EE 6882
Visual Search Engine

Prof. Shih-Fu Chang, March 19, 2012
Lecture #8

- Mid-Term Project Presentation
- Structure from Motion: Photo Synth/Tourism

Course Project

- 3/26 mid-term presentation slides due (before class)
- 4/2 final project proposal due
- 3/26, 4/2, 4/9 mid-term presentations in class
- Evaluation by peer students (50%) and instructor (50%)
- Awards: Best Presentation and Best Project

Mid-Term Presentation

- 10 mins each team
- 12 slides (title + references + 10)
- Suggested Outline
  - Topic and motivation (1)
  - Technical problem formulation (1)
  - Demo or sample results of existing techniques (1)
  - Technical review of background and state of the art (4)
  - Proposed ideas and work plans (2)
  - Evaluation: dataset and performance metrics (1)

Mid-term Presentation Schedule

Yongxu Zhang	A sketch-based search engine for 3D models	26-Mar
Yun Zhang	A sketch-based search engine for 3D models	26-Mar
Hong-Ming Chen	A statistical approach to speed up re-ranking	26-Mar
Hongzhi Li	Hierarchical Feature Representation	26-Mar
Brendan Jou	Hierarchical Feature Representation	26-Mar
Rui Cao	Library Stack Visual Search on Mobile Device	26-Mar
Mingyang	realize the SIFT algorithm by GPU	26-Mar
Li Jiao	Seam Carving in Image Resizing	26-Mar
Pengfei Weng	Seam Carving in Image Resizing	26-Mar
Zhiying Guo	Spatial-temporal Feature-Based Video Retrieval	26-Mar
Chi Hu	Spatial-temporal Feature-Based Video Retrieval	26-Mar
Mojun Zhu	Stereo Image Search using HTC EVO 3D phone	26-Mar
Wenyu Li	Traffic Sign Auto-Detection and Alert System	26-Mar
Mo Lin	Traffic Sign Auto-Detection and Alert System	26-Mar

Mid-term Project Presentation

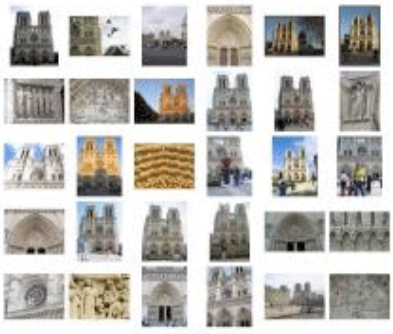
Nicholas Hwang	fast video (copy) detection	2-Apr
Rachel Kurtz	fast video (copy) detection	2-Apr
Chun-Chao Wang	Visual Search System on Fish Species	2-Apr
Fan Wei	Visual Search System on Fish Species	2-Apr
Lei Wang	Application for Image Searching (search reranking and user logs)	2-Apr
Mingkai Xu	Application for Image Searching (search reranking and user logs)	2-Apr
Nianwen Song	Chinese dictionary and menu translation	2-Apr
Teng Gu	Chinese dictionary and menu translation	2-Apr
Avijit Singh	Image search based landmark recognition within Columbia Campus	2-Apr
Vipul Singh	Image search based landmark recognition within Columbia Campus	2-Apr
Wenuwat Kaewchaijaroenkit	Local Energy based Shape Histogram (LESH)	2-Apr
Junyuan Gao	Movie information searching based on the image feature extracted from the posters	2-Apr
Di Yan	Movie information searching based on the image feature extracted from the posters	2-Apr
Dan Zhao	Painting Recognition	2-Apr
Hao Liu	Painting Recognition	2-Apr
Yi Li	Real time hand gesture recognition based on 2-D image	2-Apr
Yilin Li	Real time hand gesture recognition based on 2-D image	2-Apr
Yuandi Jin	Region Search for Object Detection	2-Apr
Chun Wang	Region Search for Object Detection	2-Apr
Wanying Lu	Sharp Eyes – Mobile Navigation Application for Visual Disable Users	2-Apr
Yuan Du	Sharp Eyes – Mobile Navigation Application for Visual Disable Users	2-Apr
Zhou Sha	Sharp Eyes – Mobile Navigation Application for Visual Disable Users	2-Apr

