994
- Erode geometry in overlapping areas
- Stitch scans together along seam
- Re-introduce all data
  - Weighted average

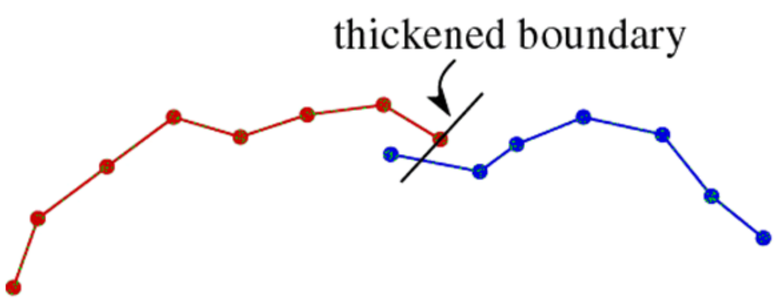
Zippering



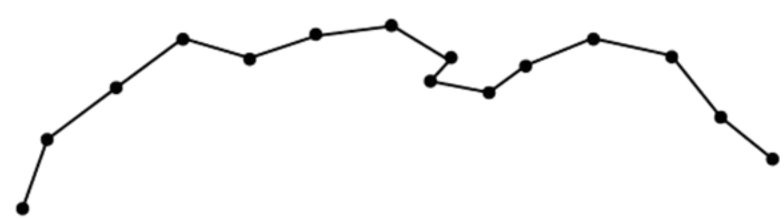
Overlapping range surfaces



Redundant geometry removed



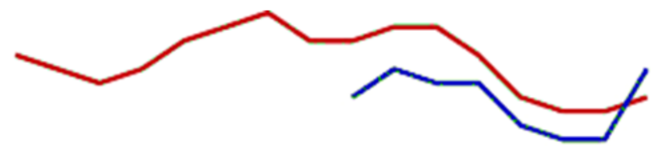
Mesh boundary "thickened" for clipping



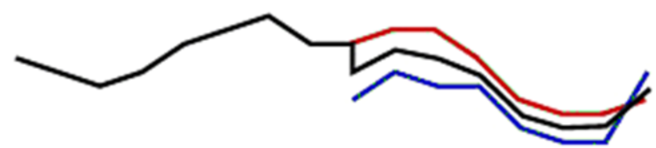
Zippered surface

Point Weighting

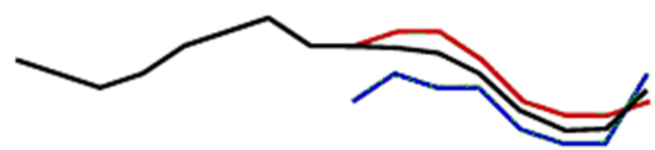
- Lower weights (tapering to 0) near boundaries
  - Smooth blends between views



