Feature Matching and Robust Fit

IACV
December 10th 2021
Giacomo Boracchi

https://boracchi.faculty.polimi.it/

Slide credits Luca Magri

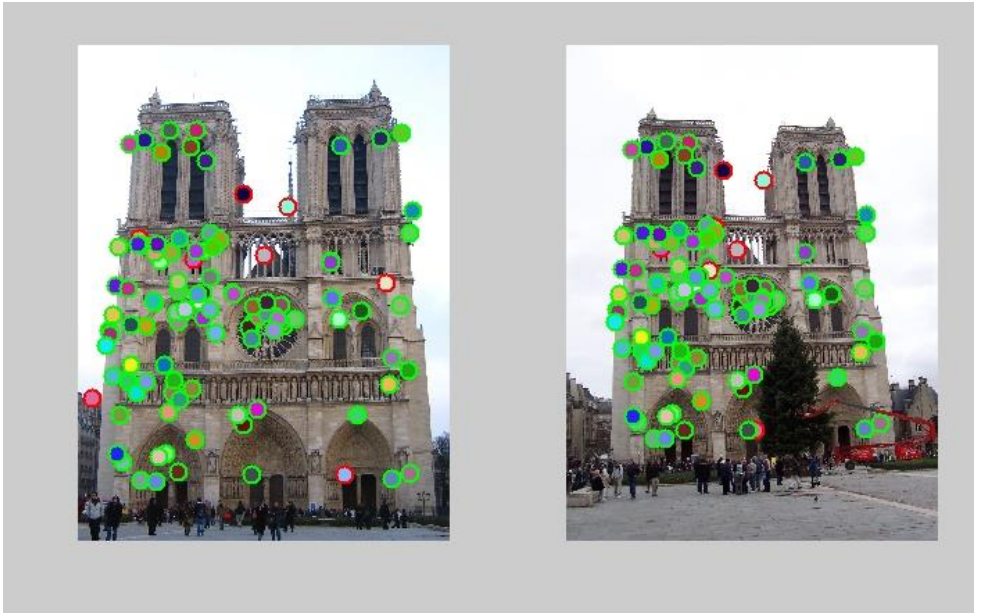
Matching of Computer Vision Features

Estimating Image Correspondences

👉 Extract features from each image

- Keypoint detection
- Descriptor Computation

- Match features between images
- Prune matches and then perform triangulation
  detect objects / stitching …



Matching of Computer Vision Features

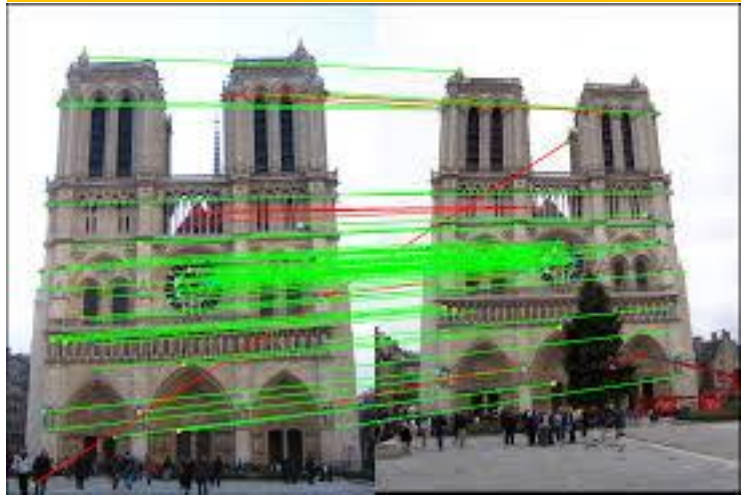
Estimating Image Correspondences

- Extract features from each image
  - Keypoint detection
  - Descriptor Computation

☛ Match features between images

- Prune matches and then perform triangulation detect objects / stitching …

Matching:
identify in a pair of images
the image points that correspond
to the same 3D point in the scene



Matching of Computer Vision Features

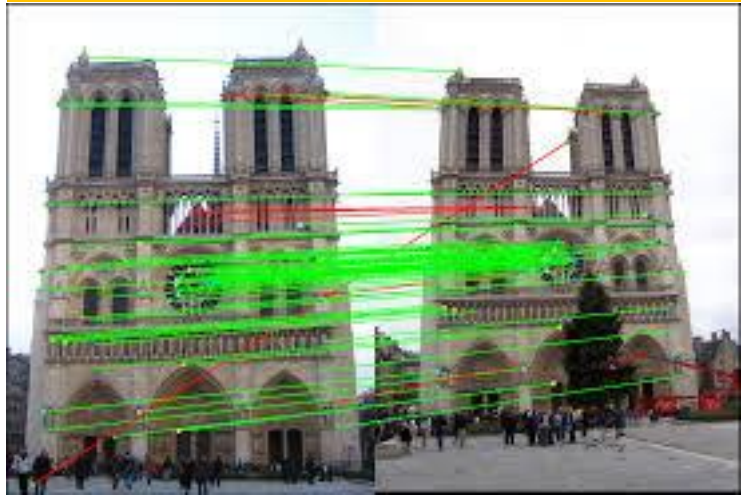
Estimating Image Correspondences

- Extract features from each image
  - Keypoint detection
  - Descriptor Computation
- Match features between images

☛ Prune matches and then perform triangulation detect objects / stitching. Geometry into play!

Robust fit:
fit a geometric model to noisy data corrupted by outliers

Matching:
identify in a pair of images the image points that correspond to the same 3D point in the scene



An application: template matching

images

Template

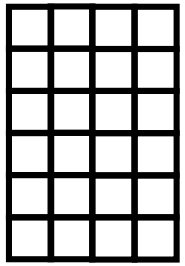
$T \in \mathbb{R}^{r_1 \times c_1}$

Target image

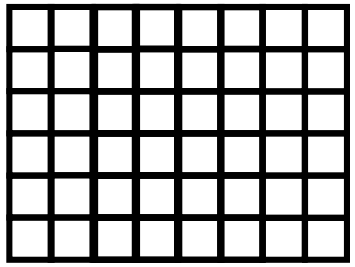
$I_1 \in \mathbb{R}^{r_2 \times c_2}$

d-dimensional vector space

Feature Extraction



$\boldsymbol{X_1} \in \mathbb{R}^{d \times N_1}$



$\boldsymbol{X_2} \in \mathbb{R}^{d \times N_2}$

$(d \ll r \times c)$

Matcher

Identification and localization

Feature Matching

Giacomo Boracchi

The General Approach to Feature Matching

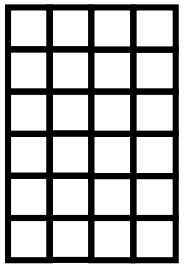
When comparing two images I_1 and I_2 one typically:

- **Extract SIFT** features (keypoint + descriptor) from **independently from I_1 and I_2**
- **For each descriptor** in I_1 we look for the most similar, descriptors in I_2, i.e., we compute **its nearest neighborhood** in I_2 in terms of the **Euclidean distance**
- **Matches are confirmed** when the **distance between the two descriptors is below a threshold**. A matched feature connects both keypoint location and scale.
- **Match clustering / Outlier removal: Geometric criteria to discard wrong matches** can be implemented.

Feature Matching

Input: X_1 and X_2, features extracted from I_1 and I_2

Template features



$X_1 \in \mathbb{R}^{d \times N_1}$

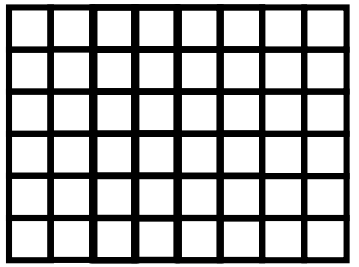
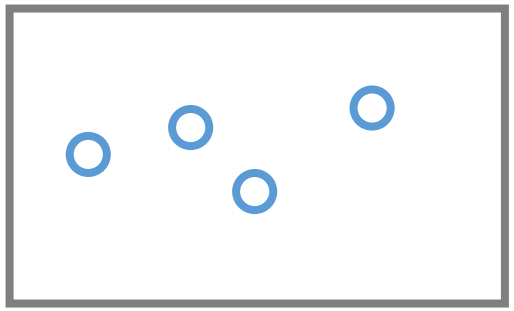
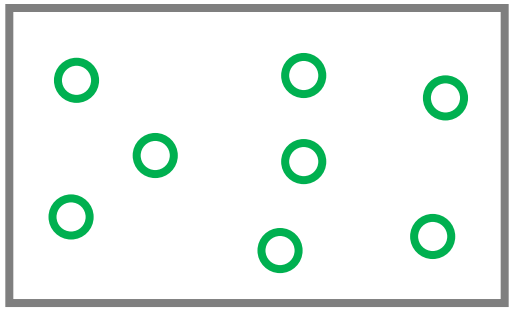


Image features

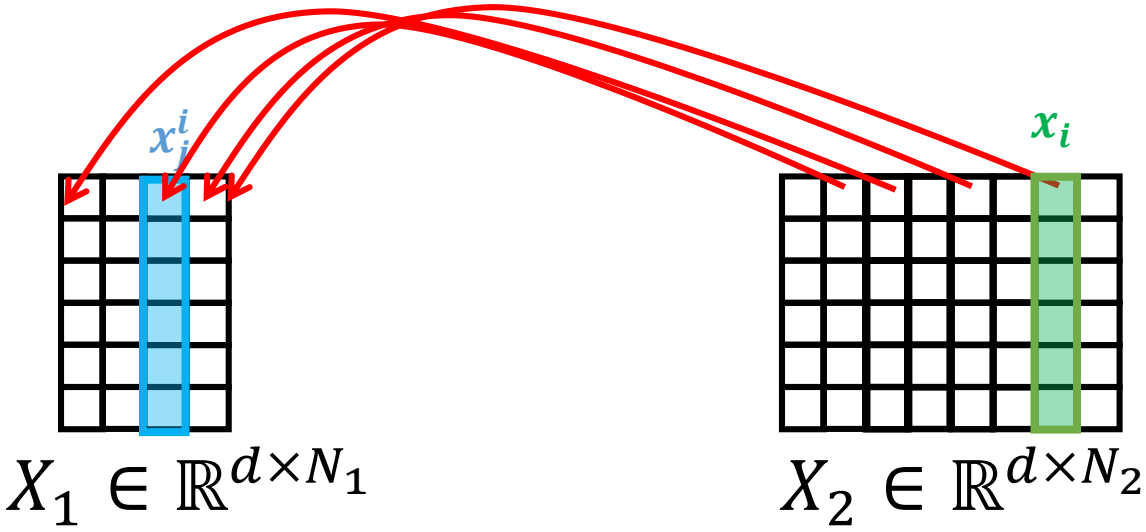
$X_2 \in \mathbb{R}^{d \times N_2}$





Feature Matching

Input: X_1 and X_2, features extracted from I_1 and I_2



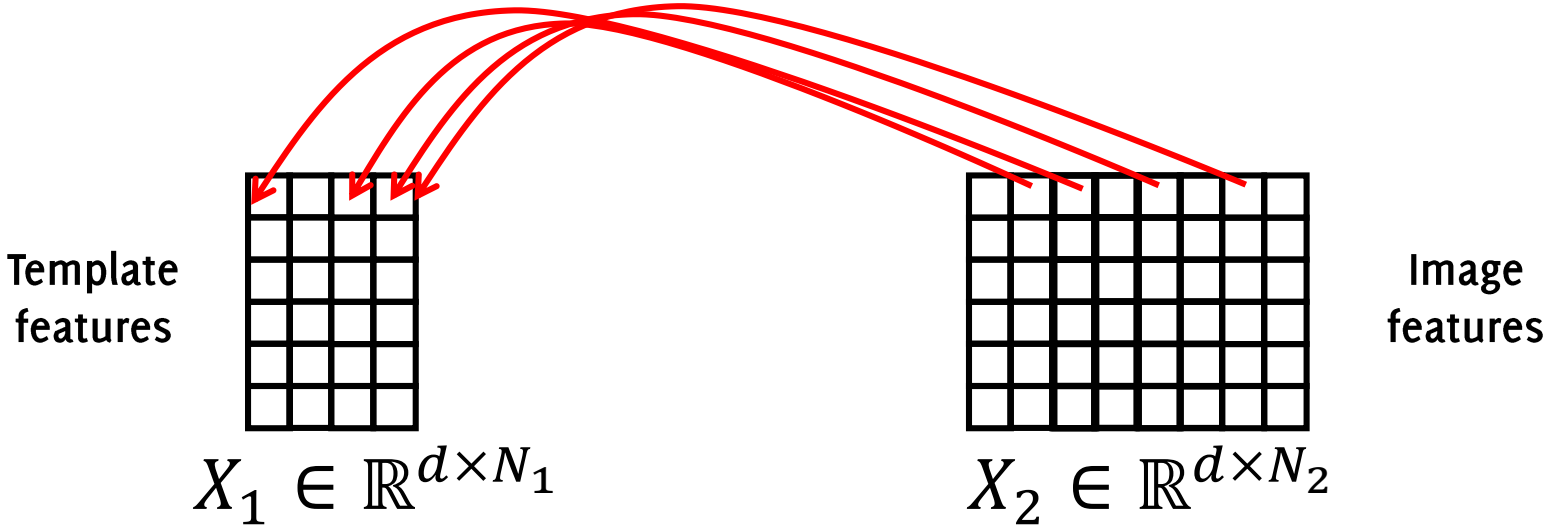
Goal: match image features on the second image to the first image features.

Identify, for each feature $\boldsymbol{x_i} \in X_2$, the feature $\boldsymbol{x_j^i} \in X_1$ minimizing the Euclidean distance

$$\boldsymbol{x_j^i} = \underset{x_j \in X_1}{\operatorname{argmin}} \left(\left|\left| \boldsymbol{x_i} - \boldsymbol{x_j} \right|\right|_2 \right)$$

Feature Matching

Input: X_1 and X_2, features extracted from I_1 and I_2



Goal: match image features on the second image to the first image features.

Identify, fo g the Euclidean distance

$x_j \in x_1$

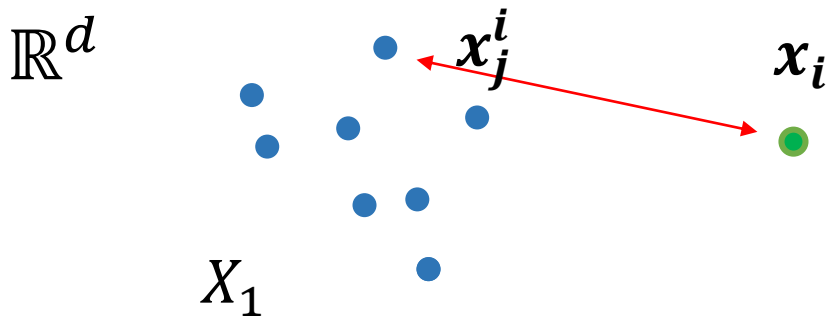
This is the **Nearest-Neighbor Matching Problem**

(on high-dimensional data)

A central problem in CV, ML, document retriveal, data analysis, bioinformatics, compression

Feature Matching

One option is linear search: exhaustively looking for the closest point in a loop



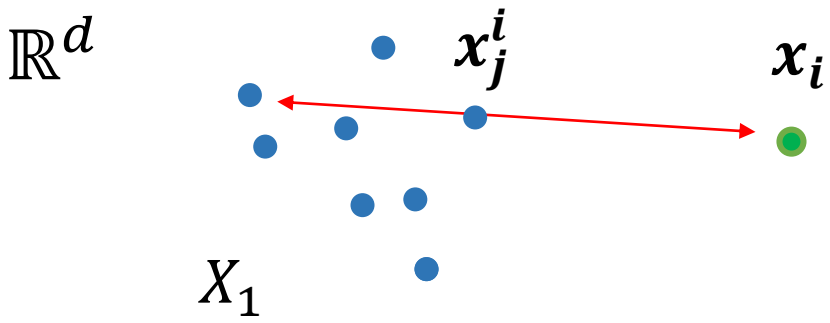
Finding nearest neighbor matches to high dimensional vectors of the training set is **one of the most computationally expensive part of CL and ML algorithms**, it is $O(n)$ but training set are typically very large

This is computationally infeasible in cases of practical interest, and **more efficient solutions are needed**

Muja and Lowe: "Scalable Nearest Neighbor Algorithms for High Dimensional Data" TPAMI 2014

Feature Matching

One option is linear search: exhaustively looking for the closest point in a loop



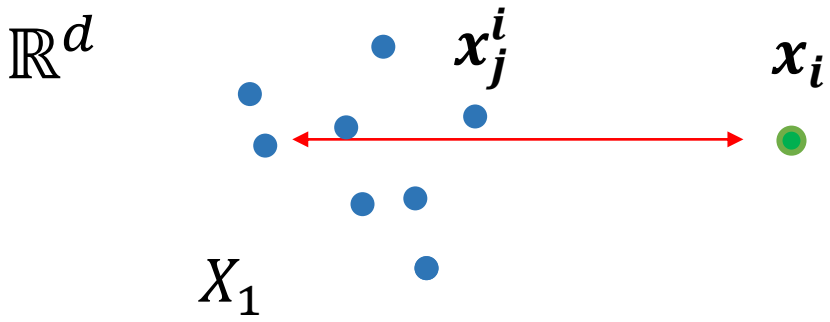
Finding nearest neighbor matches to high dimensional vectors of the training set is **one of the most computationally expensive part of CL and ML algorithms**, it is $O(n)$ but training set are typically very large

This is computationally infeasible in cases of practical interest, and **more efficient solutions are needed**

Muja and Lowe: "Scalable Nearest Neighbor Algorithms for High Dimensional Data" TPAMI 2014

Feature Matching

One option is linear search: exhaustively looking for the closest point in a loop



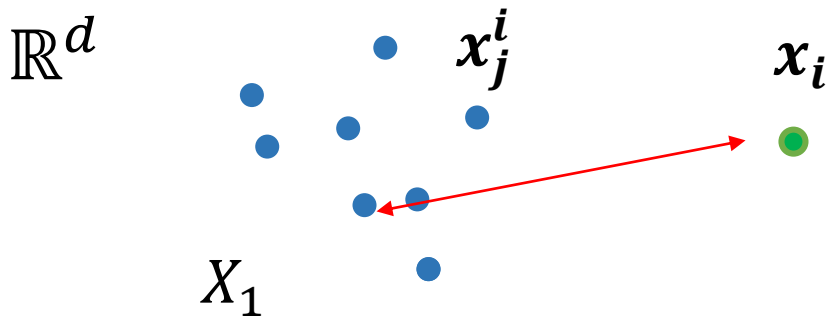
Finding nearest neighbor matches to high dimensional vectors of the training set is **one of the most computationally expensive part of CL and ML algorithms**, it is $O(n)$ but training set are typically very large

This is computationally infeasible in cases of practical interest, and **more efficient solutions are needed**

Muja and Lowe: "Scalable Nearest Neighbor Algorithms for High Dimensional Data" TPAMI 2014

Feature Matching

One option is linear search: exhaustively looking for the closest point in a loop



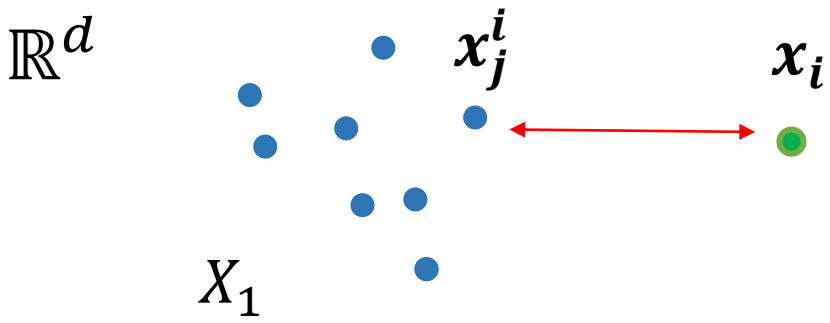
Finding nearest neighbor matches to high dimensional vectors of the training set is **one of the most computationally expensive part of CL and ML algorithms**, it is $O(n)$ but training set are typically very large

This is computationally infeasible in cases of practical interest, and **more efficient solutions are needed**

Muja and Lowe: "Scalable Nearest Neighbor Algorithms for High Dimensional Data" TPAMI 2014

Feature Matching

One option is linear search: exhaustively looking for the closest point in a loop



Finding nearest neighbor matches to high dimensional vectors of the training set is **one of the most computationally expensive part of CL and ML algorithms**, it is $O(n)$ but training set are typically very large

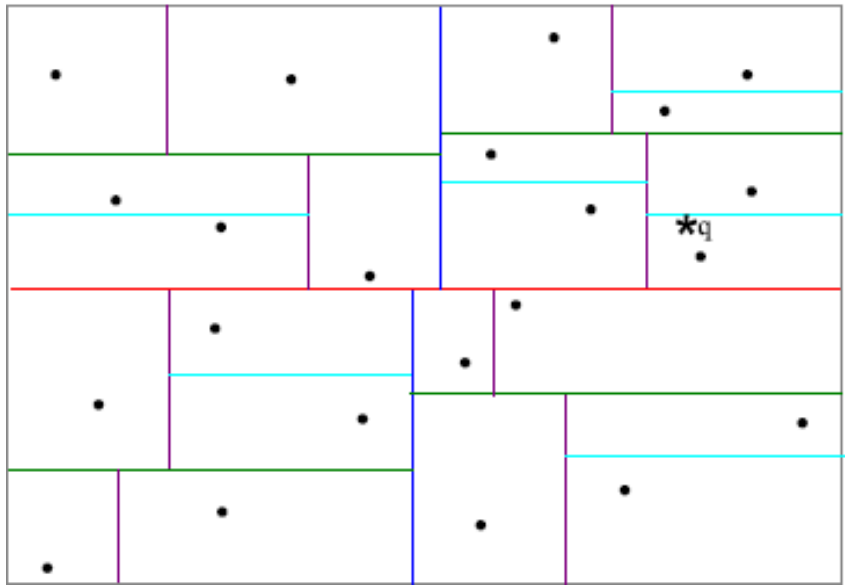
This is computationally infeasible in cases of practical interest, and **more efficient solutions are needed**

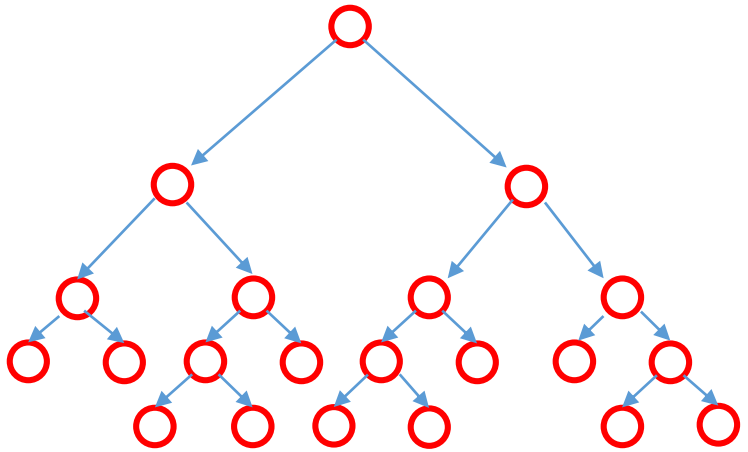
Muja and Lowe: "Scalable Nearest Neighbor Algorithms for High Dimensional Data" TPAMI 2014

K-d trees: rationale

We can do better than linear search, if we organize the descriptors in proper data-structures as K-d trees.

K-d trees are constructed by **recursive binary splits** and allow to efficiently search for nearest neighbor (average complexity $O(\log(n))$)





FLANN: Fast Library for Approximated Nearest Neighborhood

A library which implements advanced approximated seraching methods based on trees, including new algorithms:

Priority search k-means tree algorithms (which are not constructed as splits along the axis)

Hierarchical Clustering Tree (meant for binary features)

Muja and Lowe: “Scalable Nearest Neighbor Algorithms for High Dimensional Data” TPAMI 2014

Pruning Matches

Feature matching overview

Estimating Image Correspondences

- Extract features from each image
  - Keypoint detection
  - Descriptor Computation
- Match features between images
- **Prune matches and then perform triangulation detect objects / stitching:**
  - **Ratio test**
  - **Geometric verification**

Remove matches that are not good enough

The major issue:

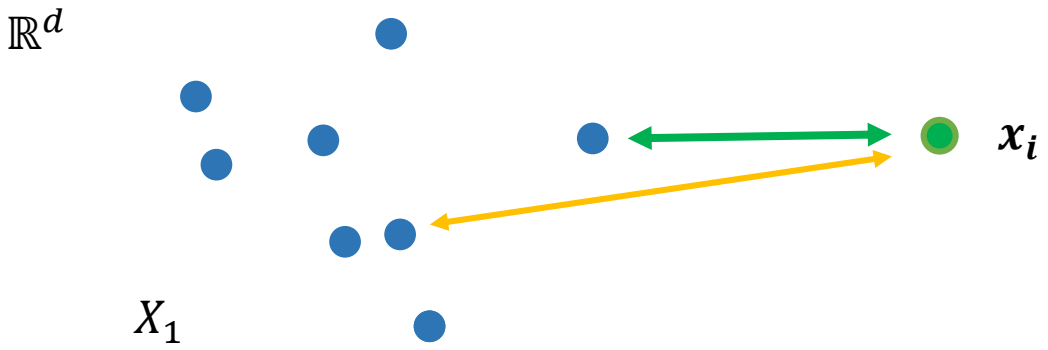
- There does not exist a reference value for a descriptor distance for good/wrong matches
- Desciptor distance can vary a lot from scene to scene, and feature to feature

Matches have to be pruned by looking at their **relative** distance

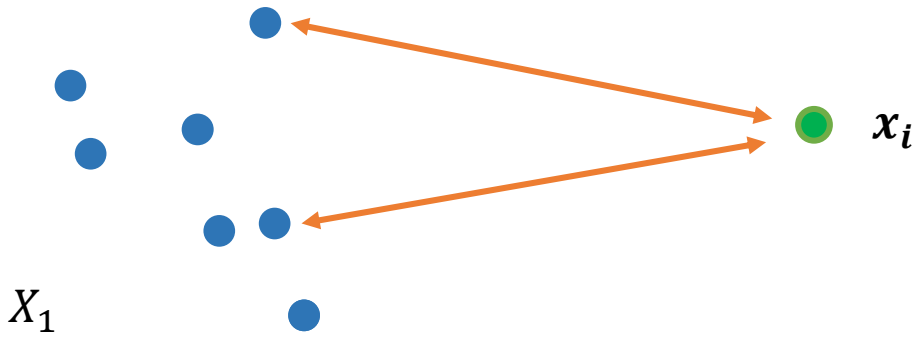
Ratio Test

By nearest neighbor search we get for each image feature $\boldsymbol{x_i} \in X_2$, the closest template feature $\boldsymbol{x_j^i} \in X_1$

Wrong matches $(\boldsymbol{x_i}, \boldsymbol{x_j^i})$ need still to be rejected.



✓ There is a clear closest match



✗ Matches are ambiguous, the distance to the best match is similar to the one with the second-best match

[Lowe 2004] Lowe "Distinctive Image Features from Scale-Invariant Keypoints" IJCV 2004

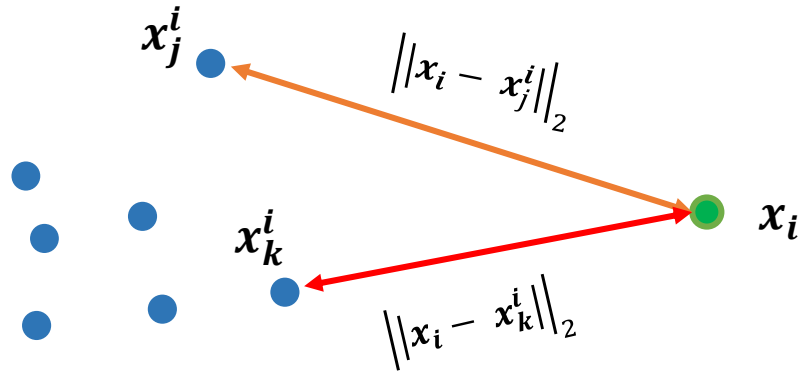
Ratio Test

During matches, retrieve the 2-NN neighbor of each $\boldsymbol{x_i}$, i.e.

$$\left(\boldsymbol{x_i}, \boldsymbol{x_j^i}\right) \text{ and } \left(\boldsymbol{x_i}, \boldsymbol{x_k^i}\right)$$

Discard keypoints where

$$\frac{\left|\left|\boldsymbol{x_i} - \boldsymbol{x_k^i}\right|\right|_2}{\left|\left|\boldsymbol{x_i} - \boldsymbol{x_j^i}\right|\right|_2} > 0.8$$

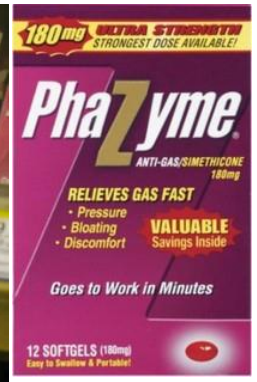


This criteria **discards matches where the second nearest neighbor is very close to the first**. These are:

- Ambiguous matches
- False matches arising from background clutter

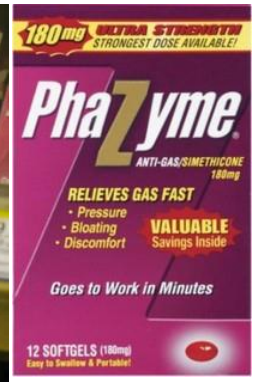
[Lowe 2004] Lowe "Distinctive Image Features from Scale-Invariant Keypoints" IJCV 2004

Application: template matching



template

Application: template matching



template

Giacomo Boracchi

Application: template matching

template

Application: template matching

Repeated structures

template

Matches preserved by the ratio test

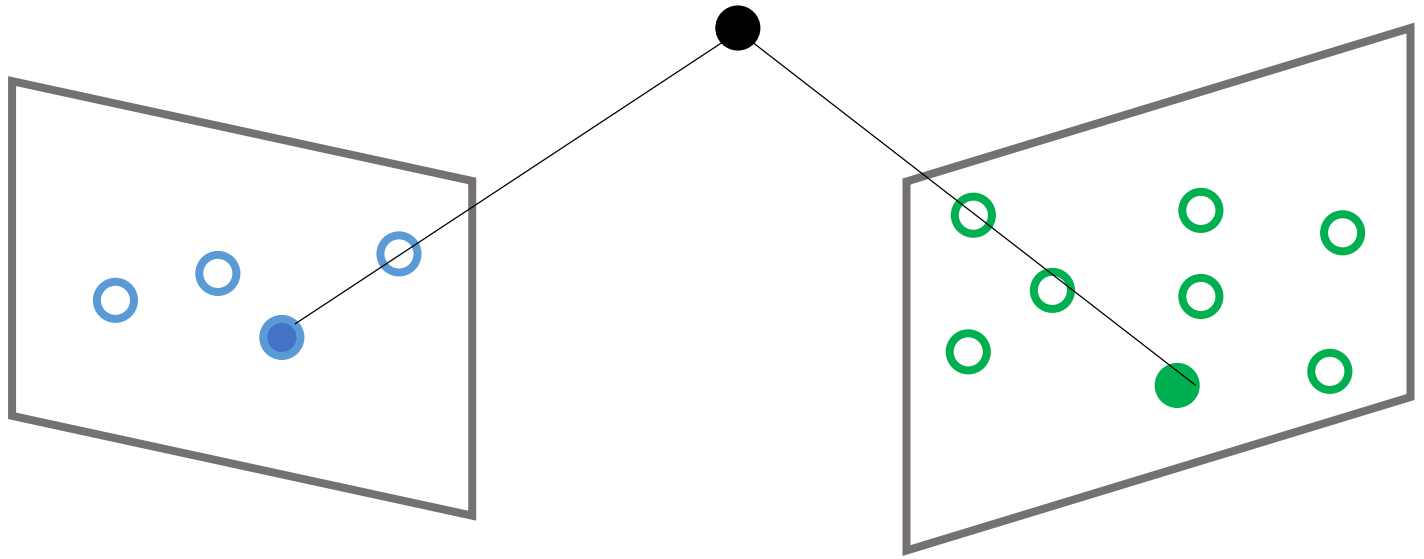
Geometric verification

Until now we have searched for corresponding points leveraging on image appearance. But we can also exploit 3D geometric constraints

Do these two points look similar?

