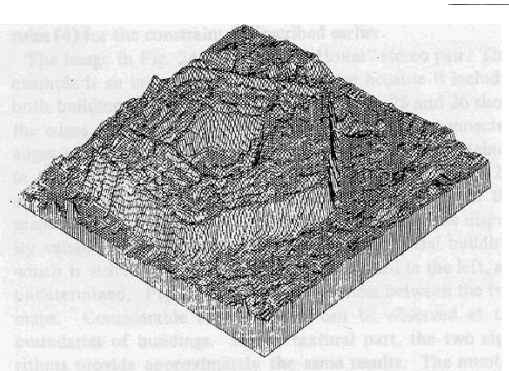
计算机视觉

Computer Vision

Lecture 11: Structure from Motion

张 超

信息科学技术学院 智能科学系




Overview

- **Structure from Motion (SfM)**
- **Large scale Structure from Motion**
- **Other approaches to obtaining 3D structure**

Overview

- **Structure from Motion (SfM)**
- **Large scale Structure from Motion**
- **Other approaches to obtaining 3D structure**

Structure from Motion

- **SFM problem statement**
- **Factorization**
- **Projective SFM**
- **Bundle Adjustment**

Structure from motion

Structure from motion

Multiple-view geometry questions

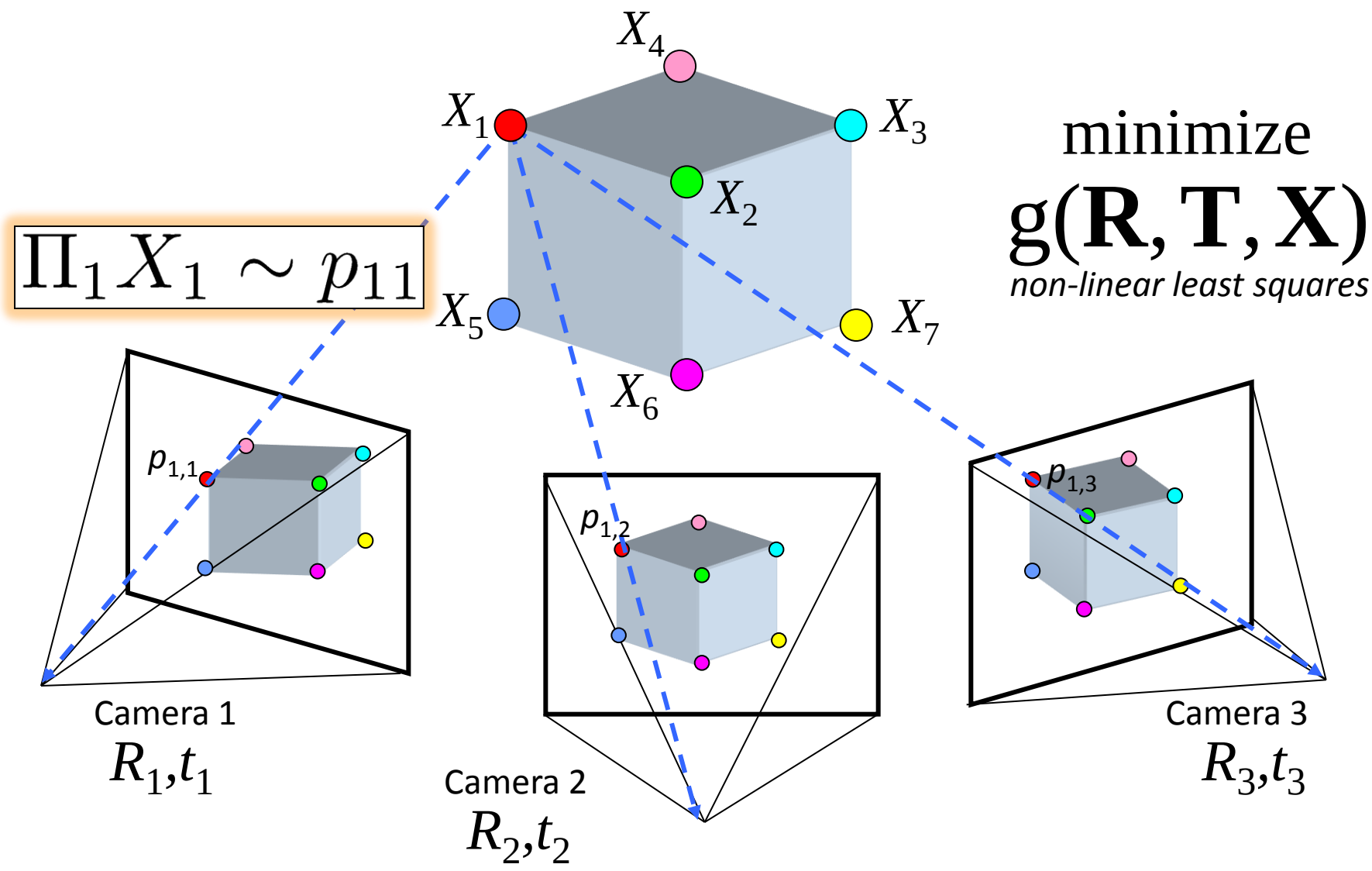
- **Scene geometry (structure): Given 2D point matches in two or more images, where are the corresponding points in 3D?**

- **Correspondence (stereo matching): Given a point in just one image, how does it constrain the position of the corresponding point in another image?**

- **Camera geometry (motion): Given a set of corresponding points in two or more images, what are the camera matrices for these views?**

Slide: S. Lazebnik

Structure from motion



minimize

$$g(\mathbf{R}, \mathbf{T}, \mathbf{X})$$

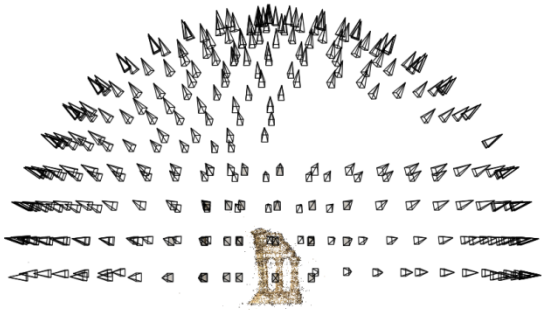
non-linear least squares

Structure from motion

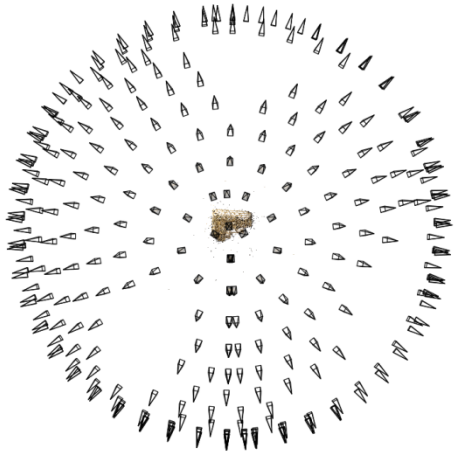
m Images

n Points

Assume correspondences



Reconstruction (side)



(top)

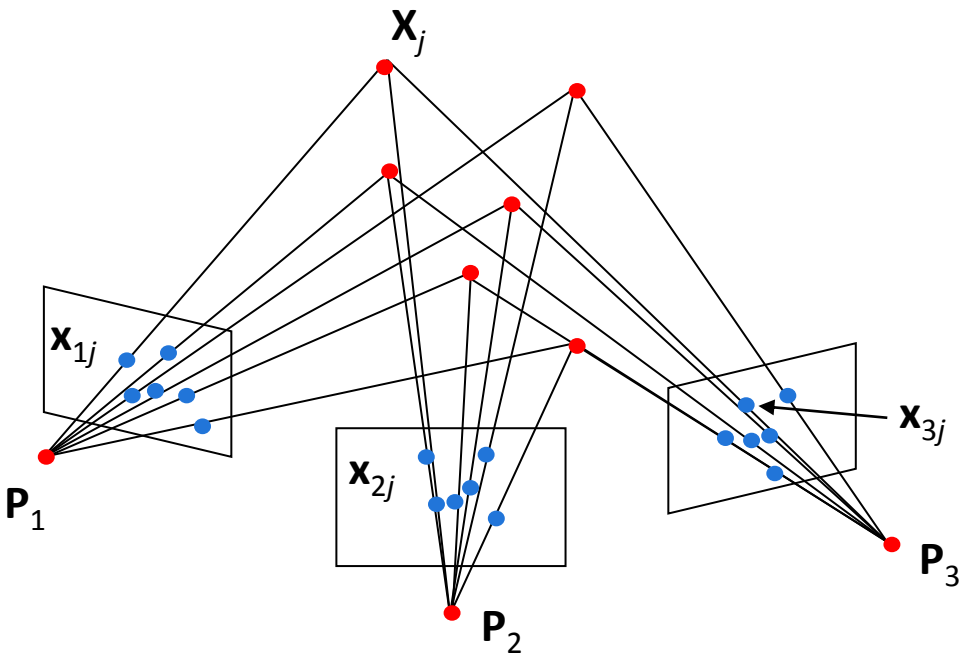
- Input: images with points in correspondence
  $p_{i,j} = (u_{i,j}, v_{i,j})$

- Output
  - structure: 3D location $\mathbf{x}_i$ for each point p_i
  - motion: camera parameters $\mathbf{R}_j$, $\mathbf{t}_j$

- Objective function: minimize *reprojection error*

Structure from motion

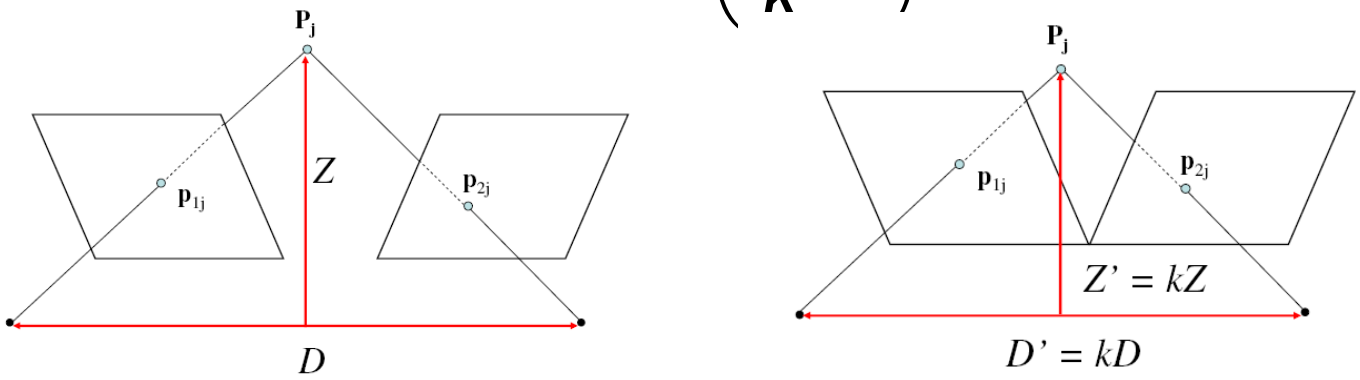
- **Given: *m* images of *n* fixed 3D points**
  - $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- **Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$**



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!

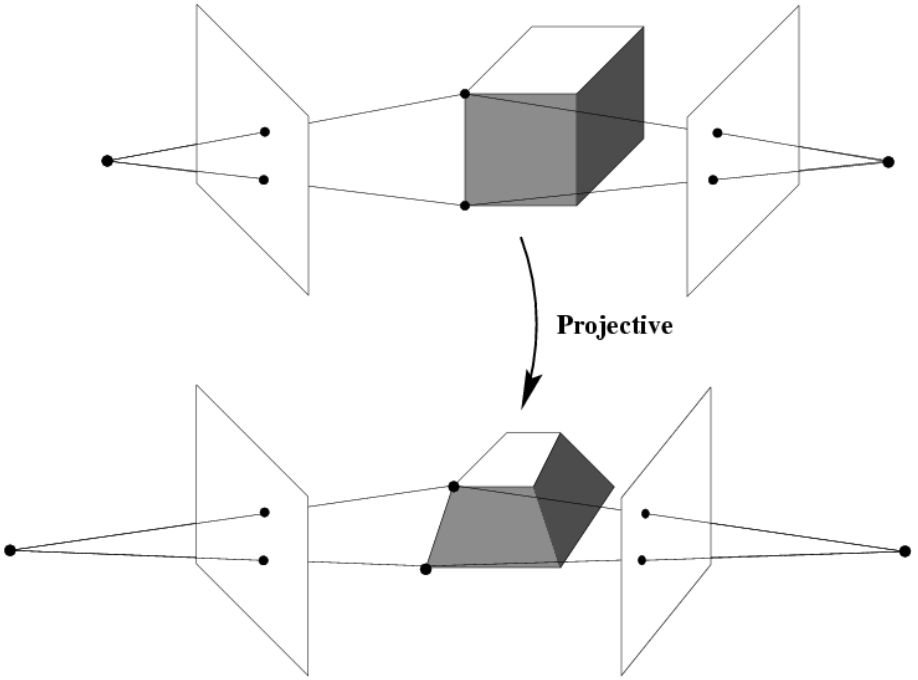
Structure from motion ambiguity

- 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 Q and apply the inverse transformation to the camera matrices, then the images do not change

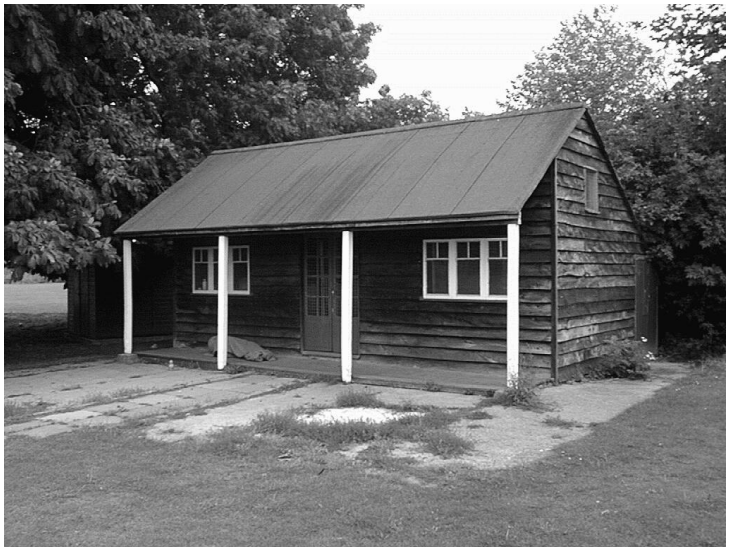
$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ}^{-1}\right)\left(\mathbf{QX}\right)$$

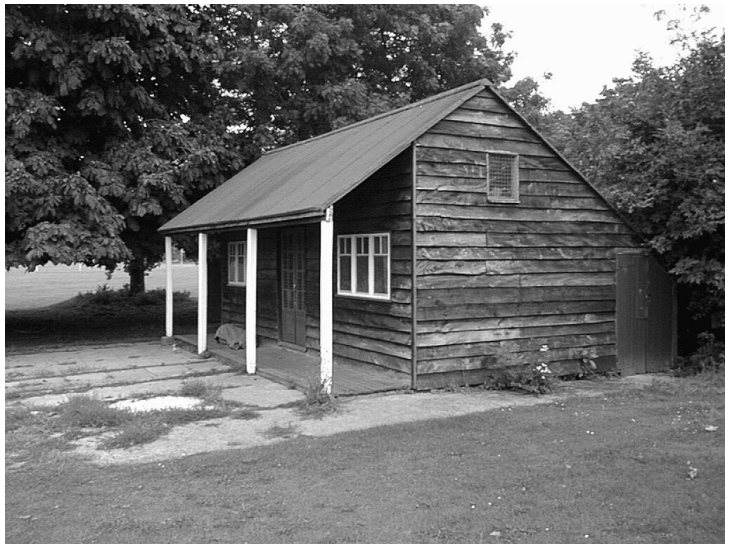
Projective ambiguity

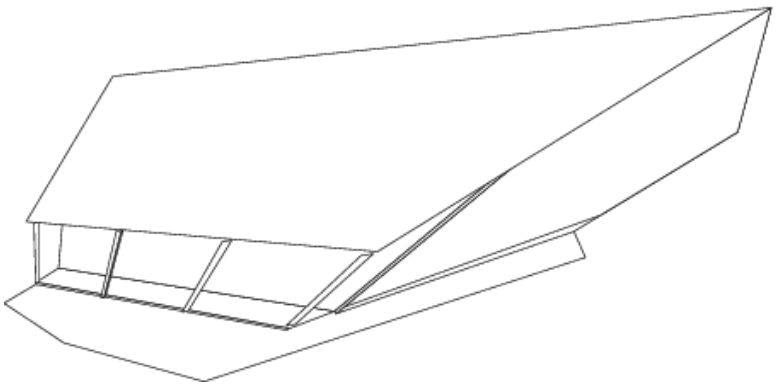


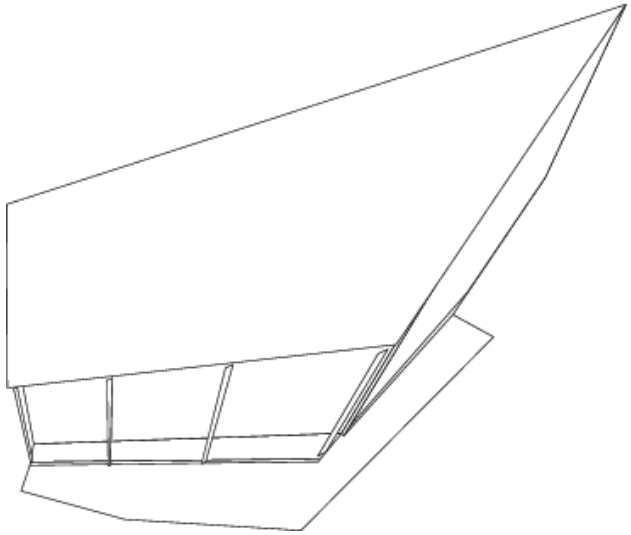
$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ}_{\mathbf{P}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{P}}\,\mathbf{X}\right)$$

Projective ambiguity

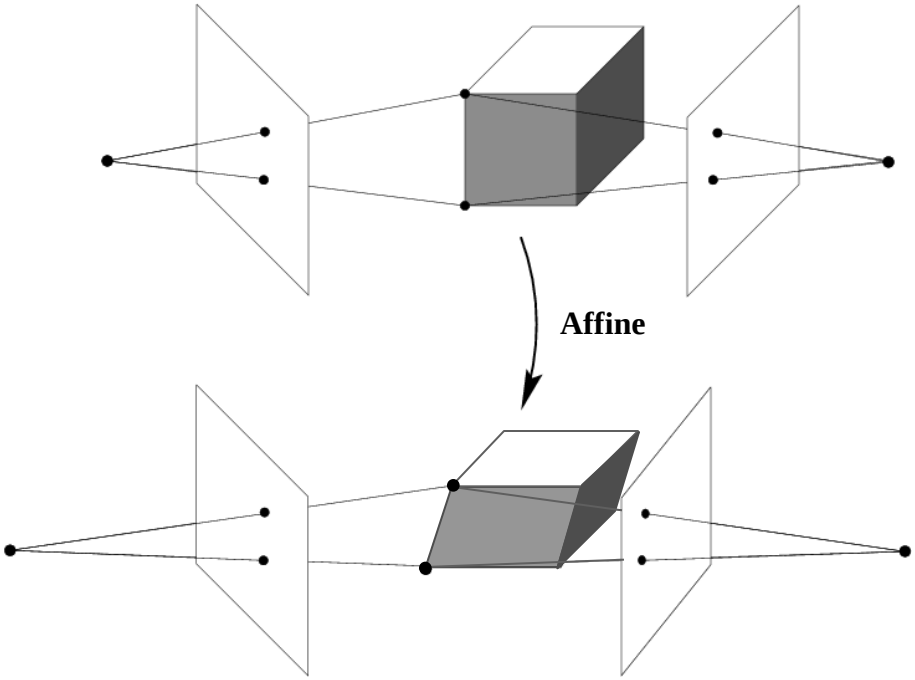








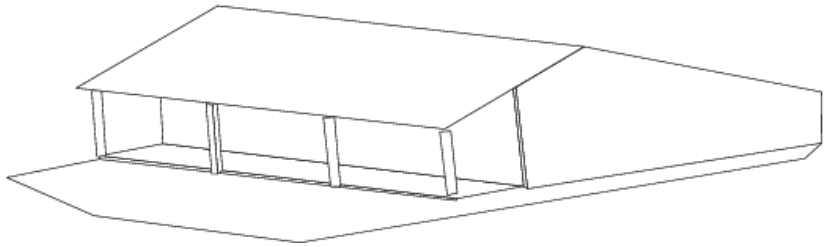
Affine ambiguity

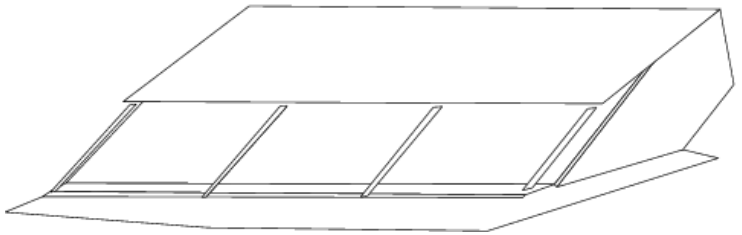


$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ}_{\mathbf{A}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{A}}\,\mathbf{X}\right)$$

