http://www.youtube.com/watch?v=lde77E4PY4Q

Alignment and Image Warping

Tuesday, Oct 6

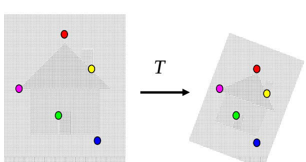
Announcements

- Midterm is next Tues, 10/13
  - In class
  - Can bring one 8.5 x 11" sheet of notes
  - Handout: 2 previous years' midterms

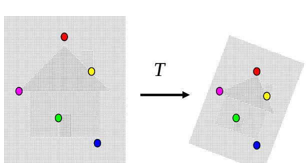
Today

- Alignment & warping
  - 2d transformations
  - Forward and inverse image warping
  - Fitting transformations
    - Affine
    - Projective
  - Application: constructing mosaics

Main questions

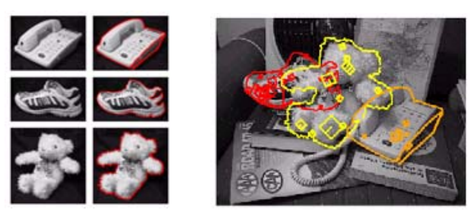


Warping: Given a source image and a transformation, what does the transformed output look like?



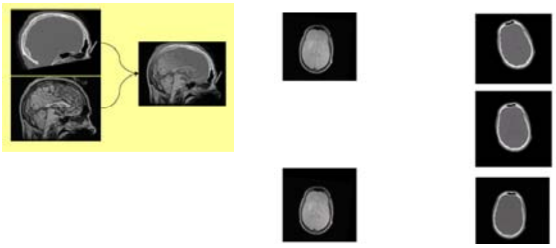
Alignment: Given two images with corresponding features, what is the transformation between them?

Motivation: Recognition



Figures from David Lowe

Motivation: medical image registration



Motivation: Mosaics

- Getting the whole picture
  - Consumer camera: 50° x 35°

Slide from Brown & Lowe 2003

Motivation: Mosaics

- Getting the whole picture
  - Consumer camera: 50° x 35°
  - Human Vision: 176° x 135°

Slide from Brown & Lowe 2003

Motivation: Mosaics

- Getting the whole picture
  - Consumer camera: 50° x 35°
  - Human Vision: 176° x 135°

- Panoramic Mosaic = up to 360 x 180°

Slide from Brown & Lowe 2003

Warping problem

- Given a set of points and a transformation, generate the warped image

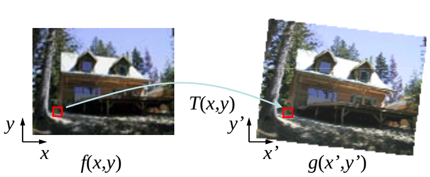
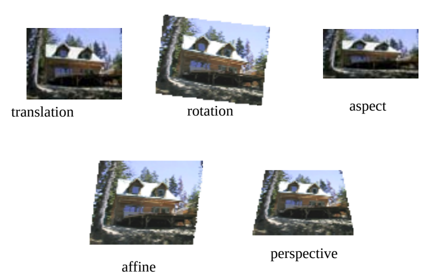


Figure Alyosha Efros

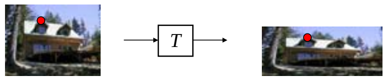
Parametric (global) warping

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

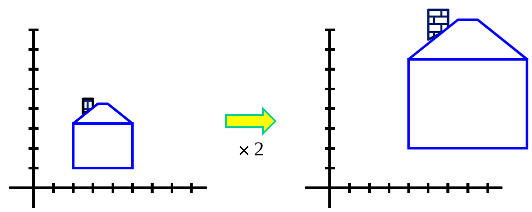
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

Scaling

Scaling a coordinate means multiplying each of its components by a scalar

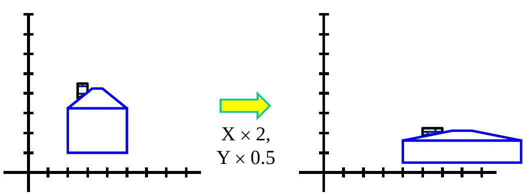
Uniform scaling means this scalar is the same for all components:



Source: Alyosha Efros

Scaling

Non-uniform scaling: different scalars per component:



Source: Alyosha Efros

Scaling

Scaling operation:

$$x' = ax$$
$$y' = by$$

Or, in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix } S} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

What transformations can be represented with a 2x2 matrix?

2D Scaling?

$$x' = s_x * x$$
$$y' = s_y * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotate around (0,0)?

$$x' = \cos\Theta * x - \sin\Theta * y$$
$$y' = \sin\Theta * x + \cos\Theta * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Shear?

$$x' = x + sh_x * y$$
$$y' = sh_y * x + y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

What transformations can be represented with a 2x2 matrix?

