Image Warping and Mosacing

15-463: Rendering and Image Processing
Alexei Efros

…with a lot of slides stolen from Steve Seitz and Rick Szeliski

Today

Mosacs

Image Warping

Homographies

Programming Assignment #2 OUT

Reading:

- Paul Heckbert, “Projective Mappings for Image Warping”, 1999
- Rick Szeliski, Chapter on Mosaicing from his new book (2005-2006), being written as we speak, hopefully on the web by Monday

Mosaics: stitching images together

virtual wide-angle camera

How to do it?

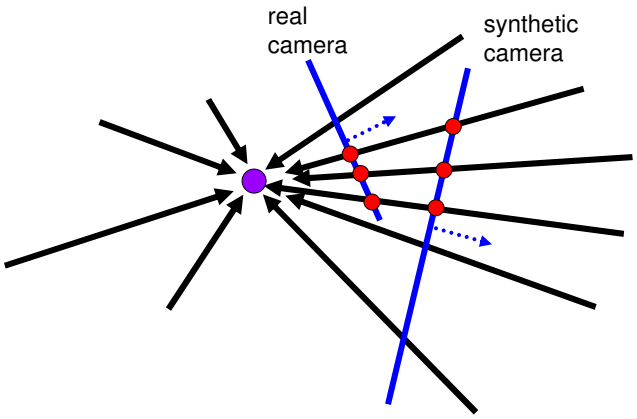
Basic Procedure

- 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

…but **wait**, why should this work at all?

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

A pencil of rays contains all views

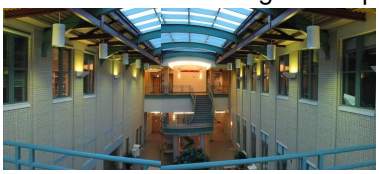


Can generate any synthetic camera view
as long as it has **the same center of projection**!

Aligning images

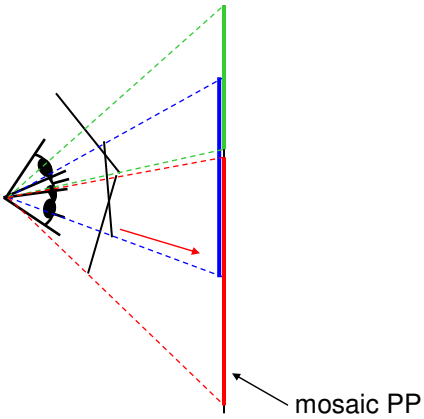
left on top

right on top



Translations are not enough to align the images

Image reprojection



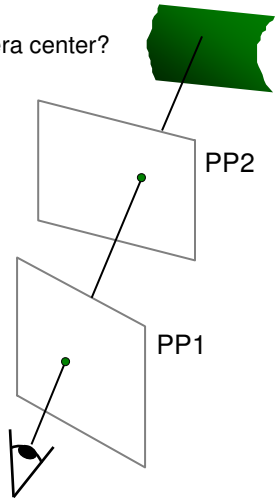
The mosaic has a natural interpretation in 3D

- 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 two images from the 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

Image Warping

image filtering: change ***range*** of image

$$g(x) = h(f(x))$$

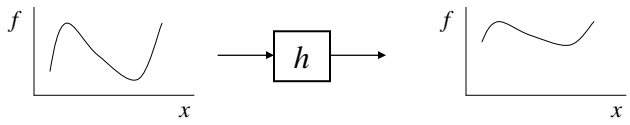


image warping: change ***domain*** of image

$$g(x) = f(h(x))$$

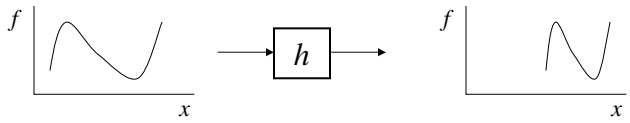


Image Warping

image filtering: change ***range*** of image

$$g(x) = h(f(x))$$

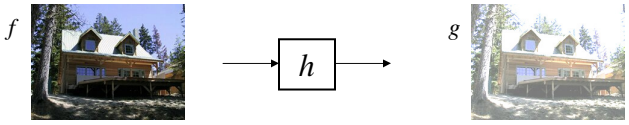
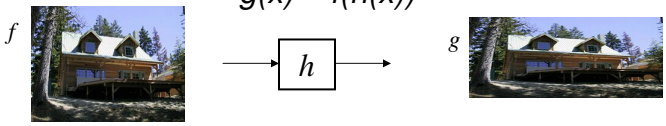


