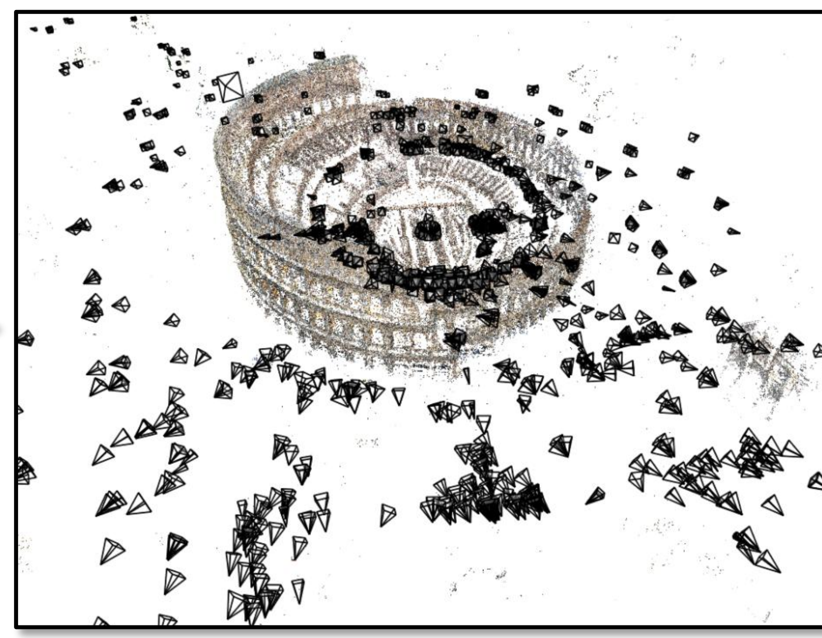
CS6670: Computer Vision

Noah Snavely

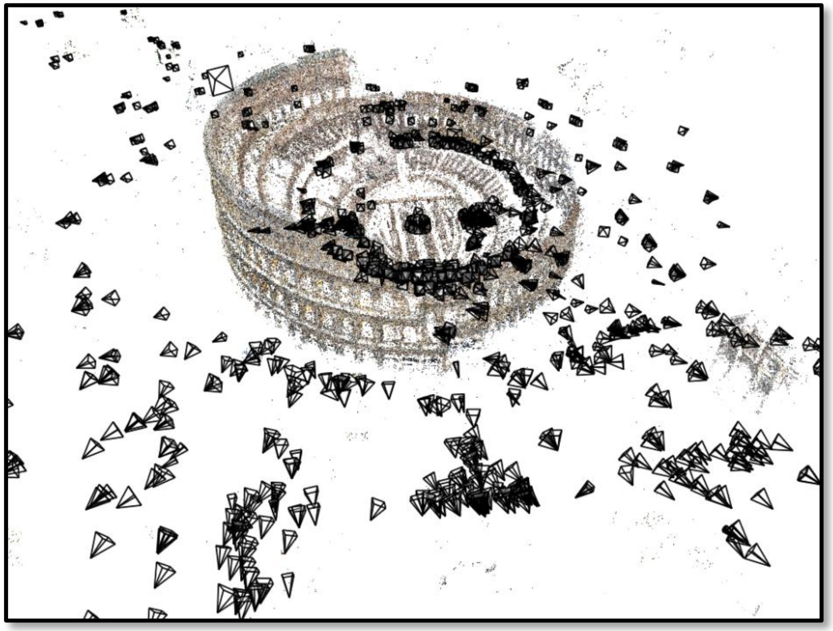
Lecture 12: Structure from motion



CS6670: Computer Vision

Noah Snavely

Lecture 13: Multi-view stereo



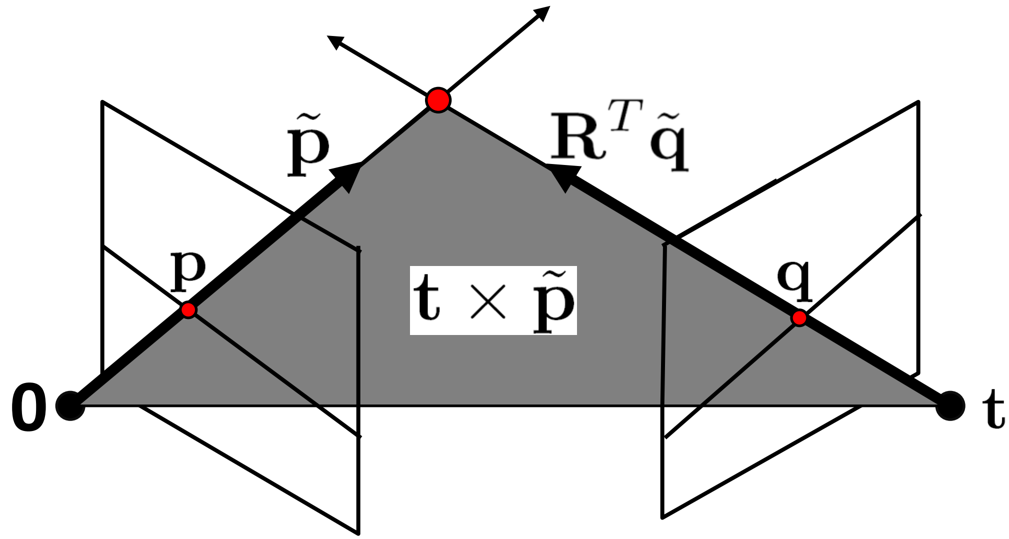
Announcements

- Project 2 voting open later today
- Final project page will be released after class
- Project 3 out soon
- Quiz 2 on Thursday, beginning of class

Readings

- Szeliski, Chapter 11.6

Fundamental matrix – calibrated case

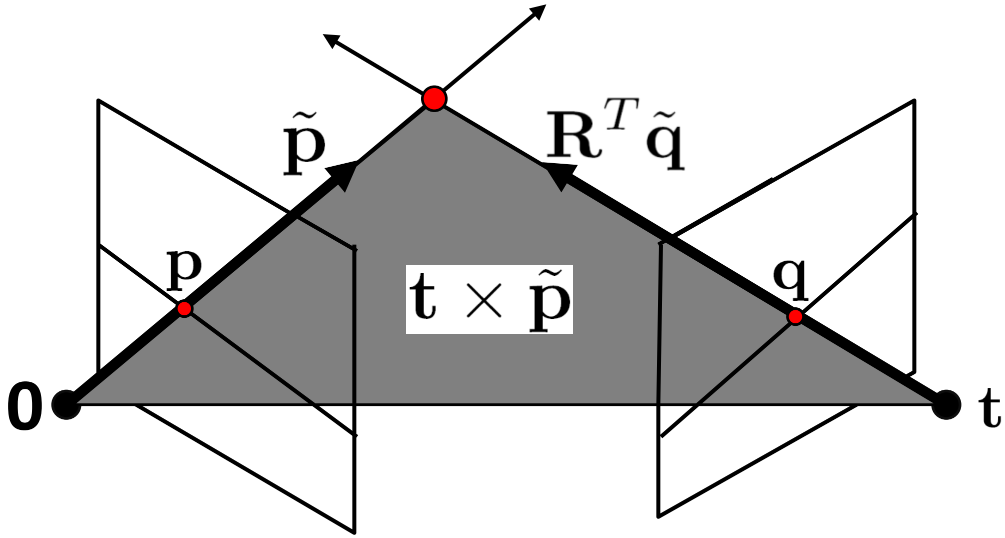


$$\underbrace{\tilde{\mathbf{q}}^T \mathbf{R}\,[\mathbf{t}]_\times}_{\mathbf{E}}\,\tilde{\mathbf{p}} = 0$$

$$\tilde{\mathbf{q}}^T \mathbf{E} \tilde{\mathbf{p}} = 0$$

$\mathbf{E}$ ← the Essential matrix

Fundamental matrix – uncalibrated case



$$\tilde{\mathbf{q}}^T \mathbf{R} \left[\mathbf{t}\right]_\times \tilde{\mathbf{p}} = 0$$

$$\mathbf{q}^T \underbrace{\mathbf{K}_2^{-T} \mathbf{R} \left[\mathbf{t}\right]_\times \mathbf{K}_1^{-1}}_{\mathbf{F}} \mathbf{p} = 0$$

$\mathbf{F}$ ← the Fundamental matrix

Properties of the Fundamental Matrix

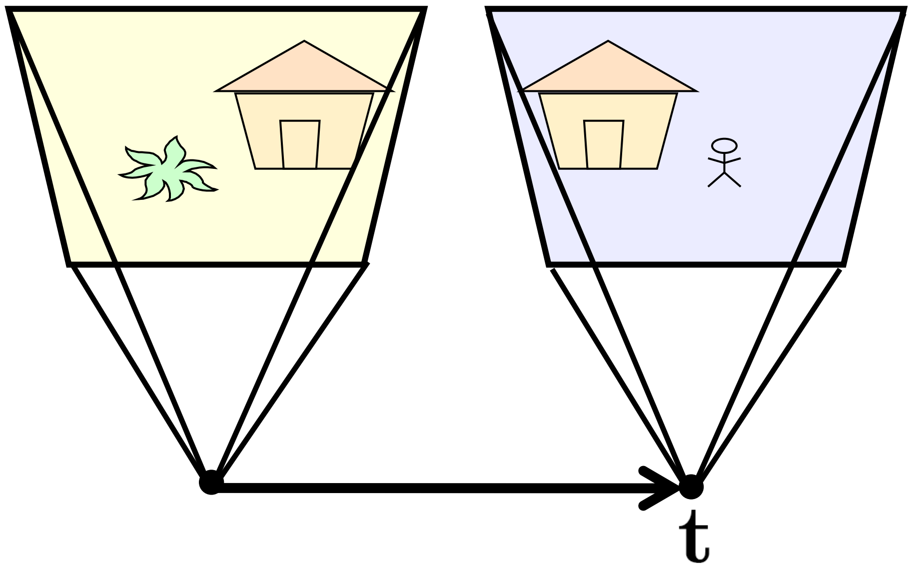
- $\mathbf{F}\mathbf{p}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T\mathbf{q}$ is the epipolar line associated with $\mathbf{q}$
- $\mathbf{F}\mathbf{e}_1 = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}_2 = \mathbf{0}$
- $\mathbf{F}$ is rank 2
- How many parameters does **F** have?

How many parameters?

- Matrix has 9 entries
  - -1 due to scale invariance $\alpha\mathbf{F} \sim \mathbf{F}$
  - -1 due to rank 2 constraint

- 7 parameters in total

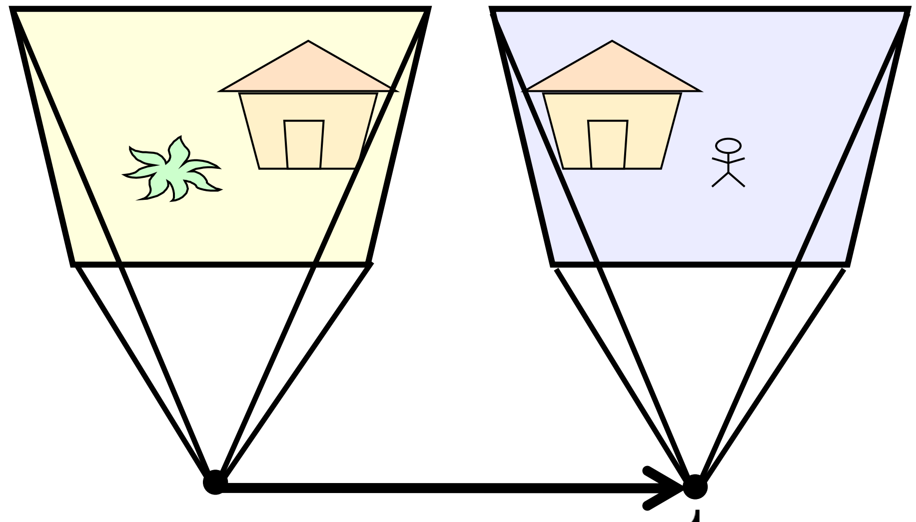
Rectified case



$$\mathbf{R} = \mathbf{I}_{3\times 3}$$
$$\mathbf{t} = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array}\right]^T$$

$$\mathbf{E} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{array}\right]$$

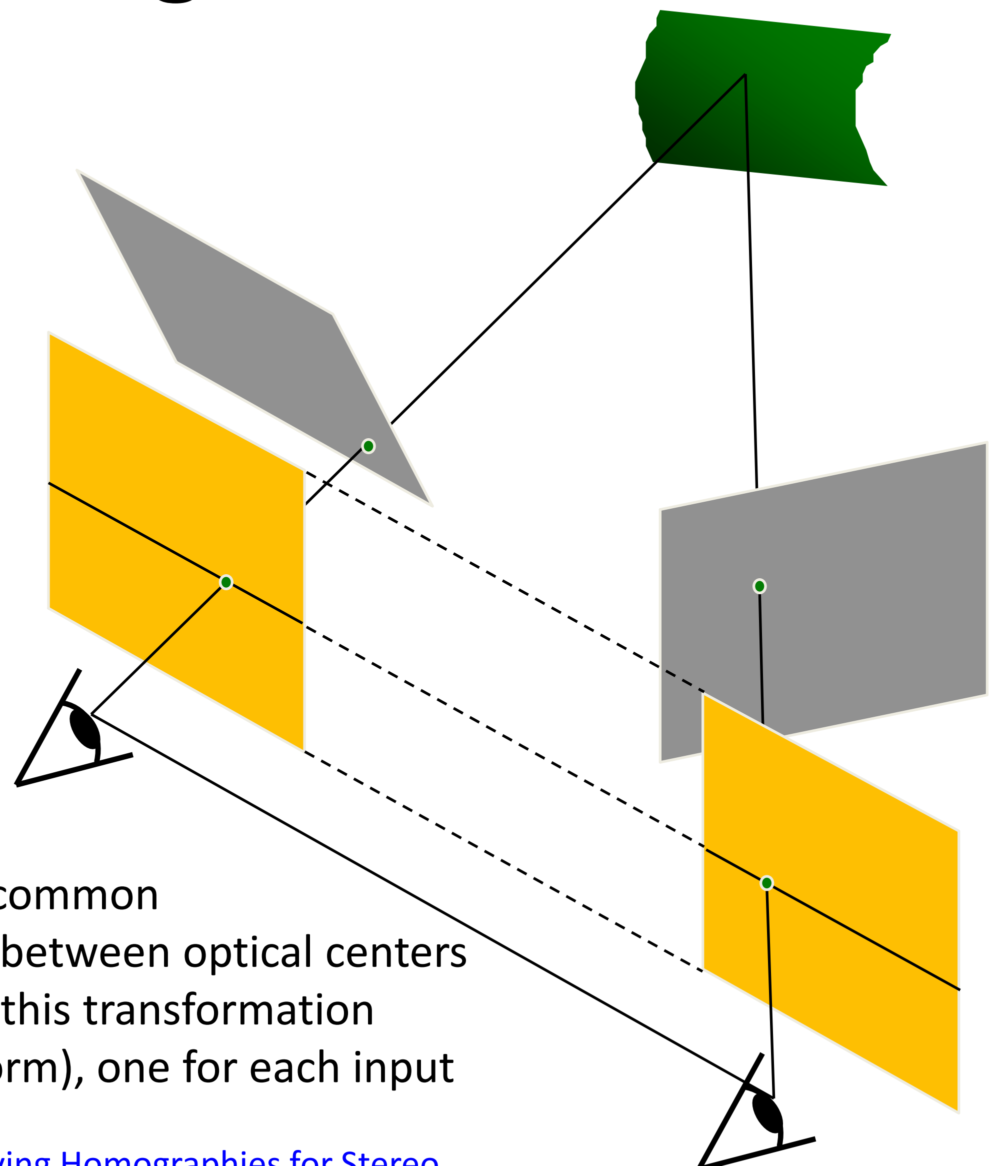
Rectified case



if $\mathbf{K}_1 = \mathbf{K}_2$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Stereo image rectification



- reproject image planes onto a common
- plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transform), one for each input image reprojection

➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectifying homographies

- Idea: compute two homographies

  $\mathbf{H}_1$ and $\mathbf{H}_2$

  such that

$$\mathbf{H}_2^{-T}\mathbf{F}\mathbf{H}_1^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Rectifying homographies

$$\mathbf{H}_2^{-T}\mathbf{F}\mathbf{H}_1^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{q}^T\mathbf{F}\mathbf{p} = 0$$

$$\mathbf{q}^T\mathbf{H}_2^T\mathbf{H}_2^{-T}\mathbf{F}\mathbf{H}_1^{-1}\mathbf{H}_1\mathbf{p} = 0$$

Estimating **F**

- If we don't know $\mathbf{K}_1$, $\mathbf{K}_2$, $\mathbf{R}$, or $\mathbf{t}$, can we estimate $\mathbf{F}$?

