CS664 Computer Vision

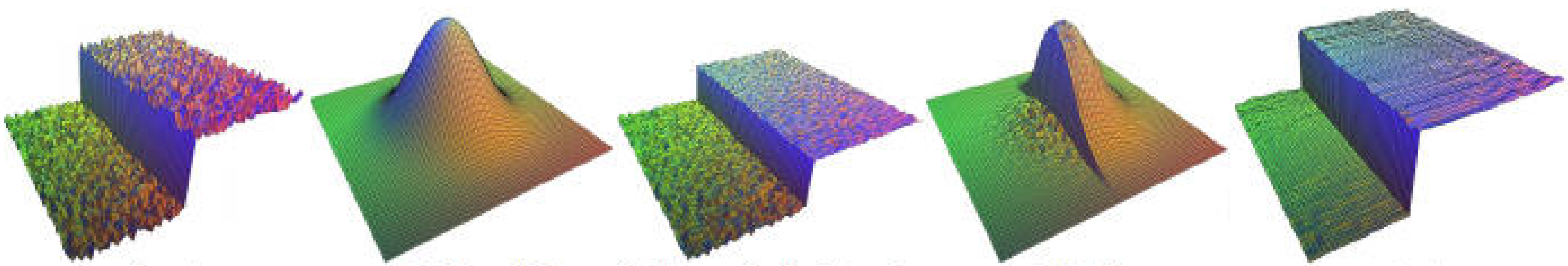
5. Image Geometry

Dan Huttenlocher

Cornell University
Faculty of Computing and Information Science

First Assignment

- Wells paper handout for question 1
- Question 2 more open ended
  - Less accurate approximations
    - Simple box filtering doesn't work
  - Anisotropic, spatially dependent

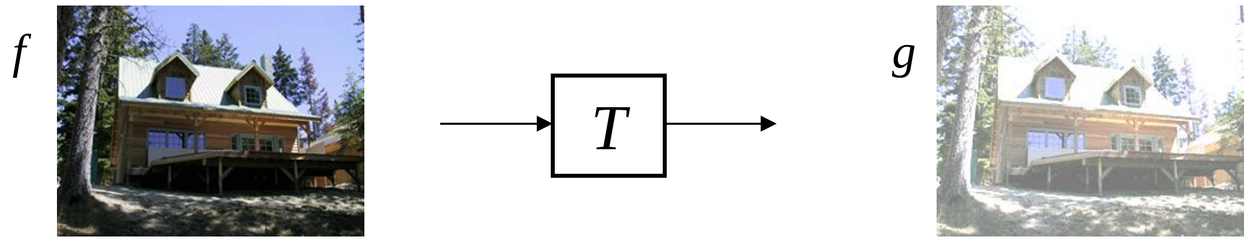


input | spatial kernel f | influence g in the intensity domain for the central pixel | weight $f \times g$ for the central pixel | output

Image Warping

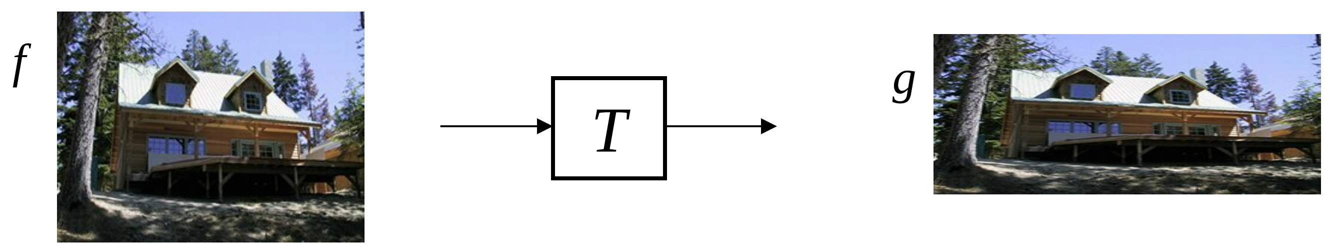
- Image filtering: change *range* of image

$$g(x) = T \star f(x)$$



- Image warping: change *domain* of image

$$g(x) = f(T(x))$$



Feature Detection in Images

- Filtering to provide area of support
  - Gaussian, bilateral, ...
- Measures of local image difference
  - Edges, corners
- More sophisticated features are invariant to certain transformations or warps of the image
  - E.g., as occur when viewing direction changes

Parametric (Global) Warping

- Examples of parametric warps:

translation

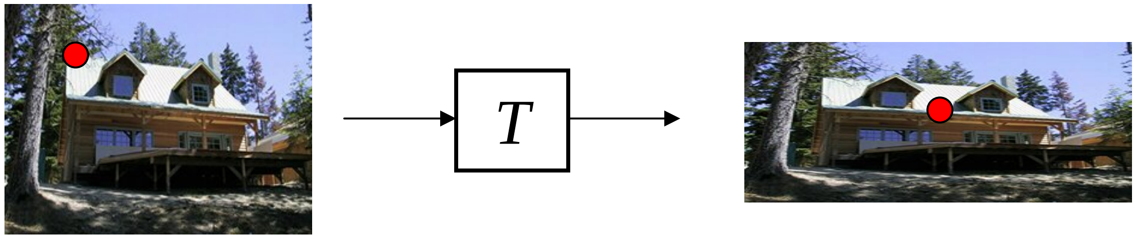
rotation

affine

projective

cylindrical

Parametric (Global) Warping



p = (x,y) **p’** = (x’,y’)