image warping: change ***domain*** of image

$$g(x) = f(h(x))$$



Parametric (global) warping

Examples of parametric warps:

translation

rotation

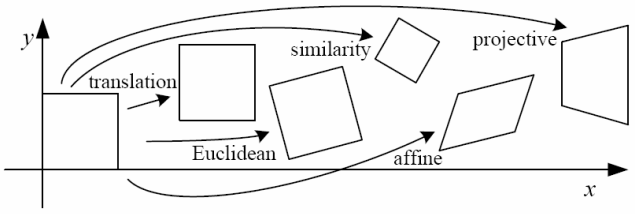
aspect

affine

perspective

cylindrical

2D image transformations

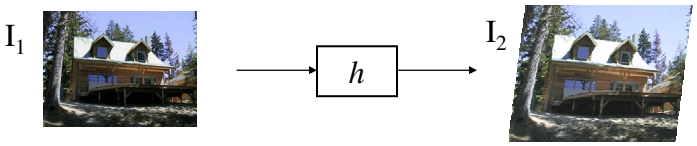


Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times3}$	8	straight lines	

These transformations are a nested set of groups

- Closed under composition and inverse is a member

Recovering warp parameters



Given two images and a transformation type, how do we recover the parameters?

1. Search
   - parameters of h() are: $\arg\min_h \sum_i [I_1(x) - I_2(h(x))]^2$
   - Easy for translation: loop through t_x and t_y
   - But harder with more DOF
   - Minimization, e.g. Newton's Method
2. Point Correspondences (often user-defined)
   - Click on the same point in two images
   - How many correspondences for:
     - translation, Euclidean, similarity, affine, projective

Homography

Q: Which t-form is the right one for warping PP1 into PP2?

- translation, Euclidean, affine, projective

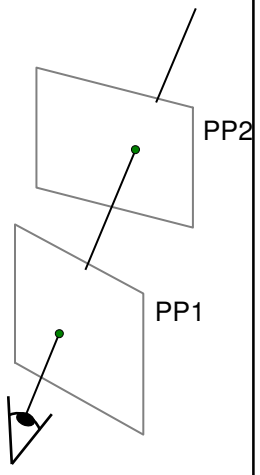
A: Projective – mapping between any two PPs with the same center of projection

- rectangle should map to arbitrary quadrilateral
- must preserve straight lines
- same as: project, rotate, reproject

called Homography

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\mathbf{p'}$ $\quad$ $\mathbf{H}$ $\quad$ $\mathbf{p}$