- Yes, given enough correspondences

Estimating F – 8-point algorithm

- The fundamental matrix F is defined by

$$\mathbf{q}^T \mathbf{F} \mathbf{p} = 0$$

for any pair of matches **p** and **q** in two images.

- Let **p**=$(u,v,1)^T$ and **q**=$(u',v',1)^T$,

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

each match gives a linear equation

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

- In reality, instead of solving $\mathbf{Af} = 0$, we seek unit vector $\mathbf{f}$ that minimizes $\|\mathbf{Af}\|^2$
  →least eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$
  - need at least 8-correspondences

8-point algorithm

- To enforce that **F** is rank 2, we replace **F** with **F**' that minimizes $\|\mathbf{F}-\mathbf{F}'\|$ subject to the rank constraint.

- This is achieved by SVD. Let $\mathbf{F}=\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, where

$$\Sigma=\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & \sigma_3\end{bmatrix}\text{, let }\Sigma'=\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & 0\end{bmatrix}$$

then $\mathbf{F}'=\mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}$ is the solution.

8-point algorithm

```
% Build the constraint matrix
   A = [x2(1,:)‘.*x1(1,:)'   x2(1,:)'.*x1(2,:)'  x2(1,:)' ...
        x2(2,:)'.*x1(1,:)'   x2(2,:)'.*x1(2,:)'  x2(2,:)' ...
        x1(1,:)'             x1(2,:)'            ones(npts,1) ];

   [U,D,V] = svd(A);

% Extract fundamental matrix from the column of V
% corresponding to the smallest singular value.
   F = reshape(V(:,9),3,3)';

% Enforce rank2 constraint
   [U,D,V] = svd(F);
   F = U*diag([D(1,1) D(2,2) 0])*V';
```

8-point algorithm

- Pros:
  - linear, easy to implement and fast
- Cons:
  - minimizes an *algebraic*, rather than geometric error
  - susceptible to noise

Problem with 8-point algorithm

$$
\begin{bmatrix}
u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\
u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1
\end{bmatrix}
\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0
$$

~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

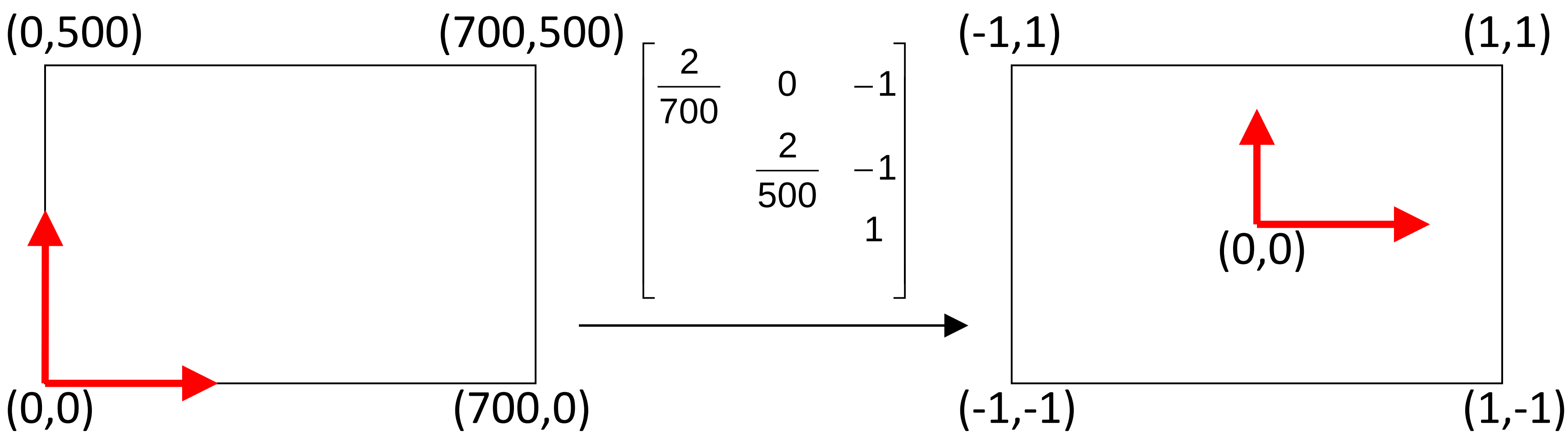
!

Orders of magnitude difference
between column of data matrix
$\rightarrow$ least-squares yields poor results

Normalized 8-point algorithm

normalized least squares yields good results

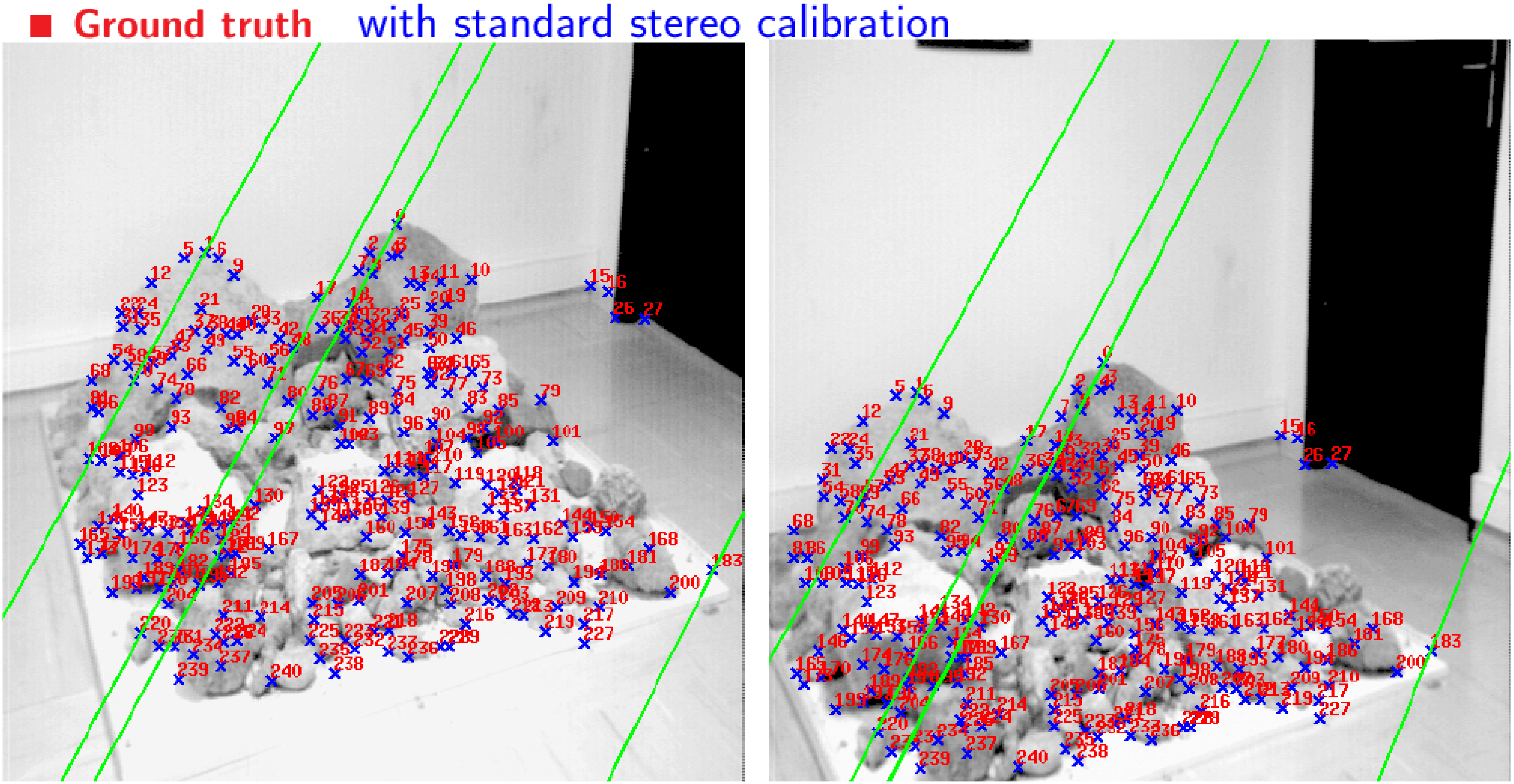
Transform image to ~[-1,1]x[-1,1]



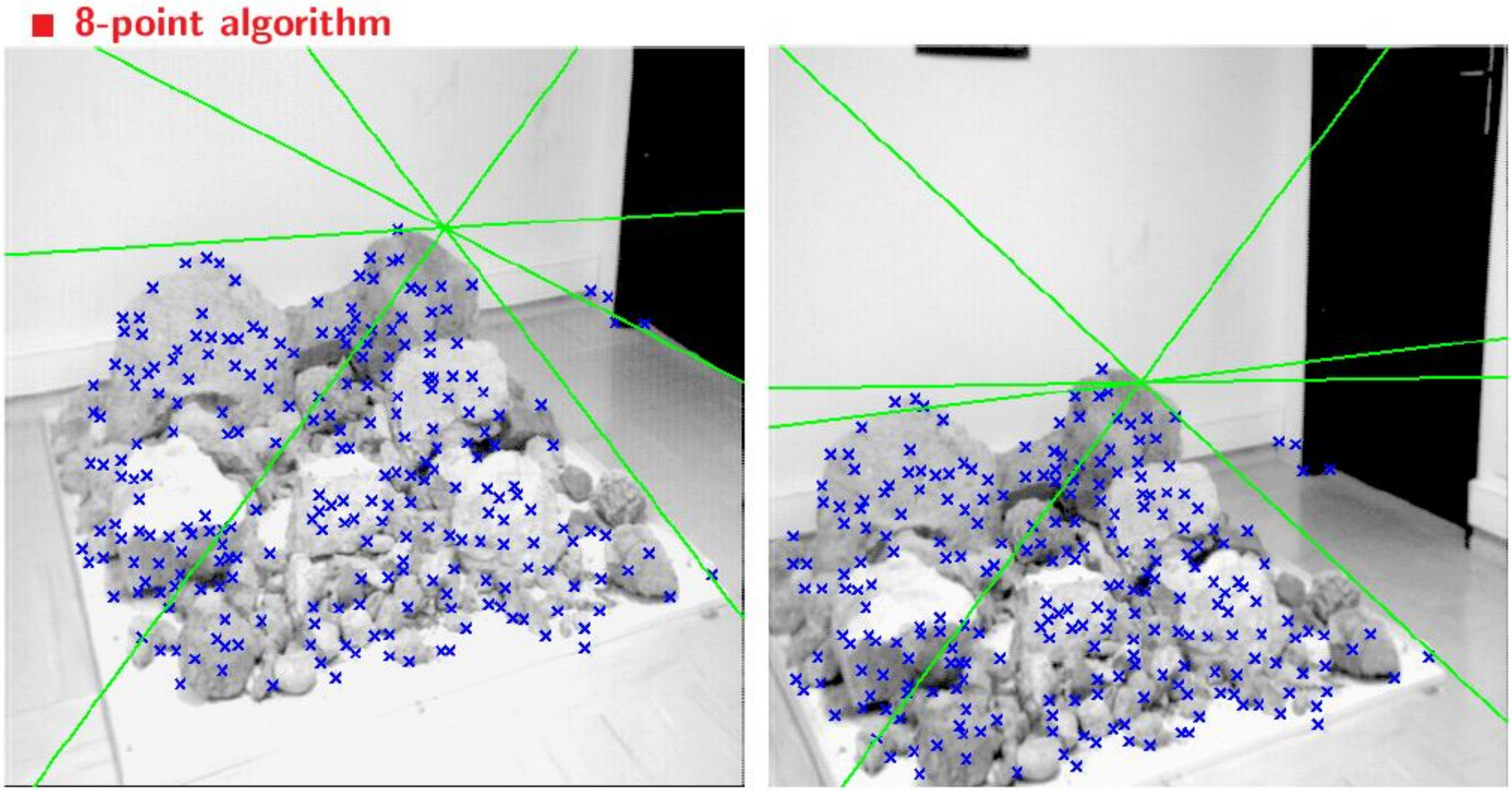
Normalized 8-point algorithm

```
[x1, T1] = normalise2dpts(x1);
[x2, T2] = normalise2dpts(x2);

A = [x2(1,:)‘.*x1(1,:)'   x2(1,:)'.*x1(2,:)'  x2(1,:)' ...
     x2(2,:)'.*x1(1,:)'   x2(2,:)'.*x1(2,:)'  x2(2,:)' ...
     x1(1,:)'             x1(2,:)'            ones(npts,1) ];

[U,D,V] = svd(A);

F = reshape(V(:,9),3,3)';

[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';

% Denormalise
F = T2'*F*T1;
```

Results (ground truth)

