Computer Vision – Lecture 17

Uncalibrated Reconstruction & SfM

08.07.2019

Bastian Leibe

Visual Computing Institute
RWTH Aachen University
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

Announcements

- No lecture tomorrow (Tuesday)
  - Due to a schedule conflict
- Last exercise will be offered on Monday, 15.07.
  - Optional, but recommended
  - Time slot & room to be announced…
- Repetition slides
  - I will provide a slide set (pdf) with summary slides for the entire lecture
  - Idea: you can use this as an index to the lecture

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Deep Learning
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Triangulation
  - Uncalibrated Reconstruction & Active Stereo
  - Structure-from-Motion

Recap: A General Point

- Equations of the form

$$\mathrm{Ax} = 0$$

- How do we solve them? (always!)
  - Apply SVD

SVD

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T = \boldsymbol{U} \begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & d_{NN} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{1N} \\ \vdots & \ddots & \vdots \\ v_{N1} & \cdots & v_{NN} \end{bmatrix}^T$$

Singular values Singular vectors

  - Singular values of A = square roots of the eigenvalues of A^TA.
  - The solution of Ax=0 is the *nullspace* vector of A.
  - This corresponds to the *smallest singular vector* of A.

Recap: Camera Parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & s & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation $\mathbf{R}$
  - Translation $\mathbf{t}$
    (both relative to world coordinate system)

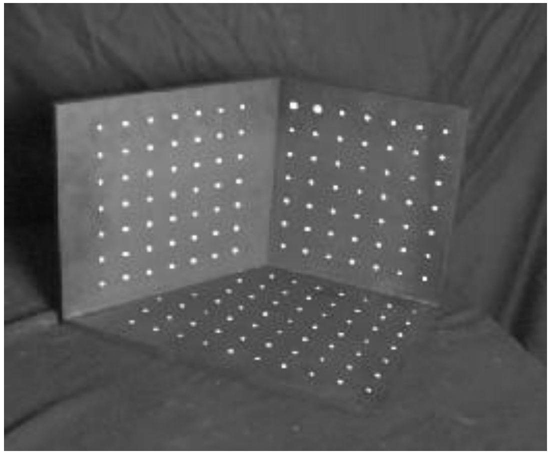
- Camera projection matrix

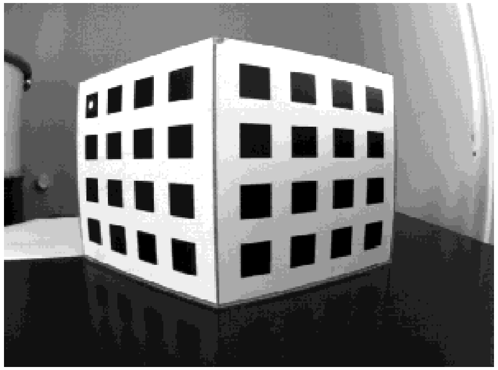
$$\mathrm{P} = \mathrm{K}\left[\mathrm{R} \mid \mathrm{t}\right]$$

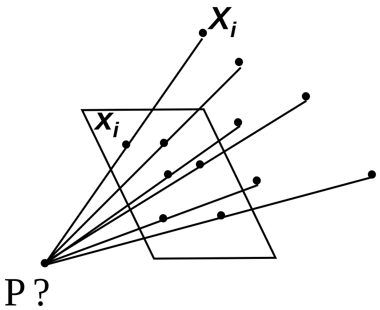
$\Rightarrow$ General pinhole camera: 9 DoF
$\Rightarrow$ CCD Camera with square pixels: 10 DoF
$\Rightarrow$ General camera: 11 DoF

Recap: Calibrating a Camera

- Goal
  - Compute intrinsic and extrinsic parameters using observed camera data.
- Main idea
  - Place "calibration object" with known geometry in the scene
  - Get correspondences
  - Solve for mapping from scene to image: estimate $P=P_{int}P_{ext}$

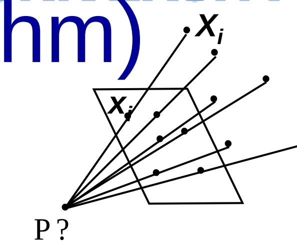






Computer Vision Summer'19

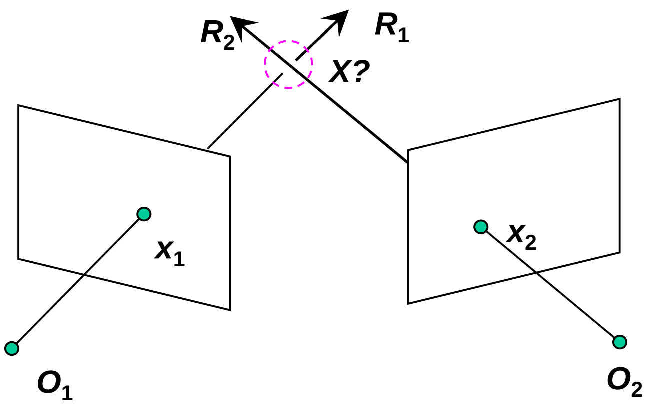
Recap: Camera Calibration (DLT Algorithm)



$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \qquad \mathrm{Ap} = 0$$

- P has 11 degrees of freedom.
- Two linearly independent equations per independent 2D/3D correspondence.
- (similar to homography estimation)
  - Solution corresponds to smallest singular vector.
- 5 ½ correspondences needed for a minimal solution.

Recap: Triangulation – Linear Algebraic Approach



$$\lambda_1 x_1 = P_1 X \qquad x_1 \times P_1 X = 0 \qquad [x_{1\times}] P_1 X = 0$$

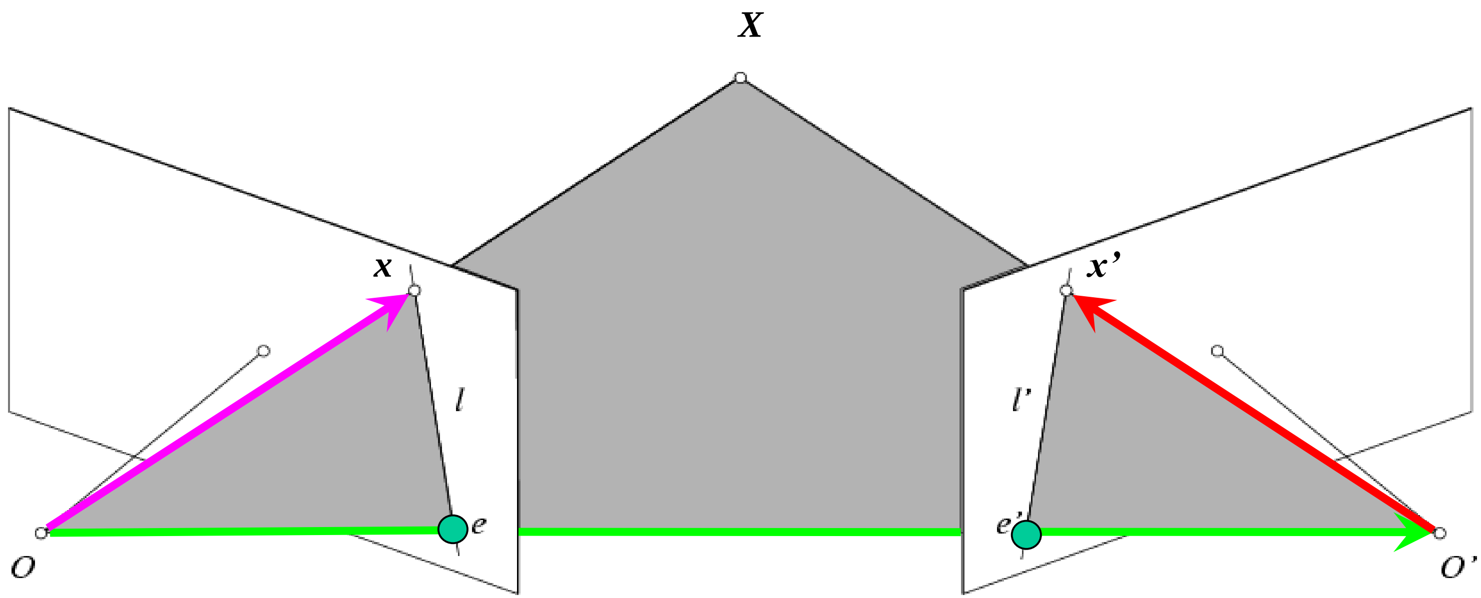
$$\lambda_2 x_2 = P_2 X \qquad x_2 \times P_2 X = 0 \qquad [x_{2\times}] P_2 X = 0$$

- Two independent equations each in terms of three unknown entries of X.
- Stack equations and solve with SVD.
- This approach nicely generalizes to multiple cameras.

Topics of This Lecture

- Revisiting Epipolar Geometry
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Kinect sensor
  - Structured Light sensing
  - Laser scanning
- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
  - Projective factorization
  - Bundle adjustment

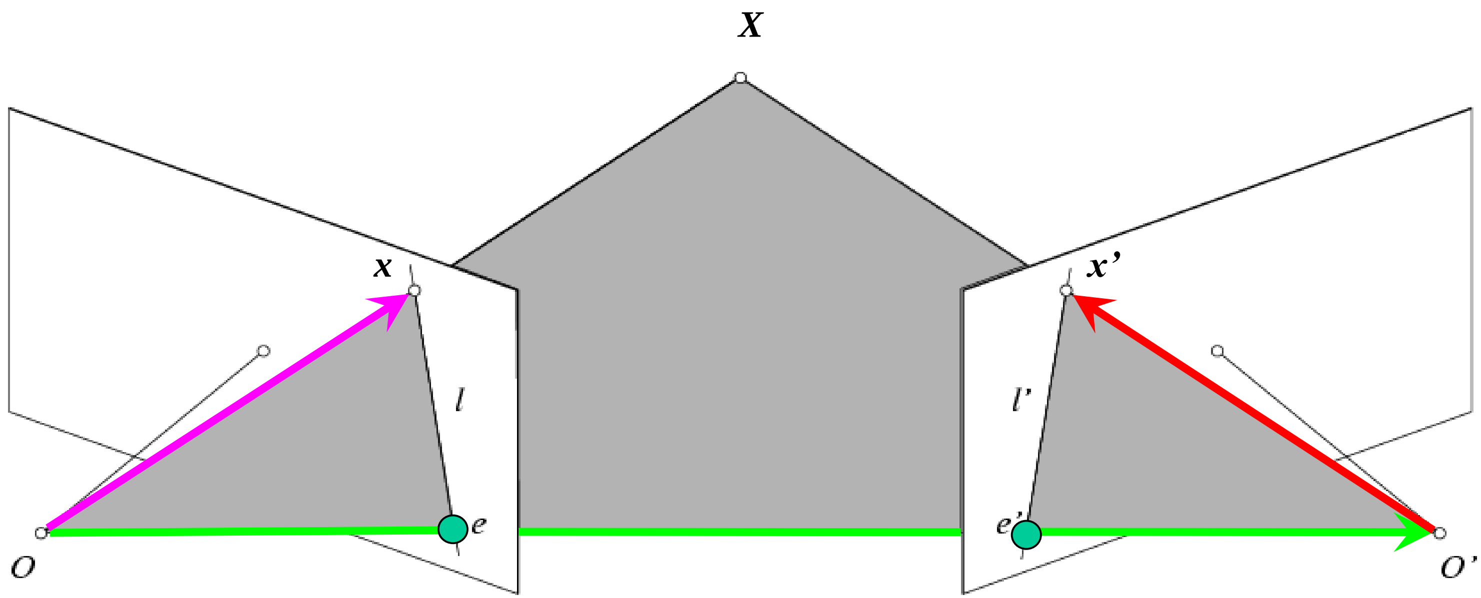
Recap: Epipolar Geometry – Calibrated Case



$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

Essential Matrix
(Longuet-Higgins, 1981)

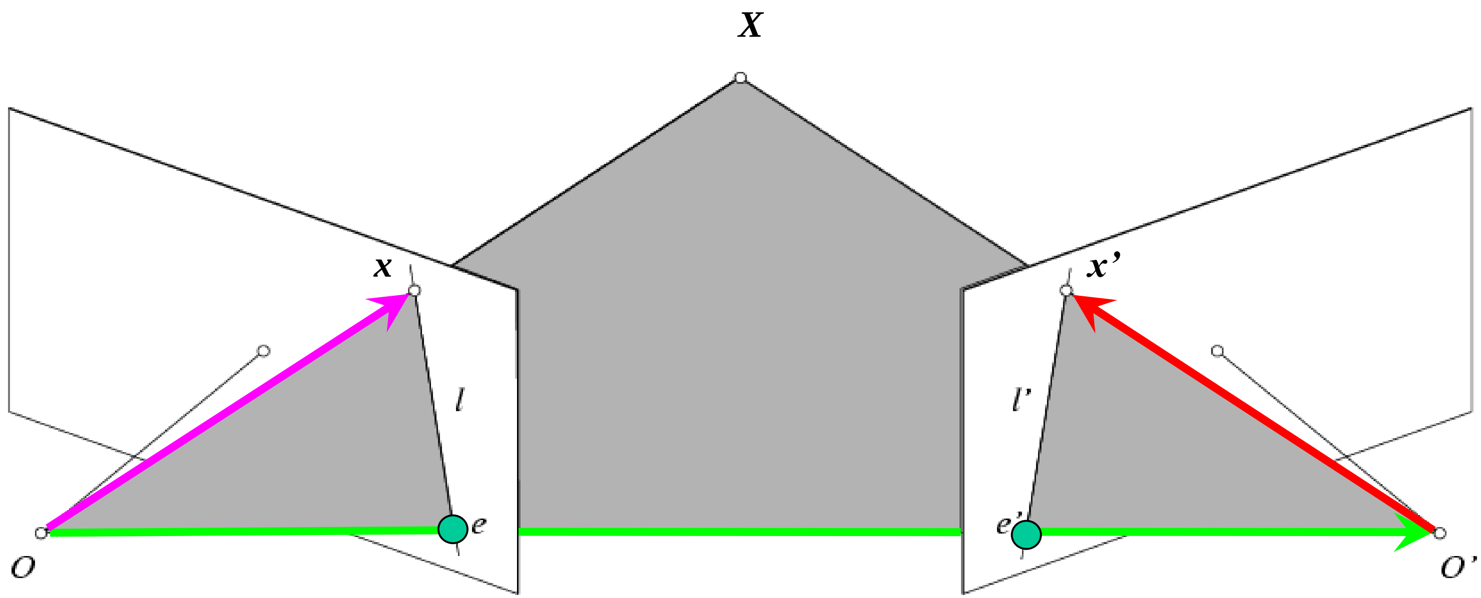
Epipolar Geometry: Calibrated Case



$$x \cdot [t \times (R x')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times] R$$

