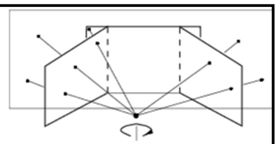


Image warping and stitching

Monday Feb 28
Prof. Kristen Grauman
UT-Austin

HP frames commercials

- http://www.youtube.com/watch?v=2RPl5vPEoQk

Announcements

- Reminder: Pset 2 due Wed March 2
- Reminder: Midterm exam is Wed March 9
  – See practice exam handout
- My office hours Wed: 12:15-1:15
- Matlab license issues – see course website
- Pset 1 and solutions were returned last week – grades online

Last time

- Interactive segmentation
- Feature-based alignment
  – 2D transformations
  – Affine fit
  – RANSAC

Today

- RANSAC for robust fitting
  – Lines, translation
- Image mosaics
  – Fitting a 2D transformation
    - Affine, Homography
  – 2D image warping
  – Computing an image mosaic
  – **Wednesday**: which local features to match?

Alignment problem

- We have previously considered how to **fit a model to image evidence**
  – e.g., a line to edge points, or a snake to a deforming contour
- In alignment, we will **fit the parameters of some transformation** according to a set of matching feature pairs (“correspondences”).

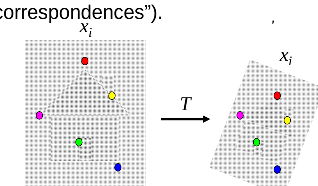
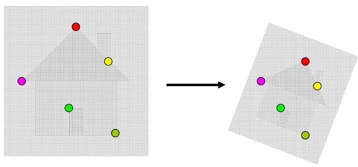
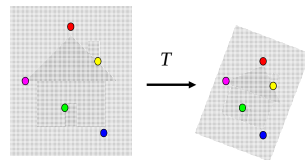


Image alignment

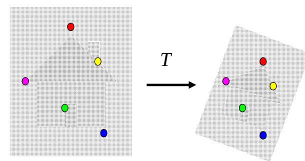


- Two broad approaches:
  - Direct (pixel-based) alignment
    - Search for alignment where most pixels agree
  - Feature-based alignment
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment

Main questions

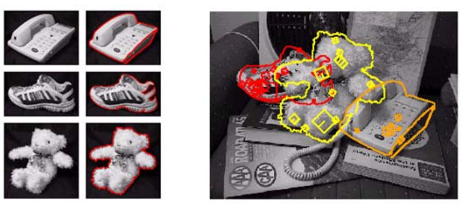


Alignment: Given two images, what is the transformation between them?



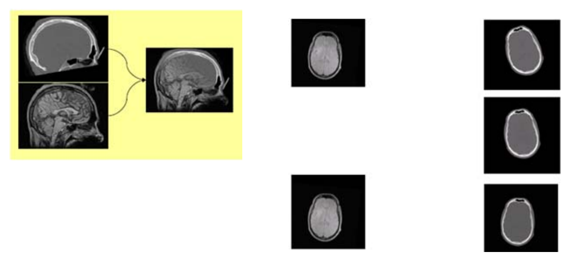
Warping: Given a source image and a transformation, what does the transformed output look like?

Motivation for feature-based alignment: Recognition



Figures from David Lowe

Motivation for feature-based alignment: Medical image registration



Motivation for feature-based alignment: Image mosaics

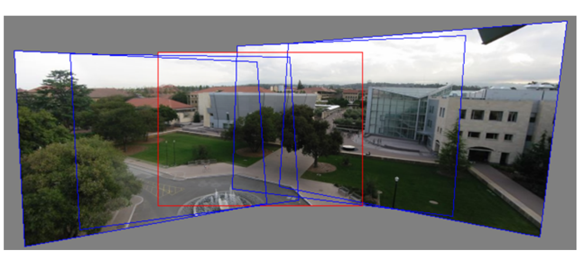
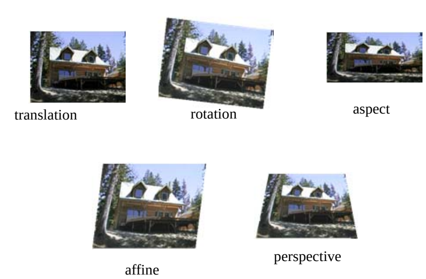


Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fa

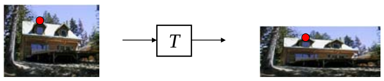
Parametric (global) warping

Examples of parametric warps:



Source: Alyosha Efros

Parametric (global) warping



p = (x,y) **p'** = (x',y')

Transformation T is a coordinate-changing machine:

p' = *T*(p)

What does it mean that *T* is **global**?

