Dense Methods 1: Stereo

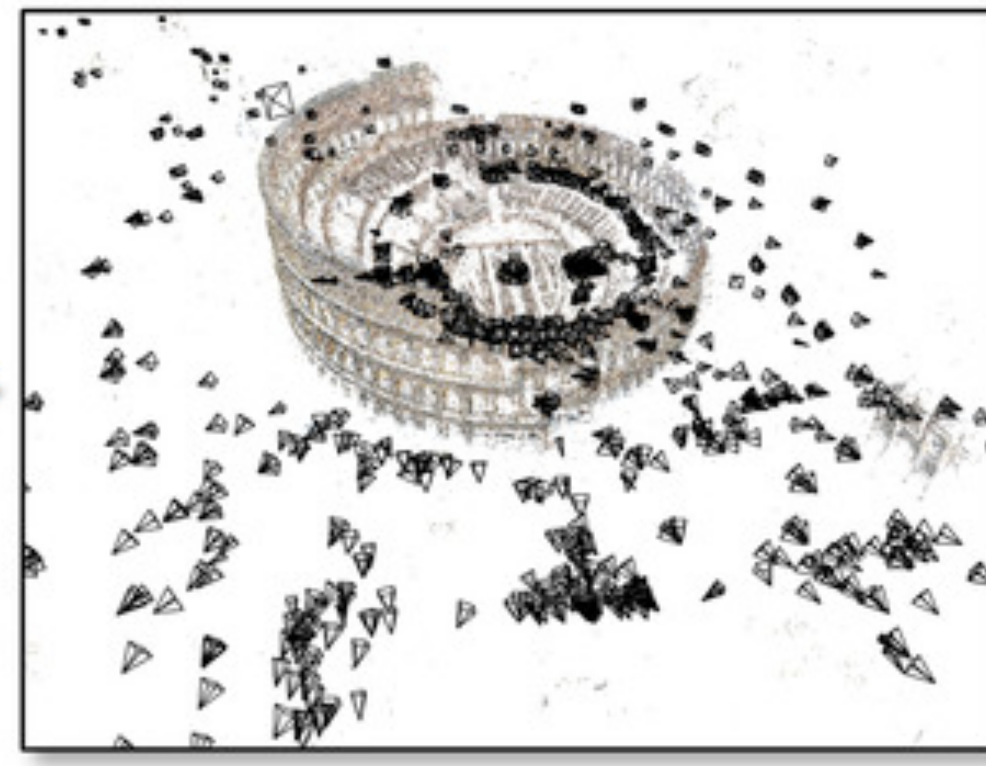
CSE P576

Dr. Matthew Brown

Dense Methods I: Stereo

- Stereo matching, local + global optimization
- Multi-view stereo, geometry representations
- Photometric Stereo

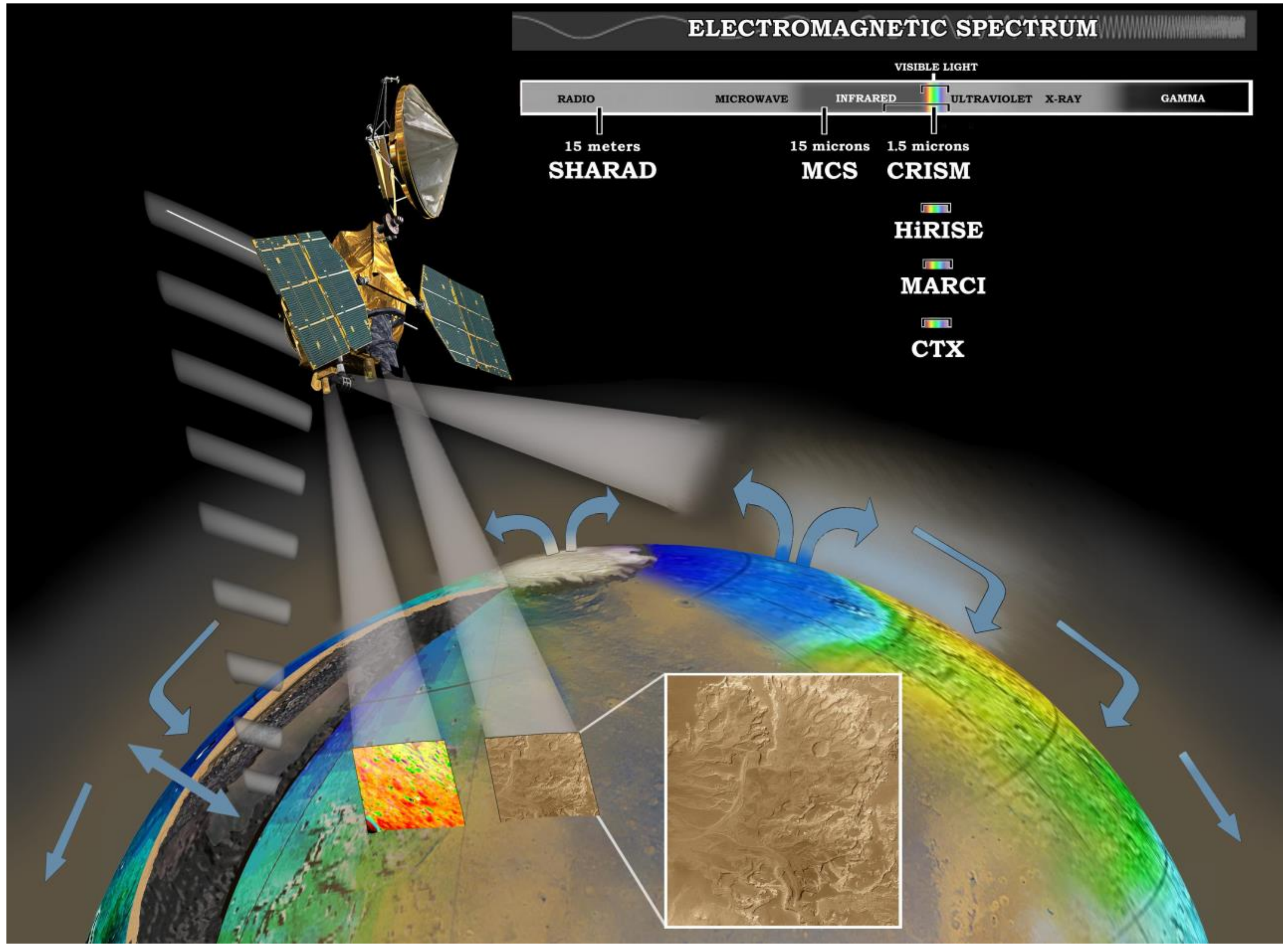
Application: Photo Collections → 3D



- Generate detailed 3D model (e.g., depth values at every pixel in input images)

[Y. Furukawa PMVS]

Application: Remote Sensing



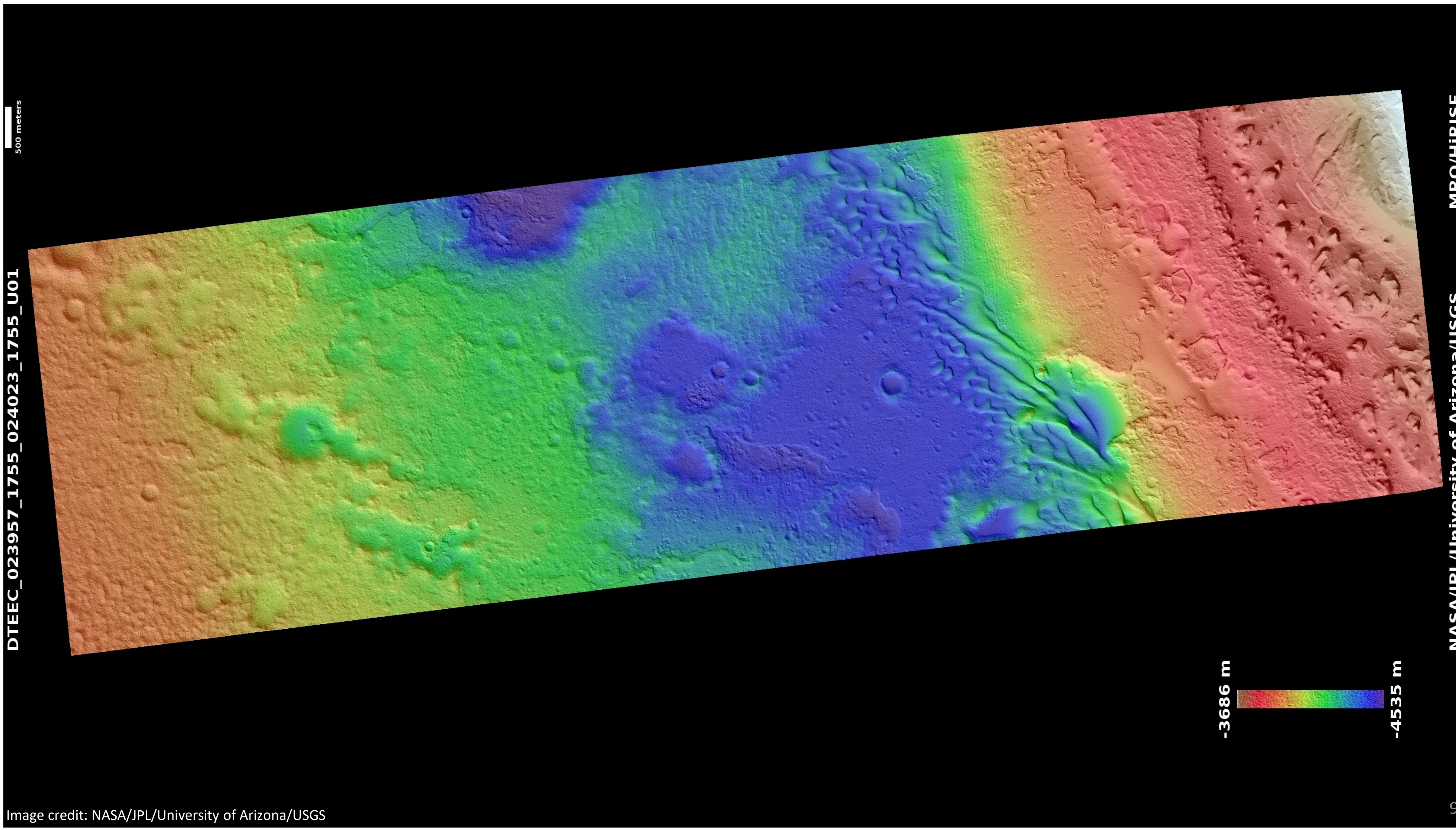
[NASA/JPL]

- Mars Reconnaissance Orbiter
- Launched 2005, ~13 orbits / earth day
- HIRISE camera pixels are 1μ radian (0.3m at 300km)
- MARCI camera has 5 visible + 2 UV bands, lower res

Application: Remote Sensing

- Martian surface elevation map

5km

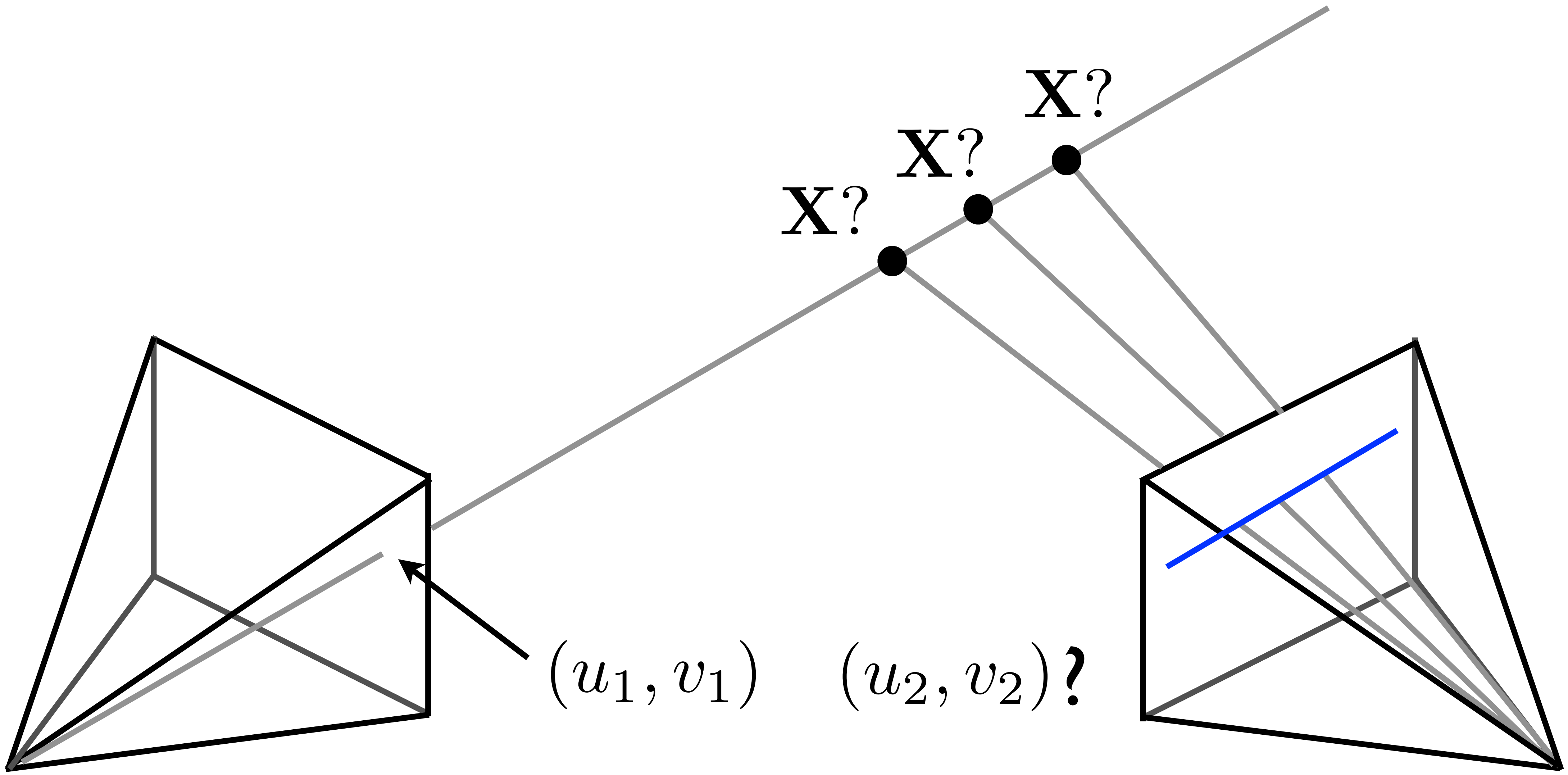


Application: Remote Sensing

- Martian surface detail

Epipolar Geometry

- A point in one view may lie on a line in the 2nd



Position in image 2 depends on the **depth** of the 3D point

2-view Stereo

- Camera motion only, points constrained to epipolar lines

1D Search

Stereo Camera Configuration

- Humans and many stereo cameras have parallel optical axes

[J. Elson]

