# CS195: Computer Vision

## Image Transformation (part 2)

Parametric Warping

December, 2nd, 2024

Md Alimoor Reza
Assistant Professor of Computer Science

Drake University

# Announcement

- **Starting on December 1st**, there will be a puzzle released daily. They typically increase in difficulty, but the first few days are typically pretty fun and easy.


**Drake Faculty & Student Advent of Code Challenge!**

Winners get prizes!!!

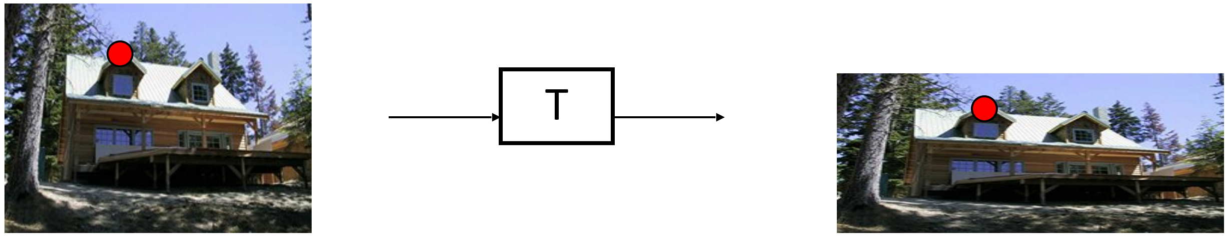
Daily puzzles – starting on December 1st

No ChatGPT Allowed

To Enter: Create an account at https://adventofcode.com/

Click on “Leaderboard” and enter: `2496015-51ebce8e`

# Recap: 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)

**For linear transformations**, we can represent T as a matrix

p’ = **M**p

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

# Recap: Parametric (global) warping

## Examples of parametric warps:

translation

rotation

aspect

affine

perspective

cylindrical

# Recap: Group Activity#6

## Find the transformed coordinates with given 2x2 transformation matrix



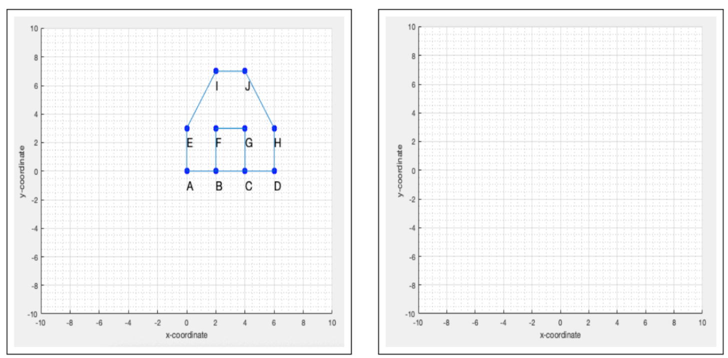
### In-class activity#6

CS 195: Computer Vision, Fall'24

**Image Transformation (*6 pts*).**

You will be transforming 2D points in this problem. As shown in the Figure below, there are ten 2D coordinates ('A', 'B', ..., 'J') collectively forming a house's shape. The origin is at location *'A'*.

a) Suppose, you are given a 2x2 transformation matrix $\mathbf{M}_1 = \begin{bmatrix} 0.87 & -0.50 \\ 0.50 & 0.87 \end{bmatrix}$. What would be the shape of the house after applying this transformation on the 2D coordinates?




Please write down the transformed coordinates or show them on the empty graph above. What type of transformation does this matrix signify?

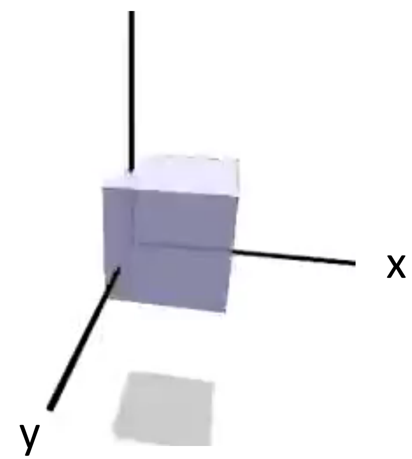
Only linear 2D transformations can be represented with a 2x2 matrix

# 2x2 Matrices

What types of transformations can be represented with a 2x2 matrix besides: rotation, scaling, shear, reflection

Can we represent 2D Translation with a 2x2 matrix from the following equation?

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




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

# 2x2 Matrices

What types of transformations can be represented with a 2x2 matrix besides:

Can we represent 2D Translation with a 2x2 matrix from the following equation?

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

NO!

ADDITIVITY RULE:

$f_P(x+y) = x+y+P$

$f_P(x) + f_P(y) = (x+P) + (y+P)$
$= x + y + 2P$

SCALING RULE

$f_P(sx) = sx + P$

$s f_P(x) = s(x+P)$
$= sx + sP$

| | |
|---|---|
| $f(\mathbf{u}+\mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$ | additivity / operation of addition |
| $f(c\mathbf{u}) = cf(\mathbf{u})$ | homogeneity of degree 1 / operation of scalar multiplication |

# Solution: Homogeneous Coordinates

- How can we represent translation as a matrix?