Geometric verification

Until now we have searched for corresponding points leveraging on image appearance. But we can also exploit 3D geometric constraints

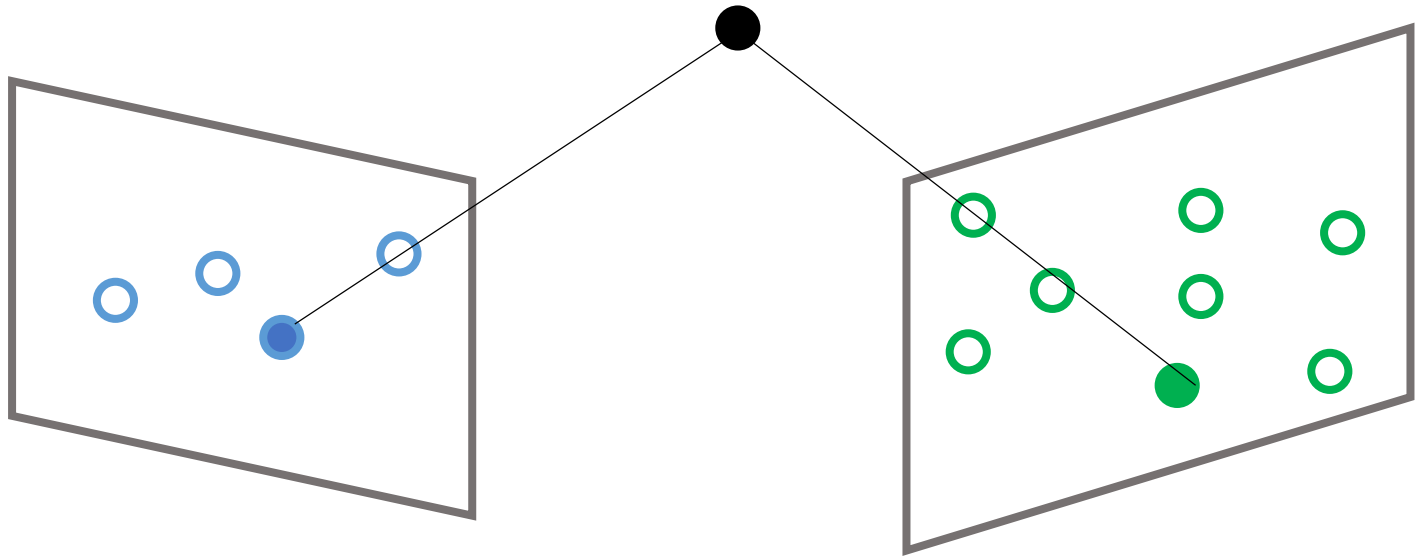


3D geometry

Do these two points look similar?
Can these two points be the image of the same 3D point in the scene?

Geometric verification

Until now we have searched for corresponding points leveraging on image appearance. But we can also exploit 3D geometric constraints



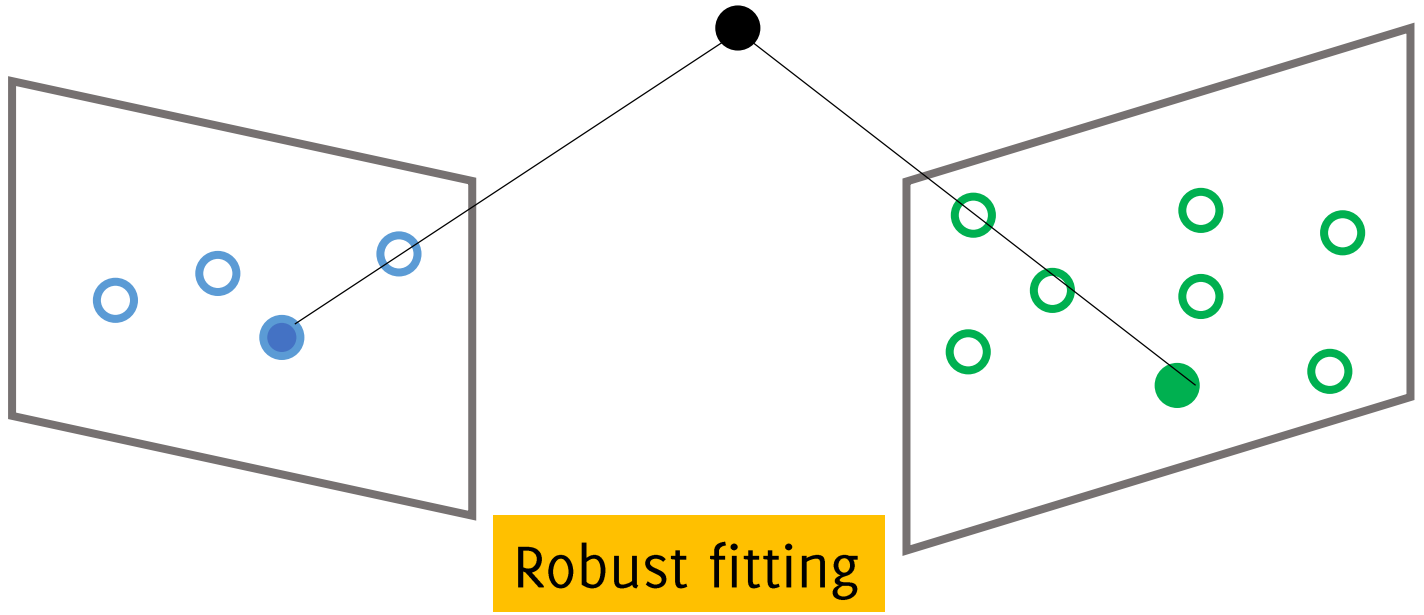
3D geometry

Can these two points be the image of the same 3D point in the scene?

Pairs of corresponding points must satisfy epipolar geometry

Geometric verification

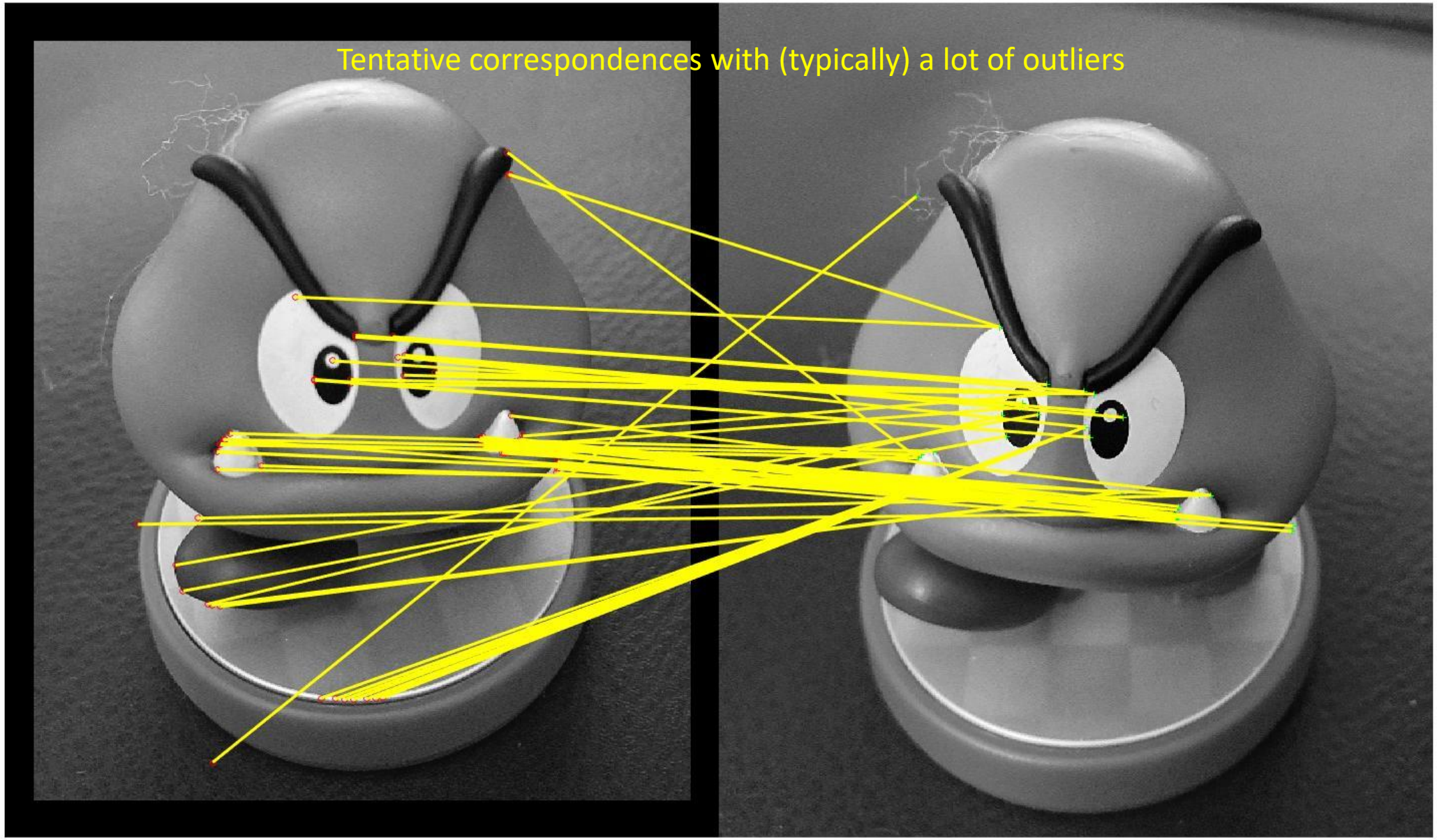
Until now we have searched for corresponding points leveraging on image appearance. But we can also exploit 3D geometric constraints



3D geometry

Robustly fit (e.g. via RanSAC) a fundamental matrix on matches and keep only the inliers

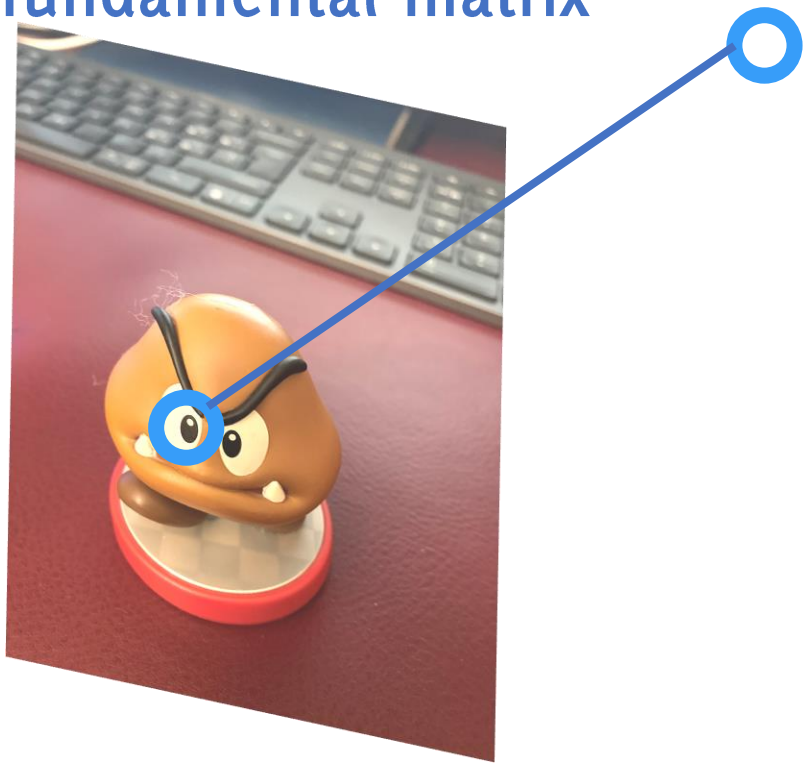
If the scene is planar or the camera movement is pure rotational an homography can be used to validate matches.

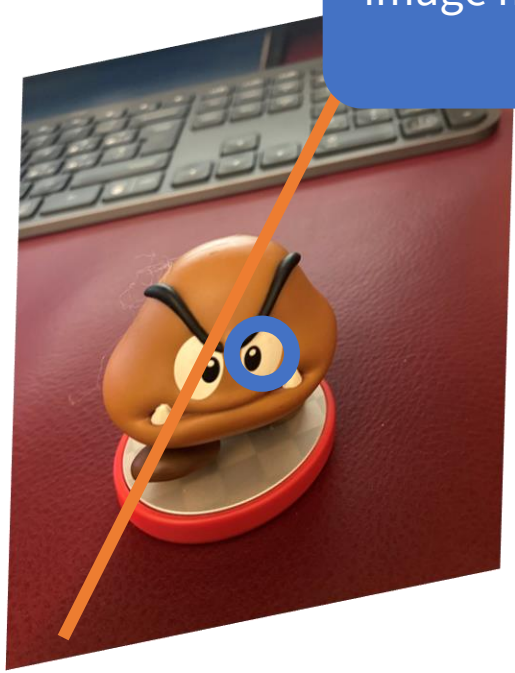

Tentative correspondences with (typically) a lot of outliers

Luca Magri

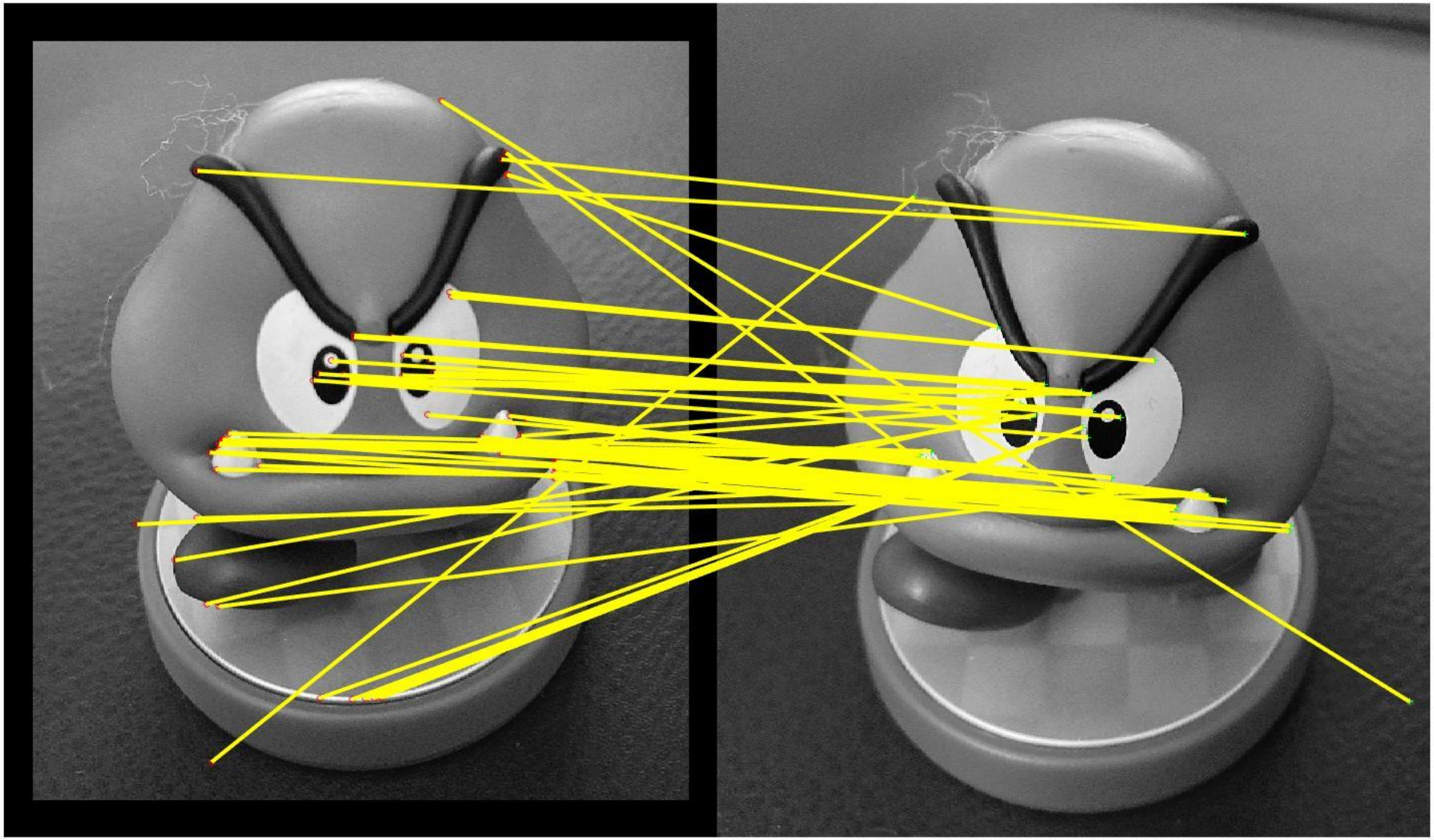
Fit the epipolar matrix

You can fit a parametric model to disambiguate between inliers and outliers: **fundamental matrix**



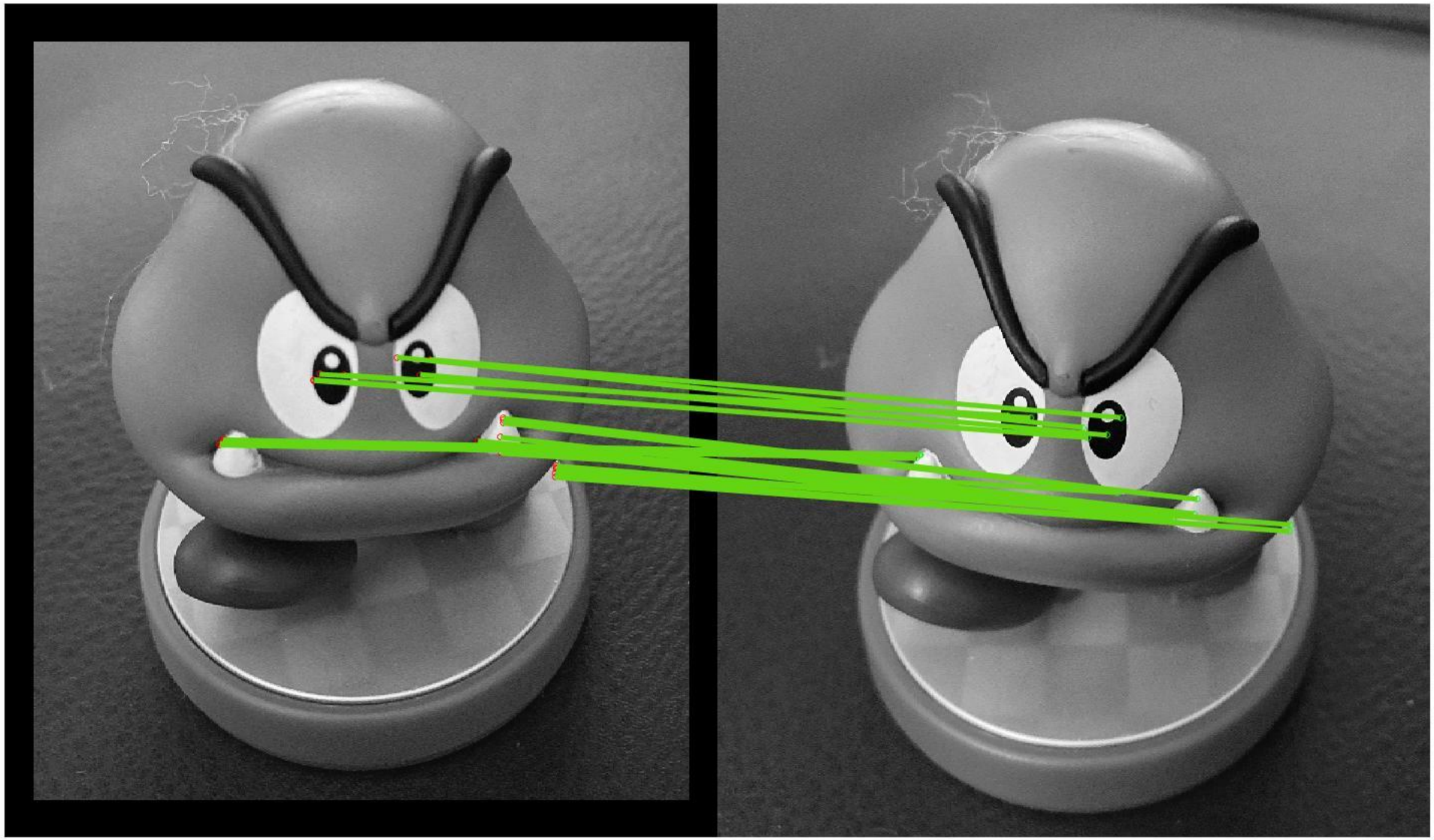


The image point on the right image must lie on the epipolar line



$X = \{(x_{\ell i}, x_{ri})\ i = 1, \ldots, n\}$ data are set of correspondences

F the model is a fundamental matrix



$X = \{(x_{\ell i}, x_{ri})\ i = 1, \ldots, n\}$ data are set of correspondences

F the model is a fundamental matrix

Luca Magri

In case of planar objects

The distance ratio test have discarded many false matches

Still, many matches belong to different objects.

We need to cluster features belonging to the same object

To this purpose **we need a model:**

- For instance: *"planar objects seen from two different views are related by an homography"*

Homographies are $H \in \mathbb{R}^{3\times 3}$ linear transformation from

$$H\colon \mathbb{R}^2 \to \mathbb{R}^2$$

which can be computed by 4 matches.

Adding Homography Constraint

The whole processing can be iterated to match more template in the image.

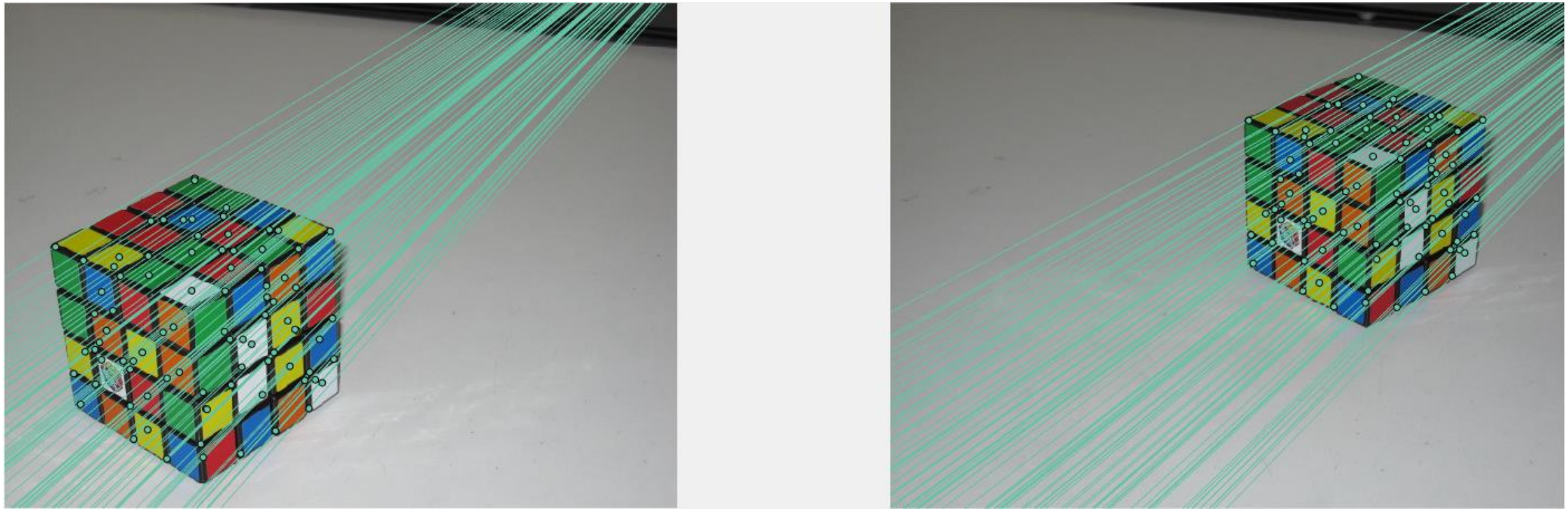
Adding Homography Constraint

The whole processing can be iterated to match more template in the image.

Each estimated homography can be used to rectify the each deteceted region and make it pixel-wise comparable with the template

Robust Fitting

Example: Estimating Fundamental Matrix

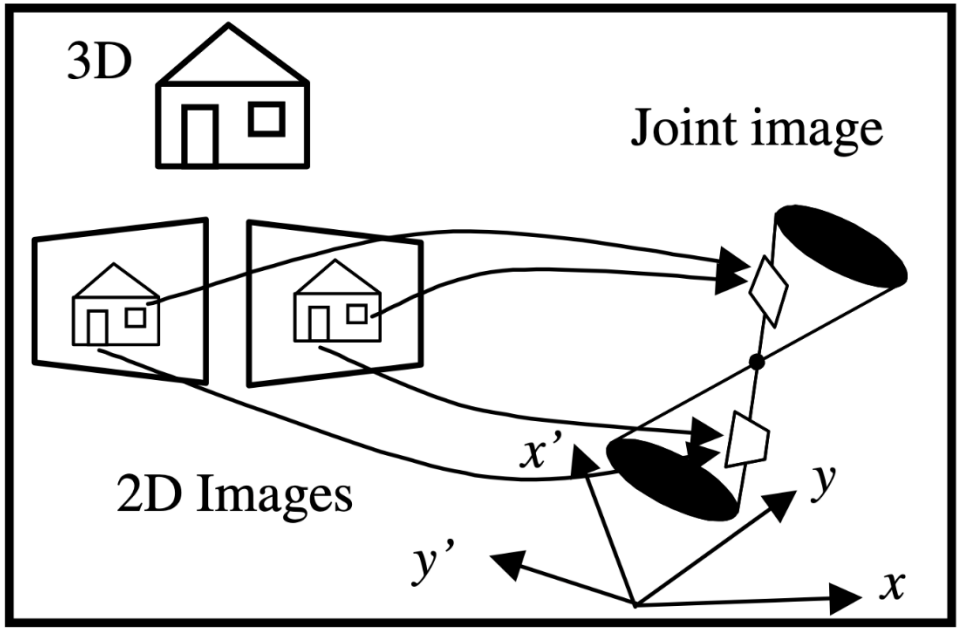


$\theta = F;$

$X = \{x_1, x_2\} \quad x_2^t F x_1 = 0$

Example: Estimating Fundamental Matrix

It can be thought as a **generalization of a conic fitting problem** in the joint image space of pairs of correspondences.



Each match is a pair of points and defines a vector in $\mathbb{R}^4$.
For each correct match, the pair of corresponding points satisfies the epipolar constraint (bilinear relation) and lies on a 4D cone that can be fitted from 8 correspondences.

$\theta = F;$
$X = \{x_1, x_2\} \quad x_2^t F x_1 = 0$

Image from Anandanand Avidan. Integrating Local Affine into Global Projective Images in the Joint Image Space. ECCV 2020

Fitting

Input: X Data

Output: θ parameters of the model that describes matches

Our aim is to fit θ to the data X to recover a “geometric pattern” described by the model (structure)

In the previous example we want to recognize those correspondences that are compatible with a rigid ego-motion (of the scene/of the cameras).

This framework can be found in several computer vision applications.

Example: Estimating Homographic Transformations

$$\theta = H;$$

$$X = \{x_1, x_2\} \Leftrightarrow x_2 \sim Hx_1 \Leftrightarrow x_2 \times Hx_1 = 0$$

Example: Line Fitting for Vanishing Point Estimation

$$\theta = (a, b, c),\ X = \{(x, y)\}$$
$$ax + by + c = 0$$

Example: Conic Fitting

$$\theta = (r, c_x, c_y); X = \{(x, y)\}$$

$$(x - c_x)^2 + (y - c_y)^2 = r^2$$

Ordinary Least Square

Least Square Regression

Given a set of N points (or matches..)

$$X = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

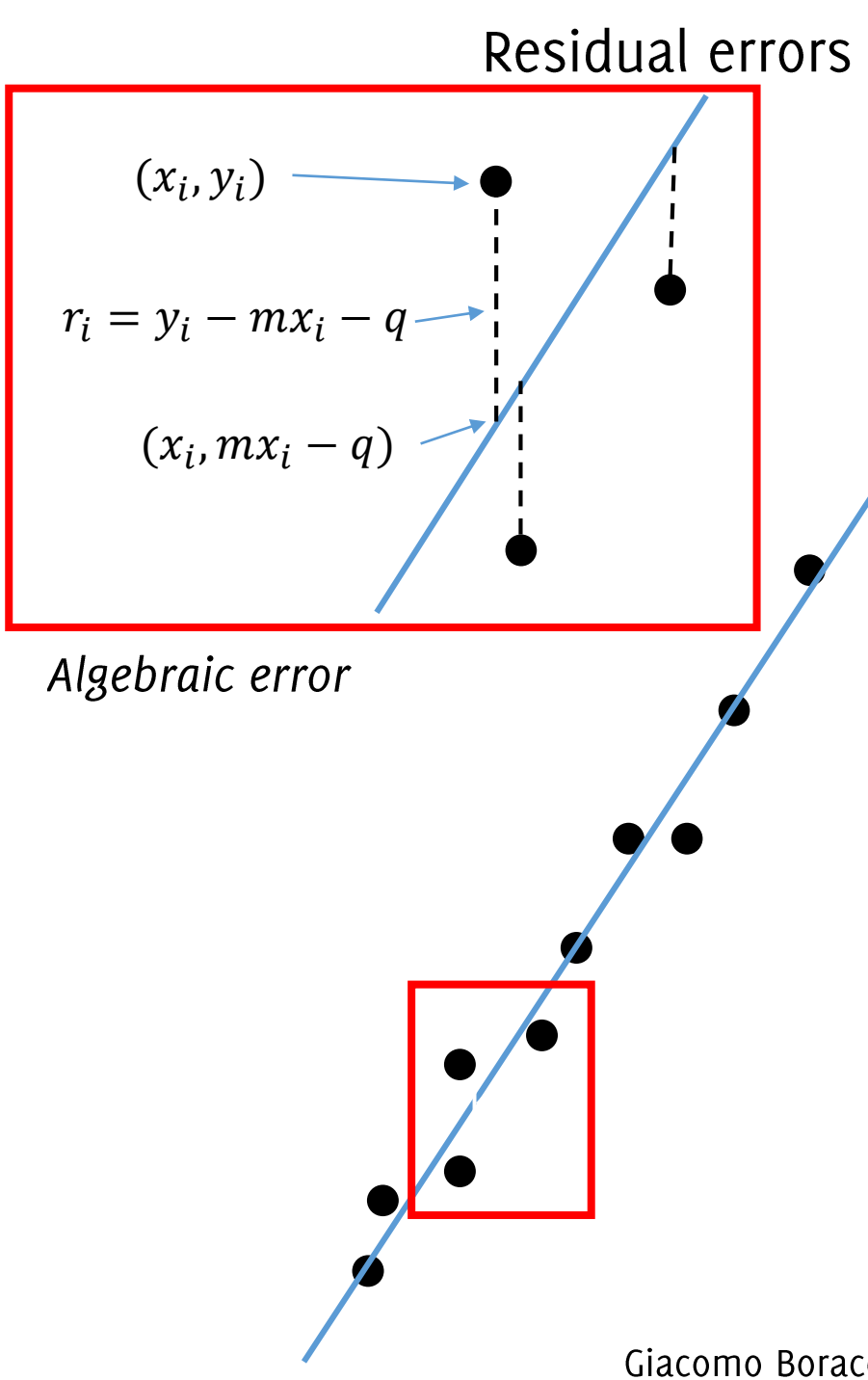
Given a parametric model

$$y = mx + q$$

Estimate the parameters $\theta = \{m, q\}$ yielding the **best fit**

The best fit is the one **minimizing some residual error over** X

$$E = \sum_{i=1}^{N} (r_i)^2 = \sum_{i=1}^{N} (y_i - mx_i - q)^2$$



This error does not make sense
when the line is vertical

Minimizing the point-line distance

Given a set of N points (or matches..)