Axis Aligned Stereo

- A common stereo configuration has camera optical axes aligned, with cameras related by a translation in the x direction

5.1

Stereo Matching

- In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

- Stereo algorithms search along scanlines for matches
- Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

5.2

[D. Scharstein]

Disparity and Depth: R

[D. Scharstein]

Disparity and Depth: L

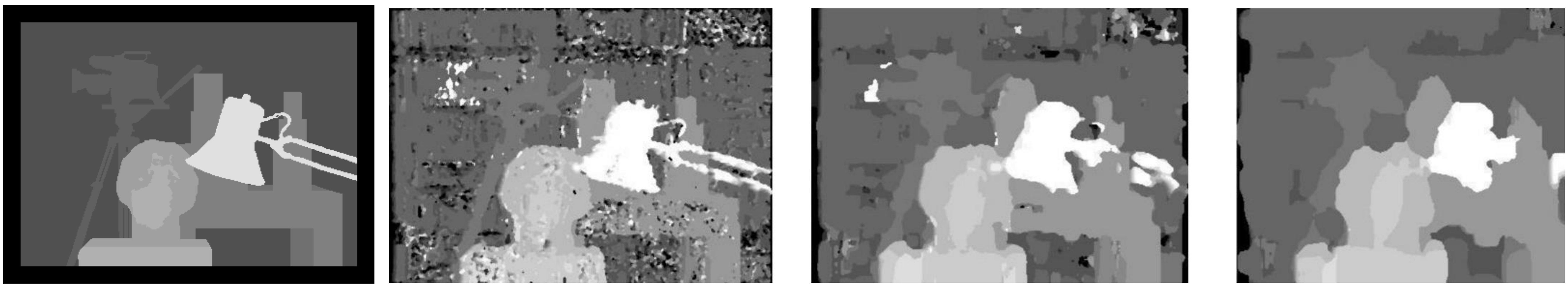
[D. Scharstein]

Disparity and Depth: R+L

[D. Scharstein]

Effect of Window Size

- Larger windows → smoothed result



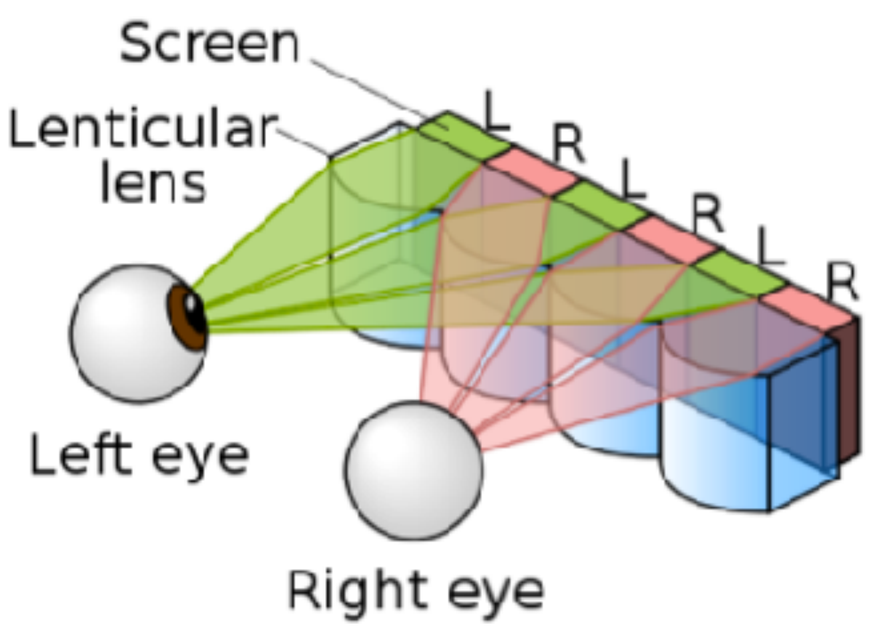
W=3 W=11 W=25

Anaglyph

- Stereo pair with images encoded in different color channels

Stereo Displays

- Field sequential (shutter) glasses transmit alternate left/right image at 120Hz



Lenticular lenses send different images directly to each eye, without the need for glasses

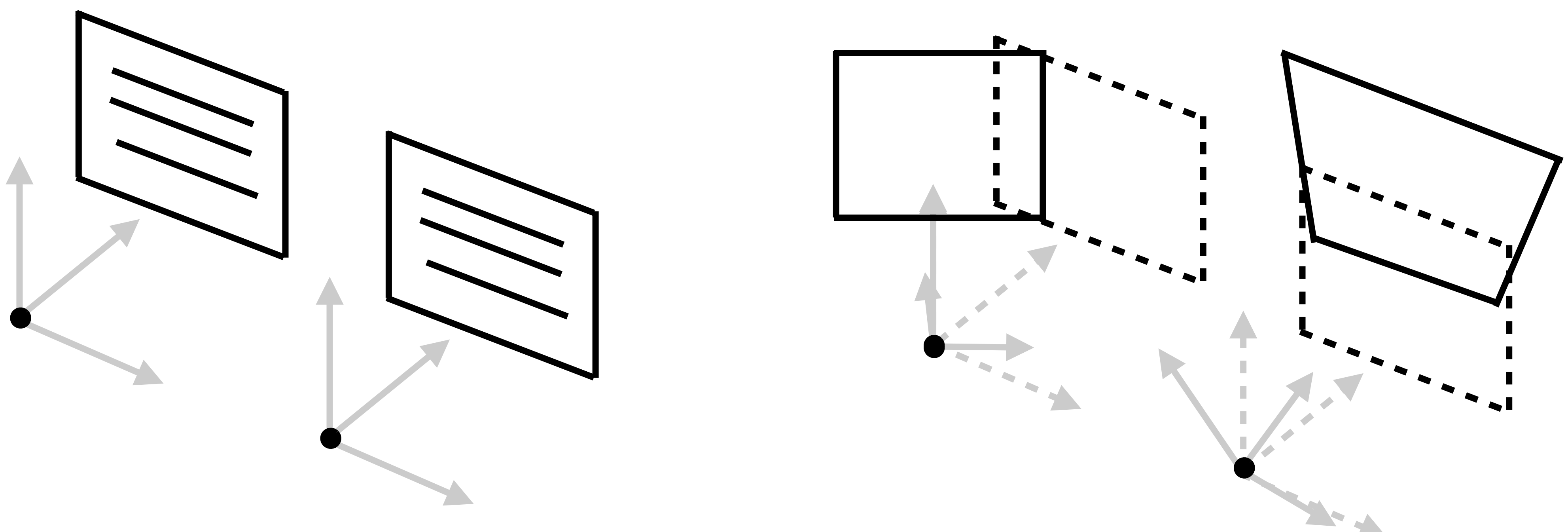
Stereo Displays

- VR headsets send L/R images directly to each eye

[Google Cardboard]

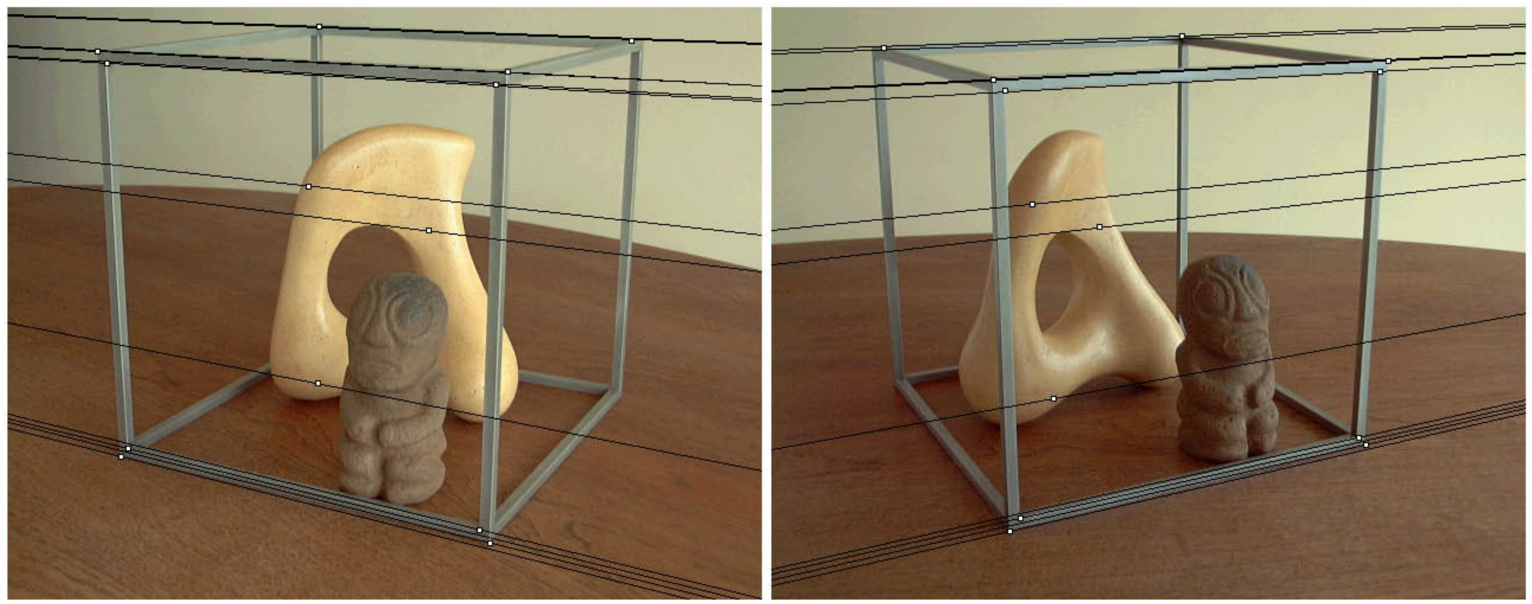
Stereo Rectification

- If the optical axes are not aligned, we can rotate the images (homography) until they are perpendicular to the baseline



Stereo Rectification

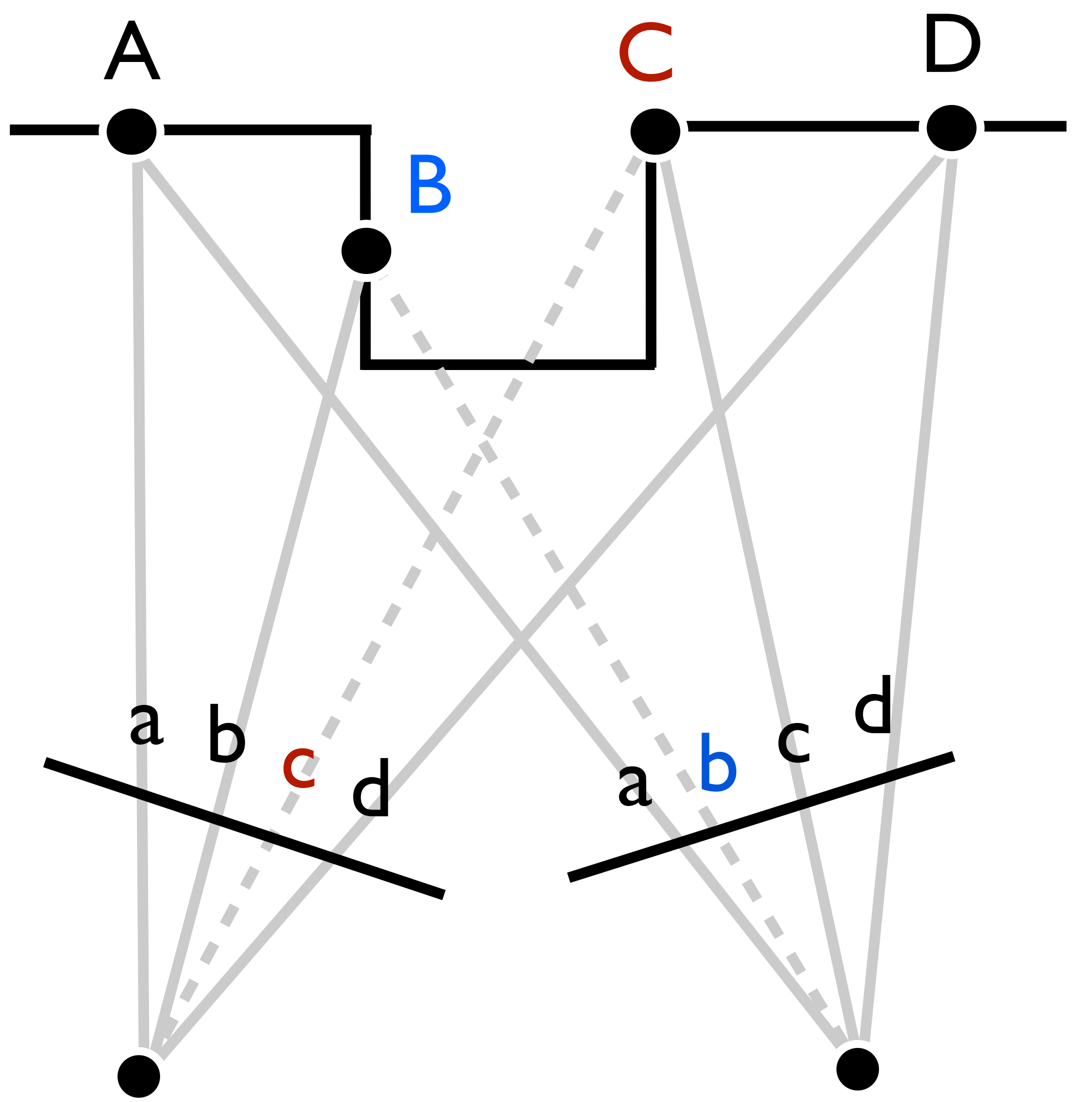
- Transform (rotate) images so that epipolar lines are horizontal



[Loop Zhang 1999]

Occlusions

- Sometimes a point in image 1 does not appear in image 2, or vice-versa (this is called an **occlusion**)



- Occlusions cause gaps in the stereo reconstruction
- + Matching is difficult nearby as aggregation windows often overlap the occluded region

