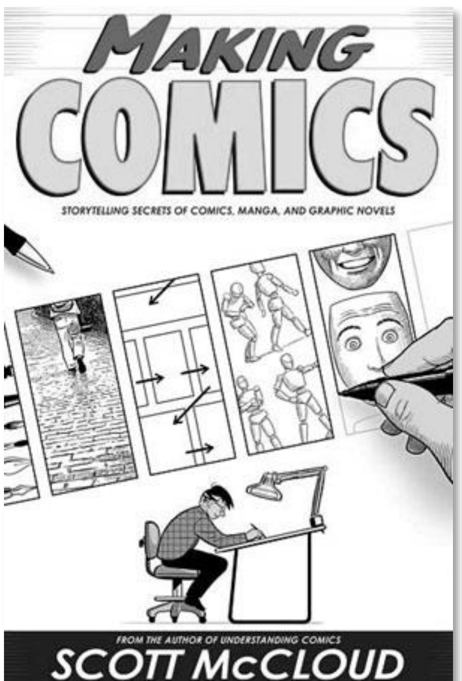
# Transformations

# What is the geometric relationship between these two images?

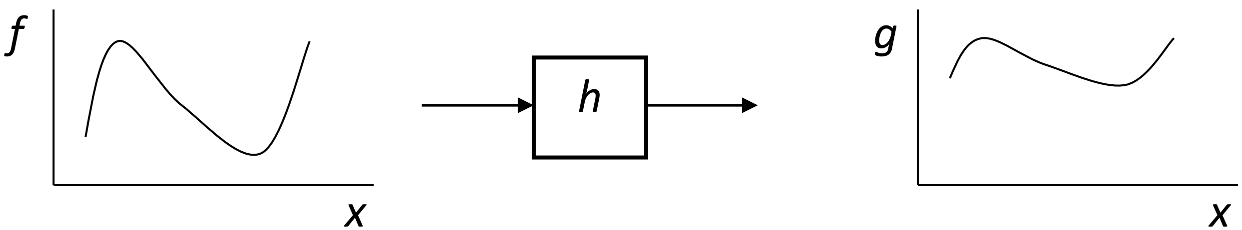



?

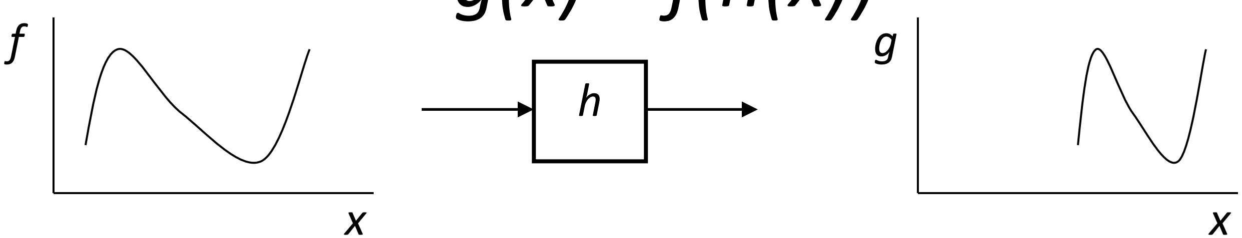
**Answer: Similarity transformation** (translation, rotation, uniform scale)

