Surface Modeling and Texture

3D Vision
University of Illinois

Derek Hoiem

Many slides adapted from Lana Lazebnik, Steve Seitz, Yasu Furukawa, Noah Snavely

This class: Surface and Texture

- Meshes and Implicit Surface Functions

- Surface reconstruction methods
  - Poisson and Screened Poisson Reconstruction
  - Floating Scale Surface Reconstruction
  - Delaunay Graph Cuts (Labatut’09)

- Texturing
  - Texrecon (Let there be color!)

Surfaces vs. points

Surfaces

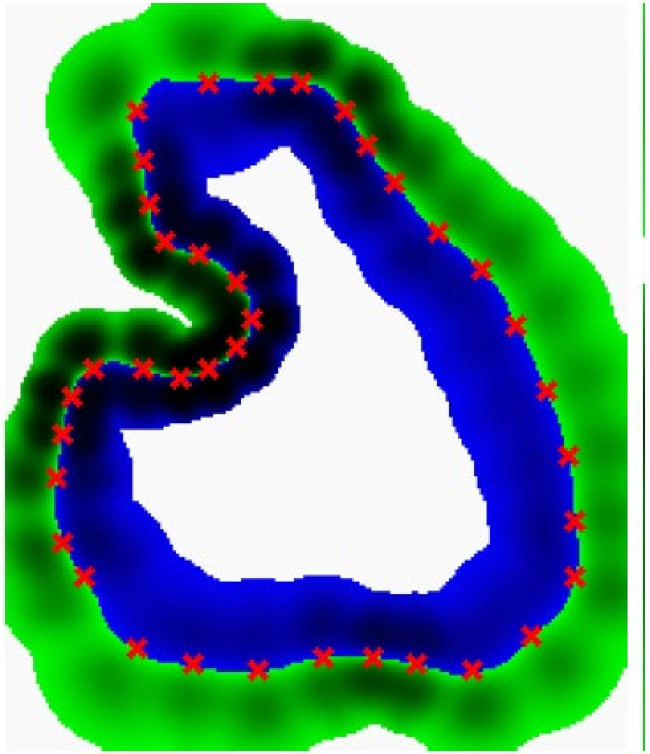
- Structured, encodes connectivity
- More precise measurement
- Render complete images
- May introduce artifacts if incorrectly estimated

Points

- Unstructured, unordered set
- Easy to stream, combine, subsample, manipulate
- More difficult to use for measurement/rendering

Important concepts

- Point cloud
- Octrees
- Oriented points
- Visibility graph
- Mesh
- Implicit surface
  - Surface defined by F(x,y,z)=0
- Explicit surface



Goals of surface generation

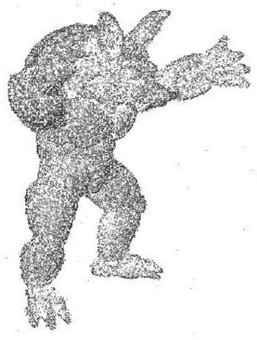
- Approximate the point positions and normals

- Smooth noise while preserving detail

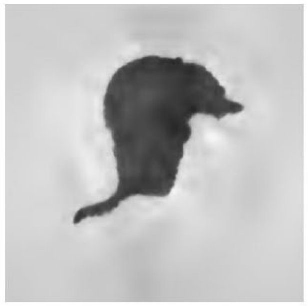
- Incorporate visibility constraints and discard outliers

- Compact and regular mesh

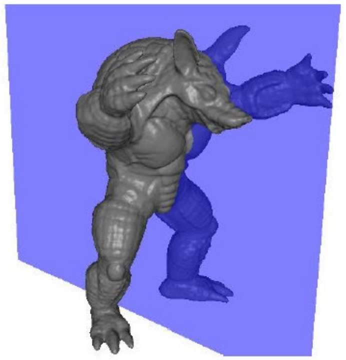
Poisson Reconstruction (Kazhdan et al. 2006)



- Input: oriented points (xyz, N)



- Approach: Solve for what volume is interior/exterior (indicator) and extract isosurface



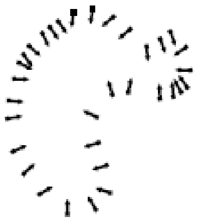
- Output: watertight triangulated mesh

Poisson Reconstruction

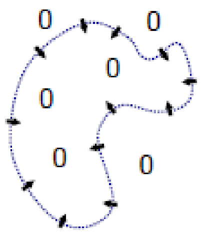
- Create vector field V
  - Populate octree of depth D with oriented points
  - Compute vector field of each point as weighted sum of normals of points in voxel neighborhood
  - Choice of basis functions is a key design decision
  - Points in lower density areas get more weight and larger neighborhoods

- Poisson solution: solve for indicator χ so that its gradient approximates V
  - Solve sparse linear system over octree nodes

