3D Vision and Machine Perception

Prof. Kyungdon Joo

3D Vision & Robotics Lab.

AI Graduate School (AIGS) & Computer Science and Engineering (CSE)

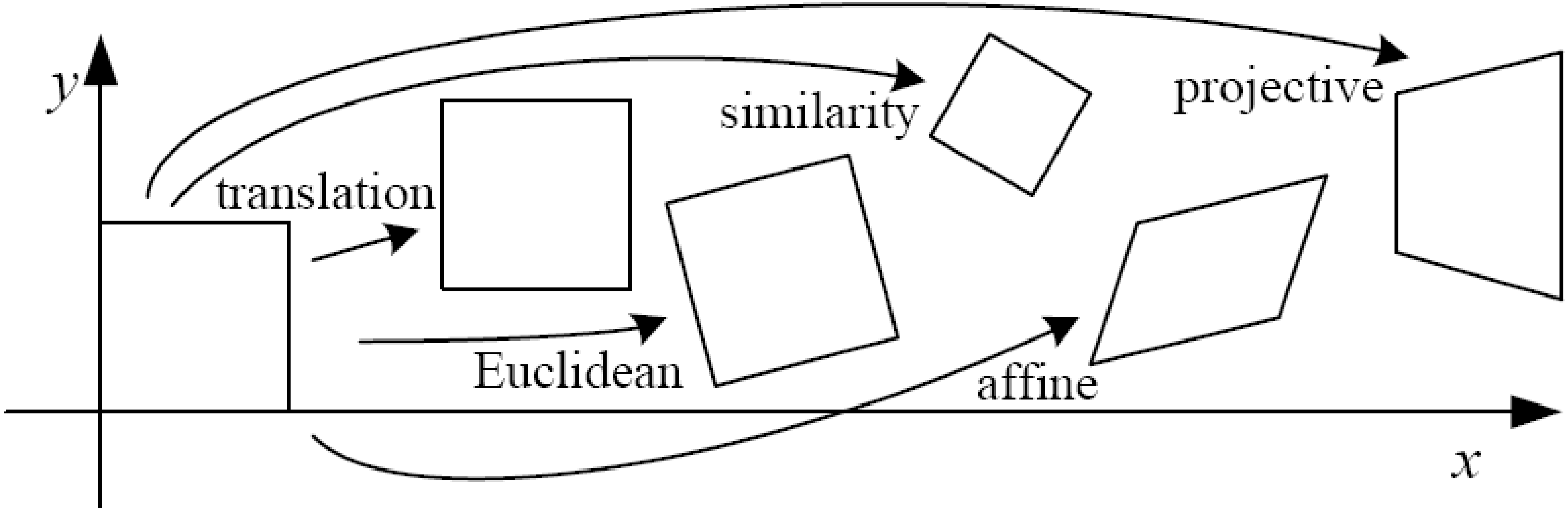
Some materials, figures, and slides (used for this course) are from textbooks, published papers, and other open lectures

Contents

- Classification of 2D transformations
- Determining unknown 2D transformations
- Determining unknown image warps

Classification of 2D transformations

Classification of 2D transformations

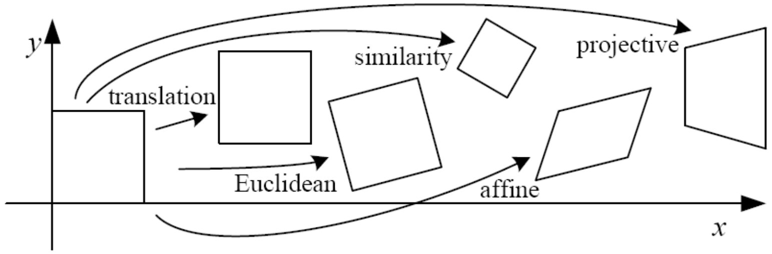


Classification of 2D transformations

- Translation:

$$\begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{bmatrix}$$

- How many degrees of freedom?

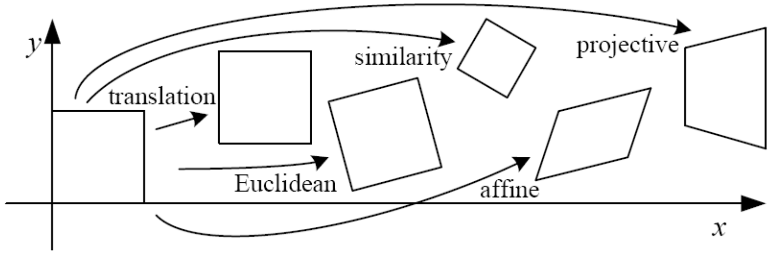


Classification of 2D transformations

- Euclidean (rigid): rotation + translation

$$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Are there any values that are related?

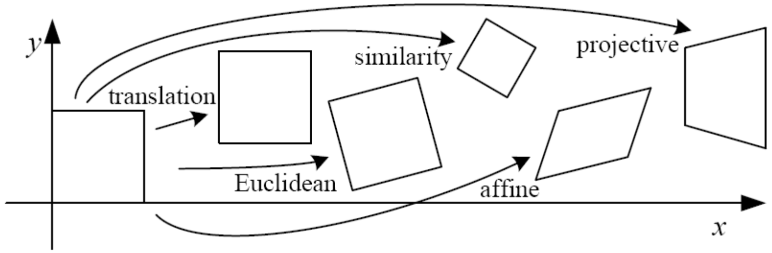


Classification of 2D transformations

- Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- How many degrees of freedom?

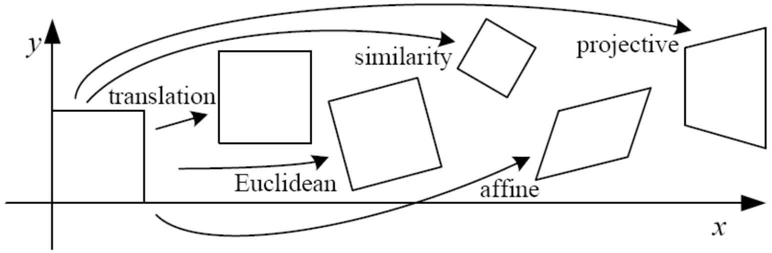


Classification of 2D transformations

- Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Which other matrix values will change if this increases?

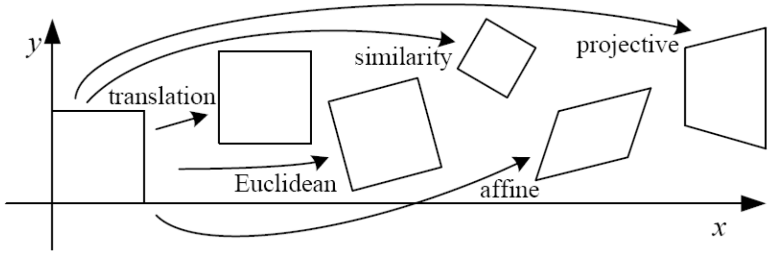


Classification of 2D transformations

- Similarity: uniform scaling + rotation + translation

$$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Are there any values that are related?



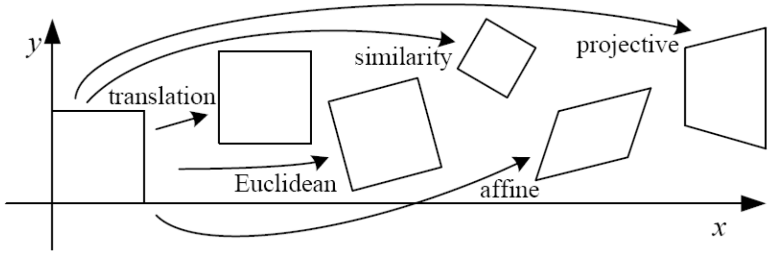
Classification of 2D transformations

- Similarity: uniform scaling + rotation + translation

$$\begin{bmatrix} \boxed{\begin{matrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{matrix}} & \begin{matrix} r_3 \\ r_6 \end{matrix} \\ \begin{matrix} 0 & \quad 0 \end{matrix} & 1 \end{bmatrix}$$

multiply these four by scale **s**

- How many degrees of freedom?

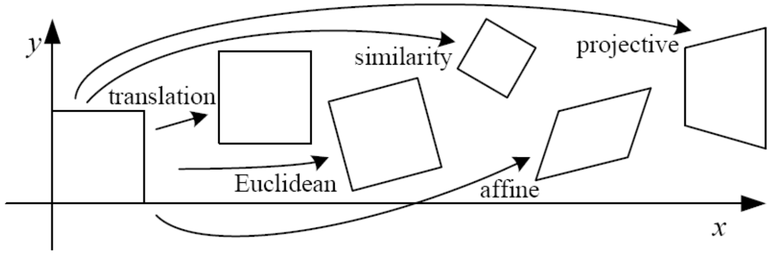


Classification of 2D transformations

- Affine transform: uniform scaling + shearing + rotation + translation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Are there any values that are related?