# Image Warping

- image filtering: change *range* of image
  - $g(x) = h(f(x))$



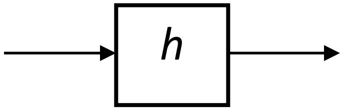

- image warping: change *domain* of image
  - $g(x) = f(h(x))$



# Image Warping

- image filtering: change *range* of image
  - $g(x) = h(f(x))$

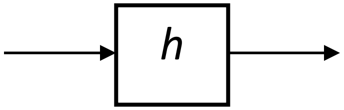
*f*




*g*

- image warping: change *domain* of image
  - $g(x) = f(h(x))$

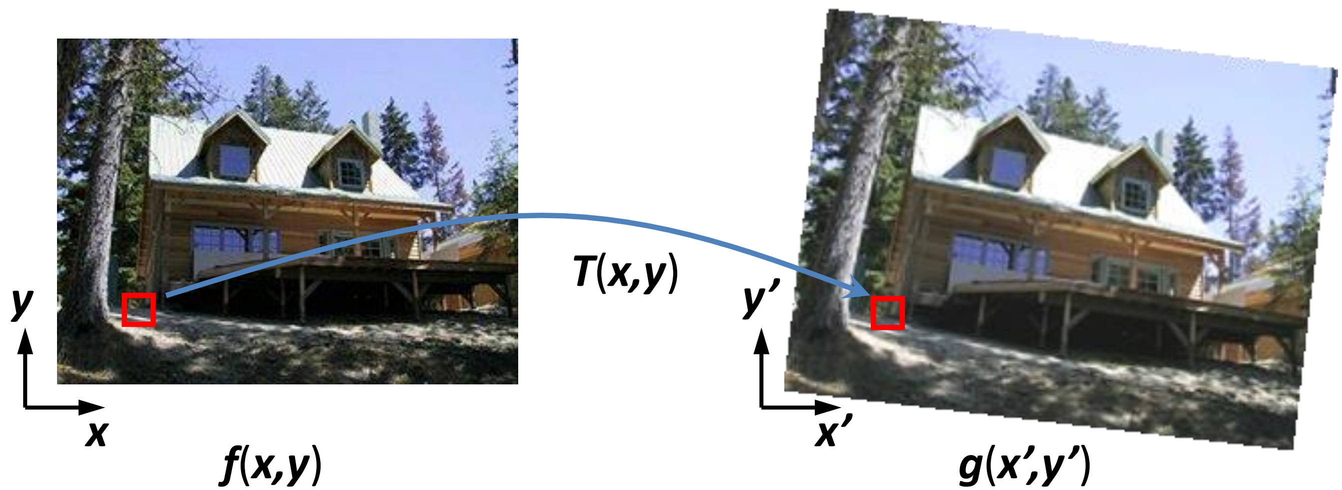
*f*




*g*

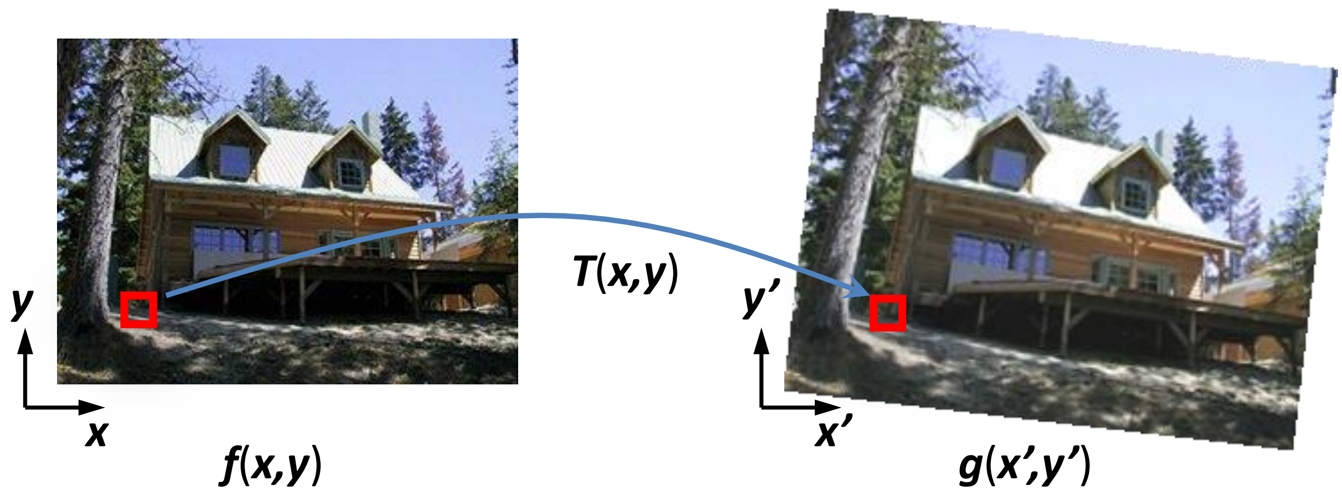
# Image Warping

- Given a coordinate xform (***x'***,***y'***) = ***T***(***x***,***y***) and a source image ***f***(***x***,***y***), how do we compute an xformed image?



