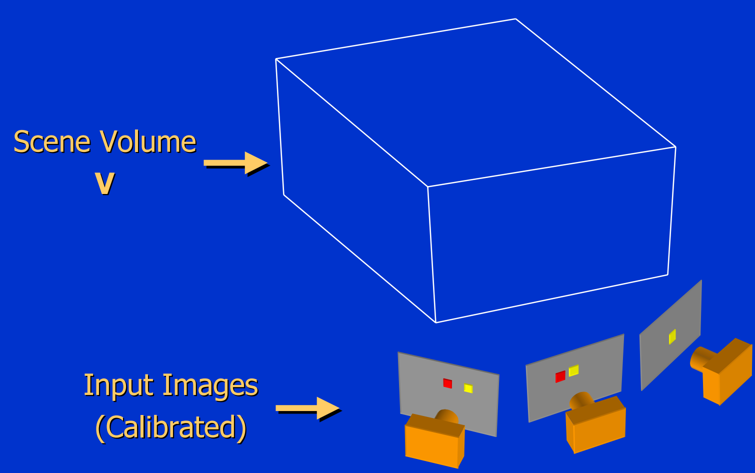
SIGGRAPH 99 Course on
3D Photography

From Images to Voxels

Steve Seitz
Carnegie Mellon University

http://www.cs.cmu.edu/~seitz

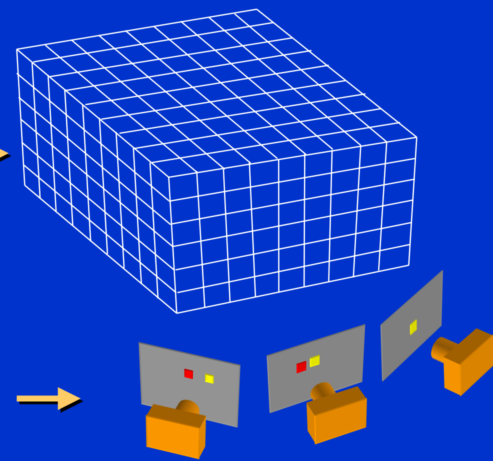
3D Reconstruction from Calibrated Images



Goal: **Determine transparency, radiance of points in V**

Discrete Formulation: Voxel Coloring

Discretized
Scene Volume



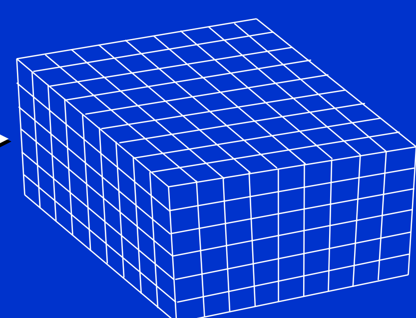
Input Images
(Calibrated)

Goal: **Assign RGBA values to voxels in V**
photo-consistent **with images**

Complexity and Computability

Discretized
Scene Volume

N^3 voxels
C colors



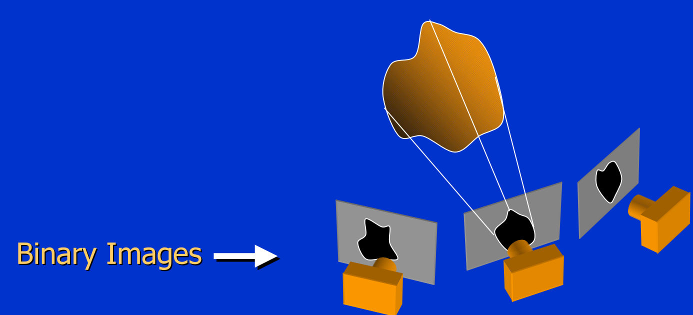
G = space of all colorings (C^{N^3})
ℵ = space of all photo-consistent colorings (computable?)
S = true scene (not computable)

$$S \in \aleph \subset G$$

Voxel Coloring Solutions

1. *C=2 (silhouettes)*
   - Volume intersection [Martin 81, Szeliski 93]
2. *C unconstrained, viewpoint constraints*
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. *General Case*
   - Space carving [Kutulakos & Seitz 98]