- Is the same for any point p
- can be described by just a few numbers (parameters)

Let's represent *T* as a matrix:

p' = **M**p

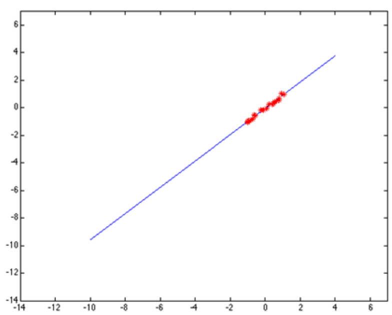
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

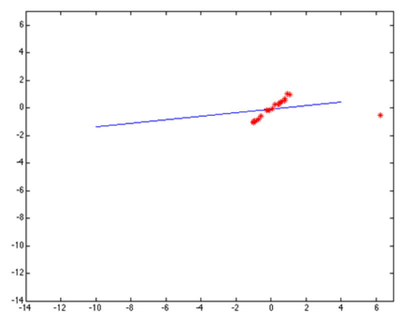
Outliers

- **Outliers** can hurt the quality of our parameter estimates, e.g.,
  - an erroneous pair of matching points from two images
  - an edge point that is noise, or doesn't belong to the line we are fitting.

Outliers affect least squares fit



Outliers affect least squares fit



RANSAC

- RANdom Sample Consensus
- **Approach**: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.
- **Intuition**: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

