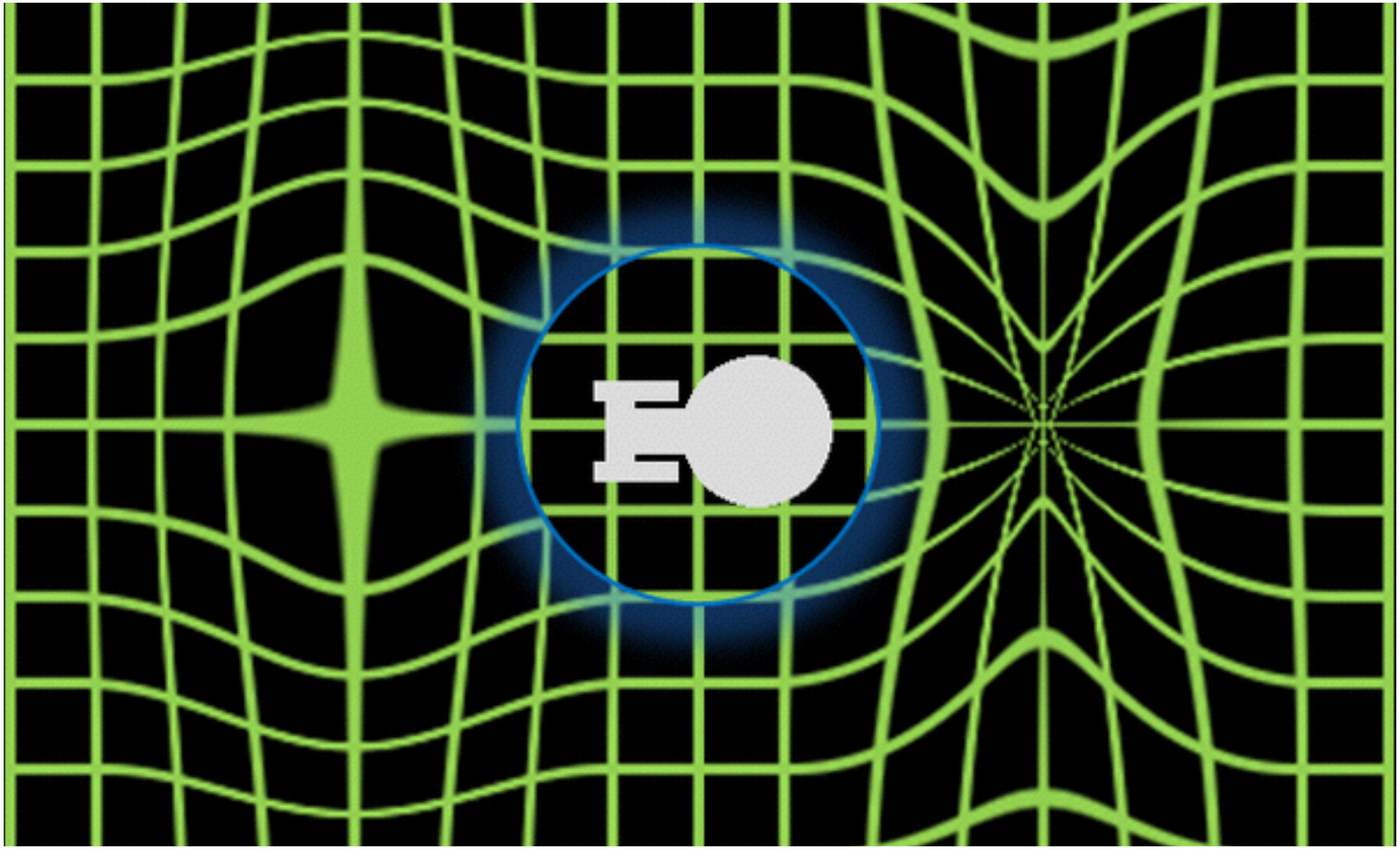


2D Image Transforms

16-385 Computer Vision

Carnegie Mellon University (Kris Kitani)


KRAFT
Macaroni & Cheese
DINNER
Family Size
ORIGINAL FLAVOR
NET WT
14.5 OZ (411g)

Extract features from an image …

what do we do next?

Feature matching

(object recognition, 3D reconstruction, augmented reality, image stitching)

How do you compute the transformation?

Given a set of matched feature points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

point in one image

point in the other image

and a transformation

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p})$$

transformation function

parameters

Find the best estimate of

$$\boldsymbol{p}$$

What kind of transformation functions are there?

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p})$$

2D Transformations

translation

rotation

aspect

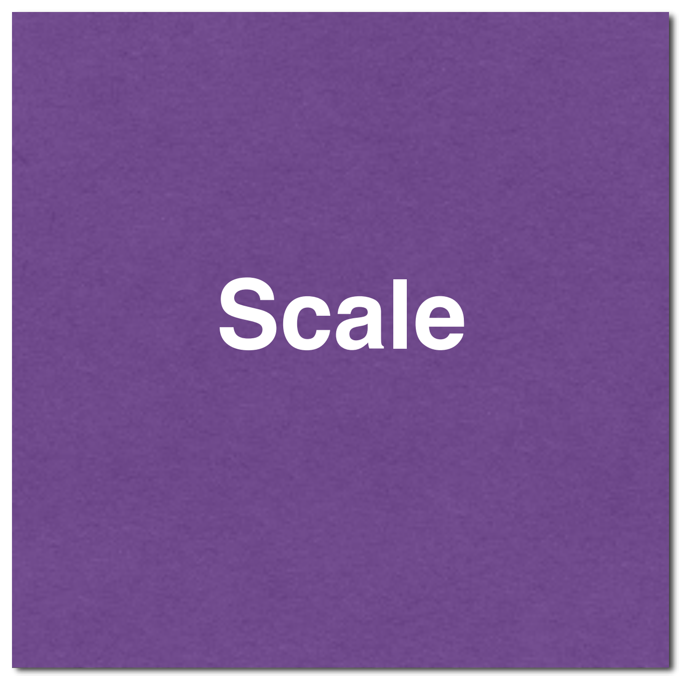
affine

perspective

cylindrical

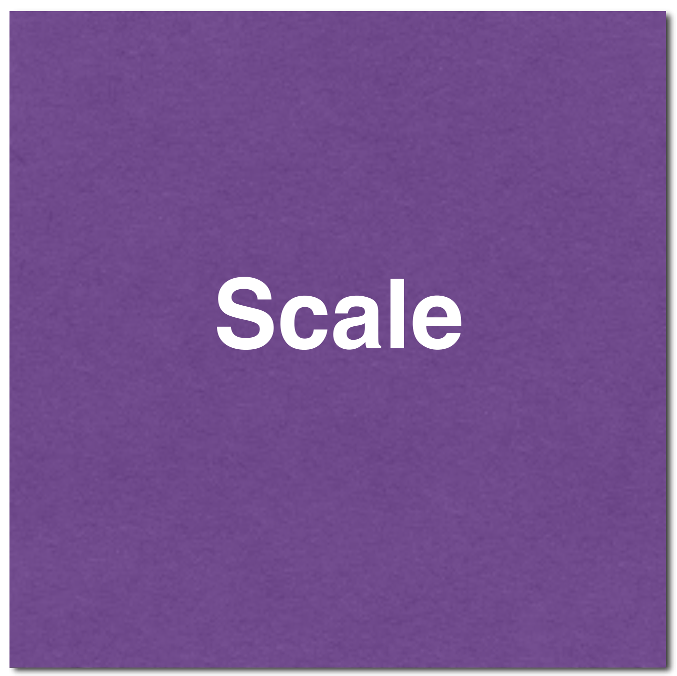
2D Planar Transformations

2D Planar Transformations



- Each component multiplied by a scalar
- Uniform scaling - same scalar for each component

2D Planar Transformations



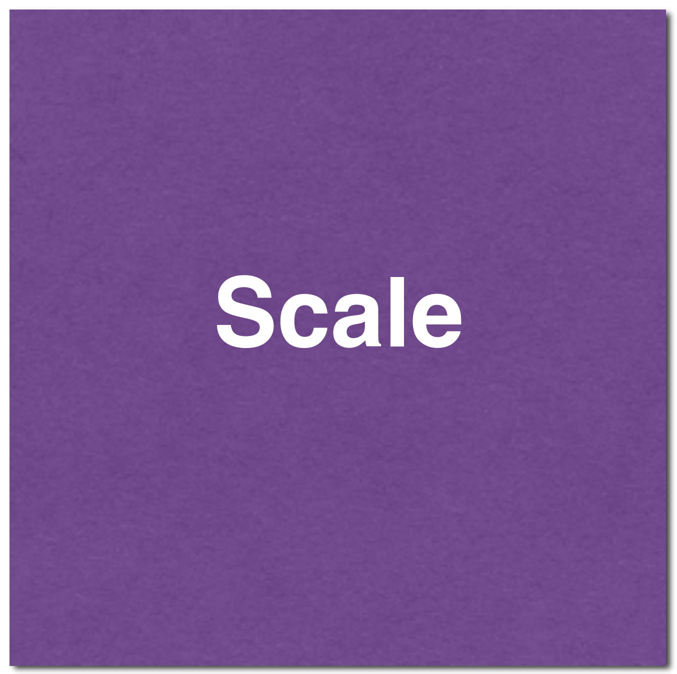
Scale

$$x' = ax$$

$$y' = by$$

- Each component multiplied by a scalar
- Uniform scaling - same scalar for each component