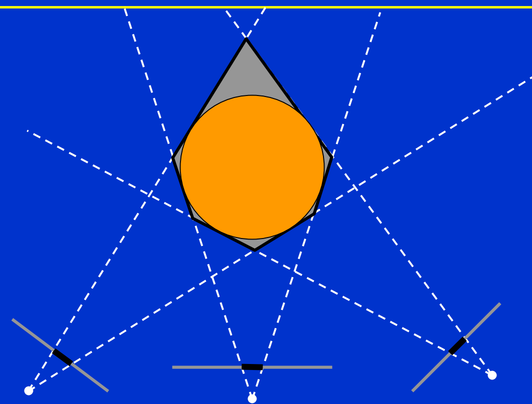
Reconstruction from Silhouettes (C = 2)



Approach:

- ***Backproject*** **each silhouette**
- **Intersect backprojected volumes**

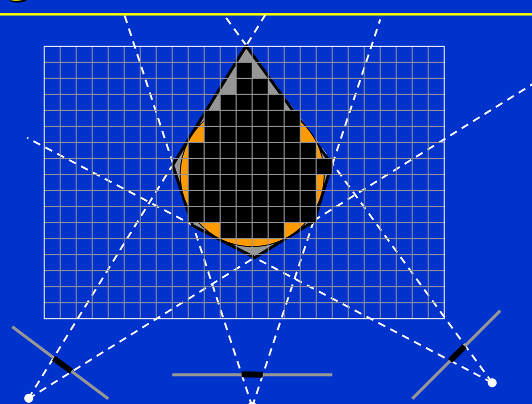
Volume Intersection



Reconstruction Contains the True Scene

- But is generally not the same (no concavities)
- In the limit (all views) get *visual hull* or *line hull*
  - > Complement of all lines that don't intersect S

Voxel Algorithm for Volume Intersection



Color voxel black if on silhouette in every image

- $O(MN^3)$, for M images, N^3 voxels
- Don't have to search 2^{N^3} possible scenes!

Properties of Volume Intersection

Pros

- Easy to implement, fast
- Accelerated via octrees [Szeliski 1993]

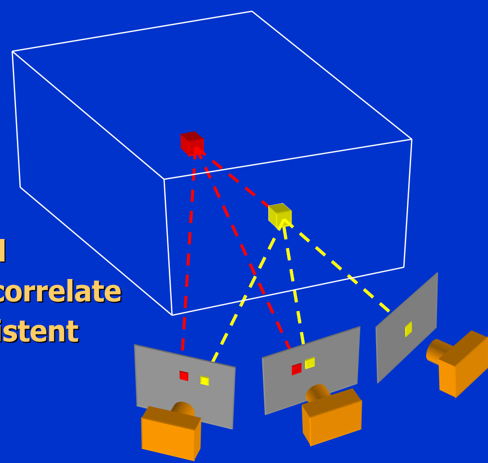
Cons

- No concavities
- Reconstruction is not photo-consistent
- Requires identification of silhouettes

Voxel Coloring Solutions

1. *C=2 (silhouettes)*
   - Volume intersection [Martin 81, Szeliski 93]
2. *C unconstrained, viewpoint constraints*
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. *General Case*
   - Space carving [Kutulakos & Seitz 98]

Voxel Coloring Approach

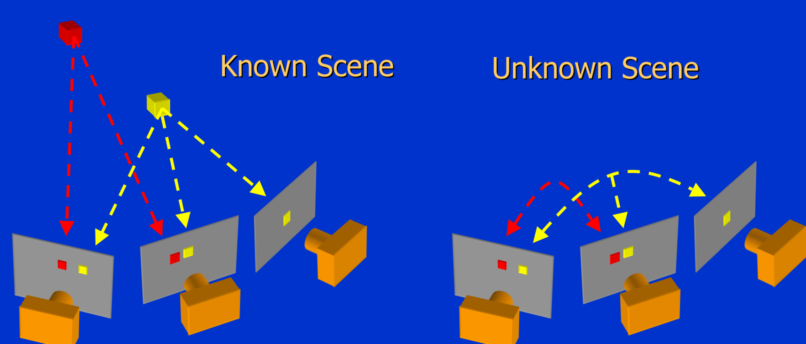


1. Choose voxel
2. Project and correlate
3. Color if consistent

Visibility Problem: in which images is each voxel visible?