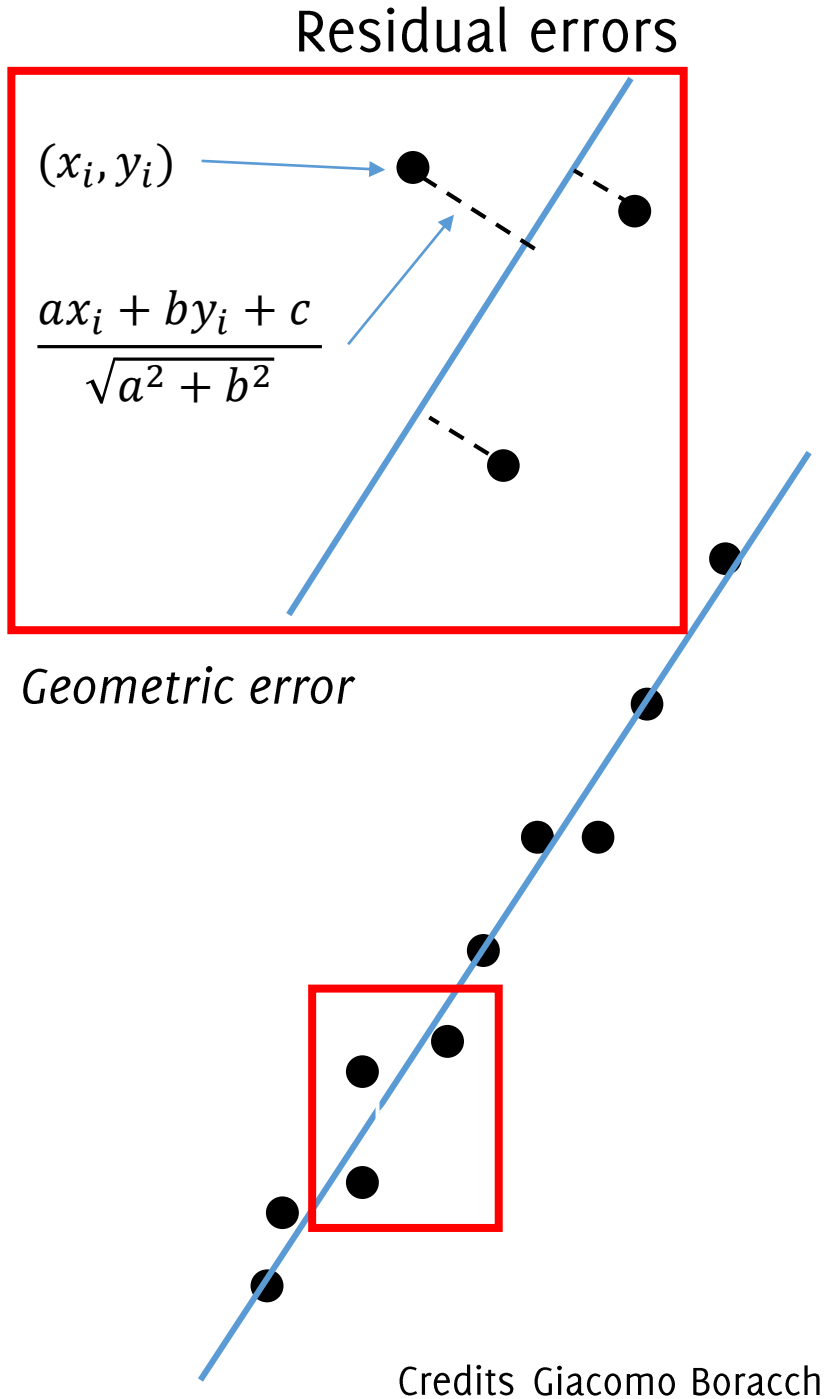
$$X = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

Given a parametric model

$$ax + by + c = 0$$

Estimate the line parameters $\theta = \{a, b, c\}$ **minimizing** the residual error

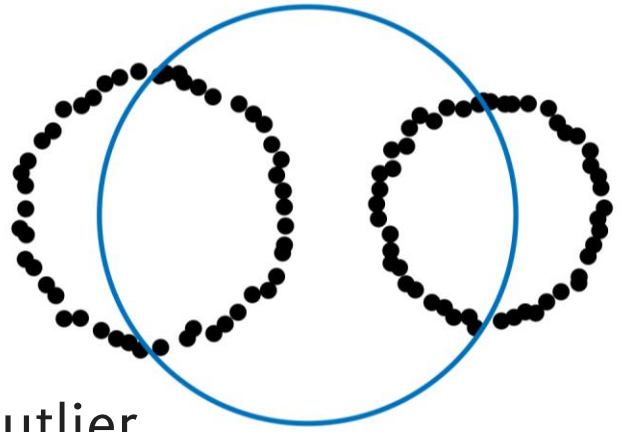
$$E = \sum_{i=1}^{N} (ax_i + by_i + c)^2$$



Credits Giacomo Boracchi

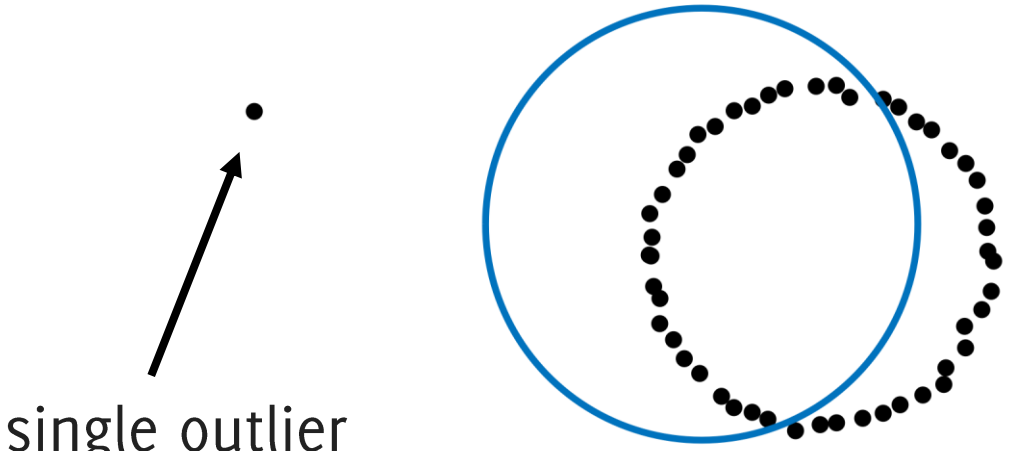
Least squares breaks down

Break down point: the proportion of incorrect observations that can be handled before giving an arbitrarily large uncorrect result.

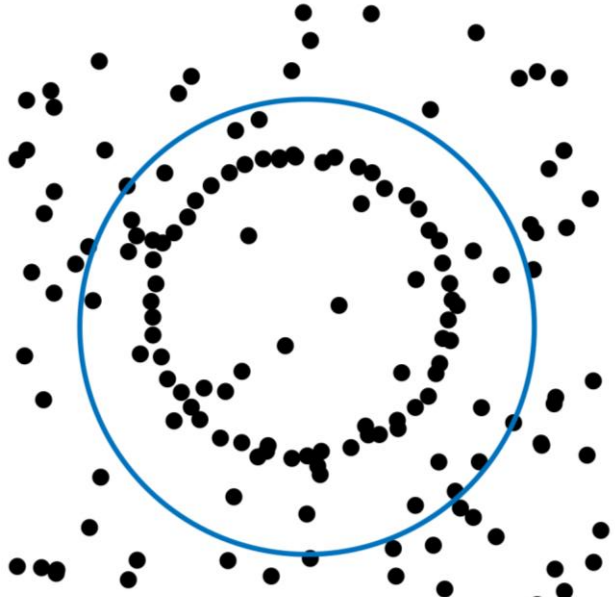


pseudo-outlier

✗ Least squares has 0% breakdown point (the outlier might be arbitrarily large)



single outlier



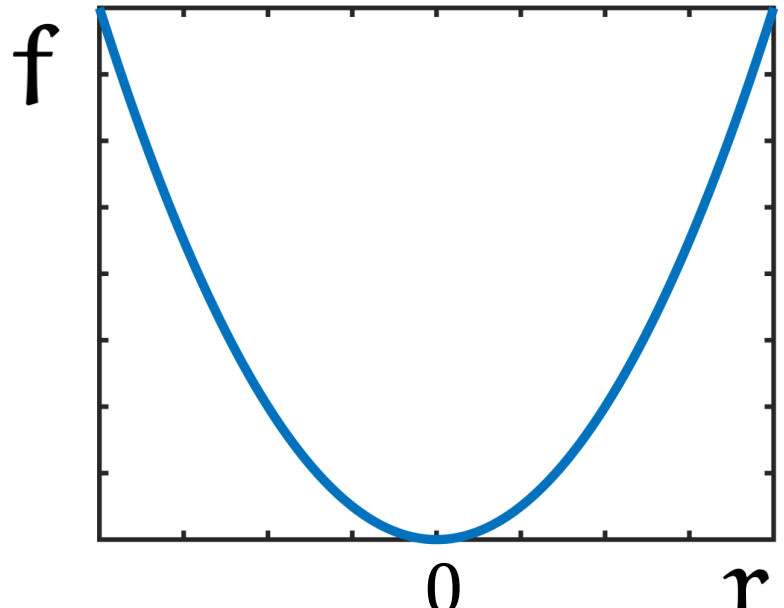
unstructured outliers

Robustness to Outliers

The loss function: least square

The loss so far is the sum of a function of all the residuals

$$E = \sum_{i=1}^{N} f(r_i), \text{where } f(r_i) = r_i^2$$

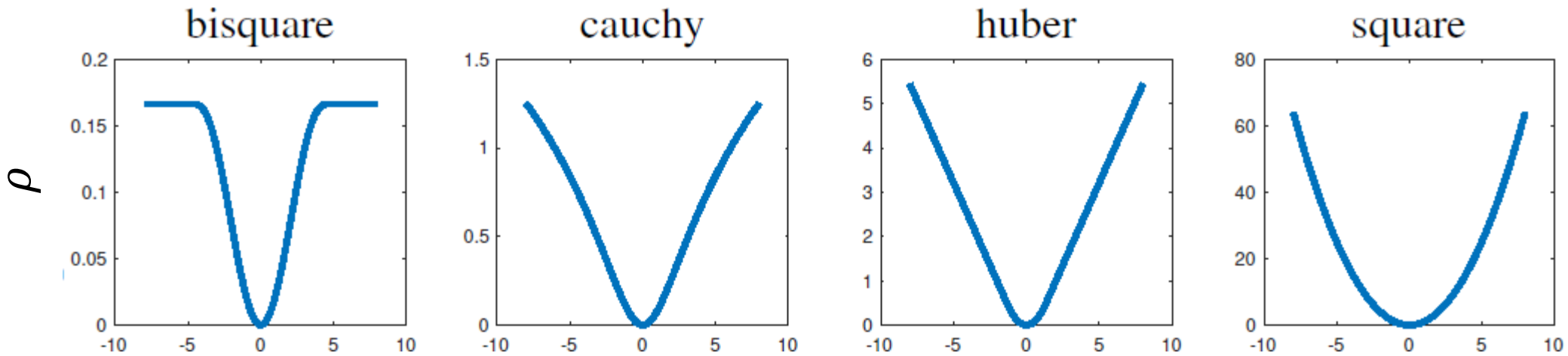


M-Estimator

Replaces the squared loss with a different loss function which penalizes less large residual values (deemed to correspond to outliers)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \rho(r_i(\theta))$$

Where ρ a symmetric, positive-definite function having a unique minimum in zero



Andrea Fusiello, Elaborazione delle Immagini: Visione Computazionale, http://www.diegm.uniud.it/fusiello/index.php/Visione_Computazionale

RanSaC

Robust single model fitting: consensus maximisation

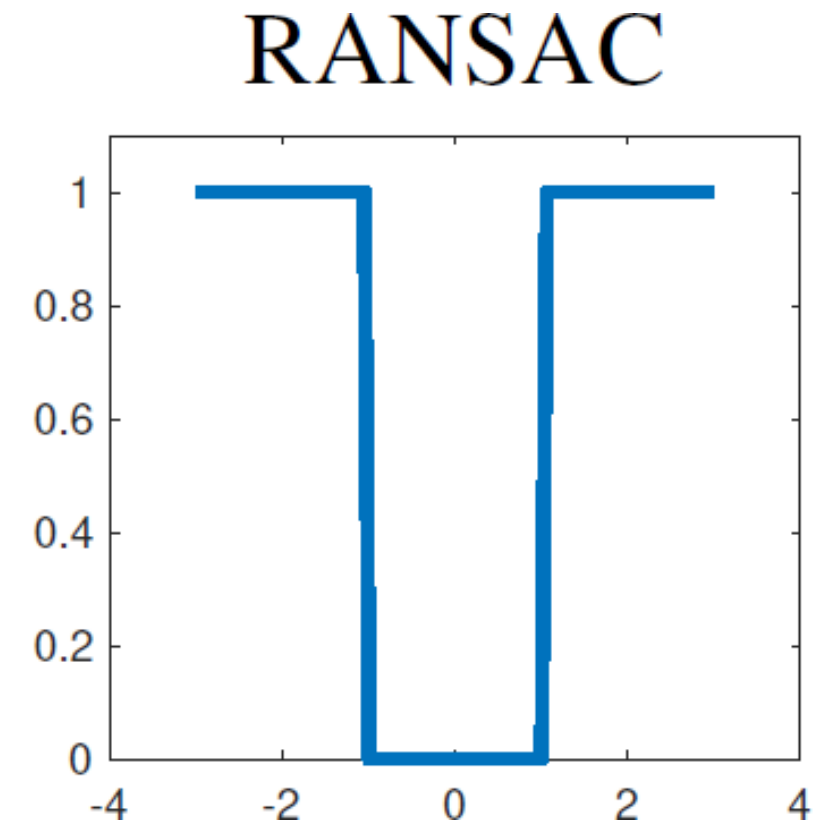
Instead of minimizing the residuals, **maximize the consensus.** Define:

- an inlier threshold $\epsilon > 0$
- a consensus function $\tilde{f}$ which is

$$\tilde{f}(r_i) = \begin{cases} 1, & r_i \leq \epsilon \\ 0, & r_i > \epsilon \end{cases}$$

Identify $\hat{\theta}$ that maximizes the consensus

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} 1 - \tilde{f}(r_i)$$



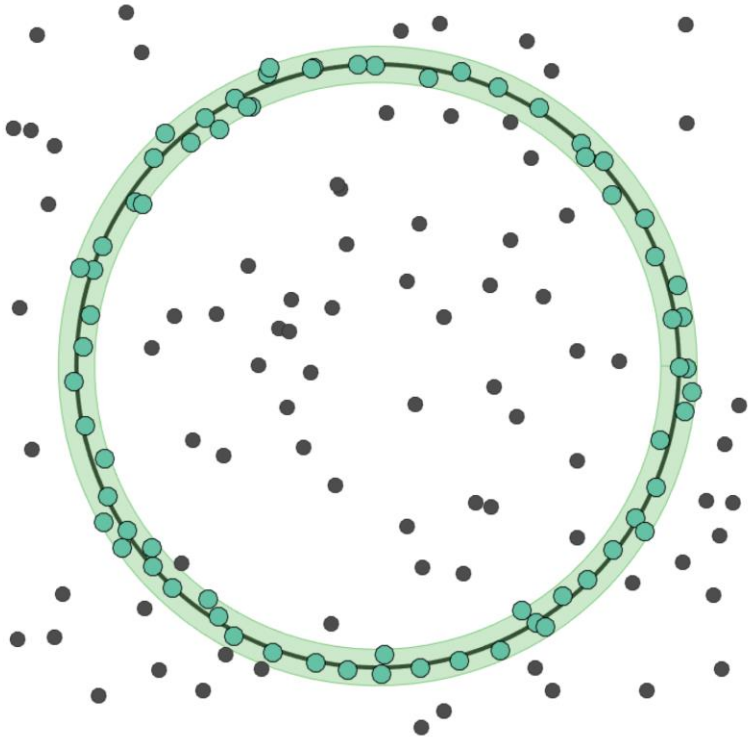
The Conensus Set

Consensus set

$$\mathrm{CS}(\theta, \epsilon) = \{x_i \mid r_i \leq \epsilon\}$$

Being $r_i = r(x_i, \theta)$, the residual of the model θ at a point x_i

The larger the consensus set $\mathrm{CS}(\theta, \epsilon)$, the better the model θ



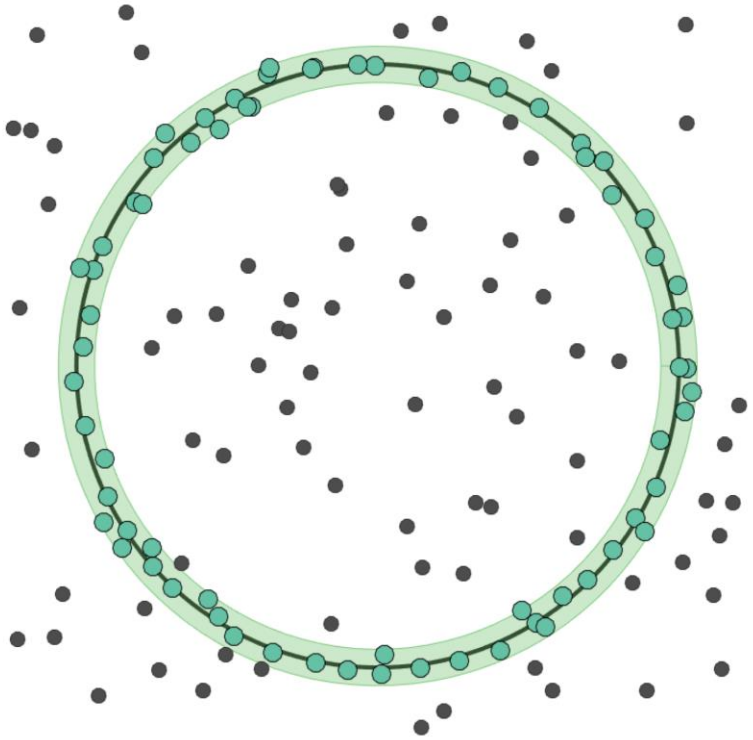
The Conensus Set

Consensus set

$$\mathrm{CS}(\theta, \epsilon) = \{x_i \mid r_i \leq \epsilon\}$$

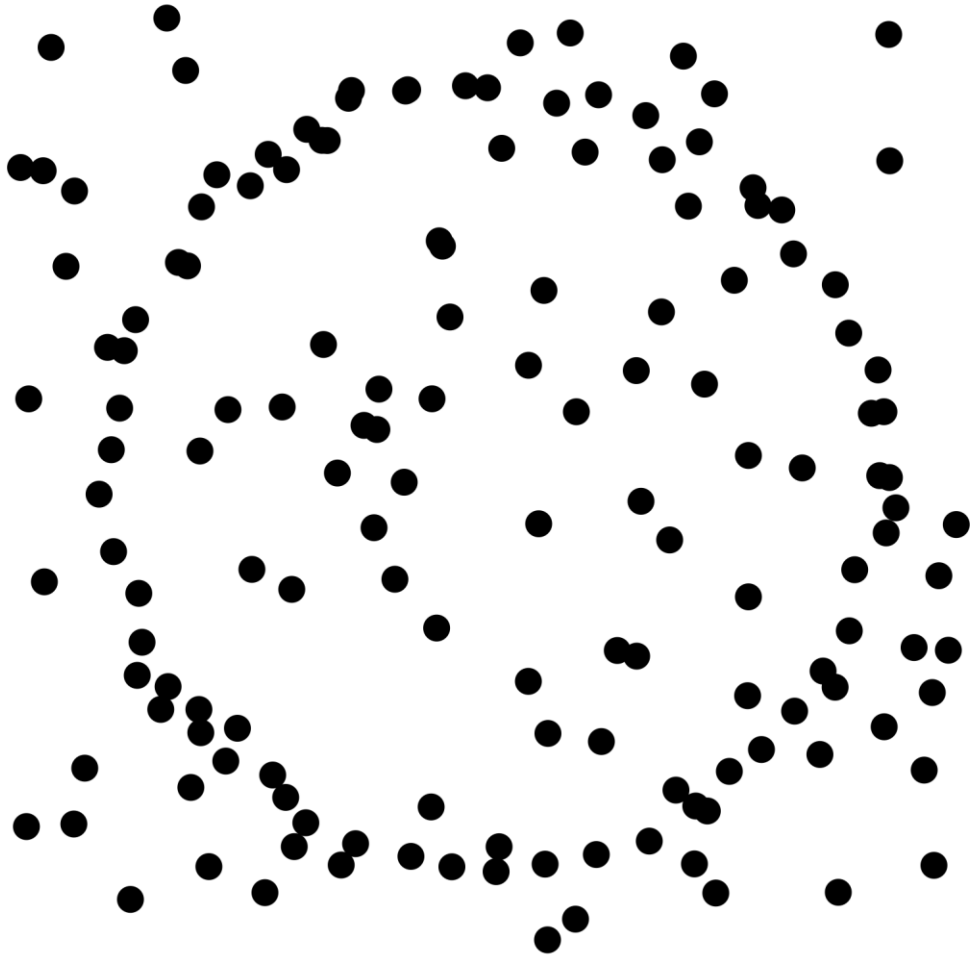
Being $r_i = r(x_i, \theta)$, the residual of the model θ at a point x_i

The larger the consensus set $\mathrm{CS}(\theta, \epsilon)$, the better the model θ



We have been fitting lines so far, but **everything** holds for any parametric model (e.g. circle, conics, homographies, fundamental matrices)

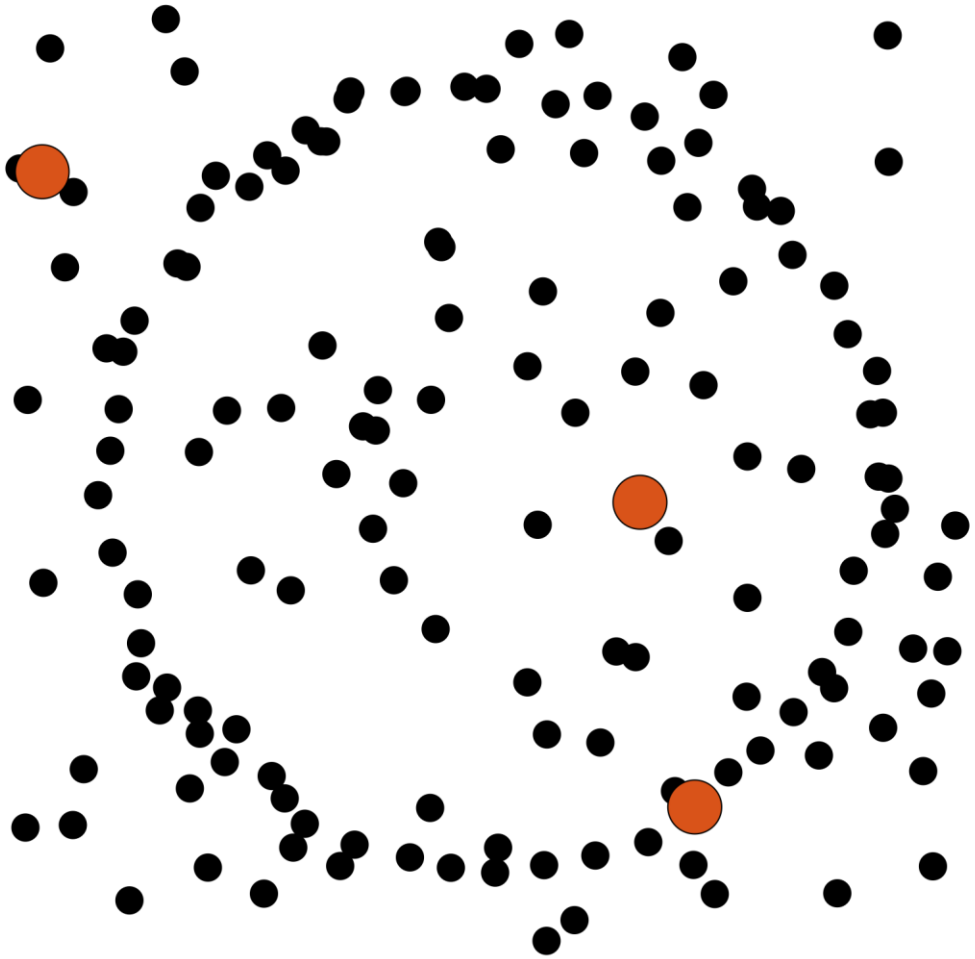
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

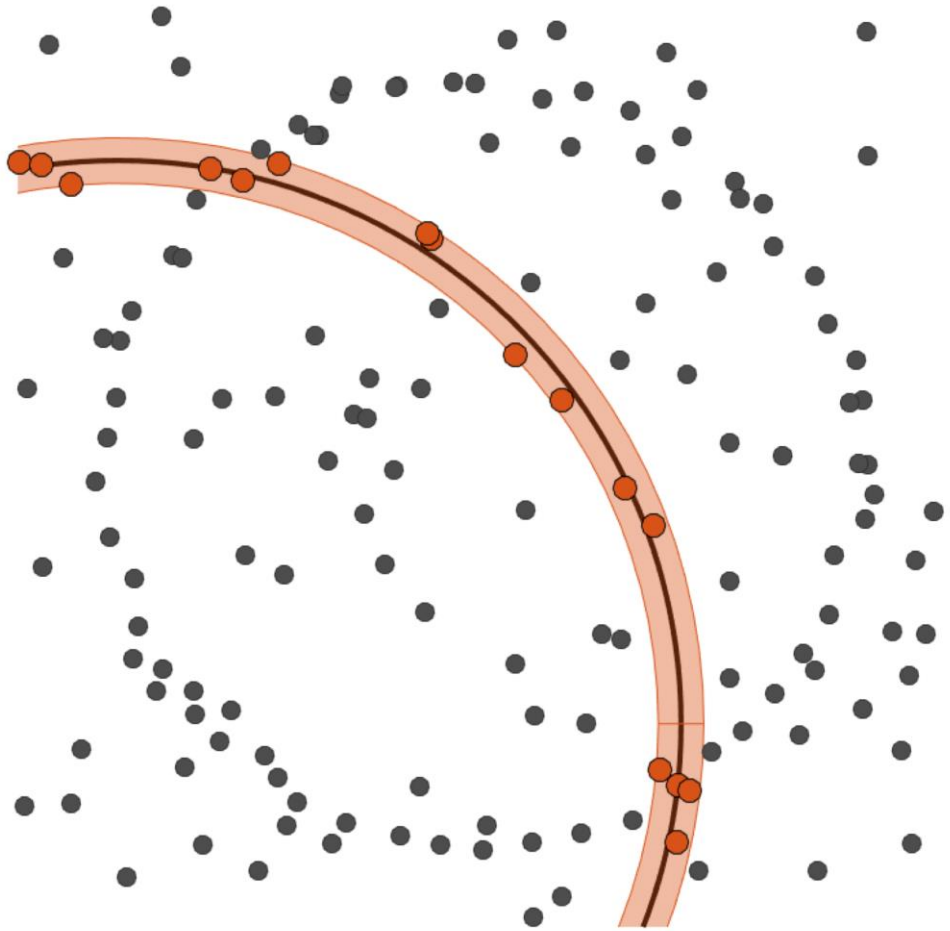
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

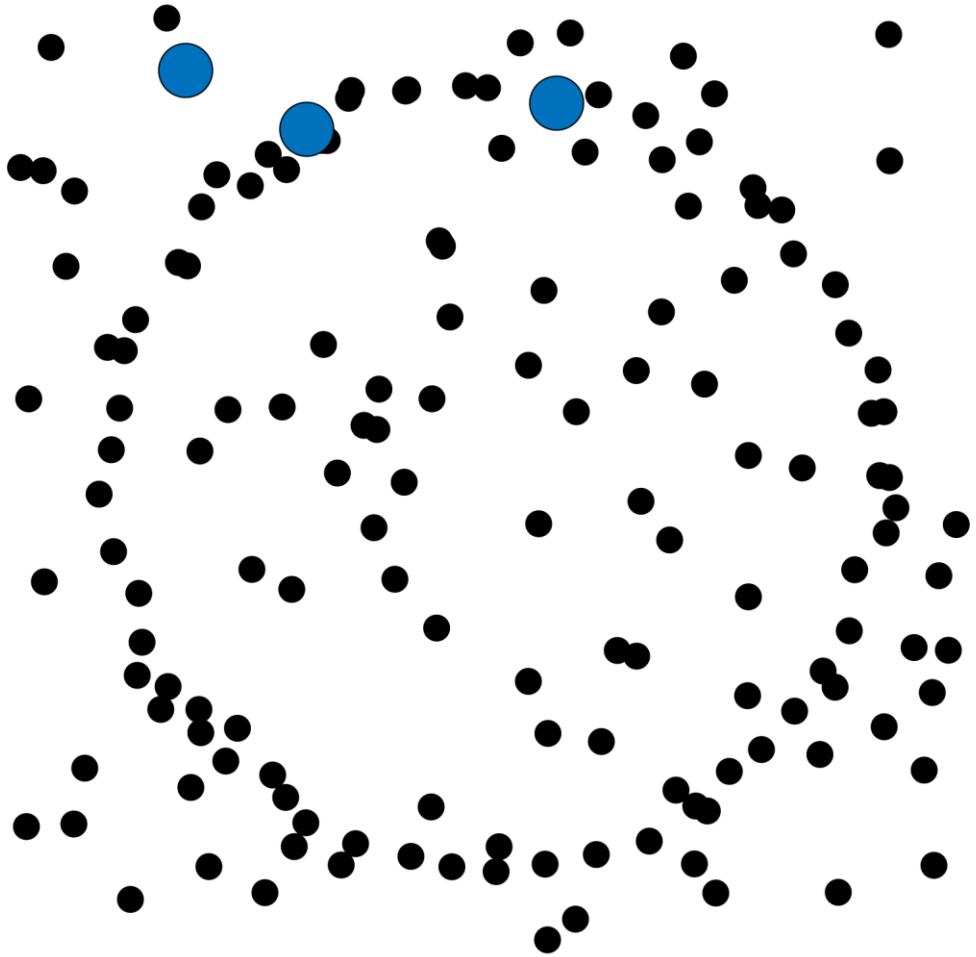
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

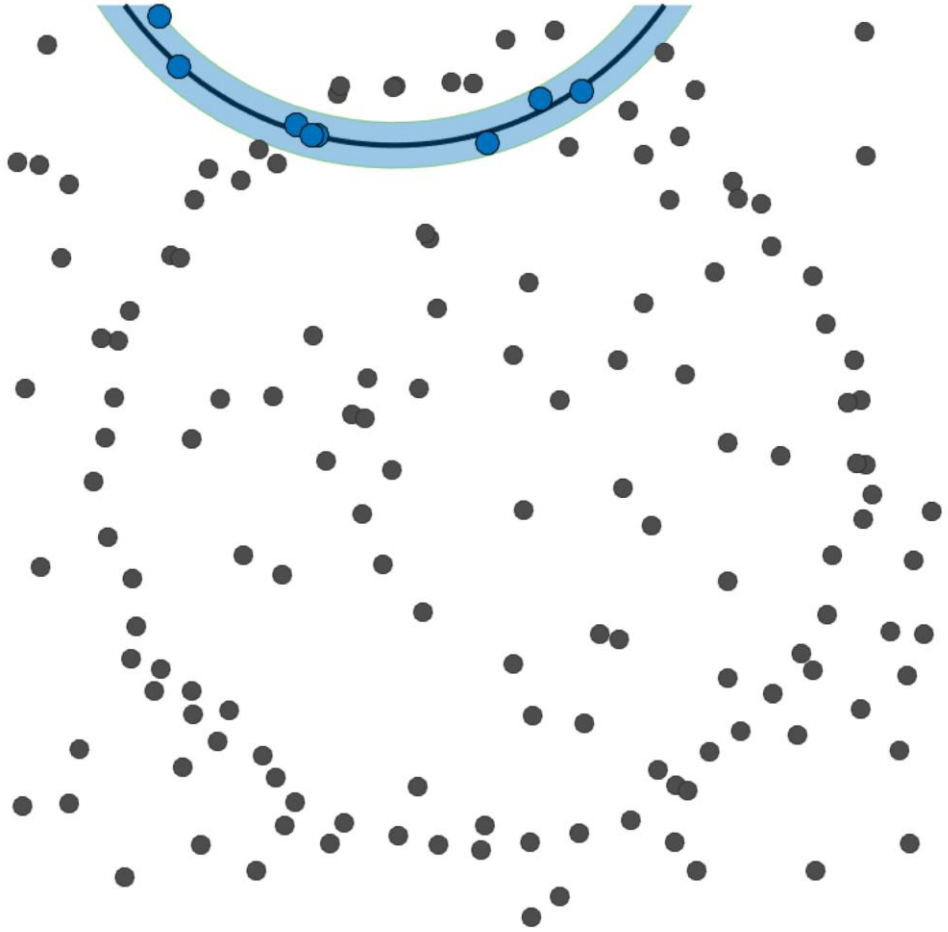
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

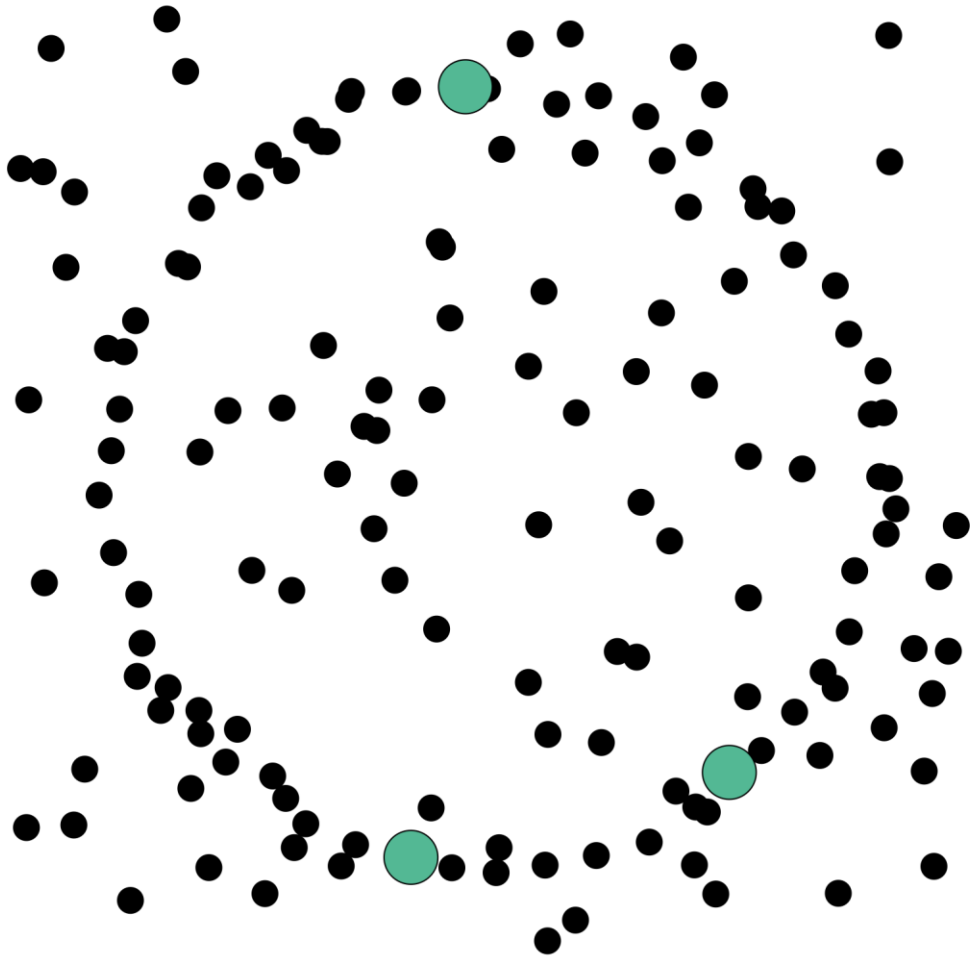
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