$$x' = x + t_x$$

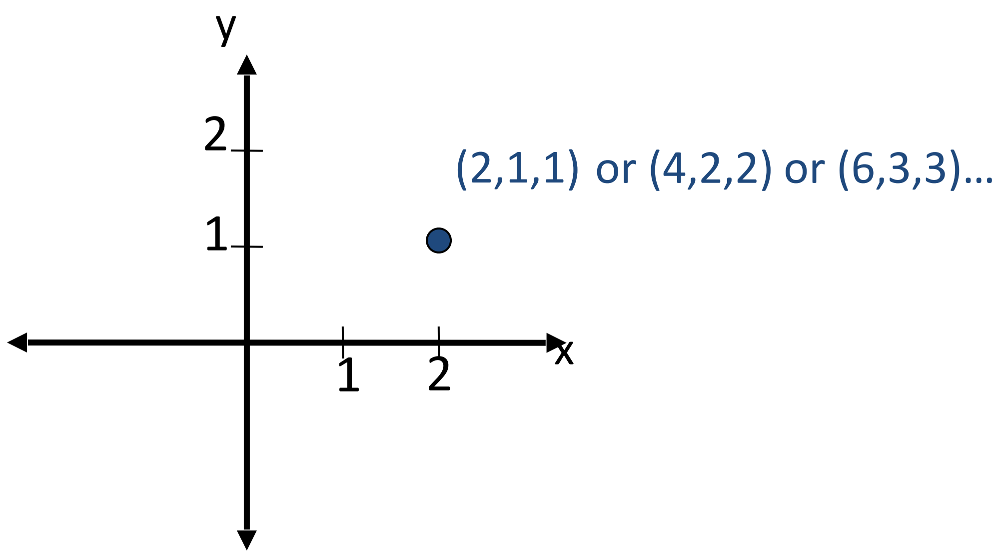
$$y' = y + t_y$$

- Homogeneous coordinates
  – Embed 2-d coordinates 3-d space

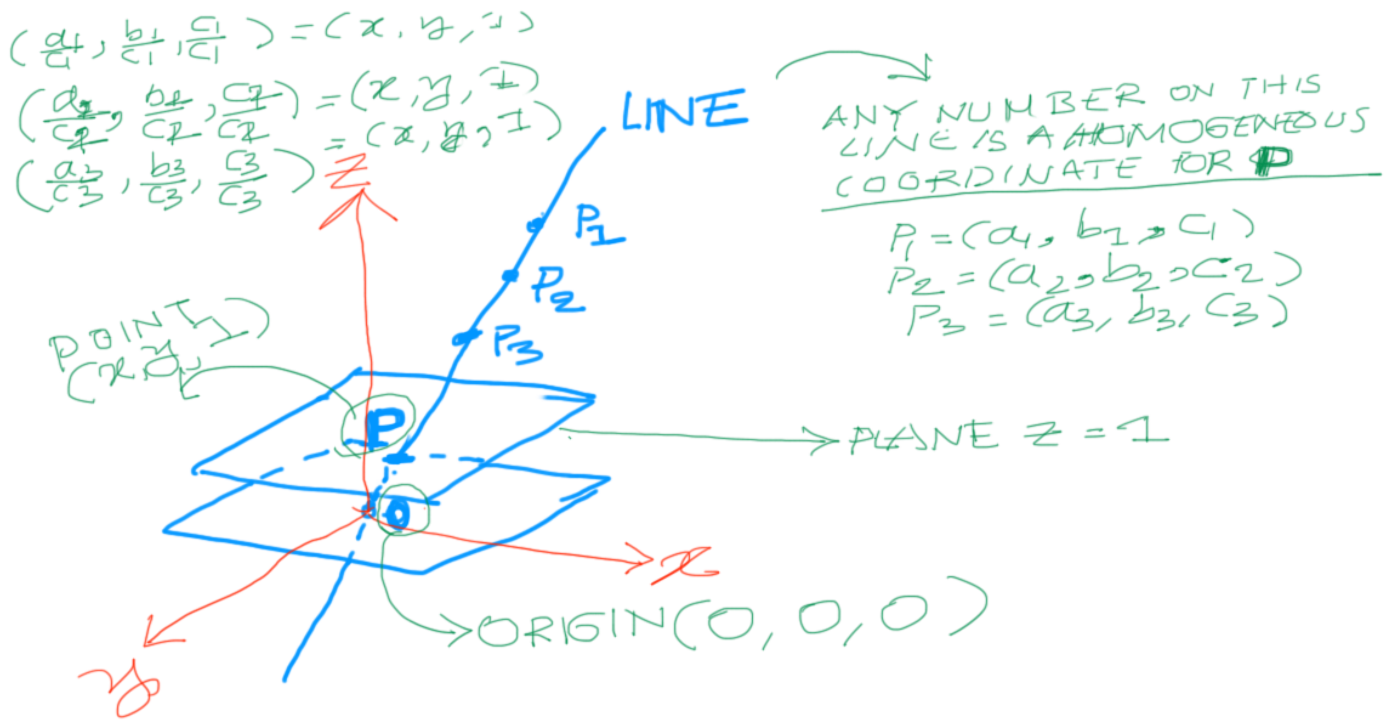
$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Homogeneous Coordinates

- Embed in higher-dimensional space
  - (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




IMAGINE, 2D PLANE THAT DOESNOT PASS THROUGH ORIGIN O

# Homogeneous Coordinates

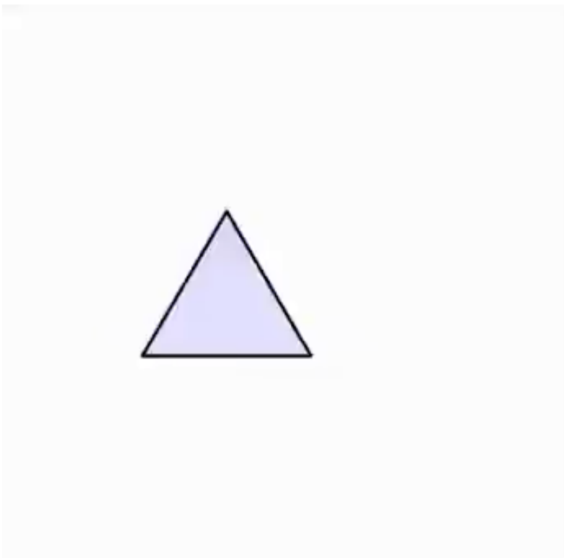


Fig. Bird-eye view or viewing from the top

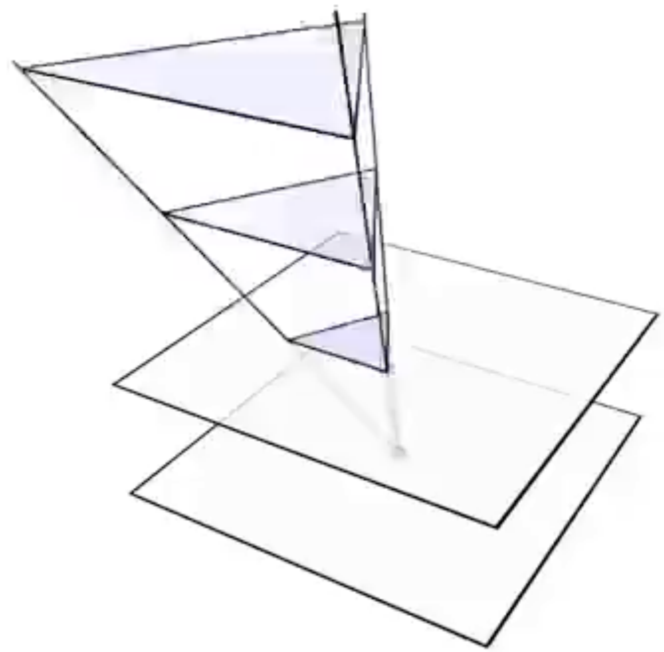


Fig. Viewing from the side

- In order to see the 2D translation, we need to observe the top surface from a top-down perspective on the x-y plane.

