# Lecture:

# Planar Transf. & Outlier Rejection

Several slides taken from D. Lowe and M. Irani

# Class Objectives

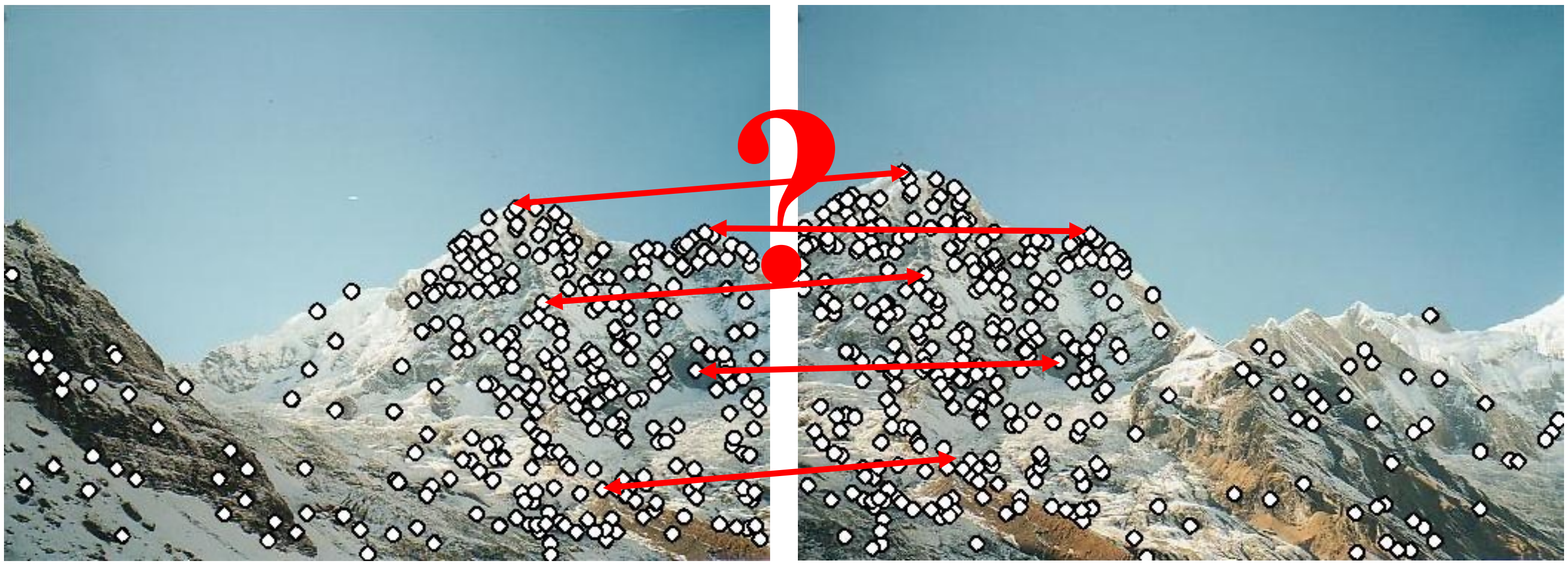
- Understand the hierarchy of different planar transformations
  - Euclidean, Similarity, Affine, Projective
- Find out how to compute an homography from a set of correspondences
- Understand the concept of Outlier
- Learn how to remove outliers with RANSAC

# Outline

- A hierarchy of transformations: Euclidean, Similarity, Affine, Projective
- Homography from a projective matrix
- Computing the homography matrix
- RANSAC

# Do you remember last week's problem?

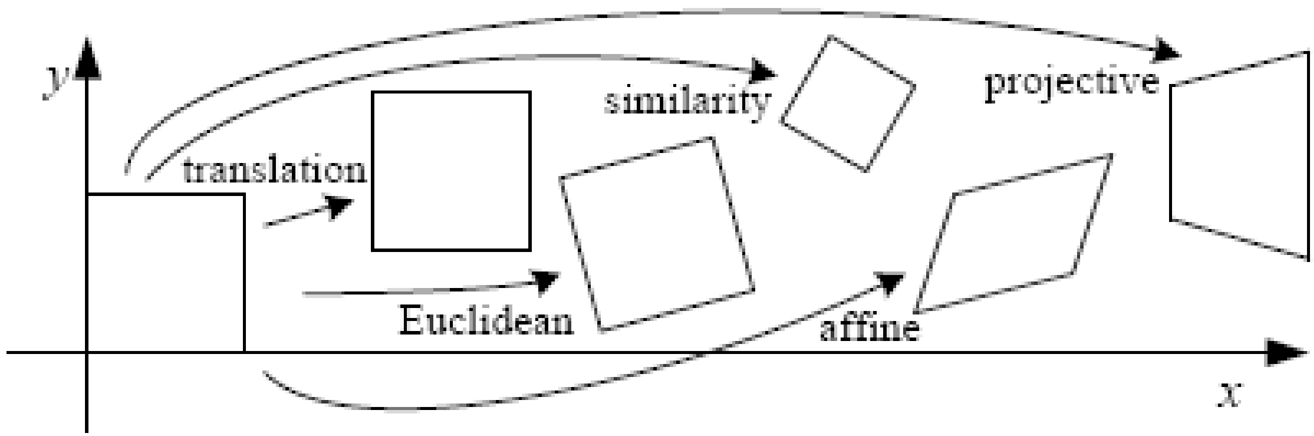
- We know how to detect points
- … and how to match them!



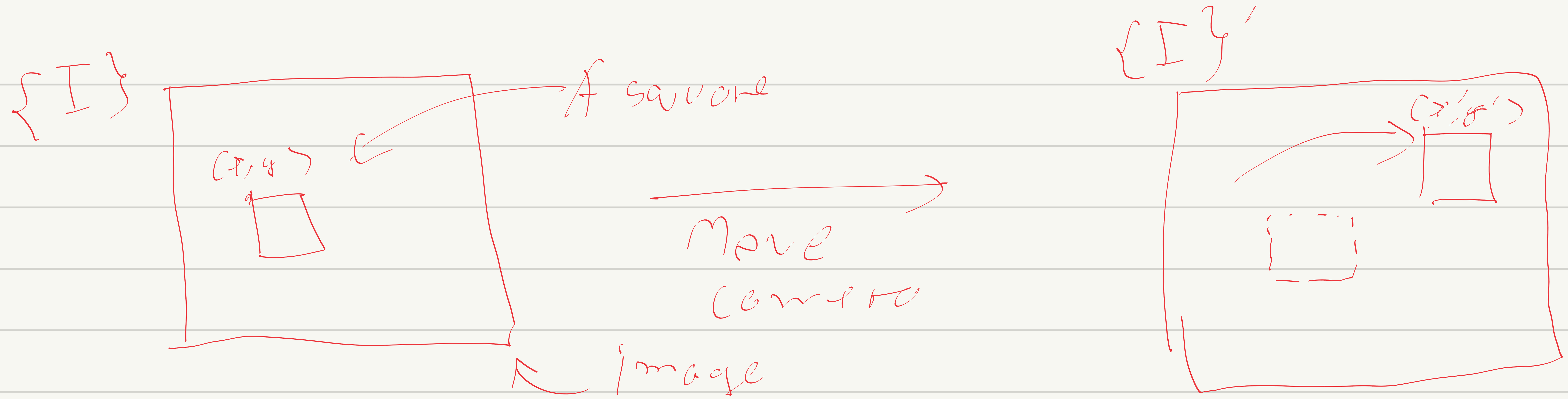

Point descriptor should be:

1. Invariant
2. Distinctive

# Summary: A hierarchy of transformations




| Name | Matrix | # D.O.F. | Preserves: | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$ | 2 | orientation + ··· | |
| rigid (Euclidean) | $\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$ | $x, y, \theta$<br>3 | lengths + ··· | |
| similarity | $\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$ | 4 | angles + ··· | |
| affine | $\begin{bmatrix} A \end{bmatrix}_{2\times 3}$ | 6 | parallelism + ··· | |
| projective | $\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$ | 8 | straight lines | |



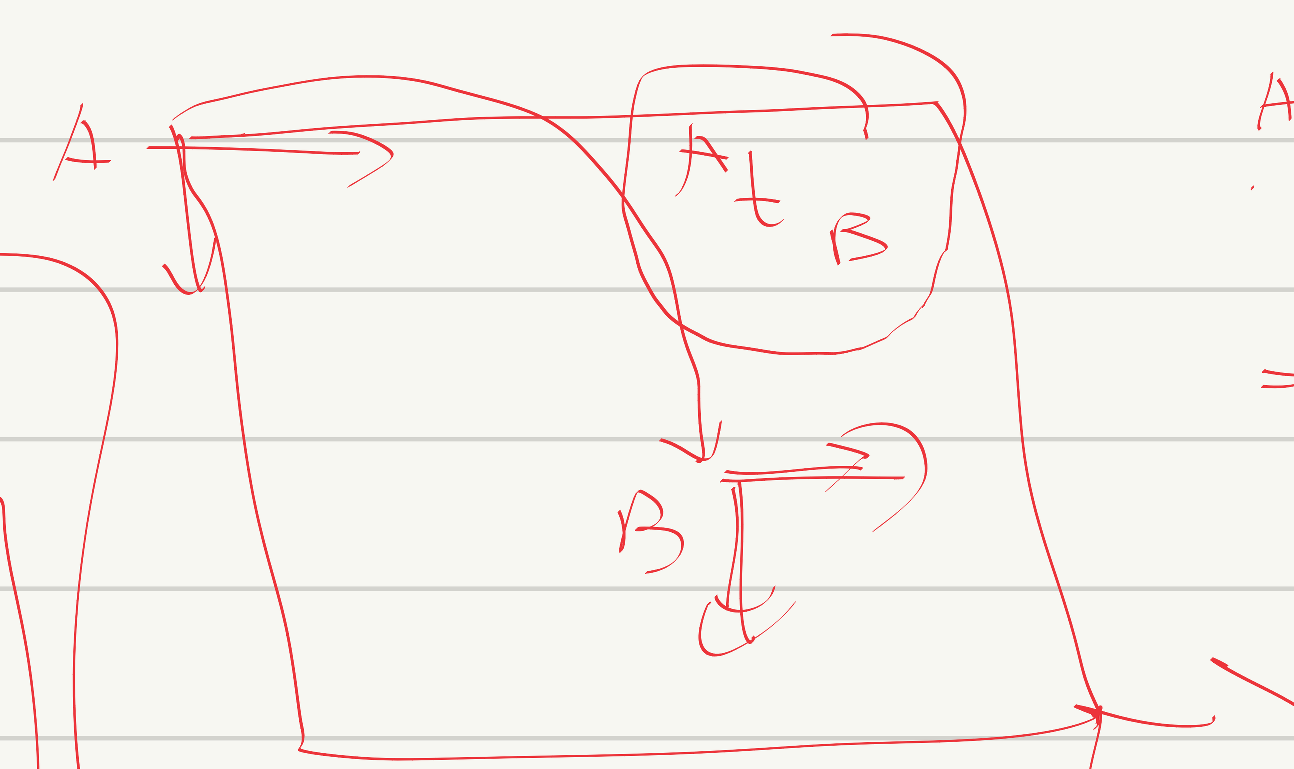

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \begin{matrix} t_x \\ t_y \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

→ we can make it more complex not just translation.