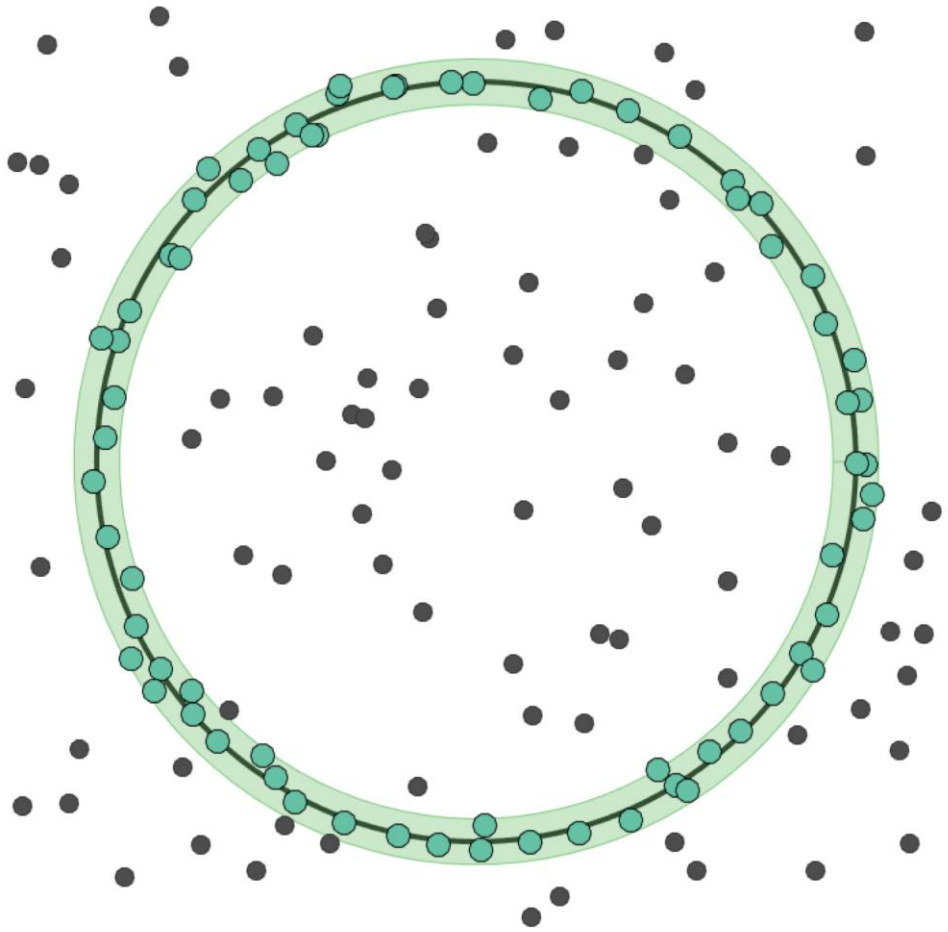
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

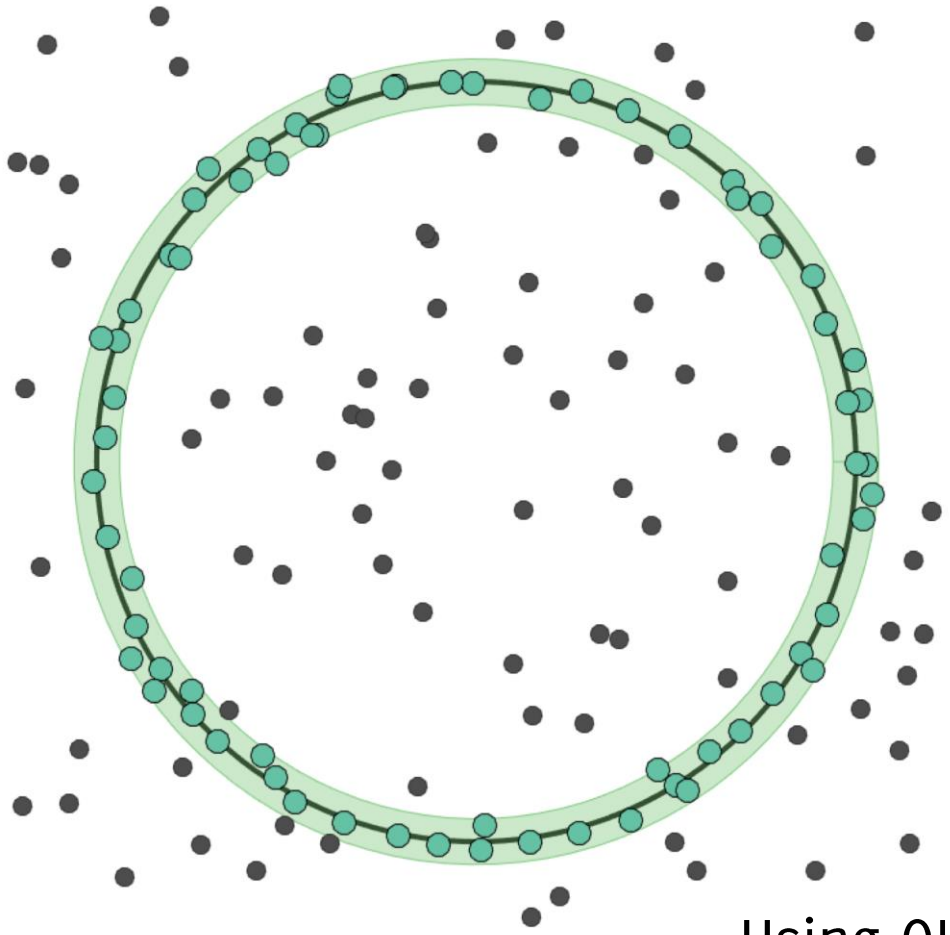
Randomized Sample Consensus [Fischler and Bolles 1981]



Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

Randomized Sample Consensus [Fischler and Bolles 1981]

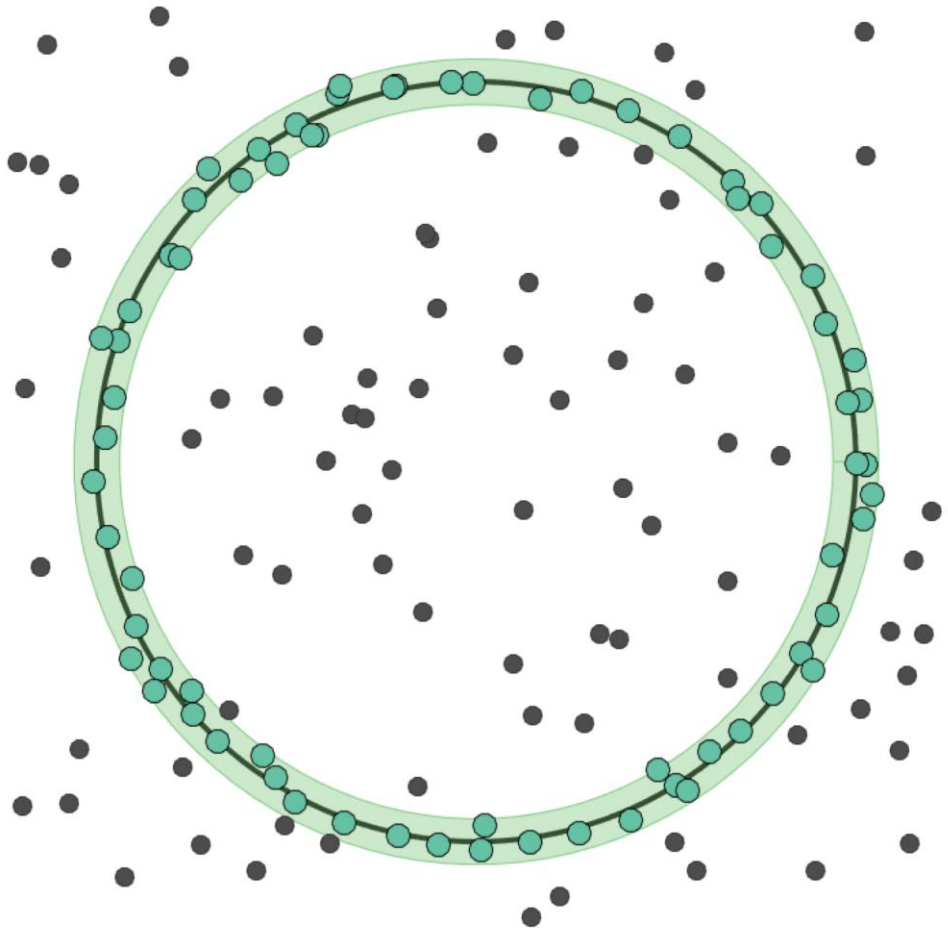


Using OLS,
Increase stability of
the results

Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

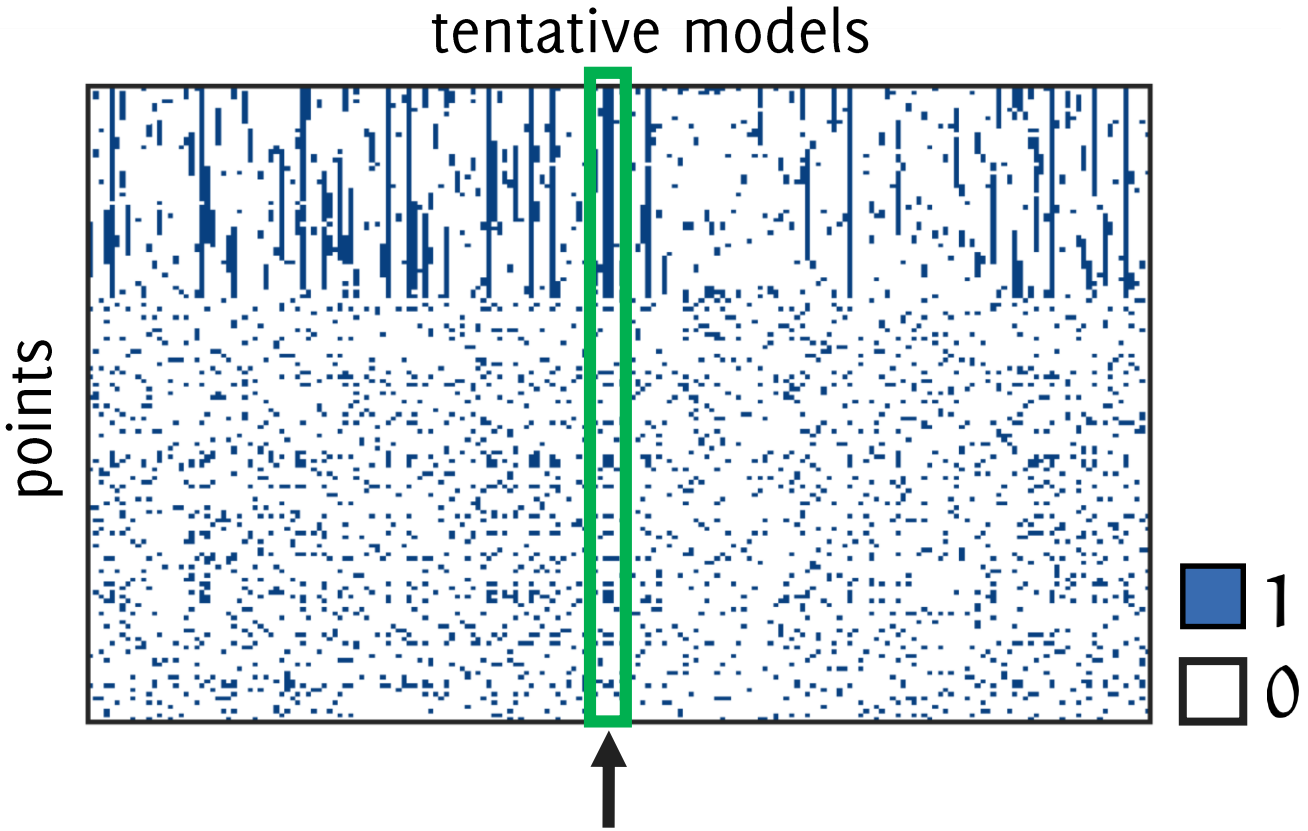
Credits Luca Magri

Randomized Sample Consensus



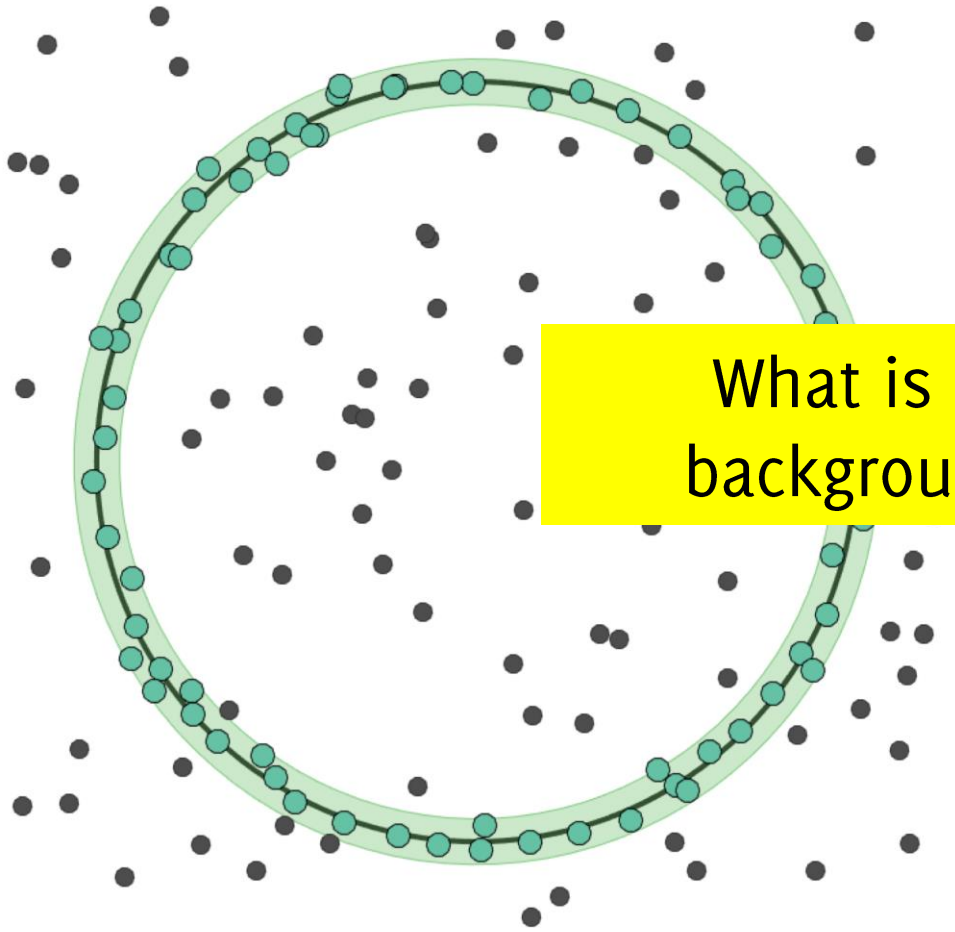
Data driven search of model space

$$H = \{\theta_1, \theta_2, \ldots, \theta_m\} \approx \Theta$$



Credits Luca Magri

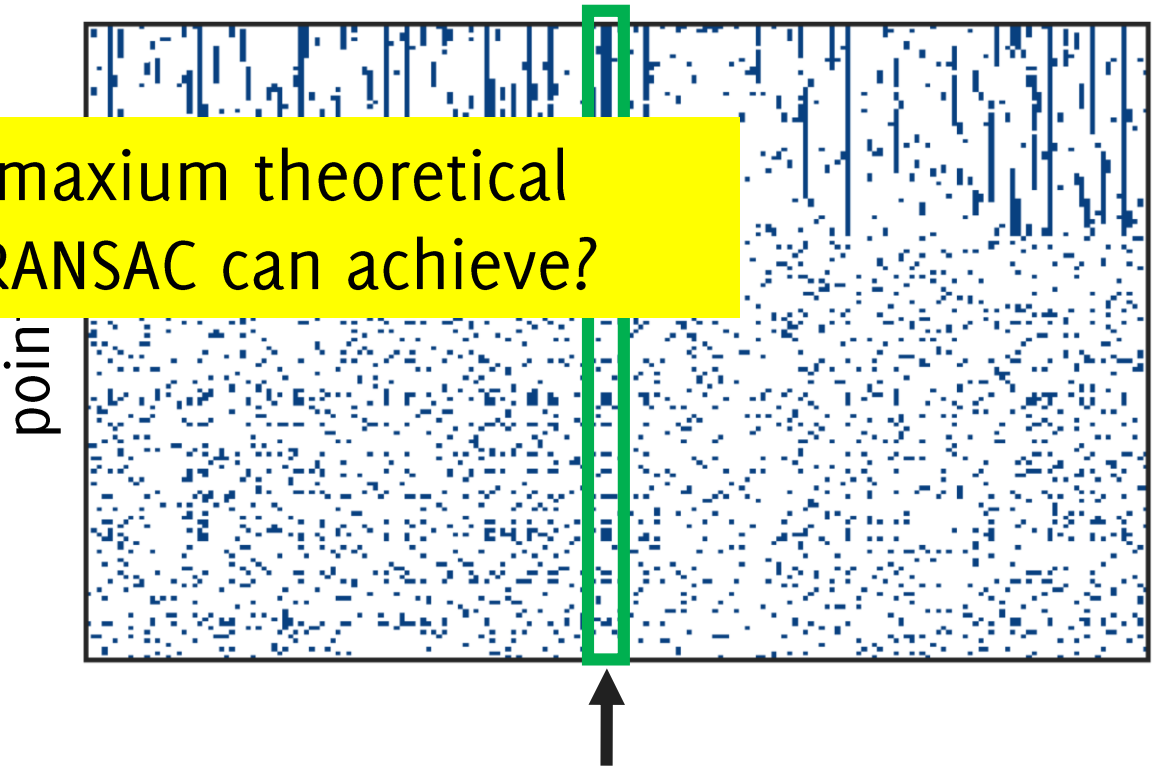
Randomized Sample Consensus



Data driven search of model space

$$H = \{\theta_1, \theta_2, \ldots, \theta_m\} \approx \Theta$$

tentative models

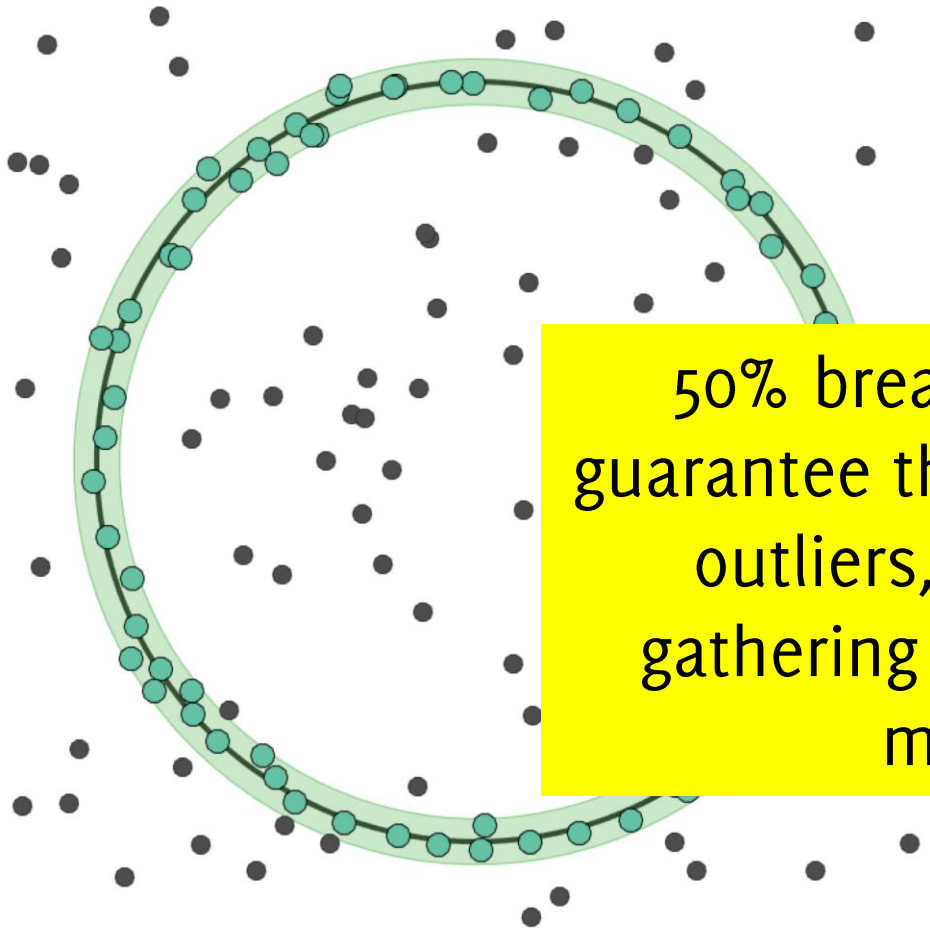


pick the column with the maximum sum

What is the maxium theoretical background RANSAC can achieve?

Credits Luca Magri

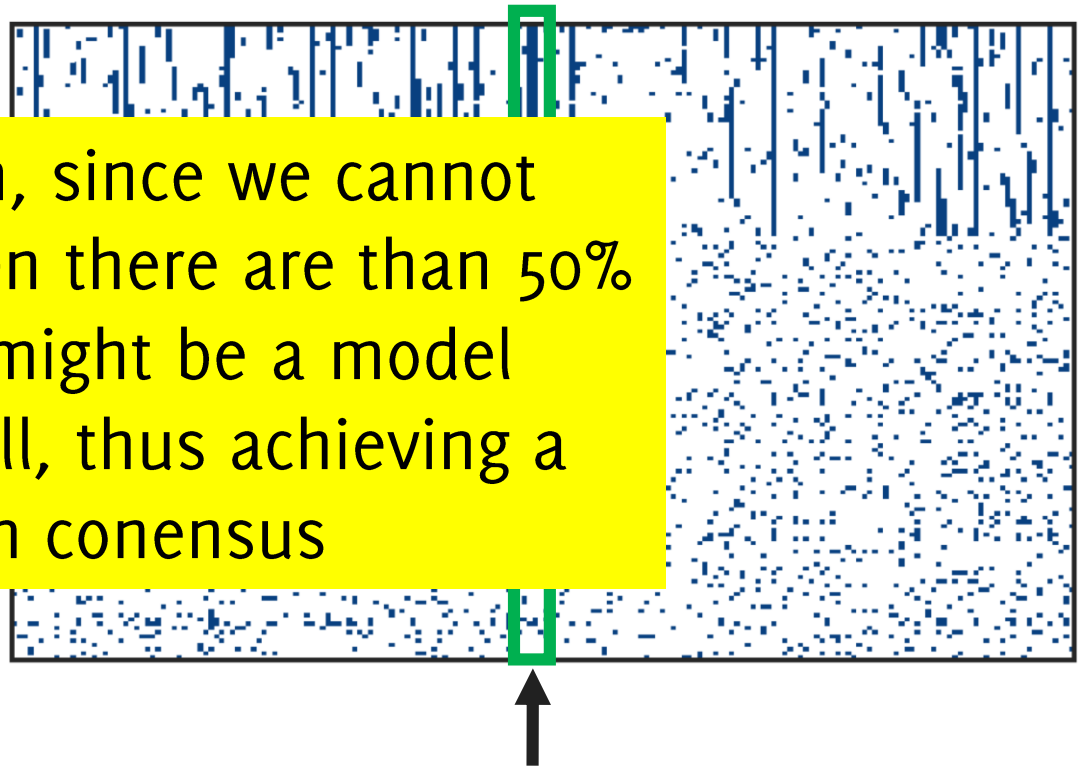
Randomized Sample Consensus



Data driven search of model space

$$H = \{\theta_1, \theta_2, \ldots, \theta_m\} \approx \Theta$$

tentative models



50% break-down, since we cannot guarantee that when there are than 50% outliers, there might be a model gathering them all, thus achieving a maximum conensus

1

0

pick the column with the maximum sum

Credits Luca Magri

Ransac: practical issues

The size of the minimum sample S: this is the bare minimum number of points to fit the parametric model at hand

The inlier threshold ϵ: this can be estimated from the noise in the data

The number of max iterations n: the criteria for selecting the number of samples n: "Choose n so that, with probability p, at least **one random sample is without outliers** (e.g. $p = 0.99$) "

The maximum number of iterations n

Let e the probability of a sample to be an outlier and $1 - e$ the probability of an inlier (can be estimated / provided by a-priori information)

The probability that all s points are inliers: $(1 - e)^s$

The probability that at least one point in S is an outlier: $1 - (1 - e)^s$ (this is the probability for a sample S to yield the right model)

The probability that all the n selected set contain outliers

$$(1 - (1 - e)^s)^n$$

The probability that at least one the n set is without outliers:

$$1 - (1 - (1 - e)^s)^n$$

Set n to have the above probability below a parameter p

$$p = (1 - (1 - (1 - e)^s)^n) \rightarrow n = \log(1 - p) / \log(1 - (1 - e)^s)$$

Ransac: practical issues

Choose n so that, with probability p, at least one random sample is free from outliers (e.g. $p = 0.99$) (outlier ratio: e)

$$n = \log(1-p) / \log(1-(1-e)^s)$$

	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

Ransac: details

Repeat n times:

- Draw s points uniformly at random
- Fit line to these s points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than t)
- Update n
  - $e = 1 - \frac{\text{number of inliers}}{\text{number of points}}$
  - $n = \log(1-p) / \log(1-(1-e)^s)$

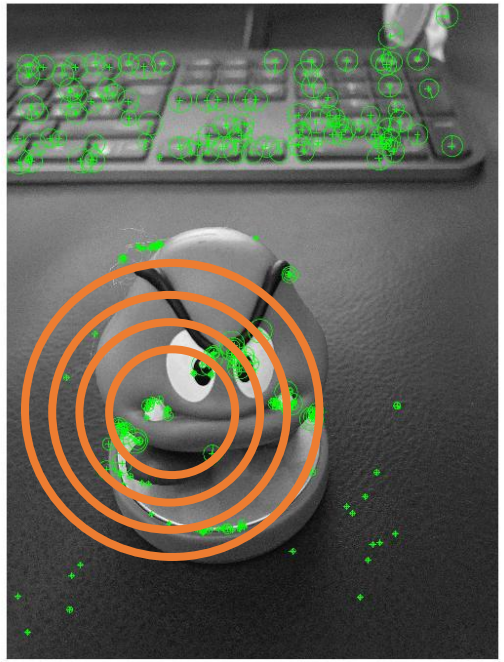
Choose the best model

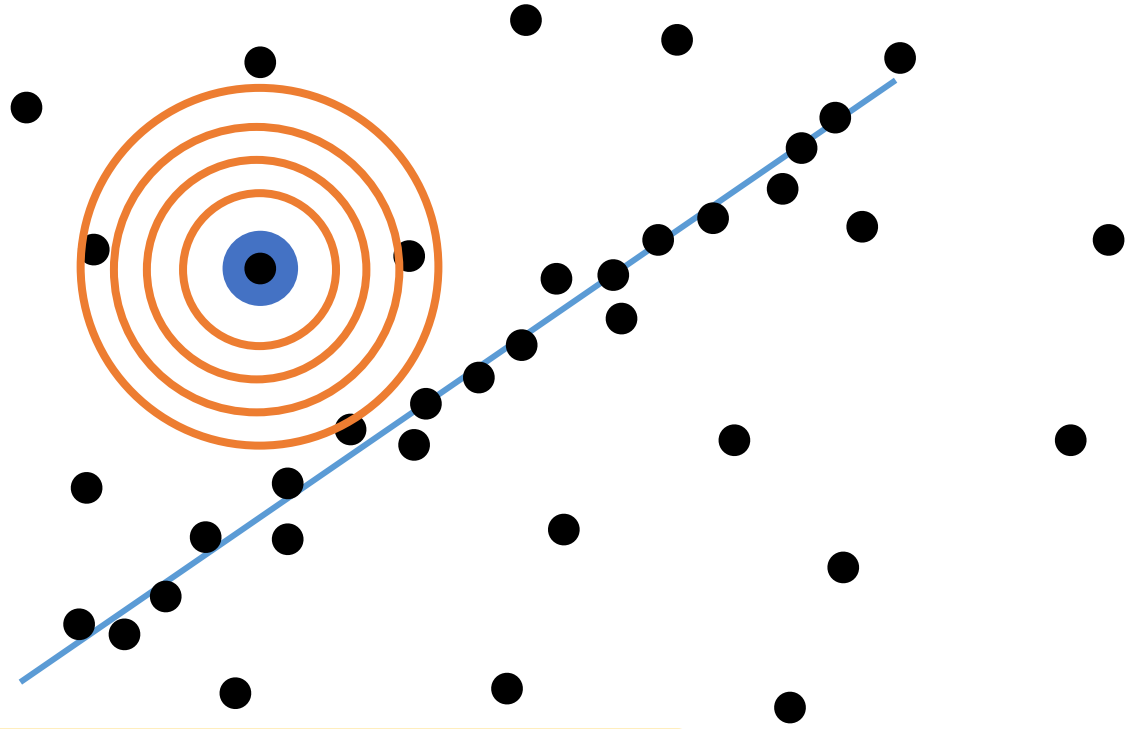
Re-estimate the line with the inliers only through ordinary least square

The sampling strategy

In some applications inliers tend to be localized:

- draw uniformly a first point
- draw the other samples from the neighborhood of the initial point



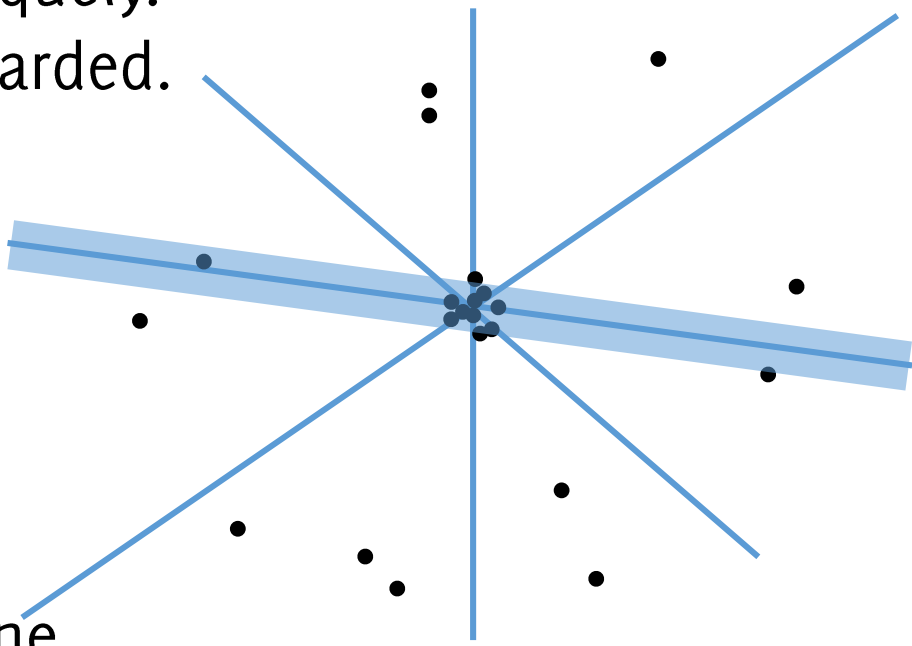


Barath, Noskova, Ivashechkin, Matas: MAGSAC++, a fast, reliable and accurate robust estimator. CVPR 2020

Daniel Barath and Jiri Matas, Graph-Cut RANSAC; CVPR 2018

Degeneracy check

A degenerate configuration is a configuration of points in the minimum sample set for which it is not possible to define the model uniquely.
Typically, has higher random support and must be discarded.



Examples:

For fundamental matrix, planar points or dominant plane.

For homography, collinear points.

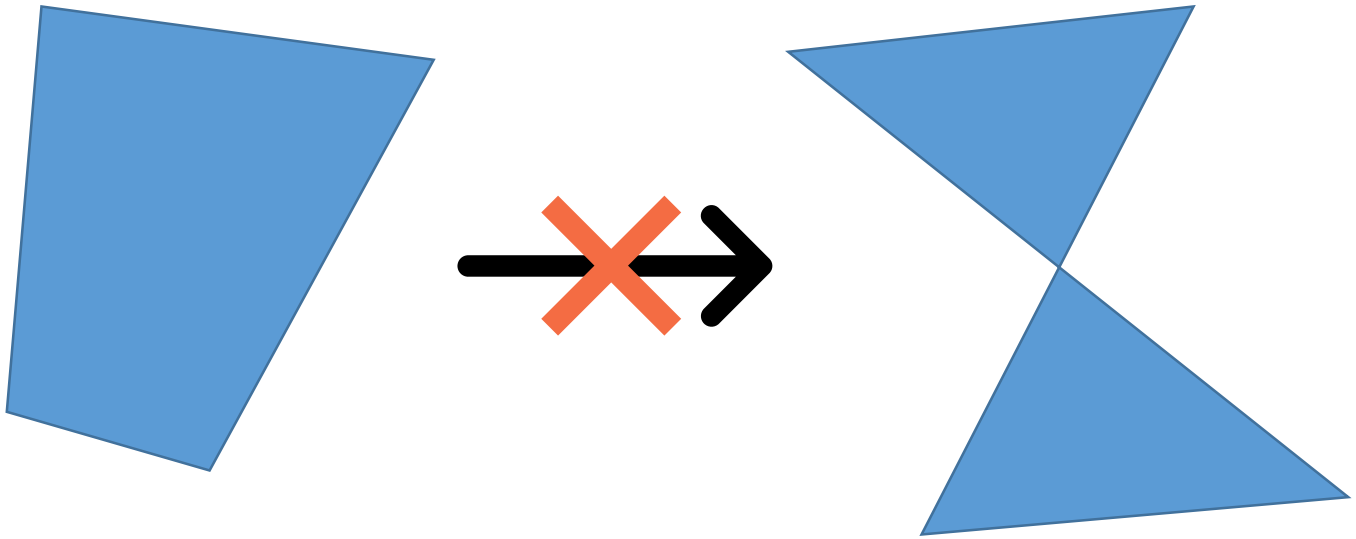
When using SVD, when the rank of the coefficient matrix drops with respect the desired dimension.