To apply a homography **H**

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

Image warping with homographies

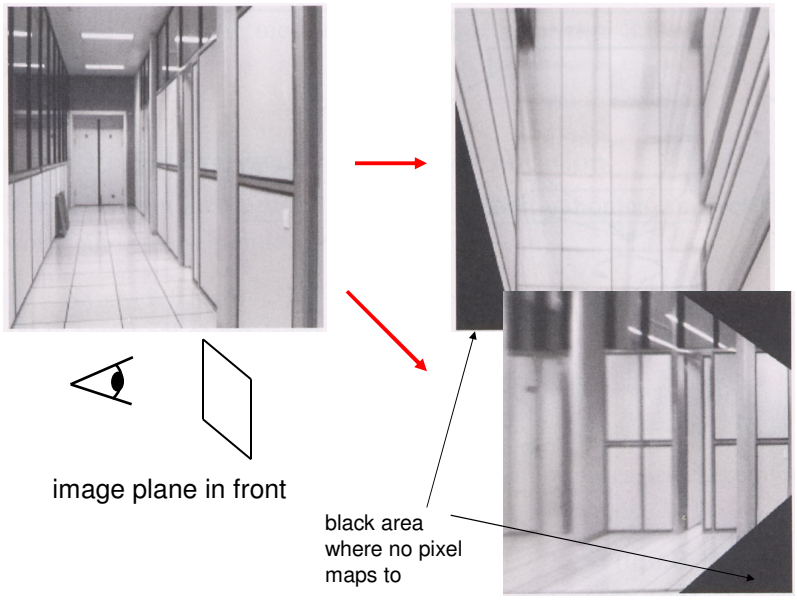
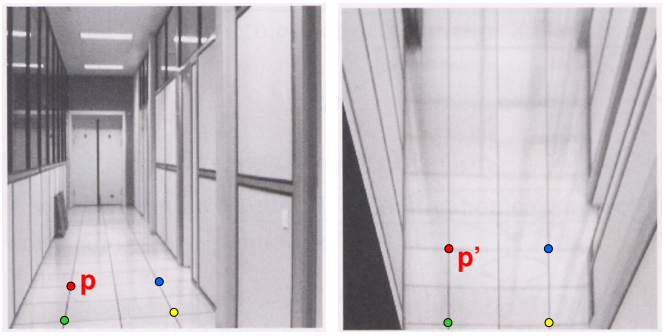


Image rectification



To unwarp (rectify) an image

- solve for homography **H** given a set of **p** and **p'** pairs
- solve equations of the form: w**p'** = **Hp**
  - linear in unknowns: w and coefficients of **H**
  - H is defined up to an arbitrary scale factor
  - how many points are necessary to solve for **H**?

Solving for homographies

$$\mathbf{p'} = \mathbf{Hp}$$

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

Can set scale factor i=1. So, there are 8 unkowns.

Set up a system of linear equations:

$$\mathbf{Ah = b}$$

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:

$$\min \|Ah - b\|^2$$

Can be done in Matlab using "\" command

- see "help lmdivide"