Two range surfaces

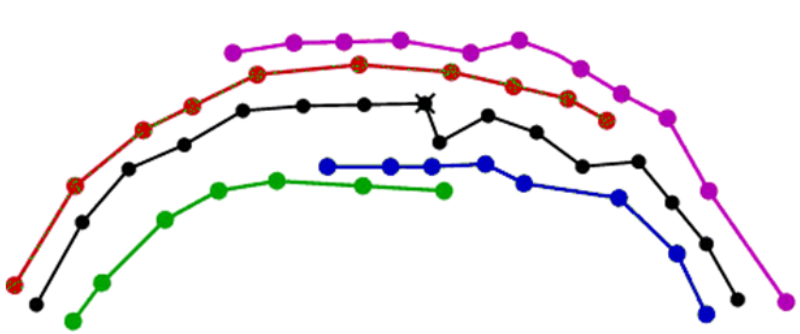


After unweighted blending

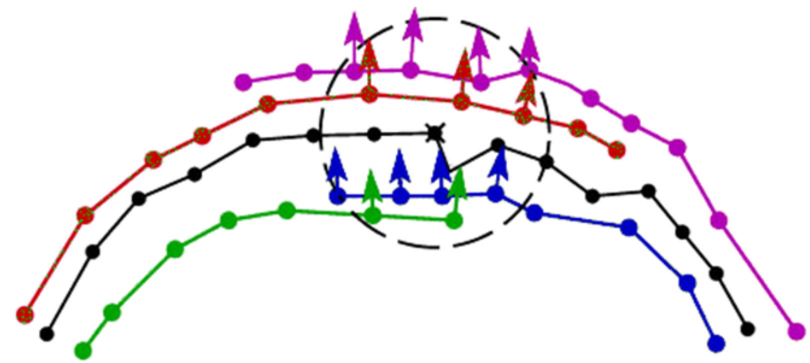


After weighted blending

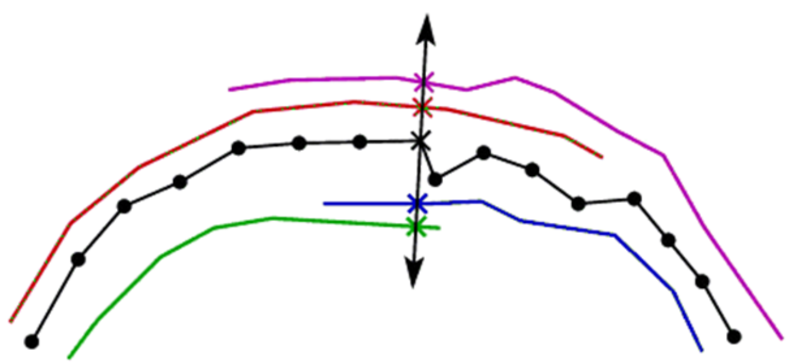
Consensus Geometry



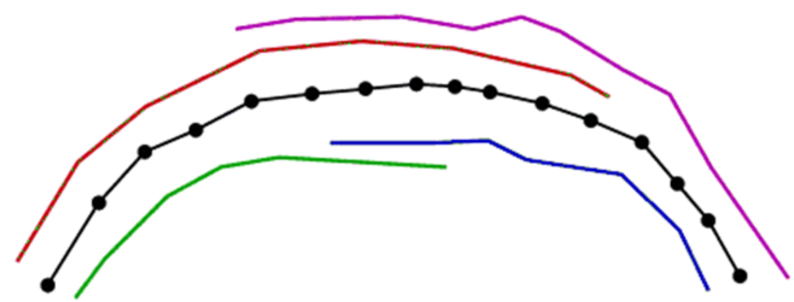
Zippered geometry + range surfaces



Compute consensus normal

