Geometric Transformations and Image Warping: Mosaicing

CS 6640

Ross Whitaker, Guido Gerig

SCI Institute, School of Computing

University of Utah

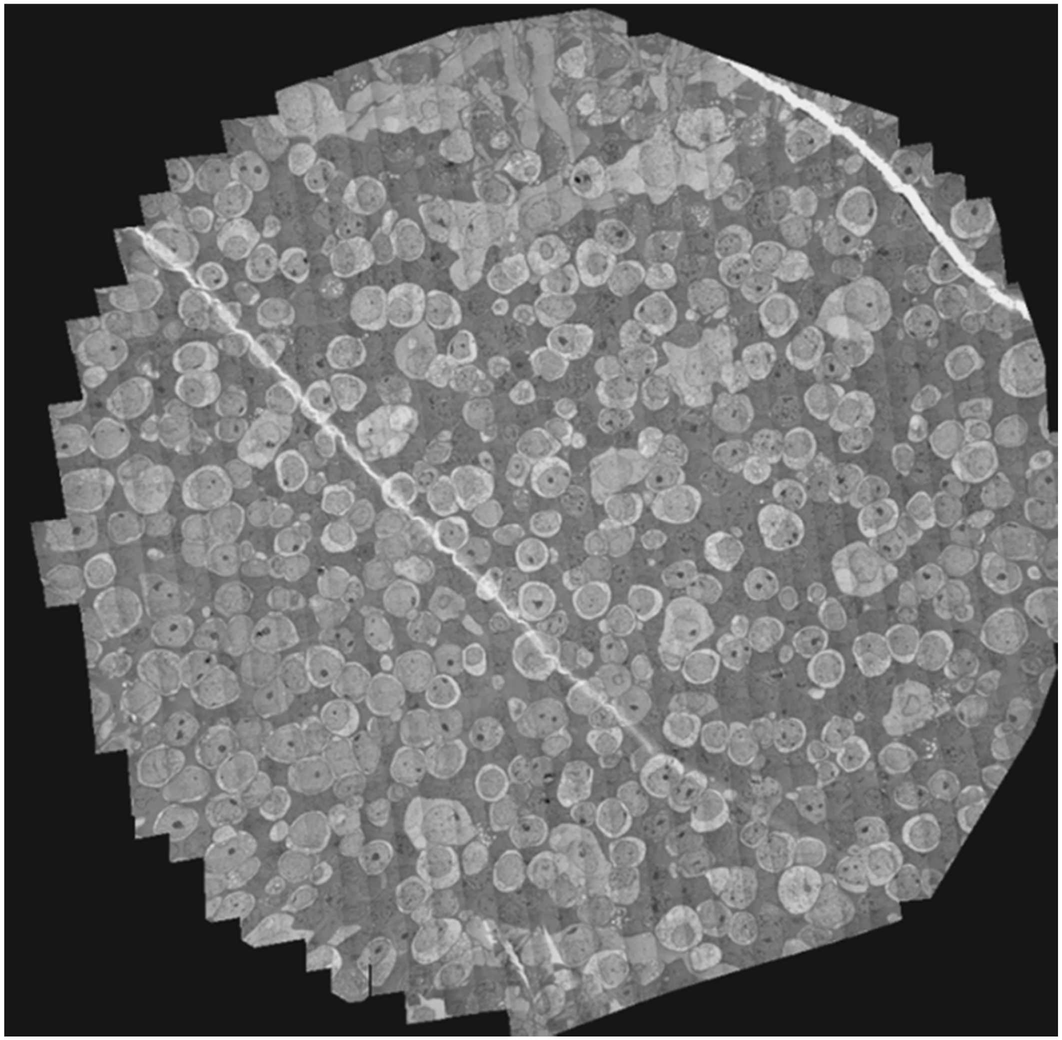
(with slides from: Jinxiang Chai, TAMU)

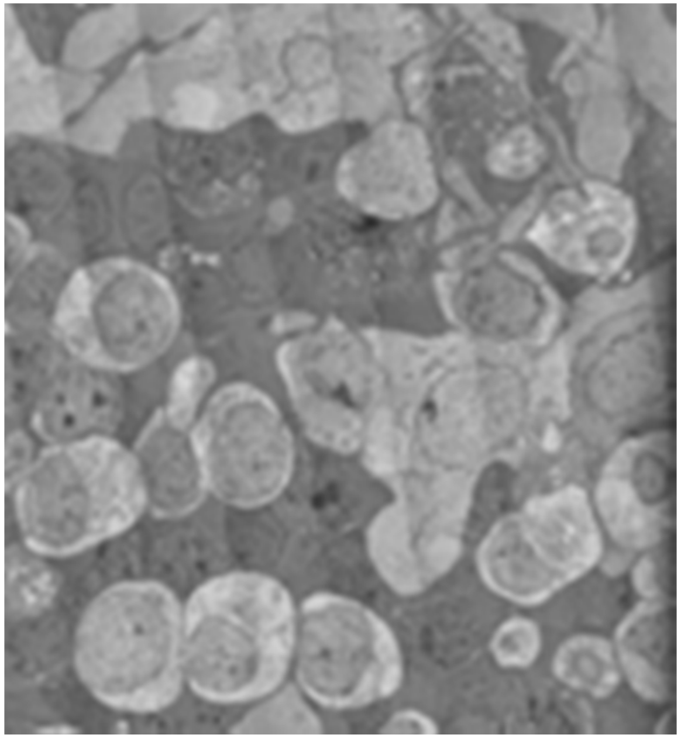
faculty.cs.tamu.edu/jchai/cpsc641_spring10/lectures/lecture8.ppt

Applications

Saint-Guénolé Church of Batz-sur-Mer Equirectangular 360º by Vincent Montibus

Microscopy (Morane Eye Inst, UofU, T. Tasdizen et al.)