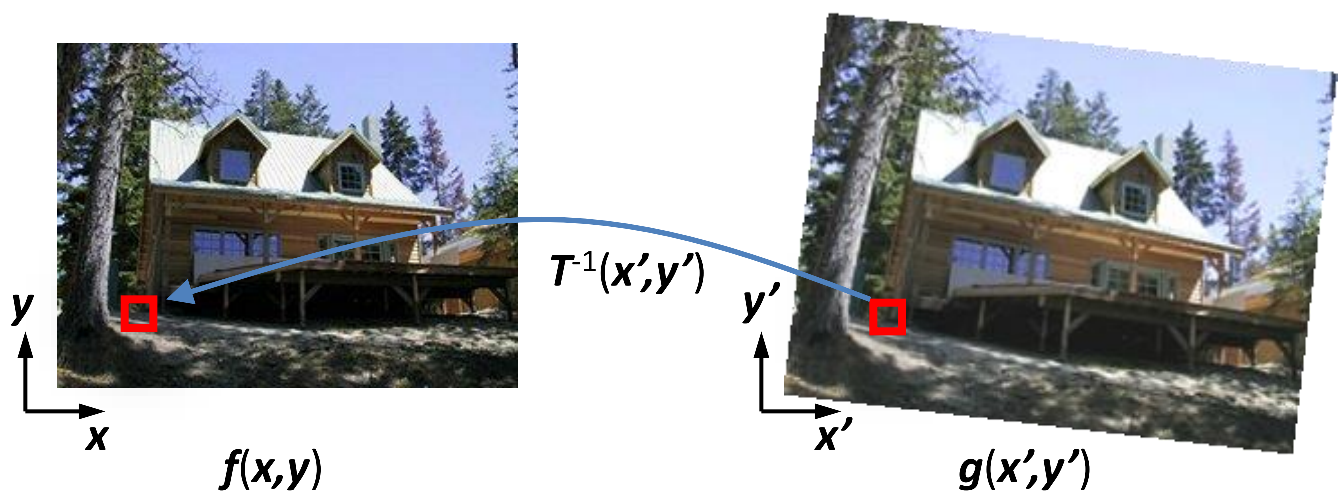
# Forward Warping

- Send each pixel $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})$ to its corresponding location $(\boldsymbol{x}',\boldsymbol{y}') = \boldsymbol{T}(\boldsymbol{x},\boldsymbol{y})$ in $\boldsymbol{g}(\boldsymbol{x}',\boldsymbol{y}')$

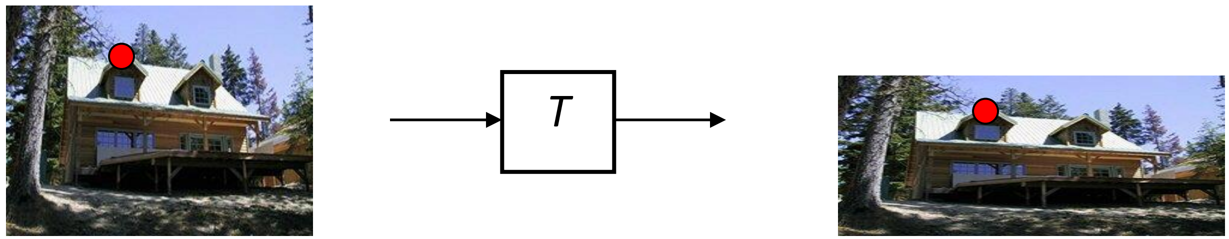


# Inverse Warping

- Get each pixel $g(x',y')$ from its corresponding location $(x,y) = T^{-1}(x',y')$ in $f(x,y)$
  - Requires taking the inverse of the transform



