Lecture:

Planar Transf. & Outlier Rejection

Several slides taken from D. Lowe and M. Irani

Class Objectives

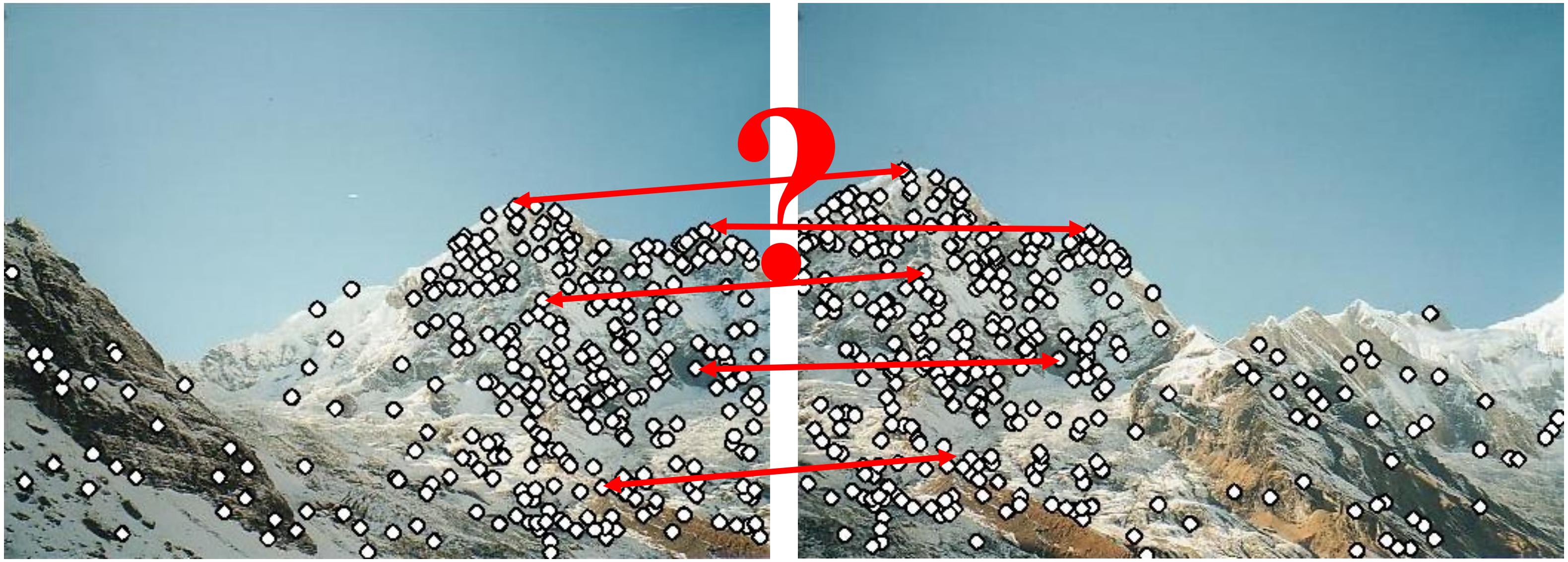
- Understand the hierarchy of different planar transformations
  - Euclidean, Similarity, Affine, Projective
- Find out how to compute an homography from a set of correspondences
- Understand the concept of Outlier
- Learn how to remove outliers with RANSAC

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?

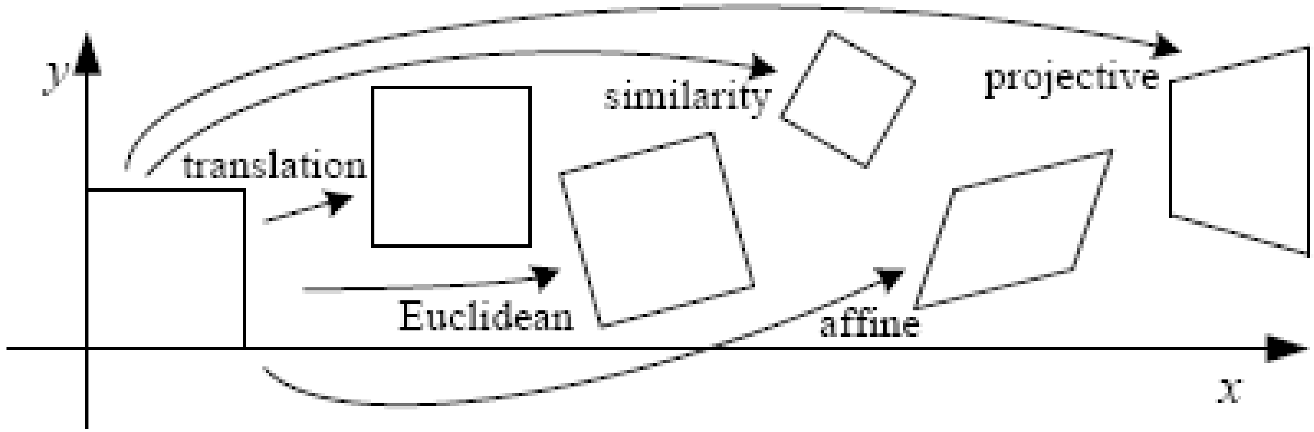
- We know how to detect points
- … and how to match them!