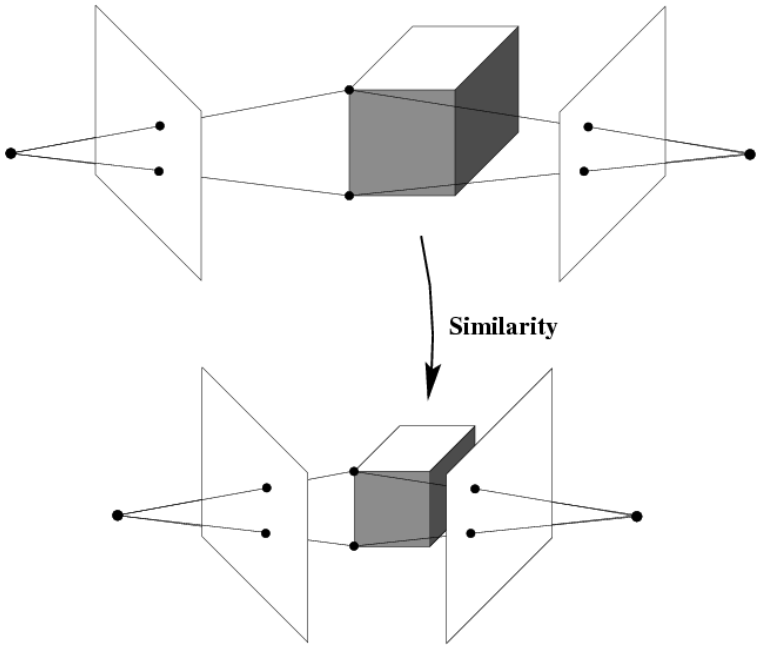
Affine ambiguity





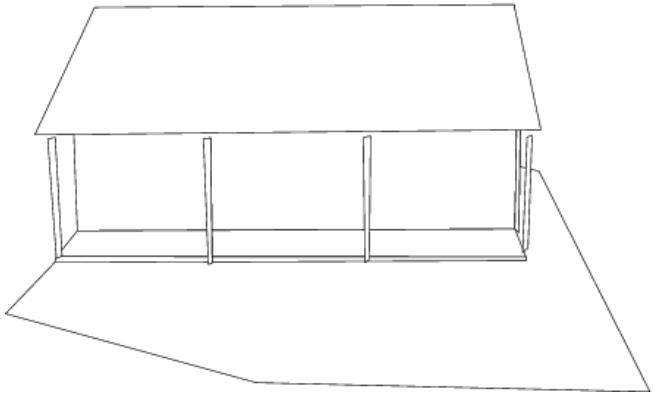


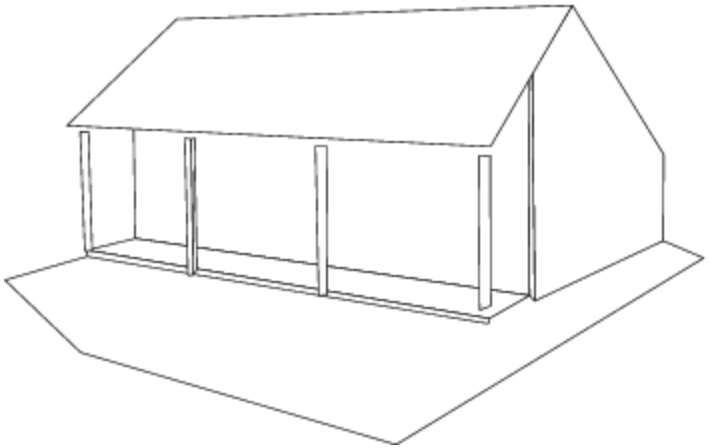
Similarity ambiguity

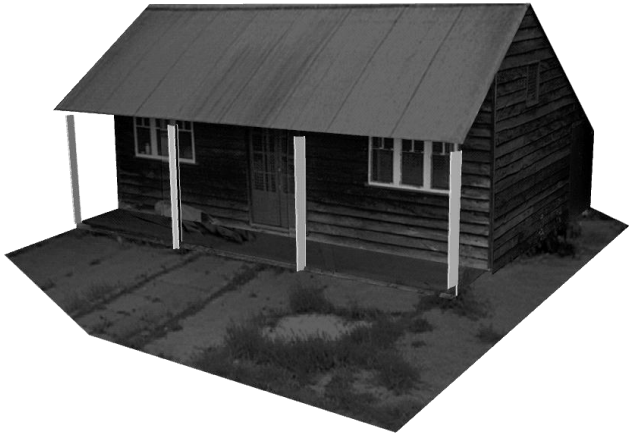


$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ}_S^{-1}\right)\left(\mathbf{Q}_S\mathbf{X}\right)$$

Similarity ambiguity

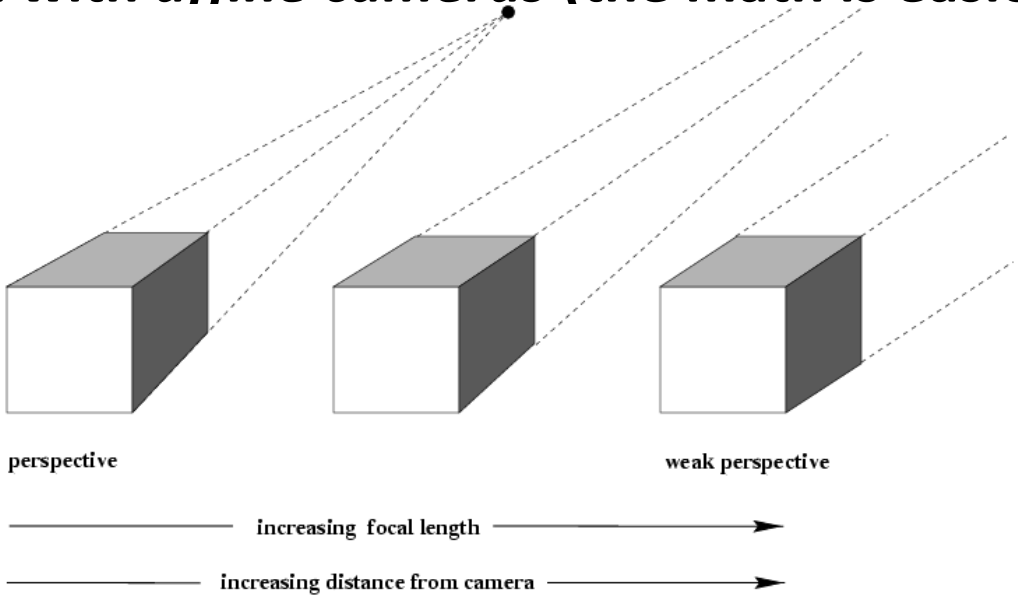




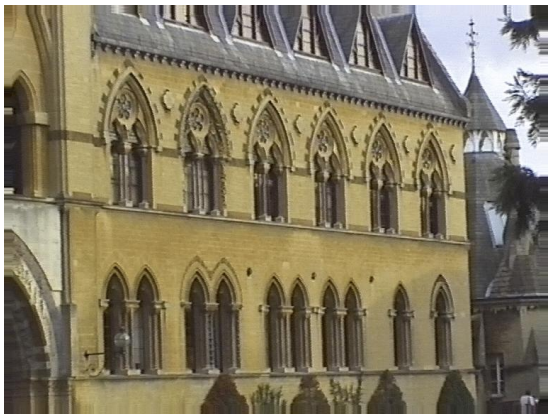


Structure from motion

- **Let's start with *affine cameras* (the math is easier)**

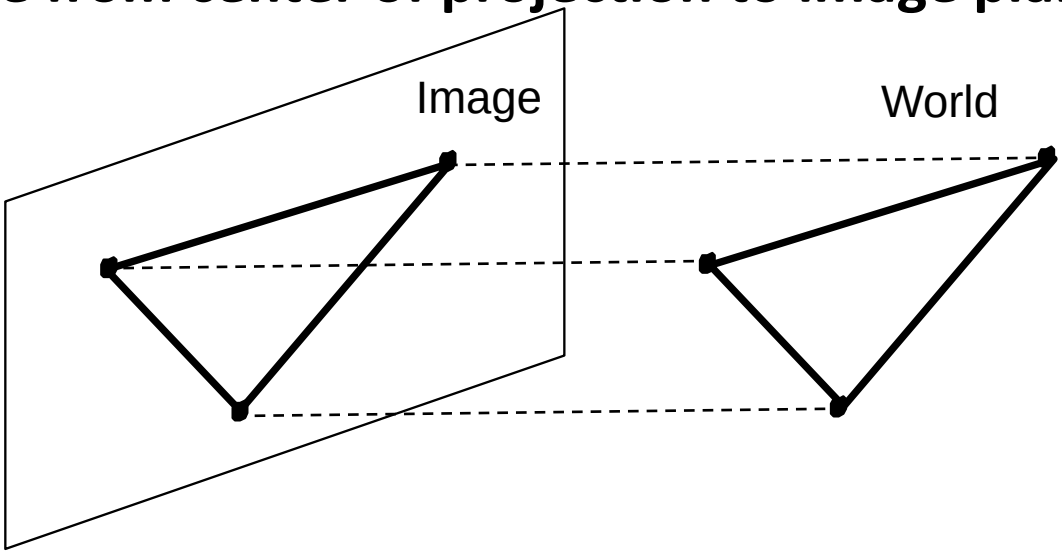


center at infinity



Recall: Orthographic Projection

- **Special case of perspective projection**
  - **Distance from center of projection to image plane is infinite**



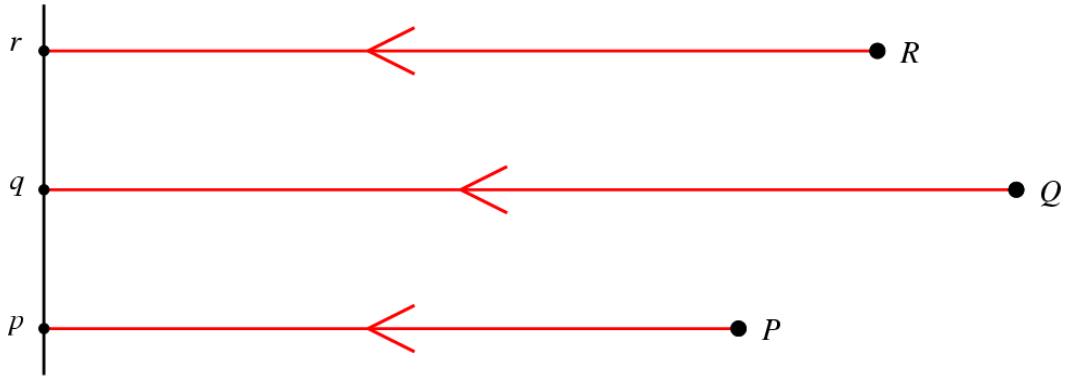
  - **Projection matrix:**

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

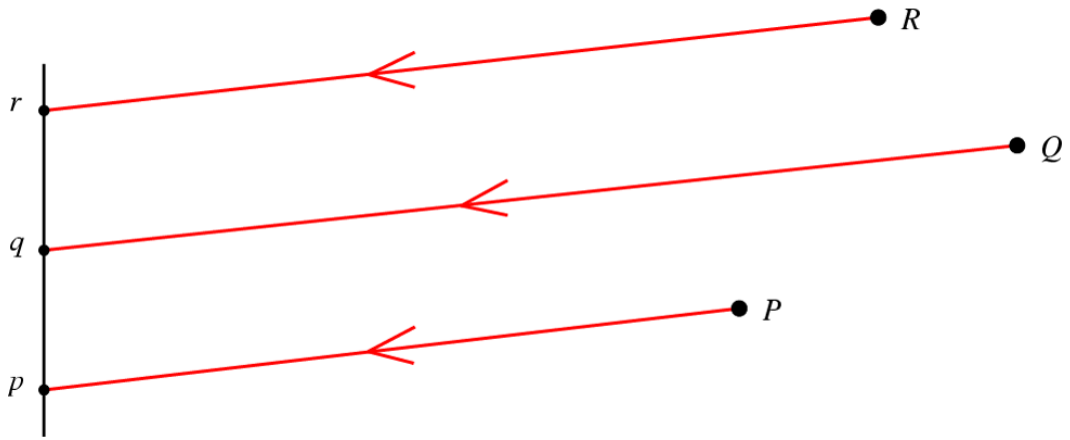
Slide by Steve Seitz

Affine cameras

Orthographic Projection



Parallel Projection

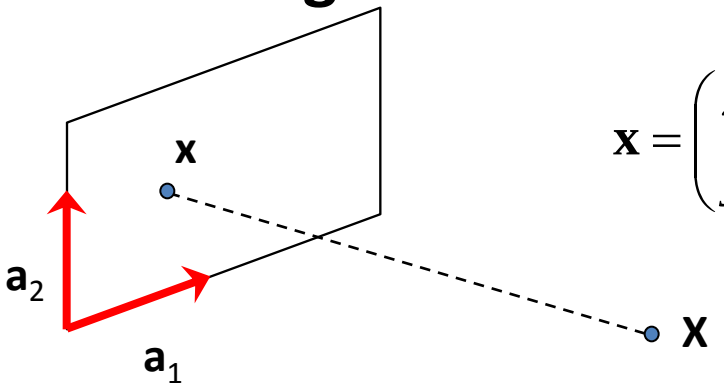


Affine cameras

- **A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:**

$$\mathbf{P} = [3\times 3\,\text{affine}]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}[4\times 4\,\text{affine}] = \begin{bmatrix}a_{11} & a_{12} & a_{13} & b_1\\ a_{21} & a_{22} & a_{23} & b_2\\ 0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{b}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}$$

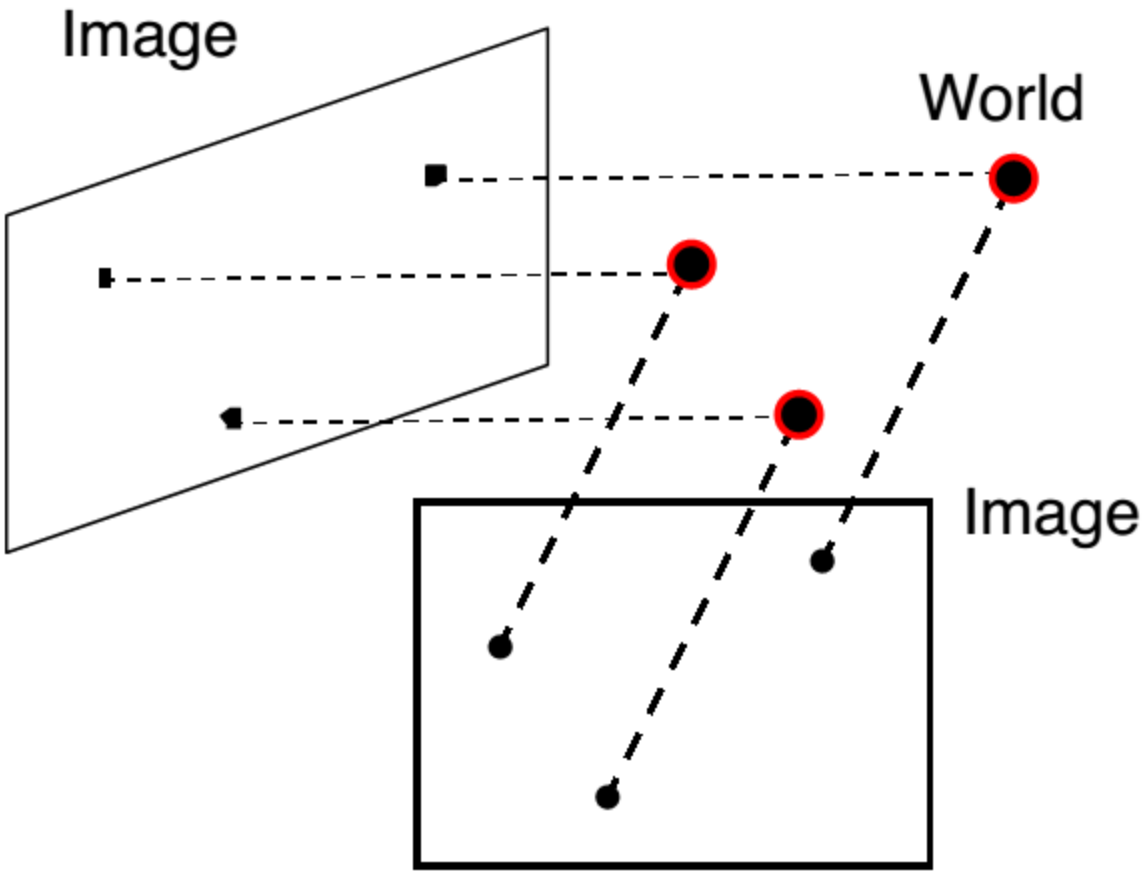
- **Affine projection is a linear mapping + translation in inhomogeneous coordinates**



$$\mathbf{x} = \begin{pmatrix}x\\ y\end{pmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\end{bmatrix}\begin{pmatrix}X\\ Y\\ Z\end{pmatrix} + \begin{pmatrix}b_1\\ b_2\end{pmatrix} = \mathbf{AX} + \mathbf{b}$$

Projection of world origin

Affine cameras



Affine structure from motion

- Given: *m* images of *n* fixed 3D points:
  - $\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i, \quad i = 1,\ldots, m, \; j = 1, \ldots, n$

- Problem: use the *mn* correspondences x_{ij} to estimate *m* projection matrices A_i and translation vectors b_i, and *n* points X_j
- We have 2*mn* knowns and 8*m* + 3*n* unknowns

- Two approaches:
  - Algebraic approach (affine epipolar geometry; estimate F; cameras; points)
  - Factorization method

A factorization method

- C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

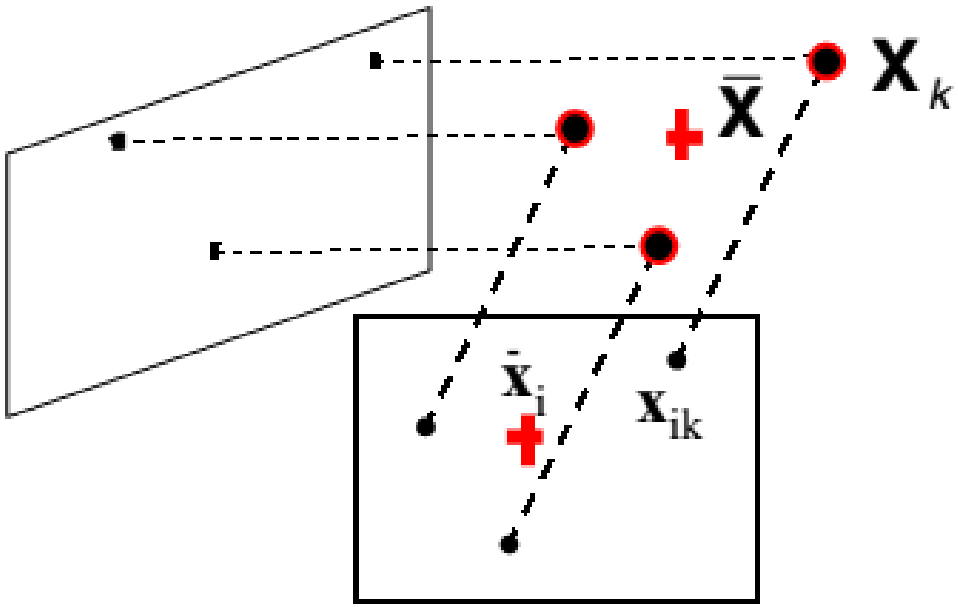
- **Two major steps:**
  - **Data centering**
  - **Factorization**

Affine structure from motion

- **Centering: subtract the centroid of the image points**

$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i\right)$$

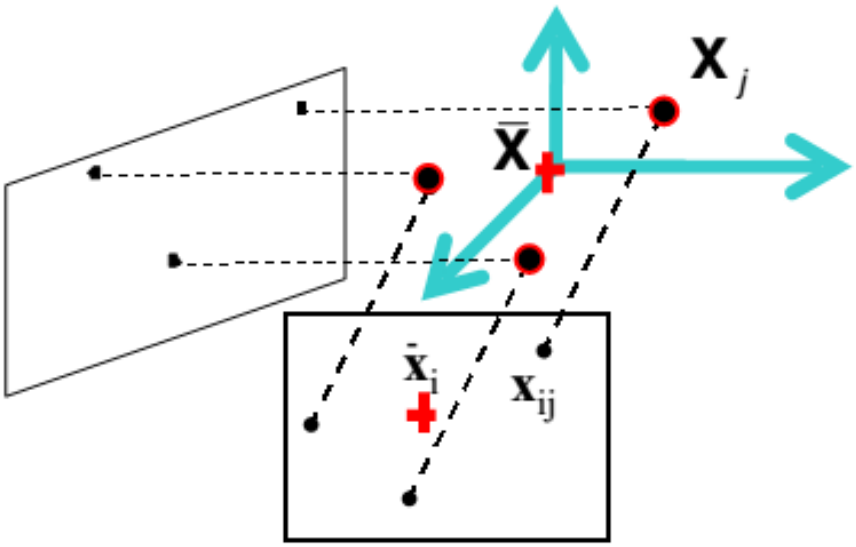
$$= \mathbf{A}_i\left(\mathbf{X}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k\right) = \mathbf{A}_i\hat{\mathbf{X}}_j$$



Affine structure from motion

- For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points
- After centering, each normalized point x_{ij} is related to the 3D point X_i by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \mathbf{X}_j$$



Affine structure from motion

- **Let's create a 2*m* × *n* data (measurement) matrix:**

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}$$

cameras (2*m*)

points (*n*)

Affine structure from motion

- **Let's create a 2*m* × *n* data (measurement) matrix:**

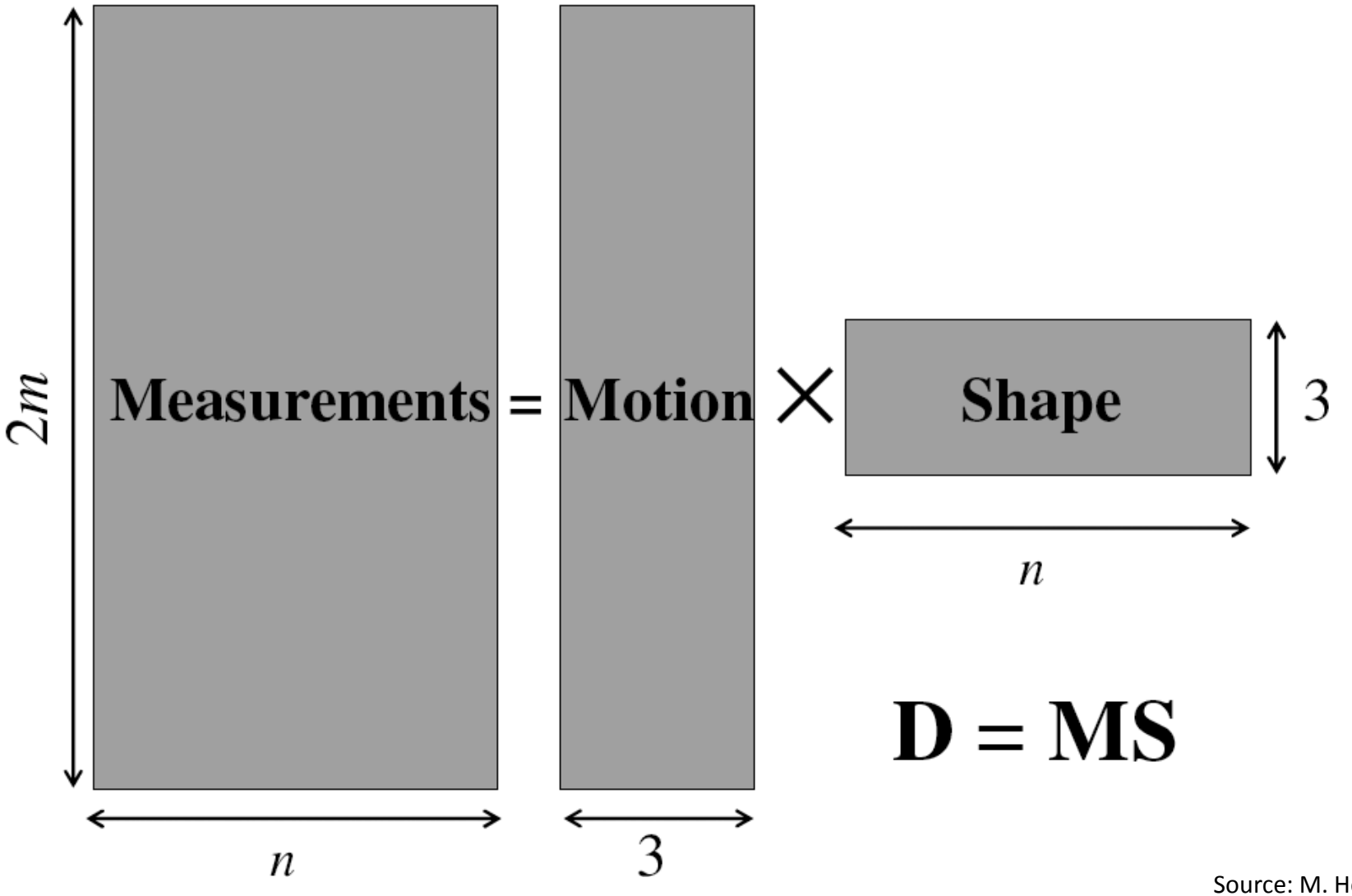
$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

cameras ($2m \times 3$)

points ($3 \times n$)

The measurement matrix $\mathbf{D} = \mathbf{MS}$ must have rank 3!