This will not be zero if there is a rotation too

This is Euclidean transformation

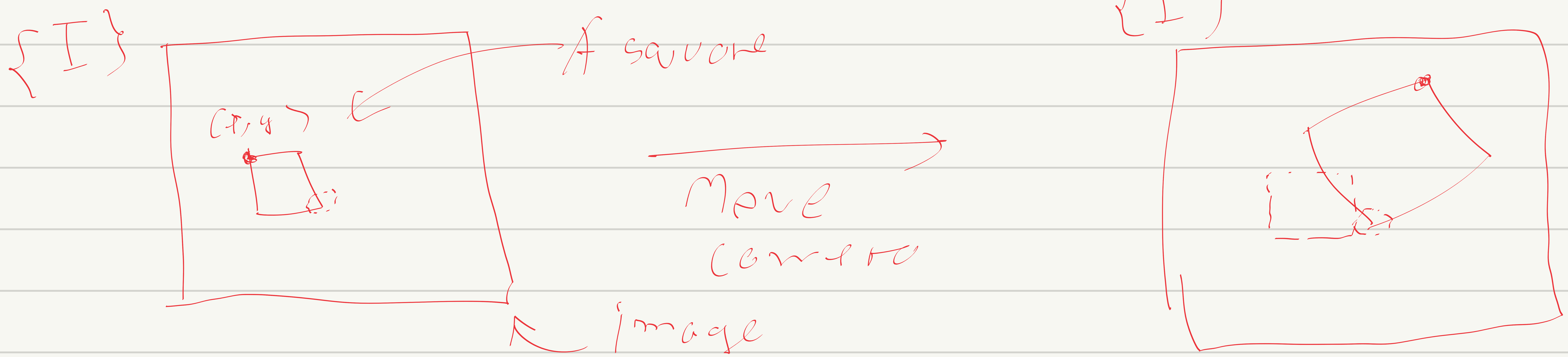
$$^{A}T_{B} = \left[ \begin{array}{c|c} 0 & ^{A}t_{B} \\ \hline 0\ 0\ 0 & \end{array} \right]$$




$^{A}p = {}^{A}T_{B}\ {}^{B}p$

$\Rightarrow {}^{B}p = {}^{B}T_{A}\ {}^{A}p$

$\Rightarrow ({}^{A}T_{B})^{-1}\ {}^{A}p$




← This is similarity transformation

DOF → 4 → $x, y, \theta, z$ ← Scale!!

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

$$S_x \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

scale is affecting this matrix

$p = (x, y, 1)^T$

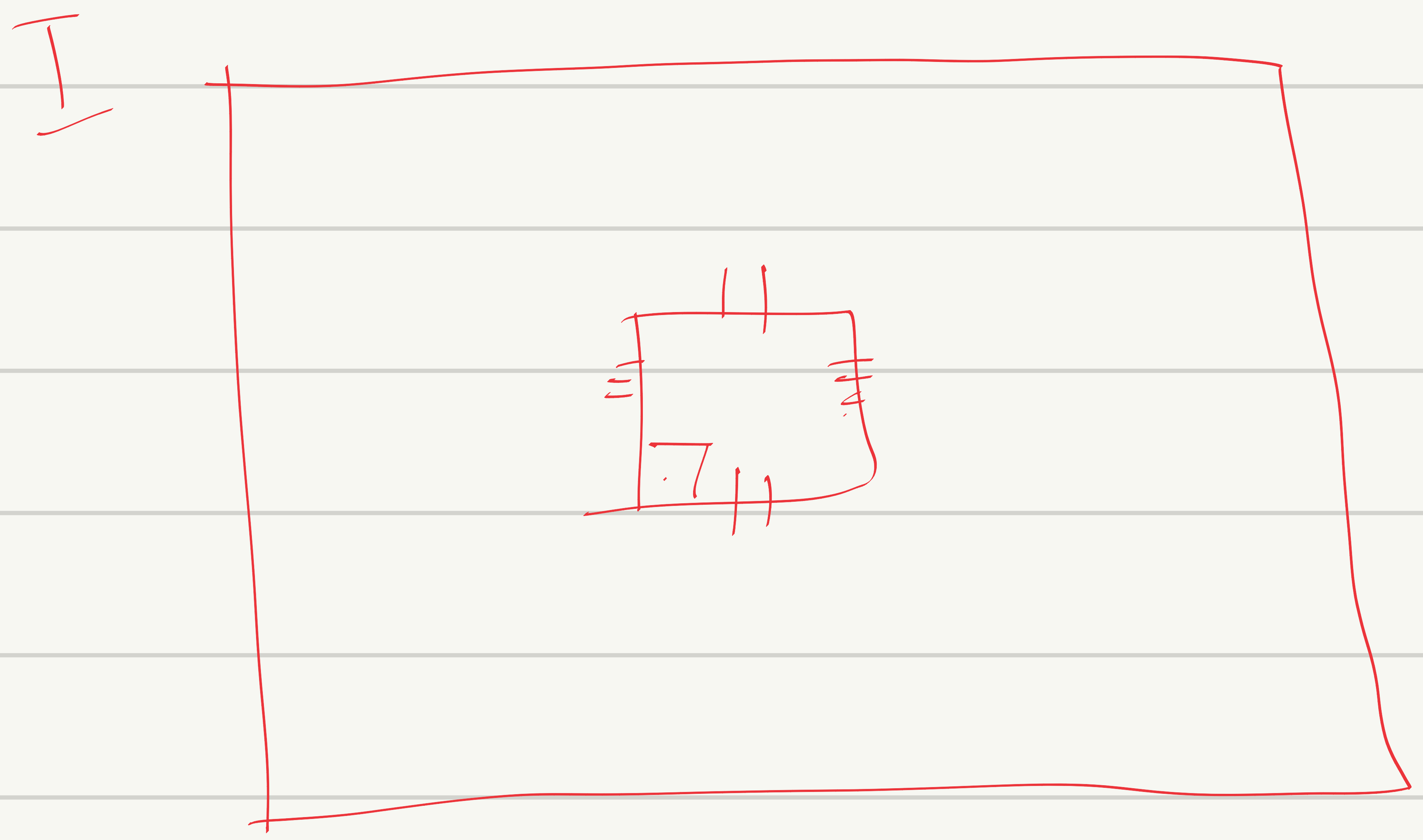
$p' = H \cdot p$

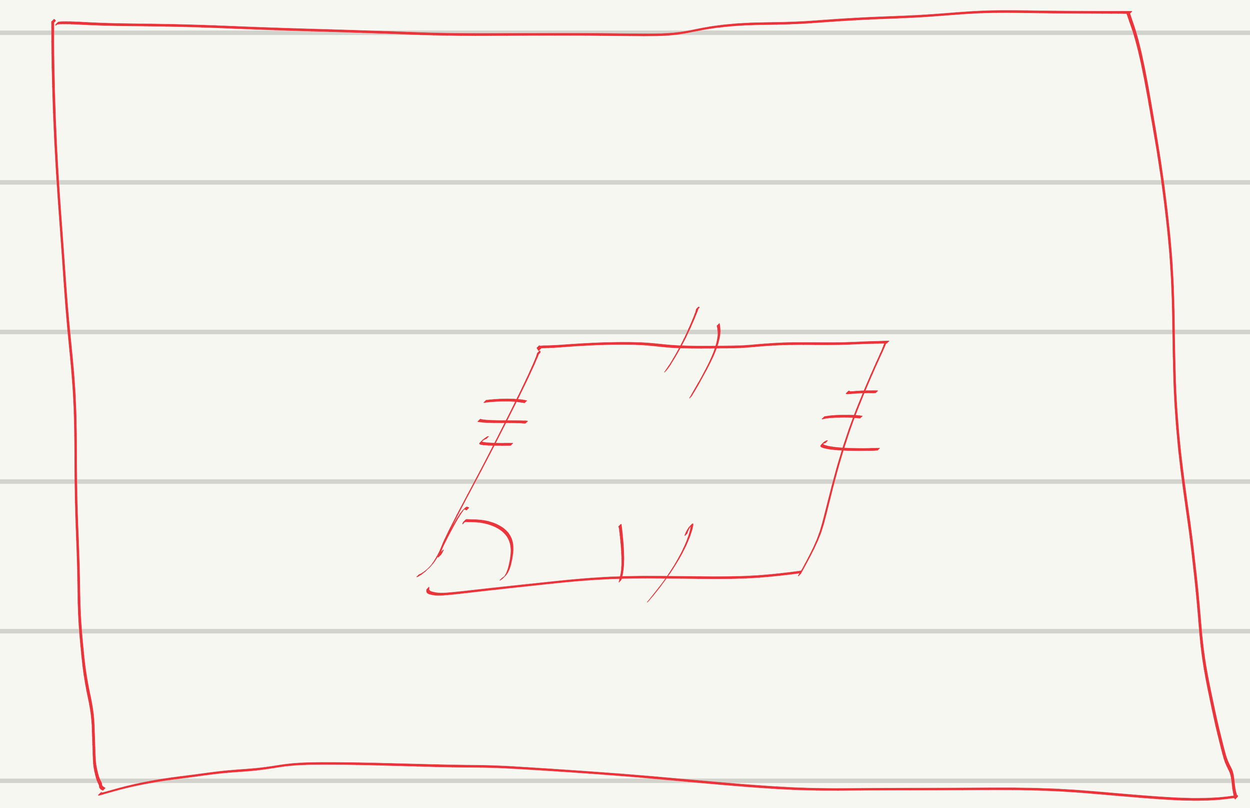
Corresponding according to the transformation

↳ then why are we using SIFT?

