Volumetric Scene Reconstruction from Multiple Views

Chuck Dyer

University of Wisconsin

dyer@cs.wisc.edu

www.cs.wisc.edu/~dyer

Image-Based Scene Reconstruction

Goal

- *Automatic construction of photo-realistic 3D models of a scene from multiple images taken from a set of arbitrary viewpoints*
- *Image-based modeling; 3D photography*

Applications

- *Interactive visualization of remote environments or objects by a virtual video camera for flybys, mission rehearsal and planning, site analysis, treaty monitoring*
- *Virtual modification of a real scene for augmented reality tasks*

Two General Approaches

World Representation

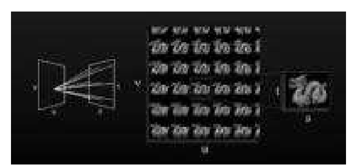
- **World centered: Recover a complete 3D geometric (and possibly photometric) model of scene**
- **Operations: feature correspondence, tracking, calibration, structure from motion, model fitting, ...**

Plenoptic Function Representation

- **Camera centered: Integration of images which sample scene geometry**
- **E.g., panoramas, light fields, LDIs**
- **Operations: image segmentation, registration, warping, compositing, interpolation, ...**

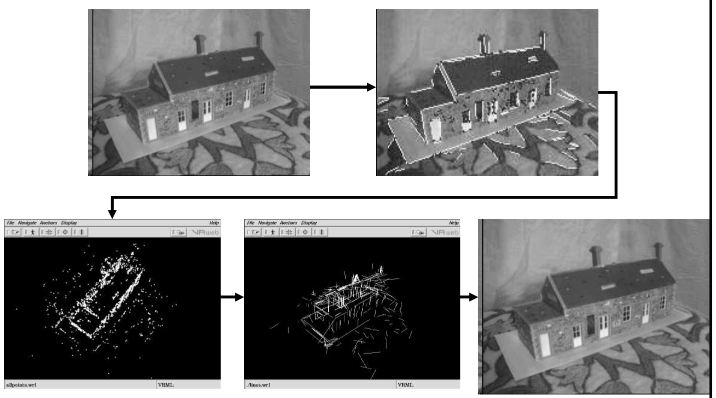
Light Fields

A range of viewpoints represented by a set of images [Levoy and Hanrahan, 1996]



Standard Approach: Multiple View Stereo

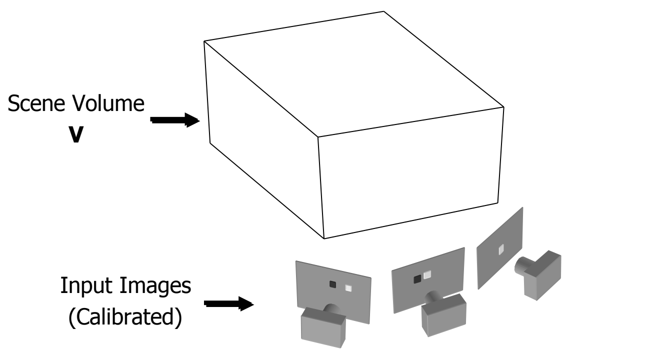
[Fitzgibbon and Zisserman, 1998]



Weaknesses of the Standard Approach

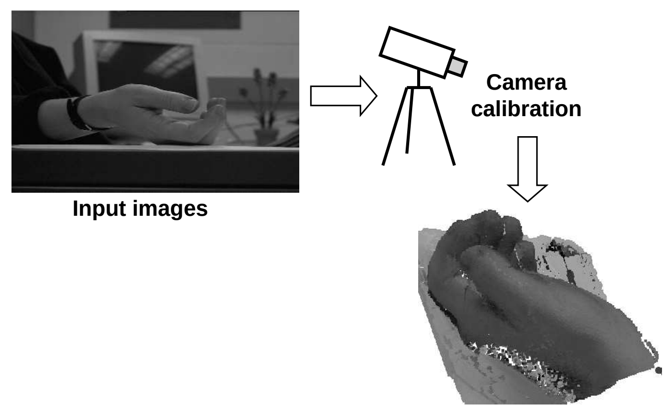
- *Views must be close together in order to obtain point correspondences*
- *Point correspondences must be tracked over many consecutive frames*
- *Many partial models must be fused*
- *Must fit a parameterized surface model to point features*
- *No explicit handling of occlusion differences between views*

Our Approach: Volumetric Scene Modeling

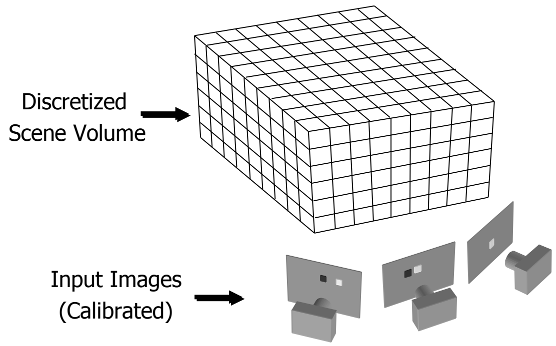


Goal: **Determine transparency and radiance of points in V**

3D Scene Reconstruction from Multiple Views

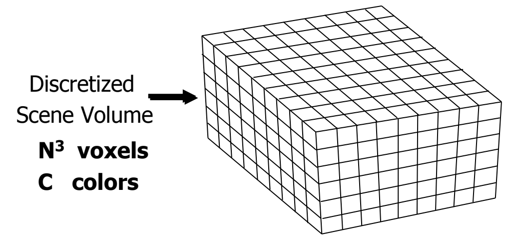


Discrete Formulation: Voxel Space



Goal: **Assign RGBA values to voxels in V that are *photo-consistent* with all input images**