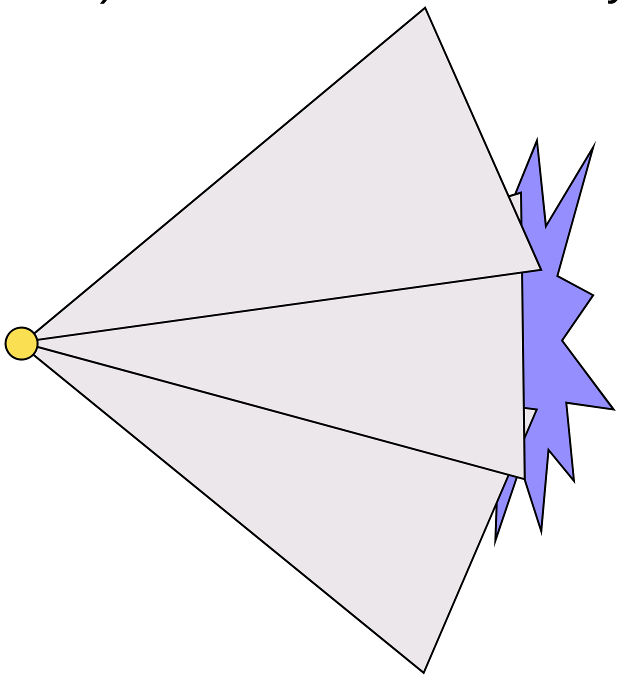

SONY
NEX-5


SONY
NEX-5

Special Cases

- Nothing new in the scene is uncovered in one view vs another
  - No ray from the camera gets behind another

1) Pure rotations–arbitrary scene



2) Arbitrary views of planar surfaces

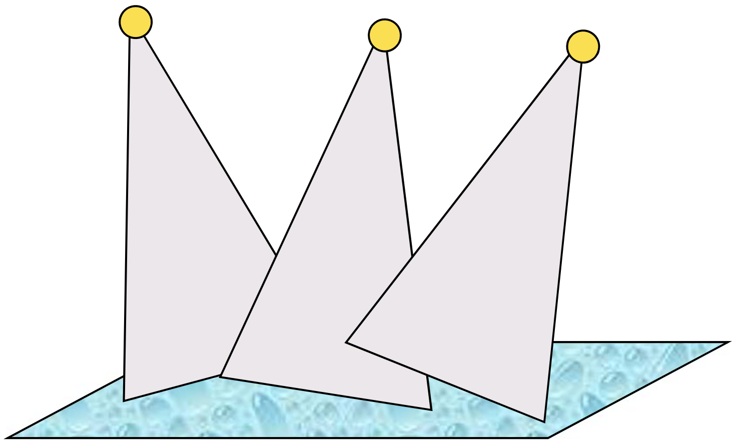


Image Homologies

- Images taken under cases 1,2 are perspectively equivalent to within a linear transformation
  - Projective relationships – equivalence is

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \equiv \begin{pmatrix} d \\ e \\ f \end{pmatrix} \Longleftrightarrow \begin{pmatrix} a/c \\ b/c \\ 1 \end{pmatrix} = \begin{pmatrix} d/f \\ e/f \\ 1 \end{pmatrix}$$