- *E x'* is the epipolar line associated with *x'* (*l* = *E x'*)
- $E^T x$ is the epipolar line associated with *x* ($l' = E^T x$)
- *E e'* = 0 and $E^T e = 0$ *Why?*
- *E* is singular (rank two) *Why?*
- *E* has five degrees of freedom (up to scale)

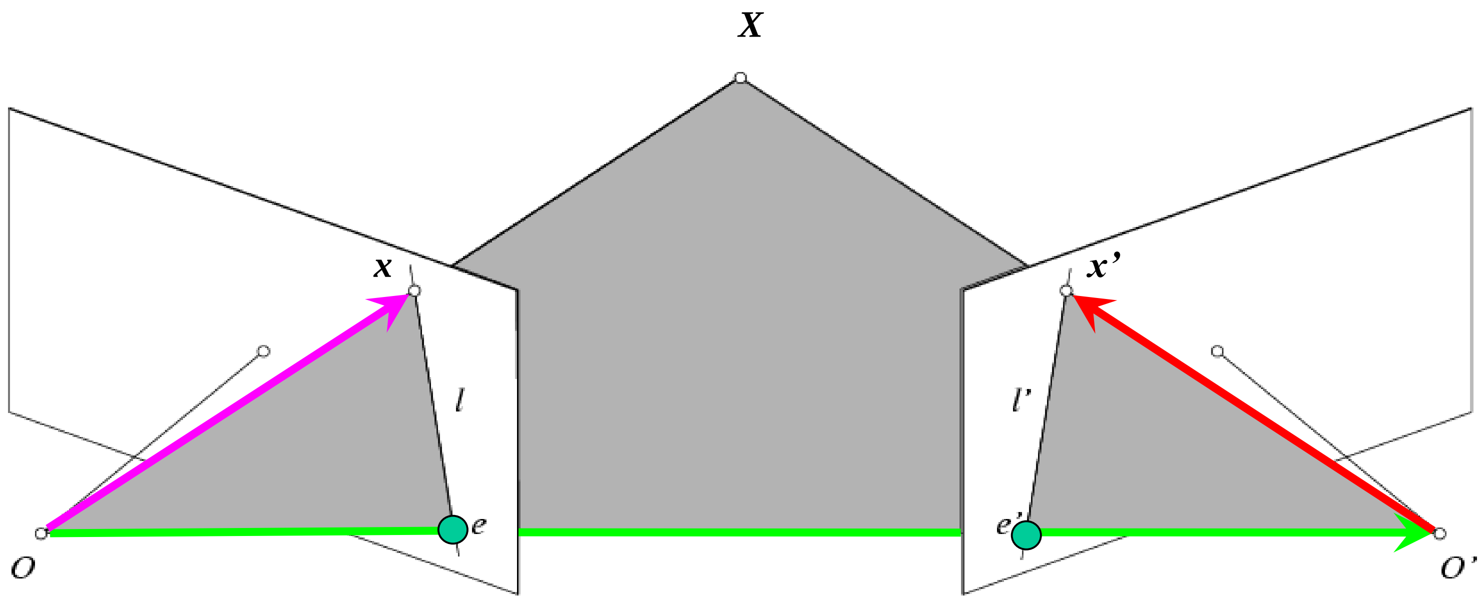
Epipolar Geometry: Uncalibrated Case



- The calibration matrices *K* and *K'* of the two cameras are unknown
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Epipolar Geometry: Uncalibrated Case



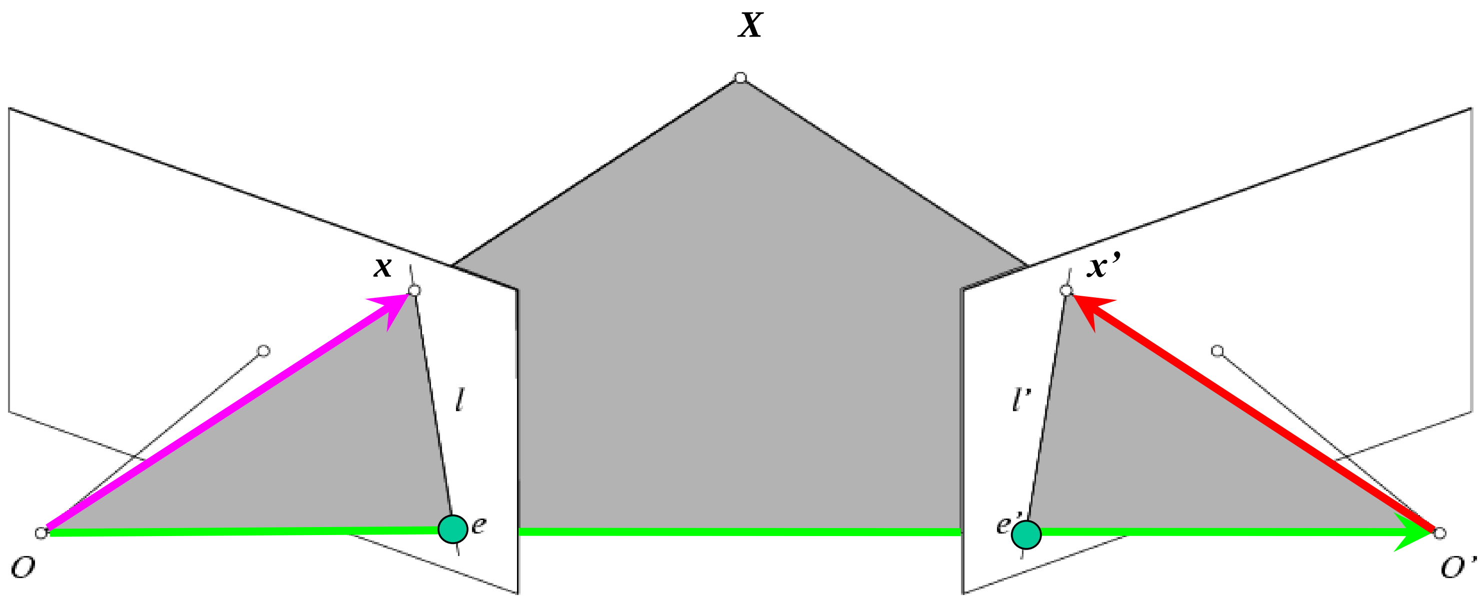
$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

$$x = K \hat{x}$$

$$x' = K' \hat{x}'$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar Geometry: Uncalibrated Case



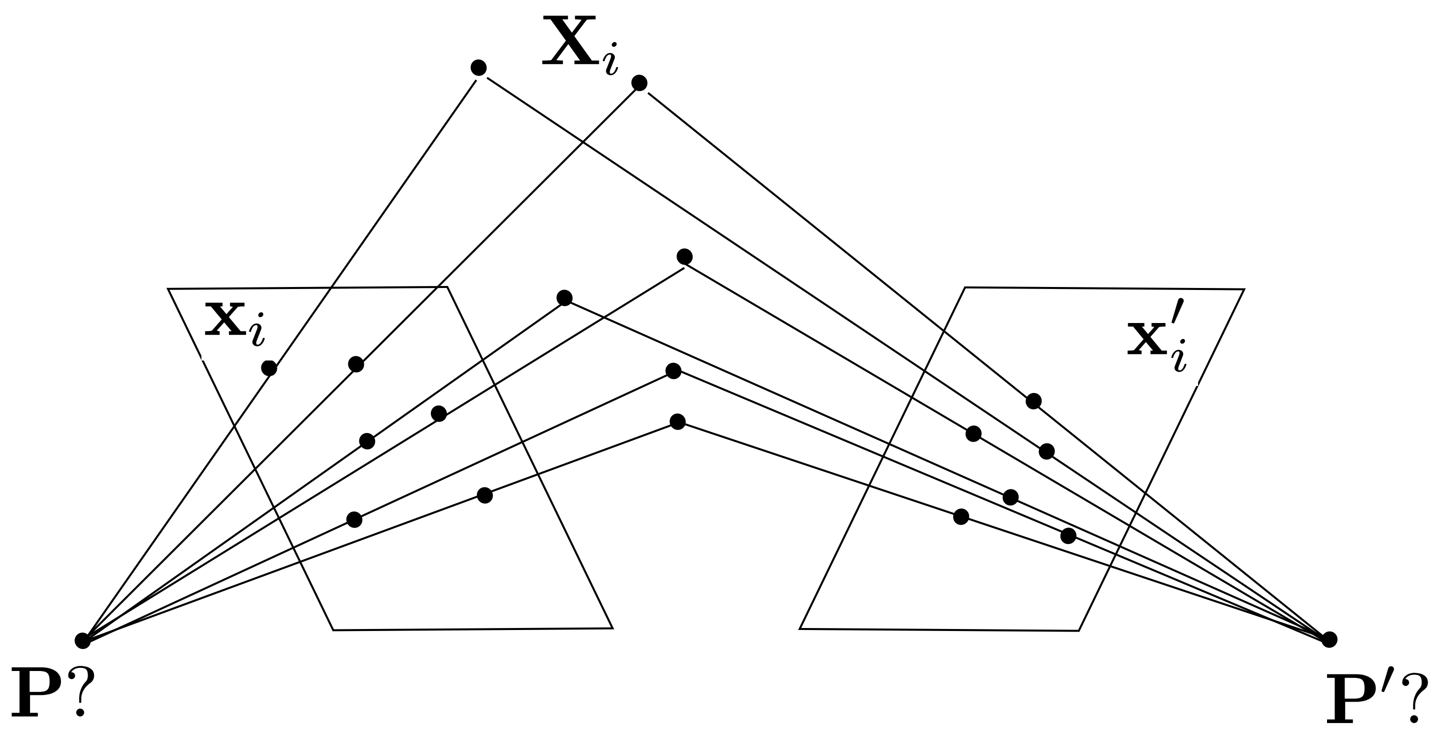
$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F x'$ is the epipolar line associated with x' ($l = F x'$)
- $F^T x$ is the epipolar line associated with x ($l' = F^T x$)
- $F e' = 0$ and $F^T e = 0$
- F is singular (rank two)
- F has seven degrees of freedom

Slide credit: Svetlana Lazebnik

Estimating the Fundamental Matrix

- The Fundamental matrix defines the epipolar geometry between two uncalibrated cameras.
- How can we estimate F from an image pair?
  - We need correspondences…



The Eight-Point Algorithm

$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x'} = (u', v', 1)^T$

$$(u,v,1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \Rightarrow [u'u, u'v, u', uv', vv', v', u, v, 1]\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

- Taking 8 correspondences:

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & u_1v'_1 & v_1v'_1 & v'_1 & u_1 & v_1 & 1 \\ u'_2u_2 & u'_2v_2 & u'_2 & u_2v'_2 & v_2v'_2 & v'_2 & u_2 & v_2 & 1 \\ u'_3u_3 & u'_3v_3 & u'_3 & u_3v'_3 & v_3v'_3 & v'_3 & u_3 & v_3 & 1 \\ u'_4u_4 & u'_4v_4 & u'_4 & u_4v'_4 & v_4v'_4 & v'_4 & u_4 & v_4 & 1 \\ u'_5u_5 & u'_5v_5 & u'_5 & u_5v'_5 & v_5v'_5 & v'_5 & u_5 & v_5 & 1 \\ u'_6u_6 & u'_6v_6 & u'_6 & u_6v'_6 & v_6v'_6 & v'_6 & u_6 & v_6 & 1 \\ u'_7u_7 & u'_7v_7 & u'_7 & u_7v'_7 & v_7v'_7 & v'_7 & u_7 & v_7 & 1 \\ u'_8u_8 & u'_8v_8 & u'_8 & u_8v'_8 & v_8v'_8 & v'_8 & u_8 & v_8 & 1 \end{bmatrix}\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Af} = \mathbf{0}$$

Solve using… SVD!

This minimizes:

$$\sum_{i=1}^{N} (x_i^T F x'_i)^2$$

Excursion: Properties of SVD

- Frobenius norm
  - Generalization of the Euclidean norm to matrices

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

- Partial reconstruction property of SVD
  - Let σ_i $i=1,\ldots,N$ be the singular values of A.
  - Let $A_p = U_p D_p V_p^T$ be the reconstruction of A when we set $\sigma_{p+1},\ldots,\ \sigma_N$ to zero.
  - Then $A_p = U_p D_p V_p^T$ is the best rank-p approximation of A in the sense of the Frobenius norm (i.e. the best least-squares approximation).

The Eight-Point Algorithm

- Problem with noisy data
  - The solution will usually not fulfill the constraint that F only has rank 2.

  $\Rightarrow$ *There will be no epipoles through which all epipolar lines pass!*

- Enforce the rank-2 constraint using SVD

$$F \overset{\text{SVD}}{=} \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U}\begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix}\begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T$$

Set d_{33} to zero and reconstruct F

- As we have just seen, this provides the best least-squares approximation to the rank-2 solution.

Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & u_1v'_1 & v_1v'_1 & v'_1 & u_1 & v_1 & 1 \\ u'_2u_2 & u'_2v_2 & u'_2 & u_2v'_2 & v_2v'_2 & v'_2 & u_2 & v_2 & 1 \\ u'_3u_3 & u'_3v_3 & u'_3 & u_3v'_3 & v_3v'_3 & v'_3 & u_3 & v_3 & 1 \\ u'_4u_4 & u'_4v_4 & u'_4 & u_4v'_4 & v_4v'_4 & v'_4 & u_4 & v_4 & 1 \\ u'_5u_5 & u'_5v_5 & u'_5 & u_5v'_5 & v_5v'_5 & v'_5 & u_5 & v_5 & 1 \\ u'_6u_6 & u'_6v_6 & u'_6 & u_6v'_6 & v_6v'_6 & v'_6 & u_6 & v_6 & 1 \\ u'_7u_7 & u'_7v_7 & u'_7 & u_7v'_7 & v_7v'_7 & v'_7 & u_7 & v_7 & 1 \\ u'_8u_8 & u'_8v_8 & u'_8 & u_8v'_8 & v_8v'_8 & v'_8 & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1

$$\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$ Poor numerical conditioning

$\Rightarrow$ Can be fixed by rescaling the data

The Normalized Eight-Point Algorithm

1. Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels.
2. Use the eight-point algorithm to compute *F* from the normalized points.
3. Enforce the rank-2 constraint using SVD.

$$F \overset{\text{SVD}}{=} \boldsymbol{UDV}^T = \boldsymbol{U}\begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix}\begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T$$

Set d_{33} to zero and reconstruct F

4. Transform fundamental matrix back to original units: if T and T' are the normalizing transformations in the two images, then the fundamental matrix in original coordinates is $T^T F T'$.

[Hartley, 1995]

The Eight-Point Algorithm

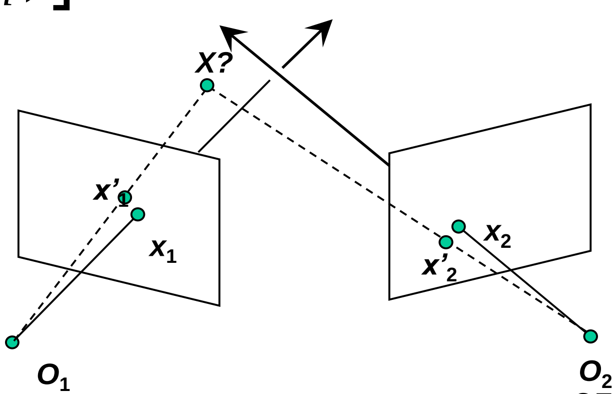
- Meaning of error $\sum_{i=1}^{N} (x_i^T F x_i')^2$:

  Sum of Euclidean distances between points x_i and epipolar lines $F x'_i$ (or points x'_i and epipolar lines $F^T x_i$), multiplied by a scale factor

- Nonlinear approach for refining the solution: minimize

$$\sum_{i=1}^{N} \left[\mathrm{d}^2(x_i, F x_i') + \mathrm{d}^2(x_i', F^T x_i) \right]$$

  - Similar to nonlinear minimization approach for triangulation.
  - Iterative approach (Gauss-Newton, Levenberg-Marquardt,…)



Comparison of Estimation Algorithms

	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

Slide credit: Svetlana Lazebnik

3D Reconstruction with Weak Calibration

- Want to estimate world geometry without requiring calibrated cameras.
- Many applications:
  - Archival videos
  - Photos from multiple unrelated users
  - Dynamic camera system

- Main idea:
  - Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras.

Stereo Pipeline with Weak Calibration

- So, where to start with uncalibrated cameras?
  - Need to find fundamental matrix F *and* the correspondences (pairs of points (u',v') ↔ (u,v)).

- Procedure
  1. Find interest points in both images
  2. Compute correspondences
  3. Compute epipolar geometry
  4. Refine

Example from Andrew Zisserman

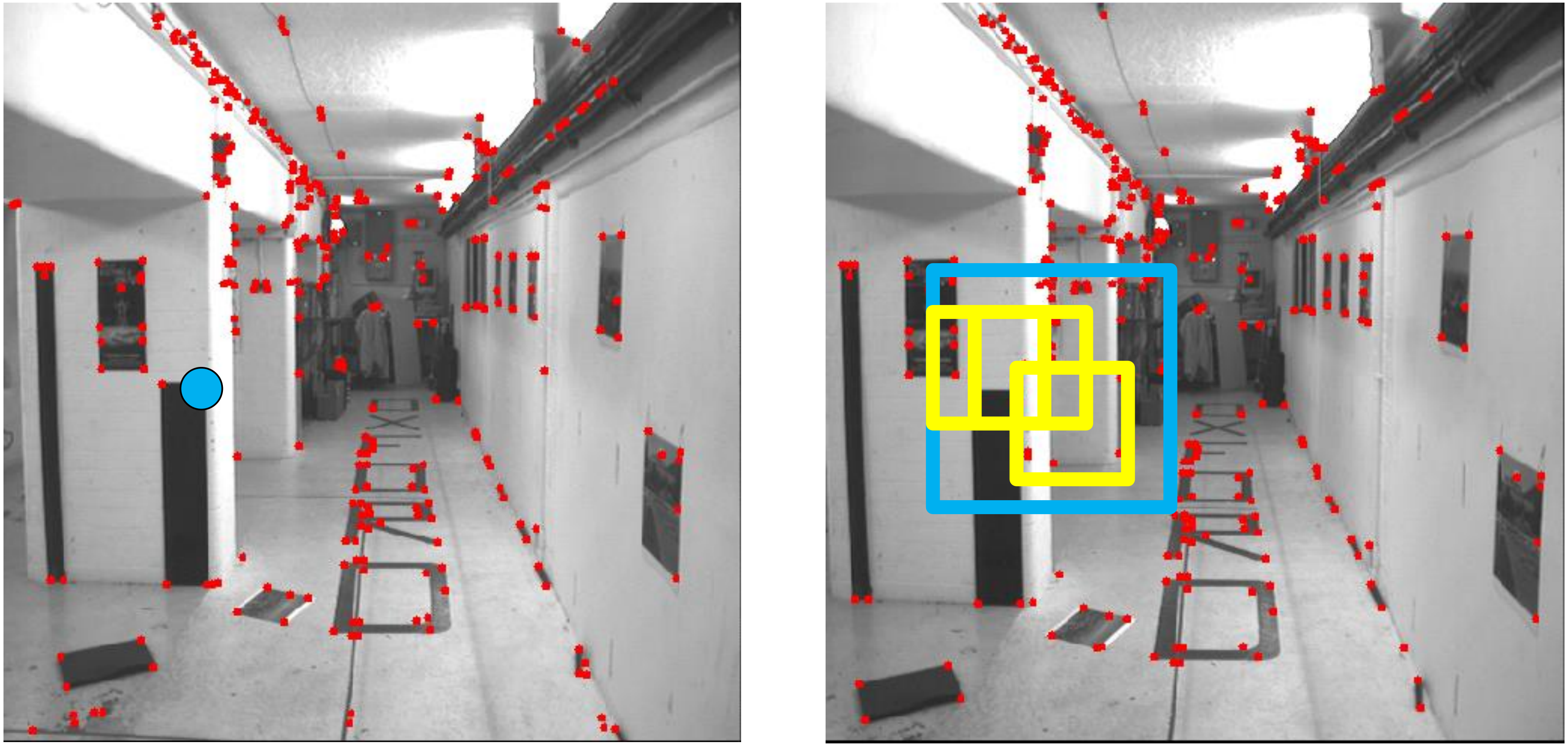
Stereo Pipeline with Weak Calibration

1. Find interest points (e.g. Harris corners)

Example from Andrew Zisserman

Stereo Pipeline with Weak Calibration

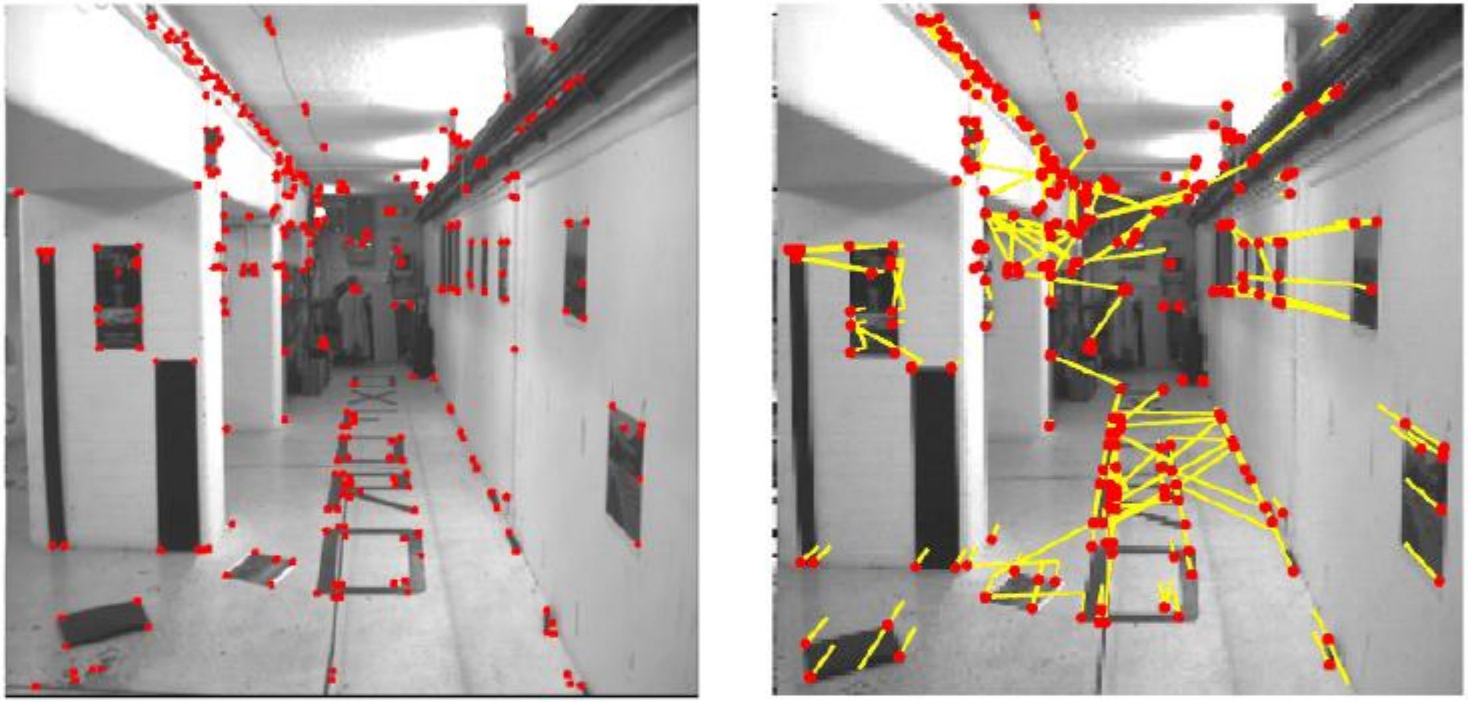
2. Match points using only proximity



Slide credit: Kristen Grauman

Example from Andrew Zisserman

Putative Matches based on Correlation Search

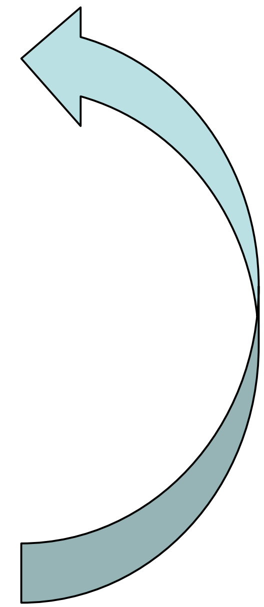


- Many wrong matches (10-50%), but enough to compute F

Example from Andrew Zisserman

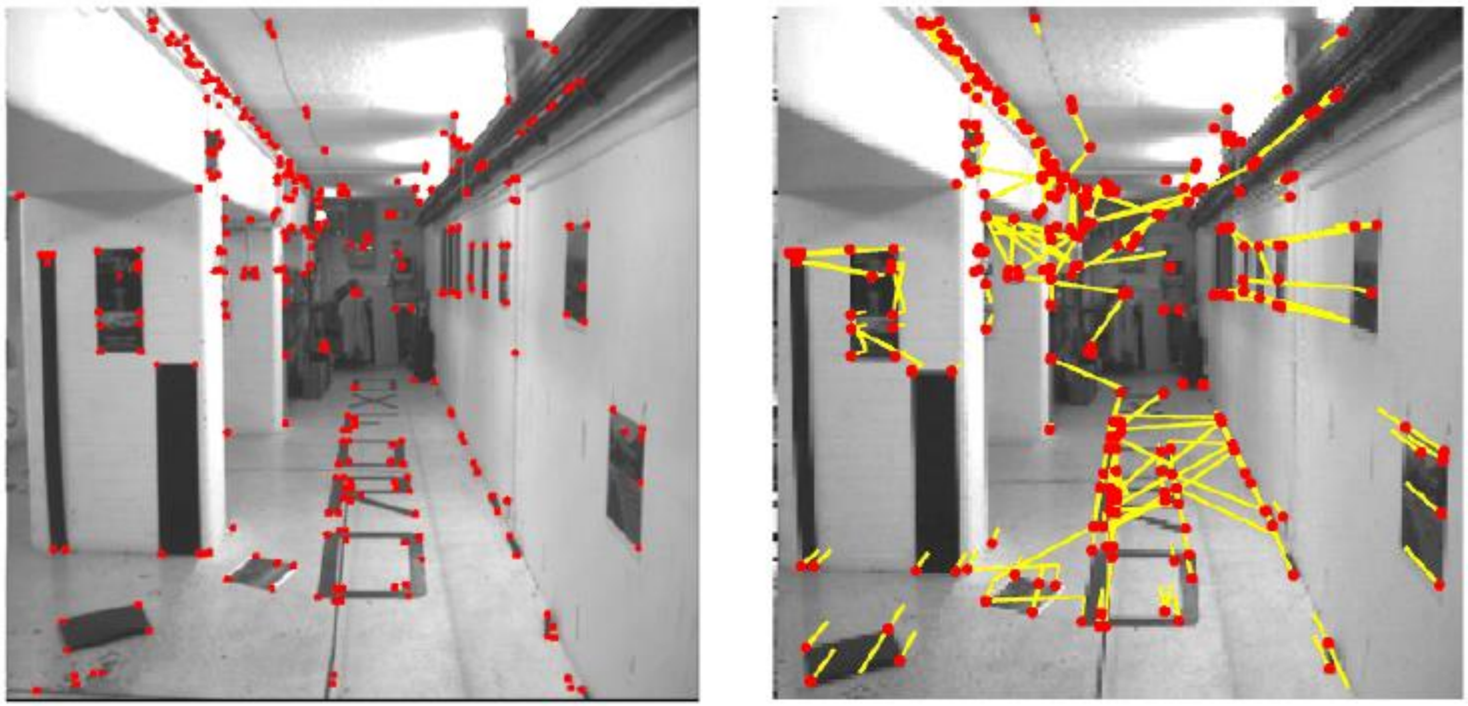
RANSAC for Robust Estimation of F

- Select random sample of correspondences
- Compute F using them
  - This determines epipolar constraint
- Evaluate amount of support – number of inliers within threshold distance of epipolar line
- Iterate until a solution with sufficient support has been found (or for max #iterations)



