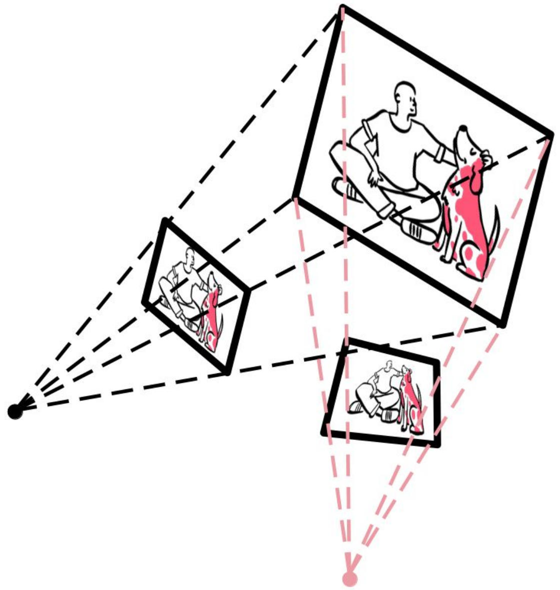
Computer Vision

Class 04



Raquel Frizera Vassallo

Class 04

01 Homography

02 Hierarchy of Transformations

03 Homography in Computer Vision

04 Estimating Homography

- How to estimate
- Key points
- Matching
- RANSAC

01

Homography

What is Homography?

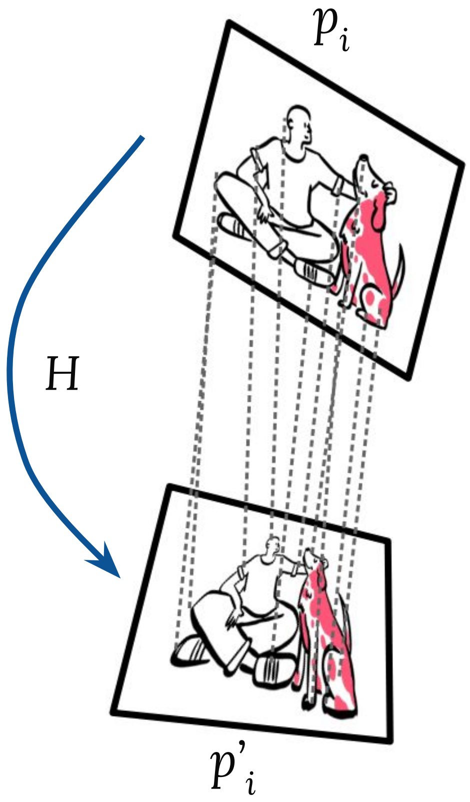
- Homography is a projective transformation that allows mapping a 2D point $p[x,y]^T$ into another 2D point $p'[x',y']^T$

$$p' = Hp$$

- Math:
  - A projective transformation that is an invertible mapping from points p_i in $\mathbb{P}^2$ (homogeneous 3-vectors) to points p'_i in $\mathbb{P}^2$ that maps lines into lines.
  - Thus, if three points p_1, p_2 and p_3 lie on a line, than p'_1, p'_2 and p'_3 also do.

- Defined as a non-singular 3 x 3 matrix:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



02

Hierarchy of Transformations

Hierarchy of transformations

Projective linear group → 8DOF

Affine group → 6DOF

Similarity group → 4DOF

Isometric group → 3DOF

DOF: Degrees of Freedom

Isometric

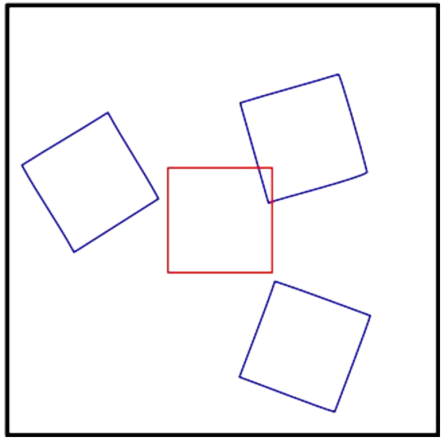
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon\cos\theta & -\sin\theta & t_x \\ \varepsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\varepsilon = \pm 1$

orientation preserving: $\varepsilon = 1$
orientation reversing: $\varepsilon = -1$

- (iso = same, metric = measure)
- 3DOF (1 rotation, 2 translations)
- Orientation preserving (Rigid Body Motion)
  Euclidean Transformation
- Invariants: length, angle, area

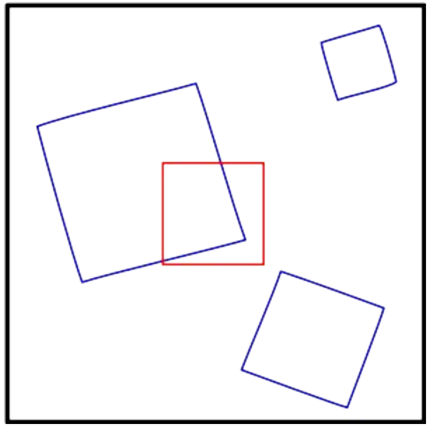
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Similarity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & T \\ \mathbf{0} & 1 \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- (isometry + scale)
- 4DOF (1 scale, 1 rotation, 2 translations)
- Also known as equi-form (shape preserving)
- Invariants: ratios of length, angle, ratios of areas, parallel lines



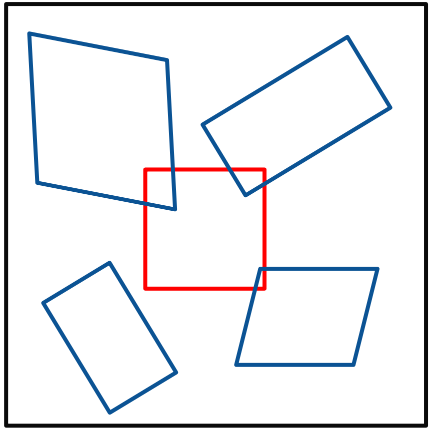
Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$A = R(\theta)R(-\phi)DR(\phi) \qquad \mathrm{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & T \\ \mathbf{0} & 1 \end{bmatrix}_{3\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

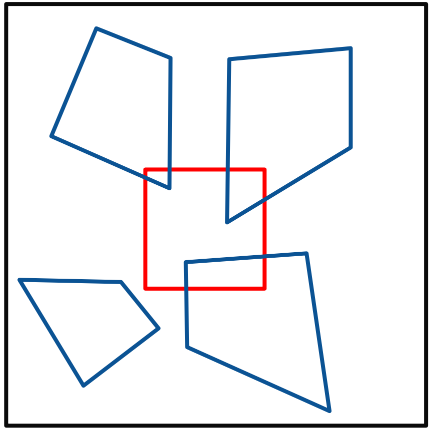
- 6DOF (2 scales, 2 rotations, 2 translations)
- non-isotropic scaling!
- Invariants: parallel lines, ratios of parallel lengths, ratios of areas