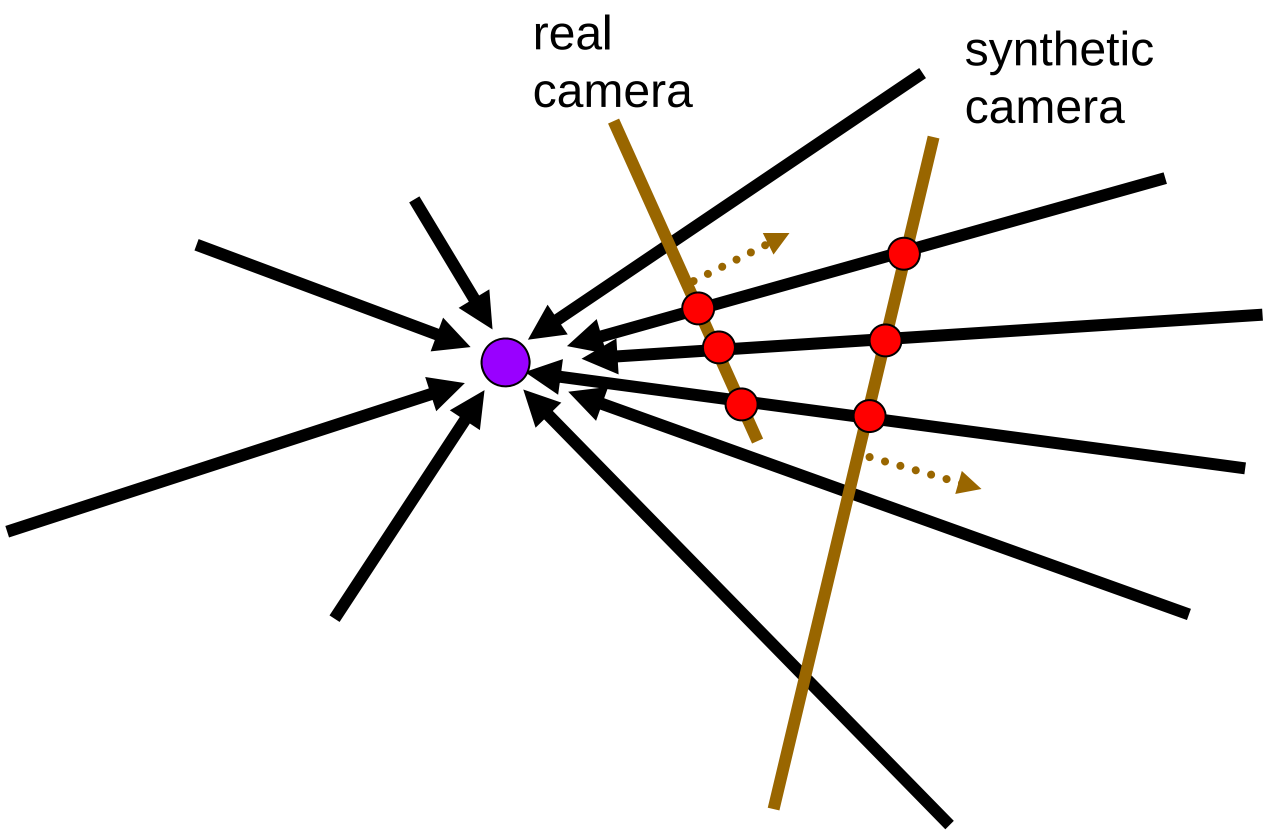
Mosaic Procedure

Basic Procedure

- Take a sequence of images from the same position
  - Rotate the camera about its optical center
- Compute transformation between second image and first
- Transform the second image to overlap with the first
- Blend the two together to create a mosaic
- If there are more images, repeat

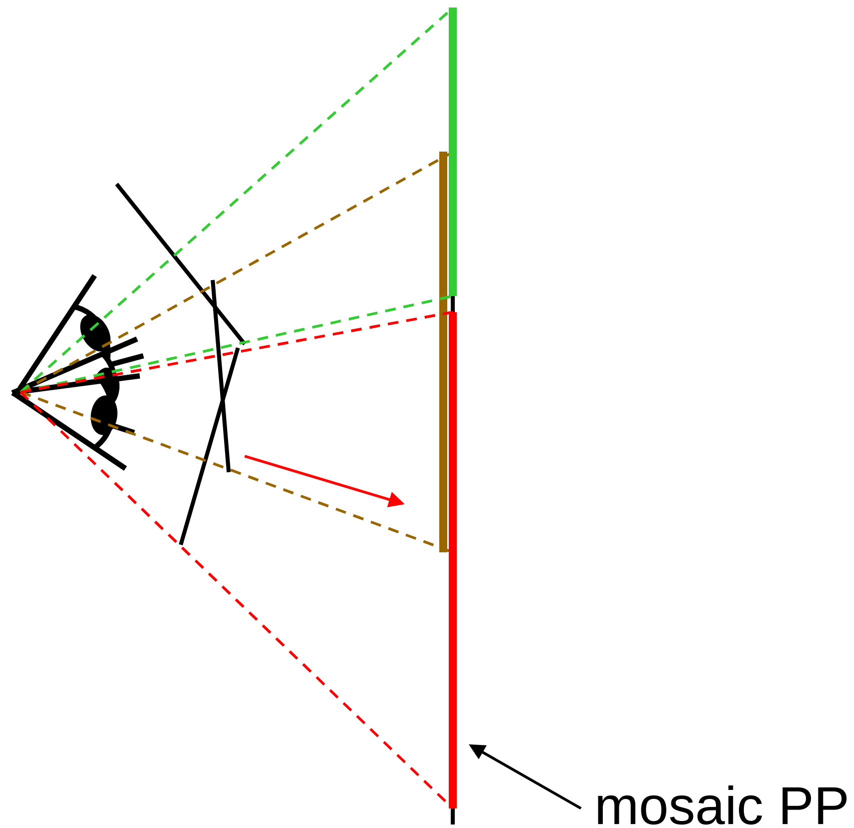
Image Mosaic

Is a pencil of rays contains all views



Can generate any synthetic camera view
as long as it has **the same center of projection**!