# Parametric (global) warping




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

- Transformation T is a coordinate-changing machine:

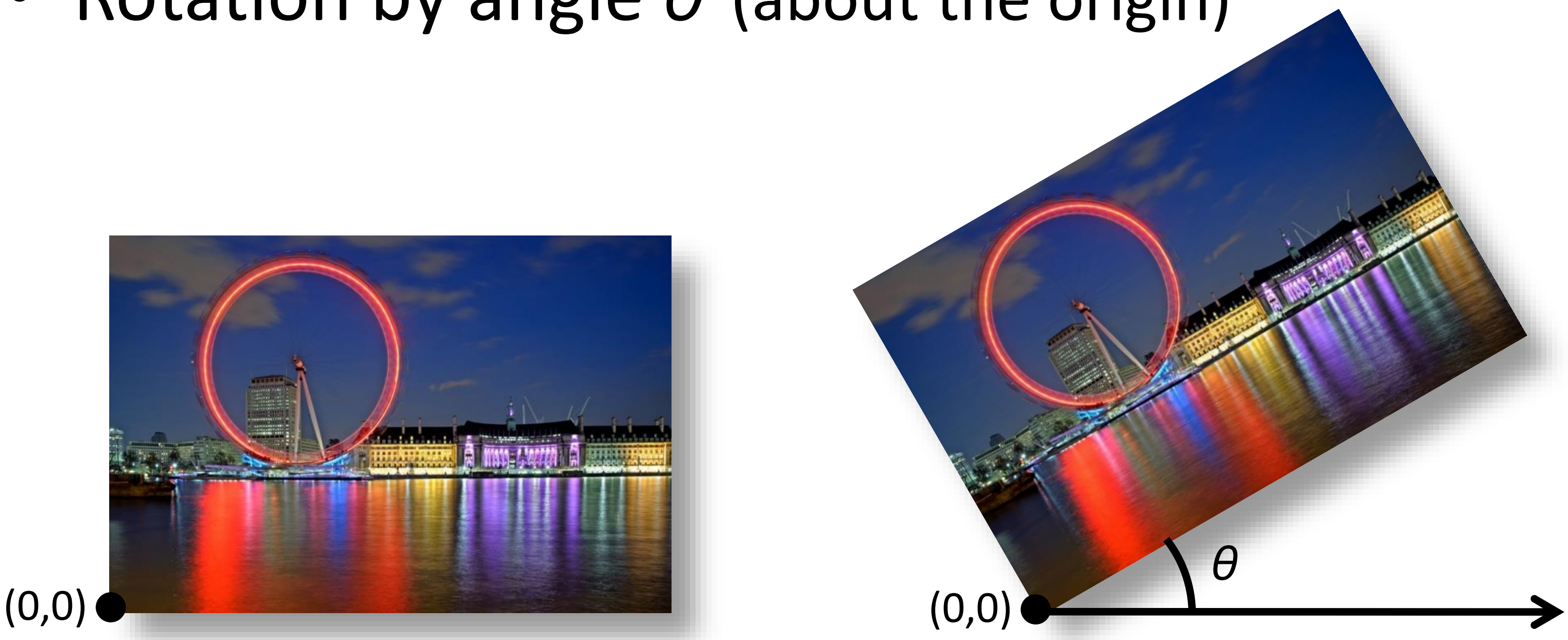
  p' = *T*(p)

- 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 consider *linear* xforms (can be represented by a 2D matrix):

$$\mathbf{p}' = \mathbf{T}\mathbf{p} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix}$$

# Common linear transformations

- Rotation by angle $\theta$ (about the origin)




$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

# 2x2 Matrices

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

2D mirror about Y axis?

$$\begin{aligned} x' &= -x \\ y' &= y \end{aligned} \qquad \mathbf{T} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

2D mirror across line y = x?

$$\begin{aligned} x' &= y \\ y' &= x \end{aligned} \qquad \mathbf{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

# 2x2 Matrices

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

2D Translation?