Factorizing the measurement matrix

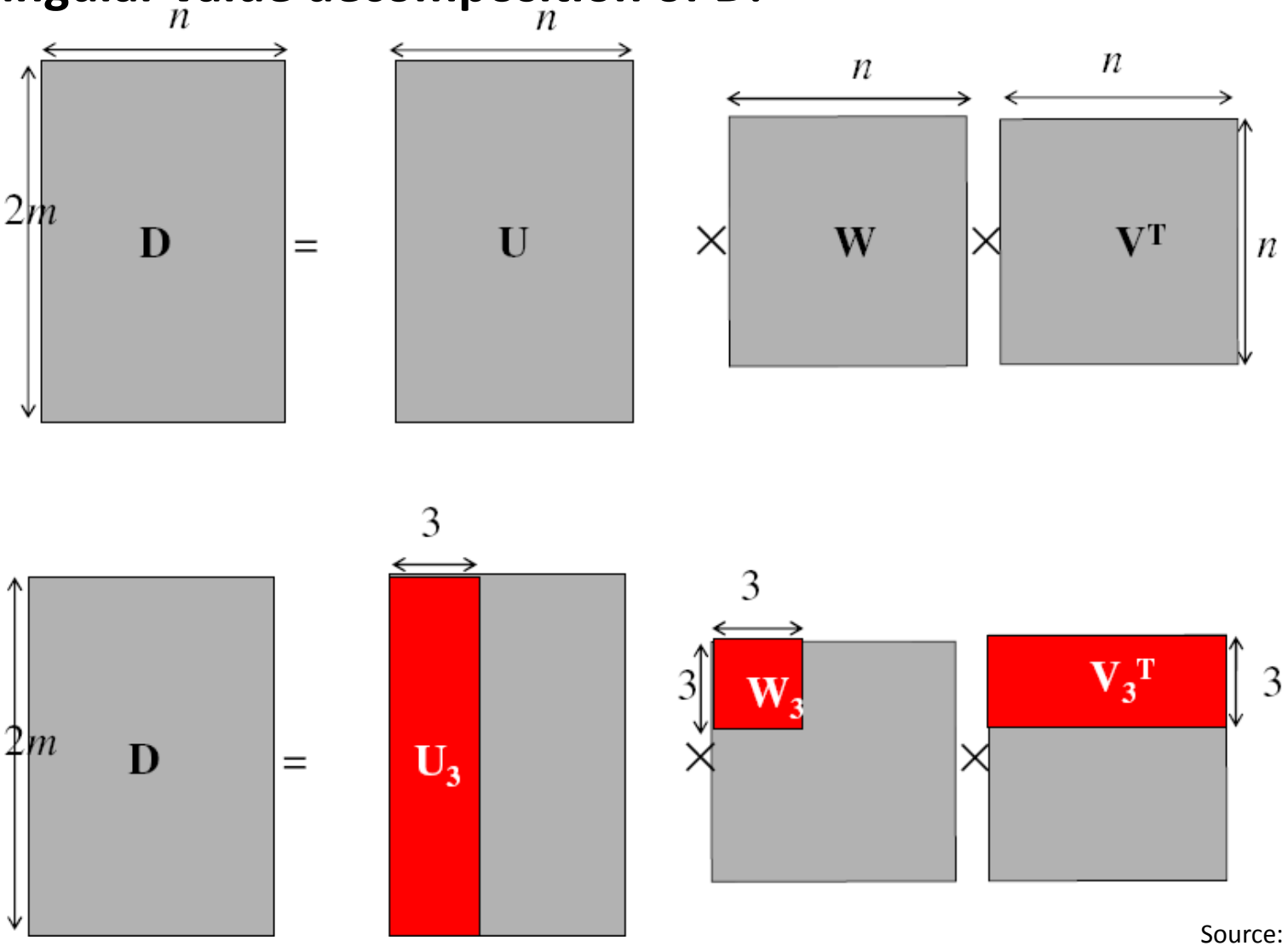


$$\mathbf{D} = \mathbf{MS}$$

Source: M. Hebert

Factorizing the measurement matrix

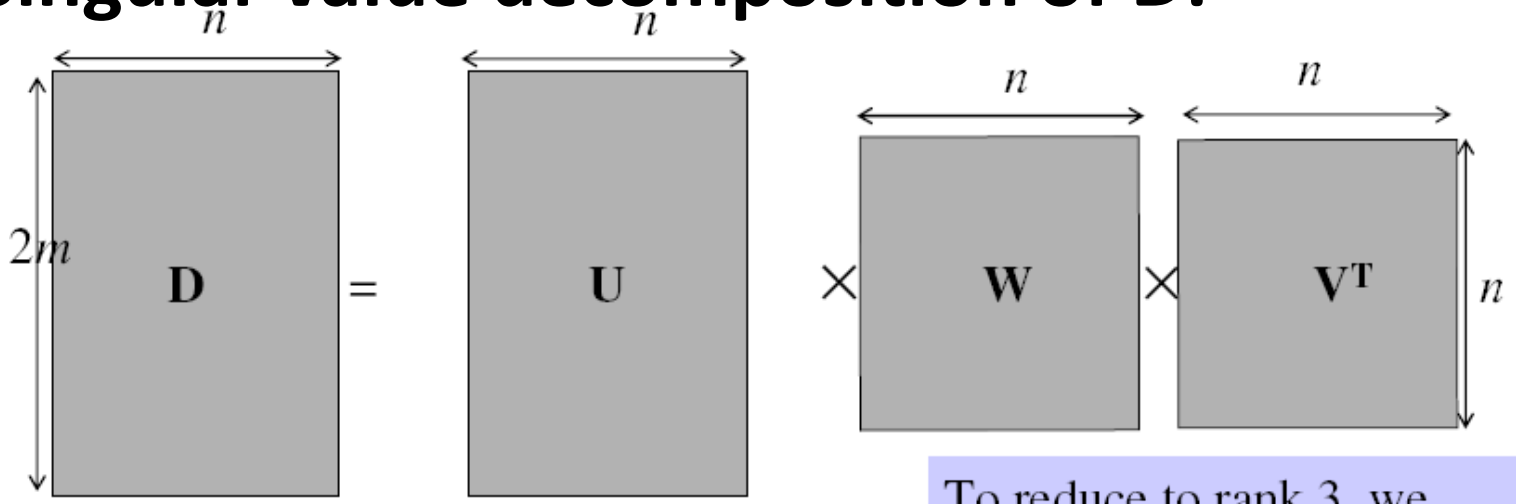
- **Singular value decomposition of D:**



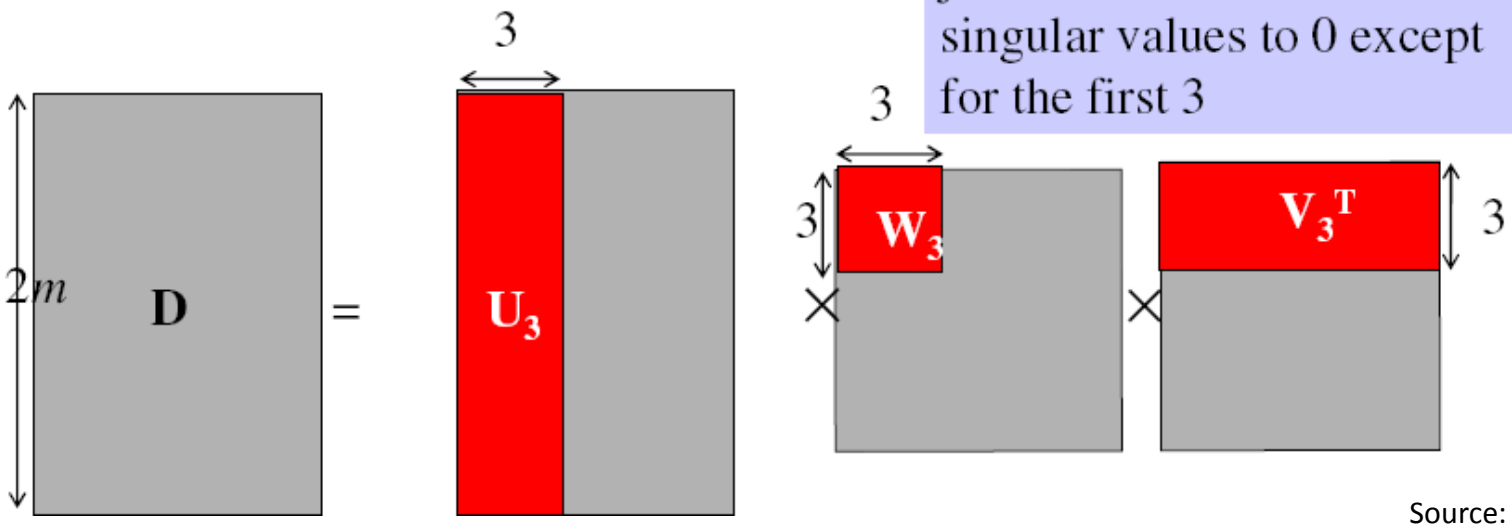
Source: M. Hebert

Factorizing the measurement matrix

- **Singular value decomposition of D:**



To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3



Source: M. Hebert

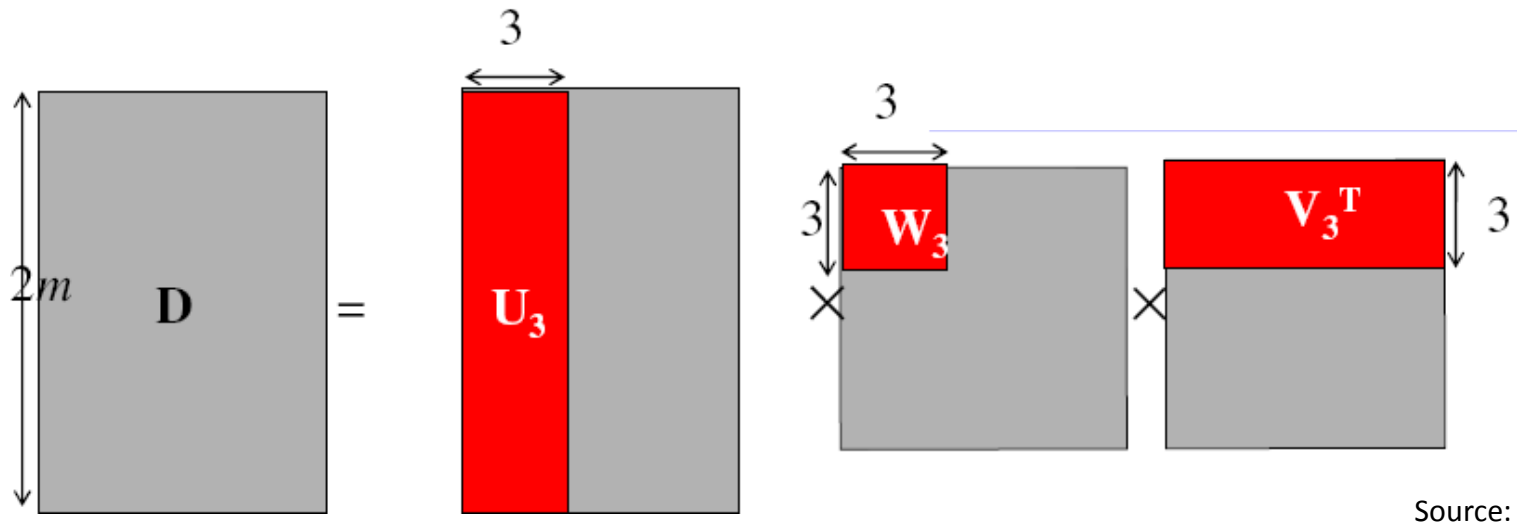
Factorizing the measurement matrix

- **D has rank>3 because of:**
  - **measurement noise**
  - **affine approximation**

Theorem: When $\mathbf{D}$ has a rank greater than 3, $\mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T$ is the best possible rank-3 approximation of $\mathbf{D}$ in the sense of the Frobenius norm.

$$\mathbf{D} = \mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T \quad \begin{cases} \mathbf{M} \approx \mathbf{U}_3 \\ \mathbf{S} \approx \mathbf{W}_3\mathbf{V}_3^T \end{cases}$$

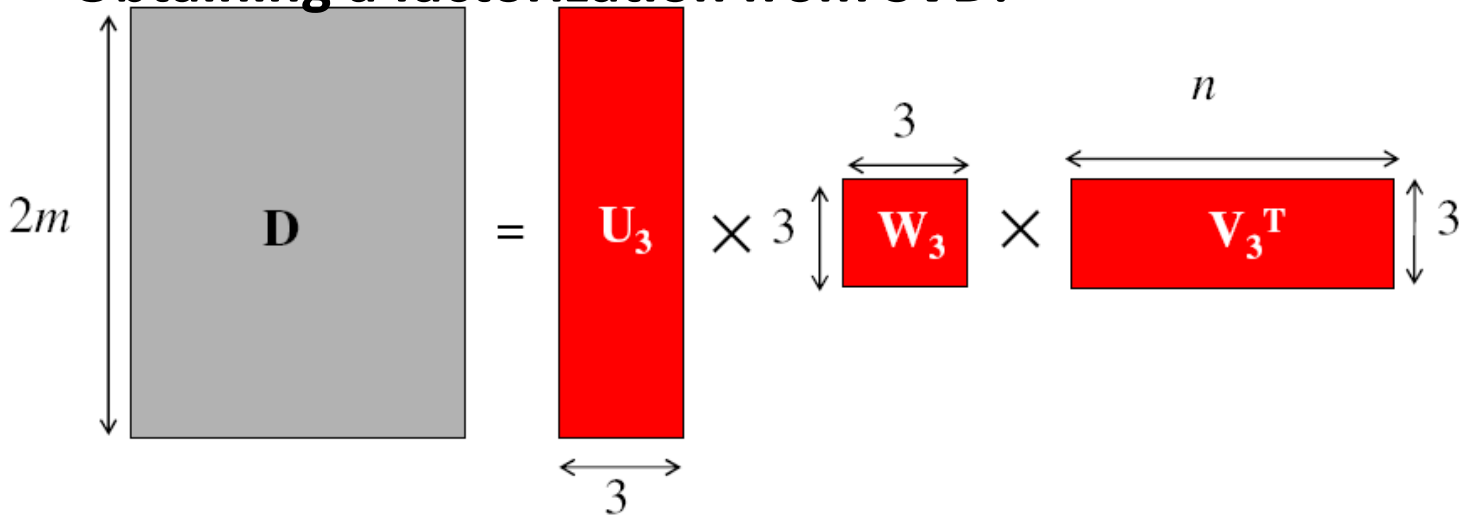
$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$$



Source: M. Hebert

Factorizing the measurement matrix

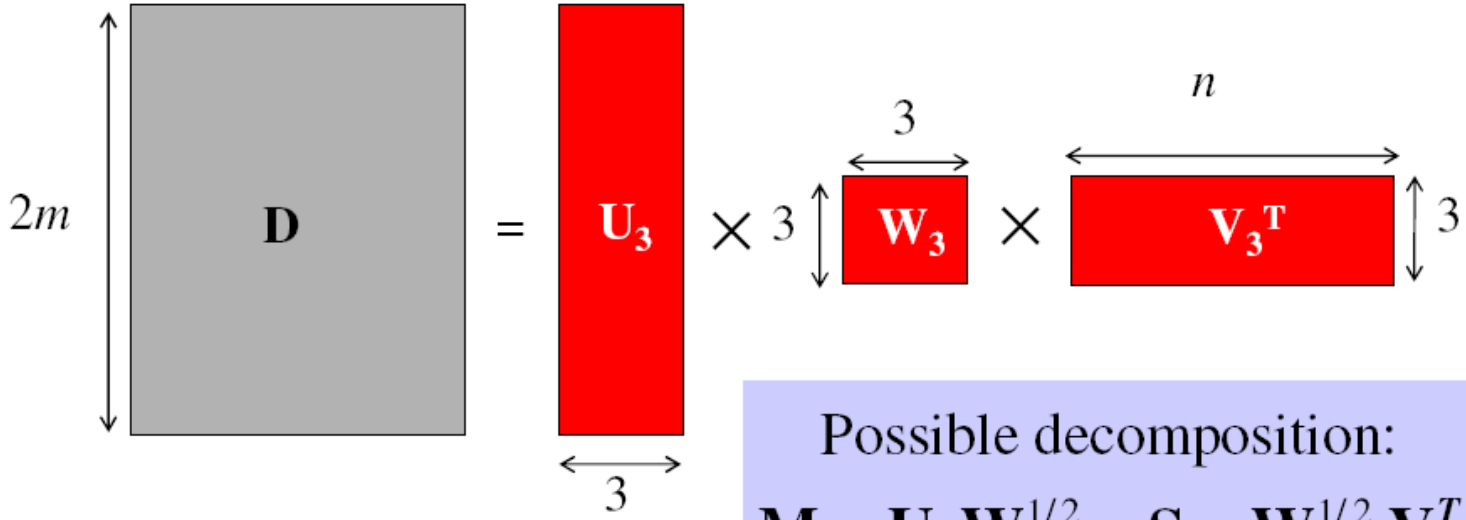
- **Obtaining a factorization from SVD:**



Source: M. Hebert

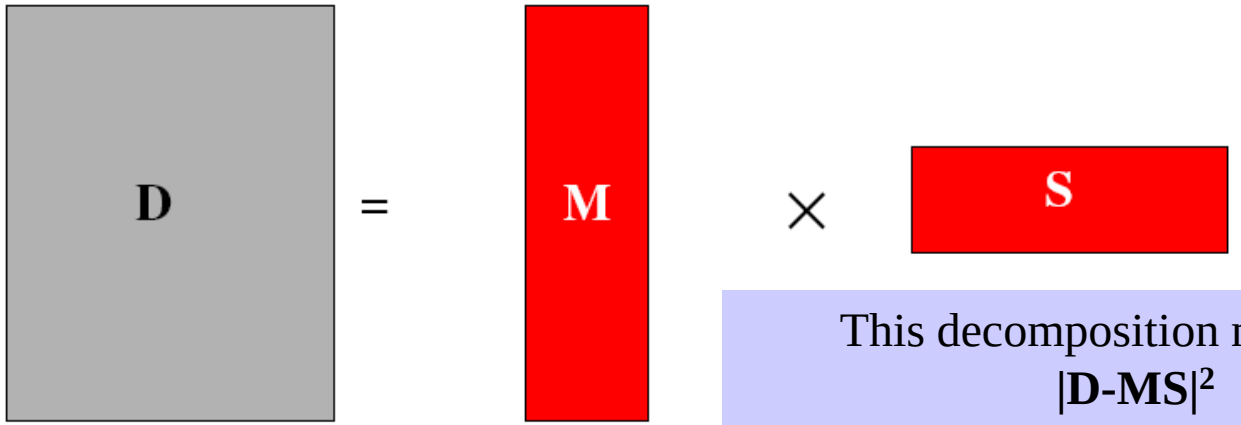
Factorizing the measurement matrix

- **Obtaining a factorization from SVD:**



Possible decomposition:

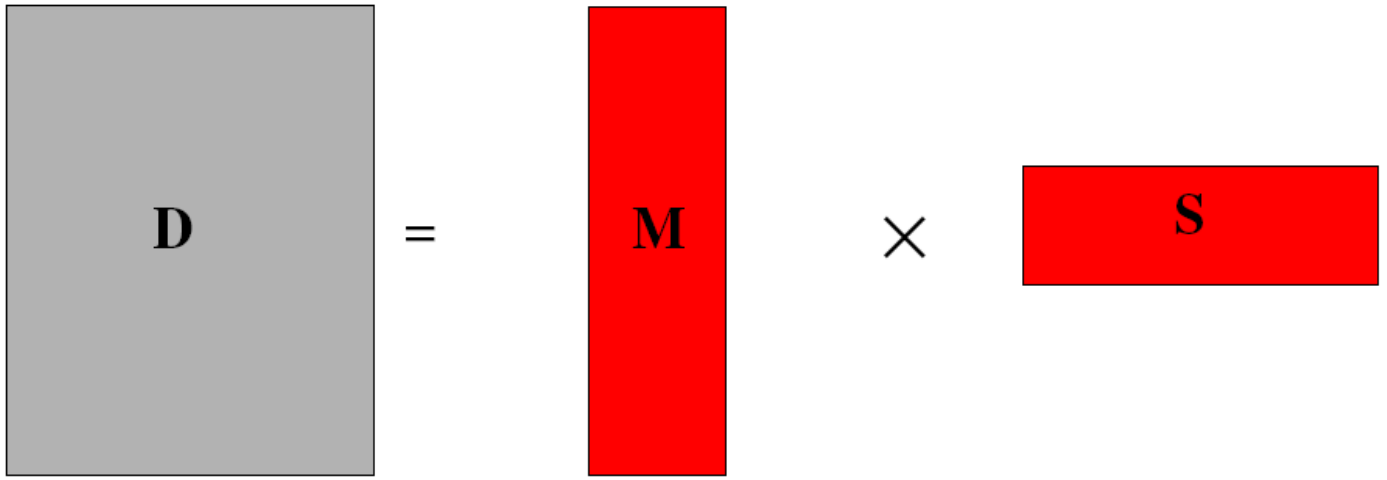
$$\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \qquad \mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T$$



This decomposition minimizes
$|\mathbf{D}-\mathbf{MS}|^2$

Source: M. Hebert

Affine ambiguity

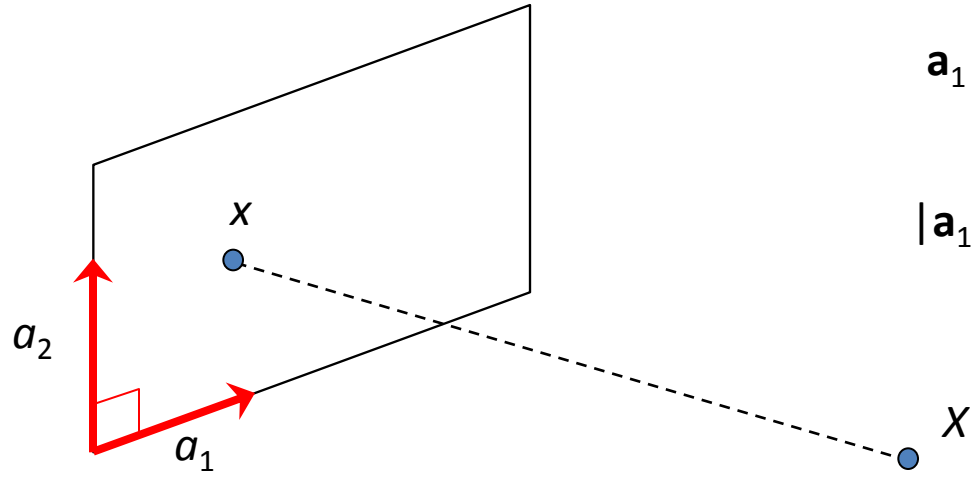


- The decomposition is not unique. We get the same D by using any 3×3 matrix C and applying the transformations M → MC, S →C^{-1}S
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example)

Source: M. Hebert

Eliminating the affine ambiguity

- **Orthographic: image axes are perpendicular and scale is 1**



$\mathbf{a}_1 \cdot \mathbf{a}_2 = 0$

$|\mathbf{a}_1|^2 = |\mathbf{a}_2|^2 = 1$

- **This translates into 3*m* equations in $L = CC^T$:**
  - $A_i L A_i^T = Id, \qquad i = 1, \ldots, m$
  - **Solve for L**
  - **Recover C from L by Cholesky decomposition: $L = CC^T$**
  - **Update M and S: $M = MC$, $S = C^{-1}S$**

Source: M. Hebert

Algorithm summary

- Given: *m* images and *n* features x_{ij}
- For each image *i*, *c*enter the feature coordinates
- Construct a $2m \times n$ measurement matrix D:
  - Column *j* contains the projection of point *j* in all views
  - Row *i* contains one coordinate of the projections of all the *n* points in image *i*
- Factorize D:
  - Compute SVD: $D = U W V^T$
  - Create U_3 by taking the first 3 columns of U
  - Create V_3 by taking the first 3 columns of V
  - Create W_3 by taking the upper left 3×3 block of W
- Create the motion and shape matrices:
  - $M = U_3 W_3^{½}$ and $S = W_3^{½} V_3^T$ (or $M = U_3$ and $S = W_3 V_3^T$)
- Eliminate affine ambiguity

Source: M. Hebert

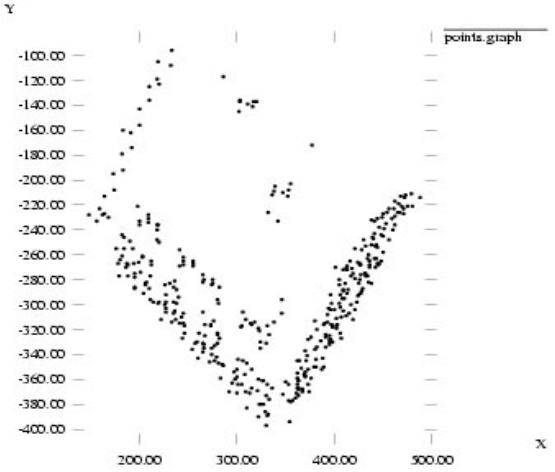
Reconstruction results

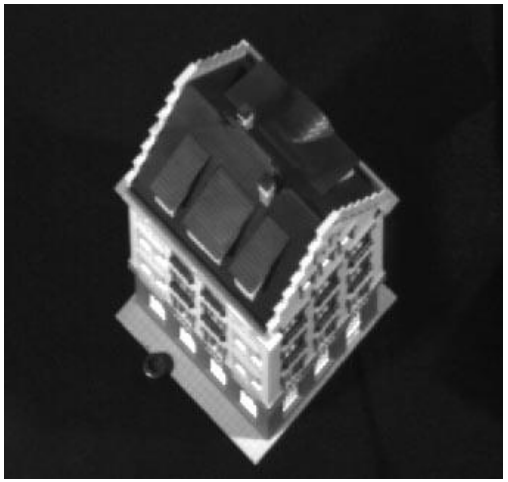
1

60

120

150

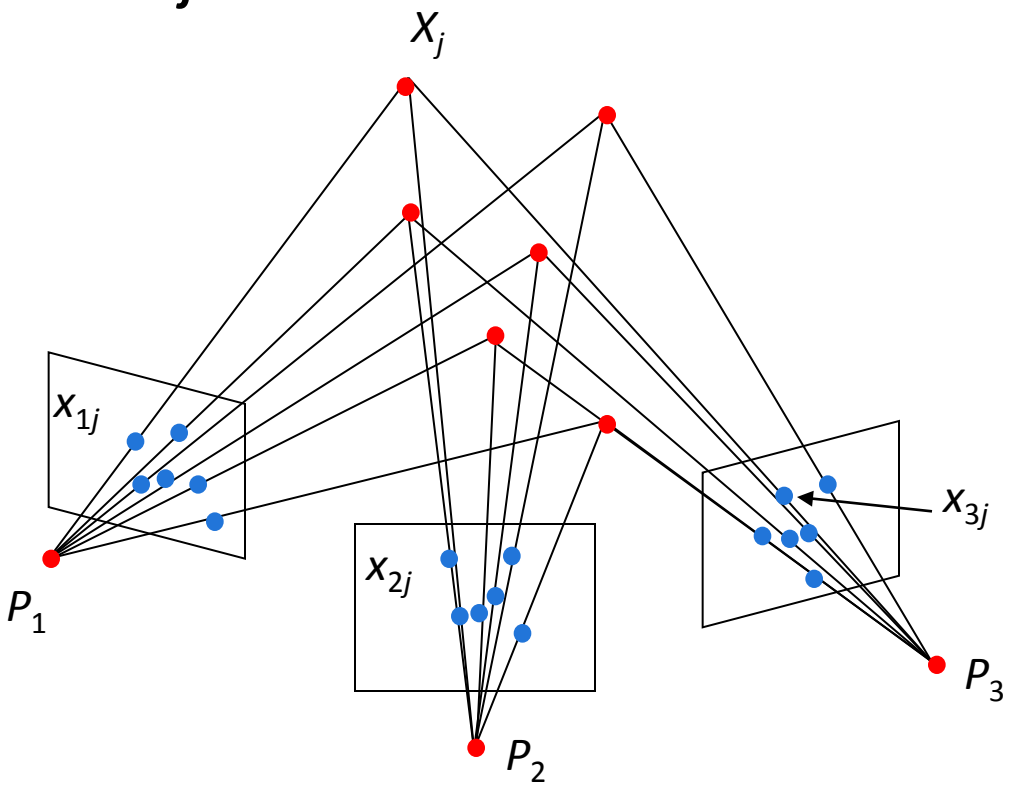




C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Projective structure from motion

- Given: *m* images of *n* fixed 3D points
  - $z_{ij}\,\mathbf{x}_{ij} = \mathbf{P}_i\,\mathbf{X}_j, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n$
- 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
  - $z_{ij}\,\mathbf{x}_{ij} = \mathbf{P}_i\,\mathbf{X}_j, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n$
- 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 4x4 projective transformation Q:
  - $\mathbf{X} \rightarrow \mathbf{QX}$, $\mathbf{P} \rightarrow \mathbf{PQ}^{-1}$
- We can solve for structure and motion when
  - $2mn >= 11m + 3n - 15$
- 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: $[\mathbf{I}|\mathbf{0}]$**
- **Second camera matrix: $[\mathbf{A}|\mathbf{b}]$**
- **Then**

$$z\mathbf{x} = [\mathbf{I}\,|\,\mathbf{0}]\mathbf{X}, \quad z'\mathbf{x}' = [\mathbf{A}\,|\,\mathbf{b}]\mathbf{X}$$

$$z'\mathbf{x}' = \mathbf{A}[\mathbf{I}\,|\,\mathbf{0}]\mathbf{X} + \mathbf{b} = z\mathbf{A}\mathbf{x} + \mathbf{b}$$

$$z'\mathbf{x}' \times \mathbf{b} = z\mathbf{A}\mathbf{x} \times \mathbf{b}$$

$$(z'\mathbf{x}' \times \mathbf{b}) \cdot \mathbf{x}' = (z\mathbf{A}\mathbf{x} \times \mathbf{b}) \cdot \mathbf{x}'$$

$$\mathbf{x}'^{\mathrm{T}}[\mathbf{b}_\times]\mathbf{A}\mathbf{x} = 0$$

$$\mathbf{F} = [\mathbf{b}_\times]\mathbf{A}$$ $\mathbf{b}$: epipole ($\mathbf{F}^{\mathrm{T}}\mathbf{b} = 0$), $\mathbf{A} = -[\mathbf{b}_\times]\mathbf{F}$

F&P sec. 8.3

Algebraic approach (2-view case)

1. Compute the fundamental matrix F from two views (eg. 8 point algorithm)

1. Use F to estimate projective cameras
   - Compute b and A from F
   - Use b and A to estimate projective cameras

2. Use these cameras to triangulate and estimate points in 3D

Projective factorization

$$\mathbf{D} = \begin{bmatrix} z_{11}\mathbf{x}_{11} & z_{12}\mathbf{x}_{12} & \cdots & z_{1n}\mathbf{x}_{1n} \\ z_{21}\mathbf{x}_{21} & z_{22}\mathbf{x}_{22} & \cdots & z_{2n}\mathbf{x}_{2n} \\ & & \ddots & \\ z_{m1}\mathbf{x}_{m1} & z_{m2}\mathbf{x}_{m2} & \cdots & z_{mn}\mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

cameras ($3m \times 4$)

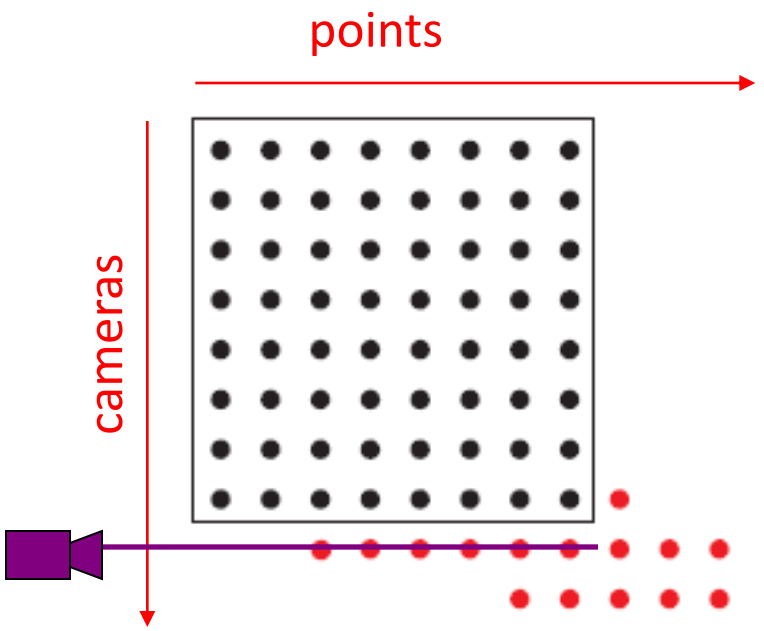
points ($4 \times n$)

$\mathbf{D} = \mathbf{MS}$ has rank 4

- **If we knew the depths *z*, we could factorize D to estimate M and S**
- **If we knew M and S, we could solve for *z***
- **Solution: iterative approach (alternate between above two steps)**

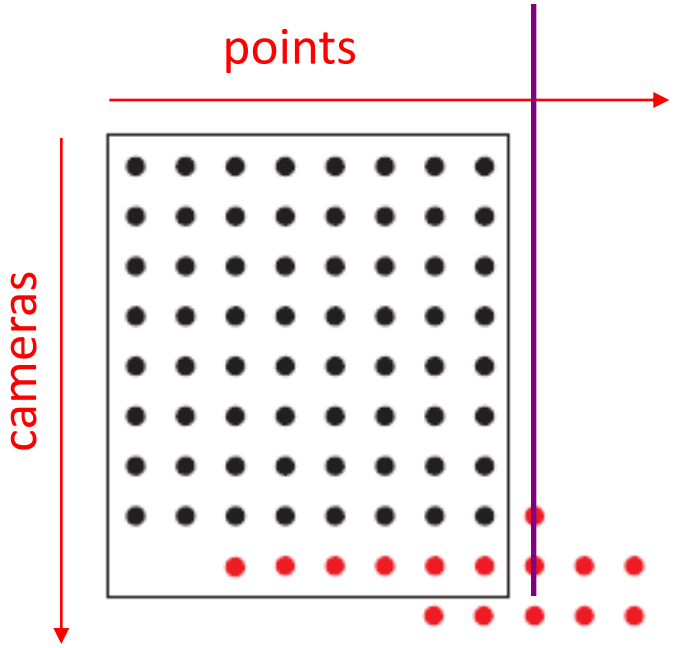
Sequential structure from motion

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



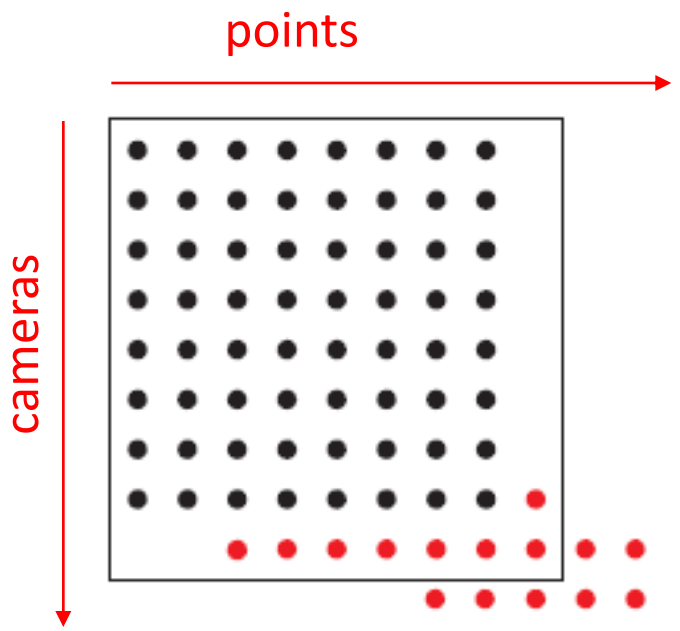
Sequential structure from motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- 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

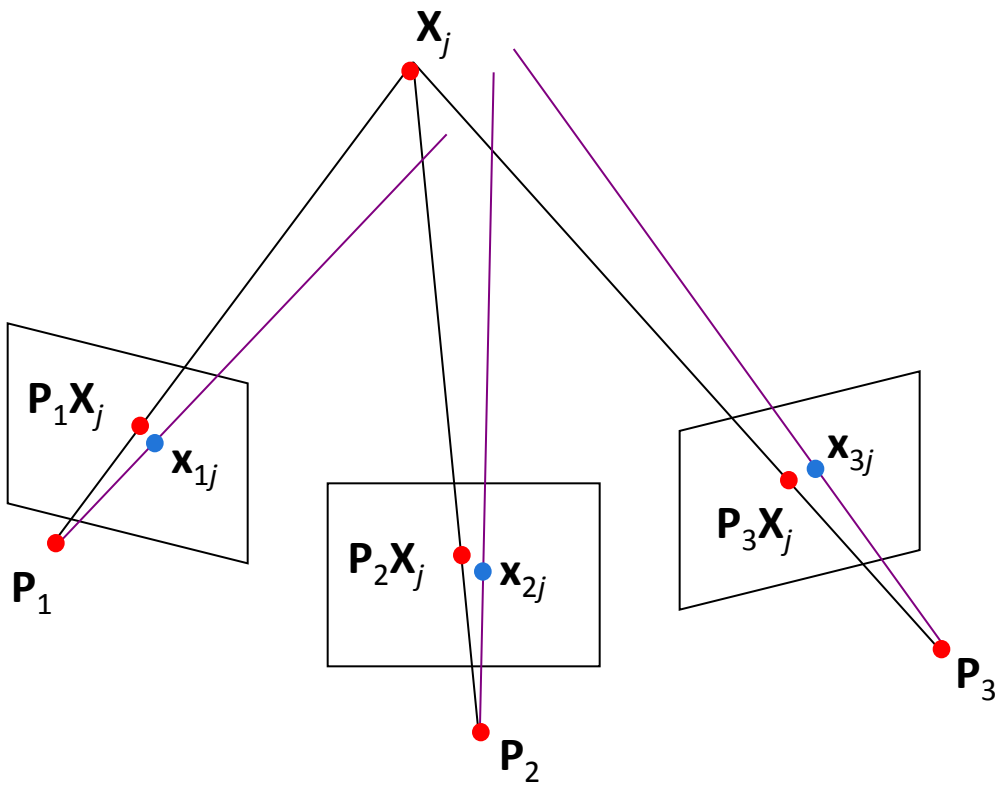
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize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



Bundle adjustment（光束平差法）

- **Non-linear method for refining structure and motion**
- **Minimizing reprojection error**

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$



Bundle adjustment

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

predicted image location

observed image location

indicator variable: is point *i* visible in image *j* ?

- Minimizing this function is called *bundle adjustment*
  - Optimized using non-linear least squares, e.g. Levenberg-Marquardt

Bundle Adjustment

- SFM = Nonlinear Least Squares problem
- Minimize through
  - Gradient Descent
  - Conjugate Gradient
  - Gauss-Newton
  - Levenberg Marquardt is common method

- Prone to local minima

Bundle adjustment

- Advantages
  - Handle large number of views
  - Handle missing data

- Limitations
  - Large minimization problem (parameters grow with number of views)
  - Requires good initial condition

- Used as the final step of SFM (i.e., after the factorization or algebraic approach)
- Factorization or algebraic approaches provide a initial solution for optimization problem

Overview

Structure from Motion (SfM)

Large scale Structure from Motion

Other approaches to obtaining 3D structure

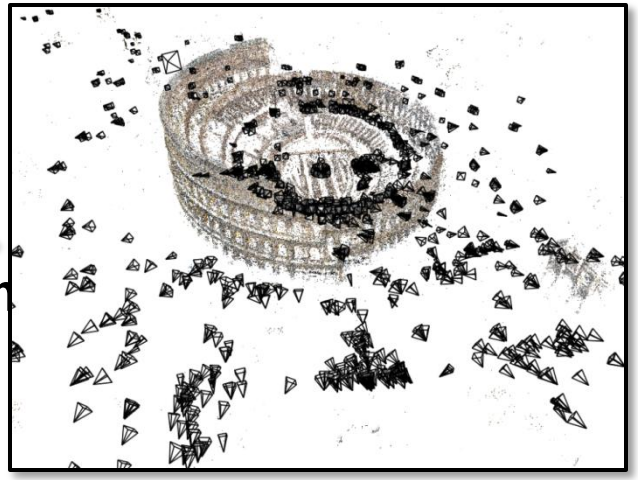
Large-scale Structure from motion

Given many images from photo collections how can we

a) figure out where they were all taken from?

b) build a 3D model of the scene?