Mid-term Project Presentation

Haosen Wang	Car Detection	9-Apr
Jian Liu	Car Detection	9-Apr
Fei Teng	face recognition	9-Apr
Yi Dai	face recognition	9-Apr
Jichuan Jiang	Iphone based mobile location search in Columbia	9-Apr
Mao Lei	Iphone based mobile location search in Columbia	9-Apr
Chenhao Lin	Logo Recognition Application on iPhone	9-Apr
Jiayu Xiao	Logo Recognition Application on iPhone	9-Apr
Mengtian Fan	Logo Recognition Application on iPhone	9-Apr
Chenglong Wang	Multiple Object Segmentation	9-Apr
Zirui Zhao	NYC Building Guide	9-Apr
Feng Zhou	NYC Building Guide	9-Apr
Ximeng Li	Panorama Recovery Based on the Shape and Color Features	9-Apr
Bo Feng	Panorama Recovery Based on the Shape and Color Features	9-Apr
Pan Deng	Panorama Recovery Based on the Shape and Color Features	9-Apr
Yaozhong Song	Recommendation Paintings	9-Apr
Shan Fu	Recommendation Paintings	9-Apr
Rui Wang	search by sketch	9-Apr

Photo synth

Noah Snavely, Steven M. Seitz, Richard Szeliski,
"Photo tourism: Exploring photo collections in 3D," SIGGRAPH 2006



http://photosynth.net/

Photo Tourism overview

Input photographs

Scene reconstruction

Relative camera positions and orientations

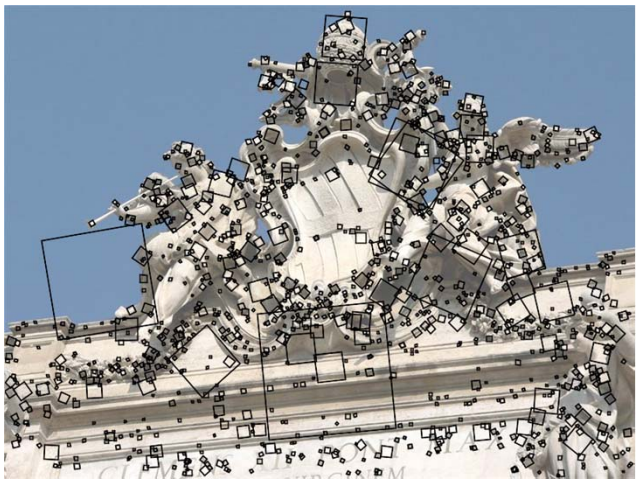
Point cloud

Sparse correspondence

Photo Explorer

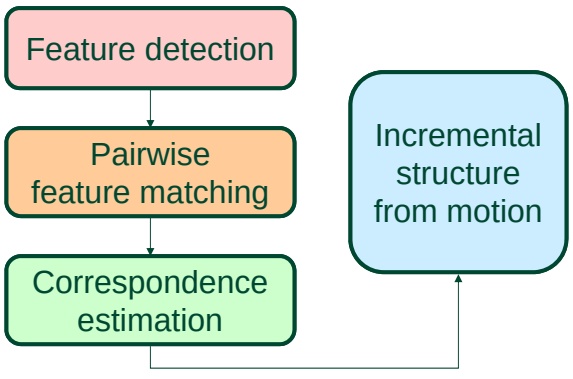
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



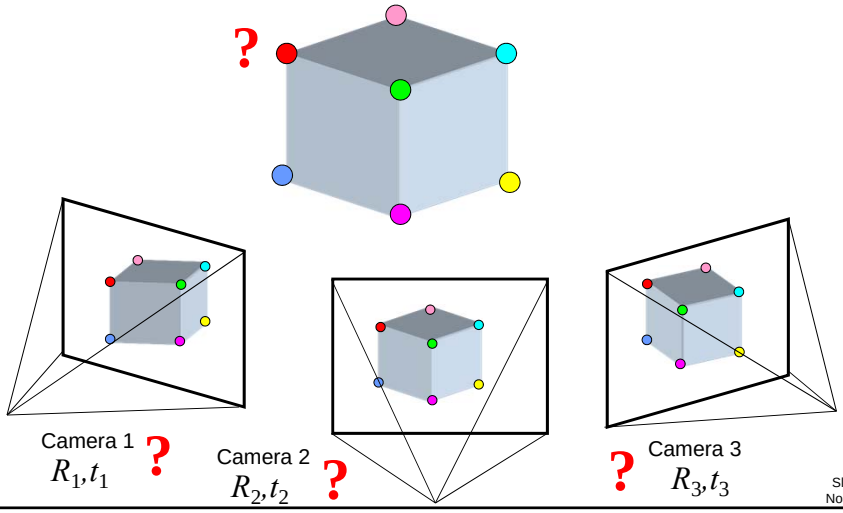
Scene reconstruction

- Automatically estimate
  - position, orientation, and focal length of cameras
  - 3D positions of feature points