- Choose F with most support (#inliers)

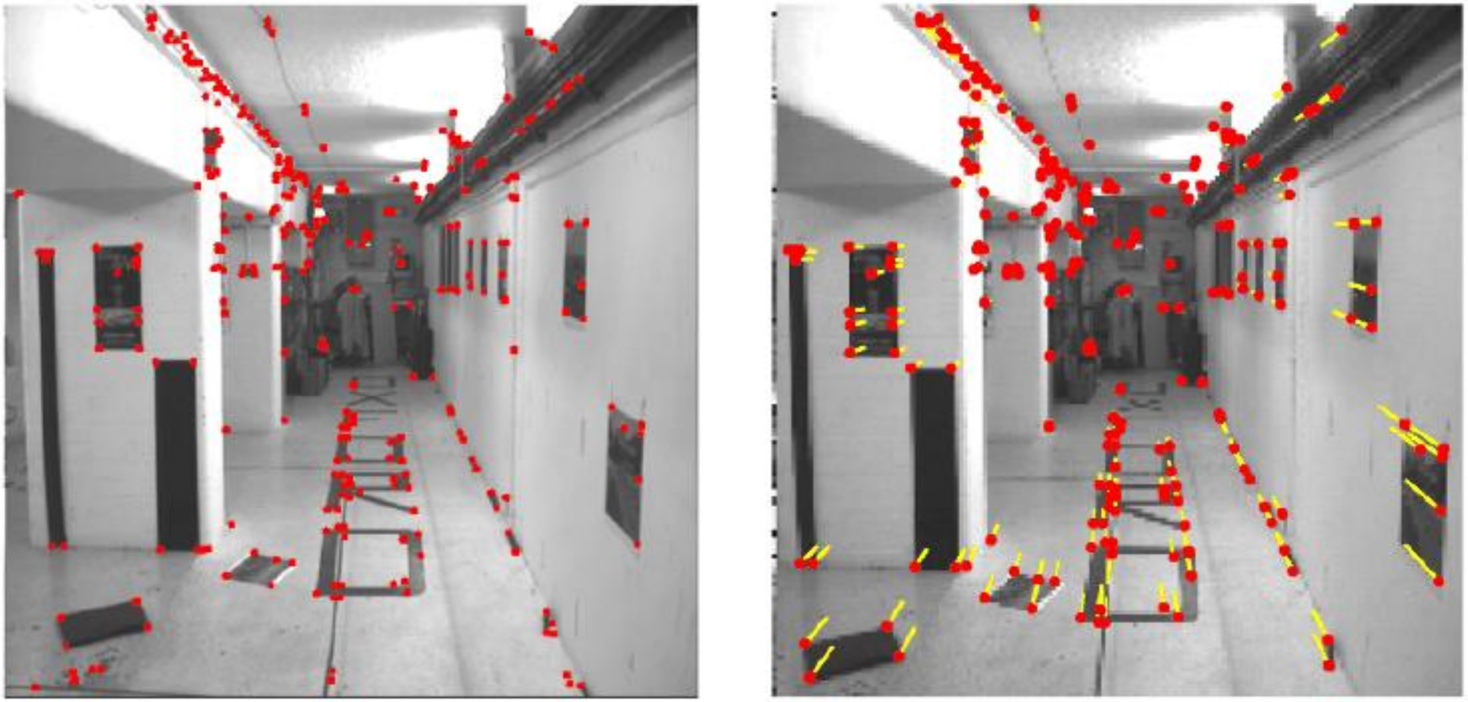
Putative Matches based on Correlation Search



- Many wrong matches (10-50%), but enough to compute F

Pruned Matches

- Correspondences consistent with epipolar geometry



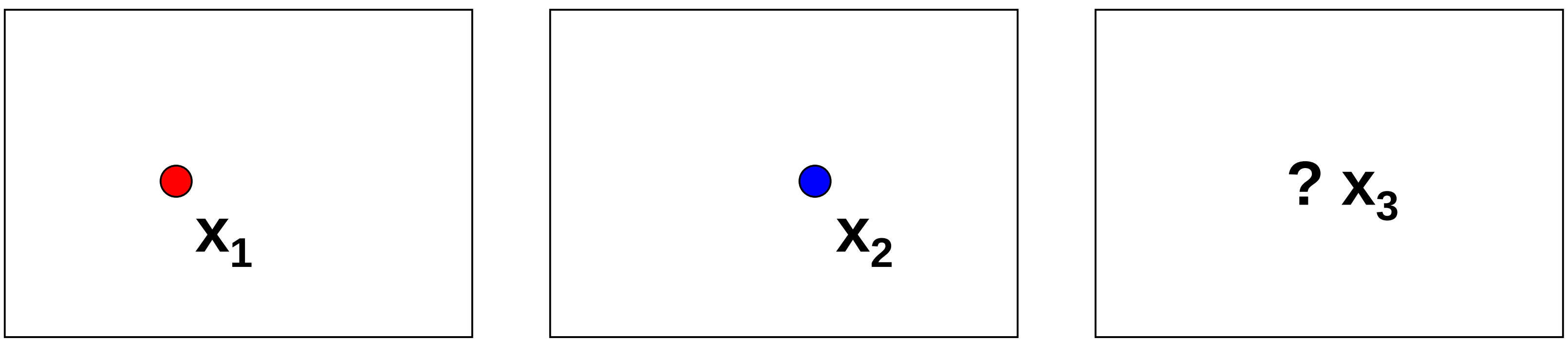
Example from Andrew Zisserman

Resulting Epipolar Geometry

Example from Andrew Zisserman

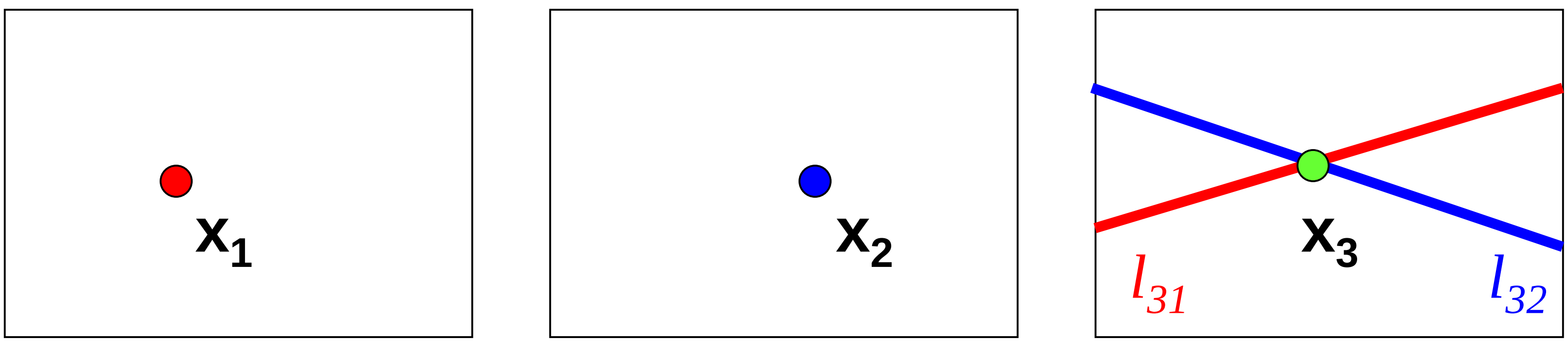
Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?



Extension: Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?



$$l_{31} = F^{T}{}_{13}\, x_1$$

$$l_{32} = F^{T}{}_{23}\, x_2$$

When does epipolar transfer fail?

Topics of This Lecture

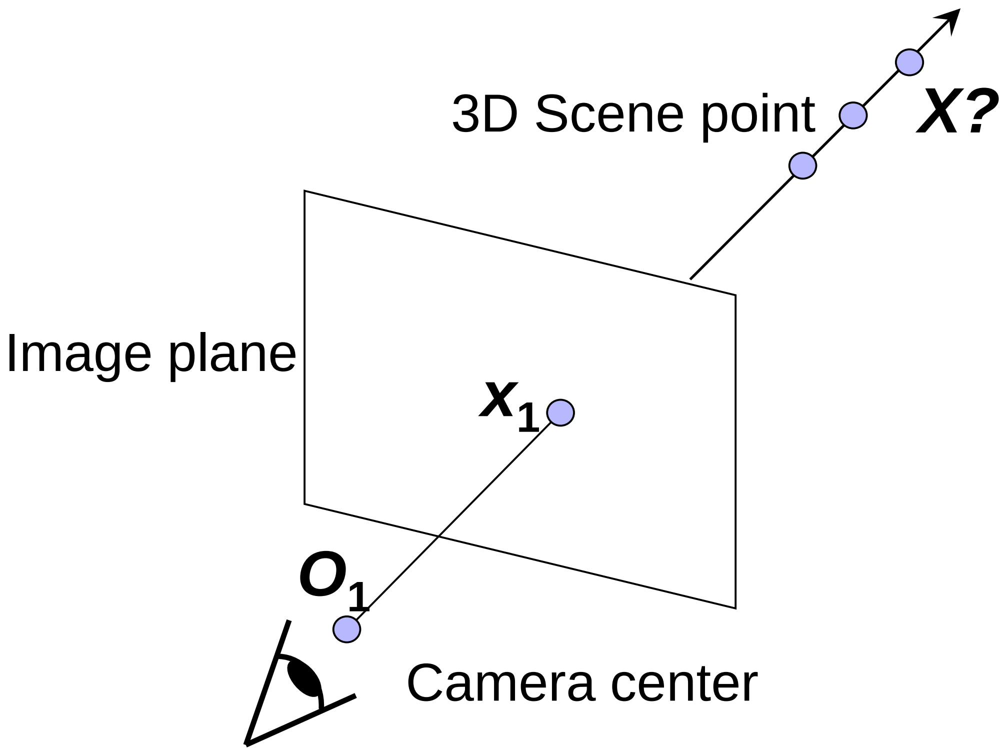
- Revisiting Epipolar Geometry
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Kinect sensor
  - Structured Light sensing
  - Laser scanning
- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
  - Projective factorization
  - Bundle adjustment

Microsoft Kinect – How Does It Work?

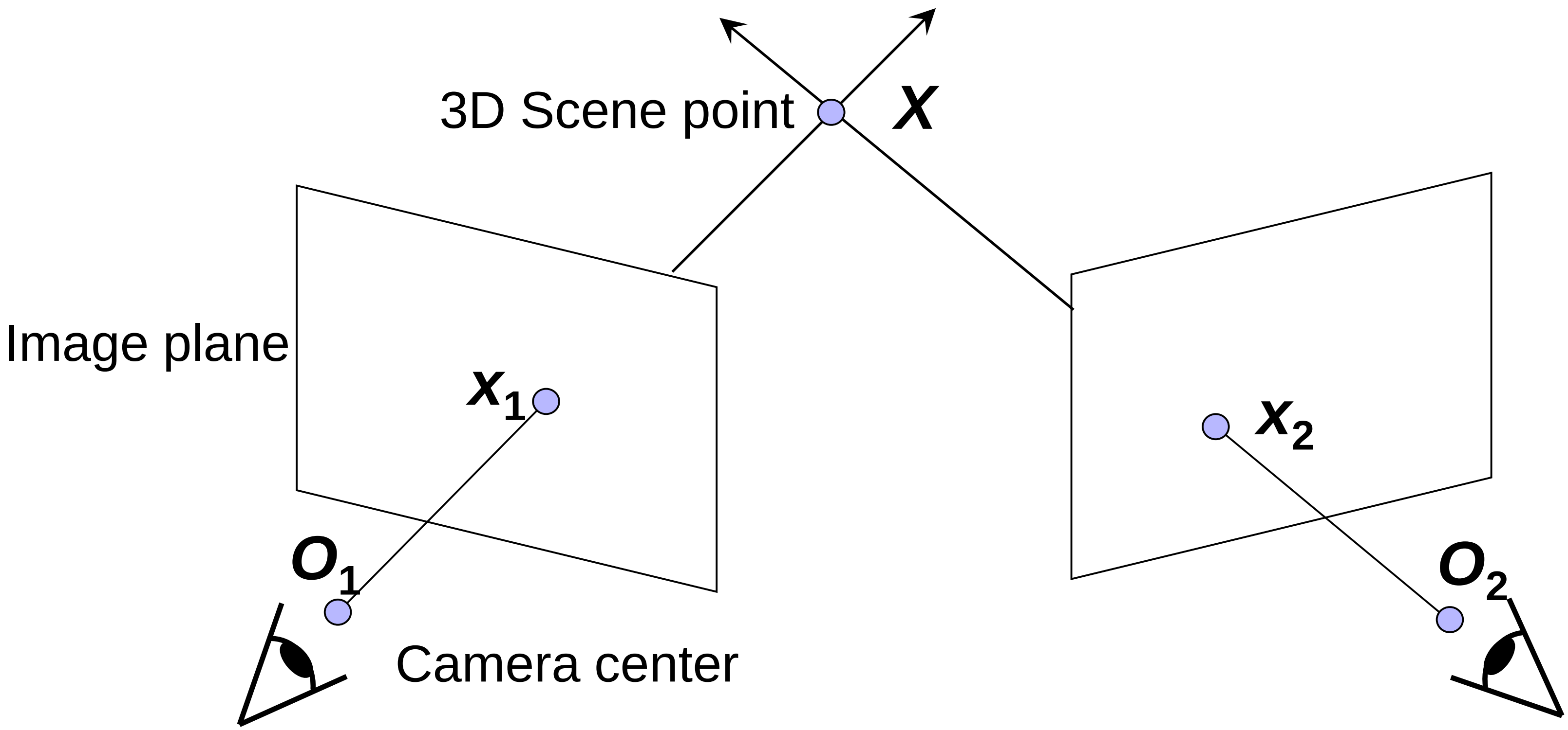
KINECT for XBOX 360.