T **e from**



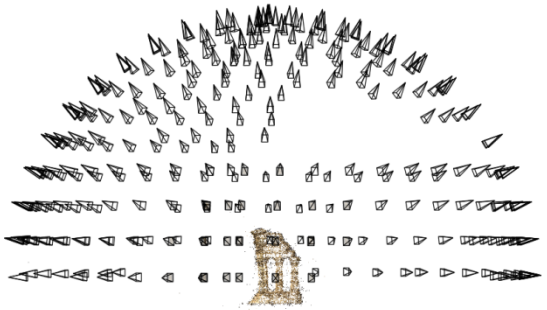
Slides from N. Snavely

Large-scale structure from motion

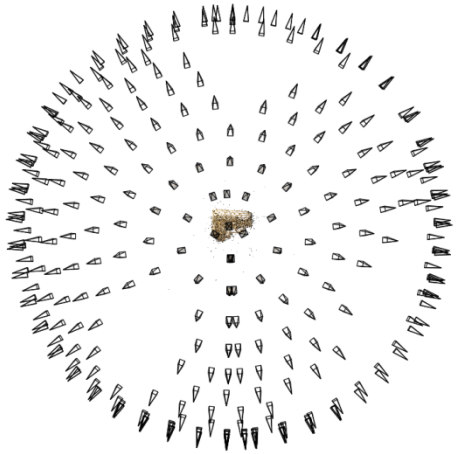
Dubrovnik, Croatia. 4,619 images (out of an initial 57,845).
Total reconstruction time: 23 hours
Number of cores: 352

Slide: N. Snavely

Structure from motion



Reconstruction (side)



(top)

- Input: images with points in correspondence
  $p_{i,j} = (u_{i,j}, v_{i,j})$

- Output
  - structure: 3D location $\mathbf{x}_i$ for each point p_i
  - motion: camera parameters $\mathbf{R}_j$, $\mathbf{t}_j$ possibly $\mathbf{K}_j$

- Objective function: minimize *reprojection error*

Photo Tourism

- **Structure from motion on Internet photo collections**
- **http://phototour.cs.washington.edu/**

Photo tourism is a system for browsing large collections of photographs in 3D. Our approach takes as input large collections of images from either personal photo collections or Internet photo sharing sites **(a)**, and automatically computes each photo's viewpoint and a sparse 3D model of the scene **(b)**. Our photo explorer interface enables the viewer to interactively move about the 3D space by seamlessly transitioning between photographs, based on user control **(c)**.

Live Demo

***New* Visit our PhotoCity project where you can help us reconstruct your neck of the woods.**

See our work on Finding Paths through the World's Photos.

Our structure from motion code is also now available at the Bundler homepage.

Photo Tourism

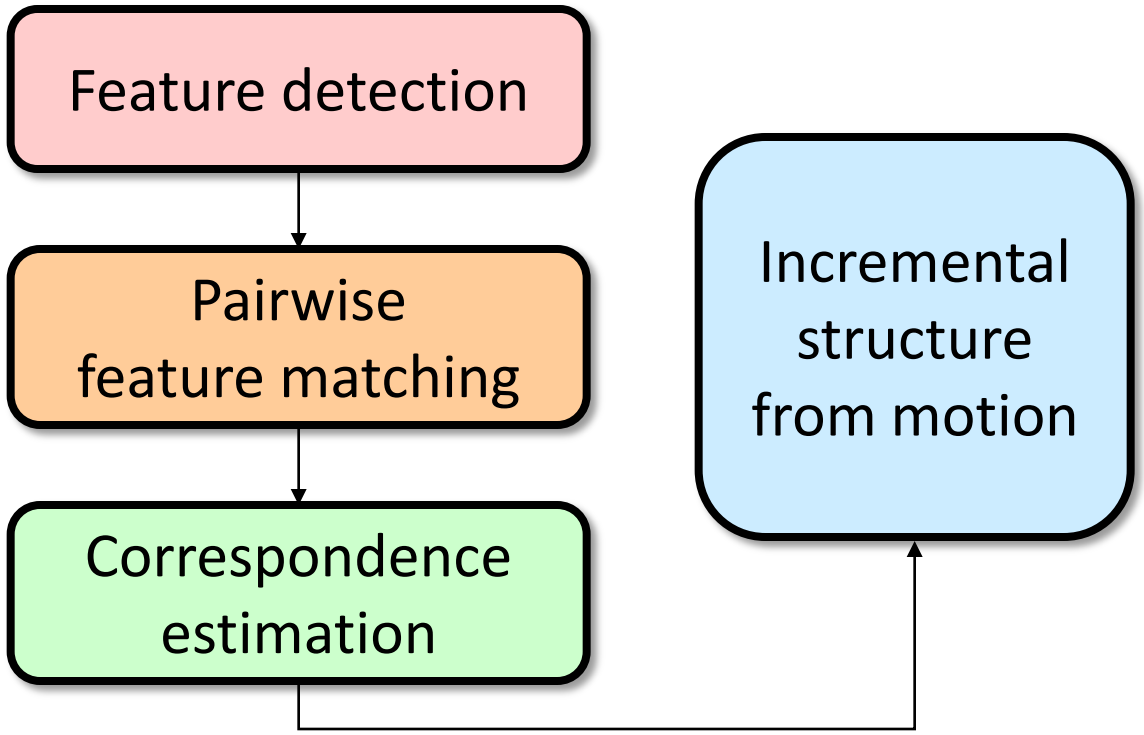
Photo Tourism

Exploring photo collections in 3D

Noah Snavely Steven M. Seitz Richard Szeliski
University of Washington *Microsoft Research*

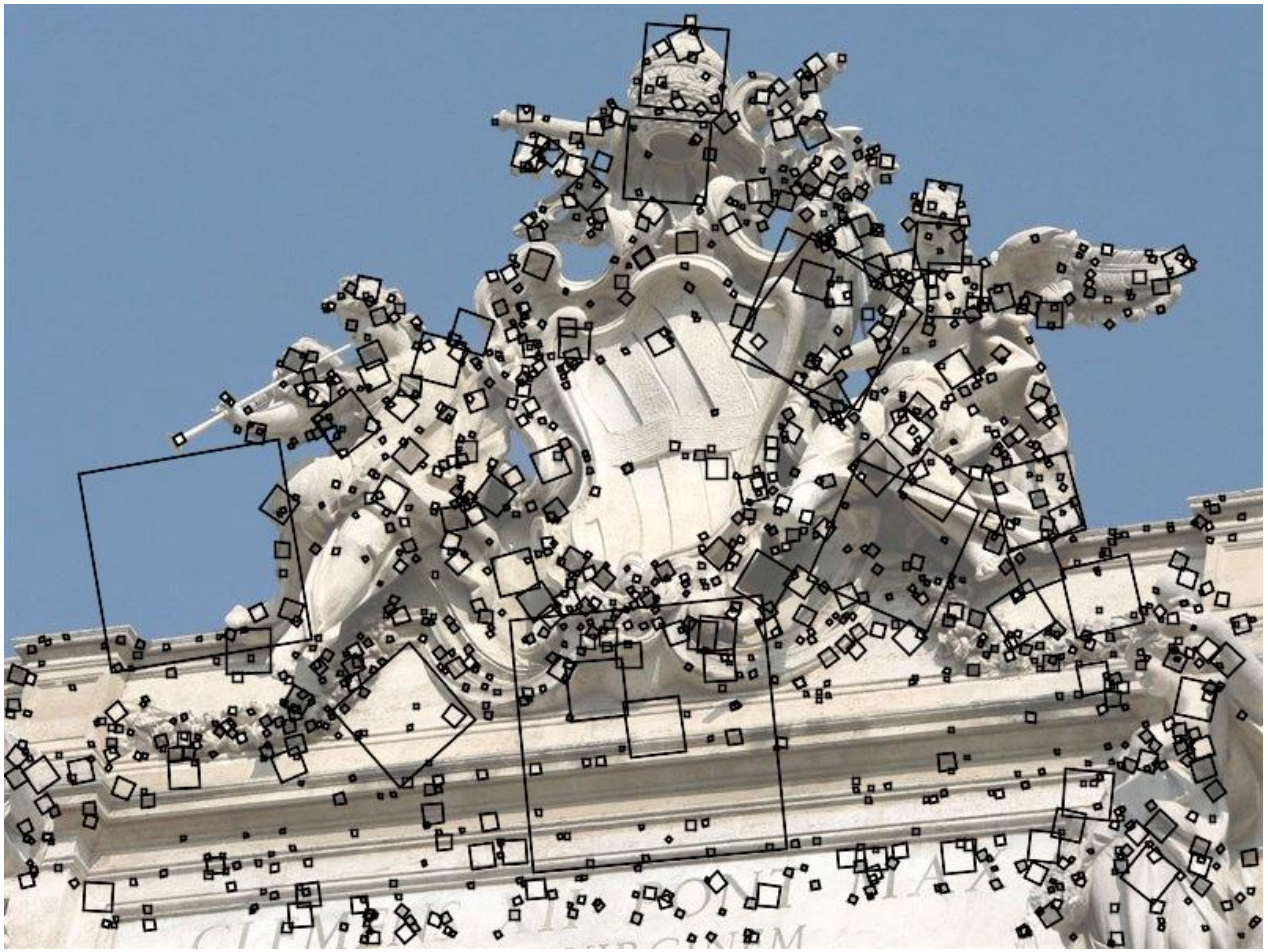
SIGGRAPH 2006

Scene reconstruction



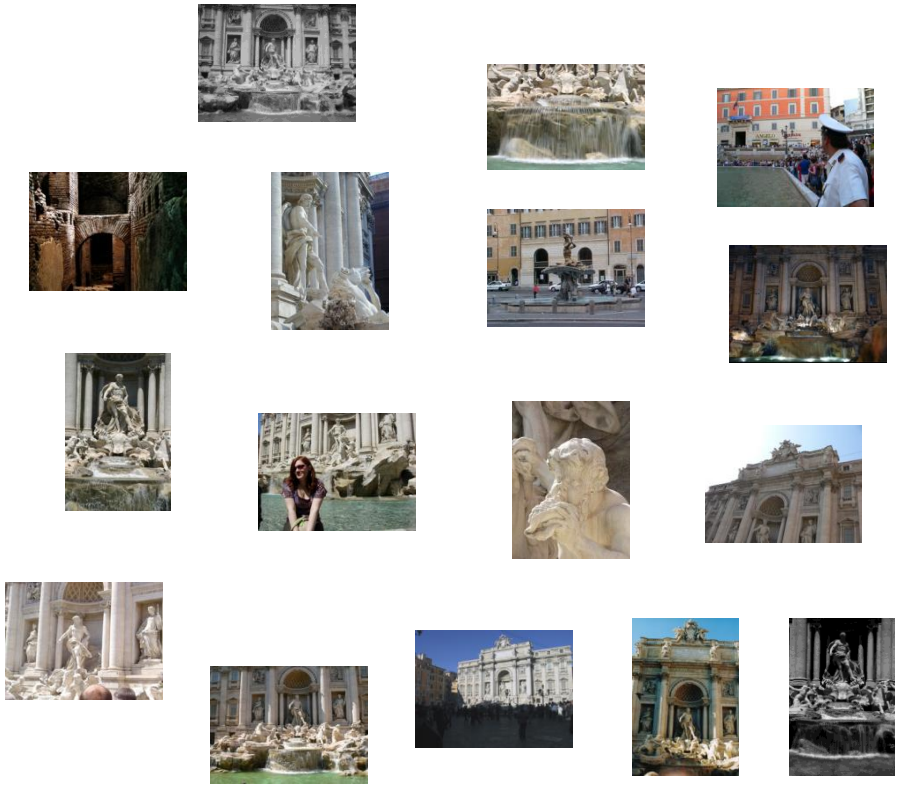
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



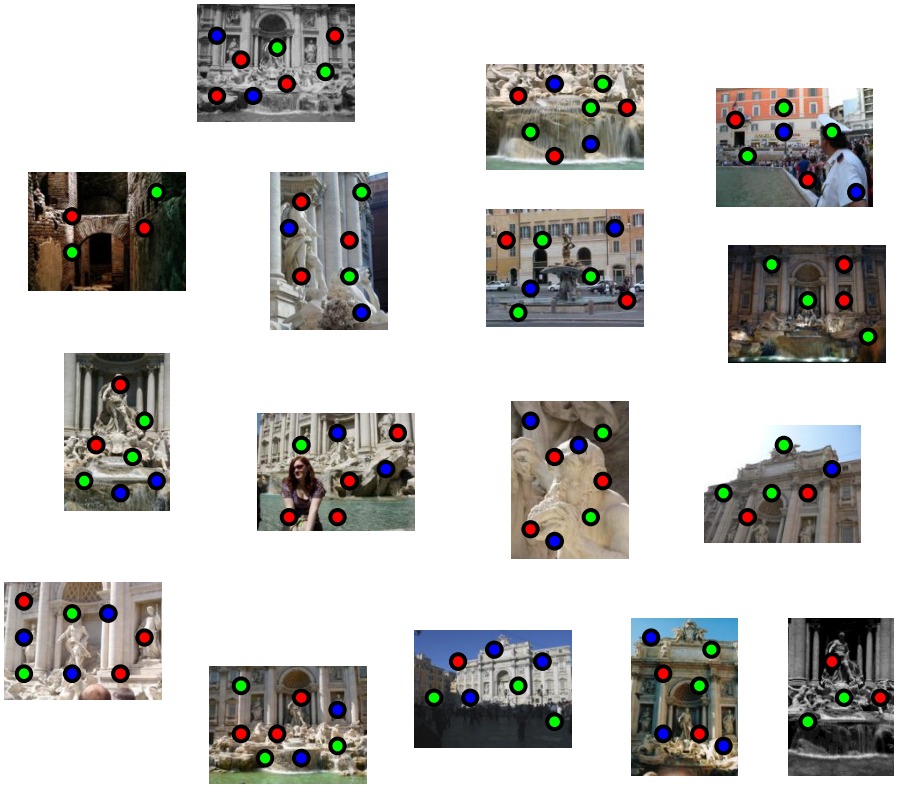
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



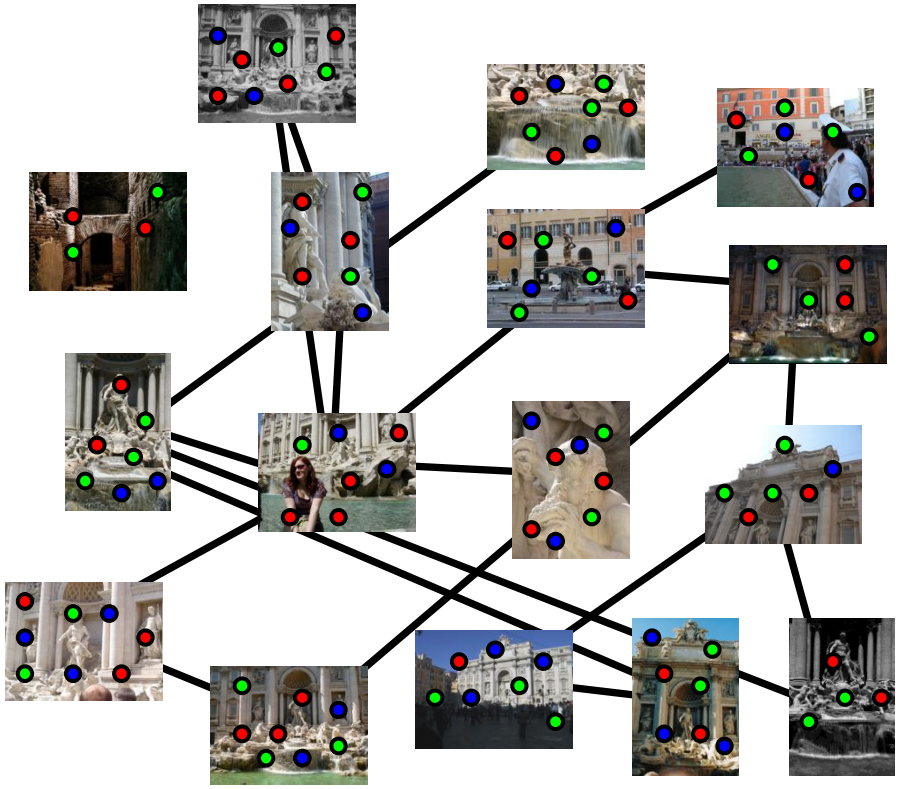
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



Feature matching

Match features between each pair of images



Feature matching

Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs

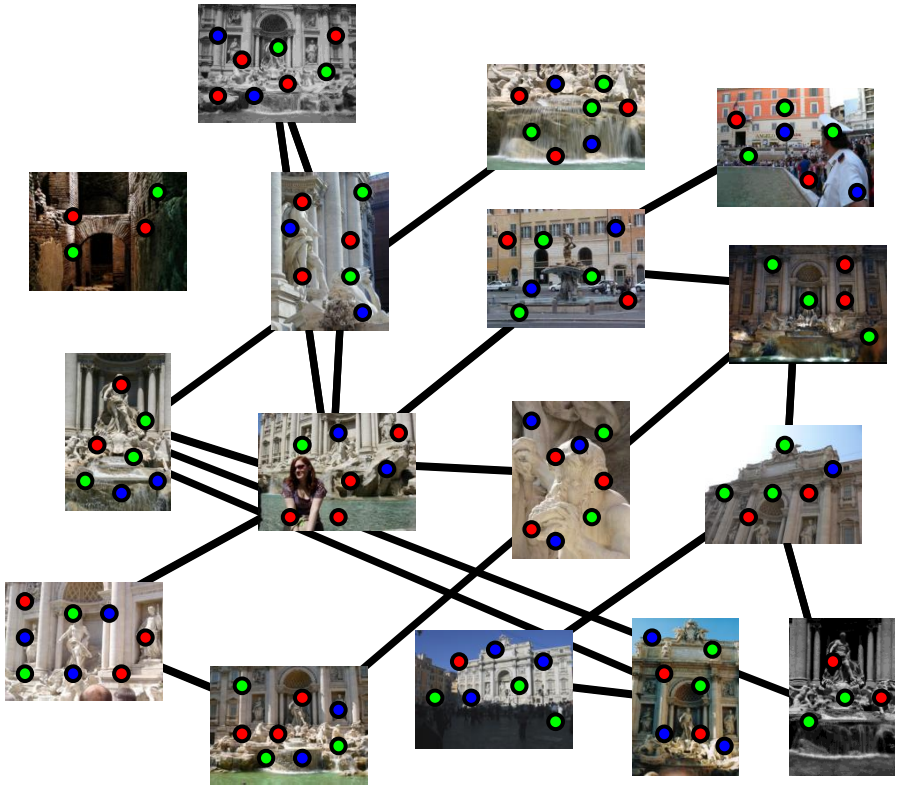
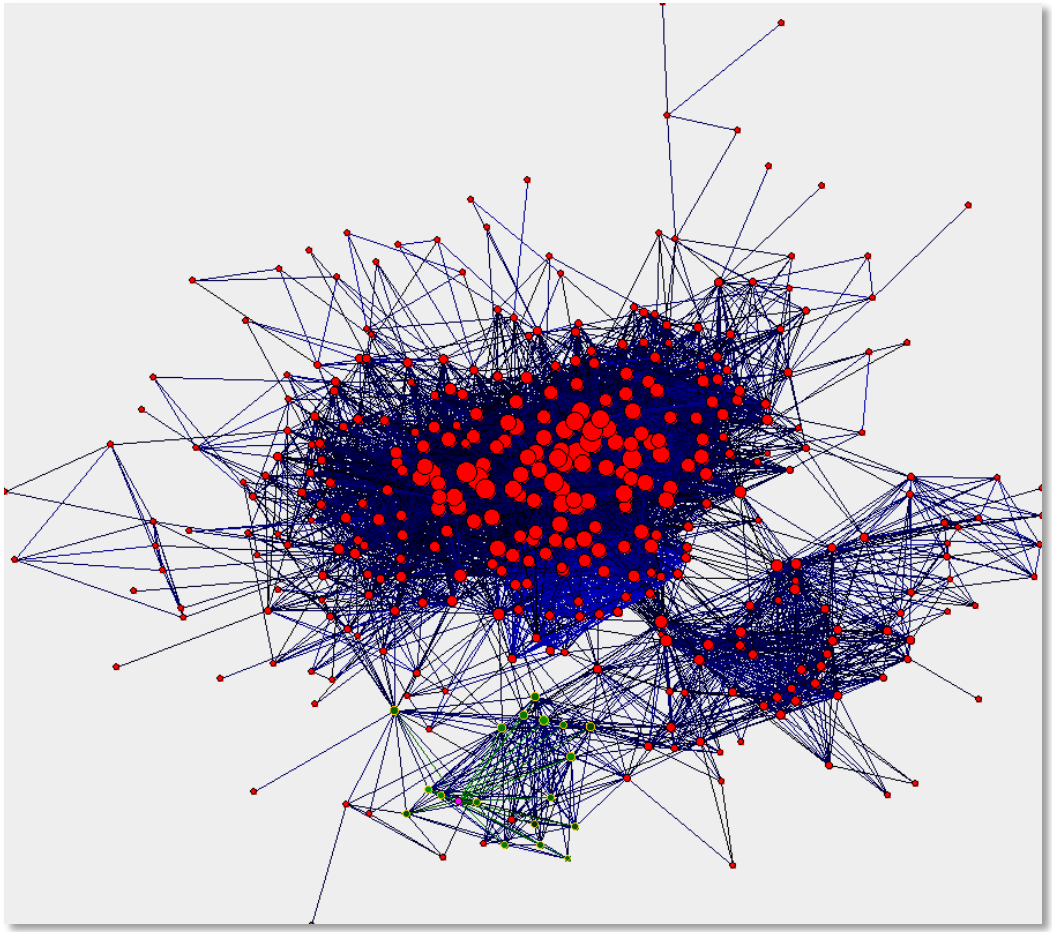
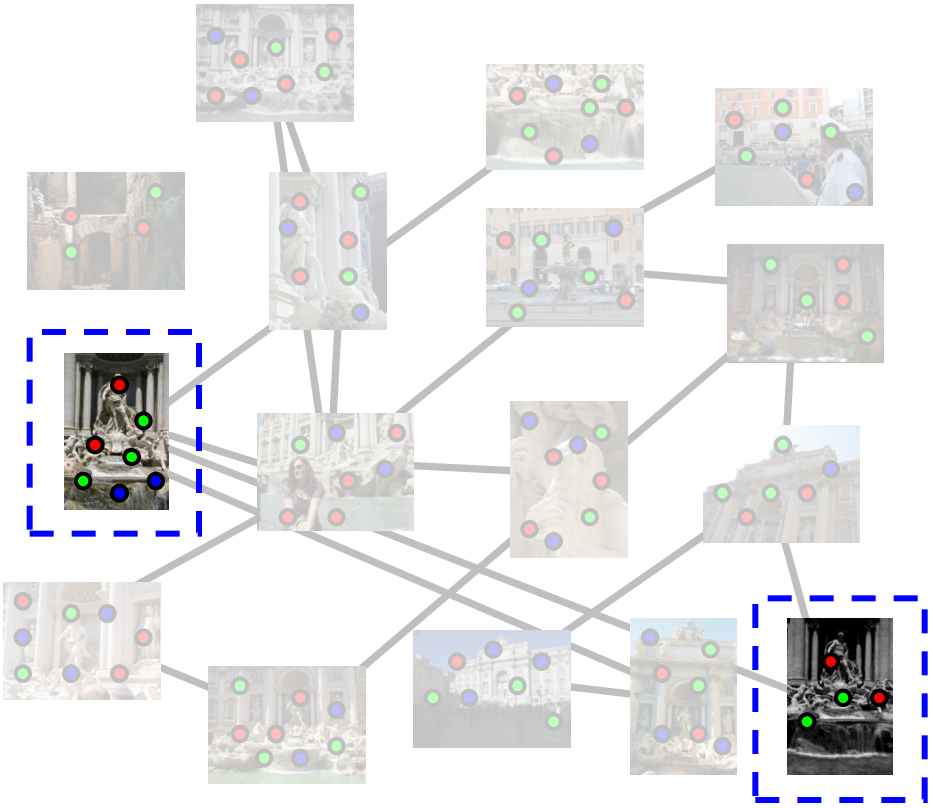