Structure from motion

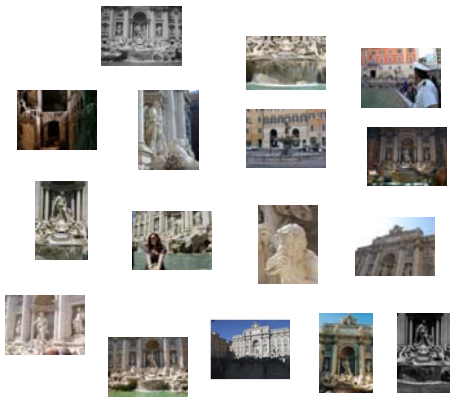
- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Slide credit: Noah Snavely

Feature detection

Detect features using SIFT [Lowe, IJCV 2004]

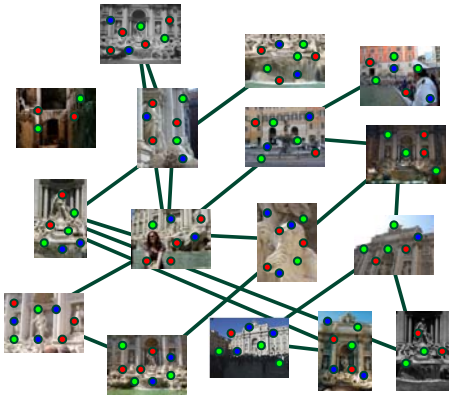


Feature detection

Detect features using SIFT [Lowe, IJCV 2004]

