

# Computer Vision

CS-E4850, 5 study credits

Lecturer: Juho Kannala

# Lecture 12: Structure from motion & multi-view stereo

- **Structure from motion** is the art of solving both the camera motion and sparse 3D structure of the scene from multiple (uncalibrated) images
- **Multi-view stereo** provides techniques for computing a complete and dense 3D scene reconstruction from multiple images (with known projection matrices)

**Acknowledgement:** many slides from Svetlana Lazebnik, Steve Seitz, Noah Snavely, and others

# Structure from motion

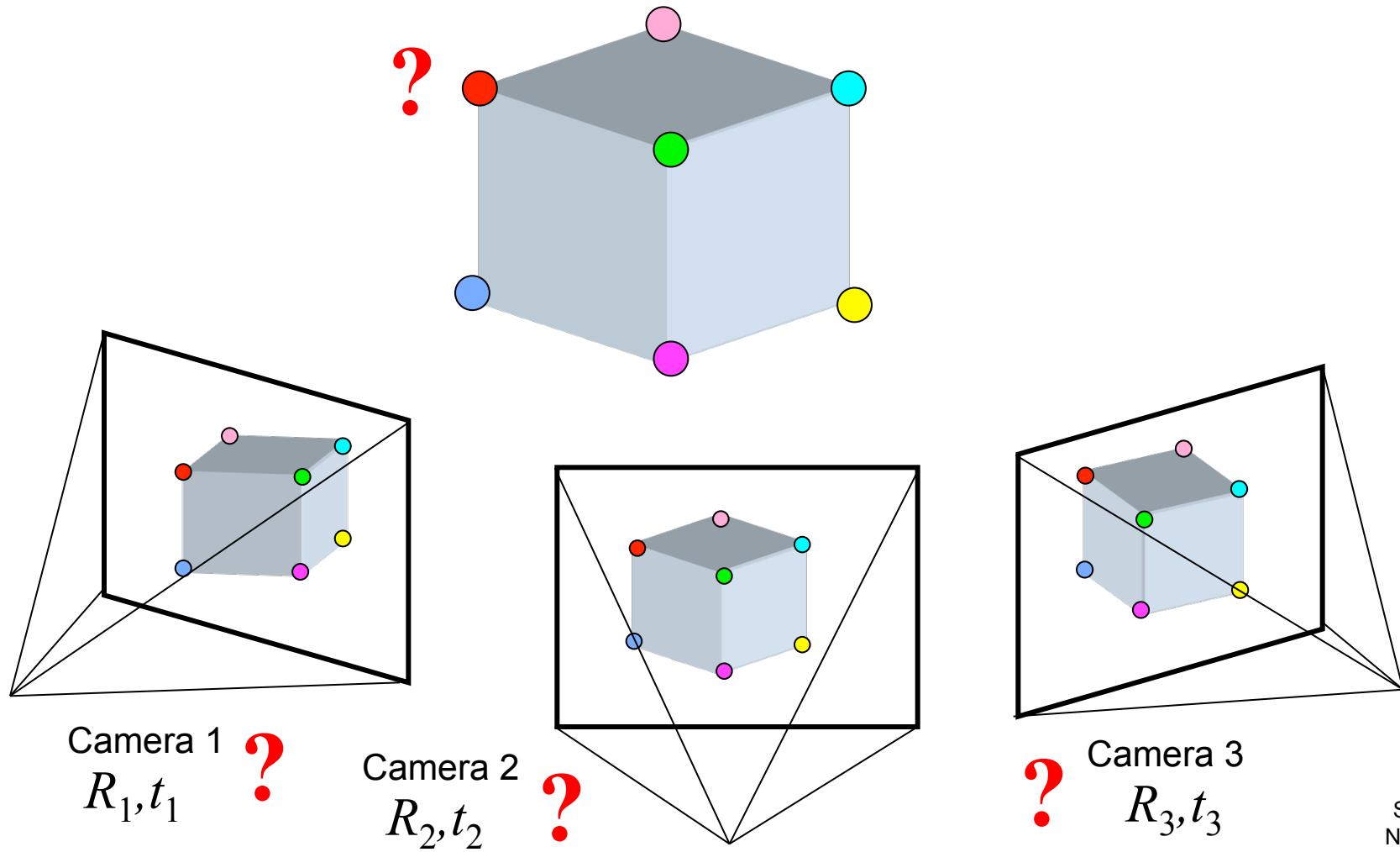
---



Драконъ, видимый подъ различными углами зренія  
По гравюре изъ изд. изъ „Oculus artificialis teledioptricus“ Гана. 1702 года.

# Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



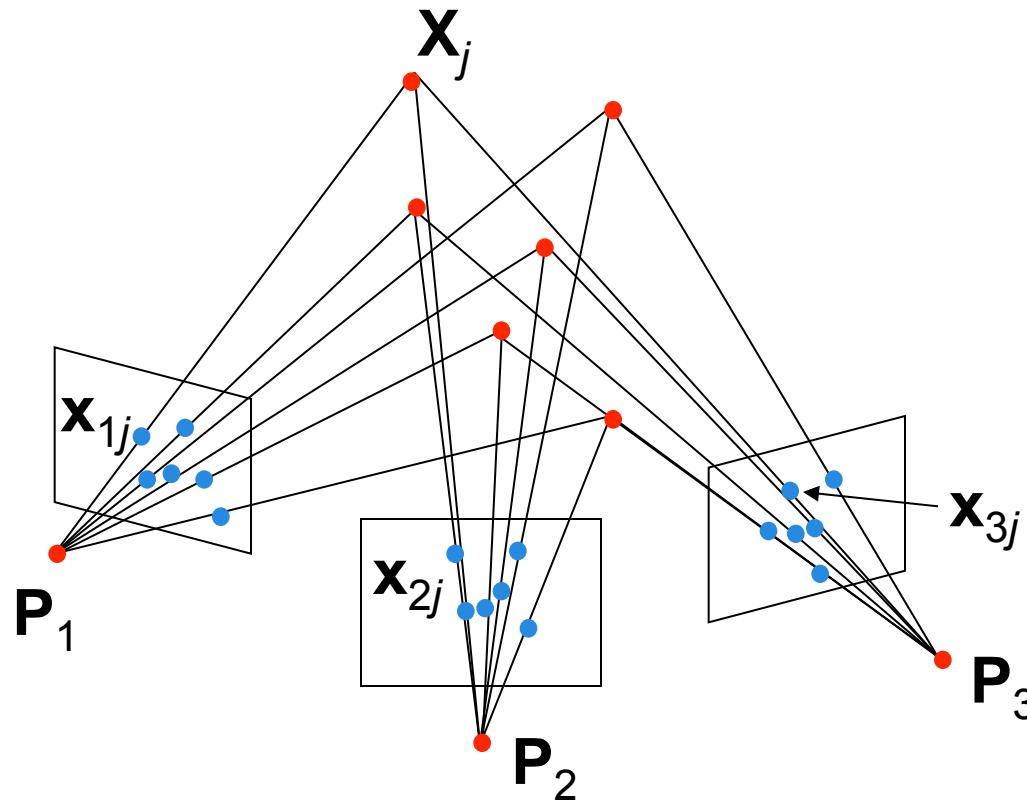
# Structure from motion

---

- Given:  $m$  images of  $n$  fixed 3D points

$$\lambda_{ij} \mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate  $m$  projection matrices  $\mathbf{P}_i$  and  $n$  3D points  $\mathbf{X}_j$  from the  $mn$  correspondences  $\mathbf{x}_{ij}$



# Structure from motion ambiguity

---

- If we scale the entire scene by some factor  $k$  and, at the same time, scale the camera matrices by the factor of  $1/k$ , the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left( \frac{1}{k} \mathbf{P} \right) (k\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

# Structure from motion ambiguity

---

- If we scale the entire scene by some factor  $k$  and, at the same time, scale the camera matrices by the factor of  $1/k$ , the projections of the scene points in the image remain exactly the same
- More generally, if we transform the scene using a transformation  $\mathbf{Q}$  and apply the inverse transformation to the camera matrices, then the images do not change:

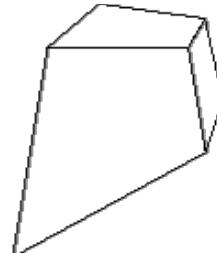
$$\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{Q}^{-1})(\mathbf{Q}\mathbf{X})$$

# Types of ambiguity

---

Projective  
15dof

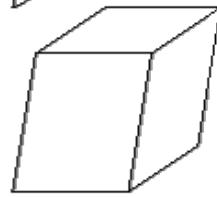
$$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$



Preserves intersection and tangency

Affine  
12dof

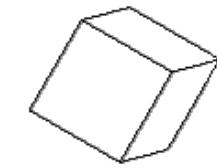
$$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$$



Preserves parallelism, volume ratios

Similarity  
7dof

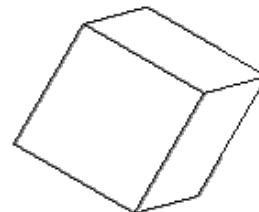
$$\begin{bmatrix} s R & t \\ 0^T & 1 \end{bmatrix}$$



Preserves angles, ratios of length

Euclidean  
6dof

$$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$



Preserves angles, lengths

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean

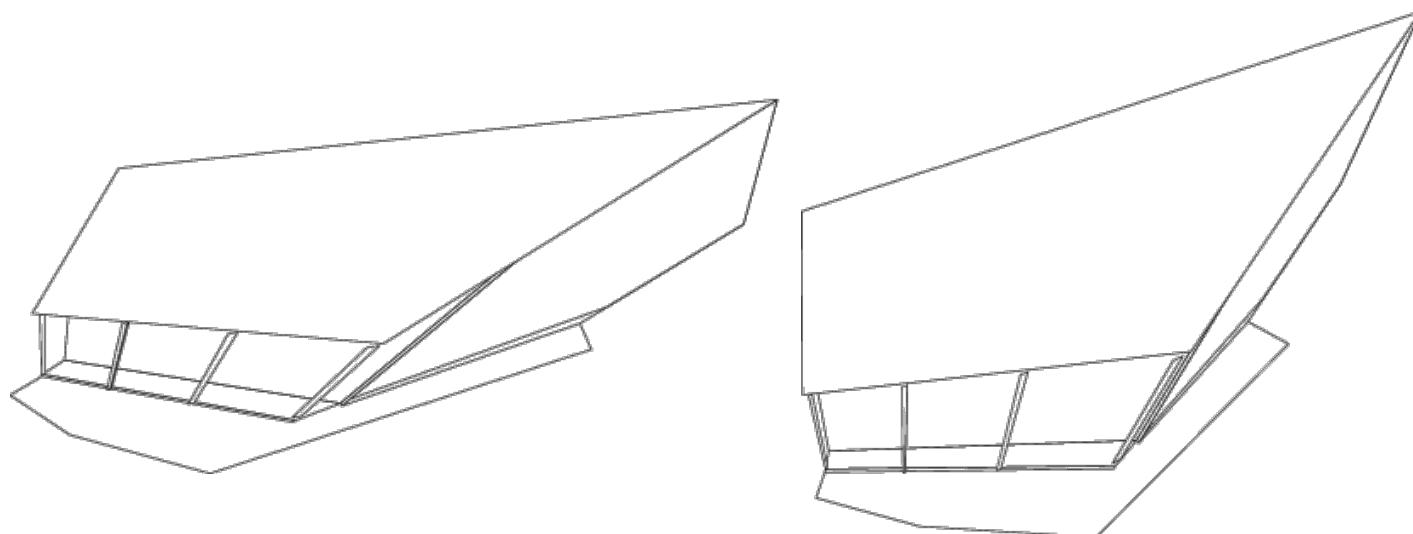
# Projective ambiguity

---

$$Q_p = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$
$$X = P X = (P Q_P^{-1})(Q_P X)$$

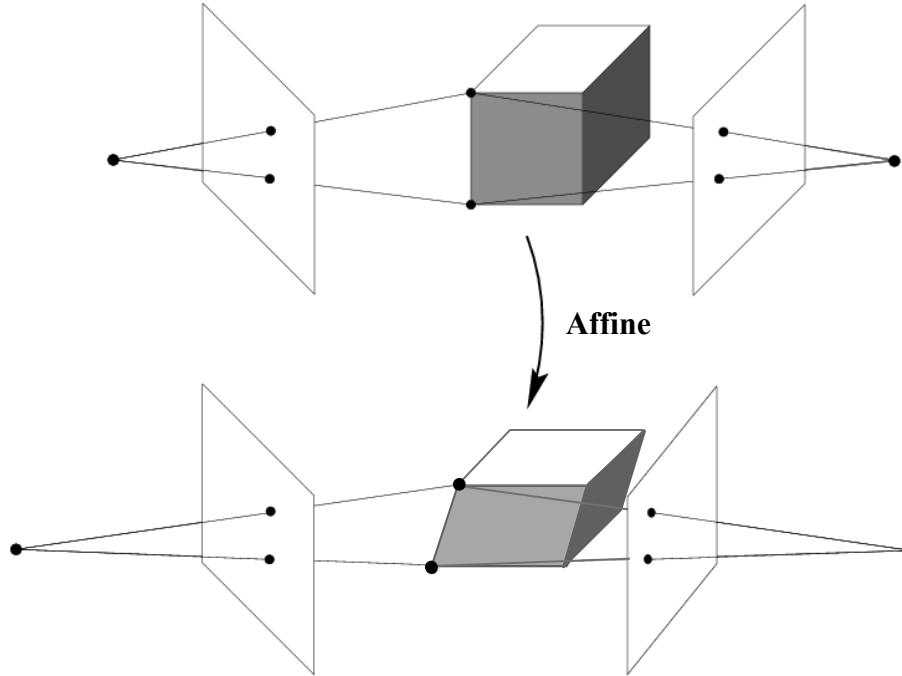
# Projective ambiguity

---



# Affine ambiguity

---

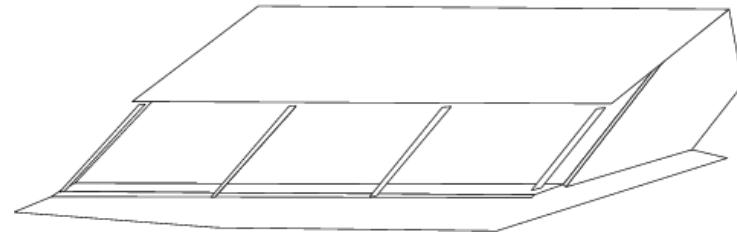
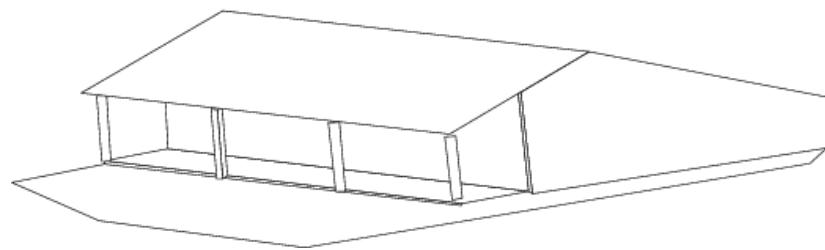


$$\mathbf{Q}_A = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{P}\mathbf{x} = (\mathbf{P}\mathbf{Q}_A^{-1})(\mathbf{Q}_A \mathbf{x})$$