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}$$

NO!

# All 2D Linear Transformations

- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror

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

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

How can we represent translation as a 3x3 matrix?

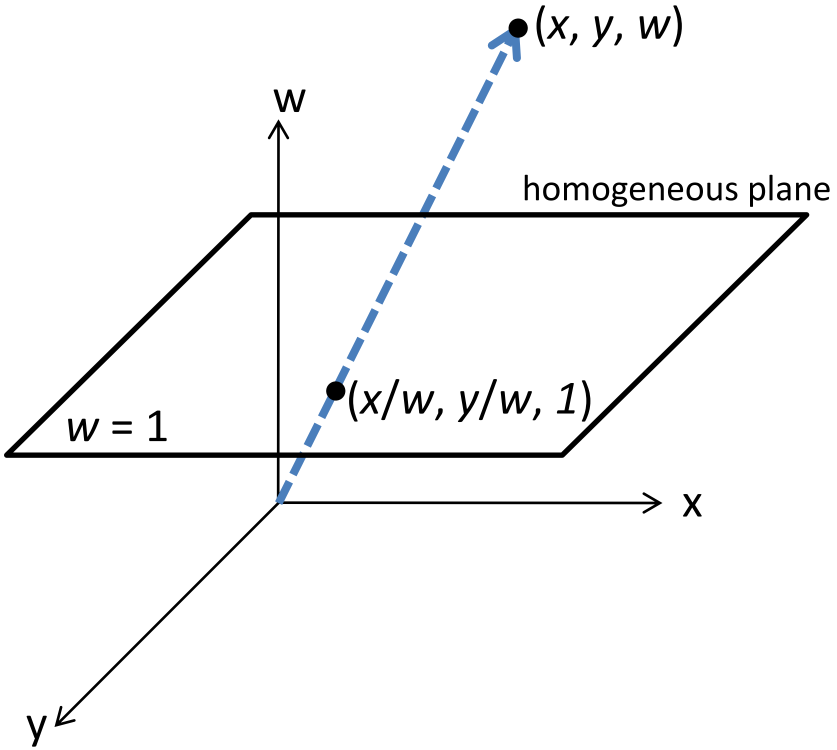
$$x' = x + t_x$$
$$y' = y + t_y$$

# Homogeneous coordinates

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates




Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

# Quiz

- Which one is true about the homogeneous coordinate?
  - The points (2,3,1) and (2,3,2) refer to the same point in image coordinate
  - Homogeneous coordinate could represent translation
  - The third dimension of the homogeneous coordinate must be greater or equal to 1