Point descriptor should be:

1. Invariant
2. Distinctive

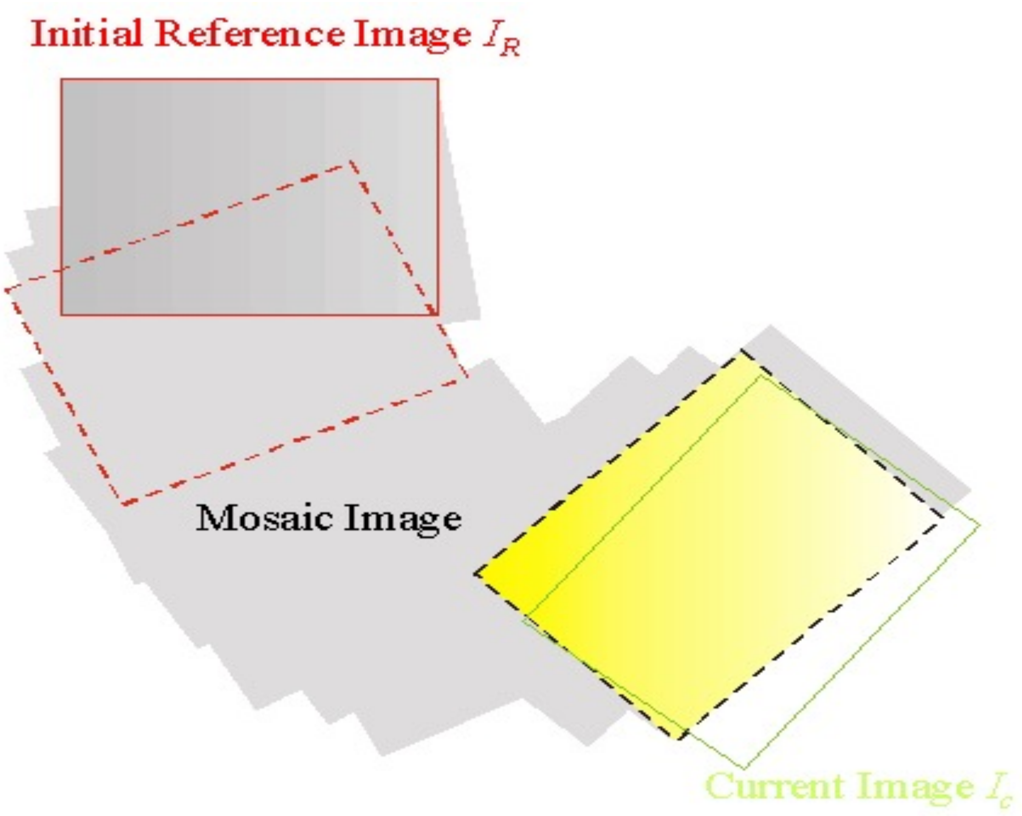
Summary: A hierarchy of transformations



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

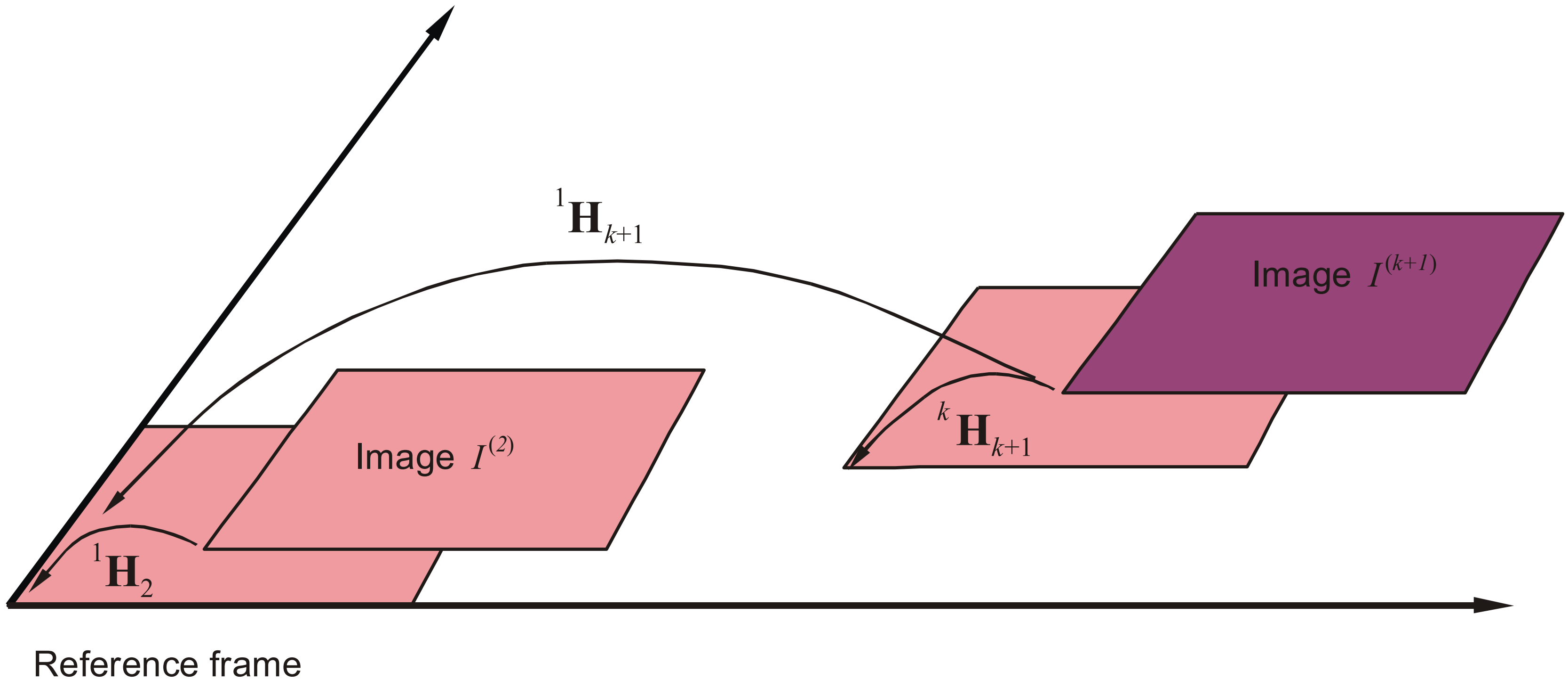
Application: mosaicing

- Cascade of homographies



Mosaicing

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

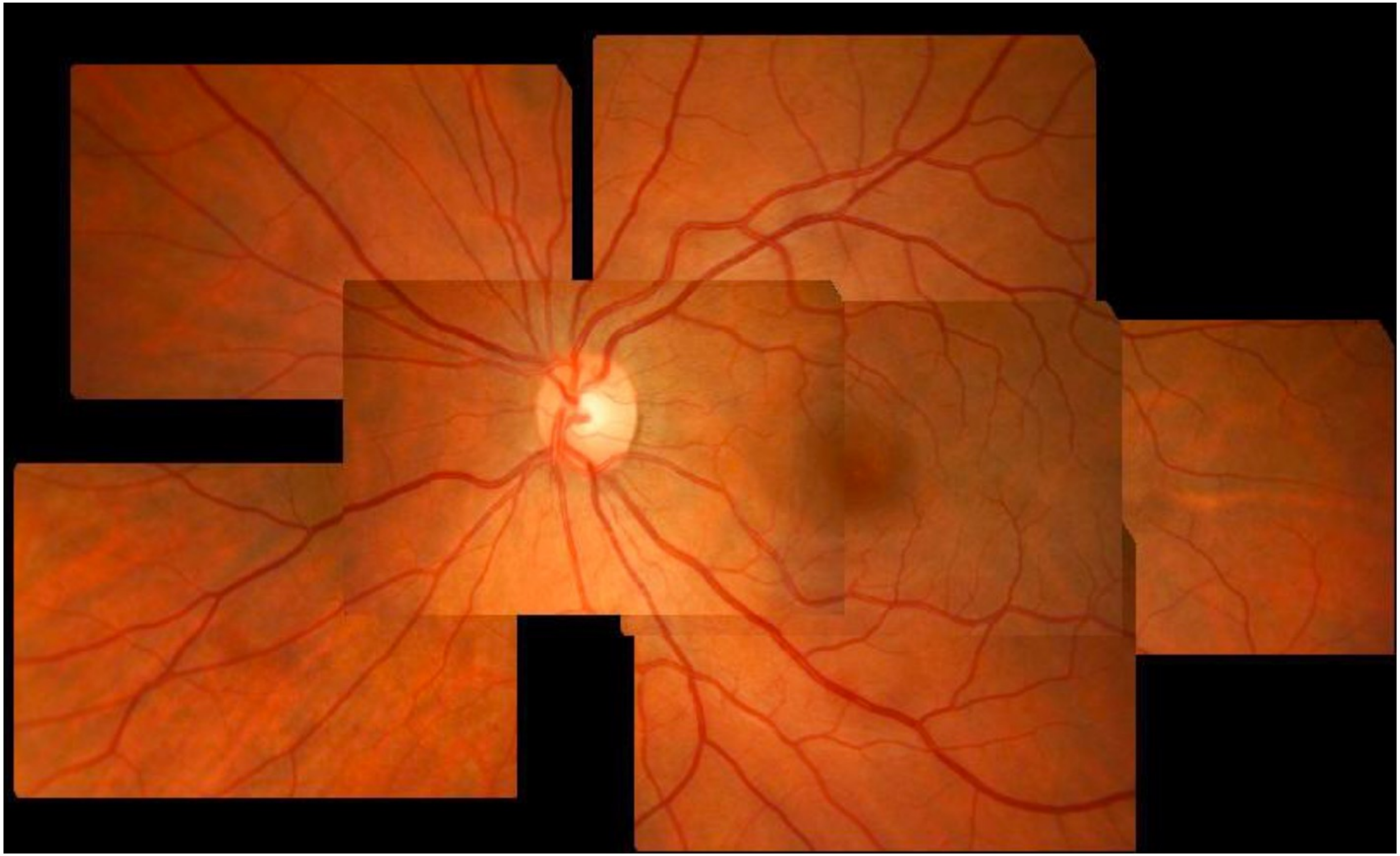


Mosaicking: Homography

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