- Built-in IR projector
- IR camera for depth
- Regular camera for color

Recall: Optical Triangulation

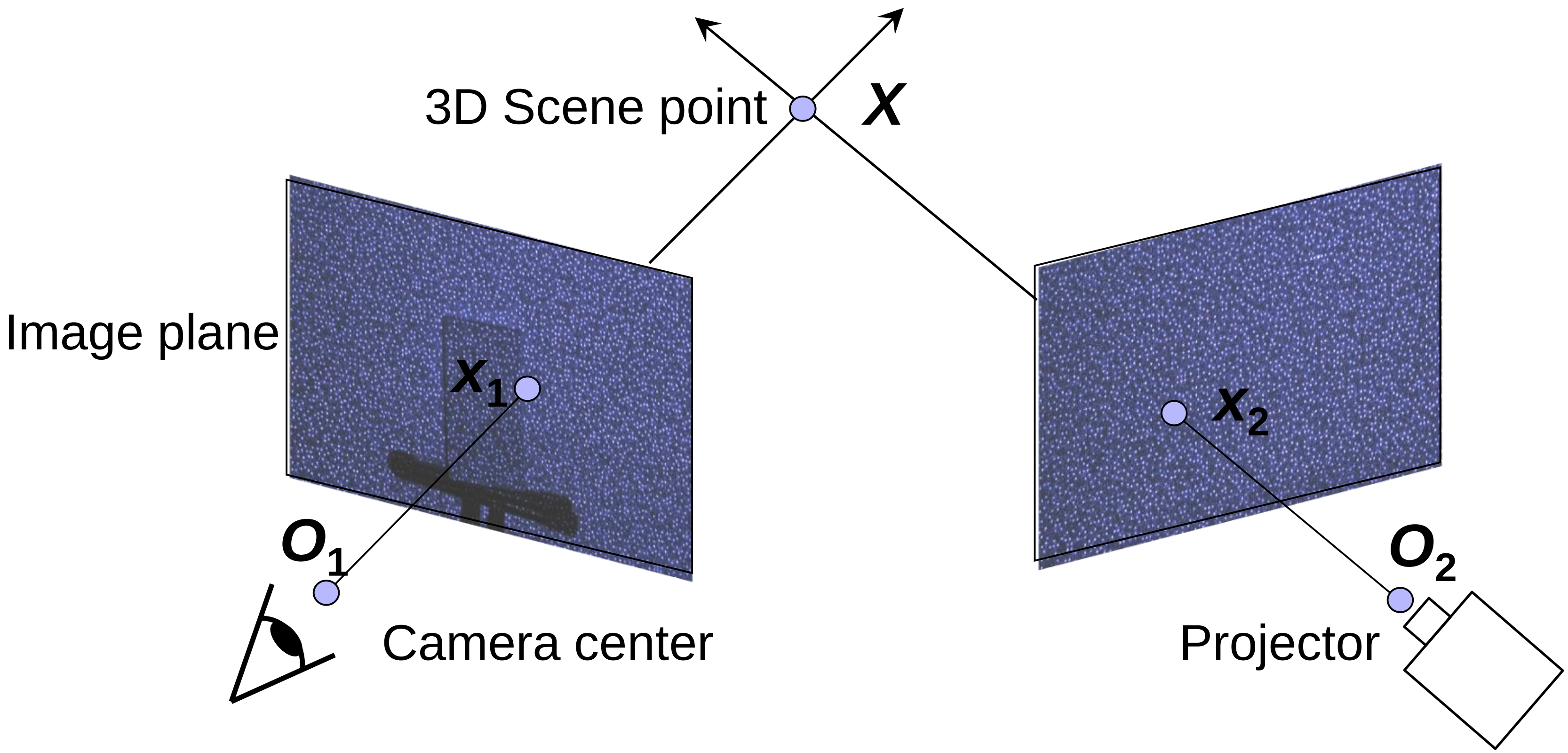


Recall: Optical Triangulation



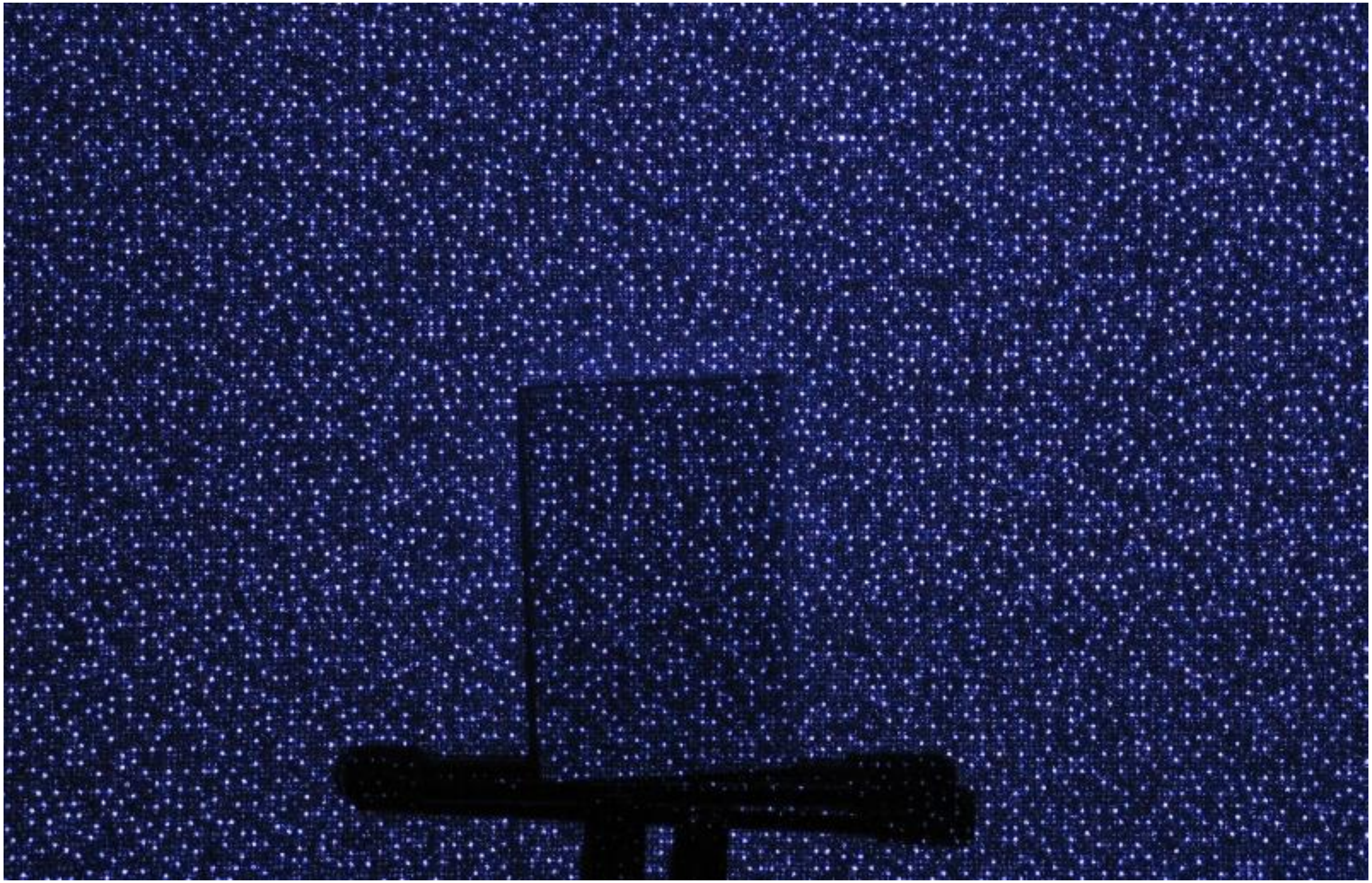
- Principle: 3D point given by intersection of two rays.
  - Crucial information: point correspondence
  - Most expensive and error-prone step in the pipeline…

Active Stereo with Structured Light



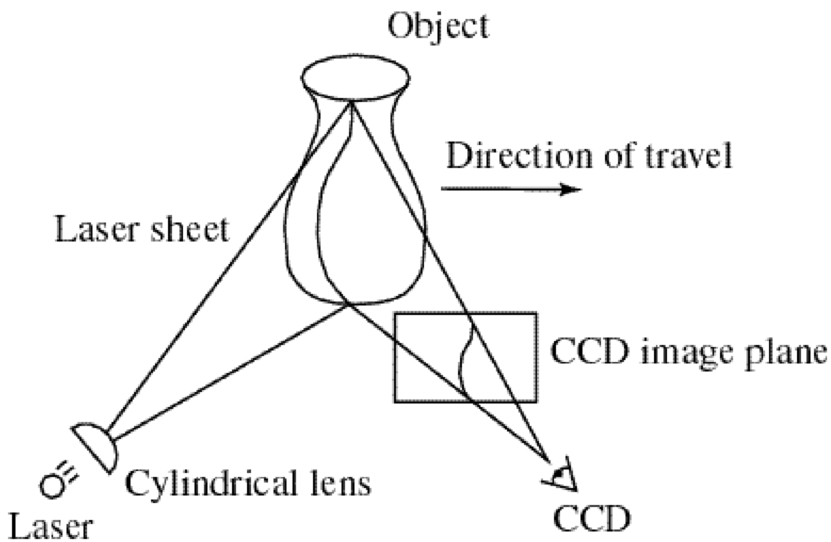
- Idea: Replace one camera by a projector.
  - Project "structured" light patterns onto the object
  - Simplifies the correspondence problem

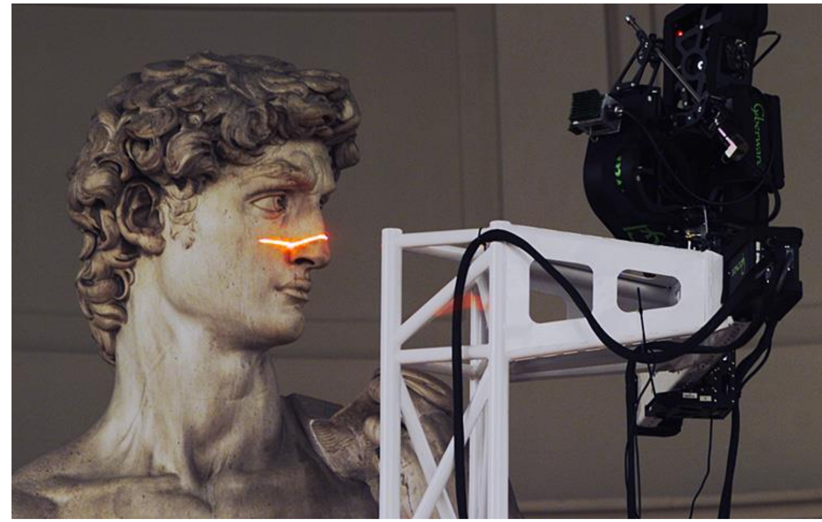
What the Kinect Sees…



3D Reconstruction with the Kinect

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

Laser Scanned Models

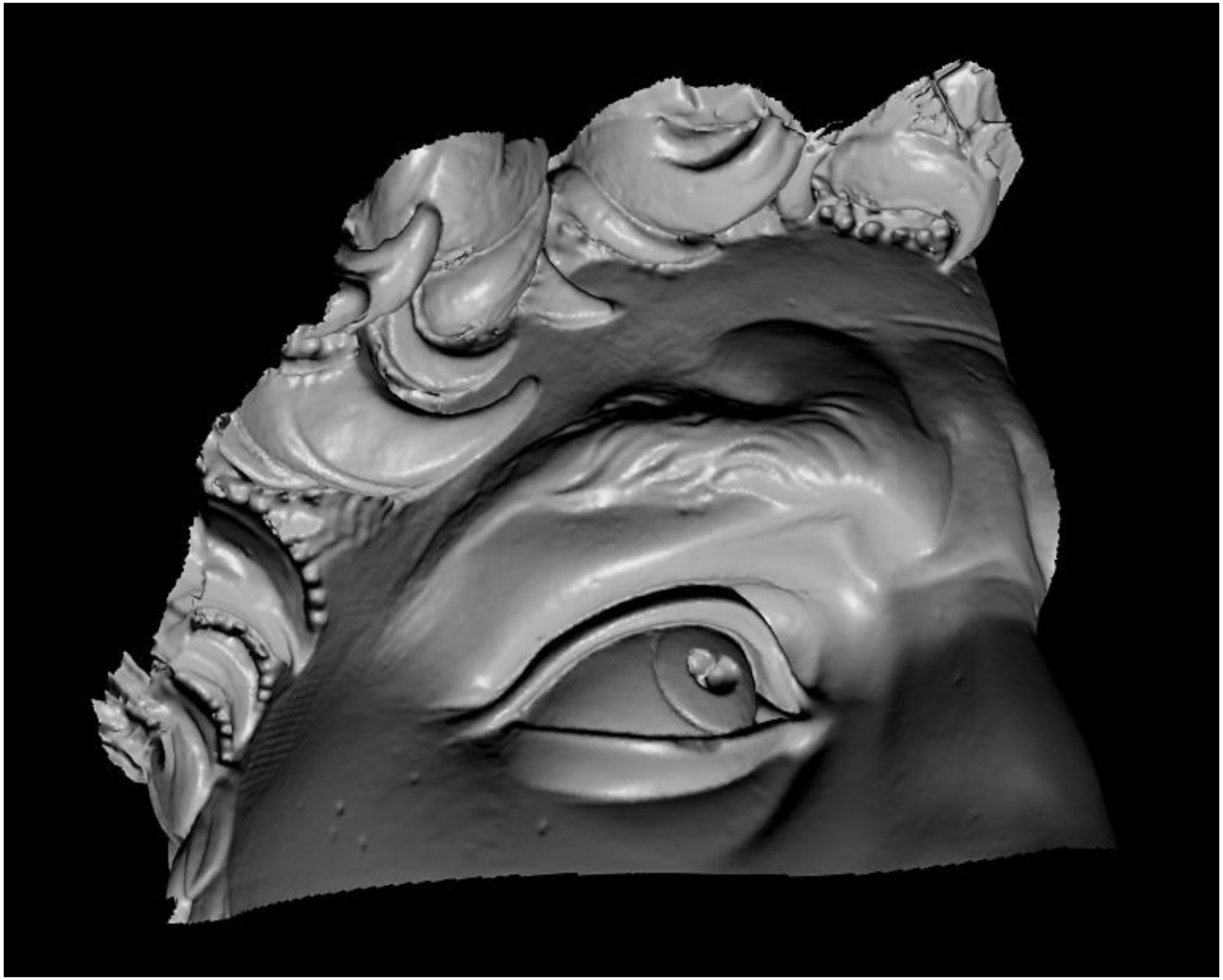
The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