↳ homography is computed using the correspondences

I





camera at infinity (never going to happen?)

Affine Transformation → 6 DOF

A

translation in x & y

Rotation & something more

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

For affine.

$$A = U D V^T$$

Diagonal Matrix

Rotation Matrices $\Rightarrow V^T V = I$ & $V^{-1} = V^T$

$$A = \underbrace{U V^T}_{R(\theta)} \underbrace{V}_{R(-\alpha)} D \underbrace{V^T}_{R(\alpha)}$$

$V^T V = I$

R(θ) 4) (if there is)

R(−α) 3)

R(α) + 1 DOF 1)

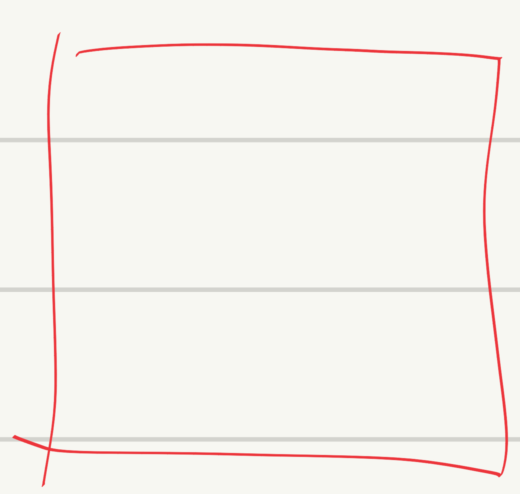
3 DOF

$$\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

2 DOF

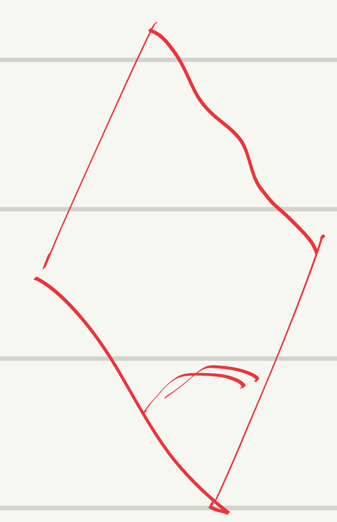
Different scale in x than in y

an isotropic scaling 2)

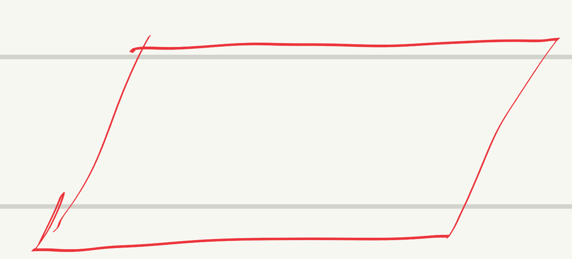


1)

2)



3)



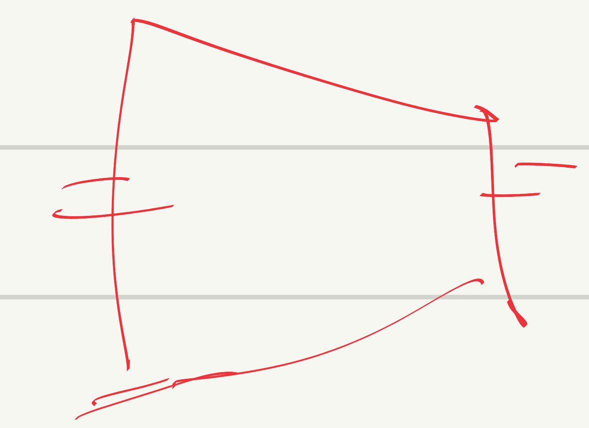
All sides equal (as its a square) just rotated

$\theta, \alpha, S_x, S_y, t_x, t_y \rightarrow 6$ DOF

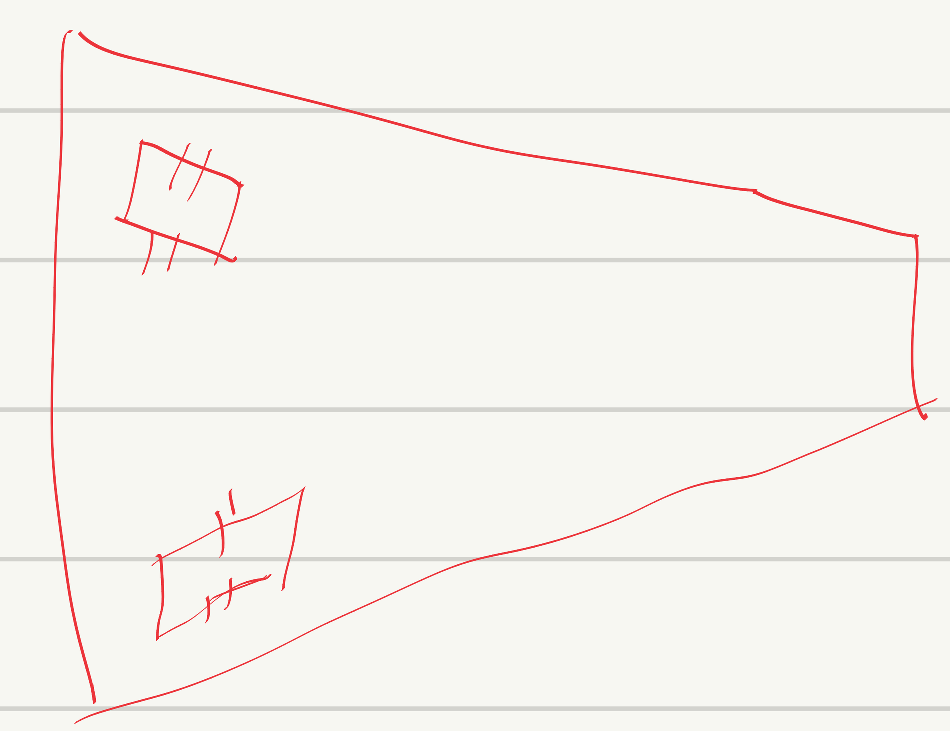
Now, if we can't place our camera at infinity, why are we talking about it?

↳ Good transformation to local patches

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



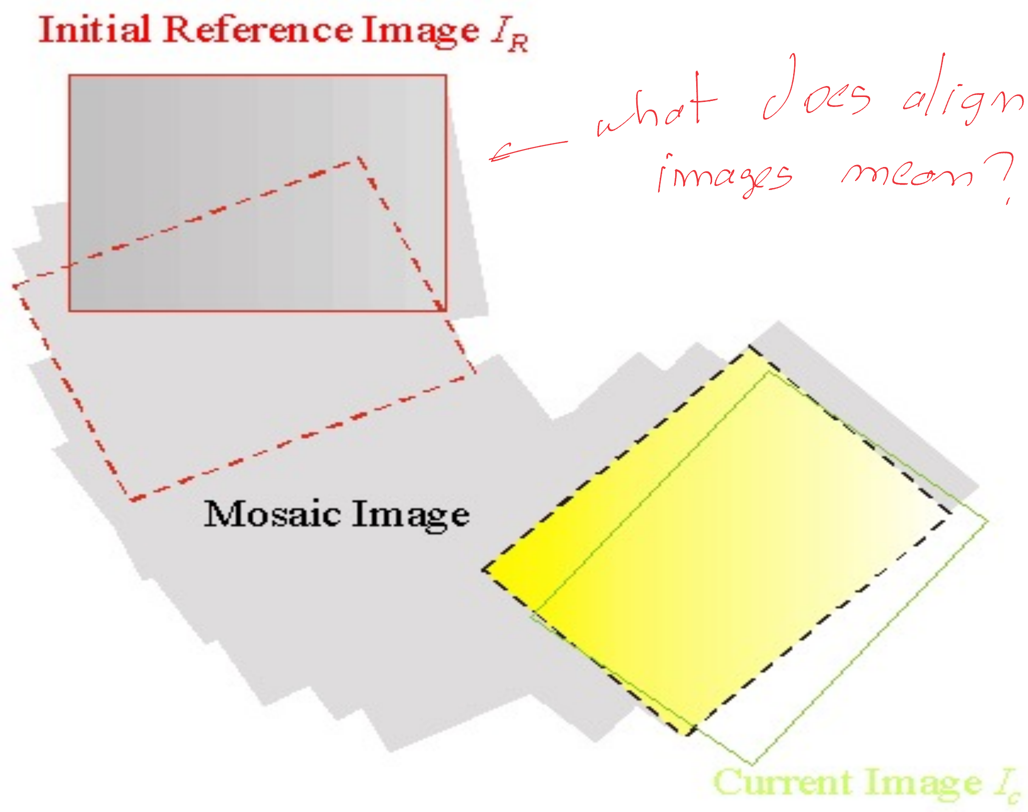
Projective



not a little bit clear how does affine transformation helps.