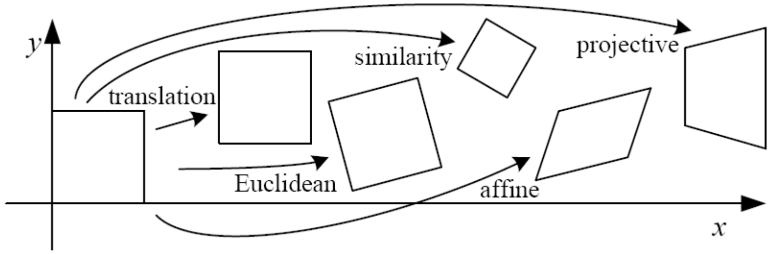
Classification of 2D transformations

- Affine transform: uniform scaling + shearing + rotation + translation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Are there any values that are related?

$$\underset{\text{similarity}}{\begin{bmatrix} sr_1 & sr_2 \\ sr_3 & sr_4 \end{bmatrix}} \underset{\text{shear}}{\begin{bmatrix} 1 & h_1 \\ h_2 & 1 \end{bmatrix}} = \begin{bmatrix} sr_1 + h_2 sr_2 & sr_2 + h_1 sr_1 \\ sr_3 + h_2 sr_4 & sr_4 + h_1 sr_3 \end{bmatrix}$$



Classification of 2D transformations

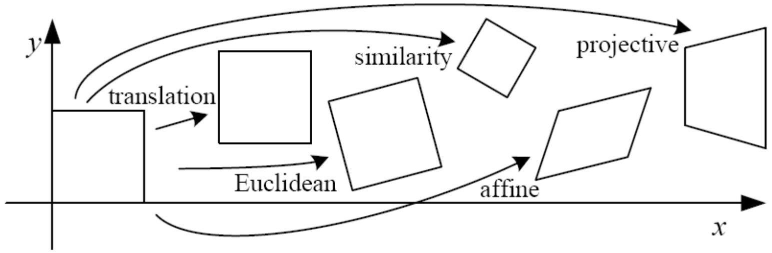
- Affine transform: uniform scaling + shearing + rotation + translation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}$$

- Are there any values that are related?

$$\underbrace{\begin{bmatrix} sr_1 & sr_2 \\ sr_3 & sr_4 \end{bmatrix}}_{\text{similarity}} \underbrace{\begin{bmatrix} 1 & h_1 \\ h_2 & 1 \end{bmatrix}}_{\text{shear}} = \begin{bmatrix} sr_1 + h_2 sr_2 & sr_2 + h_1 sr_1 \\ sr_3 + h_2 sr_4 & sr_4 + h_1 sr_3 \end{bmatrix}$$

- How many degrees of freedom?

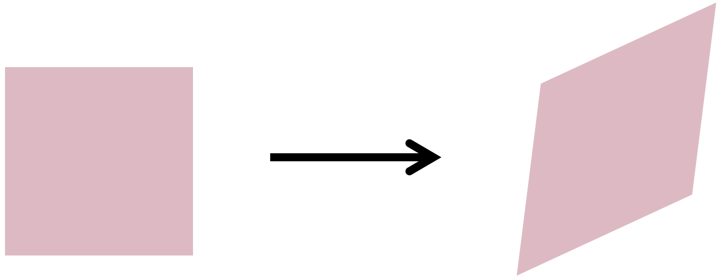


Affine transformations

- Affine transformations are combinations of
  - Arbitrary (4-DOF) 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 map to parallel lines
  - Ratios are preserved
  - Compositions of affine transforms are also affine transforms



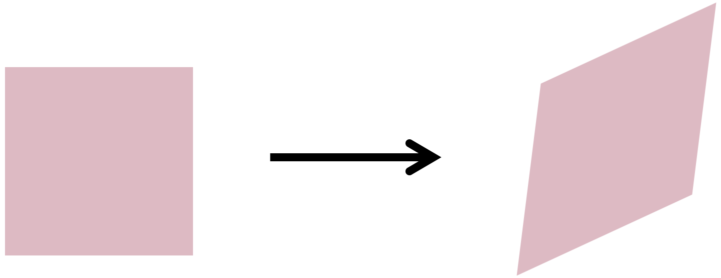
- Does the last coordinate w ever change?

Affine transformations

- Affine transformations are combinations of
  - Arbitrary (4-DOF) 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 map to parallel lines
  - Ratios are preserved
  - Compositions of affine transforms are also affine transforms



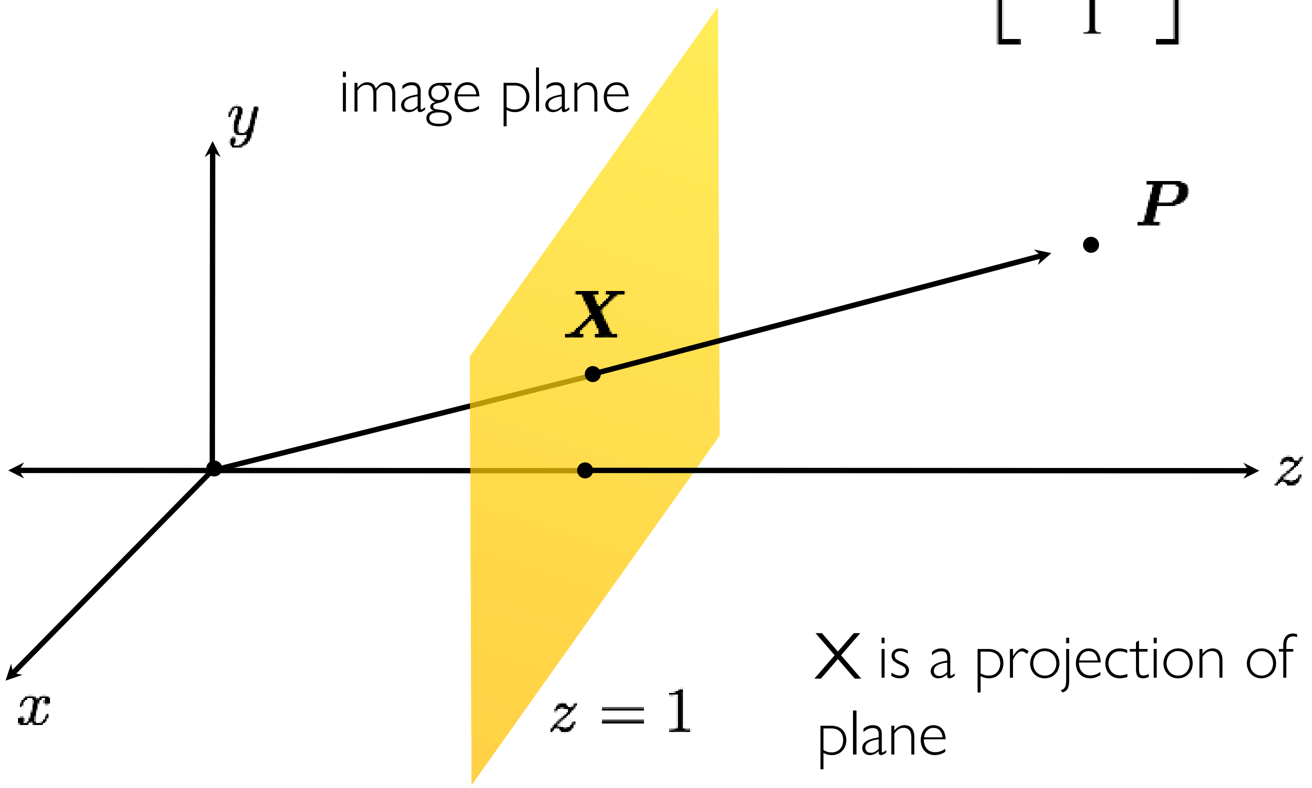
- Does the last coordinate w ever change?
  - Nope! But what does that mean?

How to interpret affine transformations here?

- Image point in pixel coordinates $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$\Updownarrow$$

- Image point in homogeneous coordinates $\boldsymbol{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$



X is a projection of a point P on the image plane

Projective transformations (aka homographies)

- Projective transformations are combinations of
  - Affine transformations; and
  - Projective wraps

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

how many degrees of freedom?