$$p' = T(p)$$

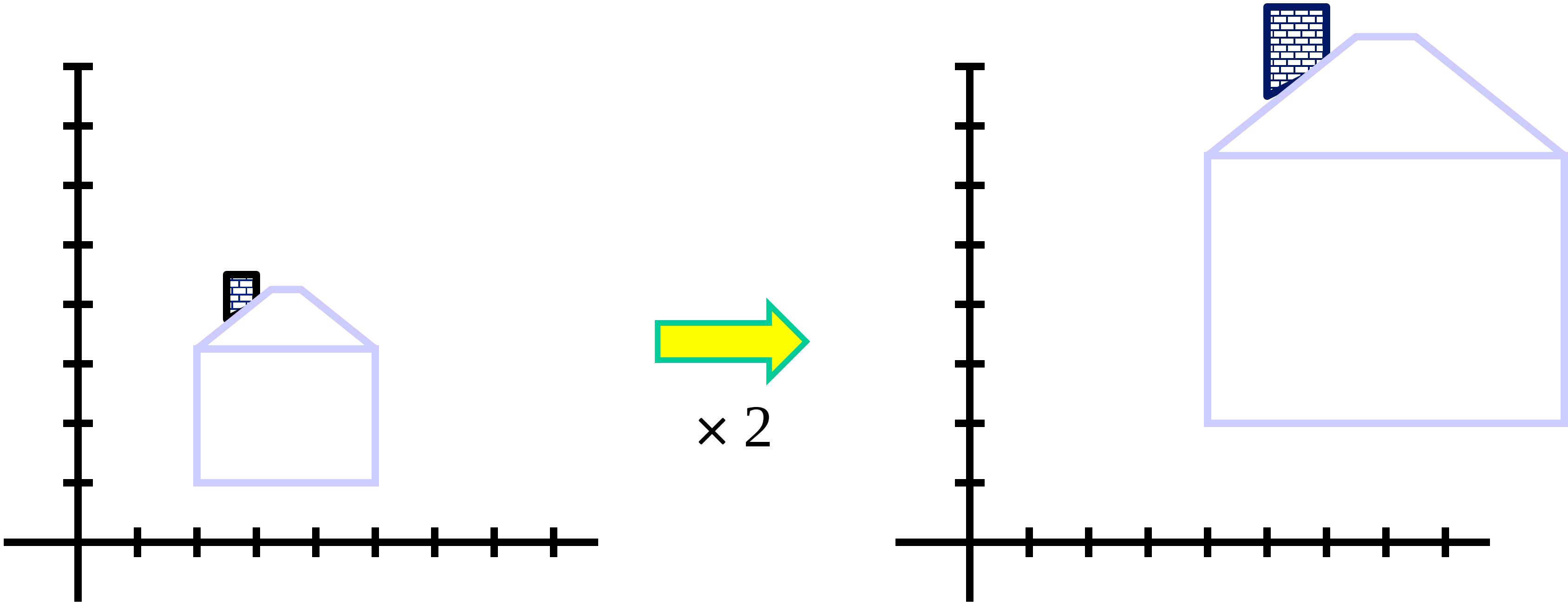
- What does it mean that *T* is global?
  - Same function for any point p
  - Described by a few parameters, often matrix

p’ = **M***p

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

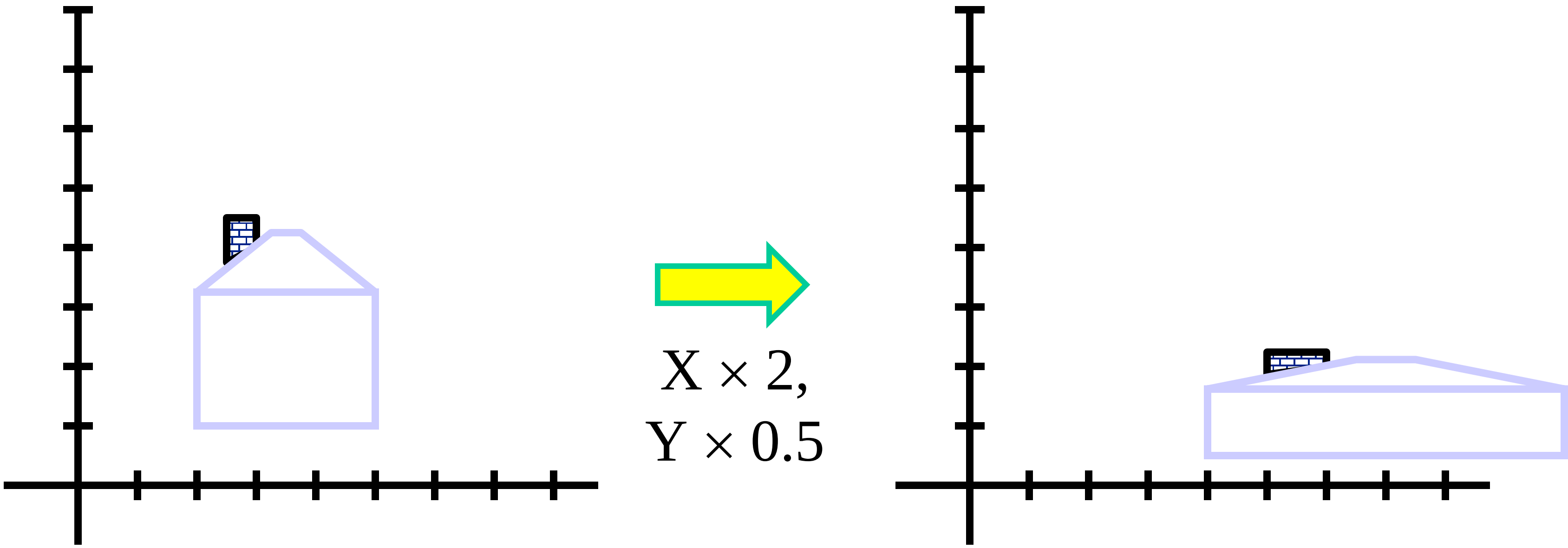
Scaling

- *Scaling* a coordinate means multiplying each of its components (axes) by a scalar
- *Uniform scaling* means this scalar is the same for all components:



Scaling

- *Non-uniform scaling*: different scalars per component:



Scaling

- Scaling operation:

$$x' = ax$$

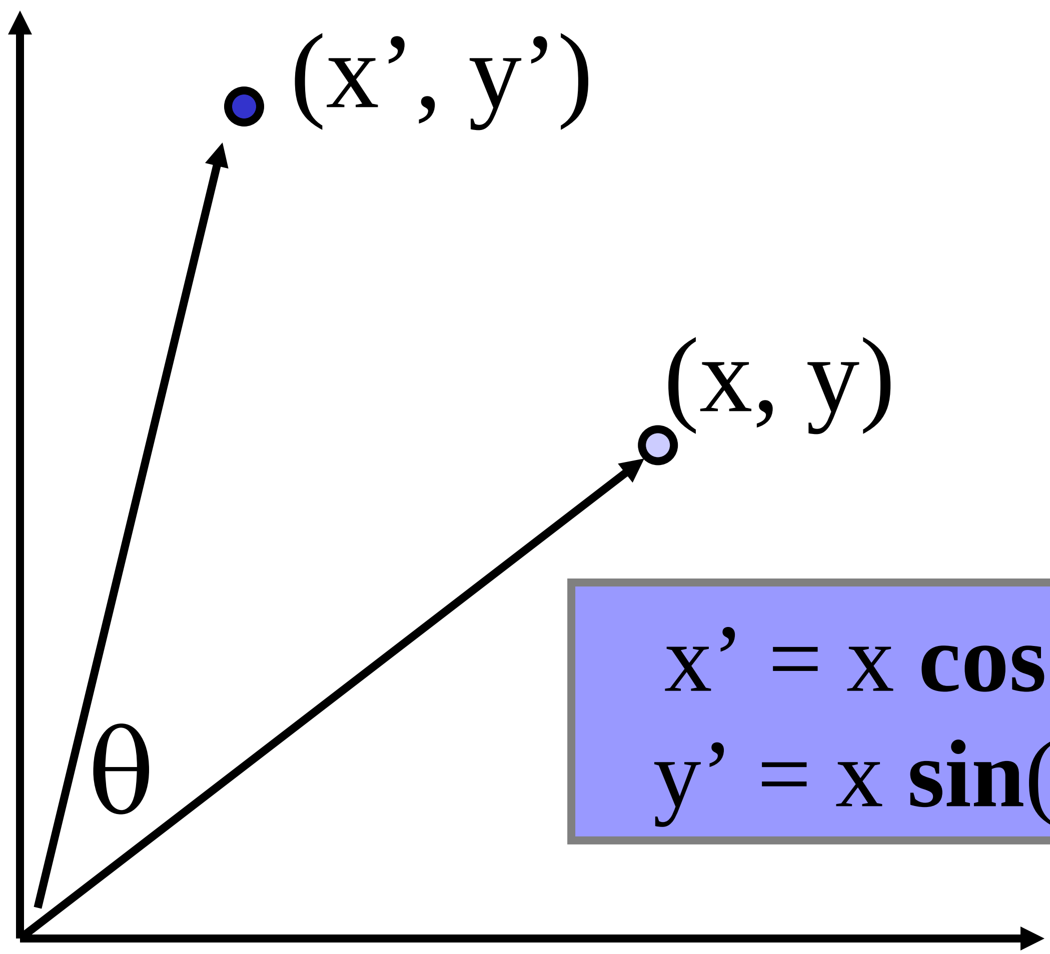
$$y' = by$$

- Or, in matrix form:

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

What's inverse of S?

2-D Rotation



About origin

$$x' = x\,\mathbf{cos}(\theta) - y\,\mathbf{sin}(\theta)$$
$$y' = x\,\mathbf{sin}(\theta) + y\,\mathbf{cos}(\theta)$$

2-D Rotation

- This is easy to capture in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Even though sin(θ) and cos(θ) are nonlinear functions of θ,
  - x’ is a linear combination of x and y
  - y’ is a linear combination of x and y
- Inverse transformation, rotation by –θ
  - For rotation matrices, det(R) = 1 and $\mathbf{R}^{-1} = \mathbf{R}^{T}$

Cornell University

2x2 Transformation Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Identity? (A rotation)

$$x' = x \qquad y' = y$$

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

2D Scale around (0,0)?

$$\boldsymbol{x}' = \boldsymbol{s}_x * \boldsymbol{x} \qquad \boldsymbol{y}' = \boldsymbol{s}_y * \boldsymbol{y}$$

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

2x2 Transformation Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Rotate around (0,0)?

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

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

2D Shear?

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

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

2x2 Transformation Matrices

- What types of 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)?