# Homogeneous Coordinates

- How can we represent translation as matrix?

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

- Use a 3x3 transformation matrix, T
  - Translation corresponds to last column of matrix

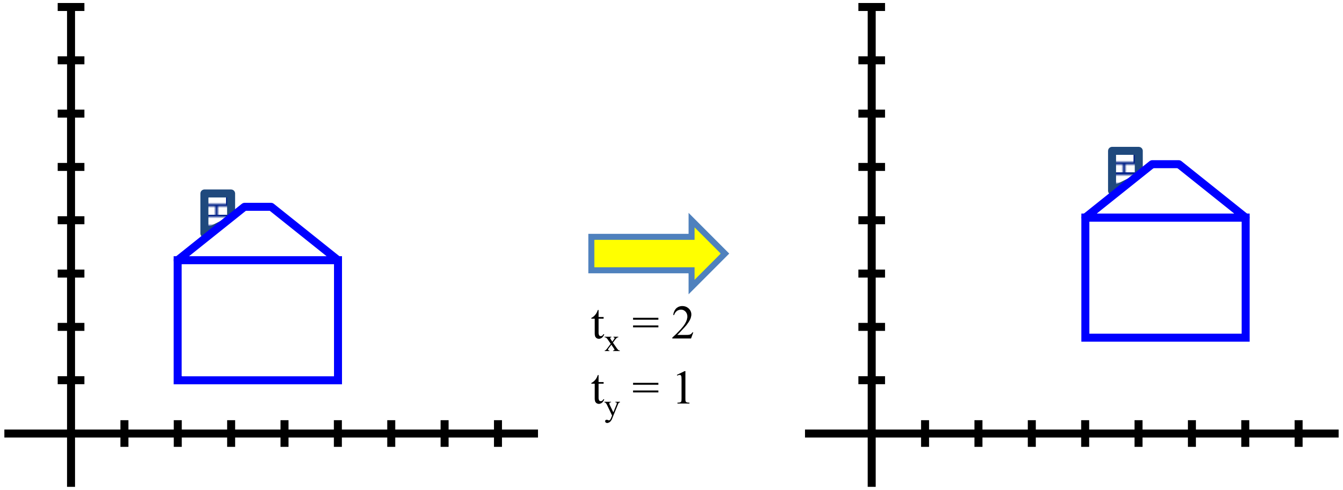
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

# Translation

- Example of translation in homogeneous coordinates

$$\begin{array}{ccc} & \textbf{Matrix} & \textbf{Vector} \end{array}$$

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



# Homogeneous 2D Transformations

- Basic 2D transformations (others including 2D translation) as 3x3 matrices

$$\begin{array}{cc} & \textbf{Matrix} \quad \textbf{Vector} \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = & \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array}$$

Translate

$$\begin{array}{cc} & \textbf{Matrix} \quad \textbf{Vector} \\ \begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = & \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix} \end{array}$$

Scale

Numbers can be placed in the **first 6 entries of each** matrix. However, they cannot be placed in the bottom-left two entries.

$$\begin{array}{cc} & \textbf{Matrix} \quad \textbf{Vector} \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = & \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array}$$

Rotate

$$\begin{array}{cc} & \textbf{Matrix} \quad \textbf{Vector} \\ \begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = & \begin{bmatrix} 1 & \boldsymbol{sh}_x & 0 \\ \boldsymbol{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix} \end{array}$$

Shear

# Affine Transformations

- Affine transformations are combinations of linear transformations (rotation, scaling, shear, mirror) and translations

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

**Affine transformation:** when any combination of numbers can be placed in the first six entries of a matrix

- 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 triangles to triangles

# Projective Transformations

- Projective transformations are combinations of affine transformations and projective warps

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

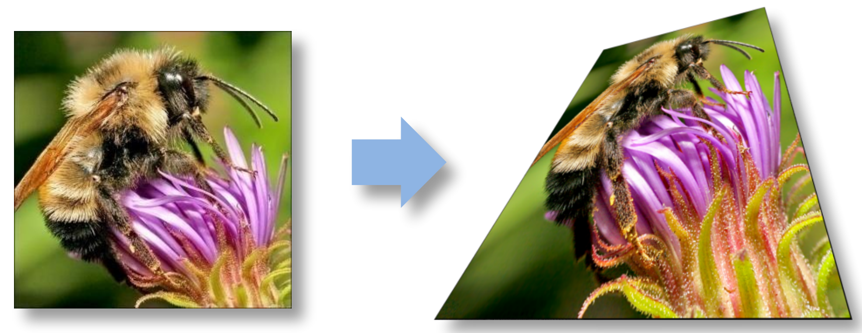
- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition
  - Models change of basis
  - Maps any quadrilateral to any quadrilateral

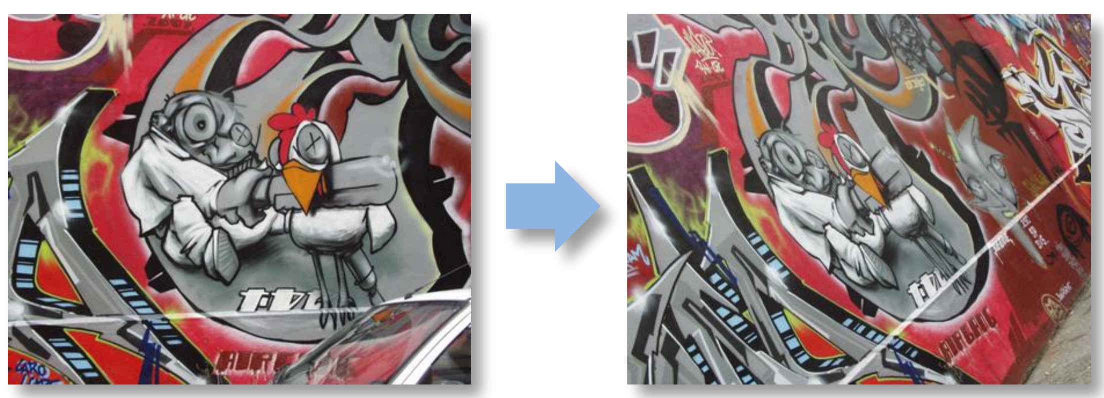
**Projective transformation:** when any combination of numbers can be placed in all 9 entries of a matrix