Image Re-projection



The mosaic has a natural interpretation in 3D

- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*

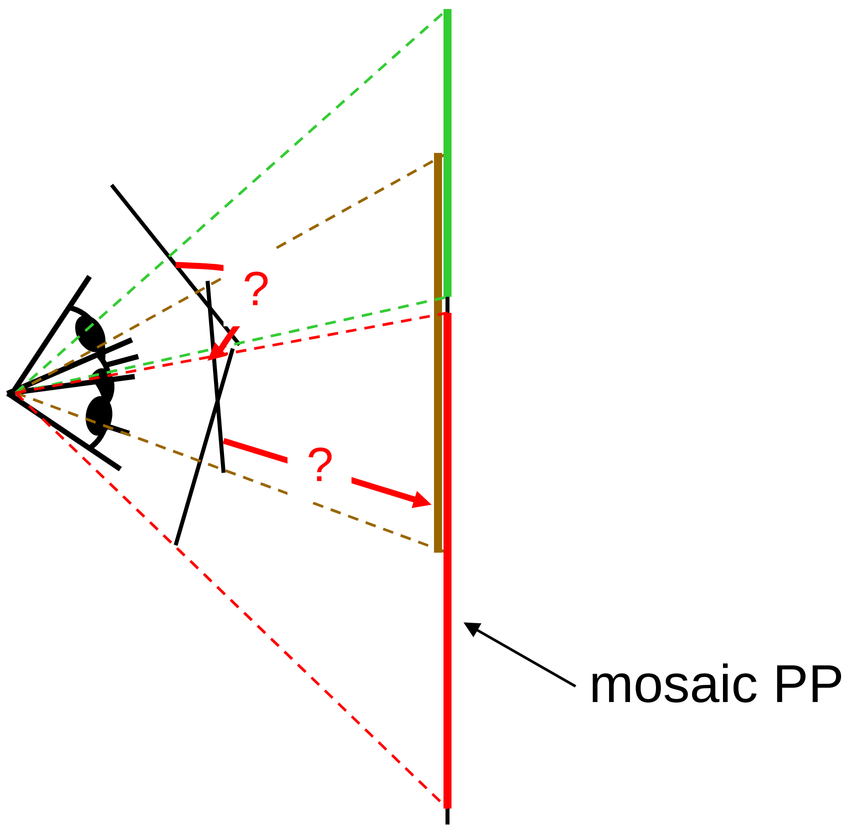
Issues in Image Mosaic

How to relate two images from the same camera center?

- image registration

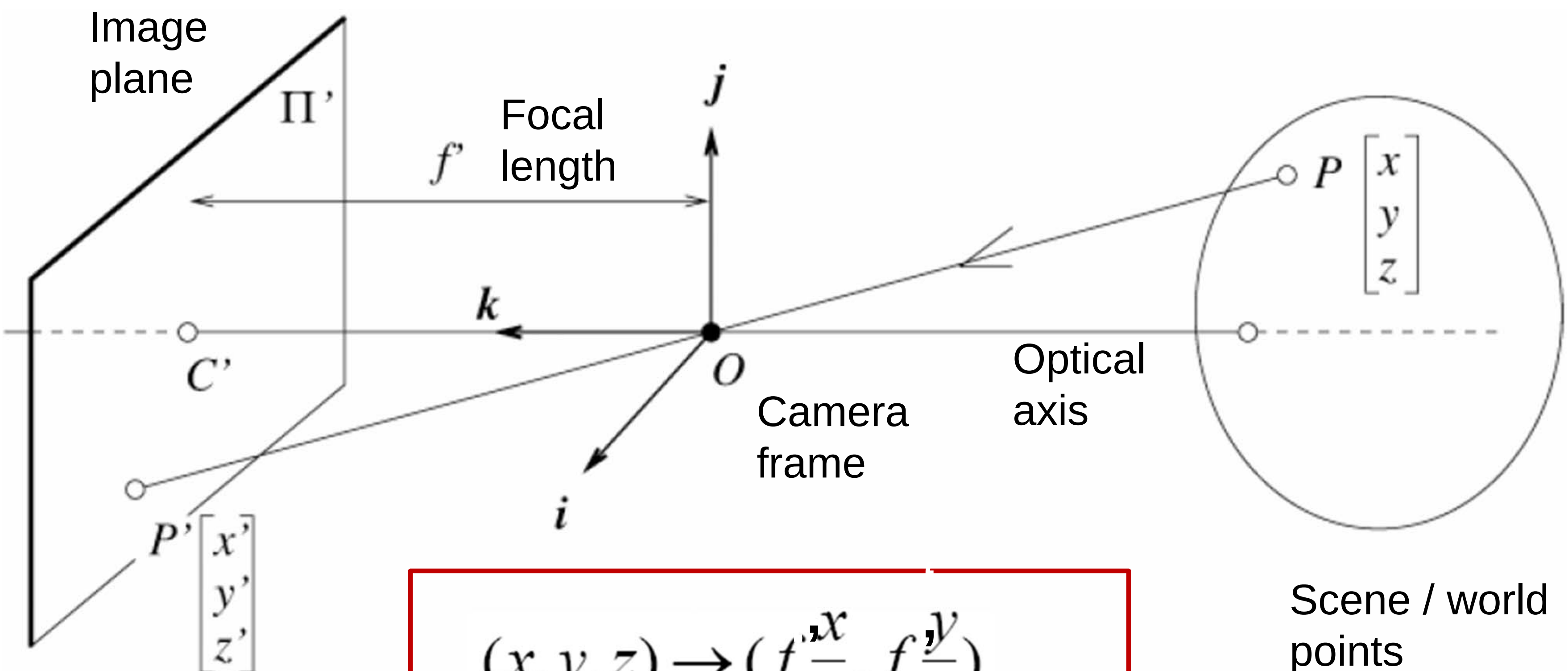
How to re-project images to a common plane?