2D Planar Transformations



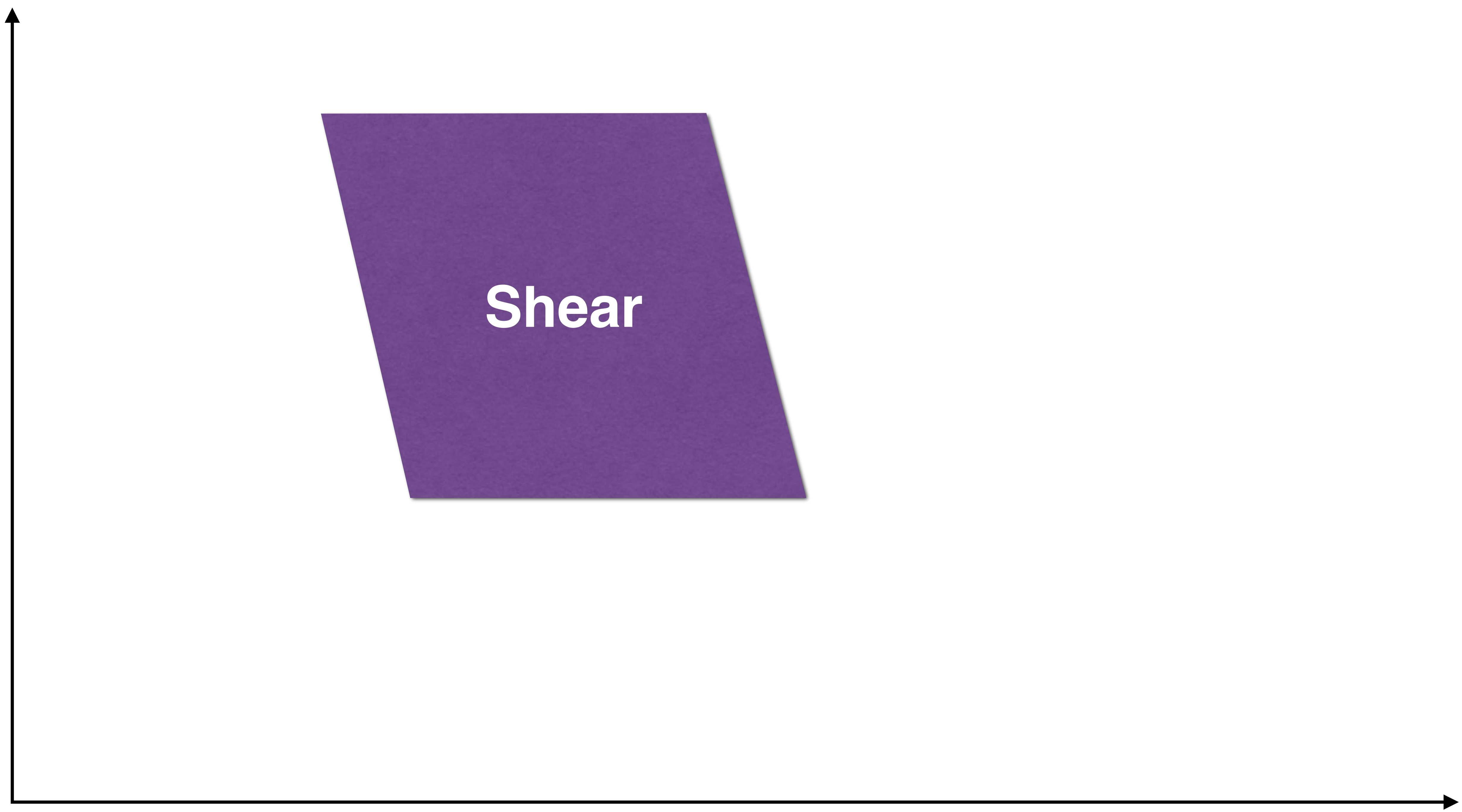
Scale

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

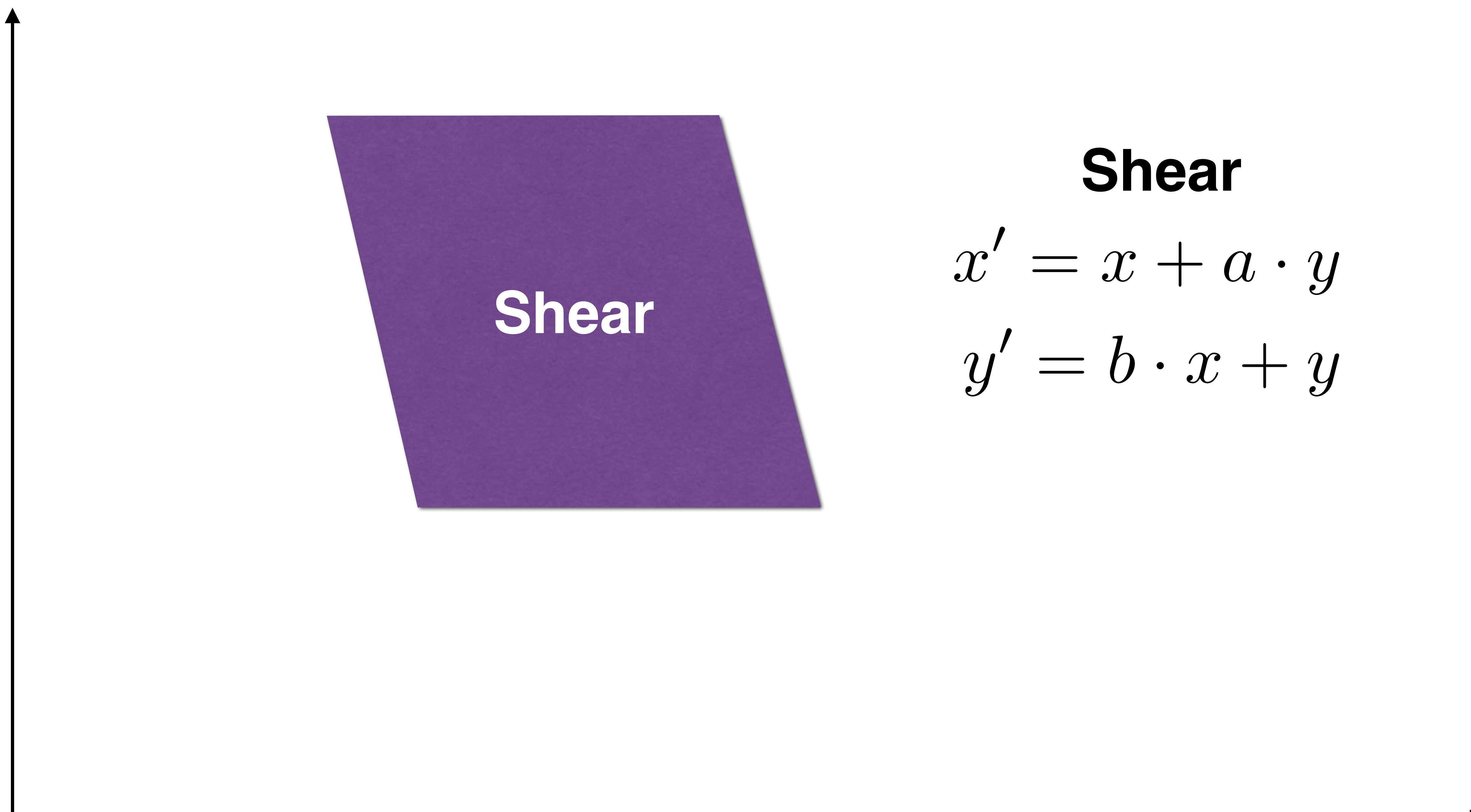
scaling matrix S

- Each component multiplied by a scalar
- Uniform scaling - same scalar for each component

2D Planar Transformations



2D Planar Transformations

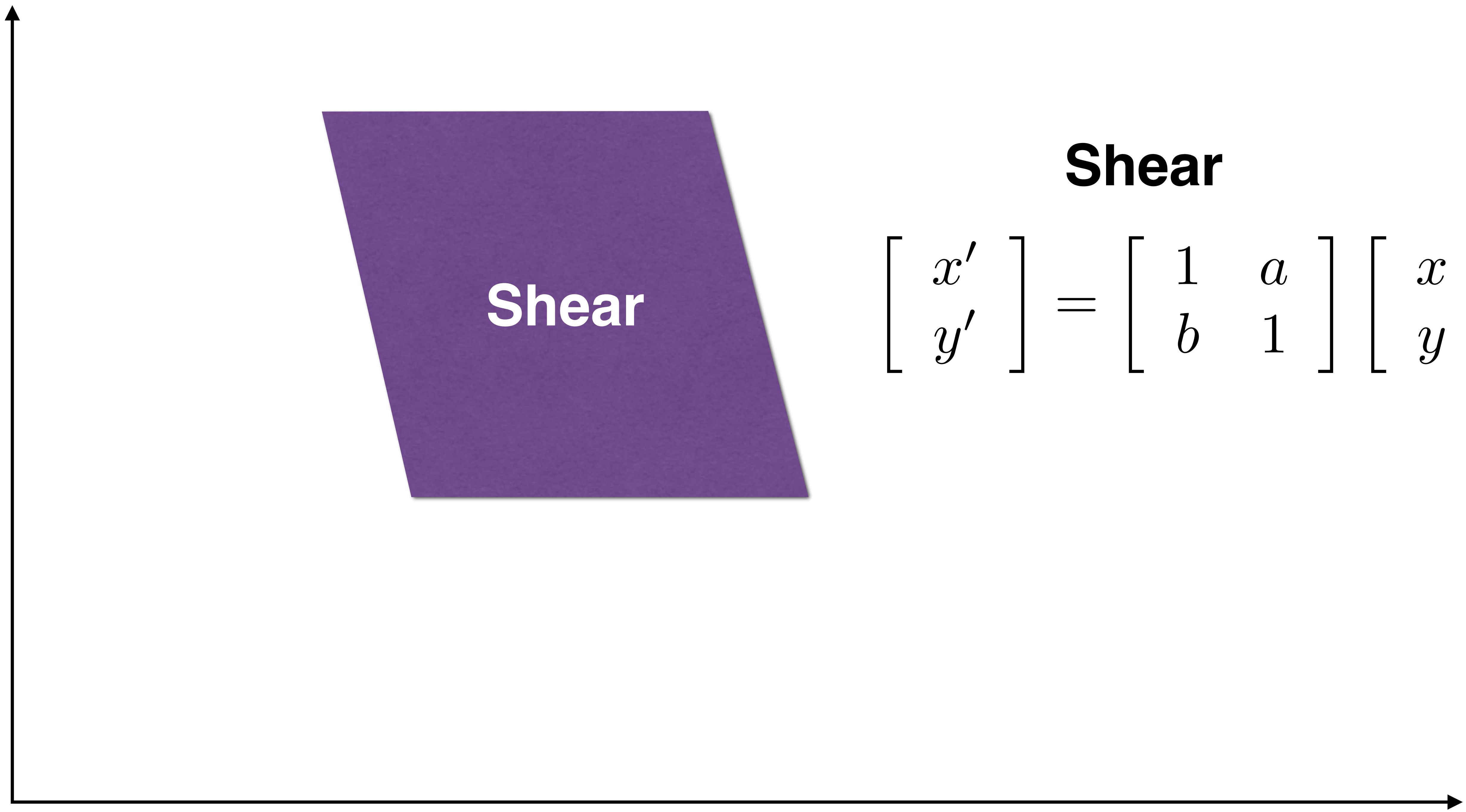


Shear

$$x' = x + a \cdot y$$

$$y' = b \cdot x + y$$

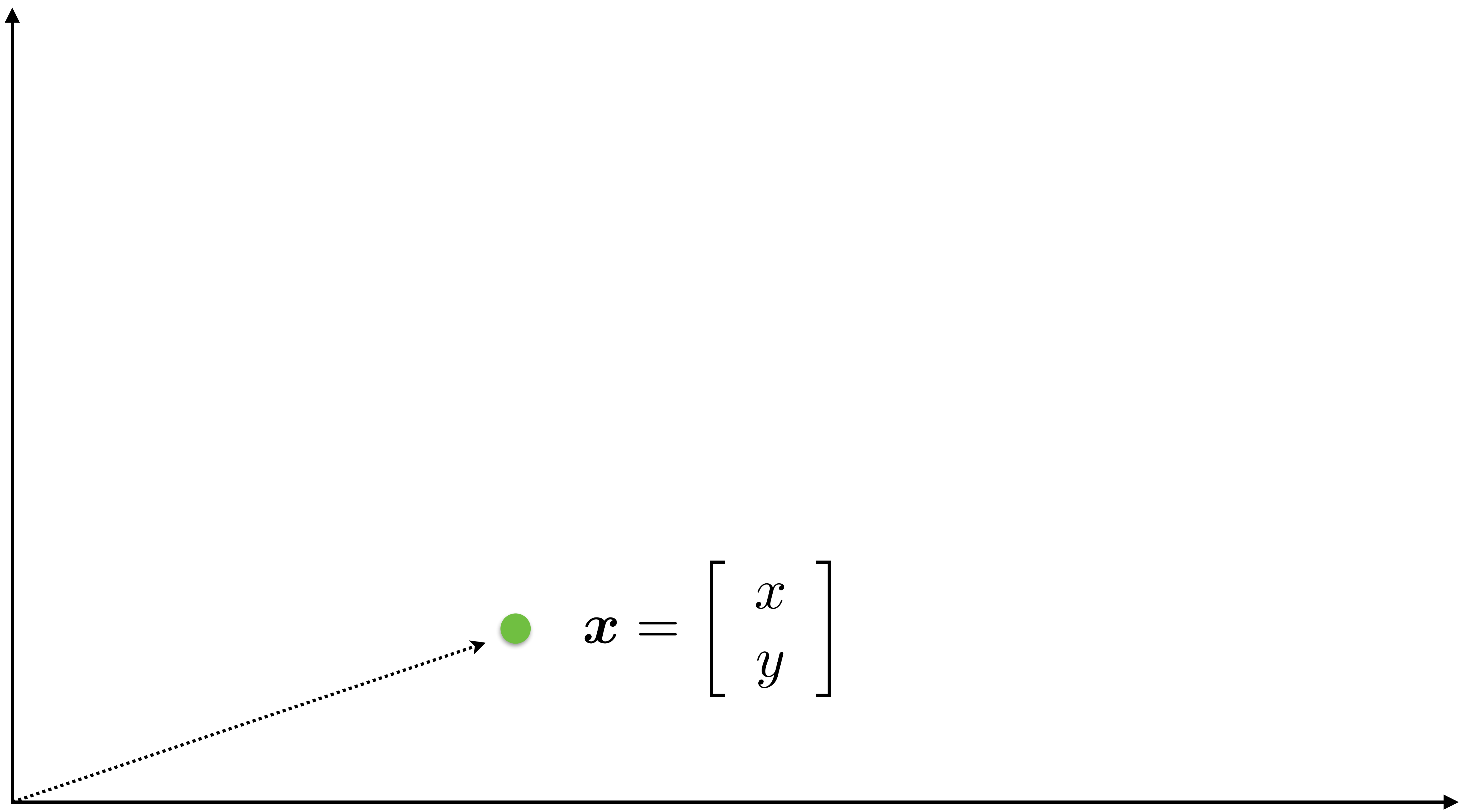
2D Planar Transformations



Shear

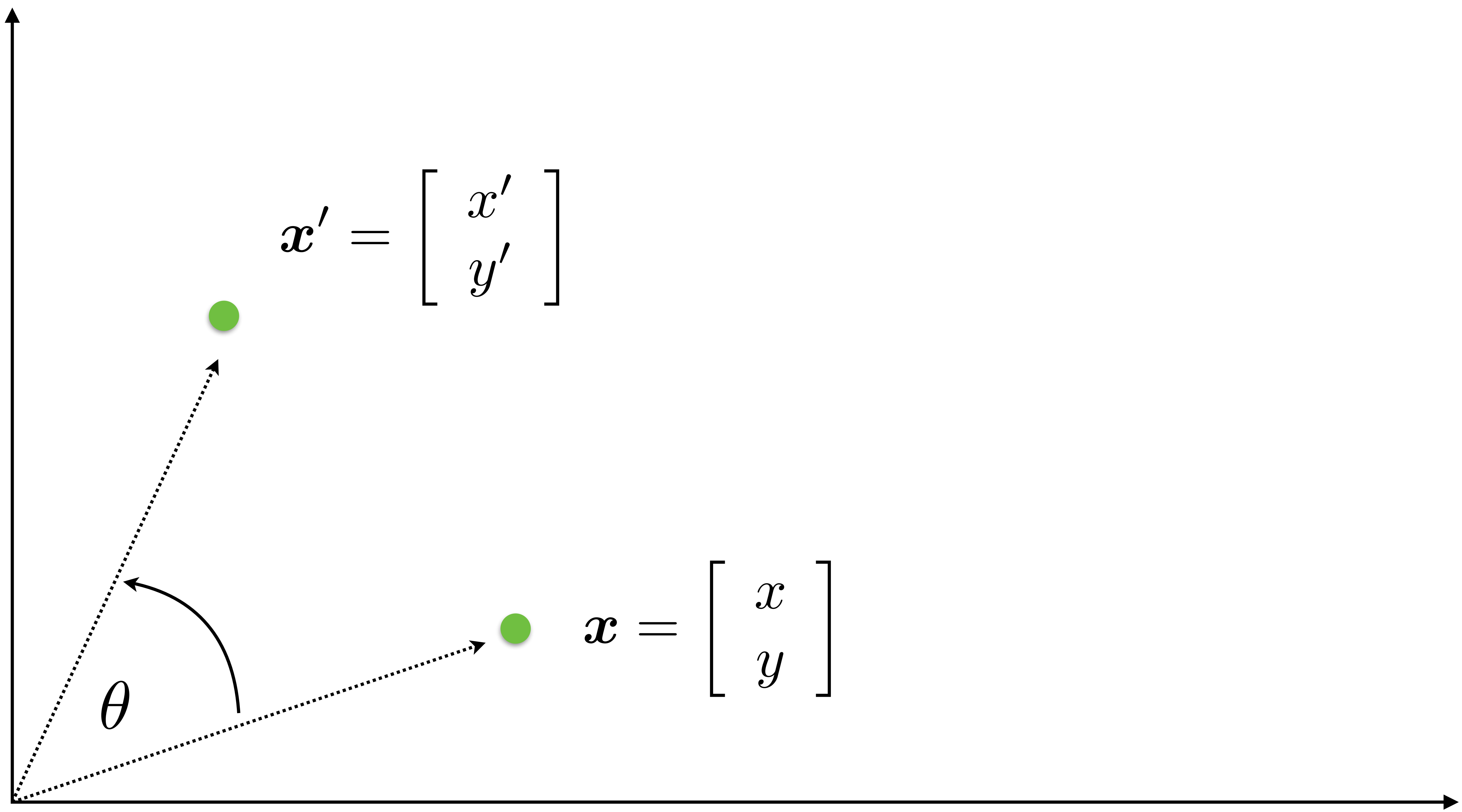
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Planar Transformations