# Application: mosaicing

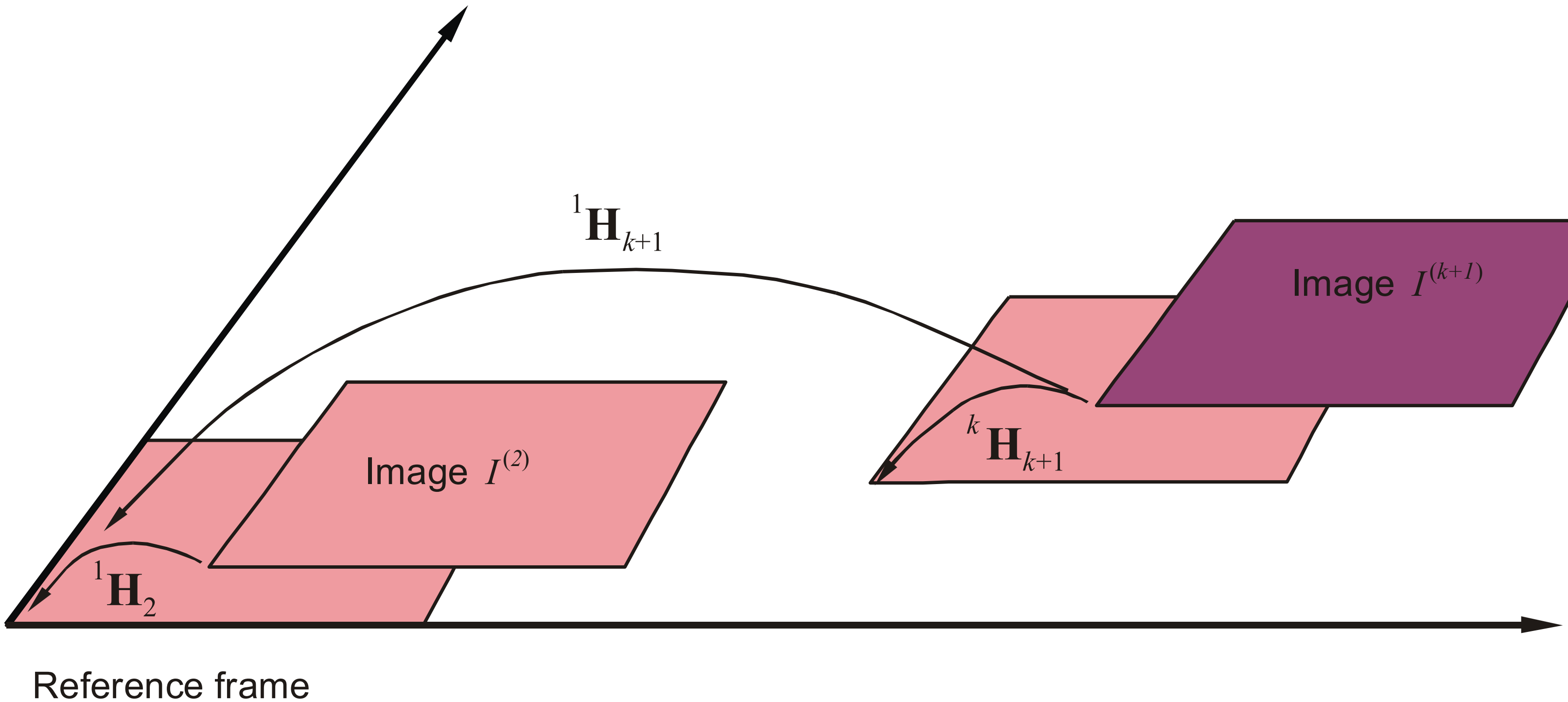
- Cascade of homographies




what does aligning images mean?

# Mosaicing

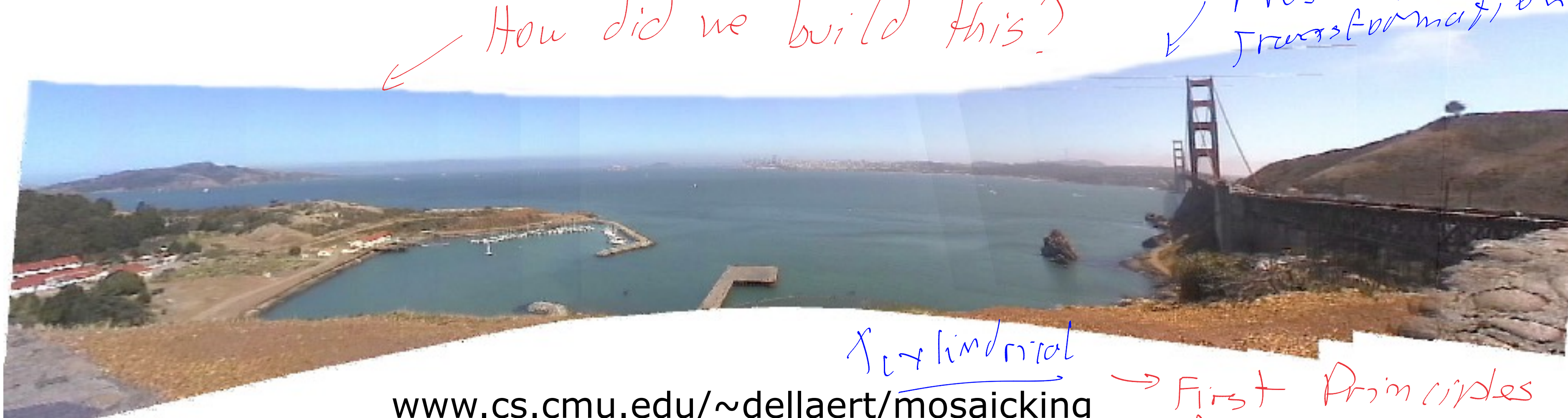
- Chain the incremental Homographies $^{1}\mathbf{H}_{k+1} = \prod_{i=1..k} {}^{i}\mathbf{H}_{i+1}$



# Mosaicking: Homography

How did we build this?

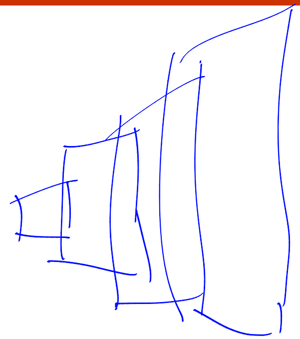
Projective Transformation.



Cylindrical

www.cs.cmu.edu/~dellaert/mosaicking

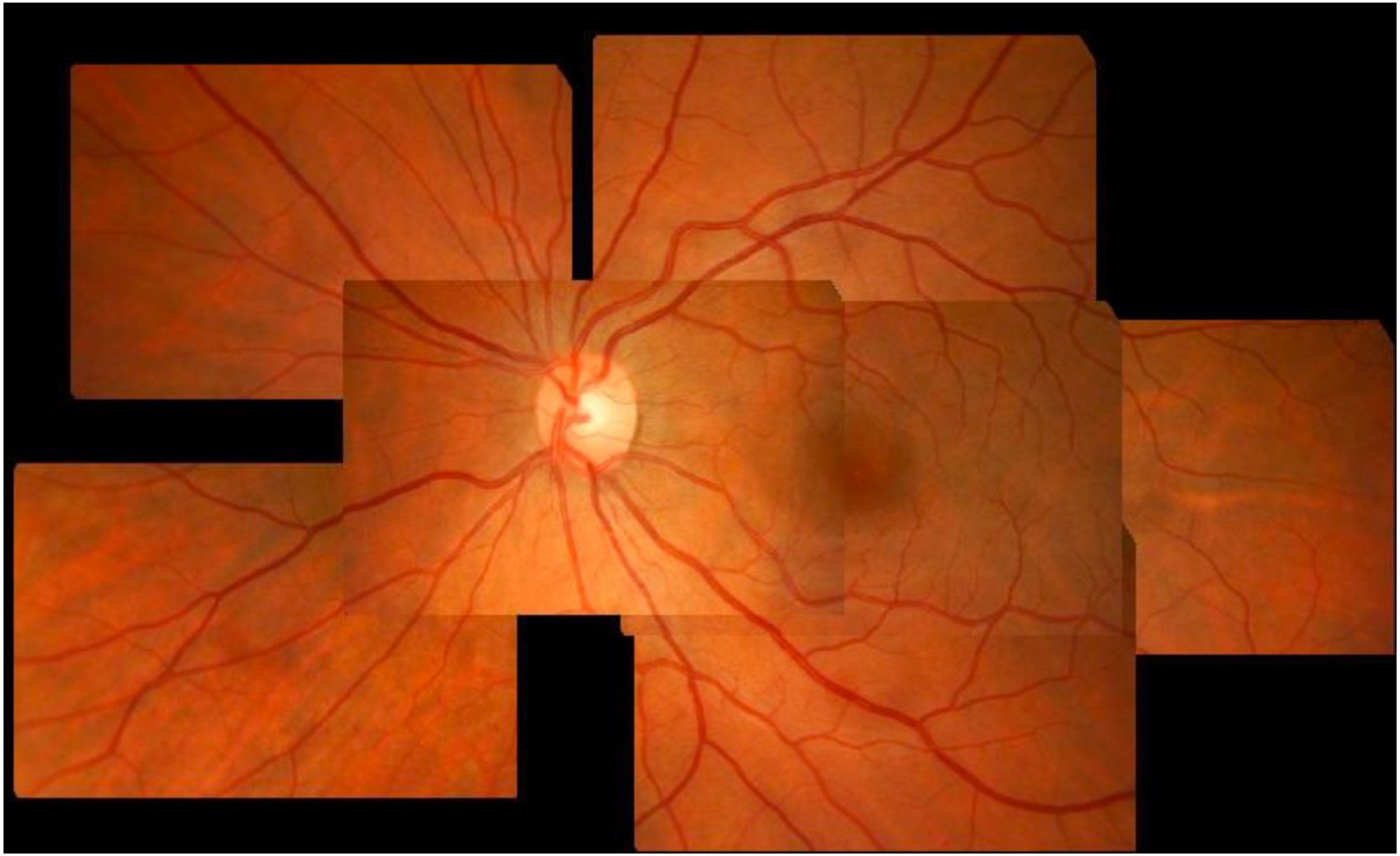
→ First Principles of Computer Vision Lecture
↳ Image Stitching.



# Building a mosaic

- *At each time, the current mosaic is represented by a graph, with images being the nodes*
- *An edge between two images indicates, that these have been stitched succesfully. The result of the pairwise stitching (a homography) is assigned to the edge.*

Mosaicing → planar scene
→ 3D scene, but camera
is just rotate..