Frahm, Pollefeys: RANSAC for (Quasi-)Degenereate Data (QDEGSAC). *CVPR 2006*

Sample checks

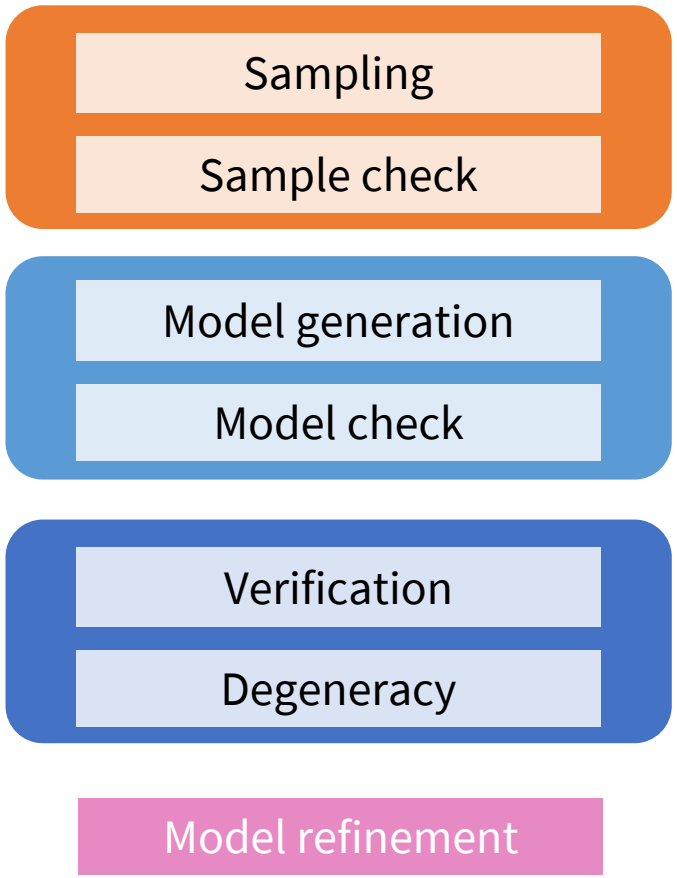
Before the verification step, minimal sample set can be checked:

Cameras can only observe points in front of them (remember chirality constraints used for the essential matrix).
Homography transformation need to preserve convexity:



Chum, Werner, Matas: Epipolar Geometry Estimation via RANSAC Benefits from the Oriented Epipolar Constraint. ICPR 2004

RanSaC implementation with bells and whistles



Sample minimal subset

Generate minimal sample model

Is the model interesting?

Generate non minimal sample model

Raguram, Chum, Pollefeys, Matas, Frham Usac a Universal Framework for Random Sample Consensus. TPAMI 2013

Ransac

Pros:

- very popular (>22900 citations in Google Scholar)
- many improvements have been proposed
- very versatile
- agnostic on outlier percentage
- mild assumption: know the scale noise
  to set the inlier threshold

Cons:

- can take longer than expected

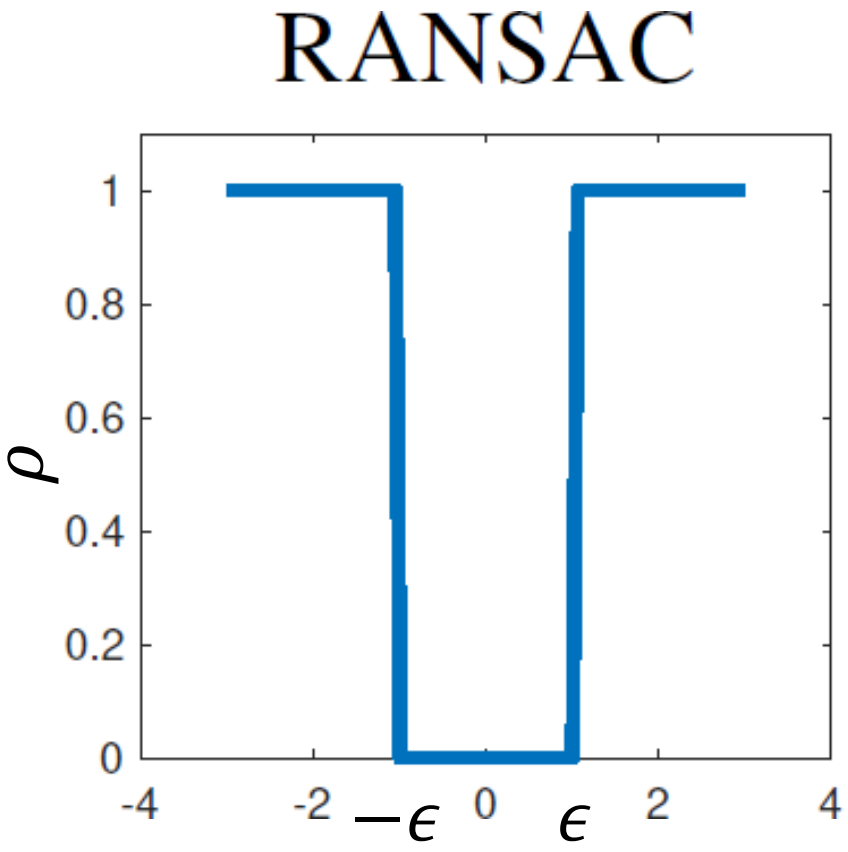
Ransac as M-estimator

(Steward 1999) RanSaC can be seen as a particular M-estimator since the **loss it minimizes** is the number of points having residual above the inlier threshold ϵ

$$f(r_i) = \begin{cases} 1, & r_i > \epsilon \\ 0, & r_i \leq \epsilon \end{cases}$$

Of course selecting inlier thrshold ϵ is very critical

Ransac achieves a theoretical breakdown of 50% of outliers, but in practice, provided a good selection of ϵ, this can be even higher

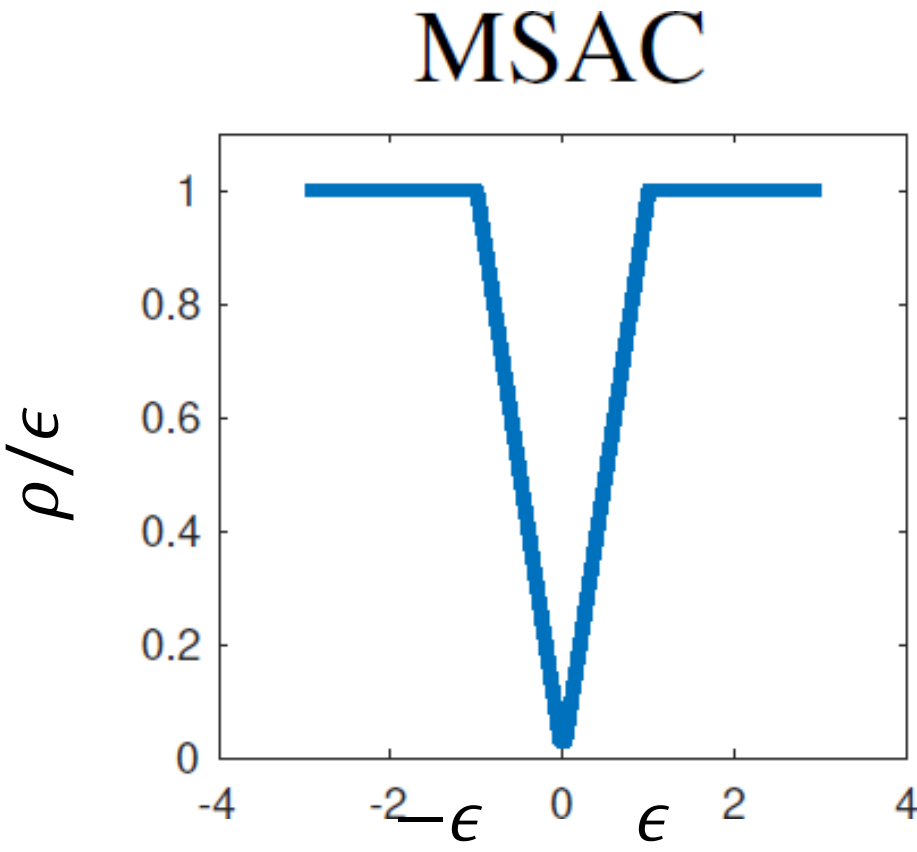


MSAC

(Torr and Zisserman 2000) a different loss function to be minimized within the RanSaC framework

$$f(r_i) = \begin{cases} r_i, & r_i > \epsilon \\ \epsilon, & r_i \leq \epsilon \end{cases}$$

This turns to be more effective and should be preferred to RanSaC



Ransac vs MSaC

Ransac

Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

MSaC

Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = +\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Estimate inlier set $I = \{x \in X : r(x, \theta)^2 < \epsilon^2\}$;
  Evaluate $J(\theta) = \sum_{x \in I} r(x, \theta) + (|X| - |I|)\epsilon$;
  if $J(\theta) < J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Credits Luca Magri

Video Analytics

math&sport

POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

math&sport


07 km/h
PACE
71 %
EFFECTIVENESS
Carter-Vickers
2033
LIKES
BACK TO REPLAY

math&sport

Powered by

MOX OFF
MATHEMATICS FOR INNOVATION

vodafone

Action for 5G,
il bando Vodafone
per le startup

math&sport

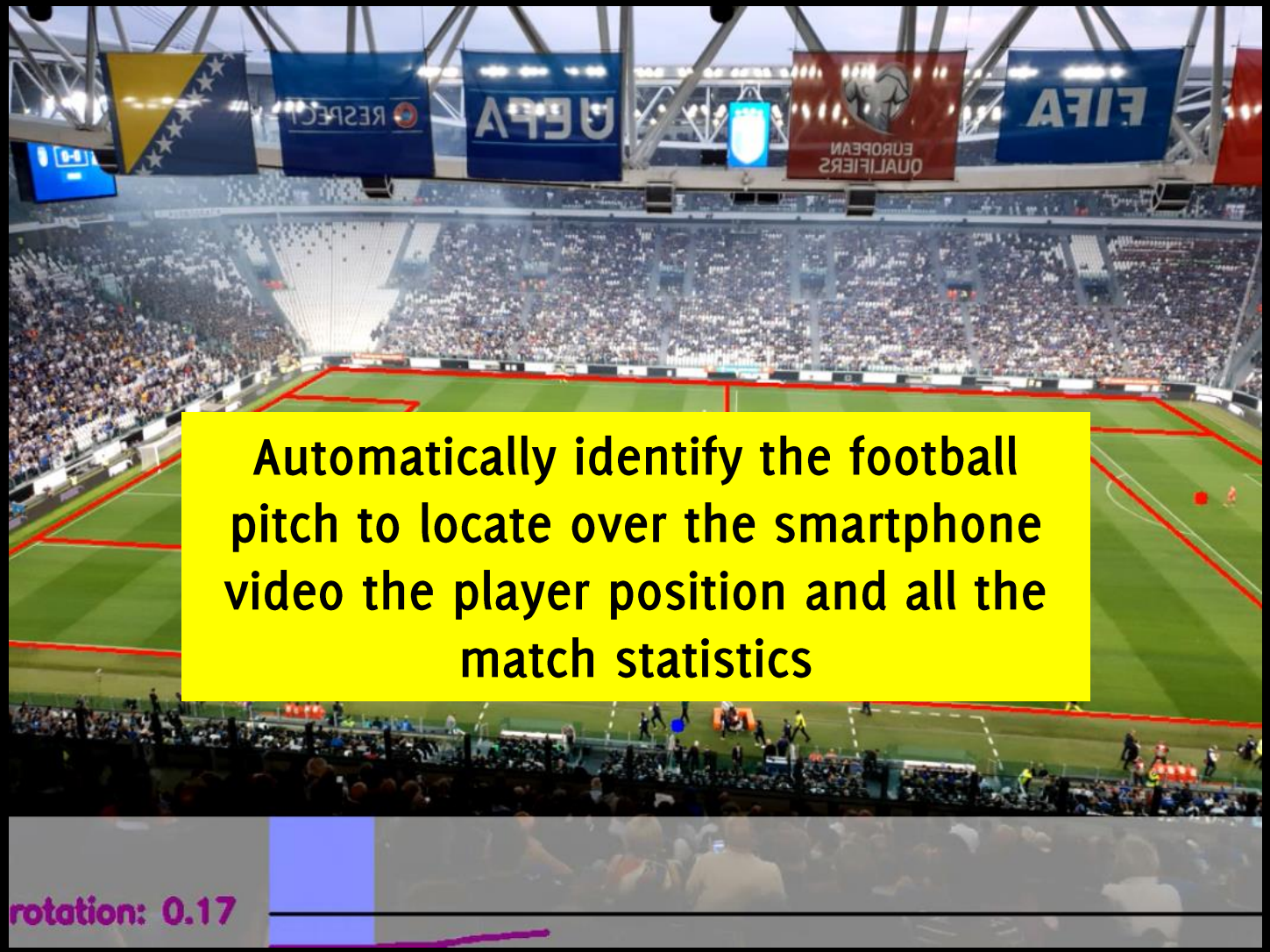
Powered by

MOX OFF
MATHEMATICS FOR INNOVATION

vodafone

Action for 5G,
il bando Vodafone
per le startup

math&sport



Powered by

MOX-OFF
MATHEMATICS FOR INNOVATION

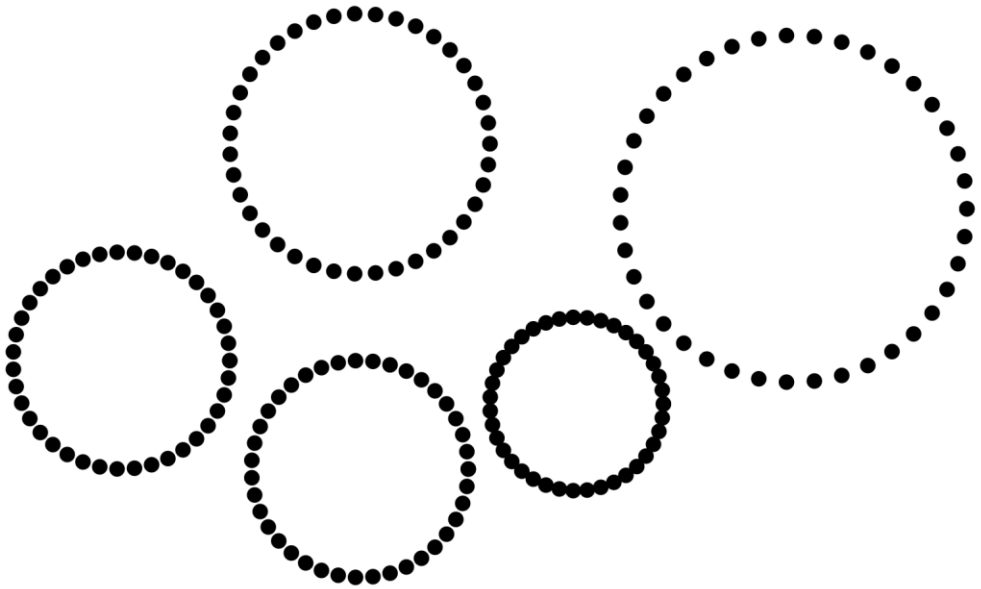
vodafone

Action for 5G,
il bando Vodafone
per le startup

Multi-Model Fitting

The problem of multi-model fitting (or structure recovery)

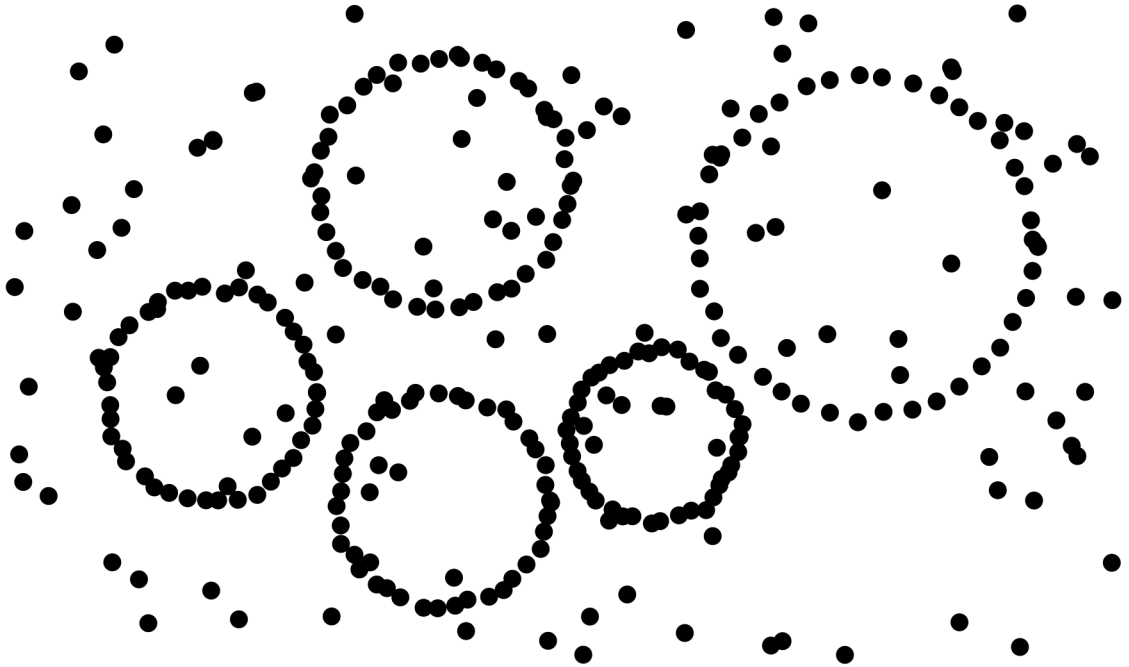
Given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, possibly corrupted by noise and outliers, and a family of geometric models Θ



Credits Luca Magri

The problem of multi-model fitting (or structure recovery)

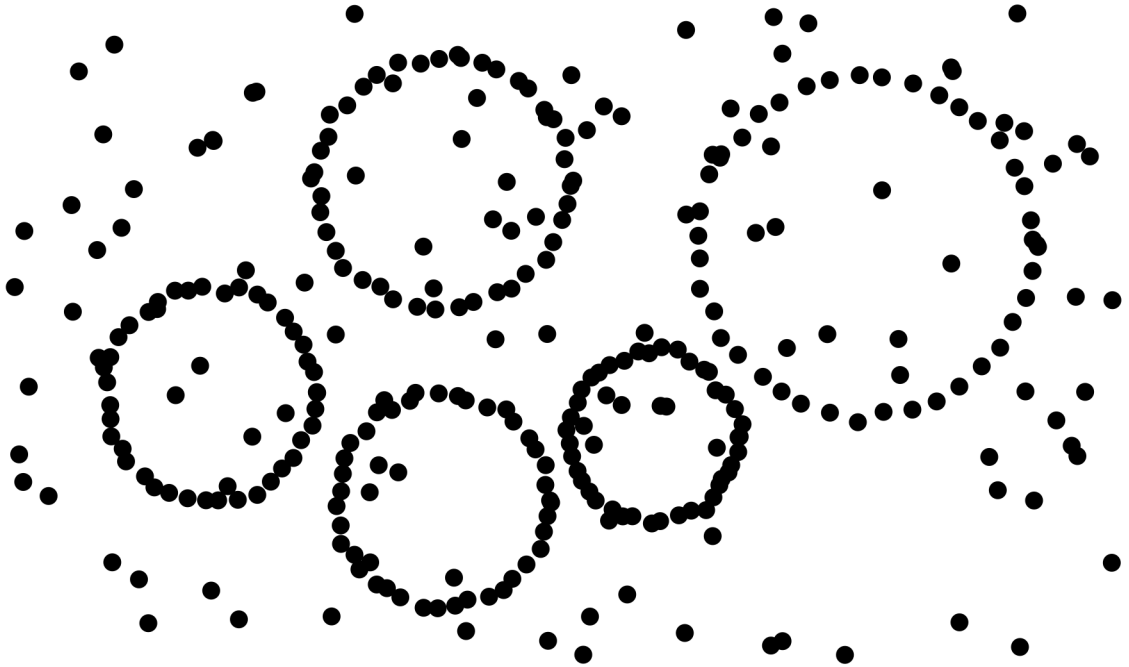
Given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, possibly corrupted by noise and outliers, and a family of geometric models Θ



Credits Luca Magri

The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, possibly corrupted by noise and outliers, and a family of geometric models Θ



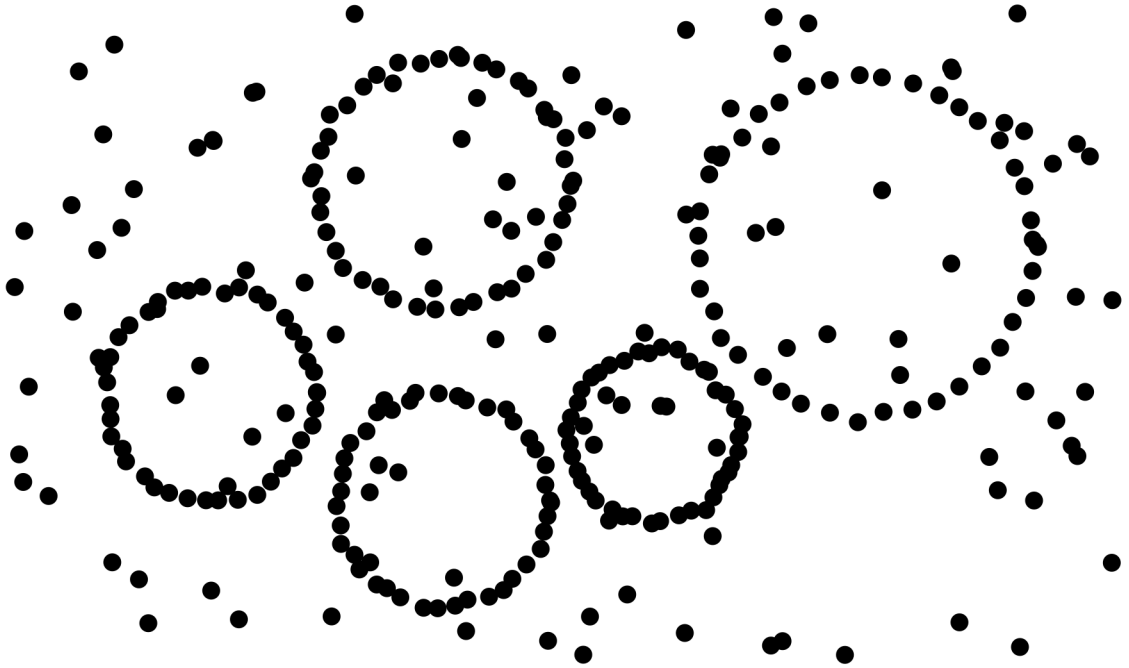
Goal: automatically estimate the models that best explain the data/discover the structures hidden in the data

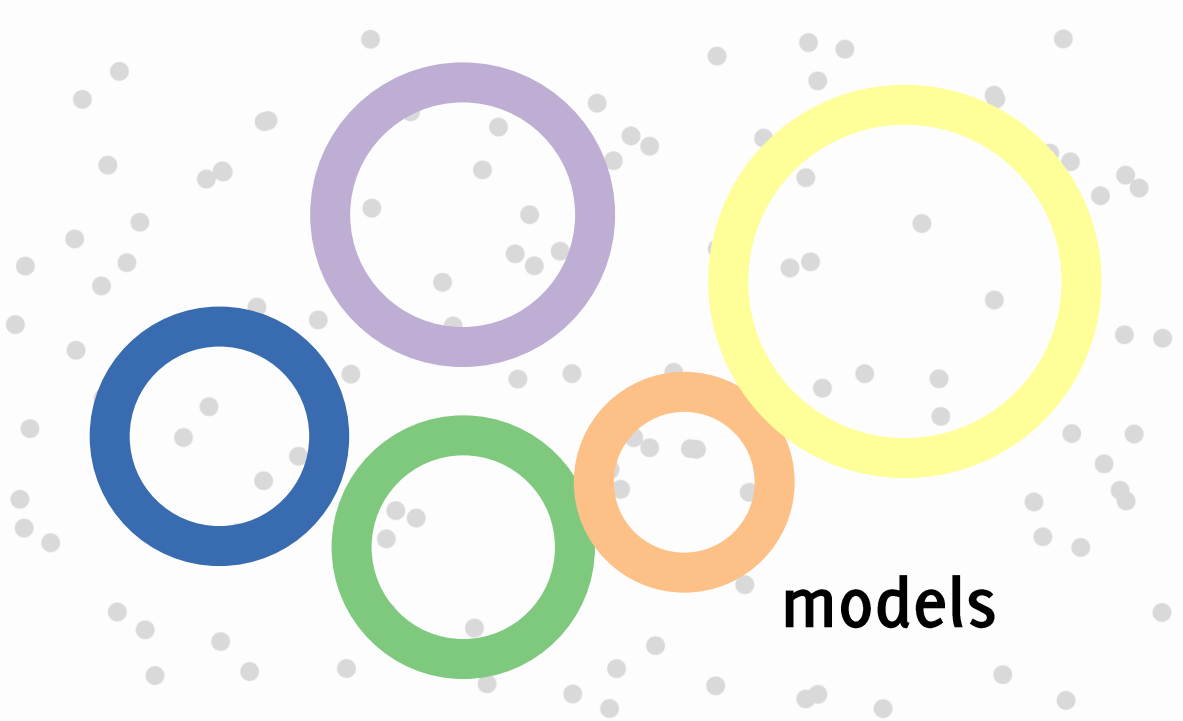
Credits Luca Magri

The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, possibly corrupted by noise and outliers, and a family of geometric models Θ

Goal: automatically estimate the models that best explain the data/discover the structures hidden in the data





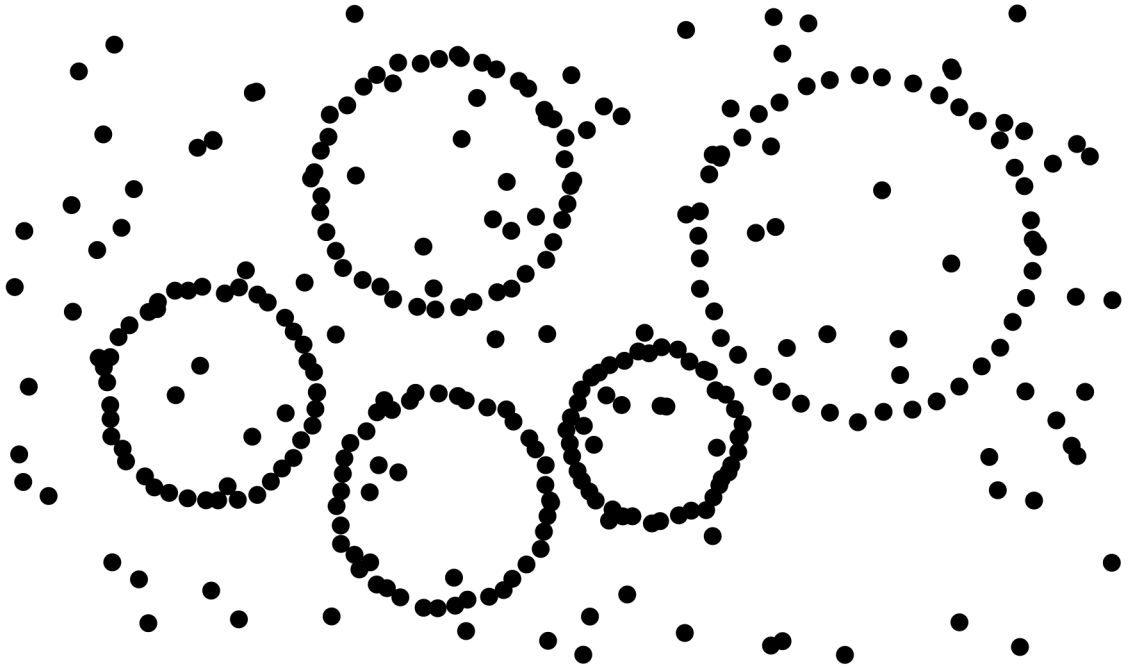
"in the eye of the beholder", mathematical descriptions of the data that an observer fits

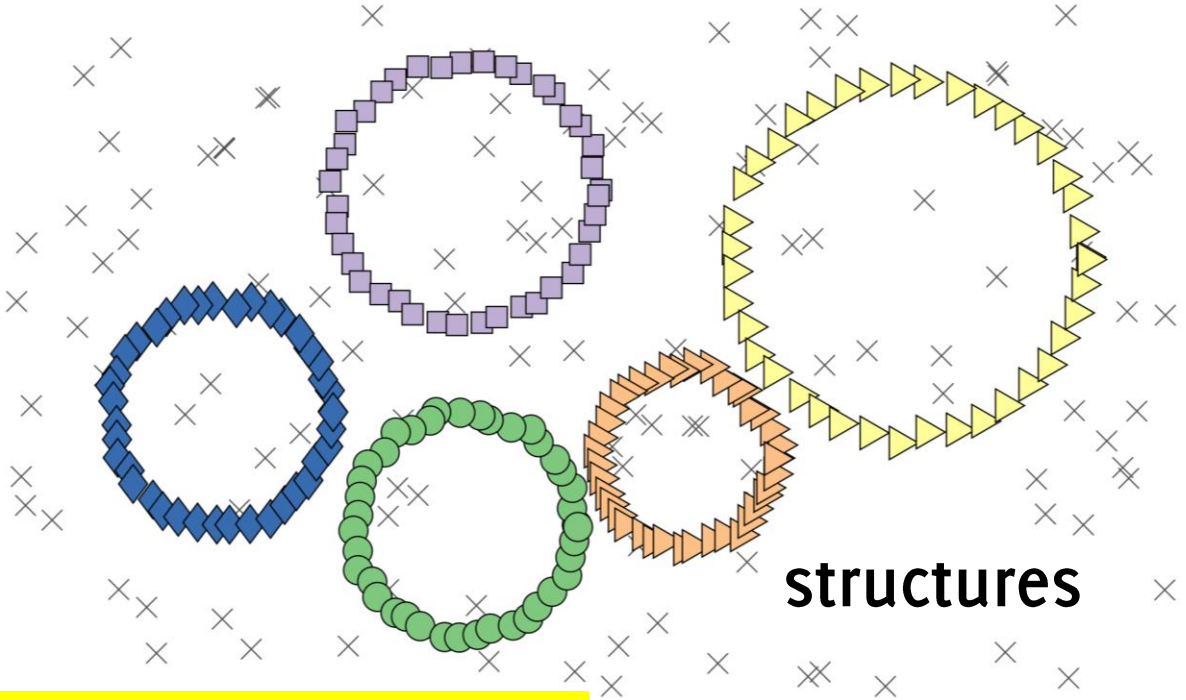
Credits Luca Magri

The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, possibly corrupted by noise and outliers, and a family of geometric models Θ

Goal: automatically estimate the models that best explain the data/discover the structures hidden in the data