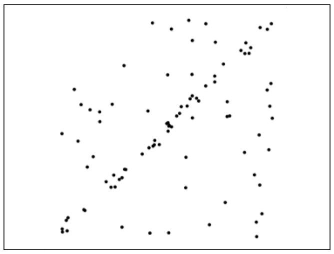
RANSAC: General form

- RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

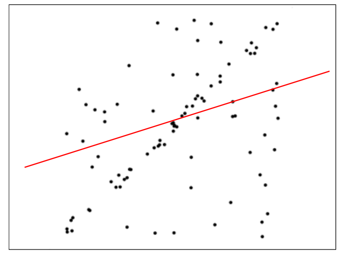
RANSAC for line fitting example



Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

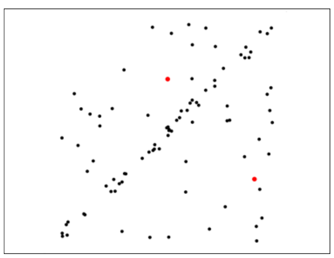


Least-squares fit

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

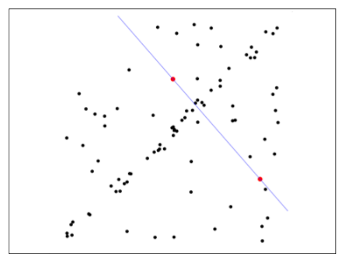


1. Randomly select minimal subset of points

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

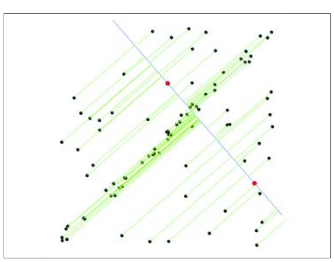


1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

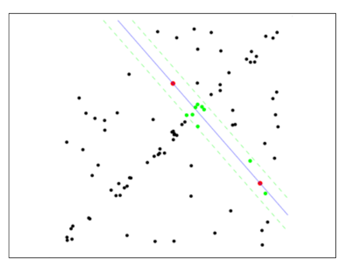


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

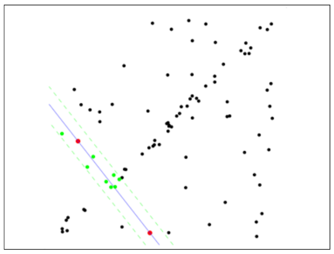


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

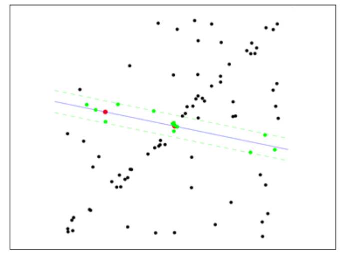


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

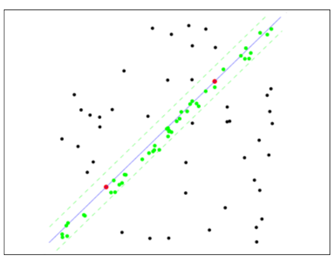
26

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

Uncontaminated sample



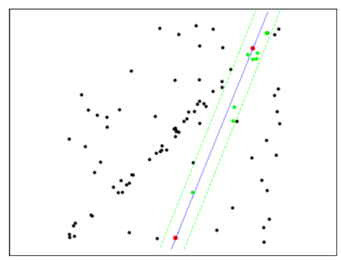
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

27

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting

Repeat ***N*** times:

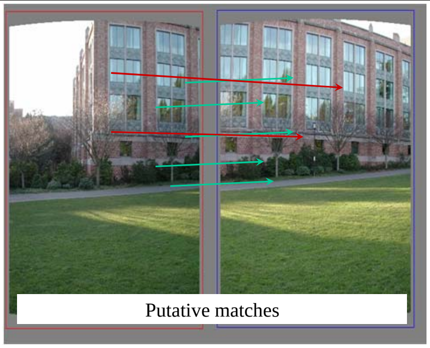
- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

Lana Lazebnik

That is an example fitting a model (line)…

What about fitting a transformation (translation)?

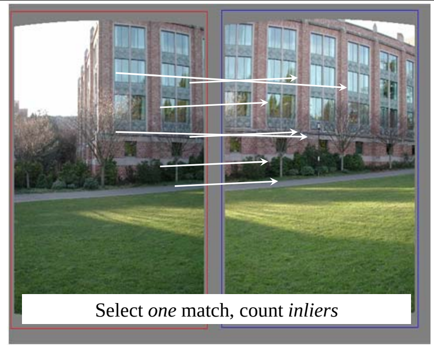
RANSAC example: Translation



Putative matches

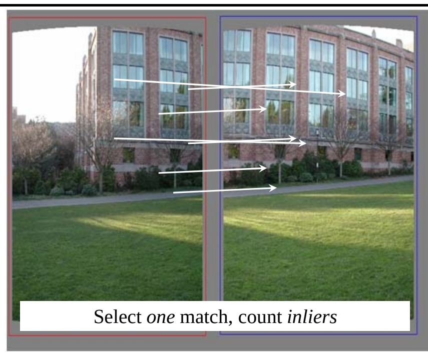
Source: Rick Szeliski

RANSAC example: Translation



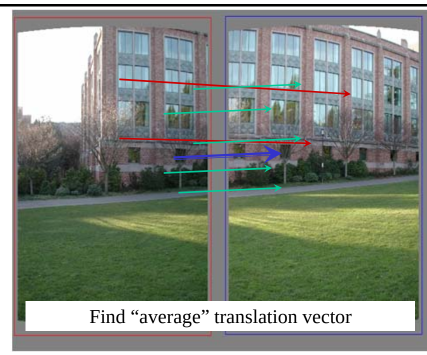
Select *one* match, count *inliers*

RANSAC example: Translation



Select *one* match, count *inliers*

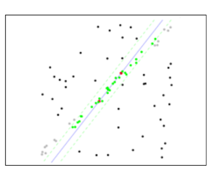
RANSAC example: Translation



Find "average" translation vector

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - Doesn't work well for low inlier ratios (too many iterations, or can fail completely)
  - Can't always get a good initialization of the model based on the minimum number of samples



Lana Lazebnik

Today

- RANSAC for robust fitting
  - Lines, translation
- Image mosaics
  - Fitting a 2D transformation
    - Affine, Homography
  - 2D image warping
  - Computing an image mosaic

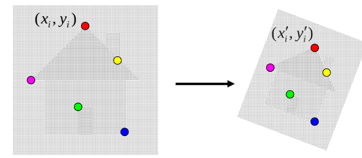
Recall: fitting an affine transformation

Affine model approximates perspective projection of planar objects.