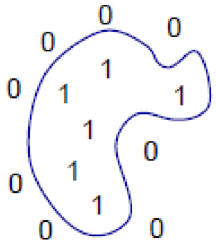
- Extract isosurface using Marching Cubes variant
  - Fit local isosurfaces to 2x2x2 blocks and fuse



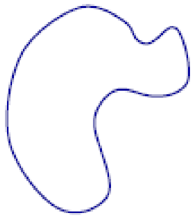
Oriented points
$\vec{V}$



Indicator gradient
$\nabla\chi_M$



Indicator function
χ_M



Surface
∂M

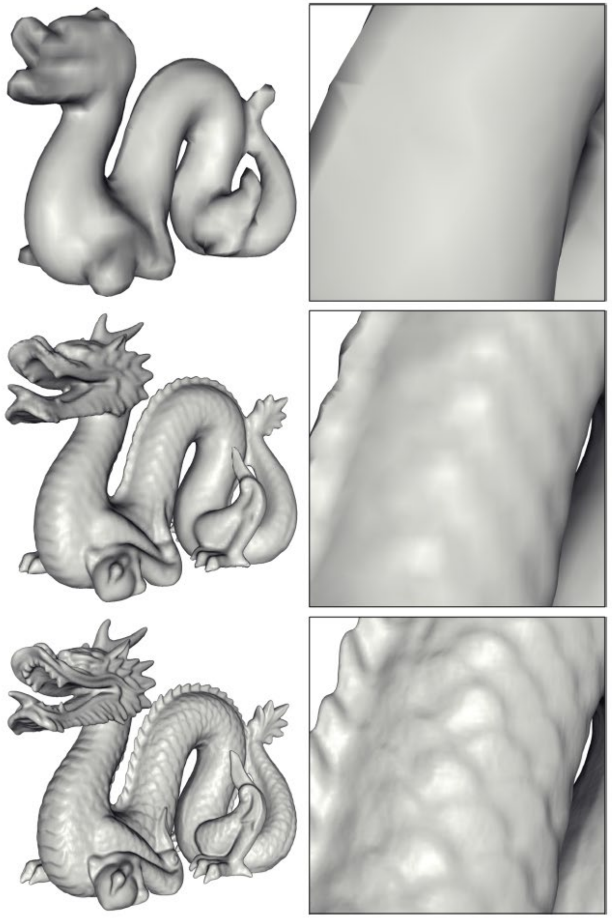


Figure 3: *Reconstructions of the dragon model at octree depths 6 (top), 8 (middle), and 10 (bottom).*

Tree Depth	Time	Peak Memory	# of Tris.
7	6	19	21,000
8	26	75	90,244
9	126	155	374,868
10	633	699	1,516,806