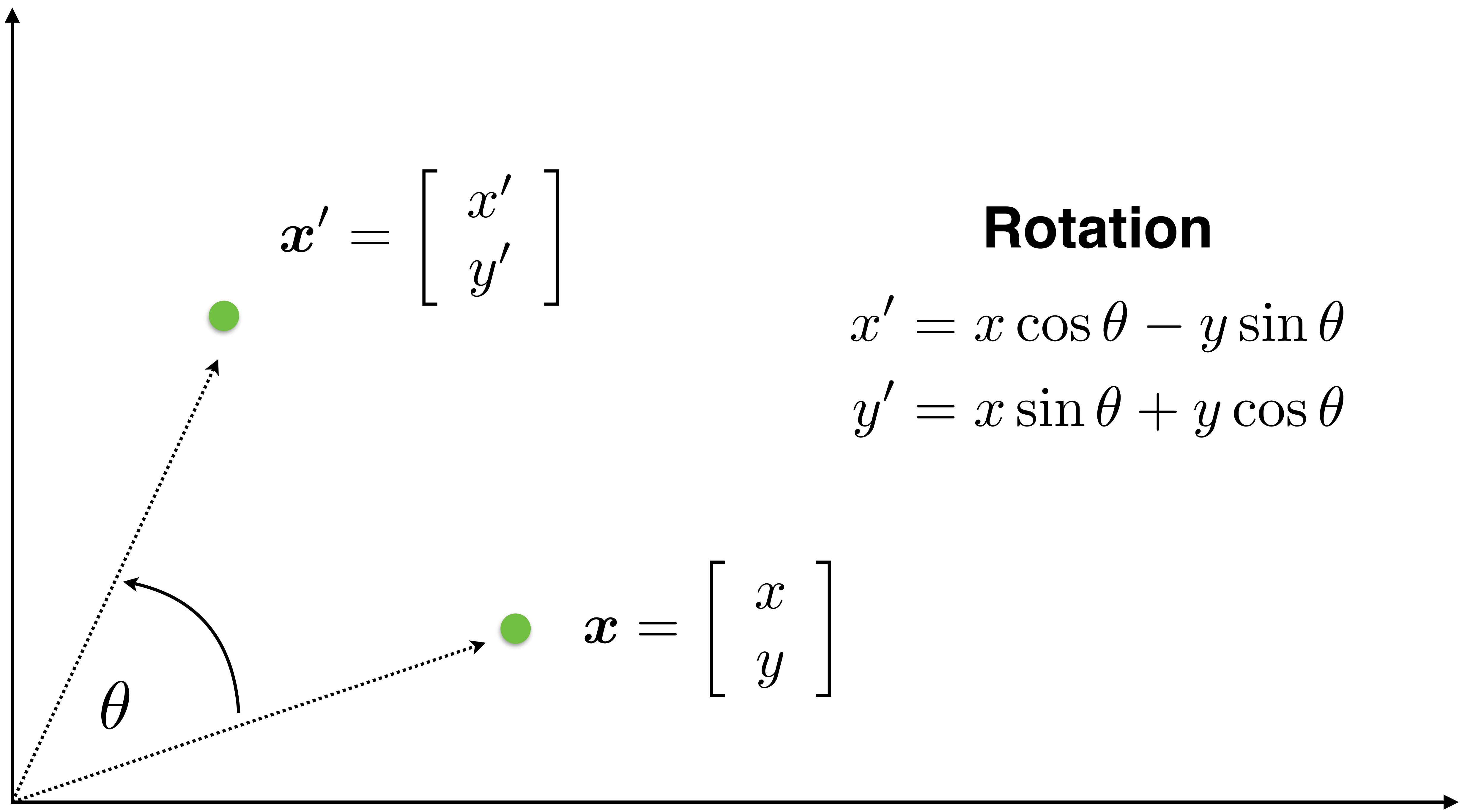


$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Planar Transformations



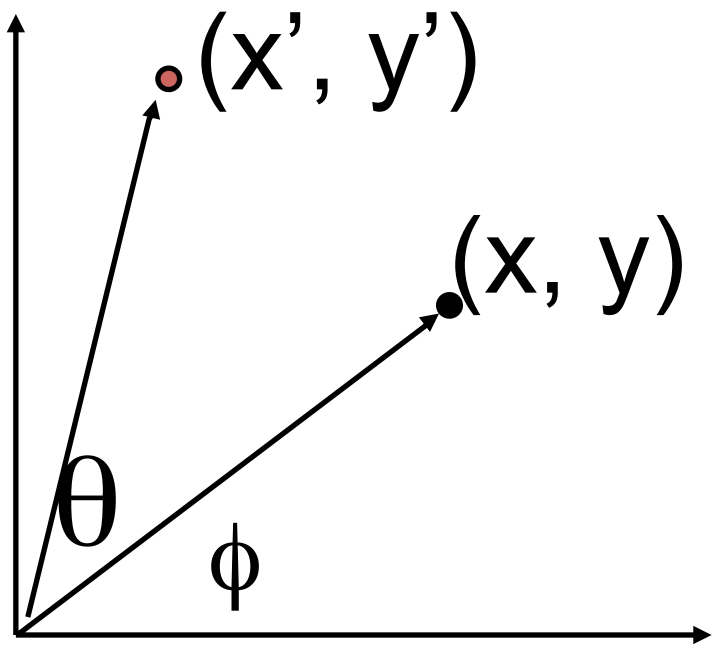
2D Planar Transformations



Rotation

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$



Polar coordinates…

$x = r \cos(\phi)$

$y = r \sin(\phi)$

$x' = r \cos(\phi + \theta)$

$y' = r \sin(\phi + \theta)$

Trig Identity…

$x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta)$

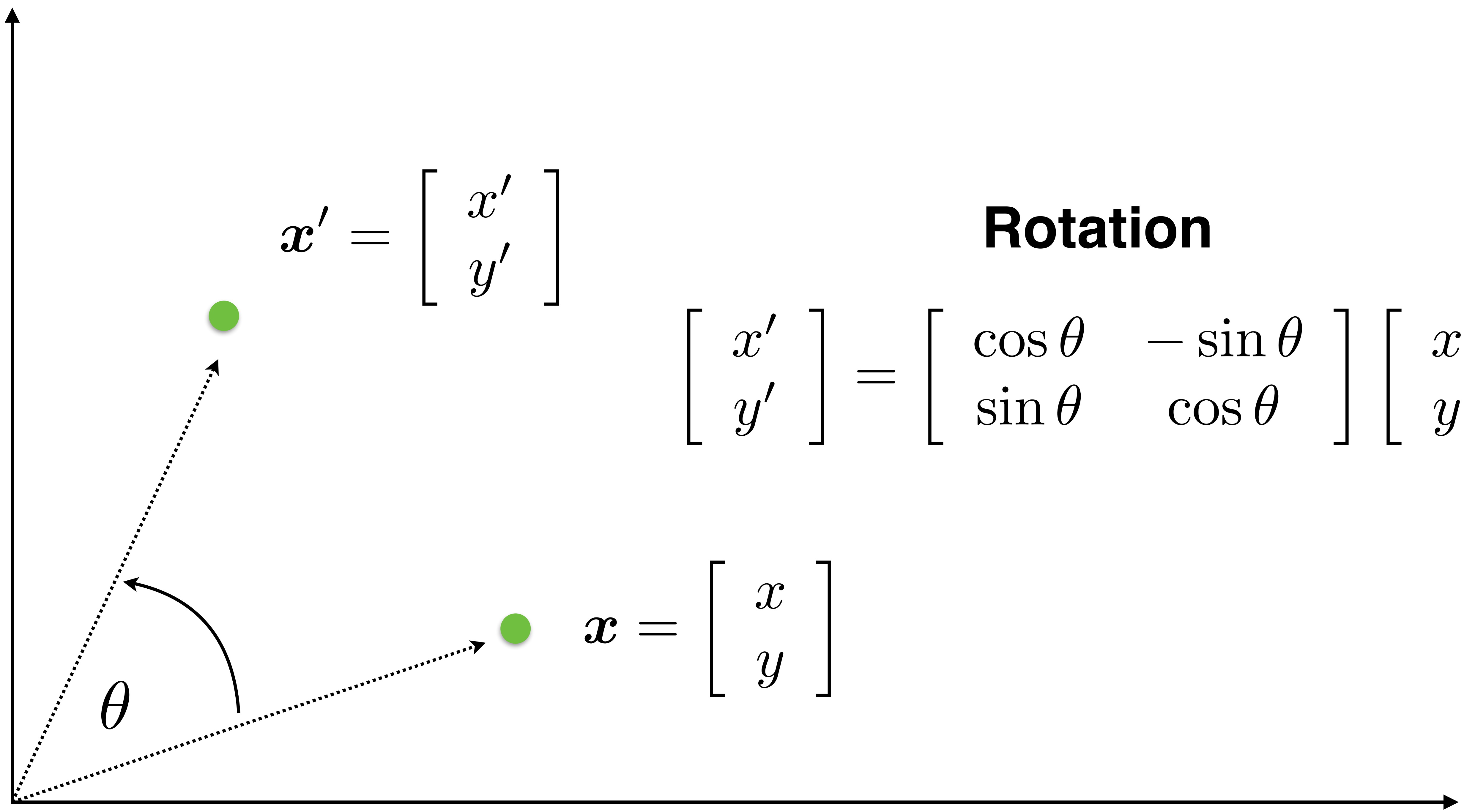
$y' = r \sin(\phi) \cos(\theta) + r \cos(\phi) \sin(\theta)$

Substitute…

$x' = x \cos(\theta) - y \sin(\theta)$

$y' = x \sin(\theta) + y \cos(\theta)$

2D Planar Transformations



Rotation

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

2D linear transformation

(can be written in matrix form)

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{p})$$

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

parameters $\boldsymbol{p}$

point $\boldsymbol{x}$

Scale

$$\mathbf{M} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

Flip across y

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Rotate

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

Flip across origin

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

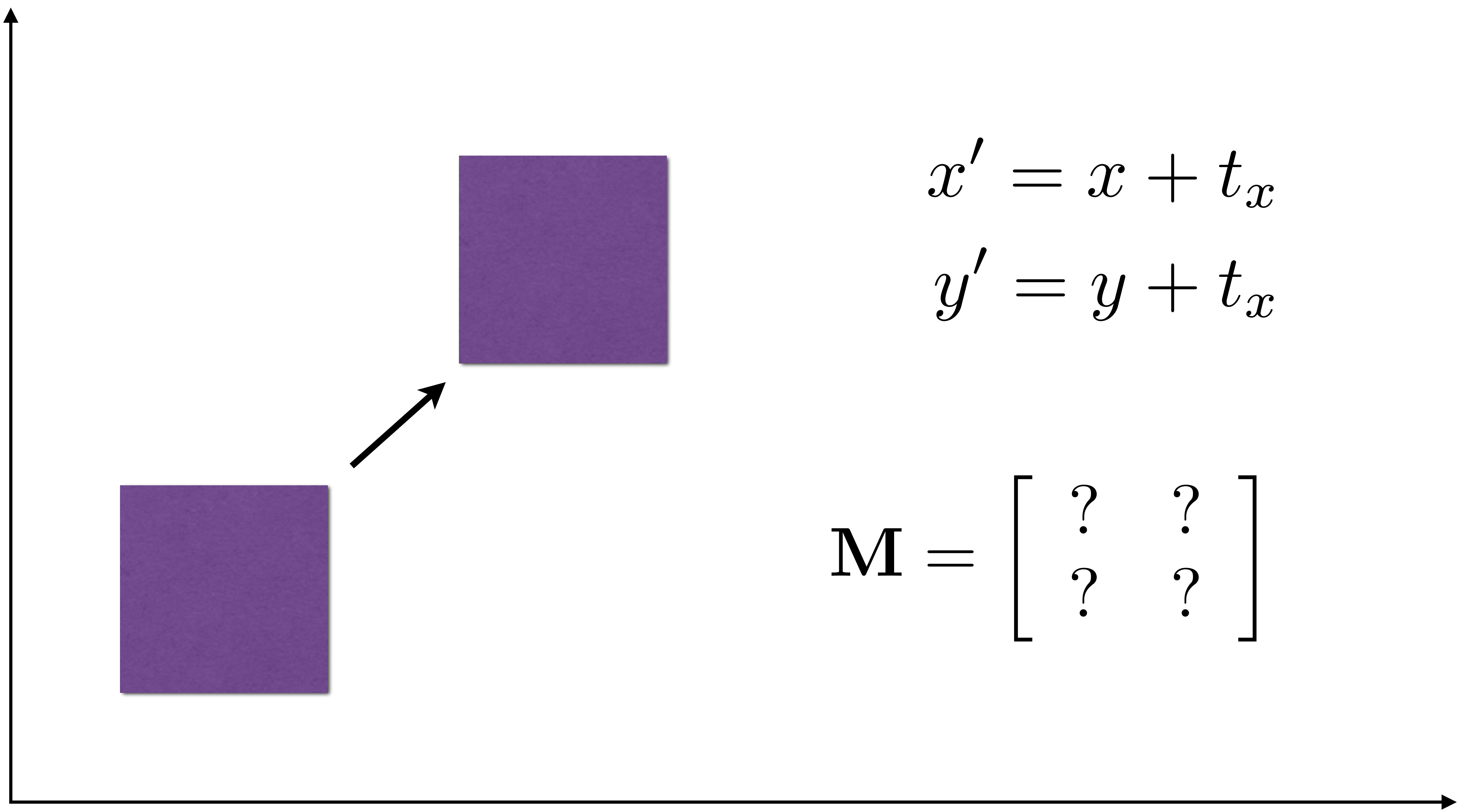
Shear

$$\mathbf{M} = \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix}$$

Identity

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

How do you represent translation with a 2 x 2 matrix?