The Global Visibility Problem

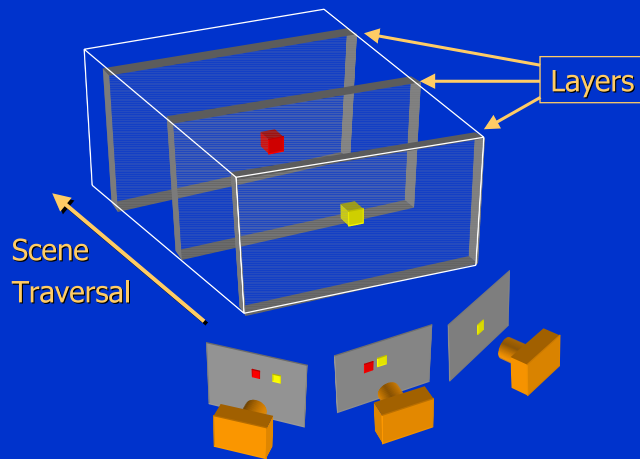
Which points are visible in which images?



Forward Visibility
known scene

Inverse Visibility
known images

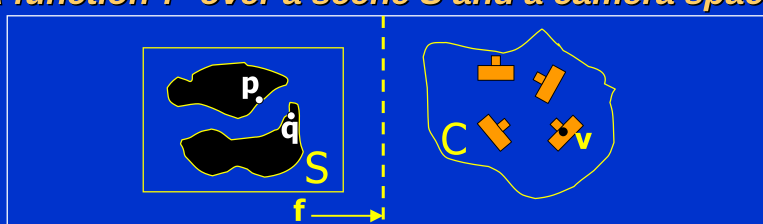
Depth Ordering: visit occluders first!



Condition: depth order is ***view-independent***

What is A *View-Independent* Depth Order?

A function f over a scene S and a camera space C



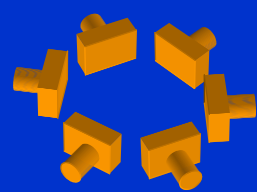
Such that for all p and q in S, v in C

p occludes q from v *only if* f(p) < f(q)

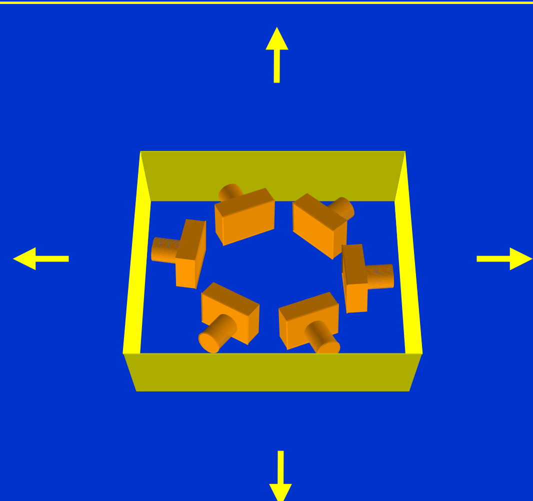
For example: f = distance from separating plane

Panoramic Depth Ordering

- Cameras oriented in many different directions
- Planar depth ordering does not apply