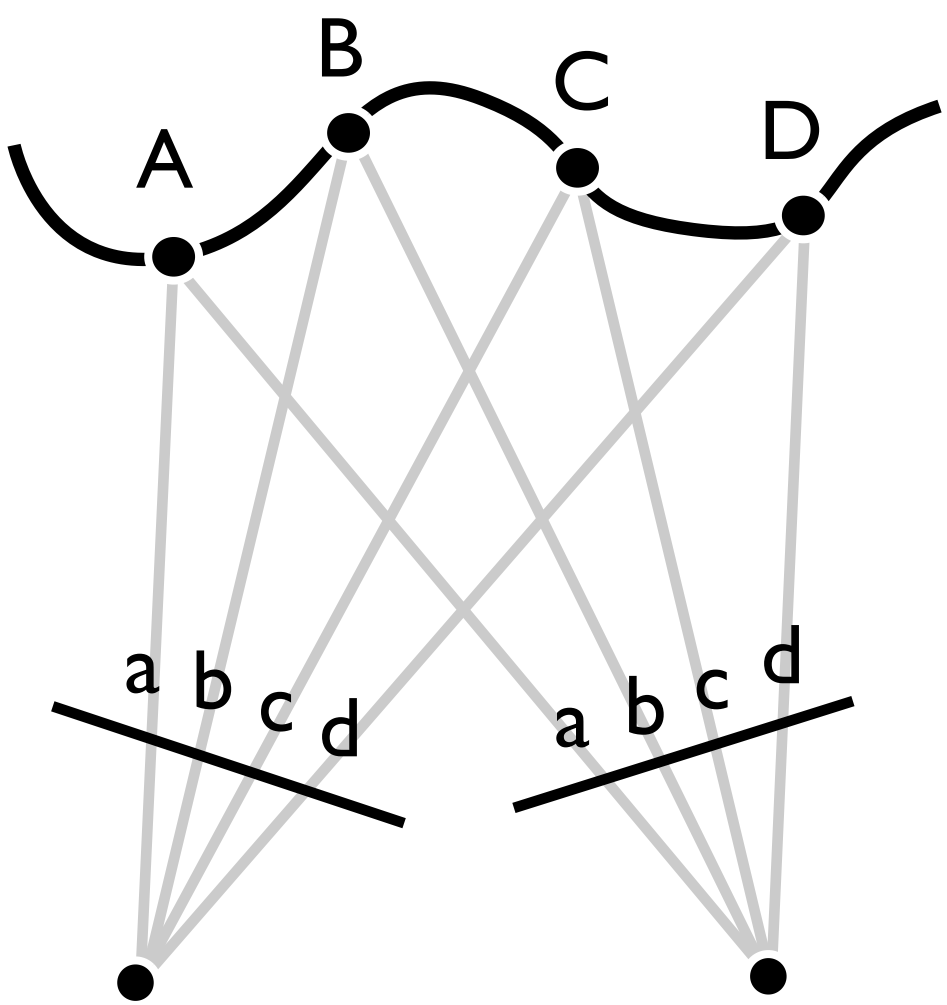
Edge Aware Stereo

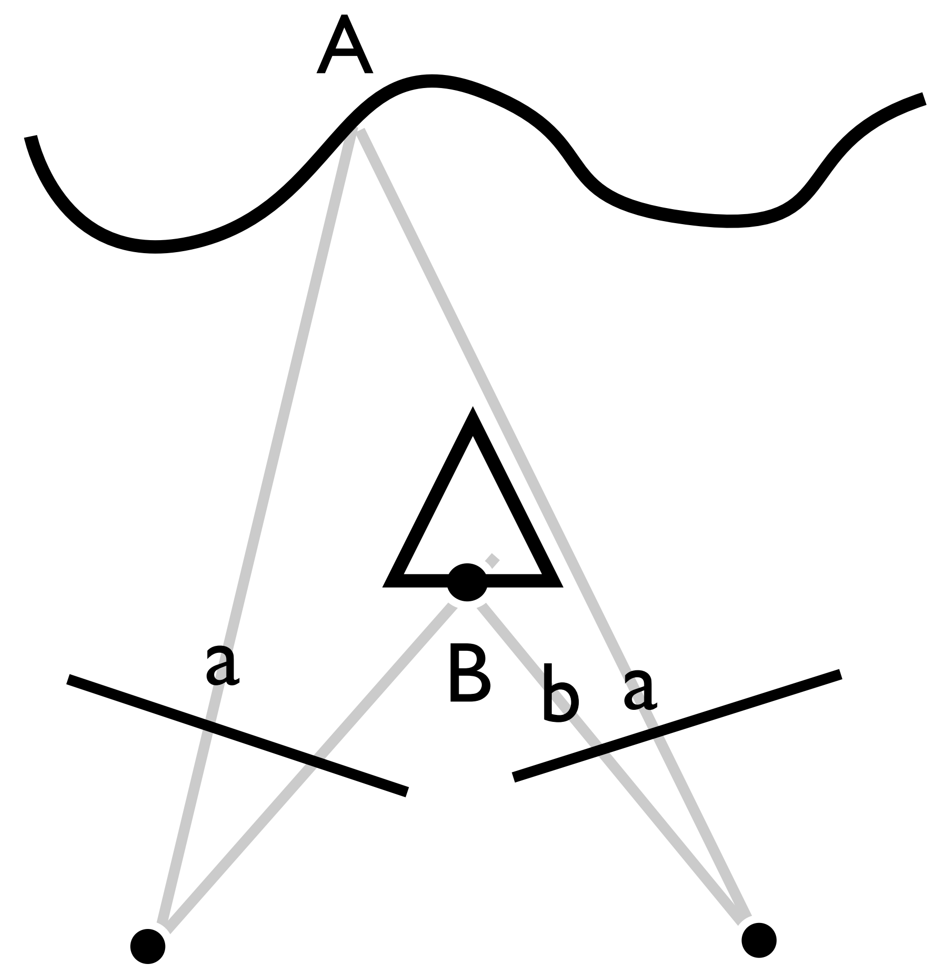
- Occlusions and depth discontinuities cause problems for stereo matching, as aggregation windows overlap multiple depths

- Segmentation-based stereo approaches aim to solve this by trying to guess the depth edges (e.g., joint segmentation and depth estimation [Taguchi et al 2008])

Ordering Constraint

- If point B is to the right of point A in image 1, the same is *usually* true in image 2

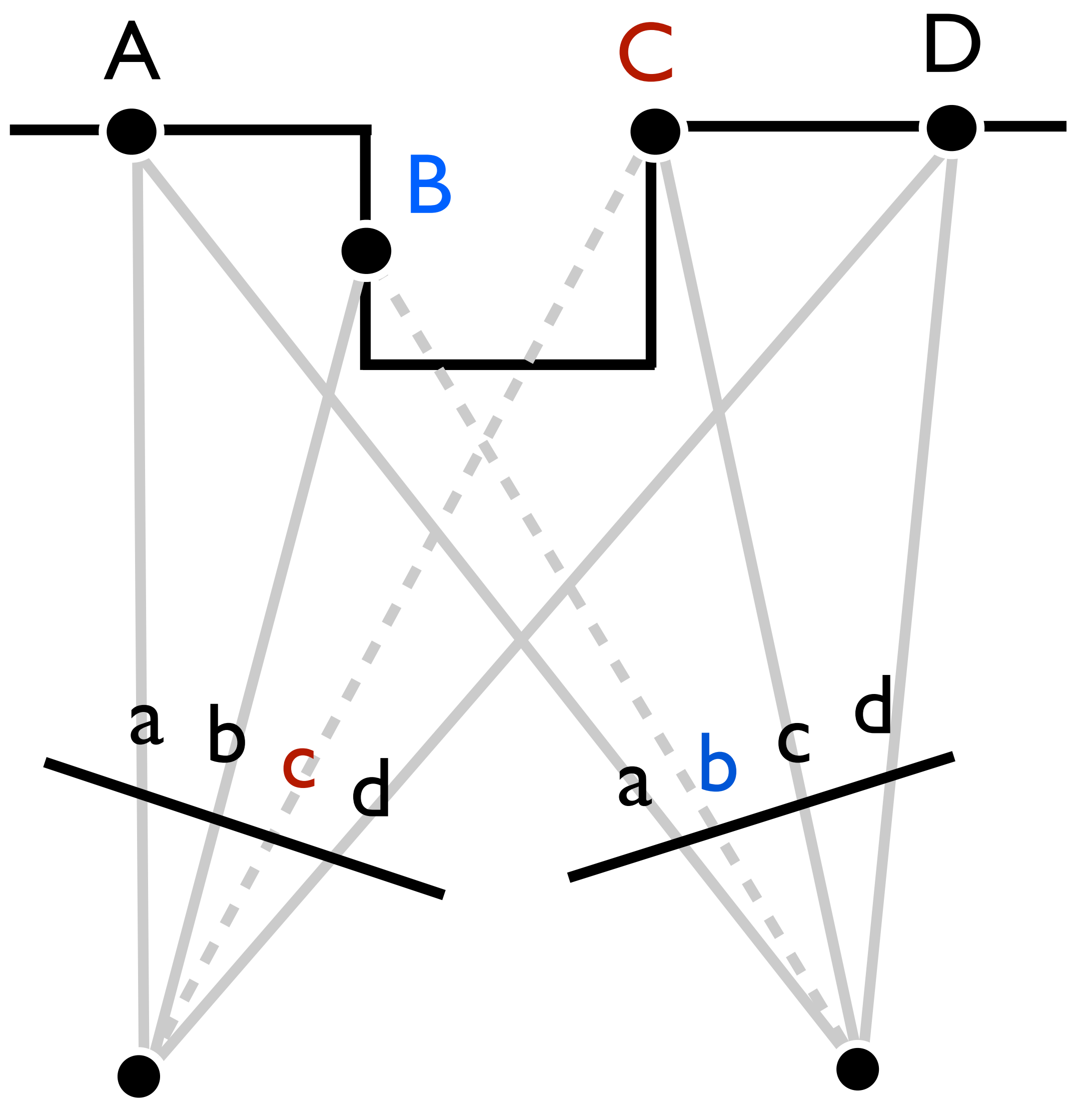




Not always, e.g., if an object is wholly within the ray triangle generated by A

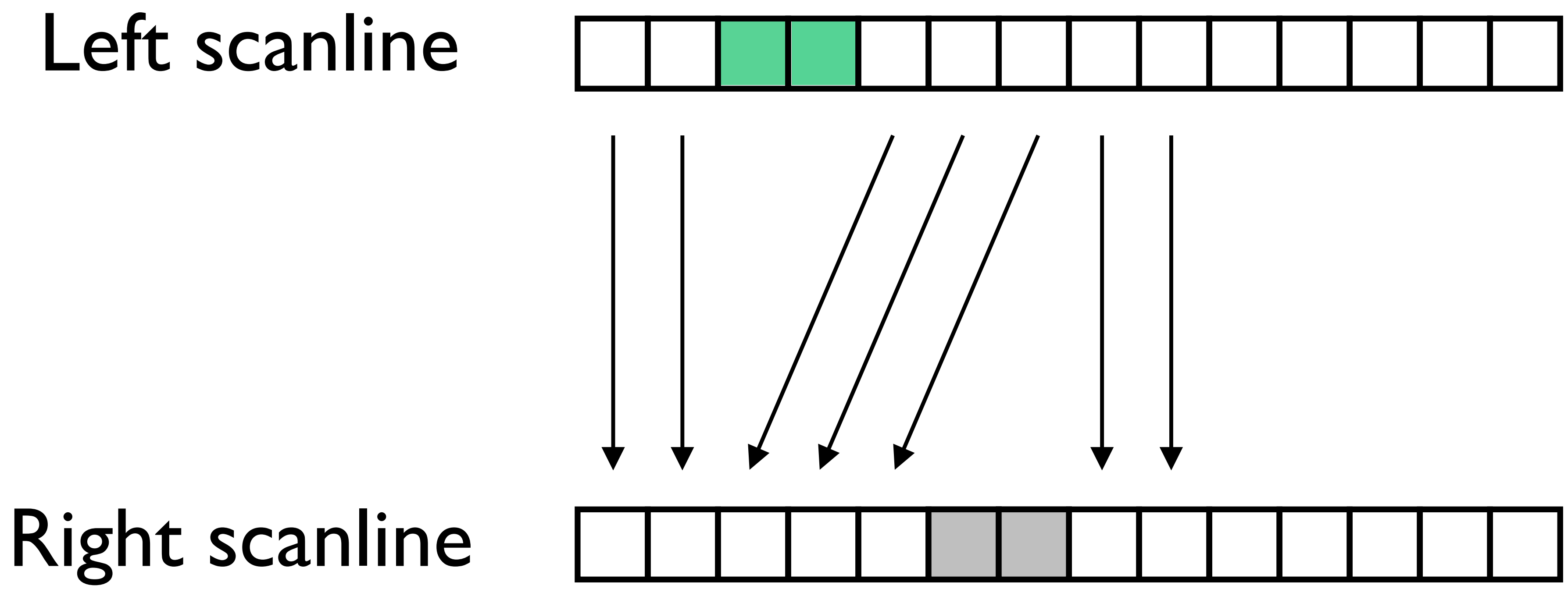
Occlusions + Ordering

- Note that the ordering constraint is still maintained in the presence of occlusions



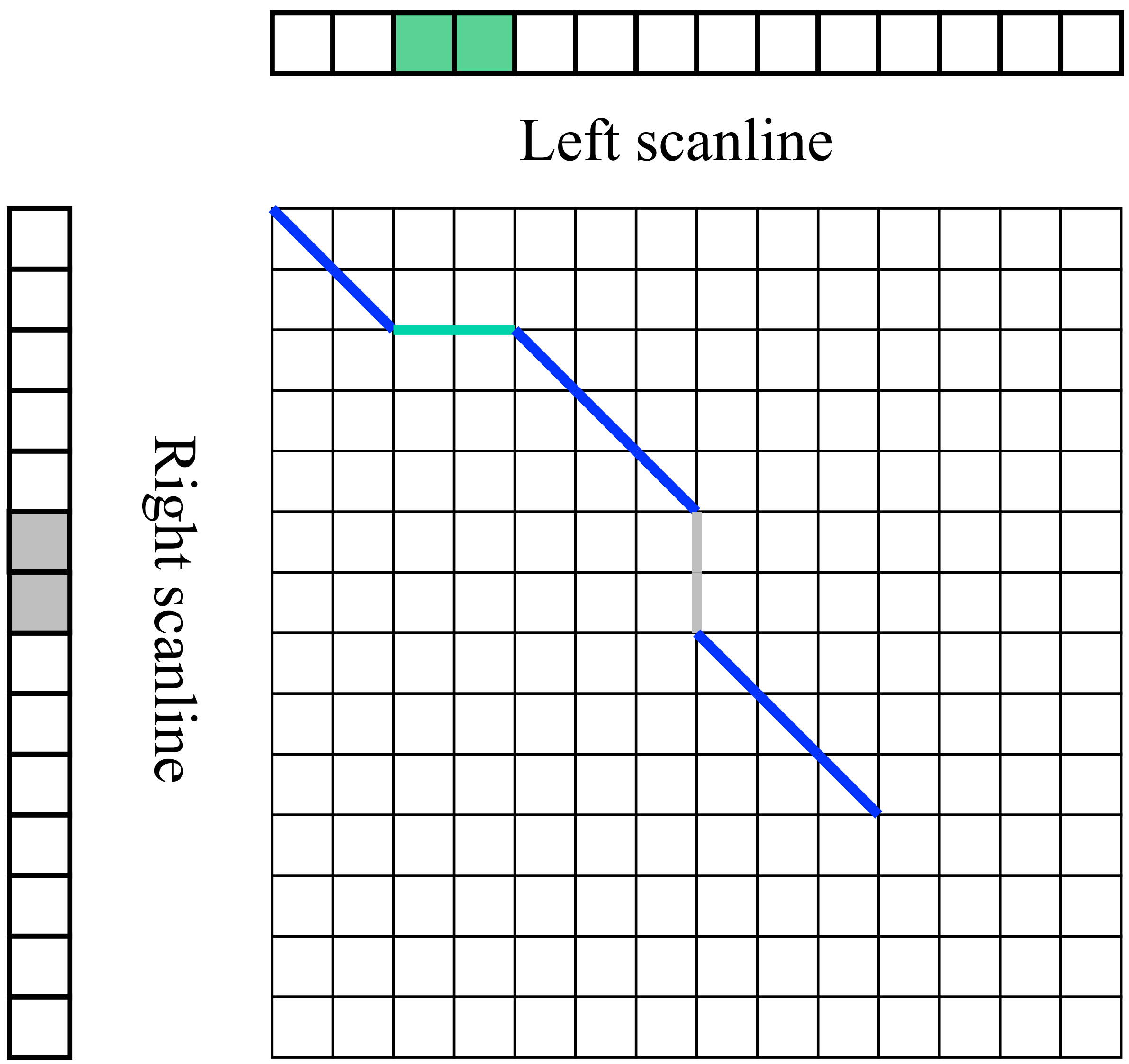
Optimal Scanline Mapping

- We can imagine a mapping between left and right scanlines
- If we assume the ordering constraint this must be monotonic, but there may be step changes (due to occlusions)
- How can we find the best monotonic sequence mapping left to right scanlines?