# Global optimization

- *In the global optimisation step, we try to find parameters for each image, such that the difference between the pairwise homgraphy and the pairwise homography induced by the global parameters is minimal.*
- *The main steps of the algorithm:*
  - *Feature extraction for each image*
  - *Matching two images*
  - *Global optimization*

# Global optimization

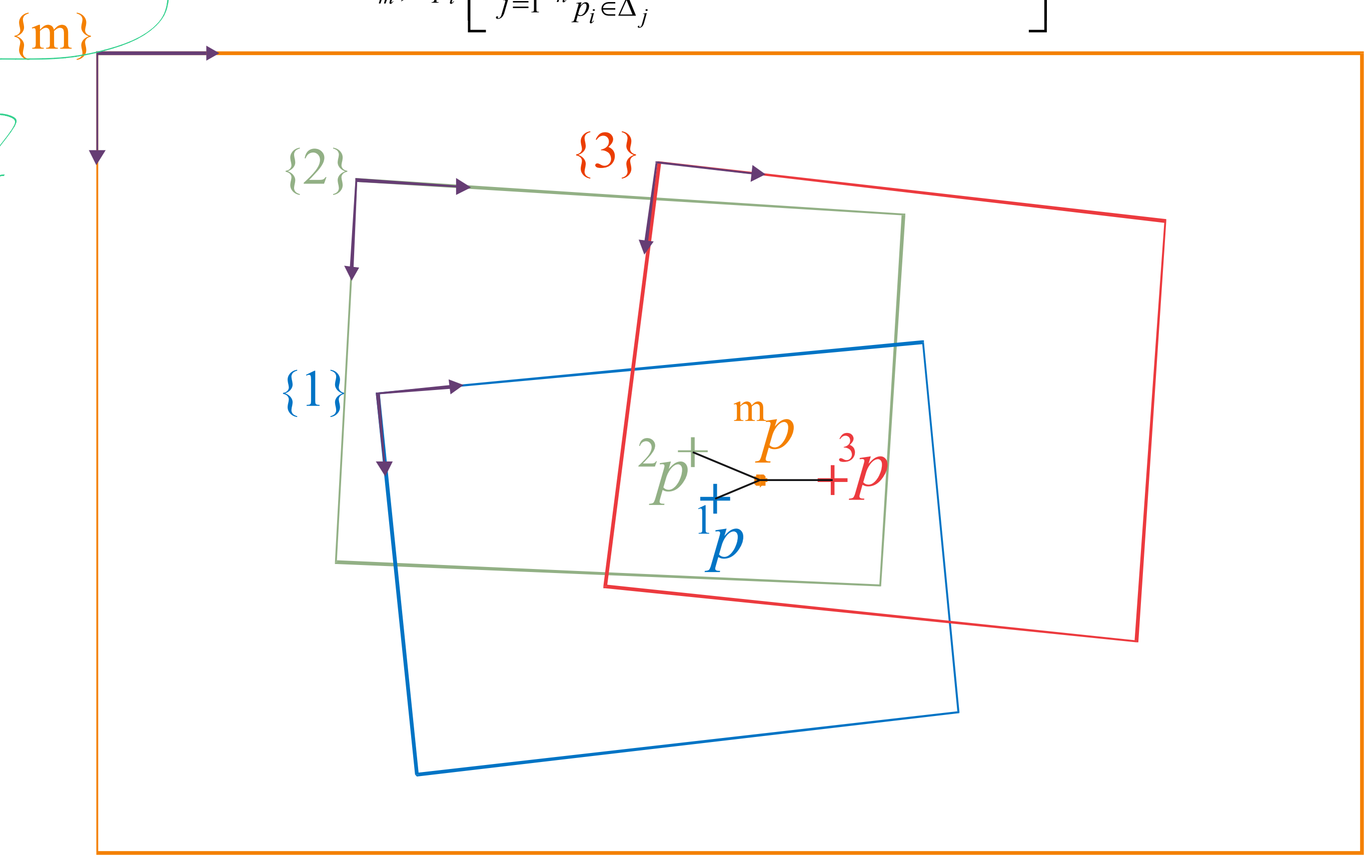
Bundle adjustment

?

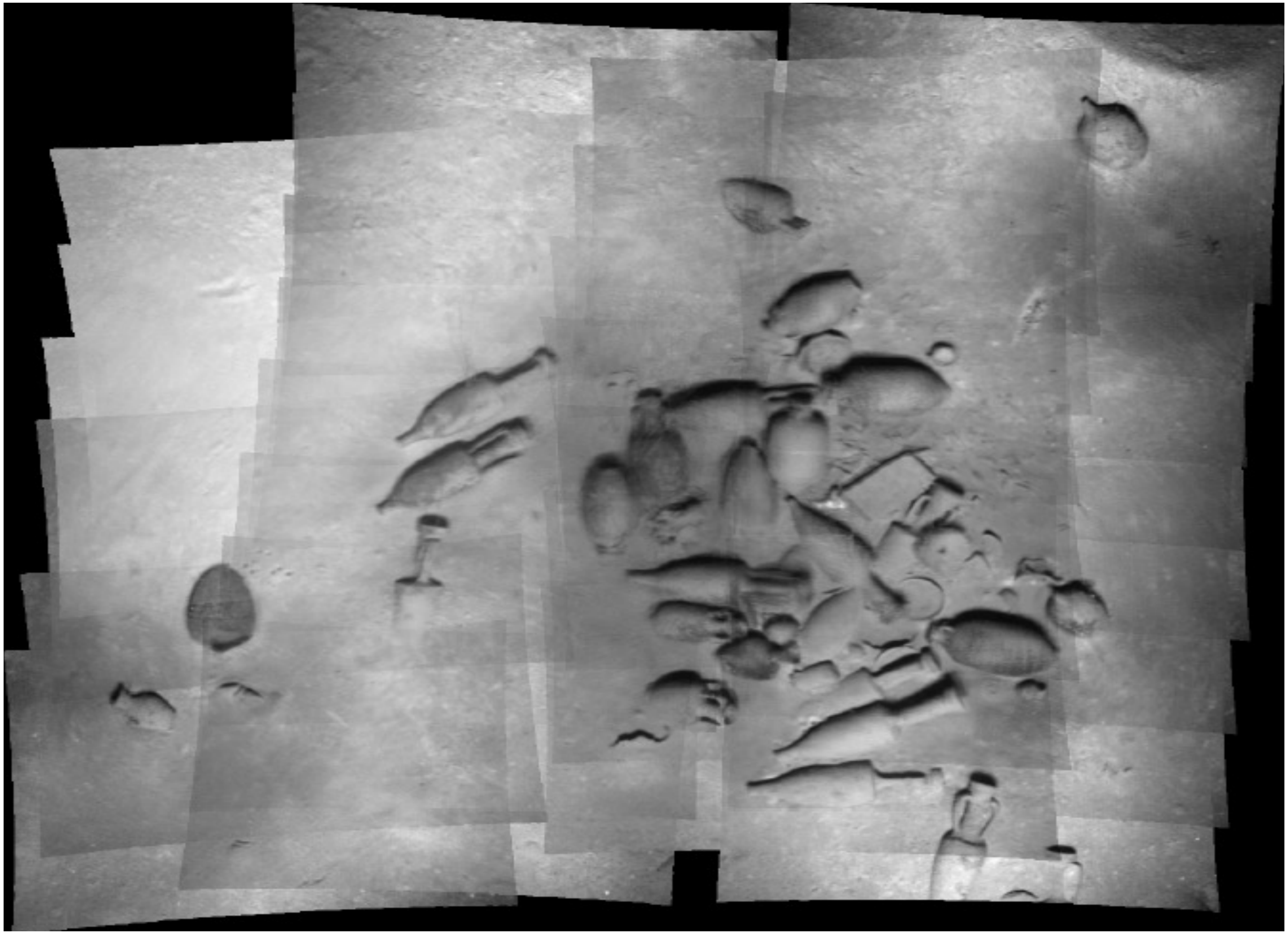
what's this?

$$c = \min_{{}^n\mathbf{H}_m,\, {}^m p_i} \left[ \sum_{j=1}^{N} \sum_{{}^n p_i \in \Delta_j} d^2\left( {}^n p_i, {}^n\mathbf{H}_m \, {}^m p_i \right) \right]$$

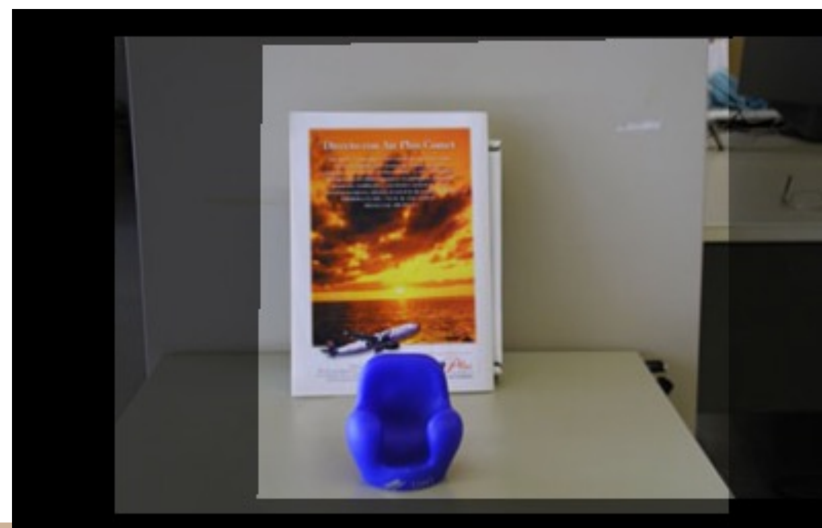
$$\left( {}^1 p, {}^2 p, {}^3 p \right) \in \Delta$$



# Global optimization



# 3D scene and a rotating camera



no parallax

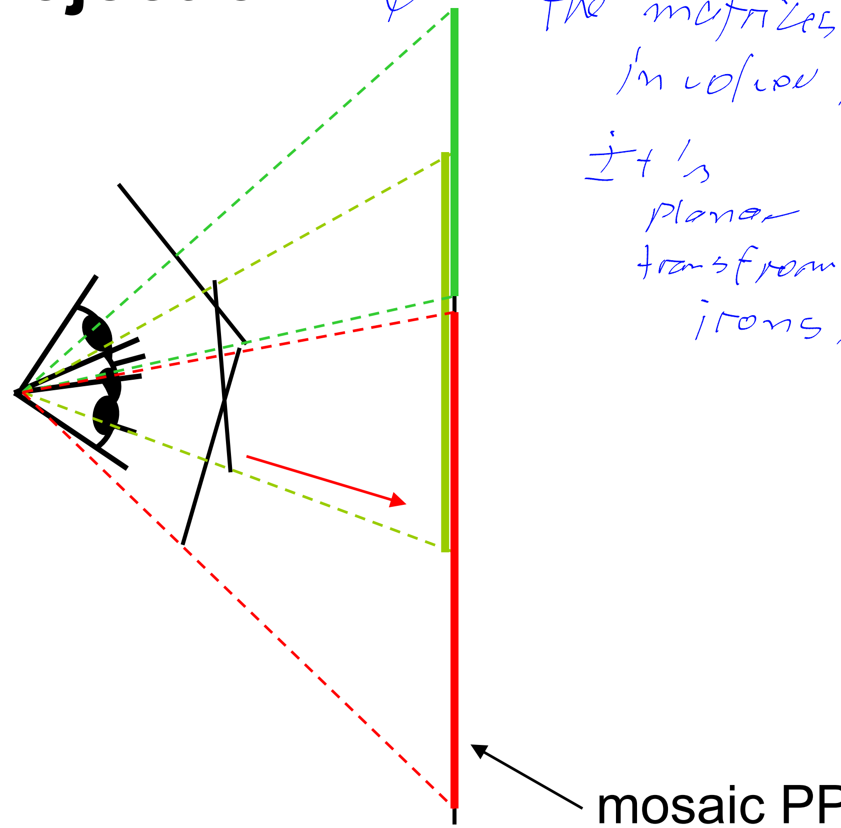
# Parallax in a translating camera

# Image reprojection

what are the matrices involved?