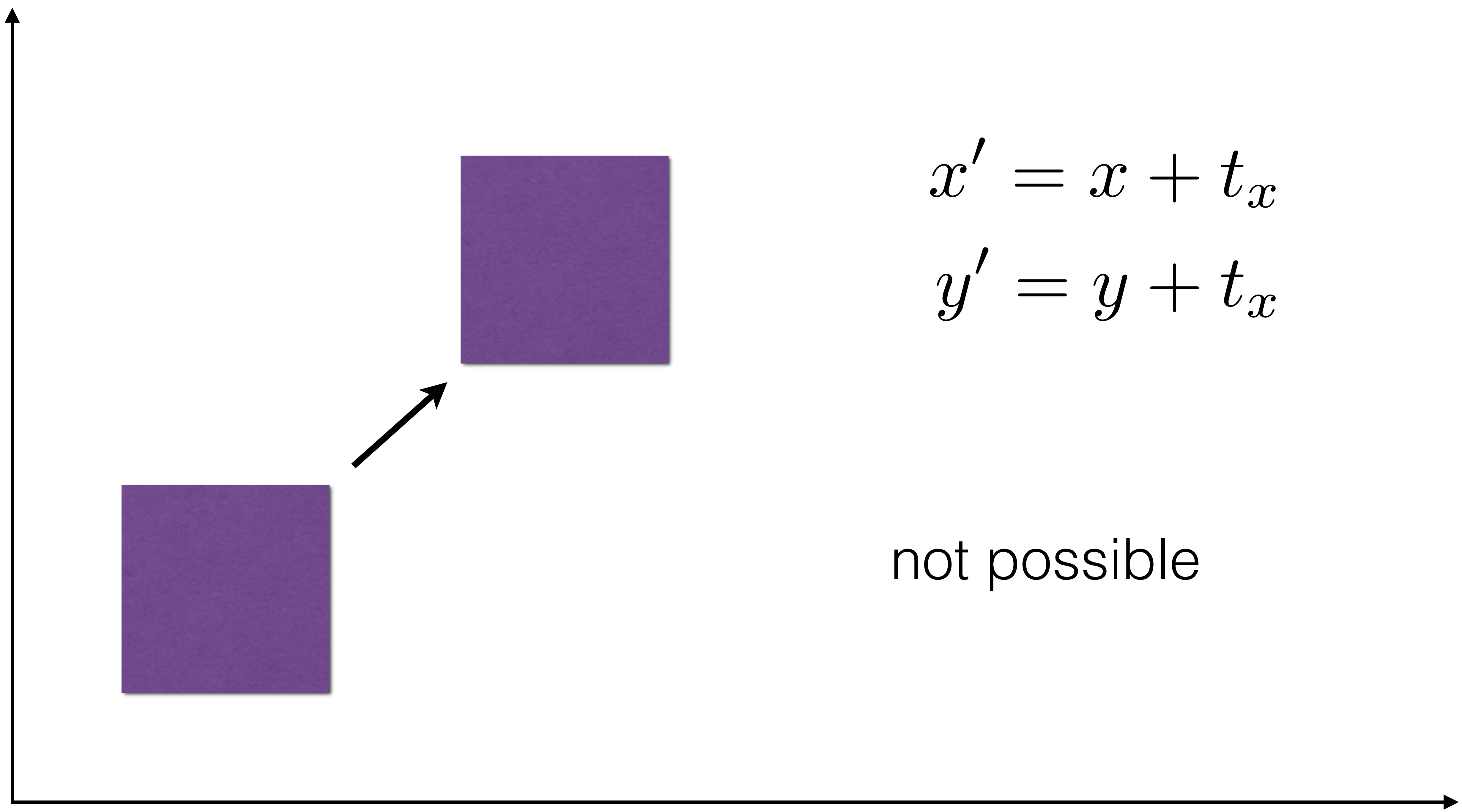


$$x' = x + t_x$$

$$y' = y + t_x$$

$$\mathbf{M} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

How do you represent translation with a 2 x 2 matrix?



$$x' = x + t_x$$

$$y' = y + t_x$$

not possible

Q: How can we represent translation in matrix form?

$$x' = x + t_x$$

$$y' = y + t_y$$

Homogeneous Coordinates

add a one here

$$\begin{bmatrix} x \\ y \end{bmatrix} \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

inhomogenous coordinates

homogenous coordinates

Represent 2D point with a 3D vector

Q: How can we represent translation in matrix form?

$$x' = x + t_x$$

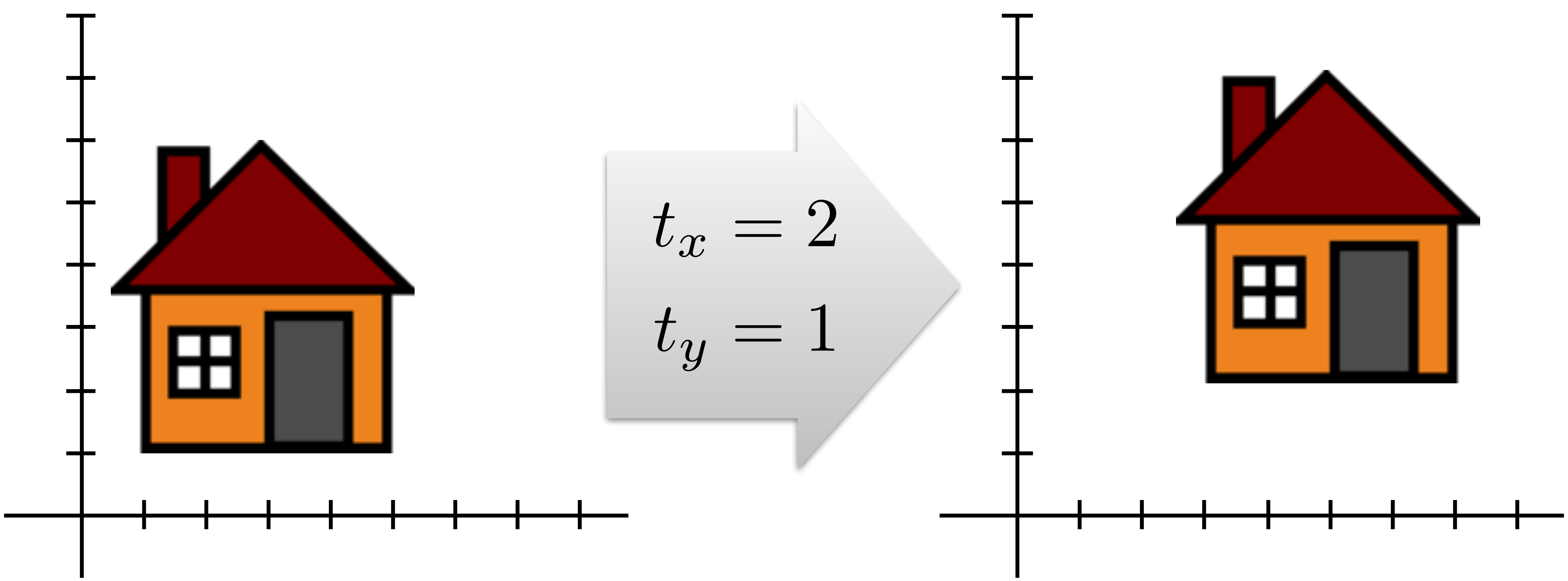
$$y' = y + t_y$$

A: append 3rd element and append 3rd column & row

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Coordinates

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



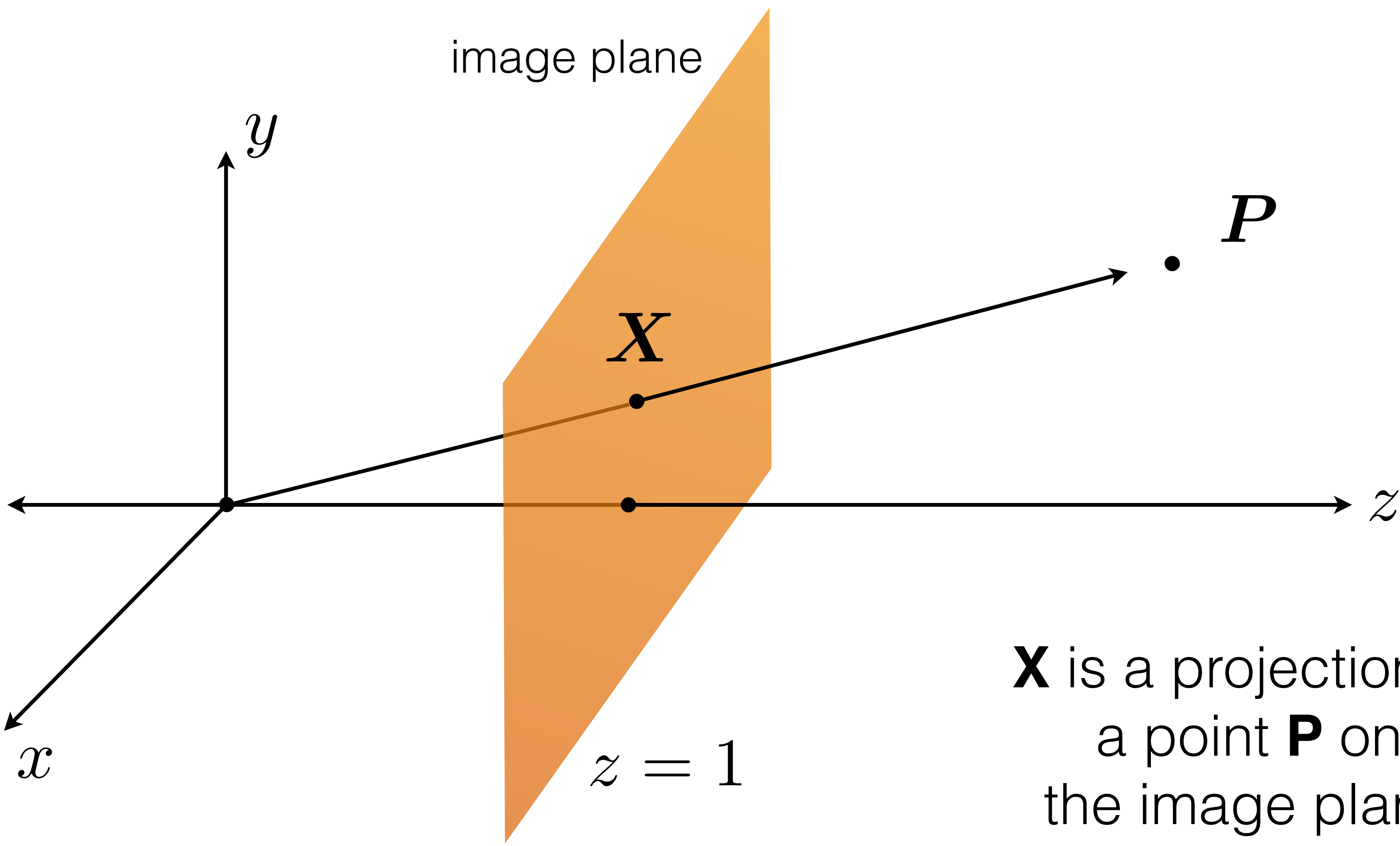
A 2D point in an image can be represented as a 3D vector

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \iff \boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where $$x = \frac{x_1}{x_3} \qquad y = \frac{x_2}{x_3}$$

Why?

Think of a point on the image plane in 3D



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

You can think of a conversion to homogenous coordinates as a conversion of a **point** to a **ray**

Conversion:

- 2D point → homogeneous point

  append 1 as 3rd coordinate

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- homogeneous point → 2D point

  divide by 3rd coordinate

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

Special Properties

- Scale invariant $\begin{bmatrix} x & y & w \end{bmatrix}^\top = \lambda \begin{bmatrix} x & y & w \end{bmatrix}^\top$

- Point at infinity $\begin{bmatrix} x & y & 0 \end{bmatrix}$

- Undefined $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$

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

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

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

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

Shear

Matrix Composition

Transformations can be combined 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(Θ) S(s_x,s_y) p

Does the order of multiplication matter?