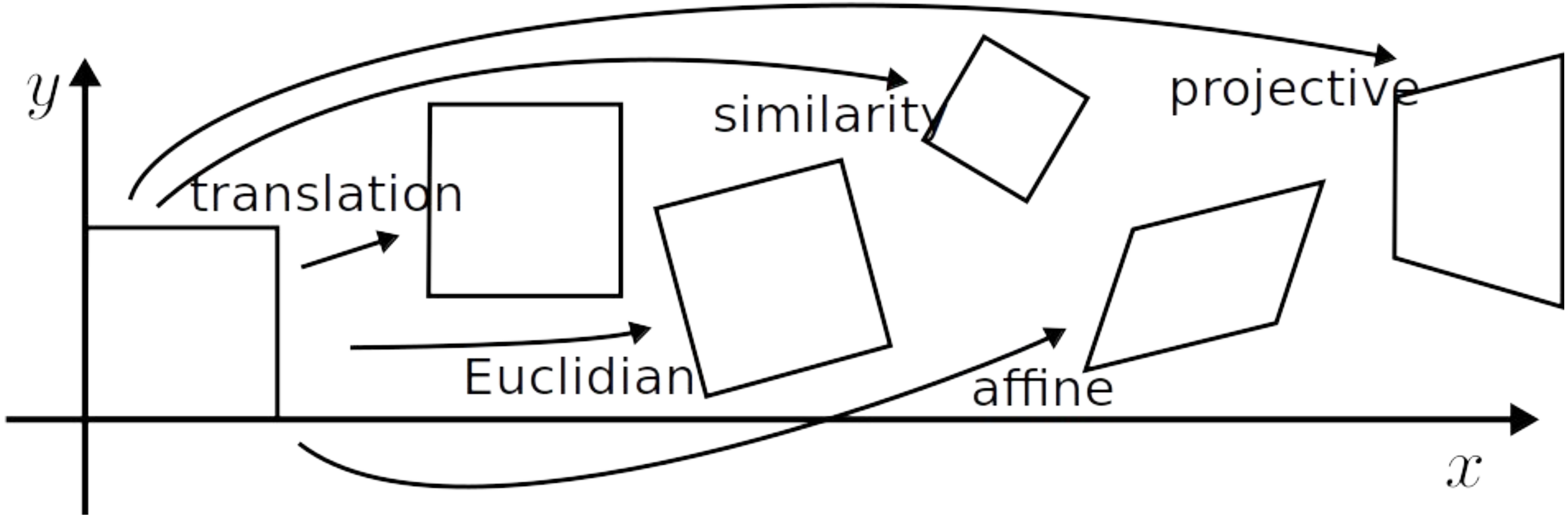
Projective transformations

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & T \\ \mathbf{v}^T & v \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{v} = [v_1, v_2]^T$$

- 8DOF (2 scales, 2 rotations, 2 translations, 2 lines at infinity)
- Invariants: cross-ratio of four points on a line (ratio of ratio)



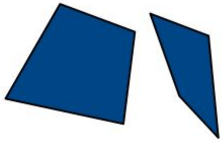
Overview of transformations



Overview of transformations

Projective
8DOF

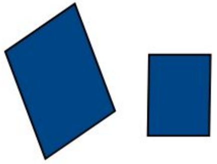
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio

Affine
6DOF

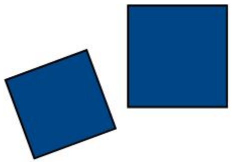
$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Parallelism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids).

Similarity
4DOF

$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Ratios of lengths, angles.

Euclidean
3DOF

$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Angles, lengths, areas, parallel lines.

03

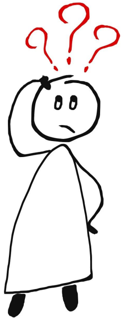
Homography in Computer Vision

Homography in Computer Vision

- A homography is a projective transformation between two planes or, alternatively, a mapping between two planar projections of an image.

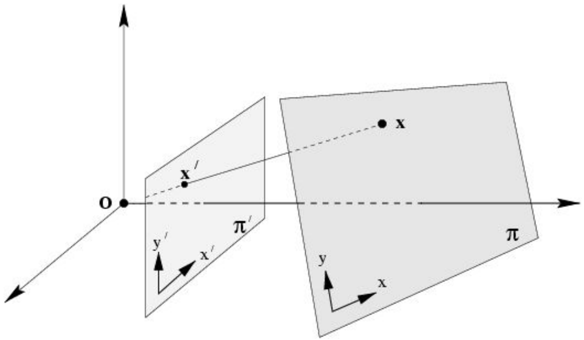
- In other words, homographies are simple image transformations that describe the relative motion between two images, when the camera (or the observed object) moves.

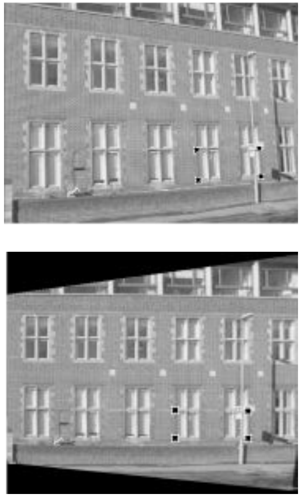
- It is the simplest kind of transformation that describes the 2D relationship between two images.



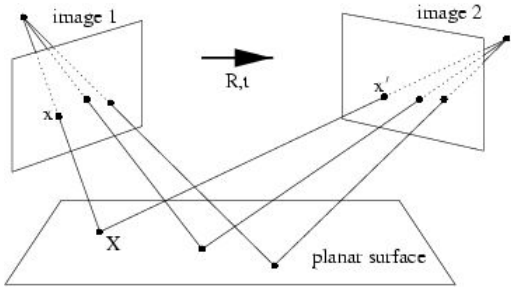
Homography in Computer Vision

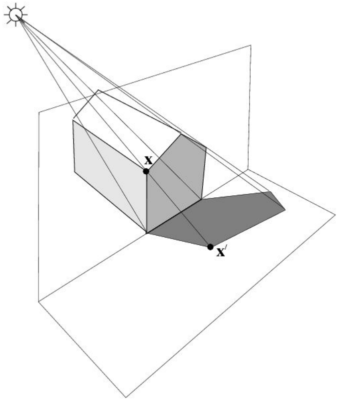
Projective transformation between two planes





Relative motion between two images, when the camera (or the observed object) moves

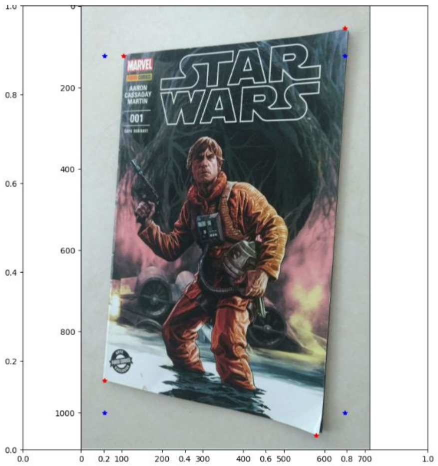


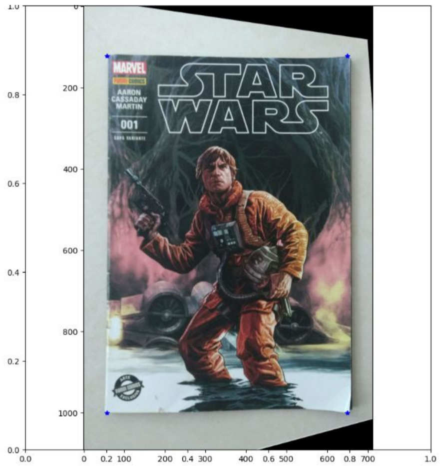


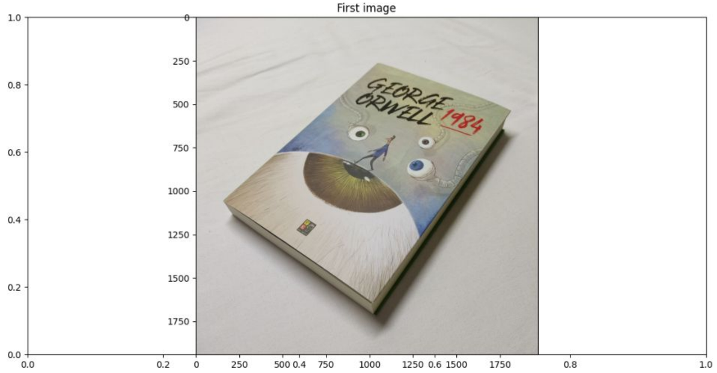
It can be the transformation between one plane in 3D space and an image