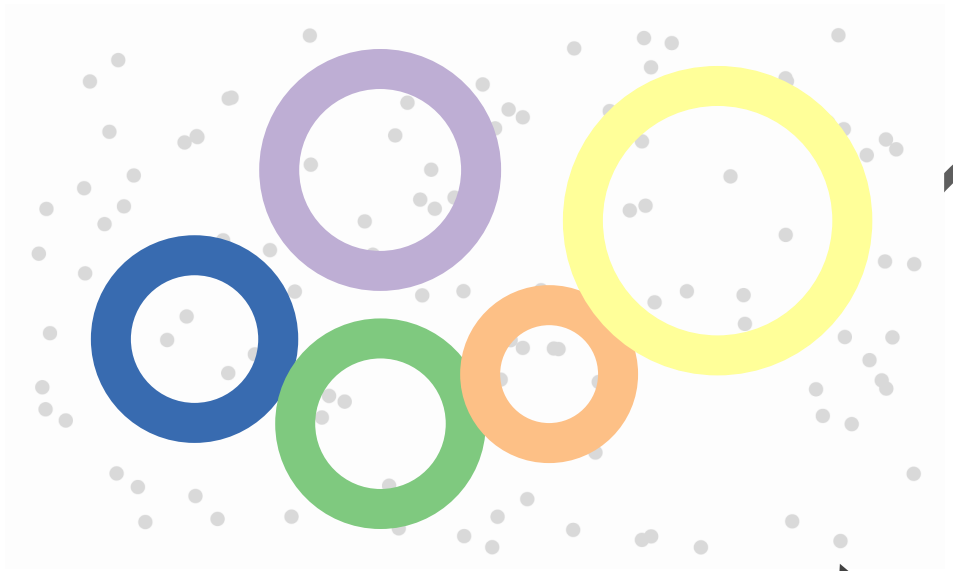




relations among the data, intrinsic to data

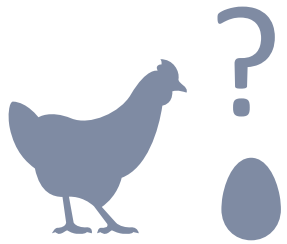
Credits Luca Magri

The Challenges of multi-model fitting



models

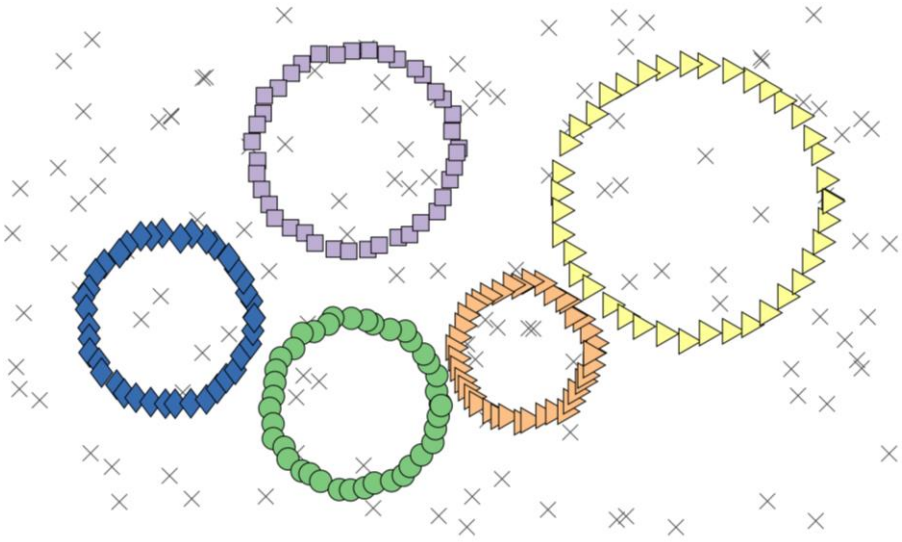
segmentation



chicken & egg dilemma

estimation

structures

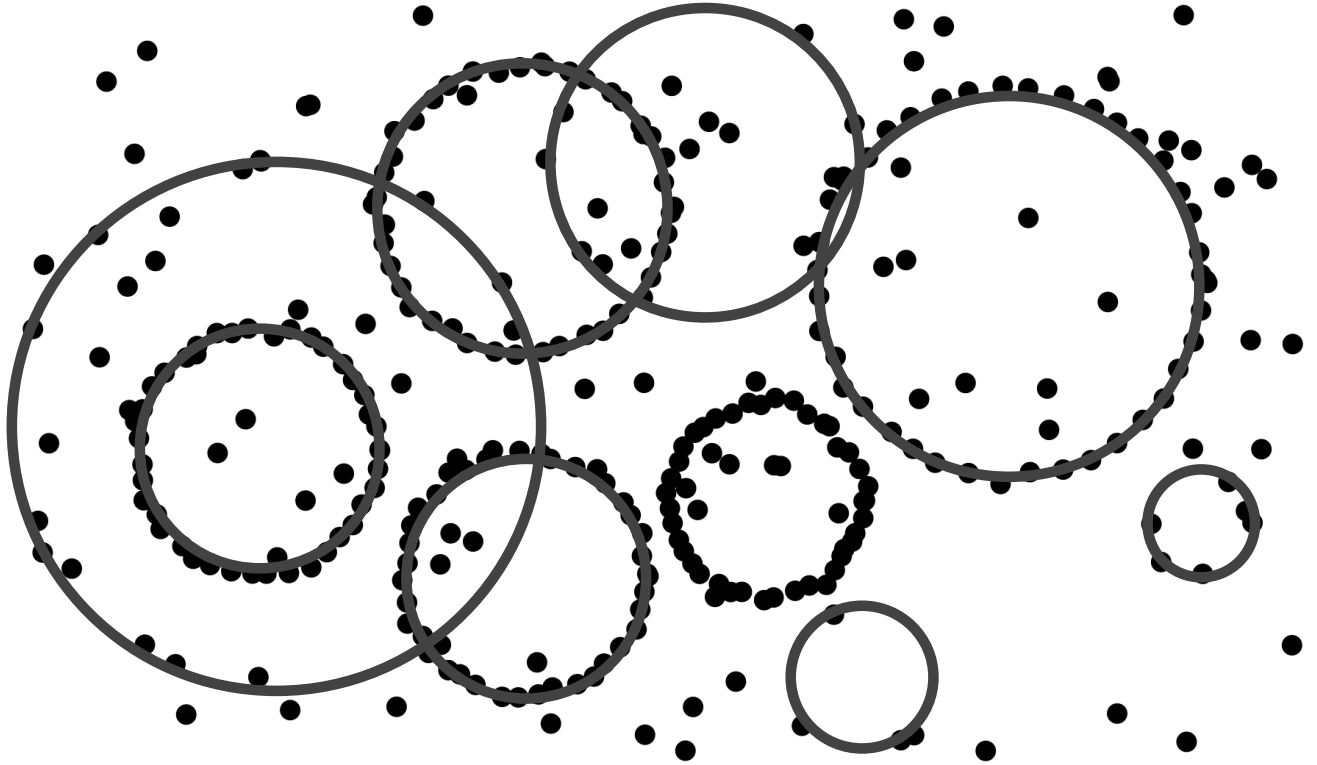


Credits Luca Magri

The Challenges of multi-model fitting

Number of models?

Inliers/outliers?



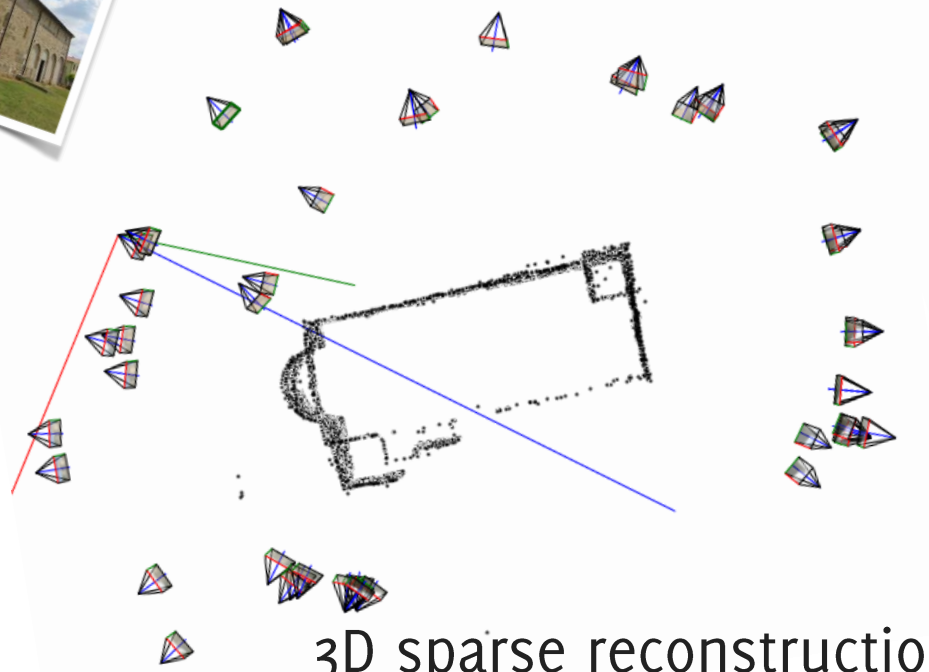
ill posed

Credits Luca Magri

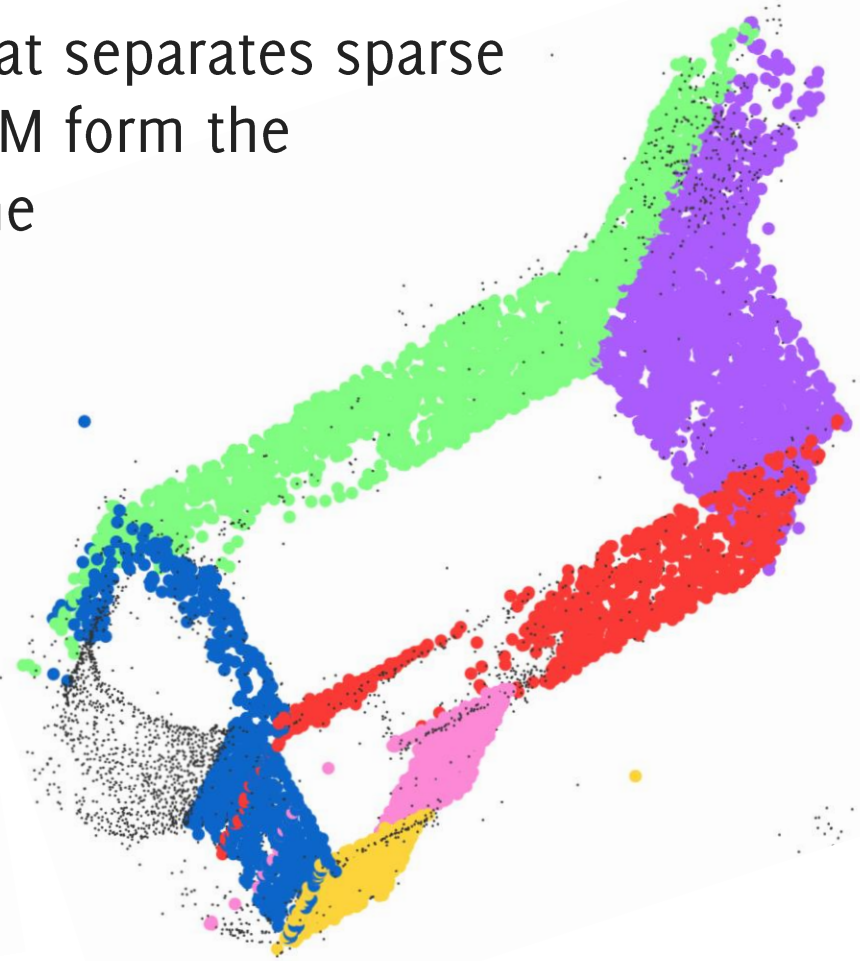
Multi model fitting applications: primitive fitting

input images

Bridge the semantic gap that separates sparse point cloud coming from SfM form the understanding of a 3D scene



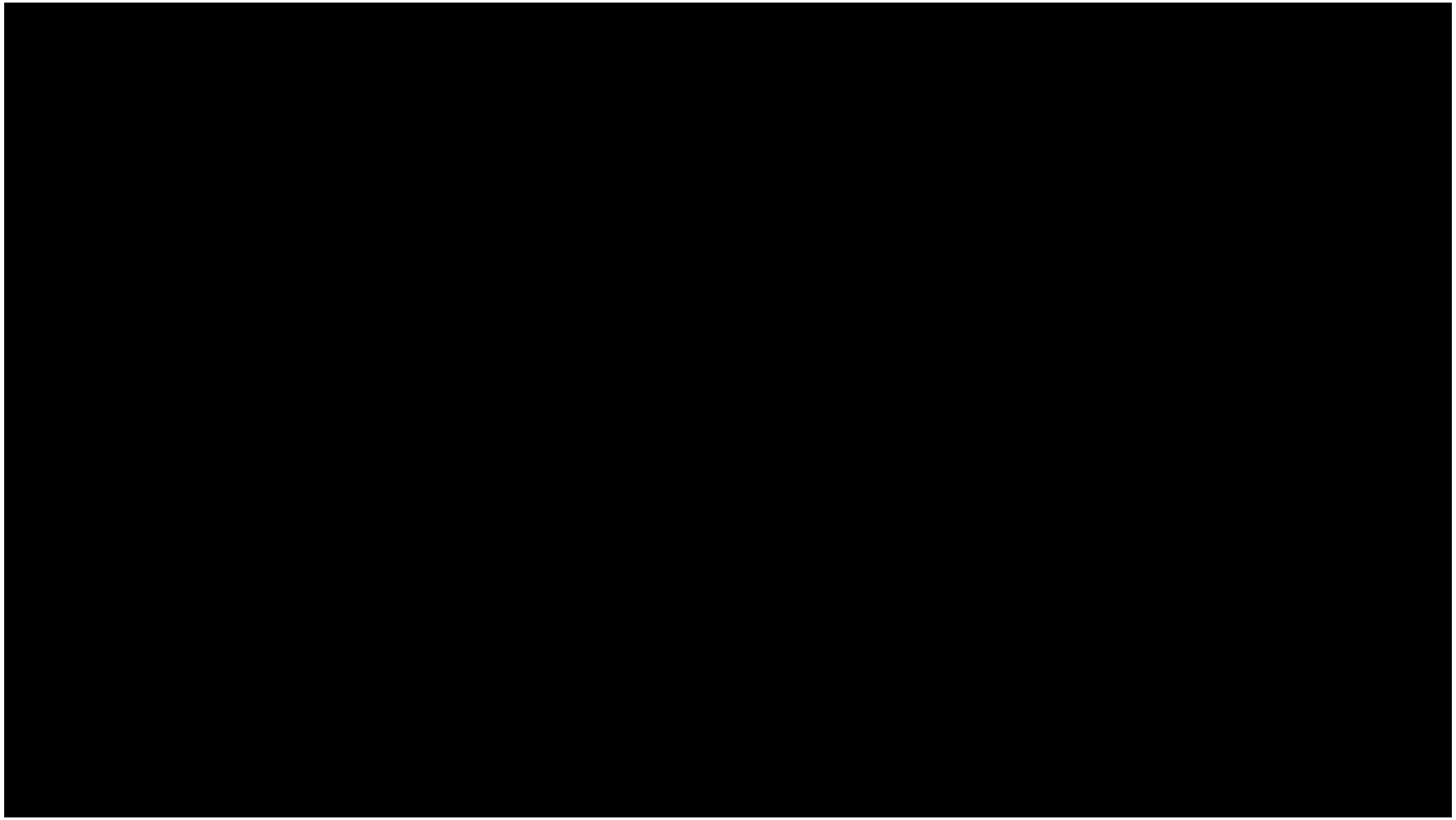
3D sparse reconstruction



$X \subset \mathbb{R}^3, \Theta =$ planes

Credits Luca Magri

Multimodel fitting for 3D scattered data



L. Magri, and A. Fusiello. "Reconstruction of interior walls from point cloud data with min-hashed J-linkage." 2018 3DV

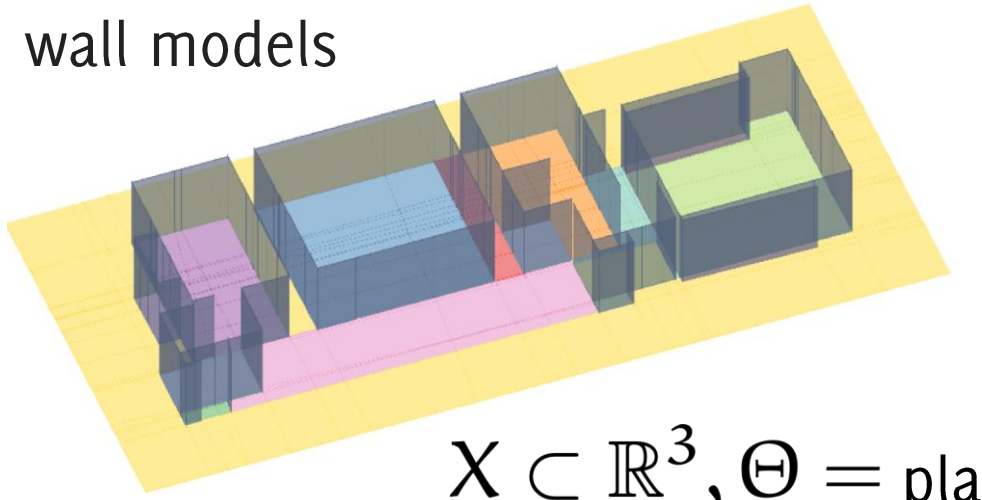
L. Magri, and A. Fusiello. "IMPROVING AUTOMATIC RECONSTRUCTION OF INTERIOR WALLS FROM POINT CLOUD DATA." International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences (2019).

L. Magri, and Andrea Fusiello. "T-linkage: A continuous relaxation of j-linkage for multi-model fitting." CVPR 2014

Multi model fitting applications: scan2bim

scanned point cloud

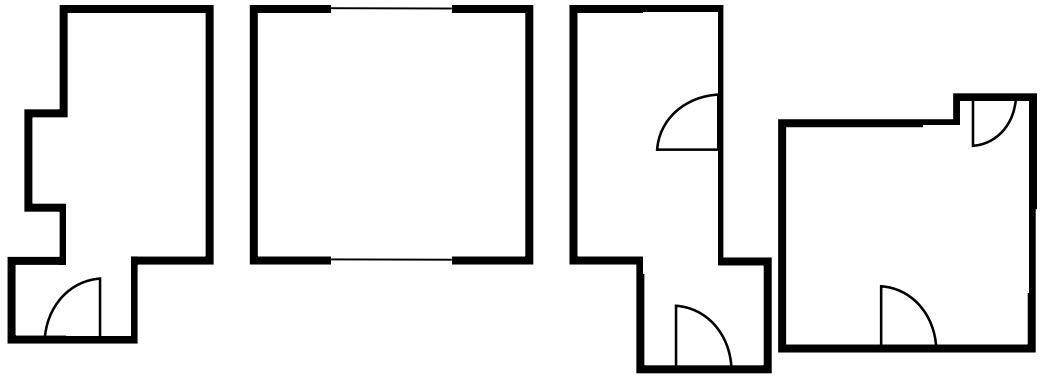
wall models



$X \subset \mathbb{R}^3, \Theta =$ planes

Given a scanned point cloud of an interior environment, detect its primary facility surfaces – such as floors, walls, and ceilings.

floor-plan



$X \subset \mathbb{R}^2, \Theta =$ lines

Credits Luca Magri

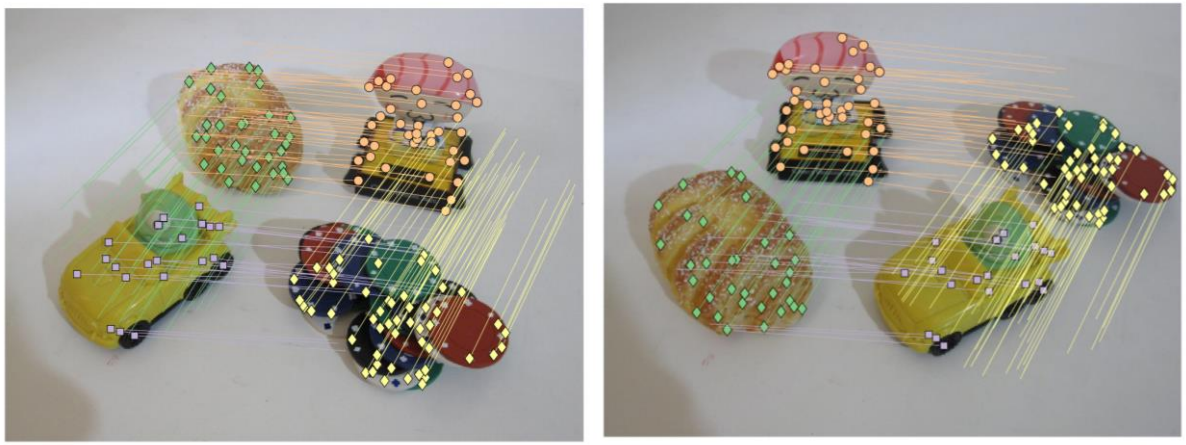
Multi model fitting applications: two view geometry

Geometric fit on corresponding matches across two images

plane detection

$X \subset \mathbb{R}^4, \Theta =$ homographies

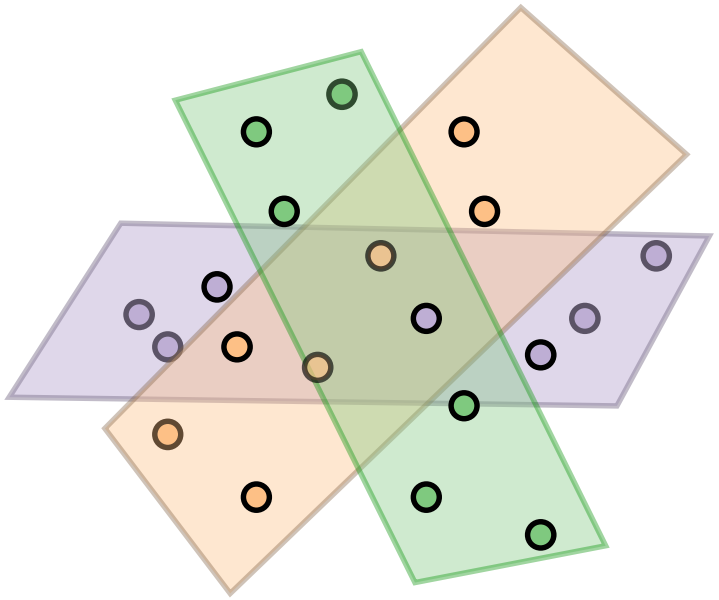
epipolar geometry



$X \subset \mathbb{R}^4, \Theta =$ fundamental matrices

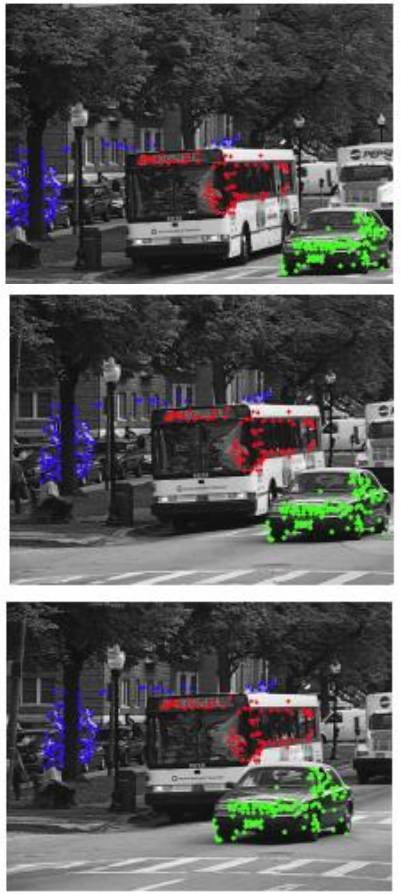
Credits Luca Magri

Multi model fitting applications: subspace clustering



$X \subset \mathbb{R}^d, \Theta =$ subspaces

3D Video segmentation



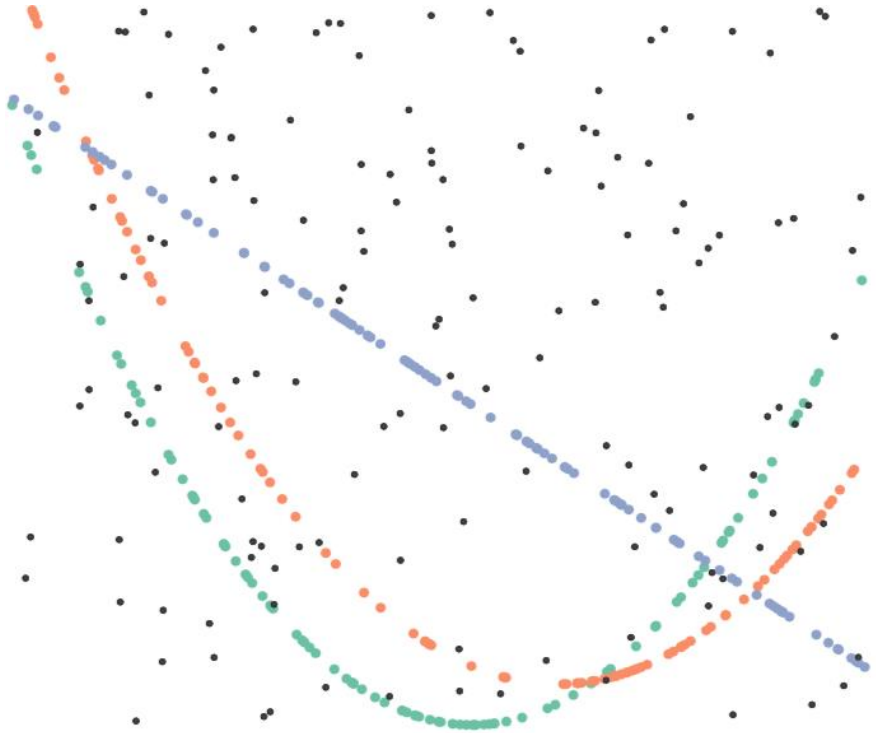
Face clustering

Credits Luca Magri

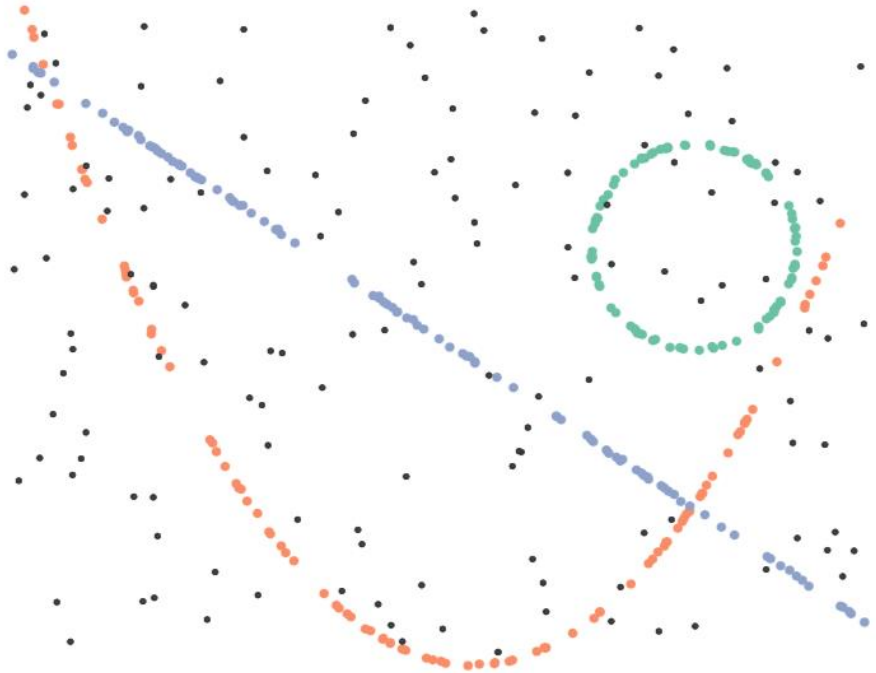
Template Detection

Our Collaboration with an Italian Company T&O

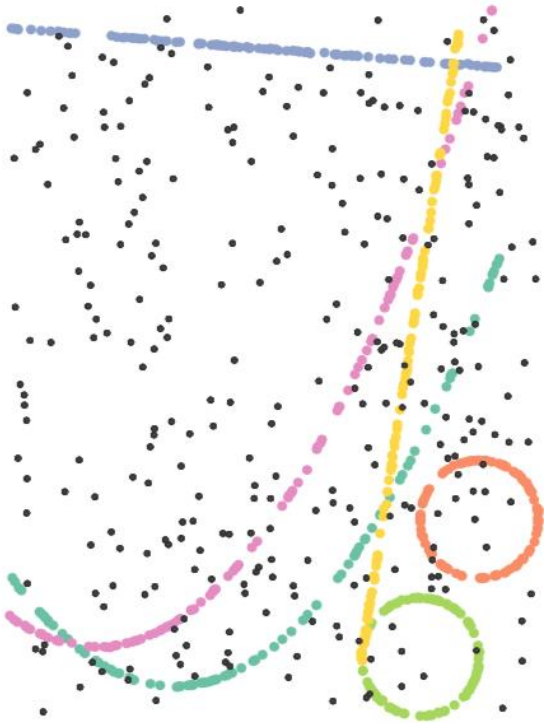
Multimodel (and multi-class) fitting



(a)



(b)

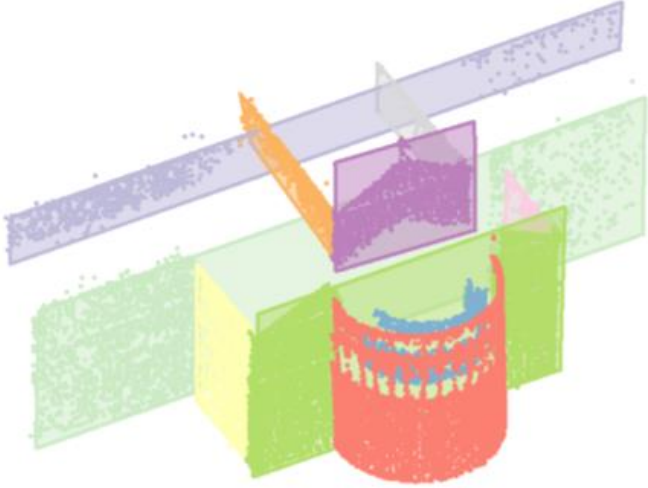


(c)

L. Magri, A. Fusiello. "Fitting Multiple Heterogeneous Models by Multi-Class Cascaded T-Linkage" CVPR 2019.

MultiLink

(a) Input point cloud



(b) Recovered structures

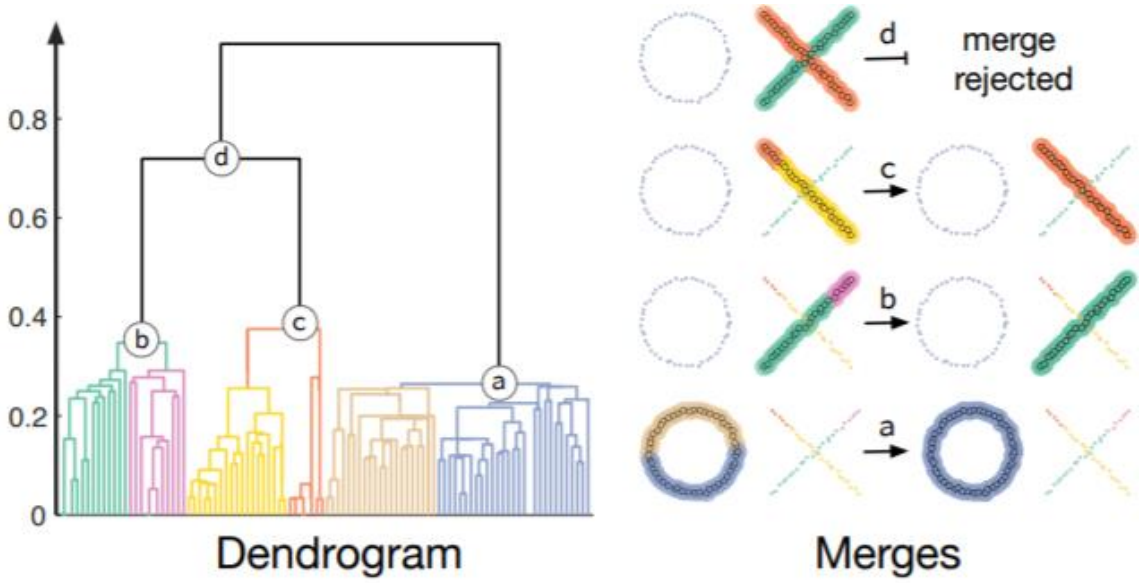


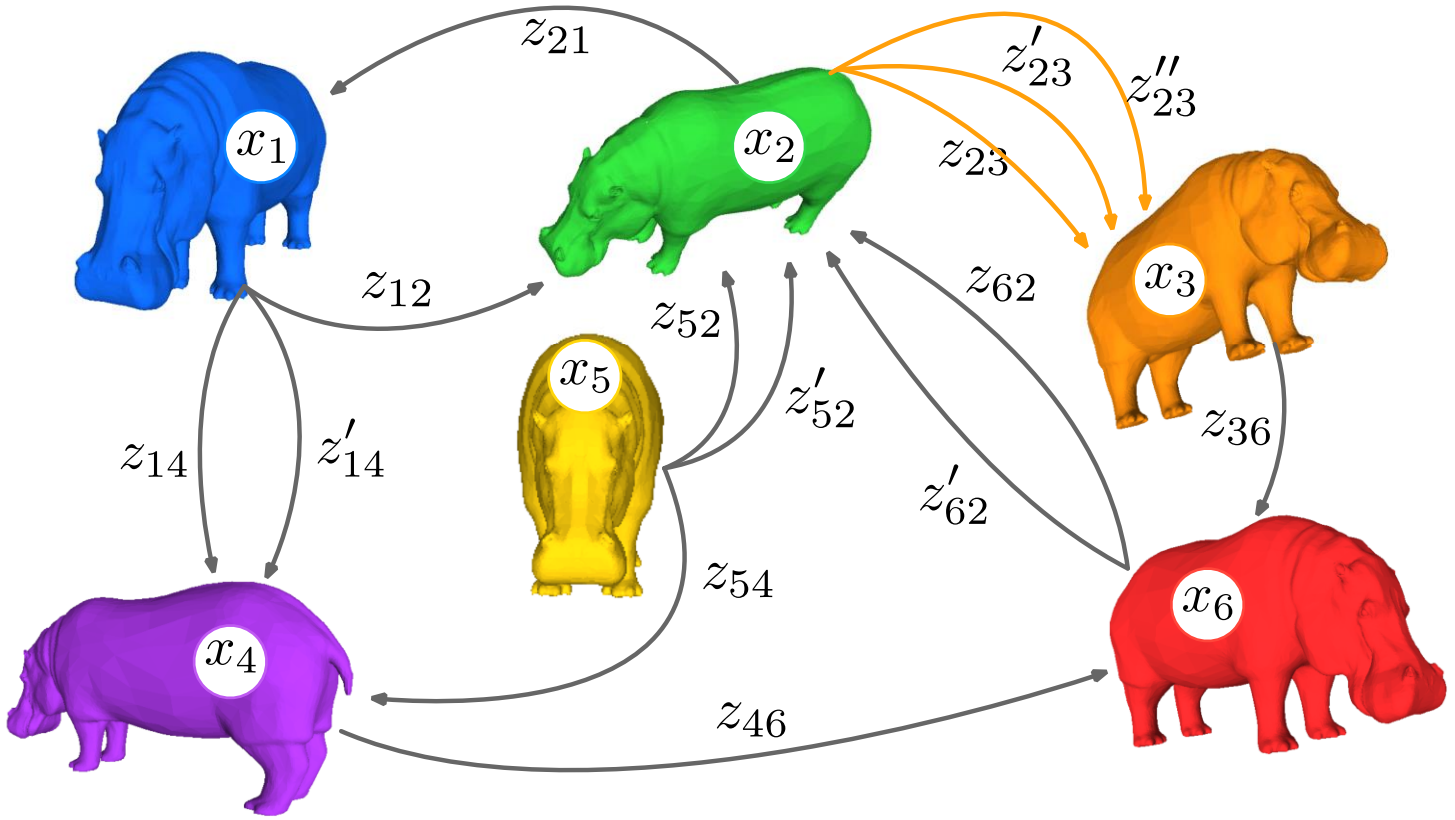
Figure 2: MultiLink combines single-linkage clustering and GRIC. Clusters are merged as long as the GRIC score improves when fitting suitable models on-the-fly. Colors indicate how cluster aggregation proceeds in the dendrogram.