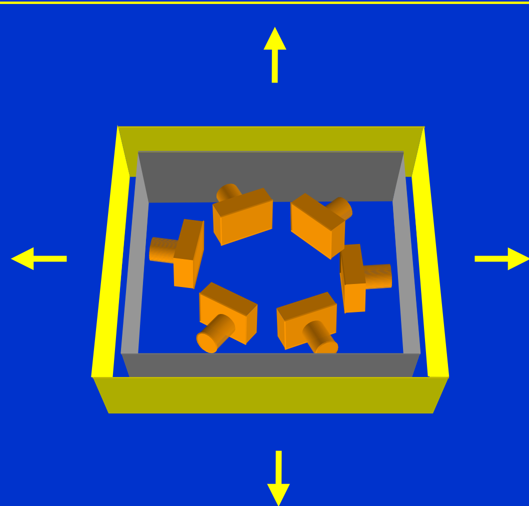


Panoramic Depth Ordering



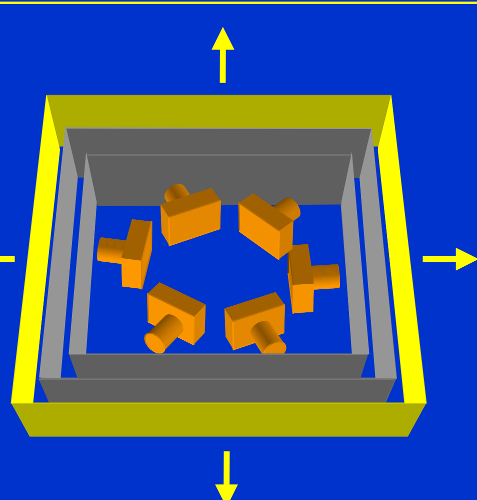
Layers radiate outwards from cameras

Panoramic Layering



Layers radiate outwards from cameras

Panoramic Layering

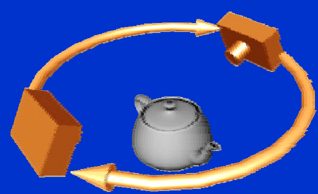


Layers radiate outwards from cameras

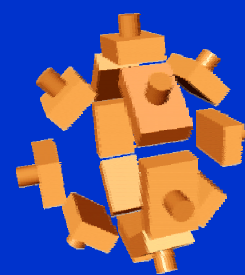
Compatible Camera Configurations

Depth-Order Constraint

- Scene outside convex hull of camera centers

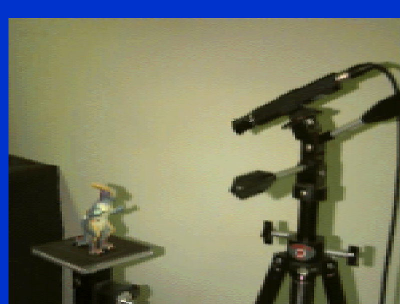


Inward-Looking
cameras above scene

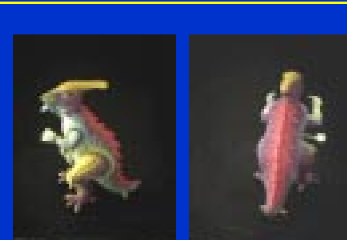


Outward-Looking
cameras inside scene

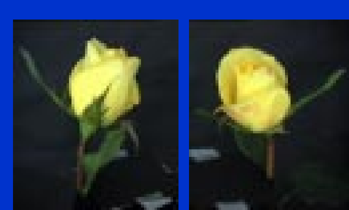
Calibrated Image Acquisition



Calibrated Turntable
360° rotation (21 images)



Selected Dinosaur Images



Selected Flower Images

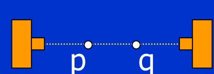
Voxel Coloring Results (Video)

Dinosaur Reconstruction
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Flower Reconstruction
70 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Limitations of Depth Ordering

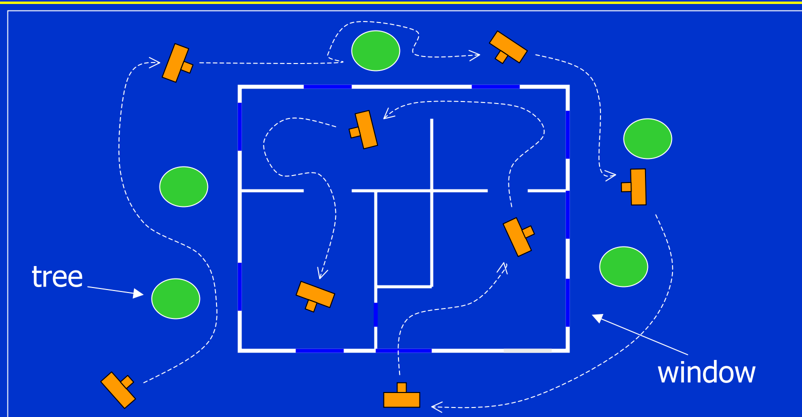
A view-independent depth order may not exist



Need more powerful general-case algorithms

- Unconstrained camera positions
- Unconstrained scene geometry/topology

A More Difficult Problem: Walkthrough



Input: calibrated images from arbitrary positions
Output: 3D model photo-consistent with all images

Voxel Coloring Solutions

1. *C=2 (silhouettes)*
   - Volume intersection [Martin 81, Szeliski 93]
2. *C unconstrained, viewpoint constraints*
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. *General Case*
   - Space carving [Kutulakos & Seitz 98]