Projective transformations (aka homographies)

- Projective transformations are combinations of
  - Affine transformations; and
  - Projective wraps

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

8 DOF: vectors (and therefore matrices) are defined up to scale)

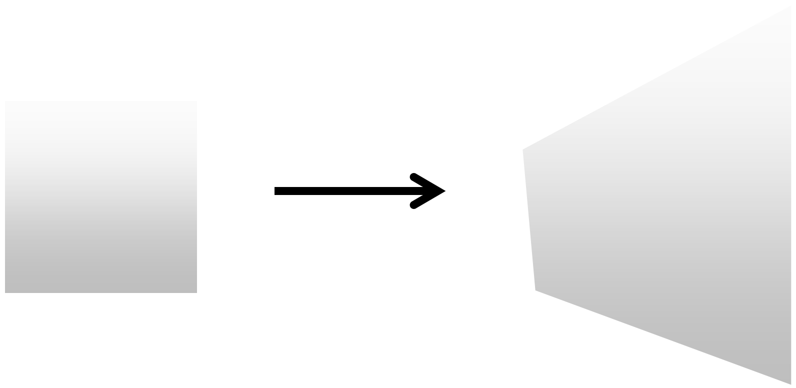
Projective transformations (aka homographies)

- Projective transformations are combinations of
  - Affine transformations; and
  - Projective wraps

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

8 DOF: vectors (and therefore matrices) are defined up to scale)

- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily map to parallel lines
  - Ratios are not necessarily preserved
  - Compositions of projective transforms are also projective transforms

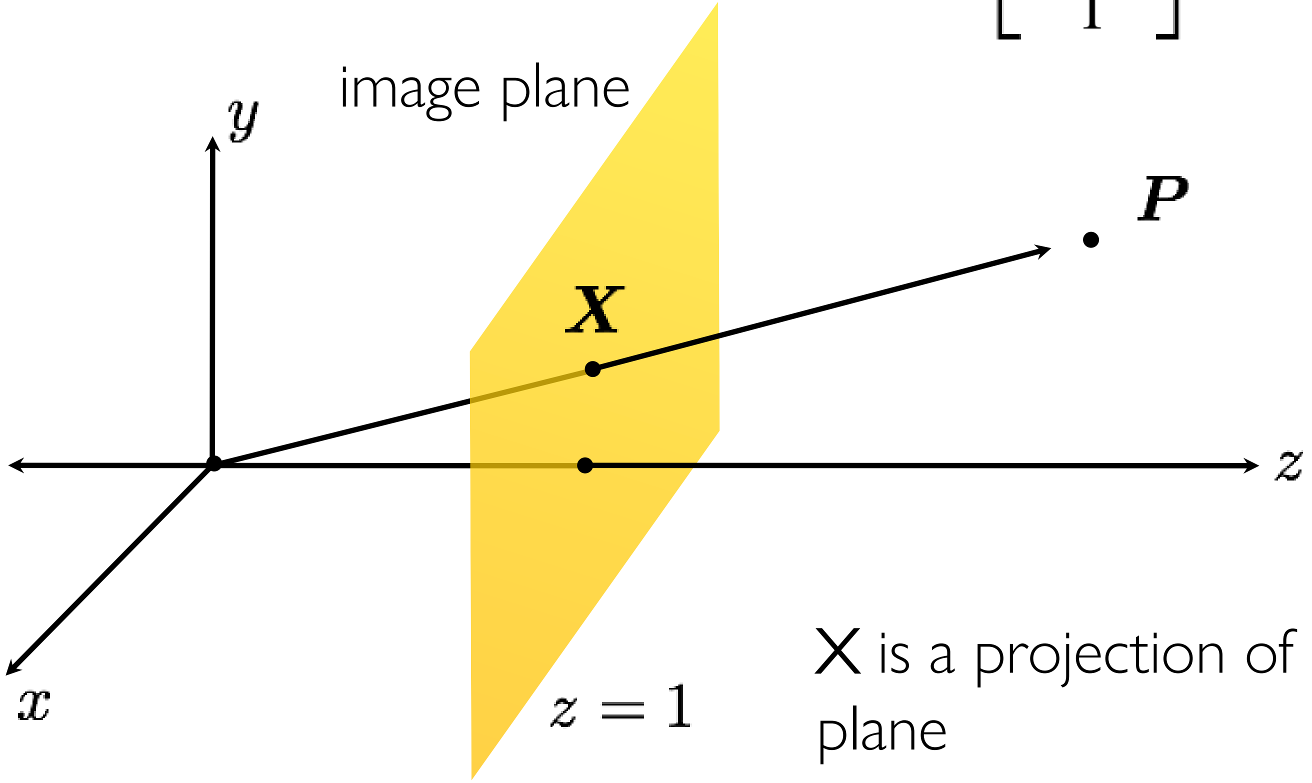


How to interpret projective transformations here?

- Image point in pixel coordinates $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$\Updownarrow$$

- Image point in heterogeneous coordinates $\boldsymbol{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$



X is a projection of a point P on the image plane

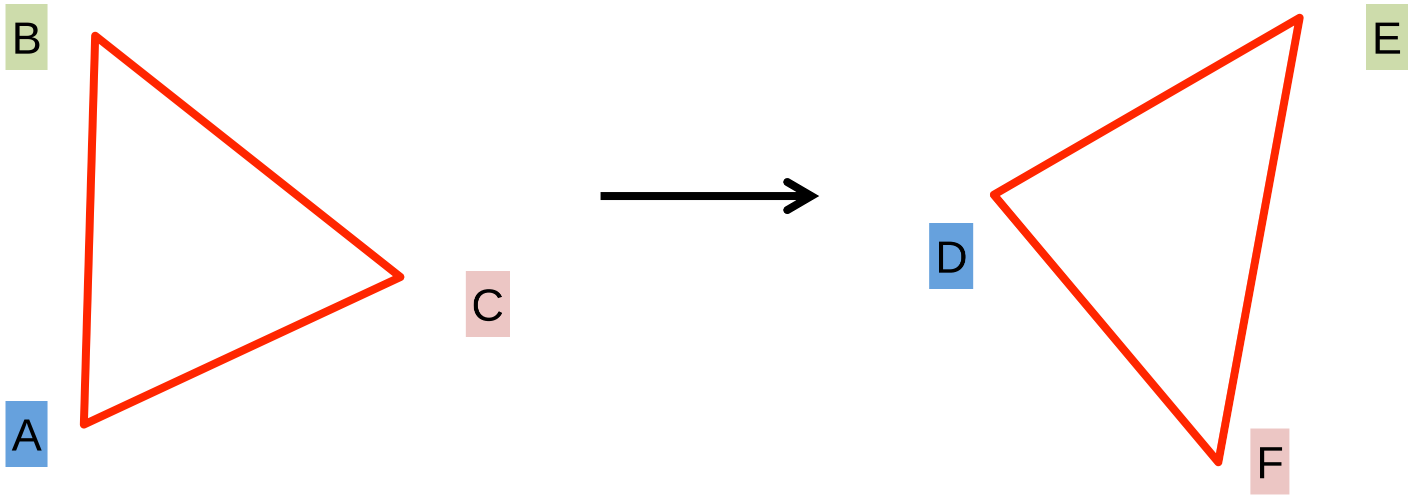
Classification of 2D transformations

Name	Matrix	# D.O.F.
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]$	2
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]$	3
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]$	4
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]$	6
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]$	8

Determining unknown (affine) 2D transformations

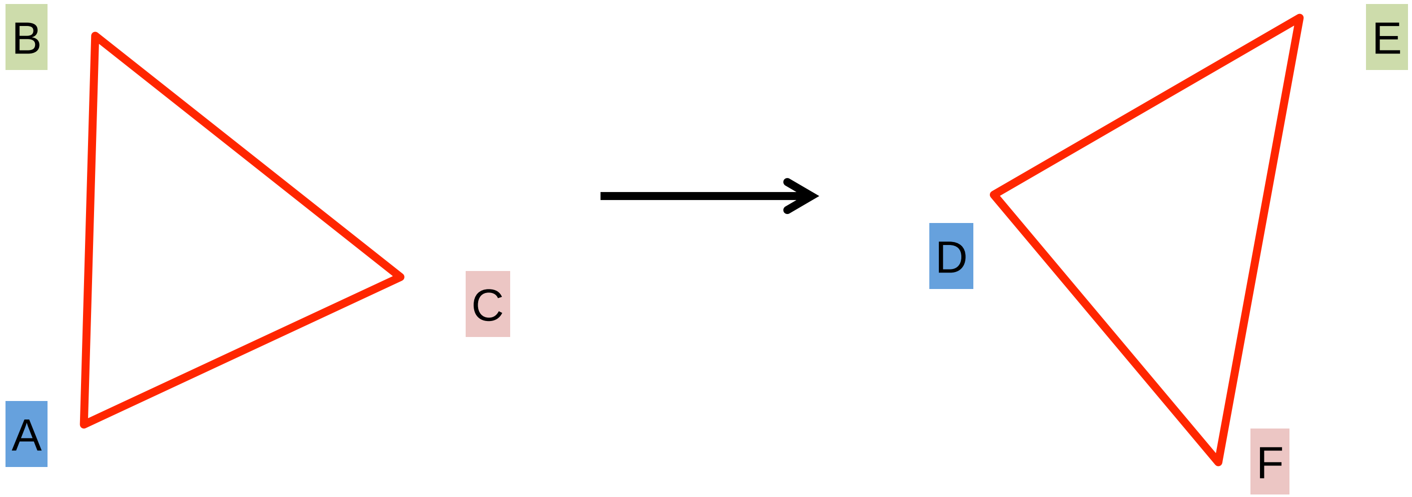
Determining unknown transformations

- Suppose we have two triangles: ABC and DEF.