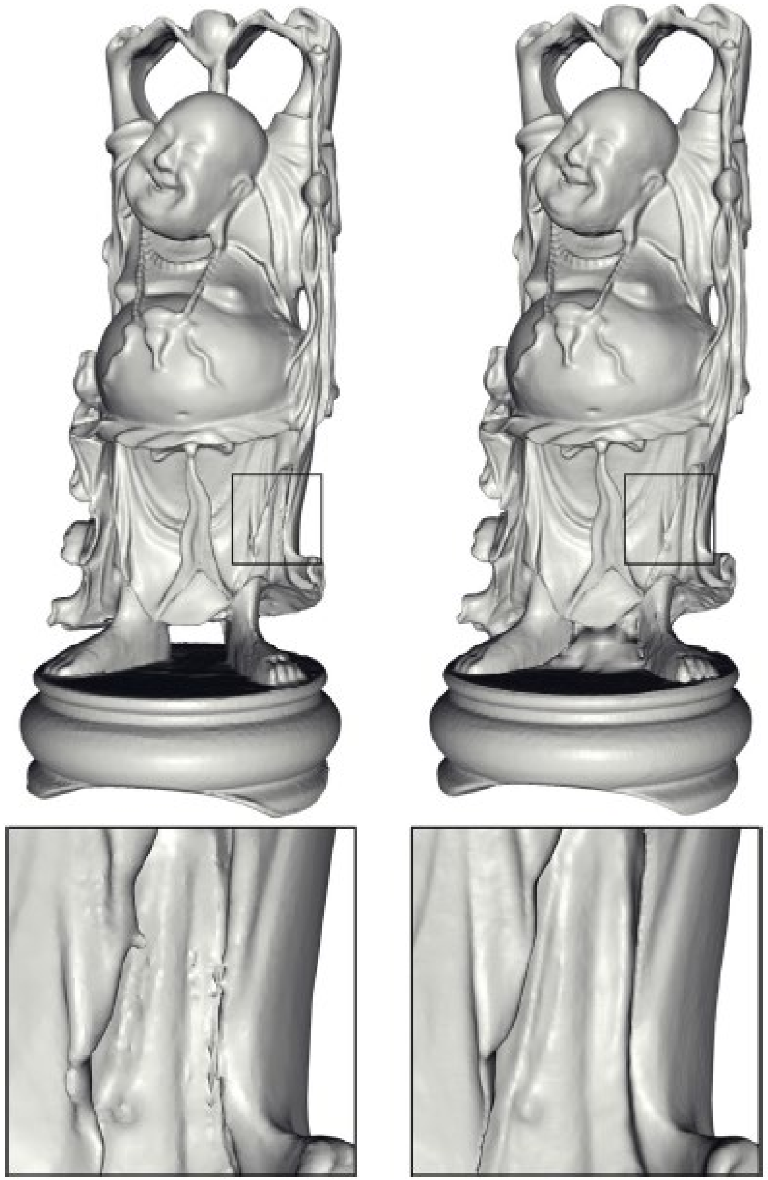


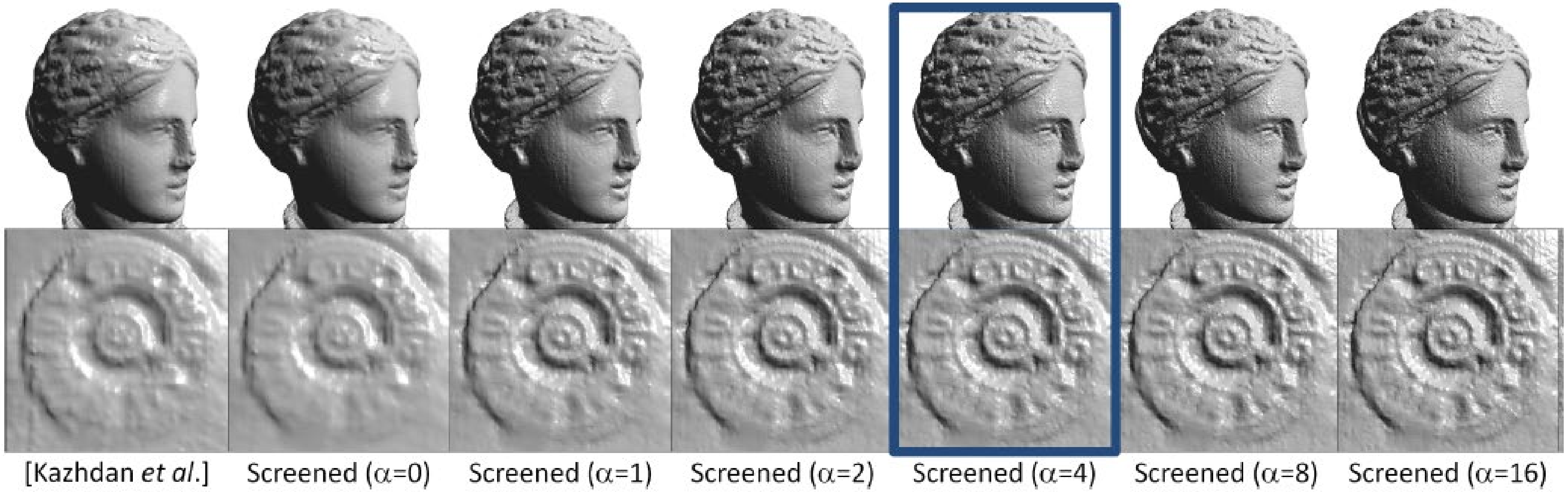
Figure 6: *Reconstructions of the "Happy Buddha" model using VRIP (left) and Poisson reconstruction (right).*

Poisson Reconstruction Pros and Cons

- Well suited to dense, accurate point clouds of objects, e.g. produced by laser scanners
  - Produces watertight surface
  - Completes gaps
  - Fits points while smoothing over noise, controlled by octree depth
- Does not account for visibility constraints or known holes
- Watertight surface can produce artifacts when entire scene is not captured (e.g. outdoors)
  - Requires “trimming” of parts of surface corresponding to shallower octree nodes
- Sensitive to outliers and can oversmooth the data

Screened Poisson Reconstruction (Kazhdan Hoppe 2013)

- Adds term to optimization that penalizes non-zero isovalues at point locations
- Resulting surface fits points more closely
- Improved optimization (faster but uses a little more memory)



Floating Scale Surface Reconstruction (Fuhrmann Goesele 2014)