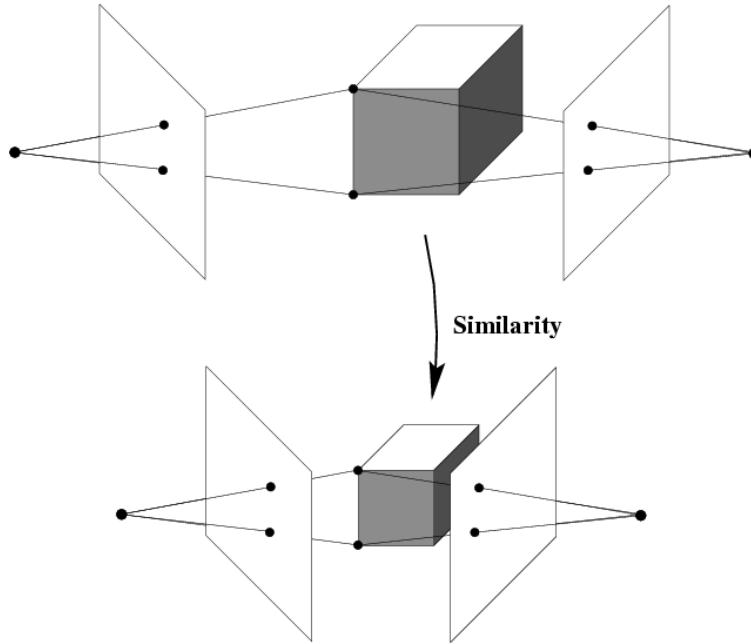
# Affine ambiguity

---



# Similarity ambiguity

---

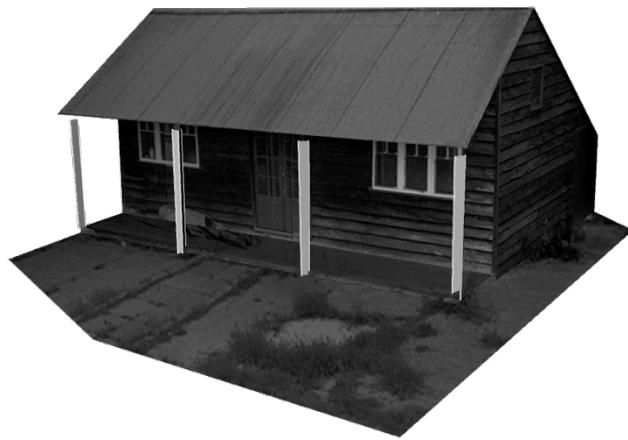
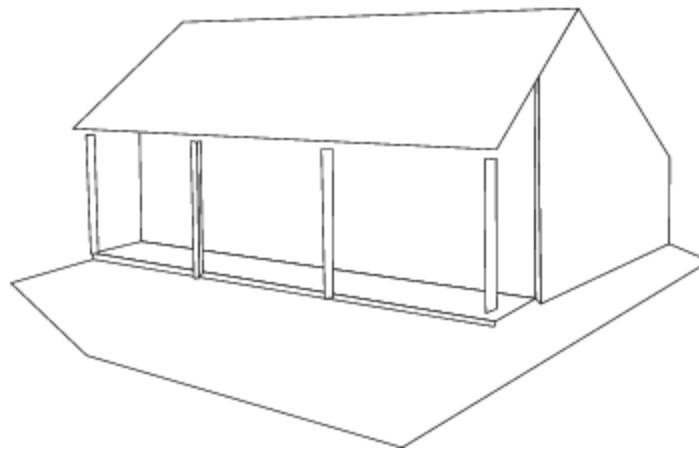
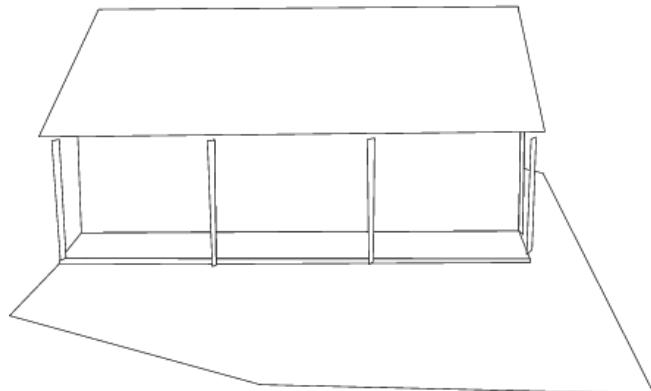


$$Q_s = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$$

$$x = Px = (PQ_s^{-1})(Q_s x)$$

# Similarity ambiguity

---



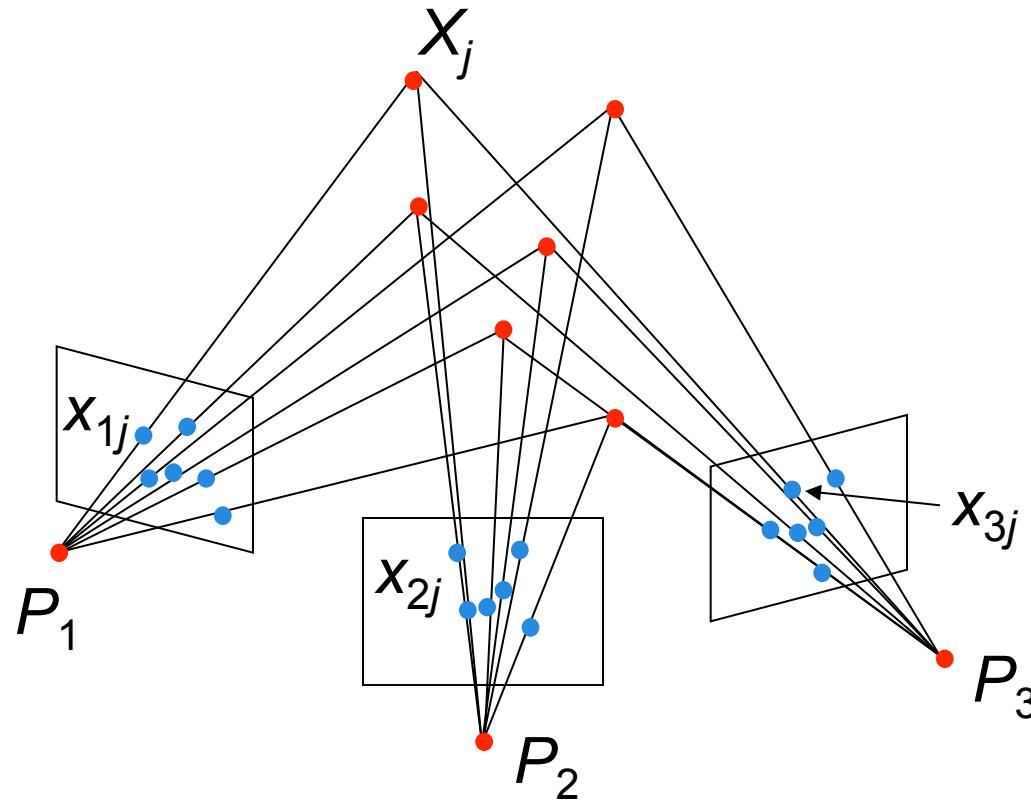
# Projective structure from motion

---

- Given:  $m$  images of  $n$  fixed 3D points

$$\lambda_{ij} \mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate  $m$  projection matrices  $\mathbf{P}_i$  and  $n$  3D points  $\mathbf{X}_j$  from the  $mn$  correspondences  $\mathbf{x}_{ij}$



# Projective structure from motion

---

- Given:  $m$  images of  $n$  fixed 3D points

$$\lambda_{ij} \mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate  $m$  projection matrices  $\mathbf{P}_i$  and  $n$  3D points  $\mathbf{X}_j$  from the  $mn$  correspondences  $\mathbf{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a  $4 \times 4$  projective transformation  $\mathbf{Q}$ :

$$\mathbf{X} \rightarrow \mathbf{Q}\mathbf{X}, \mathbf{P} \rightarrow \mathbf{P}\mathbf{Q}^{-1}$$

- We can solve for structure and motion when

$$2mn \geq 11m + 3n - 15$$

- For two cameras, at least 7 points are needed

# Projective SFM: Two-camera case

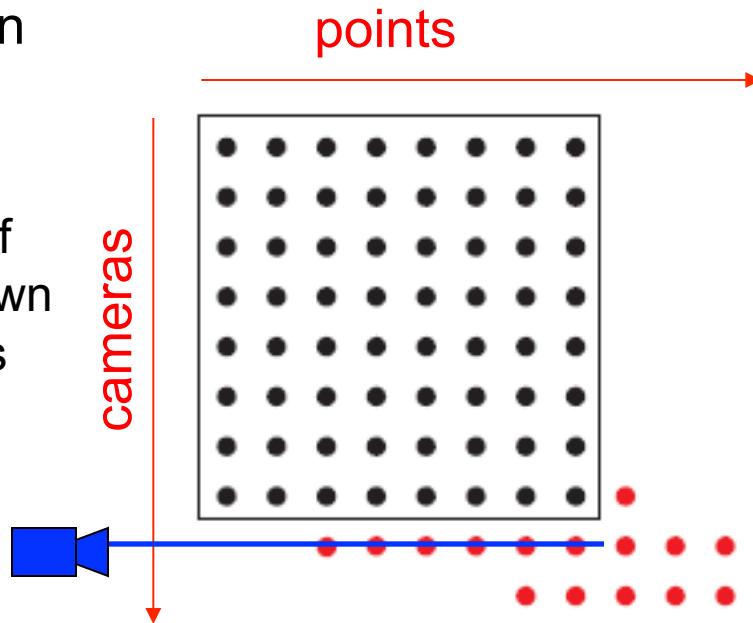
---

- Compute fundamental matrix  $\mathbf{F}$  between the two views
- First camera matrix:  $[\mathbf{I} \mid \mathbf{0}]$
- Second camera matrix:  $[\mathbf{A} \mid \mathbf{b}]$
- Then  $\mathbf{b}$  is the epipole ( $\mathbf{F}^T \mathbf{b} = 0$ ),  $\mathbf{A} = -[\mathbf{b}_\times] \mathbf{F}$

# Sequential structure from motion

---

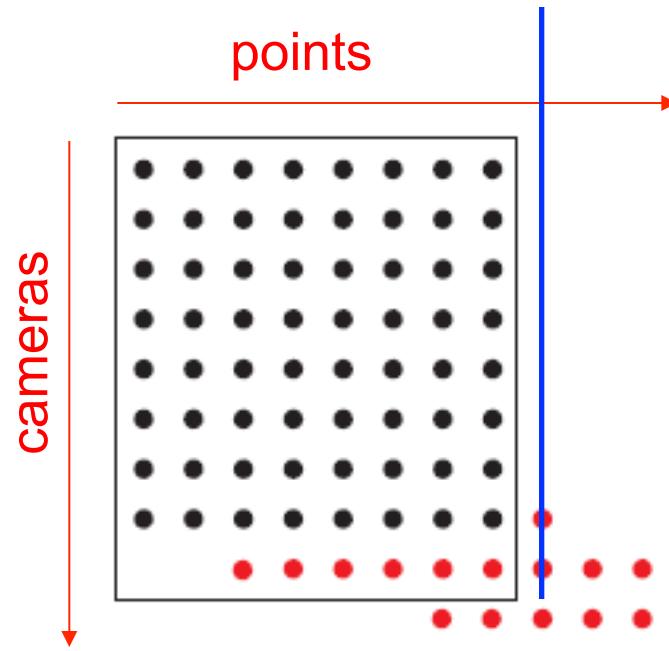
- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*



# Sequential structure from motion

---

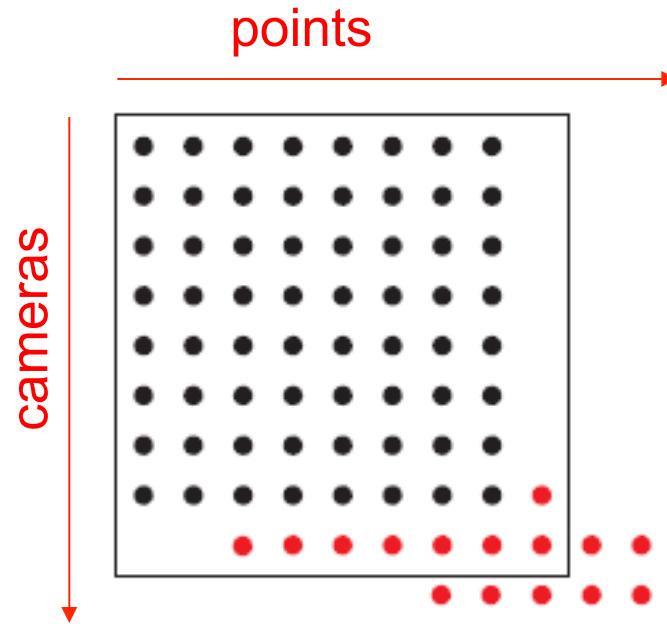
- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*



# Sequential structure from motion

---

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*
- Refine structure and motion: bundle adjustment



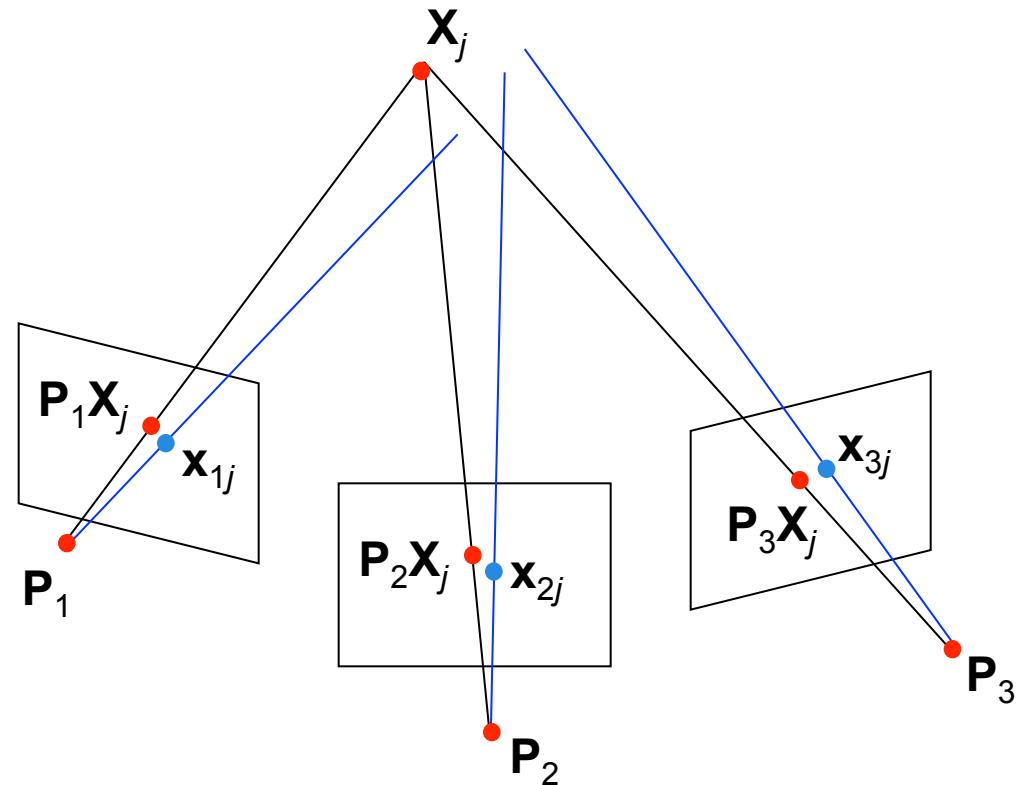
# Bundle adjustment

---

- Non-linear method for refining structure and motion
- Minimize reprojection error

$$\sum_{i=1}^m \sum_{j=1}^n w_{ij} \left\| \mathbf{x}_{ij} - \frac{1}{\lambda_{ij}} \mathbf{P}_i \mathbf{X}_j \right\|^2$$

visibility flag:  
is point j  
visible in  
view i?