It's planar transfrom irons?

- The mosaic has a natural interpretation in 3D
  - The images are reprojected onto a common plane
  - The mosaic is formed on this plane
- Only valid in **absence** of translation



# Application: video stabilization

what do we do with the points? H

Find correspondences to put them back on original so that original points didn't move.

Reference image would be changing at a frequency

# Application: video stabilization

# Application: video stabilization

# Application: video stabilization

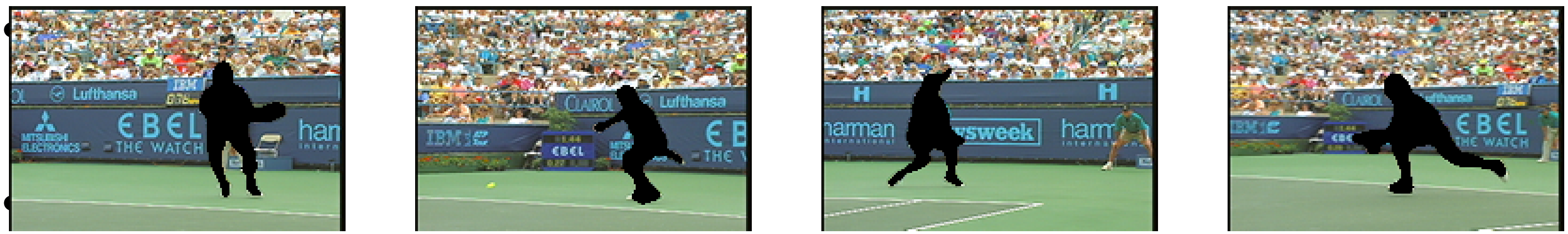
# Application: video stabilization

Universitat de Girona



# Mosaics for Video Coding

- Convert masked images into a background sprite for content-based coding



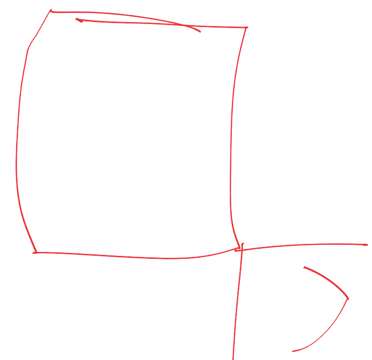

=


→ what do we do here?

# Computing the homography matrix

similarity transform

$$\begin{bmatrix} {}^{r}x_i \\ {}^{r}y_i \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_i \\ {}^{c}y_i \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} {}^{r}x_i \\ {}^{r}y_i \\ 1 \end{bmatrix} = \begin{bmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_i \\ {}^{c}y_i \\ 1 \end{bmatrix}$$

x, y

x

y

3×3

<u>Lecture Activity</u>: solve for $a$, $b$, $c$ and $d$

# 8 DOF (Projective) Case

$$\begin{bmatrix} kx_i \\ ky_i \\ k \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}$$

$$x_i = \frac{h_{11}x_i' + h_{12}y_i' + h_{13}}{h_{31}x_i' + h_{32}y_i' + 1}$$

$$y_i = \frac{h_{21}x_i' + h_{22}y_i' + h_{23}}{h_{31}x_i' + h_{32}y_i' + 1}$$

# 8 DOF (Projective) Case

$$x_i = \frac{h_{11}x_i' + h_{12}y_i' + h_{13}}{h_{31}x_i' + h_{32}y_i' + 1}$$

$$y_i = \frac{h_{21}x_i' + h_{22}y_i' + h_{23}}{h_{31}x_i' + h_{32}y_i' + 1}$$

$$\left.\begin{aligned}\left(h_{31}x_i' + h_{32}y_i' + 1\right)x_i &= h_{11}x_i' + h_{12}y_i' + h_{13}\\ \left(h_{31}x_i' + h_{32}y_i' + 1\right)y_i &= h_{21}x_i' + h_{22}y_i' + h_{23}\end{aligned}\right\}$$

$$\left.\begin{aligned}x_i &= h_{11}x_i' + h_{12}y_i' + h_{13} - h_{31}x_i x_i' - h_{32}x_i y_i'\\ y_i &= h_{21}x_i' + h_{22}y_i' + h_{23} - h_{31}y_i x_i' - h_{32}y_i y_i'\end{aligned}\right\}$$

# 8 DOF (Projective) Case

$$\left.\begin{aligned} x_i &= h_{11}x_i' + h_{12}y_i' + h_{13} - h_{31}x_i x_i' - h_{32}x_i y_i' \\ y_i &= h_{21}x_i' + h_{22}y_i' + h_{23} - h_{31}y_i x_i' - h_{32}y_i y_i' \end{aligned}\right\}$$

$$\begin{bmatrix} x_1' & y_1' & 1 & 0 & 0 & 0 & -x_1 \cdot x_1' & -x_1 \cdot y_1' \\ 0 & 0 & 0 & x_1' & y_1' & 1 & -y_1 \cdot x_1' & -y_1 \cdot y_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n' & y_n' & 1 & 0 & 0 & 0 & -x_n \cdot x_n' & -x_n \cdot y_n' \\ 0 & 0 & 1 & x_n' & y_n' & 1 & -y_n \cdot x_n' & -y_n \cdot y_n' \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix}$$

# And if some correspondences are wrong?: RANSAC Algorithm

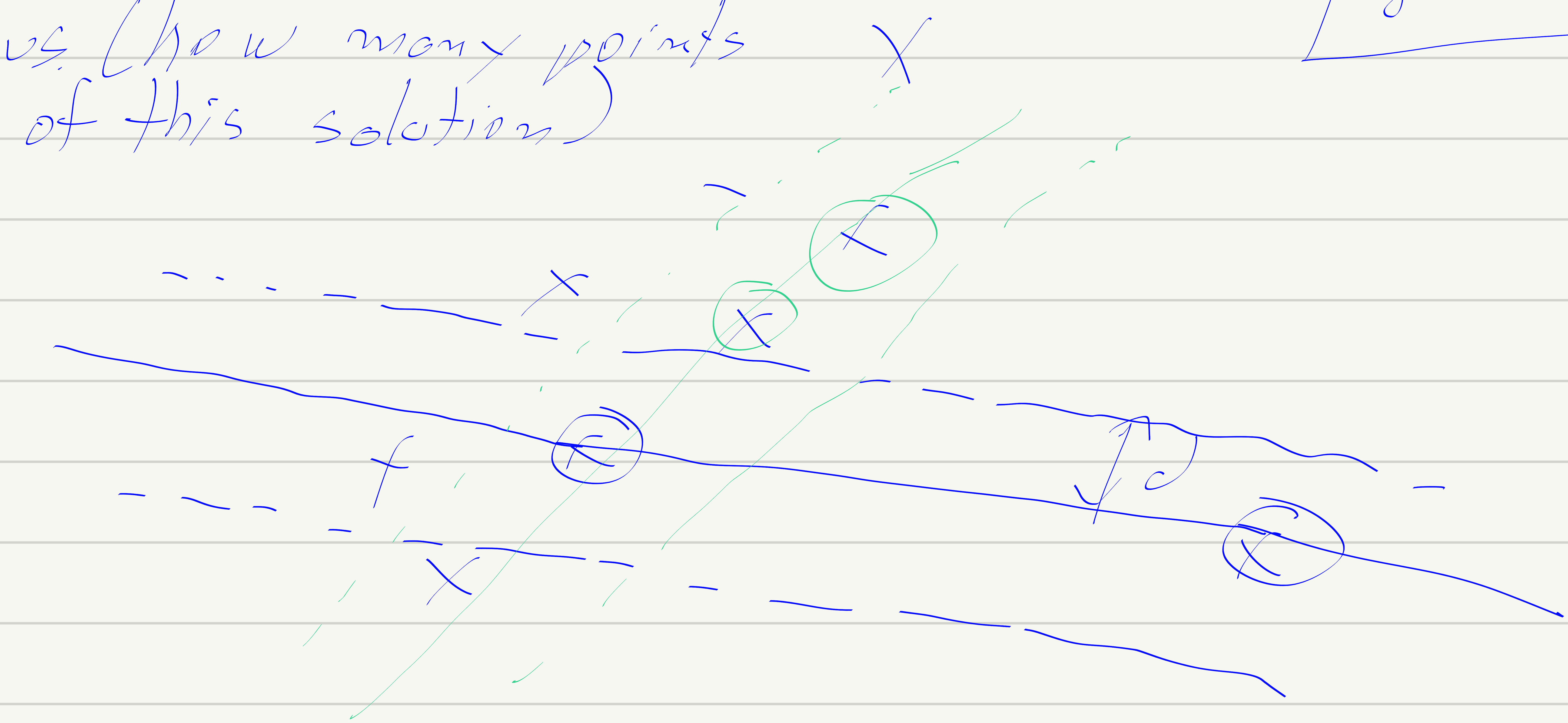
- Stands for "Random Sampling Consensus"
- Used for fitting a model to data with outliers
- Algorithm:
  - Many sets random samples are chosen from data and a model is calculated
  - 'Fitness' of the model is calculated using the entire data
  - The 'best' model and the corresponding samples are removed from the data and algorithm is run again
  - Algorithm continues until not enough samples fit any model

1) Randomly select two points for a line.
2) Within a given bound about your solution (d)
count the consensus (how many points
are within the bounds of this solution)

$y = ax + b$ ← model. ← $a, b$ ?

