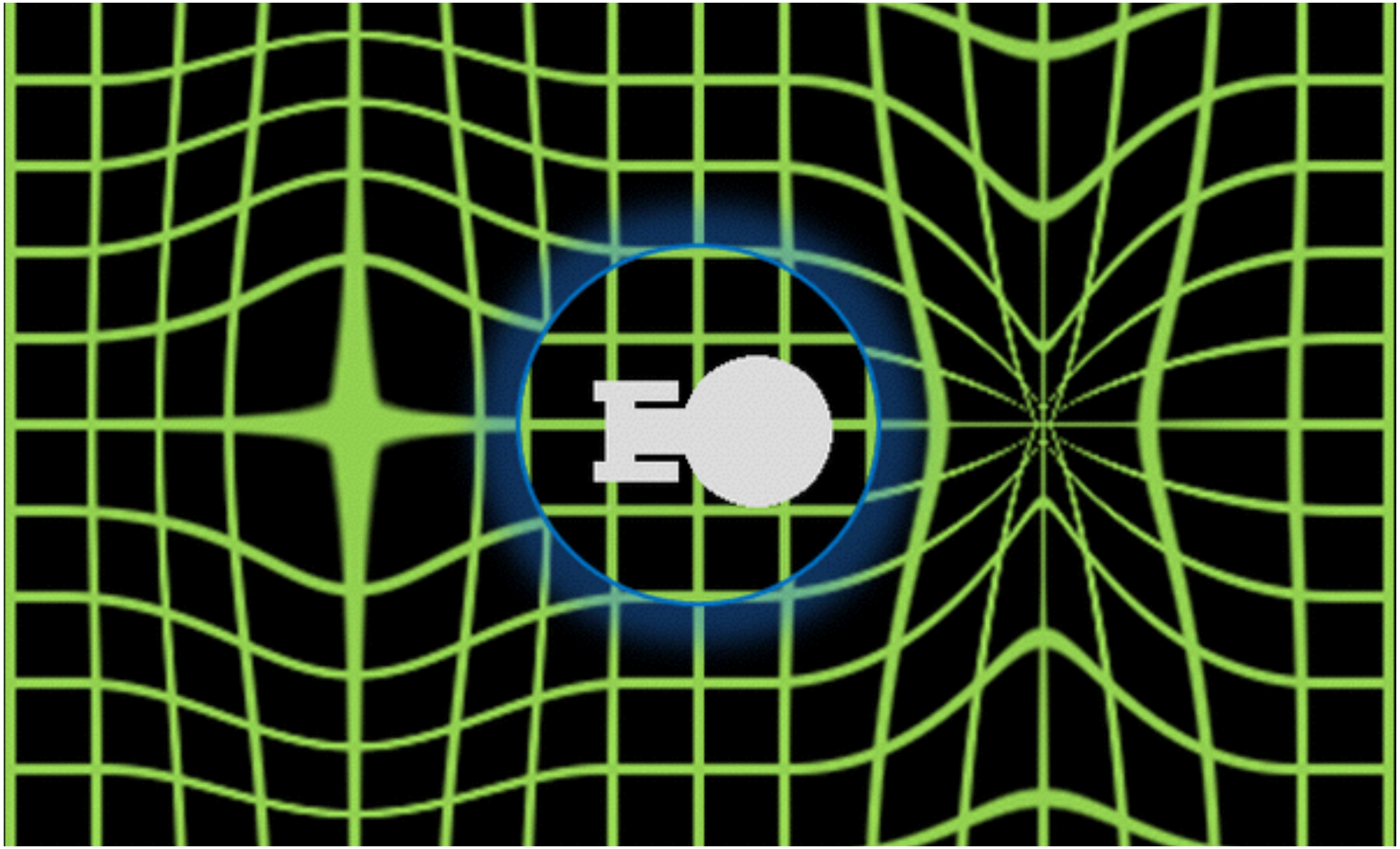


2D Image Transforms

16-385 Computer Vision

Carnegie Mellon University (Kris Kitani)


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

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

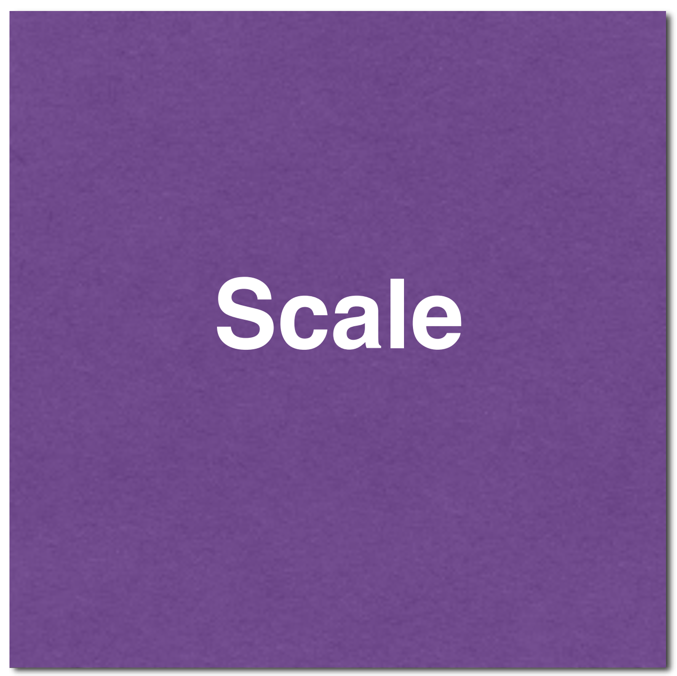
affine

perspective

cylindrical

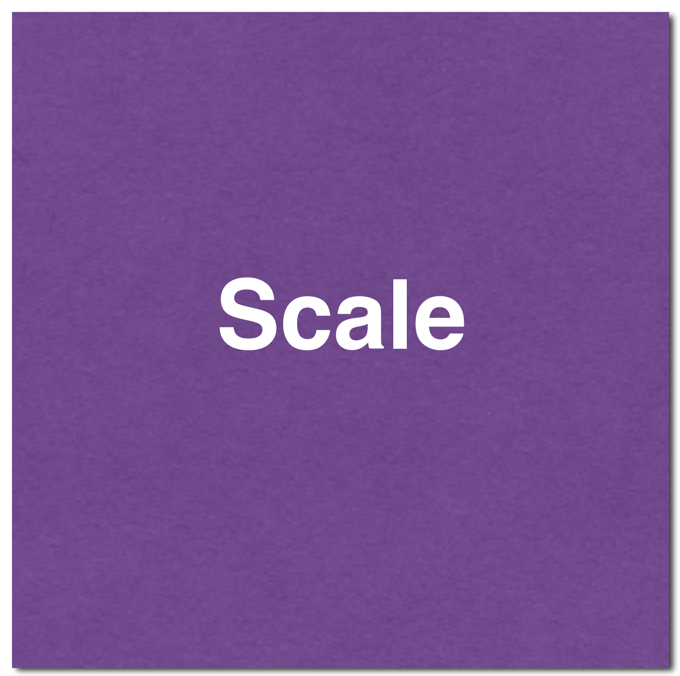
2D Planar Transformations

2D Planar Transformations



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

2D Planar Transformations



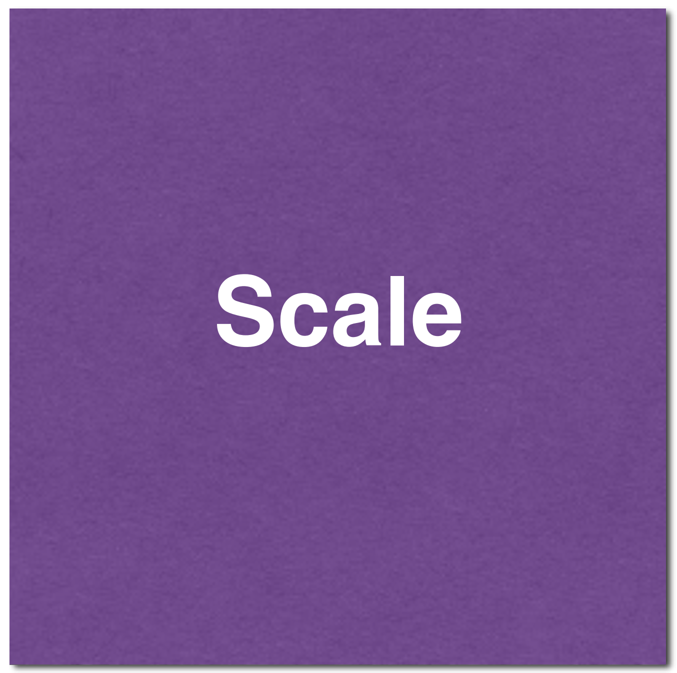
Scale

$$x' = ax$$

$$y' = by$$

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

2D Planar Transformations

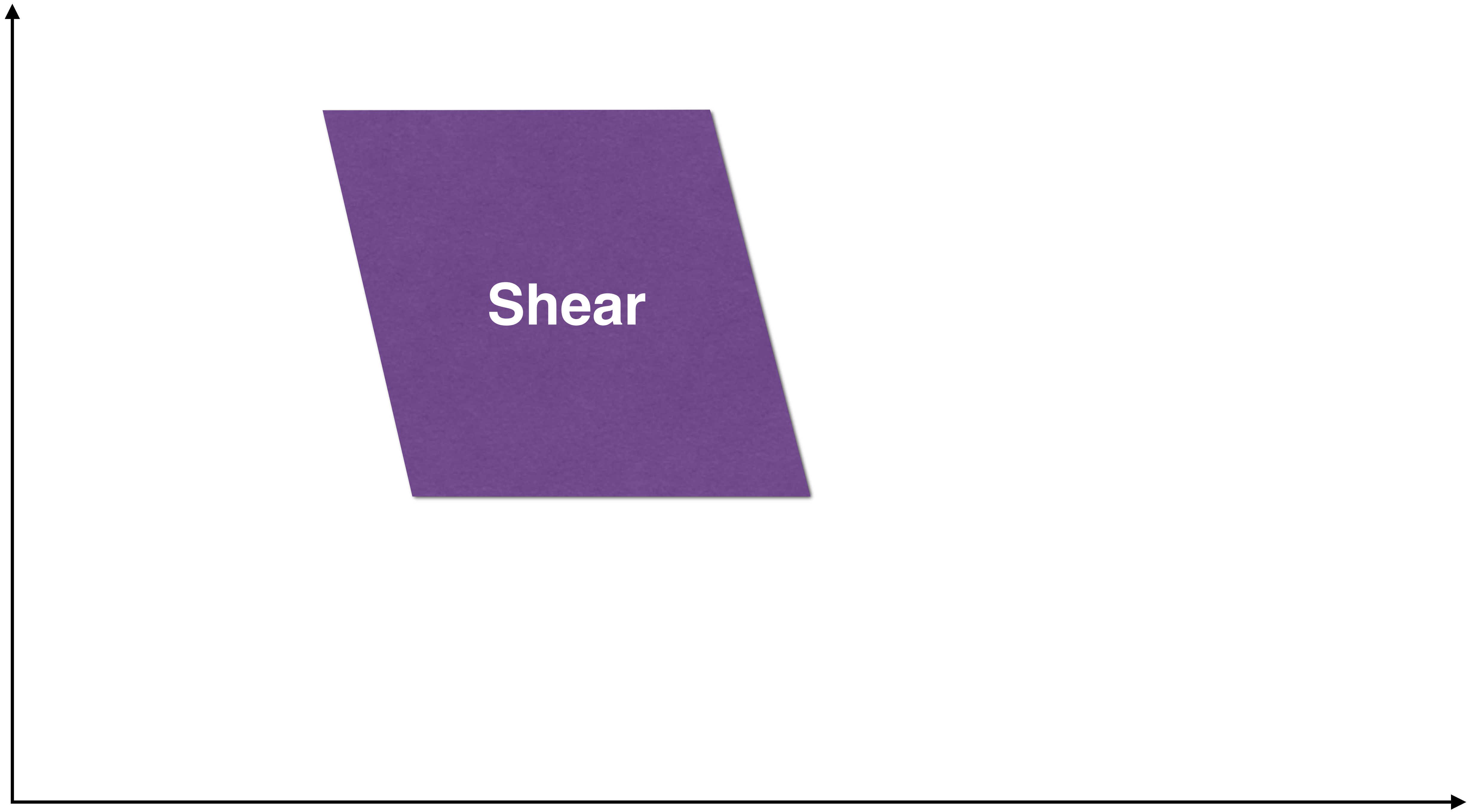


Scale

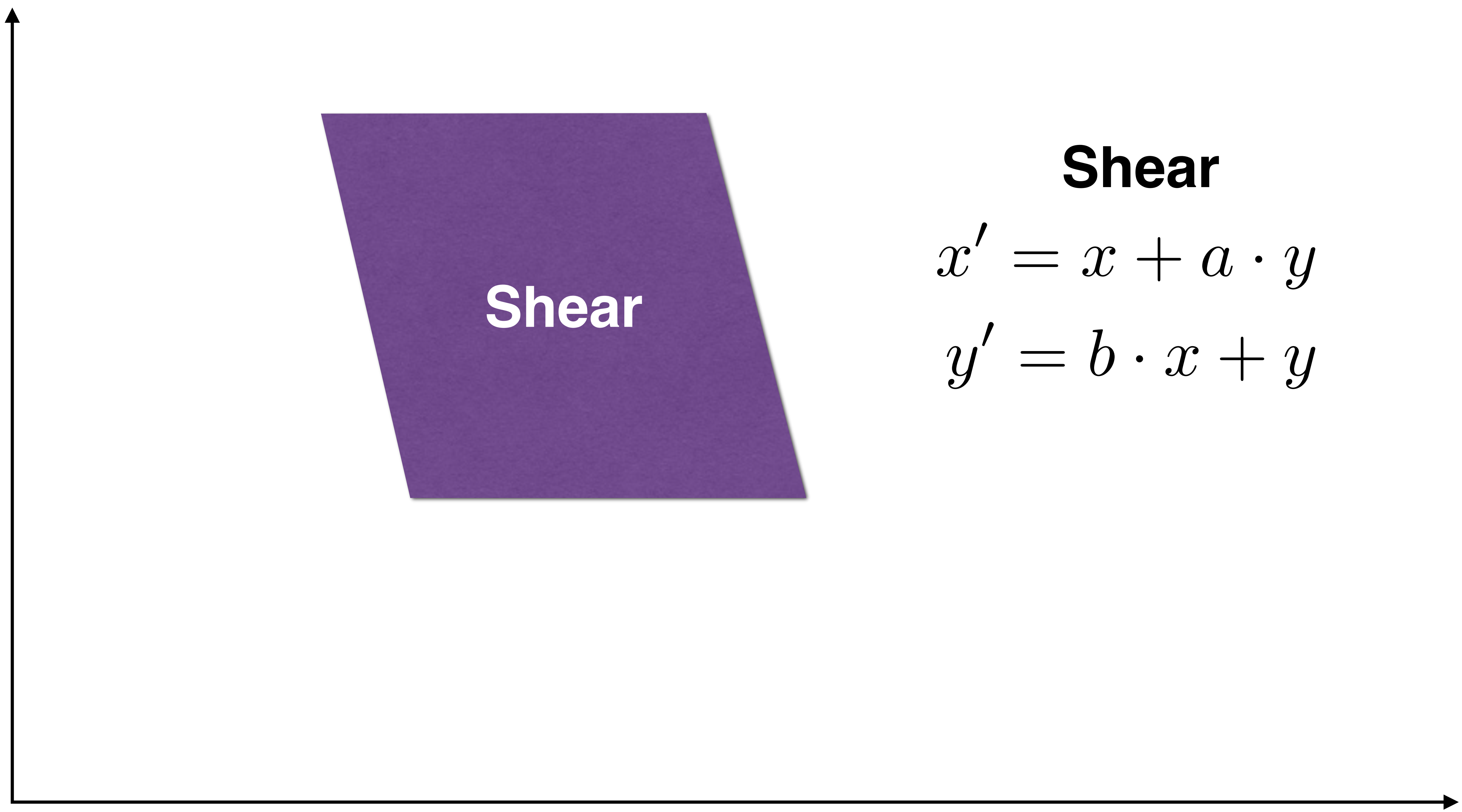
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix } S} \begin{bmatrix} x \\ y \end{bmatrix}$$