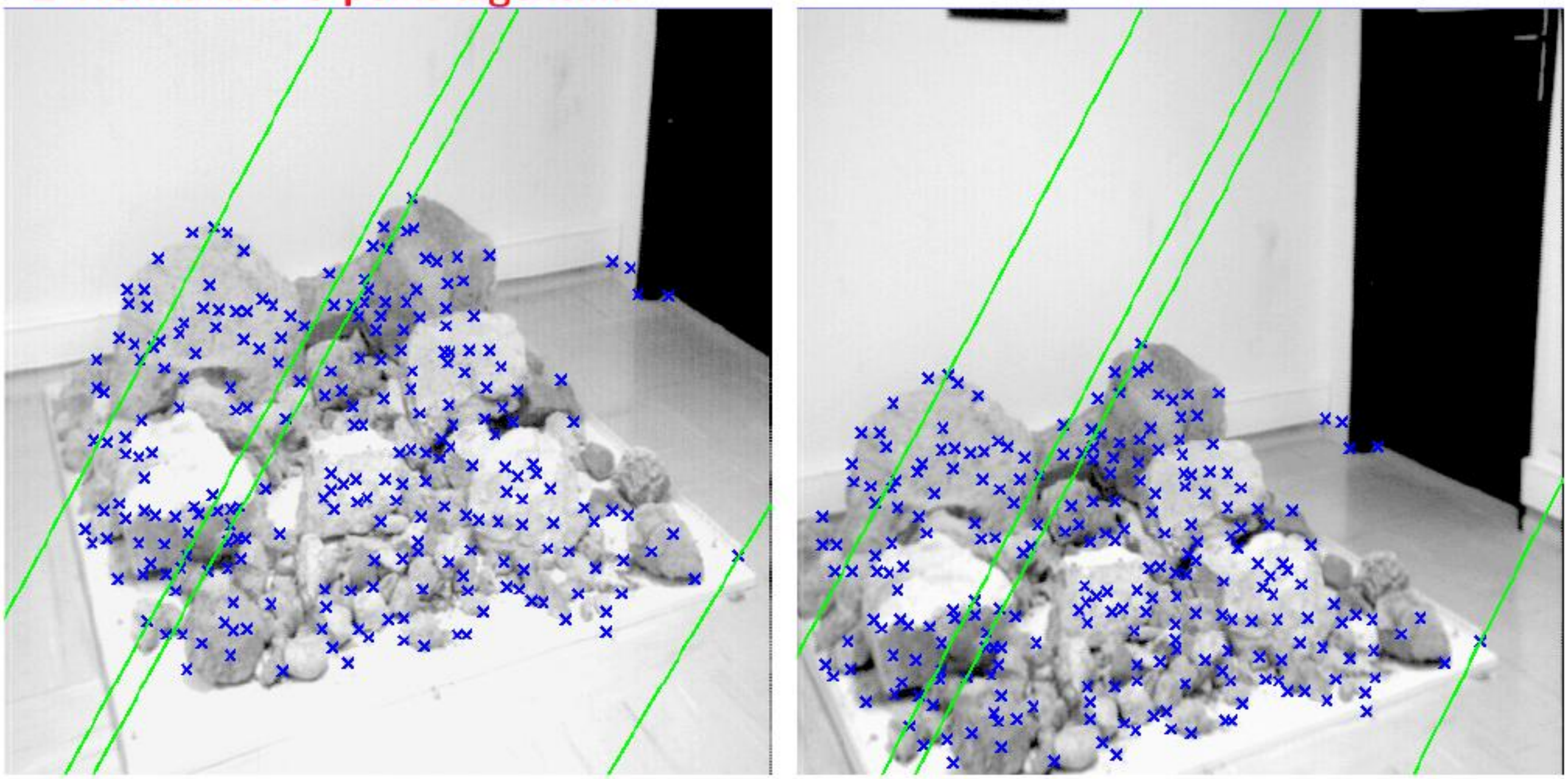


Results (8-point algorithm)



Results (normalized 8-point algorithm)

■ **Normalized 8-point algorithm**



What about more than two views?

- The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*
- The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*
- After this it starts to get complicated…
- No known closed-form solution to the general structure from motion problem

Questions?