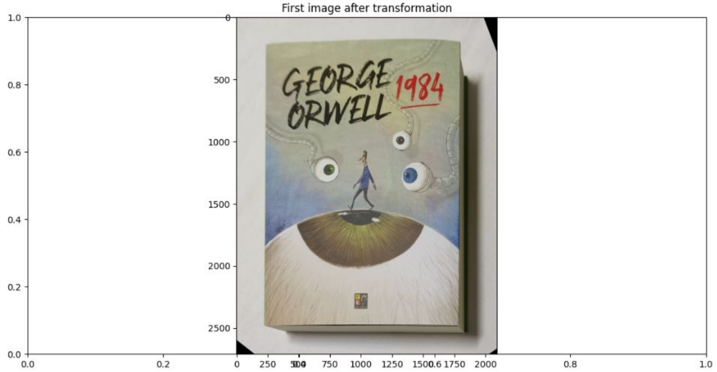
Using Homography

Rectifying documents



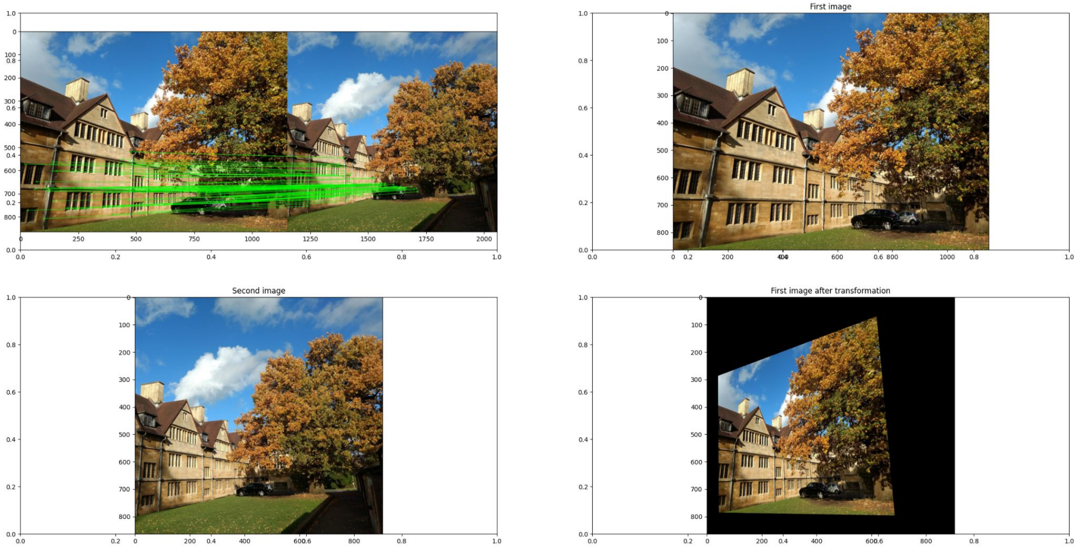






Using Homography

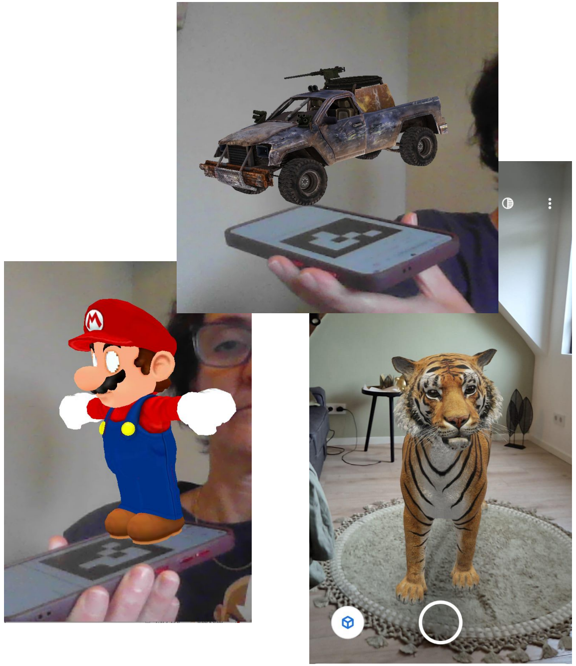
Transforming images as they were taken from the same point of view.



Creating mosaics.