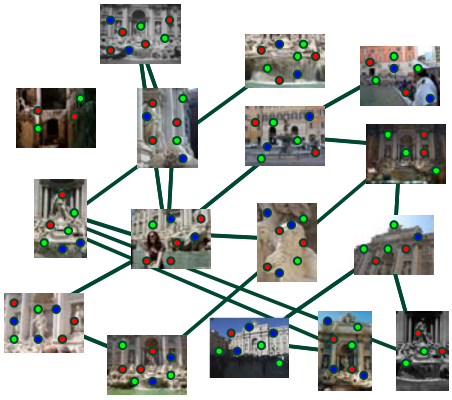
Feature matching

Match features between each pair of images

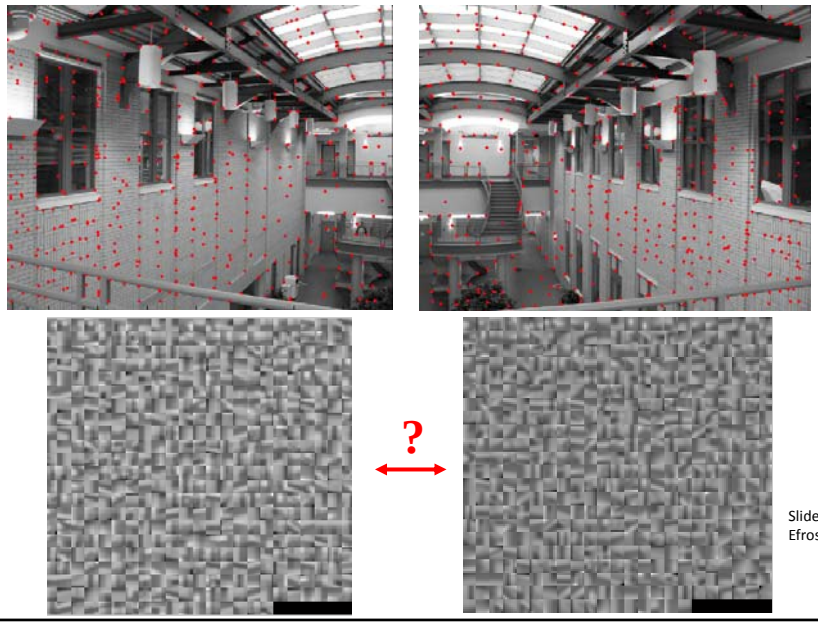


Feature matching

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



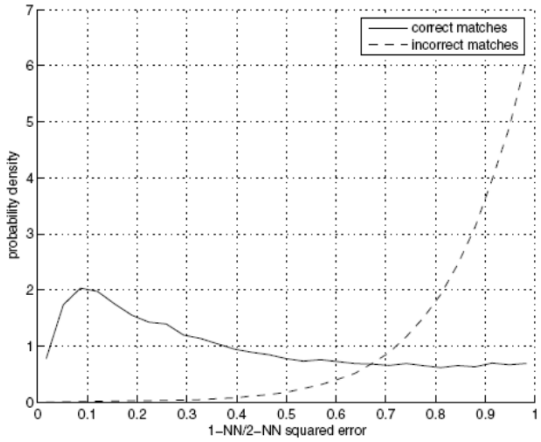
Review: Feature matching



Slide of A. Efros

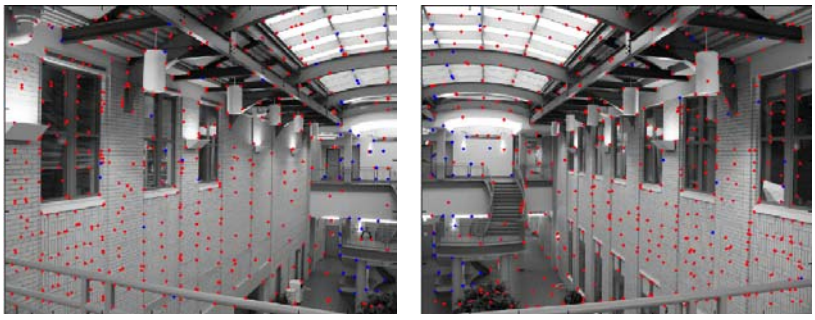
Feature-space outlier rejection [Lowe, 1999]:

- 1-NN: SSD of the closest match
- 2-NN: SSD of the second-closest match
- Look at how much the best match (1-NN) is than the 2nd best match (2-NN), e.g. 1-NN/2-NN



Slide of A. Efros

Feature-space outliner rejection



Can we now compute H from the blue points?

- No! Still too many outliers…
- What can we do?