2D Mirror about Y axis?

$$x' = -x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Mirror over (0,0)?

$$x' = -x$$
$$y' = -y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Translation?

$$x' = x + t_x$$
$$y' = y + t_y$$

NO!

Source: Alyosha Efros

2D Linear Transformations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Only linear 2D transformations can be represented with a 2x2 matrix.

Linear transformations are combinations of …

- Scale,
- Rotation,
- Shear, and
- Mirror

Source: Alyosha Efros

Homogeneous Coordinates

Q: How can we represent translation as a 3x3 matrix using homogeneous coordinates?

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}_x$$
$$\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{t}_y$$

Source: Alyosha Efros

Homogeneous Coordinates

Q: How can we represent translation as a 3x3 matrix using homogeneous coordinates?

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}_x$$
$$\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{t}_y$$

A: Using the rightmost column:

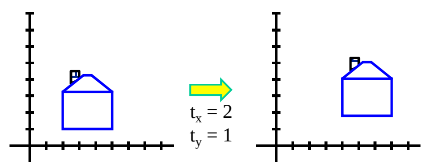
$$\boldsymbol{Translation} = \begin{bmatrix} 1 & 0 & \boldsymbol{t}_x \\ 0 & 1 & \boldsymbol{t}_y \\ 0 & 0 & 1 \end{bmatrix}$$

Source: Alyosha Efros

Translation

Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



Source: Alyosha Efros

Basic 2D Transformations

Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh}_x & 0 \\ \boldsymbol{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

Source: Alyosha Efros

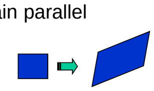
2D Affine Transformations

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

Affine transformations are combinations of …

- Linear transformations, and
- Translations

Parallel lines remain parallel



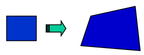
Projective Transformations

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

Projective transformations:

- Affine transformations, and
- Projective warps

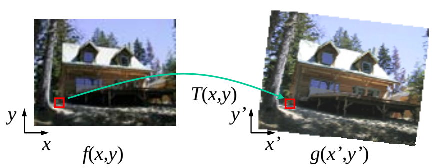
Parallel lines do not necessarily remain parallel



Today

- Alignment & warping
  - 2d transformations
  - Forward and inverse image warping
  - Fitting transformations
    - Affine
    - Projective
  - Application: constructing mosaics

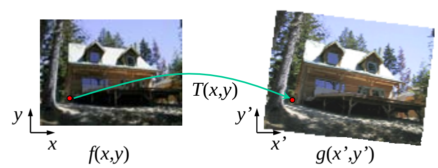
Image warping



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

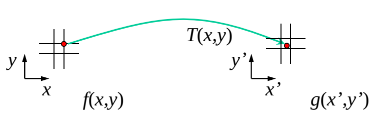


Send each pixel *f*(*x,y*) to its corresponding location (*x',y'*) = *T*(*x,y*) in the second image

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

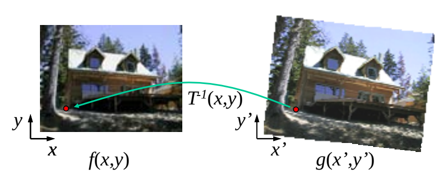
Q: what if pixel lands "between" two pixels?

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

- Known as "splatting"

Slide from Alyosha Efros, CMU

Inverse warping

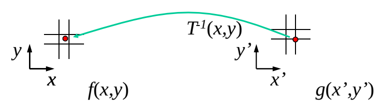


Get each pixel *g*(*x',y'*) from its corresponding location (*x,y*) = T^{-1}(*x',y'*) in the first image

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

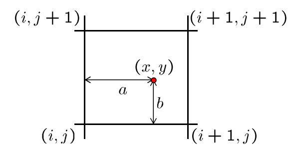
– nearest neighbor, bilinear…

Slide from Alyosha Efros, CMU

```
>> help interp2
```

Bilinear interpolation

Sampling at *f(x,y):*



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

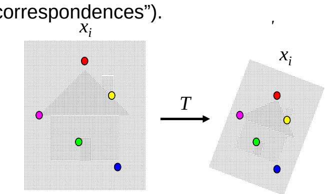
Slide from Alyosha Efros, CMU

Today

- Alignment & warping
  - 2d transformations
  - Forward and inverse image warping
  - Fitting transformations
    - Affine
    - Projective
  - Application: constructing mosaics

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").



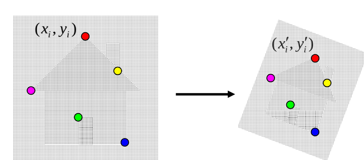
Fitting an affine transformation

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?