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.*



Global optimization

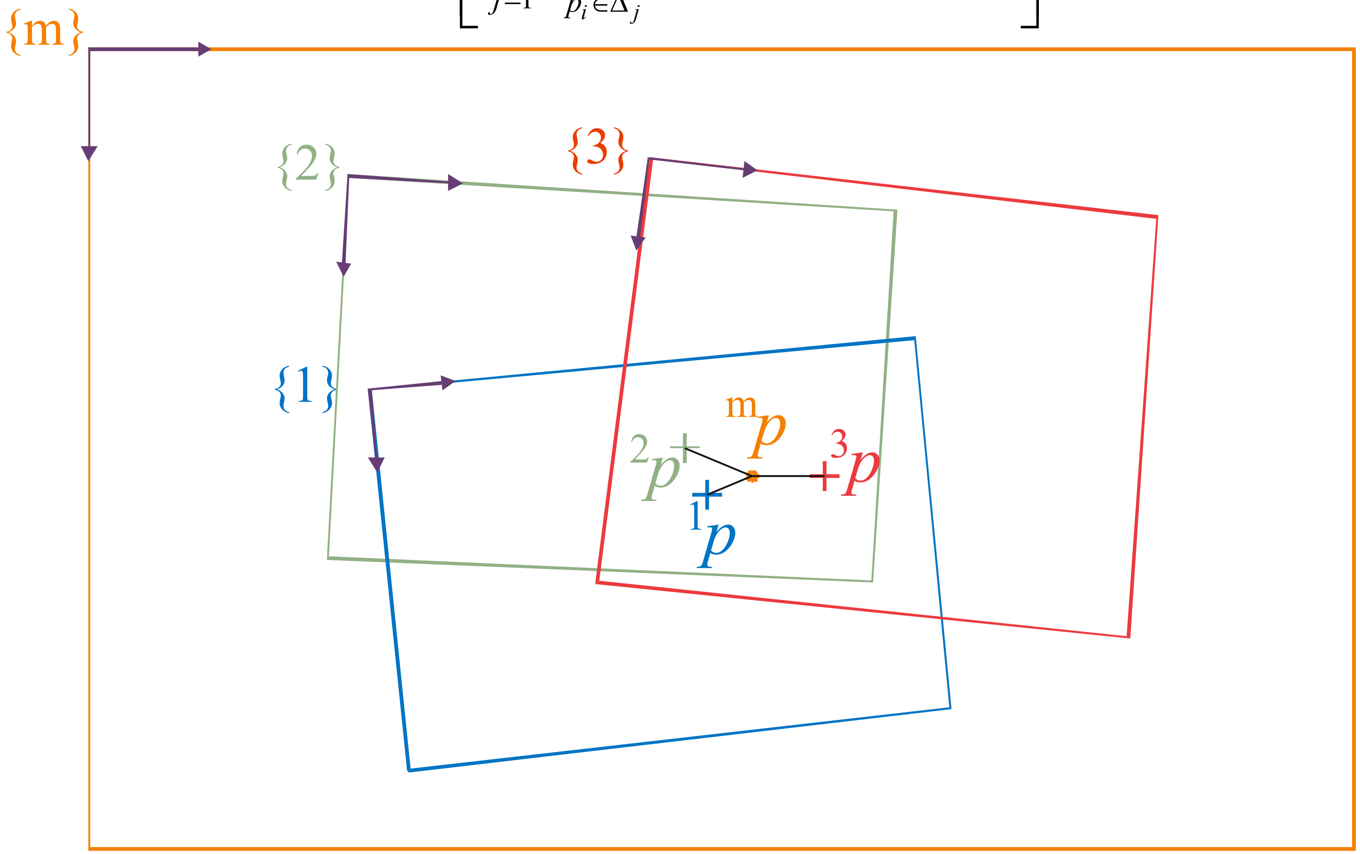
- *In the global optimisation step, we try to find parameters for each image, such that the difference between the pairwise homgraphy and the pairwise homography induced by the global parameters is minimal.*
- *The main steps of the algorithm:*
  - *Feature extraction for each image*
  - *Matching two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