Image connectivity graph



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

Incremental structure from motion



Incremental structure from motion

Incremental structure from motion

Problem size

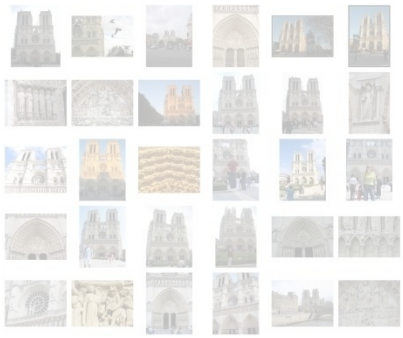
- **Trevi Fountain collection**

  466 input photos

  + > 100,000 3D points

  = very large optimization problem

Photo Tourism overview



Input photographs

Scene reconstruction

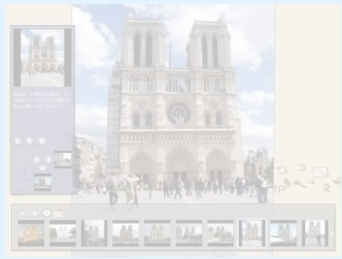


Photo Explorer

Photo Explorer

Can we reconstruct entire cities?

flickr

Signed in as Jimantha | Help | Sign Out

Home | You | Organize | Contacts | Groups | Explore

Search everyone's phot | Search

Search

Photos | Groups | People

Can't see your photos? Find out why...

Everyone's Photos | rome or roma | SEARCH

Advanced Search | Search by Camera

Full text | Tags only

We found **2,379,801 results** matching rome or roma.

View slideshow

View: Most relevant • Most recent • Most interesting

Show: Details • Thumbnails

From Giampaolo...

From Kipourax

From MrMass

From rocdam

From alessandro...

From * Toshio *

From Optical...

From egold

From egold

From egold

From donato.chiru...

From cuellar

From egold

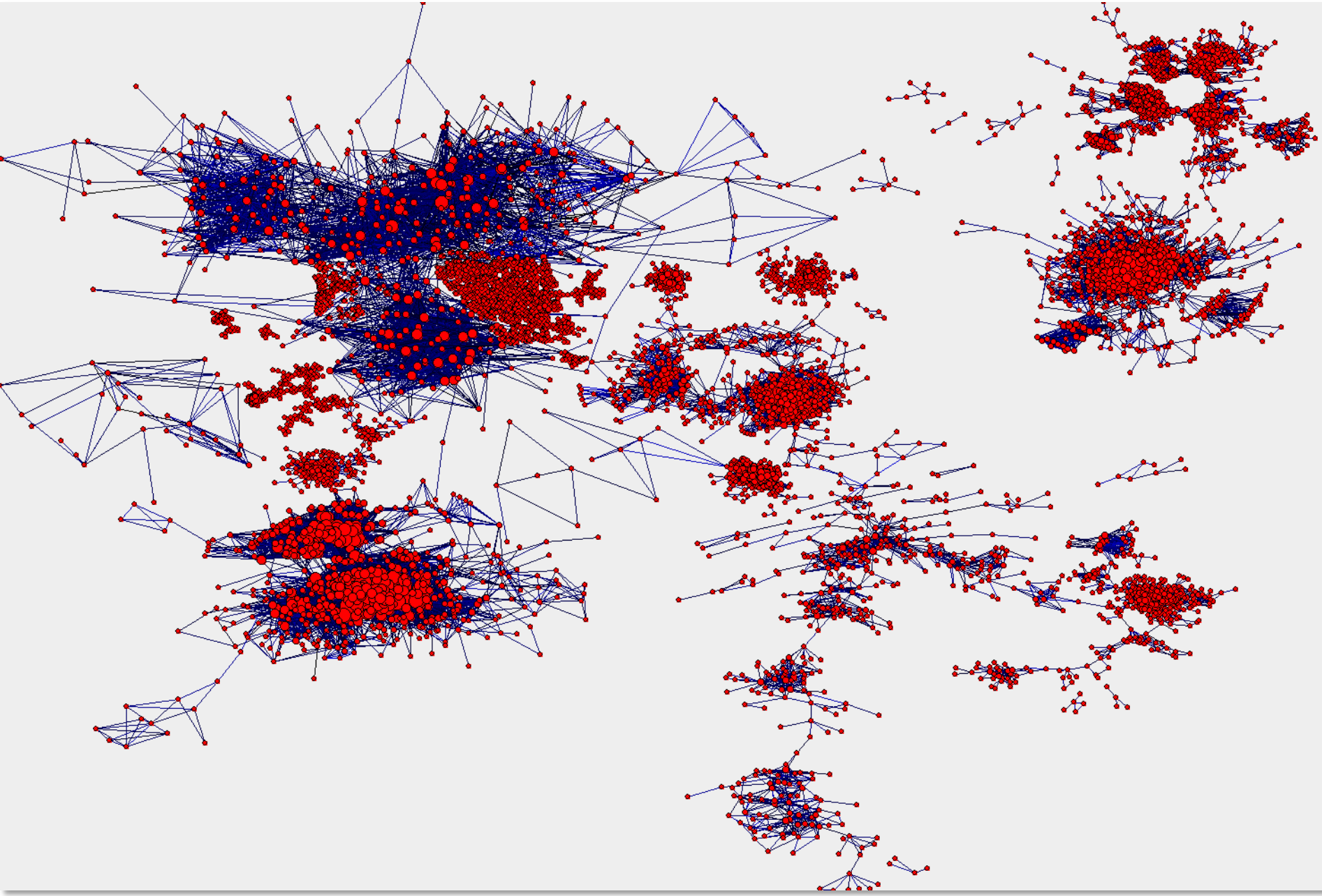
From Aquilant

From Peter...

From SDBryan

From cuellar

From david.bank



Gigantic matching problem

- **1,000,000 images → 500,000,000,000 pairs**
  - **Matching all of these on a 1,000-node cluster would take more than a year, even if we match 10,000 every second**
  - **And involves TBs of data**

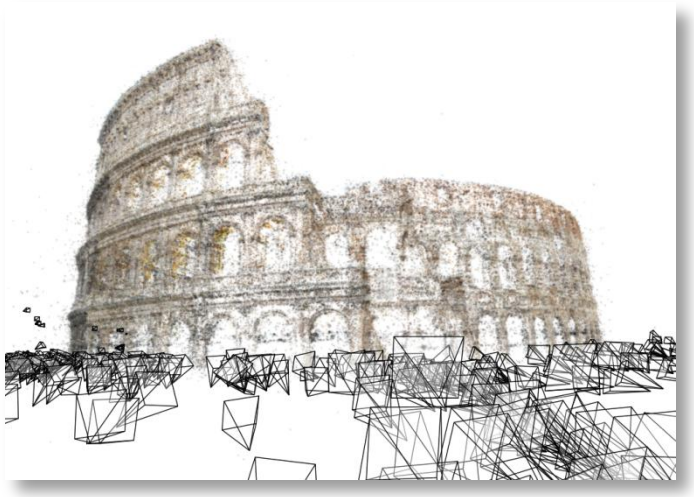
- **The vast majority (>99%) of image pairs do *not* match**
- **There are better ways of finding matching images**

Gigantic SfM problem

- **Largest problem size we've seen:**
  - **15,000 cameras**
  - **4 million 3D points**
  - **more than 12 million parameters**
  - **more than 25 million equations**
- **Huge optimization problem**
- **Requires sparse least squares techniques**

Building Rome in a Day

St. Peter's Basilica



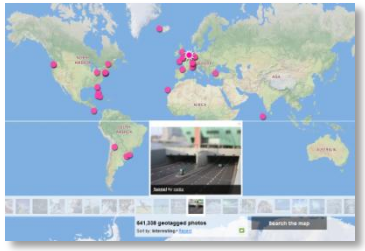
Colosseum

Trevi Fountain

Rome, Italy. Reconstructed 150,000 in 21 hours on 496 machines

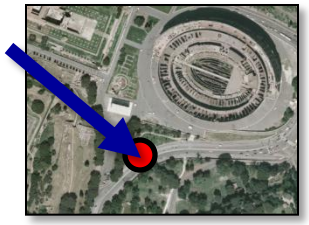
Applications

Community photo collections



- "Wikipedia for photos" – visual record of world through community of photographers
  - *Geograph British Isles* http://www.geograph.org.uk/

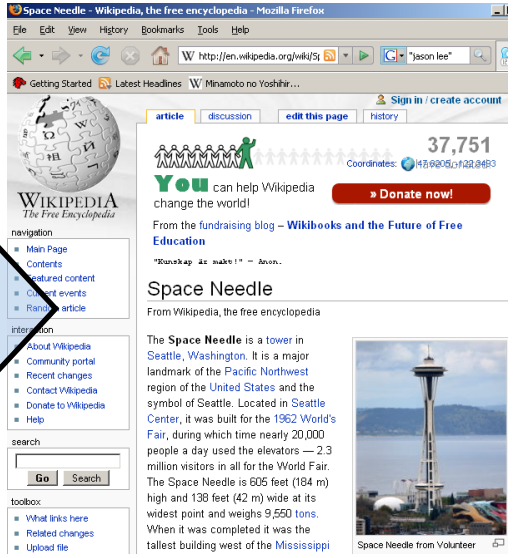
- Users can tag and comment on photos, link to other content
  - *World-wide telescope*



- "Where should I take a photo?" http://photocitygame.com/

Community photo collections

- Leveraging large databases of photos, large number of users
  - Annotations / augmented reality

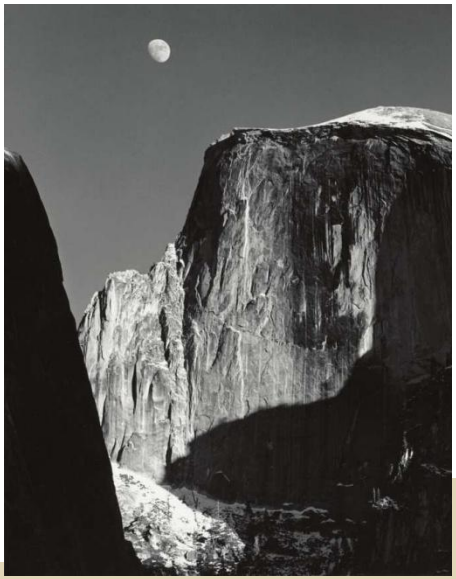


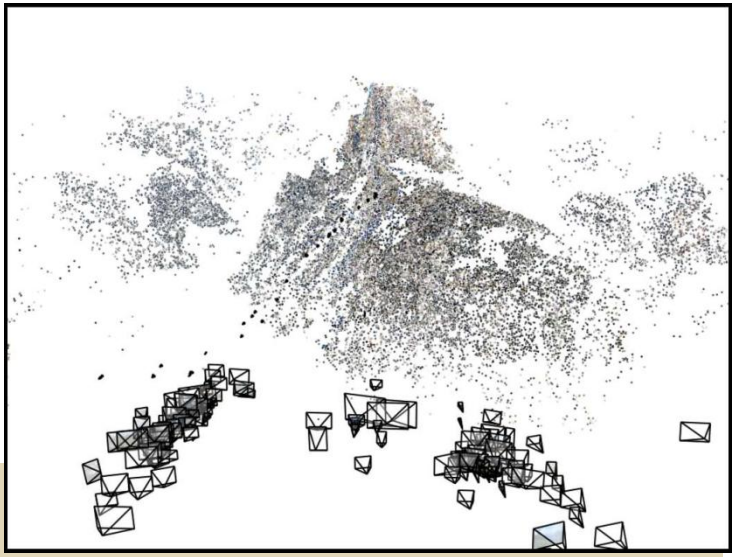
Virtual tour guide scenario

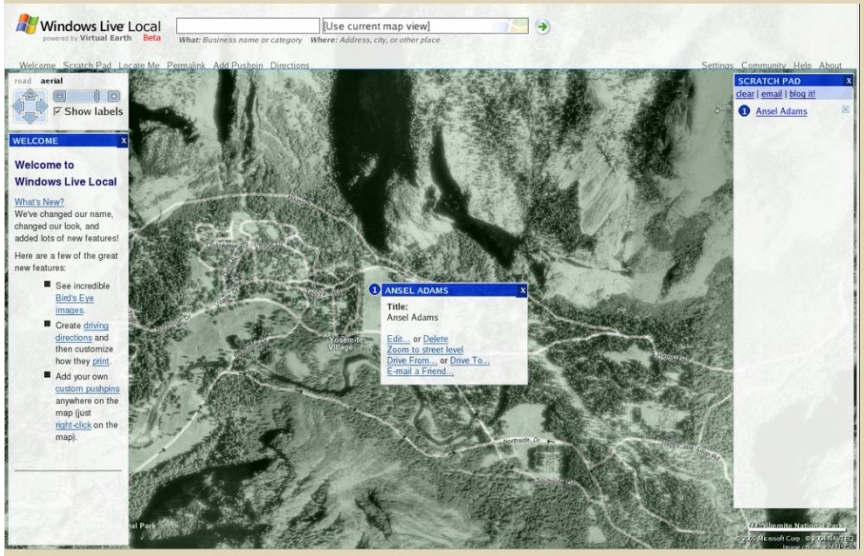
St. Peter's Basilica

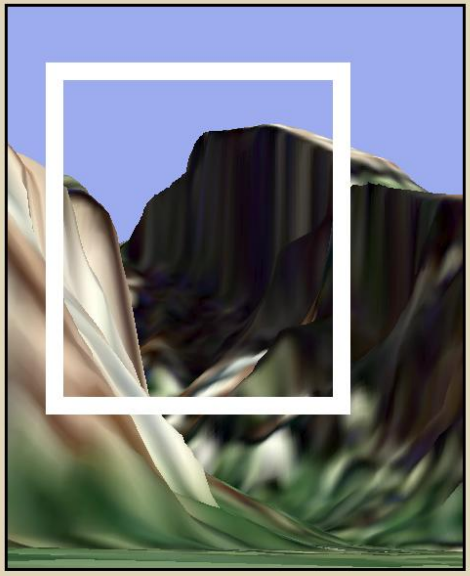
- Built: 1506-1626
- http://en.wikipedia.org/wiki/St._Peter's_Basilica

Rephotography









Topographic data courtesy USGS

Internet Computer Vision

Computer Vision and the Internet (09w5126)

Arriving Sunday, August 30 and departing Friday September 4, 2009

CO

Sec

CVF

Progr

IV Hor

Works

Areas

Progra

Proceedings of the IEEE

AUGUST 2010 / VOL. 98 / NO. 8

CONTENTS

SPECIAL ISSUE

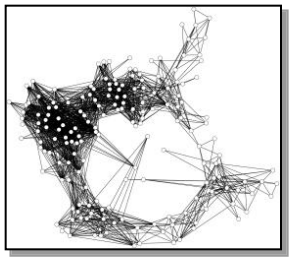
INTERNET VISION

Edited by S. Avidan, S. Baker, and Y. Shan

DEPARTMENTS

Summary

- Large (Internet) photo collections: the next frontier in computer vision



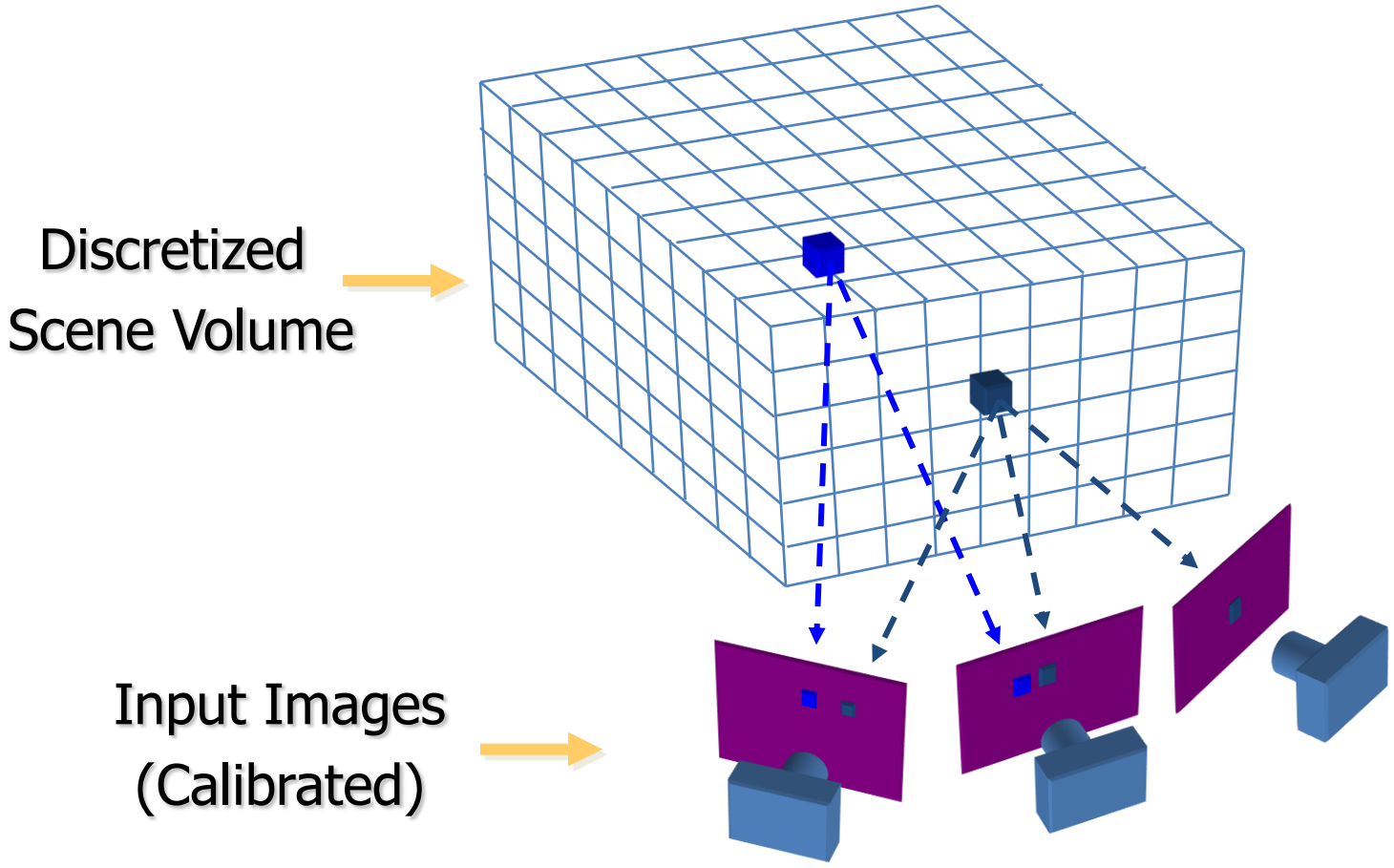
- Efficient algorithms for large-scale 3D matching and reconstruction

- New challenges for 3D recognition

Overview

- Structure from Motion (SfM)
- Large scale Structure from Motion
- Other approaches to obtaining 3D structure