Complexity and Computability



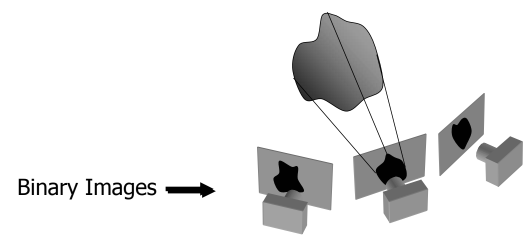
G = ***space of all colorings*** (C^{N^3})
P = ***space of all photo-consistent colorings (computable?)***
S = true scene (not computable)

$$S \in P \subset G$$

Voxel-based Scene Reconstruction Methods

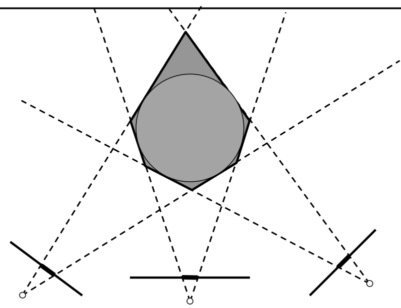
1. ***Shape from Silhouettes***
   - **Volume intersection** [Martin & Aggarwal, 1983]
2. ***Shape from Photo-Consistency***
   - **Voxel coloring** [Seitz & Dyer, 1997]
   - **Space carving** [Kutulakos & Seitz, 1999]

Reconstruction from Silhouettes



Approach:
- ***Backproject*** **each silhouette**
- **Intersect backprojected generalized-cone volumes**

Volume Intersection



Reconstruction contains the true scene

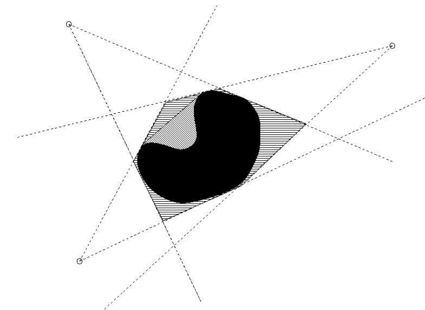
Best case (infinite # views): visual hull

(complement of all lines that don't intersect S)

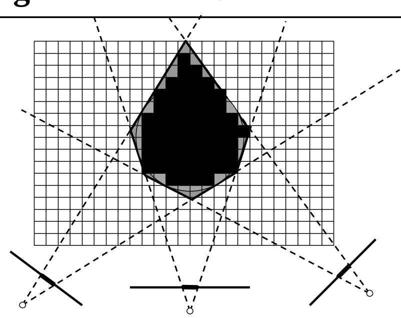
- 2D: convex hull
- 3D: convex hull – hyperbolic regions

Shape from Silhouettes

Reconstruction = object + concavities + points not visible



Voxel Algorithm for Volume Intersection

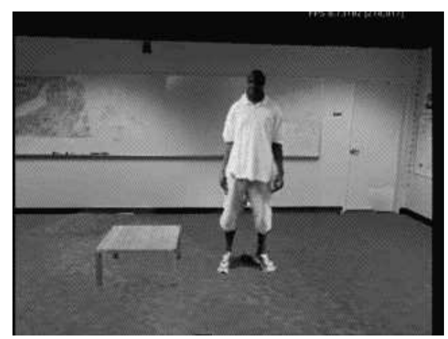


Color voxel black if in silhouette in every image

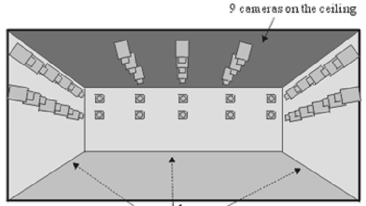
- **O(MN^3) time for M images, N^3 voxels**
- **Don't have to search 2^{N^3} possible scenes**

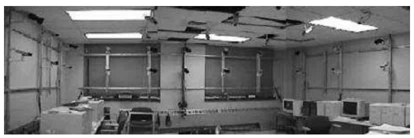
Image-based Visual Hulls

[Matusik *et al.*, 2000]