Global optimization

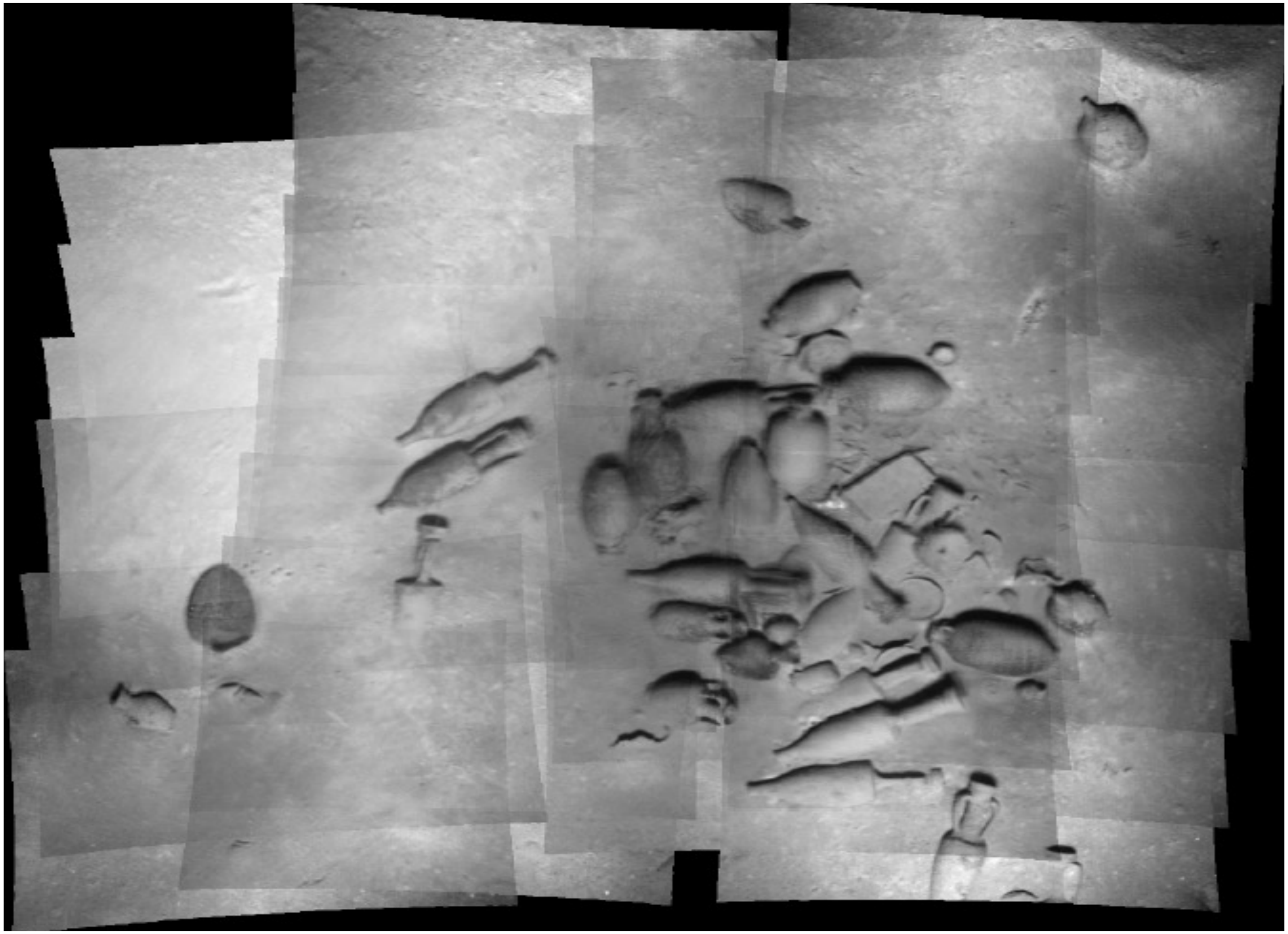
Bundle adjustment

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

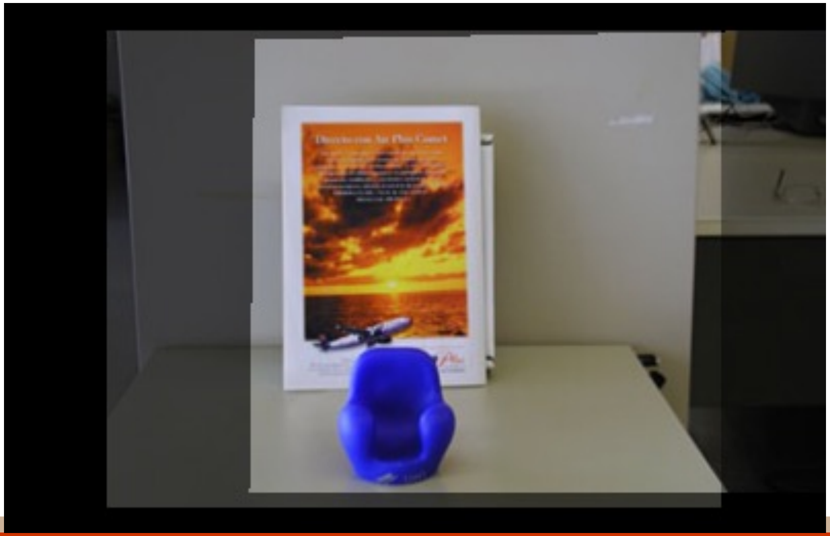
$$\left({}^{1}p, {}^{2}p, {}^{3}p\right) \in \Delta$$