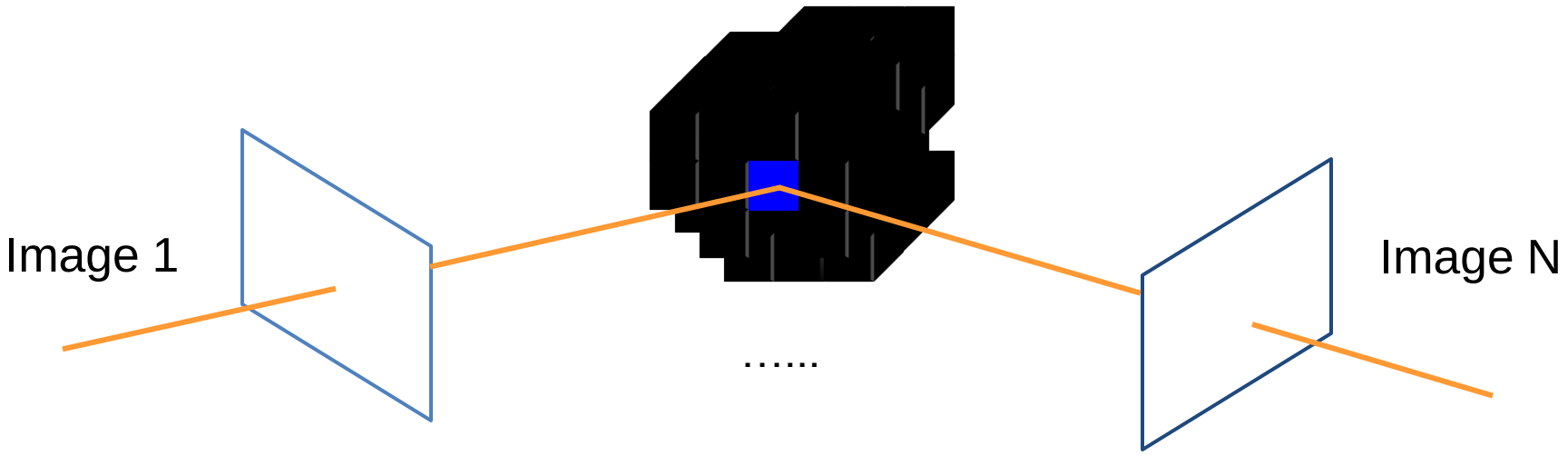
Volumetric Stereo / Voxel Coloring



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

Slide credit: S. Lazebnik

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

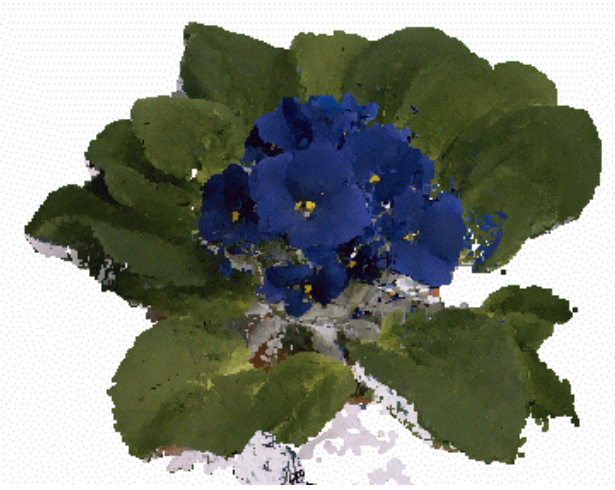
K. N. Kutulakos and S. M. Seitz, **A Theory of Shape by Space Carving**, *ICCV* 1999

Slide credit: S. Lazebnik

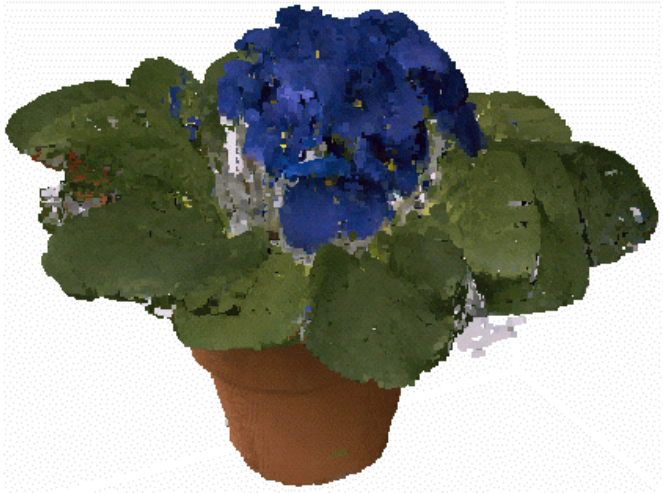
Space Carving Results: African Violet

Input Image (1 of 45)

Reconstruction



Reconstruction



Reconstruction

Source: S. Seitz

Space Carving Results: Hand

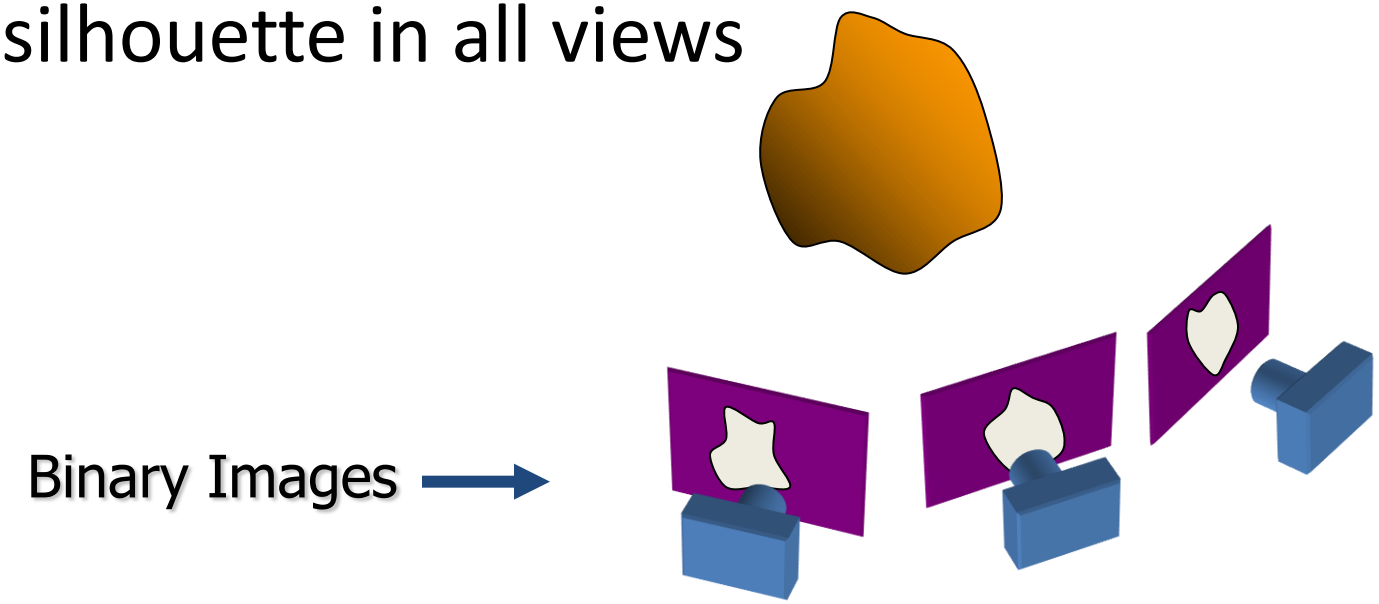
Input Image
(1 of 100)

Views of Reconstruction

Slide credit: S. Lazebnik

Reconstruction from Silhouettes

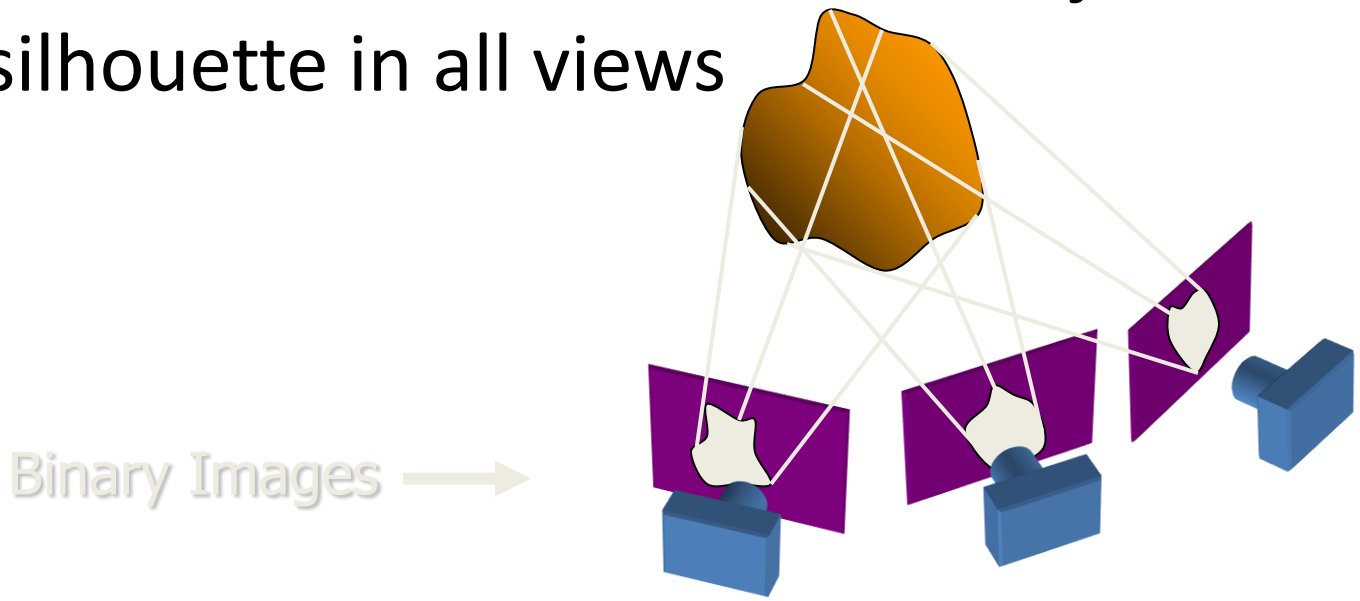
- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views



Slide credit: S. Lazebnik

Reconstruction from Silhouettes

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

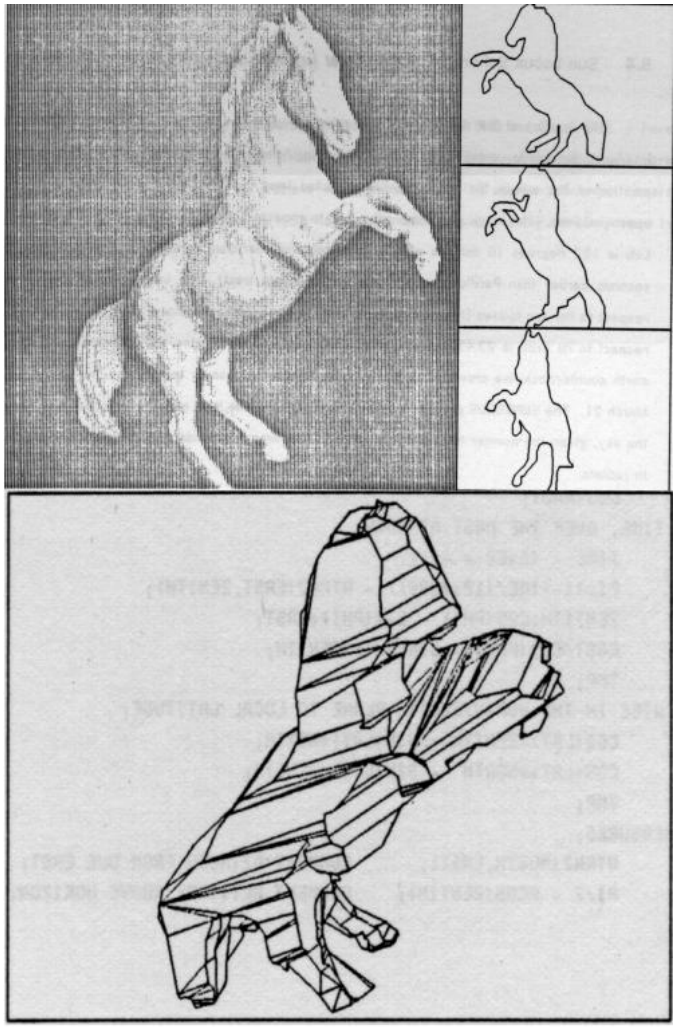


Finding the silhouette-consistent shape (*visual hull*):

- *Backproject* each silhouette
- Intersect backprojected volumes

Slide credit: S. Lazebnik

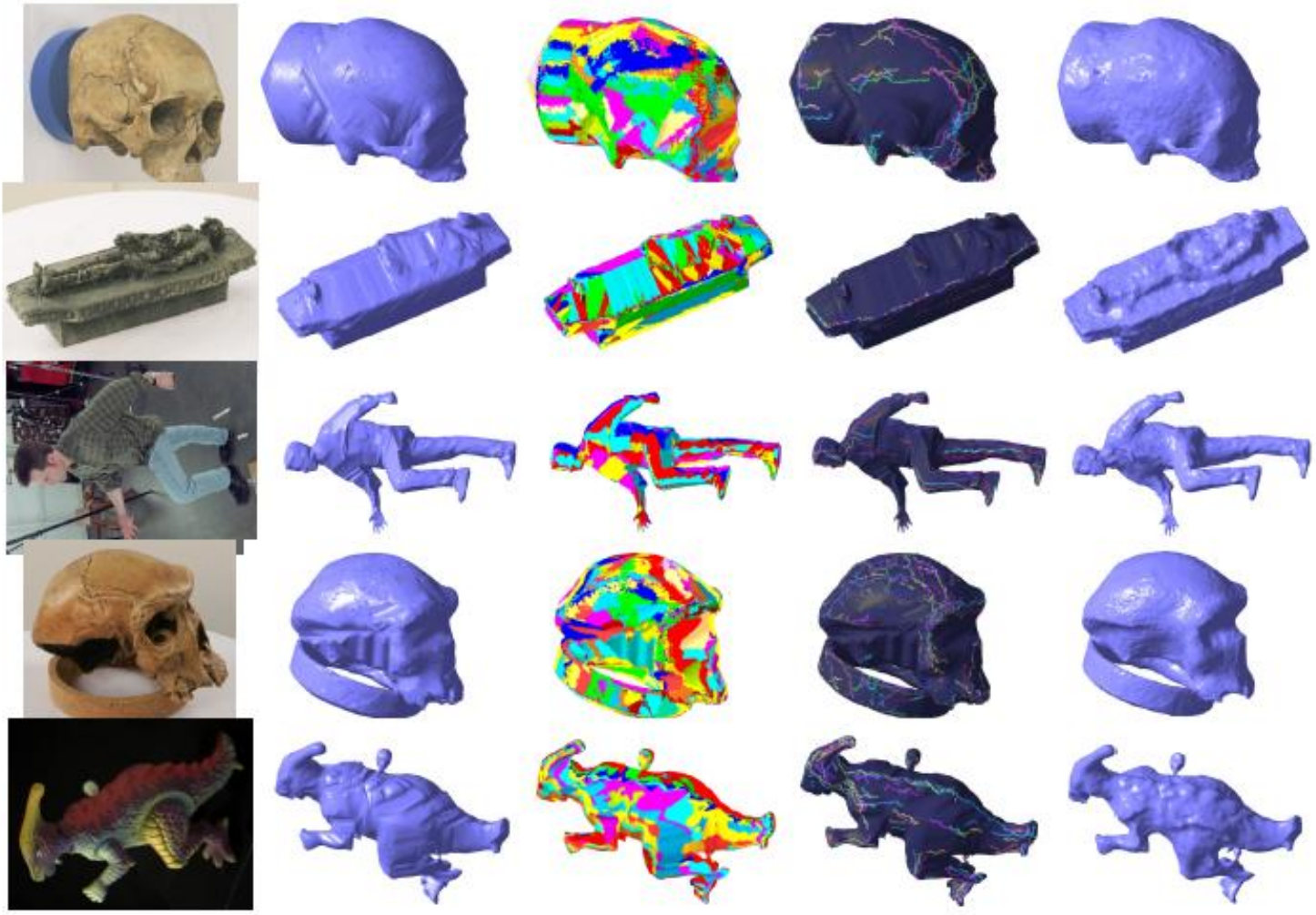
Volume intersection



B. Baumgart, *Geometric Modeling for Computer Vision*, Stanford Artificial Intelligence Laboratory, Memo no. AIM-249, Stanford University, October 1974.

Slide credit: S. Lazebnik

Carved visual hulls

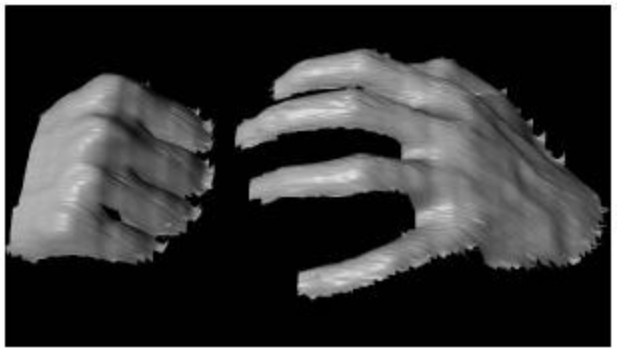


Yasutaka Furukawa and Jean Ponce, **Carved Visual Hulls for Image-Based Modeling**, ECCV 2006.

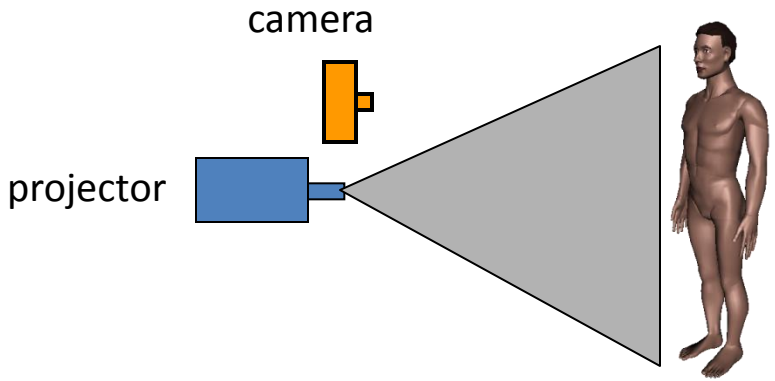
Slide credit: S. Lazebnik

Active stereo with structured light



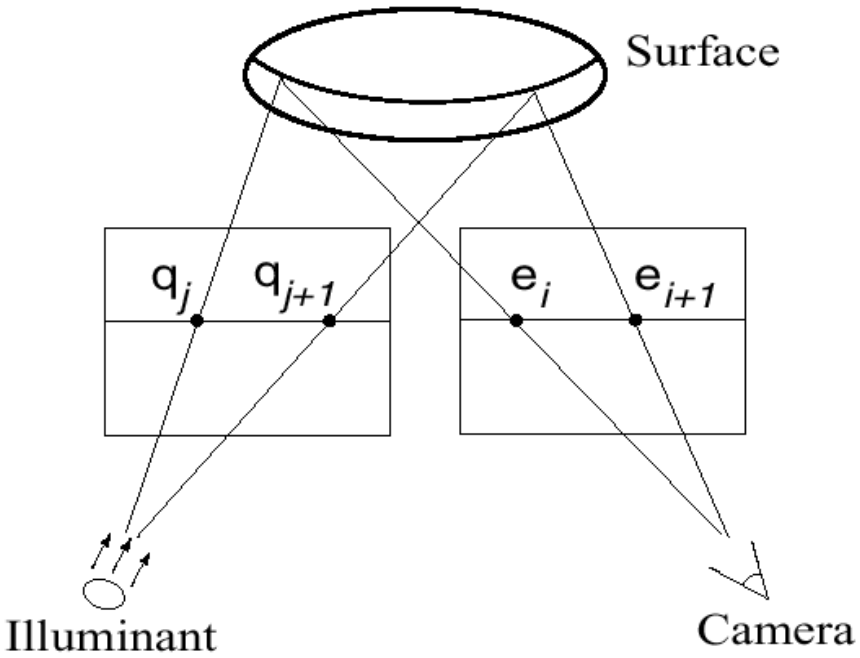


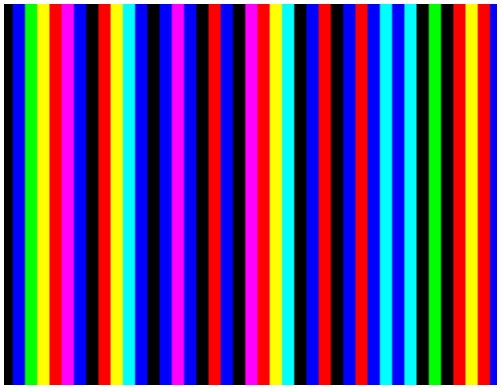
- Project “structured” light patterns onto the object
  - simplifies the correspondence problem
  - Allows us to use only one camera

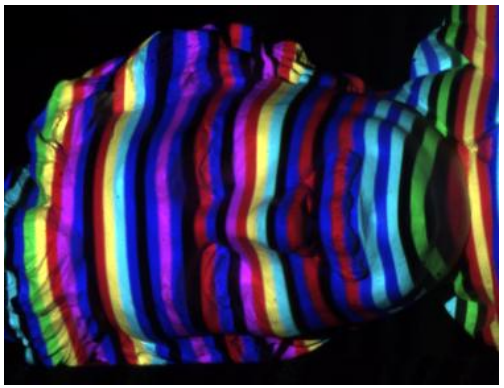


L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Active stereo with structured light