# Self-calibration

---

- Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images
- For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images
  - Compute initial projective reconstruction and find 3D projective transformation matrix  $\mathbf{Q}$  such that all camera matrices are in the form  $\mathbf{P}_i = \mathbf{K} [\mathbf{R}_i | \mathbf{t}_i]$
- Can use constraints on the form of the calibration matrix: zero skew
- Can use vanishing points

# Modern SFM pipeline

---

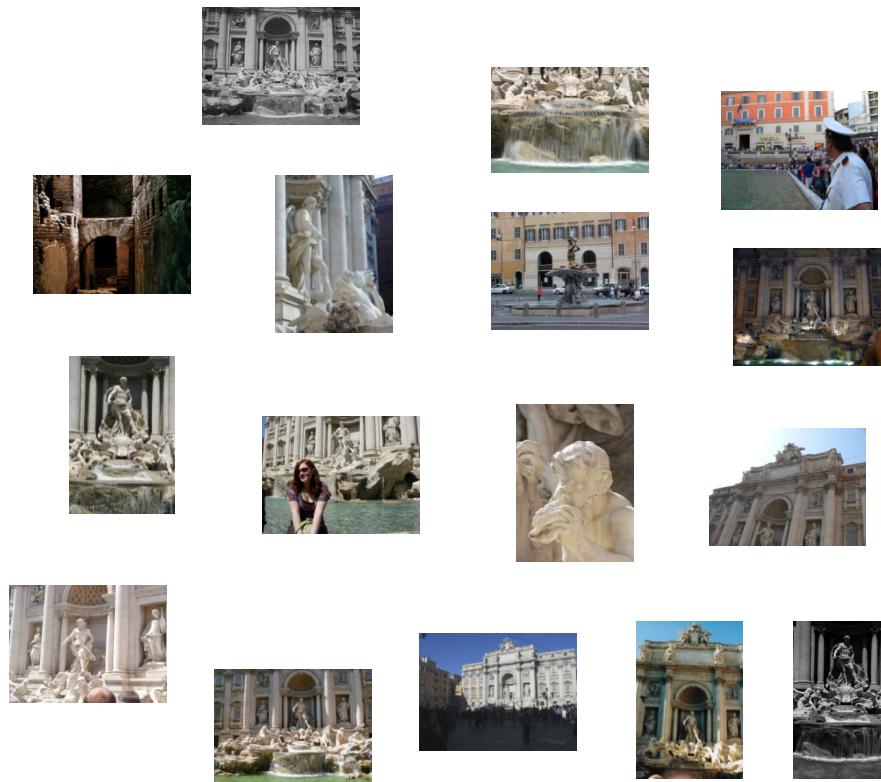


N. Snavely, S. Seitz, and R. Szeliski, "[Photo tourism: Exploring photo collections in 3D](#),"  
SIGGRAPH 2006.

# Feature detection

---

Detect features using SIFT [Lowe, IJCV 2004]



# Feature detection

---

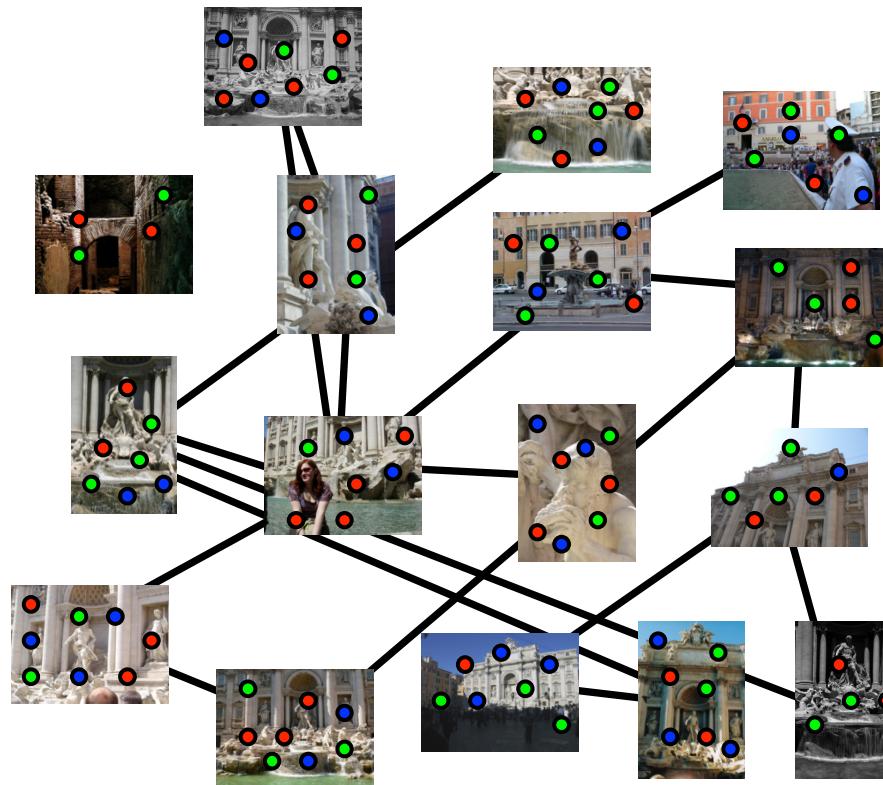
Detect features using SIFT



# Feature matching

---

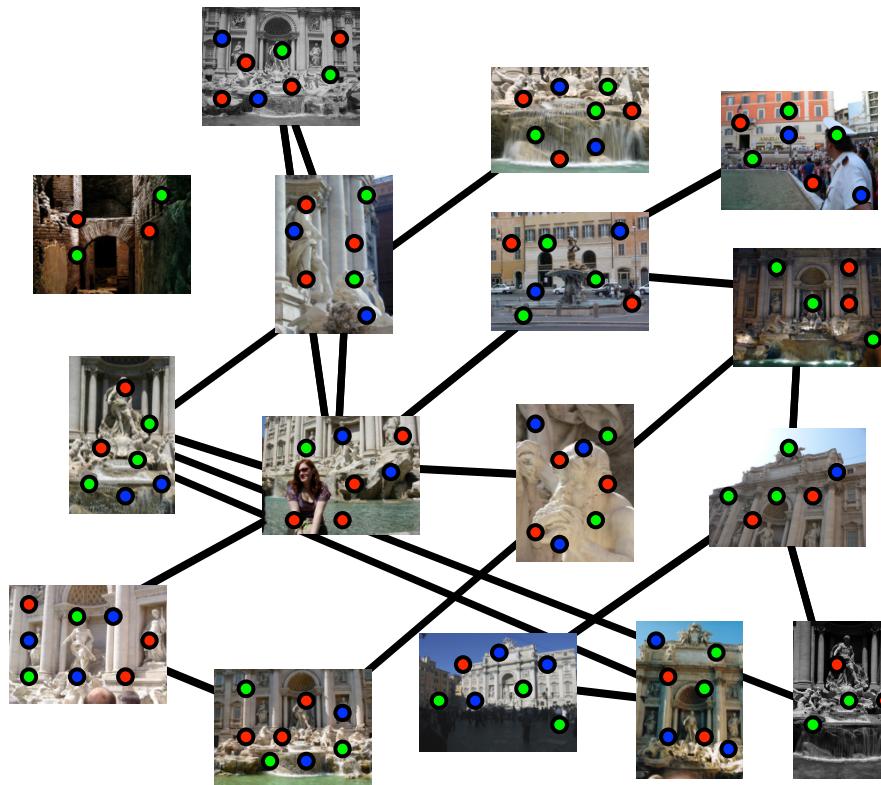
Match features between each pair of images



# Feature matching

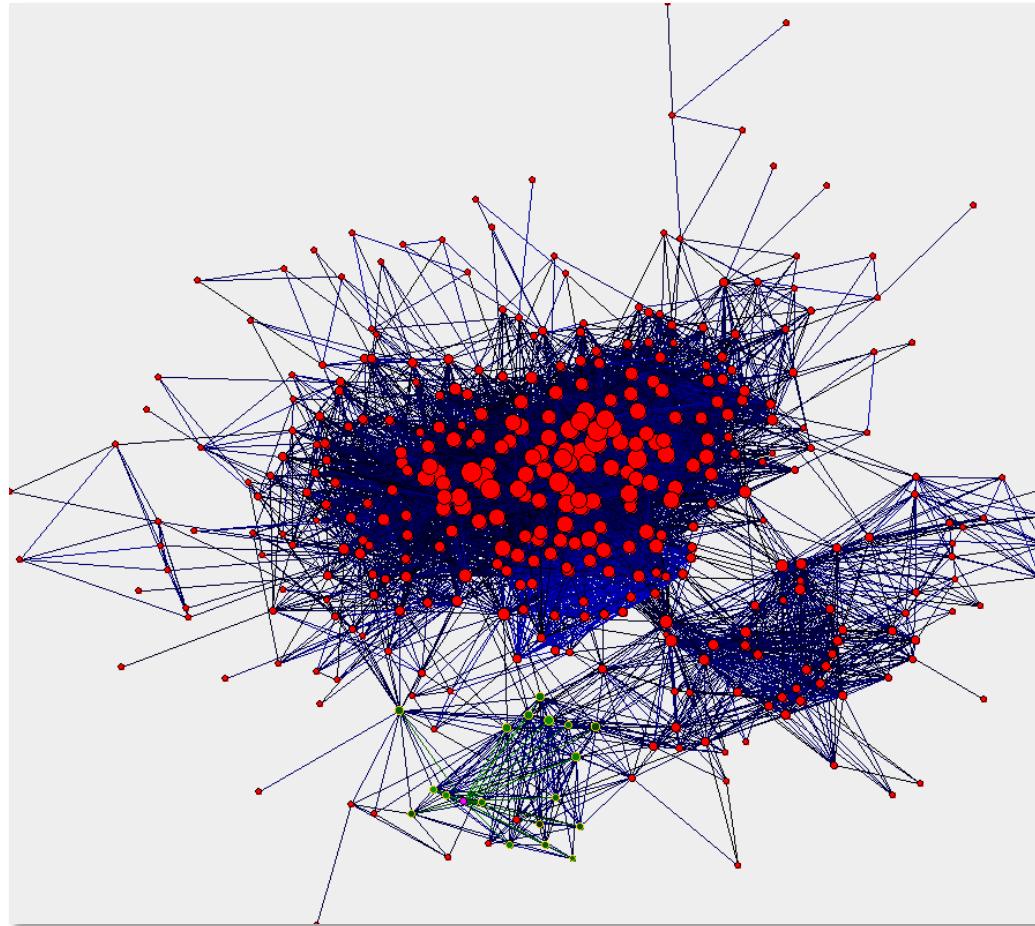
---

Use RANSAC to estimate fundamental matrix between each pair



# Image connectivity graph

---



(graph layout produced using the Graphviz toolkit: <http://www.graphviz.org/>)

# Incremental SFM

---

- Pick a pair of images with lots of inliers (and preferably, good EXIF data)
  - Initialize intrinsic parameters (focal length, principal point) from EXIF
  - Estimate extrinsic parameters ( $\mathbf{R}$  and  $\mathbf{t}$ )
    - [Five-point algorithm](#)
  - Use triangulation to initialize model points
- While remaining images exist
  - Find an image with many feature matches with images in the model
  - Run RANSAC on feature matches to register new image to model
  - Triangulate new points
  - Perform bundle adjustment to re-optimize everything

# The devil is in the details

---

- Handling degenerate configurations (e.g., homographies)
- Eliminating outliers
- Dealing with repetitions and symmetries
- Handling multiple connected components
- Closing loops
- ....

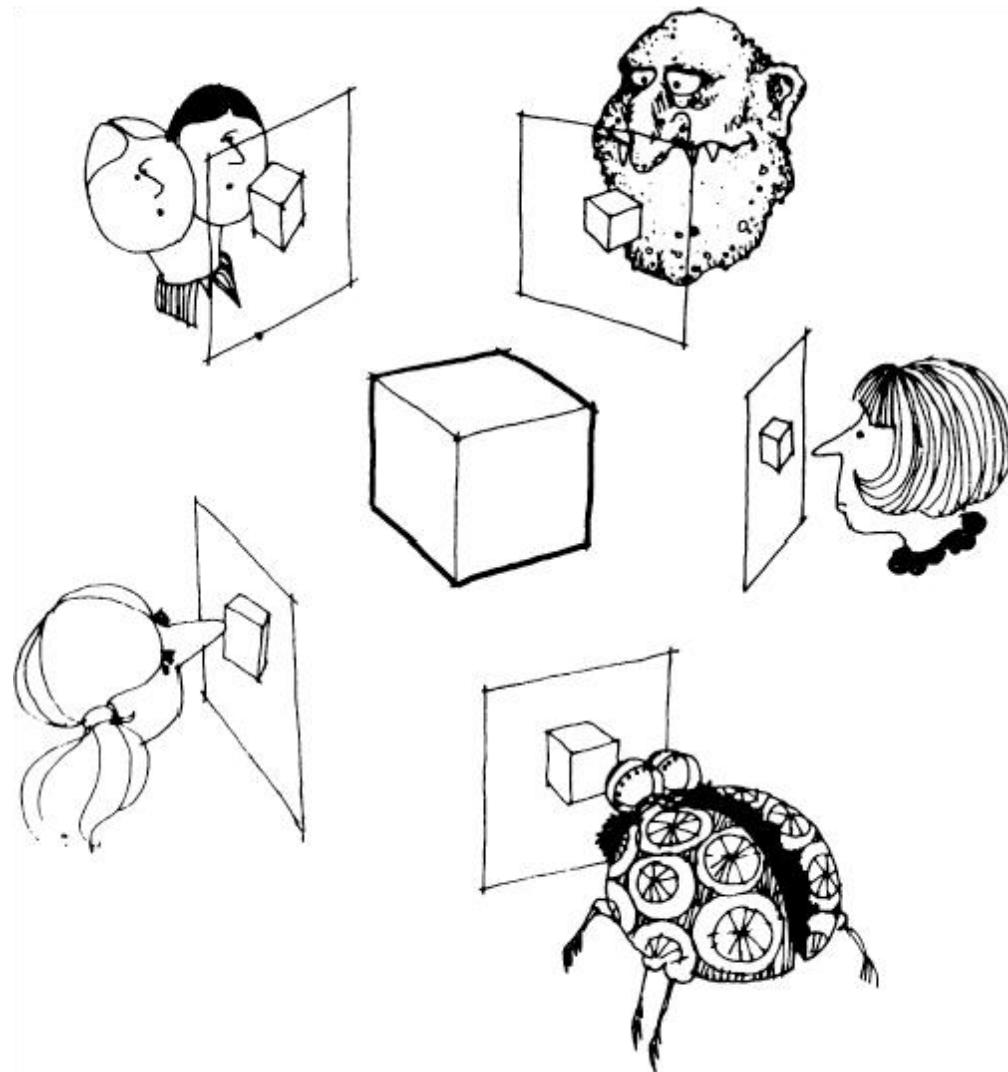
# Review: Structure from motion

---

- Ambiguity
- Projective structure from motion
  - Bundle adjustment
  - Modern structure from motion pipeline