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

2D Planar Transformations



2D Planar Transformations

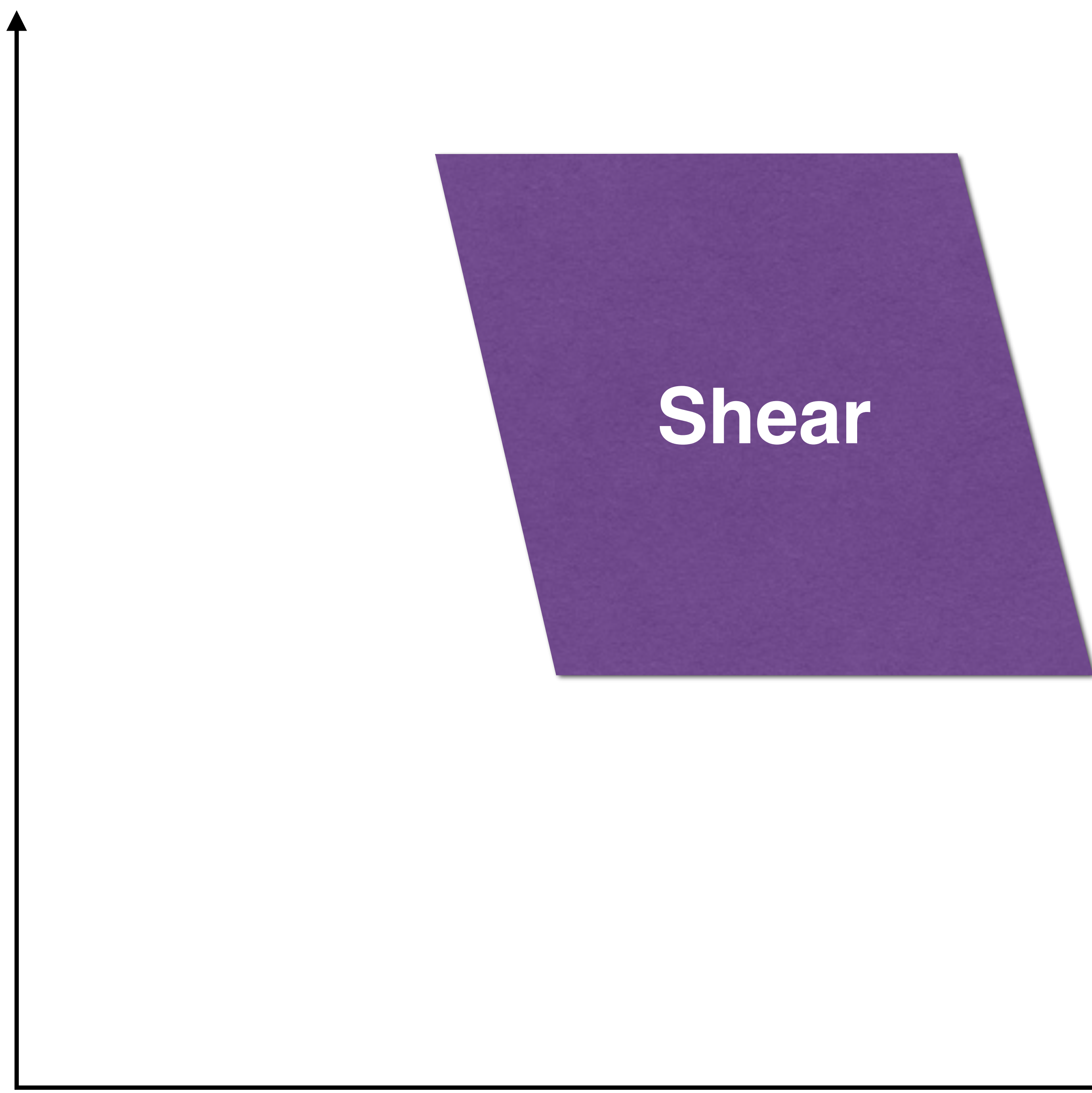


Shear

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

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

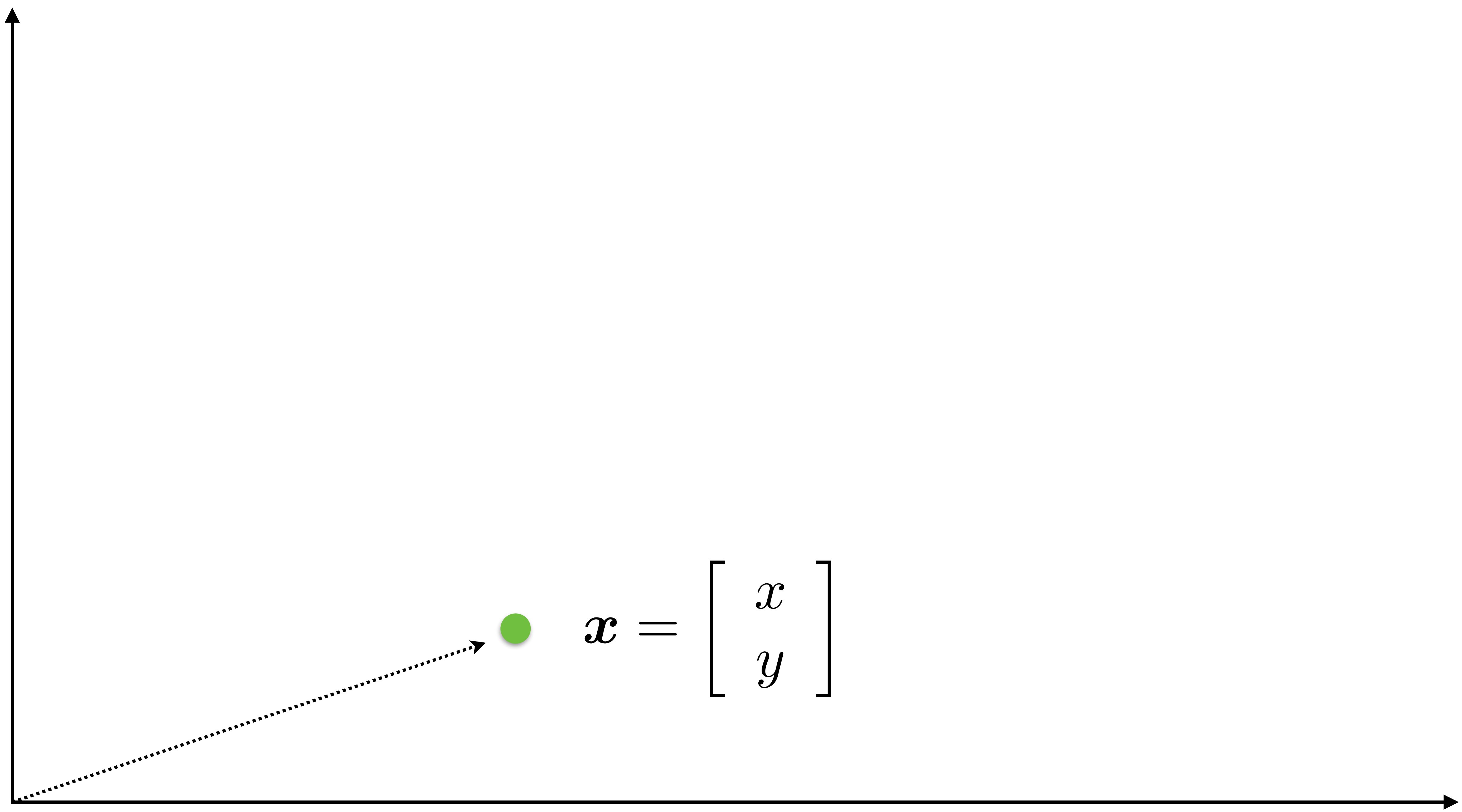
2D Planar Transformations



Shear

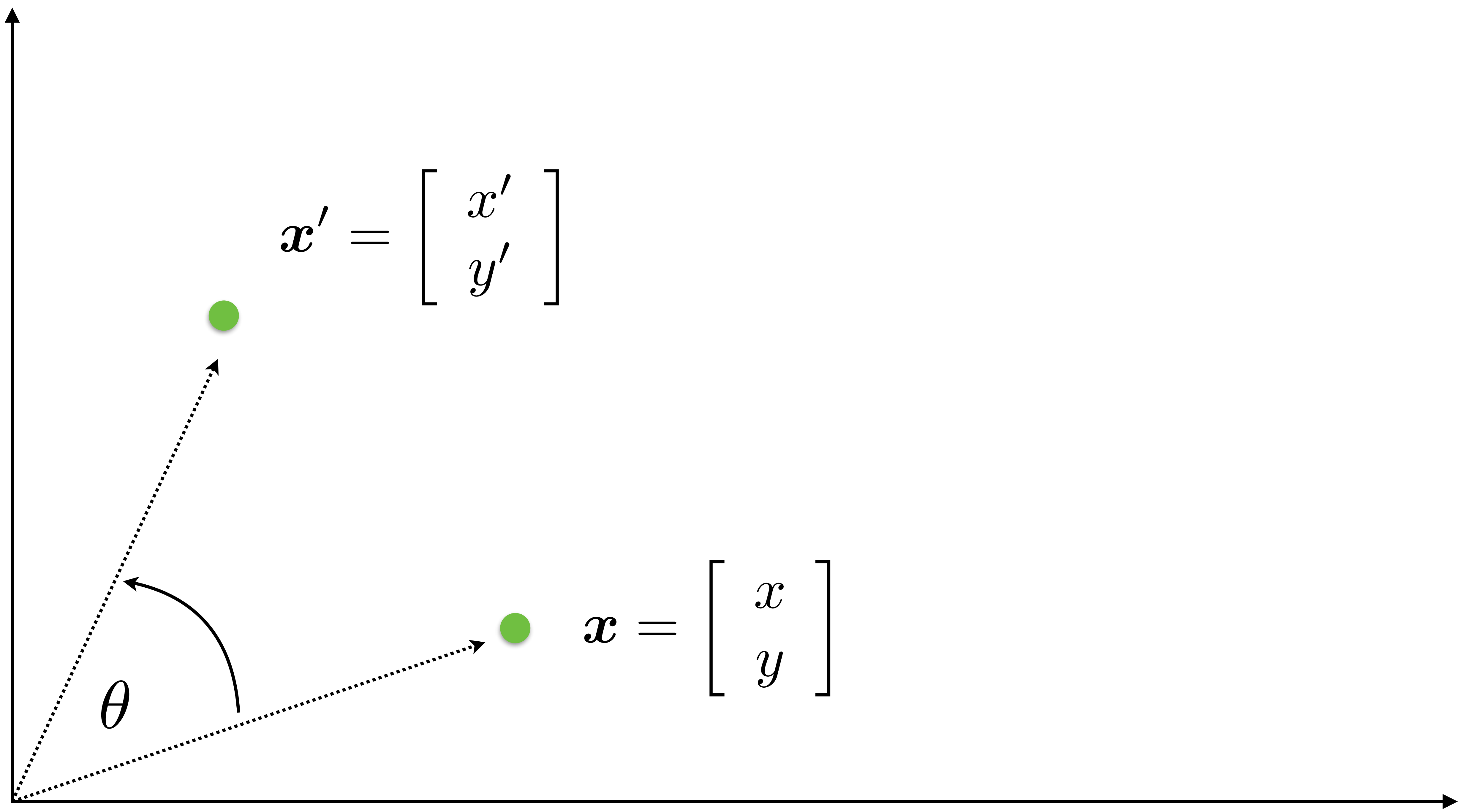
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Planar Transformations

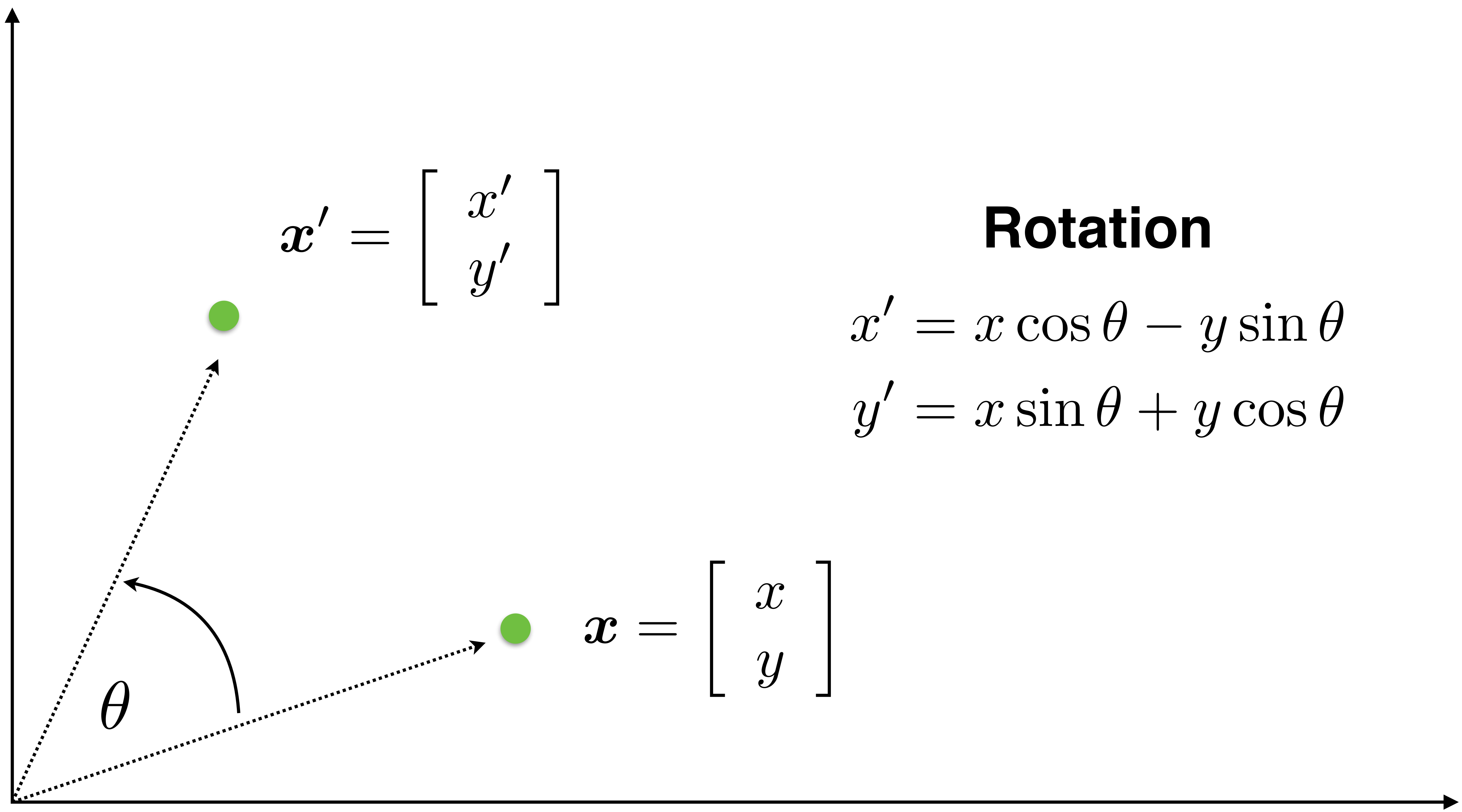


$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Planar Transformations



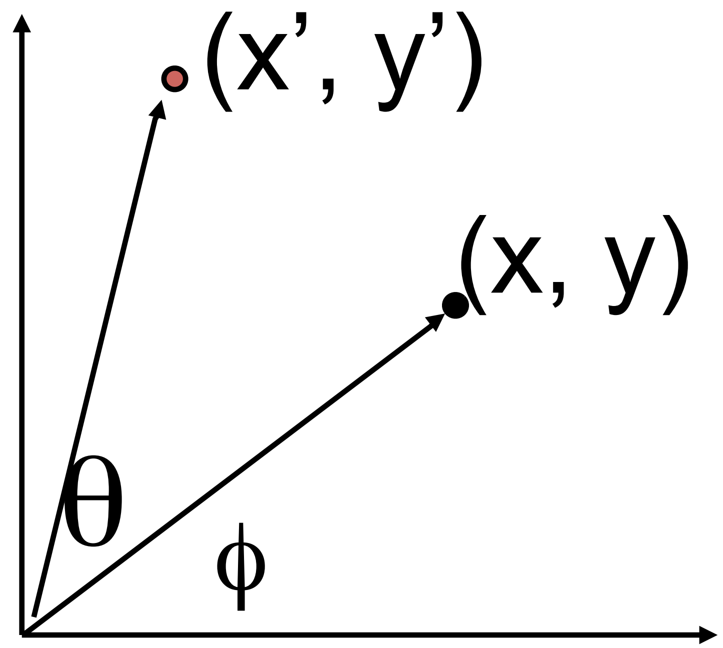
2D Planar Transformations



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…

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

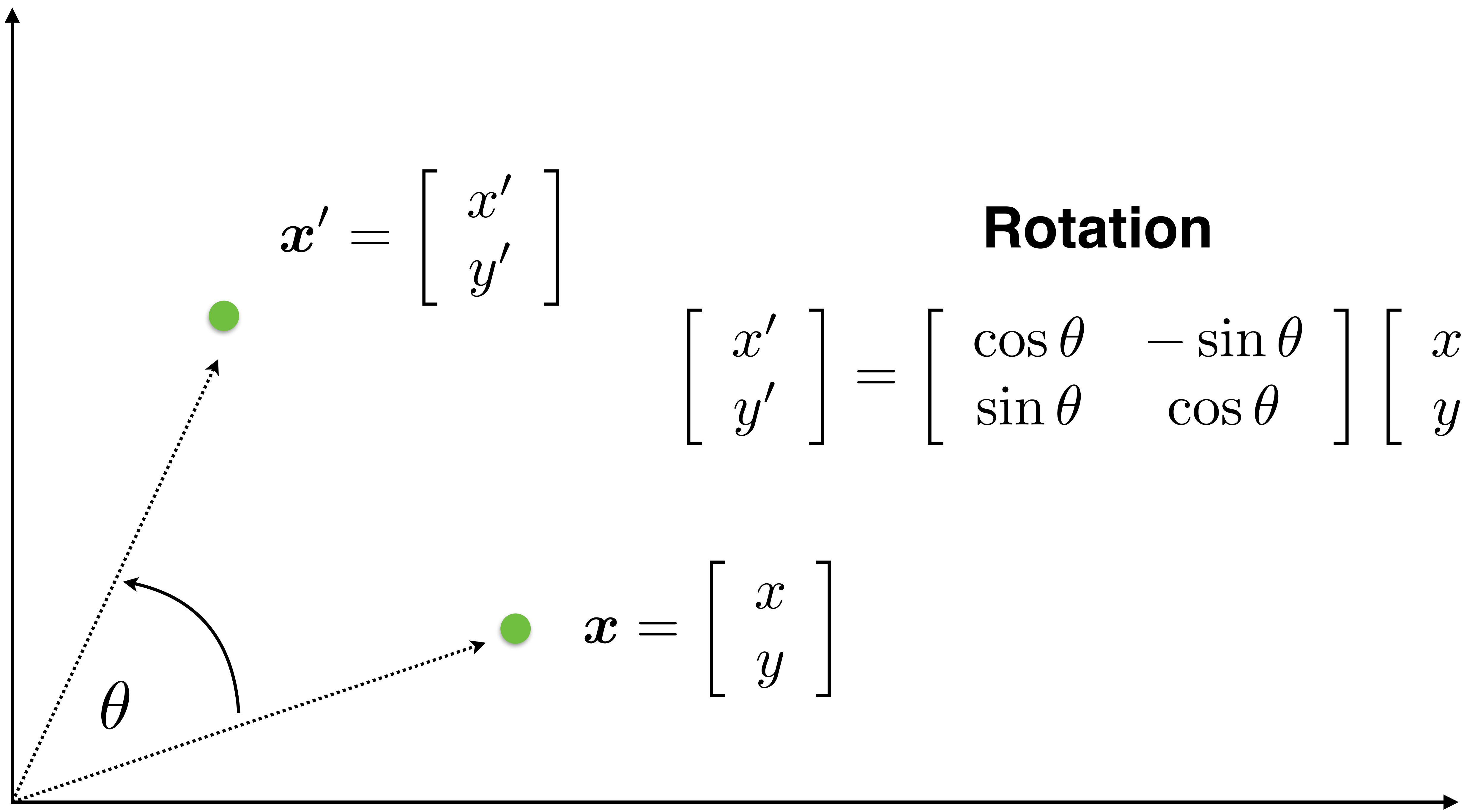
$y' = r \sin(\phi) \cos(\theta) + r \cos(\phi) \sin(\theta)$

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

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

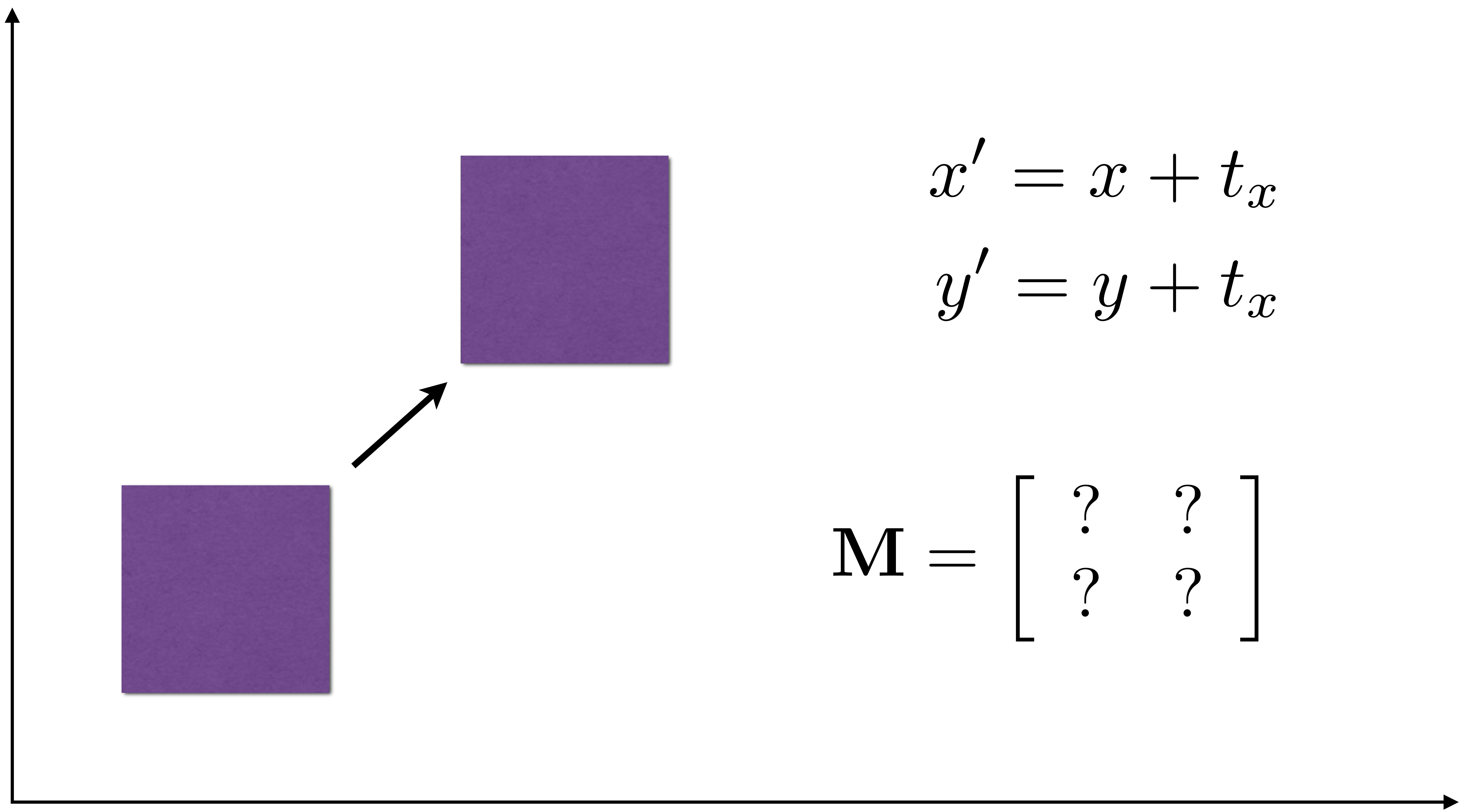
Shear

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

Identity

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

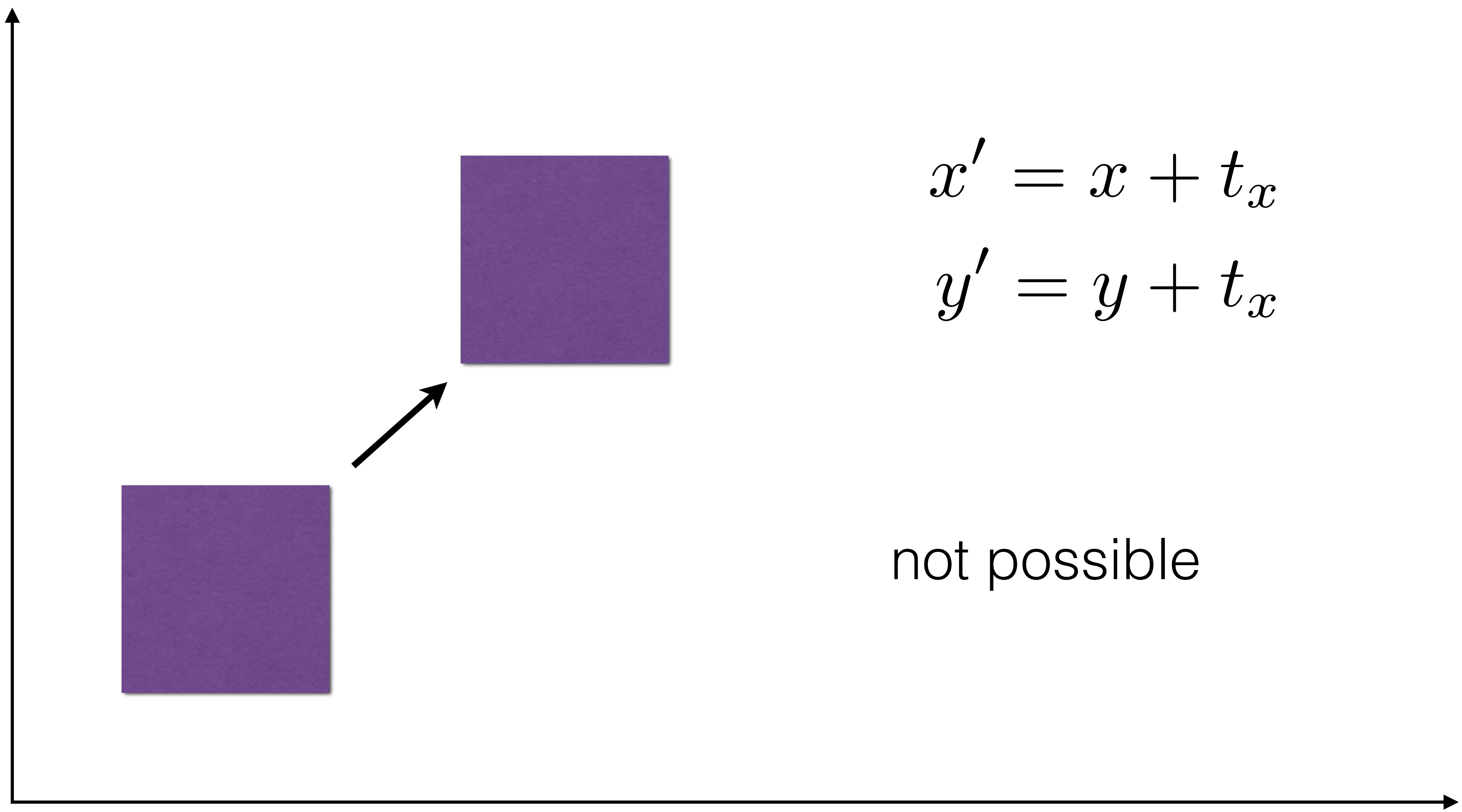
How do you represent translation with a 2 x 2 matrix?