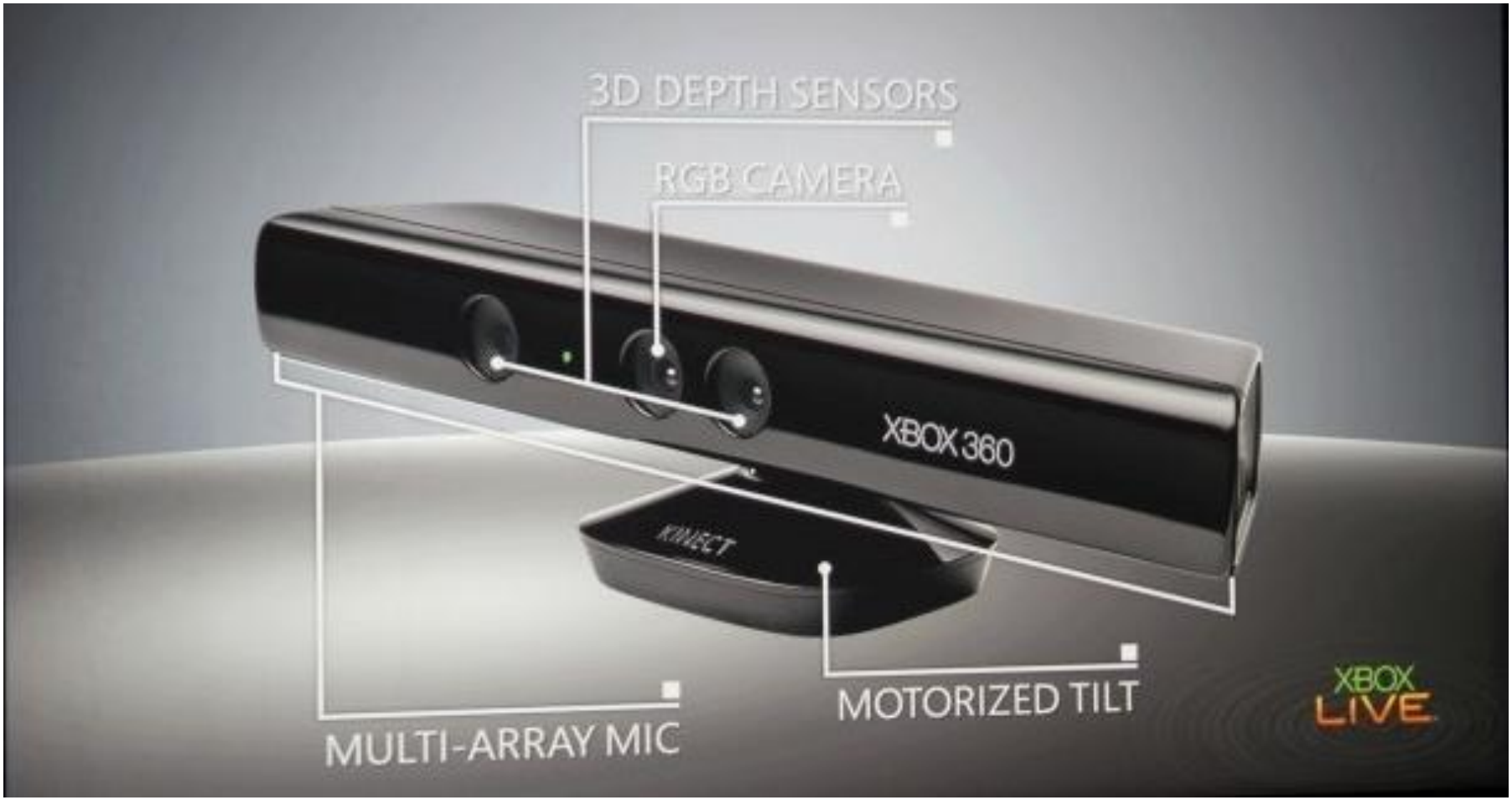






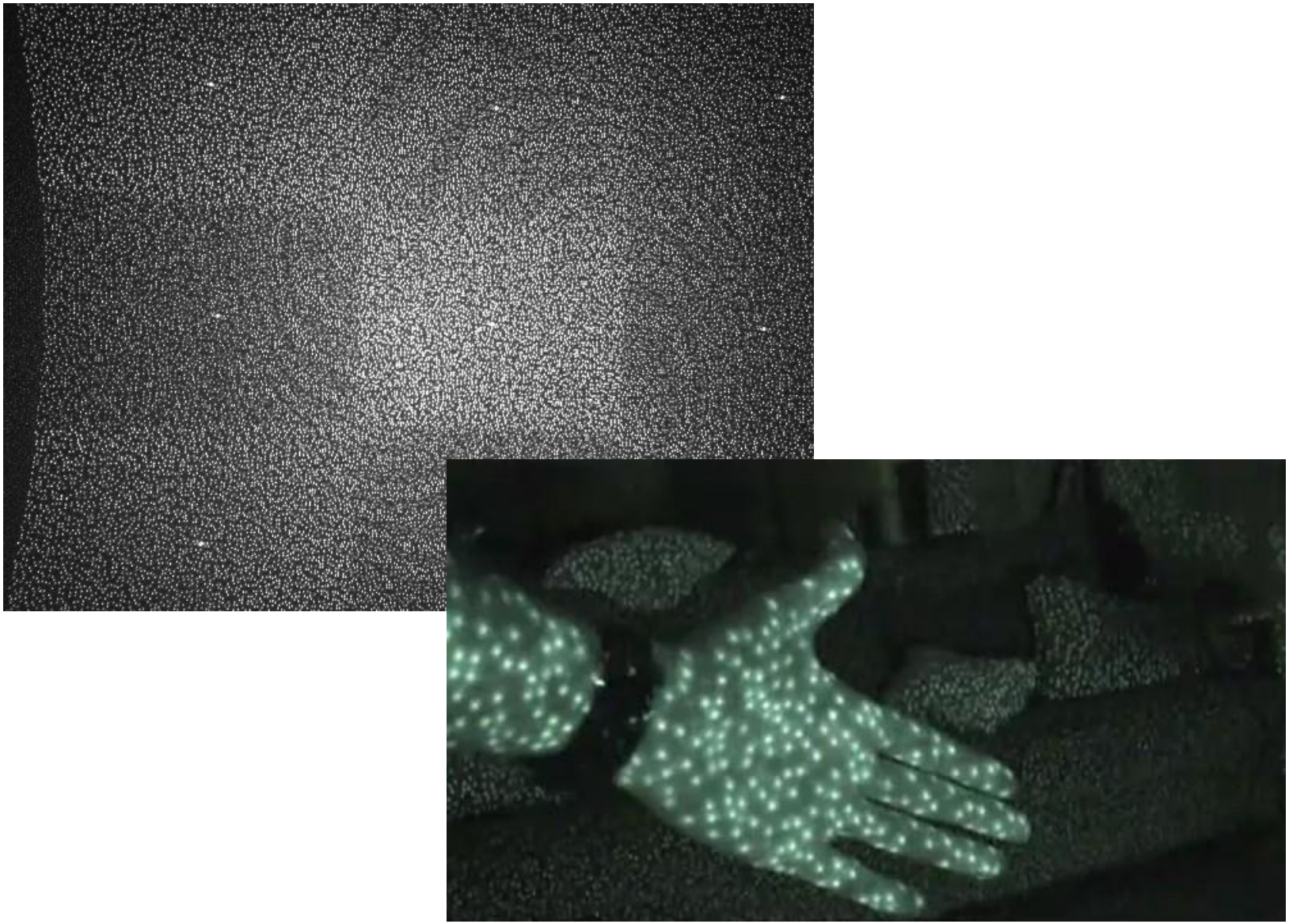
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Microsoft Kinect



Microsoft Kinect

Microsoft Kinect

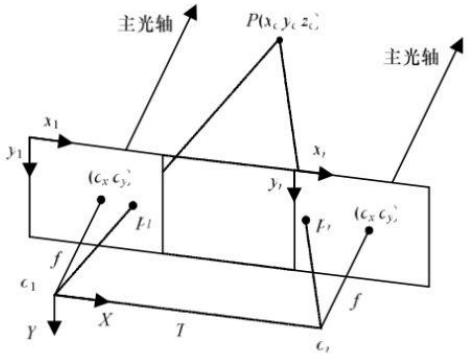


Microsoft Kinect

Microsoft Kinect

Leap Motion

Controlling Computers With Hand Motions

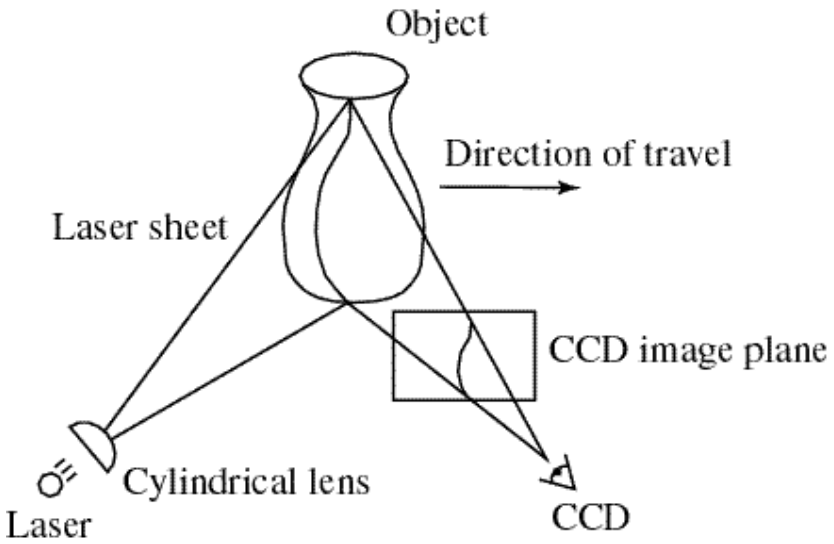


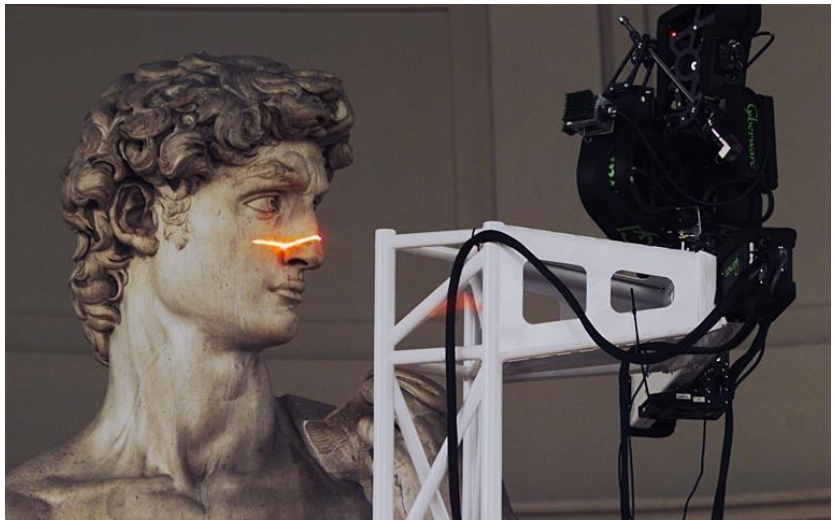
Left image

Right image

A host of companies are looking to shake-up the ways we interact with computers. Using new motion sensing technology they aim for users to replace typing and mouse-clicking with some of the gestures and movements used in everyday life.

Laser scanning





Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Source: S. Seitz

Laser scanned models

The Digital Michelangelo Project, Levoy et al.

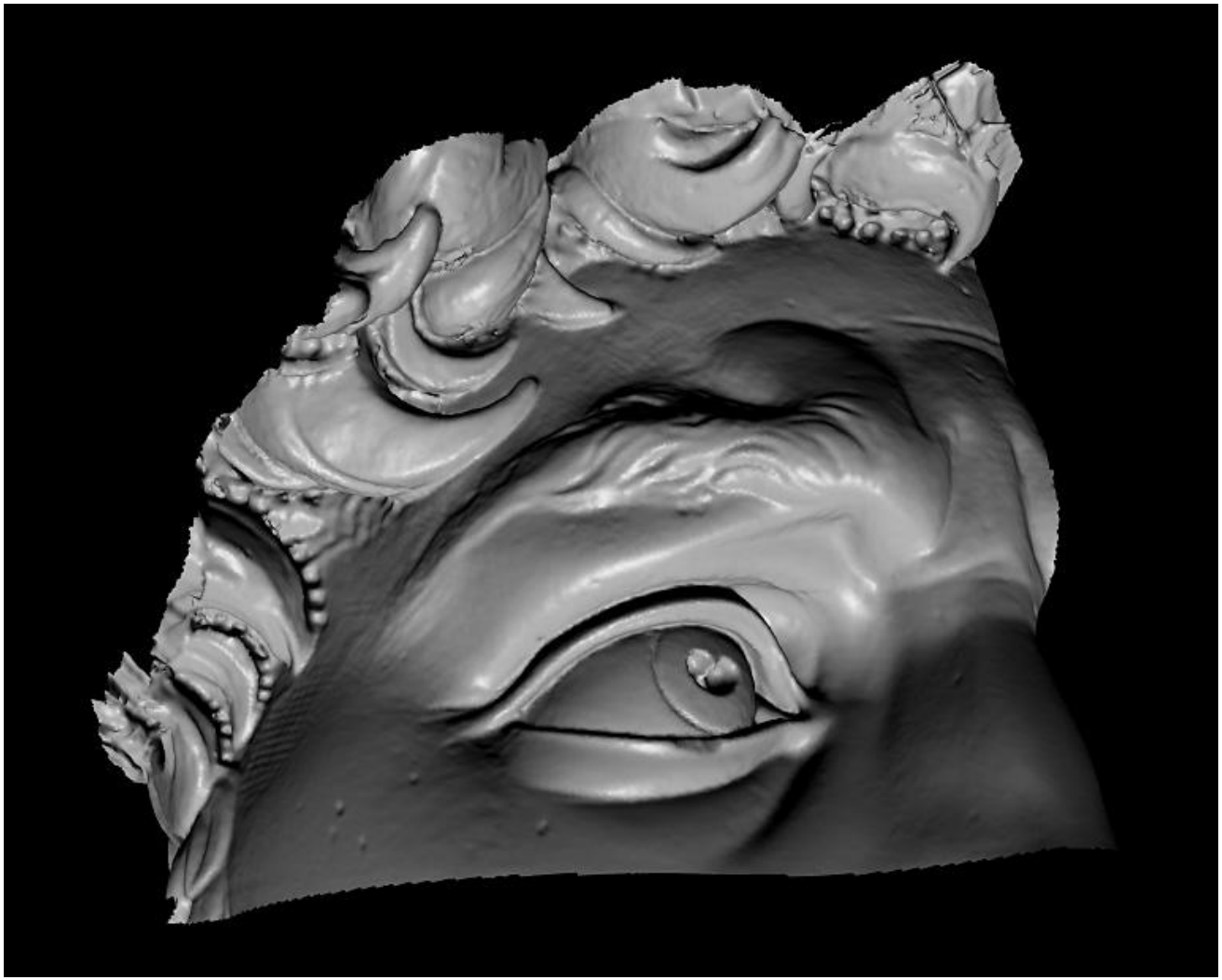
Source: S. Seitz

Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

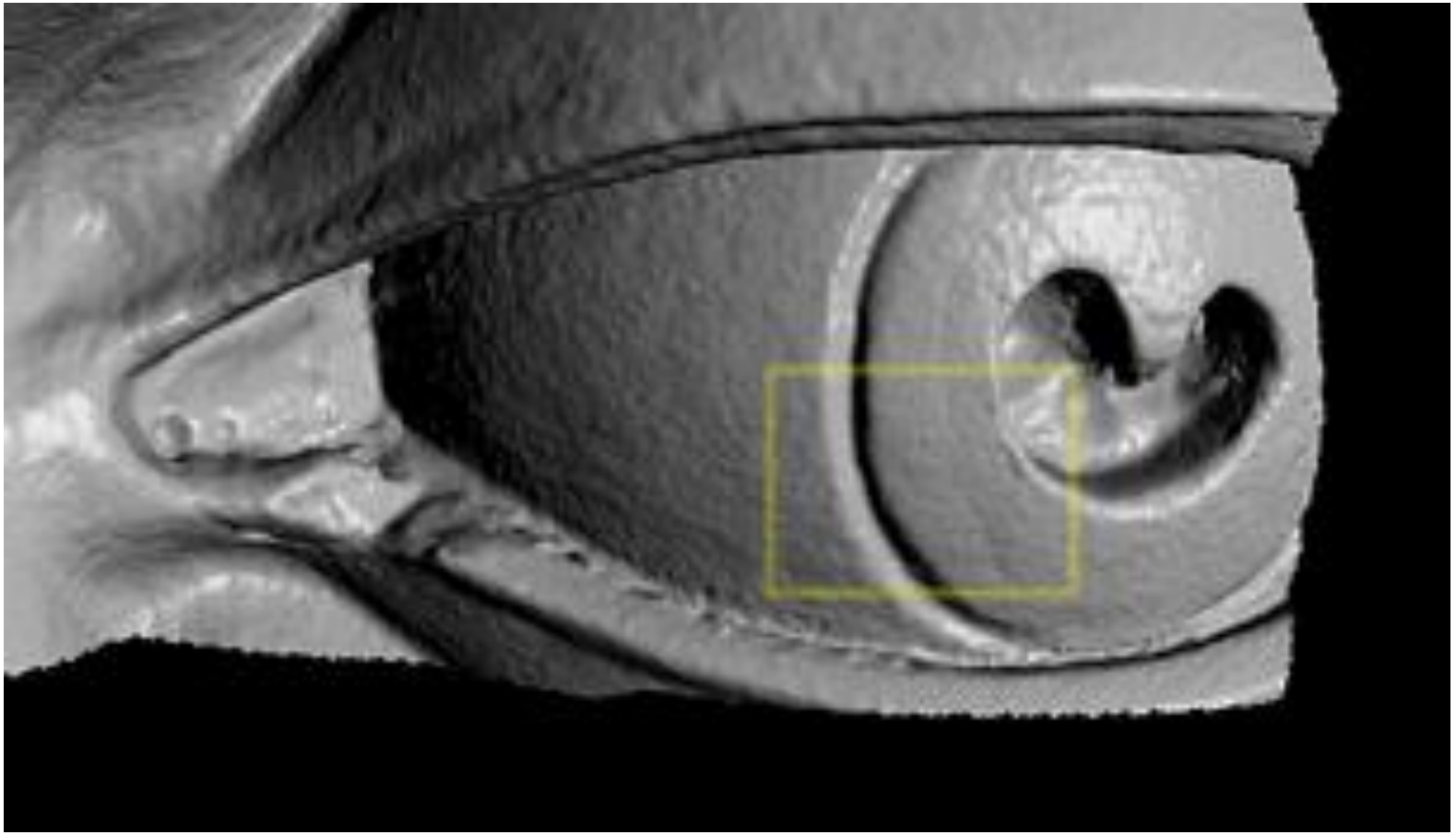
Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

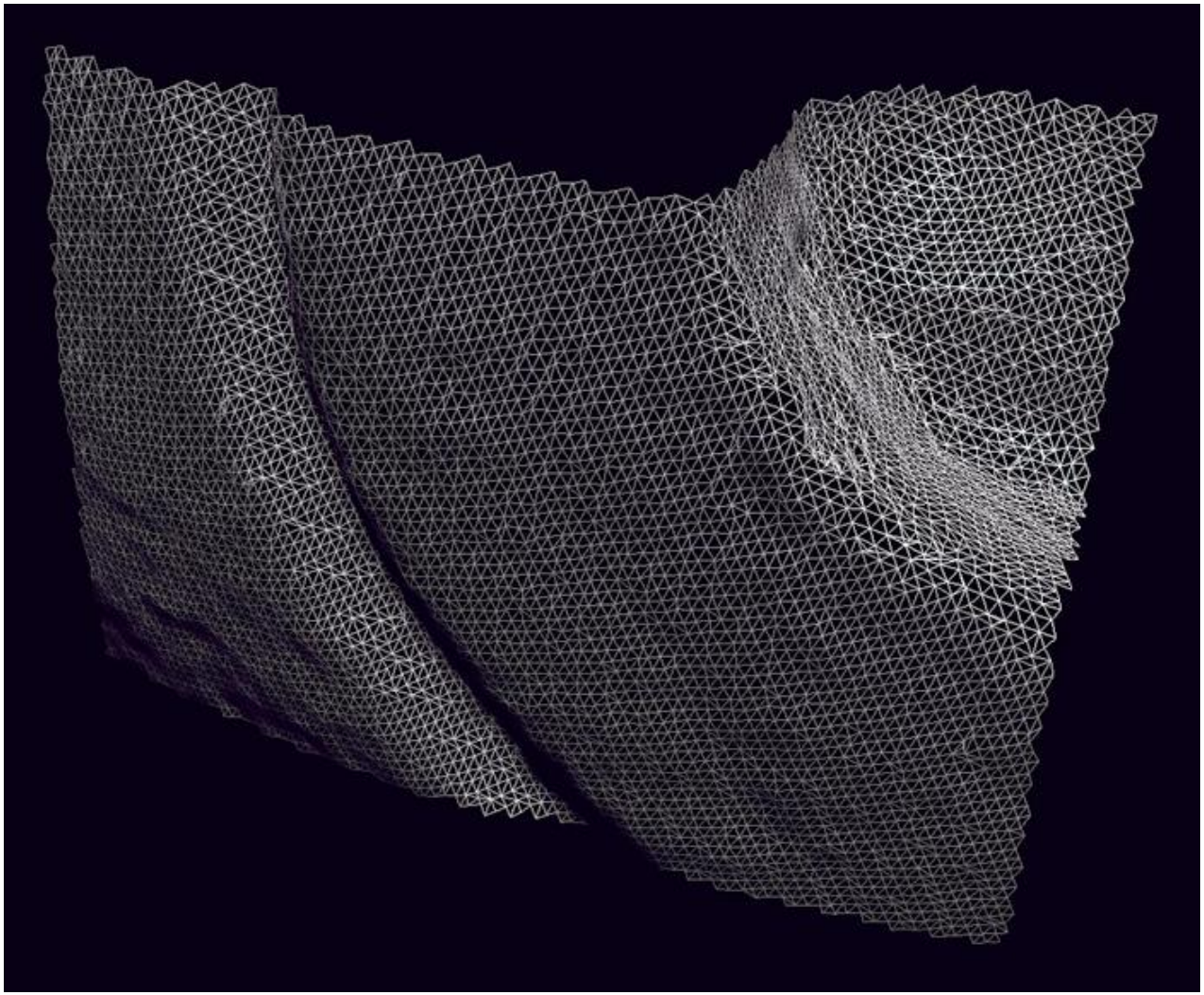
Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Aligning range images

- A single range scan is not sufficient to describe
- 

B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH 1996

Aligning range images

- A single range scan is not sufficient to describe a complex surface
- Need techniques to register multiple range images

    - … which brings us to *multi-view stereo*

Readings

- **Chapter 8**