# Multi-view stereo

---

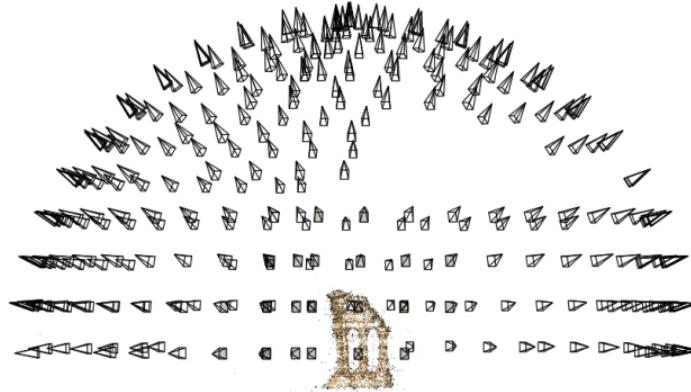


Many slides adapted from S. Seitz

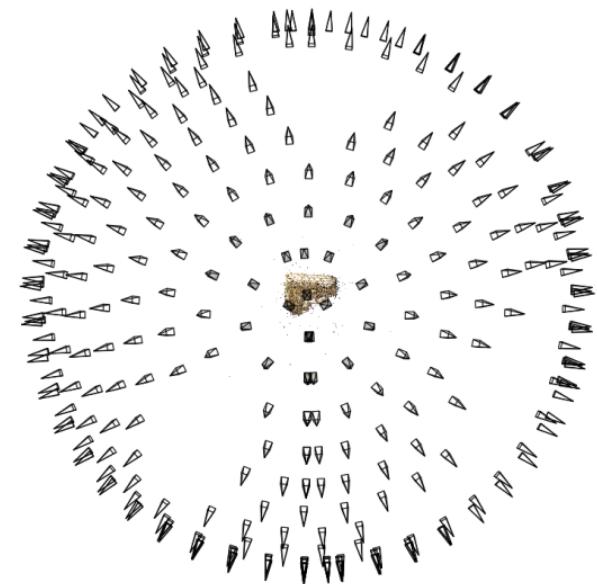
# Multi-view stereo

---

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape



Reconstruction (side)



(top)

# Multi-view stereo

---

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape
- “Images of the same object or scene”
  - Arbitrary number of images (from two to thousands)
  - Arbitrary camera positions (special rig, camera network or video sequence)
  - Camera projection matrices are assumed to be known (but internal calibration may be known or unknown)



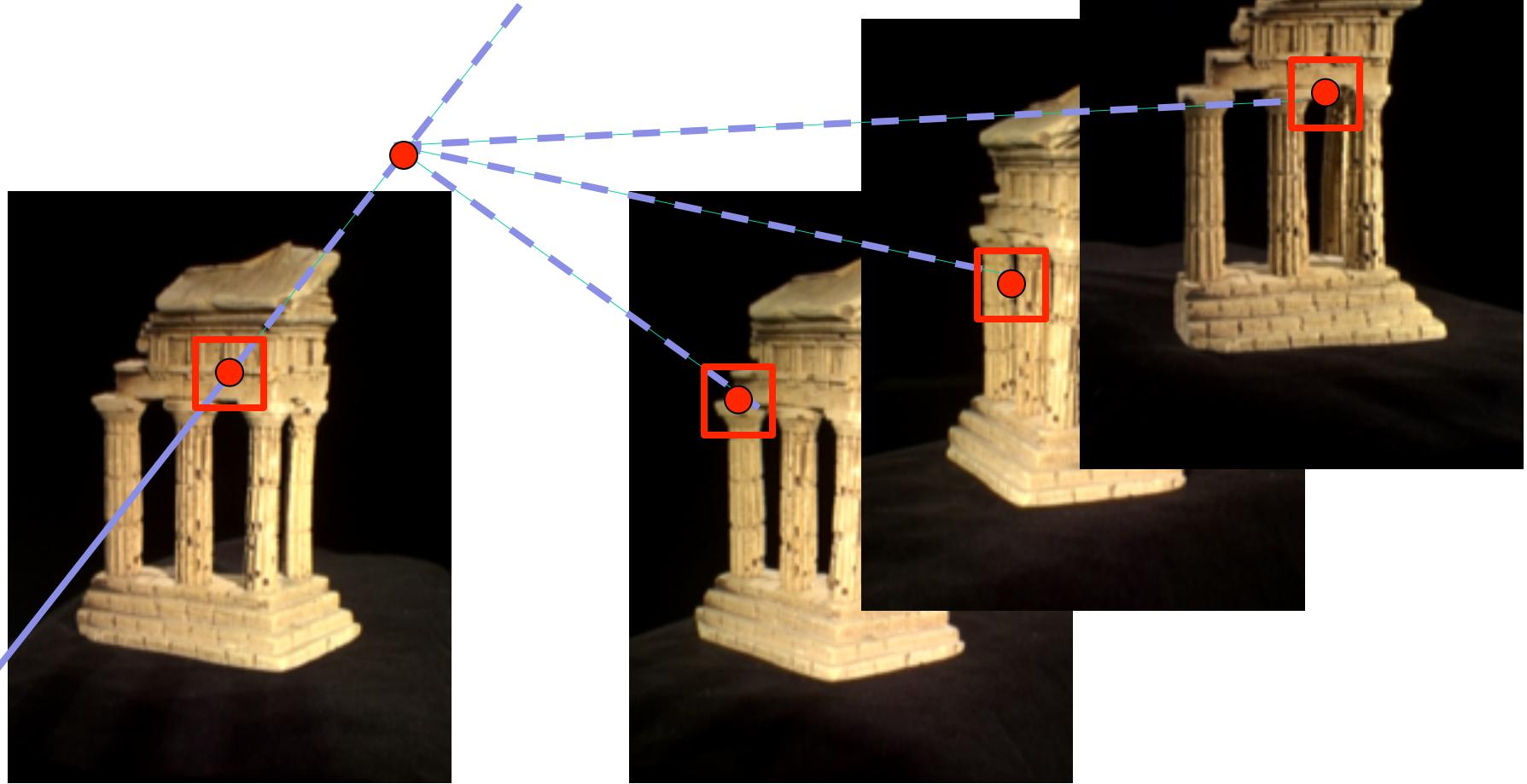
# Multi-view stereo

---

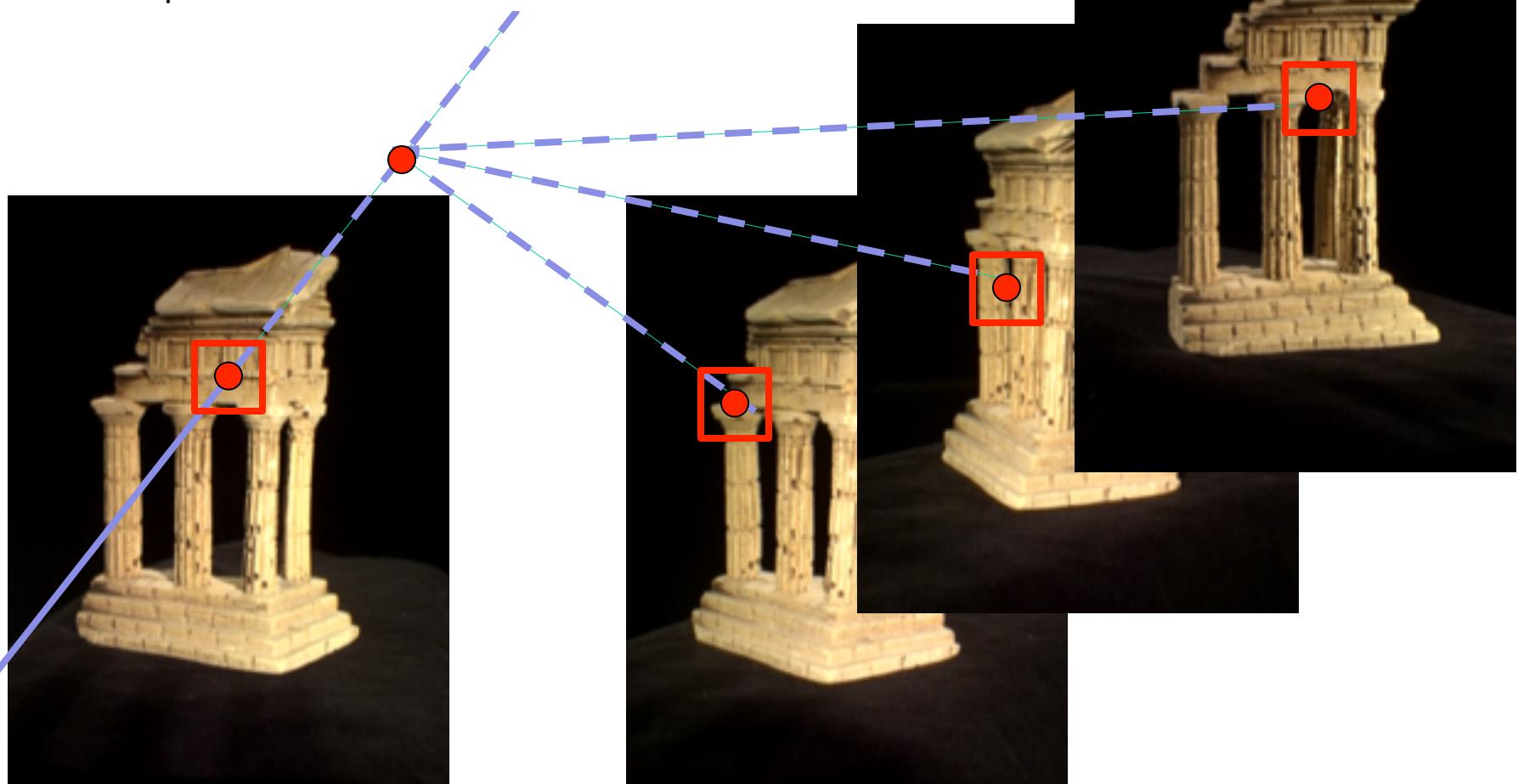
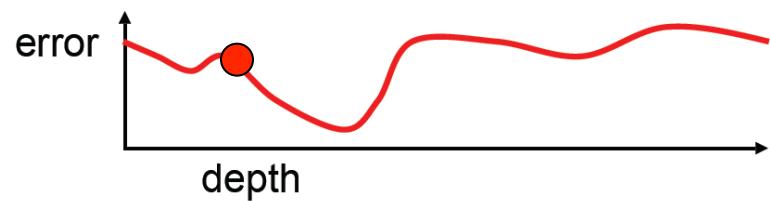
- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape
- “Images of the same object or scene”
  - Arbitrary number of images (from two to thousands)
  - Arbitrary camera positions (special rig, camera network or video sequence)
  - Calibration may be known or unknown
- “Representation of 3D shape”
  - Depth maps
  - Meshes
  - Point clouds
  - Patch clouds
  - Volumetric models
  - ....

# Multi-view stereo: Basic idea

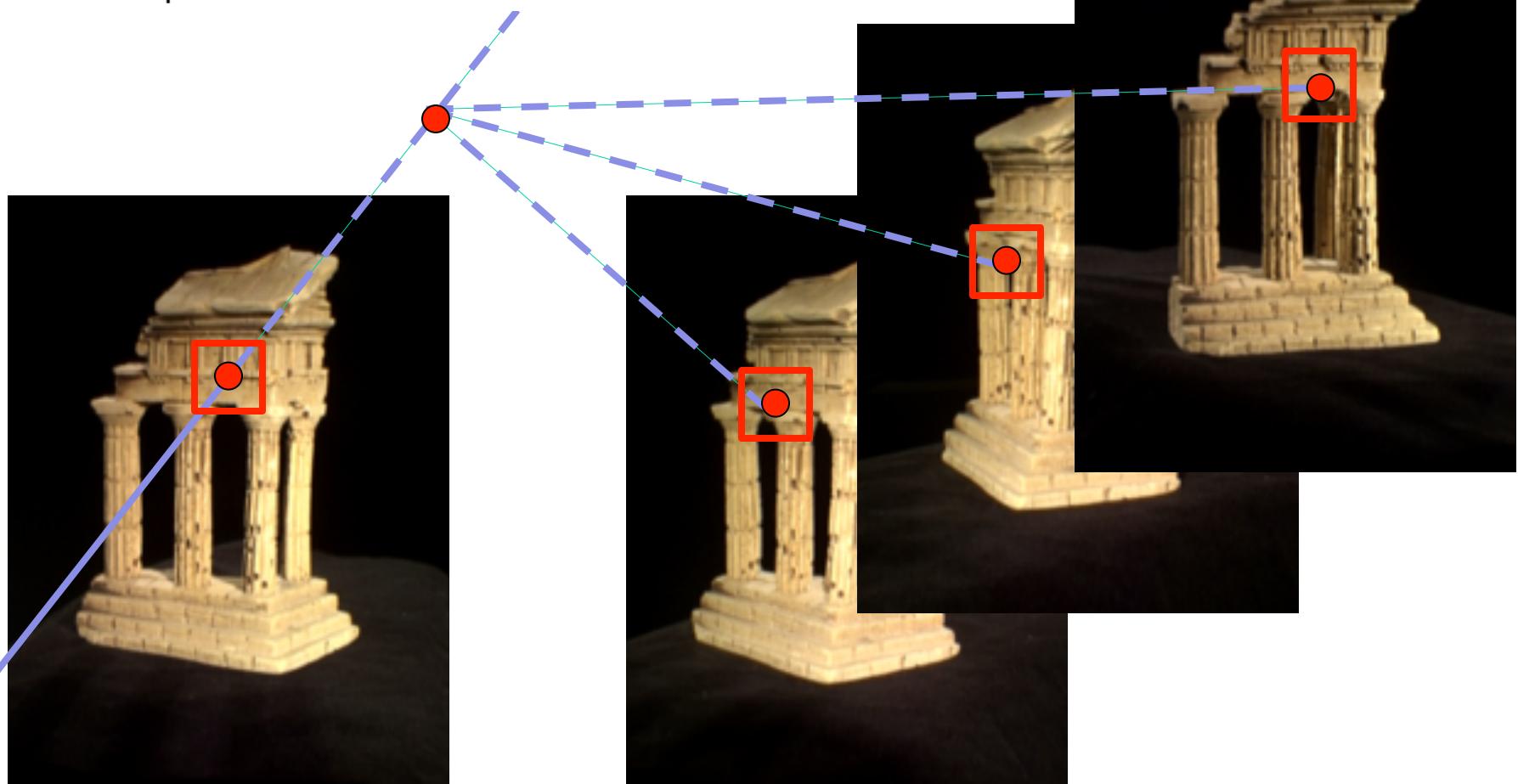
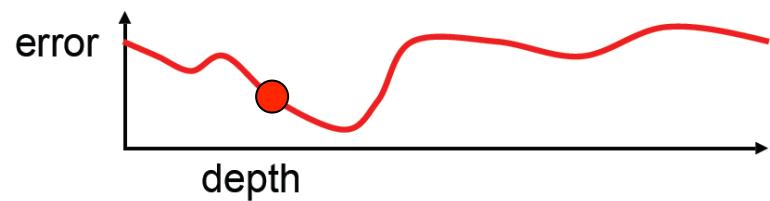
---