Multi-view stereo

Stereo

Multi-view stereo

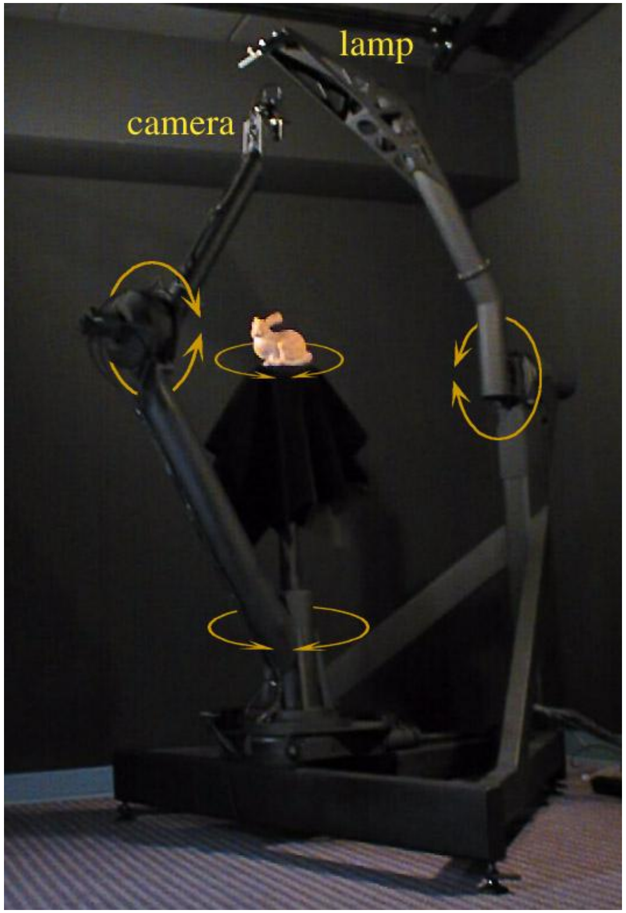
Multi-view Stereo

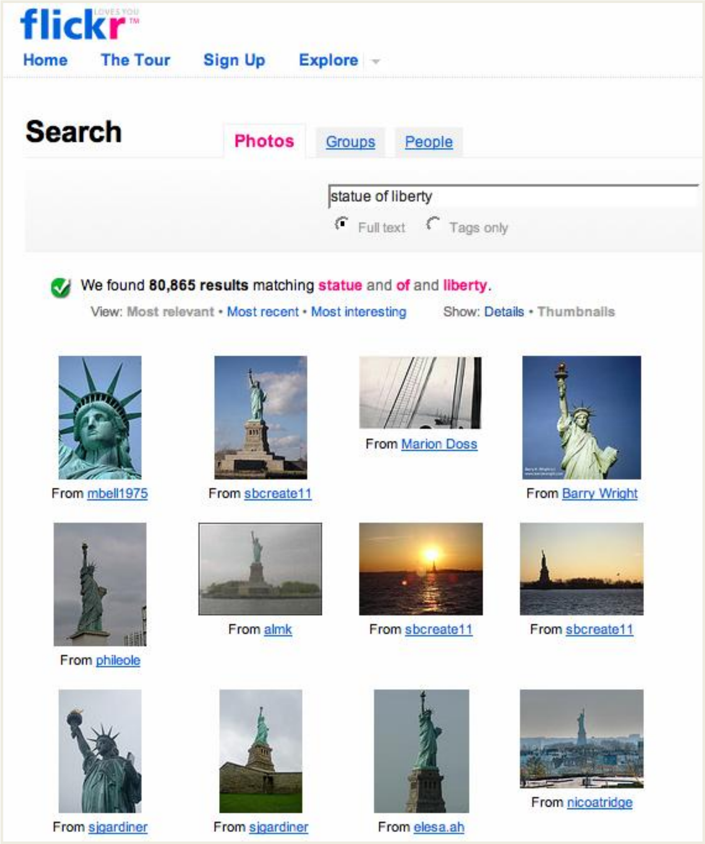
Point Grey's Bumblebee XB3

Point Grey's ProFusion 25

CMU's 3D Room

Multi-view Stereo

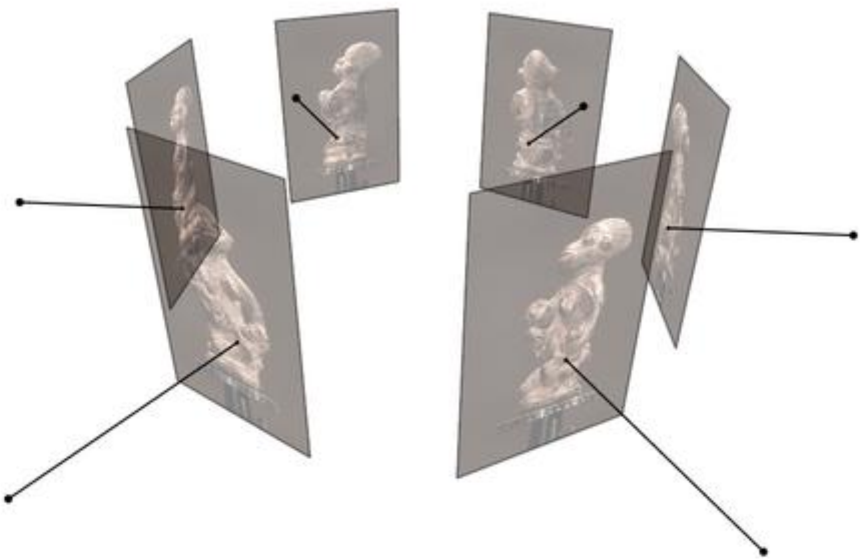




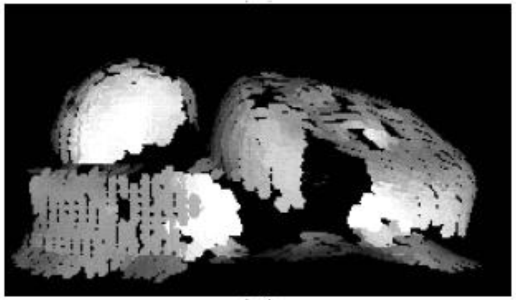
Multi-view Stereo

Input: calibrated images from several viewpoints

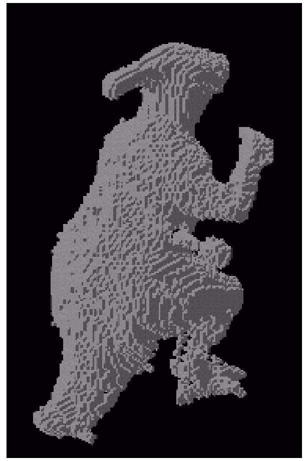
Output: 3D object model



Figures by Carlos Hernandez



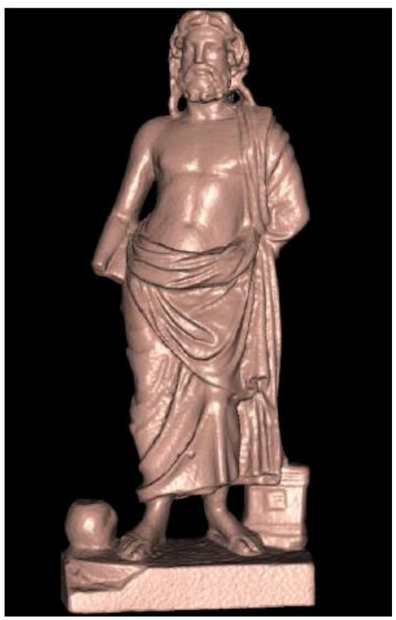
Fua
1995



Seitz, Dyer
1997

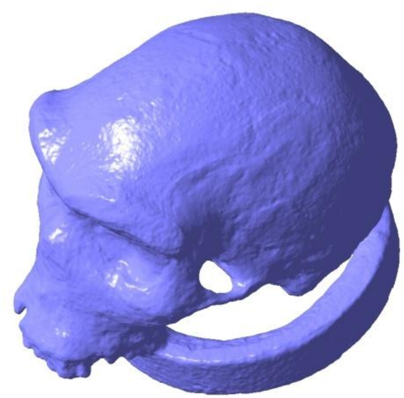
Narayanan, Rander, Kanade
1998

Faugeras, Keriven
1998



Hernandez, Schmitt
2004

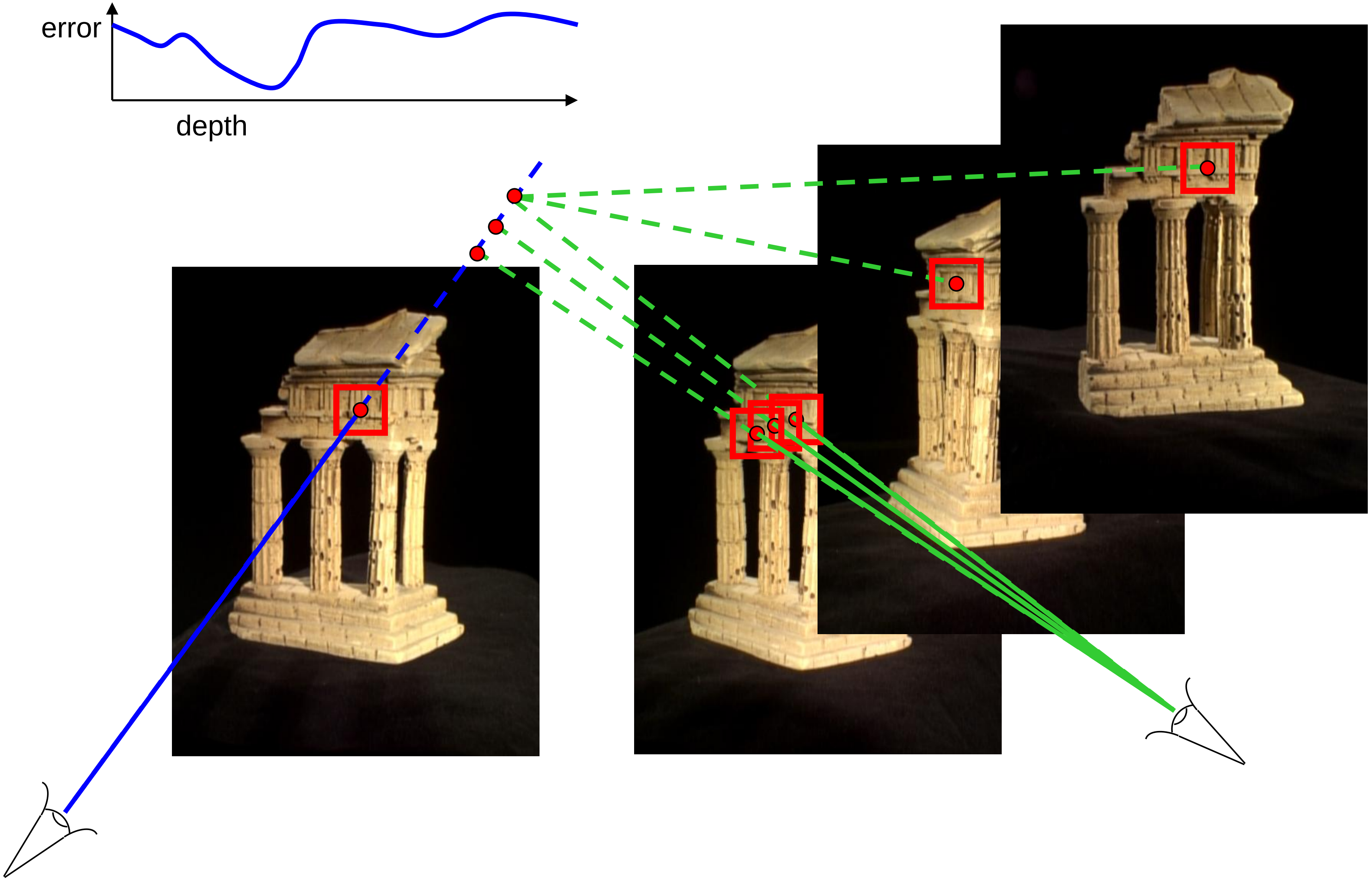
Pons, Keriven, Faugeras
2005



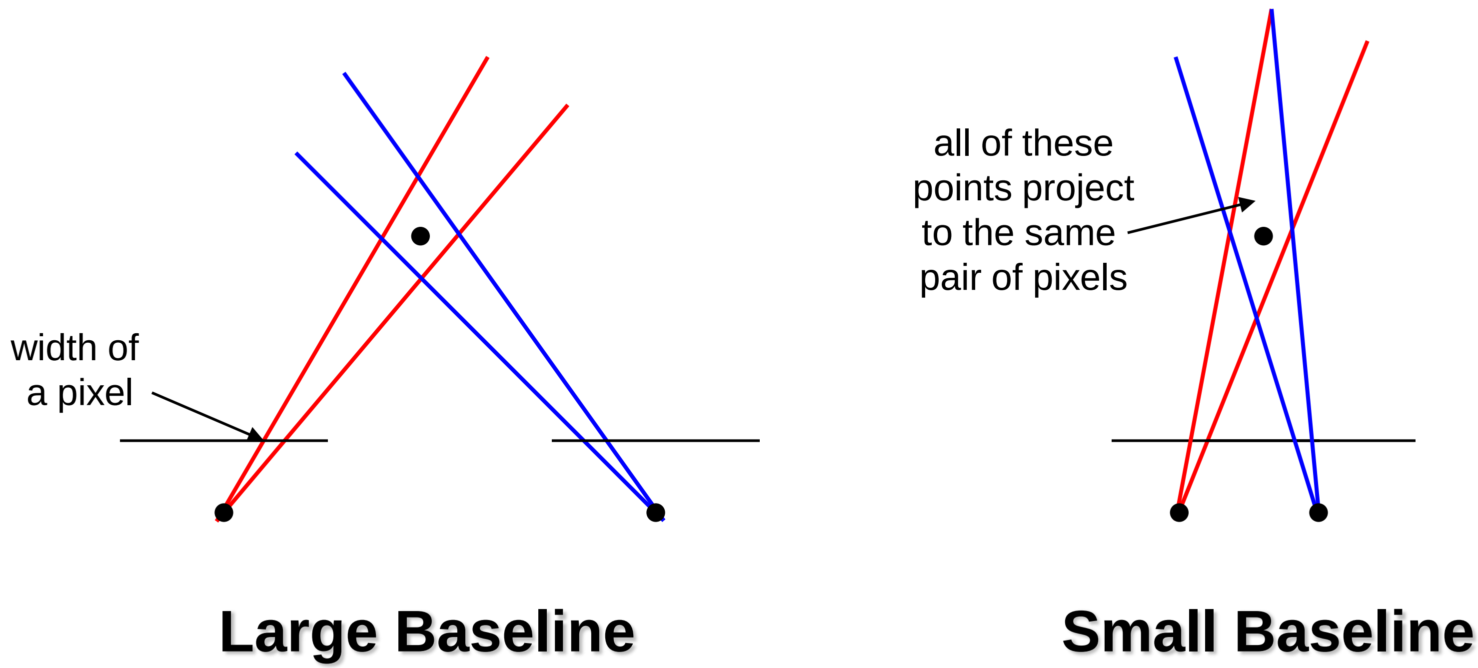
Furukawa, Ponce
2006

Goesele et al.
2007

Stereo: basic idea



Choosing the stereo baseline

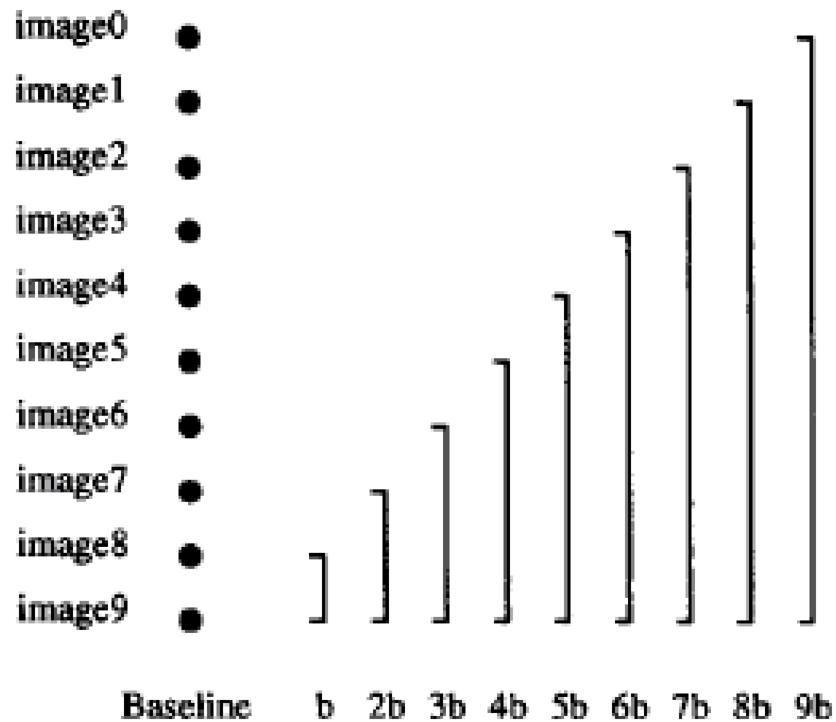


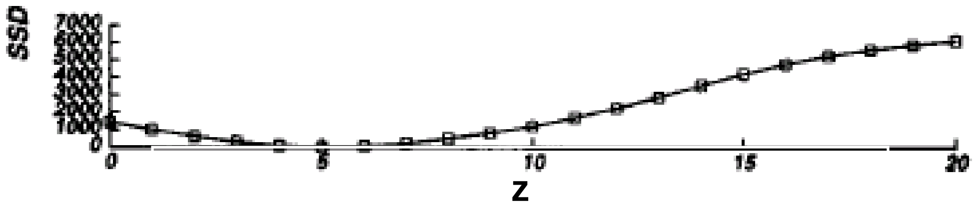
What's the optimal baseline?

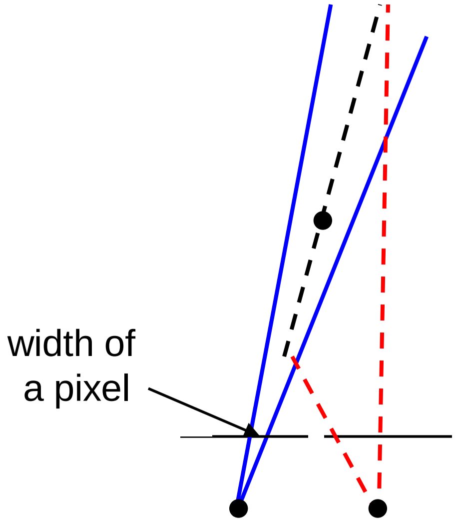
- Too small: large depth error
- Too large: difficult search problem

The Effect of Baseline on Depth Estimation

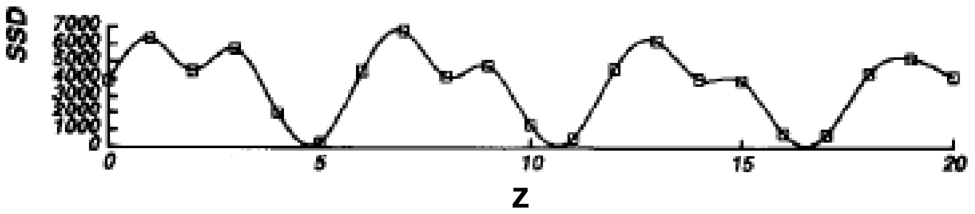
Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.

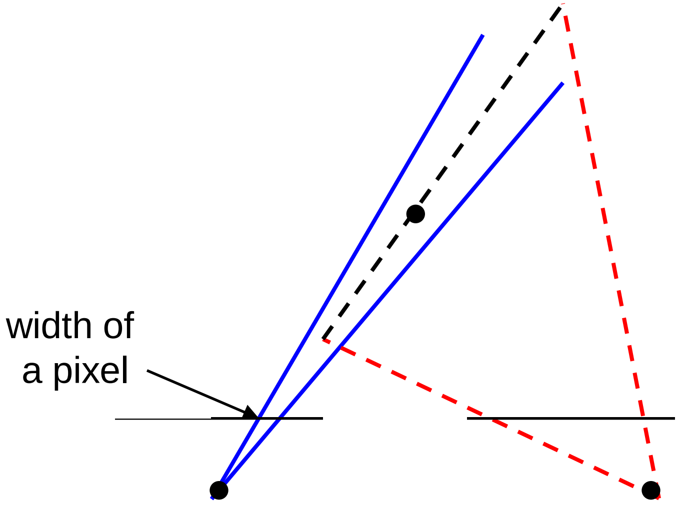






pixel matching score





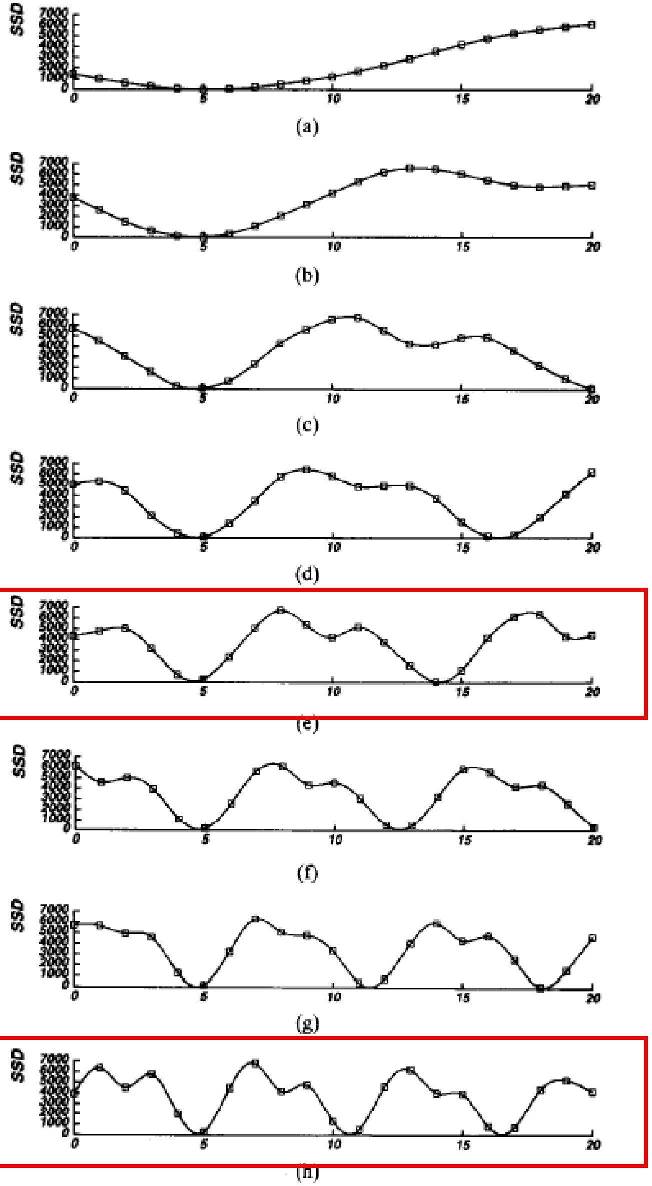


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

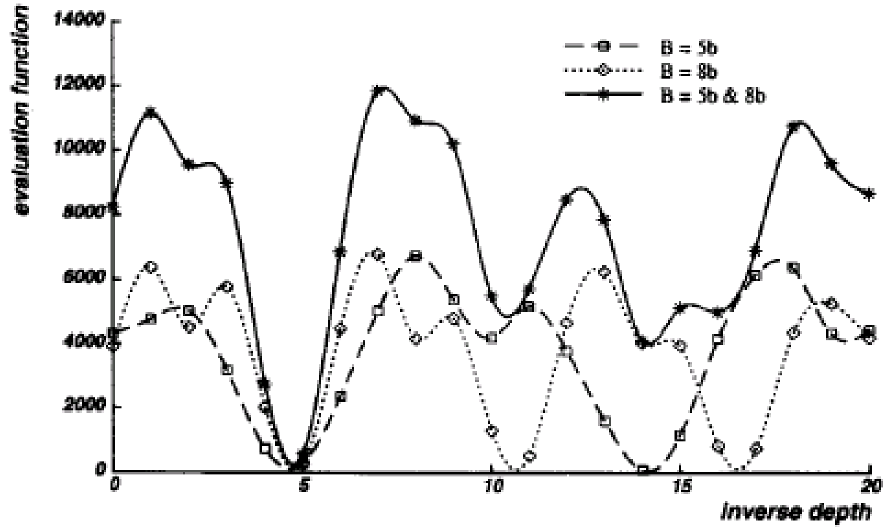


Fig. 6. Combining two stereo pairs with different baselines.

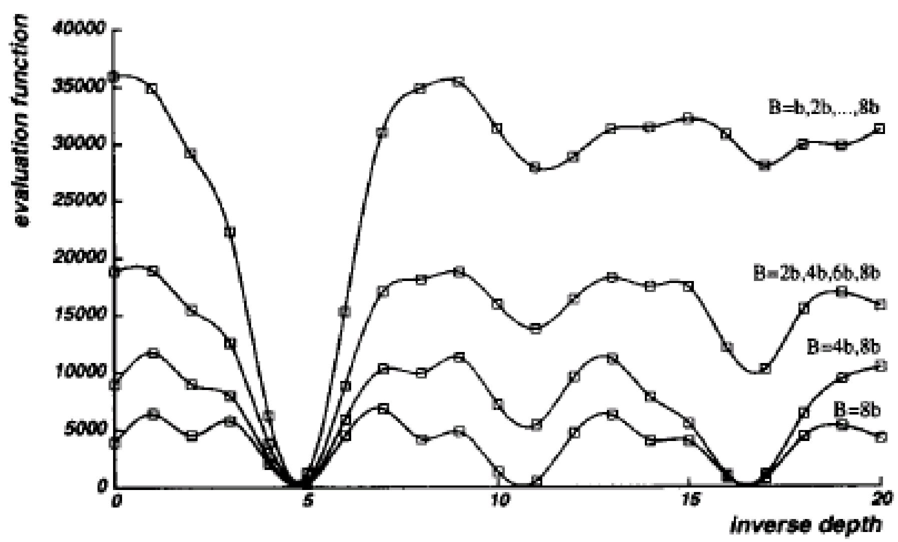


Fig. 7. Combining multiple baseline stereo pairs.