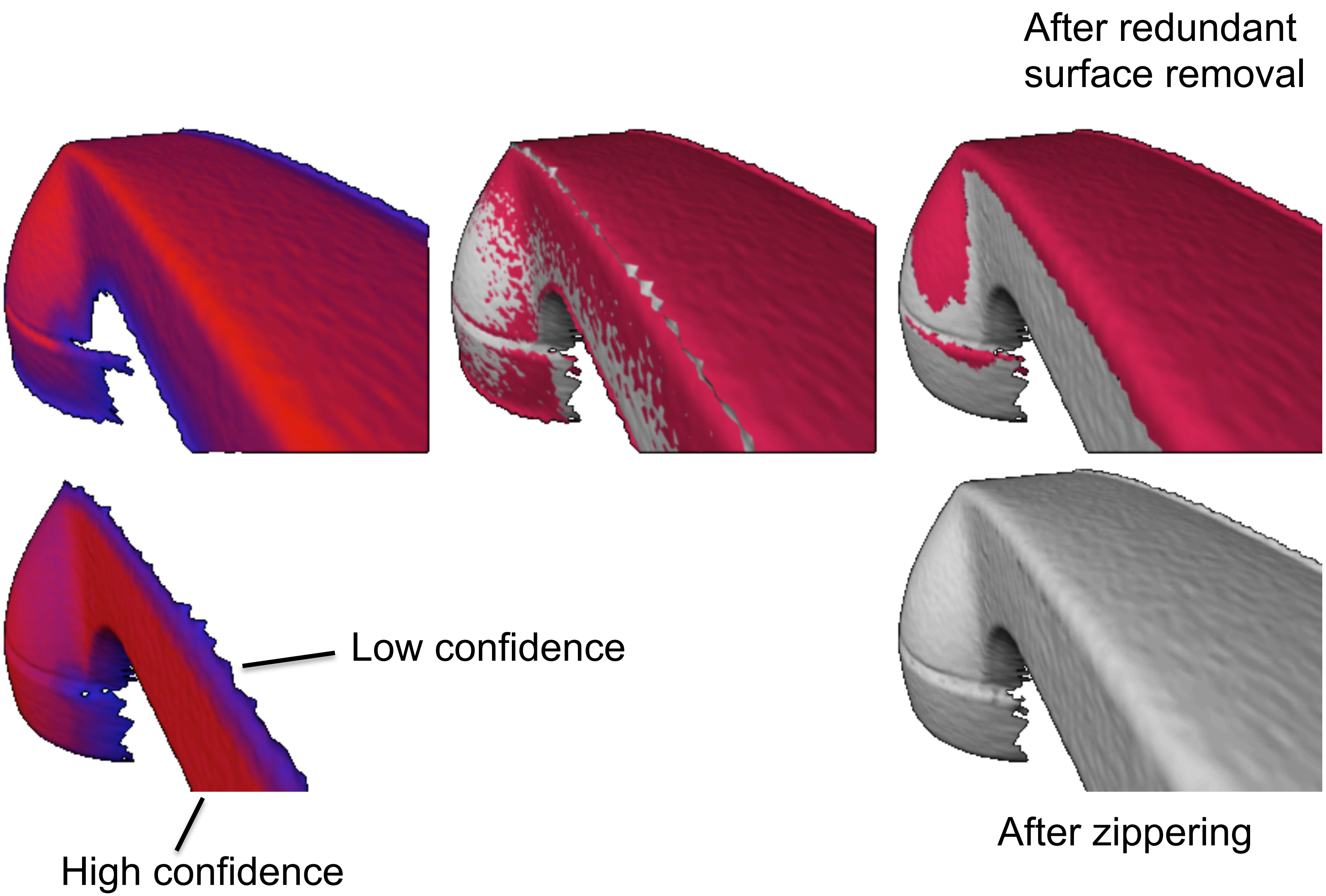


Find vertex positions on range surfaces
by intersection with consensus normal

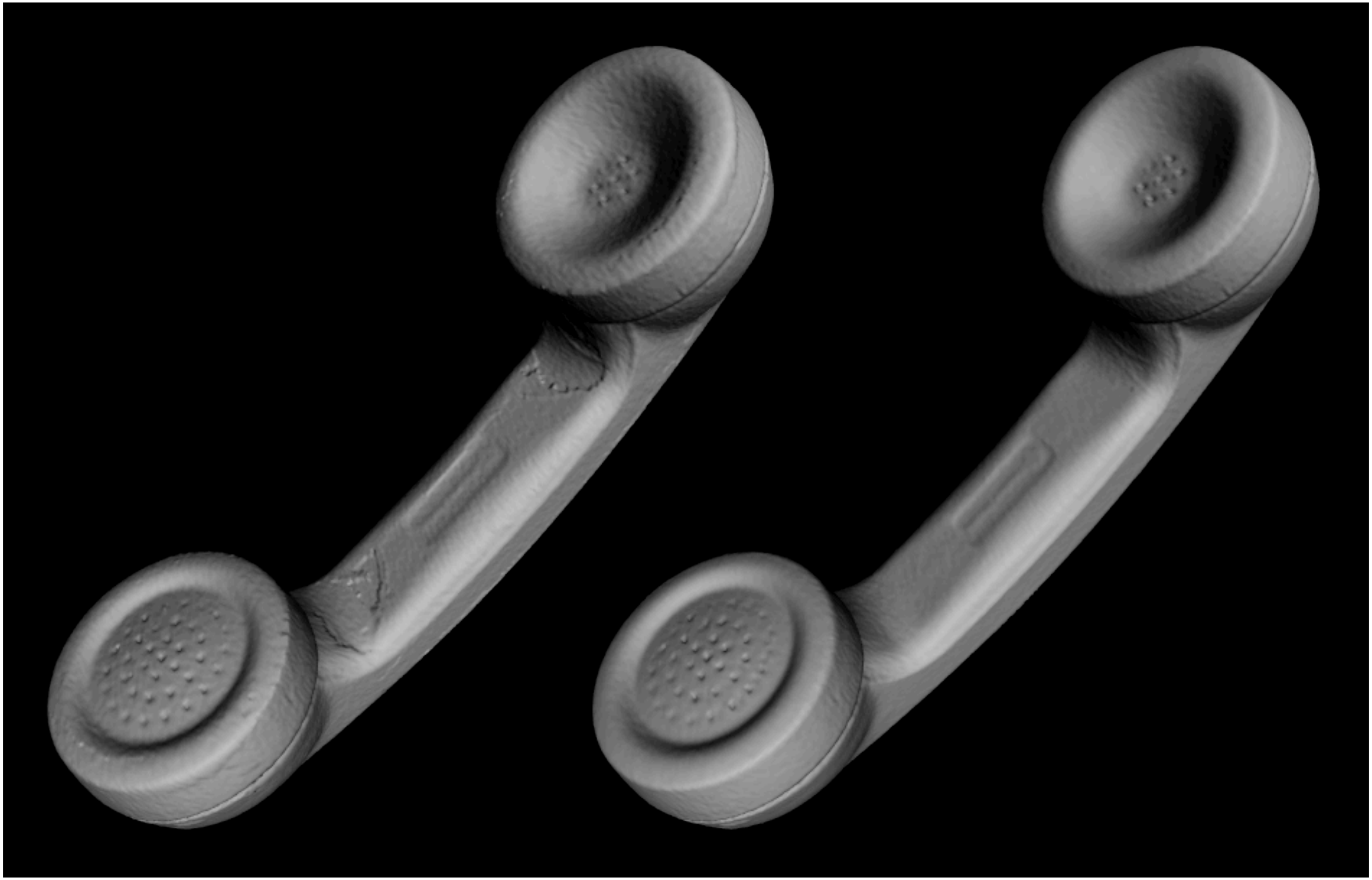


Compute weighted average of vertex positions

Zippering Example



Zippering Example

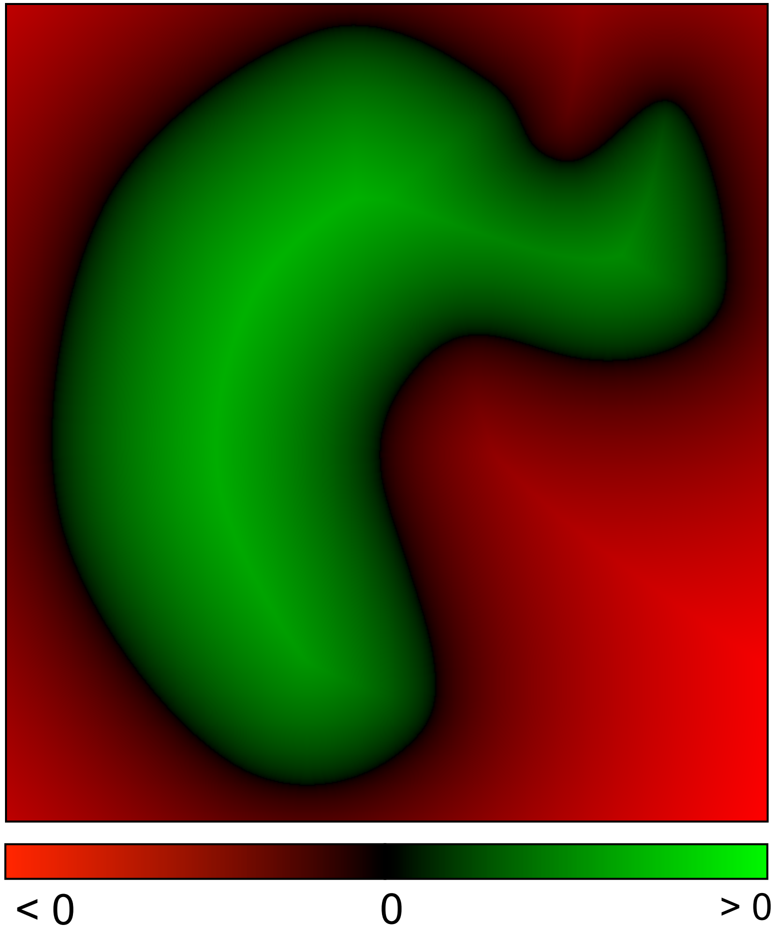


Without Consensus Geometry