Using Homography

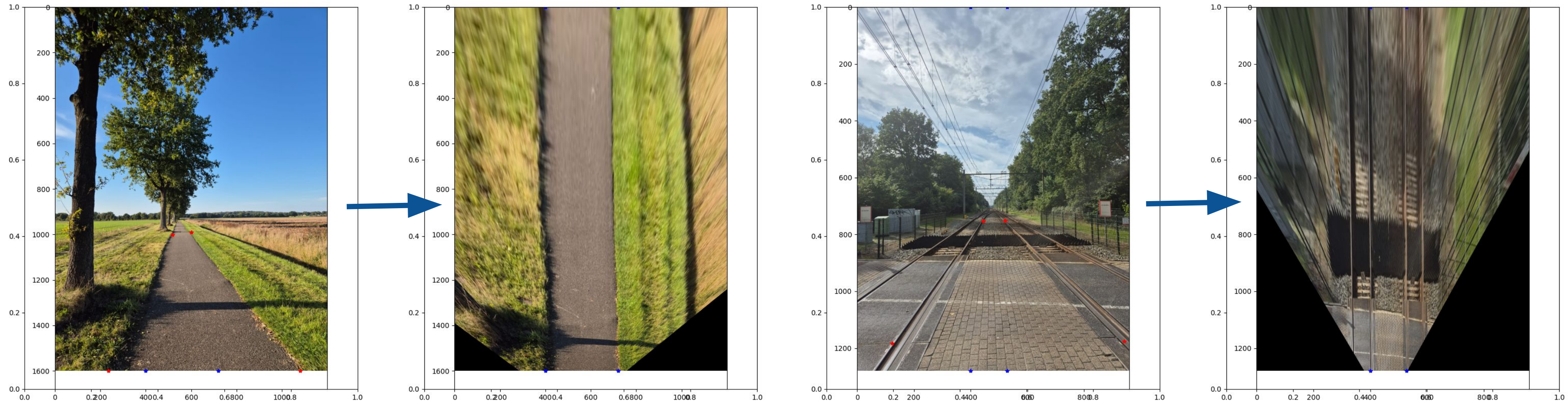
Augmented reality



Estimating the projective transformation for objects in an image

Using Homography

Remapping planes to create wrappings as Bird’s Eye View



04

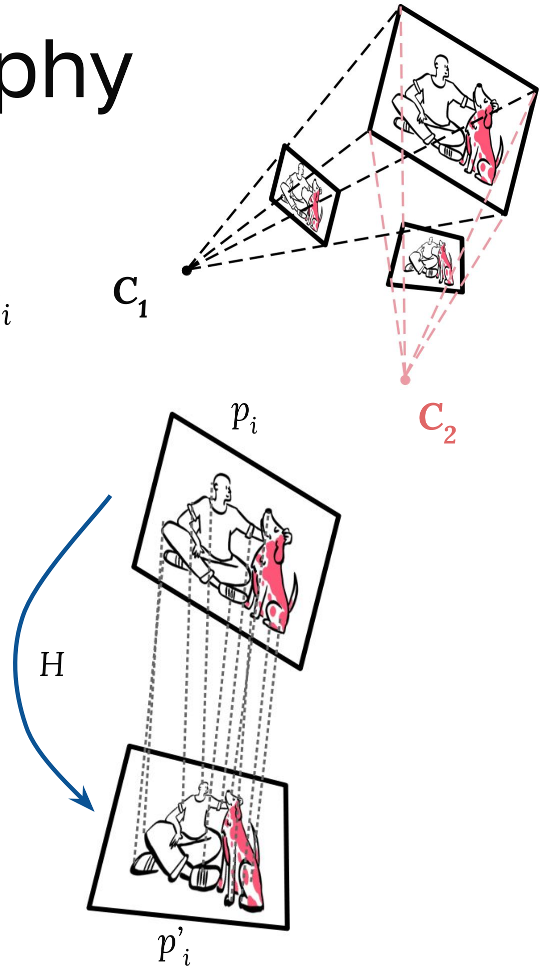
Estimating Homography

How to estimate
Key points
Matching
RANSAC

Estimating Homography

- Given a set of (p_i, p'_i), compute H that satisfies $p_i' = H\,p_i$

$$p_i' = H p_i \qquad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

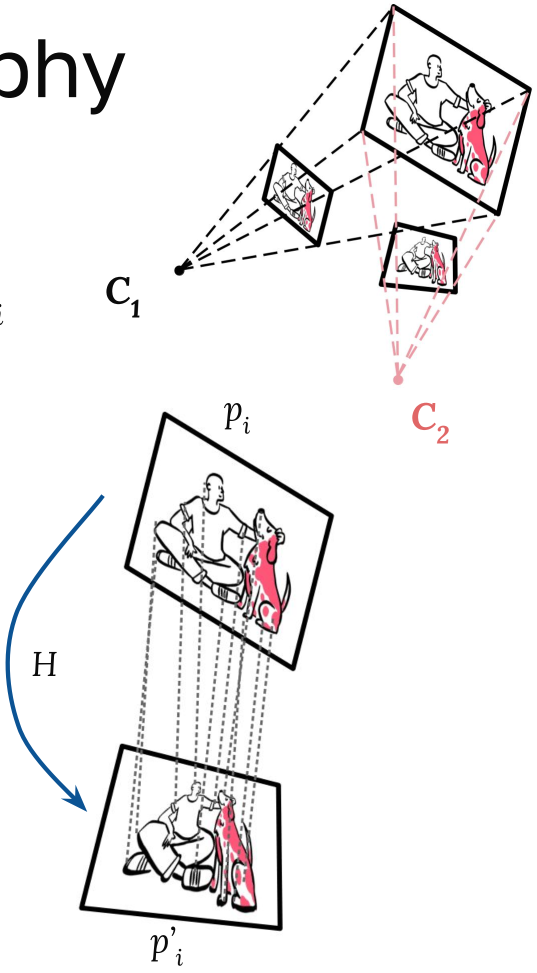


Estimating Homography

- Given a set of (p_i, p'_i), compute H that satisfies $p_i' = H\, p_i$

$$p_i' = H p_i \qquad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

- We need at least as many independent equations as the matrix degrees of freedom.

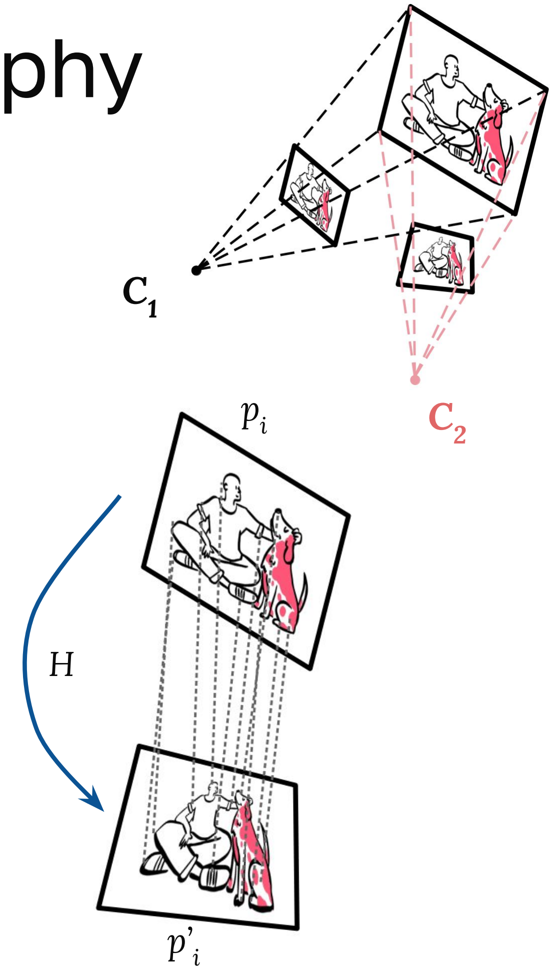


Estimating Homography

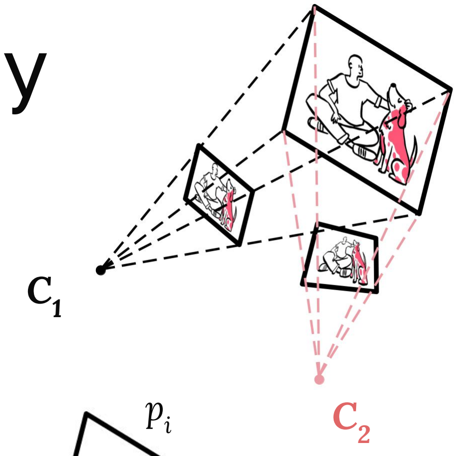
- We need at least as many independent equations as the matrix degrees of freedom:

$$p'_i = Hp_i \rightarrow \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  - Homography is defined up to a scale factor
  - **8 DOF (degrees of freedom)**
    Need at least 8 equations



Estimating Homography

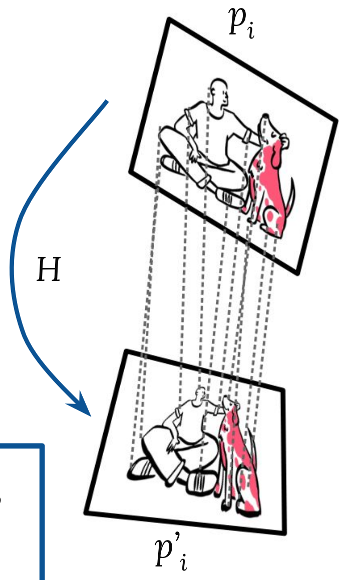


- We need at least as many independent equations as the matrix degrees of freedom:

$$p_i' = Hp_i \rightarrow \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  - Homography is defined up to a scale factor
  - **8 DOF (degrees of freedom)**
    Need at least 8 equations
  - 2 independent equations per point

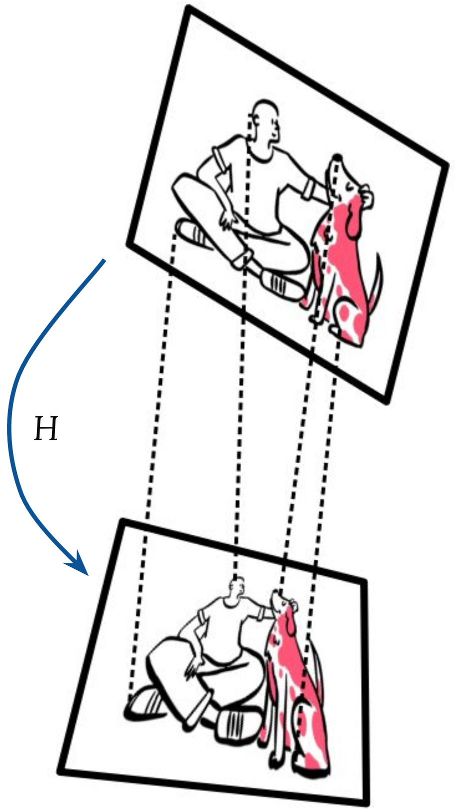
We need 4 or more matchings
$2N \geq 8 \rightarrow N \geq 4$