Dynamic Programming

- At each point, we may make one of 3 moves: left/right occlusion (higher cost), or sequential correspondence (lower cost based on patch SSD)

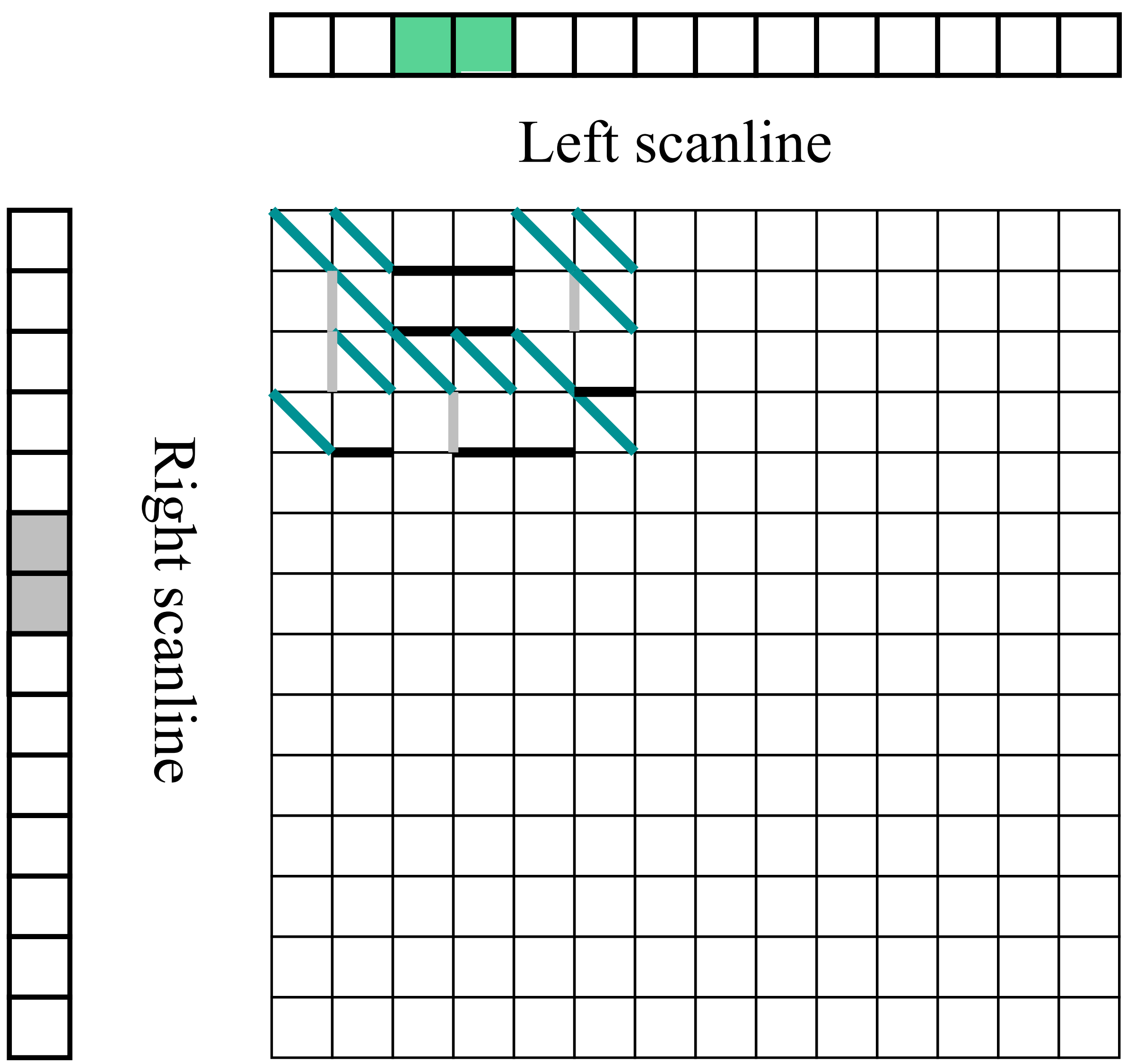


Look for a path from top left to bottom right with the lowest cost

[Cox et al 1996]

Dynamic Programming

- We need not consider all possible paths, as we only need to know the lowest cost path for reaching each state



Scan over grid finding min-cost paths and backtrack from the end

[Cox et al 1996]

Stereo Cost Functions

- Energy function for stereo matching based on disparity d(x,y)
- Sum of data and smoothness terms

$$E(d) = E_d(d) + \lambda E_s(d)$$

- Data term is cost of pixel x,y allocated disparity d (e.g., SSD)

$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y))$$

- Smoothness cost penalises disparity changes with robust ρ(.)

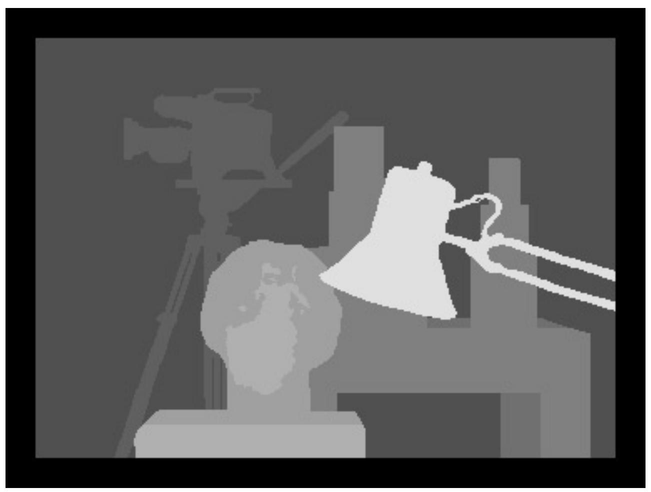
$$E_s(d) = \sum_{(x,y)} \rho(d(x, y) - d(x+1, y)) + \rho(d(x, y) - d(x, y+1))$$

- This is a Markov Random Field (MRF), which can be solved using techniques such as Graph Cuts

[Szeliski B5]

Stereo Comparison

- Global vs Scanline vs Local optimization



Ground truth

Graph Cuts [Kolmogorov Zabih 2001]

Dynamic Programming

SSD 21px aggregation

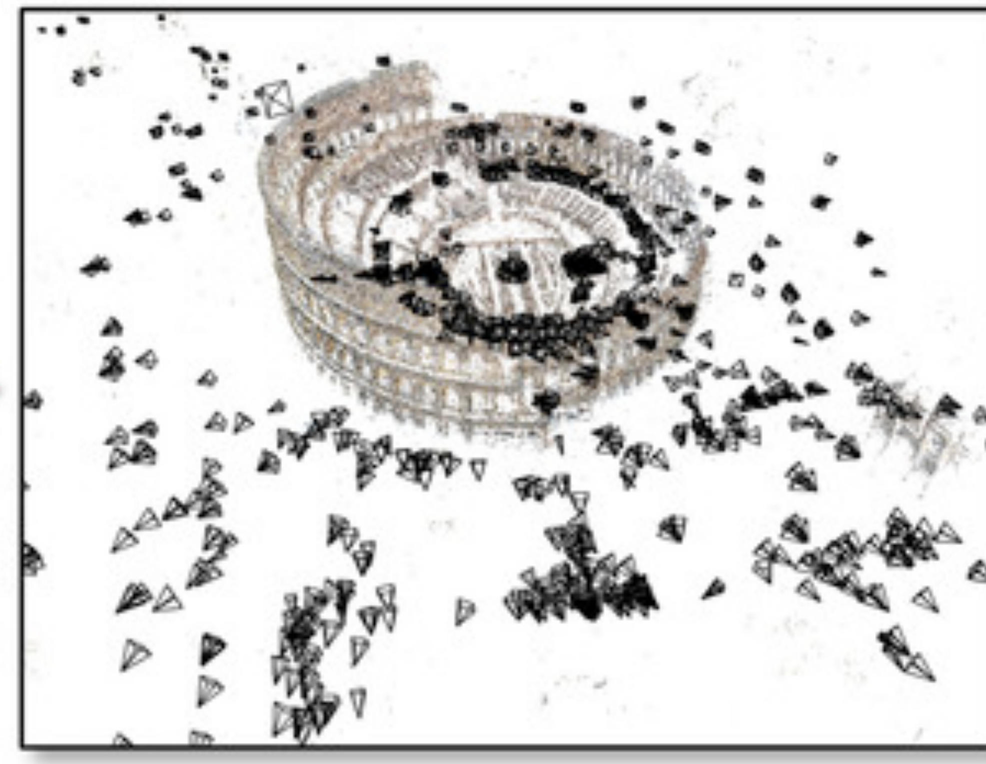
[Scharstein Szeliski 2002]

Multiview Stereo

- Plane sweep, volumetric, depth map merging

[Szeliski 11.6]