Global optimization



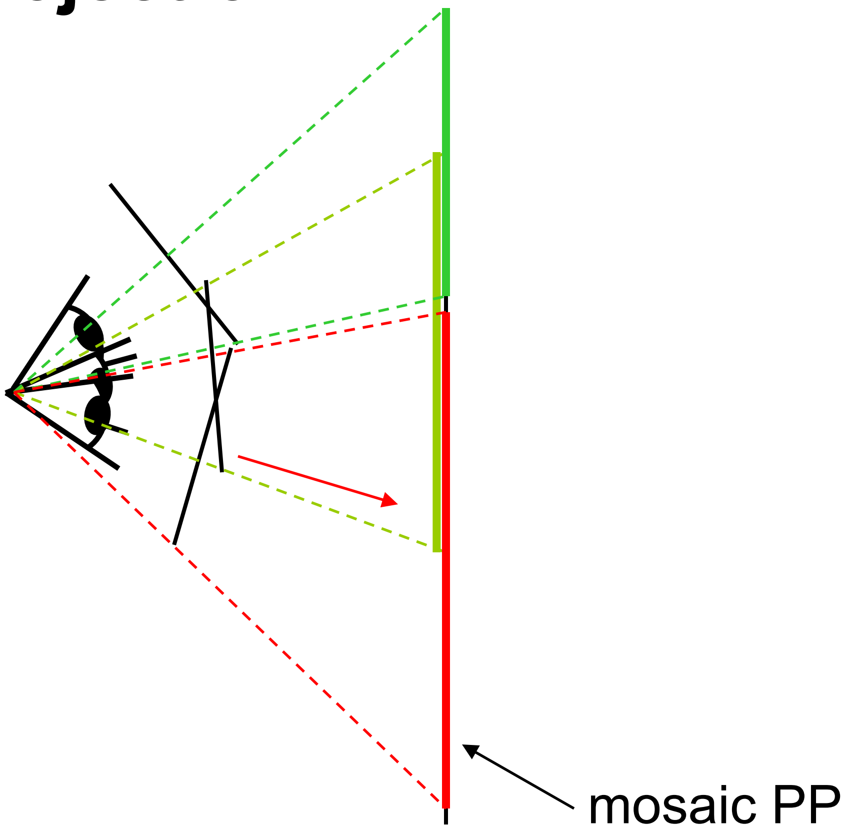
3D scene and a rotating camera



Parallax in a translating camera

Image reprojection

- 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

Application: video stabilization

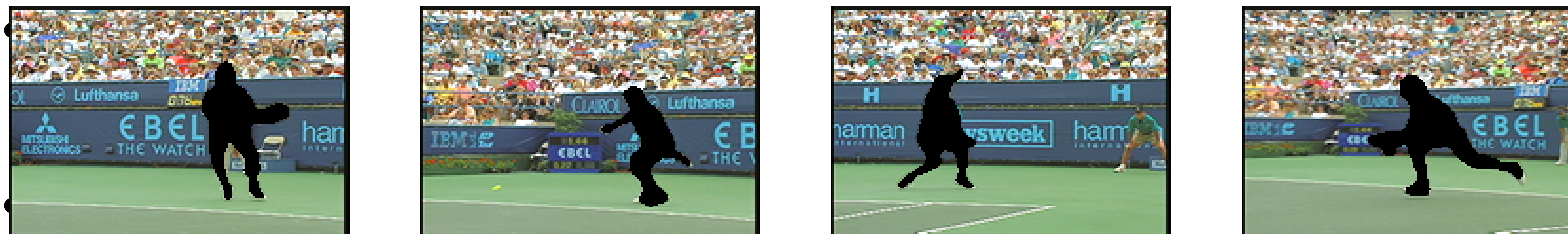
Application: video stabilization

Application: video stabilization

Application: video stabilization

Mosaics for Video Coding

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



=

Computing the homography matrix

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

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

<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

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

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

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

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

8 DOF (Projective) Case