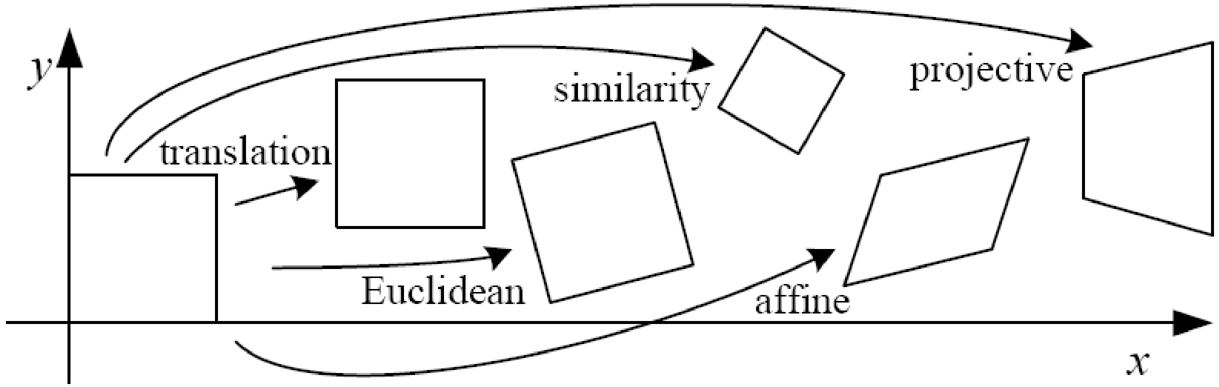


Determining unknown transformations

- Suppose we have two triangles: ABC and DEF.

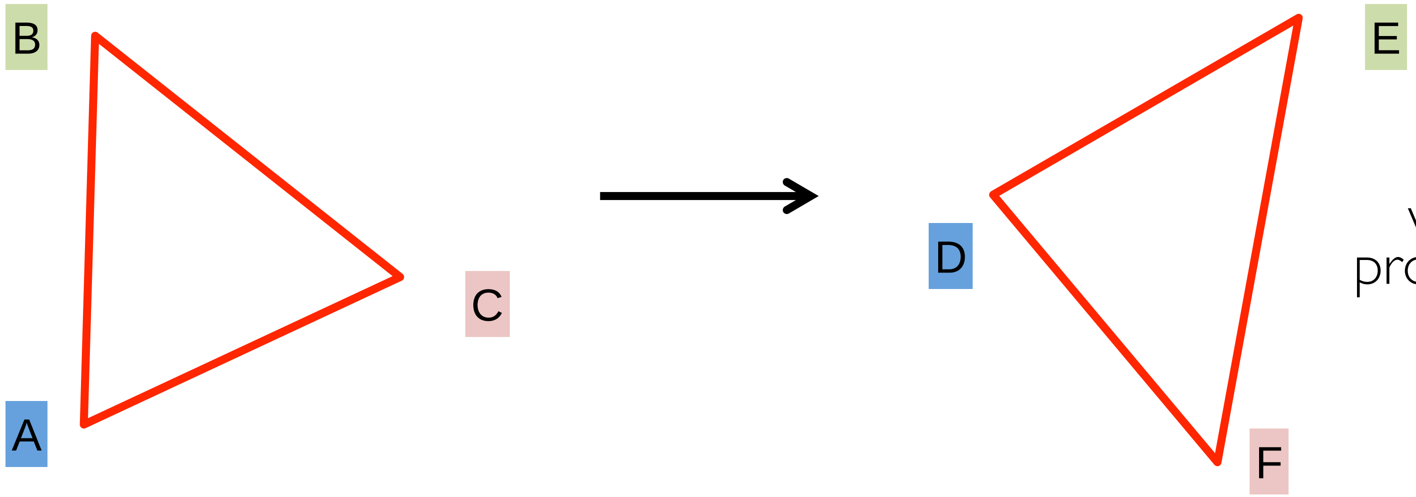


- What type of transformation will map A to D, B to E, and C to F?



Determining unknown transformations

- Suppose we have two triangles: ABC and DEF.



important:
we will see a different
procedure for dealing with
homographies!

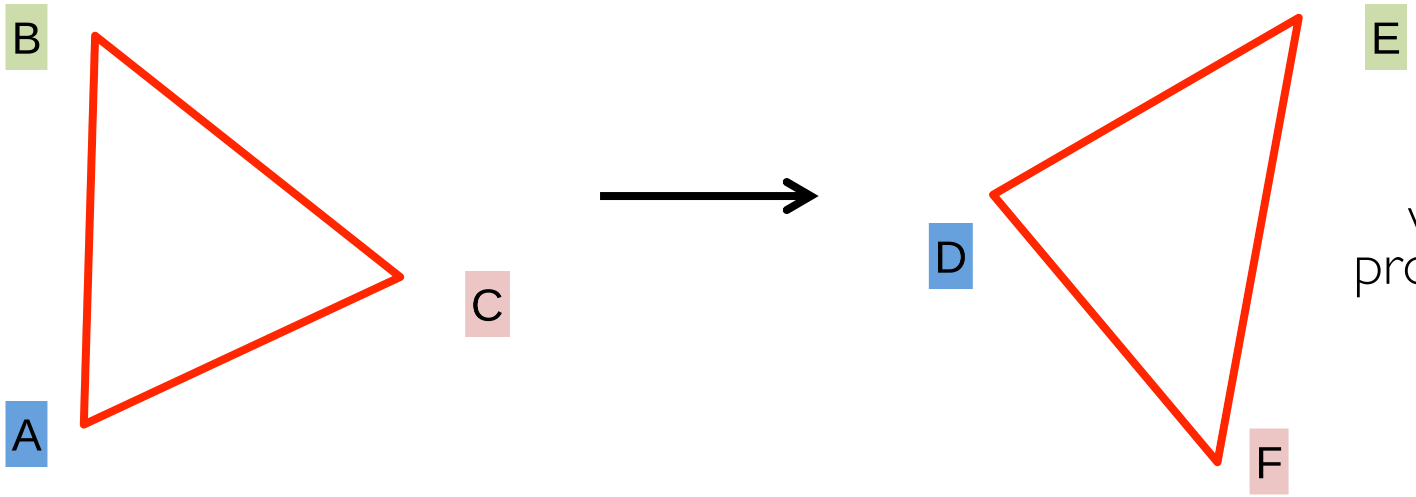
- What type of transformation will map A to D, B to E, and C to F?
- How do we determine the unknown parameters?
- How many degrees of freedom do we have?

affine transform:
uniform scaling + shearing
+ rotation + translation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}$$

Determining unknown transformations

- Suppose we have two triangles: ABC and DEF.



important:
we will see a different procedure for dealing with homographies!

- What type of transformation will map A to D, B to E, and C to F?
- How do we determine the unknown parameters?
- One point correspondence gives how many equations?
- How many point correspondences do we need?

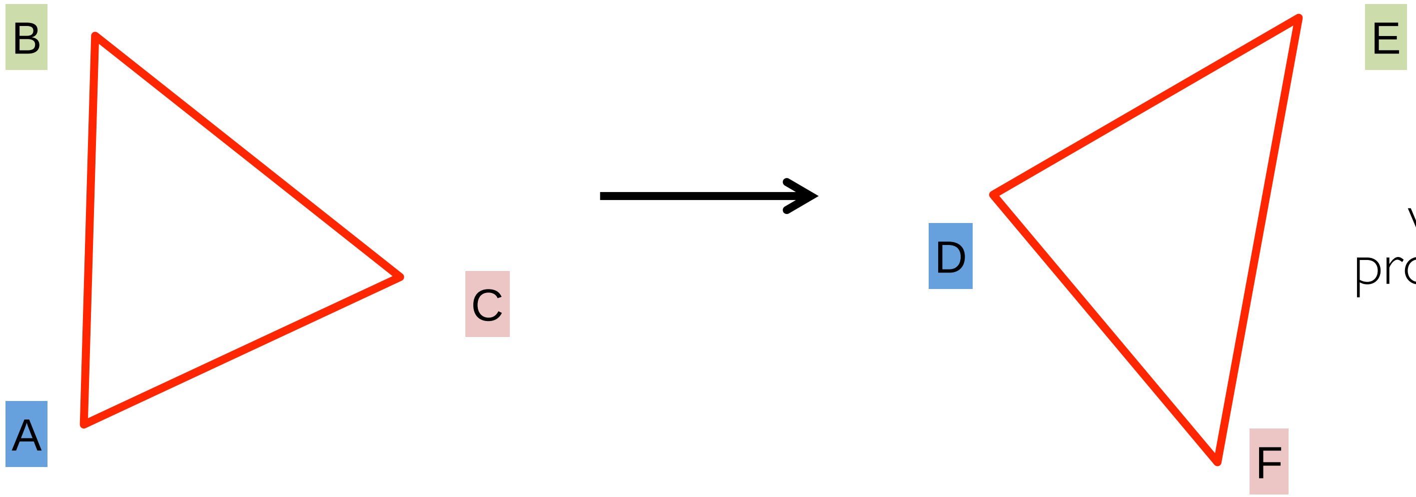
unknowns

$$\boldsymbol{x}' = \mathbf{M}\boldsymbol{x}$$

point correspondences

Determining unknown transformations

- Suppose we have two triangles: ABC and DEF.



