Lecture 10

Homographies, Mosaics and Panoramas

Lots of slides stolen from Bill Freeman & Frédo Durand, who stole them from Alyosha Efros, who stole them from Steve Seitz and Rick Szeliski

Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°

Slide from Brown & Lowe

Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°

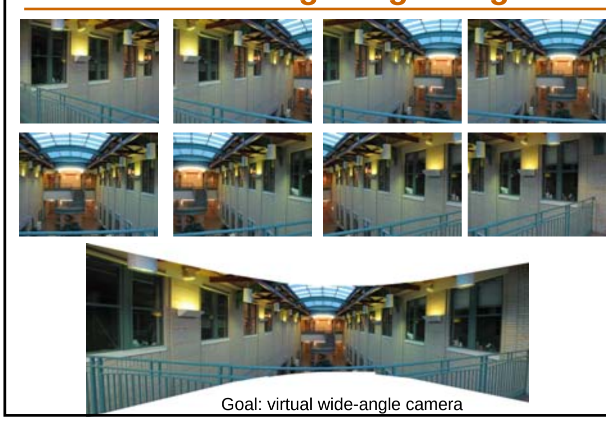
Slide from Brown & Lowe

Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Slide from Brown & Lowe

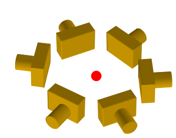
Mosaics: stiching images together



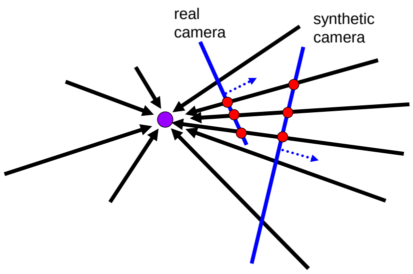
Goal: virtual wide-angle camera

Single View Point Model

- Camera rotates around a single optical center



A pencil of rays contains all views



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

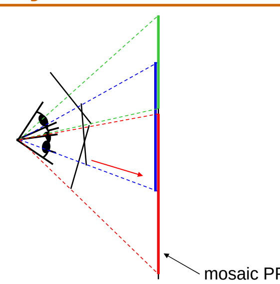
How to do it?

- 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
- What about the 3D geometry of the scene?
  - Why aren't we using it?

Aligning images: translation

Translations are not enough to align the images

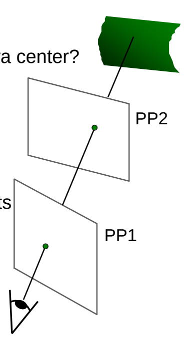
Image reprojection



- 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*

Image reprojection

- Basic question
  - How to relate 2 images from same camera center?
    - how to map a pixel from PP1 to PP2
- Answer
  - Cast a ray through each pixel in PP1
  - Draw the pixel where that ray intersects PP2

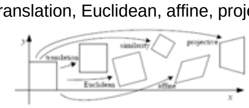


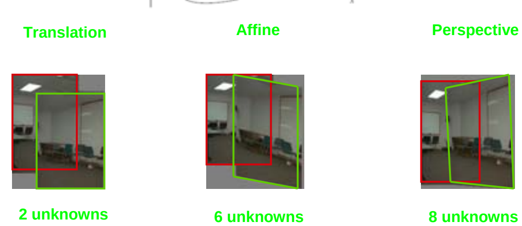
But don't we need to know the geometry of the two planes in respect to the eye?

Observation:
Rather than thinking of this as a 3D reprojection, think of it as a 2D **image warp** from one image to another

Back to Image Warping

Which t-form is the right one for warping PP1 into PP2?
e.g. translation, Euclidean, affine, projective

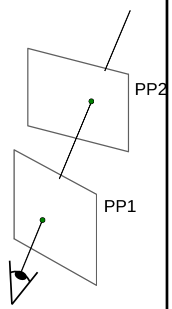




Homography

- Projective – mapping between any two PPs with the same center of projection
  - rectangle should map to arbitrary quadrilateral
  - parallel lines aren't
  - but must preserve straight lines
  - same as: project, rotate, reproject
- called Homography