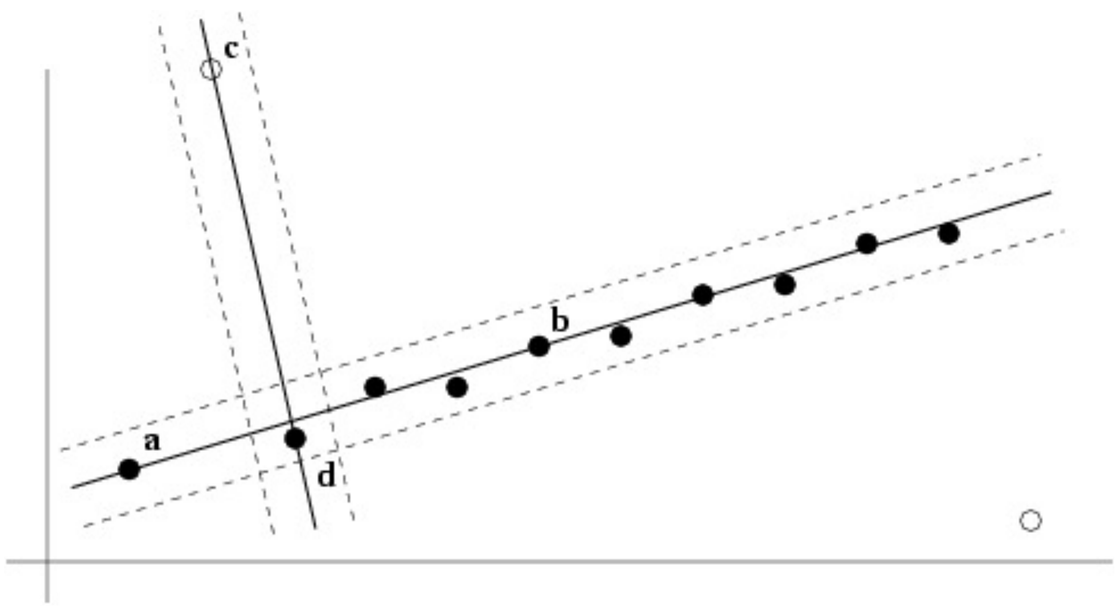
Sol 1) consensus
(c) = 3

Sol 2) c = 4



# Robust estimation

- What if set of matches contains gross outliers?
  (to keep things simple let's consider line fitting first)



For Euclidean homography

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

→ each correspondences gives 2 pair of equations
↳ : 4 unknowns
↳ 2 points required

# RANSAC

## Objective

Robust fit of model to data set S which contains outliers

## Algorithm

↳ what is the consensus for homography?

(i) Randomly select a sample of *s* data points from S and instantiate the model from this subset.

(ii) Determine the set of data points $S_i$ which are within a distance threshold *t* of the model. The set $S_i$ is the consensus set of samples and defines the inliers of S.

(iii) If the subset of $S_i$ is greater than some threshold *T*, re-estimate the model using all the points in $S_i$ and terminate

(iv) If the size of $S_i$ is less than *T*, select a new subset and repeat the above.

(v) After *N* trials the largest consensus set $S_i$ is selected, and the model is re-estimated using all the points in the subset $S_i$

Universitat de Girona

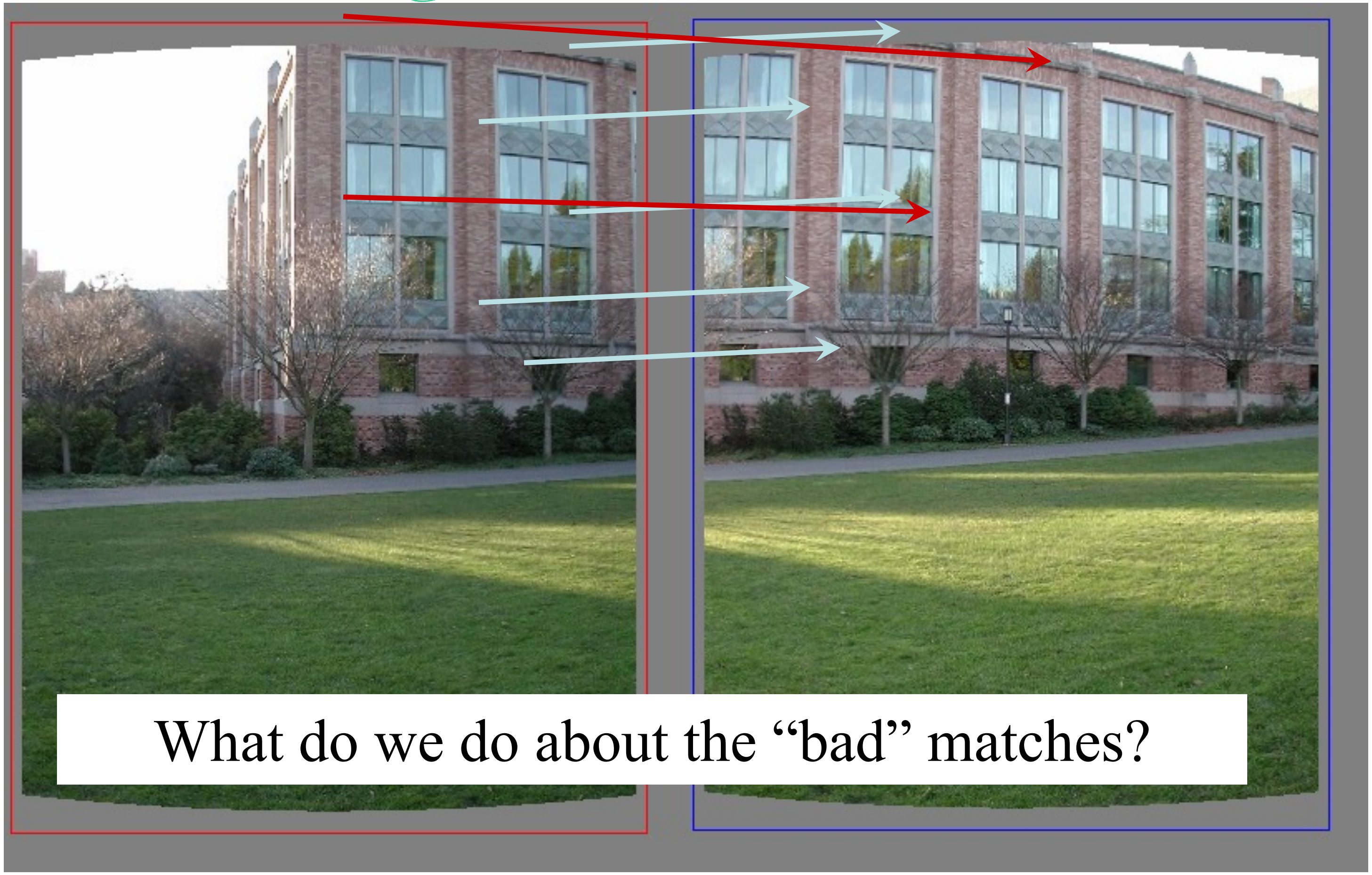


→ for all SIFT correspondences?

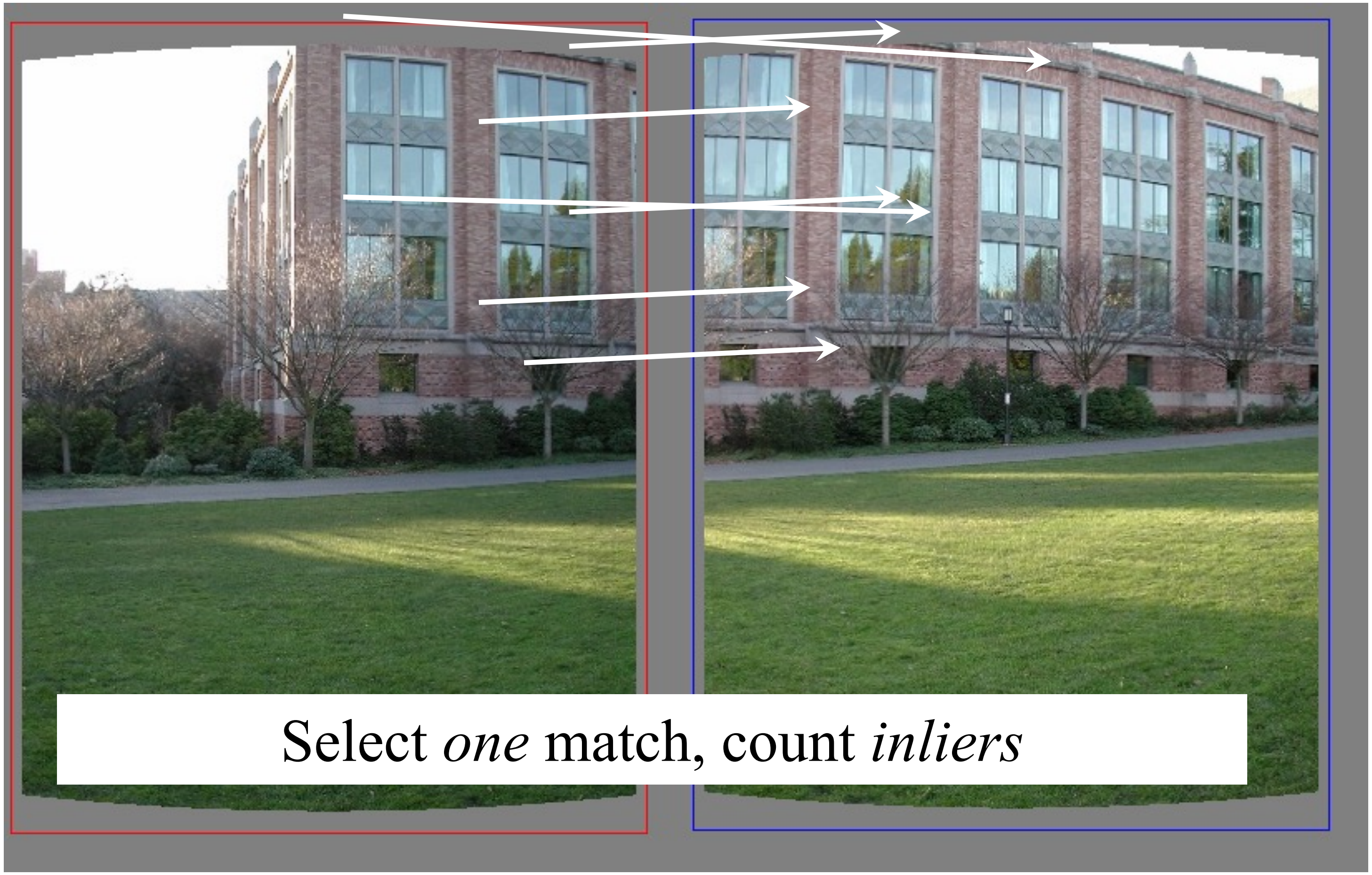
physical meaning?