2D transformations

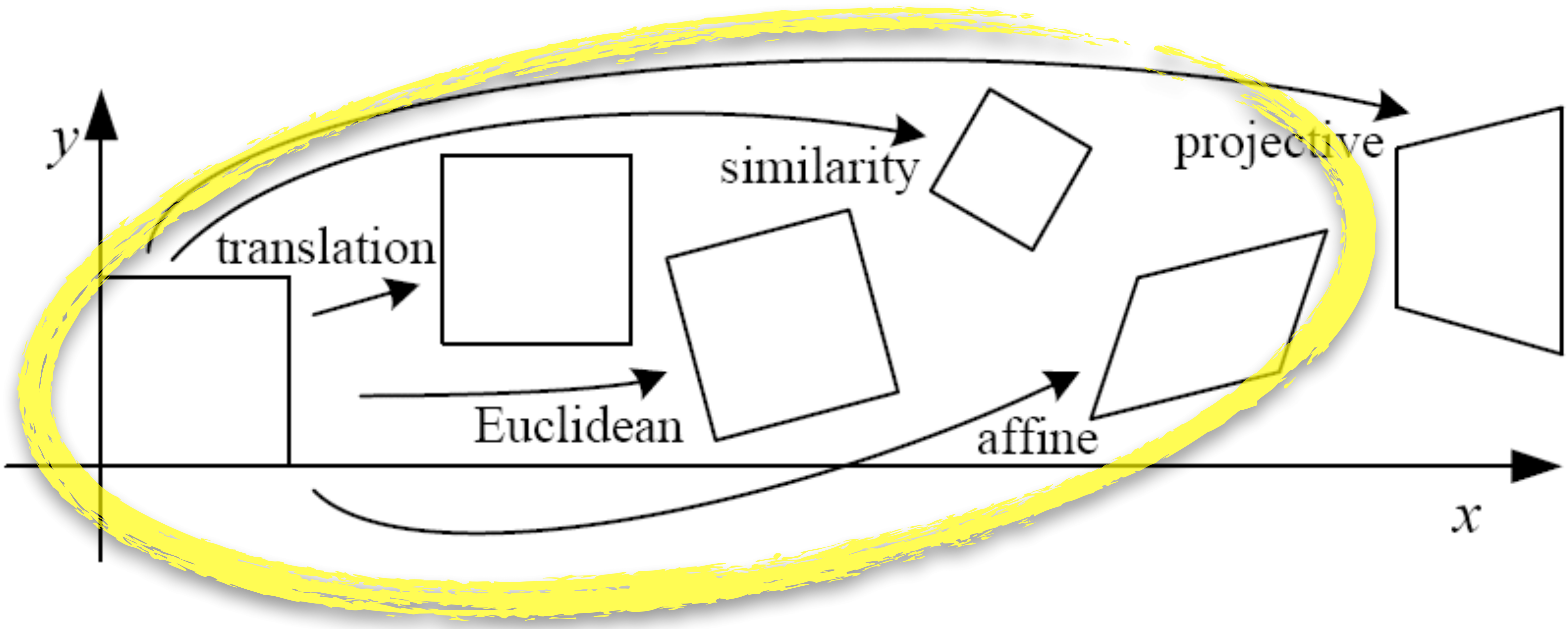


Figure 1: Basic set of 2D planar transformations

Name	Matrix	# D.O.F.
translation	$\begin{bmatrix} \boldsymbol{I} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3
similarity	$\begin{bmatrix} s\boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8

Affine Transformation

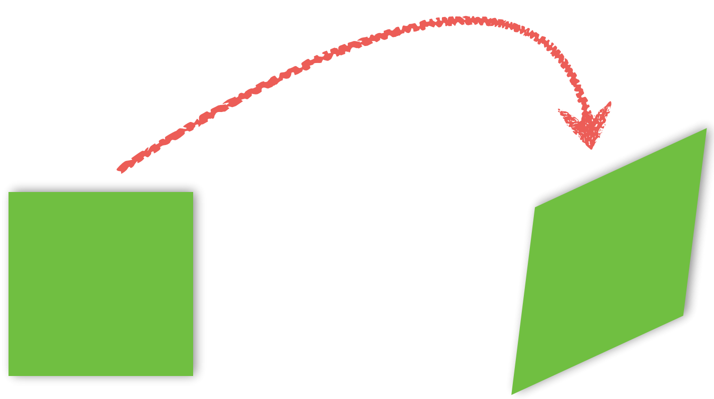
Affine transformations are combinations of

- Linear transformations, and
- Translations

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

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



Will the last coordinate w ever change?

Coming soon…

Projective Transform

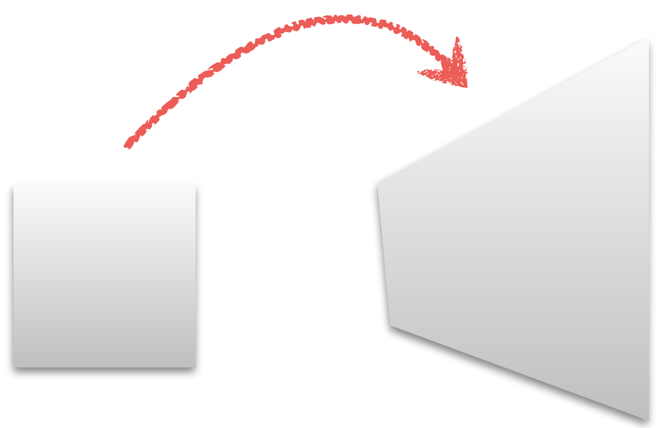
Projective transformations are combos of

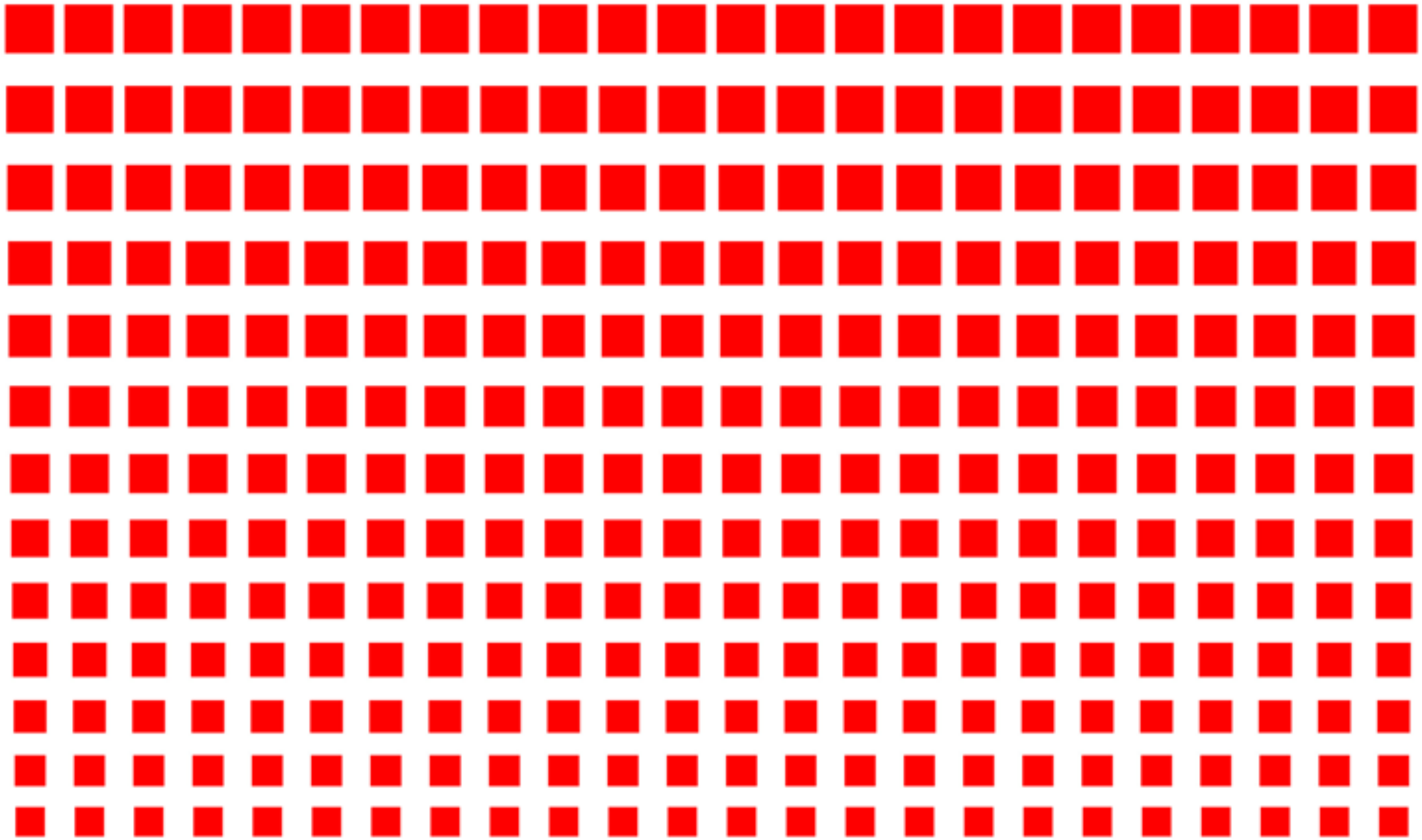
- Affine transformations, and
- Projective warps

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

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
- Projective matrix is defined up to a scale (8 DOF)





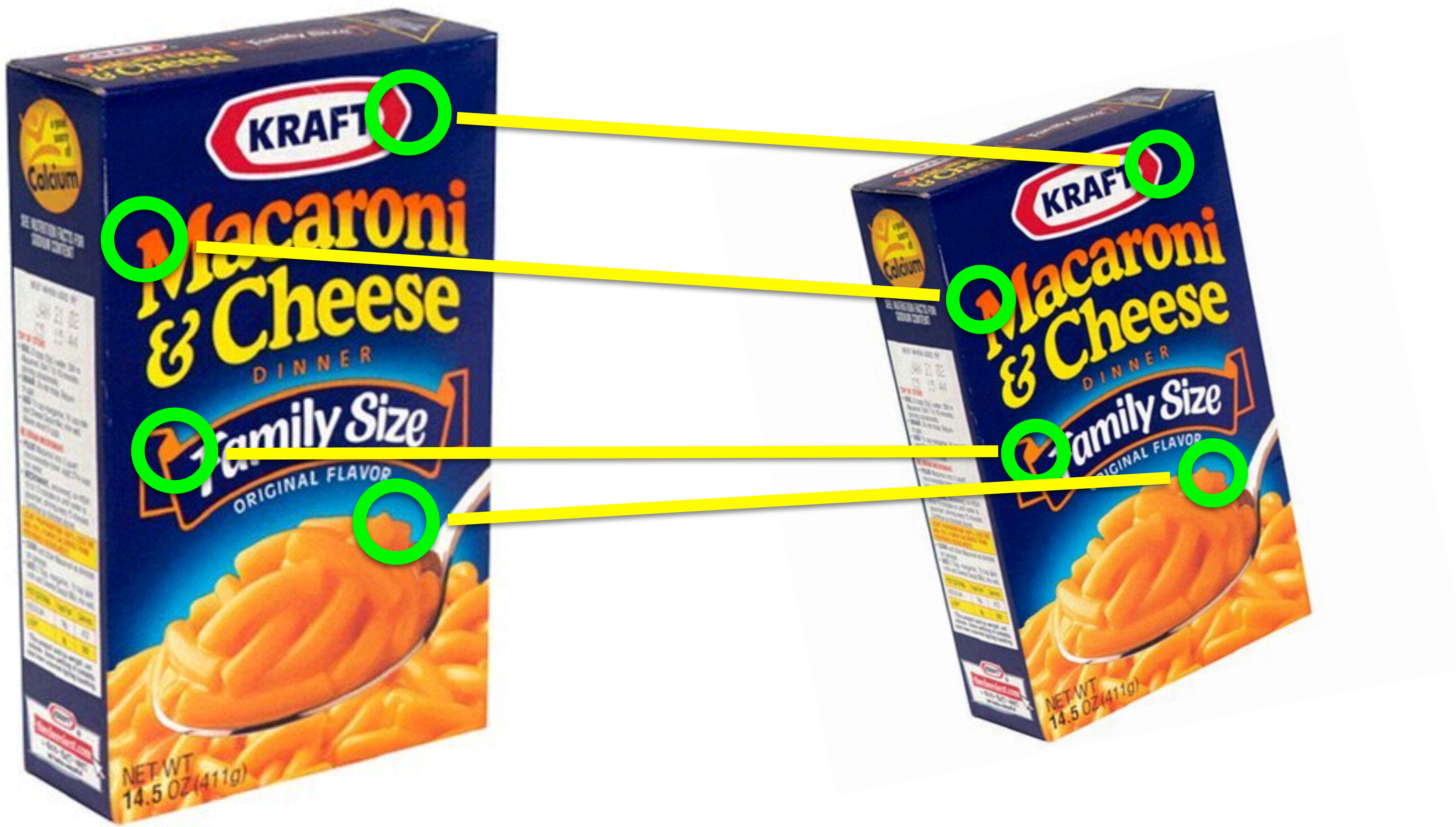
2D Alignment: Linear Least Squares

16-385 Computer Vision (Kris Kitani)

Extract features from an image …

what do we do next?

Feature matching
(object recognition, 3D reconstruction, augmented reality, image stitching)



How do we estimate the transformation?