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$



To apply a homography **H**

- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

Image warping with homographies

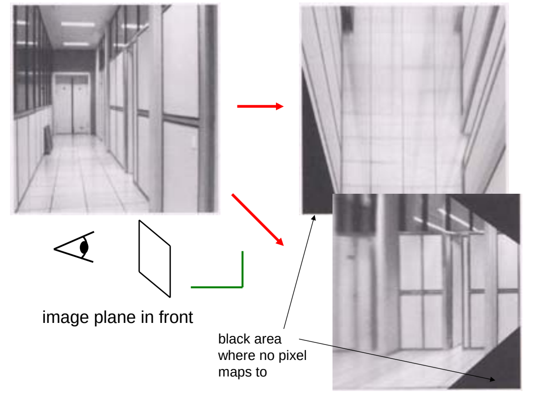
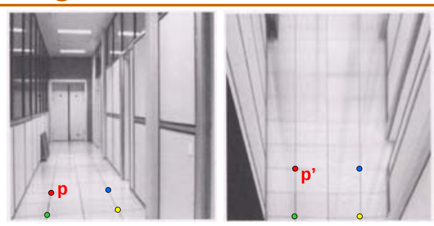


Image rectification



To unwarp (rectify) an image

- Find the homography **H** given a set of **p** and **p'** pairs
- How many correspondences are needed?
- Tricky to write H analytically, but we can solve for it!
  - Find such H that "best" transforms points p into p'
  - Use least-squares!

Least Squares Example

- Say we have a set of data points (p_1, p_1'), (p_2, p_2'), (p_3, p_3'), etc. (e.g. person's height vs. weight)
- We want a nice compact formula (line) to predict p' from p:

$$pa + b = p' \qquad \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} p_1' \\ p_2' \end{bmatrix}$$

- We want to find a and b
- How many pairs of points are needed?

rewrite

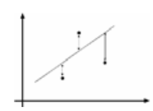
$$p_1 a + b = p_1' \qquad p_2 a + b = p_2'$$

$$\underbrace{\begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{Ah}} = \underbrace{\begin{bmatrix} p_1' \\ p_2' \end{bmatrix}}_{\mathbf{b}}$$

- What if the data is noisy?

$$\begin{bmatrix} p_1 & 1 \\ p_2 & 1 \\ p_3 & 1 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \end{bmatrix} \qquad \min \|\mathbf{Ah} - \mathbf{b}\|^2$$

overconstrained



Solving for homographies

p' = Hp

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Can set scale factor i=1. So, there are 8 unkowns.
- Set up a system of linear equations:

Ah = b

where vector of unknowns **h** = $[a,b,c,d,e,f,g,h]^T$

- Note: we do not know w, but we can compute it from x & y

$$w = gx + hy + 1$$

- The equations are linear in the unknown

Solving for homographies

p' = Hp

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Can set scale factor i=1. So, there are 8 unkowns.
- Set up a system of linear equations:

Ah = b

where vector of unknowns **h** = $[a,b,c,d,e,f,g,h]^T$

- Need at least 8 eqs, but the more the better…
- Solve for **h**. If overconstrained, solve using least-squares:

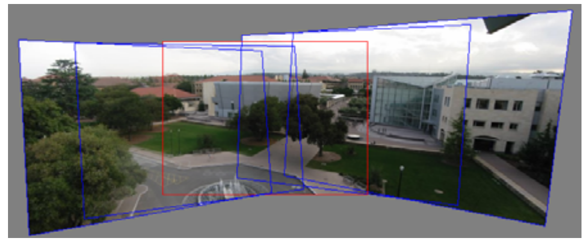
$$\min \|\mathbf{Ah} - \mathbf{b}\|^2$$

- Can be done in Matlab using "\" command - see "help lmdivide"