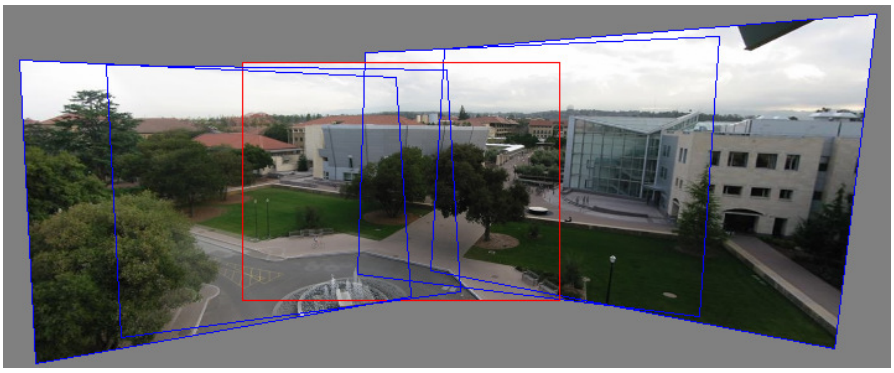
Fun with homographies

Original image

St.Petersburg
photo by A. Tikhonov

Virtual camera rotations

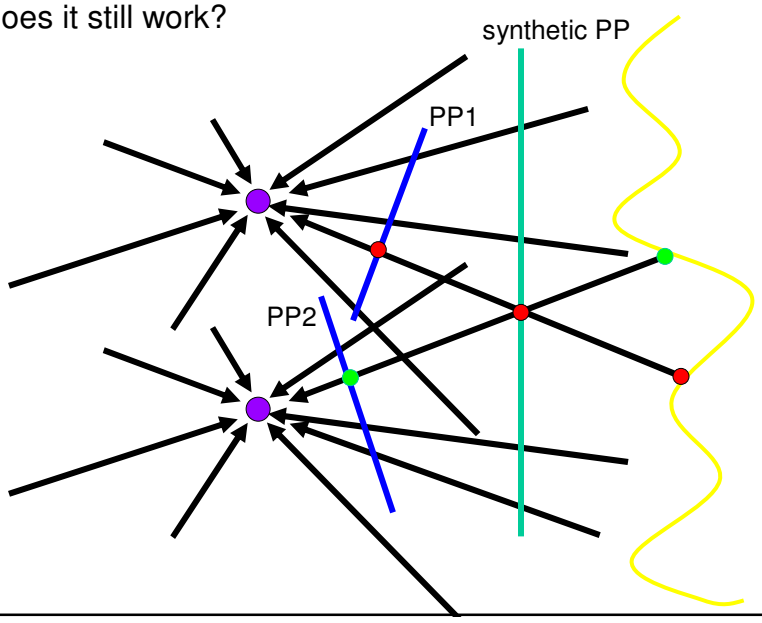
Panoramas



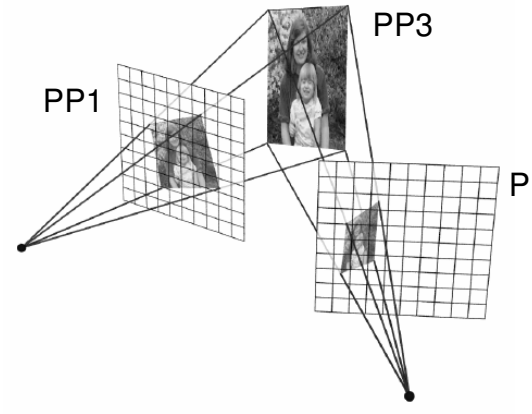
1. Pick one image (red)
2. Warp the other images towards it (usually, one by one)
3. blend

changing camera center

Does it still work?



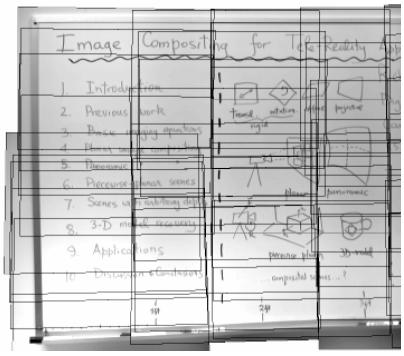
Planar scene (or far away)



PP3 is a projection plane of both centers of projection, so we are OK!

This is how big areal photographs are made

Planar mosaic



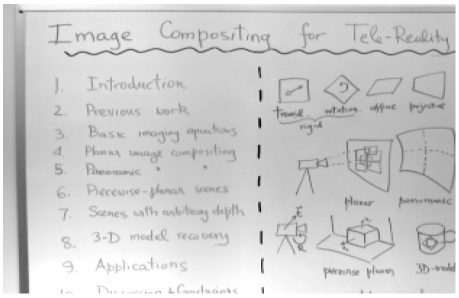
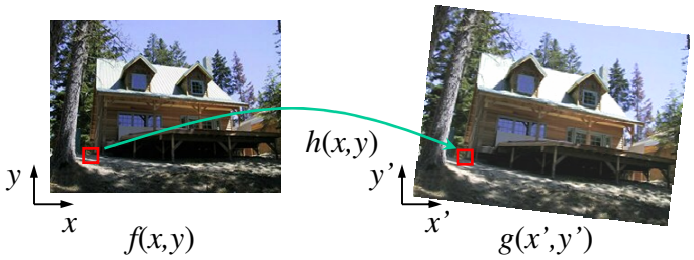
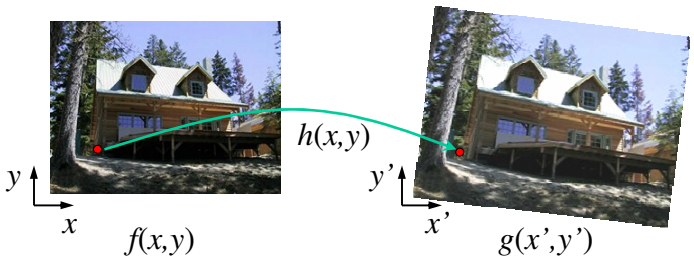


Image warping



Given a coordinate transform *(x',y')* = *h*(*x*,*y*) and a source image *f*(*x*,*y*), how do we compute a transformed image *g(x',y')* = *f*(*h*(*x*,*y*))?