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

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

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?

$$x' = x + t_x$$

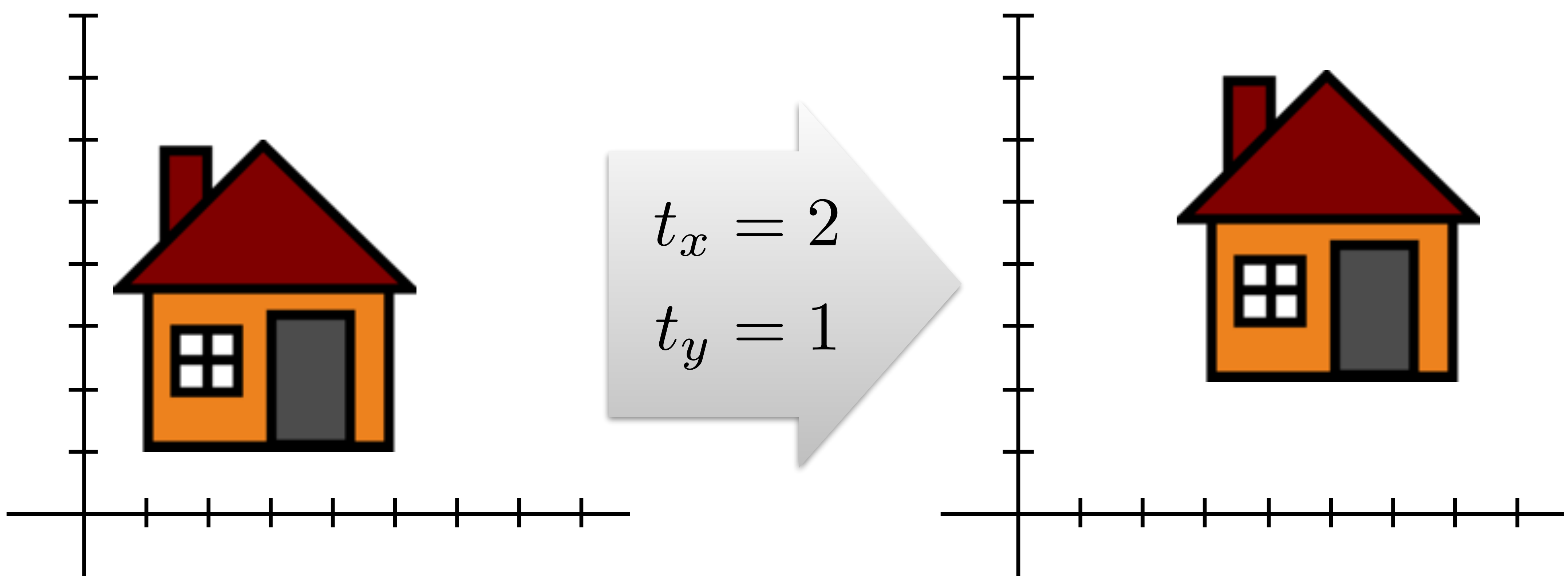
$$y' = y + t_y$$

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

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



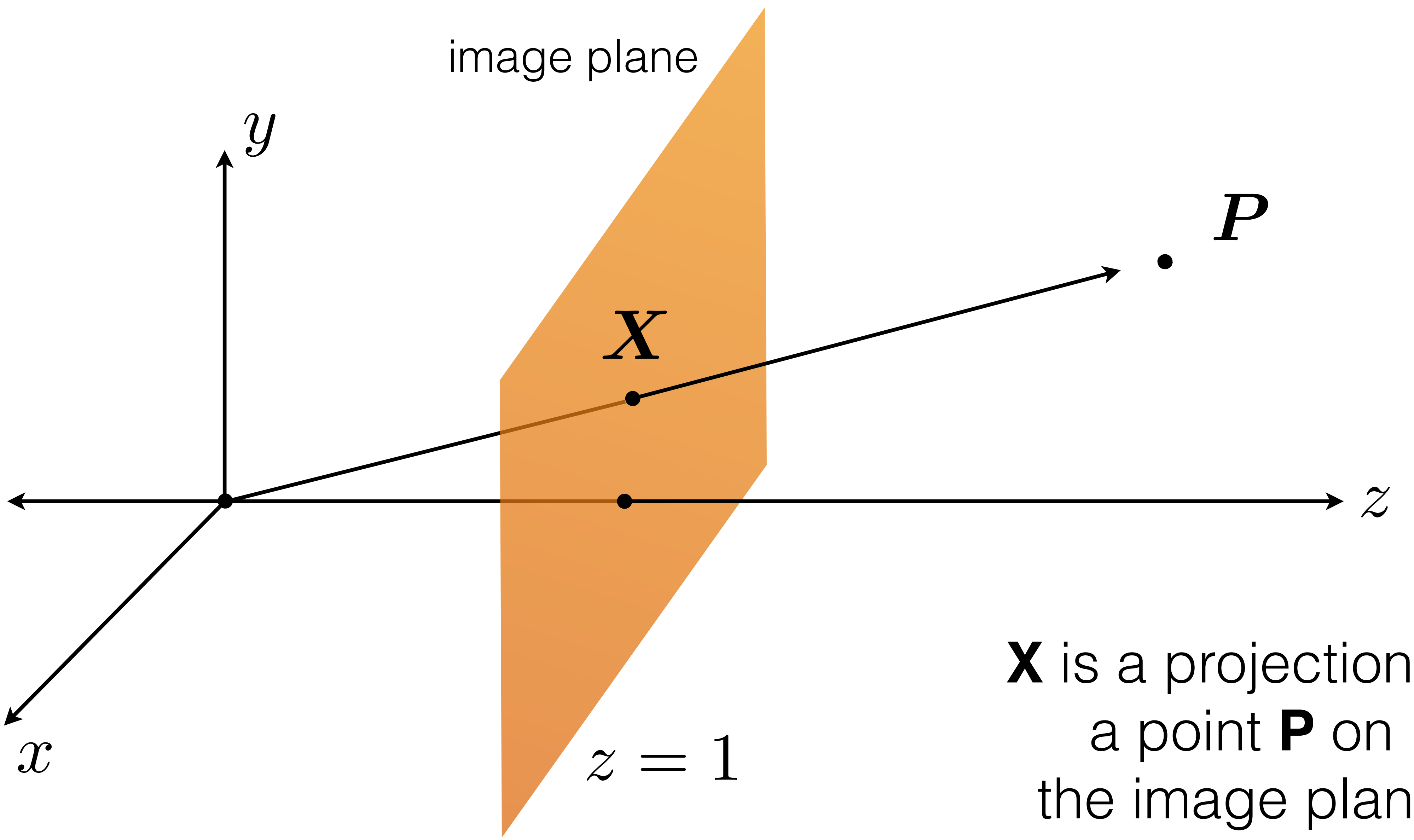
A 2D point in an image can be represented as a 3D vector

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

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

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

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

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

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

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

Does the order of multiplication matter?

2D transformations

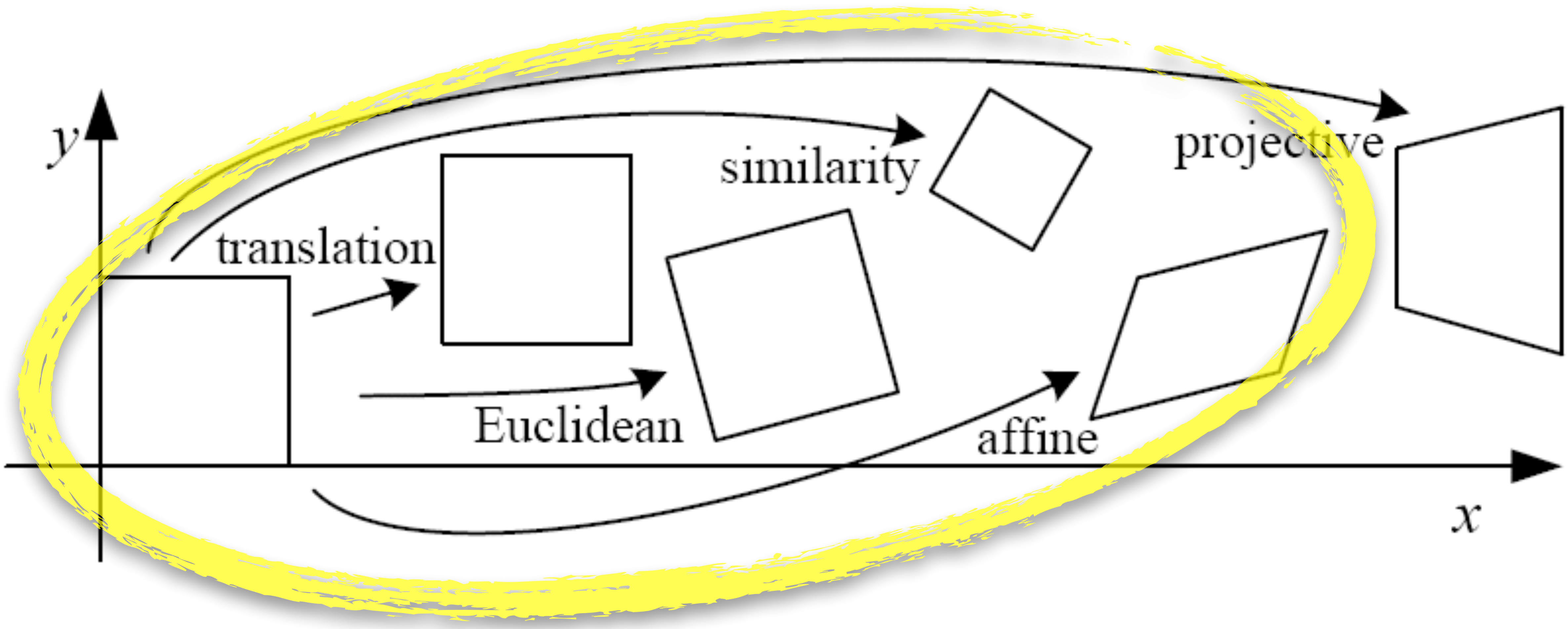


Figure 1: Basic set of 2D planar transformations

Name	Matrix	# D.O.F.
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times3}$	2
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times3}$	3
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times3}$	4
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times3}$	6
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]_{3\times3}$	8

Affine Transformation

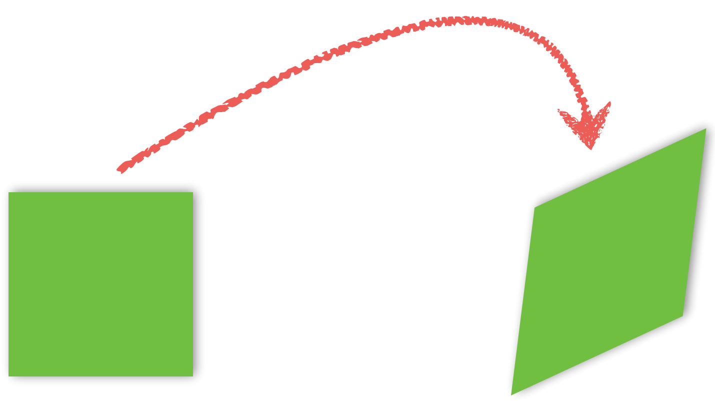
Affine transformations are combinations of

- Linear transformations, and
- Translations

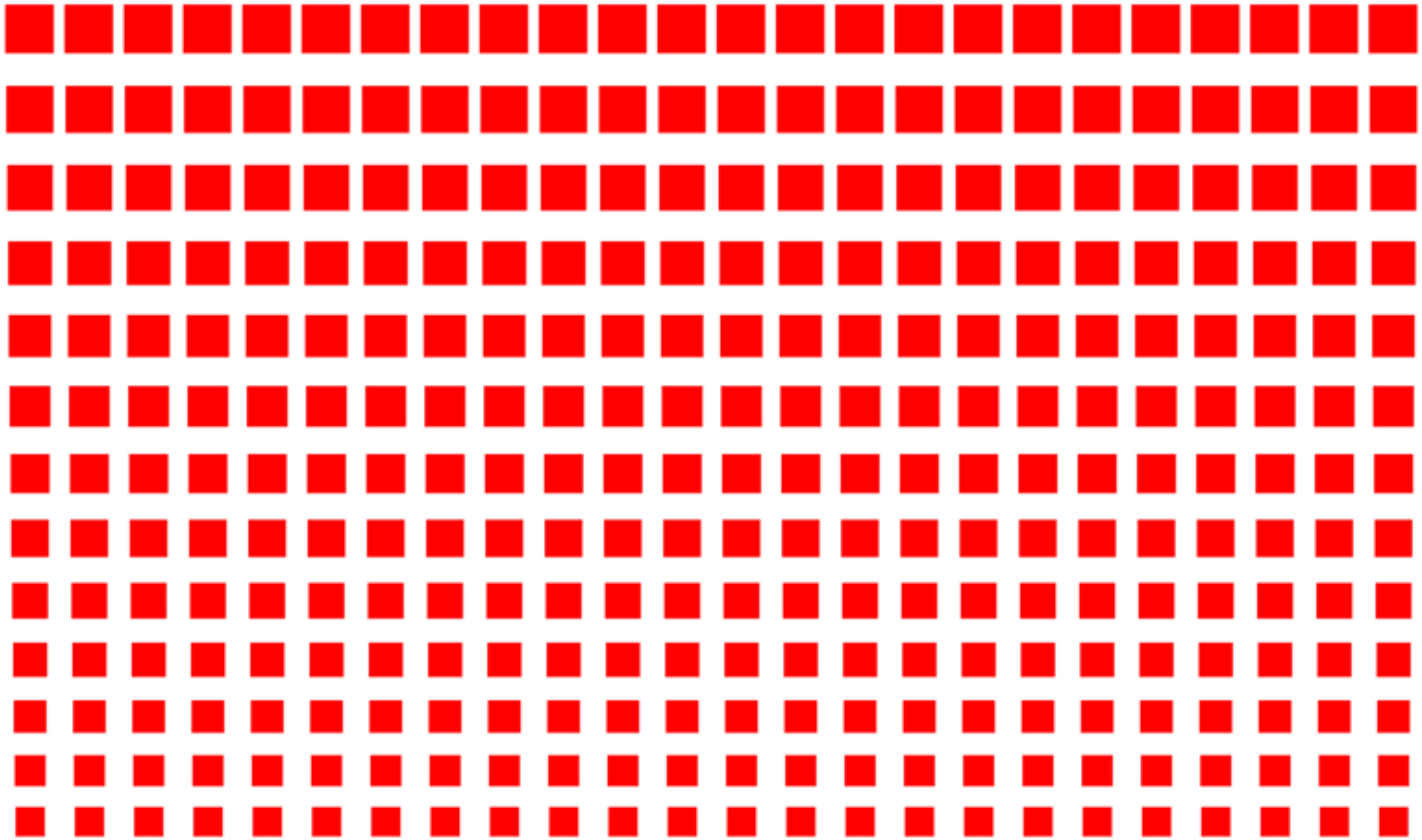
$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

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?