# Projective Transformations

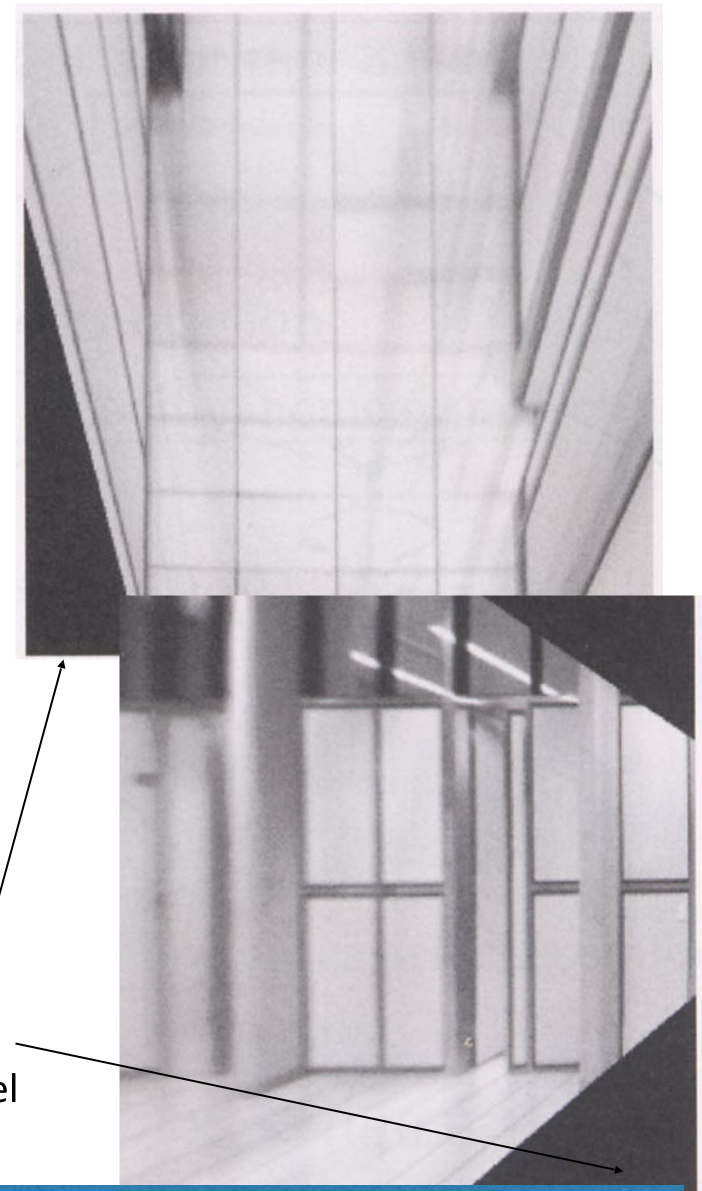
$$\mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Called a *homography*
(or *planar perspective map*)





# Image warping with homographies



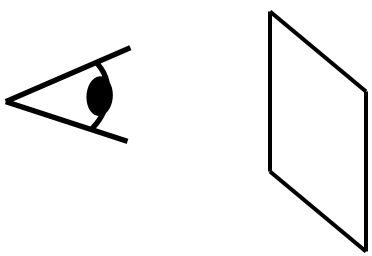
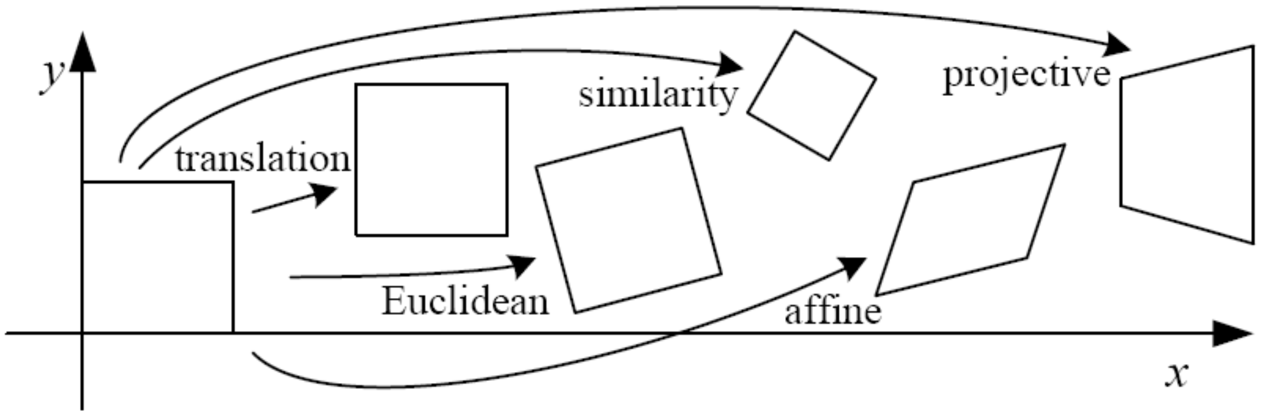