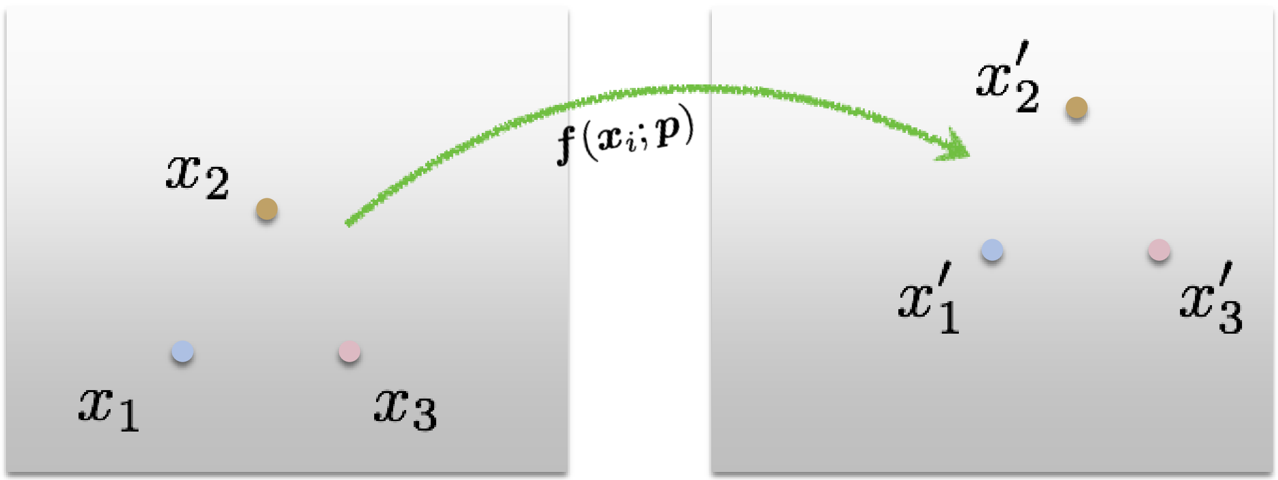
important:
we will see a different
procedure for dealing with
homographies!

- What type of transformation will map A to D, B to E, and C to F?
- How do we determine the unknown parameters?
- How do we solve this for **M**?

unknowns

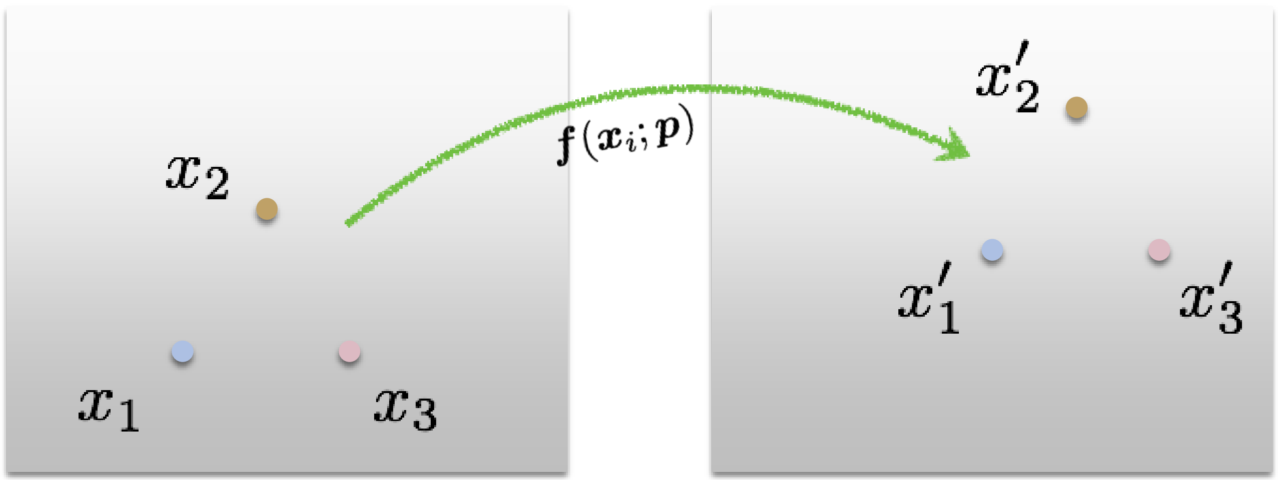
$$\boldsymbol{x}' = \mathbf{M}\boldsymbol{x}$$

point correspondences



Least Squares Error

$$E_{\mathrm{LS}} = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i\|^2$$

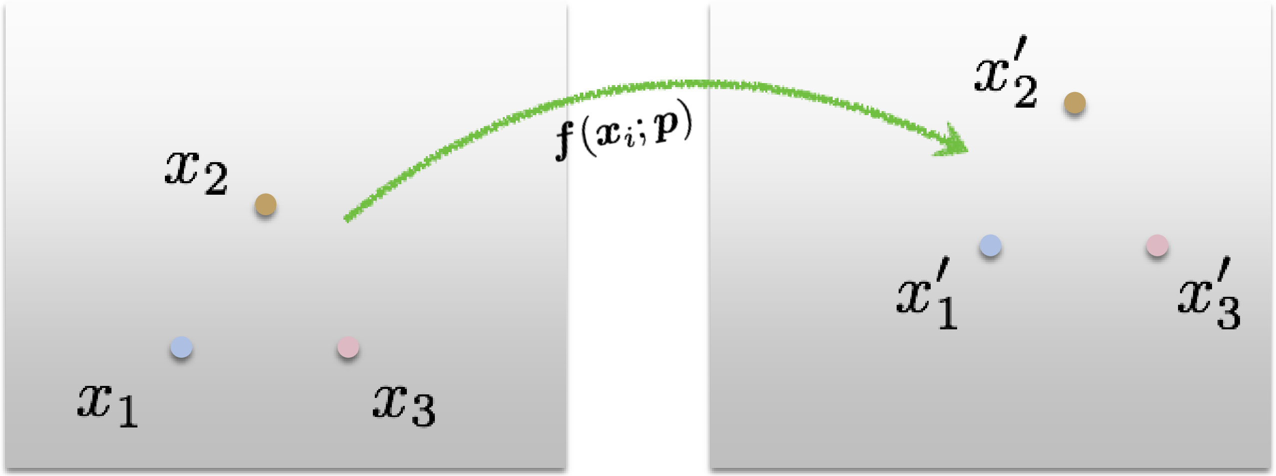


Least Squares Error

what is this?

$$E_{\mathrm{LS}} = \sum_i \| \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i \|^2$$

what is this? what is this?



$$||\boldsymbol{x}|| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Euclidean (L2) norm

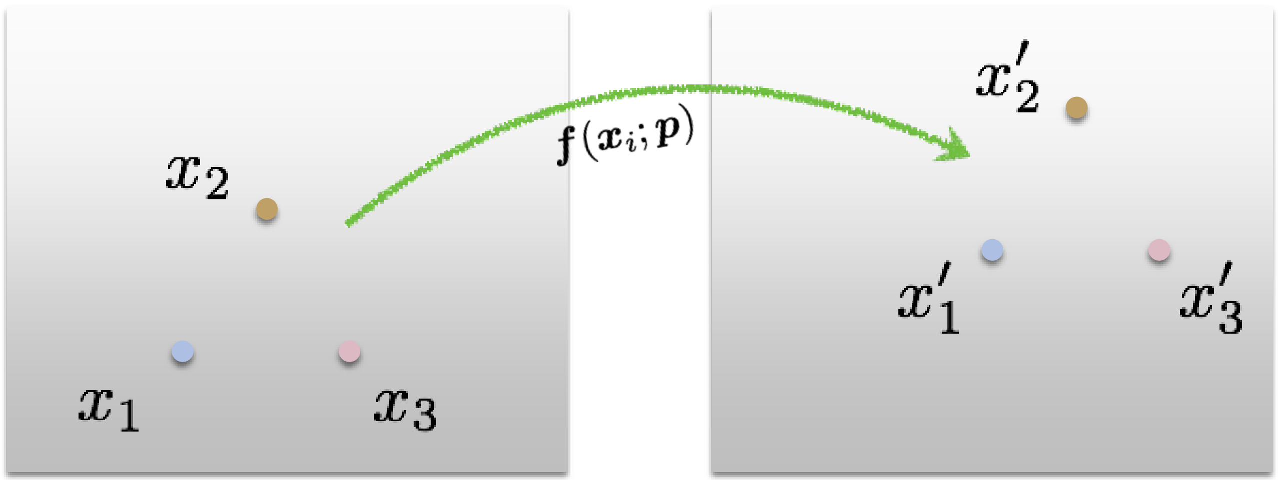
Least Squares Error

$$E_{\mathrm{LS}} = \sum_i ||\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i||^2$$

squared!

