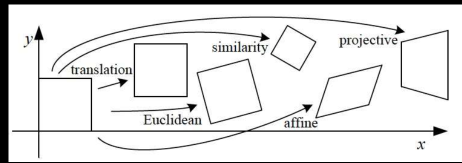
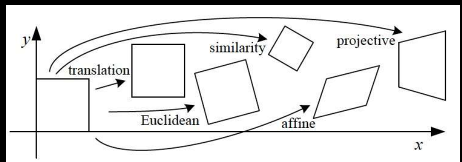
Computer Vision

Image to image projections

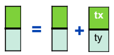
2D Transformations

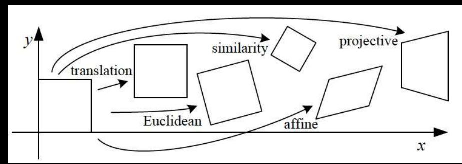




Example: translation

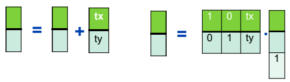
$x' = x + \vec{t}$

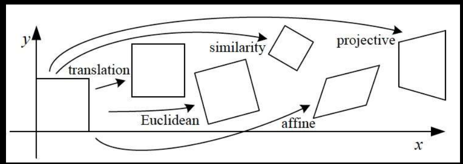




Example: translation

$x' = x + \vec{t}$ $\quad x' = \begin{bmatrix} I & \vec{t} \end{bmatrix} \bar{x}$

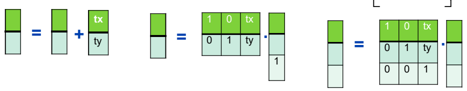




Homogenous vector

Example: translation

$x' = x + t$ $\quad$ $x' = \begin{bmatrix} I & t \end{bmatrix} \bar{x}$ $\quad$ $\bar{x}' = \begin{bmatrix} I & t \\ \mathbf{0}^T & 1 \end{bmatrix} \bar{x}$



[BTW: Now we can chain transformations]

Projective Transformations

- *Projective* transformations: for 2D images it's a 3x3 matrix applied to homogenous coordinates

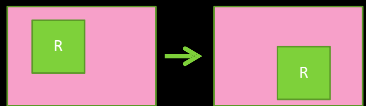
$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Special Projective Transformations

- Translation

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

- Preserves:
  - Lengths/Areas
  - Angles
  - Orientation
  - Lines

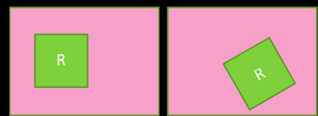


Special Projective Transformations

- Euclidean (Rigid body)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserves:
  - Lengths/Areas
  - Angles
  - Lines

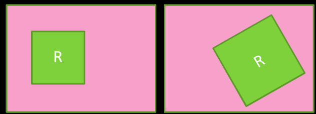


Special Projective Transformations

- Similarity (trans, rot, scale) transform

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -a\sin(\theta) & t_x \\ a\sin(\theta) & a\cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserves:
  - Ratios of Areas
  - Angles
  - Lines

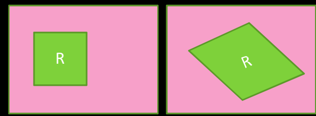


Special Projective Transformations

- Affine transform

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

- Preserves:
  - Parallel lines
  - Ratio of Areas
  - Lines



Projective Transformations

- Remember, these are homogeneous coordinates

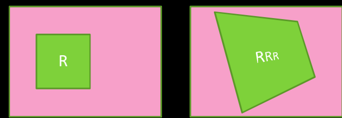
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \cong \begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Projective Transformations

- General projective transform (or Homography)

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

- Preserves:
  - Lines
  - Also cross ratios (maybe later)

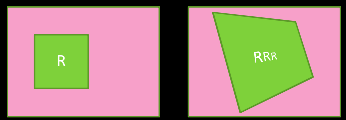


Projective Transformations

- General projective transform (or Homography)

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

- Preserves:
  - Lines
  - Also cross ratios (maybe later)



Quiz 1

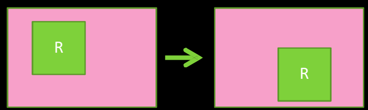
Suppose I told you the transform from image A to image B is a ***translation***. How many pairs of corresponding points would you need to know to compute the transformation?

a) 3
b) 1
c) 2
d) 4

Quiz 1 – answer

- Translation: a 1 point transformation

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



Quiz 2

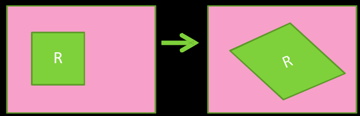
Suppose I told you the transform from image A to image B is ***affine***. How many pairs of corresponding points would you need to know to compute the transformation?