# Multi-view stereo: Basic idea

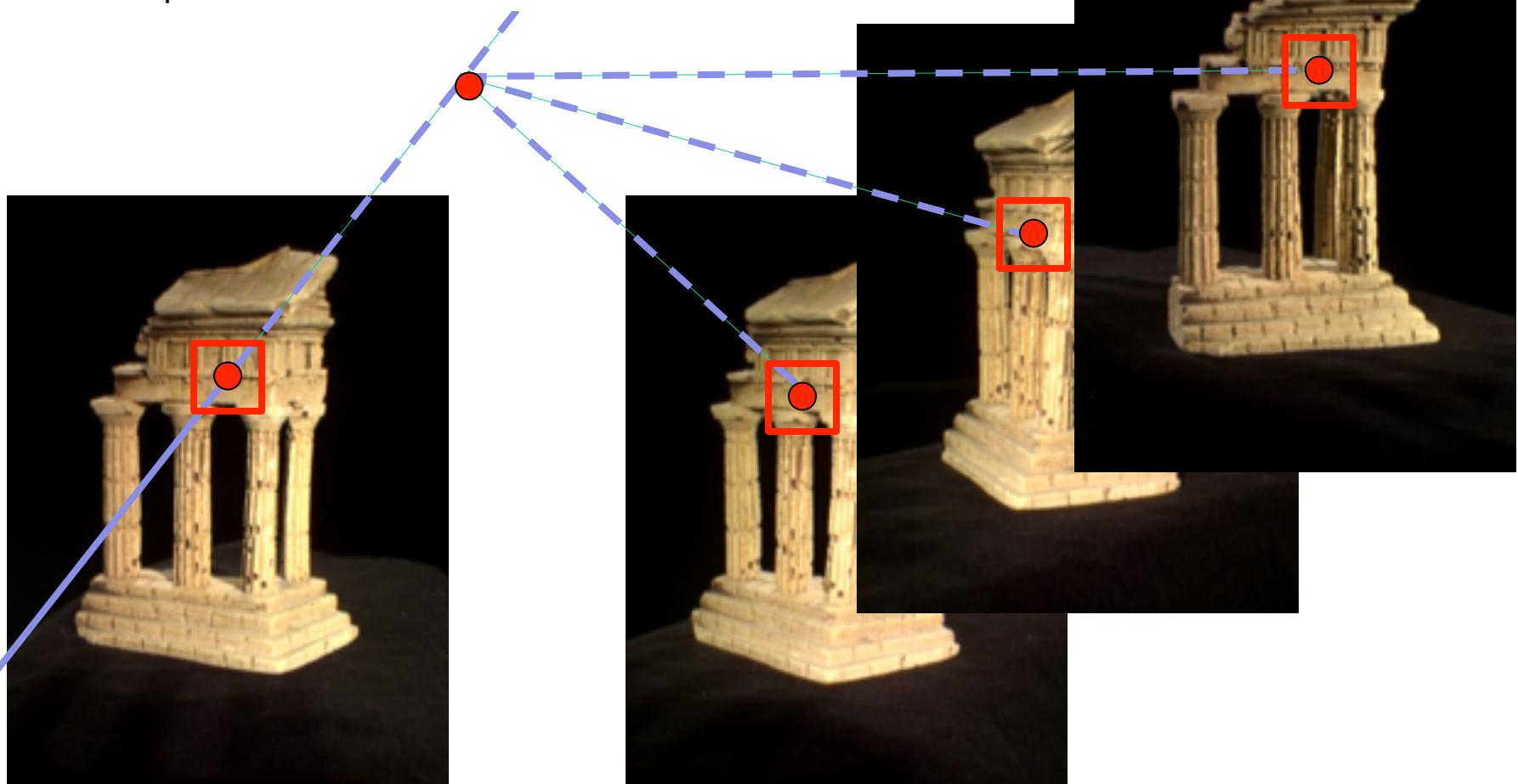
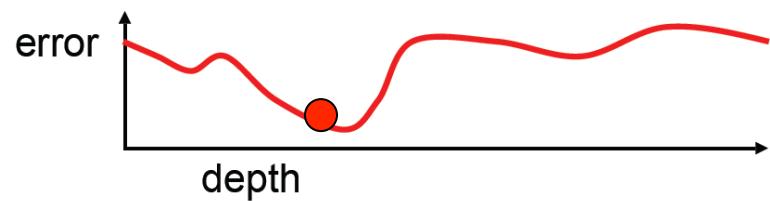


# Multi-view stereo: Basic idea



# Multi-view stereo: Basic idea

---

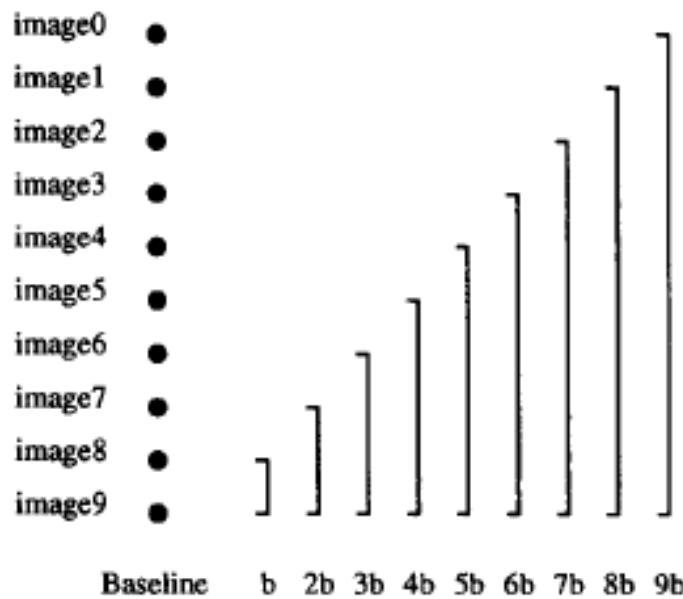


# Multiple-baseline stereo

- Pick a reference image, and slide the corresponding window along the corresponding epipolar lines of all other images, using **inverse depth** relative to the first image as the search parameter



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

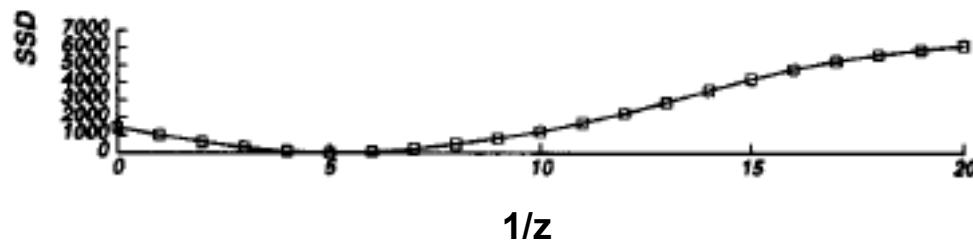


M. Okutomi and T. Kanade, [“A Multiple-Baseline Stereo System,”](#) IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

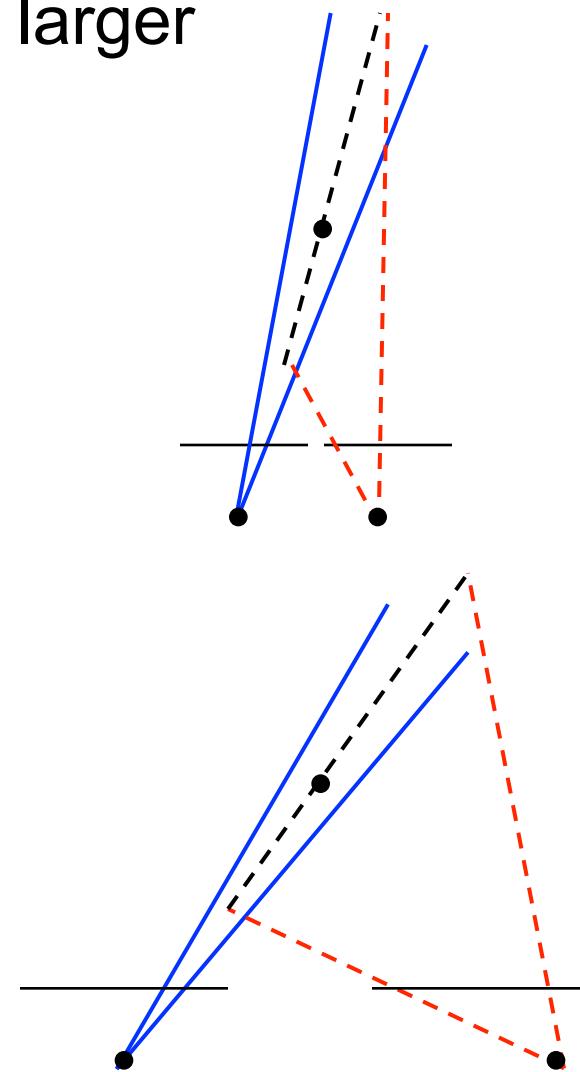
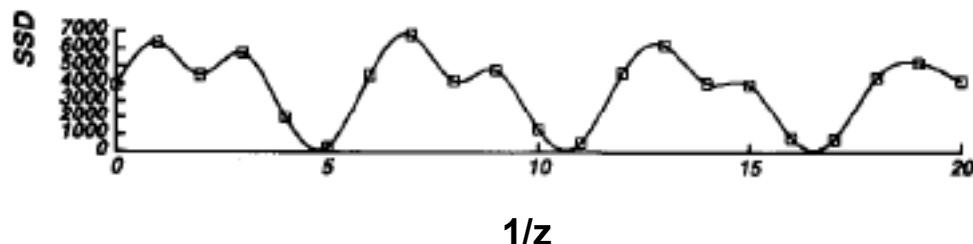
# Multiple-baseline stereo

---

- For larger baselines, must search larger area in second image



pixel matching score



# Multiple-baseline stereo

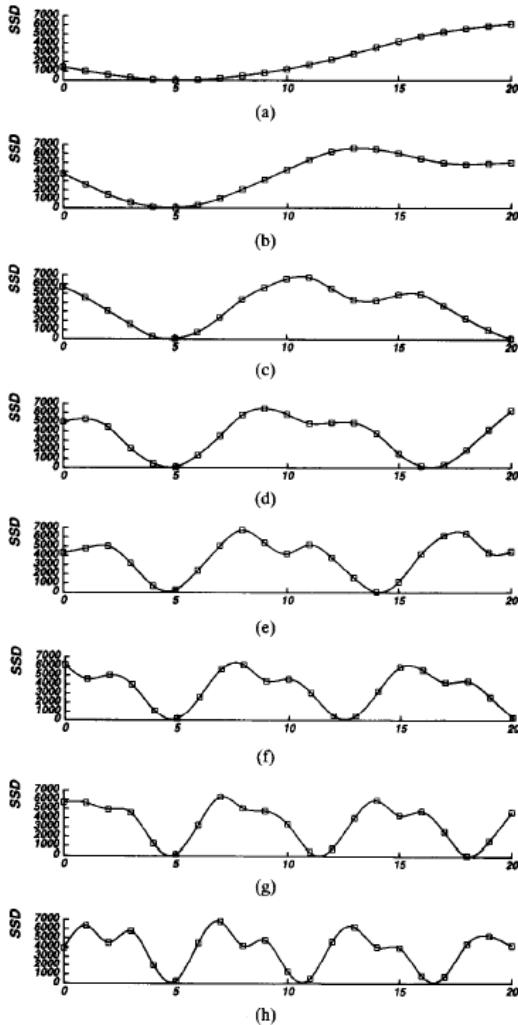


Fig. 5. SSD values versus inverse distance: (a)  $B = b$ ; (b)  $B = 2b$ ; (c)  $B = 3b$ ; (d)  $B = 4b$ ; (e)  $B = 5b$ ; (f)  $B = 6b$ ; (g)  $B = 7b$ ; (h)  $B = 8b$ . The horizontal axis is normalized such that  $8bF = 1$ .

Use the sum of  
SSD scores to rank  
matches

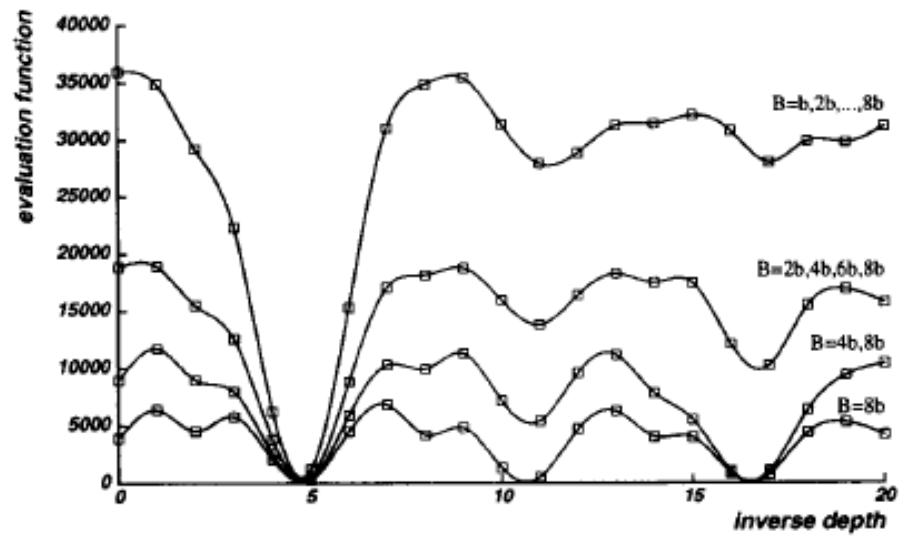


Fig. 7. Combining multiple baseline stereo pairs.

# Multiple-baseline stereo results

---



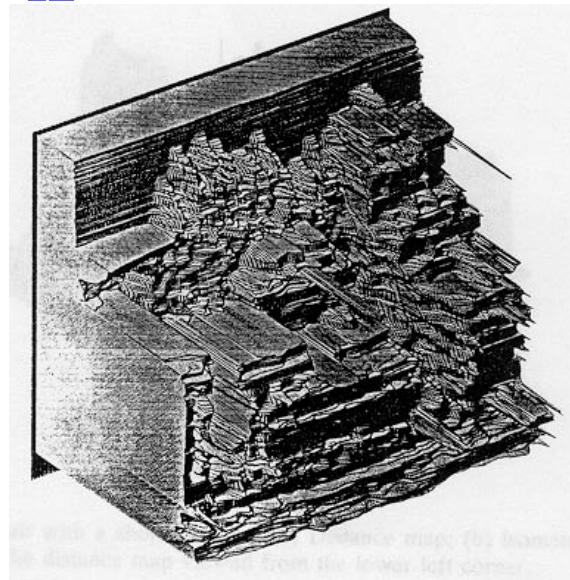
I1



I2



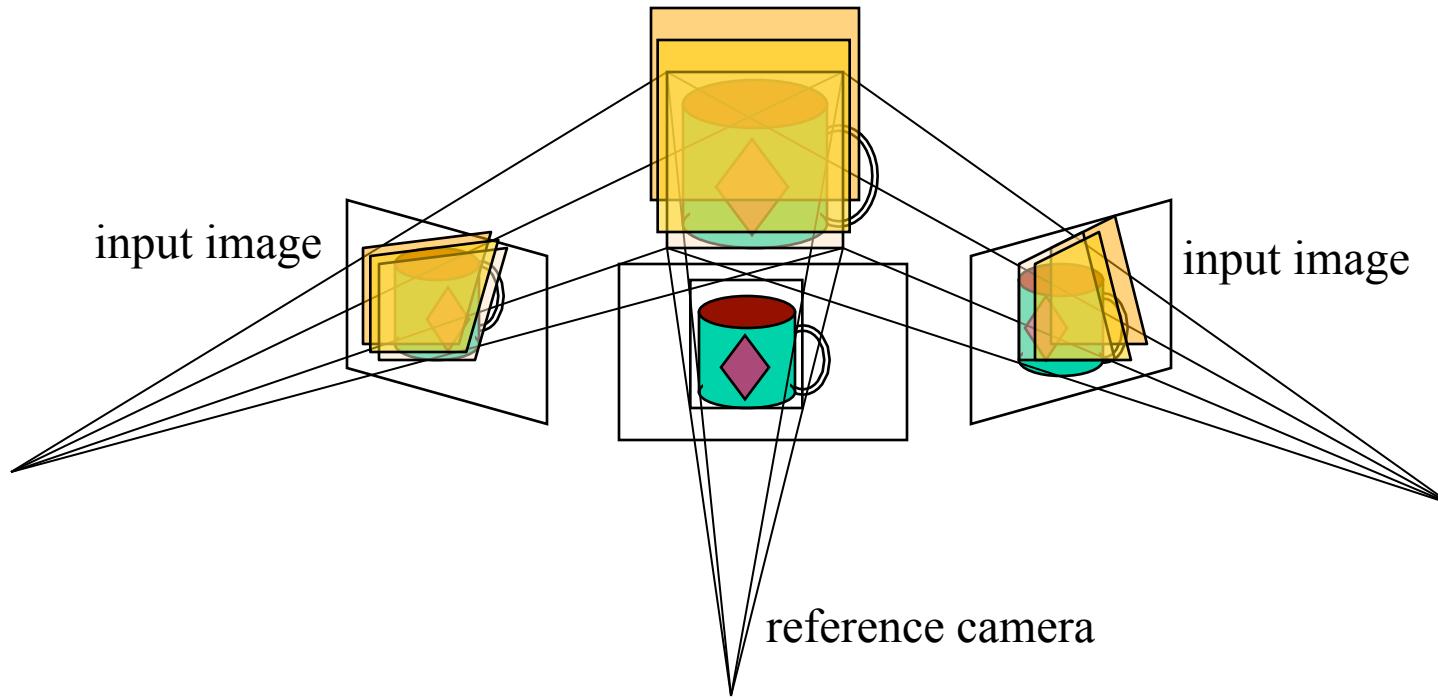
I10



M. Okutomi and T. Kanade, "[A Multiple-Baseline Stereo System](#)," IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