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

An aside: Least Squares Example

Say we have a set of data points (X1,X1'), (X2,X2'), (X3,X3'), etc. (e.g. person's height vs. weight)

We want a nice compact formula (a line) to predict X's from Xs: Xa + b = X'

We want to find a and b

How many (X,X') pairs do we need?

$$X_1 a + b = X_1' \qquad X_2 a + b = X_2'$$

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \end{bmatrix}$$

Ax=B

What if the data is noisy?

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ \dots \end{bmatrix}$$

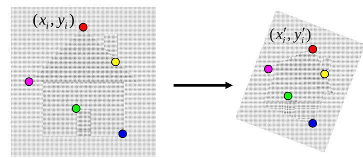
$$\min \|Ax - B\|^2$$

overconstrained

Source: Alyosha Efros

Fitting an affine transformation

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



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

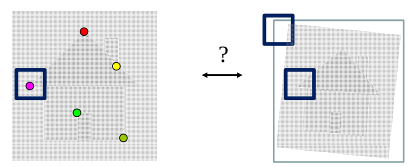
$$\begin{bmatrix} \\ \\ \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

Fitting an affine transformation

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

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?
- Once we have solved for the parameters, how do we compute the coordinates of the corresponding point for (x_{new}, y_{new}) ?

What **are** the correspondences?



- Compare content in **local** patches, find best matches.
  e.g., simplest approach: scan with template, and compute SSD or correlation between list of pixel intensities in the patch
- Later in the course: how to select regions according to the geometric changes, and more robust descriptors.

Today

- Alignment & warping
  - 2d transformations
  - Forward and inverse image warping
  - Fitting transformations
    - Affine
    - Projective
  - Application: constructing mosaics

Panoramas

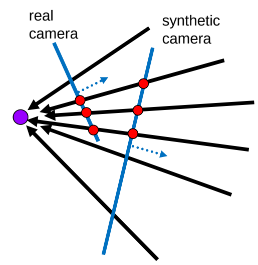
Obtain a wider angle view by combining multiple images.

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

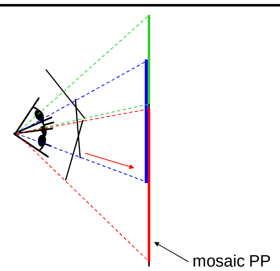
Panoramas: generating synthetic views



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*

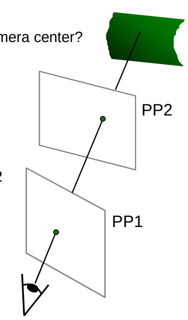
Source: Steve Seitz

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.



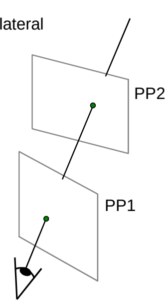
Source: Alyosha Efros

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**

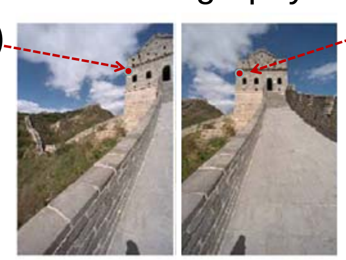
$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$



Source: Alyosha Efros

Homography

(x, y)

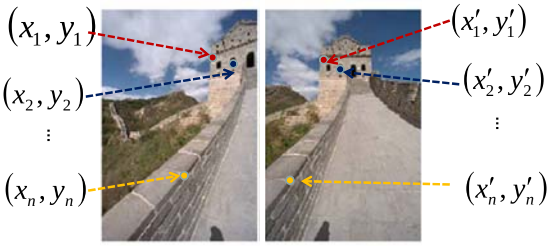


$\left(\frac{wx'}{w}, \frac{wy'}{w}\right) = (x', y')$

To **apply** a given homography **H**
- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