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

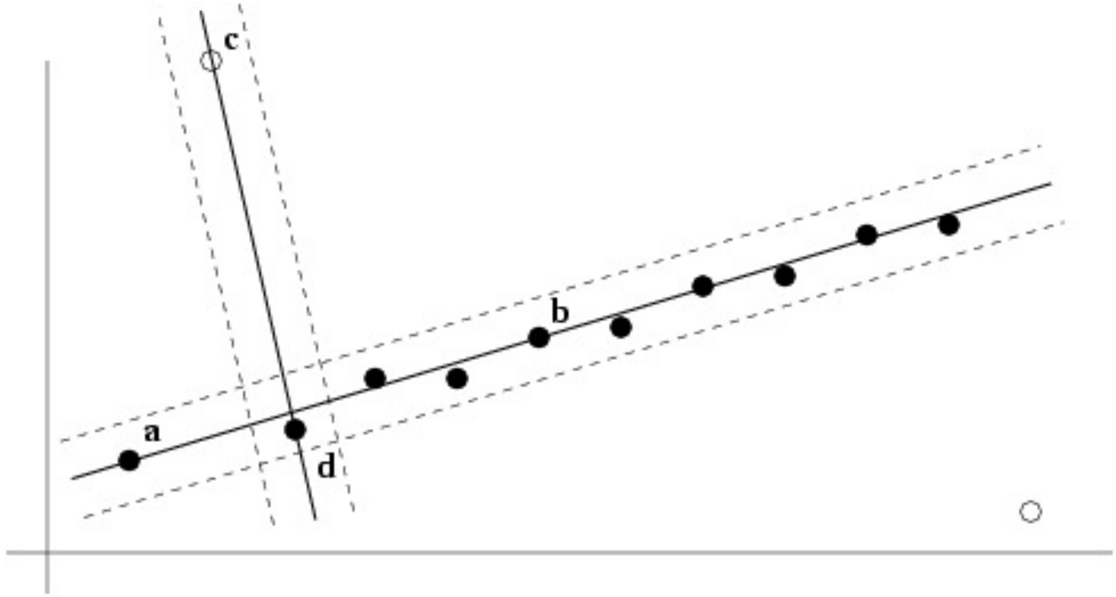
$$\begin{bmatrix} x_1' & y_1' & 1 & 0 & 0 & 0 & -x_1 \cdot x_1' & -x_1 \cdot y_1' \\ 0 & 0 & 0 & x_1' & y_1' & 1 & -y_1 \cdot x_1' & -y_1 \cdot y_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n' & y_n' & 1 & 0 & 0 & 0 & -x_n \cdot x_n' & -x_n \cdot y_n' \\ 0 & 0 & 1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

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

Robust estimation

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



RANSAC

Objective

Robust fit of model to data set S which contains outliers

Algorithm

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

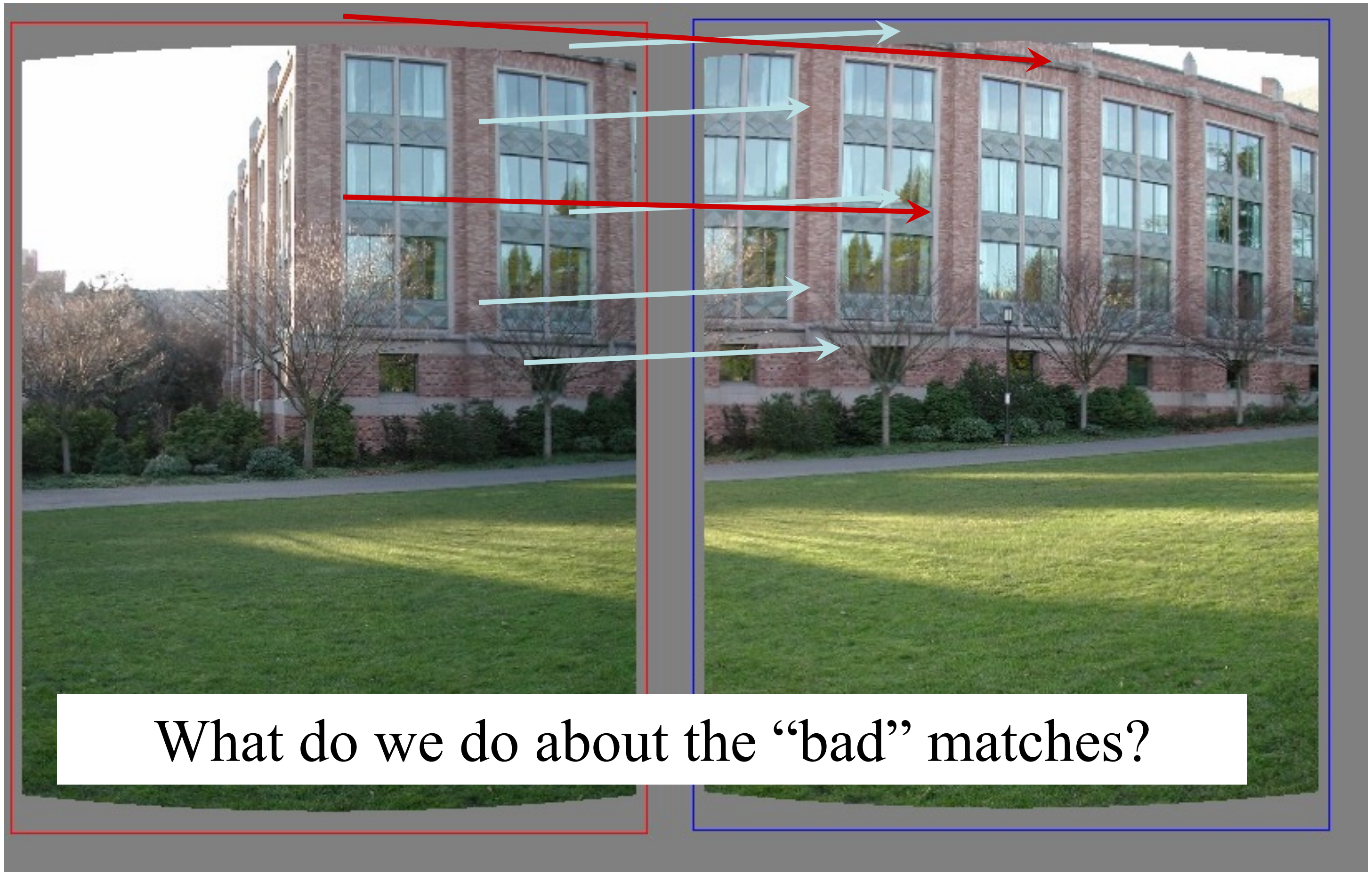
(ii) Determine the set of data points S_i which are within a distance threshold *t* of the model. The set S_i is the consensus set of samples and defines the inliers of S.

(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.