Estimating Homography

- Minimal solution:
    - 4 points yield an exact solution for H.

$$\boxed{2N \geq 8 \rightarrow N \geq 4}$$

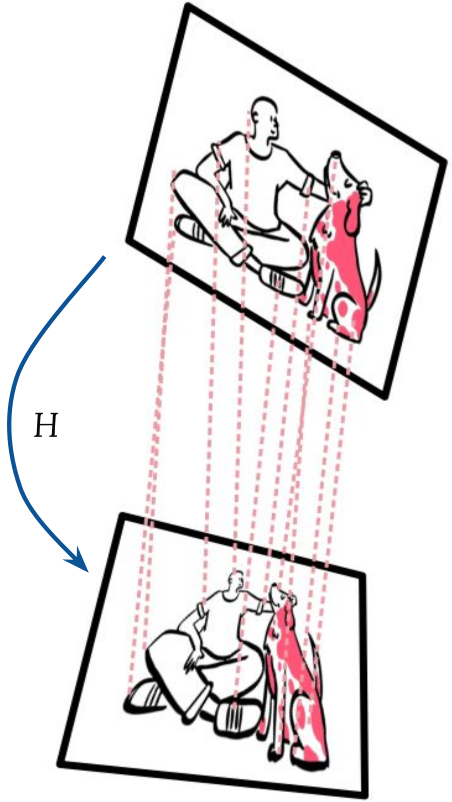


Estimating Homography

- Minimal solution:
    - 4 points yield an exact solution for H.

$$\boxed{2N \geq 8 \rightarrow N \geq 4}$$

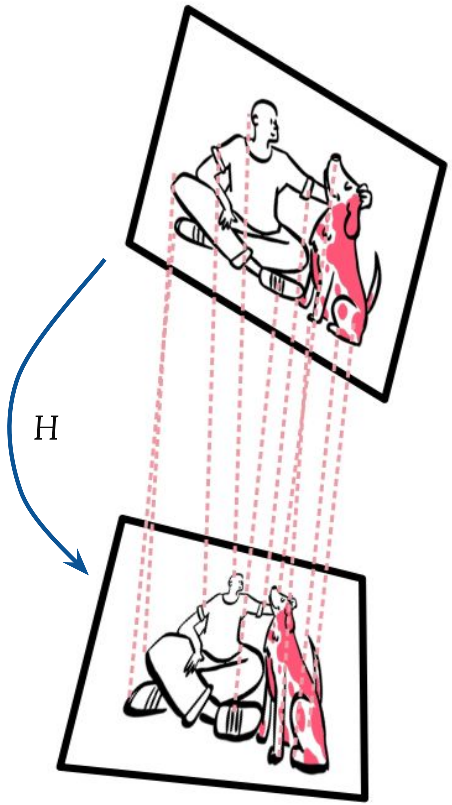
- More points:
    - No exact solution, because measurements are inexact ("noise").
    - **Search for "best" according to some cost function.**



Estimating Homography

To compute homography between two images:

1. **Find key points:** Compute interest points in each image
2. **Find correspondences:** Compute a set of matchings based on some similarity measure
3. **Estimate homography:**
    a. Using a robust method like RANSAC
    b. Optimizing the final result, if needed



Find key Points

Use features detectors and descriptors as:

- **Harris Corner Detector** – one of the earliest detectors, good for finding corners.
- **Shi–Tomasi (Good Features to Track)** – an improvement over Harris, widely used in tracking.
- **SIFT (Scale-Invariant Feature Transform)** – detects distinctive features that are robust to scale, rotation, and illumination changes.
- **SURF (Speeded-Up Robust Features)** – a faster alternative to SIFT.
- **BRIEF (Binary Robust Independent Elementary Features)** - faster method feature descriptor calculation and matching.
- **ORB (Oriented FAST and Rotated BRIEF)** – efficient and free of patent restrictions, often used in real-time applications.
- **FAST (Features from Accelerated Segment Test)** – very fast detector, often used for real-time corner detection.
- **BRISK (Binary Robust Invariant Scalable Keypoints)** – robust and efficient for matching binary descriptors.
- **AKAZE (Accelerated-KAZE)** – detects multiscale features using nonlinear diffusion filtering.