With Consensus Geometry

Implicit Function Approaches

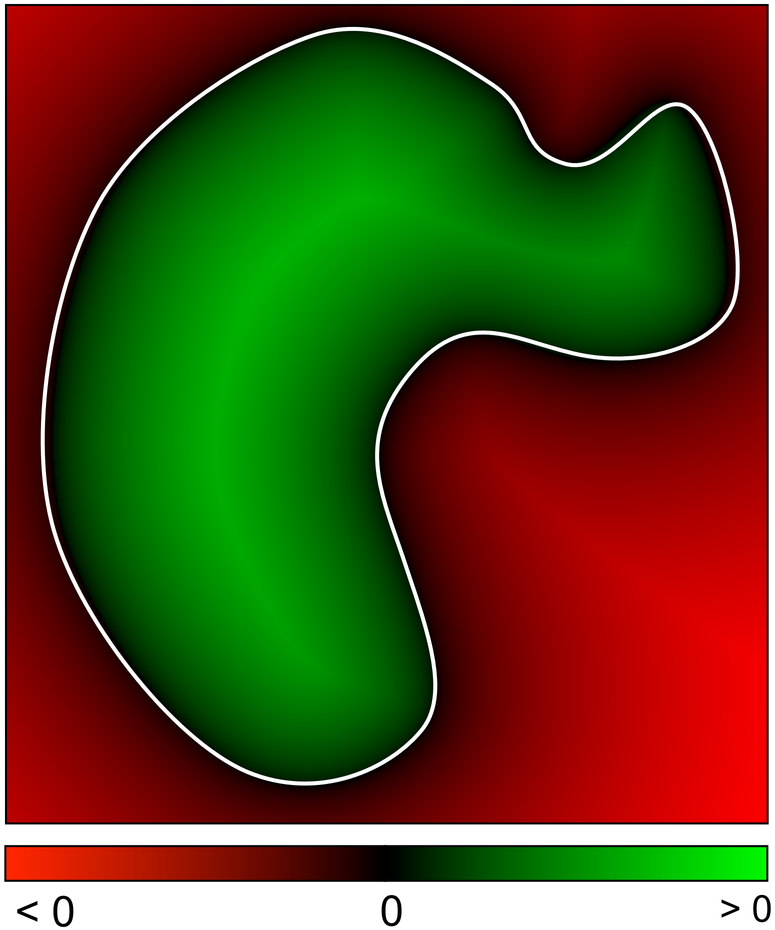
- Define a function with value less than zero outside the model and greater than zero inside



Implicit Function Approaches

- Define a function with value less than zero outside the model and greater than zero inside