Multiview Stereo

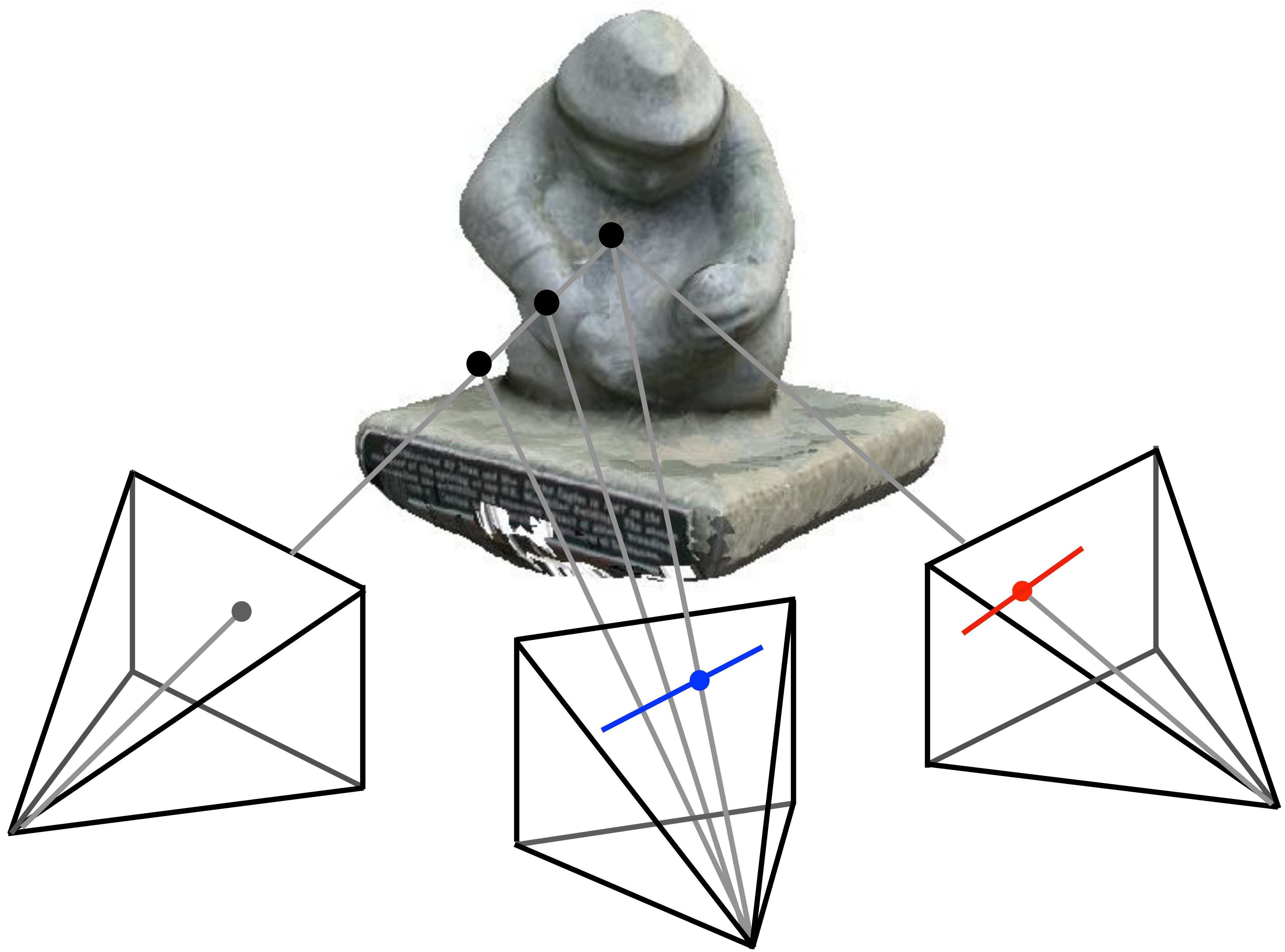


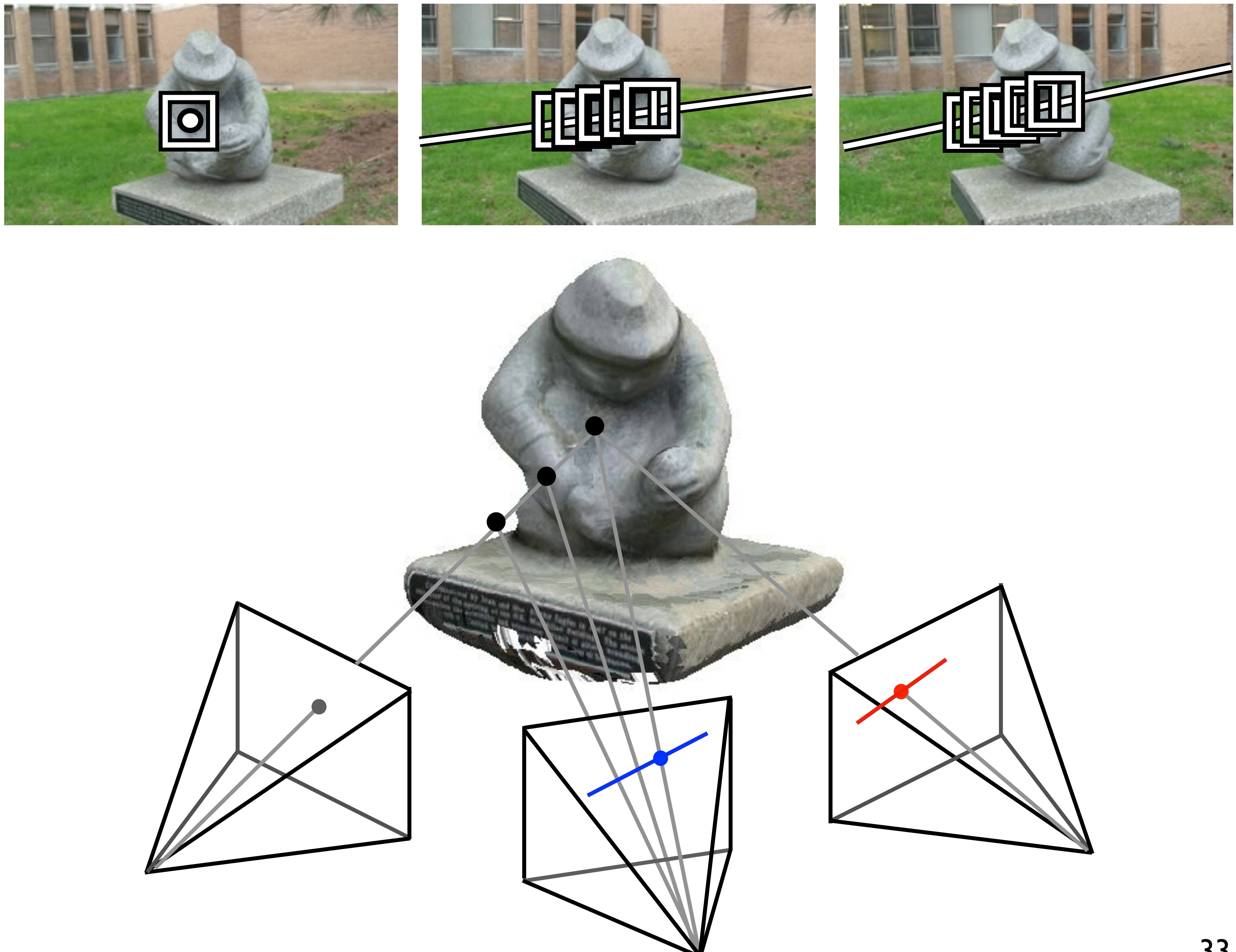
- Use information from N>2 views to form a dense 3D reconstruction

[Y. Furukawa PMVS]

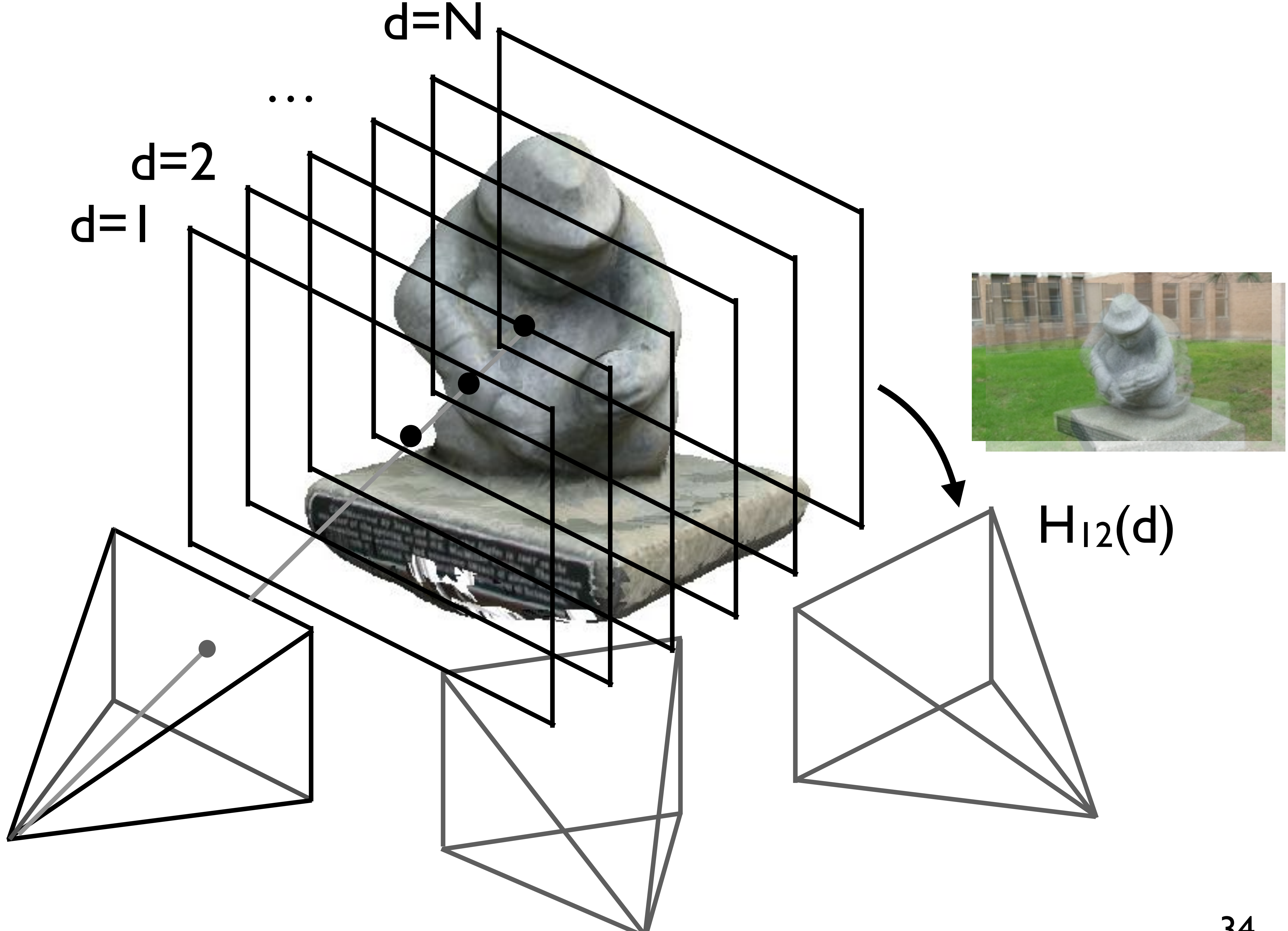
Multiview Stereo

- Search along epipolar lines to find good matches in N views





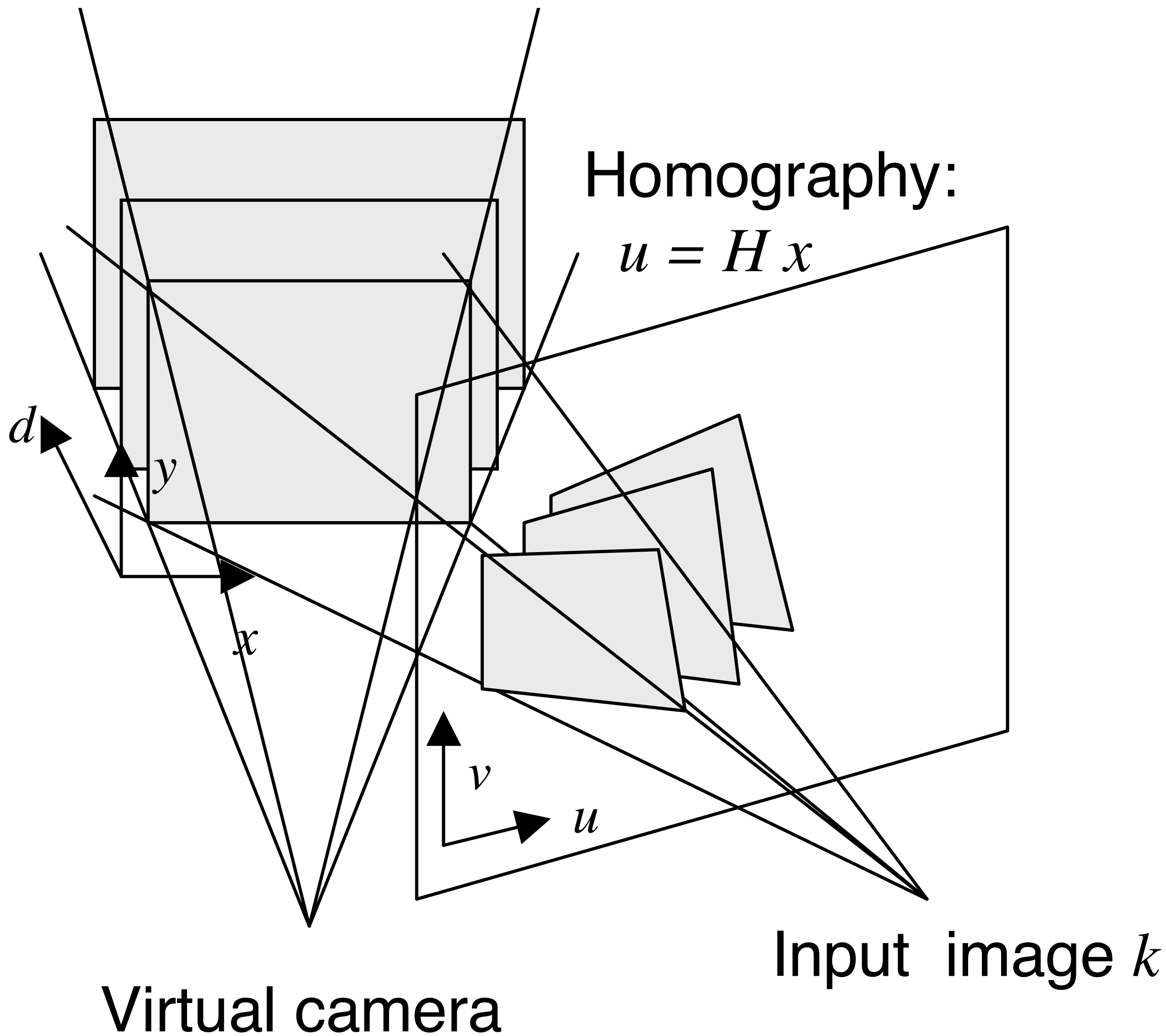
Plane Sweep Stereo



Plane Sweep Stereo

- Warp images using a set of planes in front of the camera

5.4



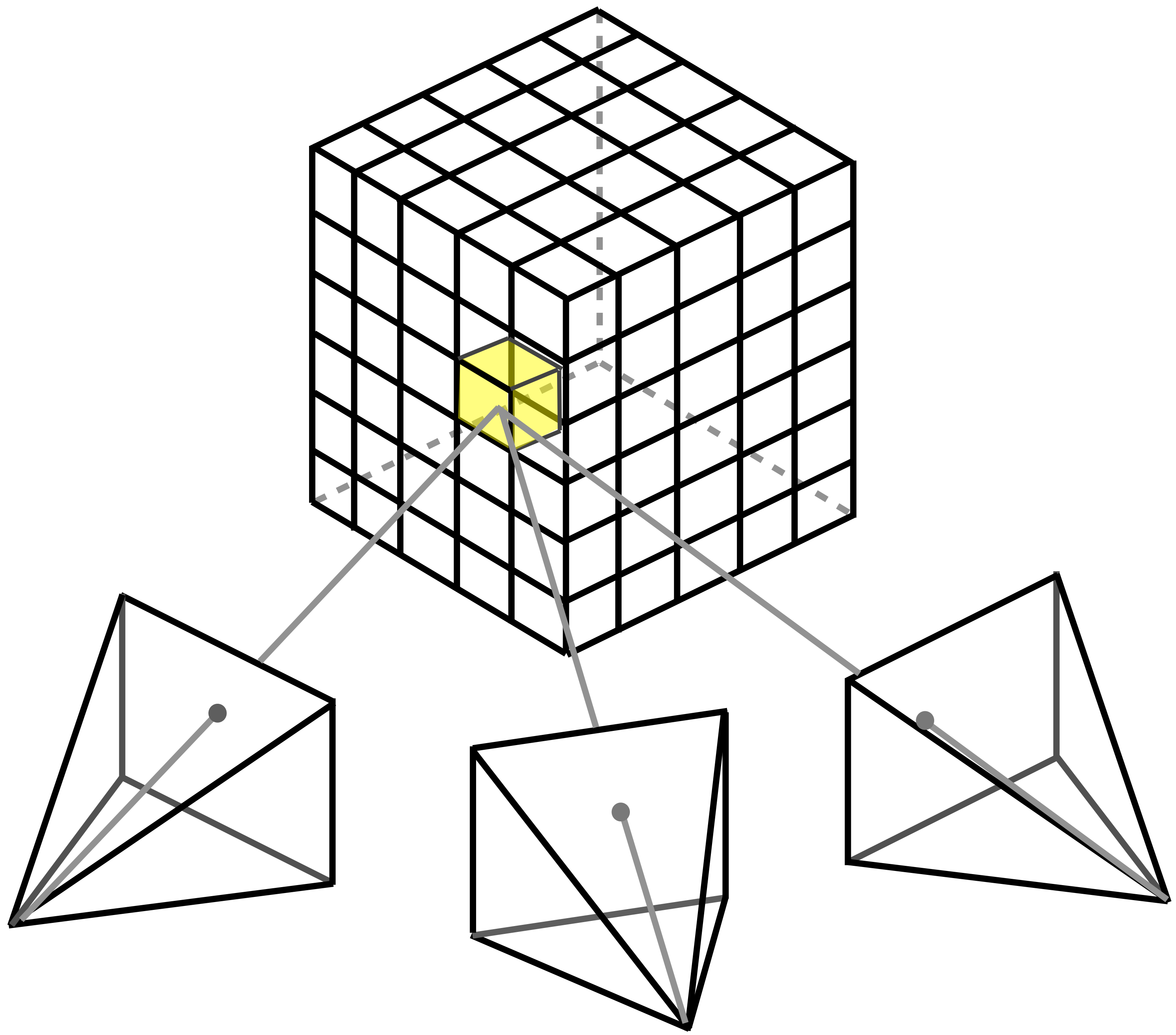
Plane Sweep Stereo

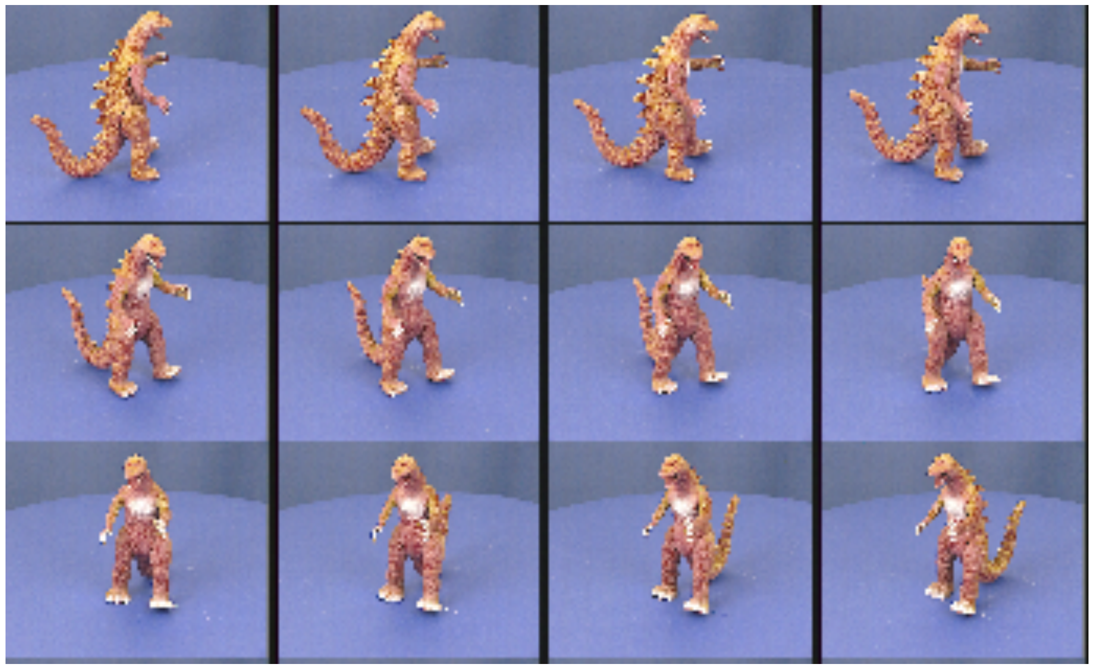
- Warp images using a set of planes in front of the camera