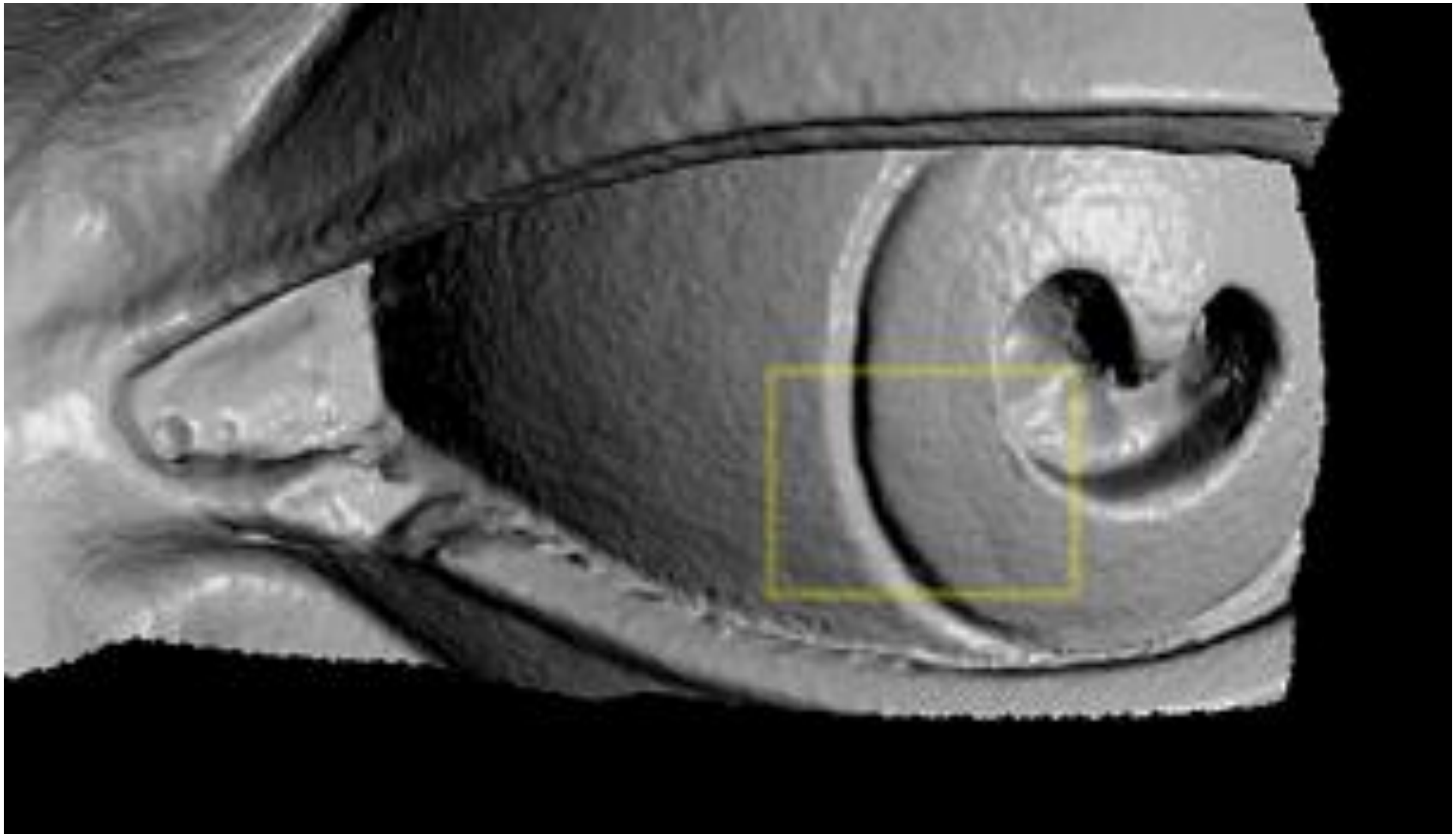
Slide credit: Steve Seitz

Laser Scanned Models



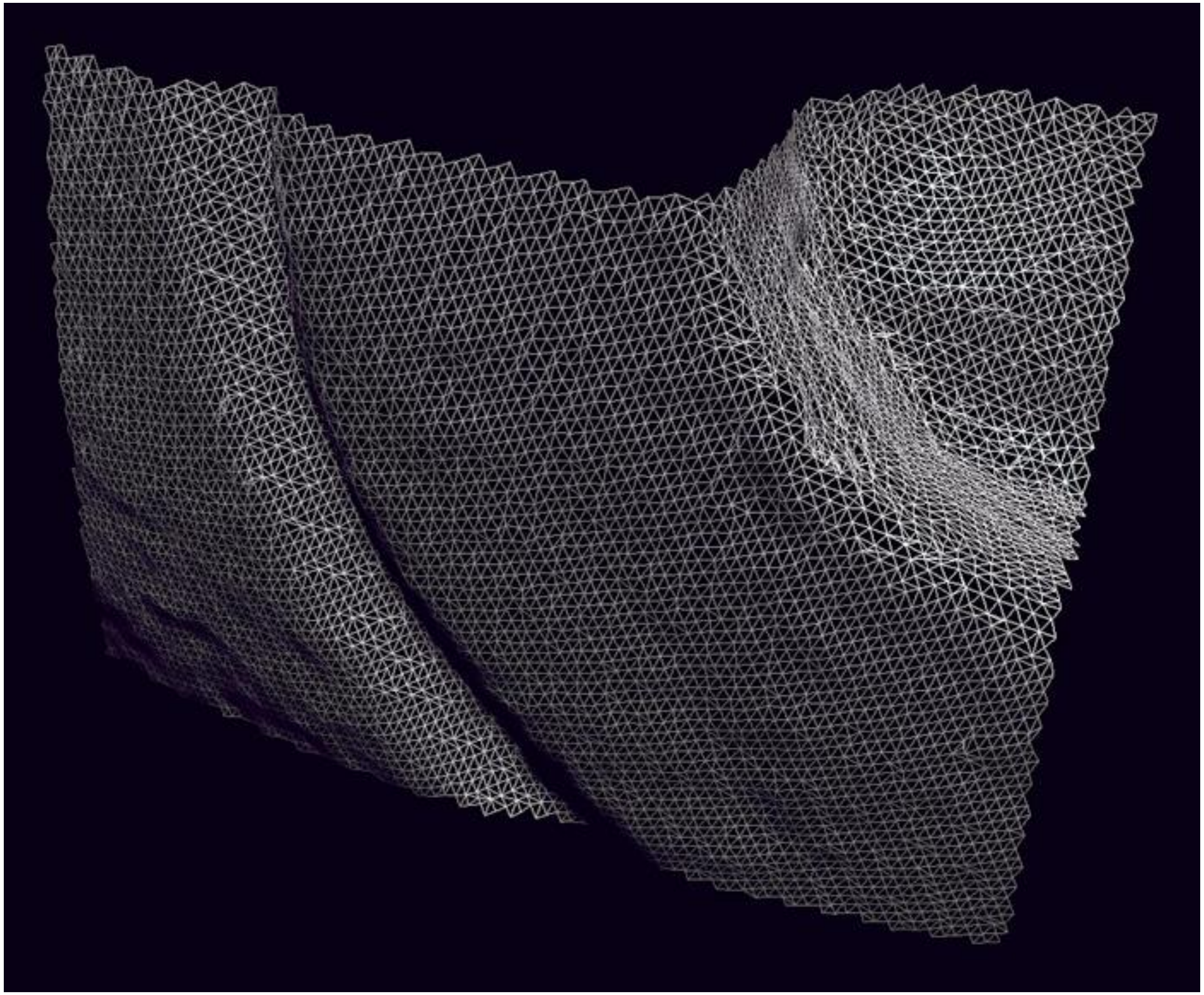
The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models



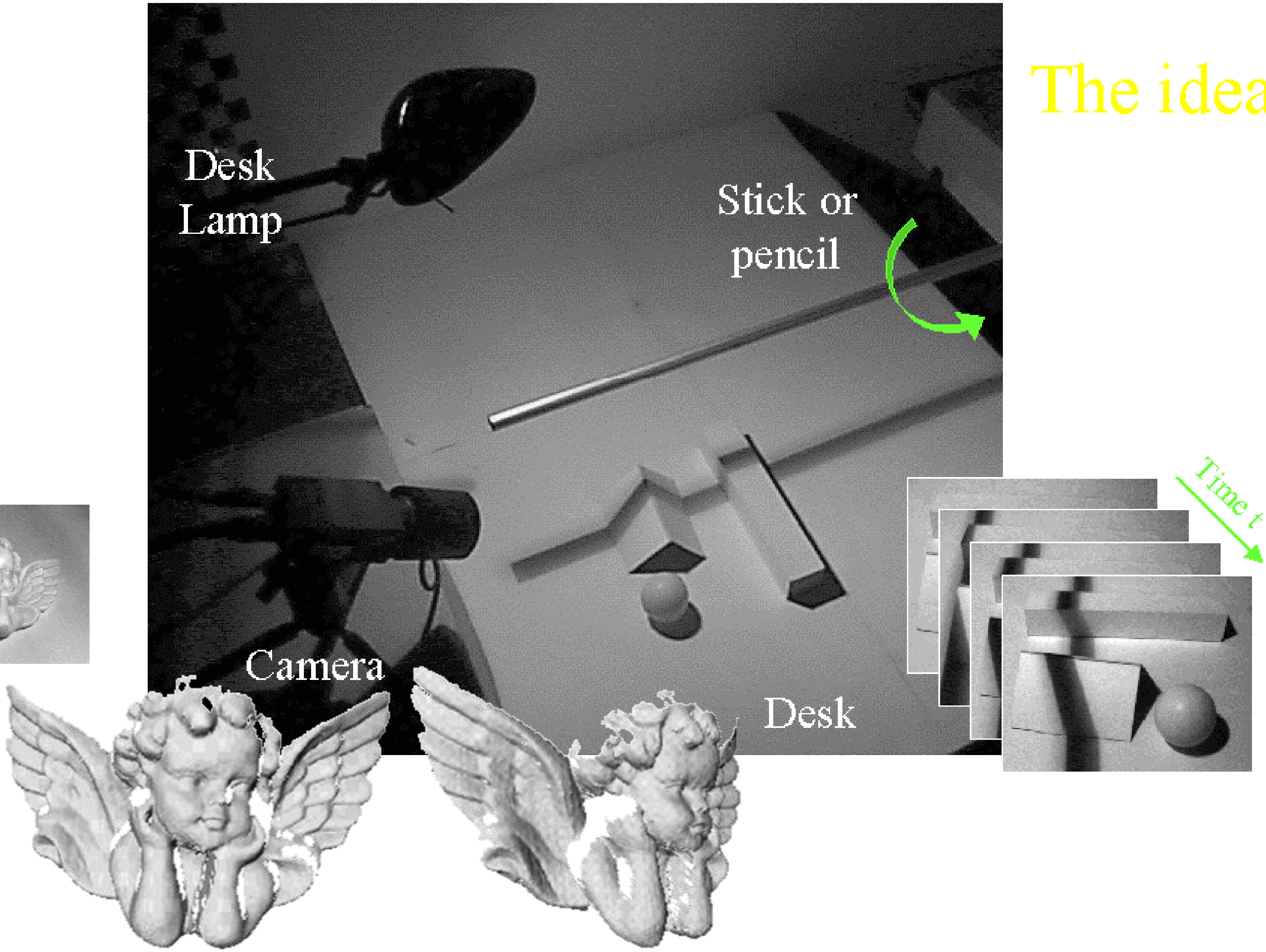
The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Poor Man’s Scanner



Bouget and Perona, ICCV’98

Slightly More Elaborate (But Still Cheap)

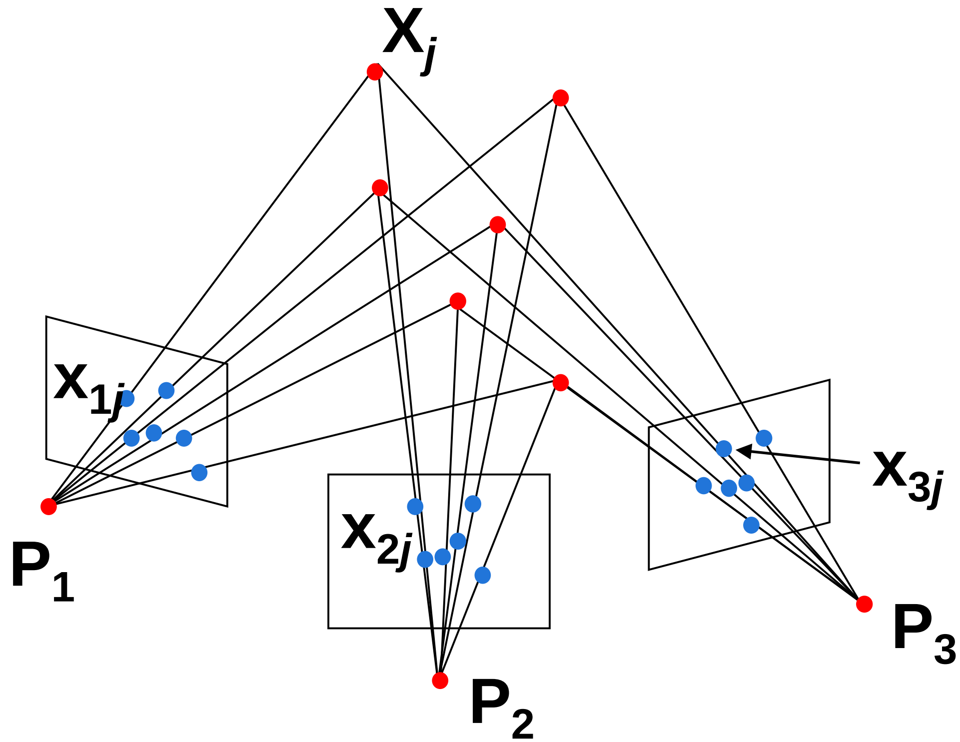
Software freely available from Robotics Institute TU Braunschweig

http://www.david-laserscanner.com/

Topics of This Lecture

- Revisiting Epipolar Geometry
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Kinect sensor
  - Structured Light sensing
  - Laser scanning
- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
  - Projective factorization
  - Bundle adjustment