# Translation

- Solution: homogeneous coordinates to the rescue

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

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

# Affine transformations

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

any transformation with last row [ 0 0 1 ] we call an *affine* transformation

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

# Affine Transformations

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

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

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

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

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

2D *in-plane* rotation

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

Shear

# Where do we go from here?

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

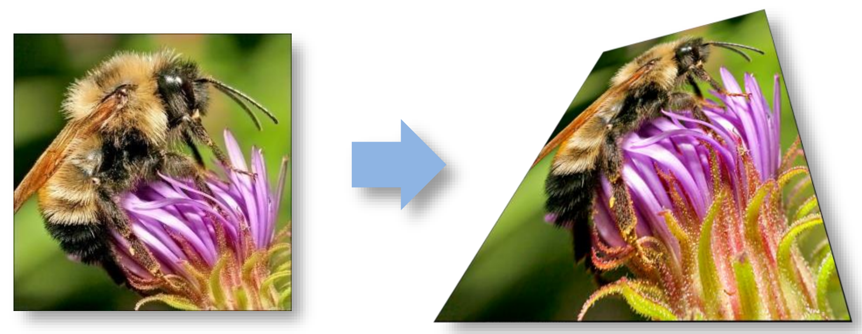
affine transformation

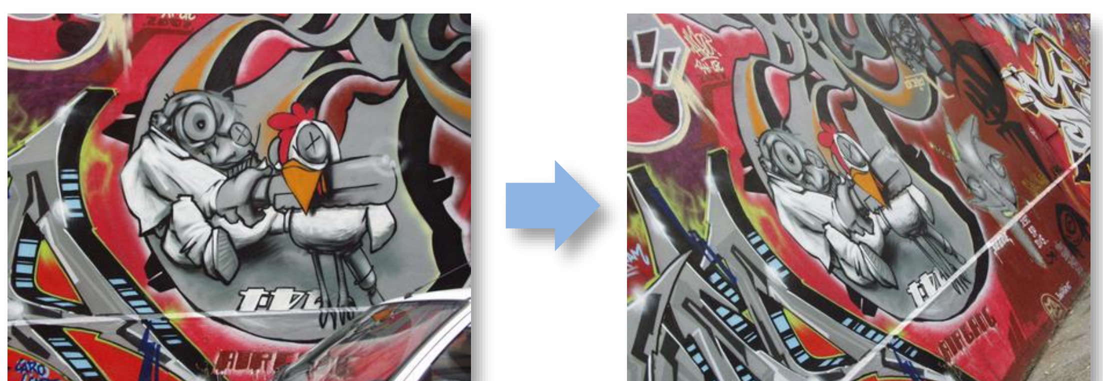
what happens when we mess with this row?

# Projective Transformations aka Homographies

$$\mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

Called a *homography*





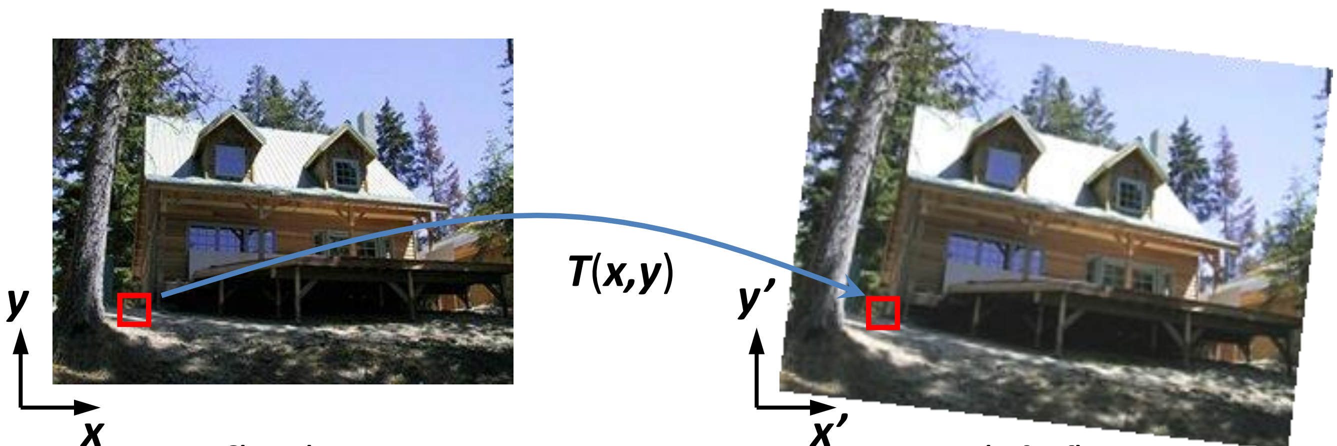
# Homographies

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

$$\sim \begin{bmatrix} \frac{ax+by+c}{gx+hy+1} \\ \frac{dx+ey+f}{gx+hy+1} \\ 1 \end{bmatrix}$$

# Image Warping

- Given a coordinate xform (***x'***,***y'***) = ***T***(***x***,***y***) and a source image ***f***(***x***,***y***), how do we compute an xformed image?