Slide of A. Efros

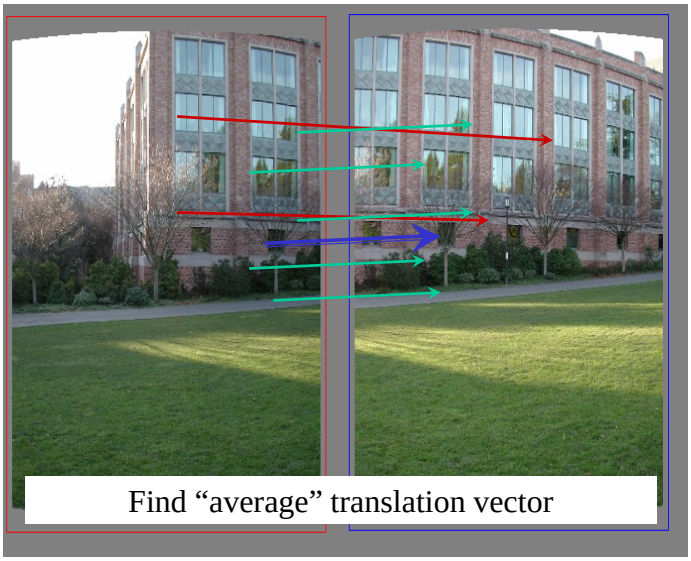
RANSAC for estimating homography

RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

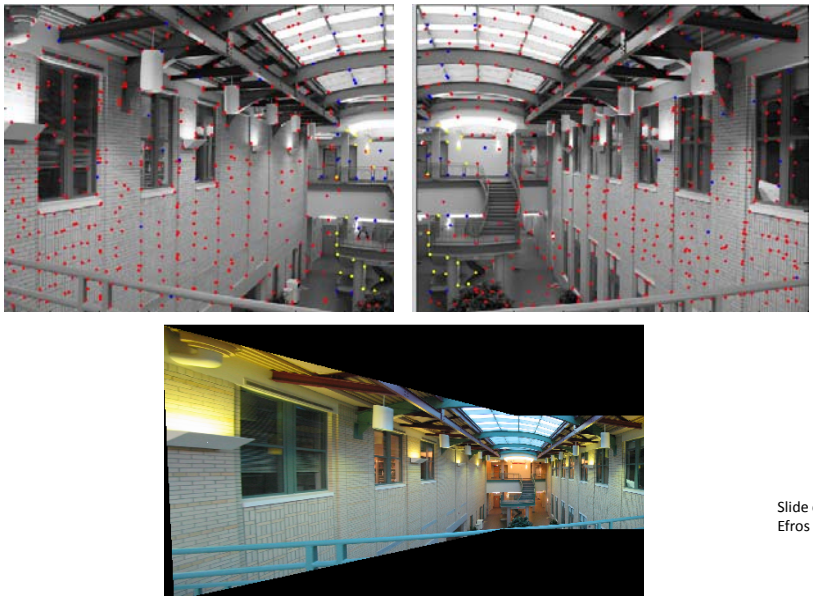
Slide of A. Efros

Least squares fit



Slide of A. Efros

RANSAC



Slide of A. Efros

Correspondence estimation

- Link up pairwise matches to form connected components of matches across several images

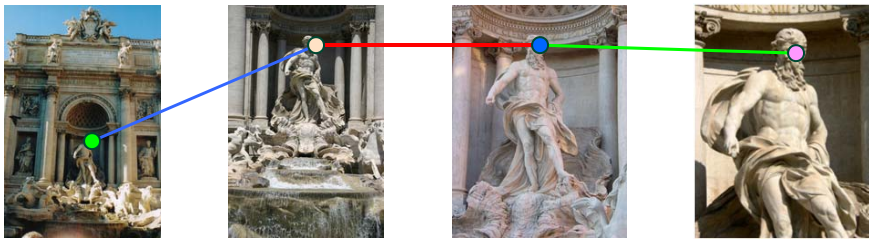
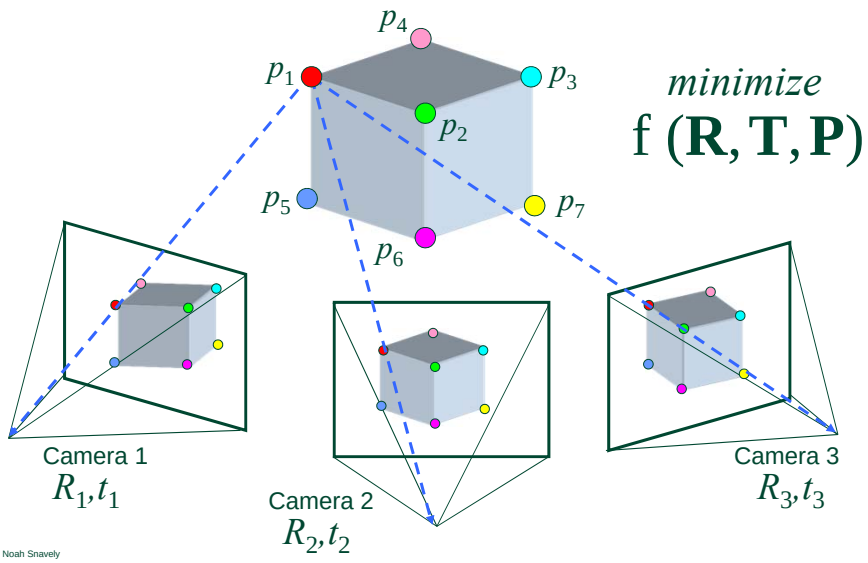


Image 1 Image 2 Image 3 Image 4

Structure from motion

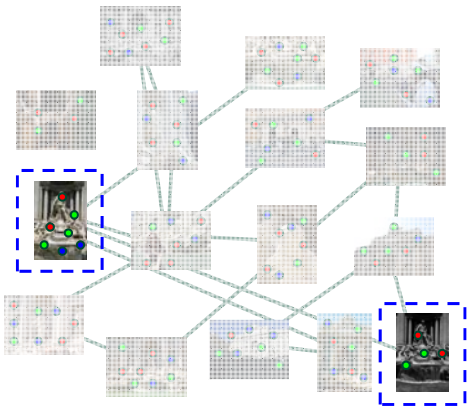


minimize
$\mathrm{f}\,(\mathbf{R}, \mathbf{T}, \mathbf{P})$

Incremental structure from motion

Incremental structure from motion

Incremental structure from motion



Reconstruction performance

- For photo sets from the Internet, 20% to 75% of the photos were registered
- Most unregistered photos belonged to different connected components