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

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

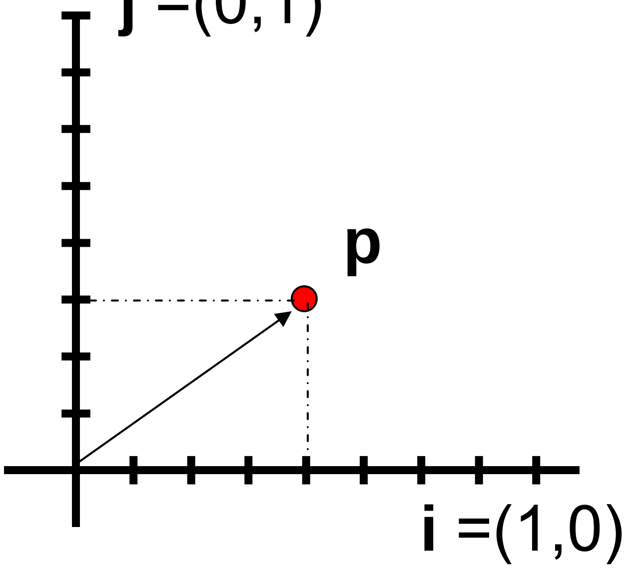
All 2D Linear Transformations

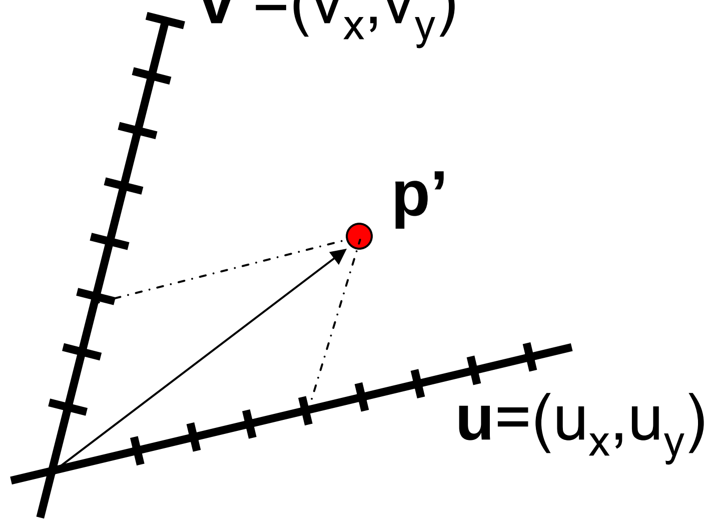
- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror
- Properties of linear transformations:
  - Origin maps to origin
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved
  - Closed under composition

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

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

Linear Transformations as Change of Basis





p=4**i**+3**j** = (4,3)

p'=4**u**+3**v**

$p_x'=4u_x+3v_x$
$p_y'=4u_y+3v_y$

$$\mathbf{p}' = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \mathbf{p}$$

- Any linear transformation is a basis!

2x2 Transformation Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Translation?

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

NO!

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

Homogeneous Coordinates

- How can we represent translation as matrix?

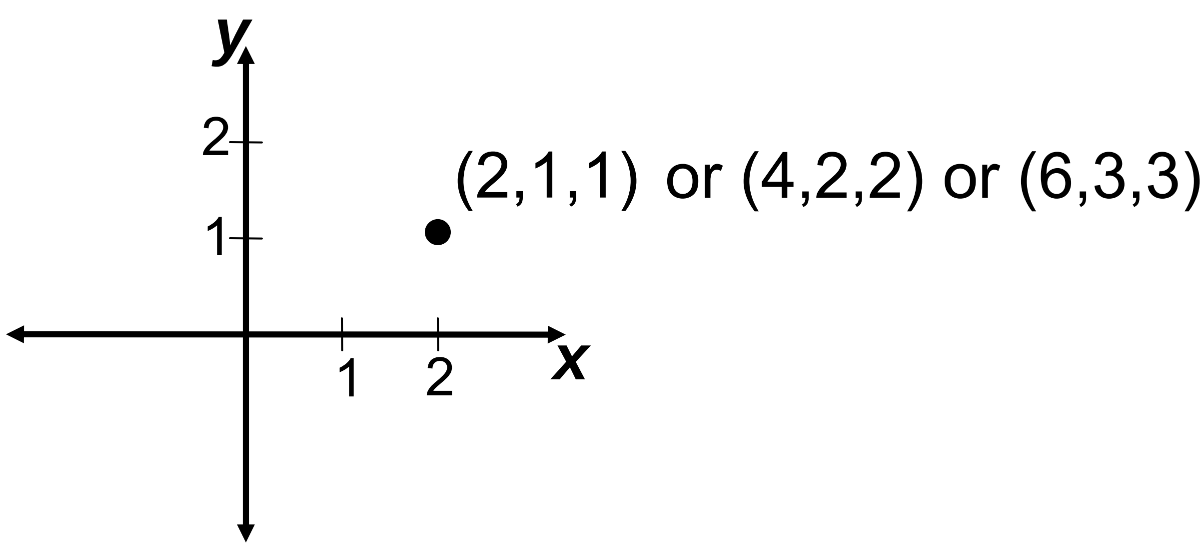
$$x' = x + t_x$$
$$y' = y + t_y$$

- Homogeneous coordinates
  - Represent coordinates in 2 dimensions with a 3-vector

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Add a 3rd coordinate to every 2D point
  - (x, y, w) represents a point at 2D location (x/w, y/w)
  - (x, y, 0) represents a point at infinity
  - (0, 0, 0) is not allowed



Homogeneous Coordinates

- How can we represent translation as matrix?