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

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

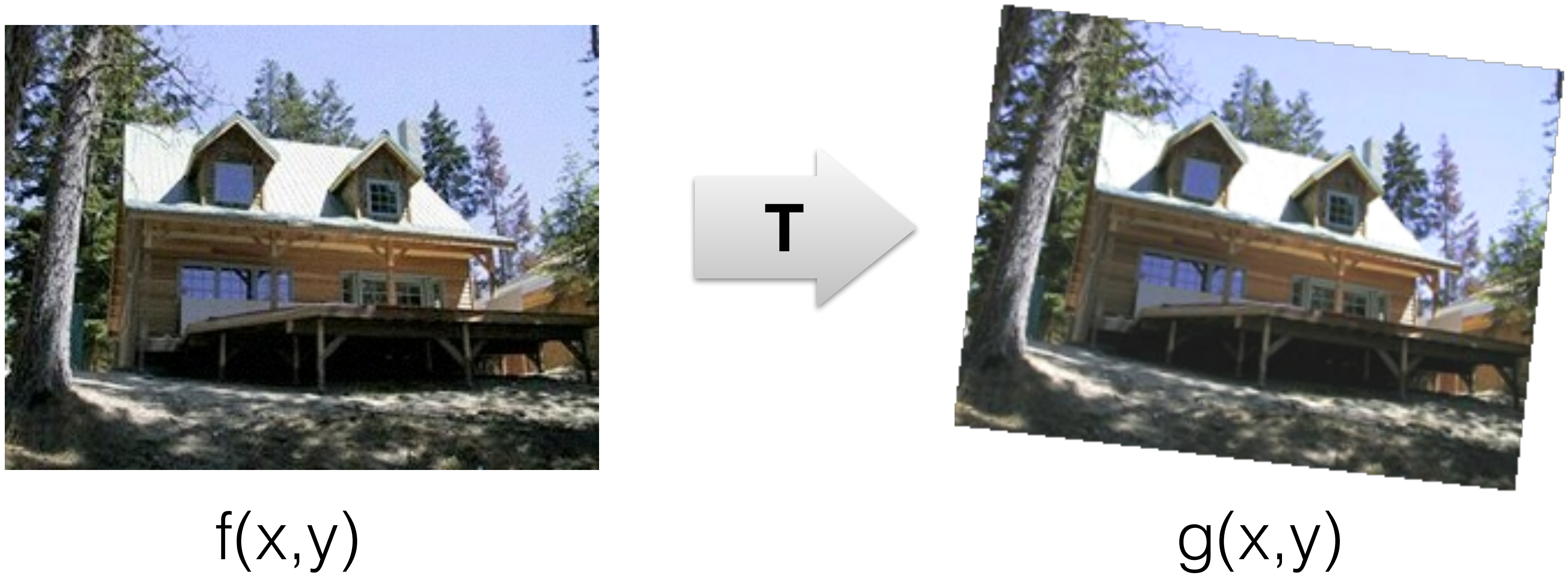
transformation function — parameters

Find the best estimate of

$$\boldsymbol{p}$$

Model fitting

Recover the transformation

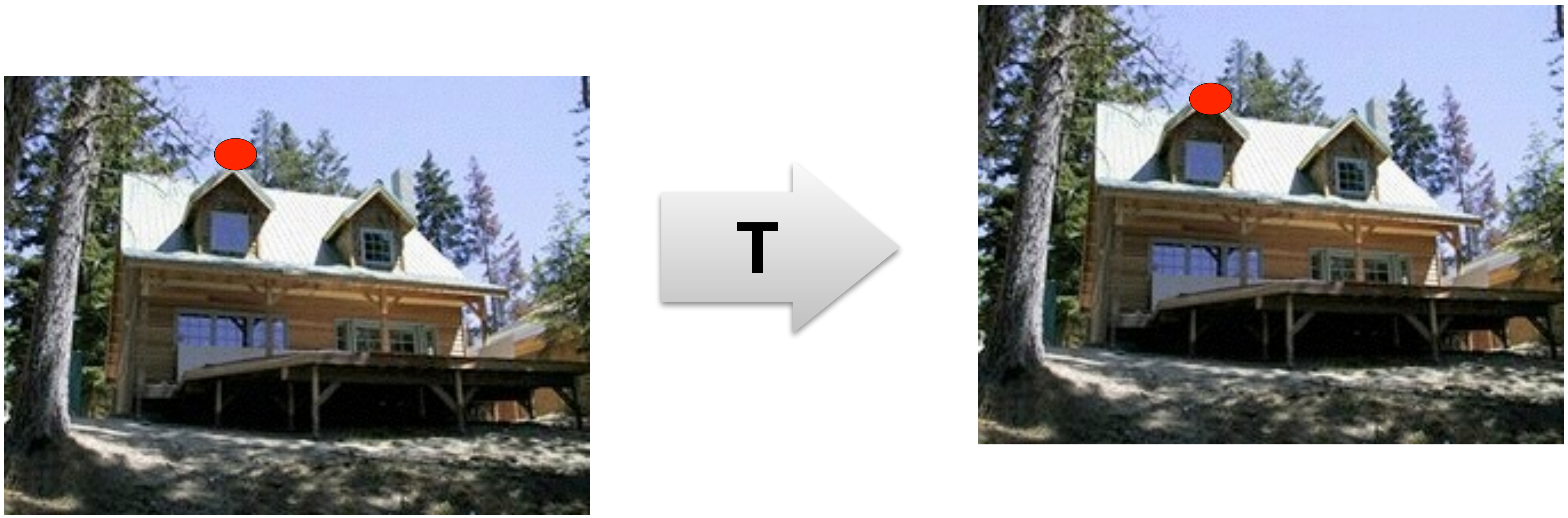


Given f and g, how would you recover the transform T?

(user will provide correspondences)

How many do we need?

Translation

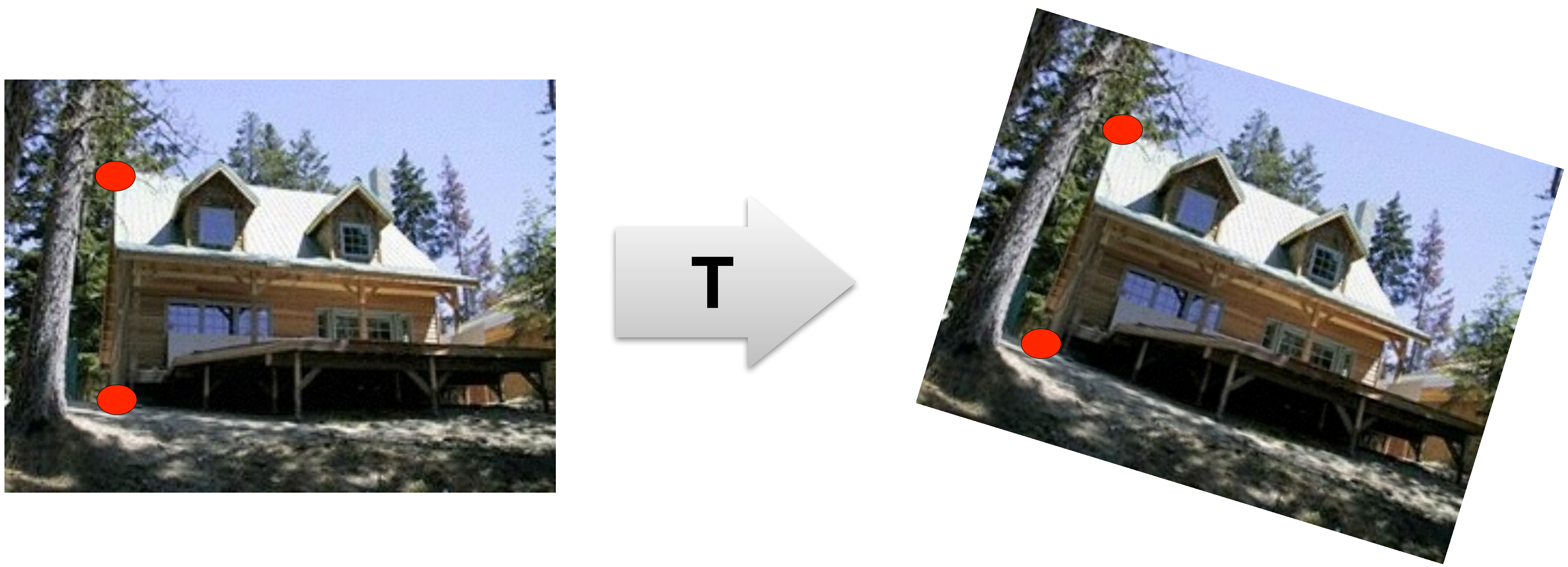


f(x,y) g(x,y)

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

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

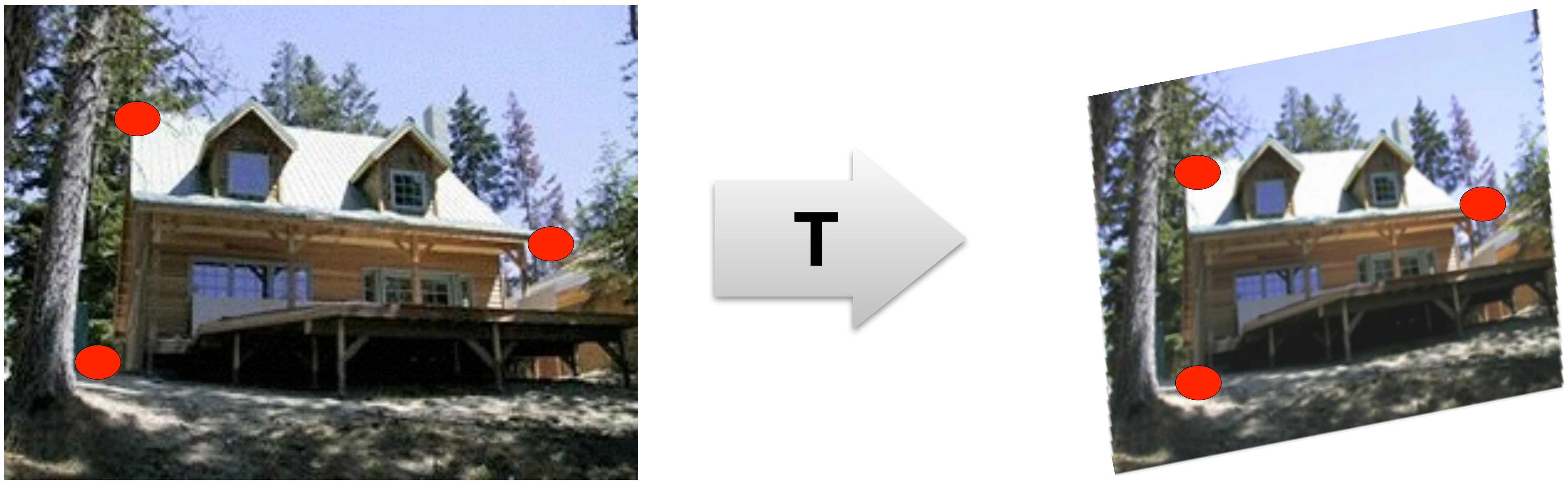
Euclidean



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

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

Affine

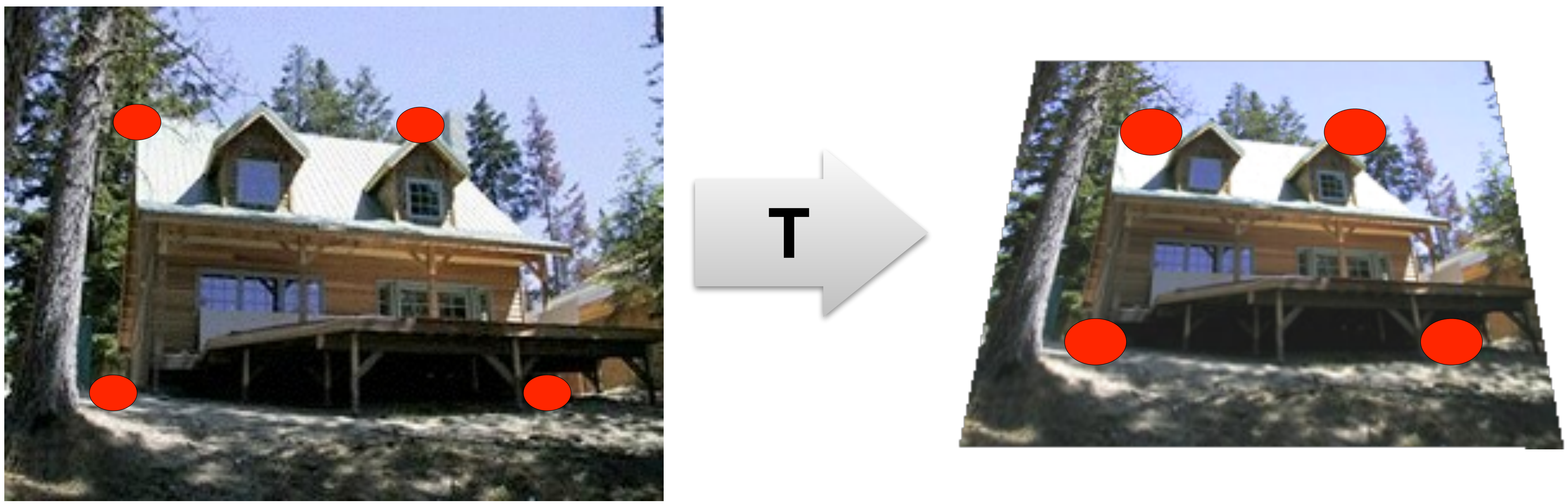


f(x,y)

g(x,y)

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

Projective



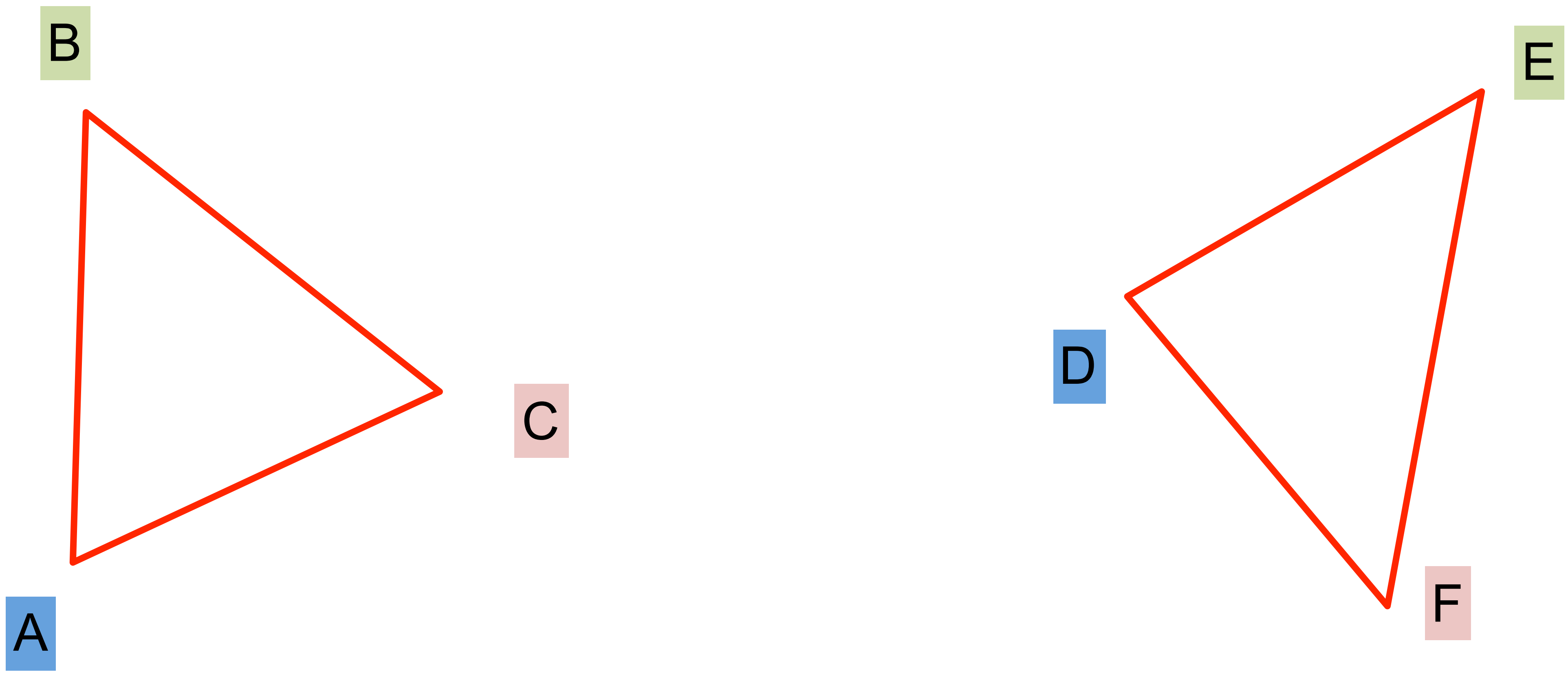
f(x,y) g(x,y)

- *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

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

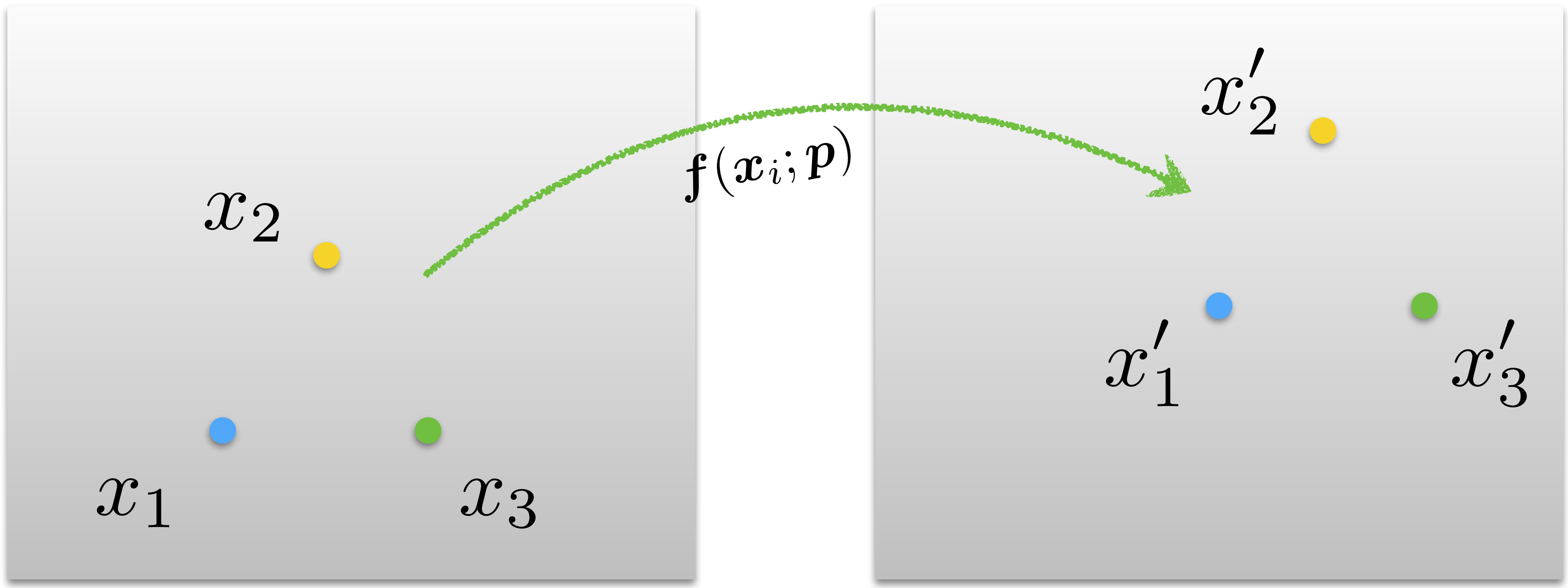
transformation function

parameters

Find the best estimate of

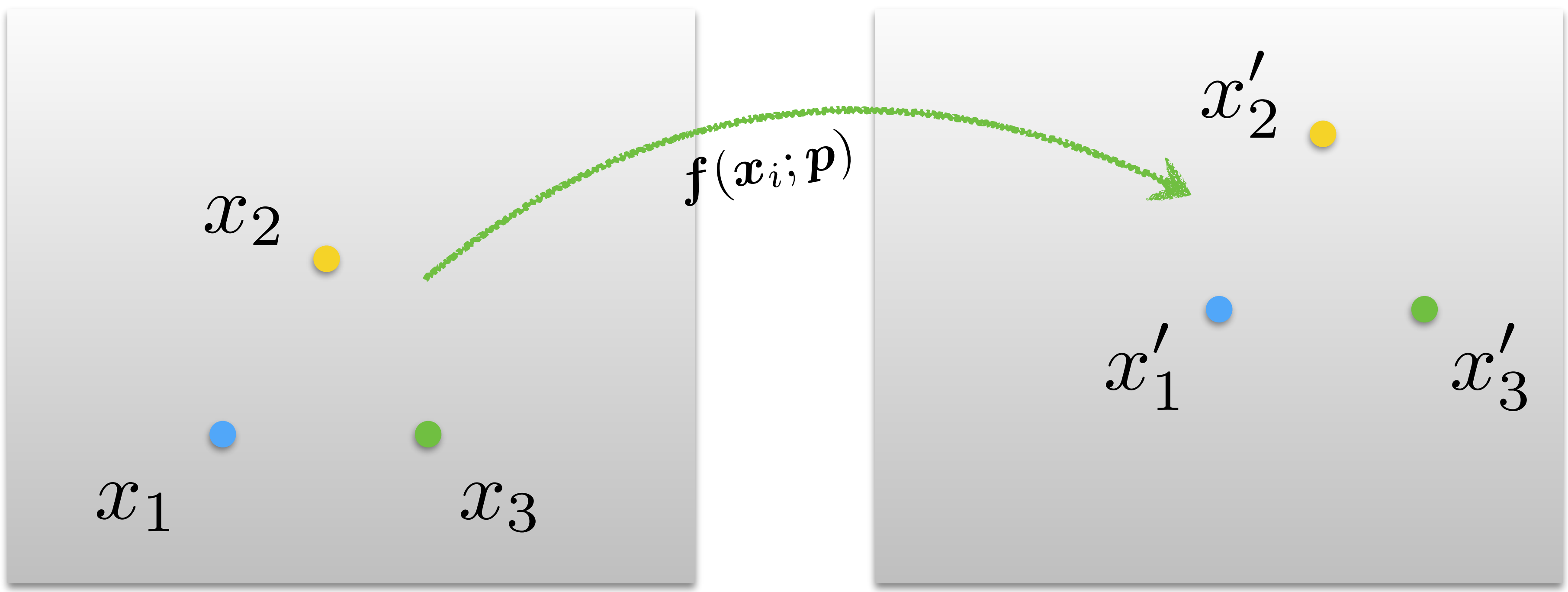
$$\boldsymbol{p}$$

Given point correspondences



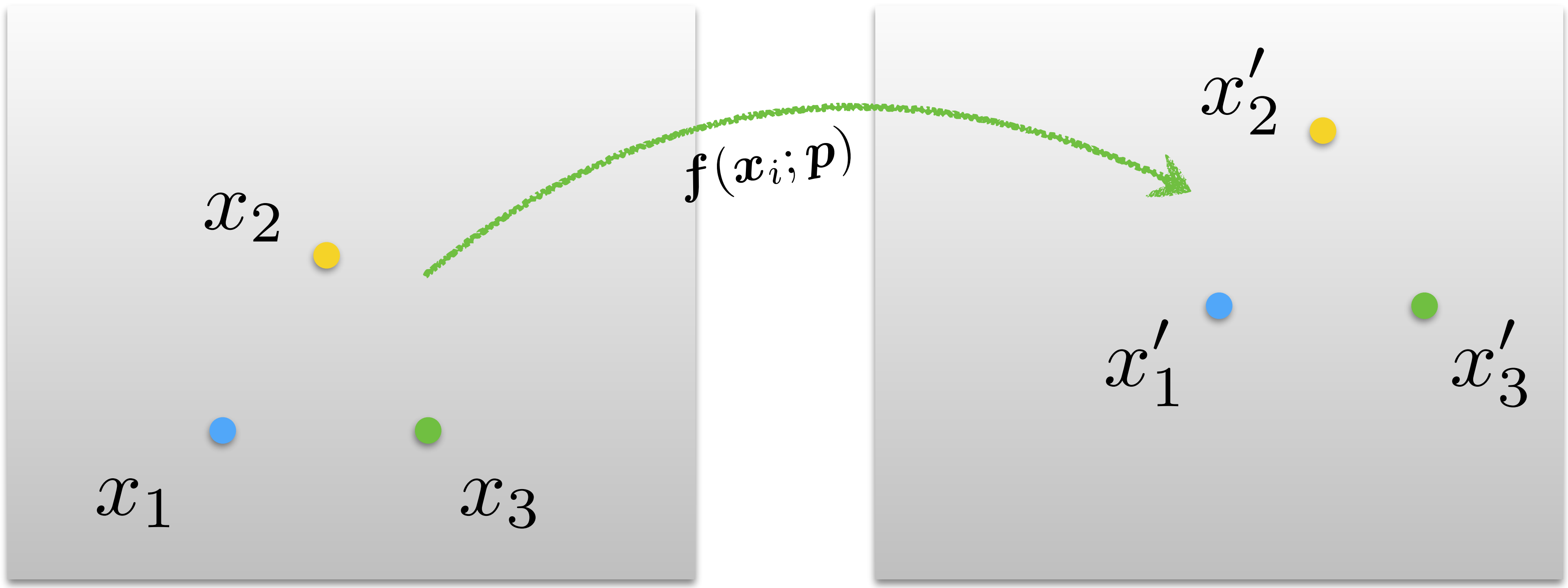
Given point correspondences

How can you solve for the transformation?