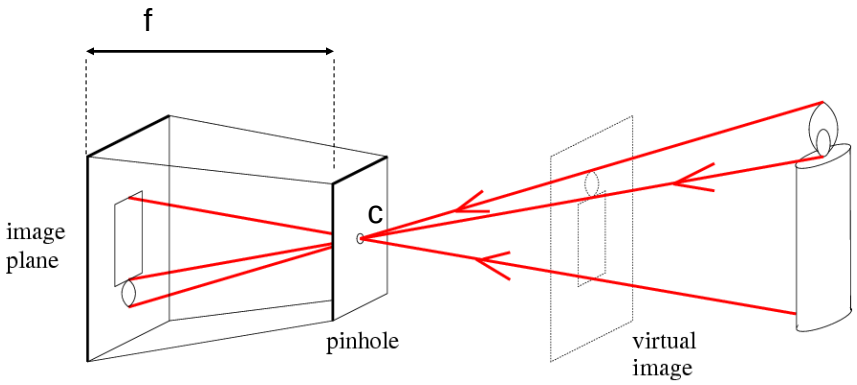
- Running time: < 1 hour for 80 photos

  > 1 week for 2600 photo

Review

- Structure from Motion

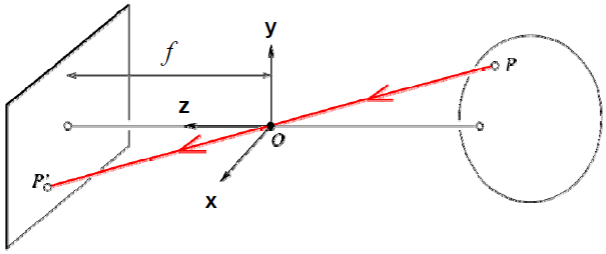
Pinhole camera



f = focal length
c = center of the camera

Figure from Forsyth

Modeling projection



Projection equations

- Compute intersection with image plane of ray from ***P*** = (x,y,z) to ***O***
- Derived using similar triangles

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z}, f)$$

- We get the projection by throwing out the last coordinate:

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

Source: J. Ponce, S. Seitz

Homogeneous coordinates

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

Is this a linear transformation?

- no—division by z is nonlinear

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

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

homogeneous scene coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Slide by Steve Seitz

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

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

divide by the third coordinate

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

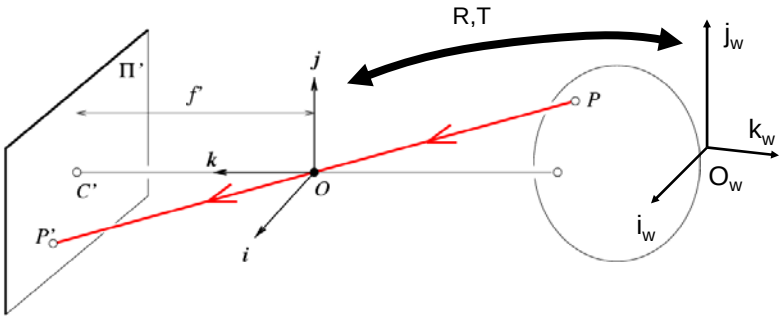
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

divide by the third coordinate

In practice: lots of coordinate transformations…

$$\begin{bmatrix} \text{2D point (3x1)} \end{bmatrix} = \begin{bmatrix} \text{Camera to pixel coord. trans. matrix (3x3)} \end{bmatrix} \begin{bmatrix} \text{Perspective projection matrix (3x4)} \end{bmatrix} \begin{bmatrix} \text{World to camera coord. trans. matrix (4x4)} \end{bmatrix} \begin{bmatrix} \text{3D point (4x1)} \end{bmatrix}$$

Projection matrix

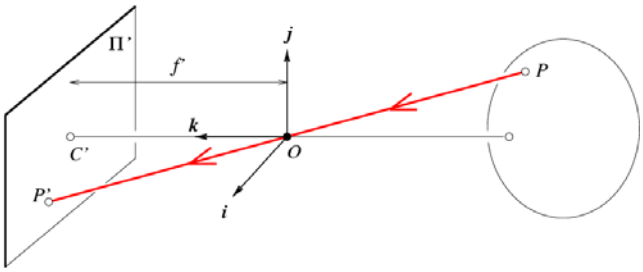


$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
X: World Coordinates: (X,Y,Z,1)