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

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
```

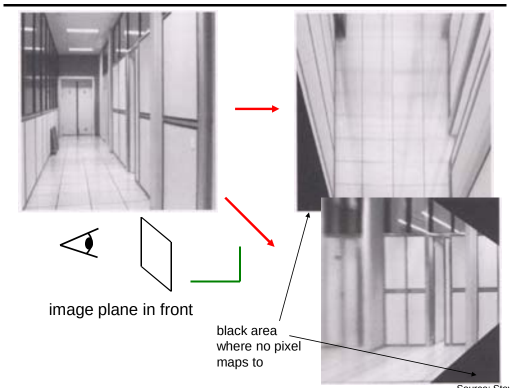
BOARD

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)

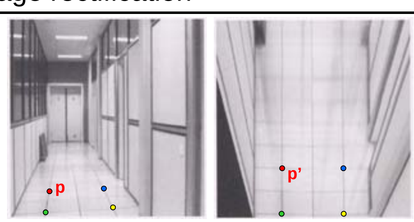
Source: Steve Seitz

Image warping with homographies

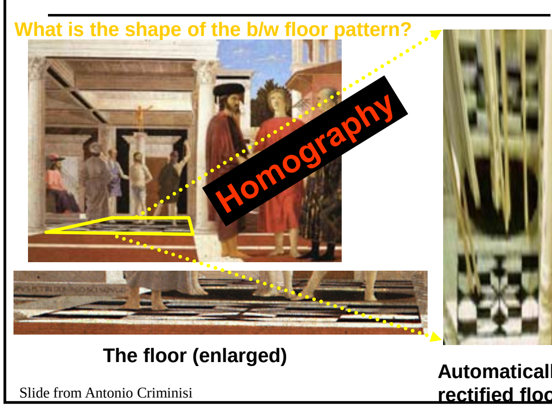


Source: Steve Seitz

Image rectification



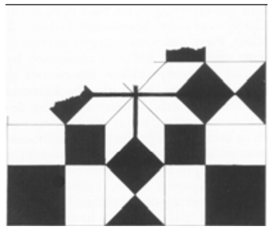
Analysing patterns and shapes



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

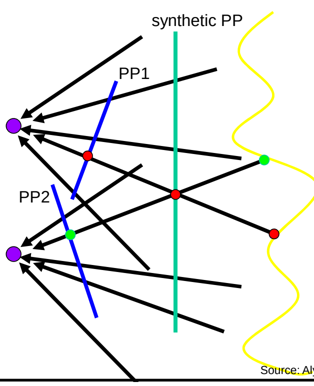
Automatic rectification

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

Slide from Criminisi

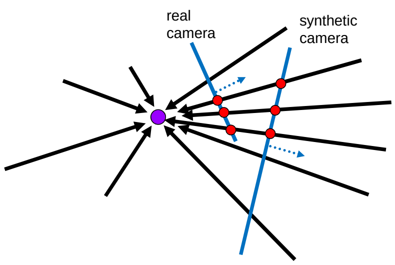
Changing camera center

Does it still work?



Source: Alyosha Efros

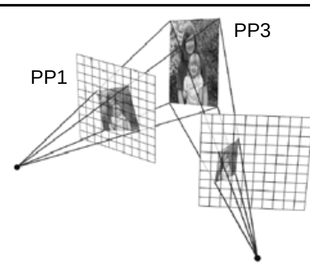
Recall: same camera center



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

Source: Alyosha Efros

…Or: Planar scene (or far away)



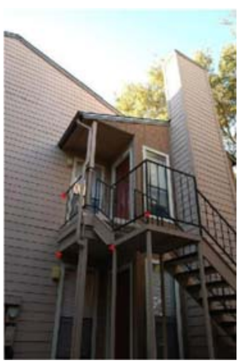
PP3 is a projection plane of both centers of projection,
so we are OK!

This is how big aerial photographs are made

Source: Alyosha Efros

Some mosaic results
from Fall 2008

Ming-Jun Chen



Ming-Jun Chen

Andrew Harp

Wei-Cheng Su

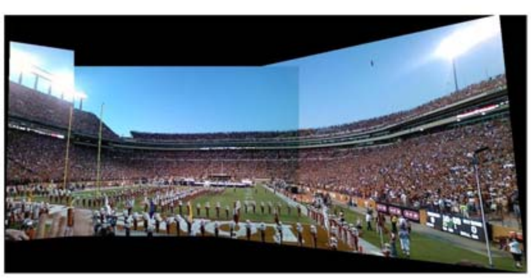
Wei-Cheng Su

Wei-Cheng Su

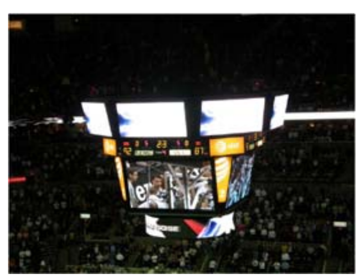
Wei-Cheng Su

Andy Luong

Andy Luong



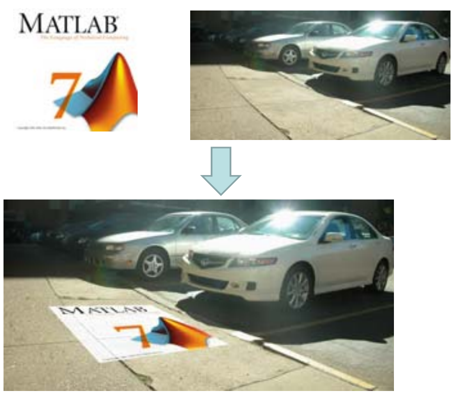
Chia-Sheng Tsai



Chia-Sheng Tsai

Chia-Sheng Tsai

Chia-Sheng Tsai



Ekapol Chuangsuwanich, CMU

Fei Li

Sung Ju Hwang

Sung Ju Hwang

Sung Ju Hwang

Sung Ju Hwang

Sung Ju Hwang

HP “Frames” commercials

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

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.