Least Squares Error

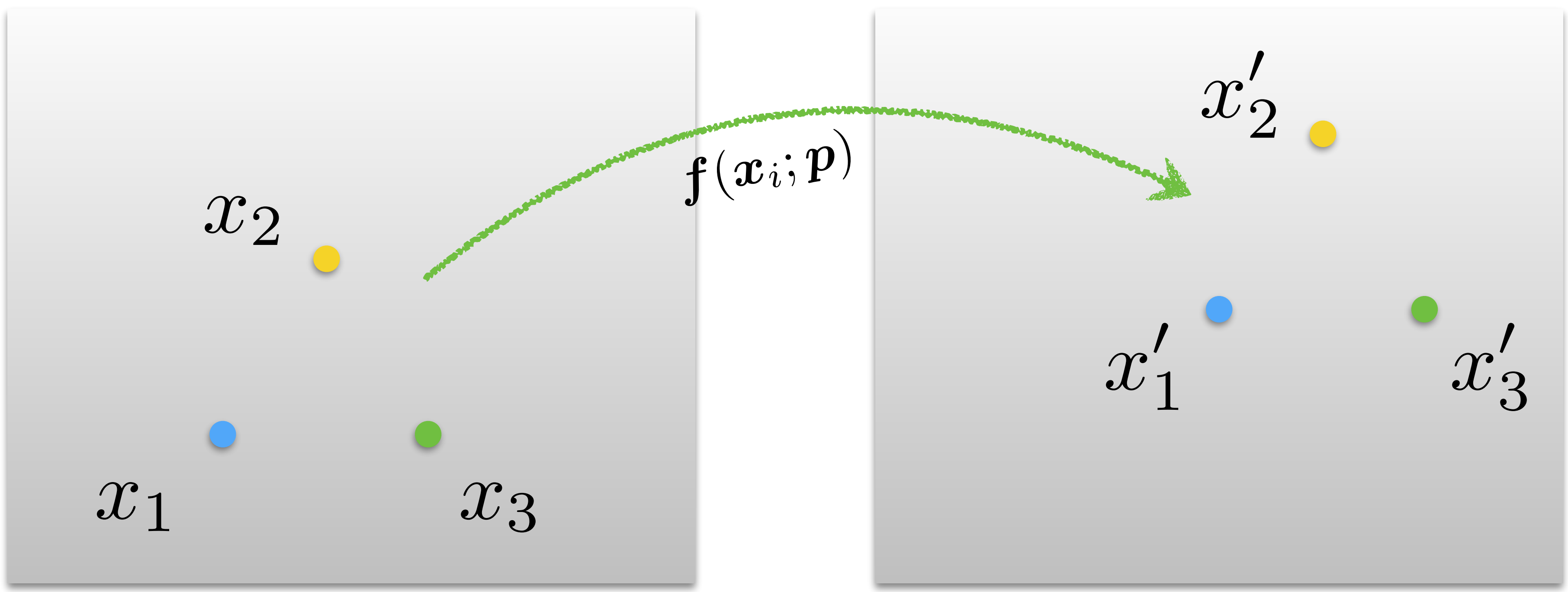
$$E_{\mathrm{LS}} = \sum_i \| \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i \|^2$$



Least Squares Error

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

What is this? *What is this?*



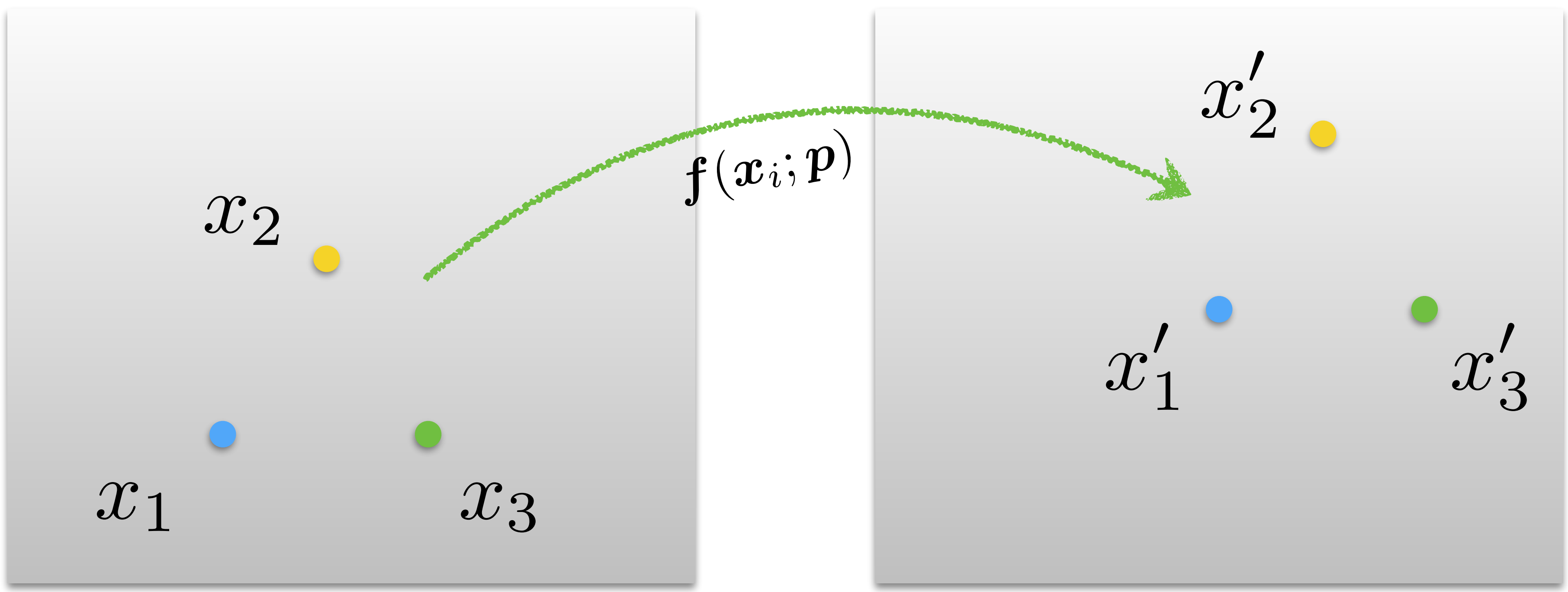
Least Squares Error

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

What is this?

What is this?

What is this?



Least Squares Error

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

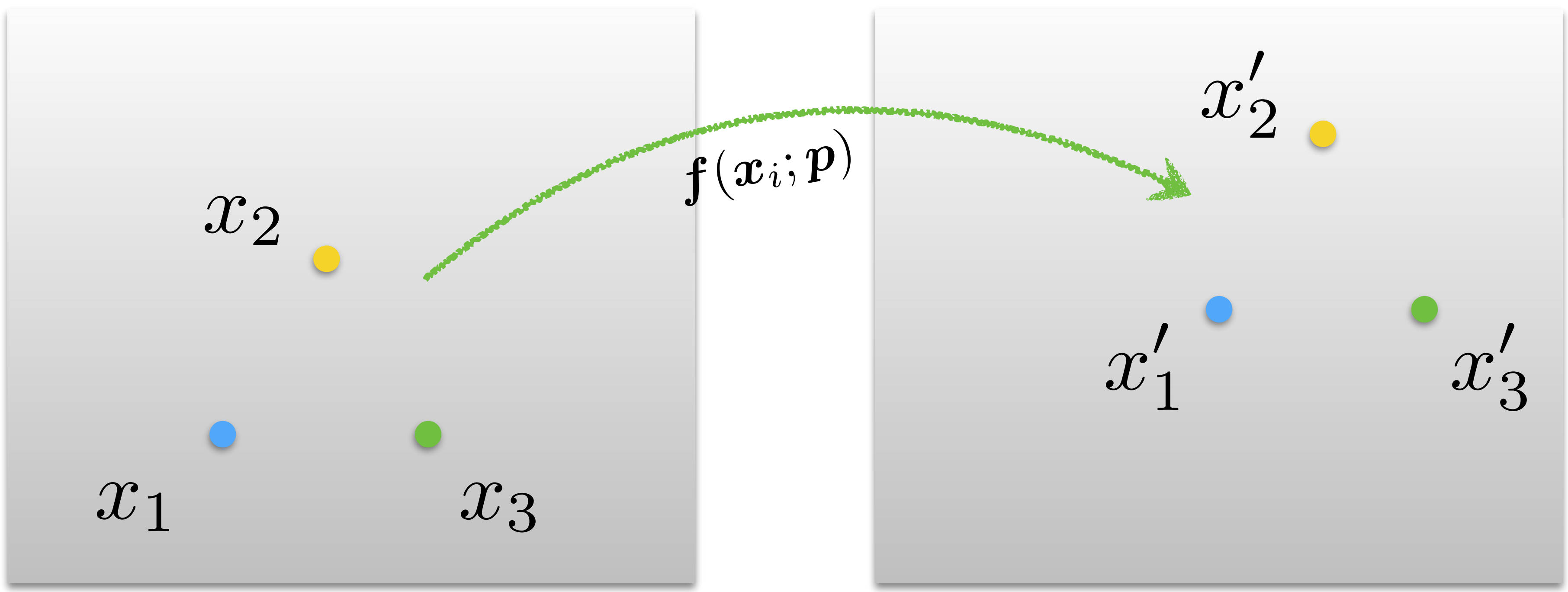
Euclidean (L2) norm

squared!

why not just squared?
(vector norm inside)

predicted location

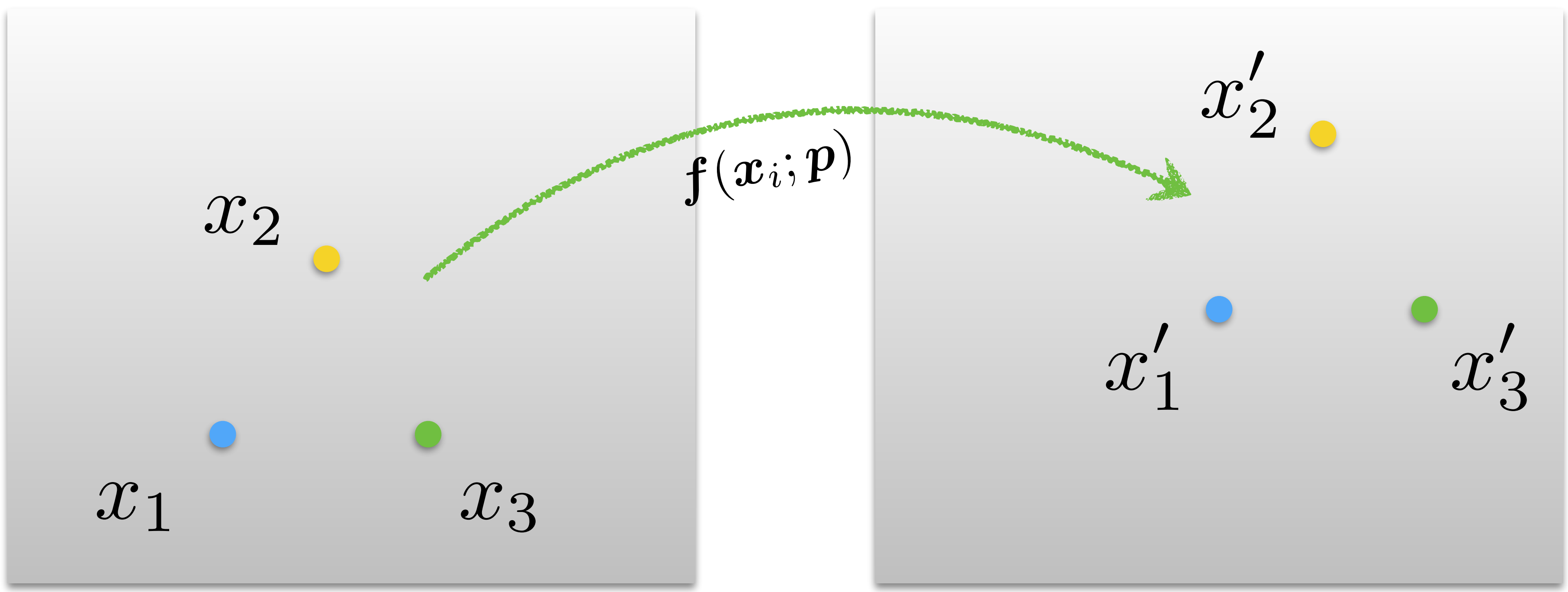
measured location



Least Squares Error

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

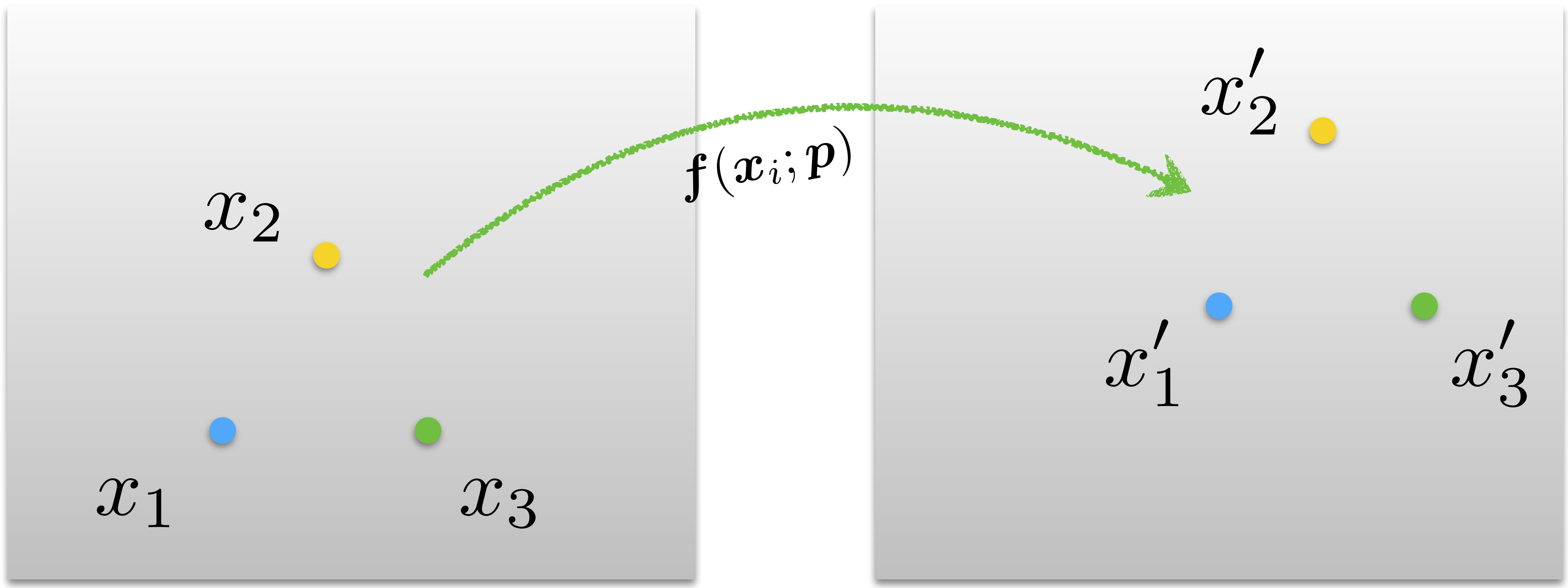
Residual
(projection error)



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!)

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

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

Minimize the error:

Expand

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

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

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

Vectorize transformation parameters

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

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

Coming soon…

Projective Transform

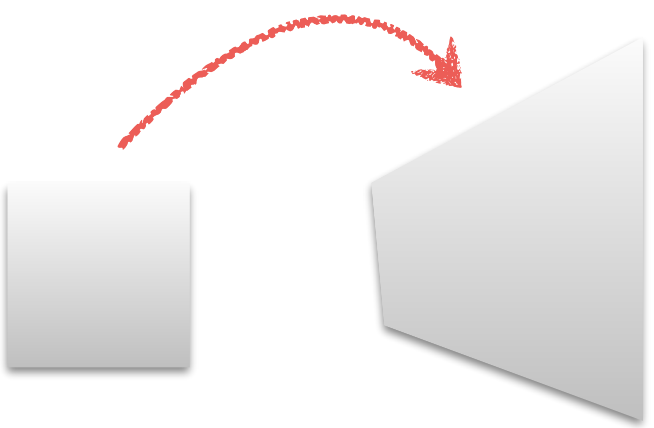
Projective transformations are combos of

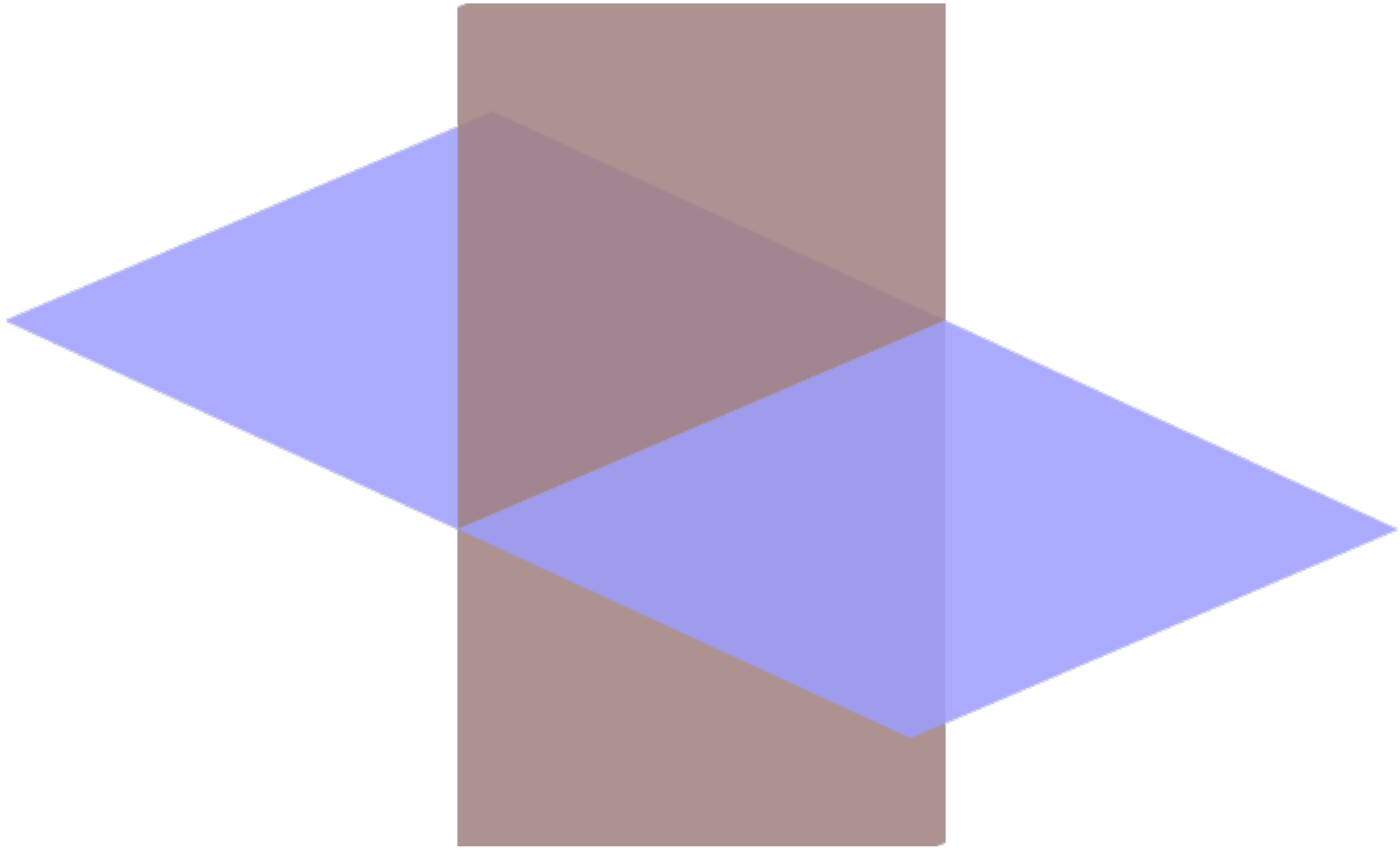
- Affine transformations, and
- Projective warps

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

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)





Direct Linear Transform

16-385 Computer Vision

Carnegie Mellon University (Kris Kitani)

We want to estimate the transformation between points…

Do you notice anything about these point correspondences?