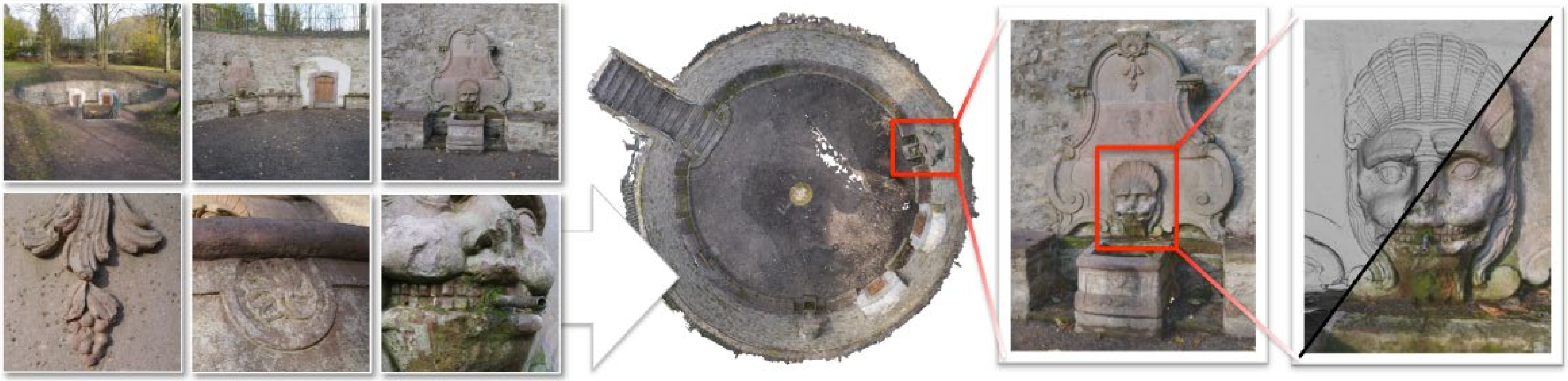
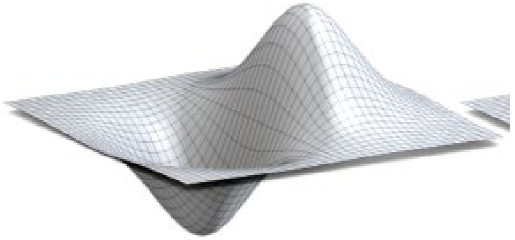


Figure 1: *Floating Scale Surface Reconstruction example. 6 out of 384 input images of a multi-scale dataset (left). Registered images are processed with multi-view stereo which yields depth maps with drastically different sampling rates of the surface. Our algorithm is able to accurately reconstruct every caputured detail of the dataset using a novel multi-scale reconstruction approach (right).*

- Reconstructs surface only where supported by points
- Accounts for scale and confidence of points (scale is different than density here)
- Avoids need for global solution (but still slow)

Floating Scale Surface Reconstruction: Algorithm

- Generate octree
- Sample implicit function at corners of leaf nodes
  - Weighted sum of signed basis functions
  - Basis functions depend on scale, confidence, position, normal of each point
  - Efficient due to octree and limited influence range of basis function
- Marching cubes to extract isosurface