$$x' = x + t_x$$

$$y' = y + t_y$$

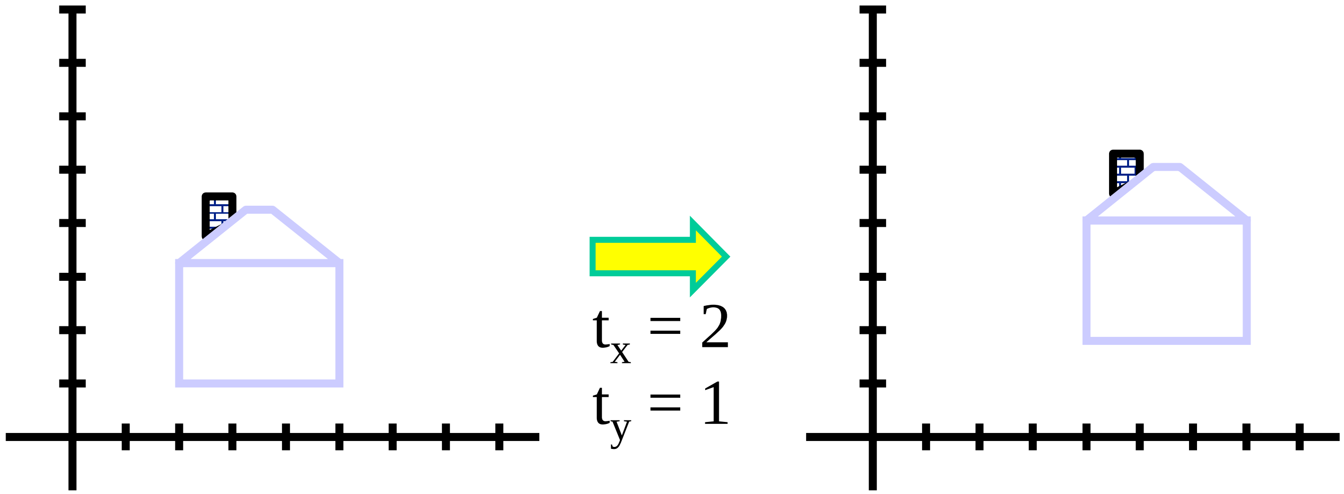
- Last column of homogeneous matrix

$$\mathbf{T}ranslation = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Translation

- Example of translation in homogeneous coordinates

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



Homogeneous 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

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

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

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

Shear

Affine Transformations

- Affine transformations are …
  - Linear transformations, and
  - Translations

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

- Properties of affine transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved
  - Closed under composition
  - Models change of basis
  - Maps any triangle to any triangle (or parallelogram)

Projective Transformations

- Projective transformations …
  - Affine transformations, and
  - Projective warps

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

- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition
  - Models change of basis
  - Maps any quadrilateral to any quadrilateral

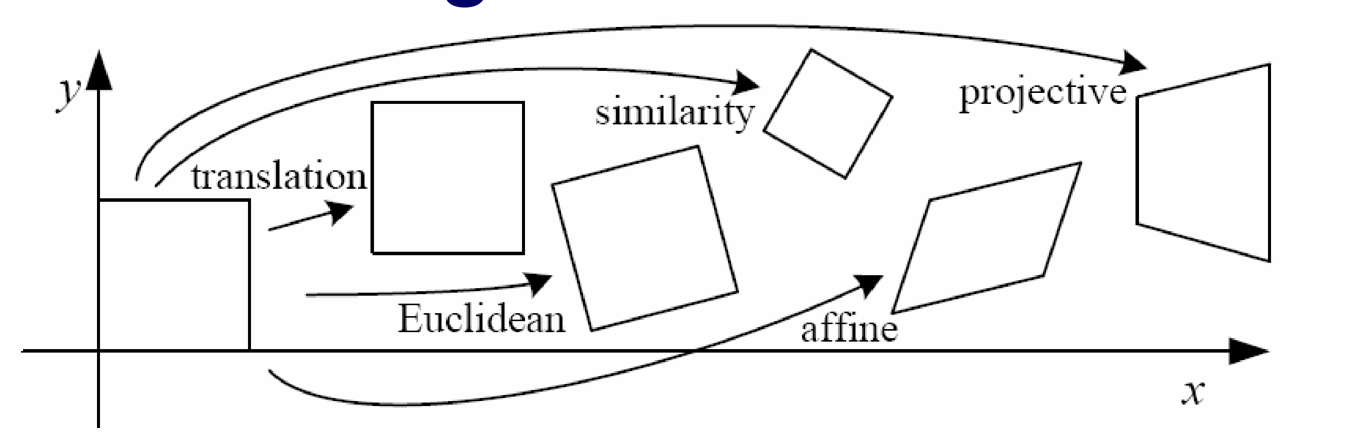
Matrix Composition

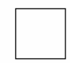
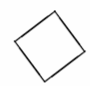
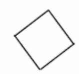
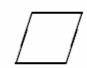
- Transformations can be combined (composed) by matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

p' = $T(t_x,t_y)$ $R(\Theta)$ $S(s_x,s_y)$ **p**

2D image transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}$			
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}$			
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}$			
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}$			
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}$			