We want to estimate the transformation between points…

The transformation of coplanar points can be described by a **projective transform**
(will NOT work for non-coplanar points)

Projective Transform (Homography)

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \alpha \mathbf{H} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

homography

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

parameters of the transform

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

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

projective transform (homography)

Find the best estimate of

$$\boldsymbol{p}$$

Given

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

how do you solve for the parameters?

parameters

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How do you deal with this?

Given

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

how do you solve for the parameters?

parameters

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How do you deal with this?

Direct Linear Transform

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Remove scale factor and get it in a linear form

(rewrite similarity equations as homogenous linear equation and solve → DLT)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Remove scale factor and get it in a linear form

(rewrite similarity equations as homogenous linear equation and solve → DLT)

Multiplied out

$$x' = \alpha(h_1 x + h_2 y + h_3)$$
$$y' = \alpha(h_4 x + h_5 y + h_6)$$
$$1 = \alpha(h_7 x + h_8 y + h_9)$$

Divide out unknown scale factor (divide line 1 and 2 by 3)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Remove scale factor and get it in a linear form

(rewrite similarity equations as homogenous linear equation and solve → DLT)

Multiplied out

$$x' = \alpha(h_1 x + h_2 y + h_3)$$
$$y' = \alpha(h_4 x + h_5 y + h_6)$$
$$1 = \alpha(h_7 x + h_8 y + h_9)$$

Divide out unknown scale factor (divide line 1 and 2 by 3)

$$x'(h_7 x + h_8 y + h_9) = (h_1 x + h_2 y + h_3)$$
$$y'(h_7 x + h_8 y + h_9) = (h_4 x + h_5 y + h_6)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Remove scale factor and get it in a linear form

(rewrite similarity equations as homogenous linear equation and solve → DLT)

Multiplied out

$$x' = \alpha(h_1 x + h_2 y + h_3)$$
$$y' = \alpha(h_4 x + h_5 y + h_6)$$
$$1 = \alpha(h_7 x + h_8 y + h_9)$$

Divide out unknown scale factor (divide line 1 and 2 by 3)

$$x'(h_7 x + h_8 y + h_9) = (h_1 x + h_2 y + h_3)$$
$$y'(h_7 x + h_8 y + h_9) = (h_4 x + h_5 y + h_6)$$

How do you rearrange terms to make it a linear system of equations?

$$x'(h_7x + h_8y + h_9) = (h_1x + h_2y + h_3)$$
$$y'(h_7x + h_8y + h_9) = (h_4x + h_5y + h_6)$$

Just rearrange the terms

$$h_7xx' + h_8yx' + h_9x' - h_1x - h_2y - h_3 = 0$$
$$h_7xy' + h_8yy' + h_9y' - h_4x - h_5y - h_6 = 0$$

$$h_7 x x' + h_8 y x' + h_9 x' - h_1 x - h_2 y - h_3 = 0$$
$$h_7 x y' + h_8 y y' + h_9 y' - h_4 x - h_5 y - h_6 = 0$$

In matrix form:

$$\mathbf{A}_i \boldsymbol{h} = \mathbf{0}$$

$$\mathbf{A}_i = \left[\begin{array}{ccccccccc} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{array} \right]$$

$$\boldsymbol{h} = \left[\begin{array}{ccccccccc} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{array} \right]^\top$$

How many equations from one point correspondence?

$$\mathbf{A}\boldsymbol{h} = \mathbf{0}$$

$$\begin{bmatrix} \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \vdots \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

'Homogeneous Linear Least Squares' problem

How do we solve this?

(usually have more constraints than variables)

$\mathbf{A}\boldsymbol{h} = \mathbf{0}$

$$
\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix}
$$
$$
\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix}
$$
$$
\vdots
$$
$$
\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix}
\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

'Homogeneous Linear Least Squares' problem

Solve with SVD!

Singular Value Decomposition

orthogonal orthogonal

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

diagonal

$$= \sum_{i=1}^{9} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

Each column of V represents a solution for $\mathbf{A}\boldsymbol{h} = \mathbf{0}$

where the eigenvalues represents the reprojection error

Solving for H using DLT

Given $\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$ solve for H such that $\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$

1. For each correspondence, create 2x9 matrix $\mathbf{A}_i$

2. Concatenate into single 2n x 9 matrix $\mathbf{A}$

3. Compute SVD of $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$

4. Store Eigenvector of the smallest Eigenvalue $\boldsymbol{h} = \boldsymbol{v}_{\hat{i}}$

5. Reshape to get $\mathbf{H}$

General form of total least squares

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

$$E_{\mathrm{TLS}} = \sum_i (\boldsymbol{a}_i \boldsymbol{x})^2$$

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

$$\|\boldsymbol{x}\|^2 = 1 \quad \text{constraint}$$

minimize $\|\mathbf{A}\boldsymbol{x}\|^2$

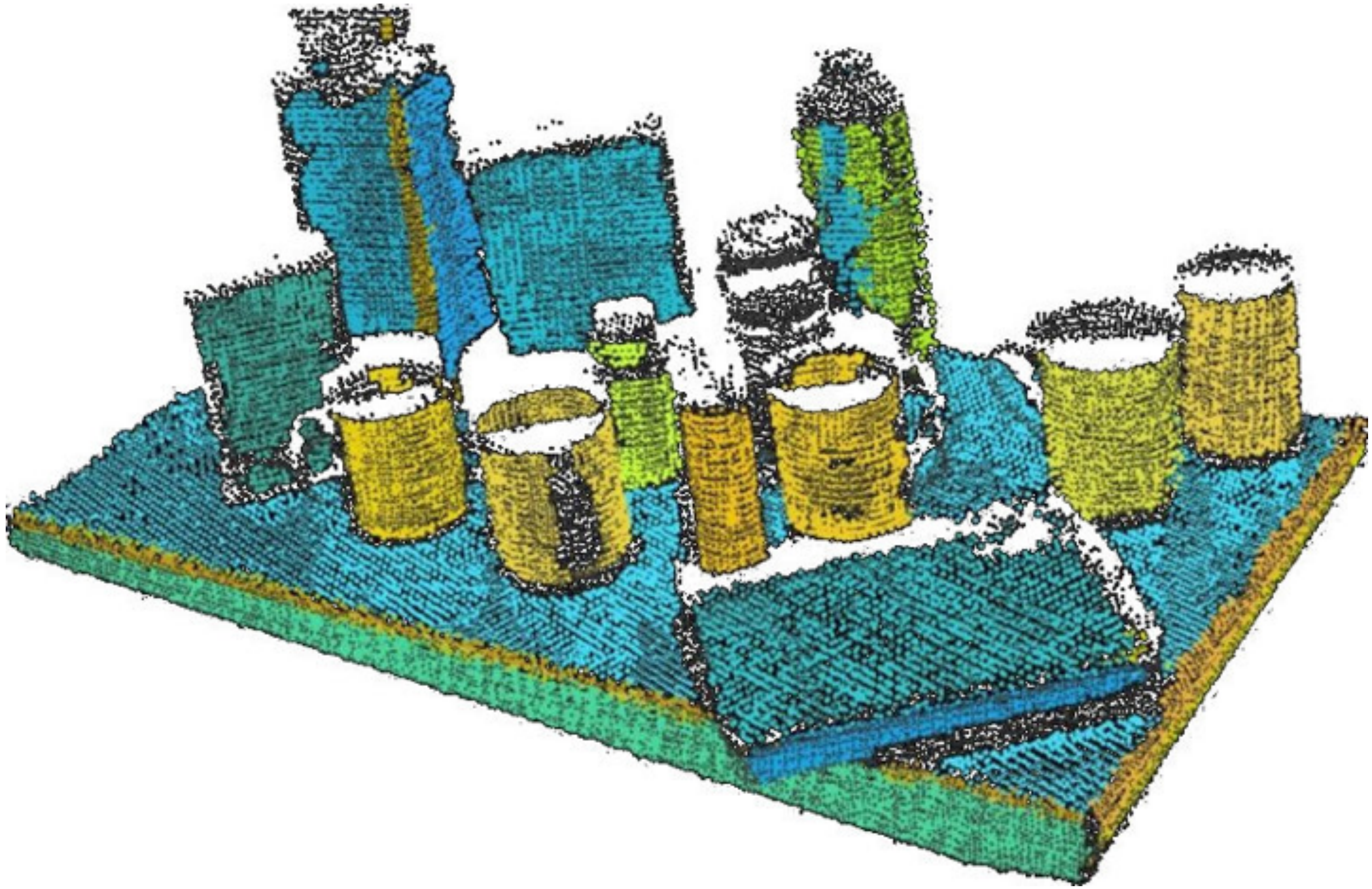
subject to $\|\boldsymbol{x}\|^2 = 1$

→ minimize $\dfrac{\|\mathbf{A}\boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^2}$

(Rayleigh quotient)

Solution is the eigenvector corresponding to smallest eigenvalue of

$$\mathbf{A}^\top \mathbf{A}$$

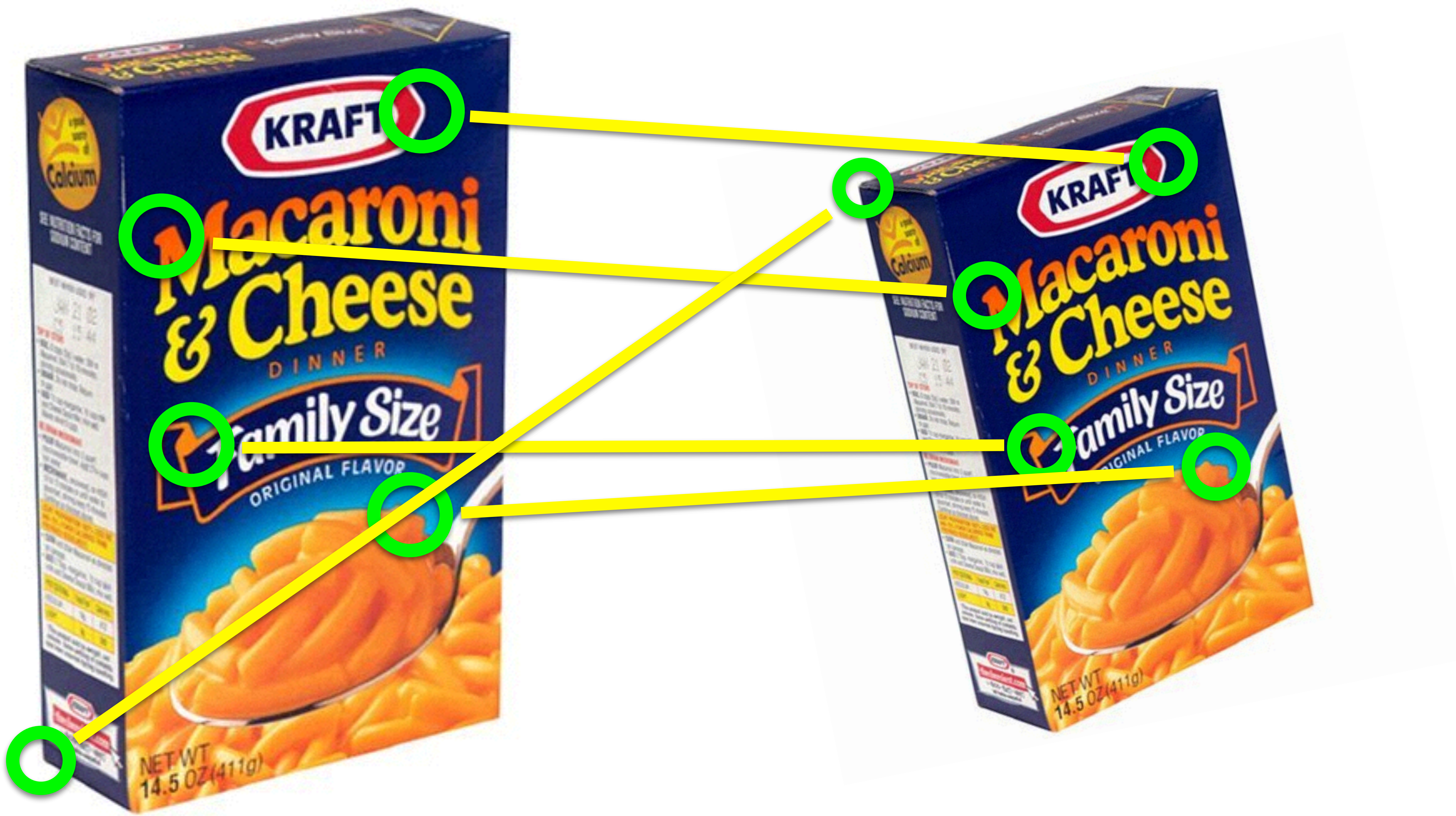


RANSAC

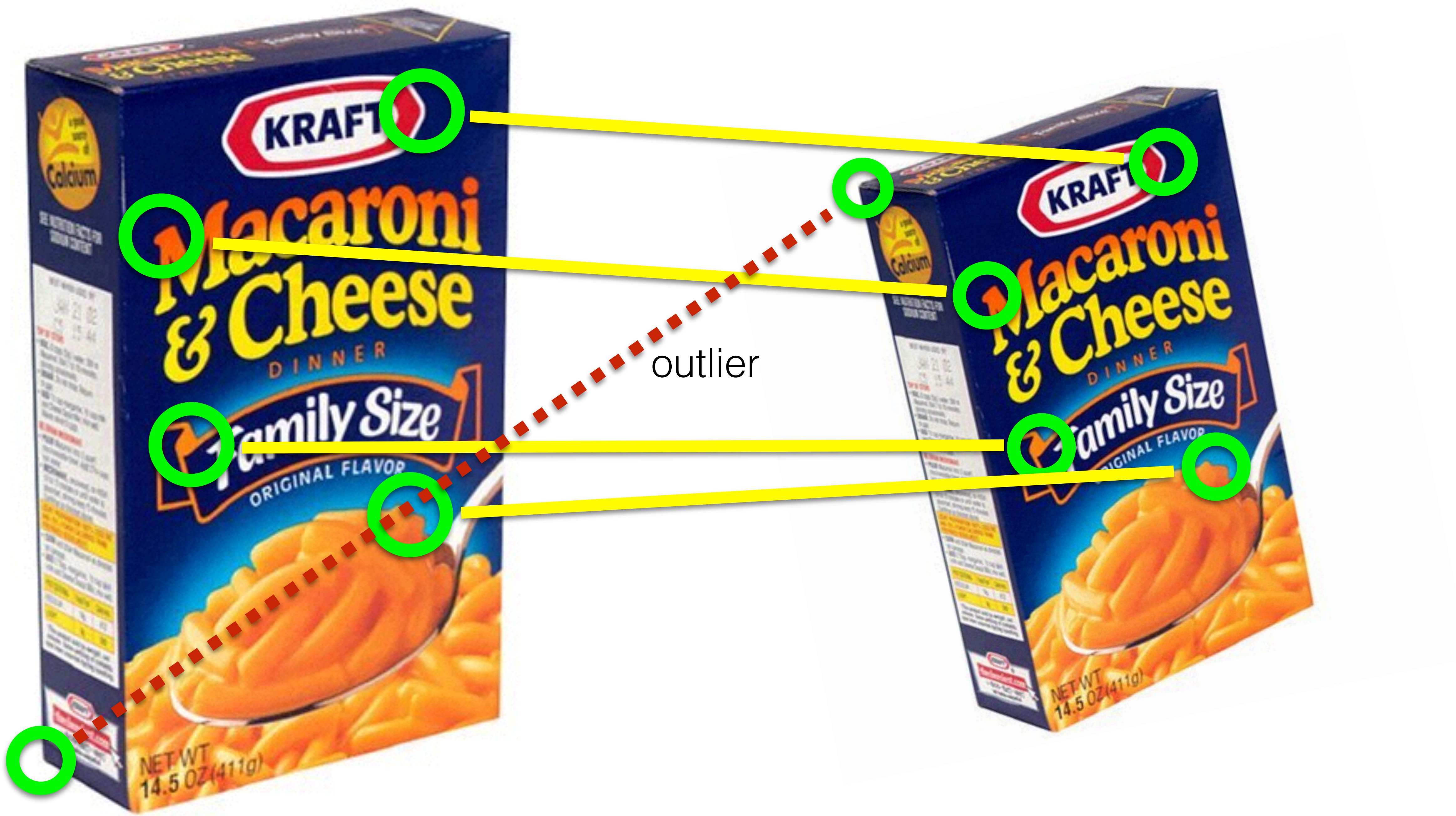
16-385 Computer vision

Carnegie Mellon University (Kris Kitani)

Up to now, we've assumed correct correspondences



What if there are mismatches?



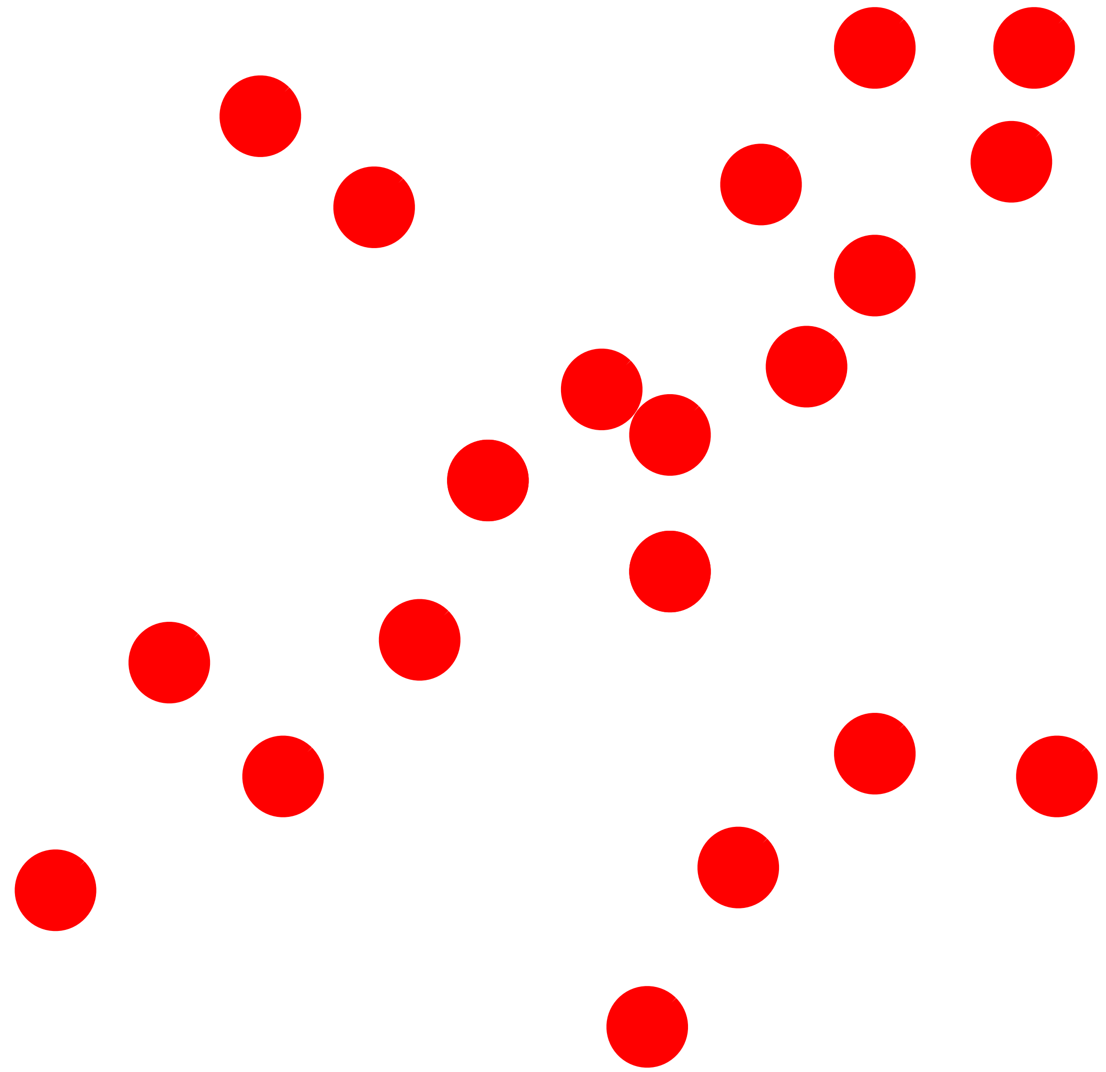
How would you find just the inliers?