a) 3
b) 1
c) 2
d) 4

Quiz 2 – answer

- Affine transform: a 3 point transformation
  - 6 unknowns – each point pair gives two equations

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



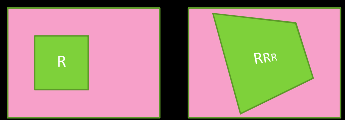
Quiz 3

Suppose I told you the transform from image A to image B is a ***homography***. How many pairs of corresponding points would you need to know to compute the transformation?

a) 3
b) 1
c) 2
d) 4

Quiz 3 – answer

- Homography:
  4 points



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

Aın 431
Introduction to Computer Vision

Homographies and mosaics

Projective Transformations

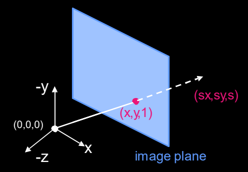
Projective transformations: for 2D images it's a 3x3 matrix applied to homogenous coordinates

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

The projective plane

What is the geometric intuition of using homogenous coordinates?

- A point in the image is a ray in projective space



The projective plane

Each *point* (x,y) on the plane (at z=1) is represented by a *ray* (sx,sy,s)

All points on the ray are equivalent:
(x, y, 1) ≅ (sx, sy, s)

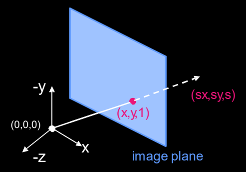
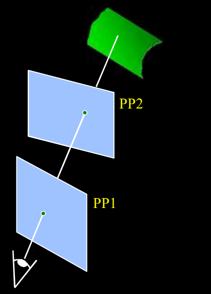


Image reprojection

Basic question:

How to relate two images from the same camera center?

How to map a pixel from projective plane PP1 to PP2?



Source: Alyosha Efros

Image reprojection

Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2

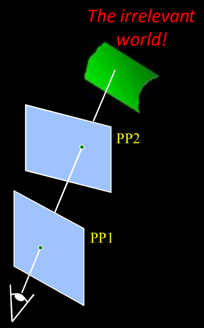
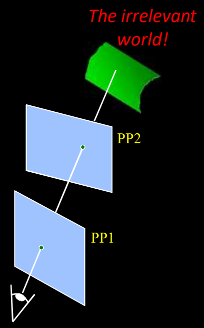


Image reprojection

Observation:

- Rather than thinking of this as a 3D reprojection, think of it as a 2D **image warp** from one image (plane) to another (plane).



Application: Simple mosaics

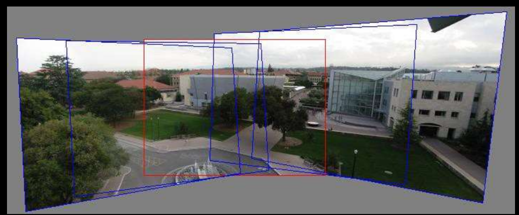


Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fall/

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

Basic Procedure

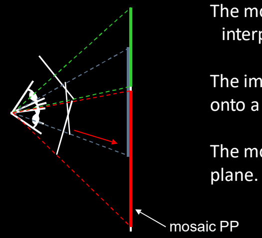
- Take a sequence of images from the same position
  - > Rotate the camera about its optical center
- Compute transformation between second image and first
- Transform the second image to overlap with the first
- Blend the two together to create a mosaic
- (If there are more images, repeat)

But wait…

Why should this work at all?

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

Image reprojection



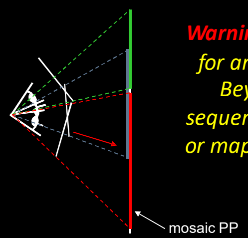
The mosaic has a natural interpretation in 3D:

The images are *reprojected* onto a common plane

The mosaic is formed on this plane.

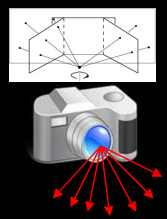
Source: Steve Seitz

Image reprojection



Warning: *This model only holds for angular views up to 180°. Beyond that need to use sequence that "bends the rays" or map onto a different surface, say, a cylinder.*

Mosaics



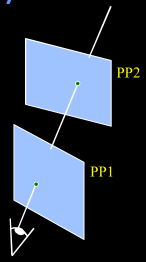
Obtain a wider angle view by combining multiple images ***all of which are taken from the same camera center***.

Image reprojection: Homography

A projective transform is a mapping between any two PPs with the same center of projection

- Lines map to lines
- So rectangle maps to arbitrary quadrilateral

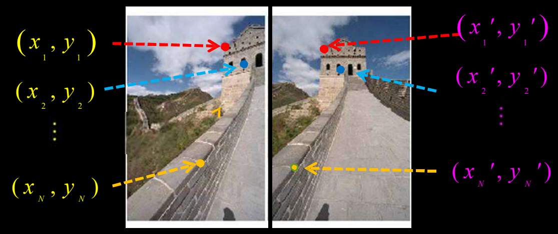
Called Homography



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

Source: Alyosha Efros

Homography



Solving for homographies

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

Solving for homographies – non-homogeneous

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

Since 8 unknowns, can set scale factor i=1.