Structure from Motion



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

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

- Problem: estimate *m* projection matrices P_i and *n* 3D points X_j from the *mn* correspondences x_{ij}

Applications

- E.g., movie special effects

Video

Video Credit: Stefan Hafeneger

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:

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

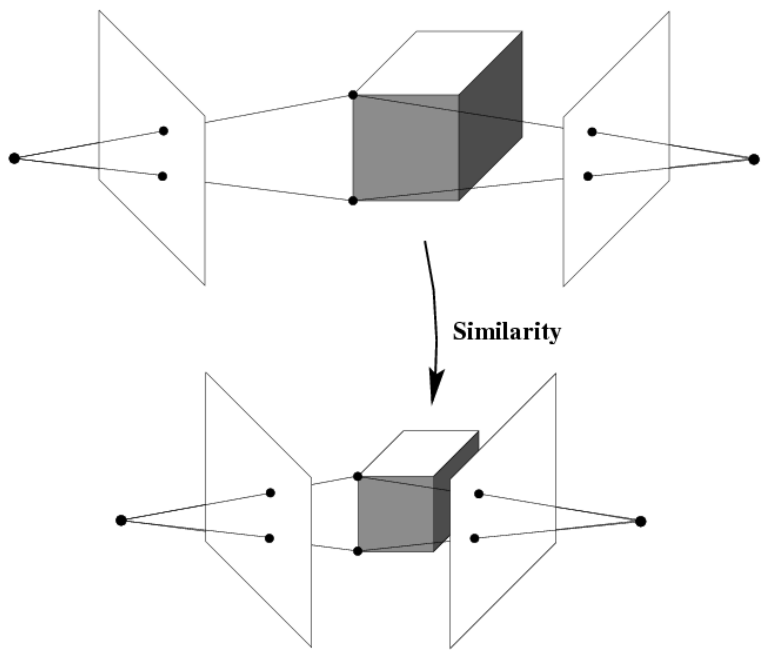
$\Rightarrow$ It is impossible to recover the absolute scale of the scene!

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

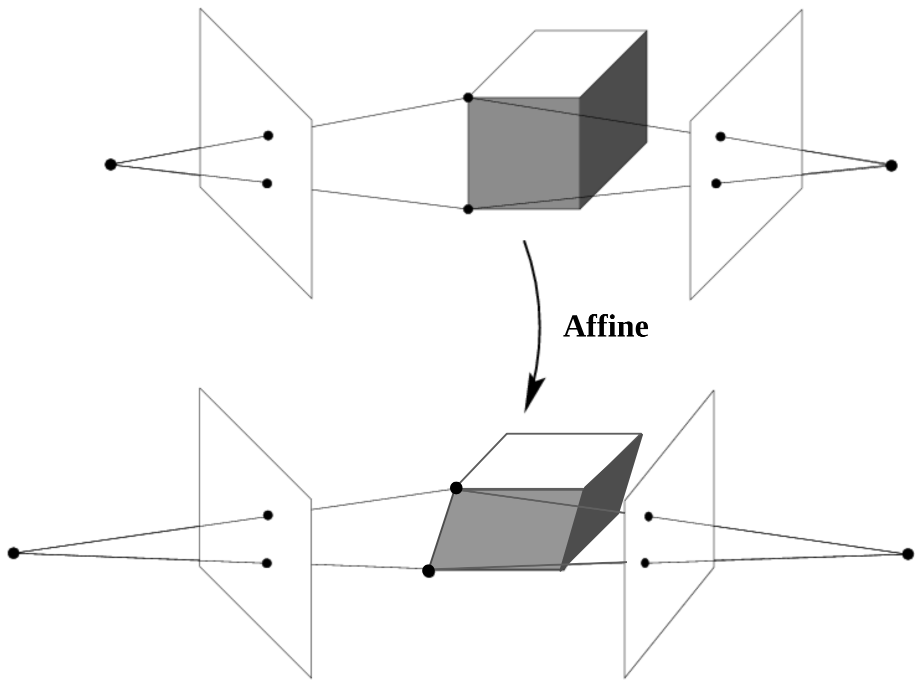
$$\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{Q}^{-1})\mathbf{Q}\mathbf{X}$$

Reconstruction Ambiguity: Similarity



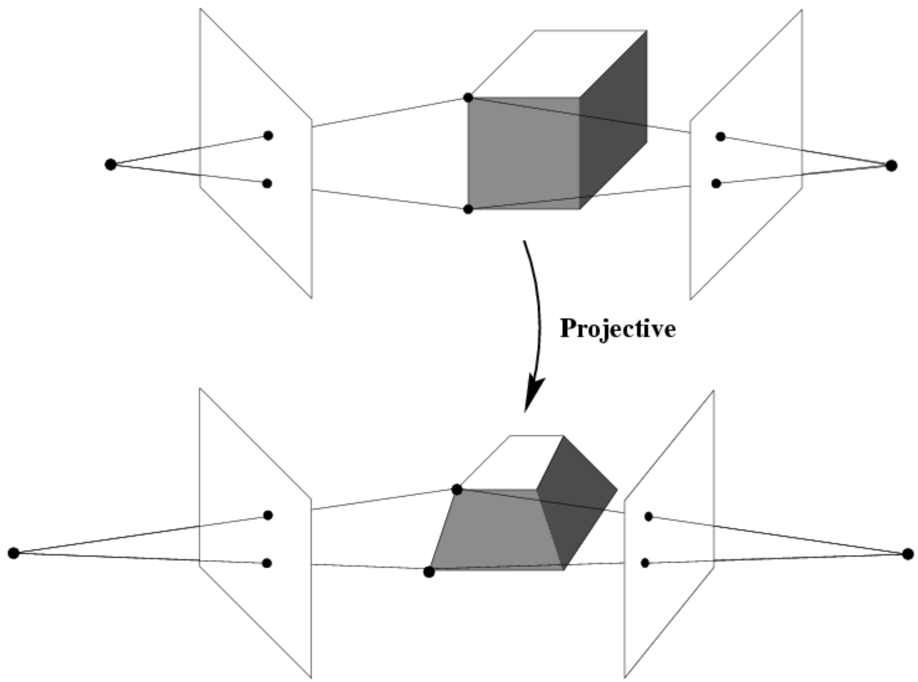
$$\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{Q}_S^{-1})\mathbf{Q}_S\mathbf{X}$$

Reconstruction Ambiguity: Affine



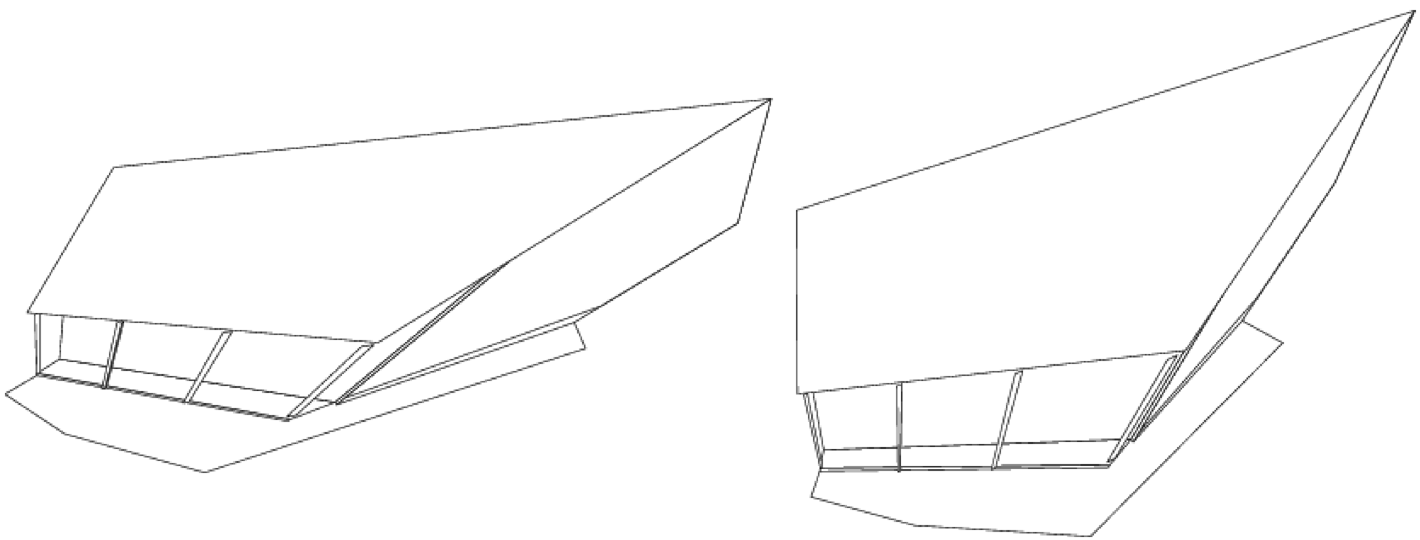
$$\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{Q}_A^{-1})\mathbf{Q}_A\mathbf{X}$$