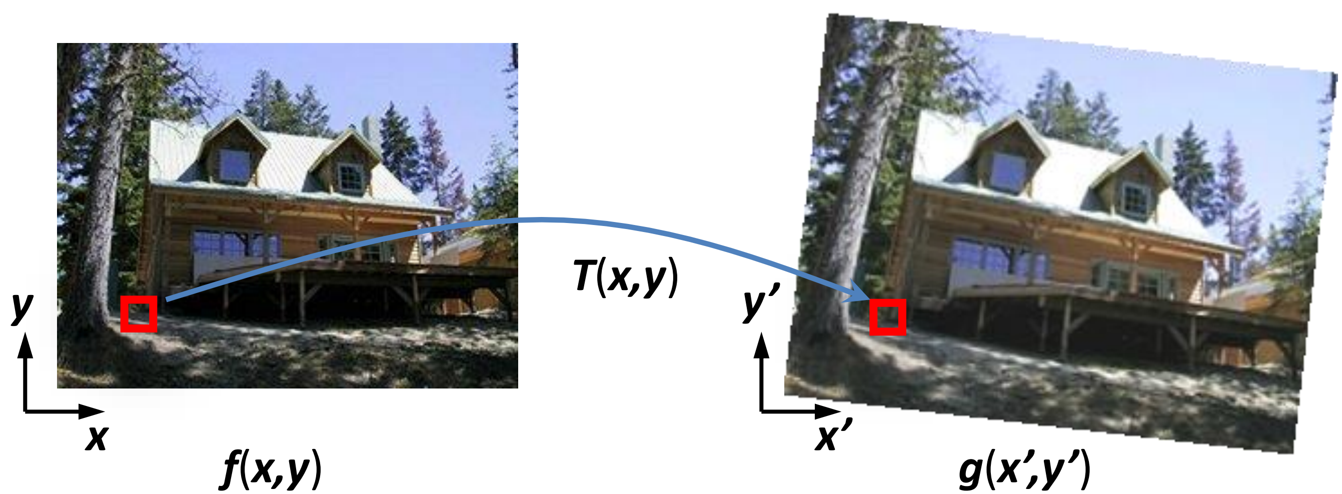



***f***(***x***,***y***)

***g***(***x'***,***y'***)

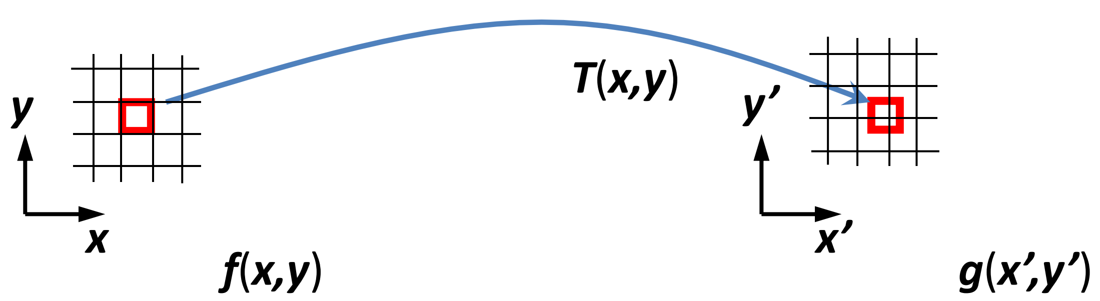
# Forward Warping

- Send each pixel $f(x,y)$ to its corresponding location $(x',y') = T(x,y)$ in $g(x',y')$
  - What if pixel lands "between" two pixels?