- Extract the zero-set

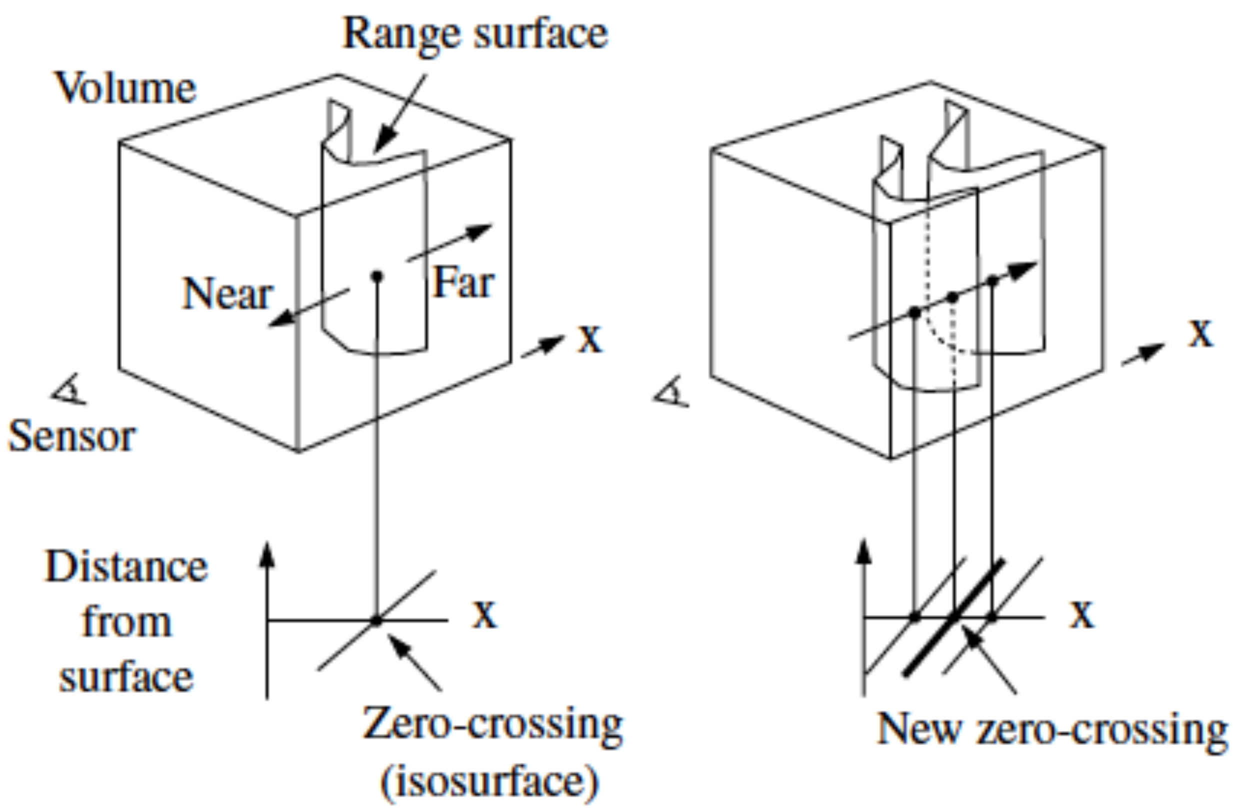


Volumetric Range Image Processing (VRIP)

- Curless & Levoy, 1996
- Algorithm:
  - Generate signed distance function for each scan
  - Combine/Compute average (possibly weighted)
  - Extract isosurface (using marching cubes)

Volumetric Range Image Processing (VRIP)

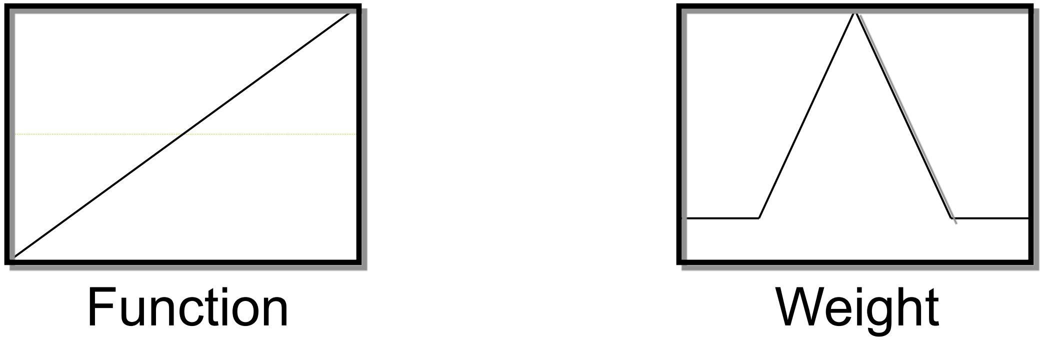
- Curless & Levoy, 1996



1. Sensors detect “zero-crossing”
2. Computer signed distance function
3. Weighted average
4. Extract isosurface

Volumetric Range Image Processing (VRIP)

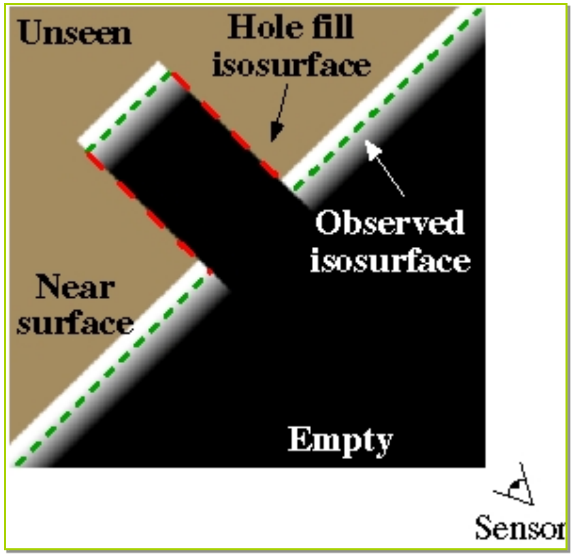
- Defined on fixed volumetric grid
- Implicit functions = ramps along line of sight to scanner
- Weighting along ramps



- Weighting across surface (similar to Zipper)

Space Carving in VRIP

- Space Carving - Mark all space between surface and scanner as "empty" (with low weight)
- Extract additional isosurfaces between "empty" and "unseen" (hole fill isosurface)