CMU's Virtualized Reality System

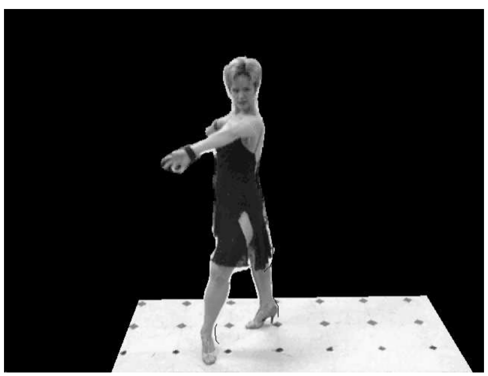




Shape from 49 Silhouettes

Surface model constructed using Marching Cubes algorithm

Virtual Camera Fly-By



Texture mapped and sound synthesized from 6 sources

Properties of Volume Intersection

Pros

- **Easy to implement**
- **Accelerated via octrees**

Cons

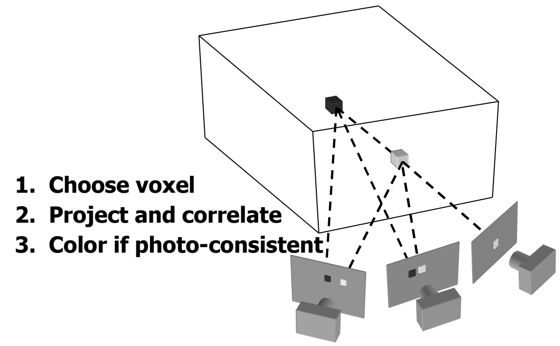
- **Concavities are not reconstructed**
- **Reconstruction does not use photometric properties in each image**
- **Requires image segmentation to extract silhouettes**

Voxel-based Scene Reconstruction Methods

1. ***Shape from Silhouettes***
   - **Volume intersection** [Martin & Aggarwal, 1983]

2. ***Shape from Photo-Consistency***
   - **Voxel coloring** [Seitz & Dyer, 1997]
   - **Space carving** [Kutulakos & Seitz, 1999]

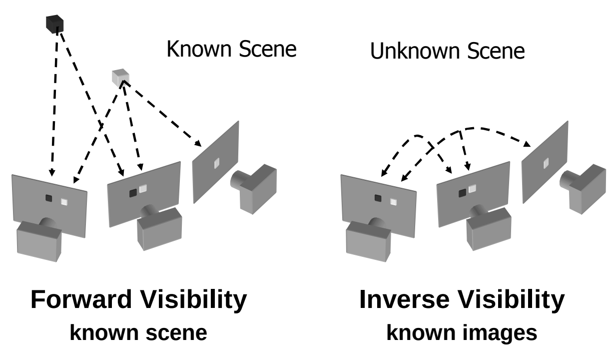
Voxel Coloring Approach



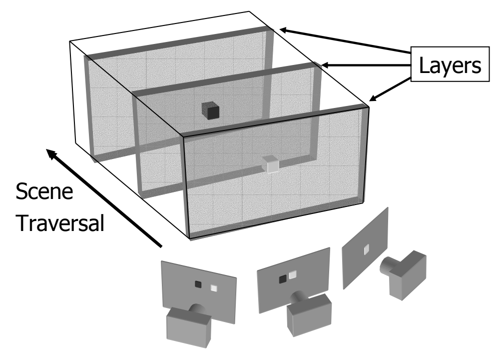
Visibility Problem: **In which images is each voxel visible?**

The Global Visibility Problem

Which points are visible in which images?



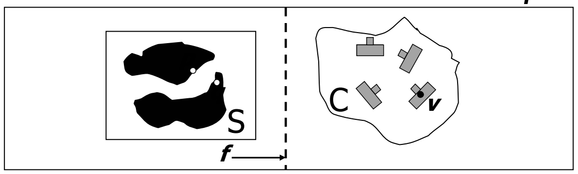
Depth Ordering: Visit Occluders First



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**

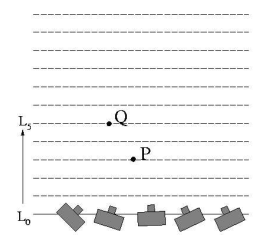
p occludes q from *v* *only if* `f(p) < f(q)`

For example: ***f*** **= distance from separating plane**

$\Rightarrow$ **Plane Sweep order** [Collins, 1996]

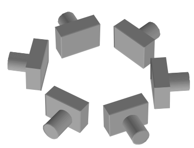
Example: 2D Scene and Line of Cameras

- ***Arrange cameras to simplify occlusion relationships***
- ***Depth-order traversal of voxels determines visibility***