- image warping



Perspective projection equations

- 3d world mapped to 2d projection in image plane



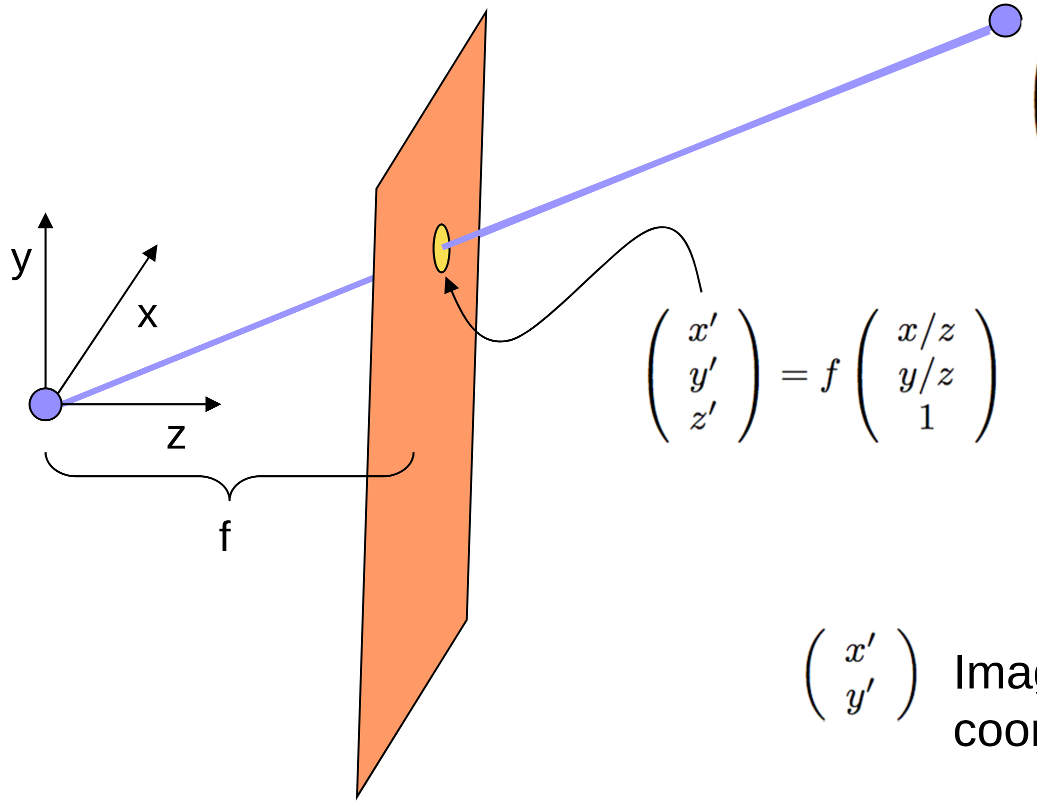
$$(x, y, z) \rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$$

Scene point → Image coordinates

Forsyth and Ponce

3D Perspective and Projection

- Camera model



$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = f \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix}$$

$\begin{pmatrix} x' \\ y' \end{pmatrix}$ Image coordinates

Homogeneous coordinates

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:

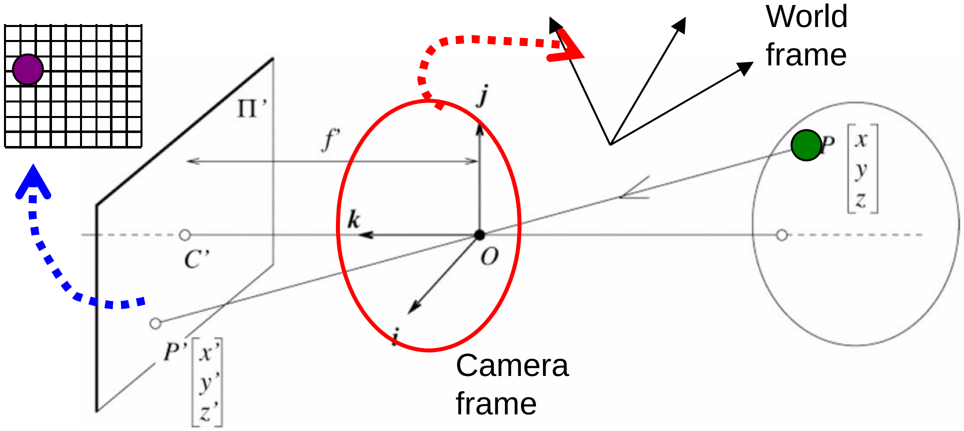
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f' \end{bmatrix} \Rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$$

divide by the third coordinate to convert back to non-homogeneous coordinates

Complete mapping from world points to image pixel positions?