Julian Beever: Manual Homographies

http://users.skynet.be/J.Beever/pave.htm

Panoramas

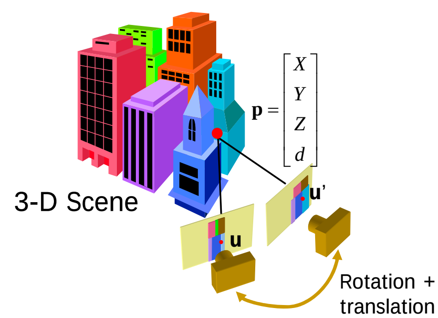


1. Pick one image (red)
2. Warp the other images towards it (usually, one by one)
3. blend

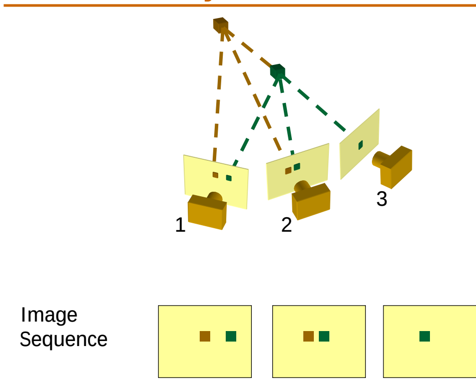
Recap

- Panorama = reprojection
- 3D rotation ➔ homography
  - Homogeneous coordinates are kewl
- Use feature correspondence
- Solve least square problem
  - Set of linear equations
- Warp all images to a reference one
- Use your favorite blending

Changing Camera Centers

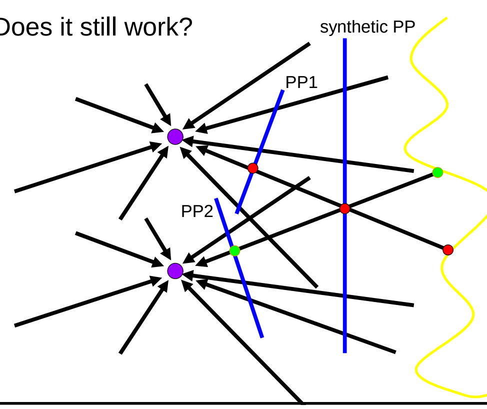


General Projective Model

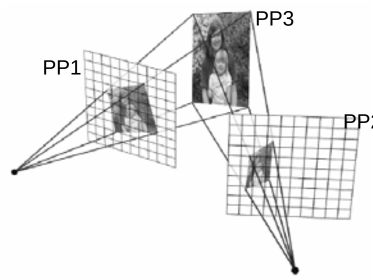


changing camera center

- Does it still work?

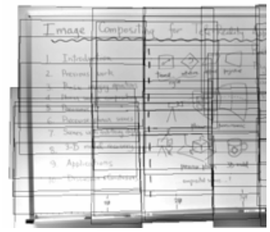


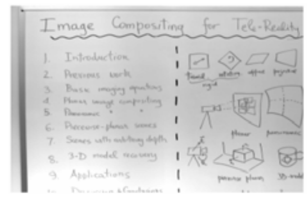
Planar scene (or far away)



- PP3 is a projection plane of both centers of projection, so we are OK!
- This is how big aerial photographs are made

Planar mosaic





Problem Set #3

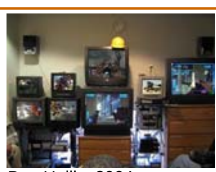
- **Homographies and Panoramic Mosaics**
- Capture photographs (and possibly video)
  - Might want to use tripod
- Compute homographies (define correspondences)
  - will need to figure out how to setup system of eqs.
- (un)warp an image (undo perspective distortion)
- Produce a panoramic mosaics (with blending)

Bells and Whistles

- Blending and Compositing
  - use homographies to combine images or video and images together in an interesting (fun) way. E.g.
    - put fake graffiti on buildings or chalk drawings on the ground
    - replace a road sign with your own poster
    - project a movie onto a building wall
    - etc.

Results from other classes

Ben Hollis, 2004

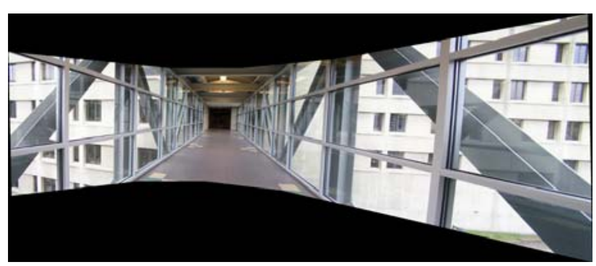


Ben Hollis, 2004

Matt Pucevich , 2004

Eunjeong Ryu (E.J), 2004

Go Explore!



Ken Chu, 2004