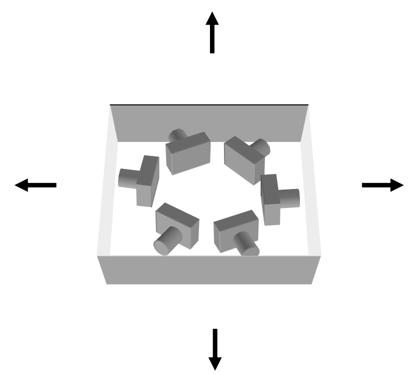


Panoramic Depth Ordering

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

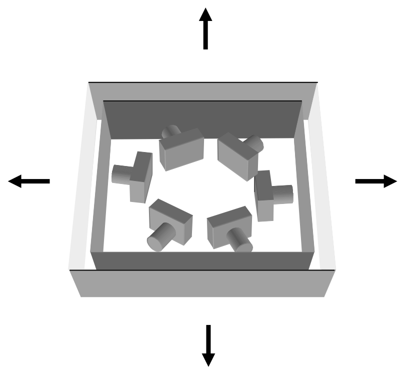


Panoramic Depth Ordering



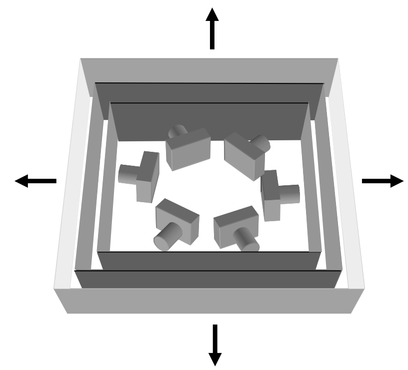
Layers radiate outwards from cameras

Panoramic Layering



Layers radiate outwards from cameras

Panoramic Layering

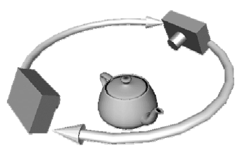


Layers radiate outwards from cameras

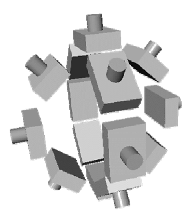
Compatible Camera Configurations

Depth-Order Constraint

- **Scene outside convex hull of camera centers**



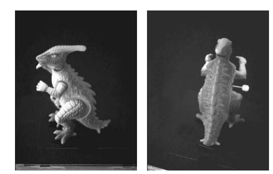
Inward-Looking
cameras above scene



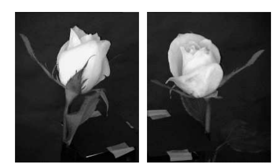
Outward-Looking
cameras inside scene

Calibrated Image Acquisition

Calibrated Turntable
360° rotation (21 images)

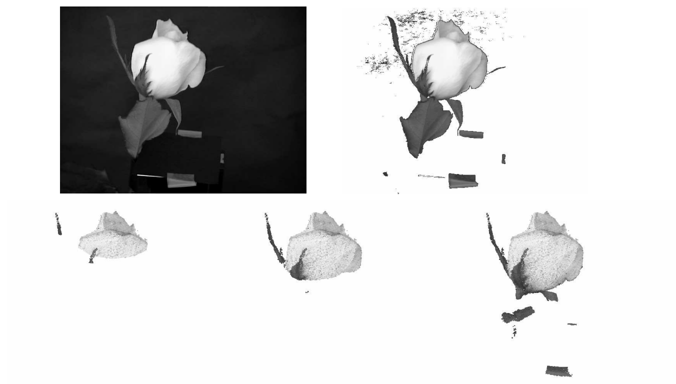


Selected Dinosaur Images



Selected Flower Images

Layered Scene Traversal

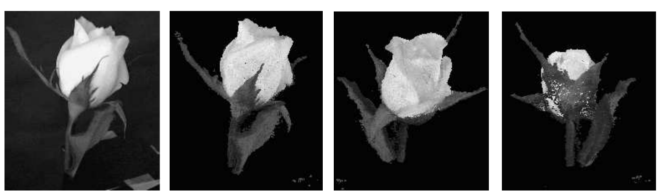


Results: Dinosaur

21 input images
spanning 360° rotation

1K voxels 5K voxels 72K voxels

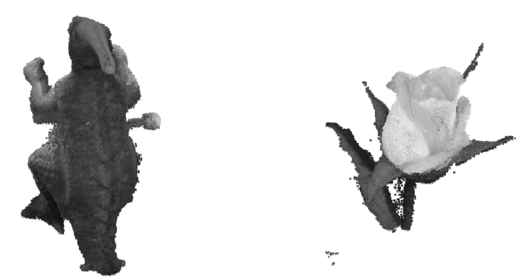
Results: Rose



1 of 21 input images

3 synthesized views

Results



Dinosaur Reconstruction
21 input images
72 K voxels colored
7.6 M voxels tested

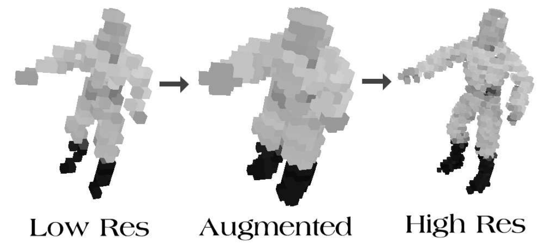
Flower Reconstruction
21 input images
70 K voxels colored
7.6 M voxels tested

Scaling Up Voxel Coloring

- *Time complexity $\propto$ #voxels $\times$ #images*
- *Too many voxels in large, high-resolution scenes*
- *Enhancements*
  - Texture mapping – use hardware to project images to each layer of voxels
  - Variable voxel resolution – use octrees and coarse-to-fine processing
  - Volumetric warping – warp voxel space to extend to an infinite domain

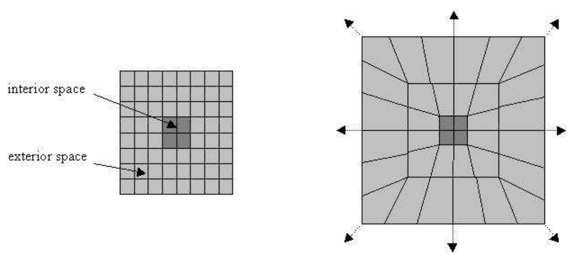
Coarse-to-Fine Voxel Coloring: Octrees

Determine colored voxels at current level
Spatial coherence $\Rightarrow$ add neighboring voxels
Decompose colored voxels into octants; repeat