Microsoft Kinect Fusion (ICP and VRIP)

SIGGRAPH Talks 2011

KinectFusion:

Real-Time Dynamic 3D Surface Reconstruction and Interaction

Shahram Izadi 1, Richard Newcombe 2, David Kim 1,3, Otmar Hilliges 1,
David Molyneaux 1,4, Pushmeet Kohli 1, Jamie Shotton 1,
Steve Hodges 1, Dustin Freeman 5, Andrew Davison 2, Andrew Fitzgibbon 1

1 Microsoft Research Cambridge 2 Imperial College London
3 Newcastle University 4 Lancaster University
5 University of Toronto

Poisson Surface Reconstruction

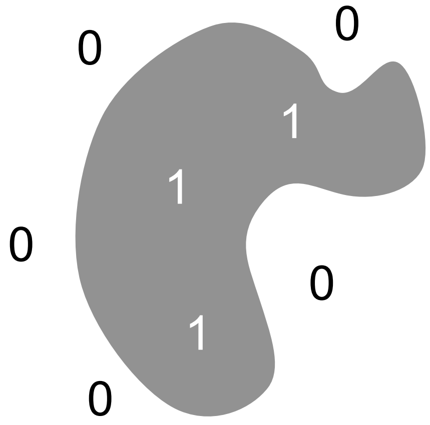
- Khazdan et al. 2006
- Theoretical Insights:
  - Relate oriented point samples to indicator gradient
  - Express reconstruction as a Poisson problem

- Empirical Advantages:
  - Is robust to noise
  - Adapts to the sampling density
  - Can handle large models

The Indicator Function

- Reconstruct the surface of the model by solving for the indicator function of the shape.

$$\chi_M(p) = \begin{cases} 1 & \text{if } p \in M \\ 0 & \text{if } p \notin M \end{cases}$$

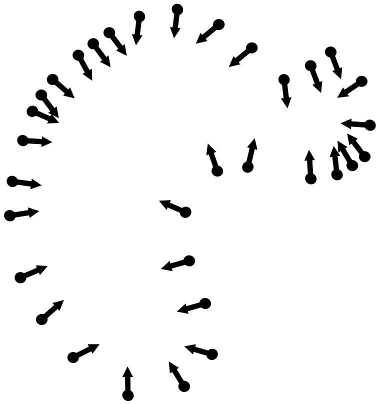


Indicator function

χ_M

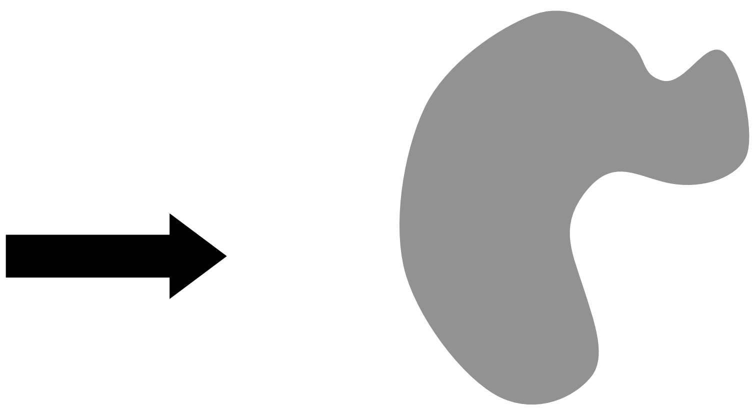
Challenge

- How to construct the indicator function?



Input: Oriented points

$\vec{V}$

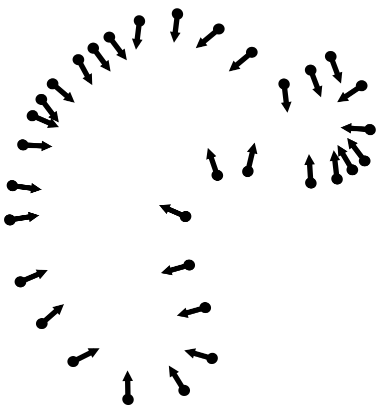


Indicator function

χ_M

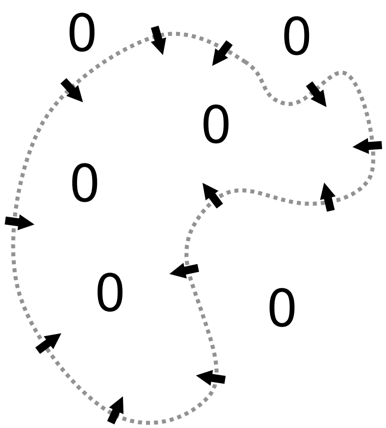
Gradient Relationship

- There is a relationship between the normal field and gradient of indicator function