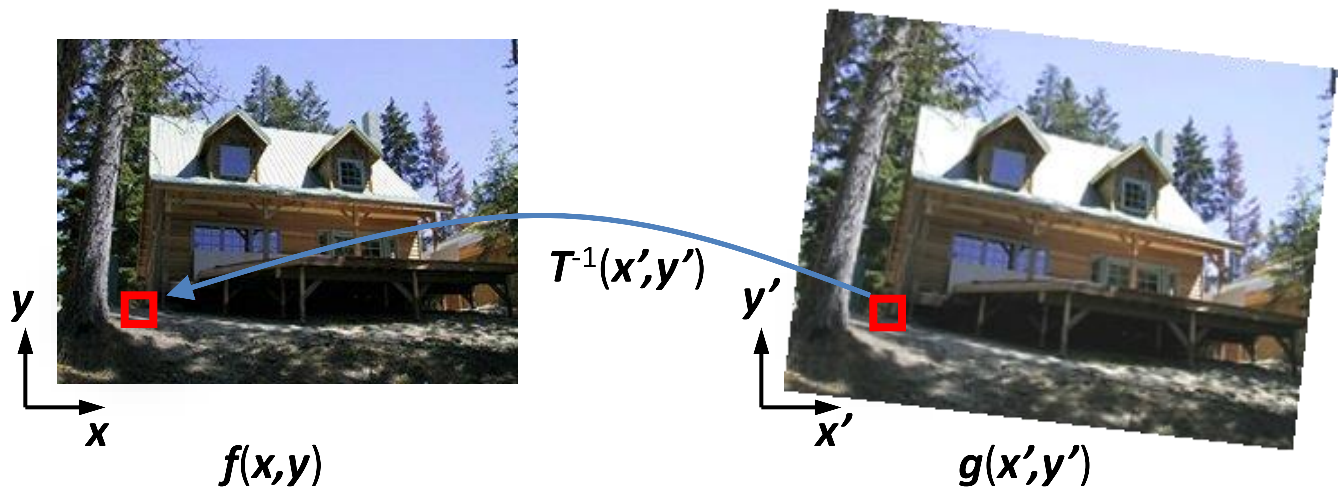
# Forward Warping

- Send each pixel $f(x,y)$ to its corresponding location $(x',y') = T(x,y)$ in $g(x',y')$
  - What if pixel lands "between" two pixels?
  - Answer: add "contribution" to several pixels, normalize later
  - Can still result in holes



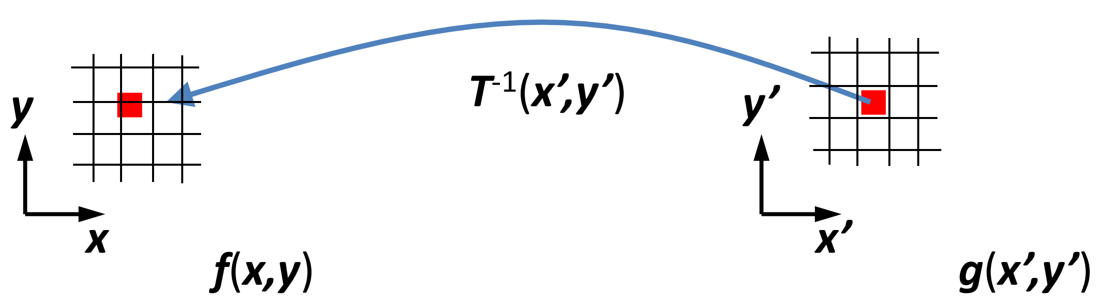
# Inverse Warping

- Get each pixel ***g***(***x'***,***y'***) from its corresponding location (***x***,***y***) = ***T***$^{-1}$(***x'***,***y'***) in ***f***(***x***,***y***)
  - Requires taking the inverse of the transform
  - What if pixel comes from "between" two pixels?



# Inverse Warping

- Get each pixel in ***g*** from its corresponding location in ***f***
  - What if pixel comes from “between” two pixels?
  - Answer: *resample* color value from *interpolated* (*prefiltered*) source image



# Interpolation

- Possible interpolation filters:
  - nearest neighbor
  - bilinear
  - bicubic (interpolating)
  - …