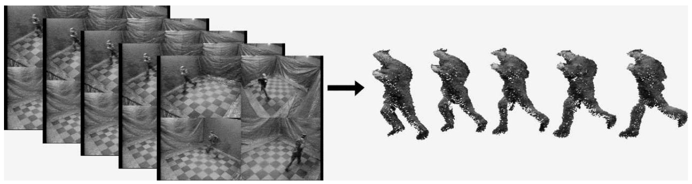
Volumetric Warping

- *G. Slabaugh, T. Malzbender, B. Culbertson, 2000*



Results

Voxel Coloring for Dynamic Scenes

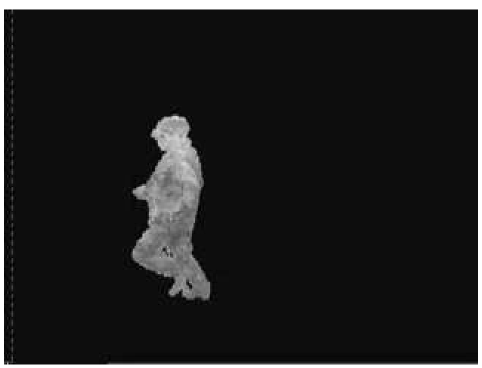


Given: Video sequences from multiple cameras

Goal: Interactive, real-time fly-by of dynamic scene

Dynamic Voxel Coloring: Input Views

Reconstruction for One Time Instant



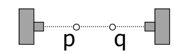
Sequence of Reconstructions

Voxel Coloring for Dynamic Scenes

- ***Coarse-to-fine recursive decomposition focuses on regions of interest***
- ***Exploit temporal coherence***
  - **Use coloring at time t_k to initialize lowest resolution voxels at time t_{k+1}**
  - **Trace rays from changed pixels only**

Limitations of Depth Ordering

A view-independent depth order may not exist:



Need more general algorithm
- **Unconstrained camera positions**
- **Unconstrained scene geometry and topology**

Voxel-based Scene Reconstruction Methods

1. ***Shape from Silhouettes***
   - **Volume intersection** [Martin & Aggarwal, 1983]
2. ***Shape from Photo-Consistency***
   - **Voxel coloring** [Seitz & Dyer, 1997]
   - **Space carving** [Kutulakos & Seitz, 1999]

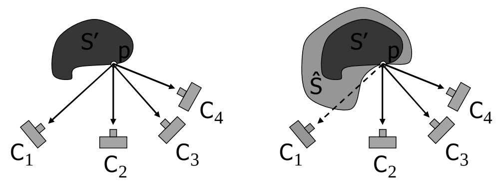
Space Carving Algorithm

Step 1: Initialize V to volume containing true scene with all voxels marked opaque

Step 2: For every voxel on surface of V
- Test *photo-consistency* of voxel with those cameras that are "in front of" it
- If voxel is inconsistent, carve it (i.e., mark it *transparent*)

Step 3: Repeat Step 2 until all voxels consistent

Visibility Property



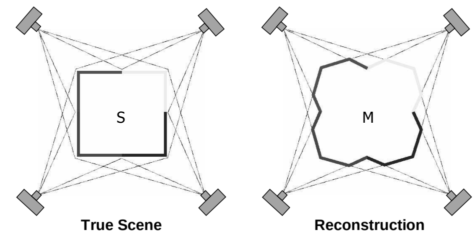
p* $\in$ *S´ consistent* $\Rightarrow$ *p* $\in$ Ŝ *consistent

p* $\in$ Ŝ *inconsistent* $\Rightarrow$ *p* $\in$ *S´ inconsistent

This property ensures that carving converges

Space Carving Convergence

- **Guaranteed convergence to the *photo hull*, i.e., union of all photo-consistent scenes**
- ***Worst case* # consistency checks: (# cameras)^2(# voxels)**



Space Carving Algorithm

Optimal algorithm is unwieldy

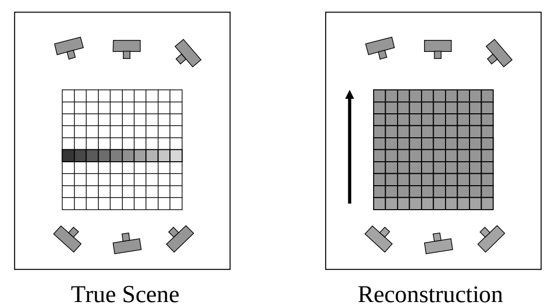
- **Complex visibility update procedure**