Figures from David Lowe, ICCV 1999

Fitting an affine transformation

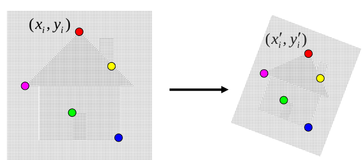
- Assuming we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

Fitting an affine transformation

- Assuming we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x'_i \\ y'_i \\ \cdots \end{bmatrix}$$

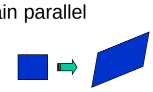
2D Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Affine transformations are combinations of …

- Linear transformations, and
- Translations

Parallel lines remain parallel



Motivation for feature-based alignment: Image mosaics

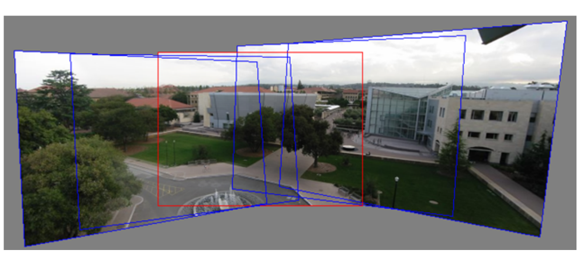


Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fa

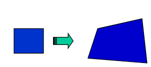
Projective Transformations

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

Projective transformations:

- Affine transformations, and
- Projective warps

Parallel lines do not necessarily remain parallel



Mosaics

Obtain a wider angle view by combining multiple images.

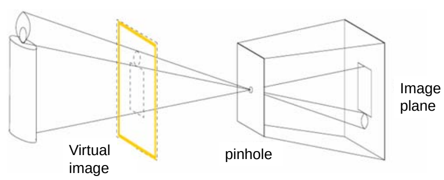
How to stitch together a panorama (a.k.a. mosaic)?

- 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?

Source: Steve Seitz

Pinhole camera

- Pinhole camera is a simple model to approximate imaging process, perspective **projection**.



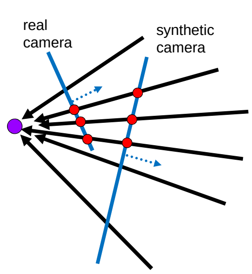
If we treat pinhole as a point, only one ray from any given point can enter the camera.

Fig from Forsyth and Ponce

Mosaics

Obtain a wider angle view by combining multiple images.

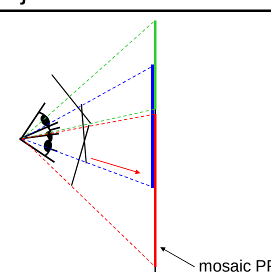
Mosaics: generating synthetic views



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

Source: Alyosha Efros

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*

Source: Steve Seitz

Image reprojection

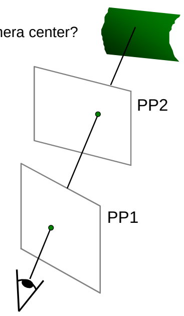
Basic question

- 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

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



Source: Alyosha Efros

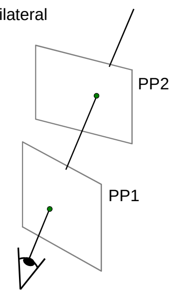
Image reprojection: Homography

A projective transform is a 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

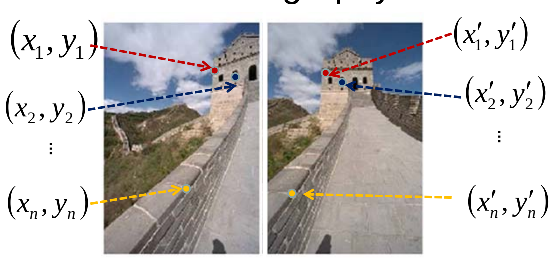
called **Homography**

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



Source: Alyosha Efros

Homography



To **compute** the homography given pairs of corresponding points in the images, we need to set up an equation where the parameters of **H** are the unknowns…

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 unknowns.
Set up a system of linear equations:

Ah = b

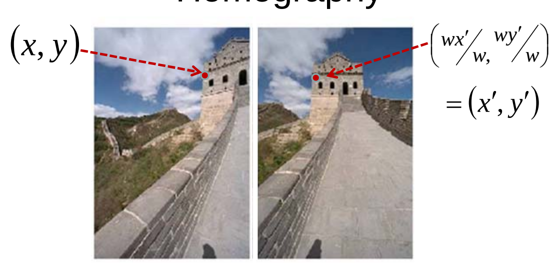
where vector of unknowns h = [a,b,c,d,e,f,g,h]^T
Need at least 8 eqs, but the more the better…
Solve for h. If overconstrained, solve using least-squares:

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

```
>> help lmdivide
```

BOARD

Homography



To **apply** a given homography **H**

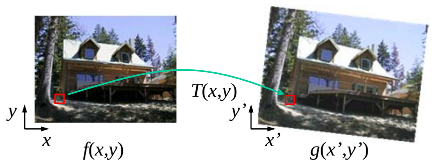
- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

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

Today

- RANSAC for robust fitting
  - Lines, translation
- Image mosaics
  - Fitting a 2D transformation
    - Affine, Homography
  - 2D image warping
  - Computing an image mosaic

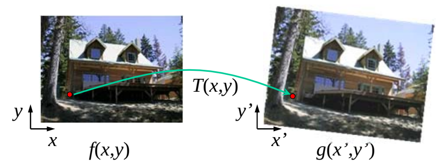
Image warping



Given a coordinate transform and a source image *f*(*x,y*), how do we compute a transformed image *g*(*x',y'*) = *f*(*T*(*x,y*))?

Slide from Alyosha Efros, CMU

Forward warping

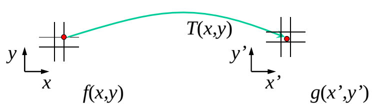


Send each pixel *f*(*x,y*) to its corresponding location (*x',y'*) = *T*(*x,y*) in the second image

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

Slide from Alyosha Efros, CMU

Forward warping



Send each pixel *f*(*x,y*) to its corresponding location (*x',y'*) = *T*(*x,y*) in the second image

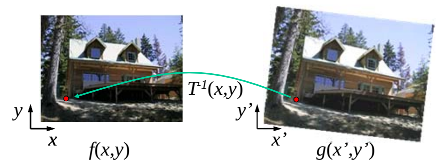
Q: what if pixel lands "between" two pixels?

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

- Known as "splatting"

Slide from Alyosha Efros, CMU

Inverse warping

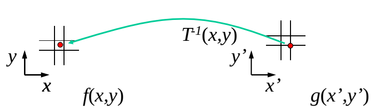


Get each pixel *g*(*x',y'*) from its corresponding location (*x,y*) = T^{-1}(*x',y'*) in the first image

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

Slide from Alyosha Efros, CMU

Inverse warping



Get each pixel *g*(*x',y'*) from its corresponding location (*x,y*) = T^{-1}(*x',y'*) in the first image

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

A: *Interpolate* color value from neighbors

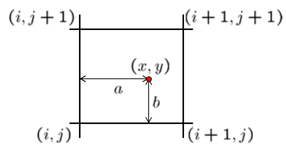
- nearest neighbor, bilinear…

Slide from Alyosha Efros, CMU

`>> help interp2`

Bilinear interpolation

Sampling at *f(x,y):*



$$
\begin{aligned}
f(x,y) = \; & (1-a)(1-b) & f[i,j] \\
& +a(1-b) & f[i+1,j] \\
& +ab & f[i+1,j+1] \\
& +(1-a)b & f[i,j+1]
\end{aligned}
$$

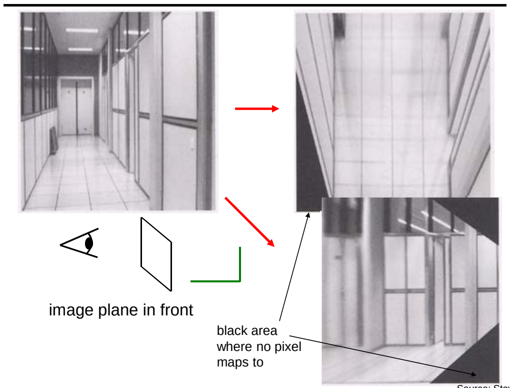
Slide from Alyosha Efros, CMU

Recap: How to stitch together a panorama (a.k.a. mosaic)?