Set up a system of linear equations $\mathbf{Ah} = \mathbf{b}$ where vector of unknowns

$$\mathrm{h} = [\,a,b,c,d,e,f,g,h,i\,]^T$$

Need at least 4 points for 8 eqs, but the more the better…

Solve for h by $\min \|\mathbf{Ah} - \mathrm{b}\|^2$ using least-squares

Solving for homographies – homogeneous

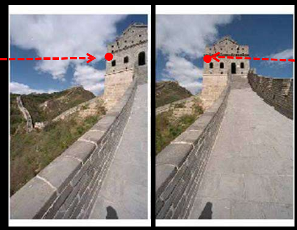
$$\mathbf{p'} = \mathbf{Hp} \qquad \begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Just like we did for the extrinsics, multiply through, and divide out by w. Gives two homogeneous equations per point.
Solve using SVD just like before. This is the cool way.

Apply the Homography

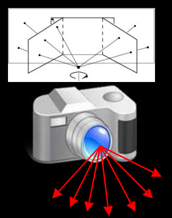
$$p' = \mathbf{H}\, p$$

$$(x, y)$$



$$\left({}^{wx'}\!/_{w},\ {}^{wy'}\!/_{w} \right) = (x', y')$$

Mosaics



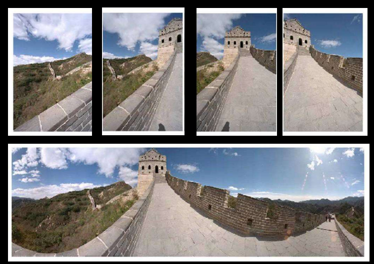


image from S. Seitz

$$p_i = \begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i x_i' & y_i x_i' & x_i' \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_i y_i' & y_i y_i' & y_i' \end{bmatrix}$$

$$PH = 0$$

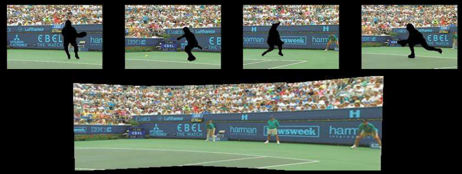
Such as:

$$PH = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 x_1' & y_1 x_1' & x_1' \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 y_1' & y_1 y_1' & y_1' \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 x_2' & y_2 x_2' & x_2' \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 y_2' & y_2 y_2' & y_2' \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3 x_3' & y_3 x_3' & x_3' \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3 y_3' & y_3 y_3' & y_3' \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4 x_4' & y_4 x_4' & x_4' \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4 y_4' & y_4 y_4' & y_4' \end{bmatrix} \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ h9 \end{bmatrix} = 0$$

SVD $P = USV^\top$ and select the last singular vector of V as the solution to H.

Mosaics for Video Coding

- Convert masked images into a background sprite for “content-based coding”



Quiz

We said that the transformation between two images taken from the same center of projection is a *homography* H. How many pairs of corresponding points do I need to compute H?

a) 6
b) 4
c) 2
d) 8

Quiz – answer

We said that the transformation between two images taken from the same center of projection is a *homography* H. How many pairs of corresponding points do I need to compute H?

a) 6
b) 4
c) 2
d) 8

Homographies and 3D planes

Remember this:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Homographies and 3D planes

- Suppose the 3D points are on a plane:

$$aX + bY + cZ + d = 0$$

Homographies and 3D planes

- On the plane *[a b c d]* can replace Z:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ (aX + bY + d)/(-c) \\ 1 \end{bmatrix}$$

Homographies and 3D planes

- So, can put the Z coefficients into the others:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} m'_{00} & m'_{01} & 0 & m'_{03} \\ m'_{10} & m'_{11} & 0 & m'_{13} \\ m'_{20} & m'_{21} & 0 & m'_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ (aX + bY + d)/(-c) \\ 1 \end{bmatrix}$$

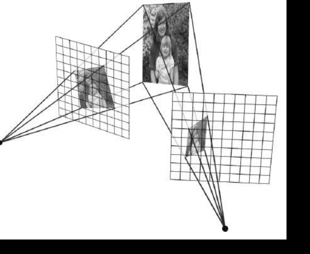
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ -\frac{aX+bY+d}{c} \\ 1 \end{bmatrix}$$

3x3 *Homography!*

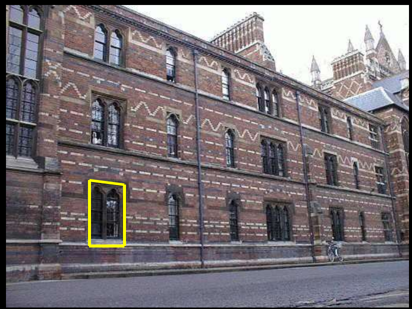
$$H = P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{a}{c} & -\frac{b}{c} & -\frac{d}{c} \\ 0 & 0 & 1 \end{bmatrix}$$

Image reprojection

- Mapping between planes is a homography.
- Whether a plane in the world to the image or between image planes.