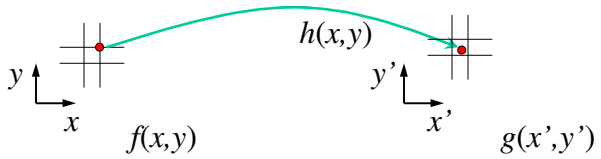
Forward warping



Send each pixel *f*(*x*,*y*) to its corresponding location (*x',y')* = *h*(*x*,*y*) in the second image

Q: what if pixel lands "between" two pixels?

Forward warping



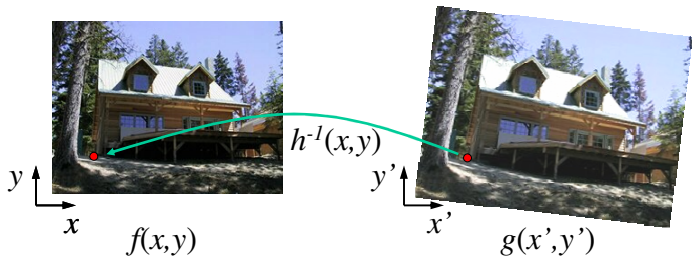
Send each pixel $f(x,y)$ to its corresponding location
$(x',y') = h(x,y)$ in the second image

Q: what if pixel lands "between" two pixels?

A: distribute color among neighboring pixels (x',y')

– Known as "splatting"

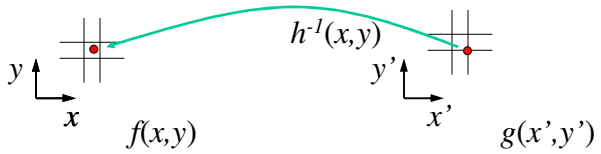
Inverse warping



Get each pixel $g(x',y')$ from its corresponding location
$(x,y) = h^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

Inverse warping



Get each pixel $g(x',y')$ from its corresponding location
$(x,y) = h^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

A: *resample* color value

- We discussed resampling techniques before
  - nearest neighbor, bilinear, Gaussian, bicubic

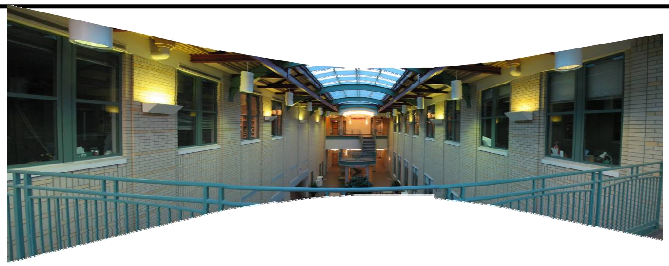
Forward vs. inverse warping

Q: which is better?

A: usually inverse—eliminates holes

- however, it requires an invertible warp function—not always possible...

Programming Assignment #2



Homographies and Panoramic Mosaics

- Capture photographs (and possibly video)
  - can check out cameras and/or tripods from us (1 day loan)
- Compute homographies (define correspondences)
  - will need to figure out how to setup system of eqs.
- Warp images (show some nice warps)
- Produce panoramic mosaics
  - will discuss blending next time
- Do some of the Bells and Whistles (can be in pairs)

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.

Bells and Whistles

Automatic stitching

- Implement automatic (or semi-automatic) stitching
- there are several methods, probably the easiest is to do cylindrical panoramas. But will need to know focal length and have tripod. Other approaches possible. Talk to me.
- Can run the algorithm on a video sequence (build a panorama from video)





Bells and Whistles

Capture creative/cool/bizzare panoramas, either rotational or planar

- Example from UW (by Brett Allen):

- Ever wondered what is happening inside your fridge while you are not looking?

Bells and Whistles

Video Panorama

- Capture two (or more) stationary videos (either from the same point, or of a planar/far-away scene). Compute homography and produce a video mosaic. Need to worry about synchronization (not too hard).
- e.g. capturing a football game from the sides of the stadium

Other interesting ideas?

- talk to me