RANSAC

RANdom **SA**mple **C**onsensus

[Fischler & Bolles in ’81]

Fitting lines
(with outliers)

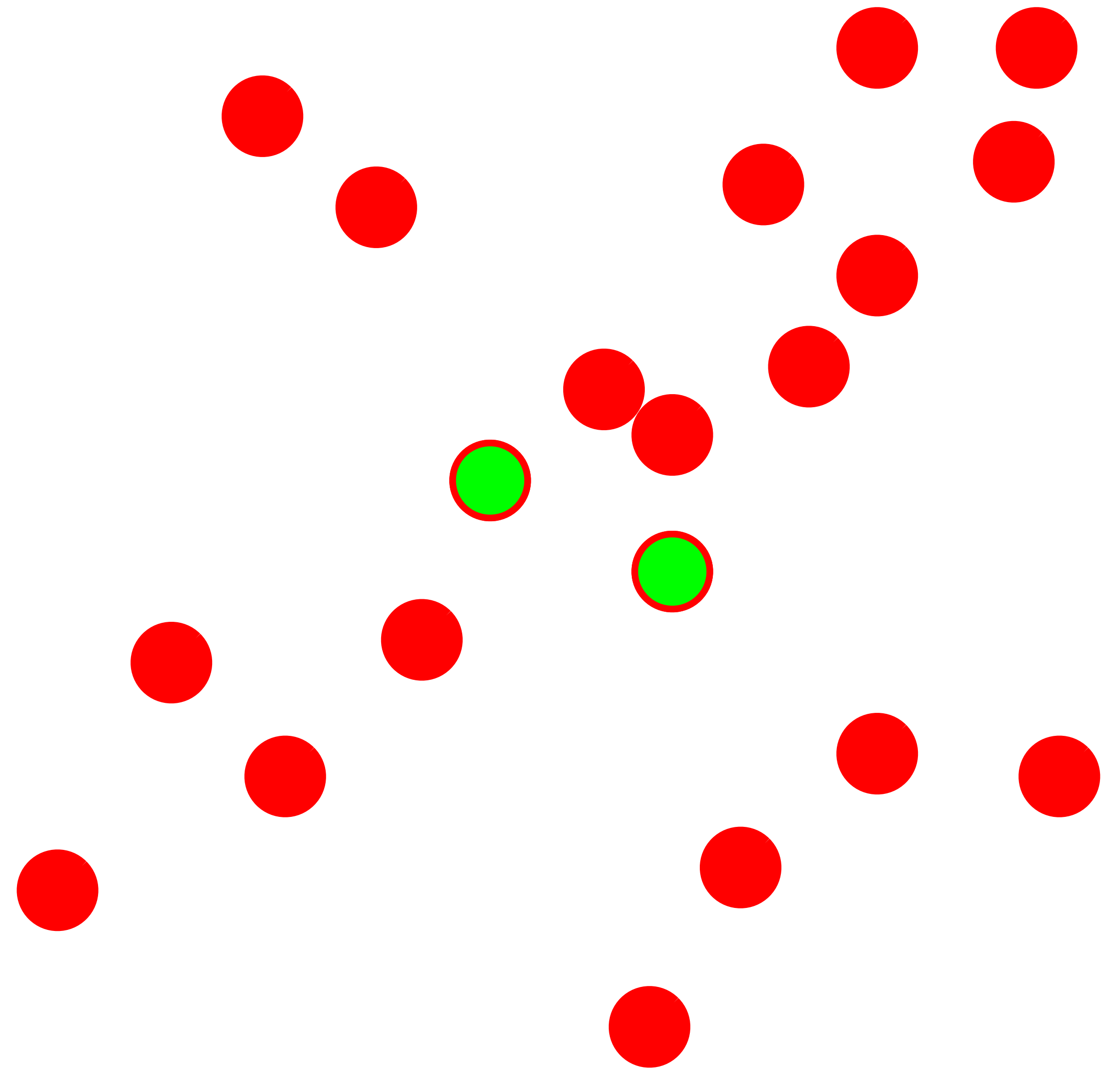


Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. Score by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Fitting lines
(with outliers)

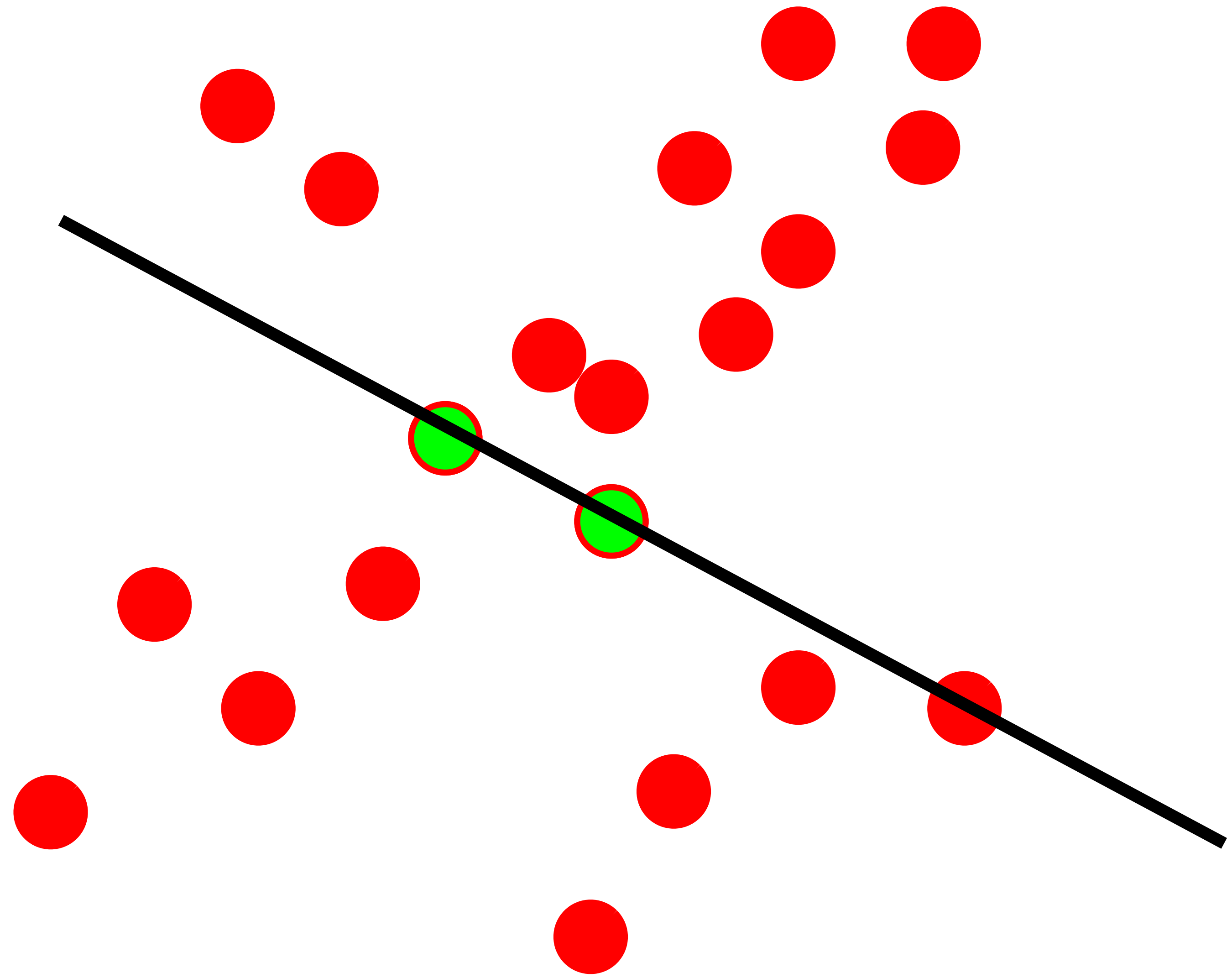


Algorithm:

1. **Sample (randomly) the number of points required to fit the model**
2. Solve for model parameters using samples
3. Score by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Fitting lines
(with outliers)

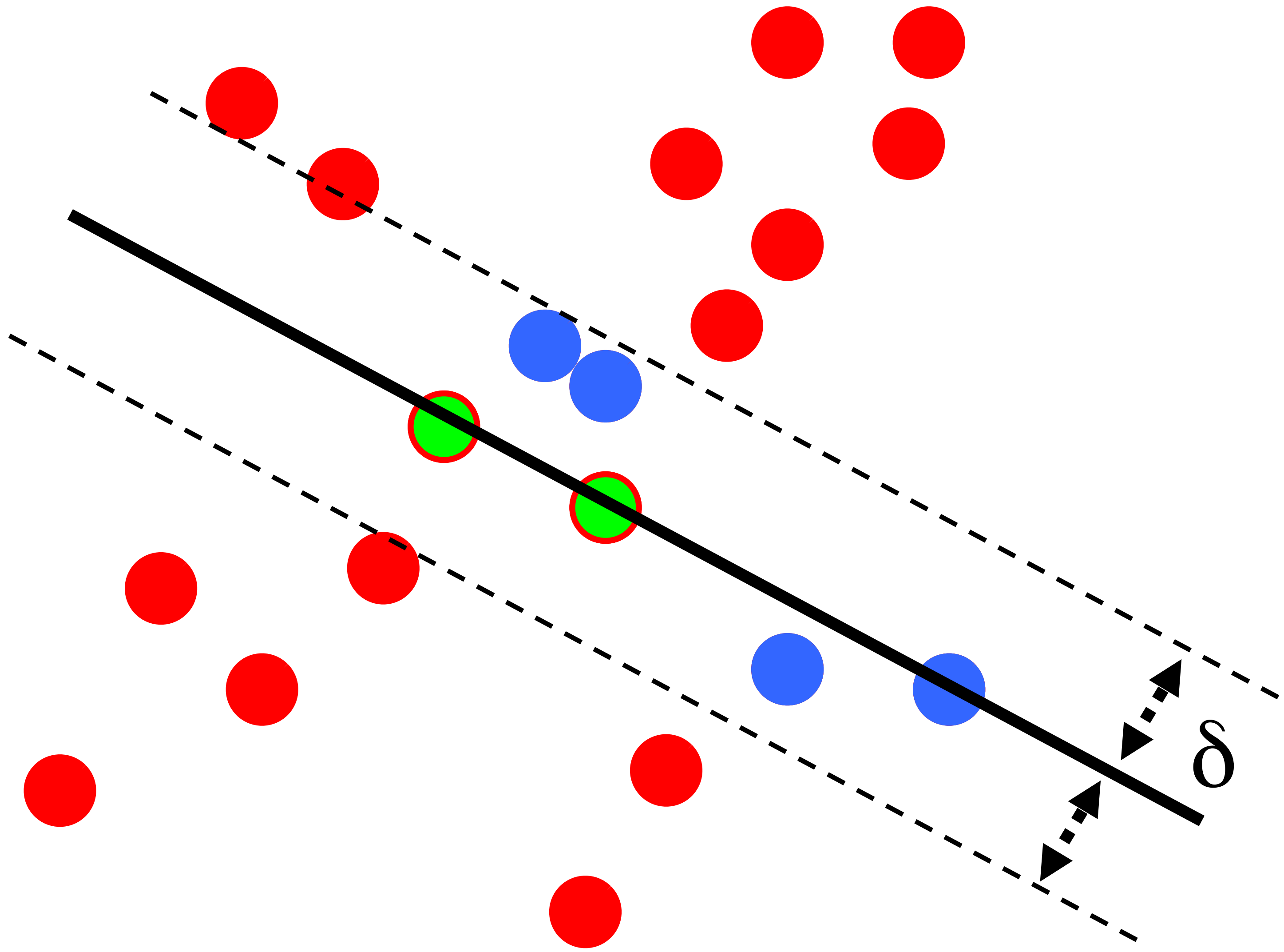


Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. **Solve for model parameters using samples**
3. Score by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Fitting lines
(with outliers)



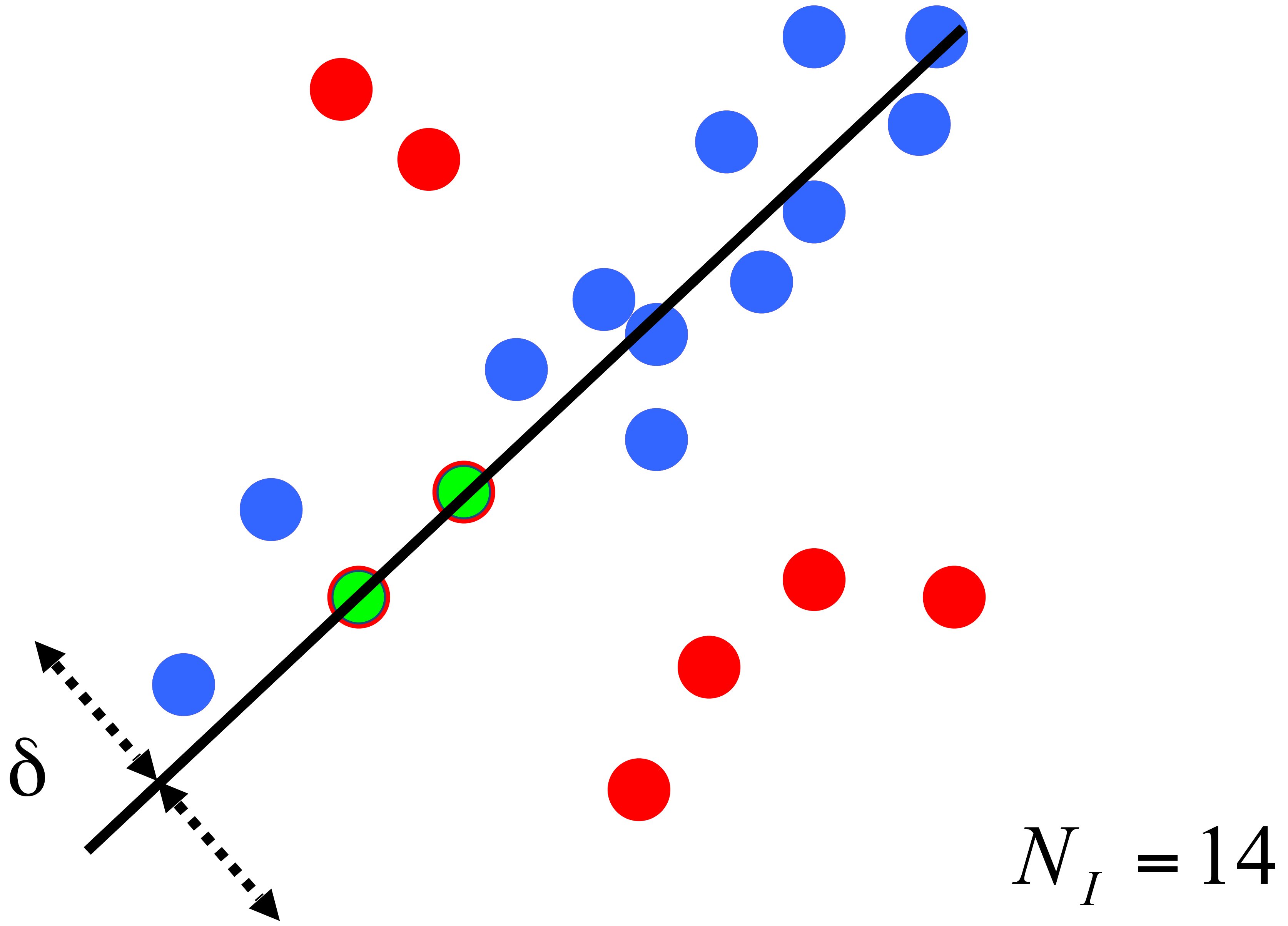
$N_I = 6$

Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. **Score by the fraction of inliers within a preset threshold of the model**

Repeat 1-3 until the best model is found with high confidence

Fitting lines
(with outliers)



Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. Score by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

How to choose parameters?