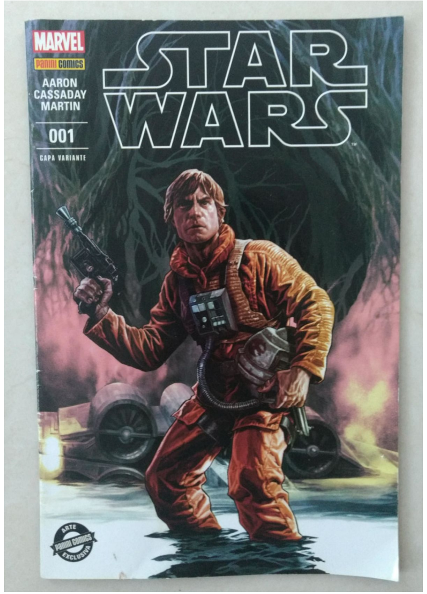
Detector without descriptor

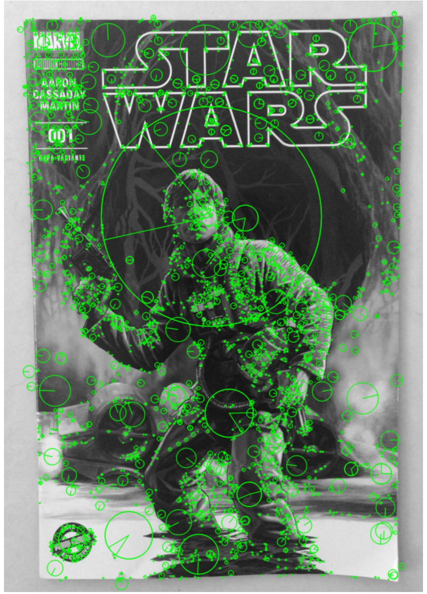
Descriptor without detector

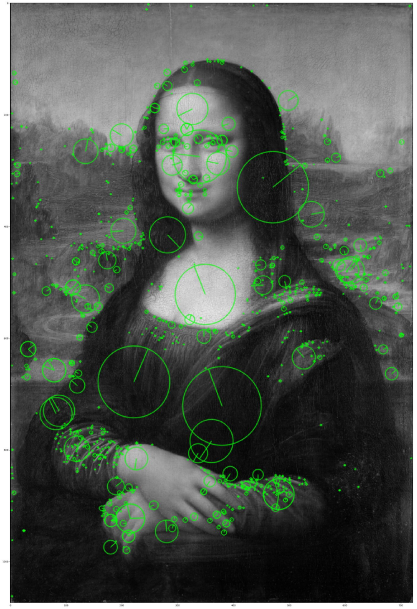
Detectors paired with descriptors

Find key Points

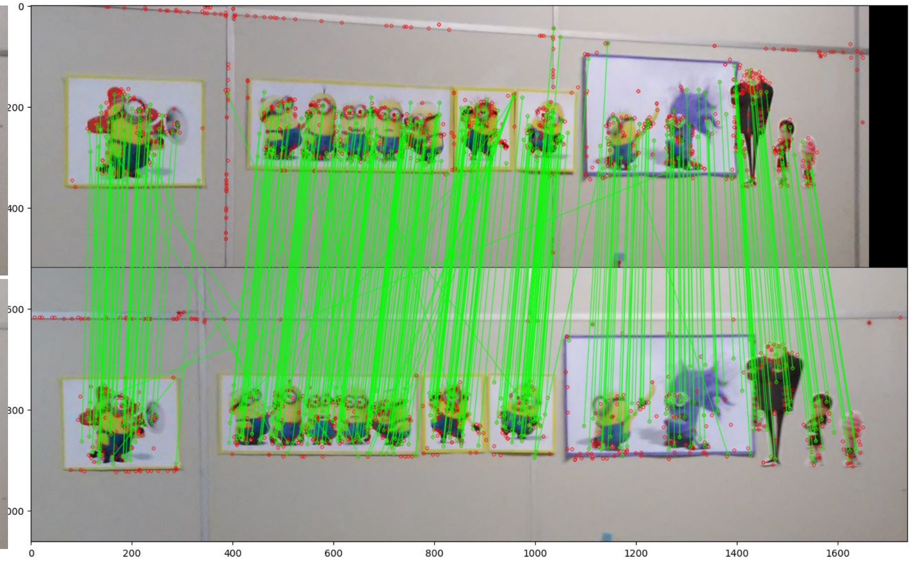
SIFT





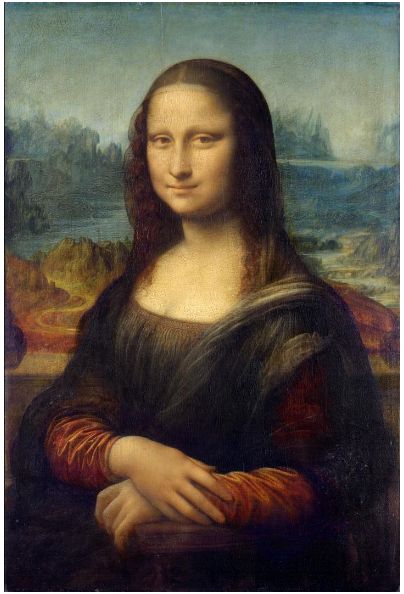


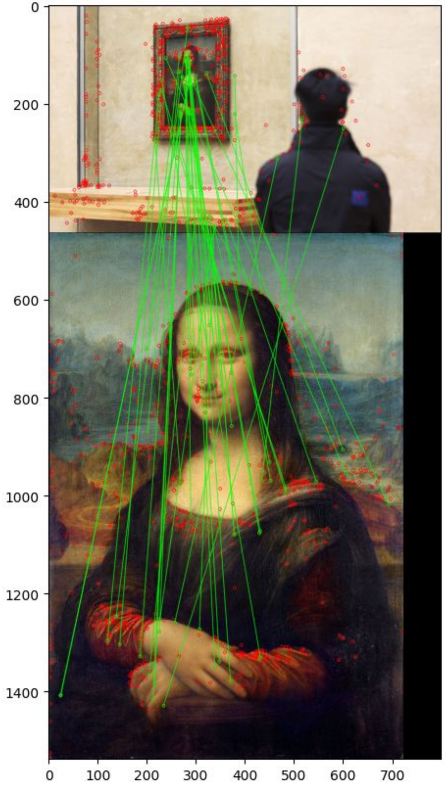
Find Correspondences



Note that there are **OUTLIERS.**

Find Correspondences

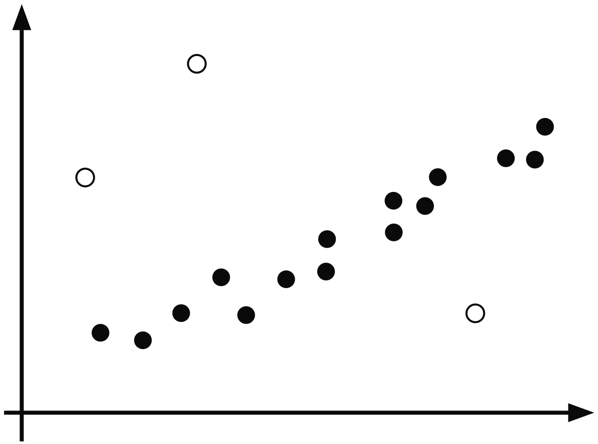




Note that there are **OUTLIERS.**

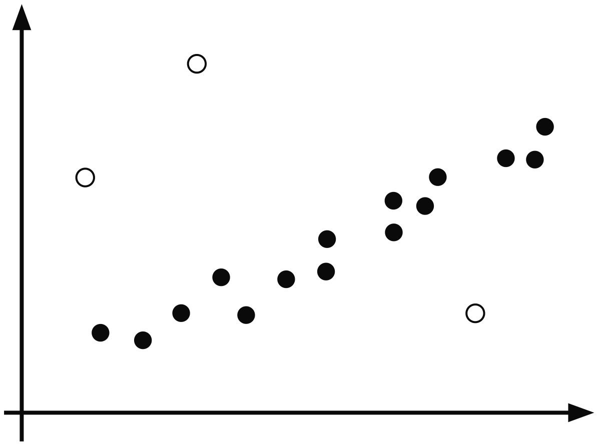
RANSAC - RANdom SAmple Consensus

Robust estimation that eliminates outliers.



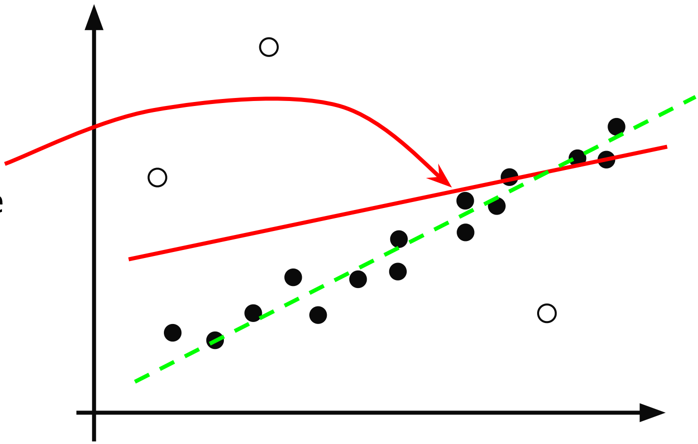
RANSAC - RANdom SAmple Consensus

Robust estimation that eliminates outliers.



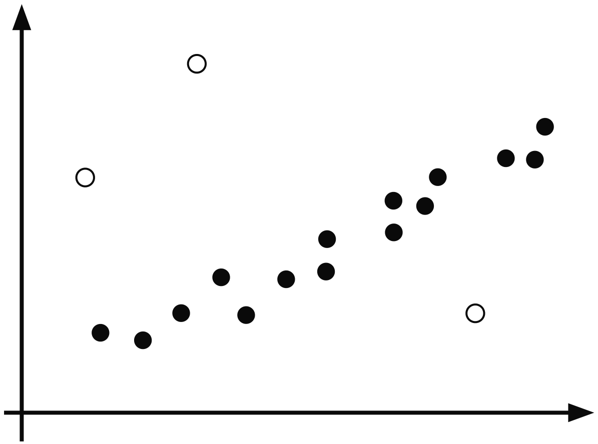
Why is that important?