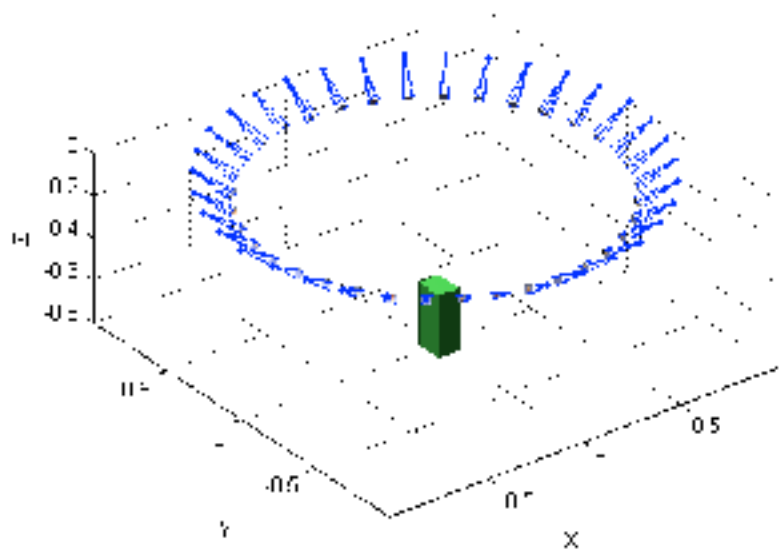
Try out `PlaneSweep.ipynb` from the course webpage

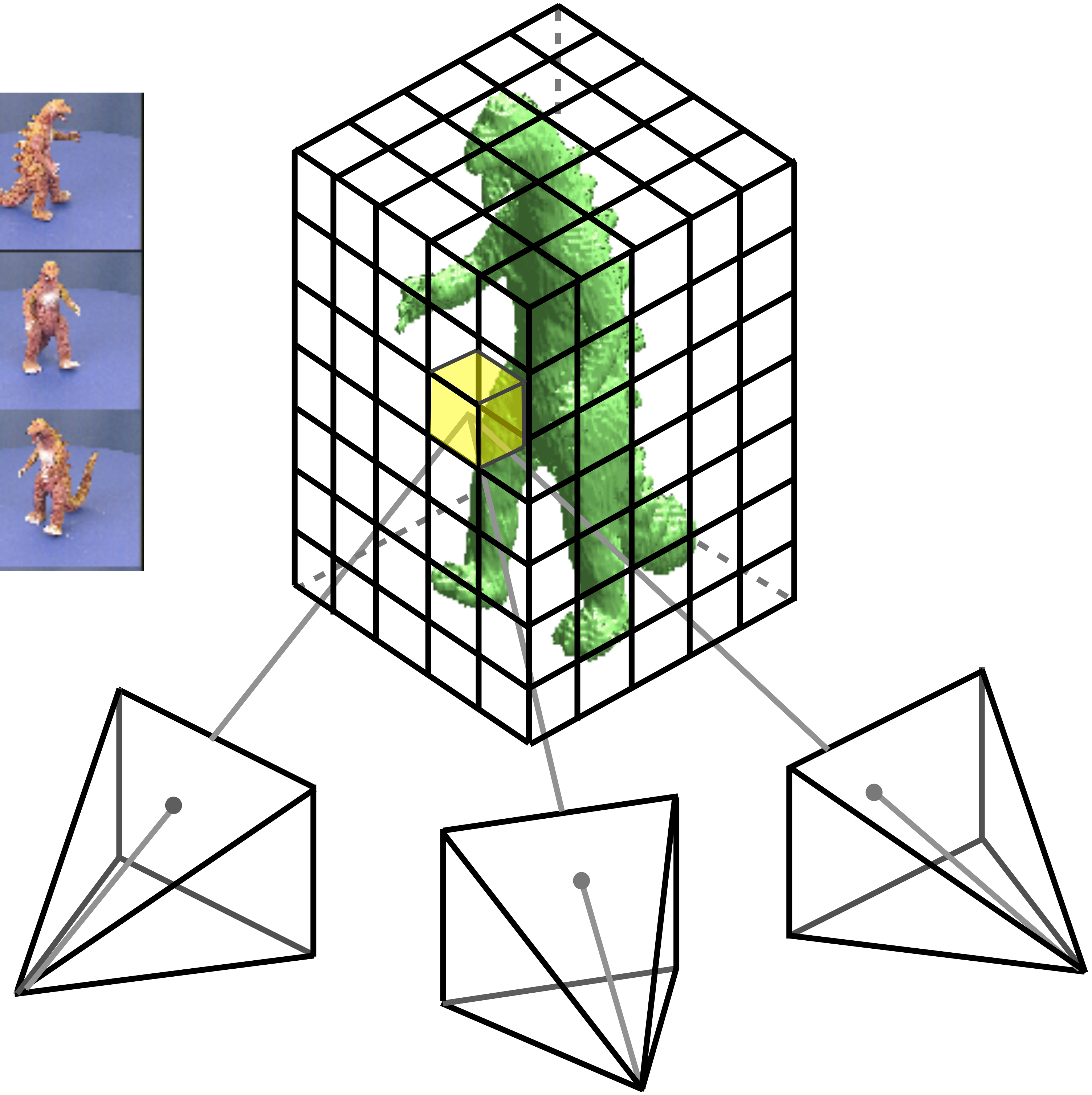
Volumetric Stereo

- Discretise the scene using a grid of **voxels**
- Infer occupancy and colour of voxels by projecting to images









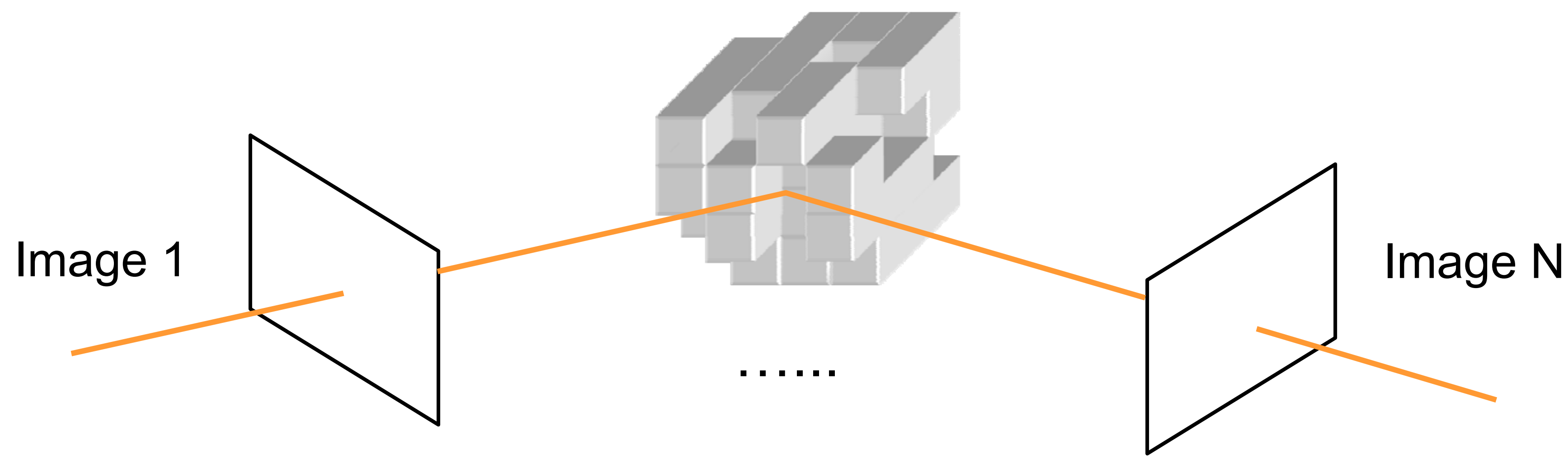
- Idea: visit all voxels in order, keep only photo-consistent voxels

What is wrong with this idea?

[Seitz Dyer 1997]

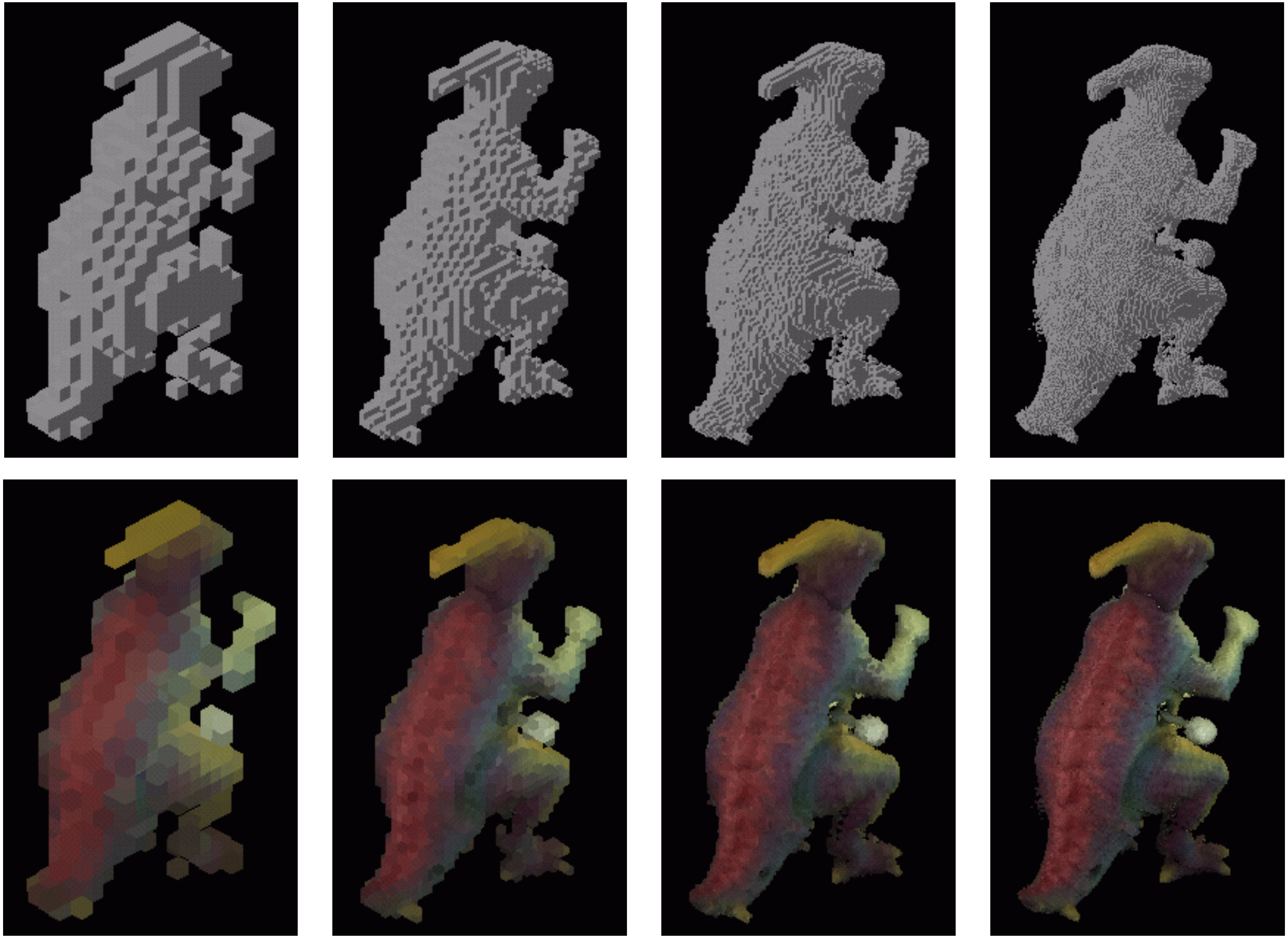
Space Carving

- Space carving finds a voxel reconstruction that is consistent with the input images, taking into account visibility



- Initialise a volume containing the true scene
- Choose a voxel v on the surface
- Project v to all views where visible
- If v is not photo-consistent, remove it from the volume
- Repeat until all voxels are photo-consistent