L. Magri, F. Leveni, G. Boracchi, MultiLink: Multi-class Structure Recovery via Agglomerative Clustering and Model Selection, CVPR 2021

Synchronization

Robustly synchronize multiple transformations that can be represented in a matrix group:

- Perform global 3D registration avoiding drifting typical of incremental approaches.
- Graph-partitioning to synchronize very large graphs



A. Porfiri Dal Cin, L. Magri, F. Arrigoni, A. Fusiello, G. Boracchi, *"Synchronization of Group-labelled Multi-graphs" ICCV 2021*

Multi-model Fitting Solutions

Several solutions

Optimization-based

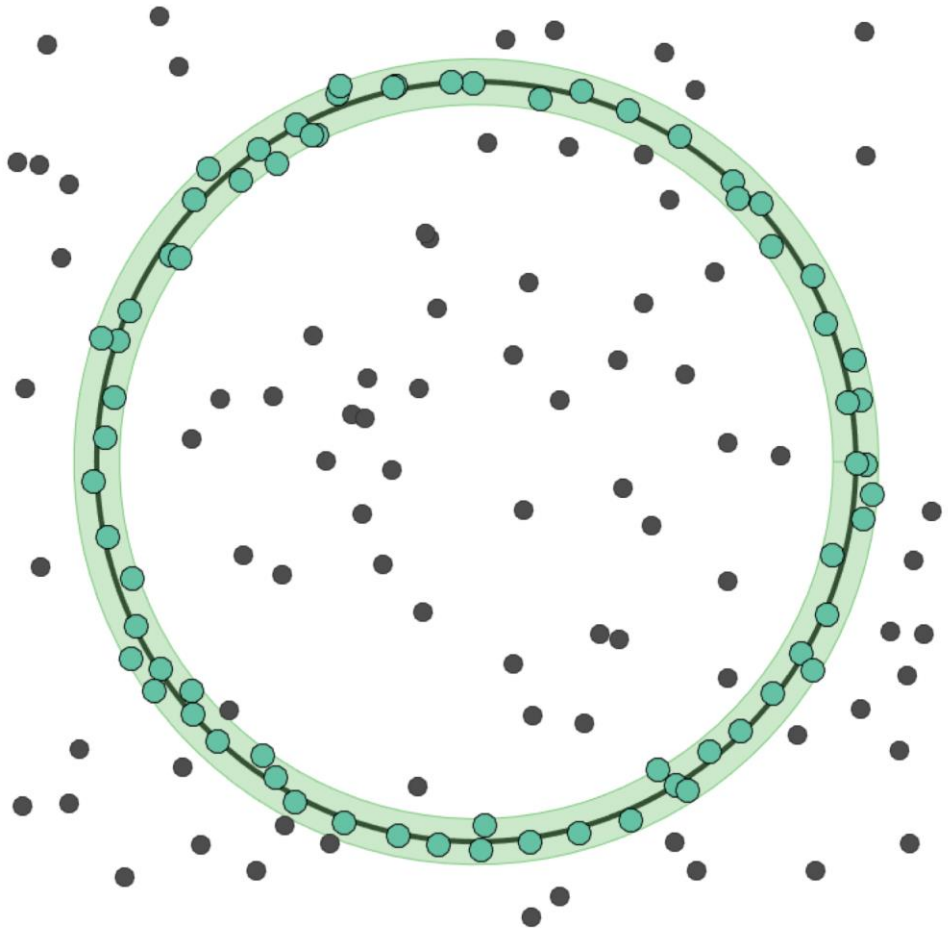
- [Xu et Al PatRecLet 90] Randomize Hough Transform
- [Zuliani ICIP 05] Multi Ransac
- [Isack and Boykov CVPR 10] Pearl
- [Magri and Fusiello CVPR 17] Set coverage
- [Barath and Matas ECCV 18] MultiX
- [Barath and Matas ICCV 19] ProgressiveX

Clustering-based

- [Zhang and Koseckà ECCV 06] Residual analysis
- [Toldo and Fusiello ECCV 08] J-linkage
- [Chin et al ICCV 09] Ordered residual kernel
- [Purkait et Al ECCV 14] Hypergraph clustering
- [Magri and Fusiello CVPR 14] T-linkage
- [Magri and Fusiello BMVC 15] Matrix factorization
- [Magri and Fusiello CVPR 19] Cascaded T-linkage
- [Magri et Al, CVPR 21] Multi-Link

Clustering provides a simple and powerful formulation based on few intelligible parameters

Let's go back to RanSaC

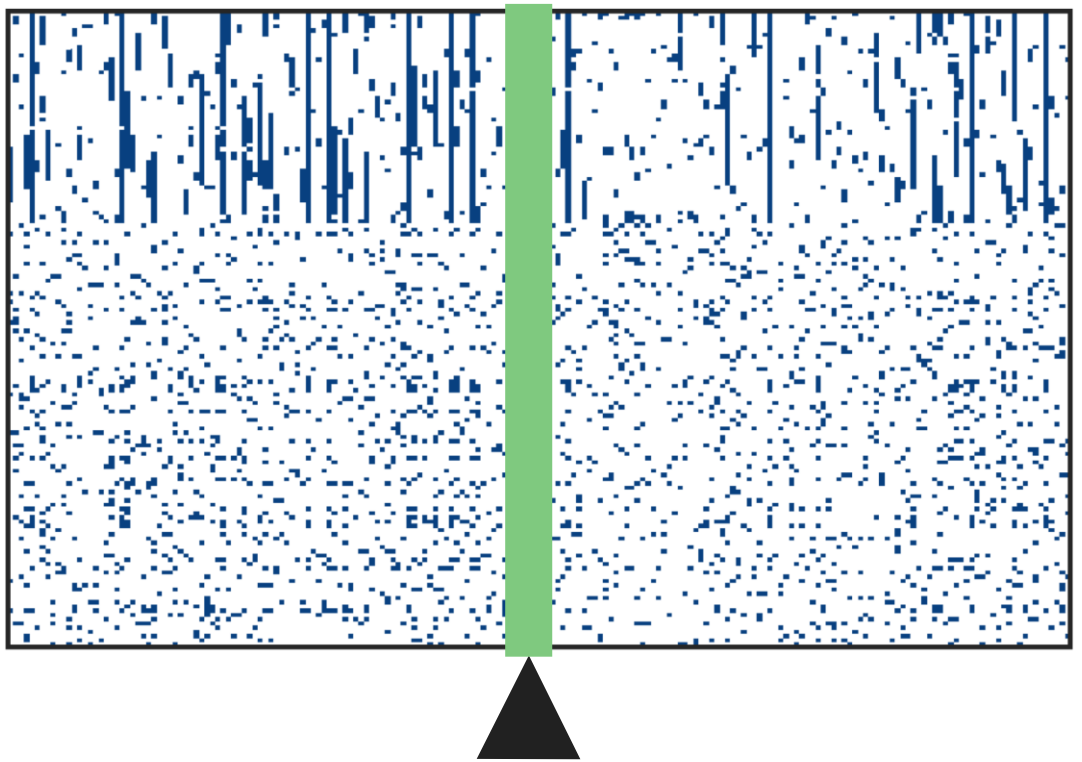


Data driven search of model space

$$H = \{\theta_1, \theta_2, \ldots, \theta_m\} \approx \Theta$$

tentative models

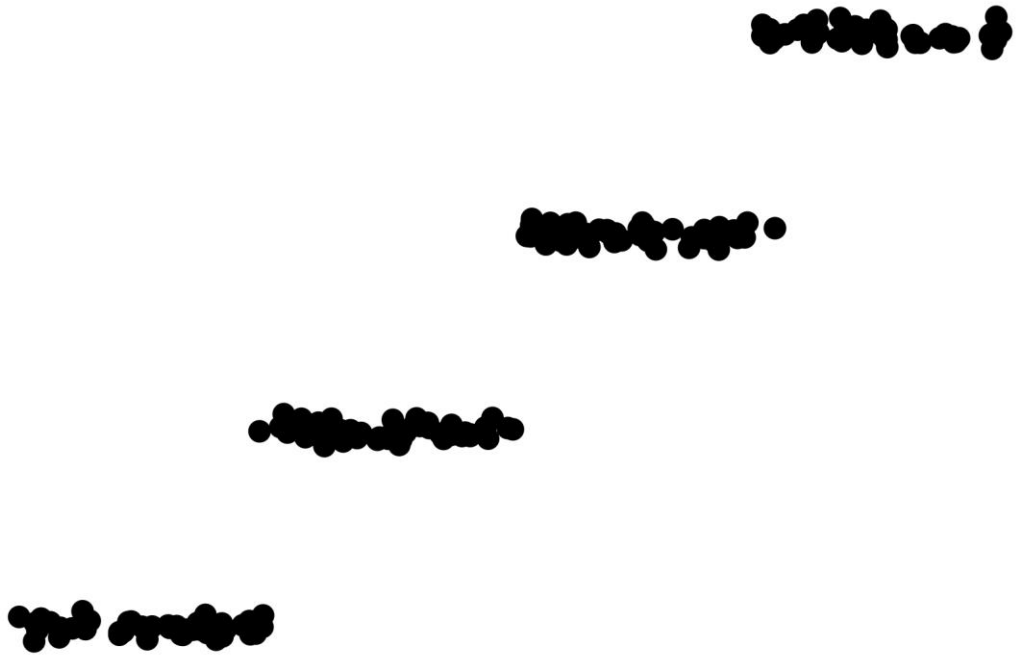
points



pick the column with the maximum sum

Sequential RanSaC

Start RanSaC on the dataset X searching for the best fit for a single instance of the model



Line fitting example

Sequential RanSaC

Start RanSaC on the dataset X searching for the best fit for a single instance of the model

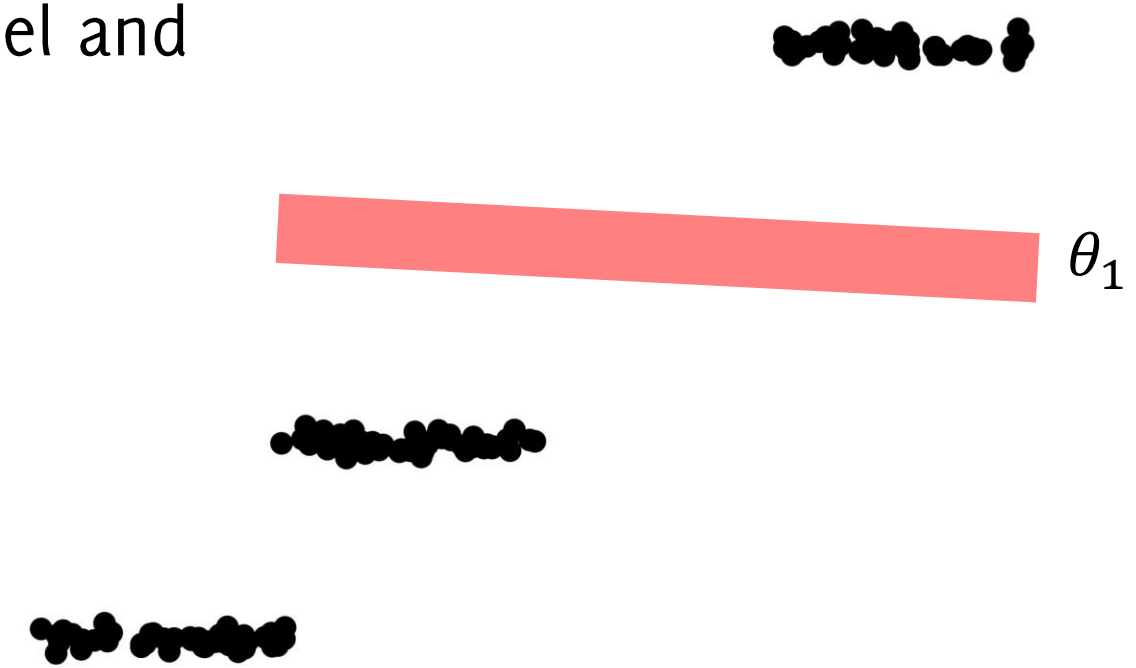
Once detected a model θ_1, keep the model and remove all the inliers

Line fitting example

Sequential RanSaC

Start RanSaC on the dataset X searching for the best fit for a single instance of the model

Once detected a model θ_1, keep the model and remove all the inliers from X



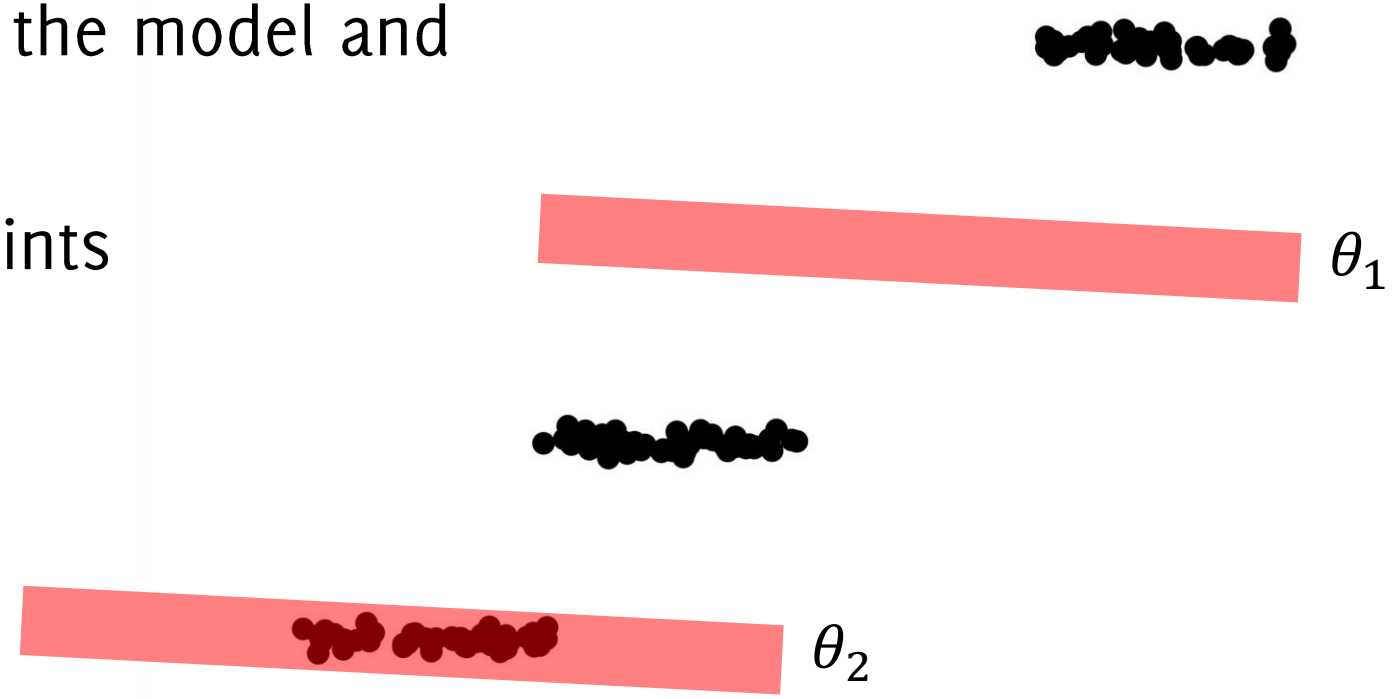
Line fitting example

Sequential RanSaC

Start RanSaC on the dataset X searching for the best fit for a single instance of the model

Once detected a model θ_1, keep the model and remove all the inliers from X

Iterate through the remaining points



Line fitting example

Sequential RanSaC

Start RanSaC on the dataset X searching for the best fit for a single instance of the model

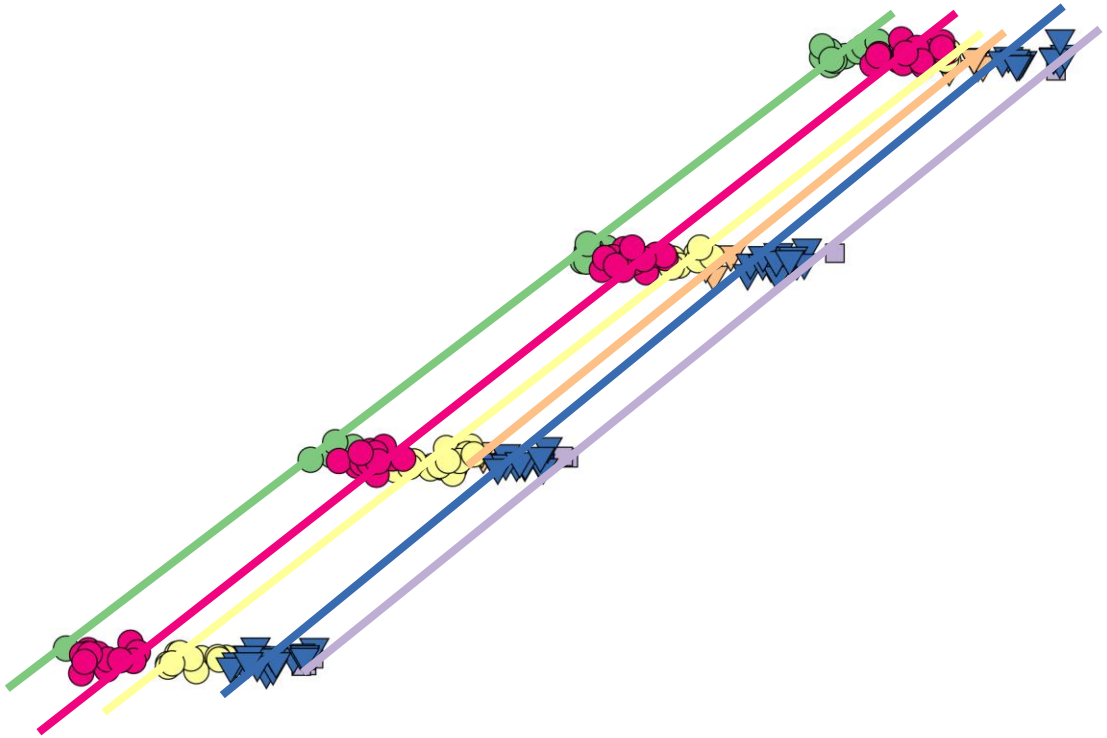
Once detected a model θ_1, keep the model and remove all the inliers from X

Iterate through the remaining points until there are no models with a sufficiently large consensus

Line fitting example

Sequential RanSaC

Unfortunately, this does not fit well with the multi-model scenario and the problem becomes even more sever in presence of outliers

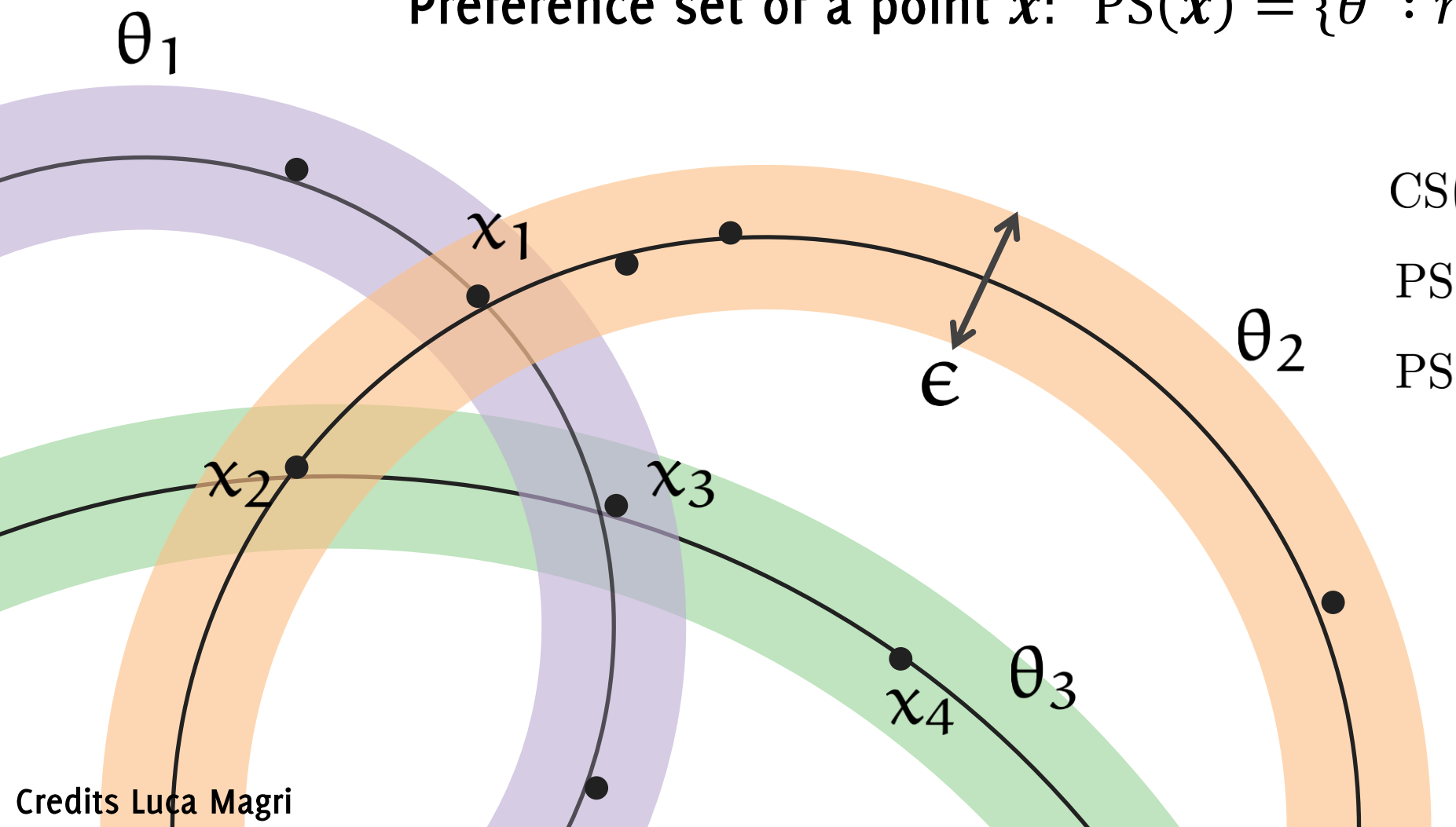


Preference-based methods

From model consensus to point preferences

Consensus set of a model θ: $\mathrm{CS}(\theta) = \{\boldsymbol{x} : r(x, \theta) < \epsilon\}$

Preference set of a point $\boldsymbol{x}$: $\mathrm{PS}(\boldsymbol{x}) = \{\theta : r(x, \theta) < \epsilon\}$



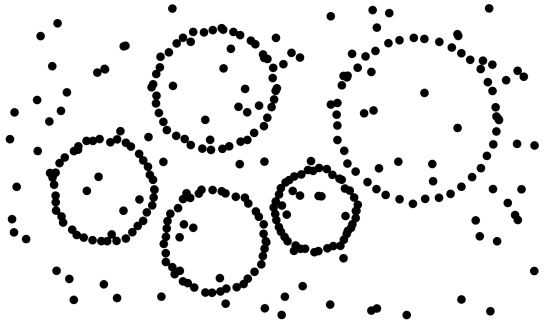
$$\mathrm{CS}(\theta_3) = \{x_2, x_3, x_4\}$$

$$\mathrm{PS}(x_1) = \{\theta_1, \theta_2\}$$

$$\mathrm{PS}(x_2) = \{\theta_2, \theta_3\}$$

Credits Luca Magri

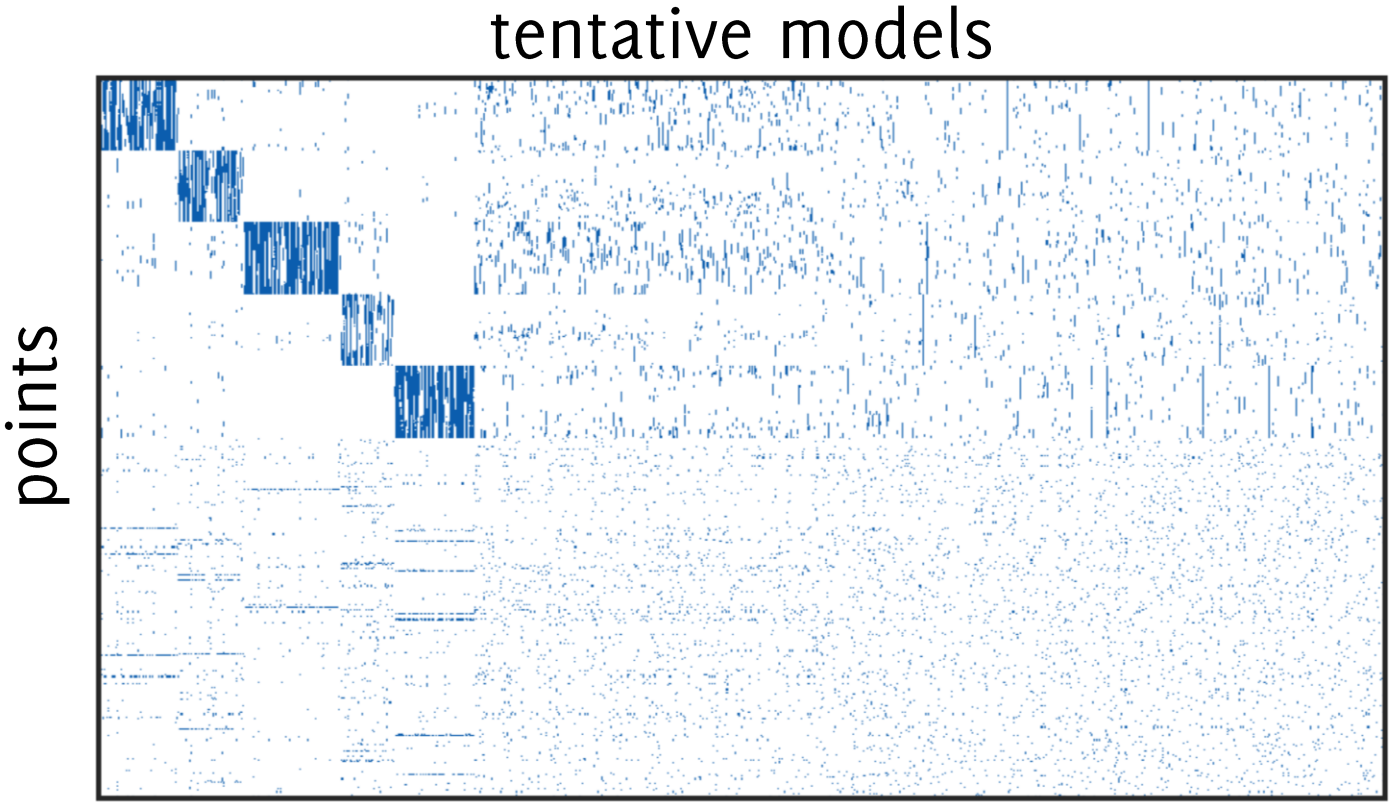
Multi model fitting: the preference trick



Generate a pool of m random models (as in RanSaC)

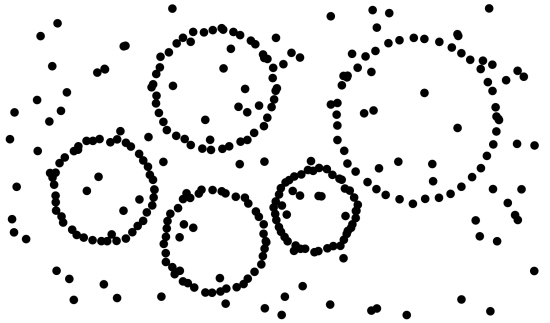
$$H = \{\theta_1, \theta_2, \ldots, \theta_m\}$$

Build the Preference Matrix



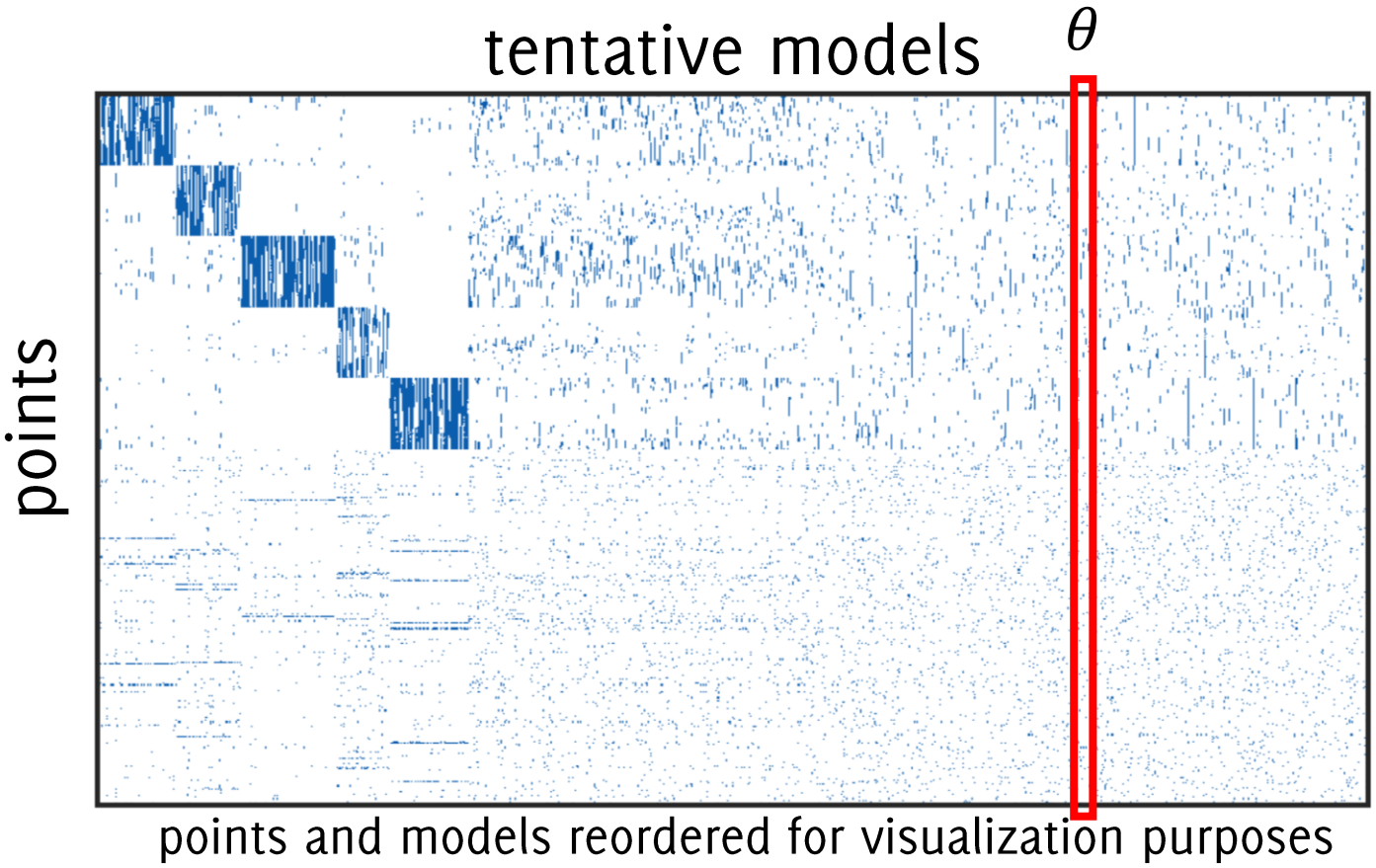
Credits Luca Magri

Multi model fitting: the preference trick



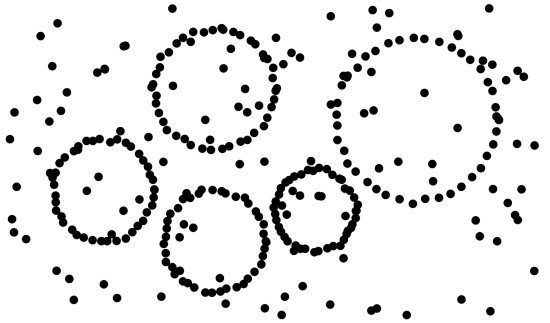
Consensus set of a model θ: $\mathrm{CS}(\theta) = \{\boldsymbol{x} : r(x, \theta) < \epsilon\}$