image plane in front

black area
where no pixel
maps to

# 2D image transformations

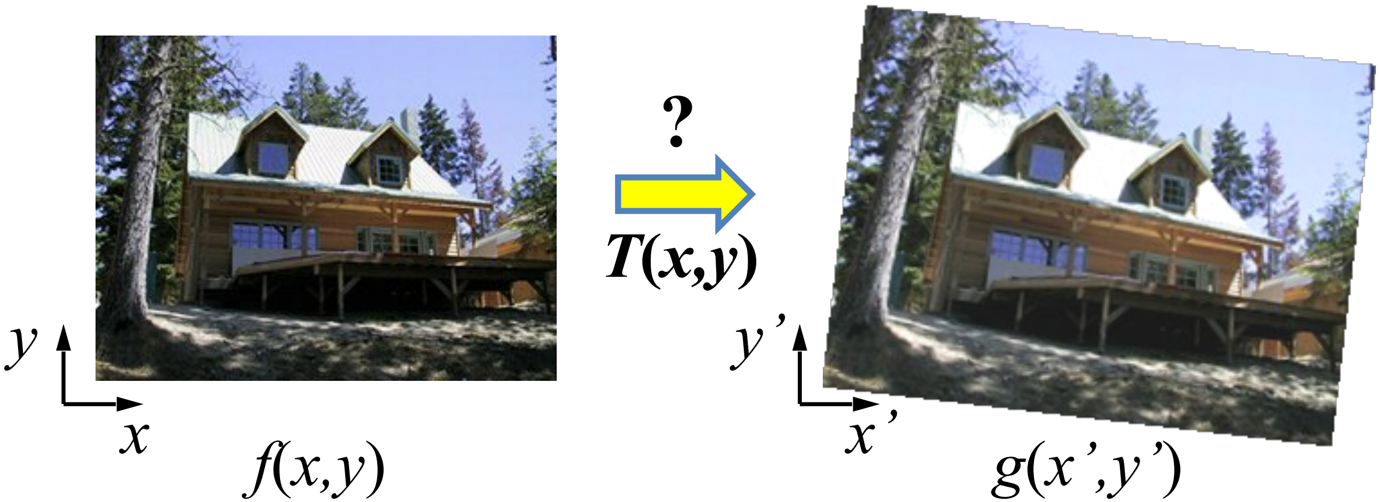



| Name | Matrix | # D.O.F. | Preserves: | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$ | | | |
| rigid (Euclidean) | $\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$ | | | |
| similarity | $\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$ | | | |
| affine | $\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times3}$ | | | |
| projective | $\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times3}$ | | | |

These transformations are a nested set of groups

- Closed under composition and inverse is a member

# Recovering Transformations




- What if we know *f* and *g* and want to recover the transform T?

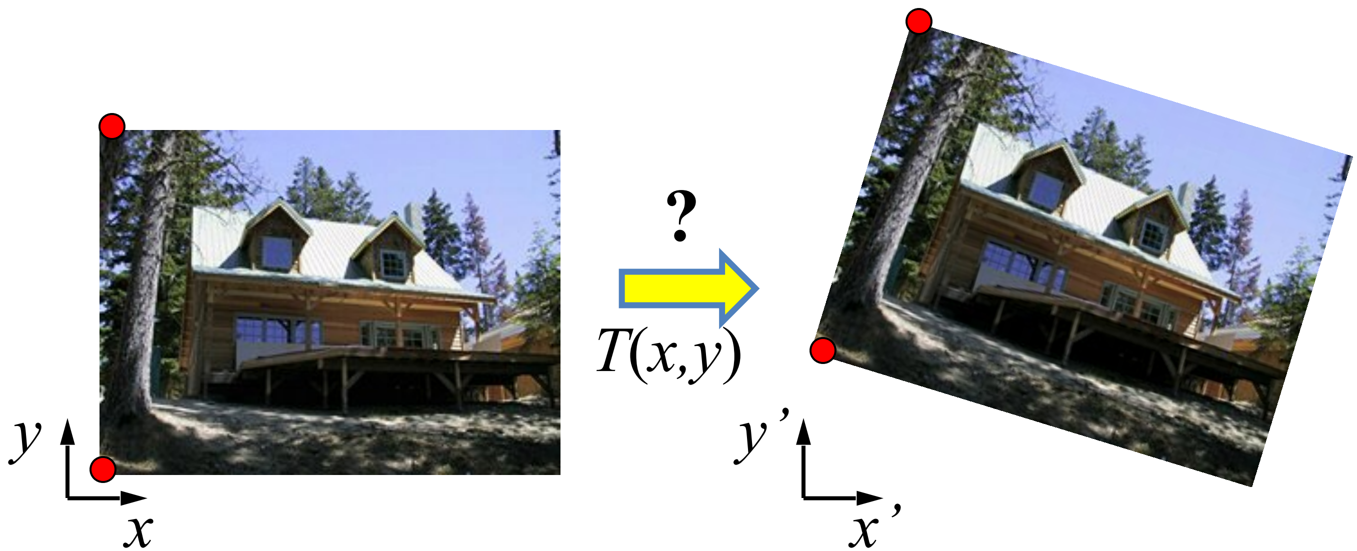
# Translation: # correspondences?


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

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

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

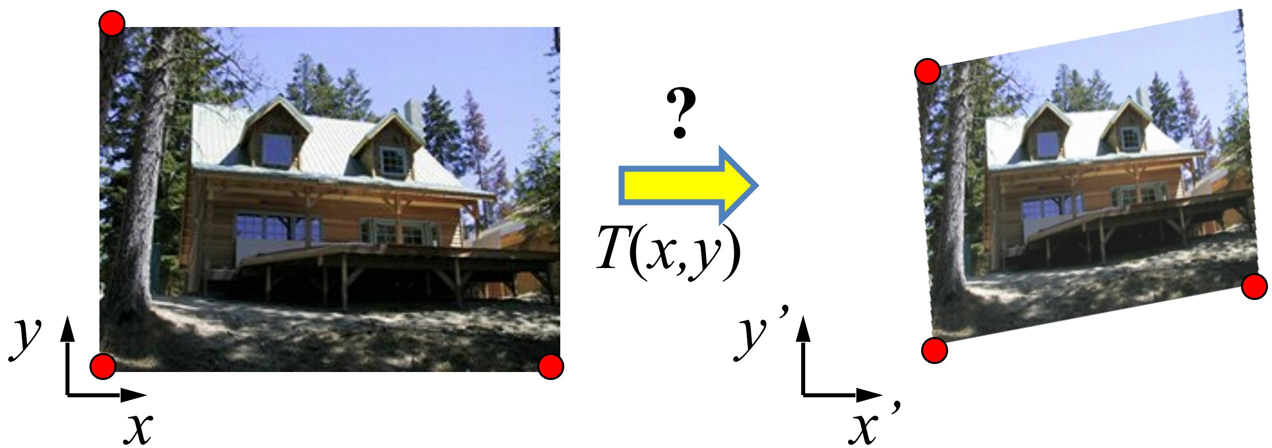
# Euclidean: # correspondences?




- How many DOF?
- How many correspondences needed for translation+rotation?