Alternative: Multi-Pass Plane Sweep Algorithm

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

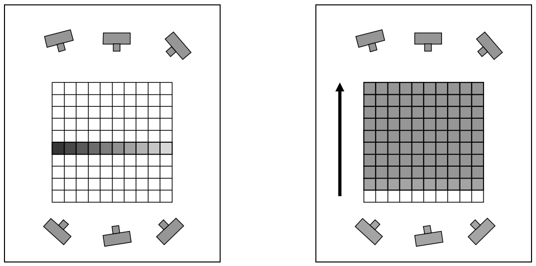
Multi-Pass Plane Sweep

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



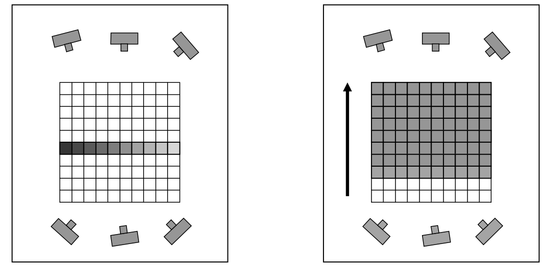
Multi-Pass Plane Sweep

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



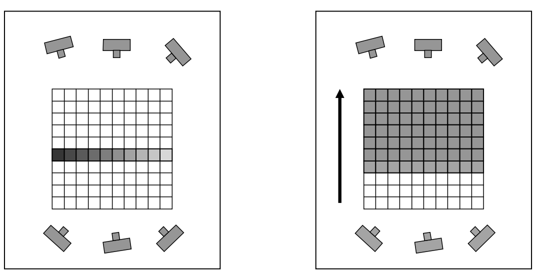
Multi-Pass Plane Sweep

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



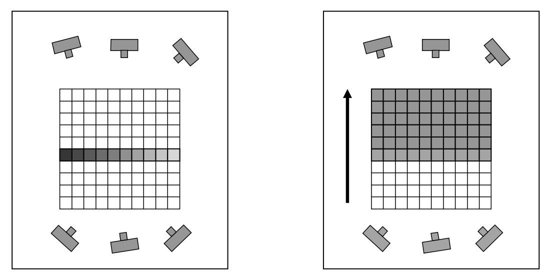
Multi-Pass Plane Sweep

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



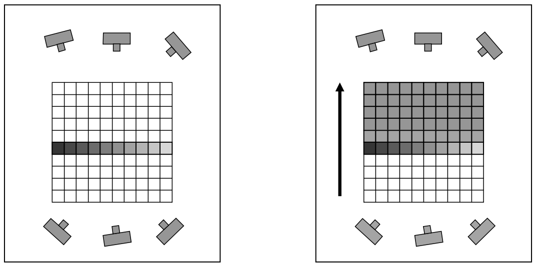
Multi-Pass Plane Sweep

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



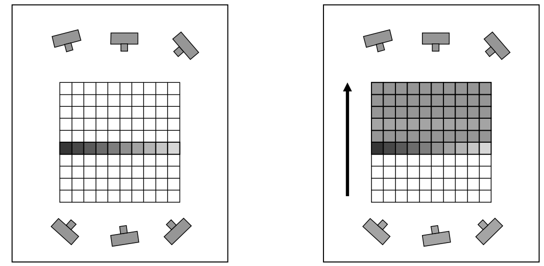
Multi-Pass Plane Sweep

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



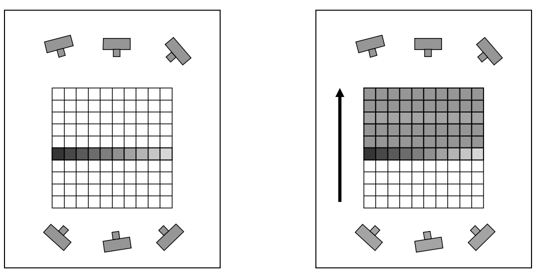
Multi-Pass Plane Sweep

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



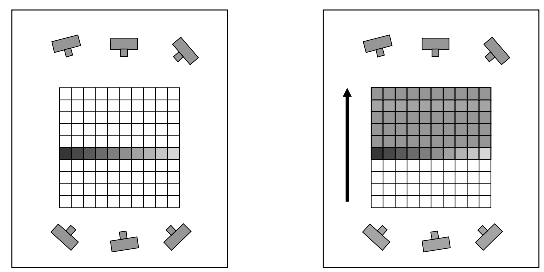
Multi-Pass Plane Sweep

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



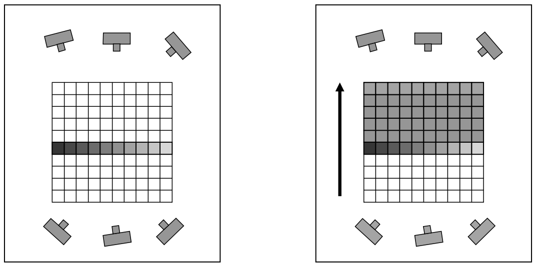
Multi-Pass Plane Sweep

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



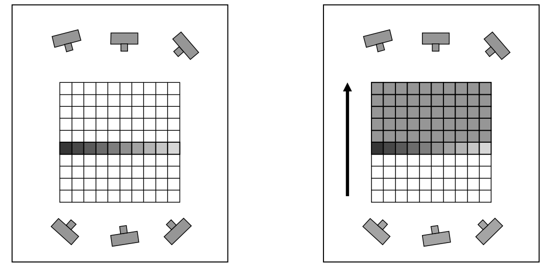
Multi-Pass Plane Sweep

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



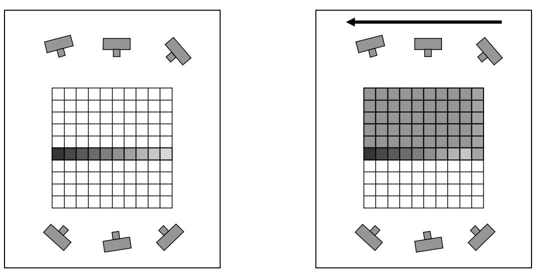
Multi-Pass Plane Sweep

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



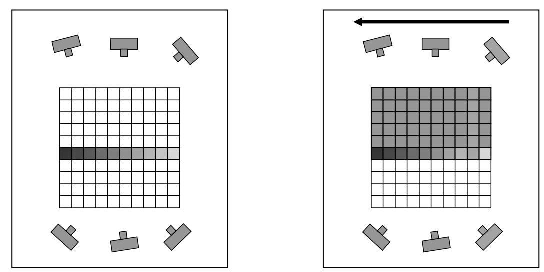
Multi-Pass Plane Sweep

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