Slide by Steve Seitz

Perspective projection

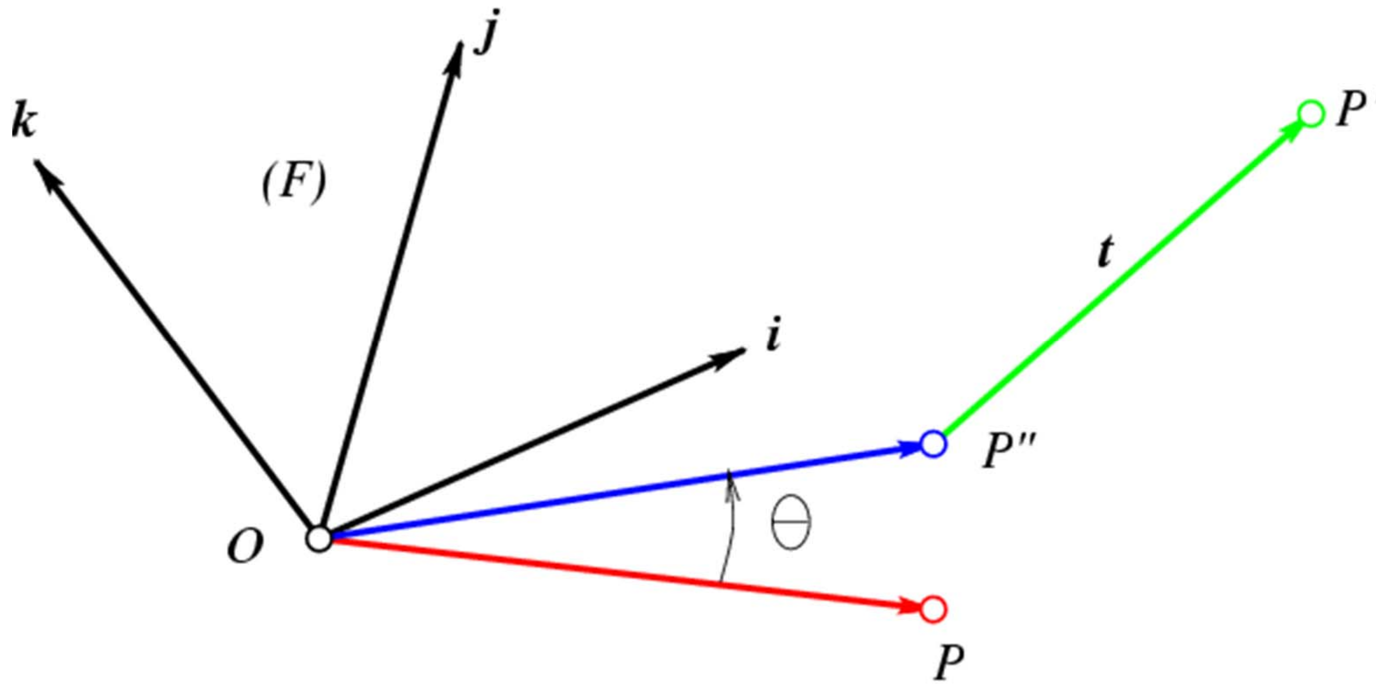


Extrinsic:
Camera frame ←→World frame

3D point (4x1)

K. Grauman

Rigid Transformations as Mappings



$$^{F}P' = \mathcal{R}\,^{F}P + \boldsymbol{t} \iff \begin{pmatrix} ^{F}P' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathcal{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \begin{pmatrix} ^{F}P \\ 1 \end{pmatrix}$$

Extrinsic parameters: translation and rotation of camera frame

$${}^{C}\vec{p} = {}^{C}_{W}R \; {}^{W}\vec{p} + {}^{C}_{W}\vec{t}$$

Non-homogeneous coordinates

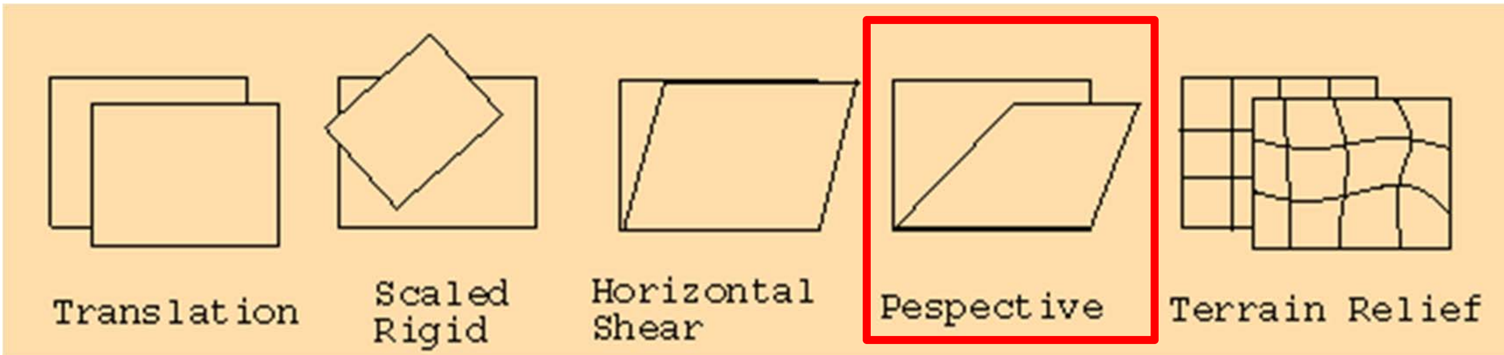
$$\begin{pmatrix} {}^{C}\vec{p} \\ \\ \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} - & - & - \\ - & {}^{C}_{W}R & - \\ - & - & - \end{bmatrix} & \begin{bmatrix} | \\ {}^{C}_{W}\vec{t} \\ | \end{bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{W}\vec{p} \\ \\ \end{pmatrix}$$