Reconstruction Ambiguity: Projective



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

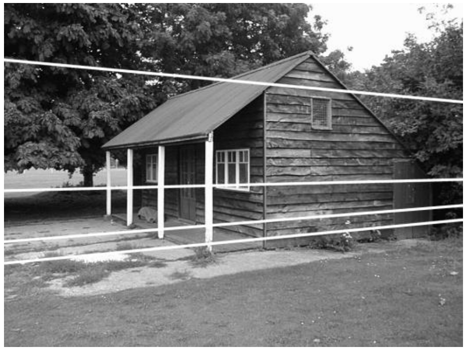
Projective Ambiguity

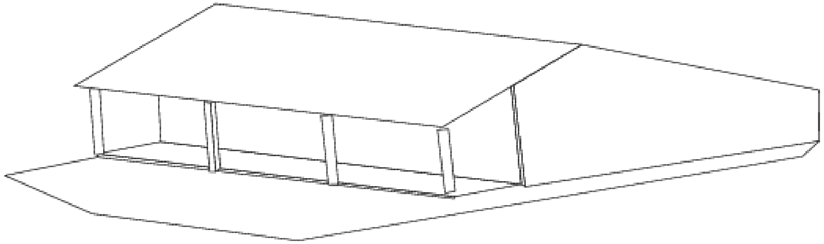


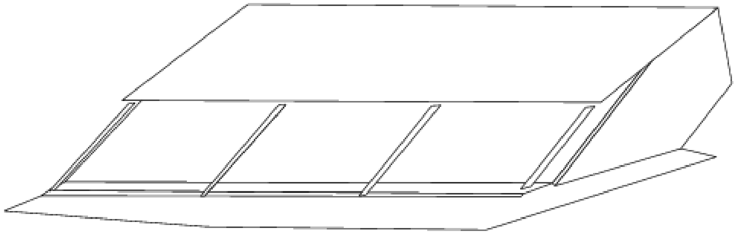
Slide credit: Svetlana Lazebnik

Images from Hartley & Zisserman

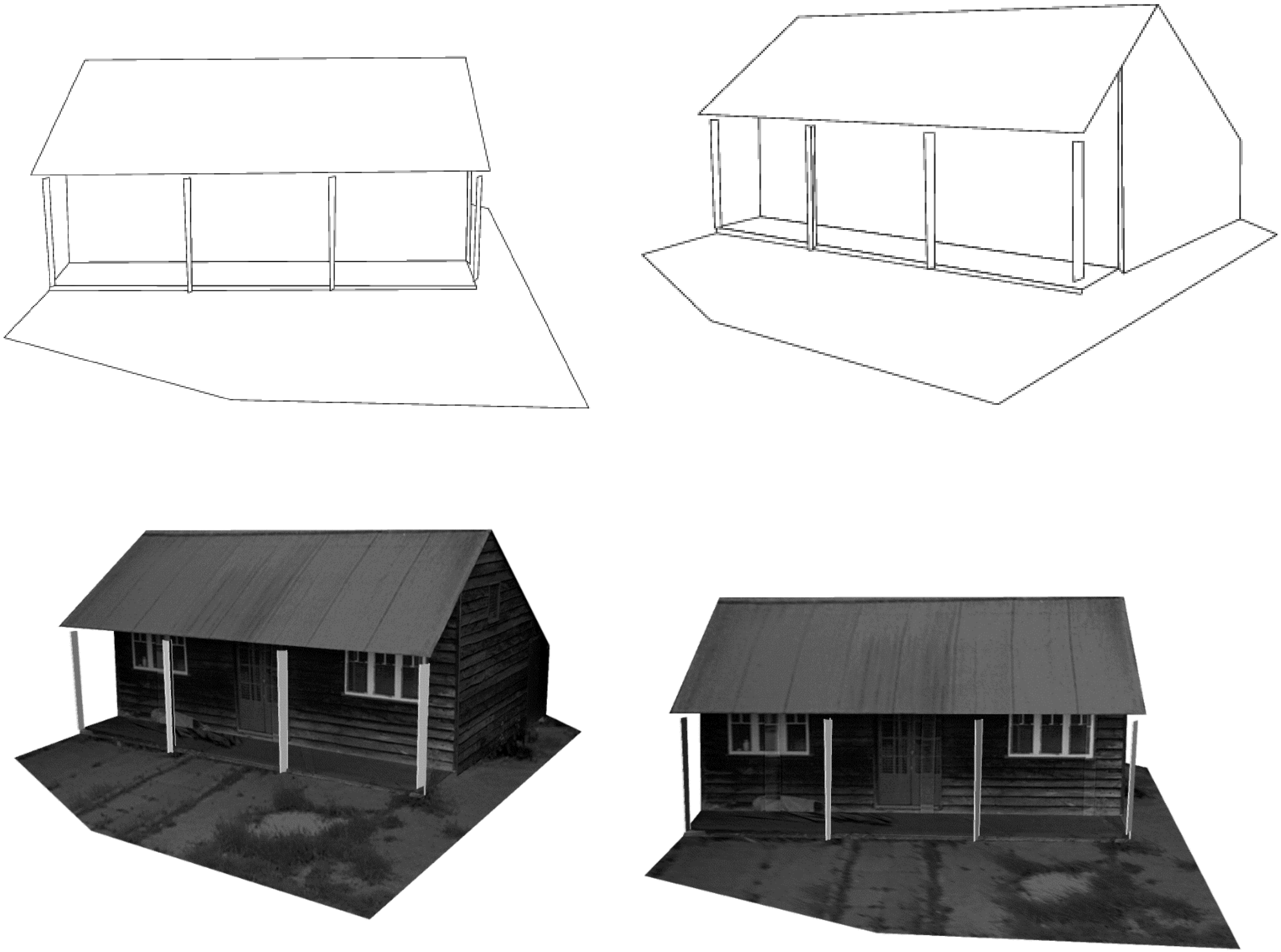
From Projective to Affine







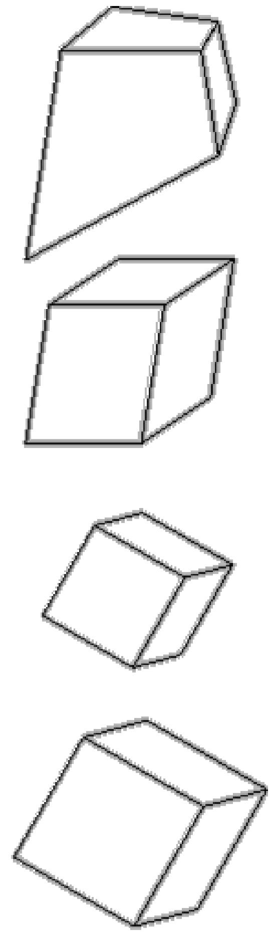
From Affine to Similarity



Images from Hartley & Zisserman

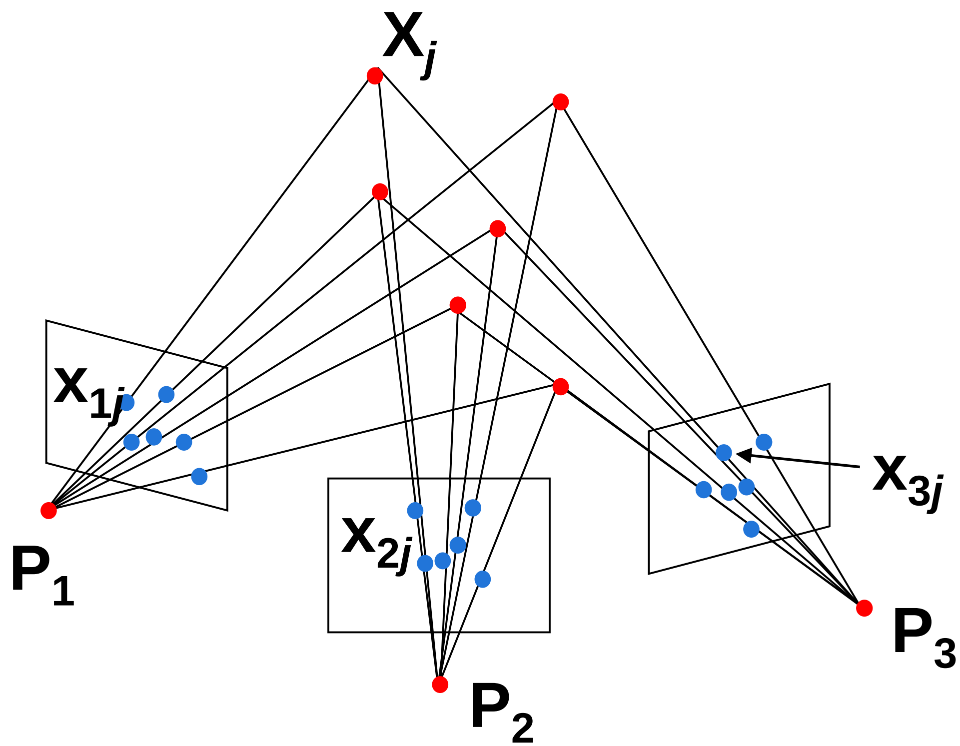
Hierarchy of 3D Transformations

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 parallellism, volume ratios**
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$	**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.

Structure from Motion



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

$$x_{ij} = P_i X_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate *m* projection matrices P_i and *n* 3D points X_j from the *mn* correspondences x_{ij}

Projective Structure from Motion

- Given: *m* images of *n* fixed 3D points
  - $z_{ij}\, x_{ij} = P_i\, X_j\,, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- Problem: estimate *m* projection matrices P_i and *n* 3D points X_j from the *mn* correspondences x_{ij}
- With no calibration info, cameras and points can only be recovered up to a 4×4 projective transformation Q:

$$X \rightarrow QX,\; P \rightarrow PQ^{-1}$$

- 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

- Assume fundamental matrix $\mathbf{F}$ between the two views
  - First camera matrix: $[I|0]Q^{-1}$
  - Second camera matrix: $[A|b]Q^{-1}$
- Let $\tilde{\mathbf{X}} = \mathbf{Q}\mathbf{X}$, then $z\boldsymbol{x} = [\boldsymbol{I}\,|\,\boldsymbol{0}]\tilde{X}, \quad z'\boldsymbol{x}' = [\boldsymbol{A}|\boldsymbol{b}]\tilde{X}$
- And $z'\boldsymbol{x}' = \boldsymbol{A}[\boldsymbol{I}\,|\,\boldsymbol{0}]\tilde{X} + \boldsymbol{b} = z\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$

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

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

$$0 = (z\boldsymbol{A}\boldsymbol{x} \times \boldsymbol{b}) \cdot \boldsymbol{x}'$$

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

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

F&P sec. 13.3.1

Projective SfM: Two-Camera Case

- Decomposing the Fundamental Matrix
  - This means that if we can compute the fundamental matrix between two cameras, we can directly estimate the two projection matrices from F.
  - Once we have the projection matrices, we can compute the 3D position of any point X by triangulation.
- How can we obtain both kinds of information at the same time?

Projective Factorization

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

Points ($4 \times n$)

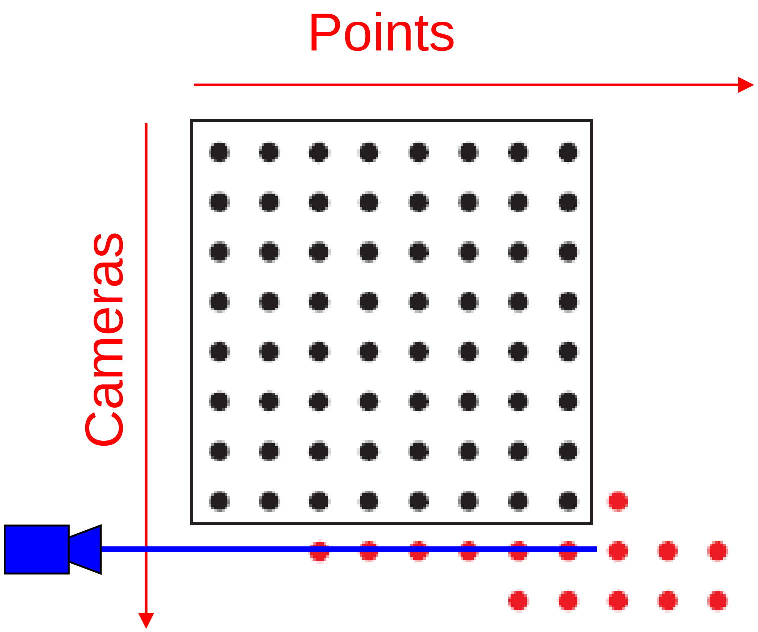
Cameras ($3m \times 4$)

$\mathbf{D} = \mathbf{MS}$ **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).

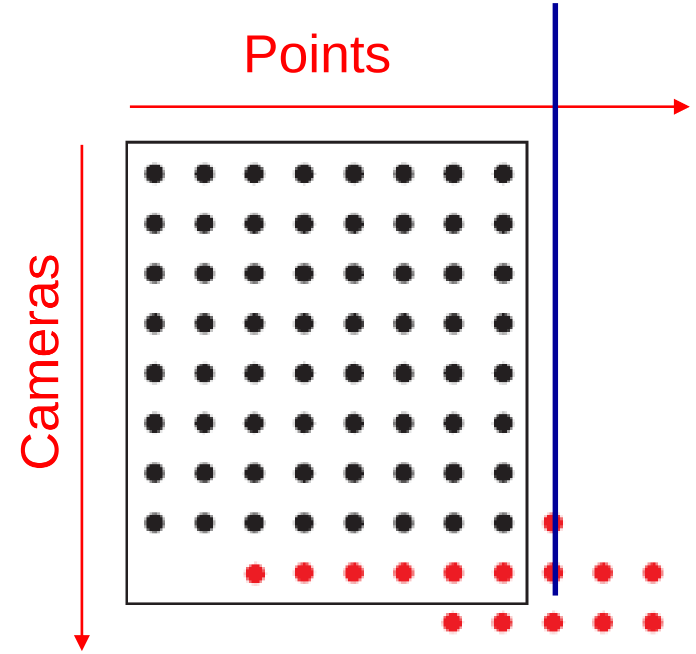
Sequential Structure from Motion

- 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*



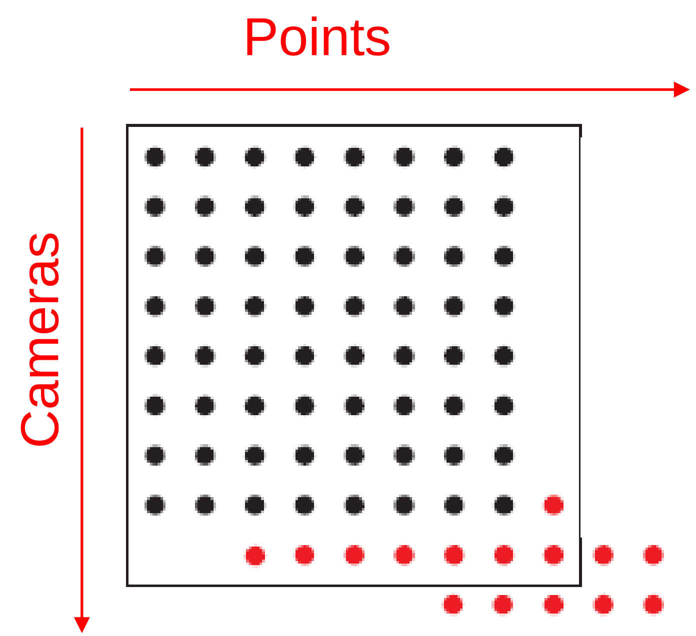
Sequential Structure from Motion

- 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

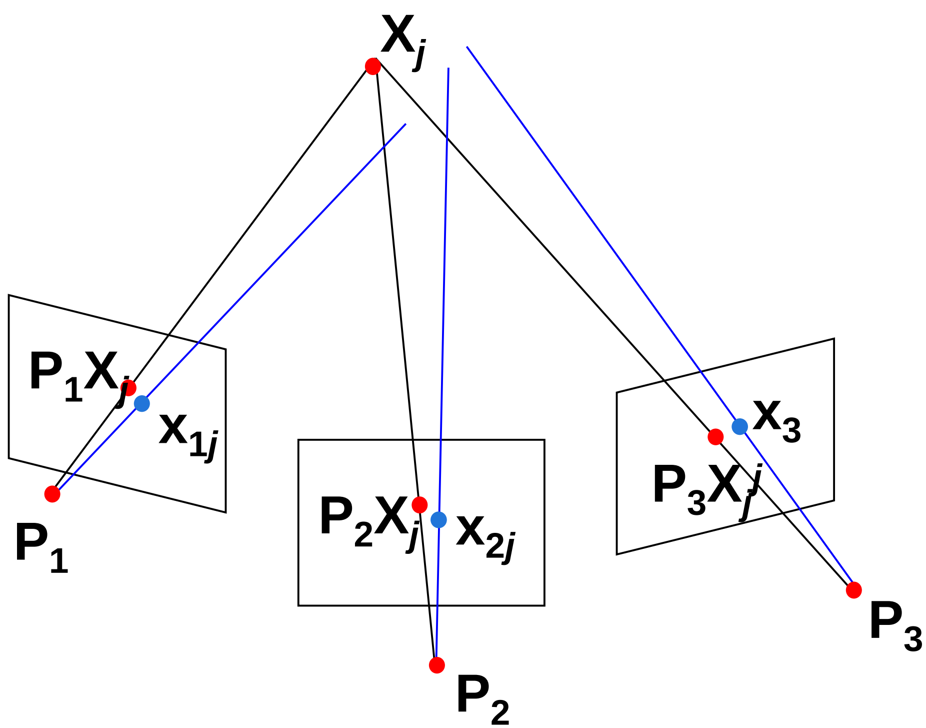
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize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



Bundle Adjustment

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

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



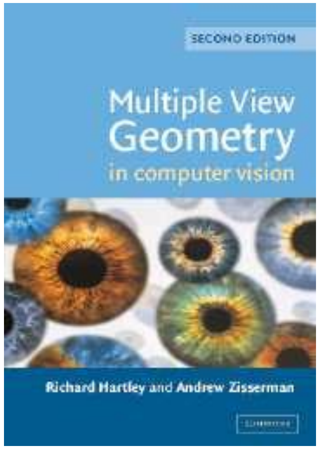
Bundle Adjustment

- Idea
  - Seek the Maximum Likelihood (ML) solution assuming the measurement noise is Gaussian.
  - It involves adjusting the bundle of rays between each camera center and the set of 3D points.
  - Bundle adjustment should generally be used as the final step of any multi-view reconstruction algorithm.
    - Considerably improves the results.
    - Allows assignment of individual covariances to each measurement.
- However…
  - It needs a good initialization.
  - It can become an extremely large minimization problem.
- Very efficient algorithms available.

References and Further Reading

- Background information on camera models and calibration algorithms can be found in Chapters 6 and 7 of

R. Hartley, A. Zisserman
Multiple View Geometry in Computer Vision
2nd Ed., Cambridge Univ. Press, 2004



- Also recommended: Chapter 9 of the same book on Epipolar geometry and the Fundamental Matrix and Chapter 11.1-11.6 on automatic computation of F.