- Nested set of groups

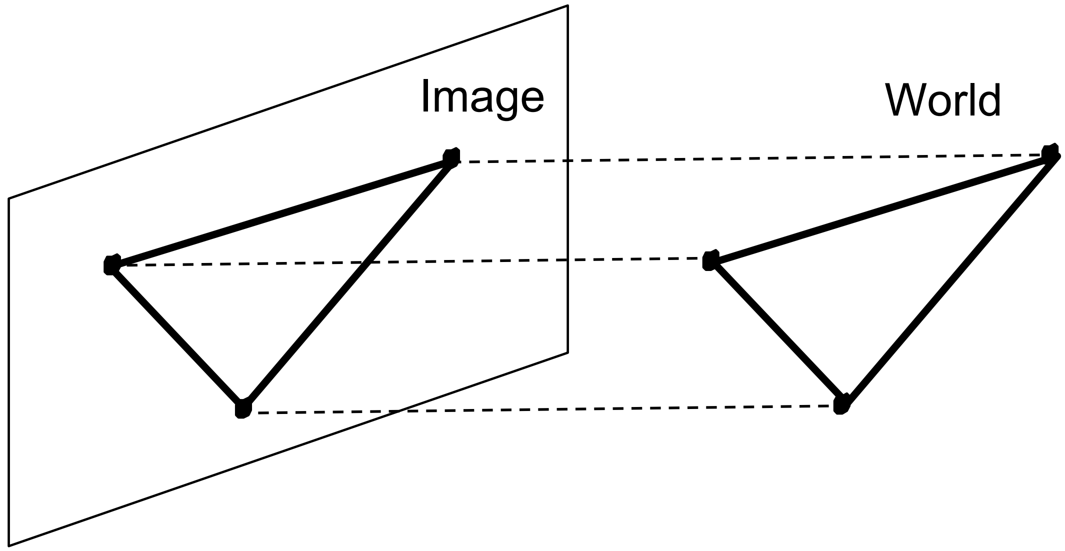
Affine Invariants

- Affine transformations preserve collinearity of points:
  - If 3 points lie on the same line, their images under affine transformations also line on the same line and,
  - The middle point remains between the other two points
- Concurrent lines remain concurrent (images of intersecting lines intersect),
- Ratio of length of line segments of a given line remains constant
- Ratio of areas of two triangles remains constant,
- Ellipses remain ellipses and the same is true for parabolas and hyperbolas

3D Interpretations

- Orthographic projection plus scaling of Euclidean motion in 3D world yields 2D affine transformation
  - Often called weak perspective imaging model
- Set of coplanar points in 3D
  - Undergo rigid body motion
  - Scaling – analogous to perspective size with distance but same scaling for all the points
  - Projection into image plane (drop coordinate)

Orthographic Projection



- Also called "parallel projection": $(x, y, z) \rightarrow (x, y)$
- What's the projection matrix (homogeneous coords)?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Weak Perspective Imaging

- Scaling in addition to parallel projection

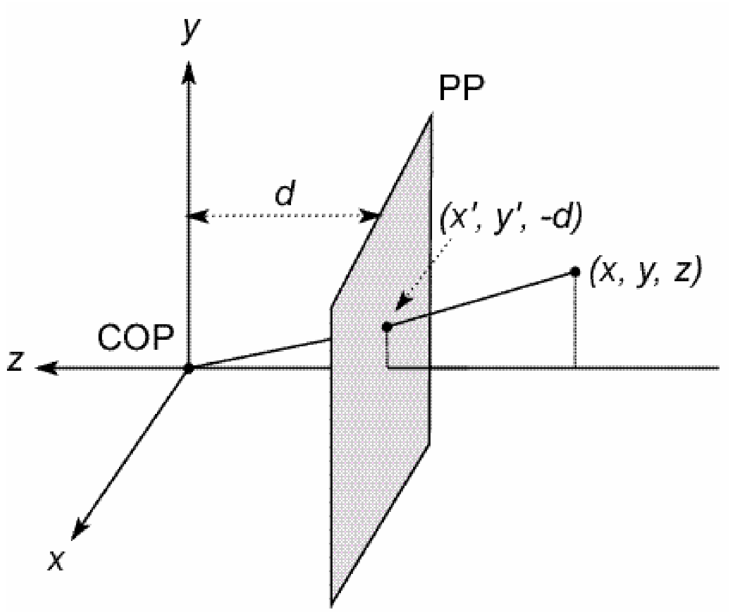
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/d \end{bmatrix} \Rightarrow (dx, dy)$$

- Composed with 3D rigid body motion (6 dof)

$$T = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & x_t \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & y_t \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- Equivalent to affine transformation of plane (6 dof) up to reflection – state without proof

Projection – Pinhole Camera Model



$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z},\ -d)$$

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$

- Projection equations
  - Compute intersection with PP of ray from (x,y,z) to COP
  - Derived using similar triangles
  - Parallel projection where d infinite

Perspective Projection

- Homogeneous coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \quad -d\frac{y}{z})$$

Divide by third coordinate

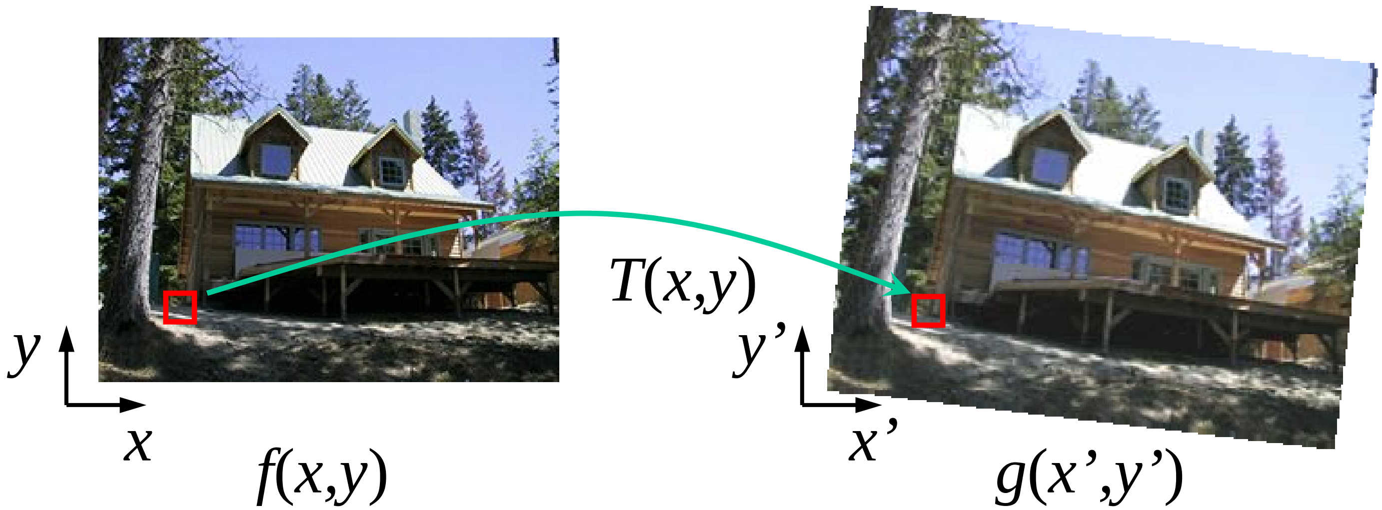
$$\begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -dx \\ -dy \\ z \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \quad -d\frac{y}{z})$$