Input: Oriented points

$\vec{V}$



Indicator gradient

$\nabla \chi_M$

remember indicator function
is constant except the boundary

Integration

- Represent the points by a vector field
- Find the function χ whose gradient best approximates $\vec{V}$:

$$\min_{\chi} \left\| \nabla \chi - \vec{V} \right\|$$

samples of the gradient

Integration as a Poisson Problem

- Represent the points by a vector field
- Find the function χ whose gradient best approximates $\vec{V}$:

$$\min_{\chi} \left\| \nabla \chi - \vec{V} \right\|$$

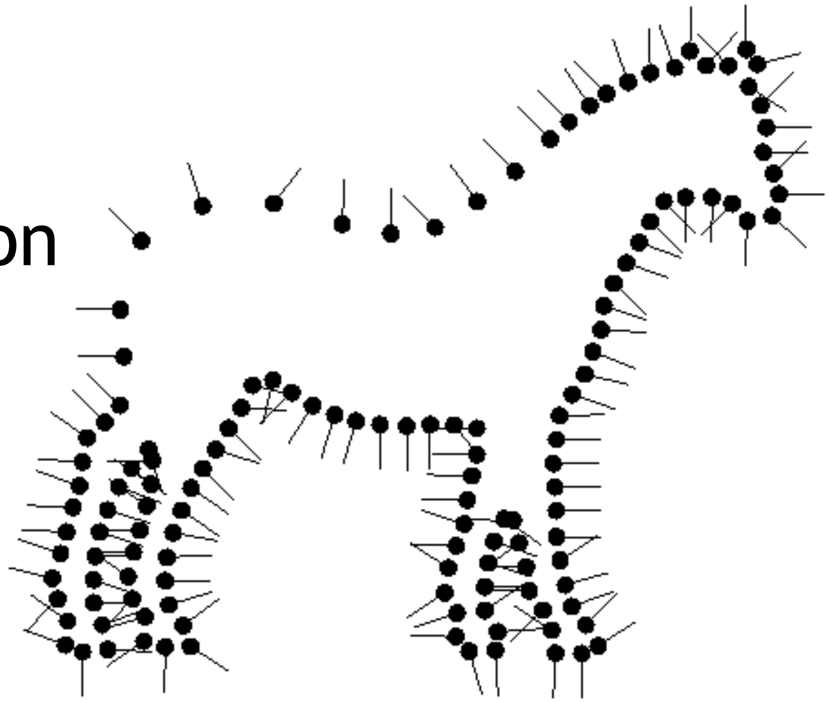
- Applying the divergence operator, we can transform this into a Poisson problem:

$$\nabla \cdot (\nabla \chi) = \nabla \cdot \vec{V} \quad \Leftrightarrow \quad \underbrace{\Delta}_{\text{Laplacian}} \chi = \nabla \cdot \vec{V}$$

Implementation

Given the Oriented Points:

- Set octree
- Compute vector field
- Compute indicator function
- Extract iso-surface



Questions?

What is the main difference between Poisson surface reconstruction v.s. Zippering or VRIP?

Questions?

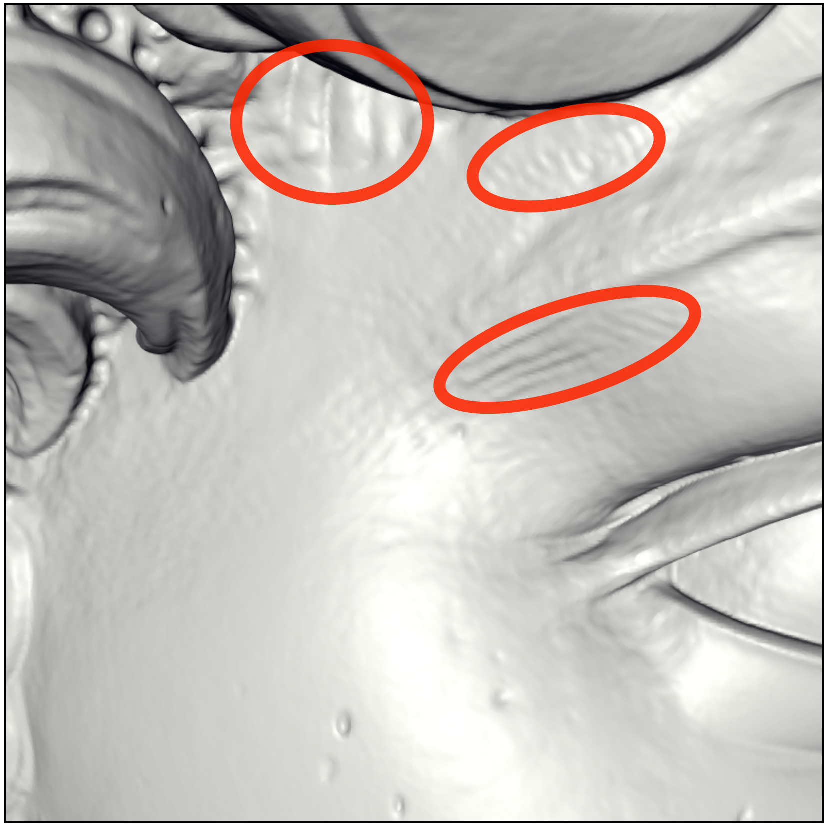
What is the main difference between Poisson surface reconstruction v.s. Zippering or VRIP?

- Global method!