Given a set of matched feature points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

point in one image — point in the other image

and a transformation

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p})$$

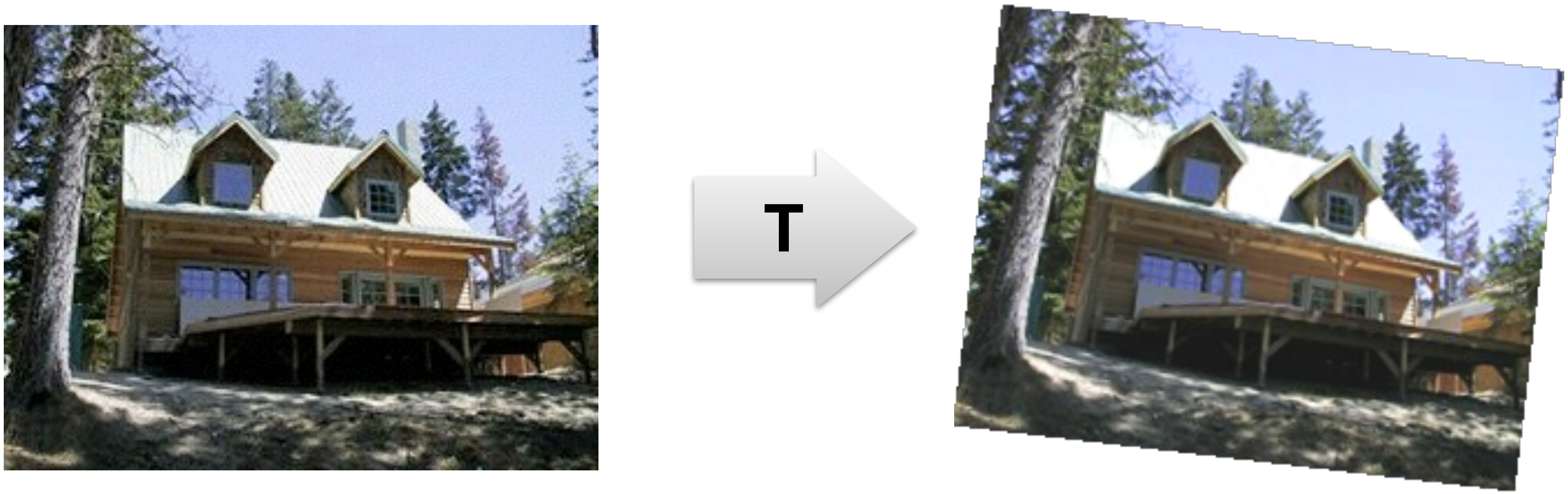
transformation function — parameters

Find the best estimate of

$$\boldsymbol{p}$$

Model fitting

Recover the transformation



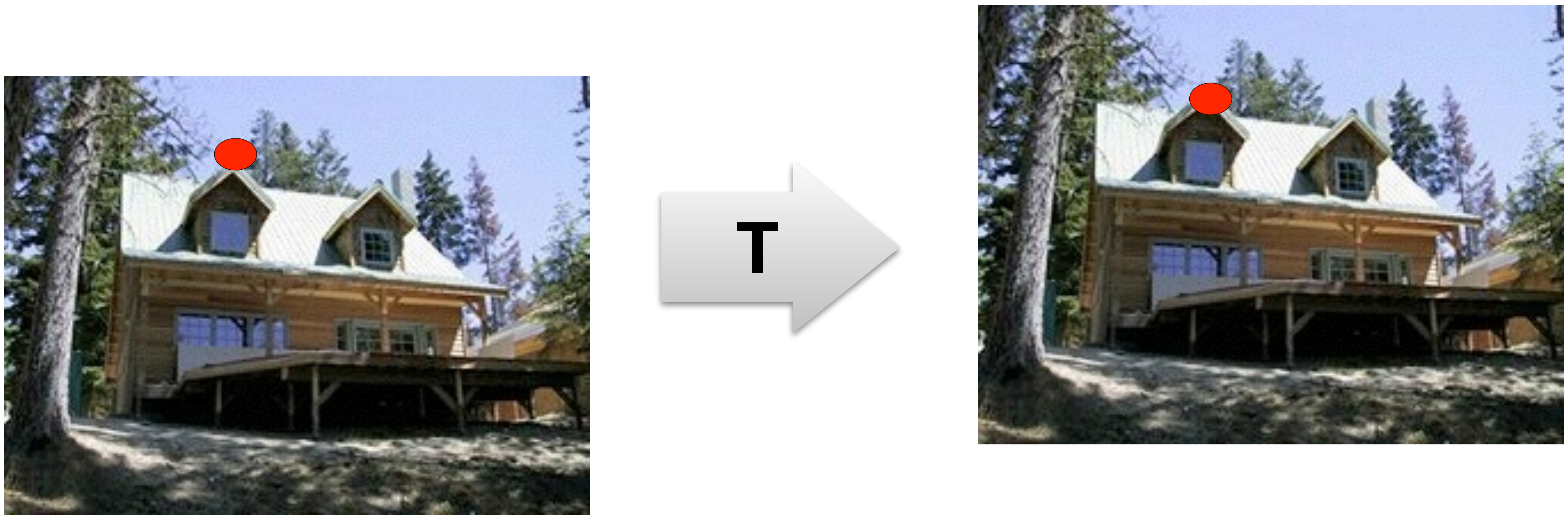
f(x,y) g(x,y)

Given f and g, how would you recover the transform T?

(user will provide correspondences)

How many do we need?

Translation

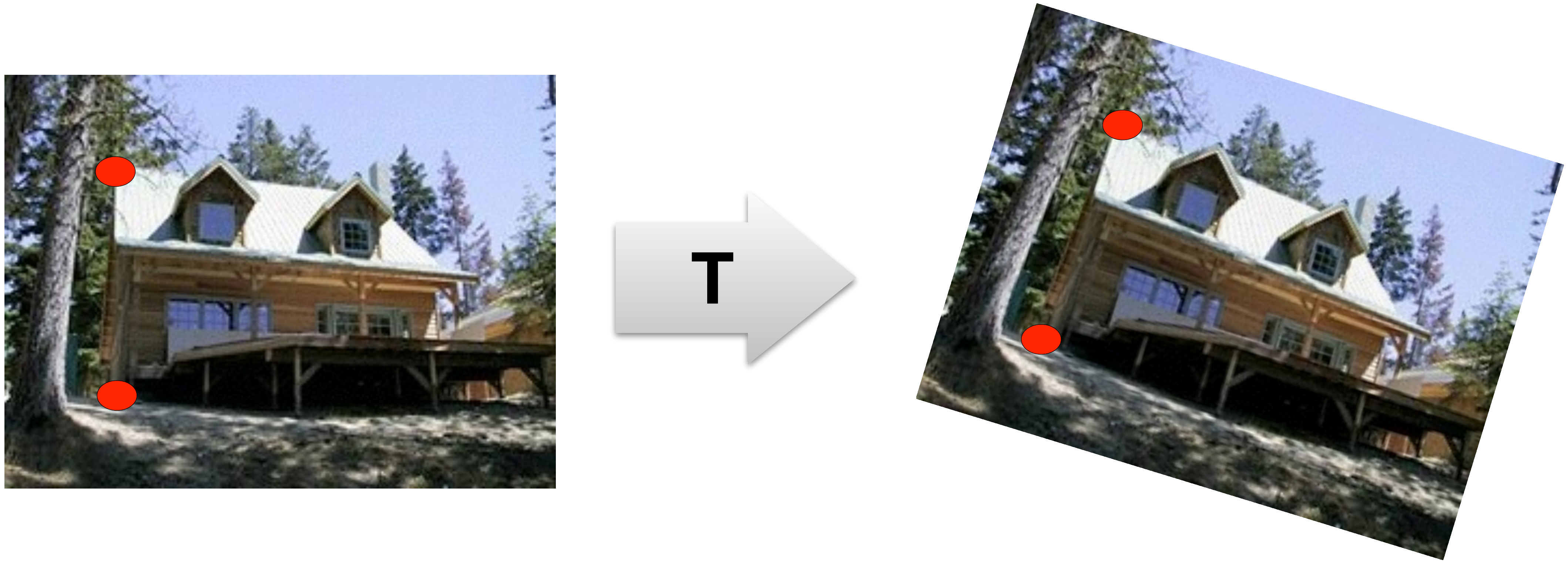


f(x,y) g(x,y)

- *How many Degrees of Freedom?*
- *How many correspondences needed?*
- *What is the transformation matrix?*

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & p'_x - p_x \\ 0 & 1 & p'_y - p_y \\ 0 & 0 & 1 \end{bmatrix}$$

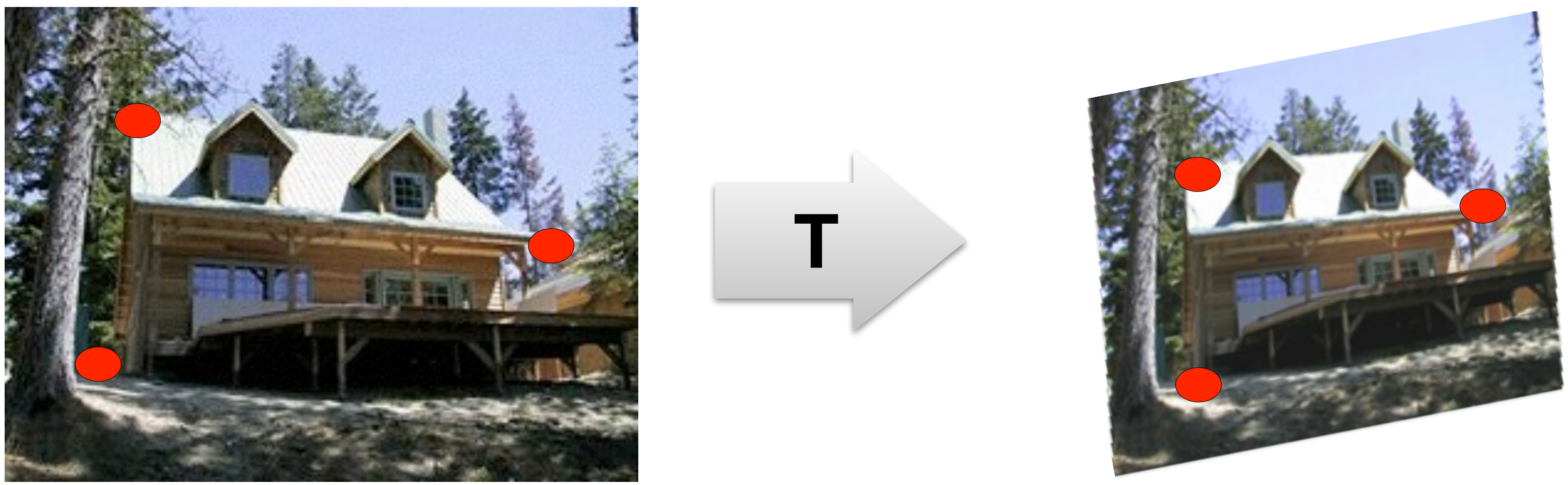
Euclidean



f(x,y) g(x,y)

- *How many Degrees of Freedom?*
- *How many correspondences needed for translation+rotation?*
- *What is the transformation matrix?*

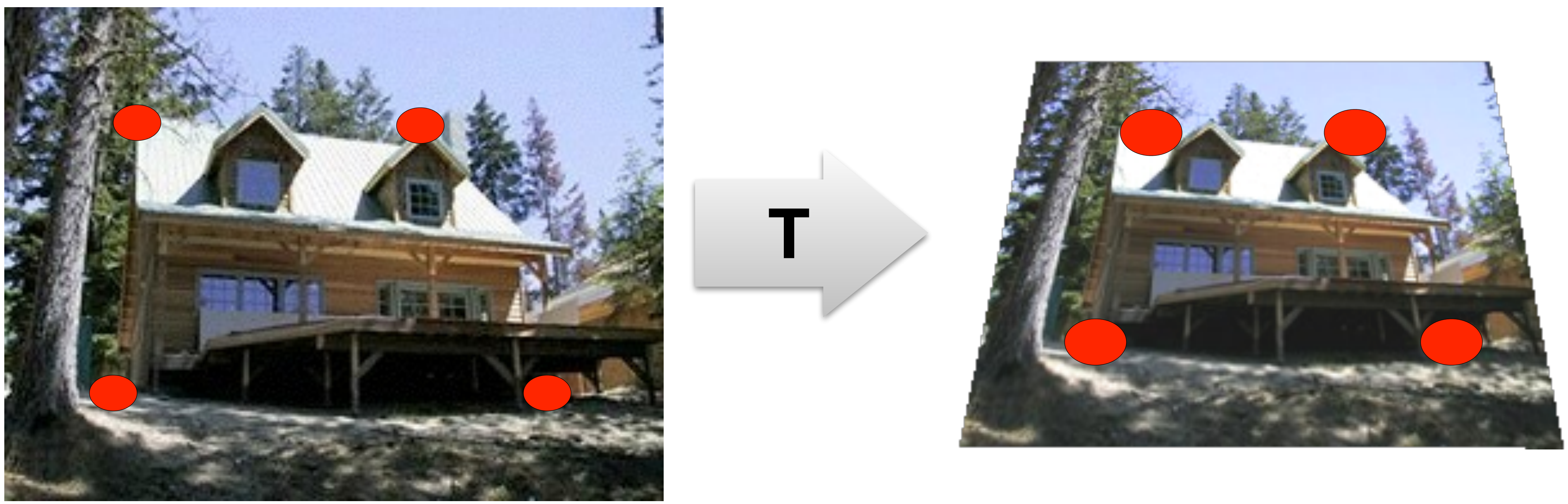
Affine



f(x,y) g(x,y)

- *How many Degrees of Freedom?*
- *How many correspondences needed for affine?*
- *What is the transformation matrix?*

Projective



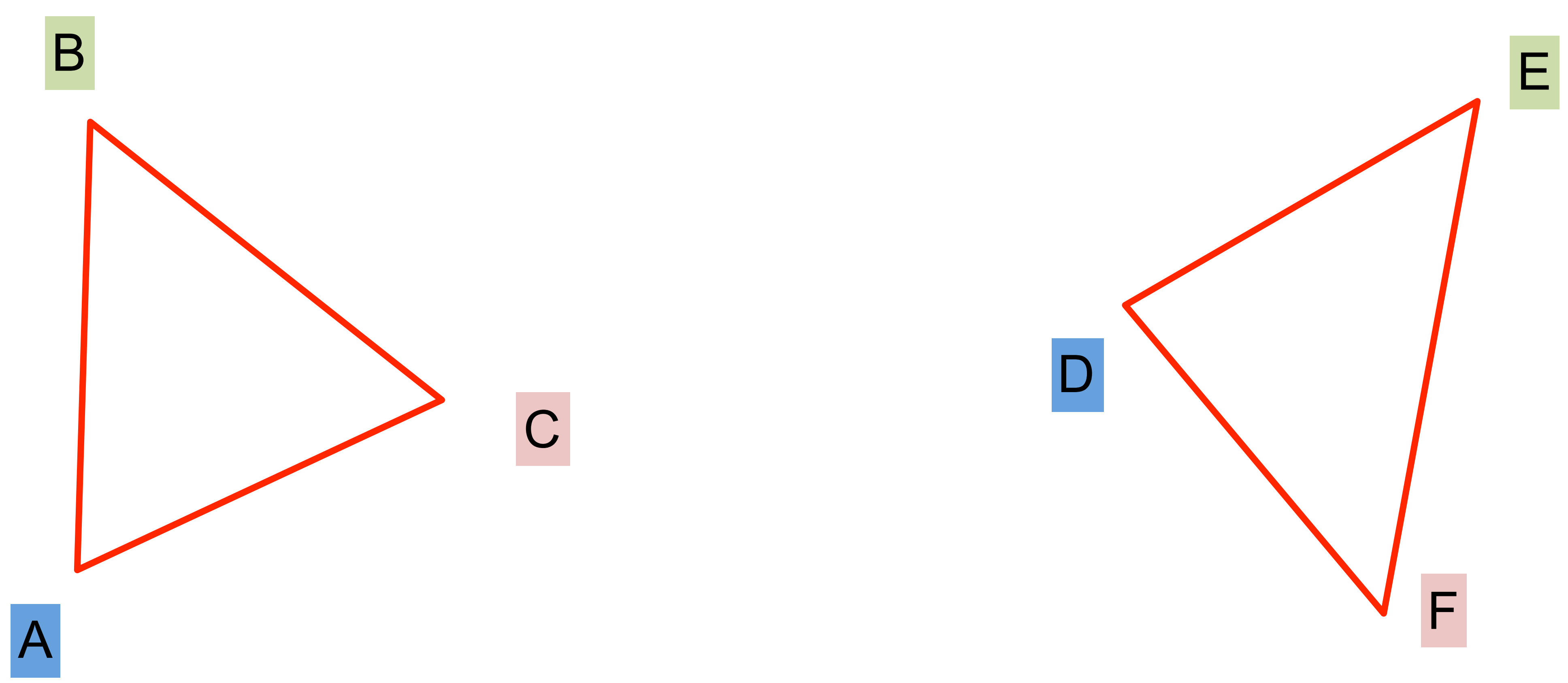
f(x,y) g(x,y)

- *How many Degrees of Freedom?*
- *How many correspondences needed for projective?*
- *What is the transformation matrix?*

Suppose we have two triangles: ABC and DEF.