Multibaseline Stereo

Basic Approach

- Choose a reference view
- Use your favorite stereo algorithm BUT
  - replace two-view SSD with SSSD over all baselines

Limitations

Problem: *visibility*

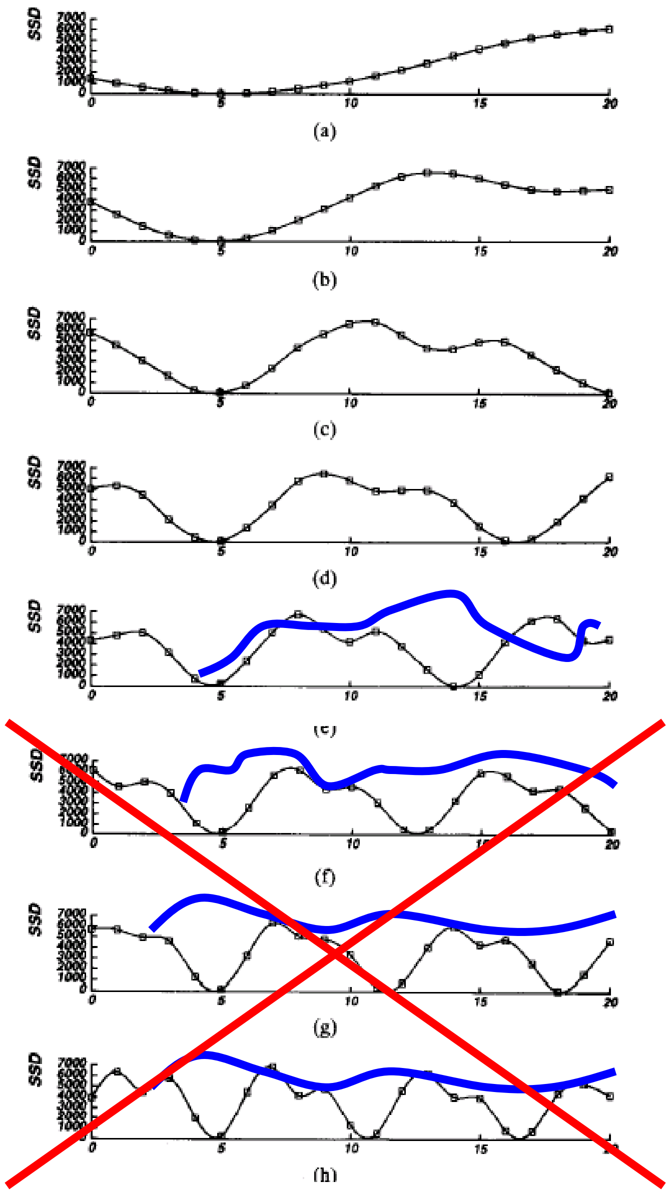


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

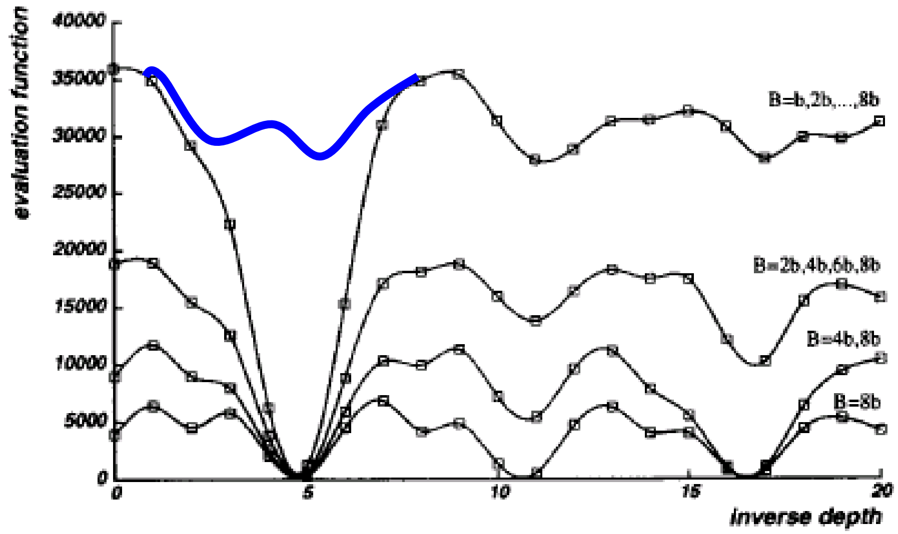


Fig. 7. Combining multiple baseline stereo pairs.

Some Solutions

- Match only nearby photos [Narayanan 98]
- Use NCC instead of SSD,
  Ignore NCC values > threshold
  [Hernandez & Schmitt 03]

Popular matching scores

- SSD (Sum Squared Distance)

$$\sum_{x,y} |W_1(x,y) - W_2(x,y)|^2$$

- NCC (Normalized Cross Correlation)

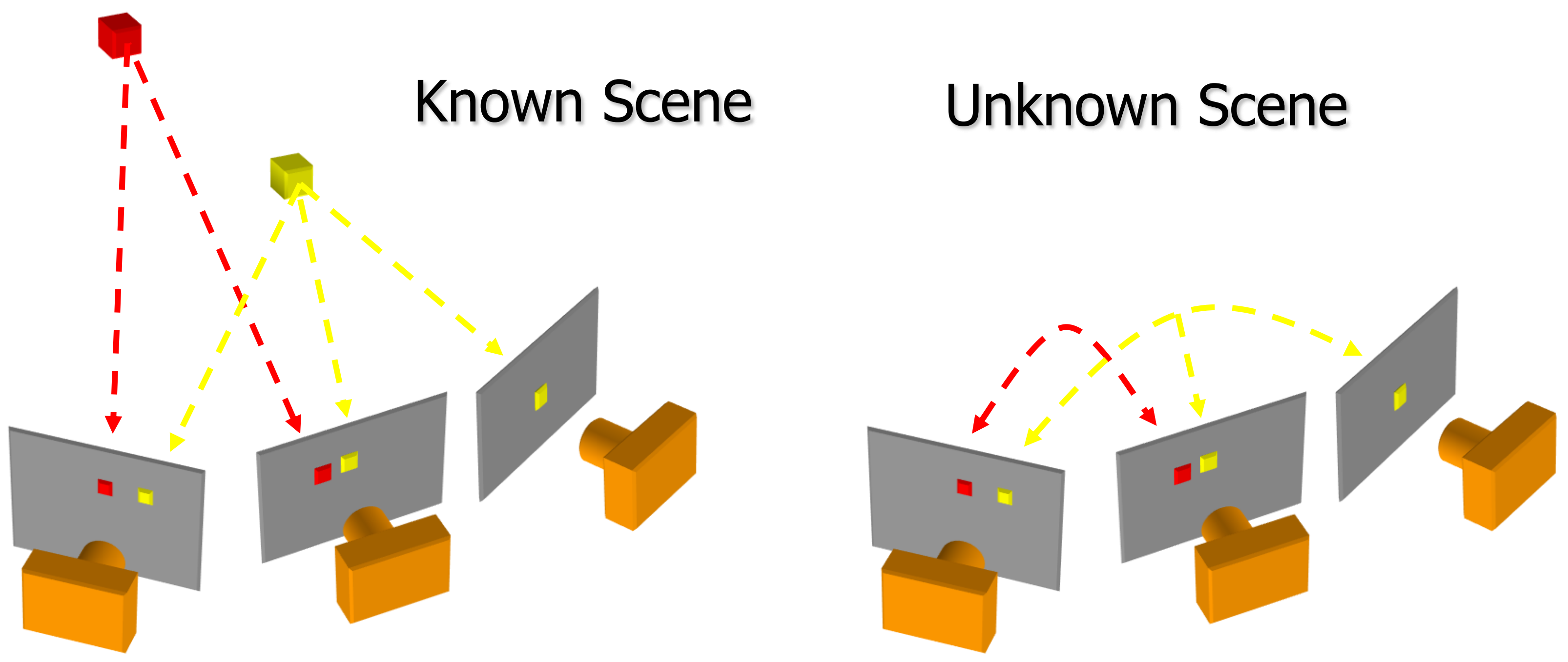
$$\frac{\sum_{x,y}(W_1(x,y) - \overline{W_1})(W_2(x,y) - \overline{W_2})}{\sigma_{W_1}\sigma_{W_2}}$$

  – where $\overline{W_i} = \frac{1}{n}\sum_{x,y} W_i \qquad \sigma_{W_i} = \sqrt{\frac{1}{n}\sum_{x,y}(W_i - \overline{W_i})^2}$

  – what advantages might NCC have?

The visibility problem

Which points are visible in which images?



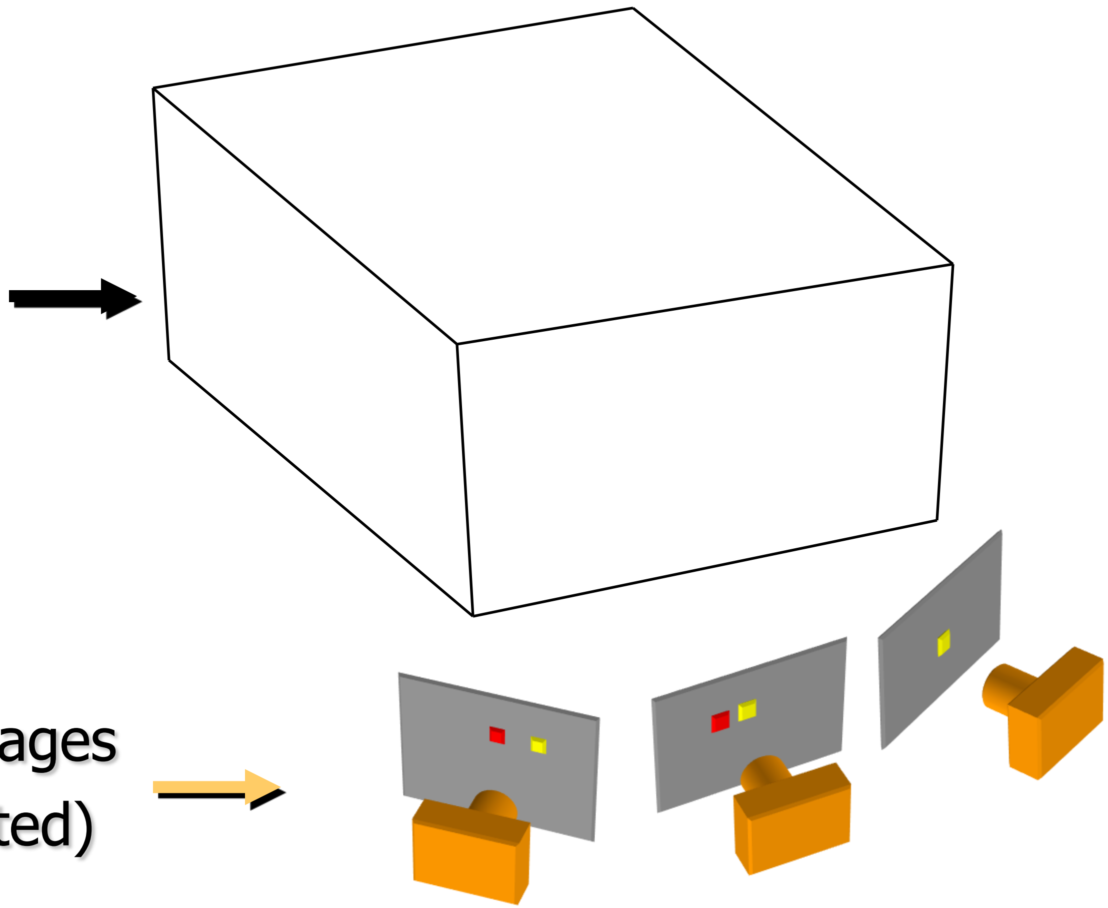
Forward Visibility

Inverse Visibility

Volumetric stereo

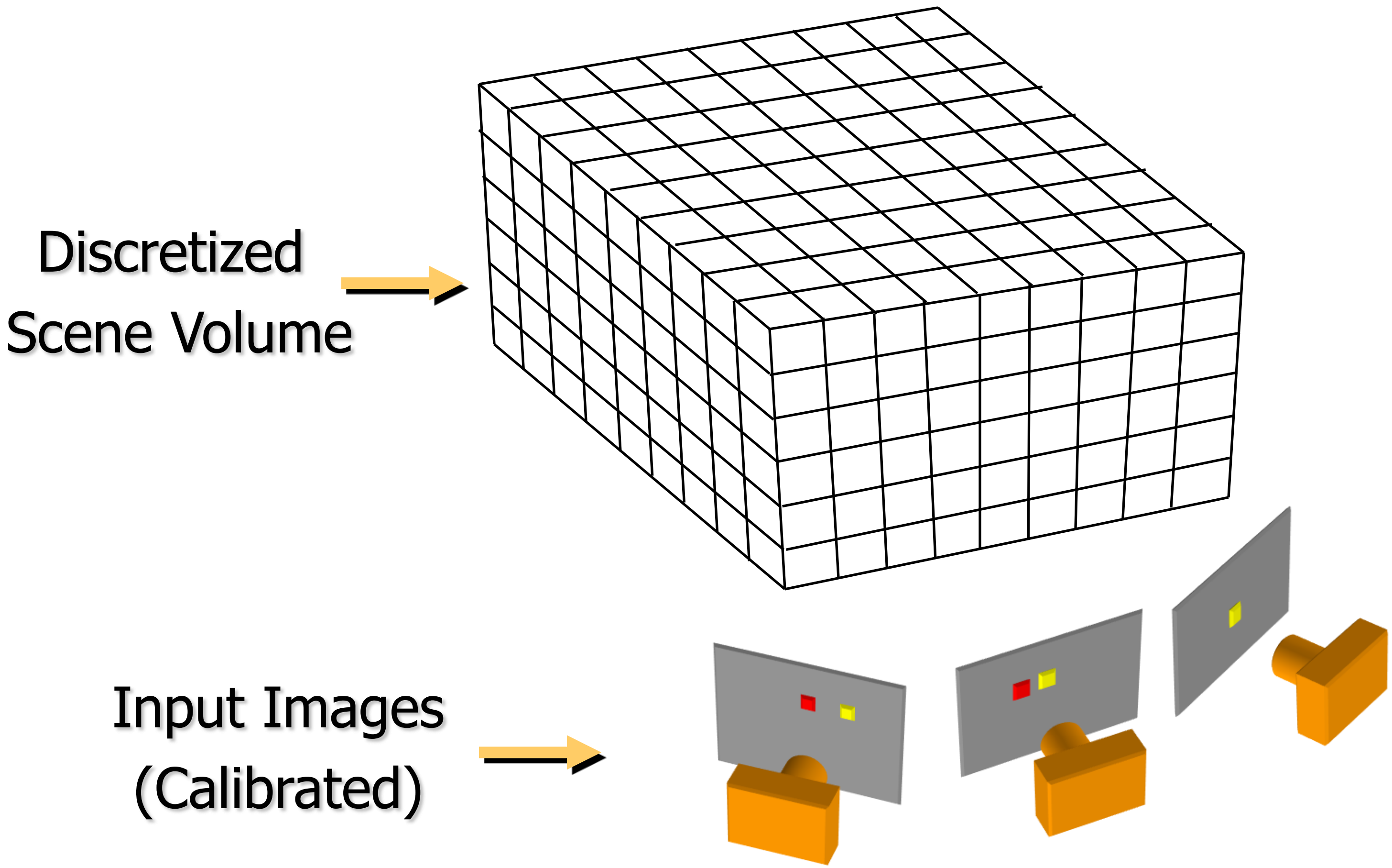
Scene Volume
V

Input Images
(Calibrated)