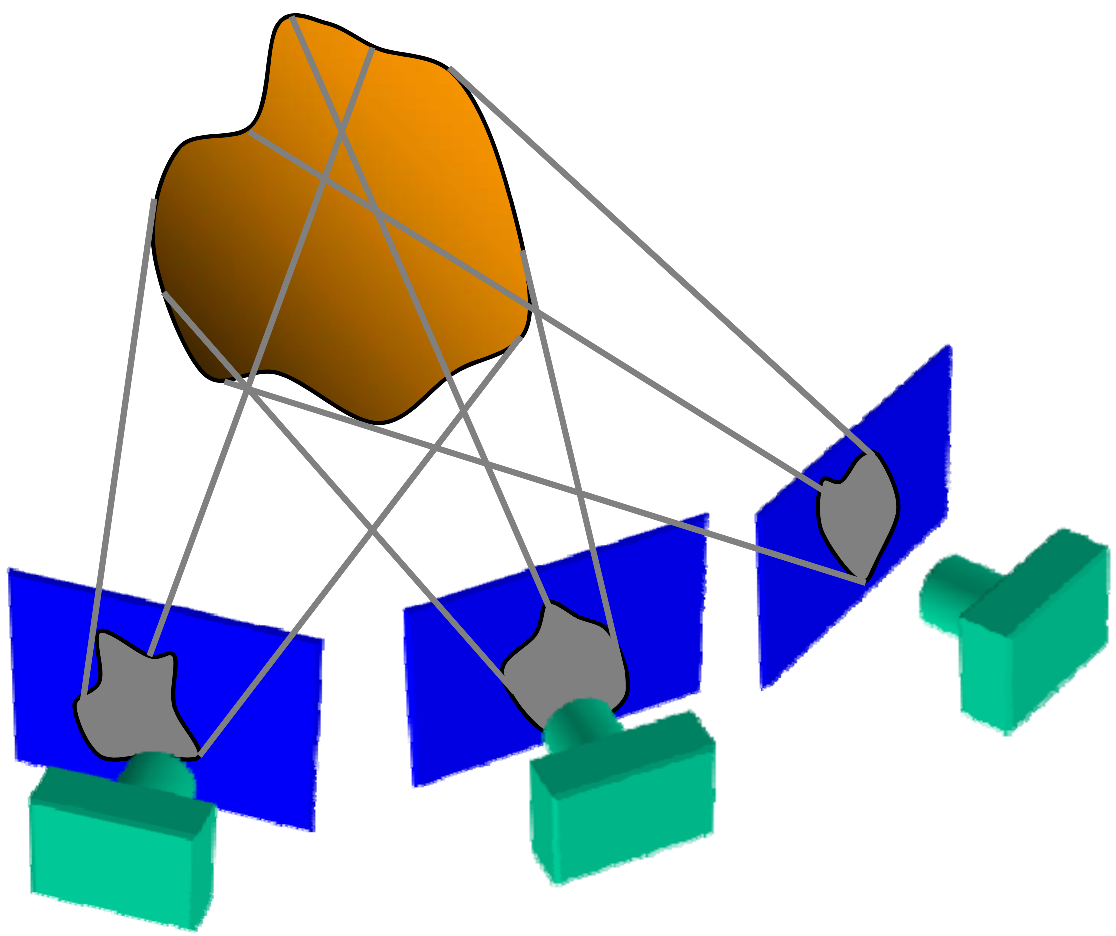
[Kutulakos Seitz 2000]



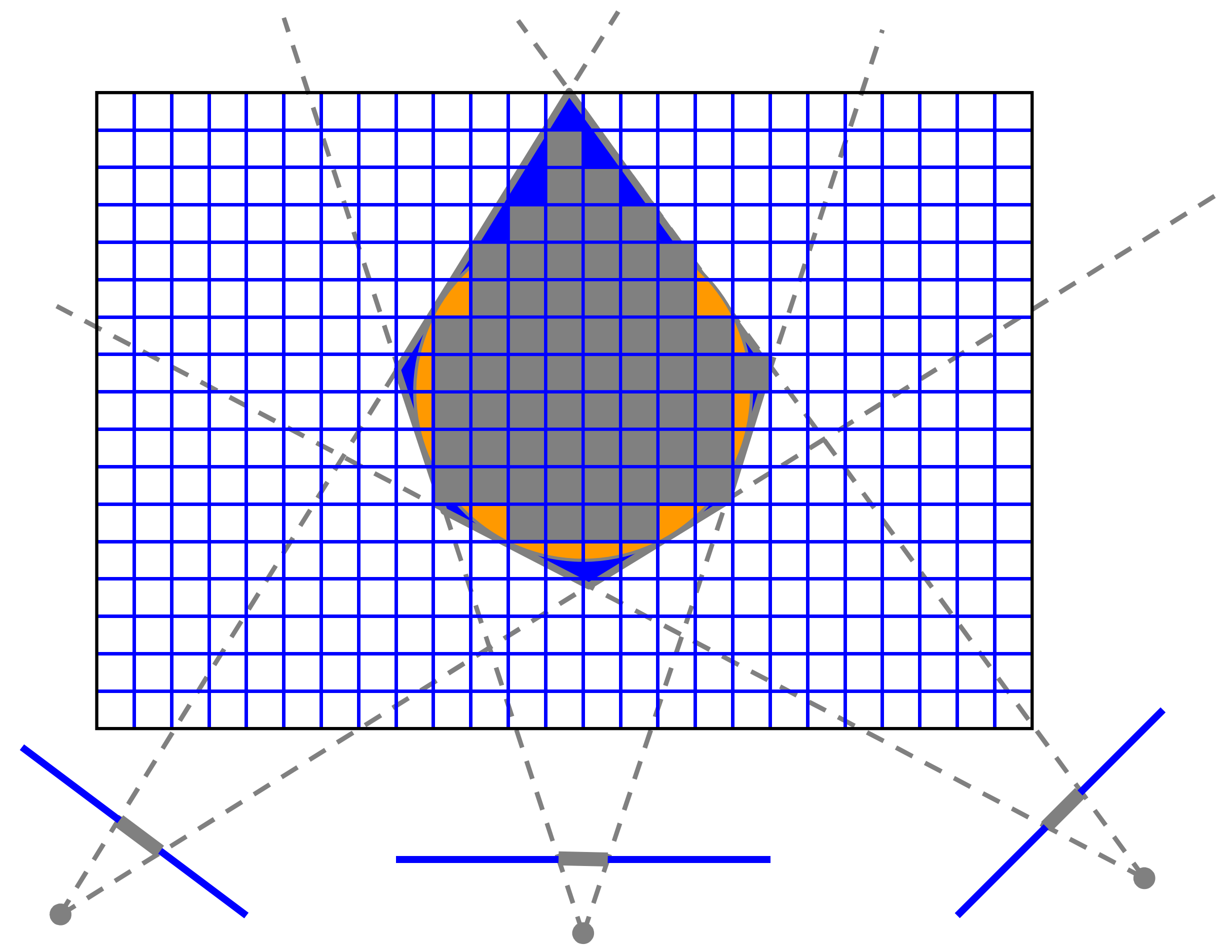
~1k voxels ~70k voxels

Silhouette Intersection

- Consider the case of binary images (silhouettes)
- Voxel is part of the object if it lies in the silhouette in all views



Project volumes from each silhouette back into scene and intersect

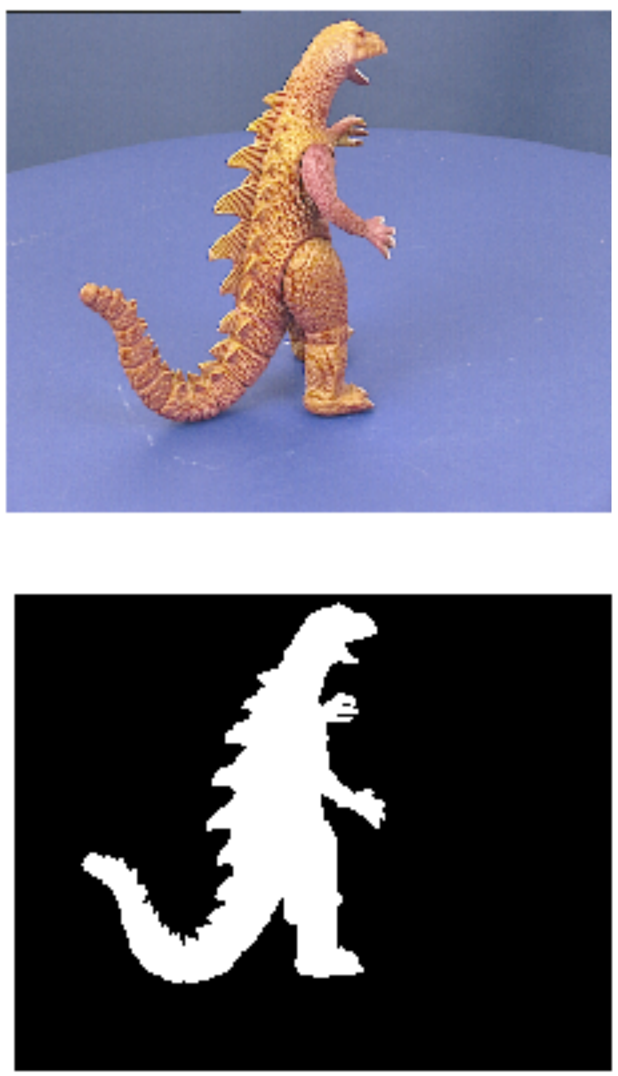


Voxel reconstruction is larger than object

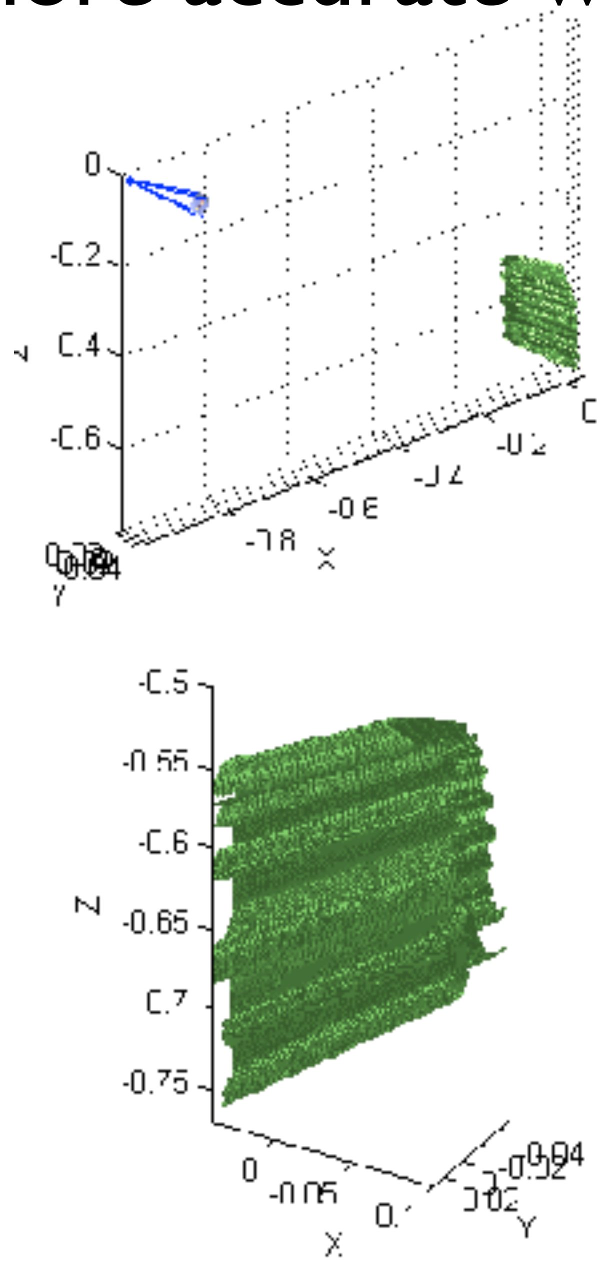
[Seitz / Lazebnik]

Silhouette Intersection

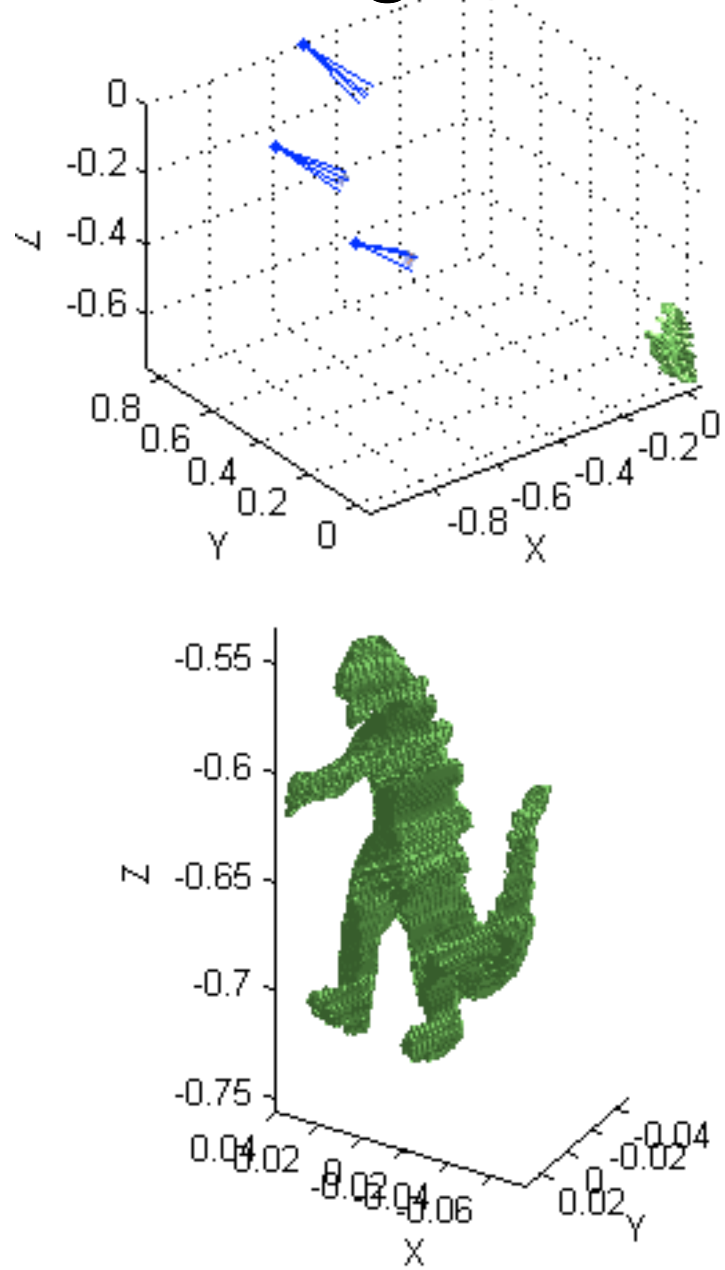
- The intersection of back-projected silhouettes is called the **visual hull**, it is more accurate with increasing # views



[Ben Tordoff / Mathworks]