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

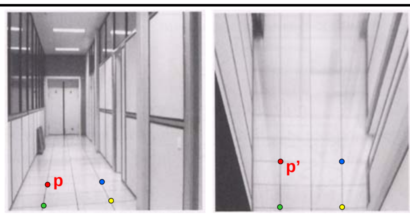
Source: Steve Seitz

Image warping with homographies



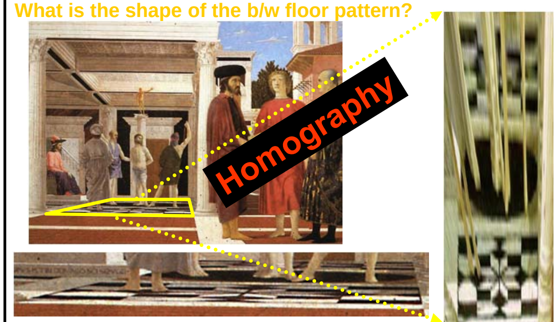
Source: Steve Seitz

Image rectification



Analysing patterns and shapes

What is the shape of the b/w floor pattern?



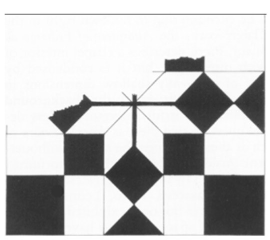
The floor (enlarged)

Automatically rectified floor

Slide from Antonio Criminisi

Analysing patterns and shapes

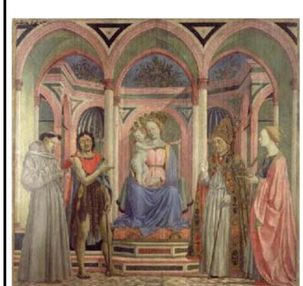
Automatic rectification



From Martin Kemp *The Science of Art* *(manual reconstruction)*

Slide from Antonio Criminisi

Analysing patterns and shapes



What is the (complicated) shape of the floor pattern?

Automatically rectified floor

St. Lucy Altarpiece, D. Veneziano

Slide from Criminisi

Analysing patterns and shapes

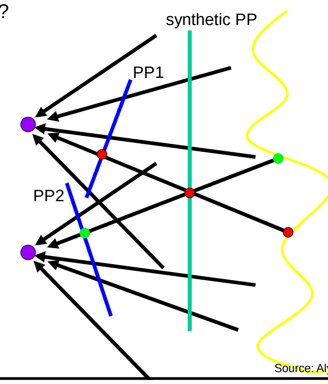
Automatic rectification

From Martin Kemp, *The Science of Art* *(manual reconstruction)*

Slide from Criminisi

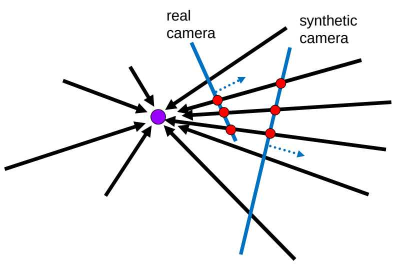
Changing camera center

Does it still work?



Source: Alyosha Efros

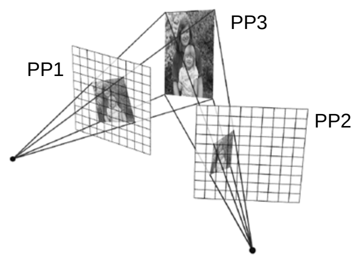
Recall: same camera center



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

Source: Alyosha Efros

…Or: 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

Source: Alyosha Efros

Summary: alignment & warping

- Write **2d transformations** as matrix-vector multiplication (including translation when we use homogeneous coordinates)
- Perform **image warping** (forward, inverse)
- **Fitting transformations**: solve for unknown parameters given corresponding points from two views (affine, projective (homography)).
- **Mosaics**: uses homography and image warping to merge views taken from same center of projection.

Next time: which features should we match?