Goal: **Determine occupancy, "color" of points in V**

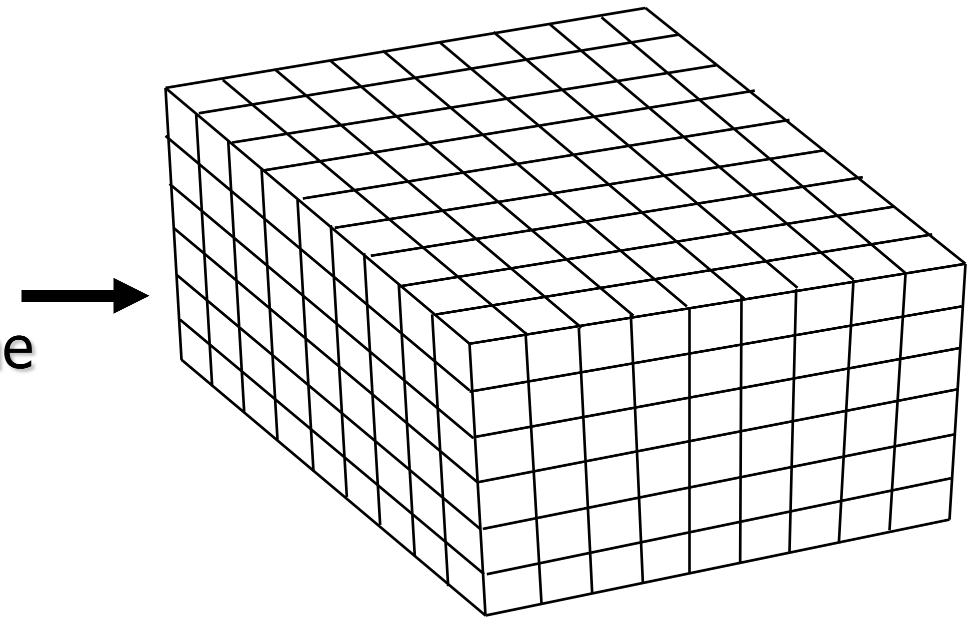
Discrete formulation: Voxel Coloring

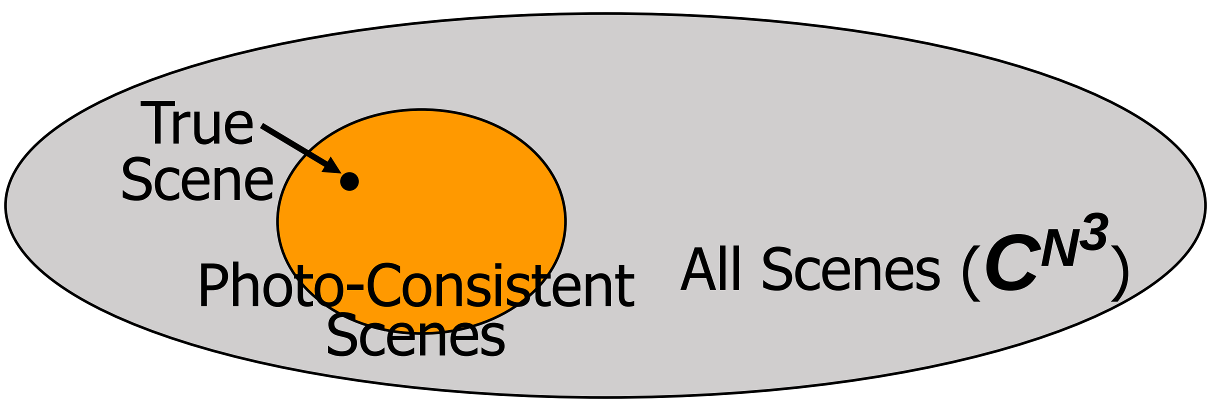


Goal: **Assign RGBA values to voxels in V**
photo-consistent **with images**

Complexity and computability

Discretized
Scene Volume
N^3 voxels
C colors





Issues

Theoretical Questions

- Identify class of *all* photo-consistent scenes

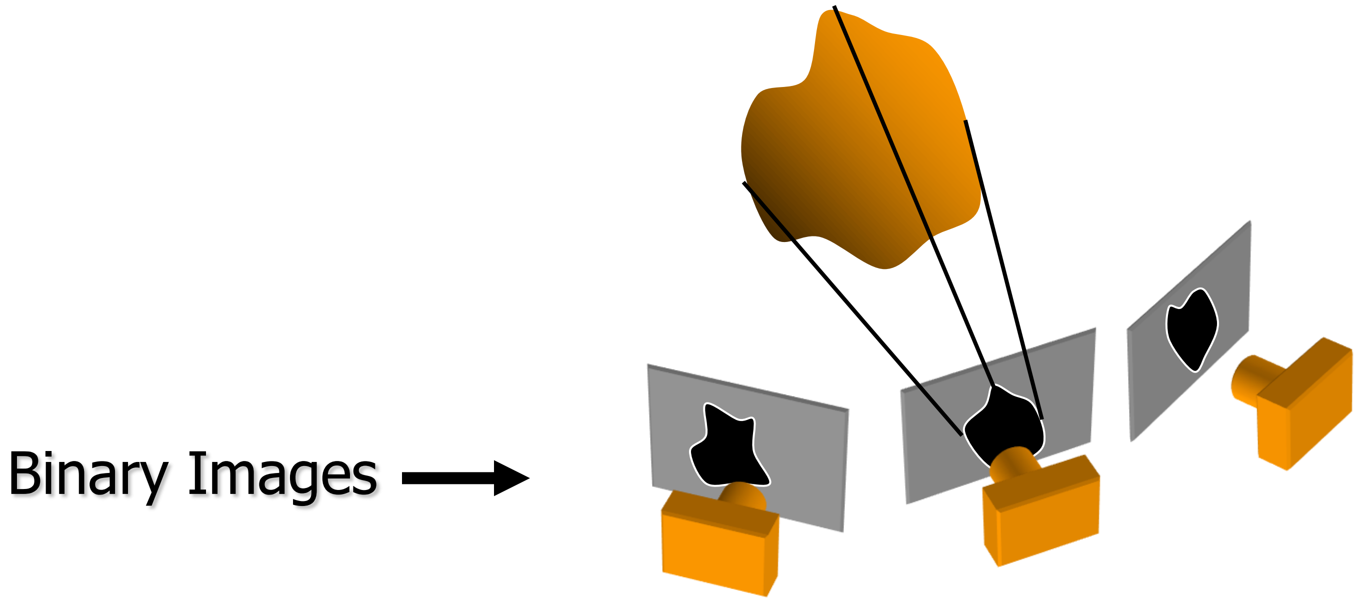
Practical Questions

- How do we compute photo-consistent models?

Voxel coloring solutions

1. C=2 (shape from silhouettes)
   - Volume intersection [Baumgart 1974]
     - > For more info: *Rapid octree construction from image sequences.* R. Szeliski, CVGIP: Image Understanding, 58(1):23-32, July 1993. (this paper is apparently not available online) or
     - > W. Matusik, C. Buehler, R. Raskar, L. McMillan, and S. J. Gortler, *Image-Based Visual Hulls*, SIGGRAPH 2000 (pdf 1.6 MB)
2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. General Case
   - Space carving [Kutulakos & Seitz 98]

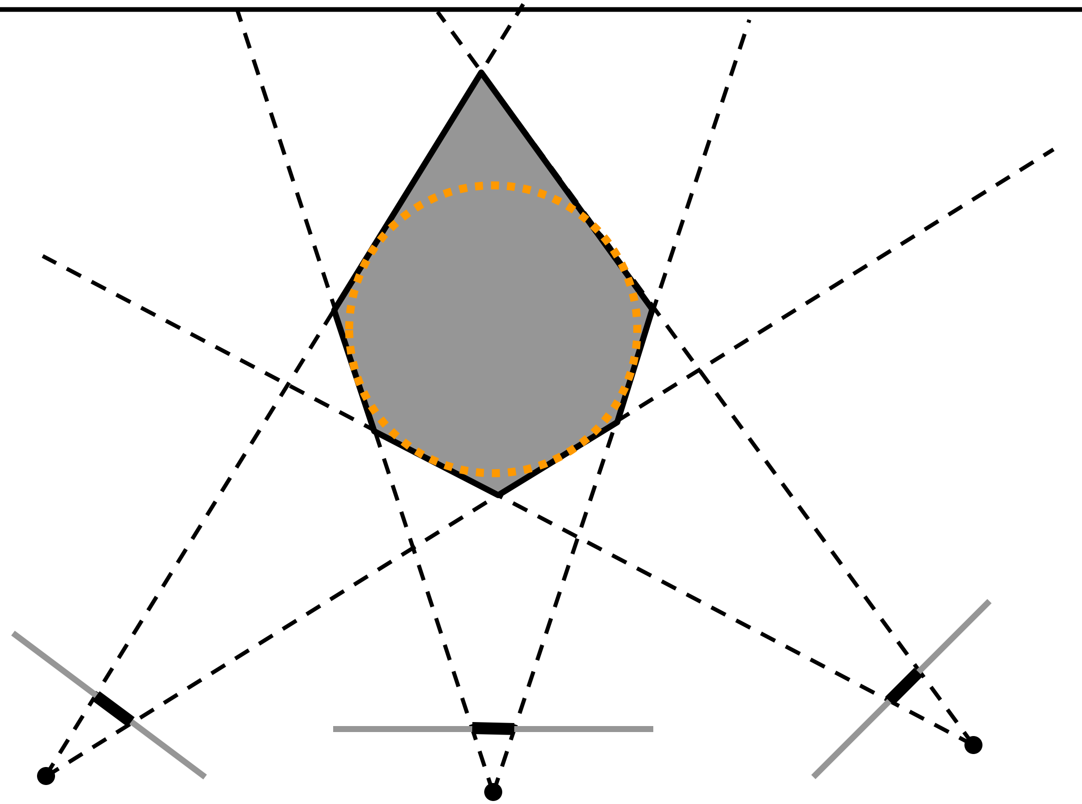
Reconstruction from Silhouettes (C = 2)



Approach:

- *Backproject* each silhouette
- Intersect backprojected volumes

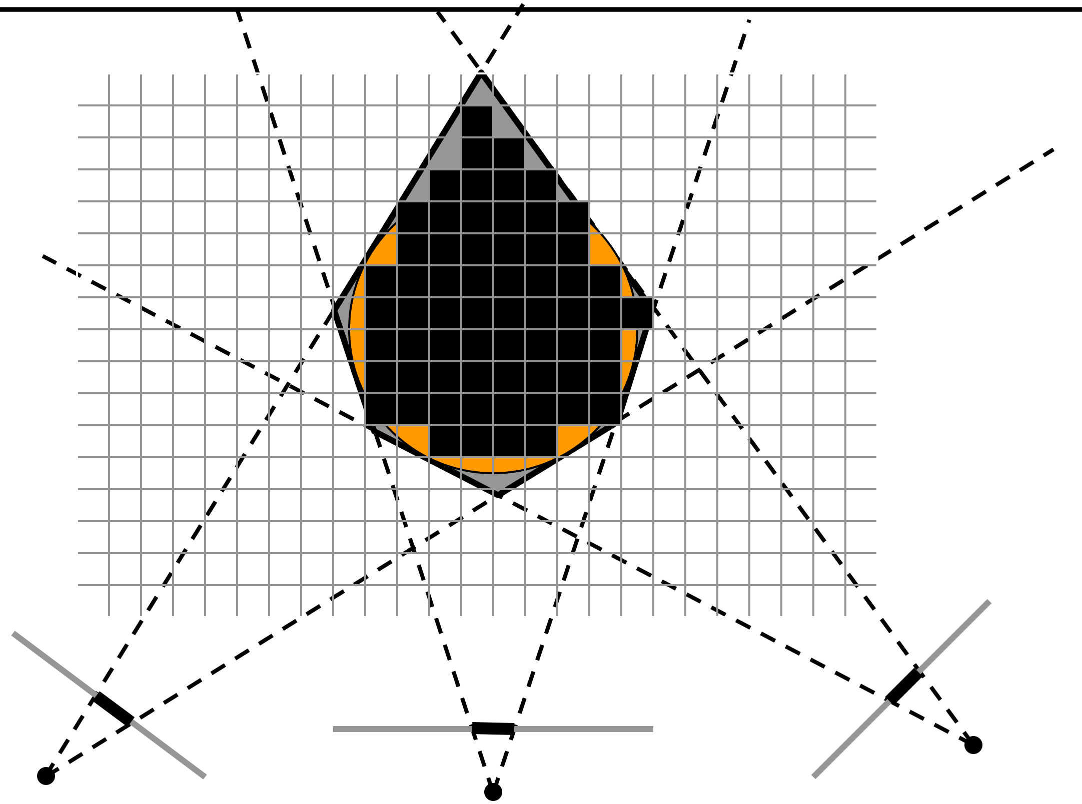
Volume intersection



Reconstruction Contains the True Scene

- But is generally not the same
- In the limit (all views) get *visual 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(?), 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] or interval techniques [Matusik 2000]