Homogeneous coordinates

Remember discussion of transformations: Rotation and Translation can be Combined into a matrix transformation via homogeneous coordinates!

W. Freeman

Transformations



$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

affine

$$\begin{pmatrix} X' \\ Y' \\ W \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

New image coordinates can be found as $x' = X'/W$, $y' = Y'/W$

x', y': homographies

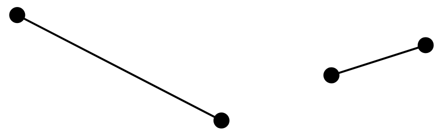
(Geom.) A relation between two figures, such that to any point of the one corresponds one and but one point in the other, and vise versa.

Materials

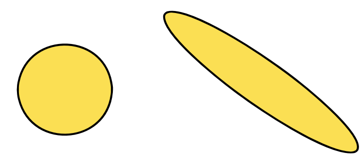
- *Excellent material to derive homography matrix:*
  - *www.cs.toronto.edu/~jepson/csc2503/**tutorial2.pdf***
  - *www.cs.toronto.edu/pub/jepson/teaching/vision/2503/**tutorial2.pdf***

Perspective Projection Properties

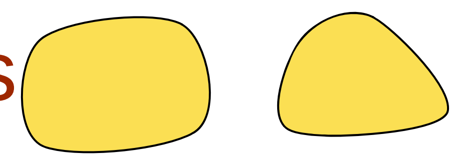
- Lines to lines (linear)



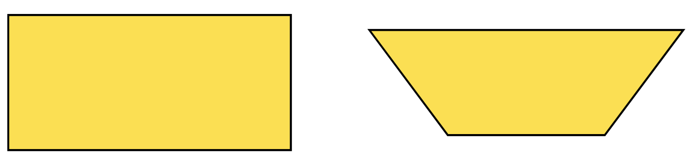
- Conic sections to conic sections



- Convex shapes to convex shapes



- Foreshortening



Transforming Images To Make Mosaics

Linear transformation with matrix P

$$\bar{x}^* = P\bar{x} \qquad P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{pmatrix} \qquad \begin{aligned} x^* &= p_{11}x + p_{12}y + p_{13} \\ y^* &= p_{21}x + p_{22}y + p_{23} \\ z^* &= p_{31}x + p_{32}y + 1 \end{aligned}$$

Perspective equivalence

$$\begin{aligned} x' &= \tfrac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + 1} \\ y' &= \tfrac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + 1} \end{aligned}$$

Multiply by denominator and reorganize terms

$$\begin{aligned} p_{31}xx' + p_{32}yx' - p_{11}x - p_{12}y - p_{13} &= -x' \\ p_{31}xy' + p_{32}yy' - p_{21}x - p_{22}y - p_{23} &= -y' \end{aligned}$$

Linear system, solve for P

$$\begin{pmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1x'_1 & y_1x'_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2x'_2 & y_2x'_2 \\ & & & \vdots & & & & \\ -x_N & -y_N & -1 & 0 & 0 & 0 & x_Nx'_N & y_Nx'_2 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1y'_1 & y_1y'_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2y'_2 & y_2y'_2 \\ & & & \vdots & & & & \\ 0 & 0 & 0 & -x_N & -y_N & -1 & x_Ny'_N & y_Ny'_N \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{23} \\ p_{23} \\ p_{31} \\ p_{32} \end{pmatrix} = \begin{pmatrix} -x'_1 \\ -x'_2 \\ \vdots \\ -x'_N \\ -y'_1 \\ -y'_2 \\ \vdots \\ -y'_N \end{pmatrix}$$

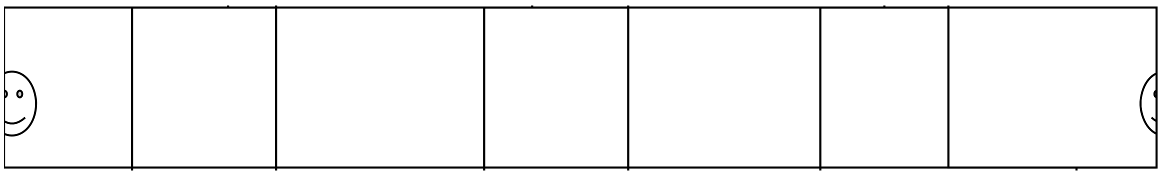
Transforming Images To Make Mosaics

Linear system, solve for P

$$\begin{pmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1x'_1 & y_1x'_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2x'_2 & y_2x'_2 \\ & & & \vdots & & & & \\ -x_N & -y_N & -1 & 0 & 0 & 0 & x_Nx'_N & y_Nx'_2 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1y'_1 & y_1y'_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2y'_2 & y_2y'_2 \\ & & & \vdots & & & & \\ 0 & 0 & 0 & -x_N & -y_N & -1 & x_Ny'_N & y_Ny'_N \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{23} \\ p_{23} \\ p_{31} \\ p_{32} \end{pmatrix} = \begin{pmatrix} -x'_1 \\ -x'_2 \\ \vdots \\ -x'_N \\ -y'_1 \\ -y'_2 \\ \vdots \\ -y'_N \end{pmatrix}$$