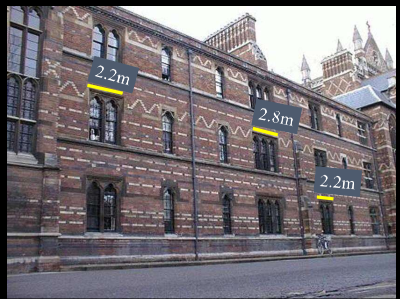
Rectifying slanted views



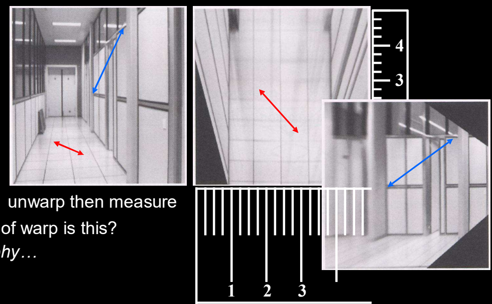
Rectifying slanted views

Corrected image (front-to-parallel)

Measuring distances



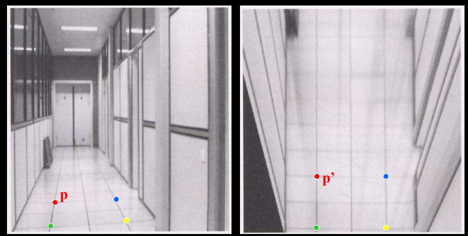
Measurements on planes



Approach: unwarp then measure
What kind of warp is this?
Homography…

Image rectification

If there is a planar rectangular grid in the scene you can map it into a rectangular grid in the image…



Some other images of rectangular grids…

Same pixels – via a homography

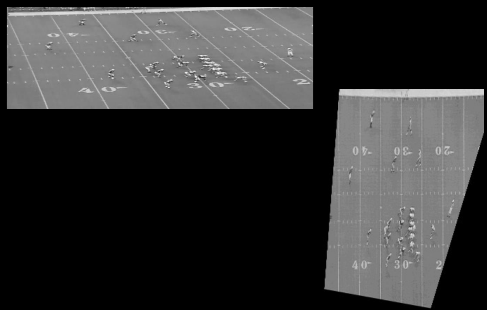
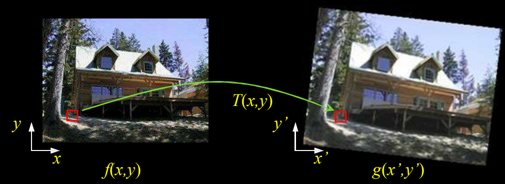


Image warping

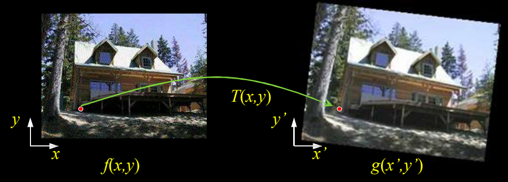
Given a coordinate transform and a source image *f(x,y)*, how do we compute a transformed image *g(x',y') = f(T(x,y))*?



Slide from Alyosha Efros,

Forward warping

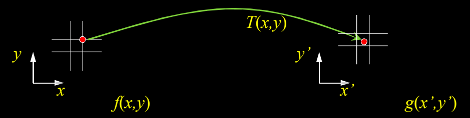
Send each pixel $f(x,y)$ to its corresponding location
$(x',y') = T(x,y)$ in the second image



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

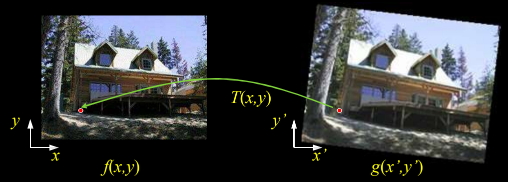
Forward warping

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



Inverse warping

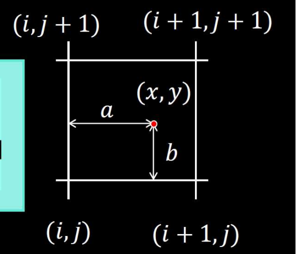
Get each pixel $g(x',y')$ from its corresponding location
$(x,y) = T^{-1}(x',y')$ in the first image



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

Bilinear interpolation

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



See Matlab (Octave) function **`interp2`**

Review: How to make a panorama (or 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)