# Matching features

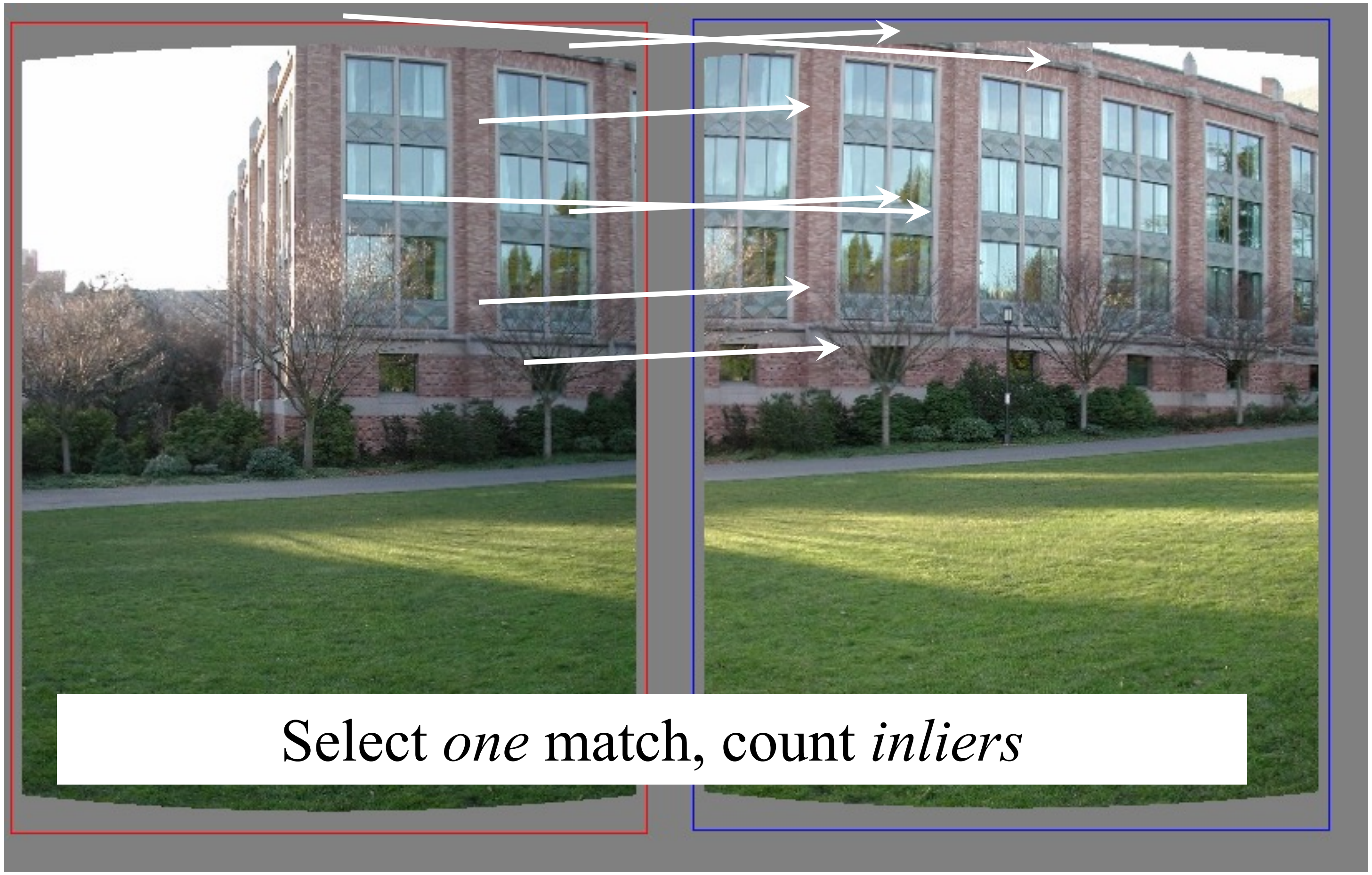
what is the meaning of threshold?



# RAndom SAmple Consensus



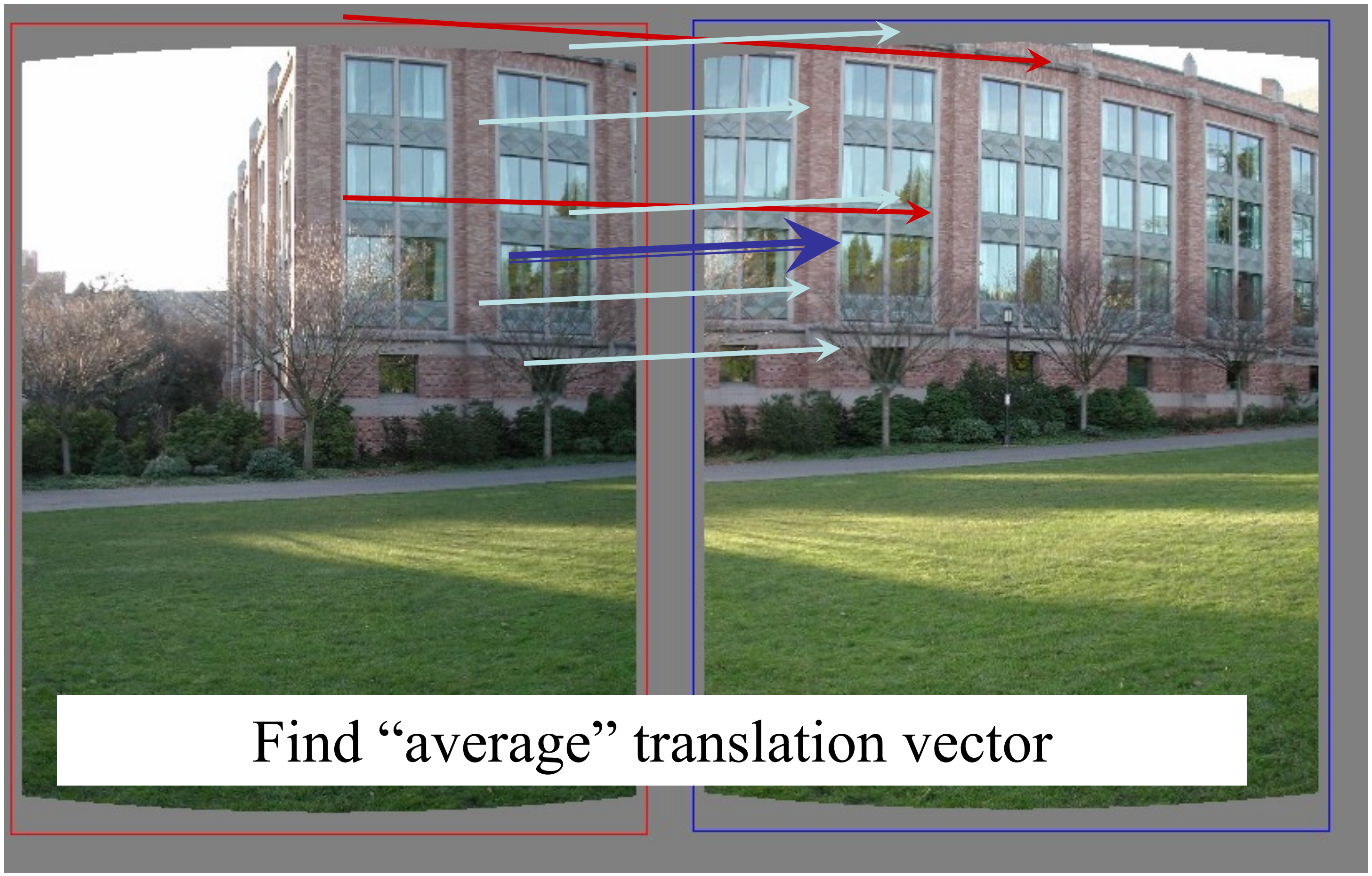
# RAndom SAmple Consensus



what is SIFT used for
what is RANSAC used for



# Least squares fit



# Distance threshold

Choose $t$ so probability for inlier is α (e.g. 0.95)

- Often empirically
- Zero-mean Gaussian noise σ then $d_{\perp}^{2}$ follows a $\chi_{m}^{2}$ distribution with $m$ DOFs ($m$ = codimension of model)

*(dimension+codimension=dimension space)*

| Codimension | Model | $t^2$ |
|---|---|---|
| 1 | line,F | $3.84\sigma^2$ |
| 2 | H,P | $5.99\sigma^2$ |
| 3 | T | $7.81\sigma^2$ |

# How many samples?

- Choose $N$ so that, with probability $p$, at least one random sample is free from outliers. e.g. $p$=0.99

Each point is a correspondence. How do we know that correspondence is an outlier?

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

Probability to be an outlier

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

Amount of data to instantiate homography

10% outliers

| | 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 |

# Acceptable consensus set?

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)\ n$$

## Adaptively determining the number of samples

$e$ is often unknown a priori, so pick worst case, i.e. 0, and adapt if more inliers are found, e.g. 80% would yield $e$=0.2

- *N*=∞, *sample_count* =0
- While *N* >*sample_count* repeat
    - Choose a sample and count the number of inliers
    - Set e=1-(number of inliers)/(total number of points)
    - Recompute *N* from *e* $\left(N = \log(1-p)/\log\left(1-(1-e)^s\right)\right)$
    - Increment the *sample_count* by 1
- Terminate

## Summary RANSAC – the RANdom SAmple Consensus

- Randomly pick up enough data points (sample) for estimation.
- Count the number of the supporting points for each estimation
- Get the best estimation which have the most supports
- Use all the supporting points to get the best estimation

# Summary

We have seen today:

- a hierarchy of transformations: Euclidean, Similarity, Affine, Projective
- What is a Homography
- How to compute a homography matrix using the adequate (motion) model
- Applications of planar transformations
- How to detect outliers

# References

- RANSAC:
  - M. Fischler and R. Bolles. Random sampling consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of ACM*, vol. 24, no.6, pp. 381–395, 1981.