# Plane Sweep Stereo

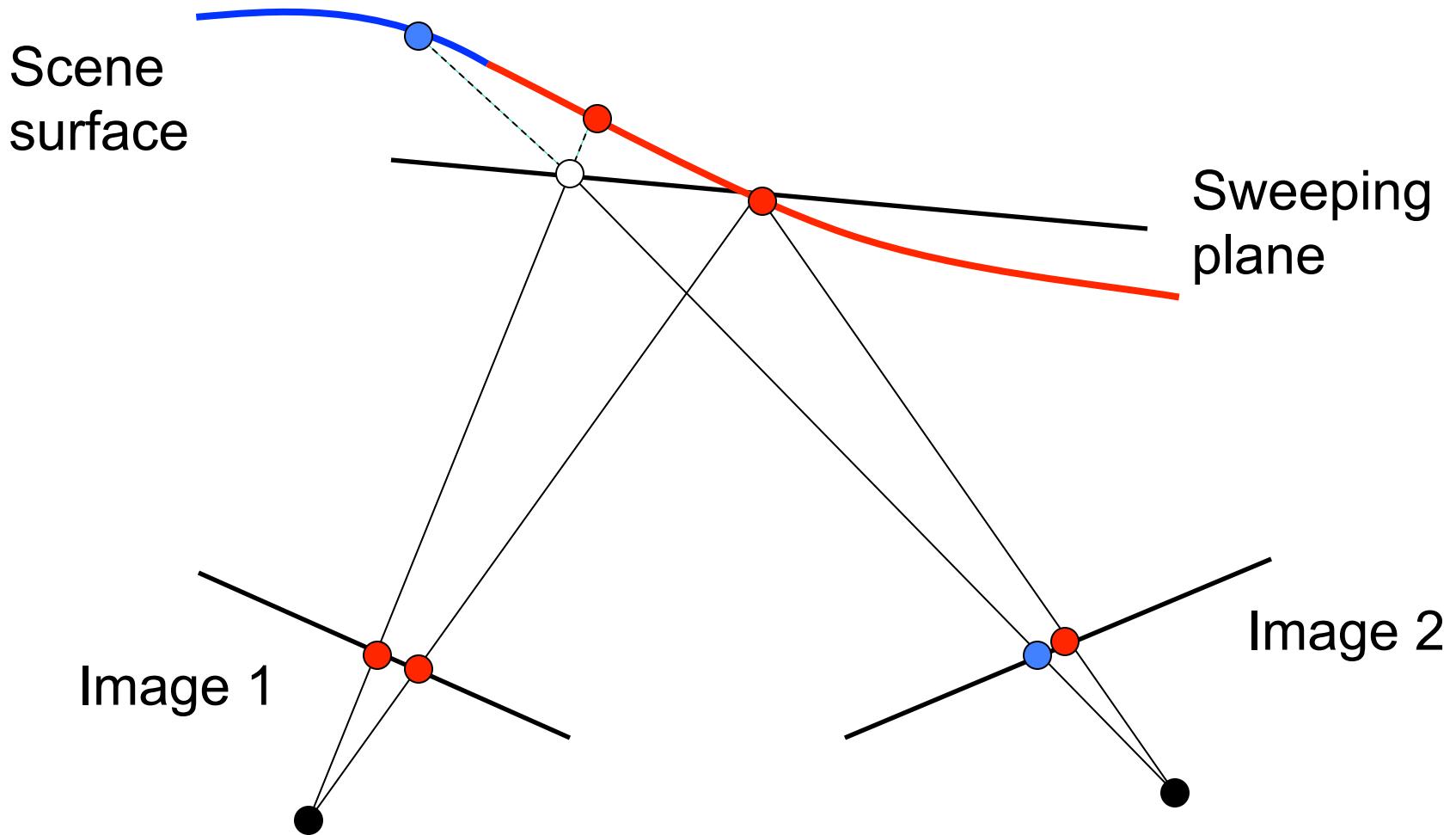
---



- Sweep family of planes at different depths w.r.t. a reference camera
- For each depth, project each input image onto that plane
- This is equivalent to a homography warping each input image into the reference view
- What can we say about the scene points that are at the right depth?

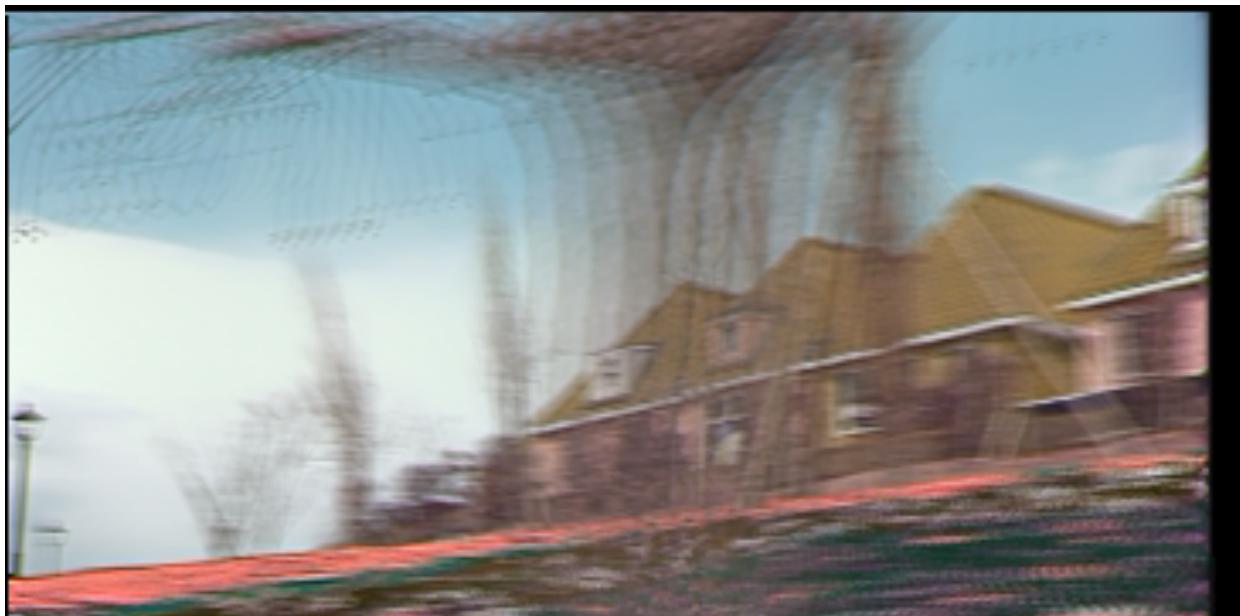
# Plane Sweep Stereo

---



# Plane Sweep Stereo

---



- For each depth plane
  - For each pixel in the composite image stack, compute the variance
  - For each pixel, select the depth that gives the lowest variance
- Can be accelerated using graphics hardware

# Merging depth maps

---

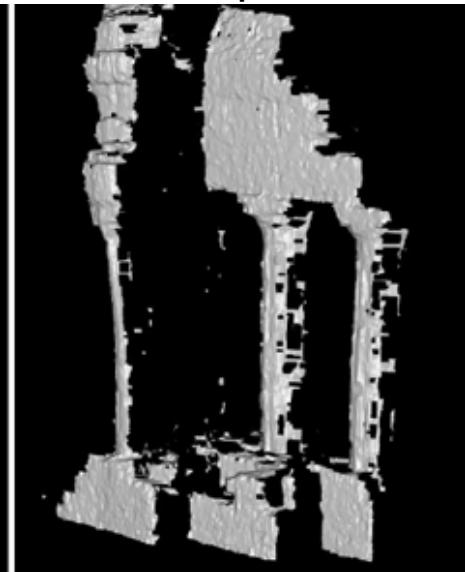


- Given a group of images, choose each one as reference and compute a depth map w.r.t. that view using a multi-baseline approach
- Merge multiple depth maps to a volume or a mesh (see, e.g., Curless and Levoy 96)

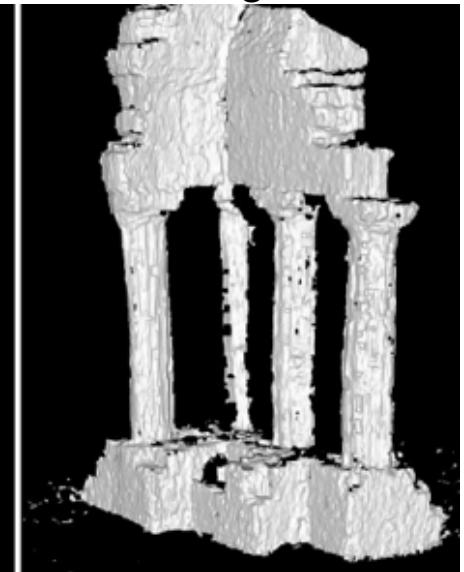
Map 1



Map 2



Merged



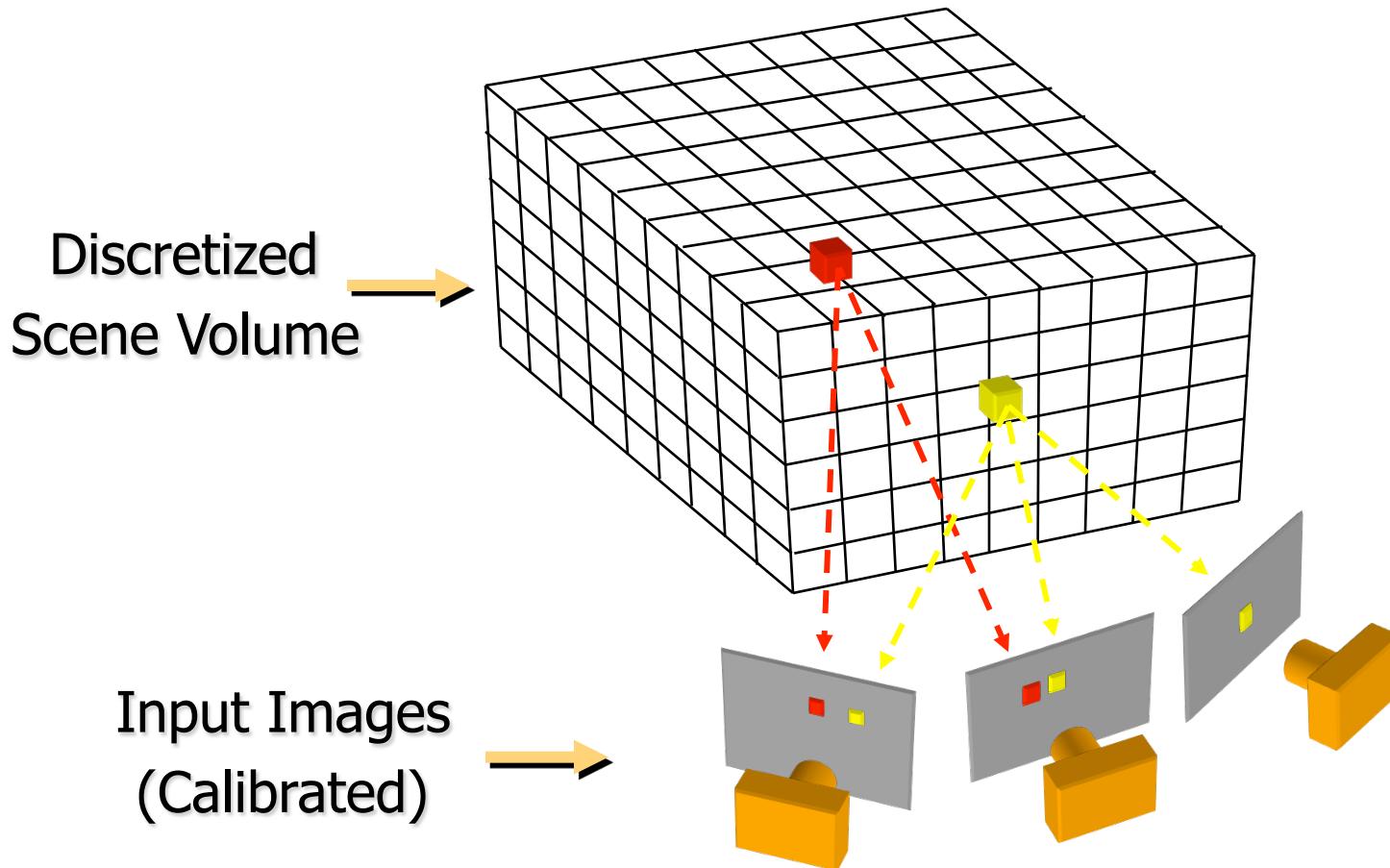
# Volumetric stereo

---

- In plane sweep stereo, the sampling of the scene depends on the reference view
- We can use a voxel volume to get a view-independent representation