- Choose sets of corresponding landmarks in two images A and B: x_i and x'_i
- Calculate matrix P
- Transform image A to image B

Image Stitching



Stitch pairs together, blend, then crop

Image Stitching

A big image stitched from 5 small images

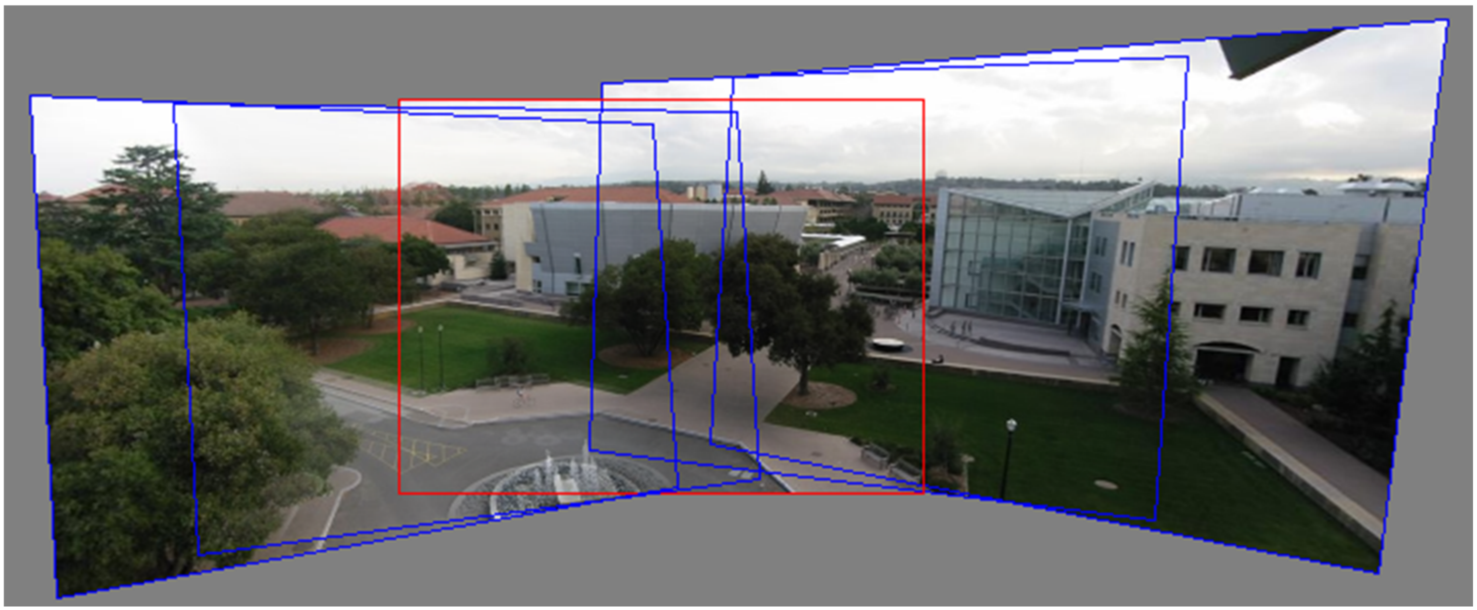


Image Mosaicing

4 Correspondences

5 Correspondences

6 Correspondences

Mosaicing Issues

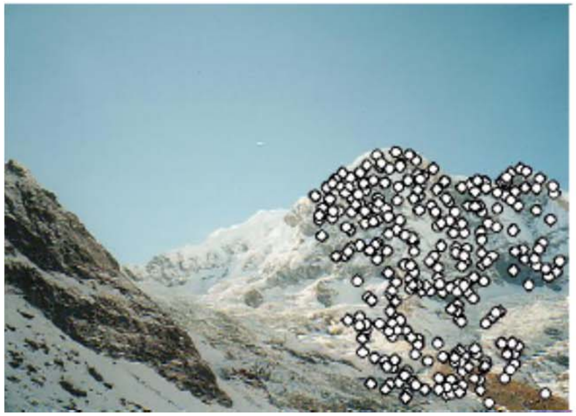
- Need a canvas (adjust coordinates/origin)
- Blending at edges of images (avoid sharp transitions)
- Adjusting brightnesses
- Cascading transformations

Recognizing panoramas

- A fully automatic 2D image stitcher system

Recognizing panoramas

- A fully automatic 2D image stitcher system

- Image matching with SIFT features

- For every image, find the M best images with RANSAC

- Form a graph and find connected component in the graph

- Stitching and blending.