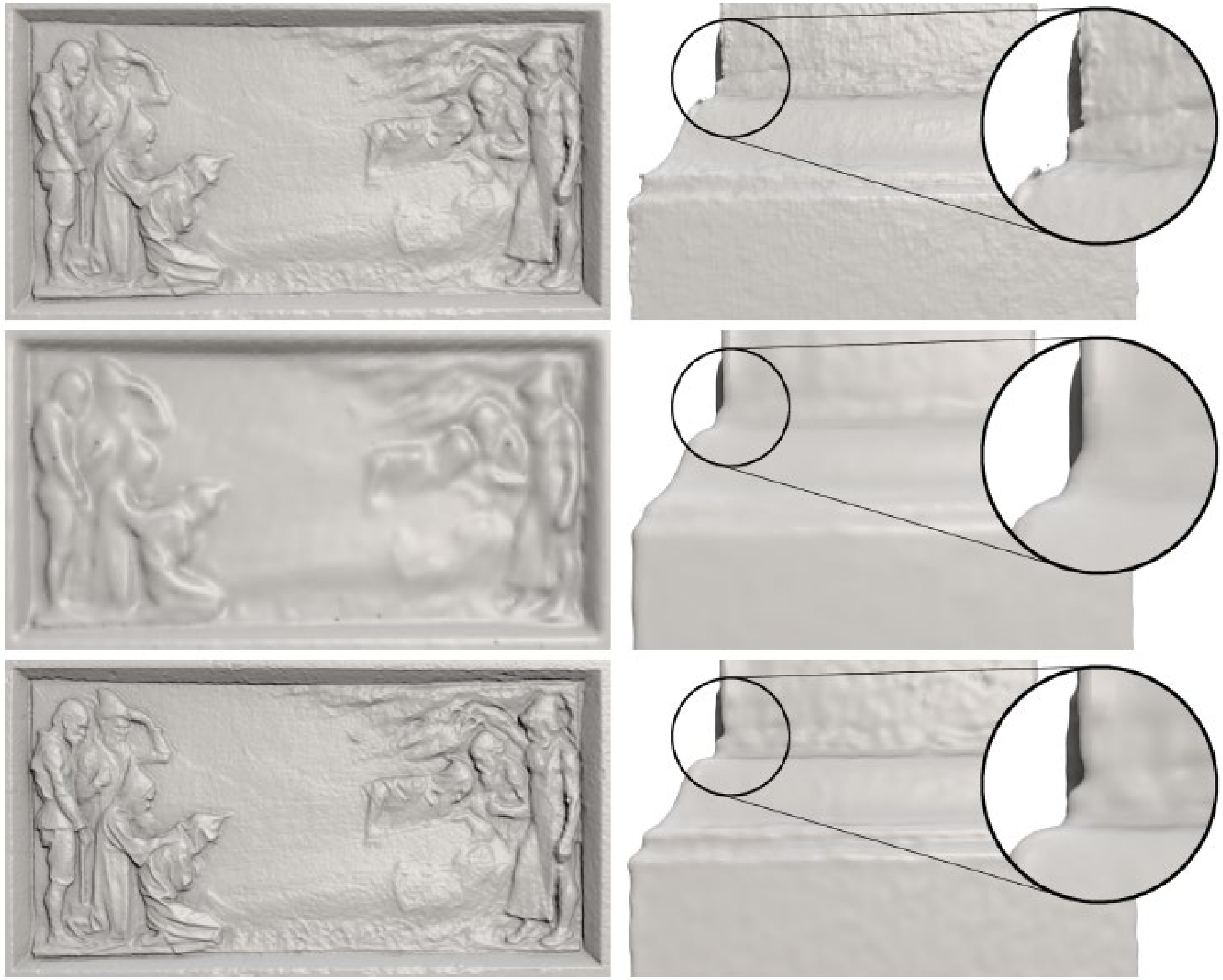


Figure 15: *Comparison with PSR on the* Elisabeth *dataset.* Top row: *Reconstruction with PSR at level 11, which reconstructs details (left) but produces noise in low resolution regions (right).* Middle row: *PSR at level 9 smoothly reconstructs low resolution regions but fails on the details.* Bottom row: *Our method reproduces both high- and low resolution regions appropriately.*

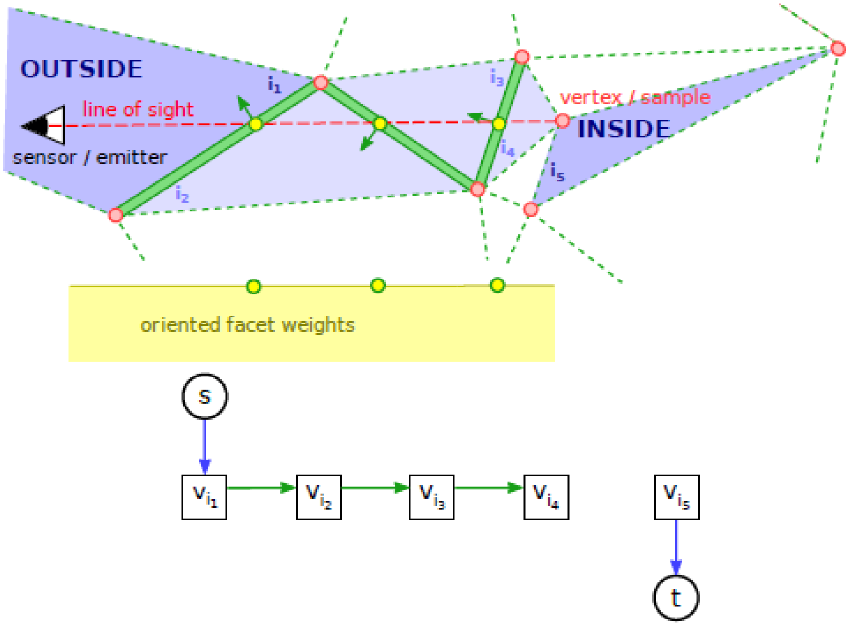
Dataset Name	Number of Samples	Recon. Time	Peak Memory	Output Vertices
Bunny	362 K	30s + 9s	320 MB	277 K
Dragon	2.3 M	83s + 17s	603 MB	455 K
Armadillo	2.4 M	63s + 13s	553 MB	293 K
David	472 M	247m + 38m	114 GB	81.9 M
Temple	22.8 M	5m + 5s	1.96 GB	176 K
Elisabeth	39.3 M	19m + 1m	4.39 GB	2.3 M
Fountain	196 M	178m + 6m	19.9 GB	10.2 M

Delaunay + Graph cuts (Labatut et al. 2009)
“Robust and efficient surface reconstruction from range data”

- Perform Delaunay triangulation on points
  - Results in good and bad faces

$$\mathrm{E}(S) = \mathrm{E}_{\mathrm{vis}}(S) + \lambda_{\mathrm{qual}}\,\mathrm{E}_{\mathrm{qual}}(S)$$

- Label tetrahedra as inside or outside based on visibility and quality
  - Roughly, each ray from visibility graph votes for which tetrahedra are inside and outside and which facets are on the surface
  - “Soft visibility” to allow for noise