Suppose we use all the points to interpolate the line using Least Squares



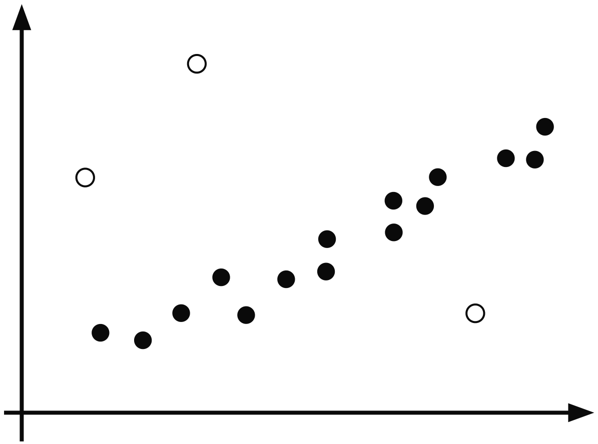
RANSAC - RANdom SAmple Consensus

How we get rid of the outliers?



RANSAC - RANdom SAmple Consensus

How we get rid of the outliers?



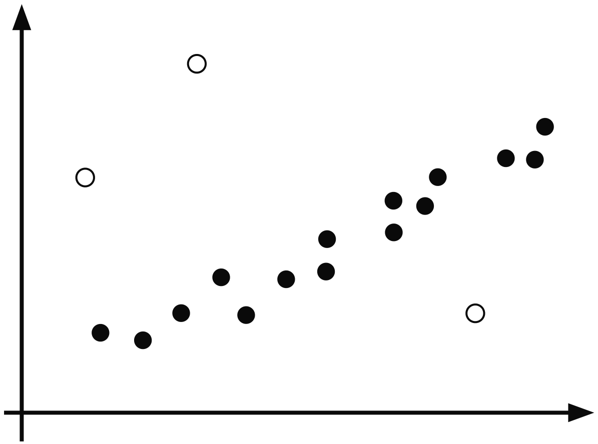
Let's iteratively:

- randomly select 2 points
- estimate the line
- test the model for all the other points, until we can get a consensus set that can be considered a good set of inliers after some number of iterations.

RANSAC - RANdom SAmple Consensus

Select:

- Randomly select a sample of data points from all the points and instantiate the model from this subset.



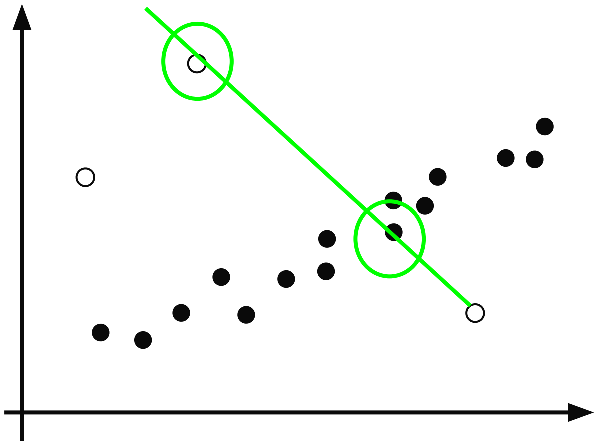
We can use the minimal set of samples to instantiate the model.

In this case, we need just 2 points.

RANSAC - RANdom SAmple Consensus

Select:

- Randomly select a sample of data points from all the points and instantiate the model from this subset.



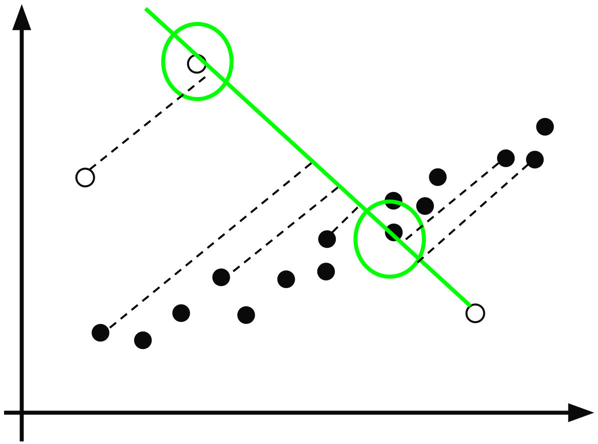
We can use the minimal set of samples to instantiate the model.

In this case, we need just 2 points.

RANSAC - RANdom SAmple Consensus

Test:

- Determine the set of data points which are within a distance threshold of the model. This set is the consensus set of samples and defines the inliers for that iteration.



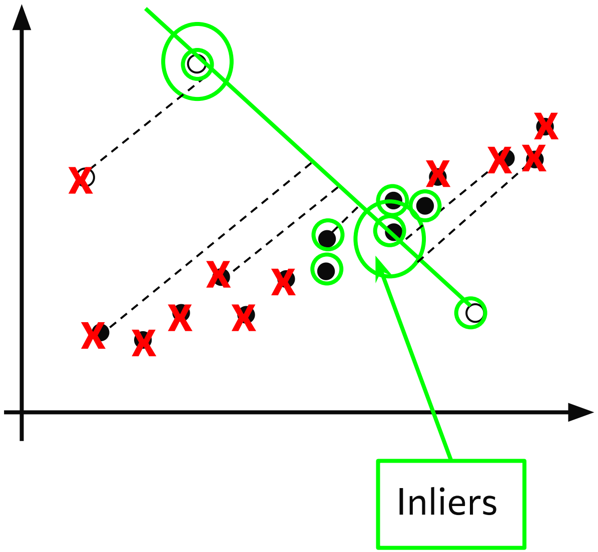
If $d_i < threshold$:

Store the point as an inlier.
Count the number of inliers of that iteration.

RANSAC - RANdom SAmple Consensus

Test:

- Determine the set of data points which are within a distance threshold of the model. This set is the consensus set of samples and defines the inliers for that iteration.



If d_i < *threshold:*

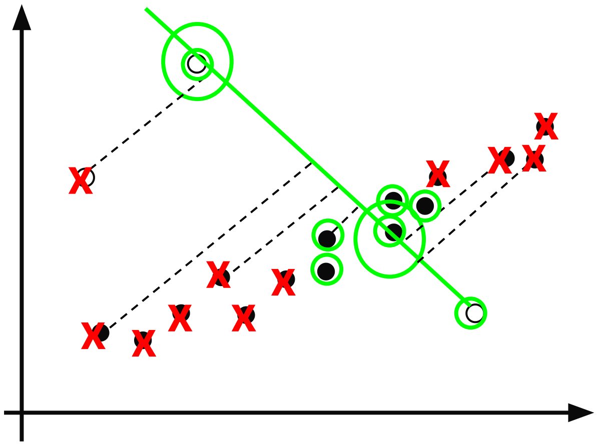
Store the point as an inlier.
Count the number of inliers of that itaration.

Note that points that don't meet this threshold are not stored as inliers. Thus they are eliminated from the consensus set.