Scaling projection matrix, no effect

Perspective Projection

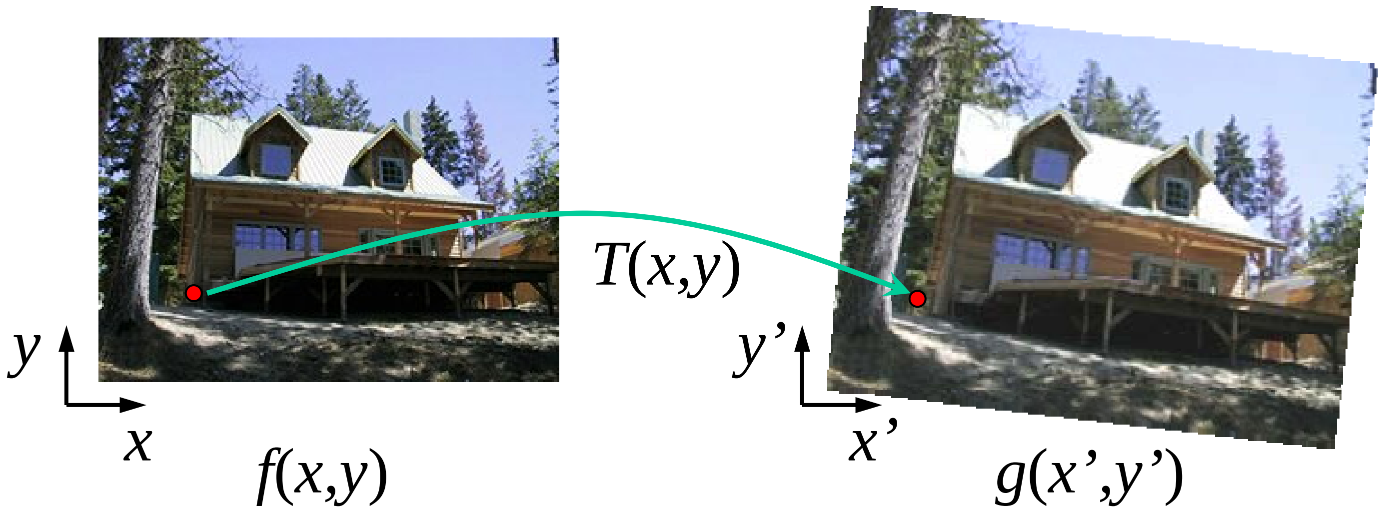
- Composed with 3D rigid body motion (6 dof)
- Focal length, d, and distance z two additional parameters
- Equivalent to projective transformation of plane (homography)
  - 3x3 matrix in homogeneous coordinates, 8 dof
  - Again state without proof
- 2D affine and projective transformations correspond to images of plane in space under rigid motion, different imaging models

Image Warping



- Given coordinate transform $(x',y') = T(x,y)$ and source image $f(x,y)$
- How do we compute transformed image $g(x',y') = f(T(x,y))$?

Forward Warping

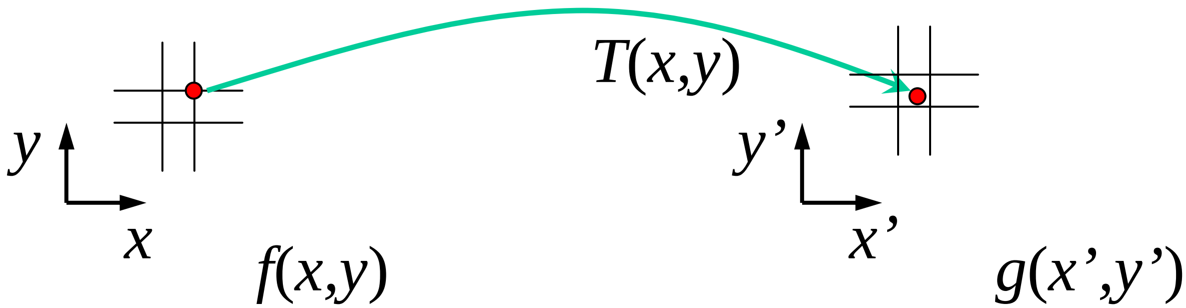


- Send each pixel $f(x,y)$ to its corresponding location

  $(x',y') = T(x,y)$ in the second image

Q: What about when pixel lands "between" two pixels?