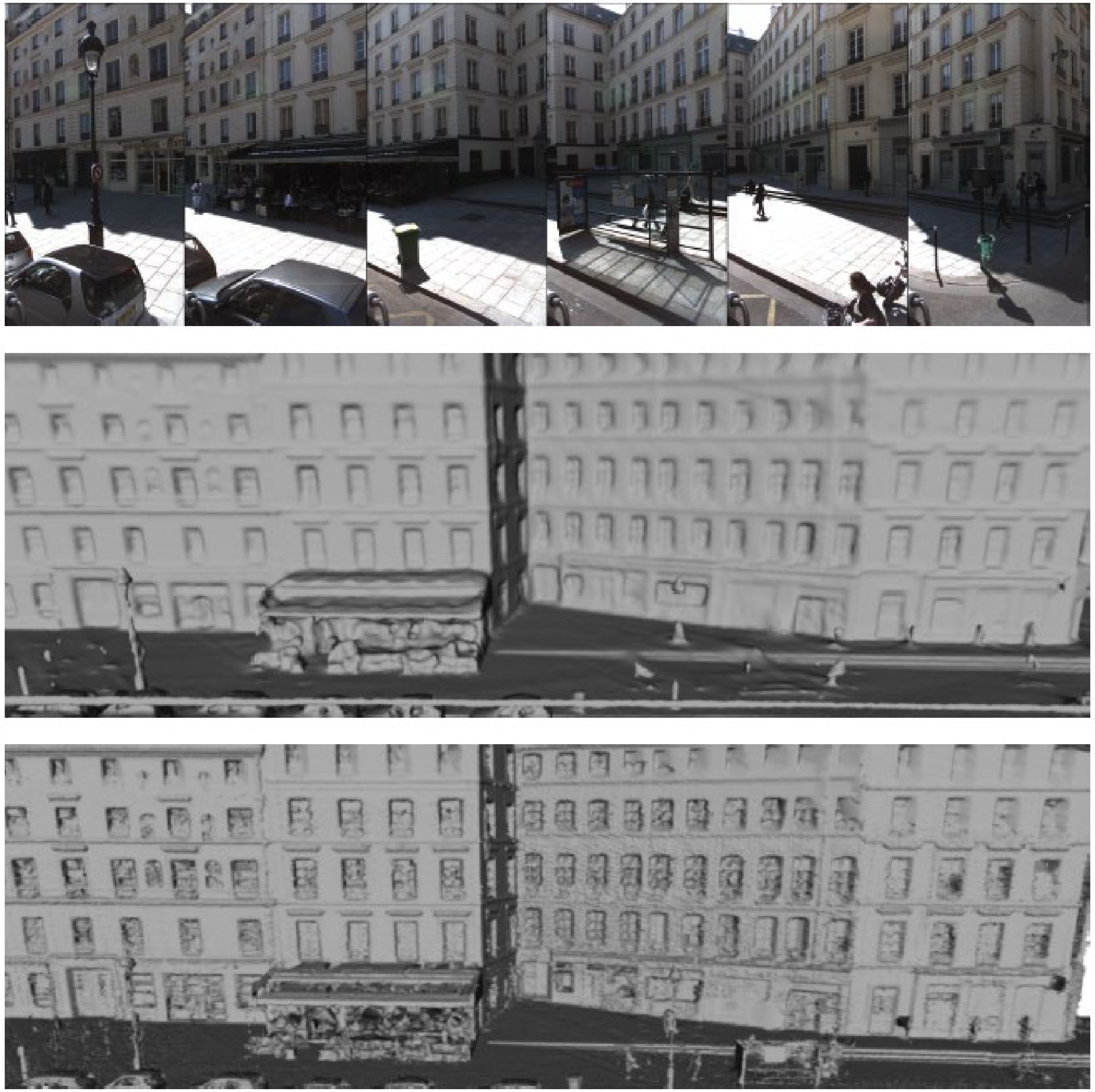
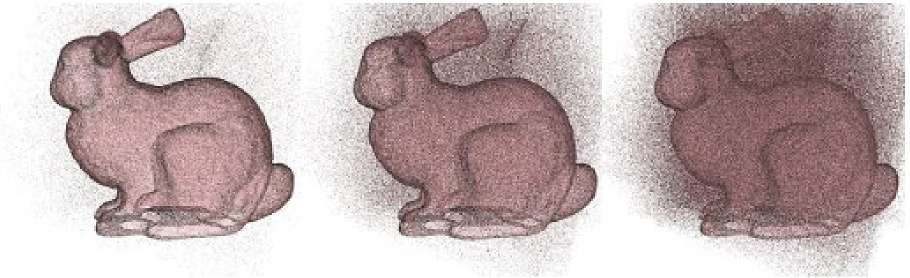
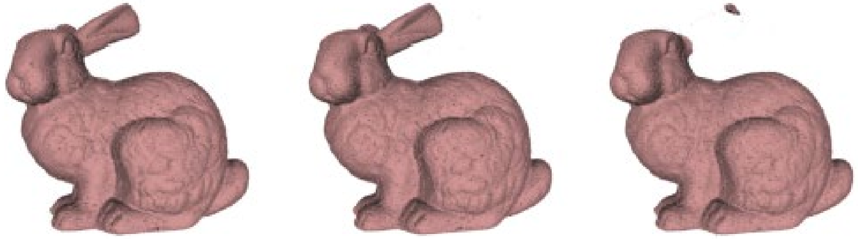


Figure 15: ***Reconstruction details of rue Soufflot.*** Acquired images (top), corresponding reconstruction results of Poisson (middle) and our method (bottom).