- Number of samples N
  - Choose N so that, with probability p, at least one random sample is free from outliers (e.g. p=0.99) (outlier ratio: e)
- Number of sampled points s
  - Minimum number needed to fit the model
- Distance threshold δ
  - Choose δ so that a good point with noise is likely (e.g., prob=0.95) within threshold
  - Zero-mean Gaussian noise with std. dev. σ: $t^2=3.84\sigma^2$

$$N = \frac{\log(1-p)}{\log\left(1-(1-e)^s\right)}$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Good

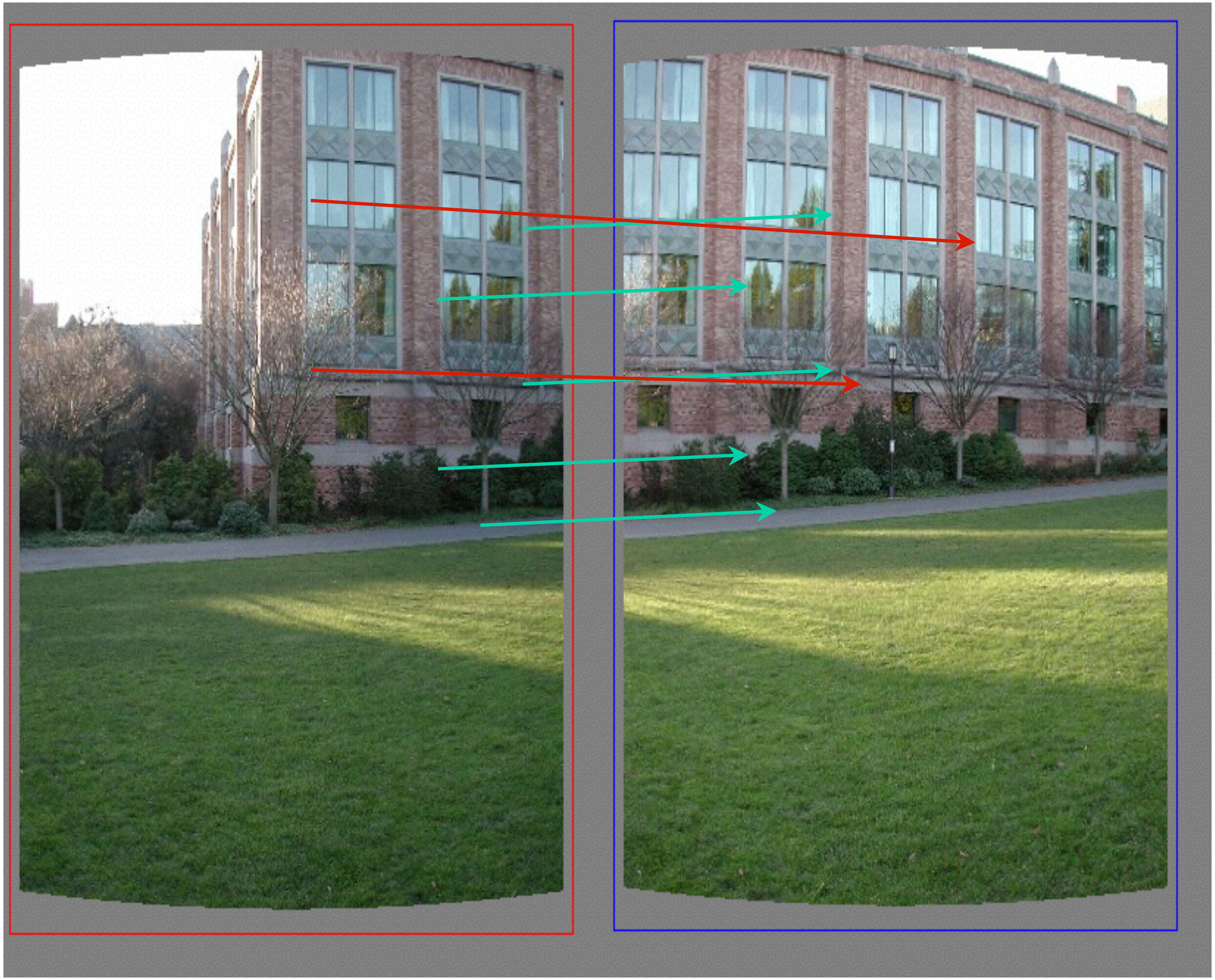
- Robust to outliers
- Applicable for larger number of parameters than Hough transform
- Parameters are easier to choose than Hough transform

Bad

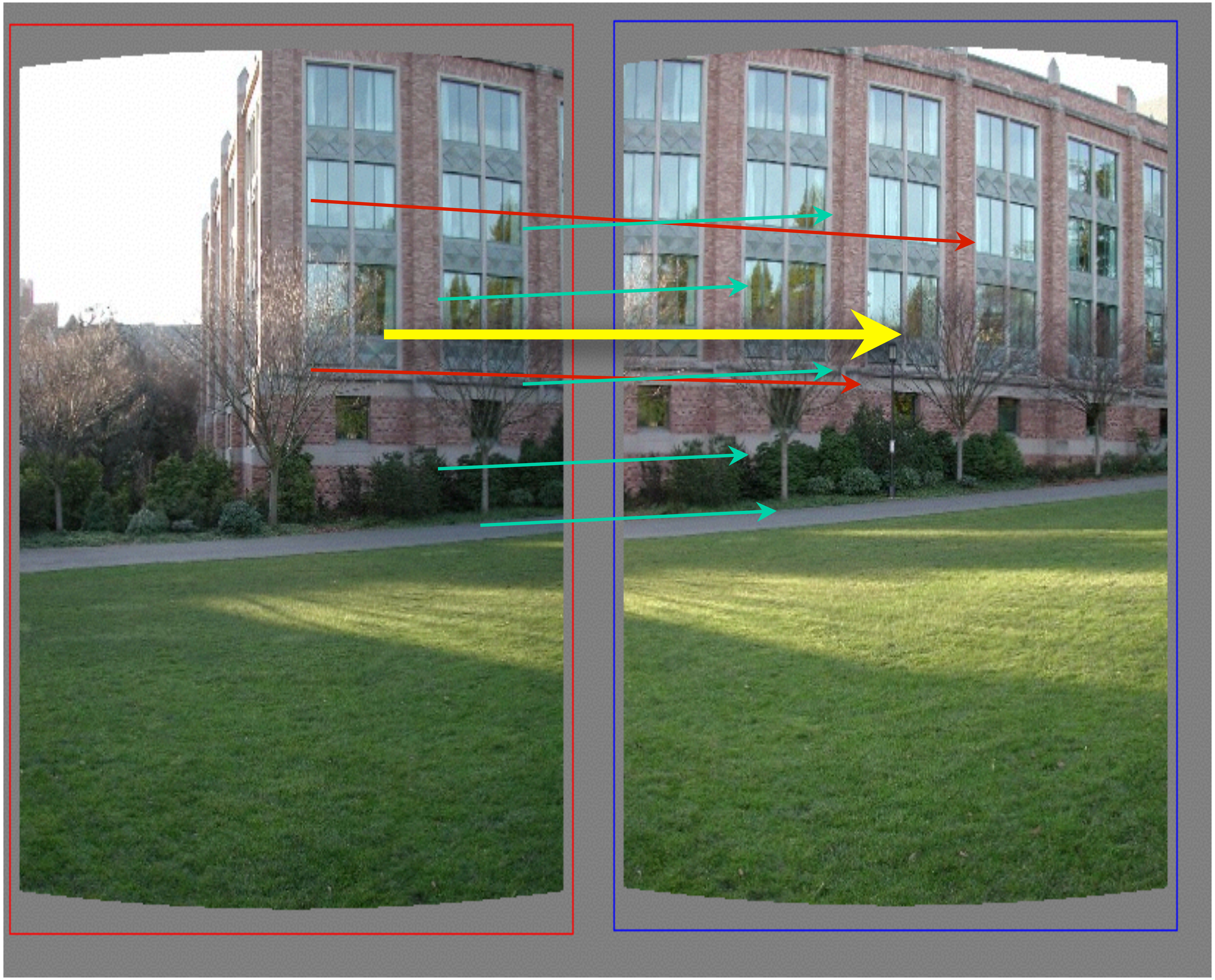
- Computational time grows quickly with fraction of outliers and number of parameters
- Not good for getting multiple fits

Common applications

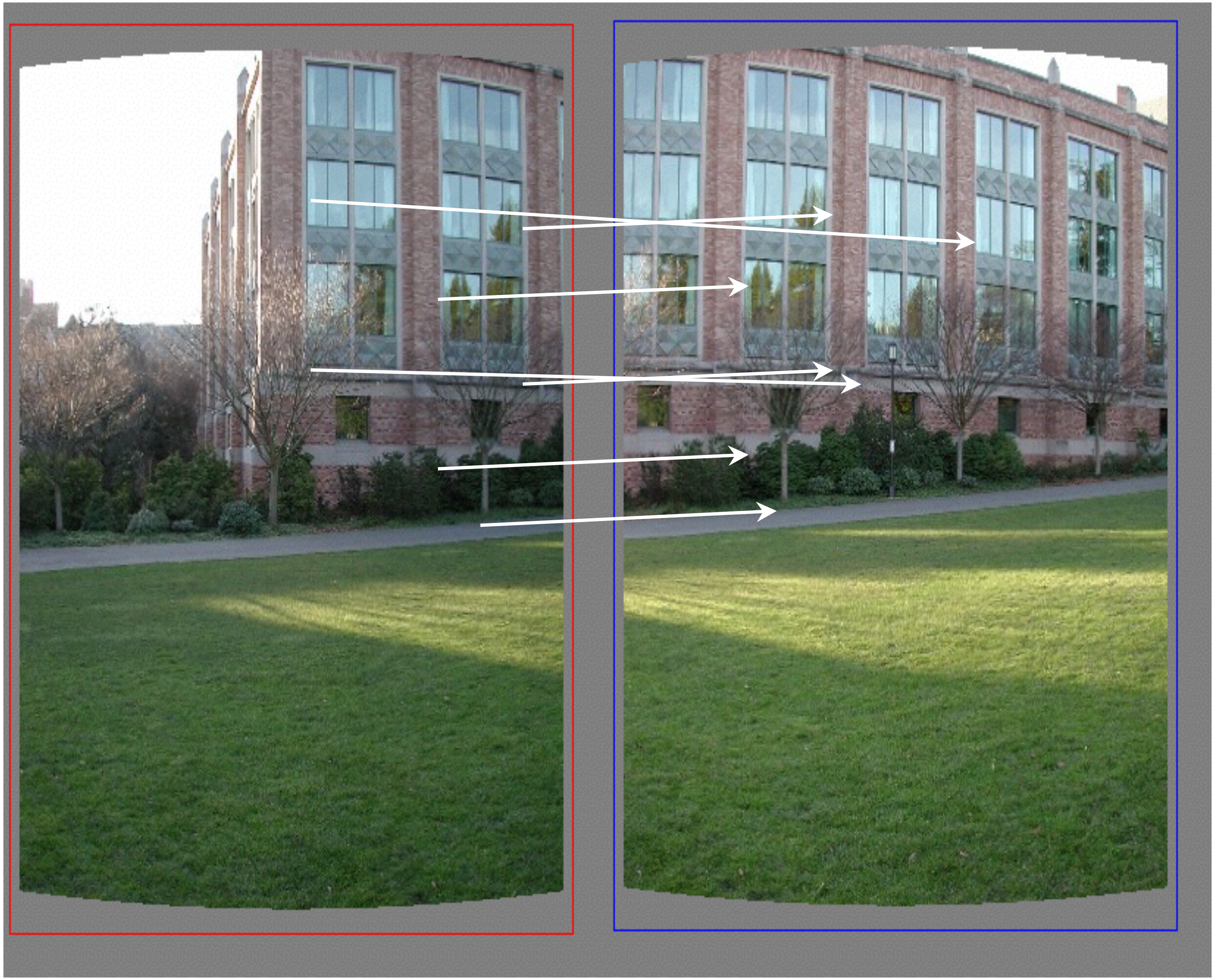
- Computing a homography (e.g., image stitching)
- Estimating fundamental matrix (relating two views)



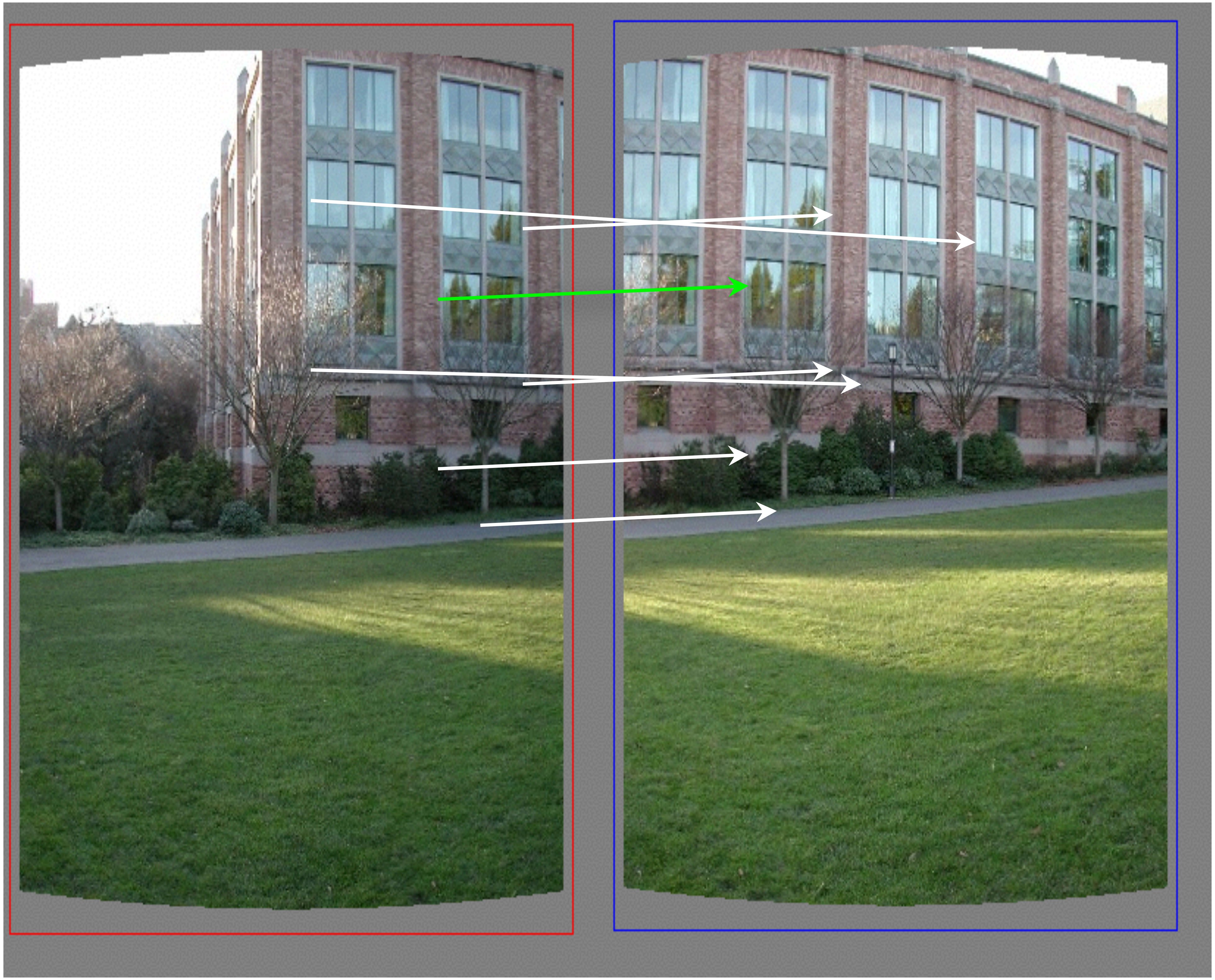
Matched points



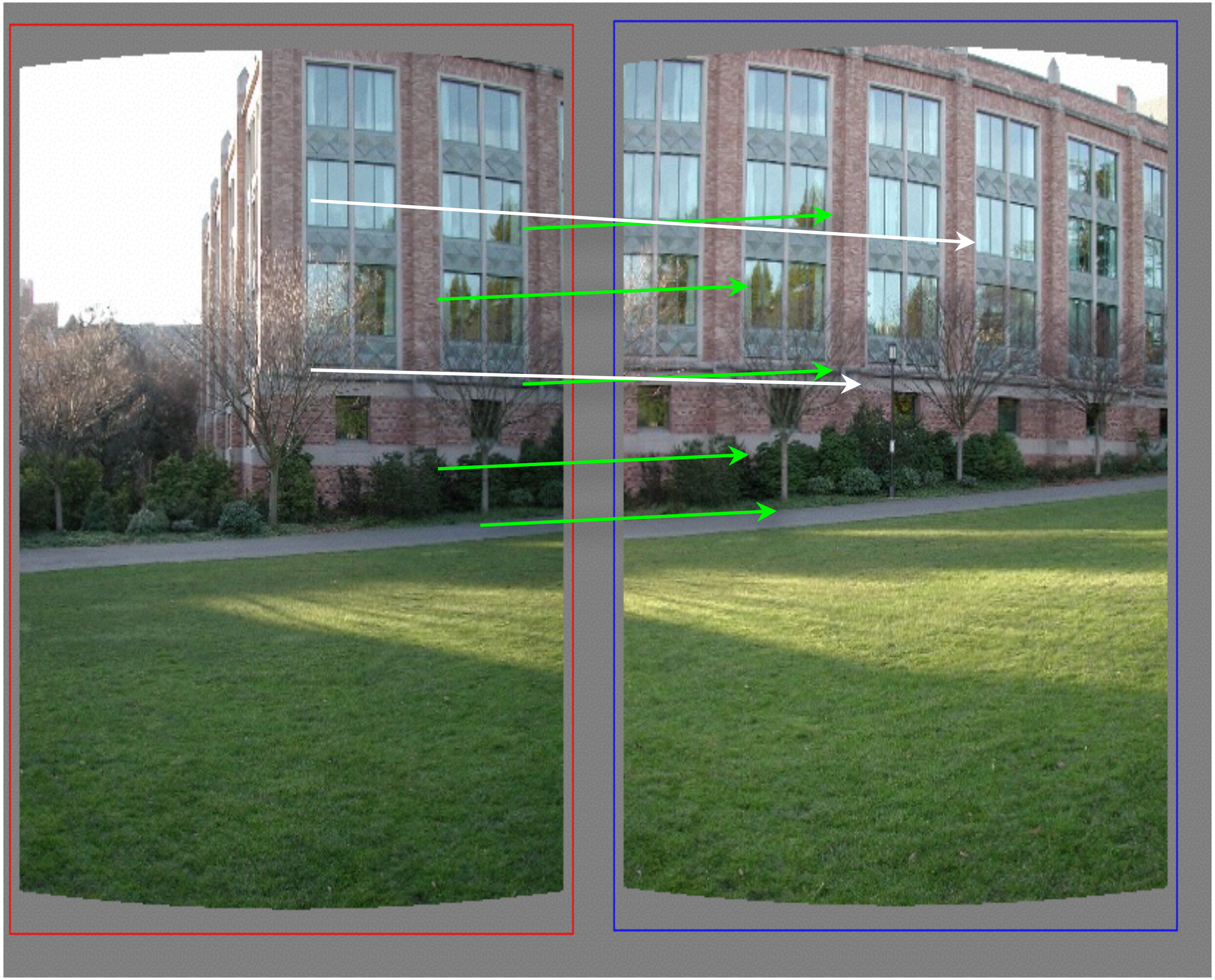
Least Square fit finds the ‘average’ transform



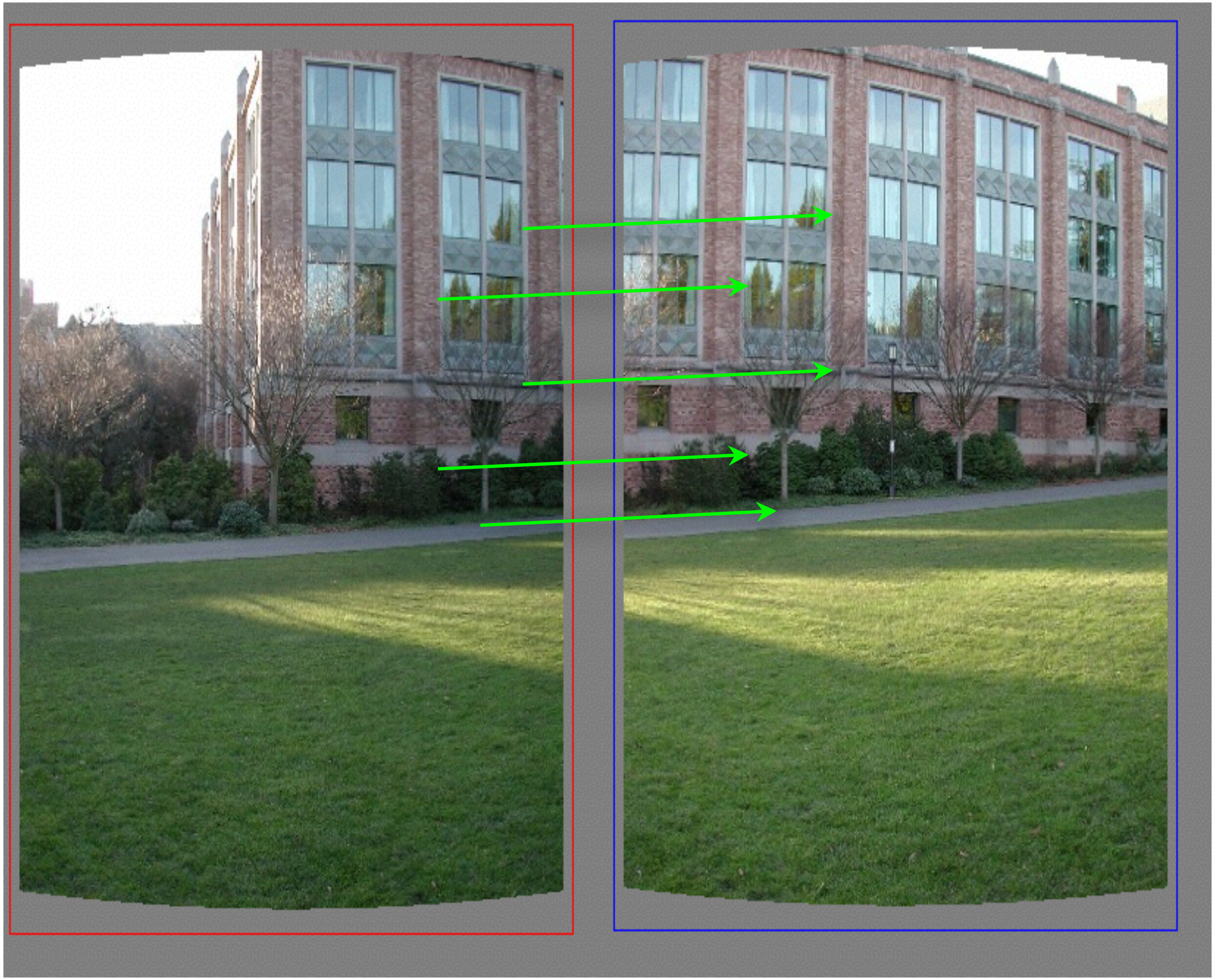
RANSAC: Use one correspondence, find inliers



RANSAC: **Use one correspondence**, find inliers



RANSAC: Use one correspondence, **find inliers**



RANSAC: Use one correspondence, **find inliers**

Estimating homography using RANSAC

- RANSAC loop
    1. Get four point correspondences (randomly)
    2. Compute H (DLT)
    3. Count inliers
    4. Keep if largest number of inliers
- Recompute H using all inliers

Useful for…