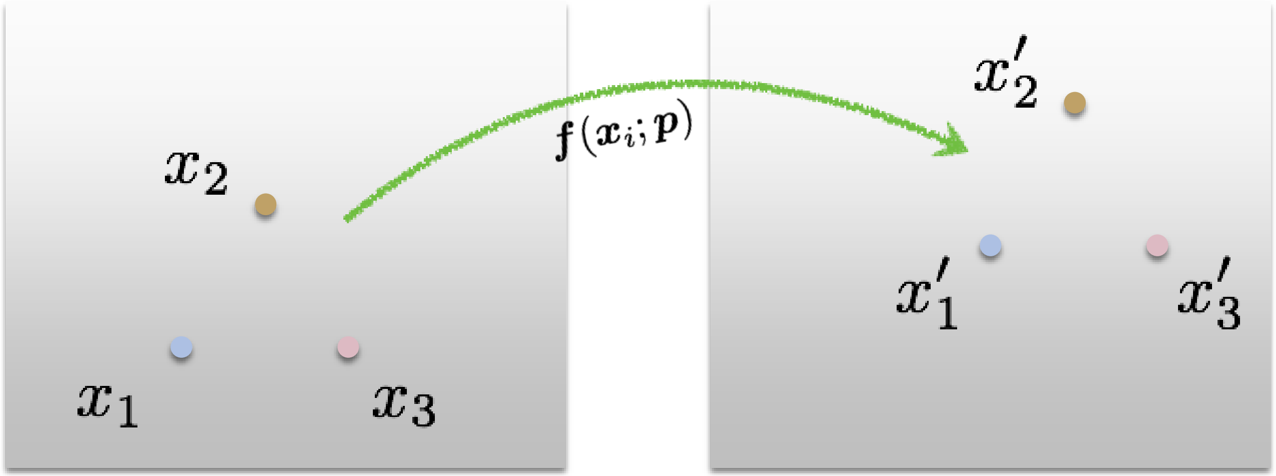
predicted location

measured location



Least Squares Error

$$E_{\mathrm{LS}} = \sum_i \underbrace{\| \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i \|}_{\substack{\text{residual} \\ \text{(projection error)}}}{}^2$$



- Find parameters that minimize squared error

$$\hat{\boldsymbol{p}} = \arg\min_{\boldsymbol{p}} \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i\|^2$$

- General form of linear least squares
(warning: change of notation. x is a vector of parameters!)

$$
\begin{aligned}
E_{\mathrm{LLS}} &= \sum_i |\boldsymbol{a}_i \boldsymbol{x} - \boldsymbol{b}_i|^2 \\
&= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \quad \text{(matrix form)}
\end{aligned}
$$

Determining unknown transformations

- Affine transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

why can we drop the last line?

- Vectorize transformation parameters:

- Stack equations from point correspondences:

$$\underbrace{\begin{bmatrix} x' \\ y' \\ x' \\ y' \\ \vdots \\ x' \\ y' \end{bmatrix}}_{\boldsymbol{b}} = \underbrace{\begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ & & & \vdots & & \\ x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix}}_{\boldsymbol{x}}$$

- Notation in system form:

- General form of linear least squares
  (warning: change of notation. x is a vector of parameters!)

$$\begin{aligned} E_{\mathrm{LLS}} &= \sum_i |\boldsymbol{a}_i \boldsymbol{x} - \boldsymbol{b}_i|^2 \\ &= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \quad \text{(matrix form)} \end{aligned}$$

- This function is quadratic.
- How do you find the root of a quadratic?

Solving the linear system

- Convert the system to a linear least-squares problem: $E_{\mathrm{LLS}} = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$

- Expand the error: $E_{\mathrm{LLS}} = \boldsymbol{x}^\top(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} - 2\boldsymbol{x}^\top(\mathbf{A}^\top\boldsymbol{b}) + \|\boldsymbol{b}\|^2$

- Minimize the error: $(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} = \mathbf{A}^\top\boldsymbol{b}$

- Set derivative to 0, solve for x: $\boldsymbol{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{b}$

✓In Matlab: x = A \ b (search `mldivide`)

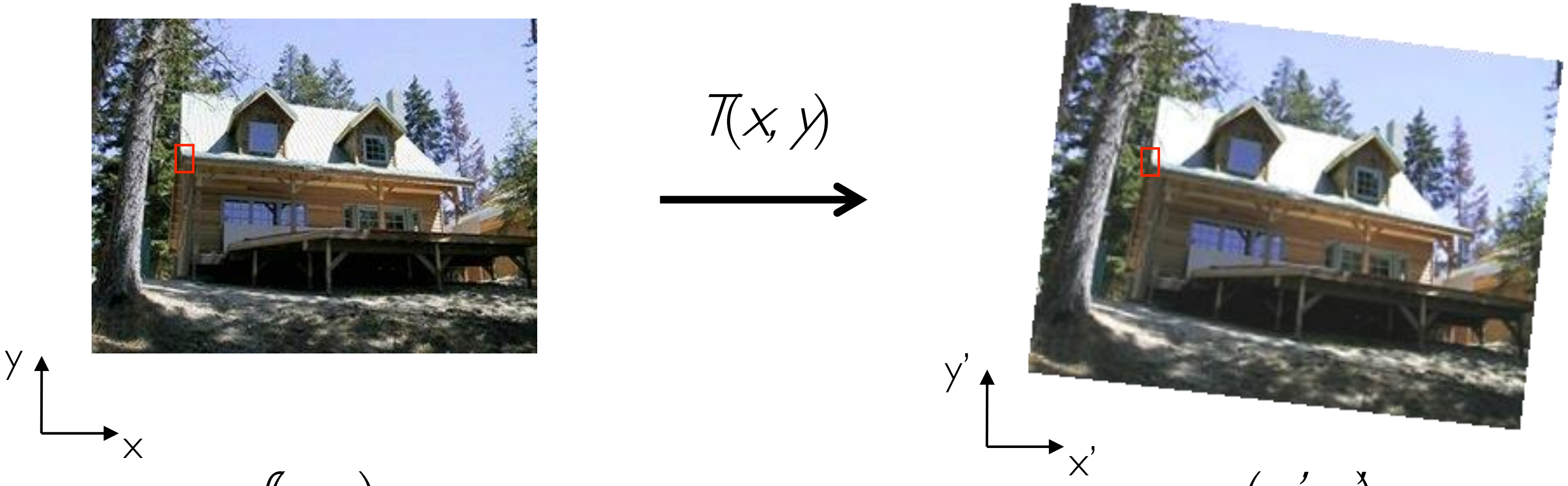
- Linear least squares estimation only works when the transform function is ?

- **Linear** least squares estimation only works when the transform function is **linear!**
  - Think about similarity transform
- Also doesn't deal well with outliers

Determining unknown image warps

Determining unknown image warps

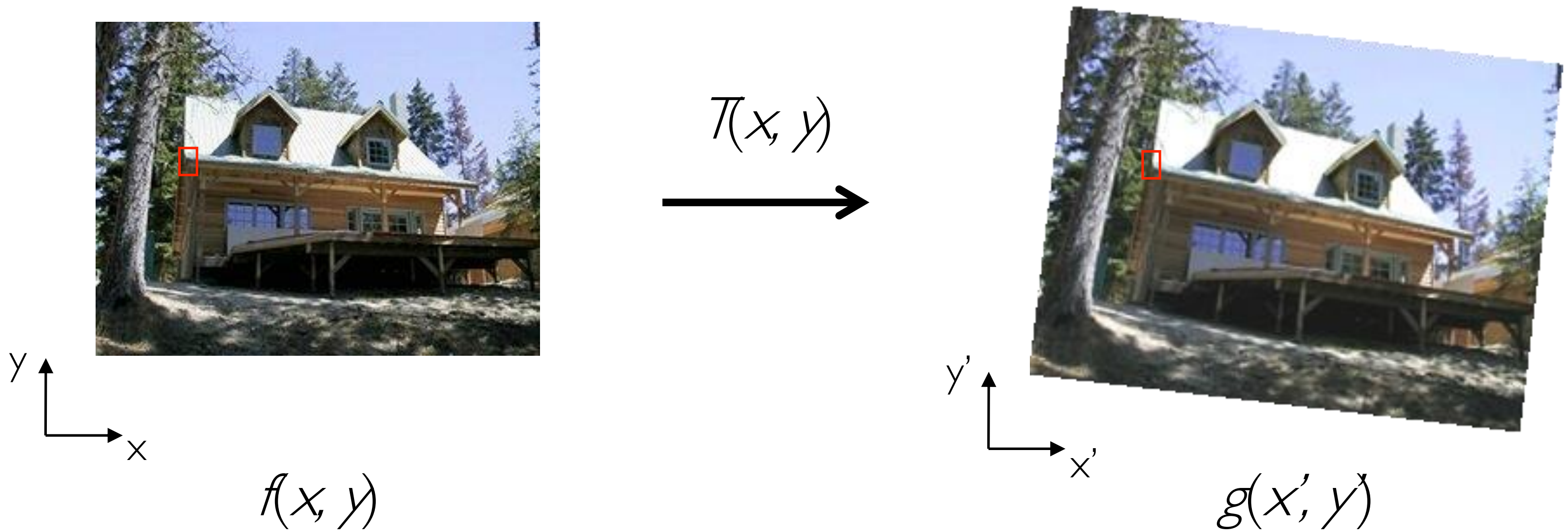
- Suppose we have two images.
- How do we compute the transform that takes one to the other?