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

How can we get the parameters?



Estimate transformation parameters using

Linear least squares

Given a set of matched feature points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

point in one image | point in the other image

and a transformation

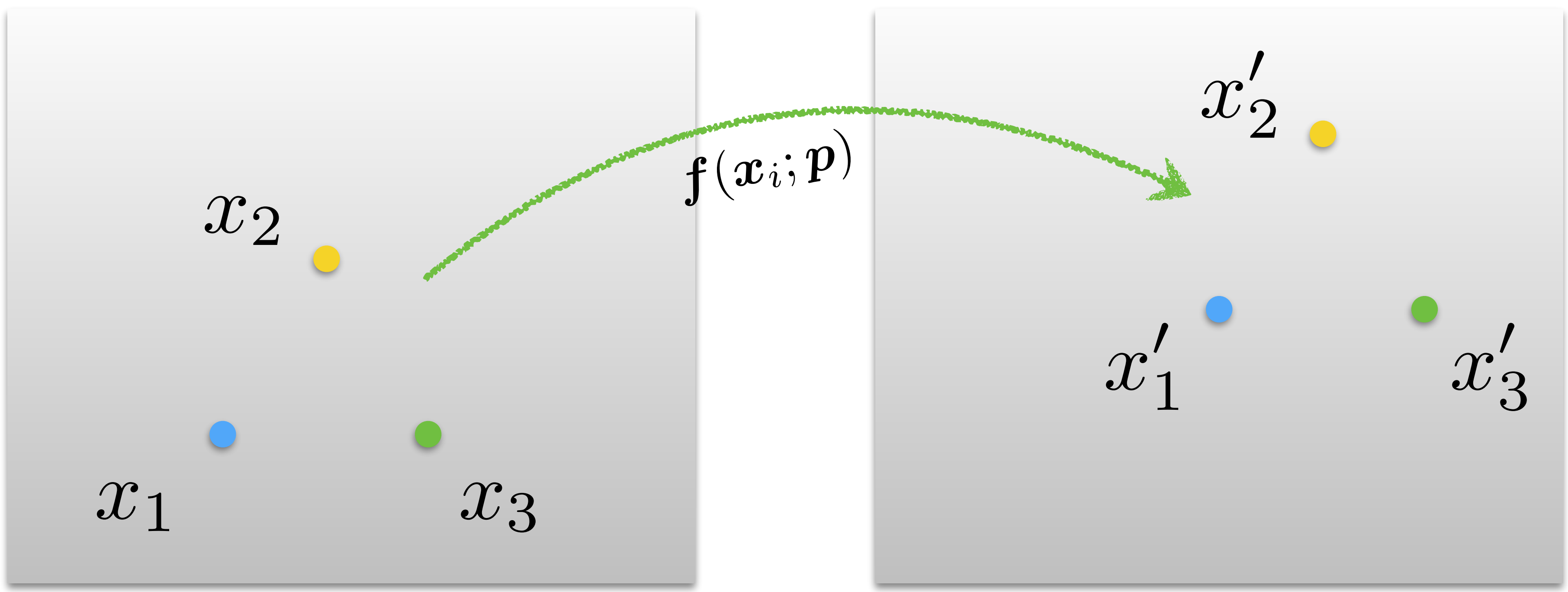
$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p})$$

transformation function | parameters

Find the best estimate of

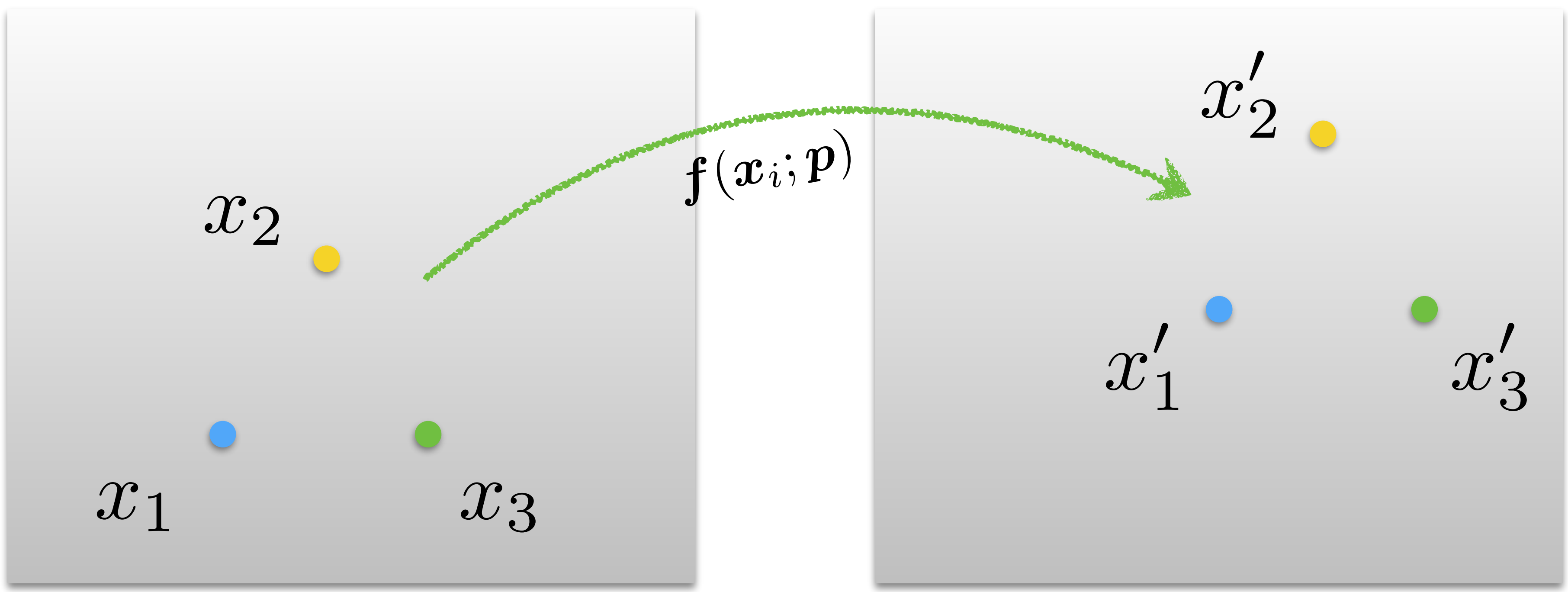
$$\boldsymbol{p}$$

Given point correspondences



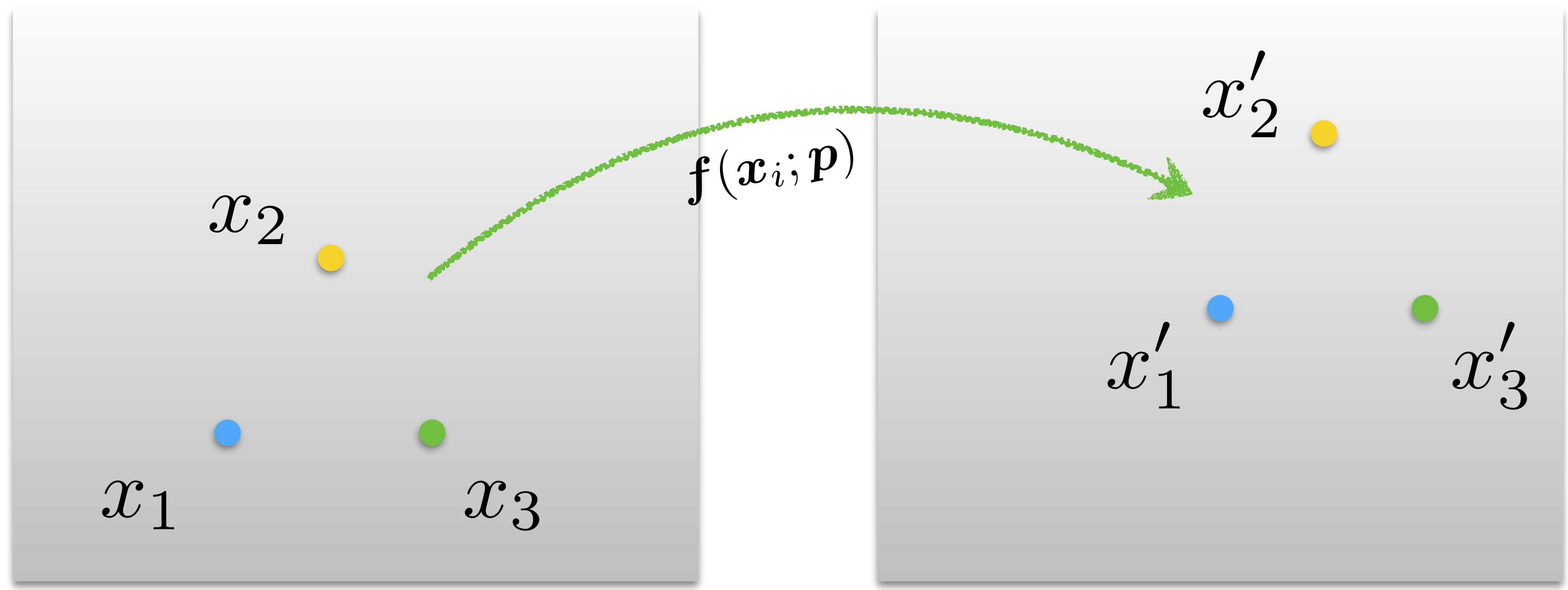
Given point correspondences

How can you solve for the transformation?



Least Squares Error

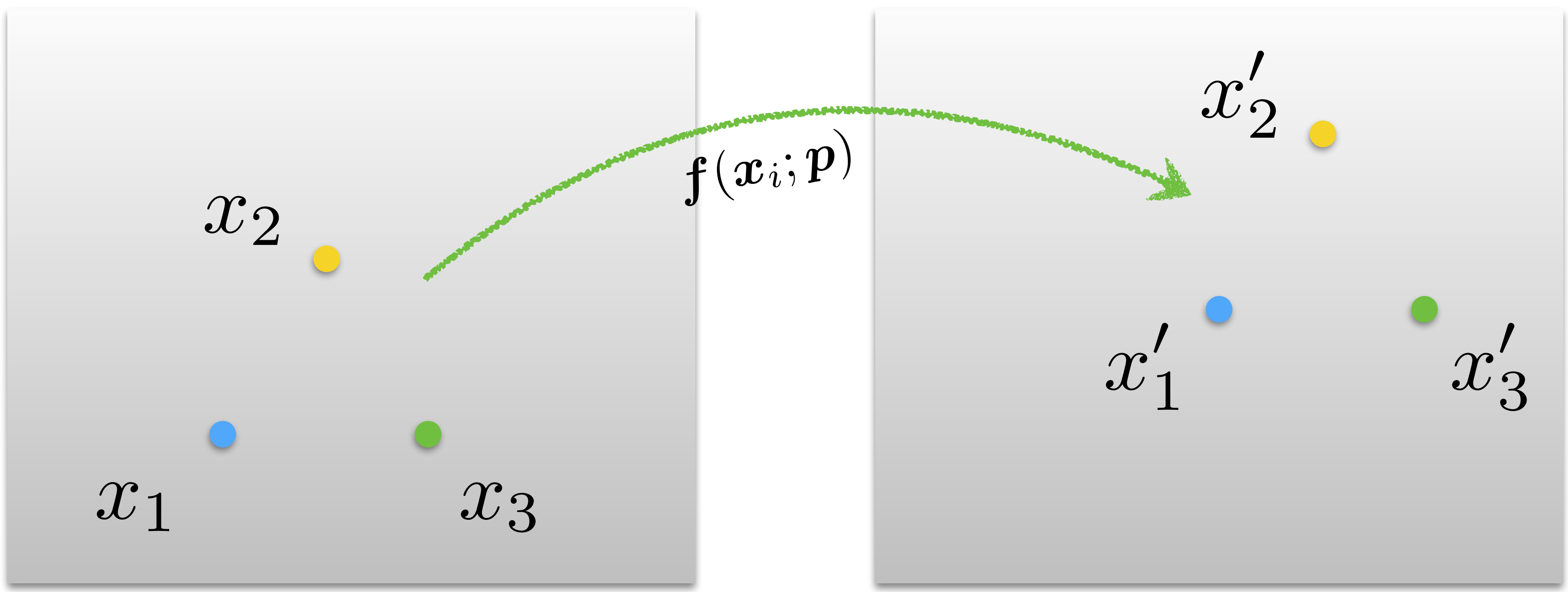
$$E_{\text{LS}} = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i;\boldsymbol{p}) - \boldsymbol{x}'_i\|^2$$