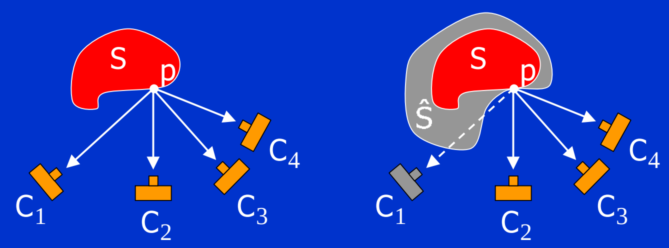
Space Carving Algorithm

- **Step 1: Initialize V to volume containing true scene**
- **Step 2: For every voxel on surface of V**
  - > test *photo-consistency* of voxel
  - > if voxel is inconsistent, carve it
- **Step 3: Repeat Step 2 until all voxels consistent**

Convergence:

- ***Always* converges to a photo-consistent model (when all assumptions are met)**
- **Good results on difficult real-world scenes**

Visibility Property



p ∈ S consistent ⇒ p ∈ Ŝ consistent

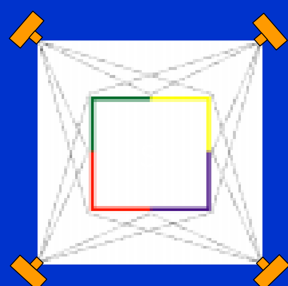
p ∈ Ŝ inconsistent ⇒ p ∈ S inconsistent

This property ensures that carving converges

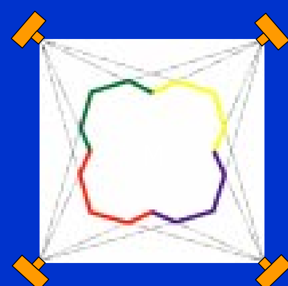
Space Carving Convergence Properties

Properties

- Guaranteed convergence to photo-consistent reconstruction (M) called the *photo hull*
  - > $M = \cup \aleph$ --- union of all photo-consistent scenes
- Tightest possible bound on true scene
- *Worst case #* consistency checks: (# cameras)^2(# voxels)



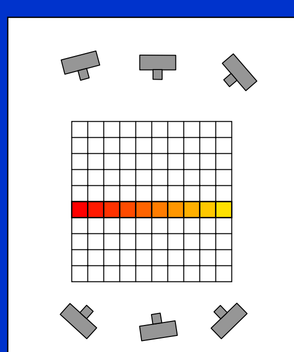
True Scene



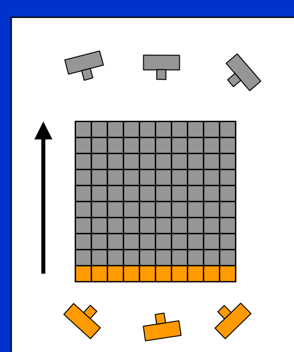
Reconstruction

Multi-Pass Plane Sweep

- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence



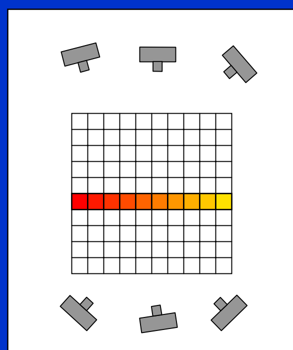
True Scene

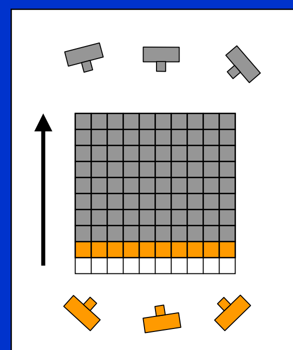


Reconstruction

Multi-Pass Plane Sweep

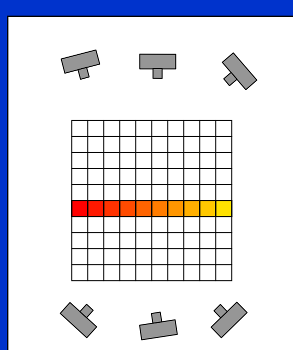
- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence

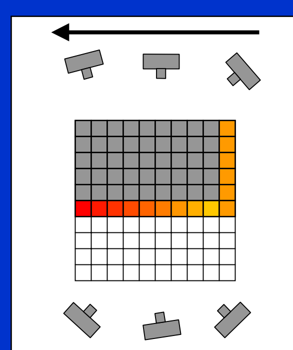




Multi-Pass Plane Sweep

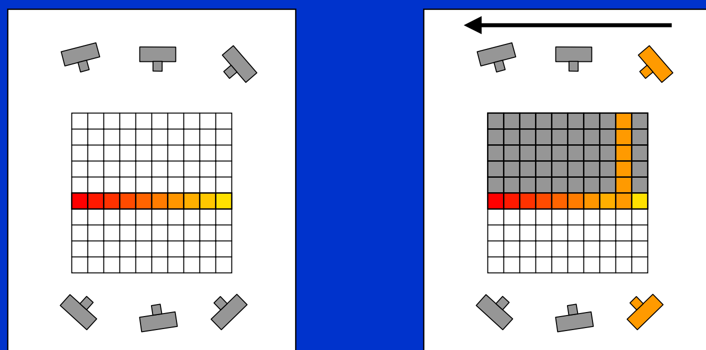
- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence





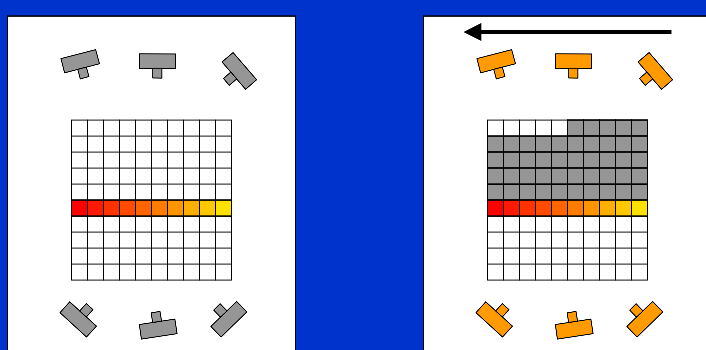
Multi-Pass Plane Sweep

- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence



Multi-Pass Plane Sweep

- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence



Space Carving Algorithm

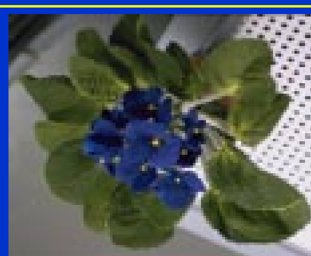
Optimal algorithm is unwieldy

- Complex visibility update procedure

Alternative: multi-pass plane sweep

- Efficient, can use texture-mapping hardware
- Converges quickly in practice
- Easy to implement

Space Carving Results: African Violet



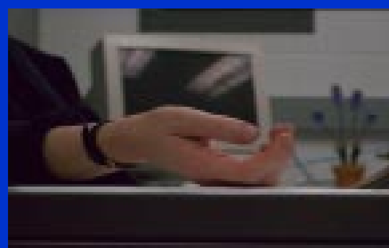
Input Image (1 of 45)

Reconstruction

Reconstruction

Reconstruction

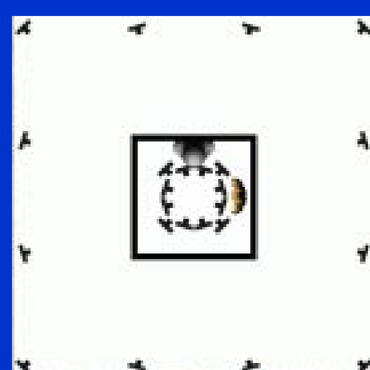
Space Carving Results: Hand



Input Image
(1 of 100)