# Volumetric stereo

---

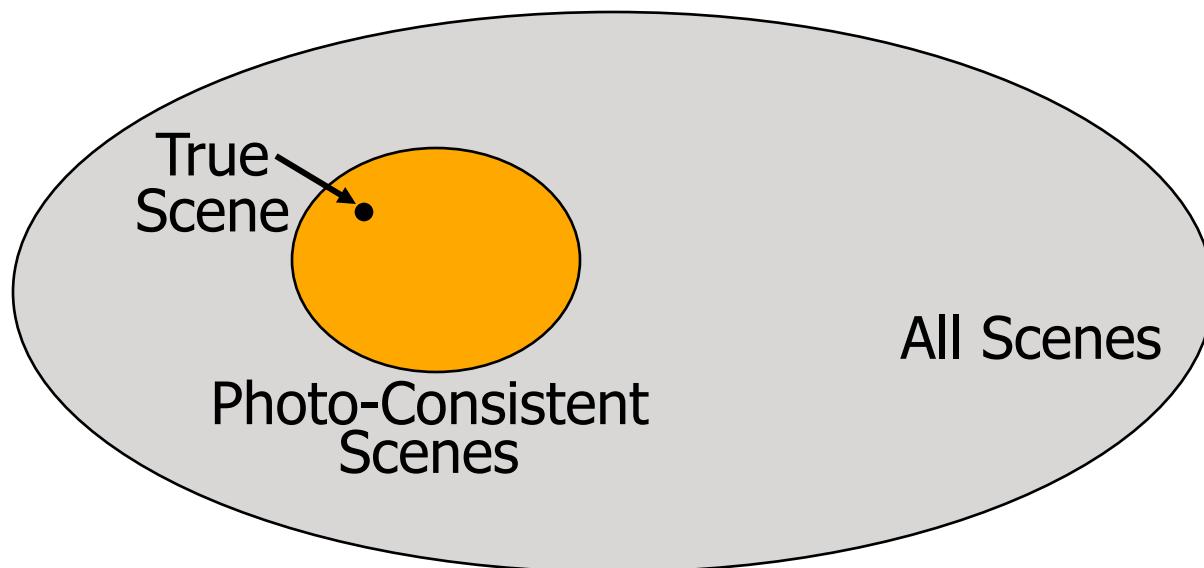


**Goal:** Assign RGB values to voxels in  $V$   
*photo-consistent* with images

# Photo-consistency

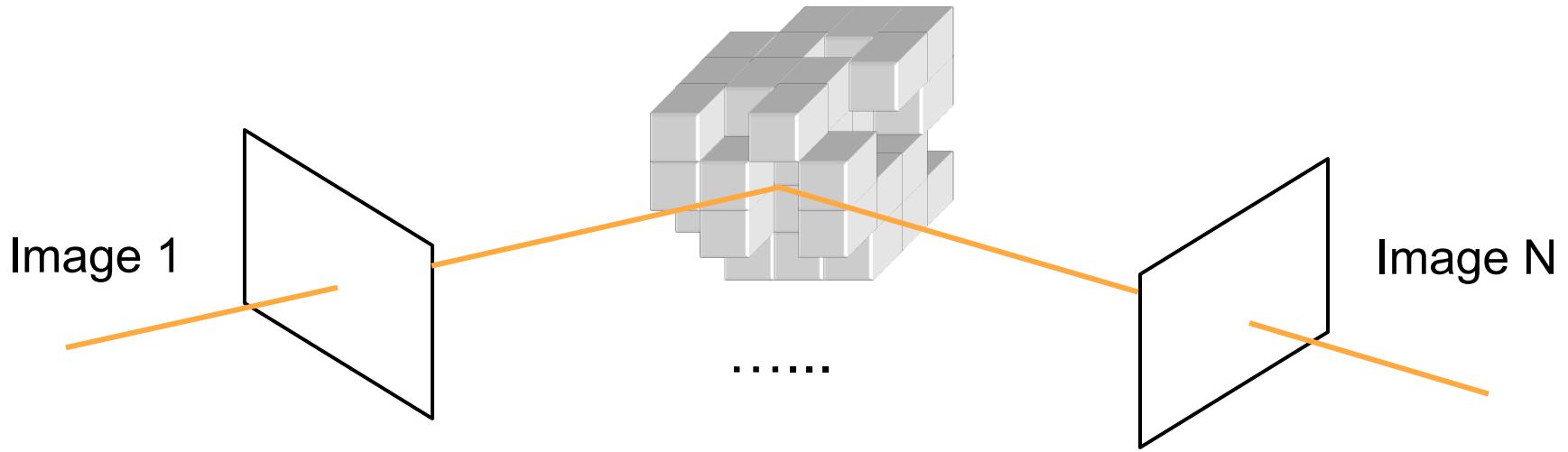
---

- A *photo-consistent scene* is a scene that exactly reproduces your input images from the same camera viewpoints
  - You can't use your input cameras and images to tell the difference between a photo-consistent scene and the true scene



# Space Carving

---



## Space Carving Algorithm

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

# Space Carving Results: African Violet

---



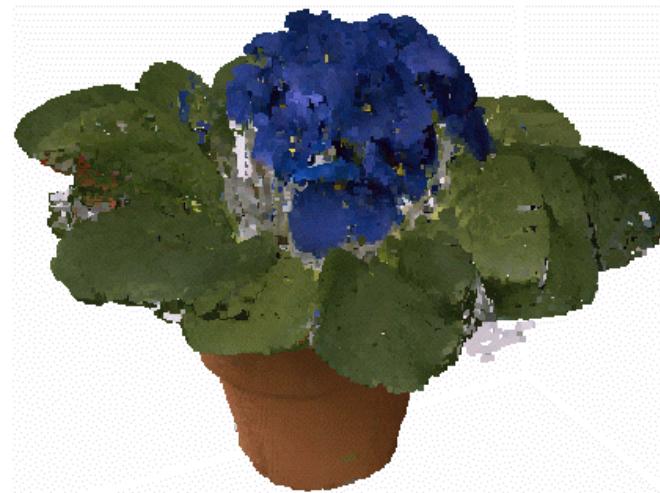
**Input Image (1 of 45)**



**Reconstruction**



**Reconstruction**



**Reconstruction**

Source: S. Seitz

# Space Carving Results: Hand

---



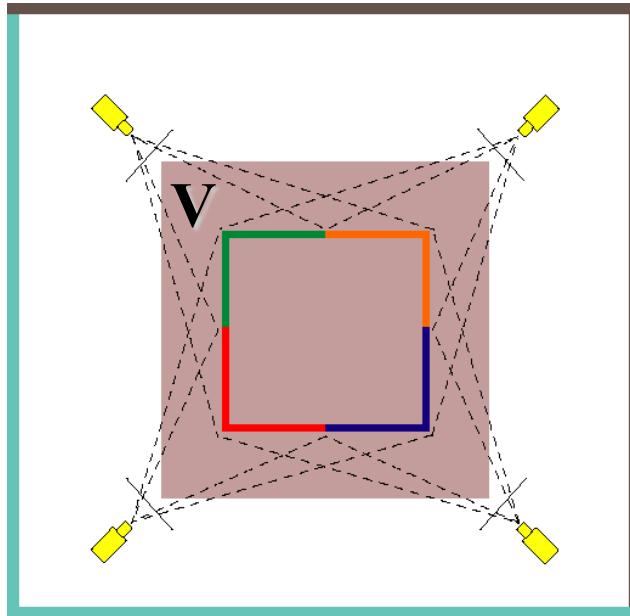
**Input Image  
(1 of 100)**



**Views of Reconstruction**

# Which shape do you get?

---



True Scene

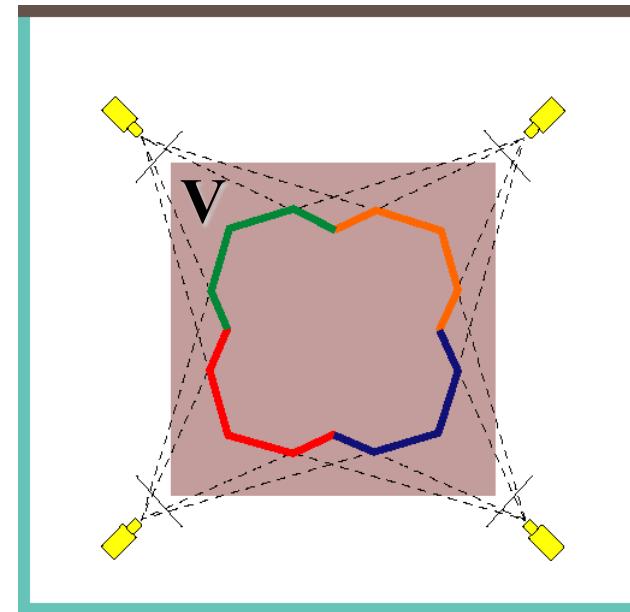


Photo Hull

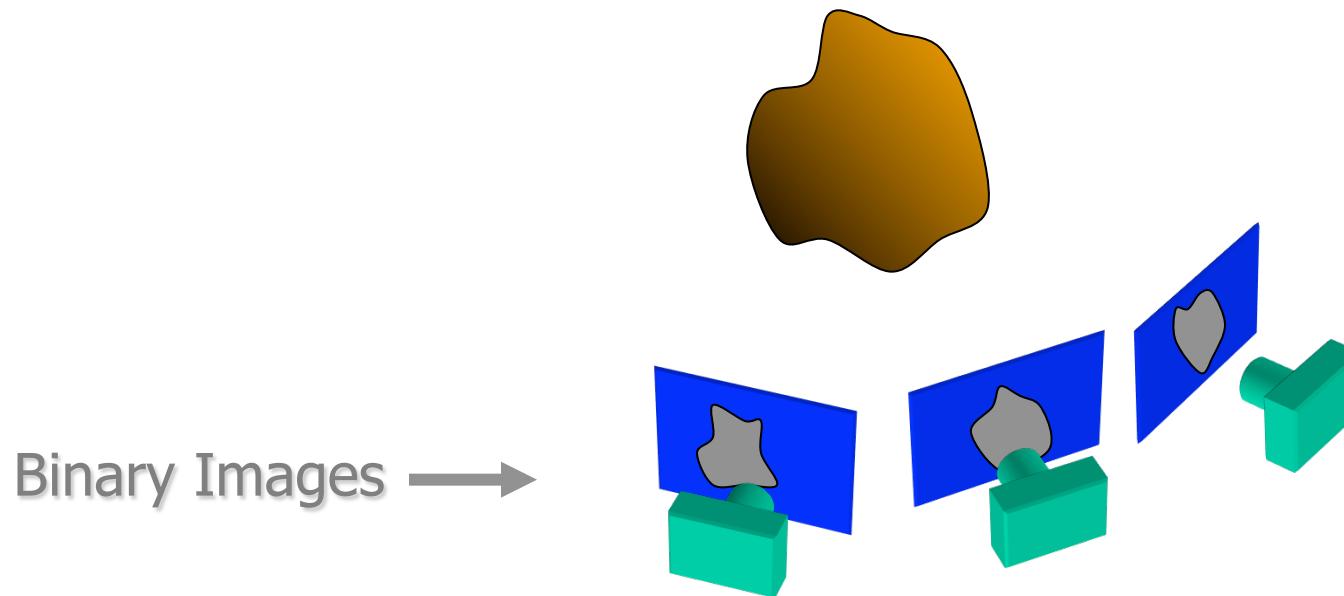
The **Photo Hull** is the UNION of all photo-consistent scenes in  $V$

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

# Reconstruction from Silhouettes

---

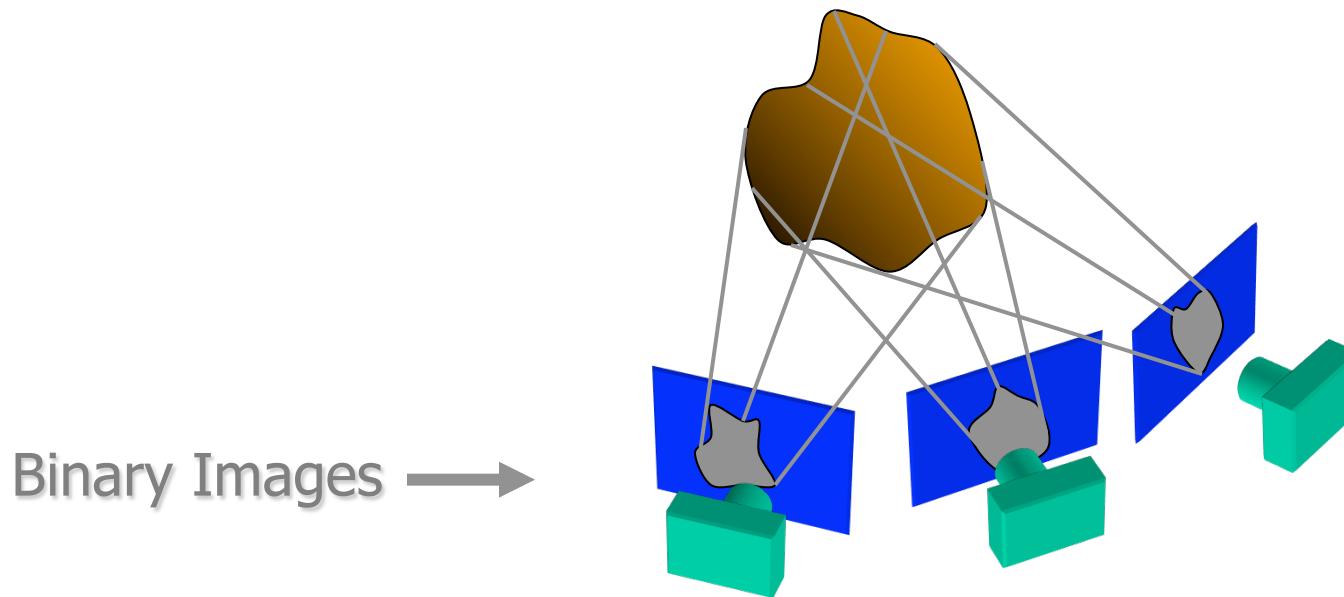
- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views



# Reconstruction from Silhouettes

---

- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views

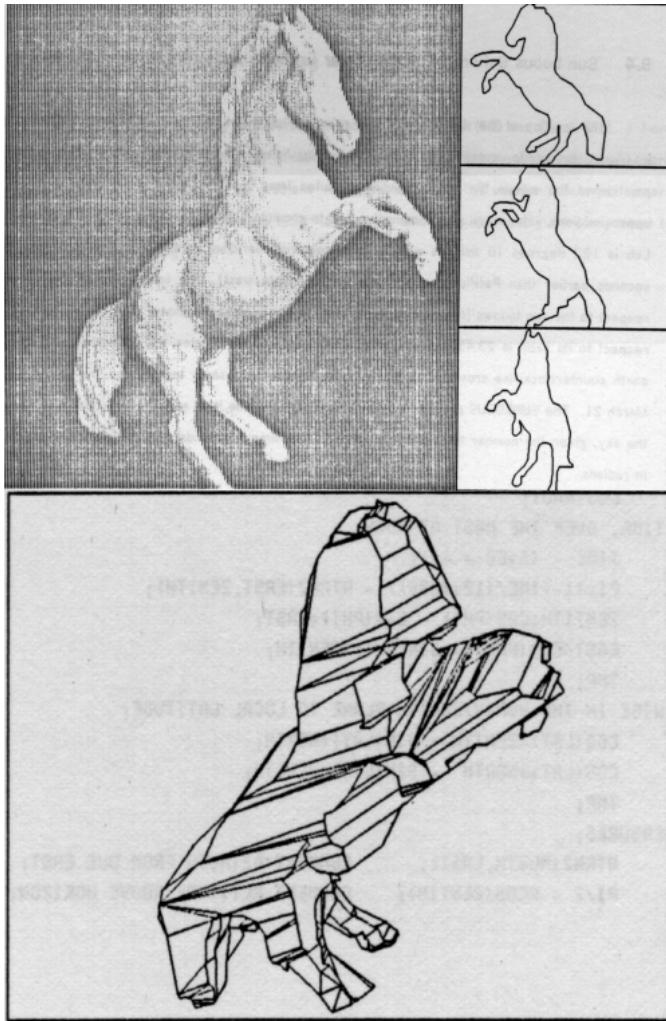


Finding the silhouette-consistent shape (*visual hull*):

- *Backproject* each silhouette
- Intersect backprojected volumes

# Volume intersection

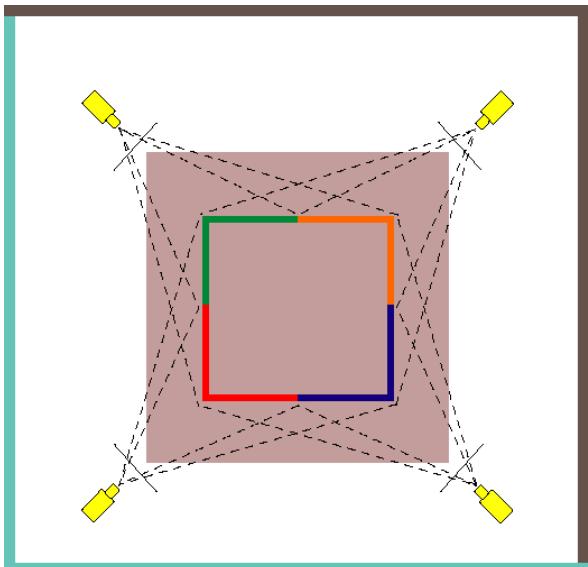
---



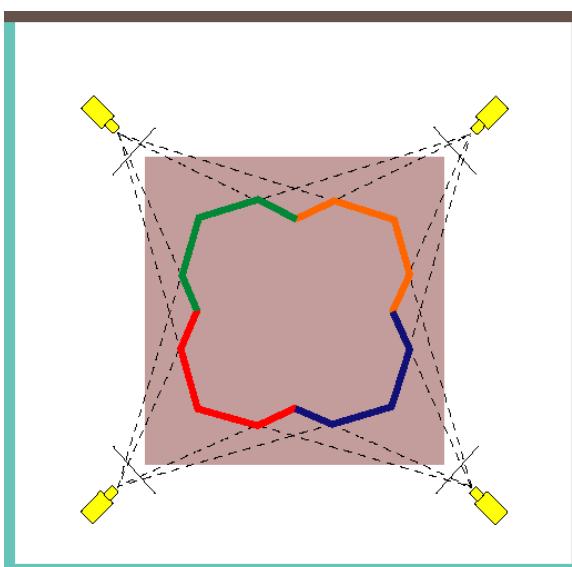
B. Baumgart, [\*Geometric Modeling for Computer Vision\*](#), Stanford Artificial Intelligence Laboratory, Memo no. AIM-249, Stanford University, October 1974.