Cons

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

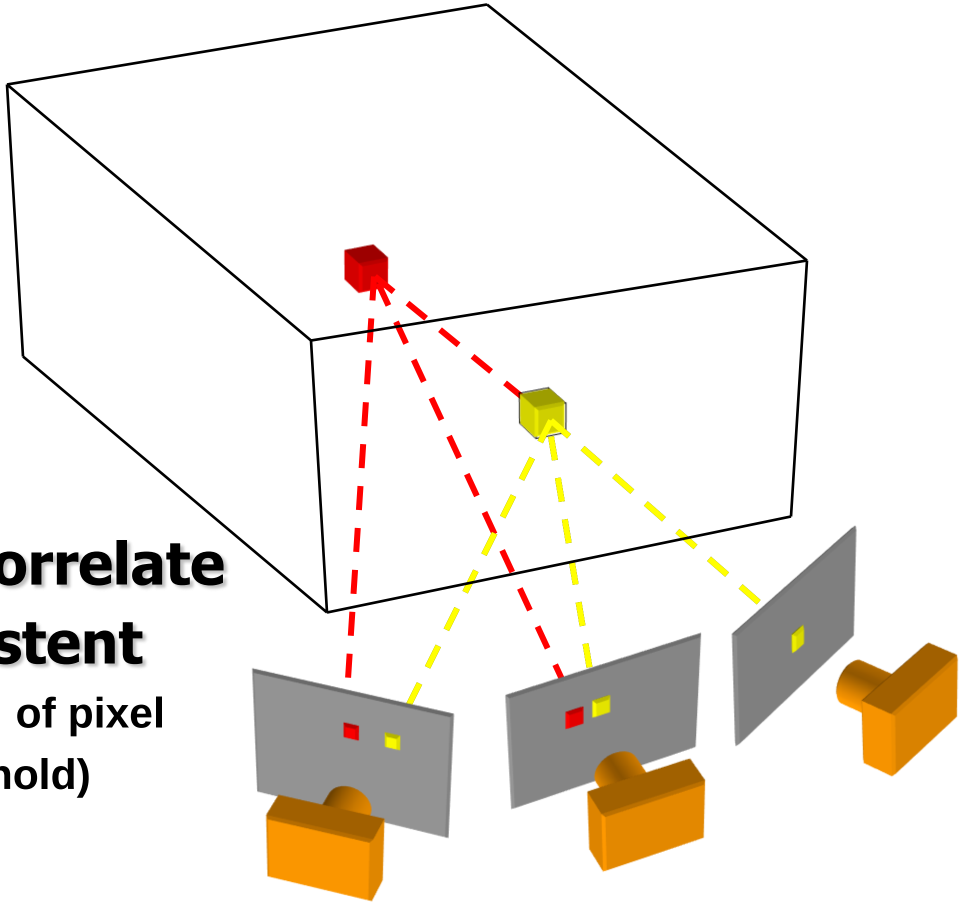
Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Baumgart 1974]

2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
     - > For more info: http://www.cs.washington.edu/homes/seitz/papers/ijcv99.pdf

3. General Case
   - Space carving [Kutulakos & Seitz 98]

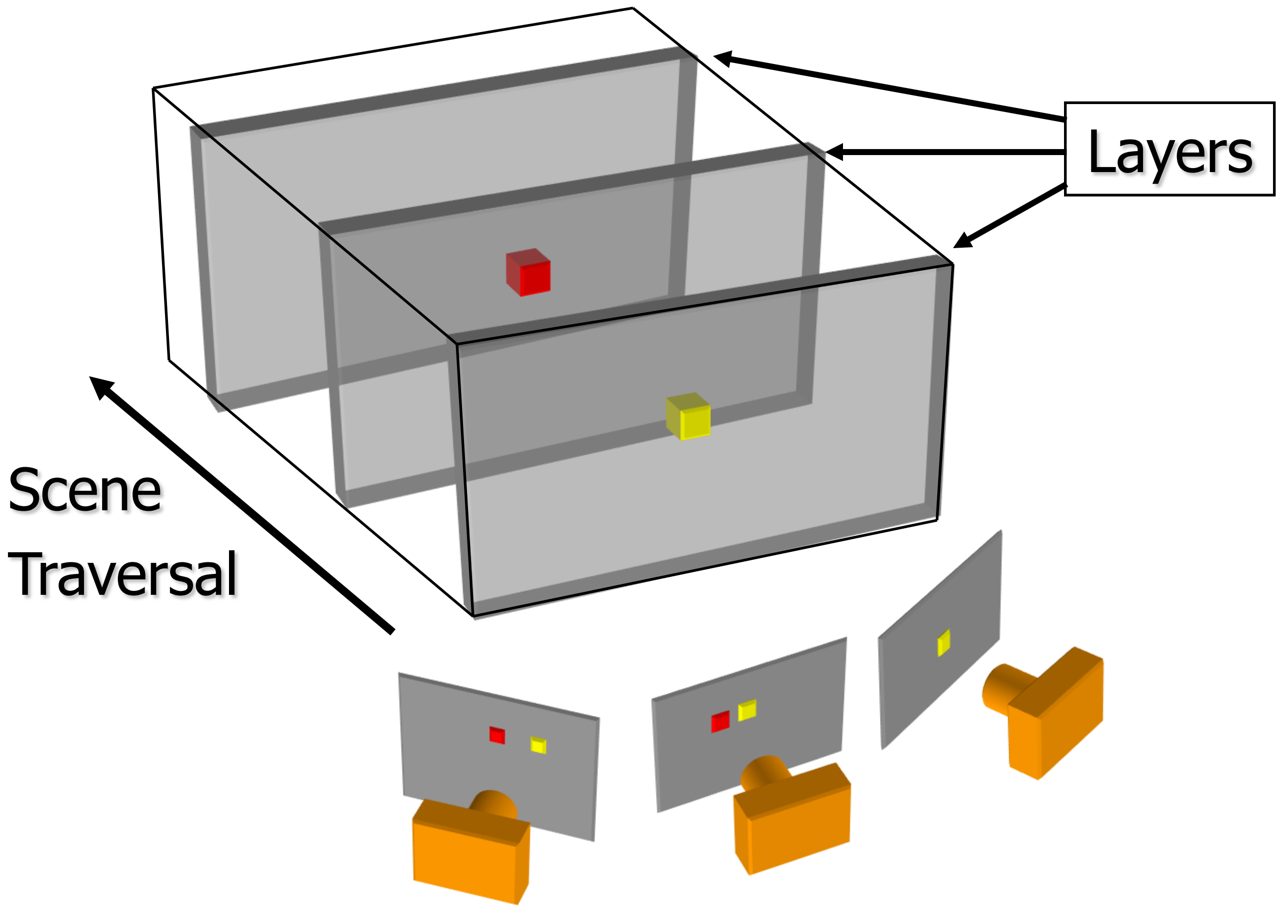
Voxel Coloring Approach



1. **Choose voxel**
2. **Project and correlate**
3. **Color if consistent**
   (standard deviation of pixel colors below threshold)

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

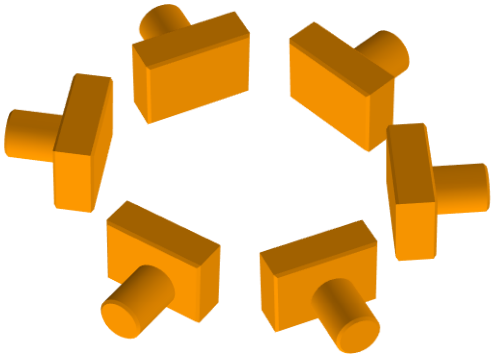
Depth Ordering: visit occluders first!



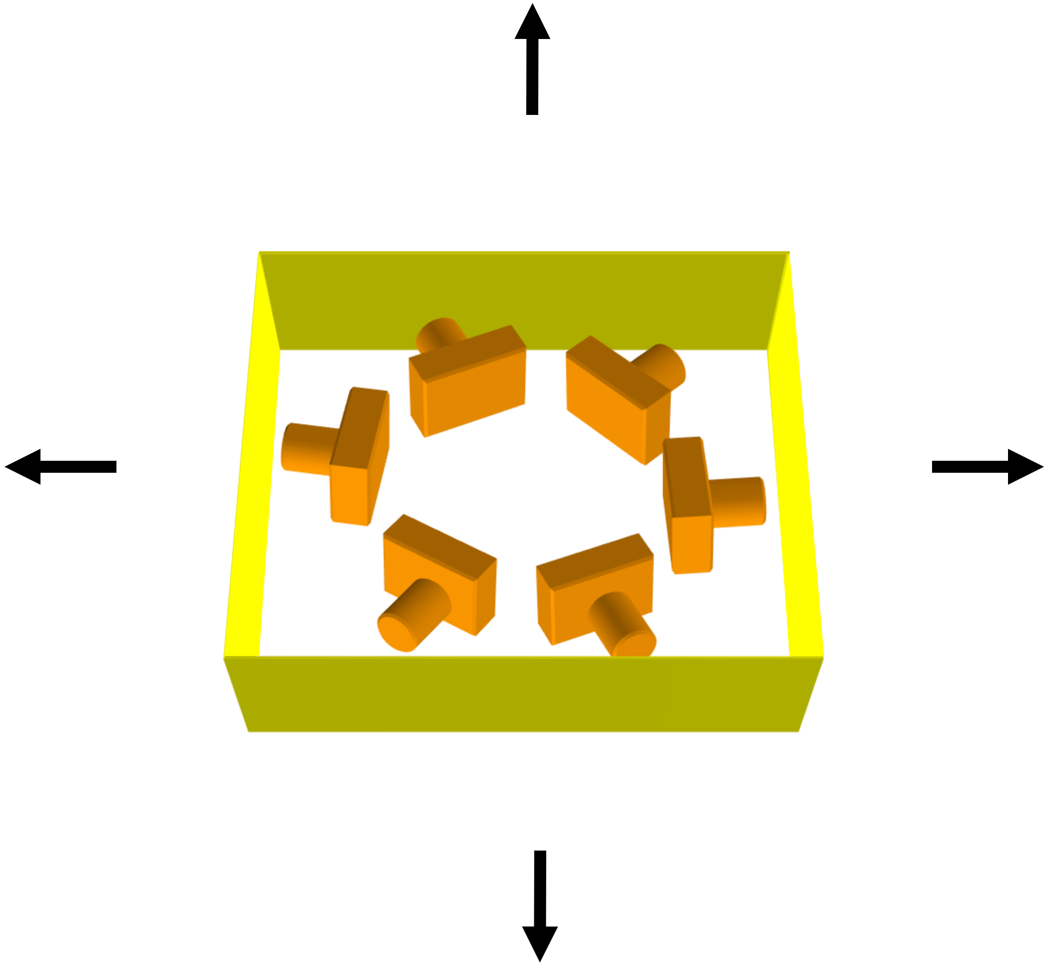
Condition: depth order is the *same for all input views*