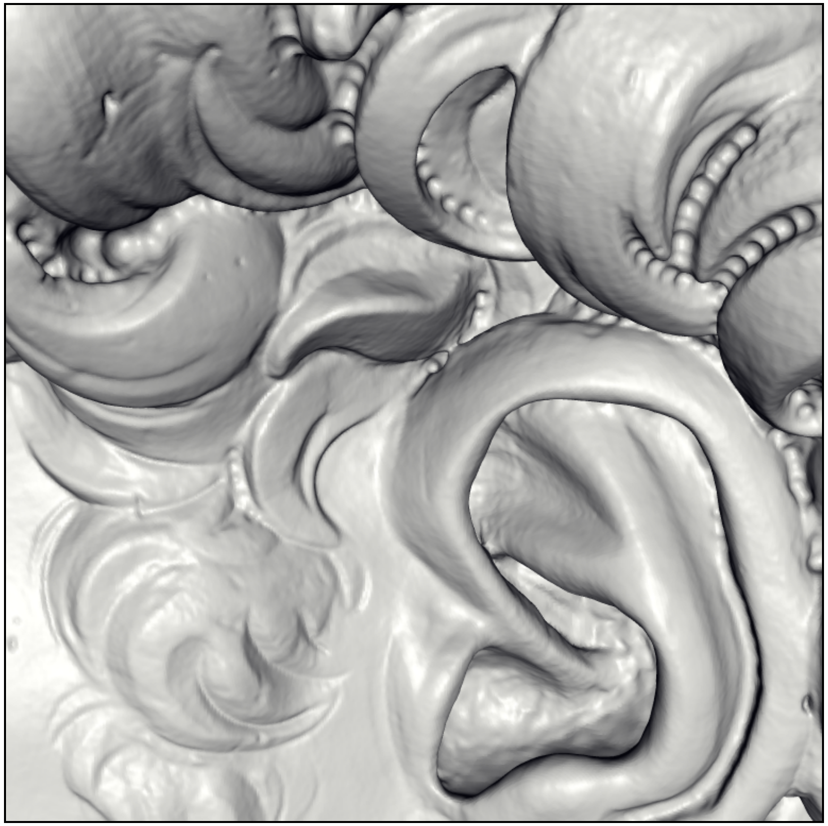
Michelangelo’s David

- 215 million data points from 1000 scans
- 22 million triangle reconstruction
- Compute Time: 2.1 hours
- Peak Memory: 6600MB

David - Chisel Marks

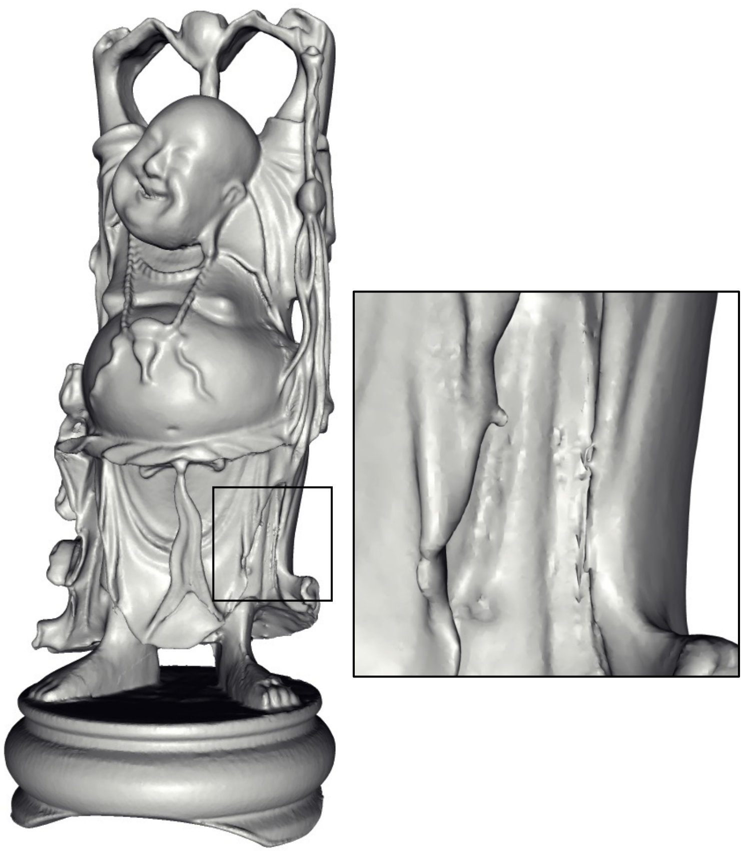


David - Drill Marks



David - Eye

VRIP Comparison



VRIP

Poisson

That's All for Today

- Further Readings:
  - "Efficient Variants of the ICP Algorithm", Szymon Rusinkiewicz and Marc Levoy
    - http://www.cs.princeton.edu/~smr/papers/fasticp/
  - "Zippered Polygon Meshes from Range Images", Greg Turk and Marc Levoy
    - http://graphics.stanford.edu/papers/zipper/
  - "A Volumetric Method for Building Complex Models from Range Images", Brian Curless and Marc Levoy
    - http://www-graphics.stanford.edu/papers/volrange/
  - "Poisson Surface Reconstruction", Kazhdan et al.
    - http://www.cs.jhu.edu/~misha/MyPapers/SGP06.pdf