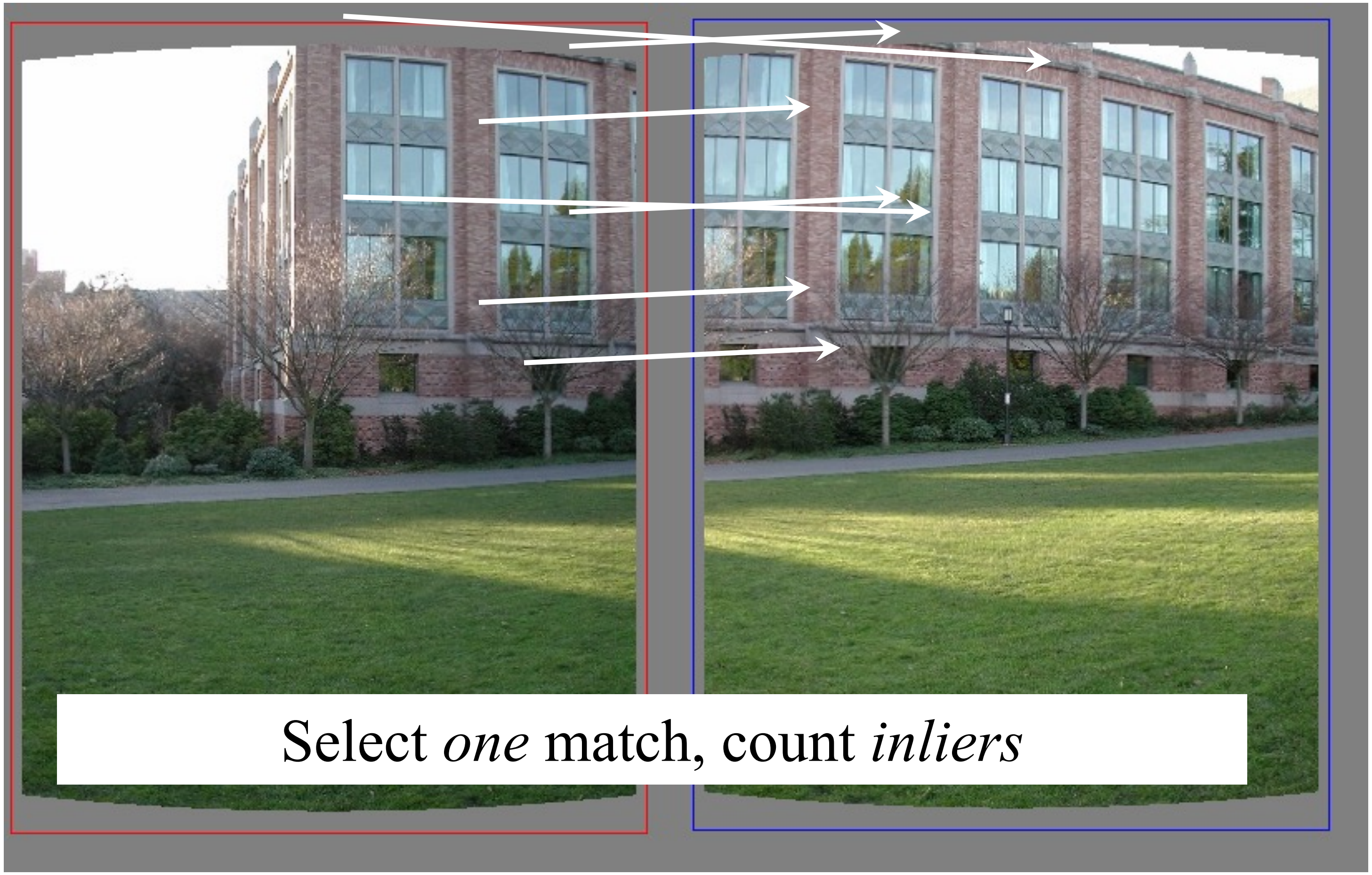
(v) After *N* trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