$f(x, y)$ $g(x', y')$

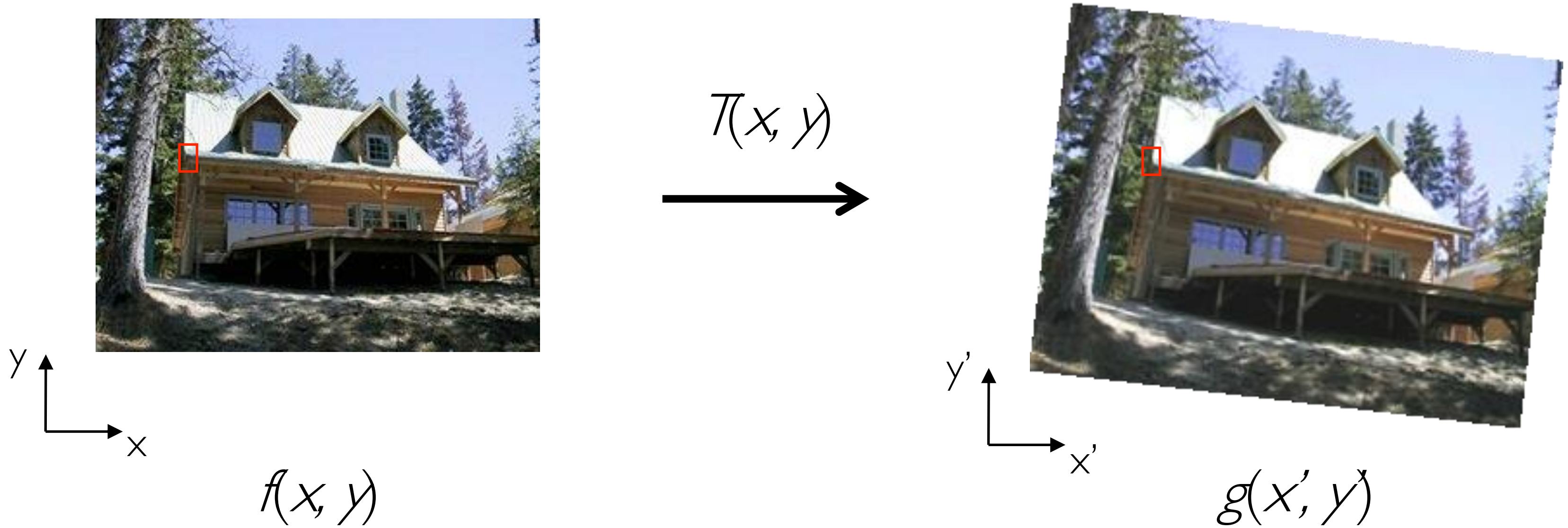
Forward warping

- Suppose we have two images.
- How do we compute the transform that takes one to the other?



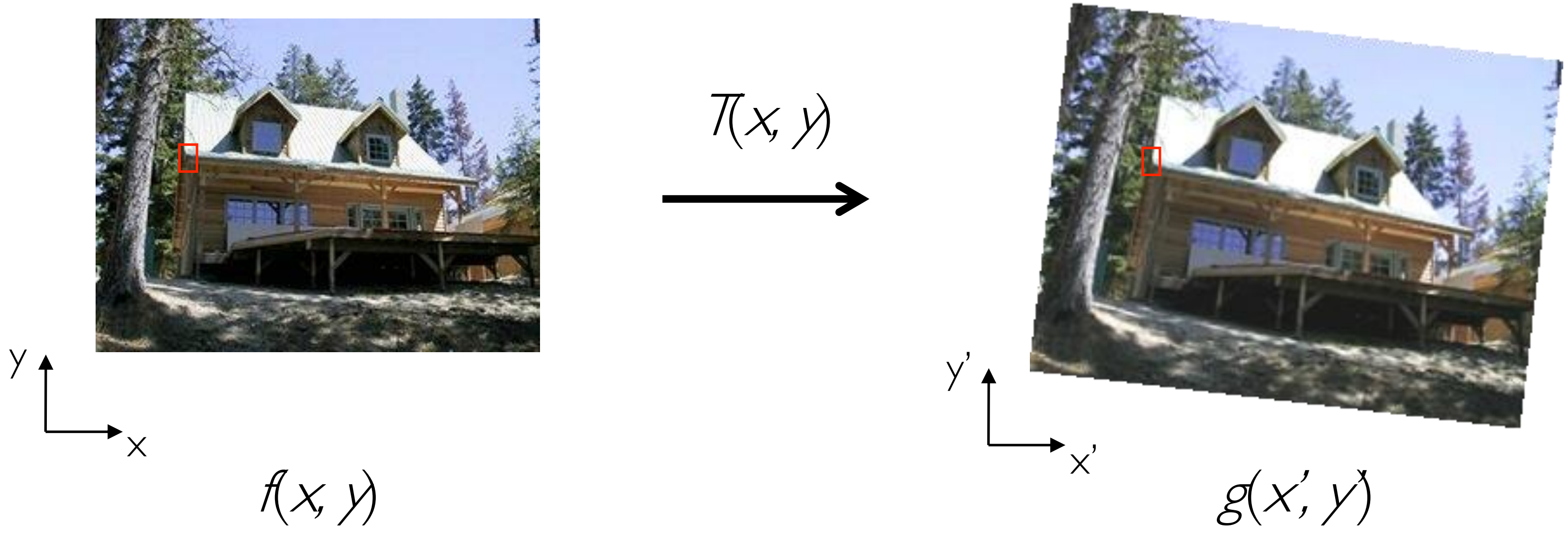
Forward warping

- Form enough pixel-to-pixel correspondences between two images
- Solve for linear transform parameters as before
- Send intensities f(x,y) in first image to their corresponding location in the second image



Forward warping

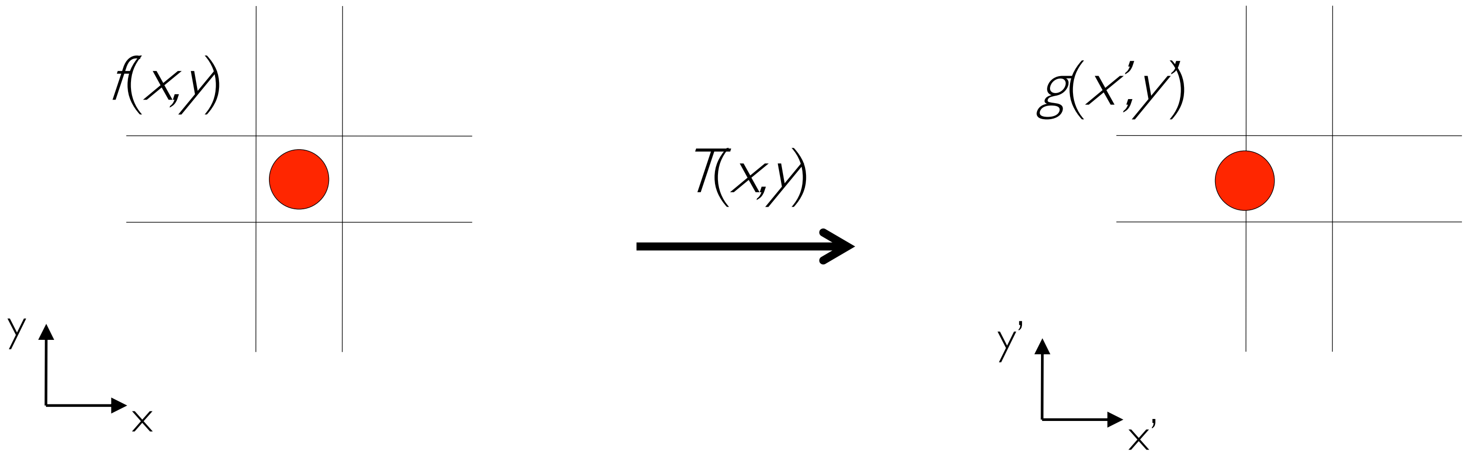
- Form enough pixel-to-pixel correspondences between two images
- Solve for linear transform parameters as before
- Send intensities f(x,y) in first image to their corresponding location in the second image



- What is the problem with this?

Forward warping

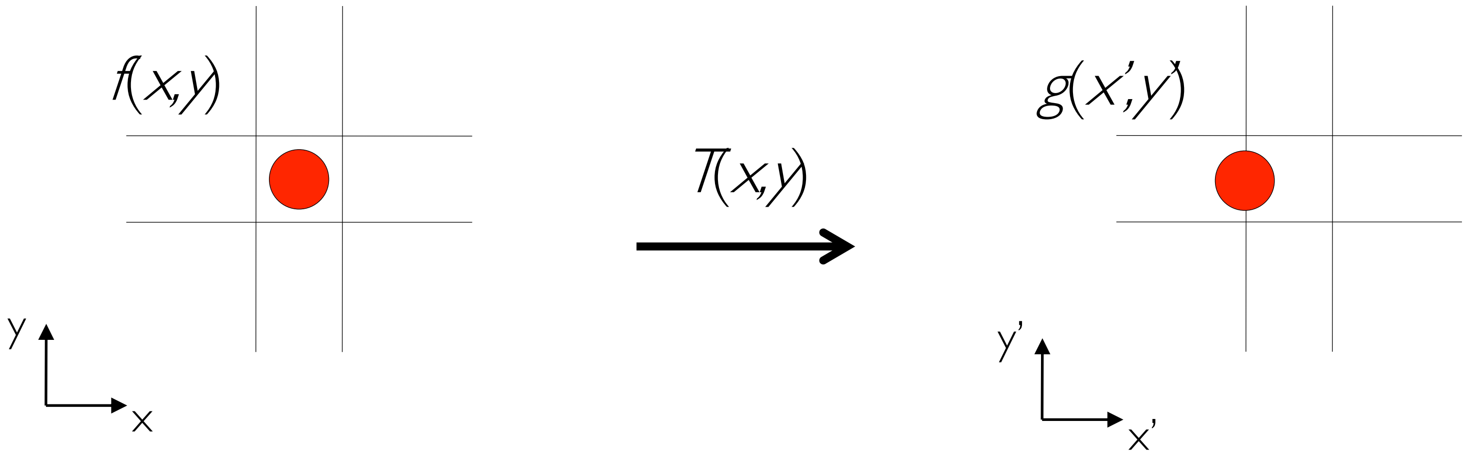
- Pixels may end up between two points
- How do we determine the intensity of each point?