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

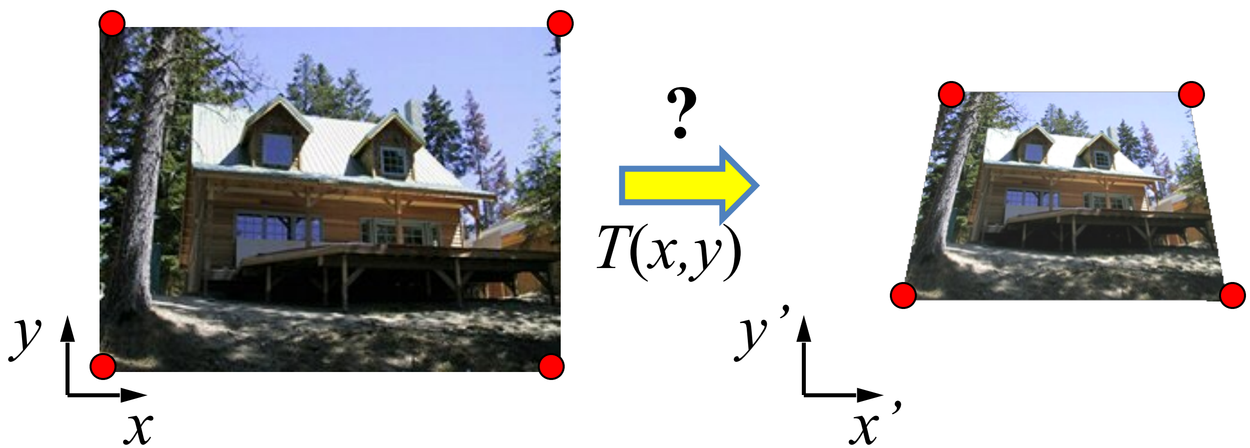
# Affine: # correspondences?




- How many DOF?
- How many correspondences needed for affine?

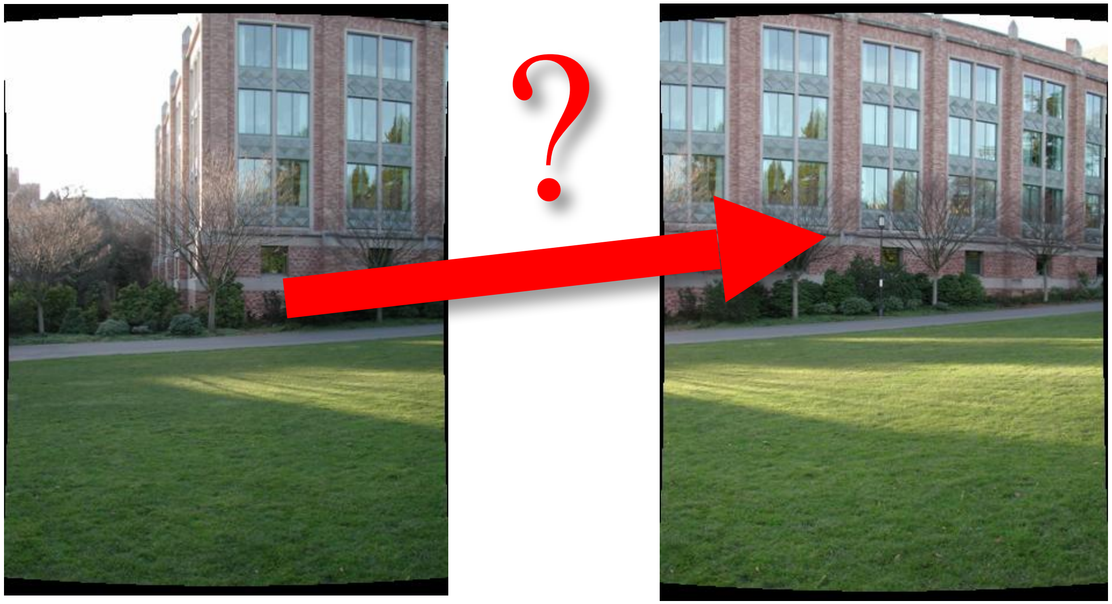
$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

# Projective: # correspondences?



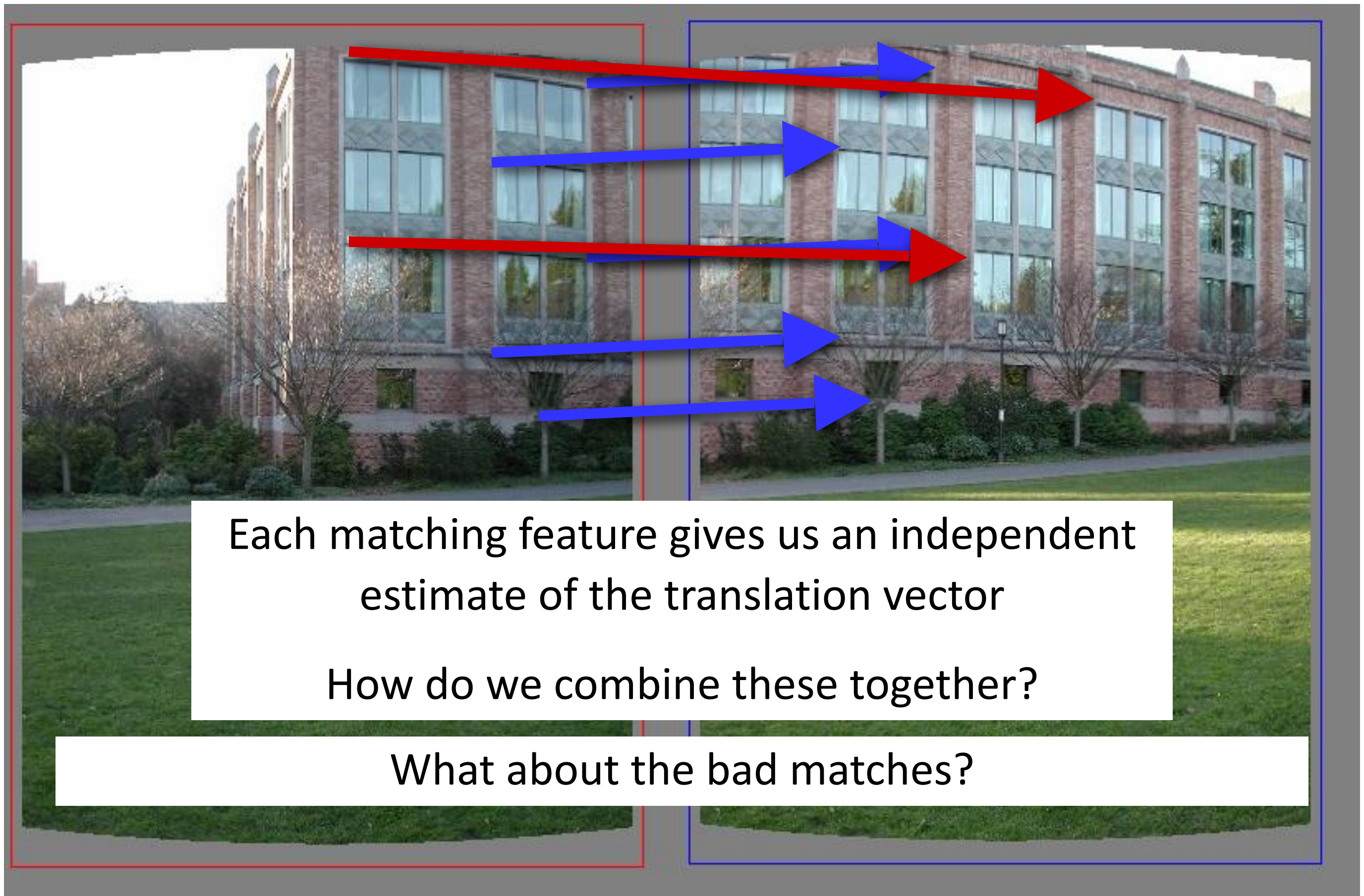

- How many DOF?
- How many correspondences needed for projective?