(a) Point cloud plus 50,000, 300,000 and 850,000 outliers

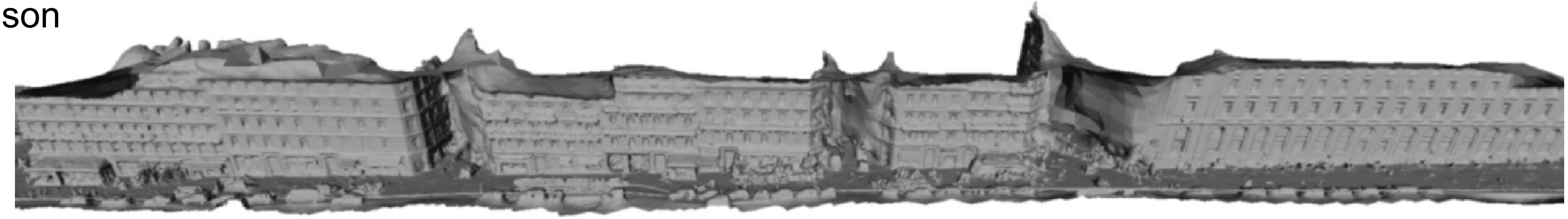
(b) Poisson



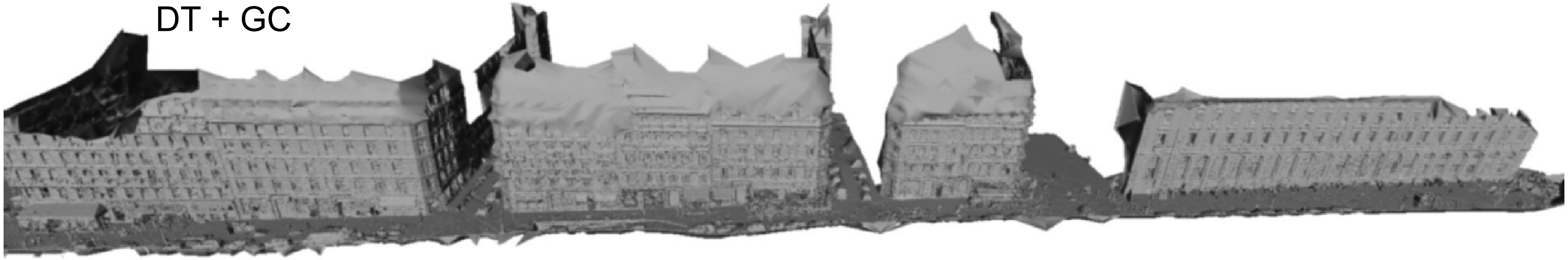
(c) Our method

Figure 13: ***Robustness to large amounts of outliers.***

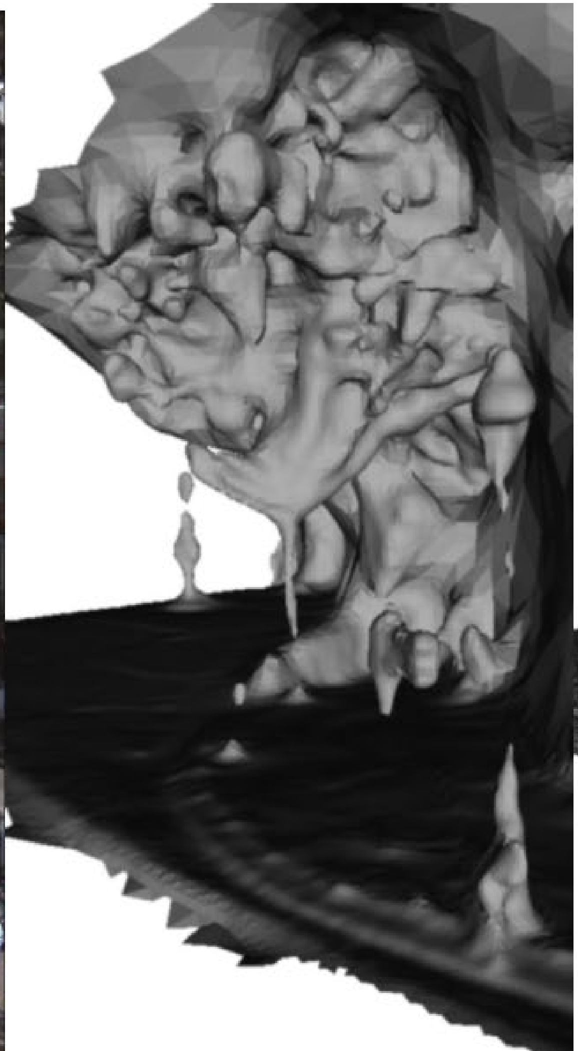
Poisson



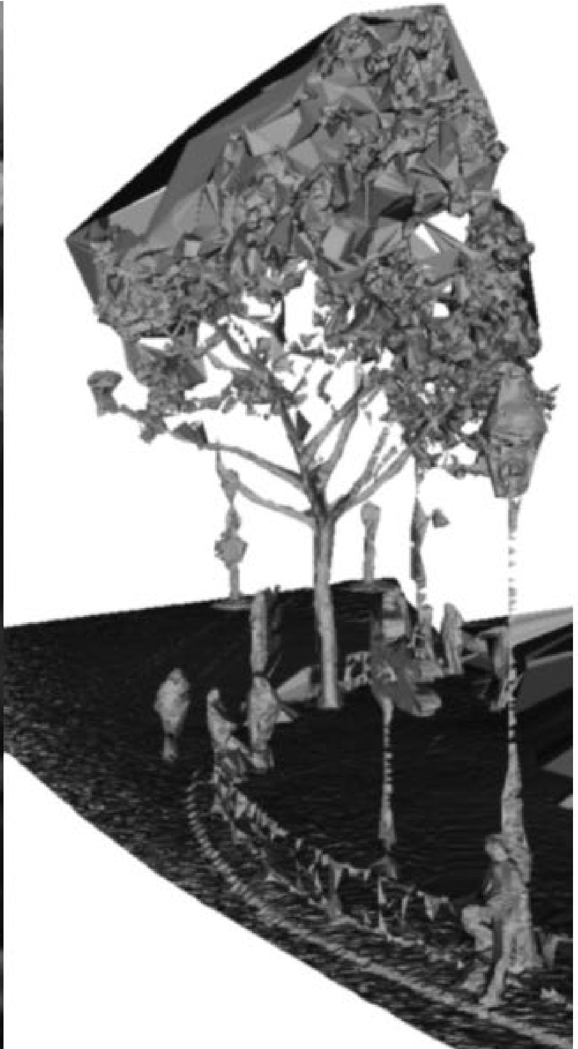
DT + GC



Poisson



DT + GC



Delaunay + Graph Cuts pros and cons

- Preserves most original points
- Removes outliers that violate visibility constraints
- Completes surfaces where not contradicted by visibility graph
- Very detailed (but noisy) mesh
  - Can be smoothed using laplacian smoothing
- Not easy to parallelize, high memory usage
  - Similar running time to Poisson on single thread

Surface reconstruction, bottom line

- Poisson Reconstruction is well used and has continued to be extended; good code available: https://github.com/mkazhdan/PoissonRecon
- Poisson has annoying bubbly artifacts that are avoided by FSSR and DT+GC
- DT+GC effectively removes outliers (especially helpful for MVS inputs) but is more affected by noisy estimates

Mesh Texturing

How to texture meshes (texrecon)

1. Select which view to use to obtain texture for each face
   - Face must be visible to the selected view
   - Prefer close, frontal views, and camera axis passes near face
   - Prefer to use same image for neighboring faces (solve MRF)
   - Apply photo consistency check

2. Adjust colors and blend seams
   - Where neighboring faces are textured by different images, seams will occur