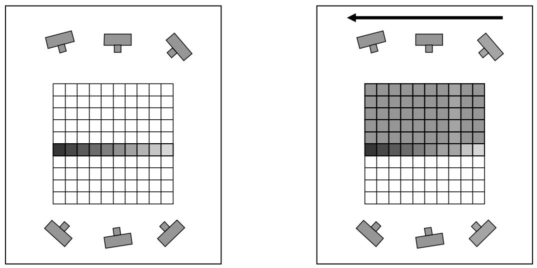
Multi-Pass Plane Sweep

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



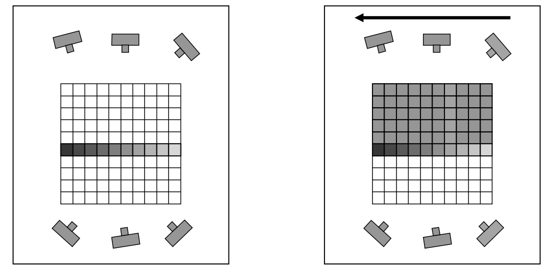
Multi-Pass Plane Sweep

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



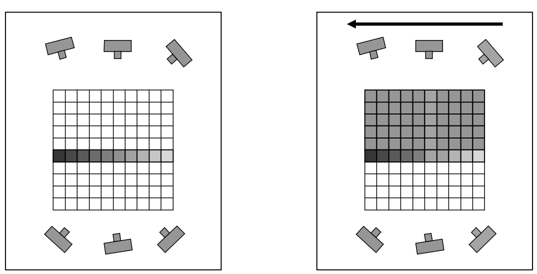
Multi-Pass Plane Sweep

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



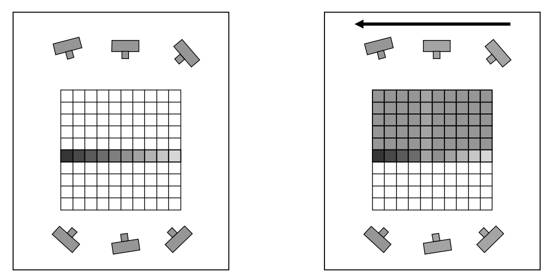
Multi-Pass Plane Sweep

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



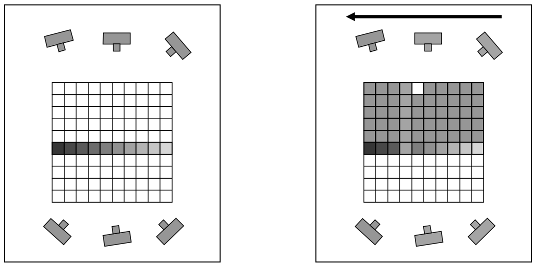
Multi-Pass Plane Sweep

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



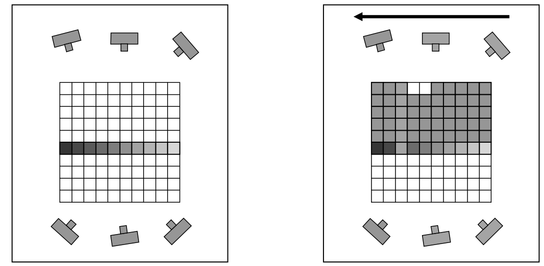
Multi-Pass Plane Sweep

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



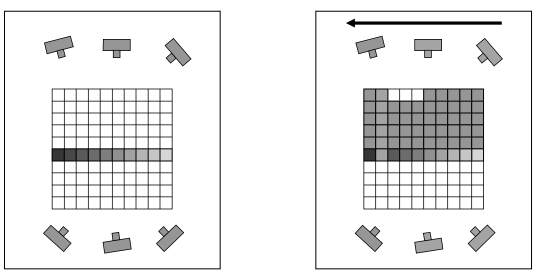
Multi-Pass Plane Sweep

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



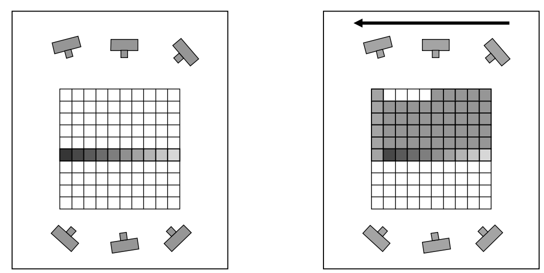
Multi-Pass Plane Sweep

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



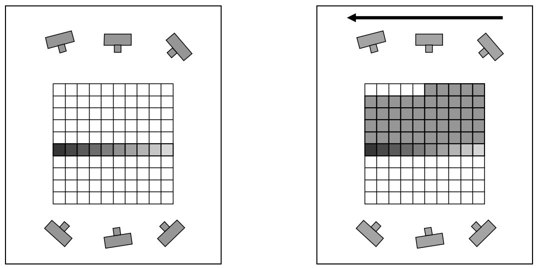
Multi-Pass Plane Sweep

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



Multi-Pass Plane Sweep

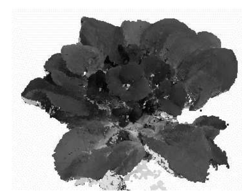
- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence



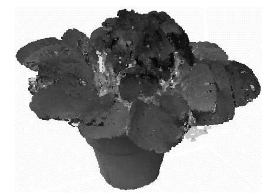
Results: African Violet

Input Image (1 of 45)

Reconstruction



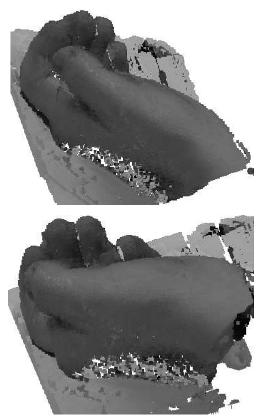
Reconstruction



Reconstruction

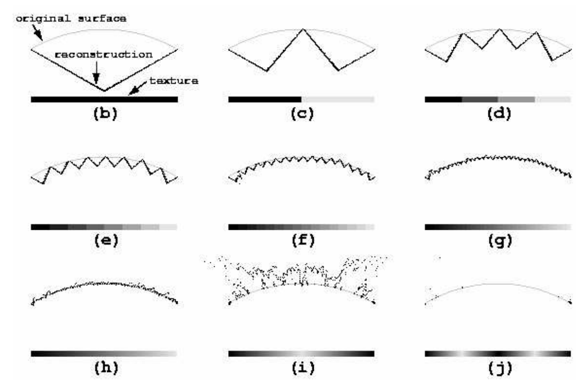
Results: Hand

Input Image
(1 of 100)

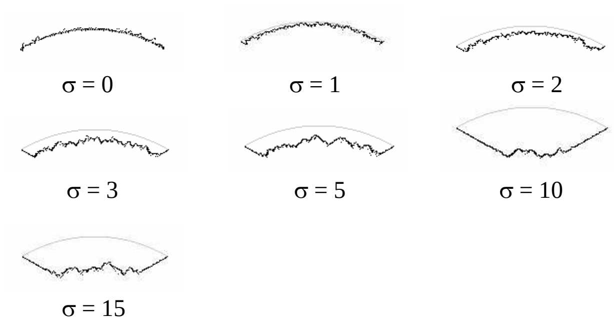


Views of Reconstruction