# Image alignment




- Given two images, how do we compute the transformation that aligns them?
  - Answer: use feature matching

# Feature-based alignment

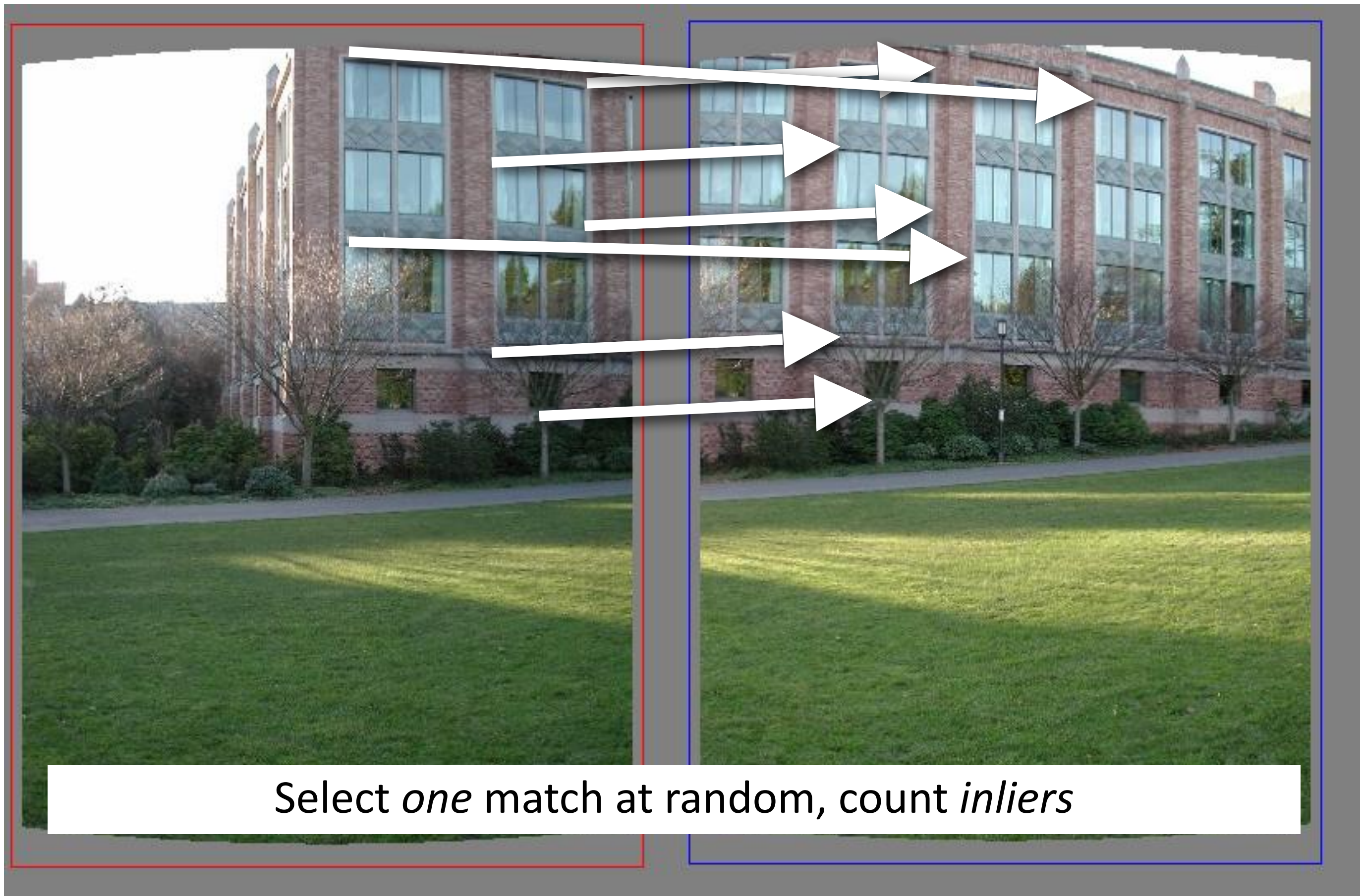


Each matching feature gives us an independent estimate of the translation vector