Panoramic Depth Ordering

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

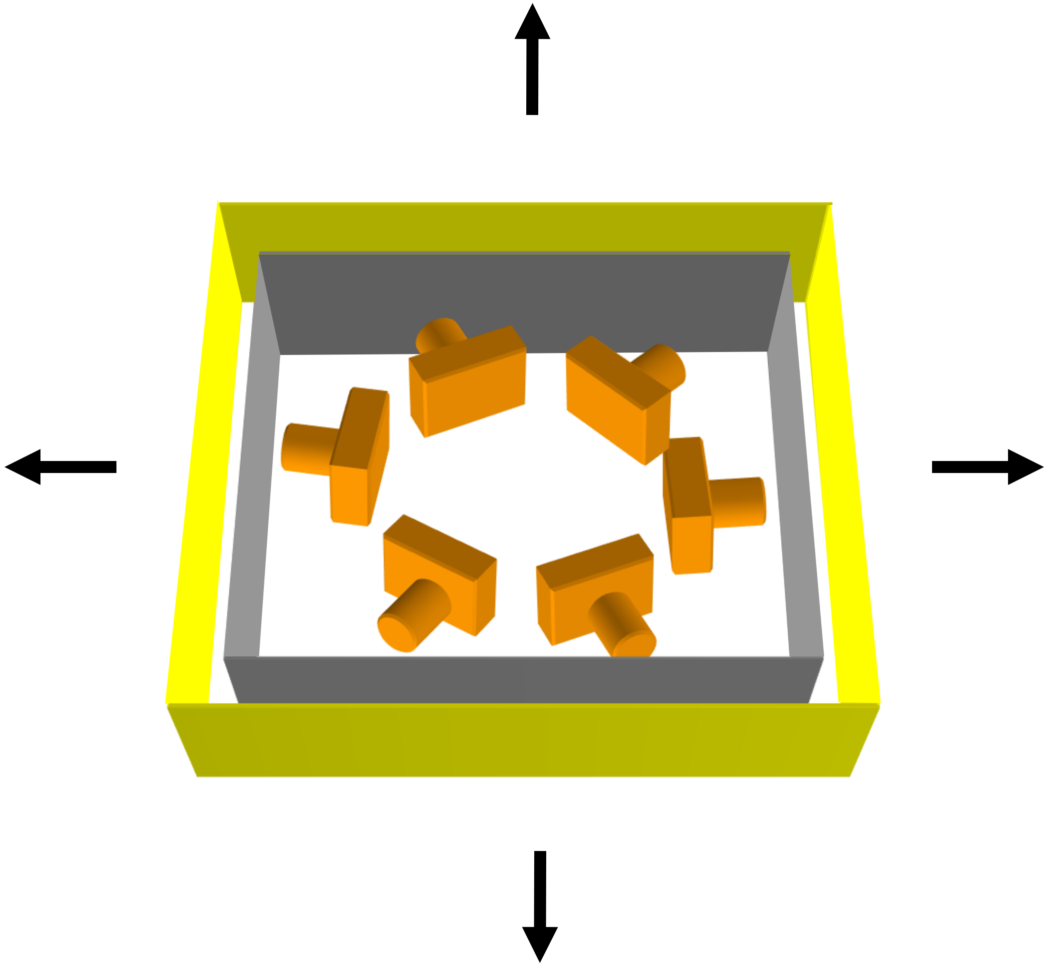


Panoramic Depth Ordering



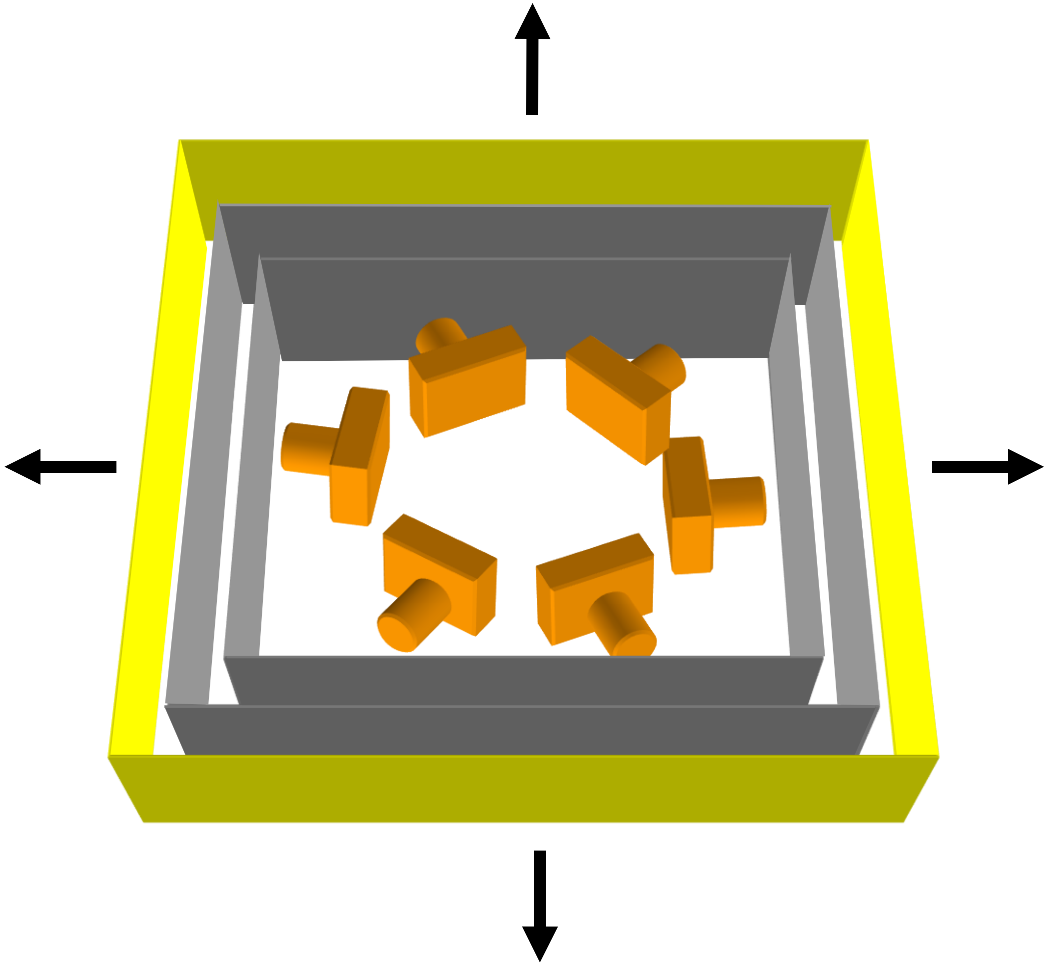
Layers radiate outwards from cameras

Panoramic Layering



Layers radiate outwards from cameras

Panoramic Layering

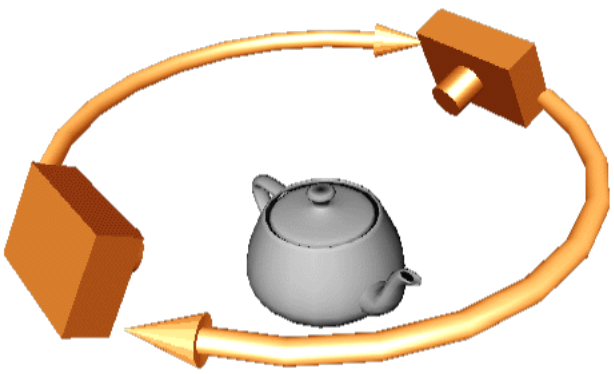


Layers radiate outwards from cameras

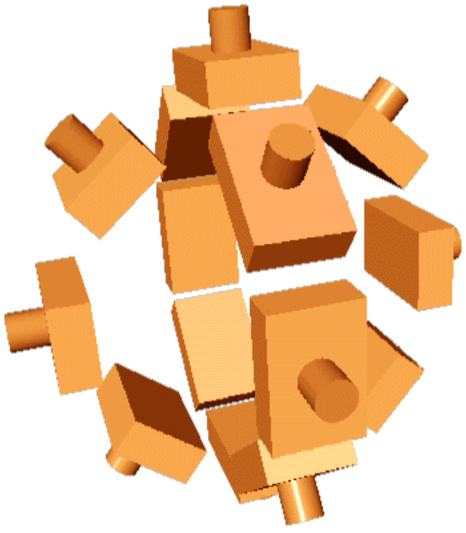
Compatible Camera Configurations

Depth-Order Constraint

- Scene outside convex hull of camera centers



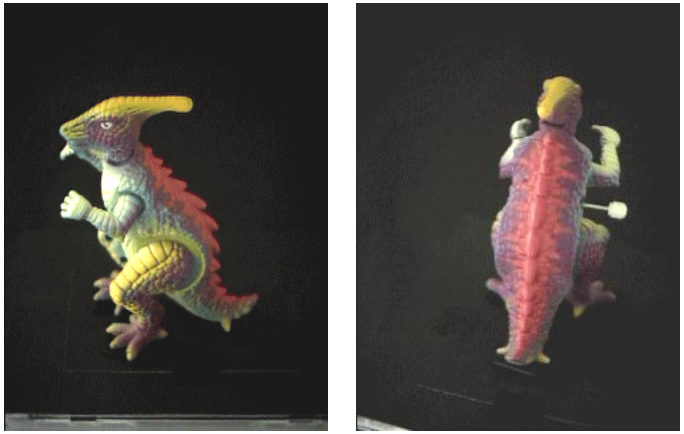
Inward-Looking



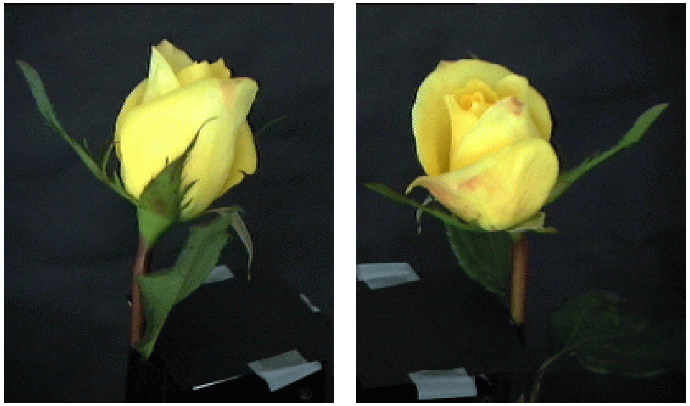
Outward-Looking

Calibrated Image Acquisition

Calibrated Turntable



Selected Dinosaur Images



Selected Flower Images

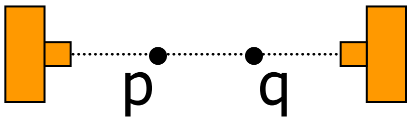
Voxel Coloring Results

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

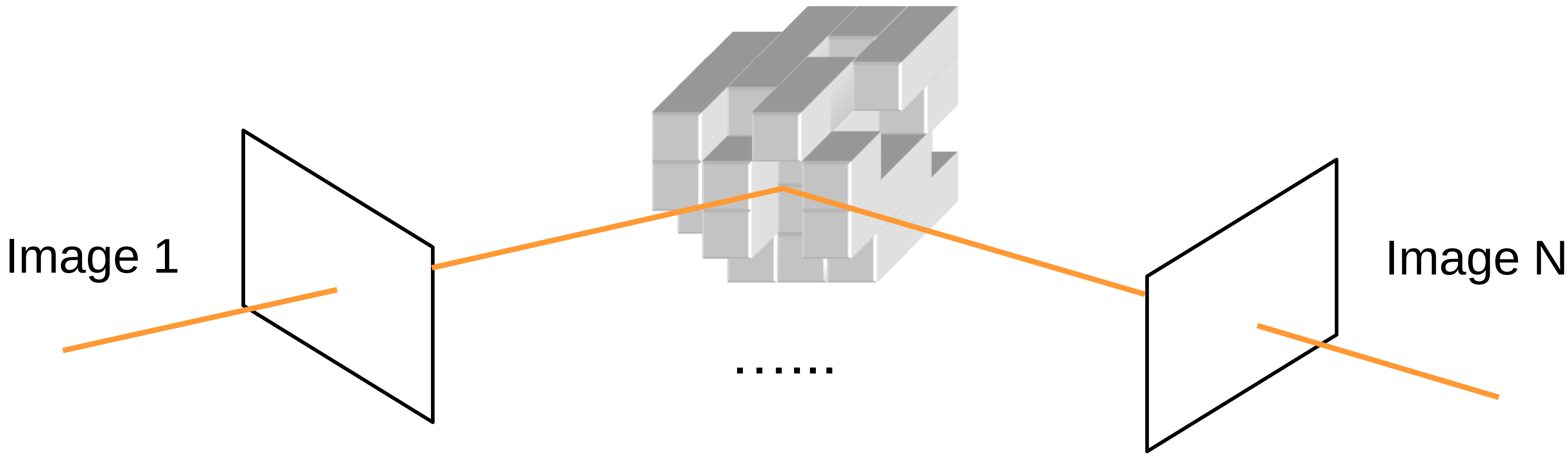
Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Baumgart 1974]

2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]

3. General Case
   - Space carving [Kutulakos & Seitz 98]
     - > For more info: http://www.cs.washington.edu/homes/seitz/papers/kutu-ijcv00.pdf

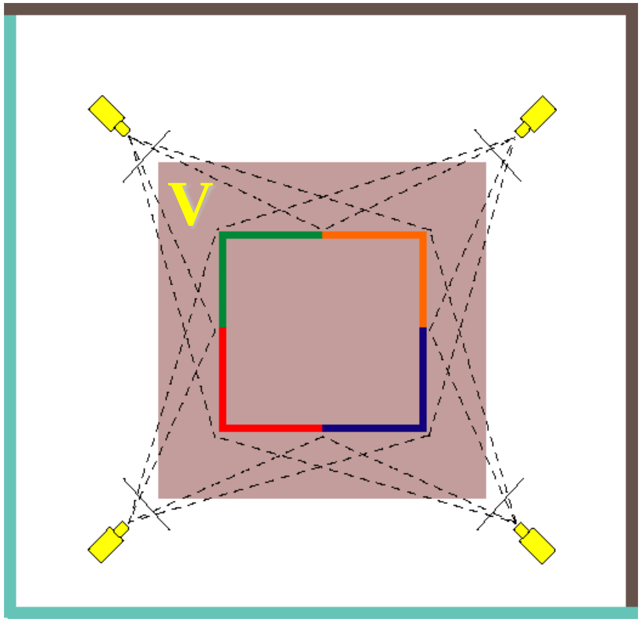
Space Carving Algorithm



Space Carving Algorithm

- Initialize to a volume V containing the true scene
- Choose a voxel on the current surface
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence

Which shape do you get?



True Scene

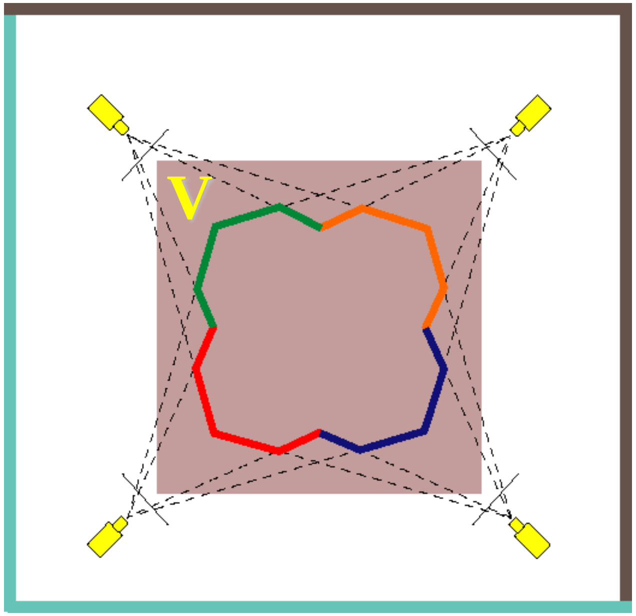


Photo Hull

The Photo Hull *is the UNION of all photo-consistent scenes in V*

- It is a photo-consistent scene reconstruction
- Tightest possible bound on the true scene

Space Carving Algorithm

Basic algorithm is unwieldy

- Complex 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)

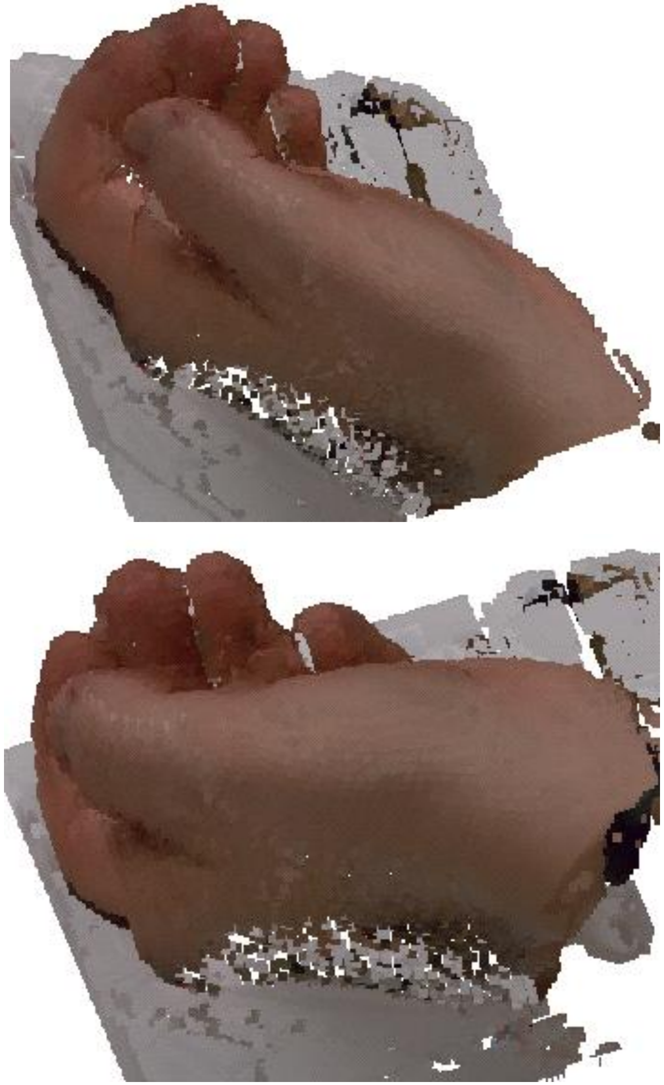
Reconstruction

Reconstruction

Reconstruction

Space Carving Results: Hand

Input Image
(1 of 100)



Views of Reconstruction

Properties of Space Carving

Pros

- Voxel coloring version is easy to implement, fast
- Photo-consistent results
- No smoothness prior

Cons

- Bulging
- No smoothness prior

Improvements

Unconstrained camera viewpoints

- Space carving [Kutulakos & Seitz 98]

Evolving a surface

- Level sets [Faugeras & Keriven 98]
- More recent work by Pons et al.

Global optimization

- Graph cut approaches
  - [Kolmogoriv & Zabih, ECCV 2002]
  - [Vogiatzis et al., PAMI 2007]

Modeling shiny (and other reflective) surfaces

- e.g., Zickler et al., *Helmholtz Stereopsis*

See today's reading for an overview of the state of the art