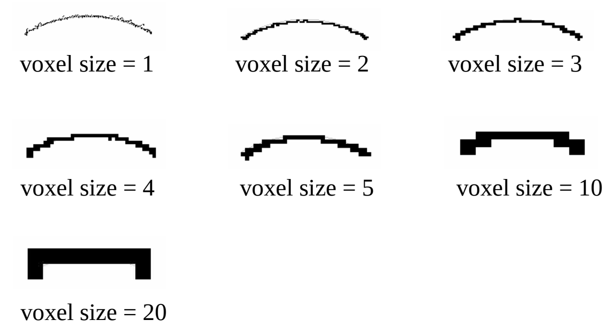
Texture Effects on Voxel Coloring



Effects of Noise



Effects of Voxel Resolution



Other Extensions

- ***Dealing with calibration errors***
  - **Kutulakos, 2000**
  - **Construct approximate photo hull defined by weakening the definition of photo-consistency so that it requires only that there exists a photo-consistent pixel within distance *r* of the ideal position**

- ***Partly transparent scenes***
  - **De Bonet and Viola, 1999**
  - **Compute at each voxel the probability that it is visible (or the degree of opacity)**
  - **Optimization algorithm finds best linear combination of colors and opacities at the voxels along each visual ray to minimize the error with the input image colors**

Voxel Coloring / Space Carving Summary

"The more the marble wastes, the more the statue grows."
– Michelangelo

Pros

- **Non-parametric**
  - Can model arbitrary geometry and topology
- **Camera positions unconstrained**
- **Guaranteed convergence**

Cons

- **Expensive to process high resolution voxel grids**
- **Carving stops at *first* consistent voxel, not *best***
- **Assumes simple, known surface reflectance model, usually Lambertian**

Collaborators

- **Steve Seitz, Andrew Prock, Kyros Kutulakos**

Current Work

- ***BRDF estimation from multiple views***
  - ***Modeling is more than geometry – need to simultaneously recover surface reflectance models***
- **Wide-baseline feature point correspondence**
- **Calibration from multiple moving objects**
- **Metric self-calibration from static scenes**