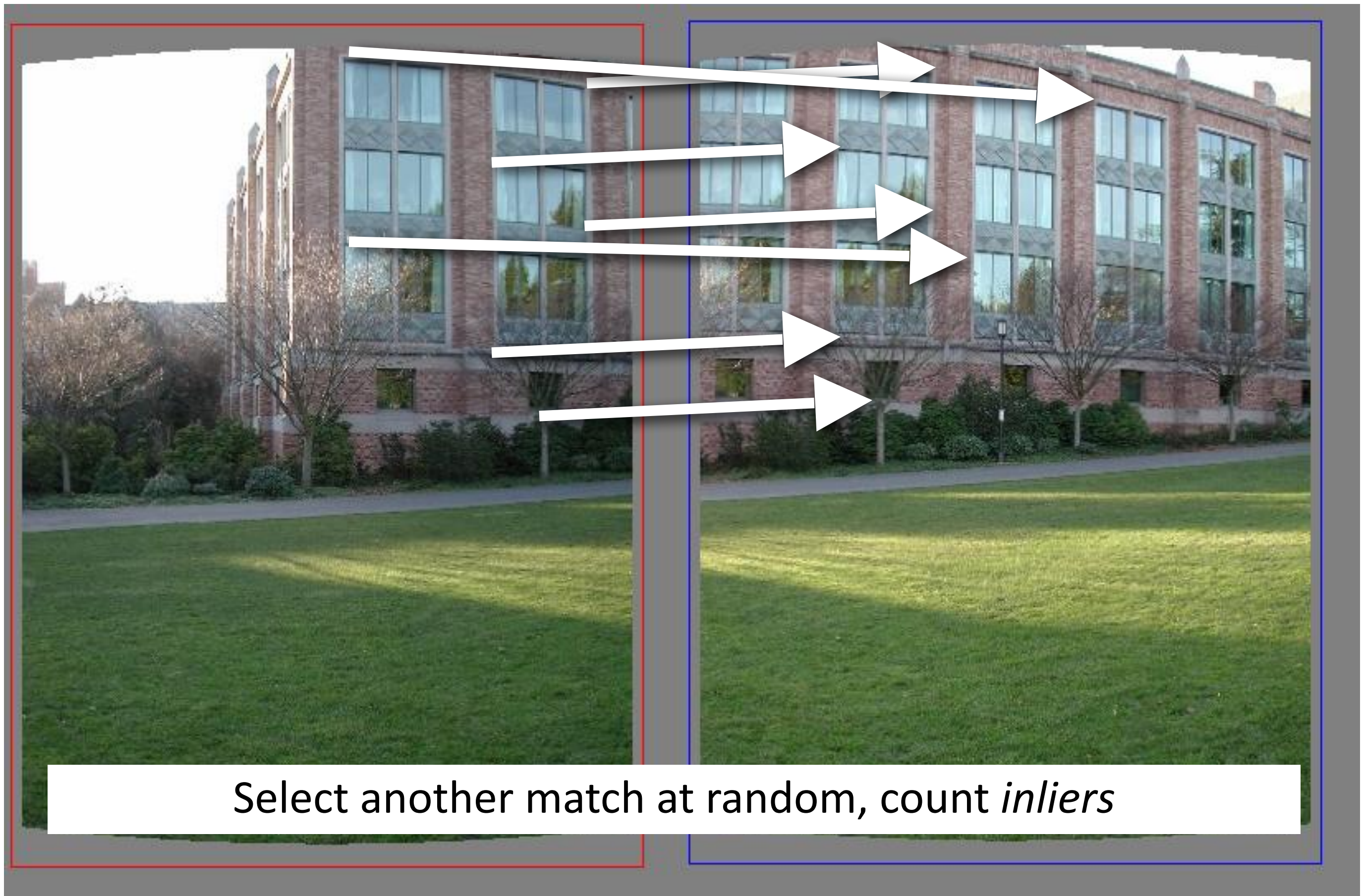
How do we combine these together?

What about the bad matches?

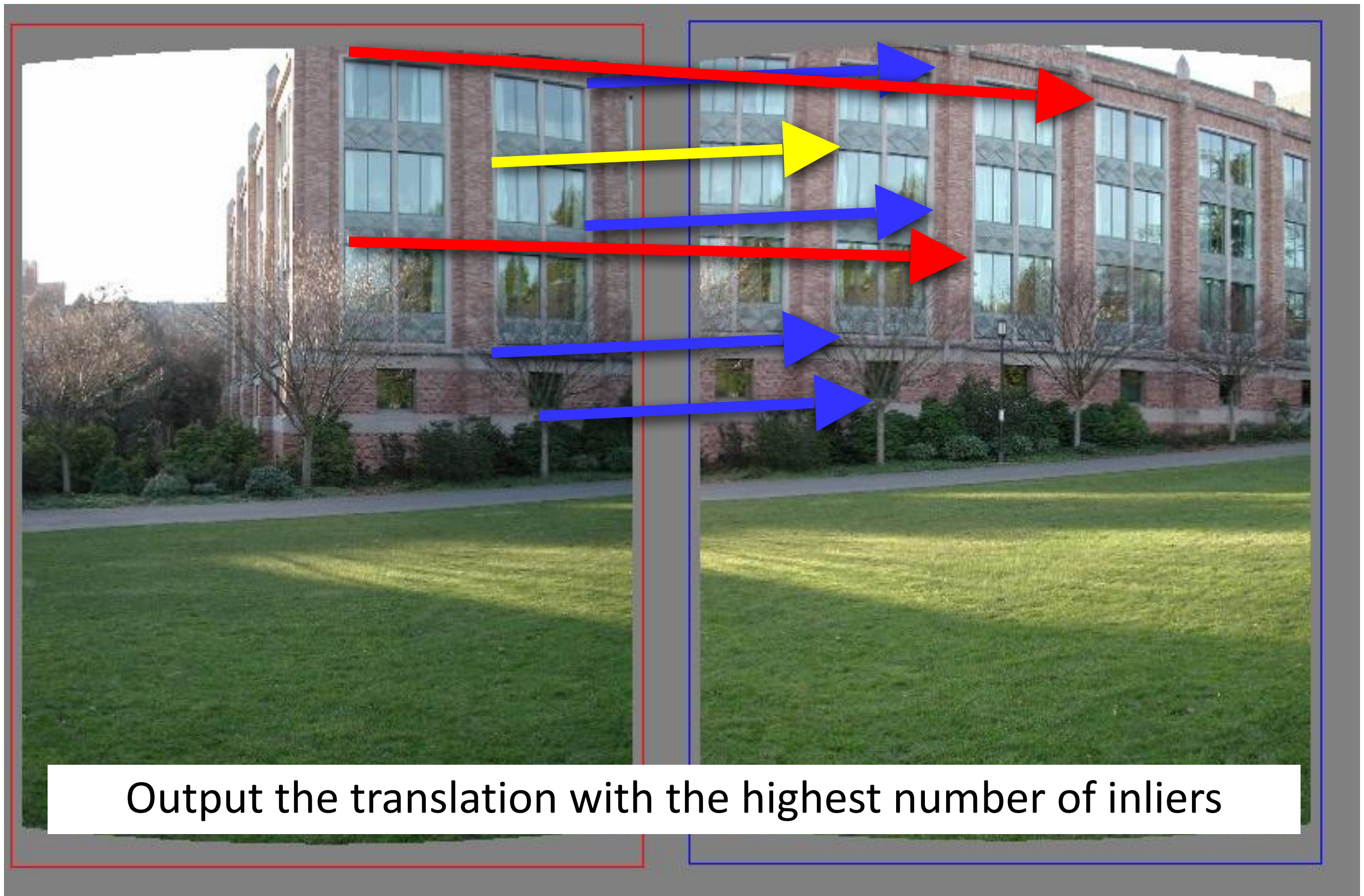
# RAndom SAmple Consensus



# RAndom SAmple Consensus



# <u>RA</u>ndom <u>SA</u>mple <u>C</u>onsensus



Output the translation with the highest number of inliers