Matching features

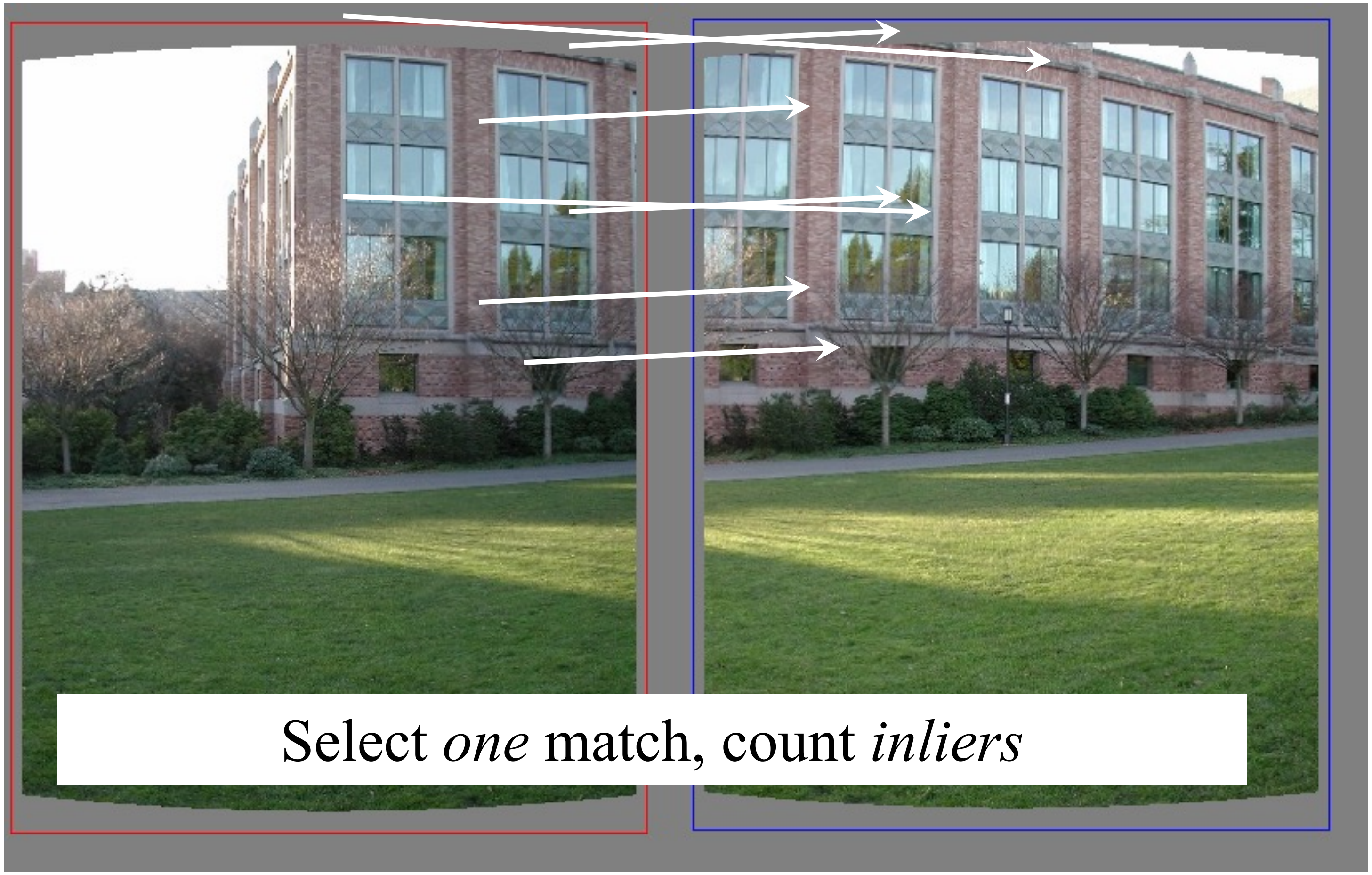


What do we do about the “bad” matches?

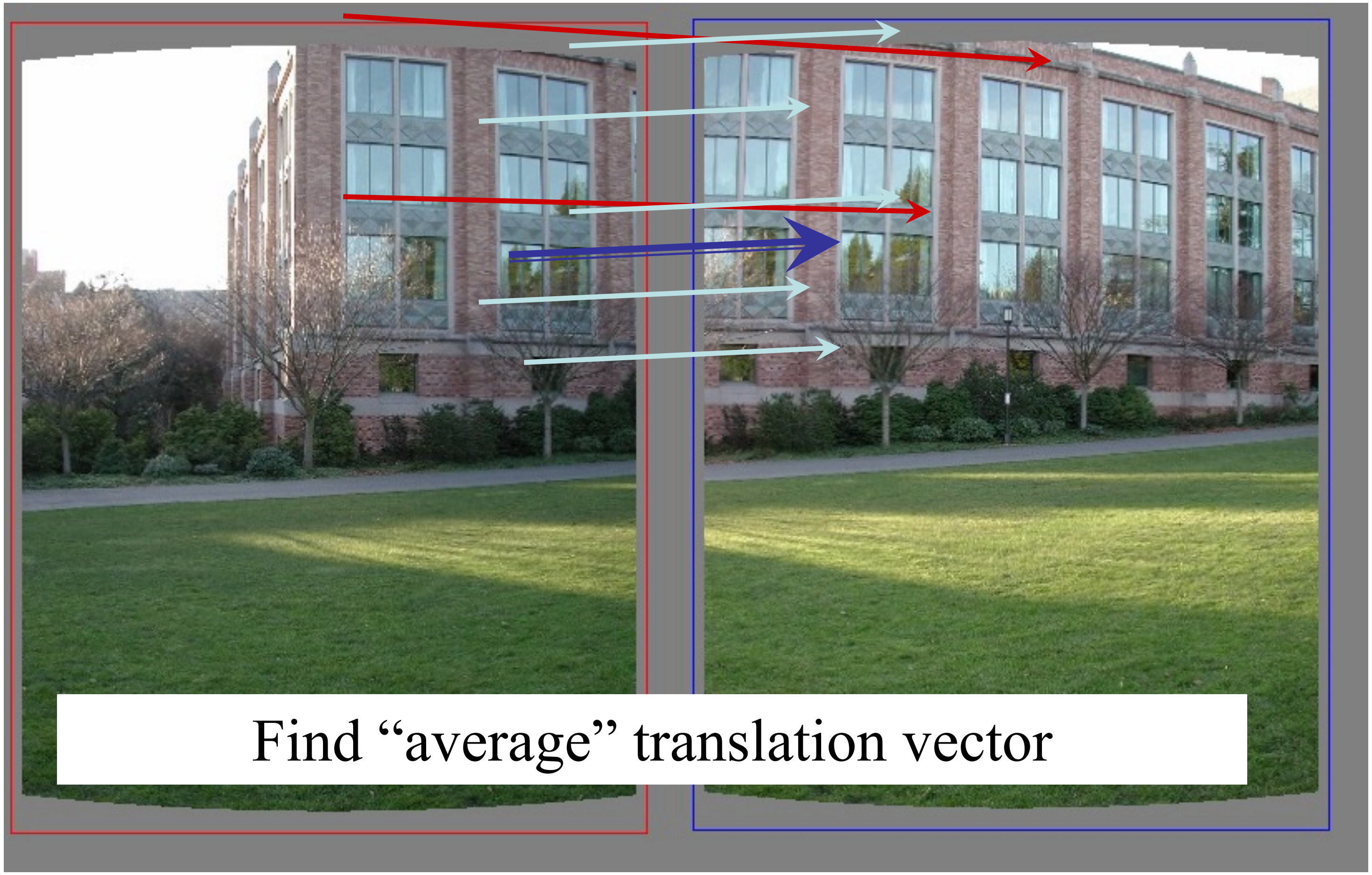
RAndom SAmple Consensus



RAndom SAmple Consensus



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 χ_{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

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

Probability to be an outlier

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

	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

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

- $N=\infty$, *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.