# Photo-consistency vs. silhouette-consistency

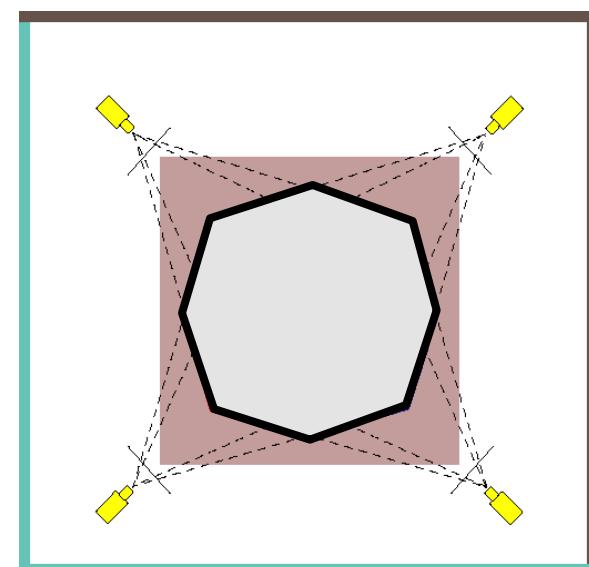
---



**True Scene**



**Photo Hull**



**Visual Hull**

# Carved visual hulls

---

- The visual hull is a good starting point for optimizing photo-consistency
  - Easy to compute
  - Tight outer boundary of the object
  - Parts of the visual hull (rims) already lie on the surface and are already photo-consistent

# Carved visual hulls

---

1. Compute visual hull
2. Use dynamic programming to find rims (photo-consistent parts of visual hull)
3. Carve the visual hull to optimize photo-consistency keeping the rims fixed



# From feature matching to dense stereo

---

1. Extract features
2. Get a sparse set of initial matches
3. Iteratively expand matches to nearby locations
4. Use visibility constraints to filter out false matches
5. Perform surface reconstruction



Yasutaka Furukawa and Jean Ponce,  
[Accurate, Dense, and Robust Multi-View Stereopsis](#), CVPR 2007.

# From feature matching to dense stereo

---



<http://www.cs.washington.edu/homes/furukawa/gallery/>

Yasutaka Furukawa and Jean Ponce,  
**Accurate, Dense, and Robust Multi-View Stereopsis**, CVPR 2007.

# Stereo from community photo collections

- Need *structure from motion* to recover unknown camera parameters
- Need *view selection* to find good groups of images on which to run dense stereo

**flickr®** from YAHOO!

Home You Organize & Create Contacts Groups Explore Upload

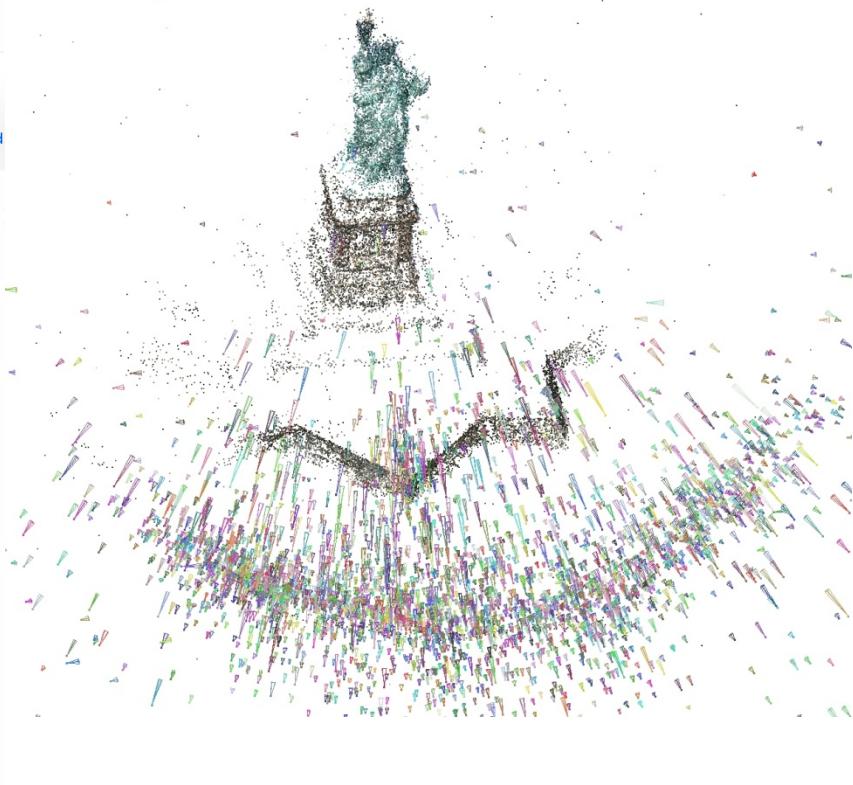
Search Photos Groups People

Everyone's Uploads statue of liberty SEARCH Full Text Advanced

Sort: Relevant | Recent | Interesting View: Small | Medium | Detail | Slideshow

The screenshot shows the Flickr search interface with the query 'statue of liberty'. The search results page displays a grid of 15 images, each with a caption indicating the user who uploaded it. The images show various views of the Statue of Liberty.

Image	User
	From EdZa...
	From micbaun
	From rafaj
	From lepublicname
	From Jesus...
	From Julio...
	From StephiGra...
	From alabs
	From BigMs.Take
	From laurvre
	From dmp0309
	From laurenbou...
	From StephiGra...
	From Mojumbo22...



# Towards Internet-Scale Multi-View Stereo



[YouTube video](#), [high-quality video](#)

Yasutaka Furukawa, Brian Curless, Steven M. Seitz and Richard Szeliski,  
[Towards Internet-scale Multi-view Stereo](#), CVPR 2010.

# The Visual Turing Test for Scene Reconstruction

Rendered Images (Right) vs. Ground Truth Images (Left)

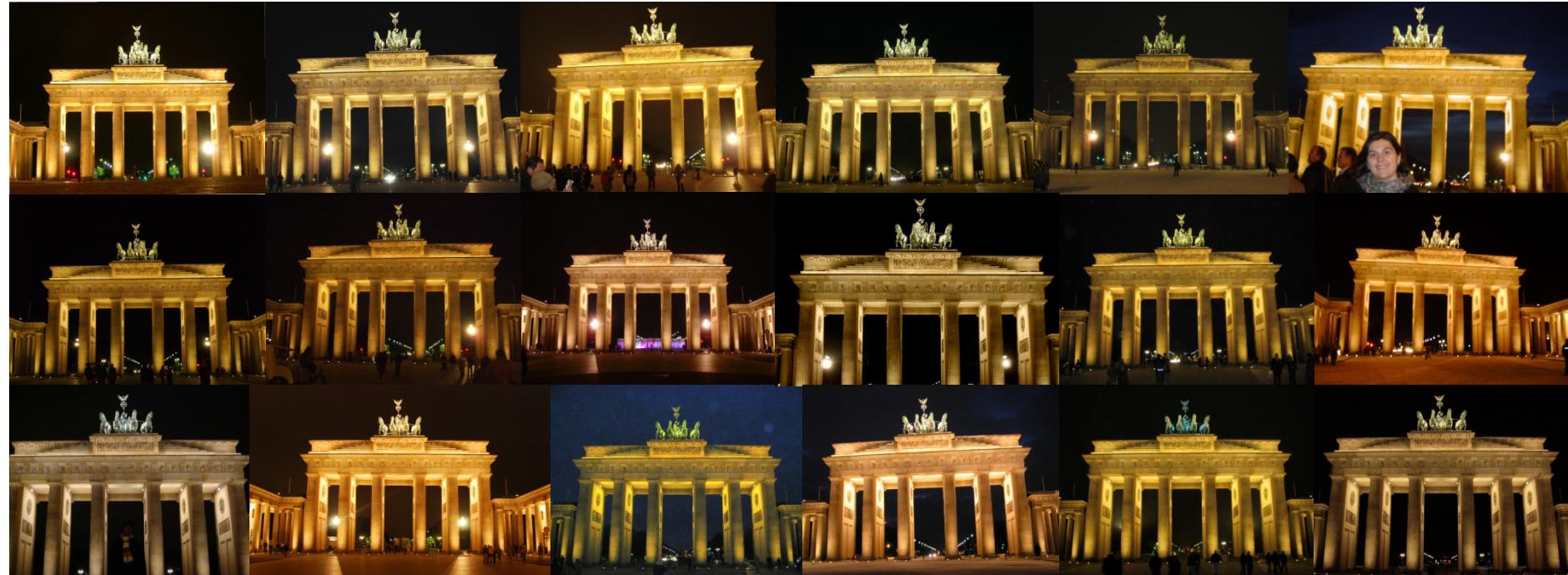


Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. Seitz,  
["The Visual Turing Test for Scene Reconstruction,"](#) 3DV 2013.

# Fast stereo for Internet photo collections

---

- Start with a cluster of registered views
- Obtain a depth map for every view using plane sweeping stereo with normalized cross-correlation

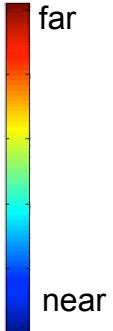
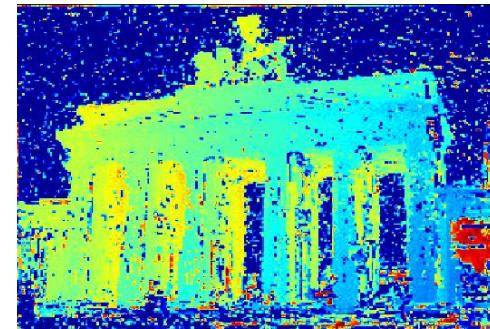
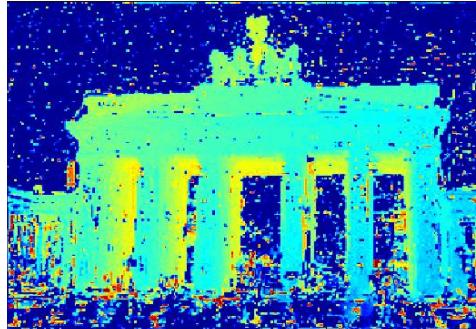
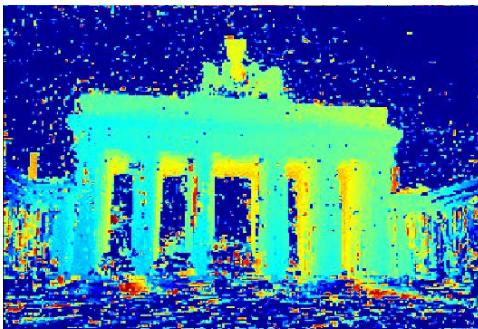


Frahm et al., [“Building Rome on a Cloudless Day,”](#) ECCV 2010.

# Plane sweeping stereo

---

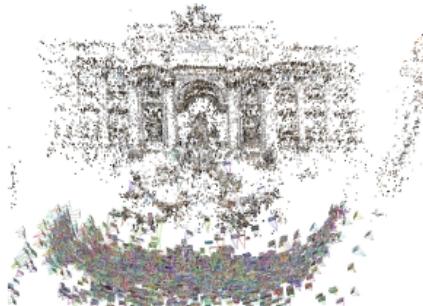
- Need to register individual depth maps into a single 3D model
- Problem: depth maps are very noisy



Frahm et al., [“Building Rome on a Cloudless Day,”](#) ECCV 2010.

# Results

---



[YouTube Video](#)

Frahm et al., "[Building Rome on a Cloudless Day,](#)" ECCV 2010.

# Kinect: Structured infrared light

---



<http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/>

# KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera\*

*Shahram Izadi<sup>1</sup>, David Kim<sup>1,3</sup>, Otmar Hilliges<sup>1</sup>, David Molyneaux<sup>1,4</sup>, Richard Newcombe<sup>2</sup>, Pushmeet Kohli<sup>1</sup>, Jamie Shotton<sup>1</sup>, Steve Hodges<sup>1</sup>, Dustin Freeman<sup>1,5</sup>, Andrew Davison<sup>2</sup>, Andrew Fitzgibbon<sup>1</sup>*

<sup>1</sup>Microsoft Research Cambridge, UK      <sup>2</sup>Imperial College London, UK

<sup>3</sup>Newcastle University, UK      <sup>4</sup>Lancaster University, UK      <sup>5</sup>University of Toronto, Canada

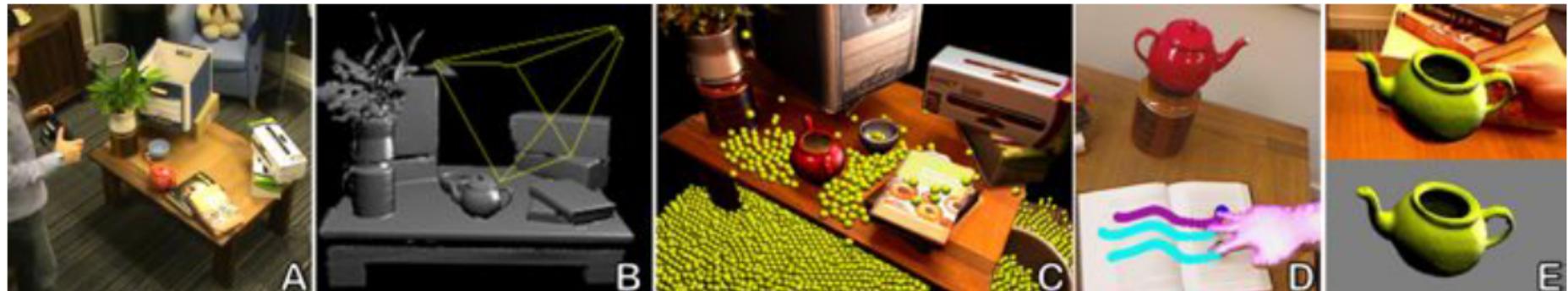


Figure 1: KinectFusion enables real-time detailed 3D reconstructions of indoor scenes using only the depth data from a standard Kinect camera. A) user points Kinect at coffee table scene. B) Phong shaded reconstructed 3D model (the wireframe frustum shows current tracked 3D pose of Kinect). C) 3D model texture mapped using Kinect RGB data with real-time particles simulated on the 3D model as reconstruction occurs. D) Multi-touch interactions performed on any reconstructed surface. E) Real-time segmentation and 3D tracking of a physical object.

[Paper link](#) (ACM Symposium on User Interface Software and Technology,  
October 2011)

[YouTube Video](#)

# Summary: 3D geometric vision

---

- Single-view geometry
  - The pinhole camera model
  - The perspective projection matrix
  - Intrinsic and extrinsic parameters
  - Calibration
  - Single-view metrology, calibration using vanishing points
- Multiple-view geometry
  - Triangulation
  - The epipolar constraint
    - Essential matrix and fundamental matrix
  - Stereo
    - Binocular, multi-view
  - Structure from motion
    - Reconstruction ambiguity
    - Projective SFM