Forward warping

- Pixels may end up between two points
- How do we determine the intensity of each point?

- We distribute color among neighboring pixels (x',y')

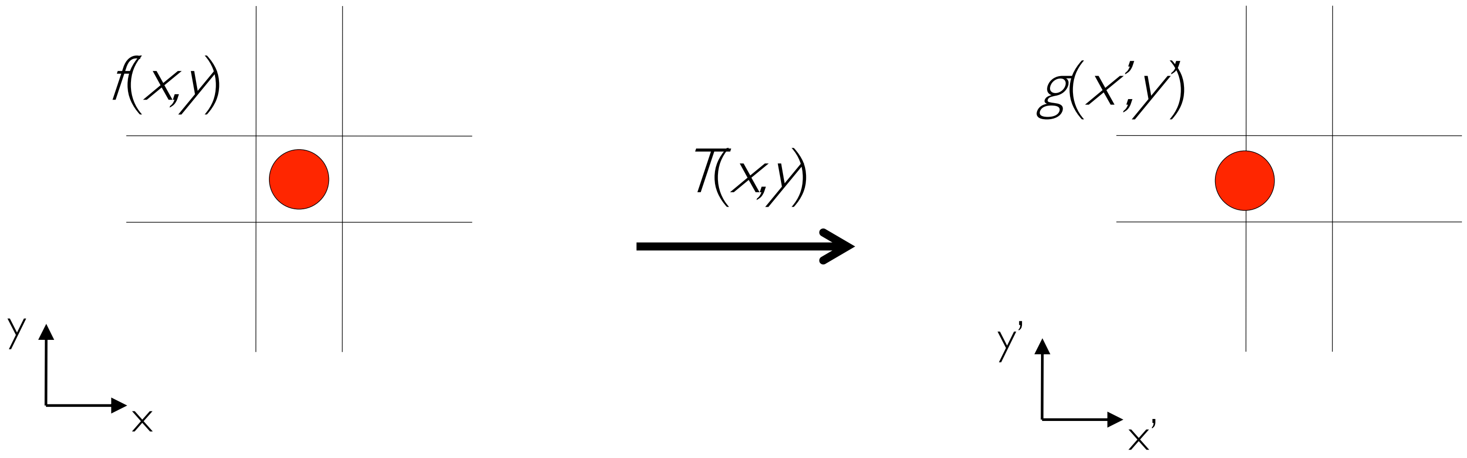


- What if a pixel (x',y') receives intensity from more than one pixels (x,y)?

Forward warping

- Pixels may end up between two points
- How do we determine the intensity of each point?

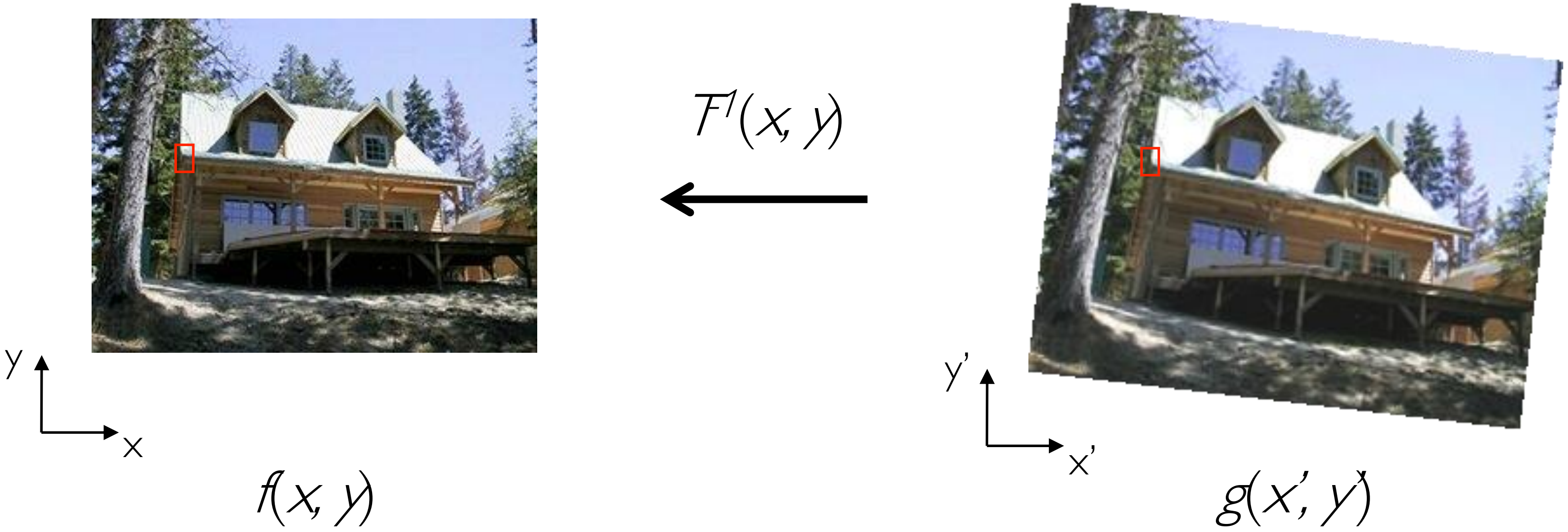
- We distribute color among neighboring pixels (x',y')



- What if a pixel (x',y') receives intensity from more than one pixels (x,y)?
- We average their intensity contributions.

Inverse warping

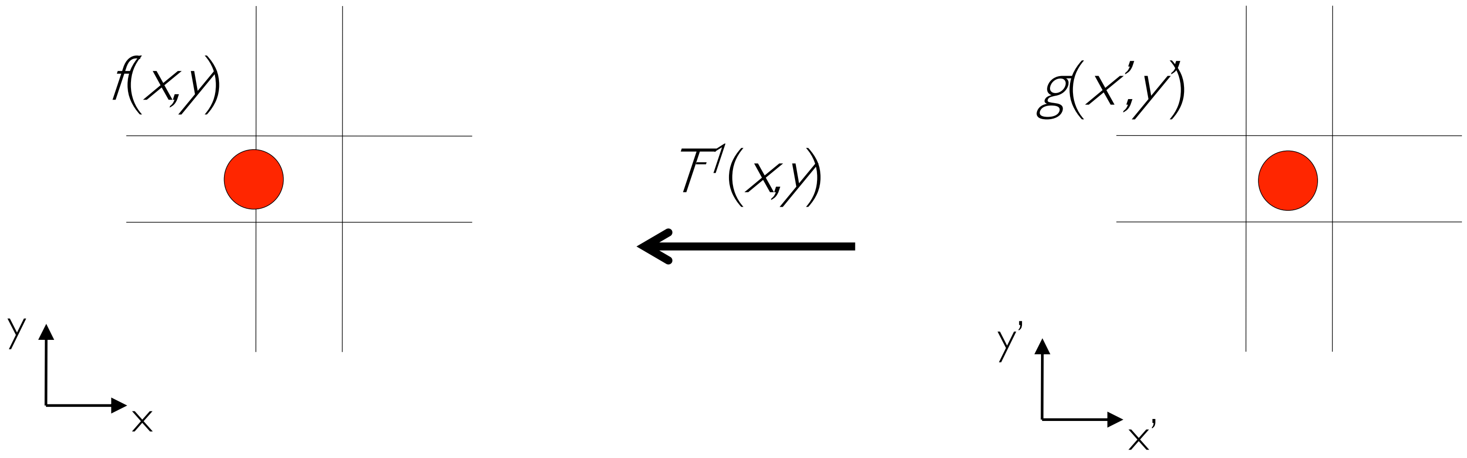
- Form enough pixel-to-pixel correspondences between two images
- Solve for linear transform parameters as before, then compute its inverse
- Get intensities g(x',y') in the second image from point (x,y) = T^{-1} (x',y') in first image



- What is the problem with this?

Inverse warping

- Pixel may come from between two points
- How do we determine its intensity?



Inverse warping

- Pixel may come from between two points
- How do we determine its intensity?

- Use interpolation