   - Adjust colors to minimize difference at boundary
   - Apply Poisson image editing on strip around boundary to force gradient closer to zero

https://github.com/nmoehrle/mvs-texturing

Open problems

- Not aware of surface reconstruction algorithms that handle large numbers of outliers and smooth away noise (without oversmoothing) well

- Surface reconstruction does not account for learned priors or appearance, e.g.
  - Shape priors, such as planes and cones
  - Repeated structures within and across scenes, e.g. window ledges and columns
  - Geometric smoothness based on image texture

- Texturing is still challenging when there are frequent occlusions (e.g. 360 capture) and relies on quality mesh

Research idea

- Goal: Given an MVS point cloud, produce a mesh that when resampled maximizes accuracy/completion

- Evaluation:
  - Accuracy/completeness on MVS benchmarks, compare raw point cloud to resampled from mesh
  - Speed of mesh construction, compactness of mesh (# faces)

- Rough ideas for approach:
  - Encode points with position, normal, view rays, image features
  - Fill octree and predict occupancy, position, and connectivity for leaf nodes
  - Potentially use graph CNNs or PointNet variant

Summary

- Surface reconstruction is extensively studied for reconstructing laser scanned objects, often fitting isosurface
  - A lot of room for improvement for turning MVS point clouds + images into nice surfaces
- Texture mapping involves finding close, direct, unobstructed views and blending