Slide Credit: Saverese

Projection matrix



Intrinsic Assumptions
- Unit aspect ratio
- Optical center at (0,0)
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \overbrace{\begin{bmatrix}f & 0 & 0\\0 & f & 0\\0 & 0 & 1\end{bmatrix}}^{\mathbf{K}}\begin{bmatrix}0\\0\\0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Slide Credit: Saverese

Remove assumption: known optical center

Intrinsic Assumptions
- Unit aspect ratio
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f & 0 & u_0\\0 & f & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Remove assumption: square pixels

Intrinsic Assumptions
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & 0 & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Remove assumption: non-skewed pixels

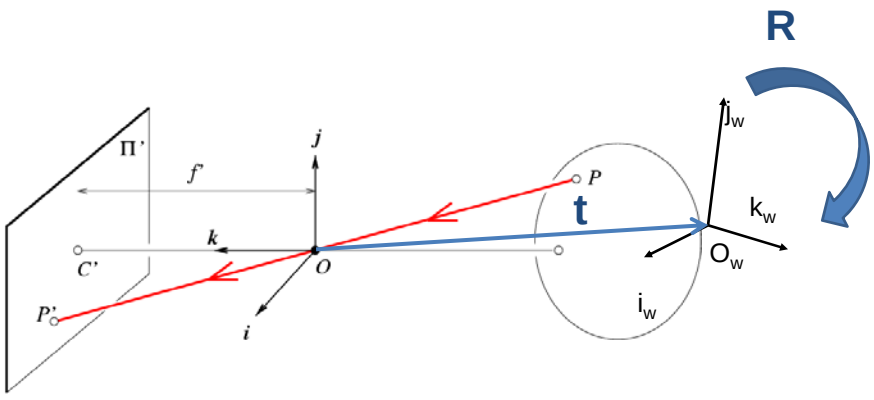
Intrinsic Assumptions

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Note: different books use different notation for parameters

Oriented and Translated Camera



Allow camera translation

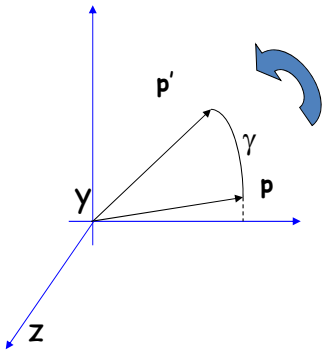
Intrinsic Assumptions Extrinsic Assumptions
- No rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & 0 & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_x\\0 & 1 & 0 & t_y\\0 & 0 & 1 & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Slide Credit: Saverese

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Allow camera rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

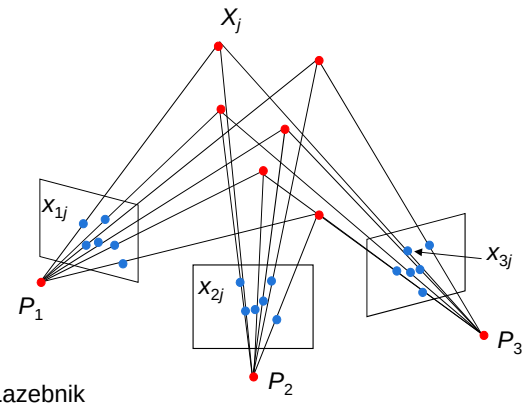
Degrees of freedom

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \underbrace{\begin{bmatrix}\alpha & s & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}}_{5}\underbrace{\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}}_{6}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}$$

Projective structure from motion

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



Slides from Lana Lazebnik

Projective structure from motion

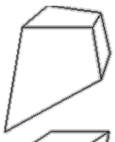
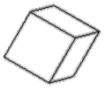
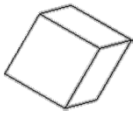
- Given: *m* images of *n* fixed 3D points
  - $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- Problem: estimate *m* projection matrices $\mathbf{P}_j$ and *n* 3D points $\mathbf{X}_j$ from the *mn* corresponding points $\mathbf{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation **Q**:
  - **X → QX, P → PQ^{-1}**
- We can solve for structure and motion when
  - $2mn >= 11m + 3n - 15$
- For two cameras, at least 7 points are needed

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

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

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 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

Slide of James Hays

Photo Tourism overview

Input photographs

Scene reconstruction

- Navigation
- Rendering
- Annotations

More slides from Photo Tourism

http://phototour.cs.washington.edu/Photo_Tourism.ppt