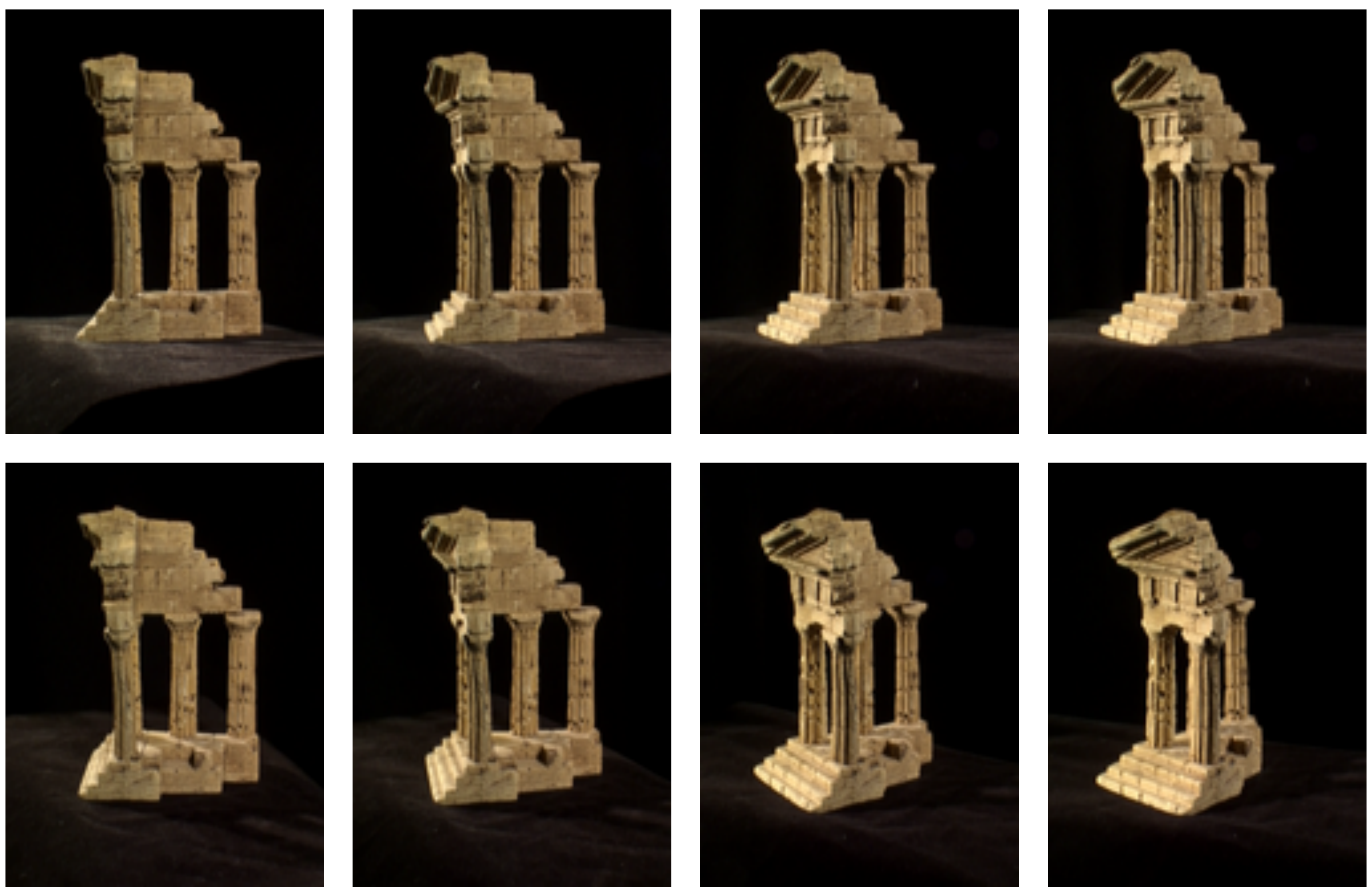
1 camera

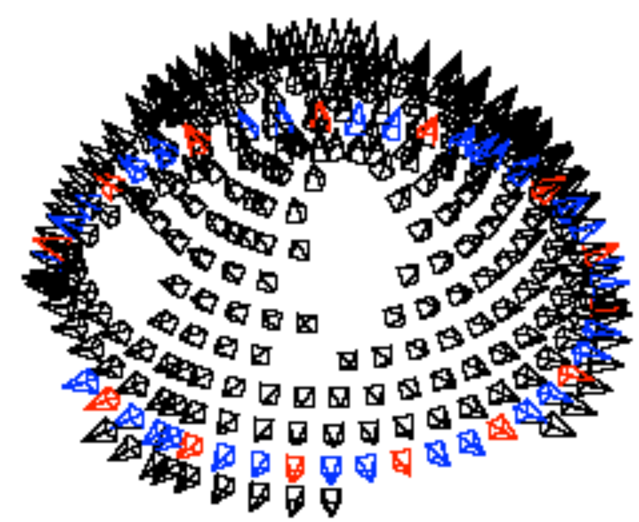


3 cameras

Depth Map Merging

- Idea: Nearby images have the most reliable stereo matches
- If we have a lot of images/pixels, we may not need to perform wide baseline matching

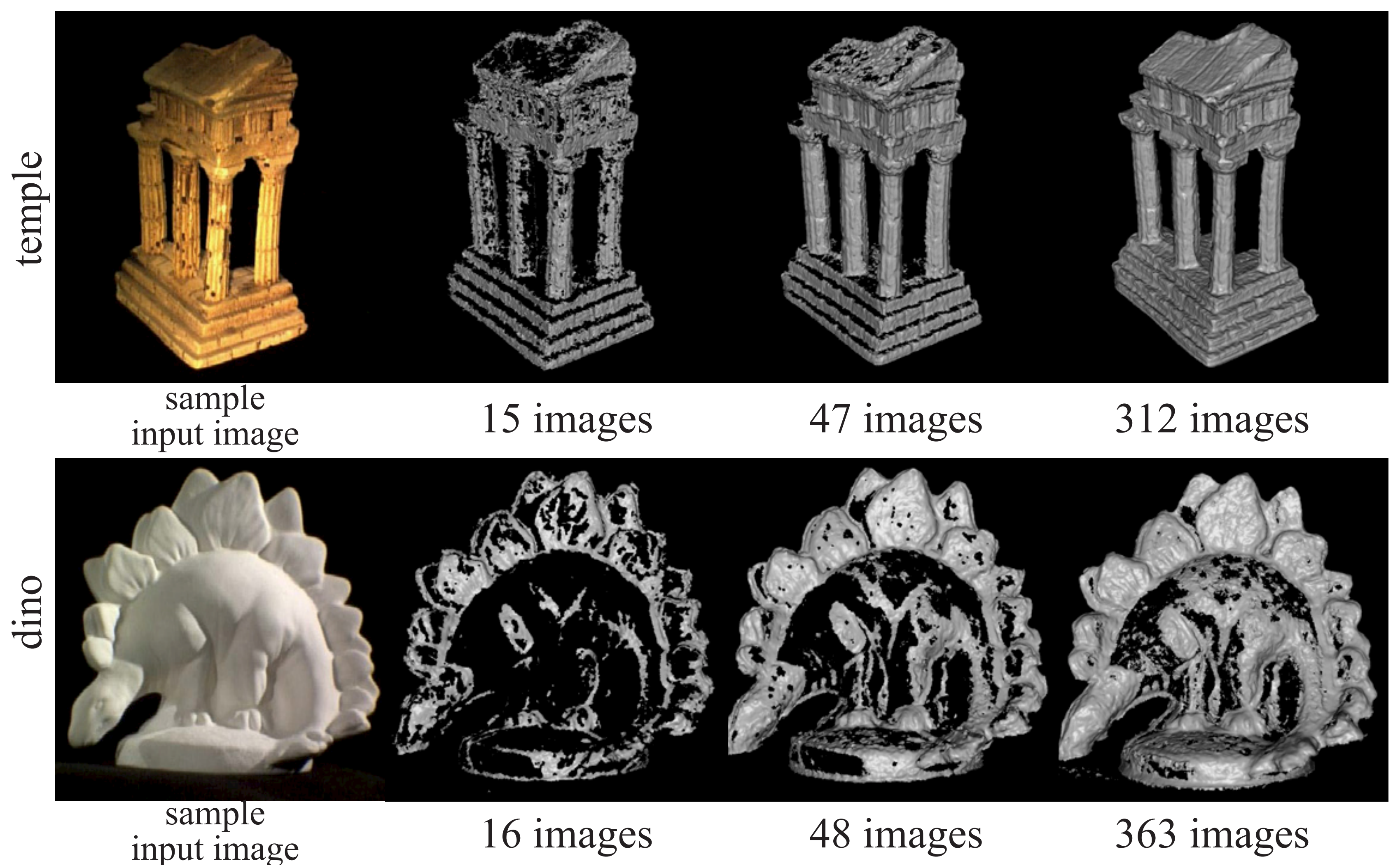




[http://vision.middlebury.edu/mview]

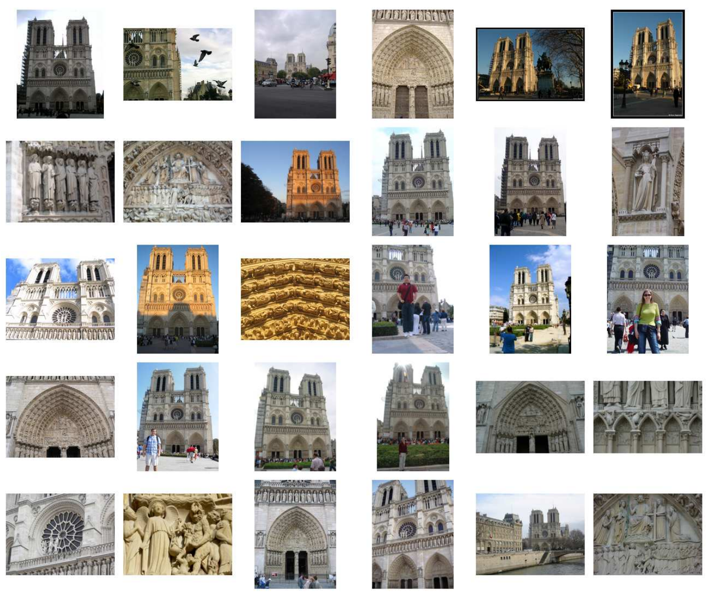
Depth Map Merging

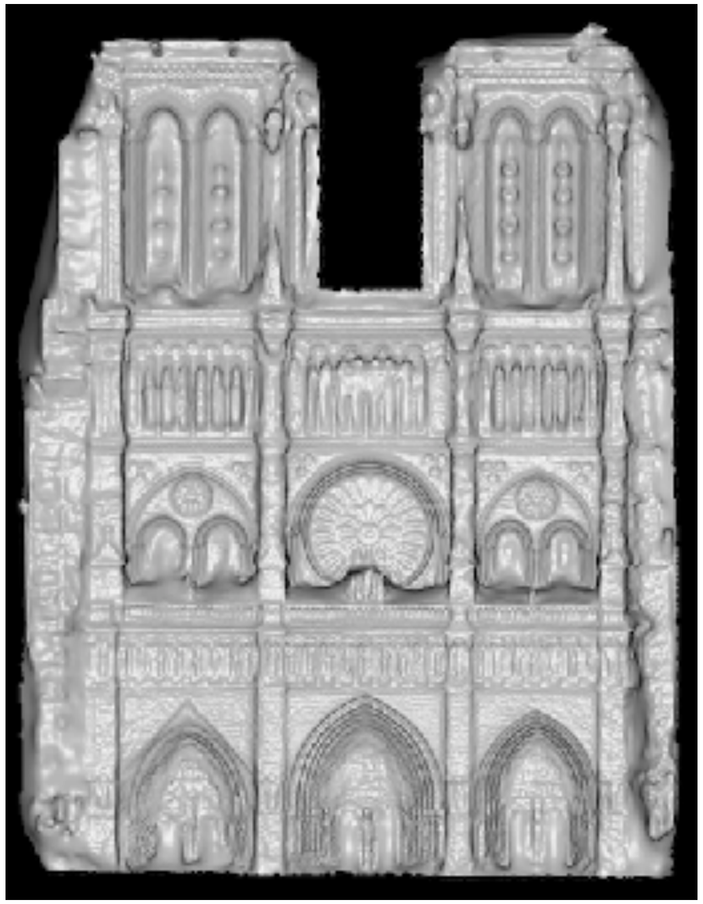
- Select subsets of images and compute high confidence depth maps (e.g., keep only low SSD matches)
- Merge depth maps using robust fusion, e.g., using signed distance functions [Curless Levoy 1996]



[Goesele Curless Seitz 2006]

Photo Collections→3D





- Depth map merging is practical for photo collections:
- Adaptable to complex geometry and large-scale scenes
- Robust to varied imagery and noise — select only subsets with good matches (don't try to match everything)

[N. Snavely, M. Goesele]

Neural Scene Representation

- Neural Radiance Fields, ~10s of input views

matthewtancik.com/nerf

Photometric Stereo

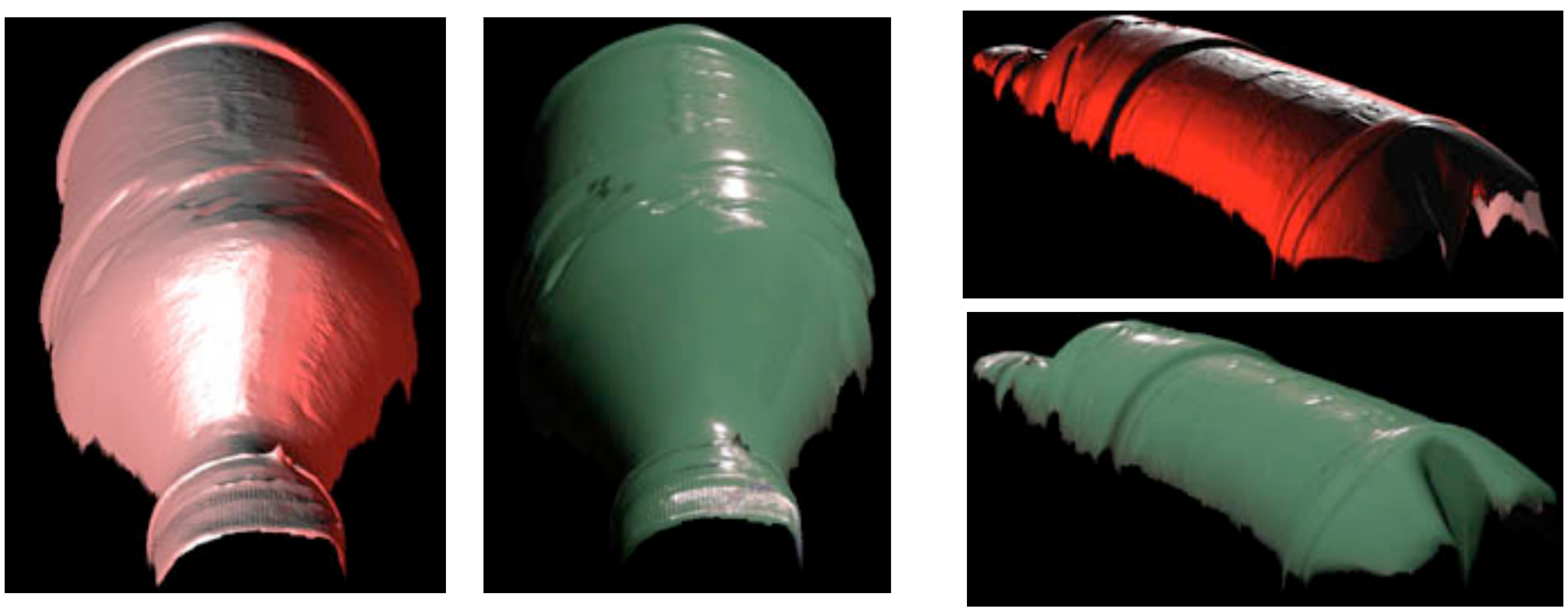
- We can also get 3D information about the scene using one camera and **multiple lights**

- The most straightforward case of photometric stereo is to assume Lambertian reflectance

5.5

Photometric Stereo by Example

- Use object of known geometry, match colour patterns



[Hertzmann Seitz 2003]

Non-rigid Photometric Stereo with Colored Lights

C. Hernández[1], G. Vogiatzis[1], G.J. Brostow[2], B. Stenger[1] and R. Cipolla[2]

Toshiba Research Cambridge[1]

University of Cambridge[2]

Next Lecture

- Depth, Flow