Forward Warping



Points at pixel centers

- Send each pixel $f(x,y)$ to its corresponding location

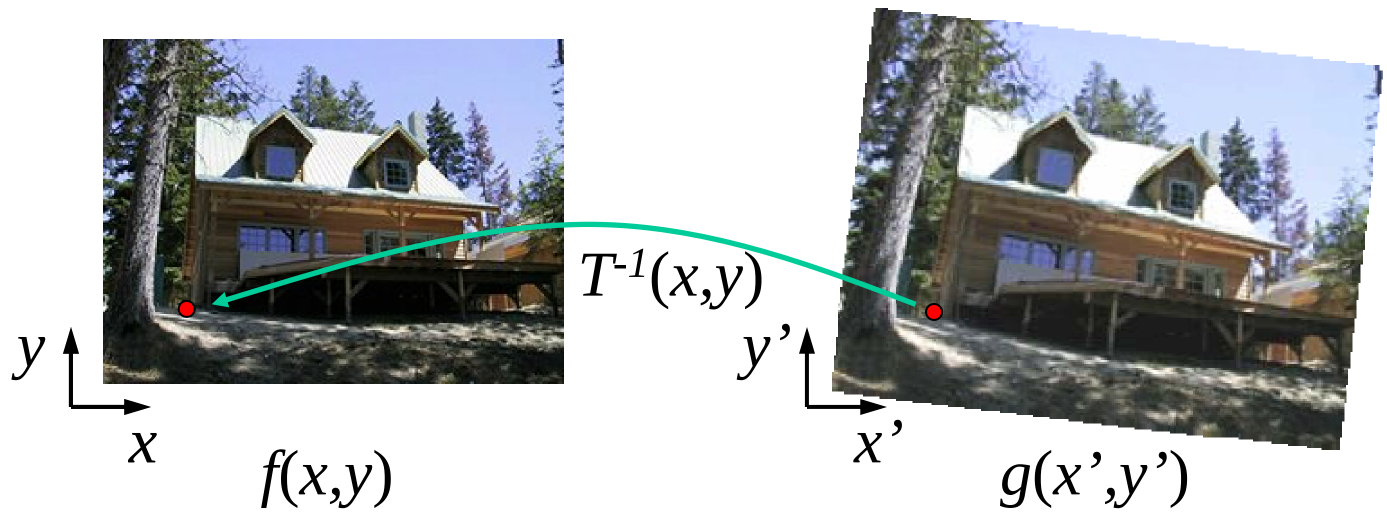
$$(x', y') = T(x,y) \text{ in the second image}$$

Q: What about when pixel lands "between" two pixels?

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

- – Known as "splatting"

Inverse Warping

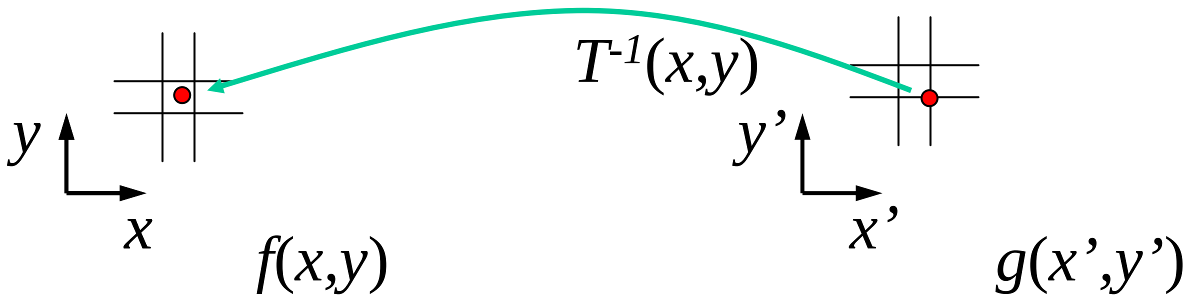


- Get each pixel $g(x',y')$ from its corresponding location

  $(x,y) = T^{-1}(x',y')$ in the first image

Q: What about when pixel comes from "between" two pixels?

Inverse Warping



- Get each pixel $g(x',y')$ from its corresponding location

  $(x,y) = T^{-1}(x',y')$ in the first image

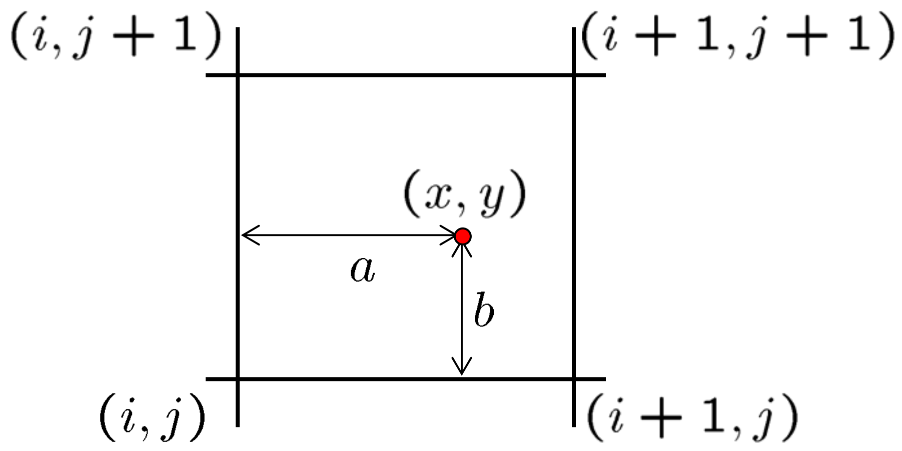
Q: What about when pixel comes from "between" two pixels?

A: *Interpolate* color value from neighbors

- Nearest neighbor, bilinear, Gaussian, bicubic

Bilinear Interpolation (Reminder)

- Sampling at *f(x,y):*



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

Forward vs. Inverse Warping

- Q: Which is better?
- A: Usually inverse – eliminates holes
  - However, requires an invertible warp function – not always possible...

affine

projective

Reference

- R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Ed., Cambridge Univ. Press, 2003.