RANSAC - RANdom SAmple Consensus

Check:

- Verify if that is the best set of inliers obtained until this moment. If that is the case, store this set as the Best Set of Inliers.
- Also verify if the number of inliers meet the requirement for stopping the loop (i.e. is higher than the number of inliers required by the problem)



If the $N_{inliers} >= Max_{inliers}$:

$$Best_{inliers} = Current_{inliers}$$

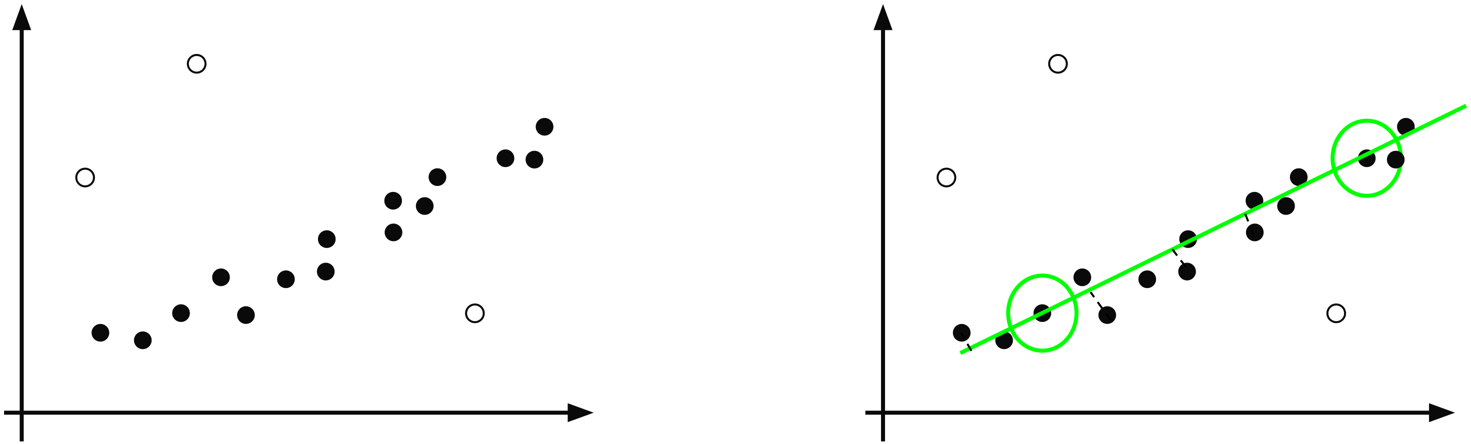
If the $N_{inliers} >= Needed_{inliers}$:

Break loop

RANSAC - RANdom SAmple Consensus

Repeat Loop:

- If the inliers criteria is not achieved
- If the maximum number of trials is also not achieved
- Select a new subset and repeat the loop

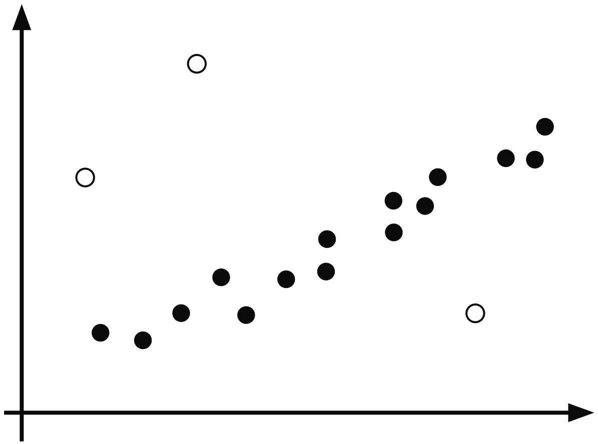


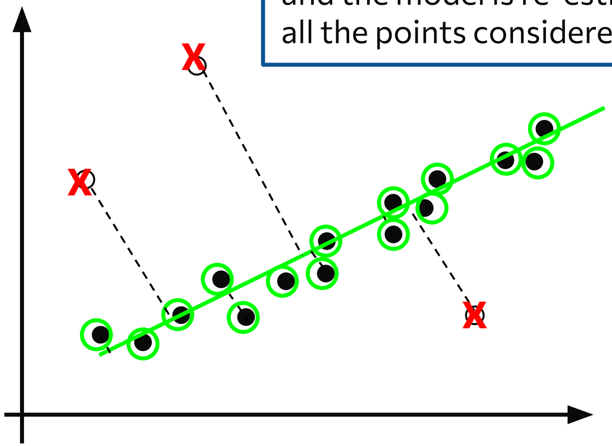
RANSAC - RANdom SAmple Consensus

Repeat Loop:

- If the inliers criteria is not achieved
- If the maximum number of trials is also not achieved
- Select a new subset and repeat the loop

After a certain number of trials the largest consensus set is selected, and the model is re-estimated using all the points considered as inliers.





RANSAC - RANdom SAmple Consensus

Objective

- Robust fit a model to a data set S which contains outliers

Algorithm

1. Randomly select a sample of k data points from $Samples$ (complete set) and instantiate the model from this subset.

2. Determine the set of data points $Current_{inliers}$ which are within a distance threshold of the model. The $Current_{inliers}$ is the consensus set of samples and defines the inliers of $Samples$.

3. If the subset of $Current_{inliers}$ is greater than $Max_{inliers}$ store the set as $Best_{inliers}$

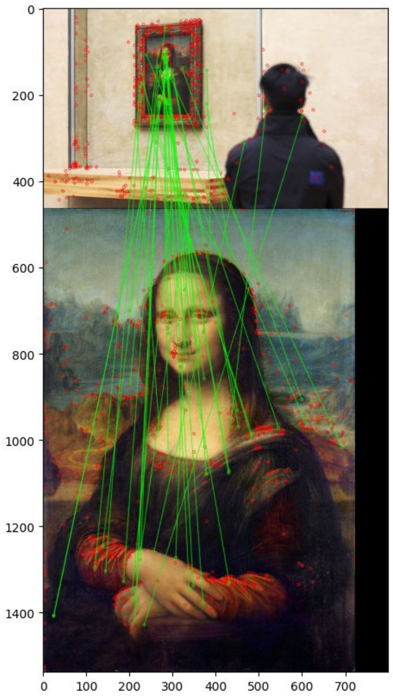
4. If the set $Best_{inliers}$ is greater than the desired number of inliers, stop the loop and re-estimate the model using all the points in $Best_{inliers}$ and terminate

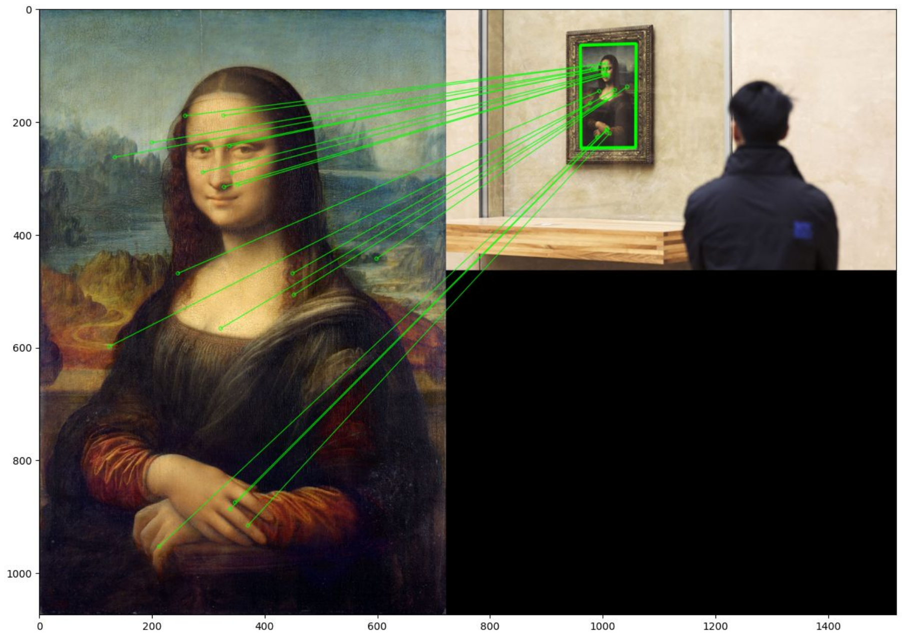
5. But if the size of $Best_{inliers}$ is less, select a new subset and repeat the above steps.

6. After N trials the largest consensus set $Best_{inliers}$ is selected, and the model is re-estimated using all the points in that subset.

Estimating the Homography

Note that the outliers are gone after using RANSAC.





So now…
Let’s play with Homography

Credits

Richard Hartley and Andrew Zisserman.
Multiple View Geometry in Computer Vision.
Cambridge, ISBN 0521623049