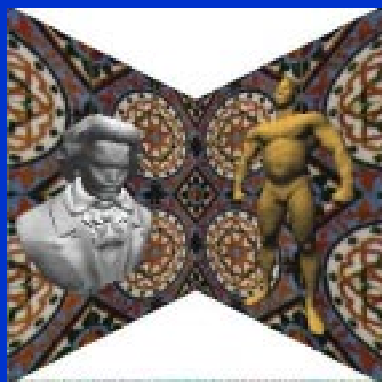
Views of Reconstruction

House Walkthrough

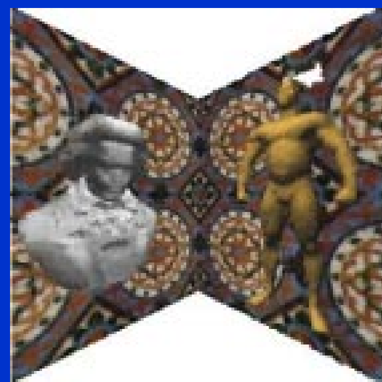


24 rendered input views from inside and *outside*

Space Carving Results: House

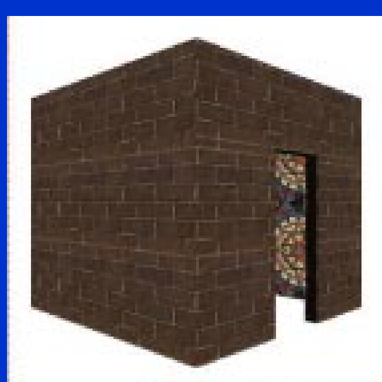


Input Image
(true scene)



Reconstruction
370,000 voxels

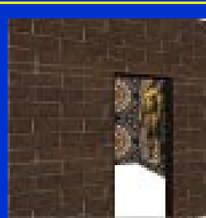
Space Carving Results: House



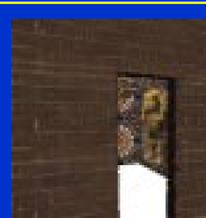
Input Image
(true scene)

Reconstruction
370,000 voxels

Space Carving Results: House



New View (true scene)



Reconstruction

New View
(true scene)

Reconstruction

Reconstruction
(with new input view)

Other Features

Coarse-to-fine Reconstruction

- Represent scene as octree
- Reconstruct low-res model first, then refine

Hardware-Acceleration

- Use texture-mapping to compute voxel projections
- Process voxels an entire plane at a time

Limitations

- Need to acquire calibrated images
- Restriction to simple radiance models
- Bias toward maximal (fat) reconstructions
- Transparency not supported

Other Approaches

Level-Set Methods [Faugeras & Keriven 1998]

- Evolve implicit function by solving PDE's

Transparency and Matting [Szeliski & Golland 1998]

- Compute voxels with alpha-channel

Max Flow/Min Cut [Roy & Cox 1998]

- Graph theoretic formulation

Mesh-Based Stereo [Fua & Leclerc 95]

- Mesh-based but similar consistency formulation

Virtualized Reality [Narayan, Rander, Kanade 1998]

- Perform stereo 3 images at a time, merge results

Conclusions

Advantages of Voxels

- Non-parametric
  - > can model arbitrary geometry
  - > can model arbitrary topology
- Good reconstruction algorithms
- Good rendering algorithms (splatting, LDI)

Disadvantages

- Expensive to process hi-res voxel grids
- Large number of parameters
  - > Simple scenes (e.g., planes) require lots of voxels
- Meshes simplify better

Bibliography

Volume Intersection

- Martin & Aggarwal, "Volumetric description of objects from multiple views", Trans. Pattern Analysis and Machine Intelligence, 5(2), 1991, pp. 150-158.
- Szeliski, "Rapid Octree Construction from Image Sequences", Computer Vision, Graphics, and Image Processing: Image Understanding, 58(1), 1993, pp. 23-32.

Voxel Coloring and Space Carving

- Seitz & Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring", Proc. Computer Vision and Pattern Recognition (CVPR), 1997, pp. 1067-1073.
- Seitz & Kutulakos, "Plenoptic Image Editing", Proc. Int. Conf. on Computer Vision (ICCV), 1998, pp. 17-24.
- Kutulakos & Seitz, "A Theory of Shape by Space Carving", U. Rochester C.S. Dept. TR #692, May 1998, to appear in Proc. ICCV 99.

Bibliography

Related References

- Faugeras & Keriven, "Variational principles, surface evolution, PDE's, level set methods and the stereo problem", IEEE Trans. on Image Processing, 7(3), 1998, pp. 336-344.
- Szeliski & Golland, "Stereo Matching with Transparency and Matting", Proc. Int. Conf. on Computer Vision (ICCV), 1998, 517-524.
- Roy & Cox, "A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem", Proc. ICCV, 1998, pp. 492-499.
- Fua & Leclerc, "Object-centered surface reconstruction: Combining multi-image stereo and shading", Int. Journal of Computer Vision, 16, 1995, pp. 35-56.
- Narayanan, Rander, & Kanade, "Constructing Virtual Worlds Using Dense Stereo", Proc. ICCV, 1998, pp. 3-10.