Least Squares Error

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

What is this? *What is this?*



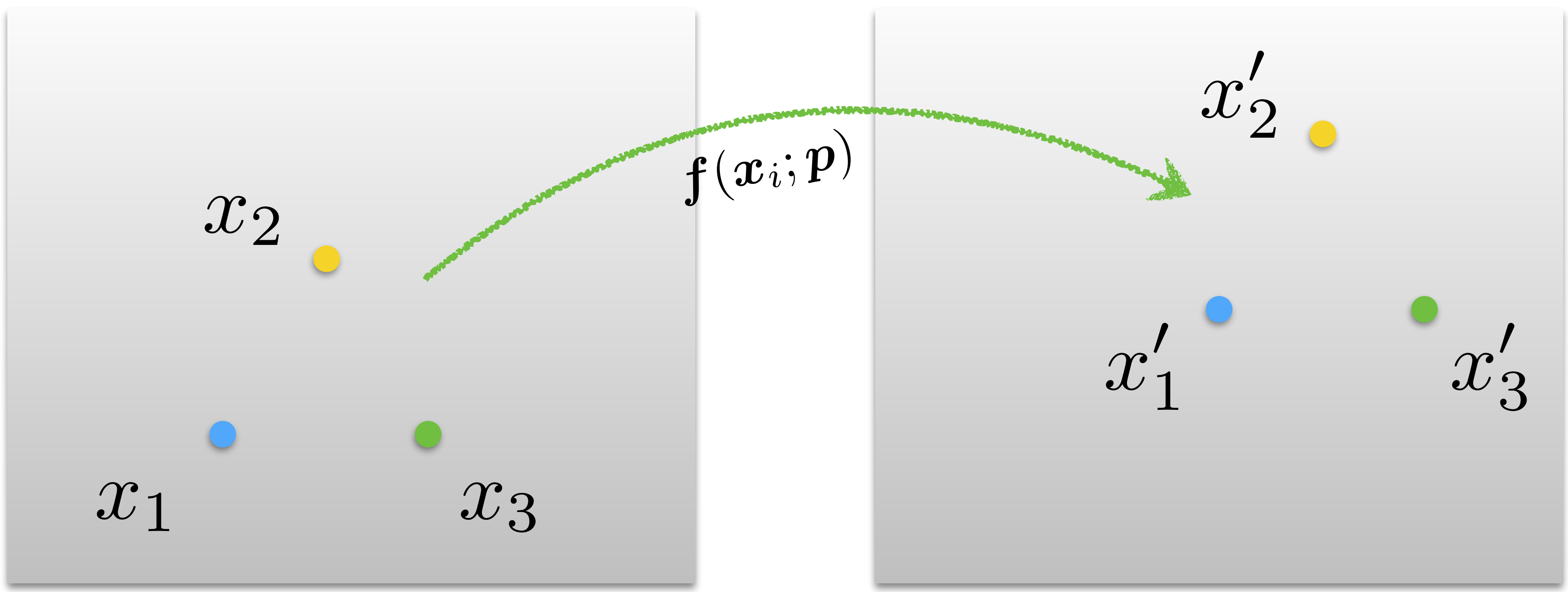
Least Squares Error

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

What is this?

What is this?

What is this?



Least Squares Error

Euclidean (L2) norm

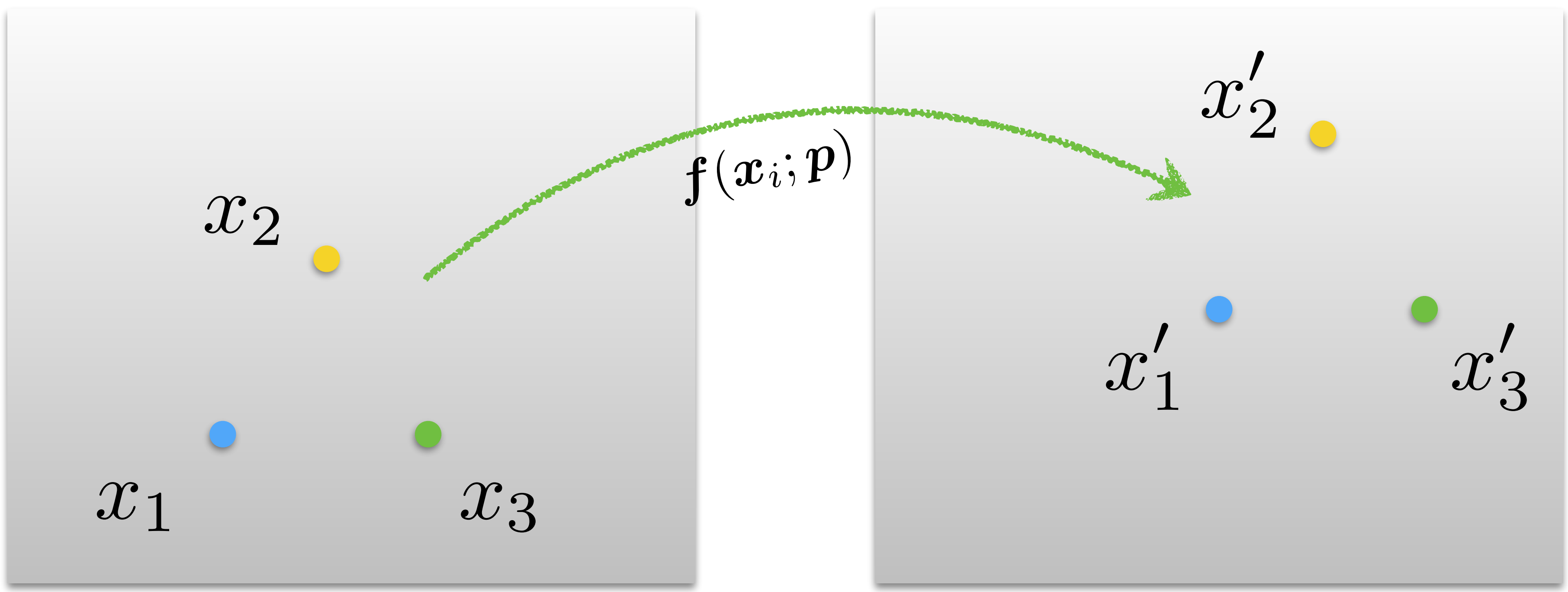
$$E_{\mathrm{LS}} = \sum_i \| \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i \|^2$$

squared!

predicted location

measured location

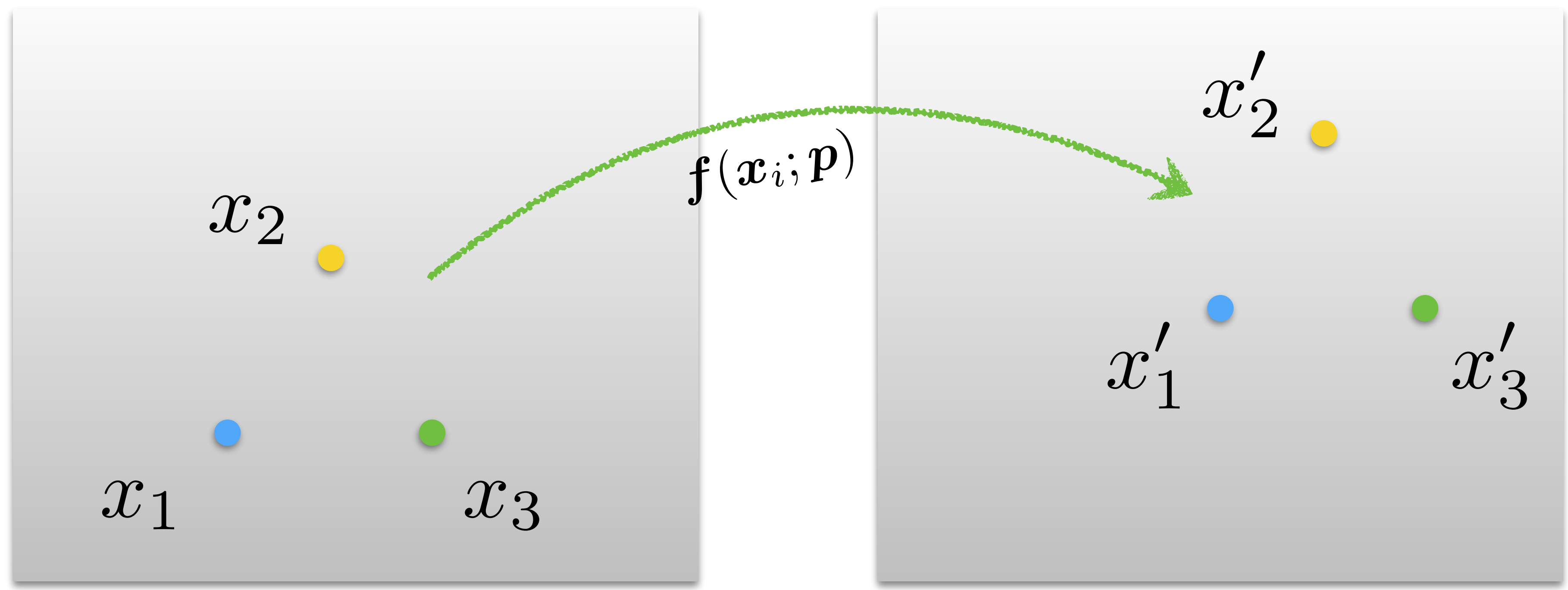
why not just squared?
(vector norm inside)



Least Squares Error

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

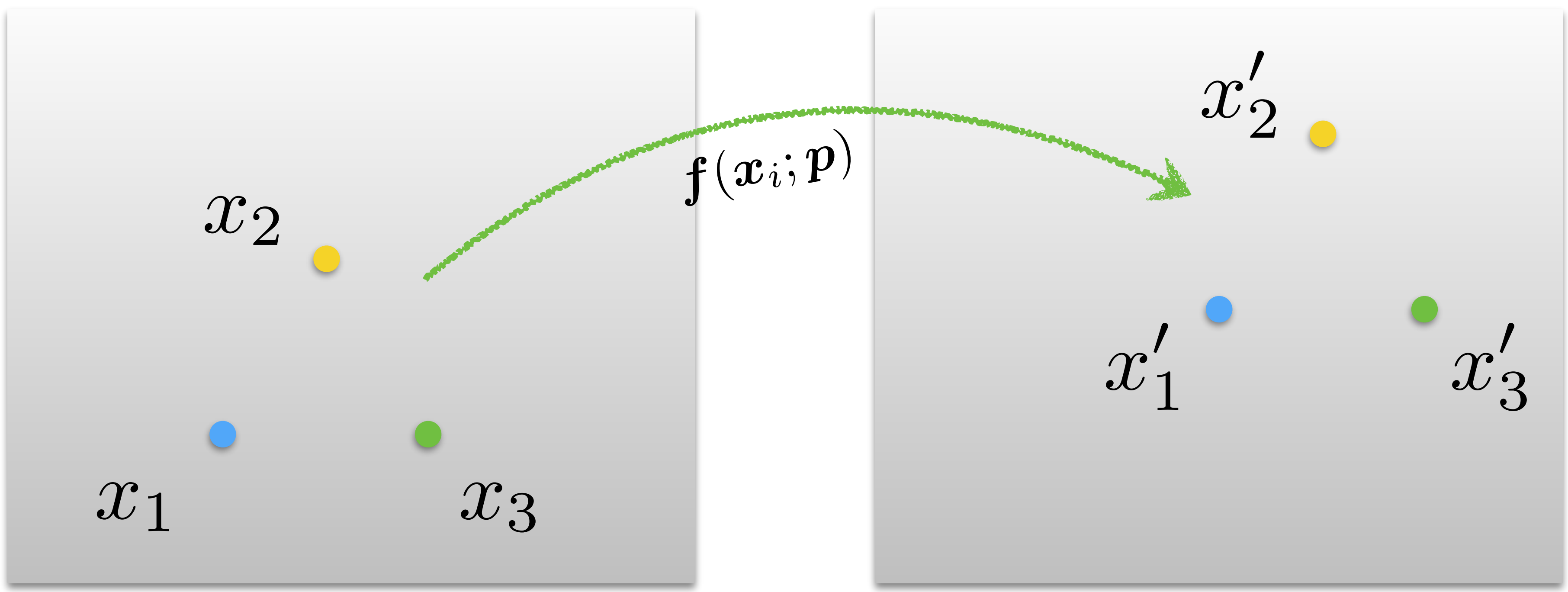
Residual
(projection error)



Least Squares Error

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

What is the free variable?
What do we want to optimize?



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!)

$$E_{\text{LLS}} = \sum_i |\boldsymbol{a}_i \boldsymbol{x} - \boldsymbol{b}_i|^2$$

$$= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \quad \text{(matrix form)}$$

This function is quadratic.

How do you find the root of a quadratic?

General form of linear least squares

(**Warning:** change of notation. x is a vector of parameters!)

$$E_{\mathrm{LLS}} = \sum_i |\boldsymbol{a}_i \boldsymbol{x} - \boldsymbol{b}_i|^2$$

$$= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$$ (matrix form)

Minimize the error:

Expand

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

Take derivative, set to zero

$$(\mathbf{A}^\top \mathbf{A}) \boldsymbol{x} = \mathbf{A}^\top \boldsymbol{b}$$ (normal equation)

Solve for x

$$\boldsymbol{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \boldsymbol{b}$$

For the Affine transformation $\quad \boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{p})$

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

Vectorize transformation parameters

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix}$$

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

$$\vdots \qquad\qquad \vdots$$

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

Notation in general form $\quad \boldsymbol{b} \qquad \mathbf{A} \qquad \boldsymbol{x}$

Linear

least squares

estimation

only works

when the

transform function

is

?

Linear

least squares

estimation

only works

when the

transform function

is

linear!

Also

doesn't

deal well

with

outliers