Preference set of a point $\boldsymbol{x}$: $\mathrm{PS}(\boldsymbol{x}) = \{\theta : r(x, \theta) < \epsilon\}$



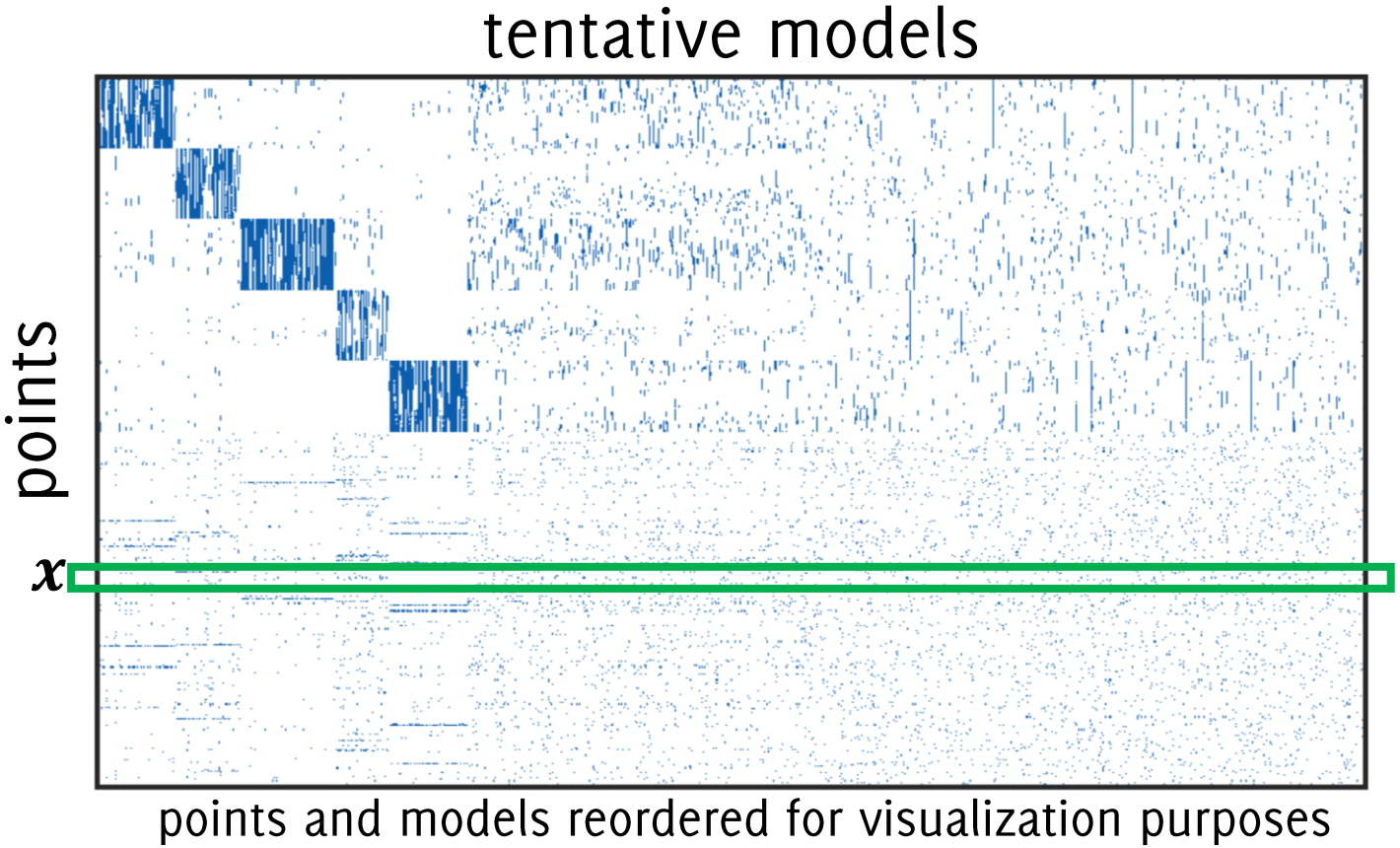
Credits Luca Magri

Multi model fitting: the preference trick



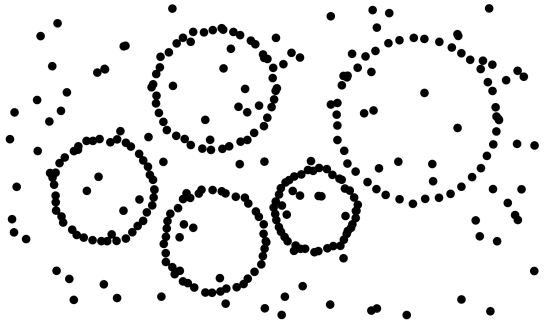
Consensus set of a model θ: $\mathrm{CS}(\theta) = \{\boldsymbol{x} : r(x, \theta) < \epsilon\}$

Preference set of a point $\boldsymbol{x}$: $\mathrm{PS}(\boldsymbol{x}) = \{\theta : r(x, \theta) < \epsilon\}$



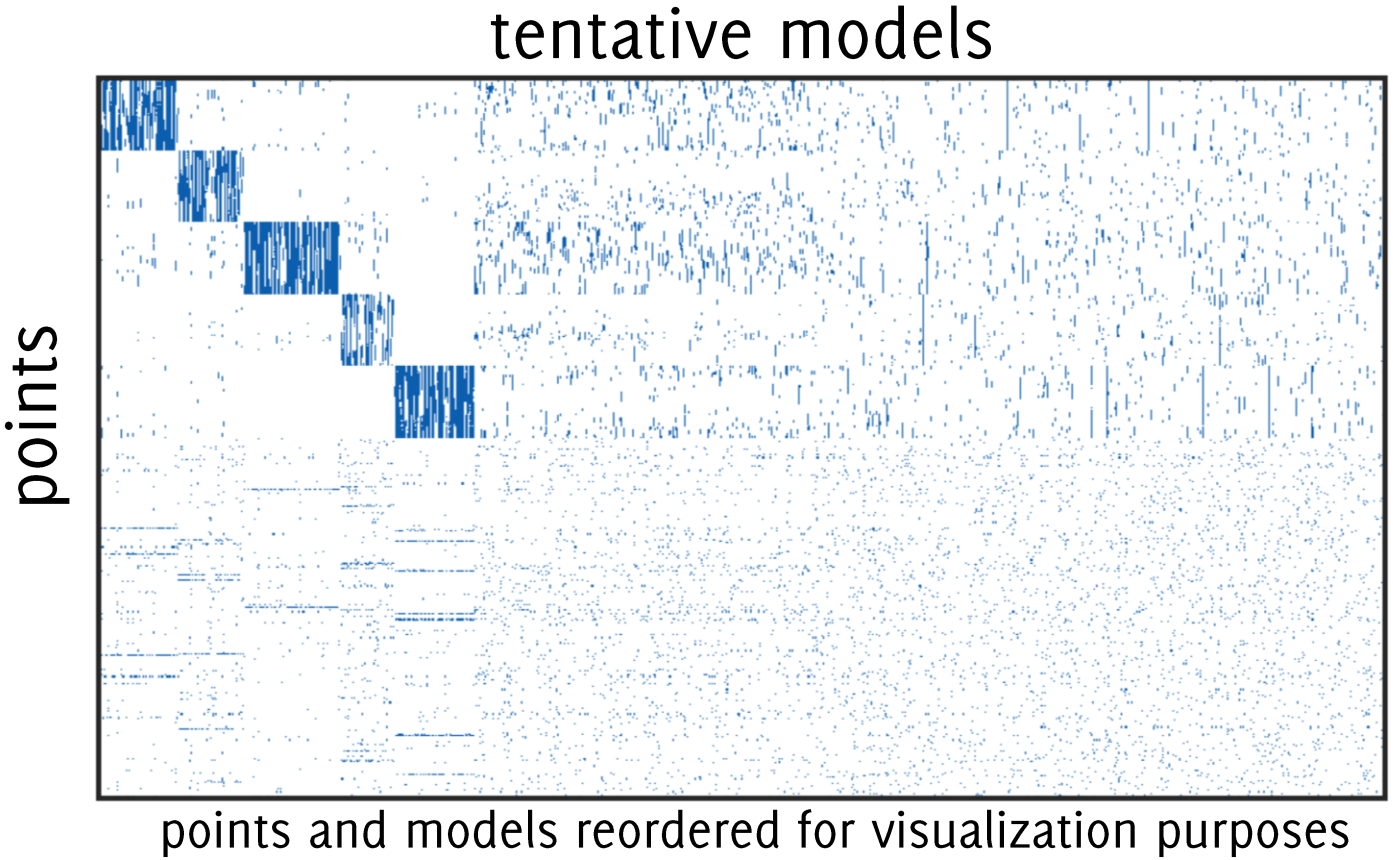
Credits Luca Magri

Multi model fitting: the preference trick



Point $\longleftrightarrow$ subset of preferred sampled models

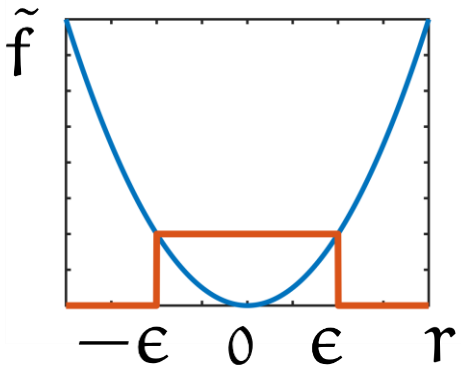
Block diagonal matrix $\Longrightarrow$ point of the same structure have similar preferences



Credits Luca Magri

Multi model fitting: a lift to Preference Space

Voting function

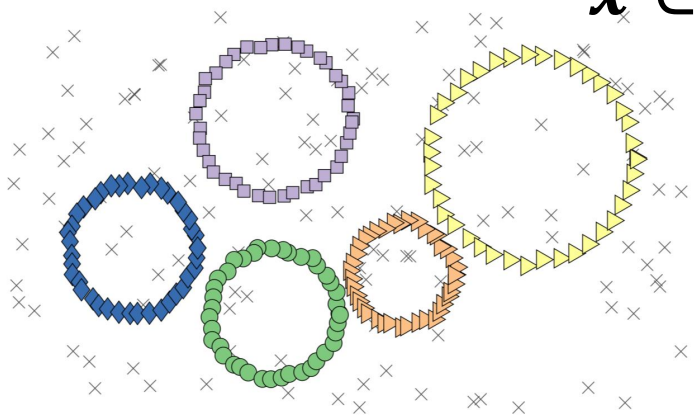


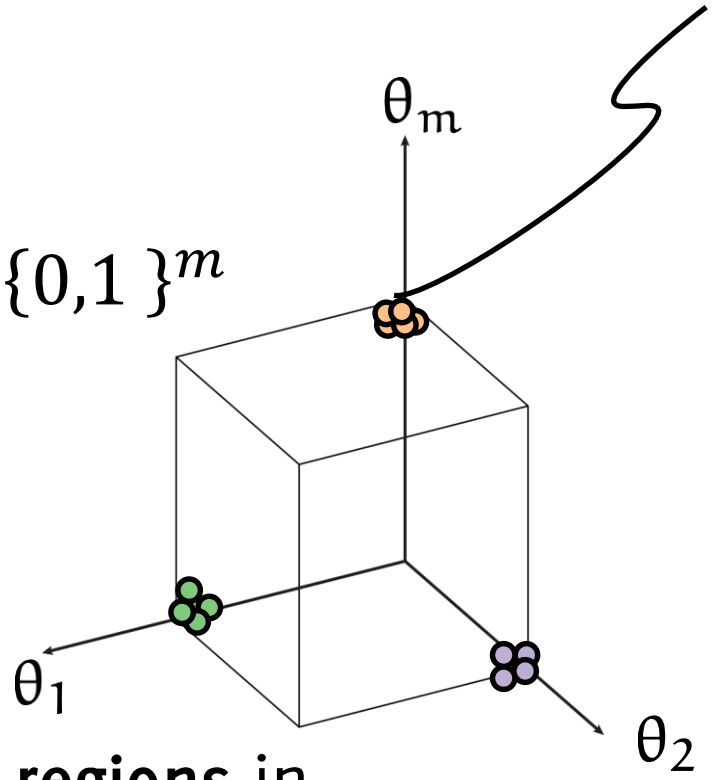
Vector of binary votes to sampled models

$$x \mapsto [\hat{f}(r(x,\theta_1)),\ldots,\hat{f}(r(x,\theta_m))] \in \{0,1\}^m$$

$\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$

$\mathrm{PS}(\boldsymbol{x}) \in \{0,1\}^{m}$



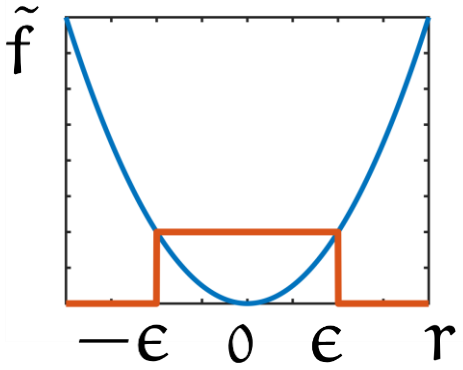


Structures correspond to **high-density regions** in the preference space

Credits Luca Magri

Multi model fitting: a lift to Preference Space

Voting function

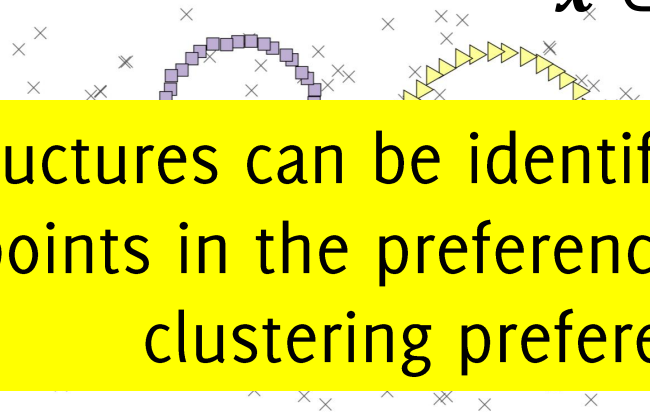


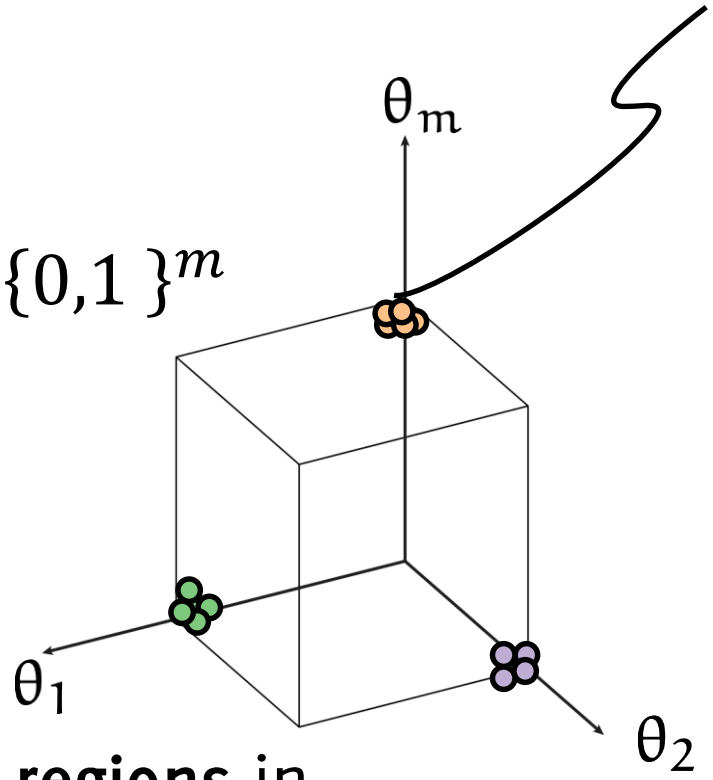
Vector of binary votes to sampled models

$$x \mapsto [\hat{f}(r(x,\theta_1)),\ldots,\hat{f}(r(x,\theta_m))] \in \{0,1\}^m$$

$\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$

$\mathrm{PS}(\boldsymbol{x}) \in \{0,1\}^{m}$





Structures can be identified by clustering points in the preference space, i.e. by clustering preference sets.

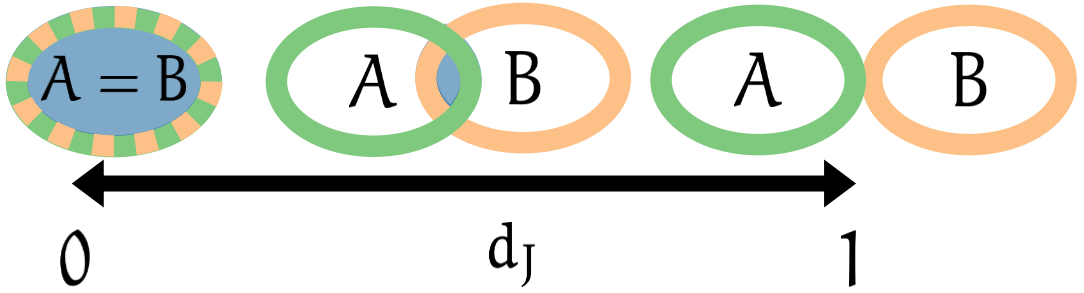
Structures correspond to **high-density regions** in the preference space

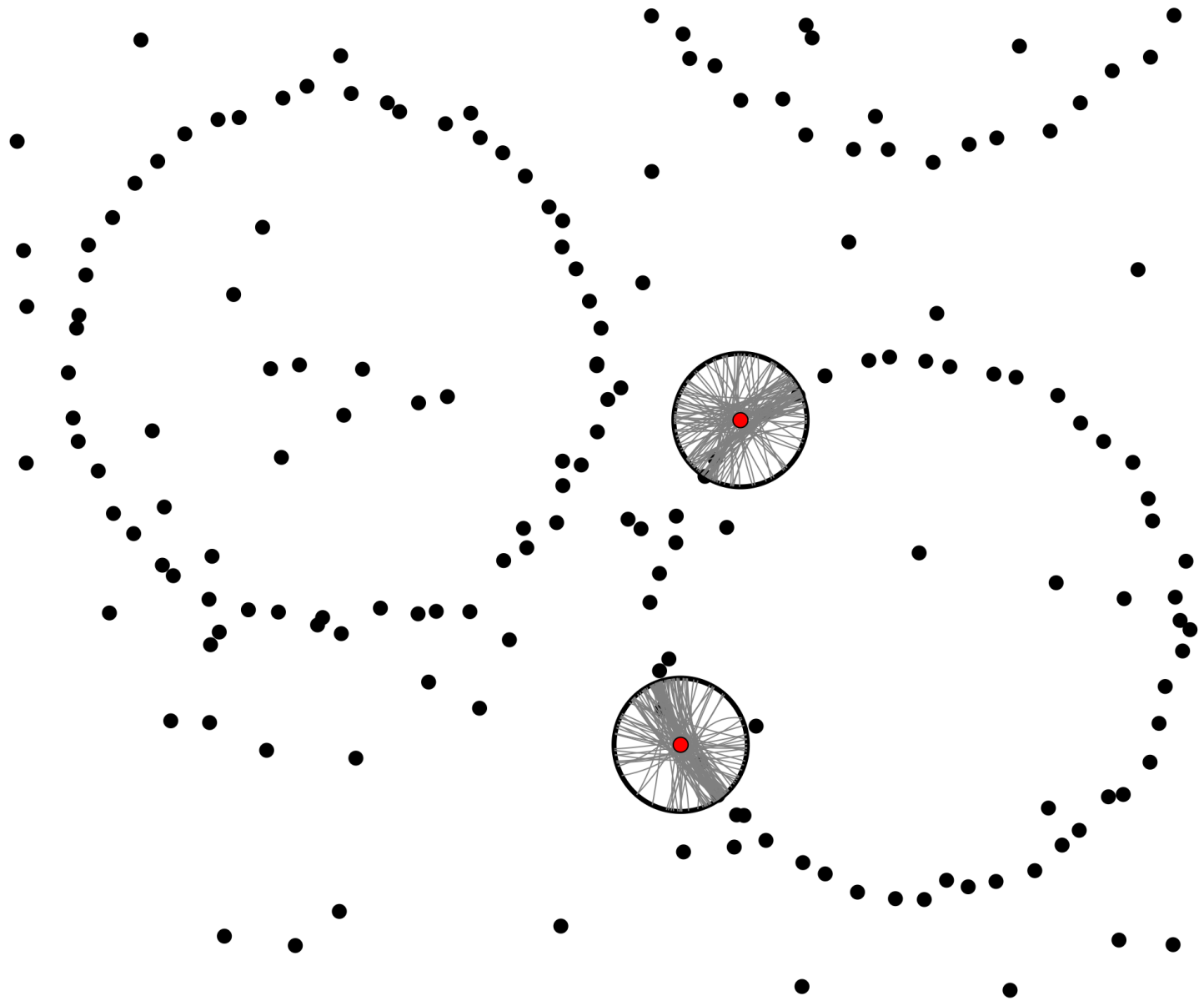
Credits Luca Magri

Multi model fitting: a lift to Preference Space

The Jaccard distance can be used to measure distance between preference sets.

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$



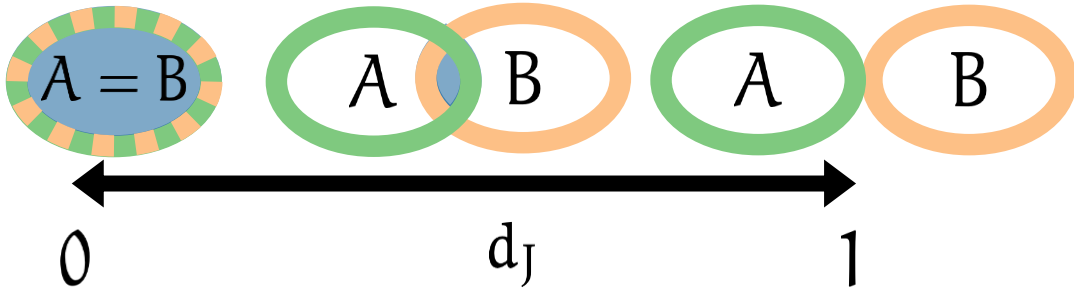


Credits Luca Magri

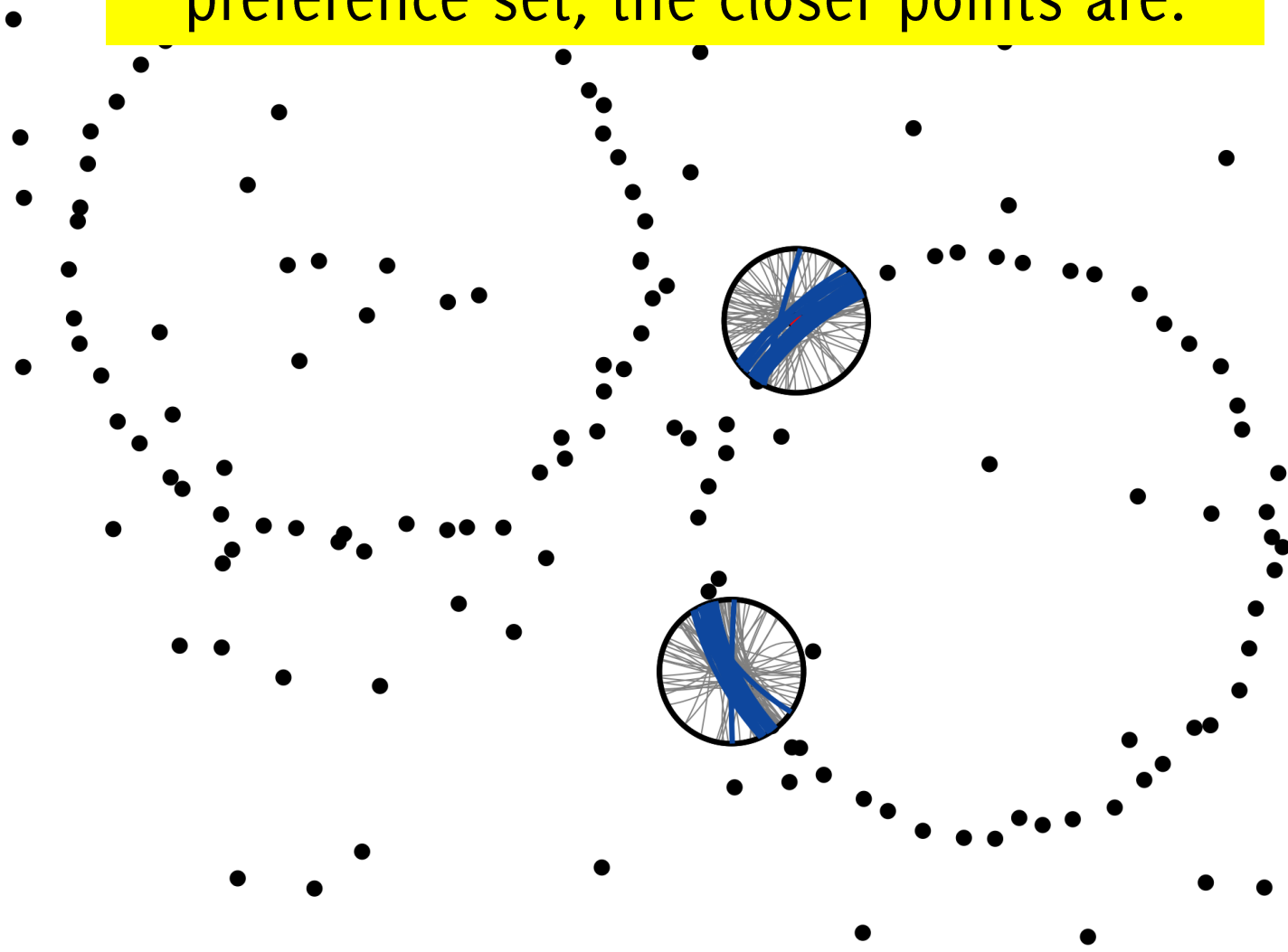
Multi model fitting: a lift to Preference Space

The Jaccard distance can be used to measure distance between preference sets.

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$



the more models in common in their preference set, the closer points are.

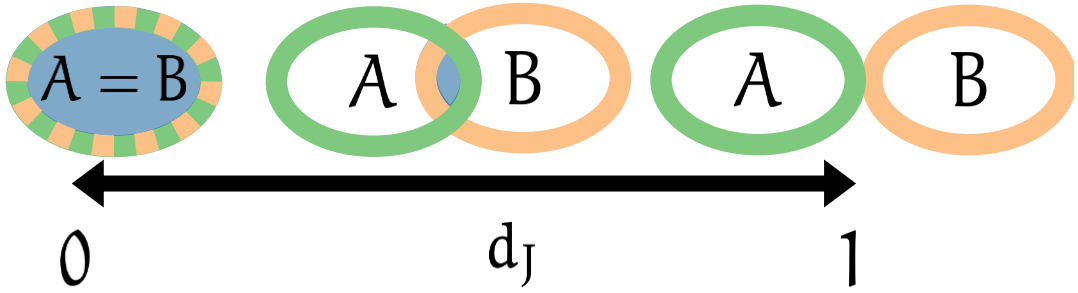


Credits Luca Magri

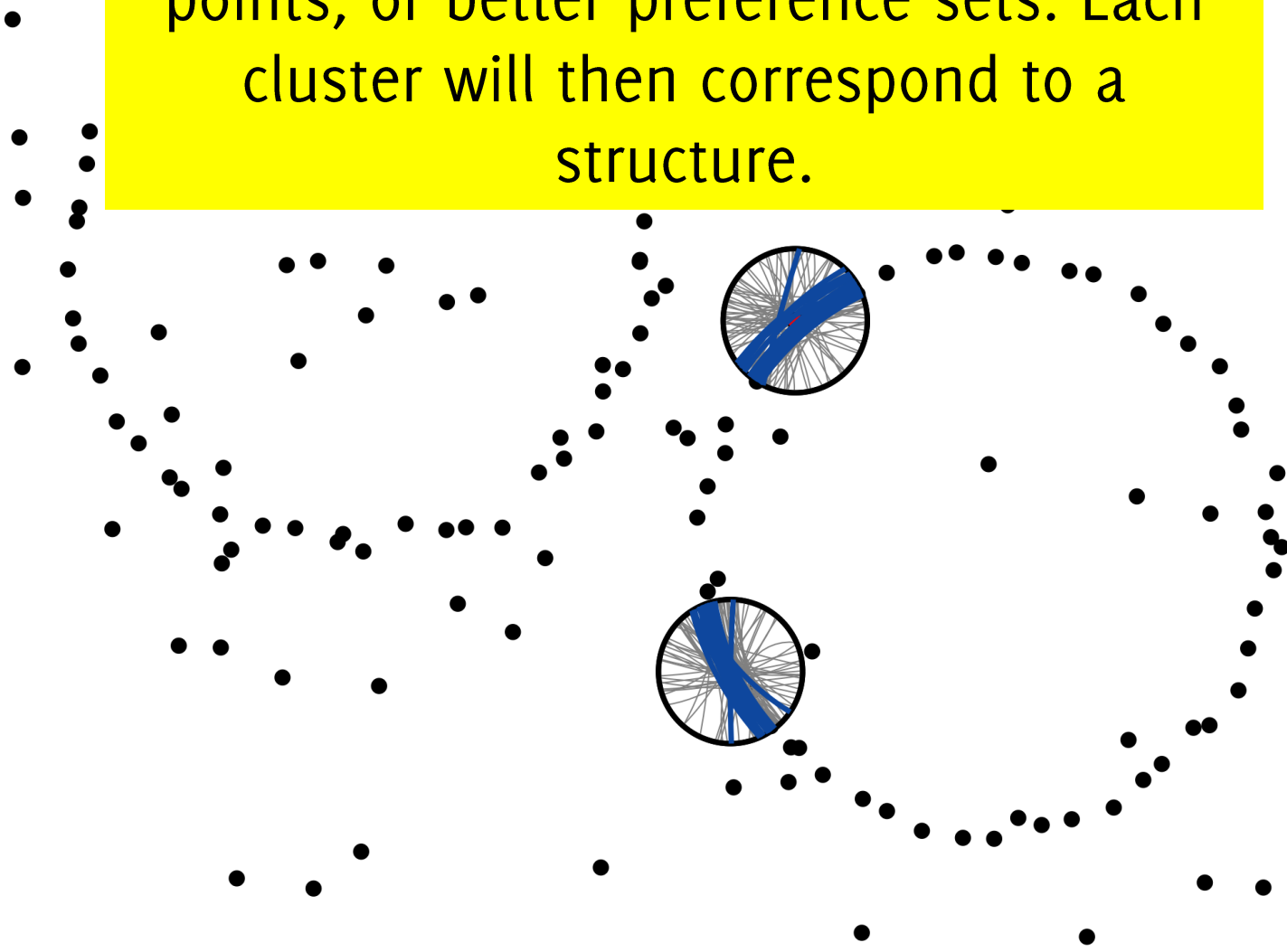
Identify Structures by Clustering Preferences

The Jaccard distance can be used to measure distance between preference sets.

$$d_J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$



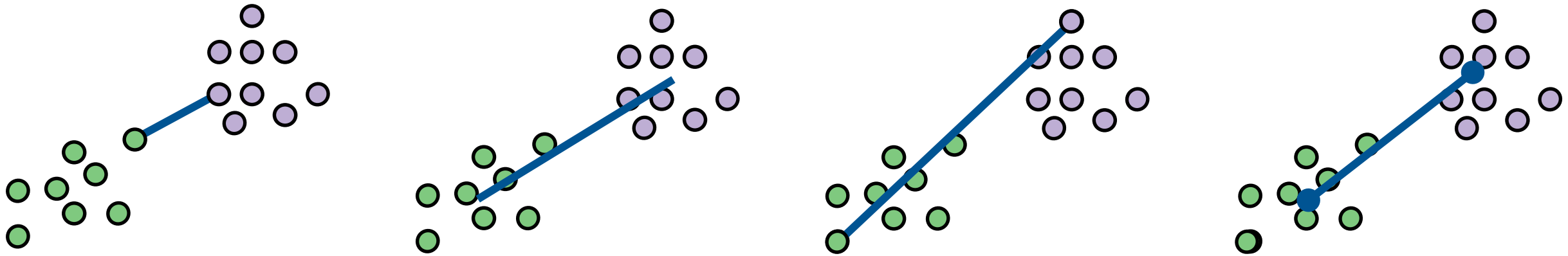
Structures are identified by clustering points, or better preference sets. Each cluster will then correspond to a structure.



Credits Luca Magri

Structure Identification by Clustering in PS

Hierarchical clustering can be used in the Preference Space to recover the structures.



Single linkage Average linkage Complete linkage Centroid linkage

Distances are measured in the Preference Space, and each element of clustering is identified by a preference set.

Instead of using centroids, we derive a conceptual representation for each cluster (which also lives in the preference space).

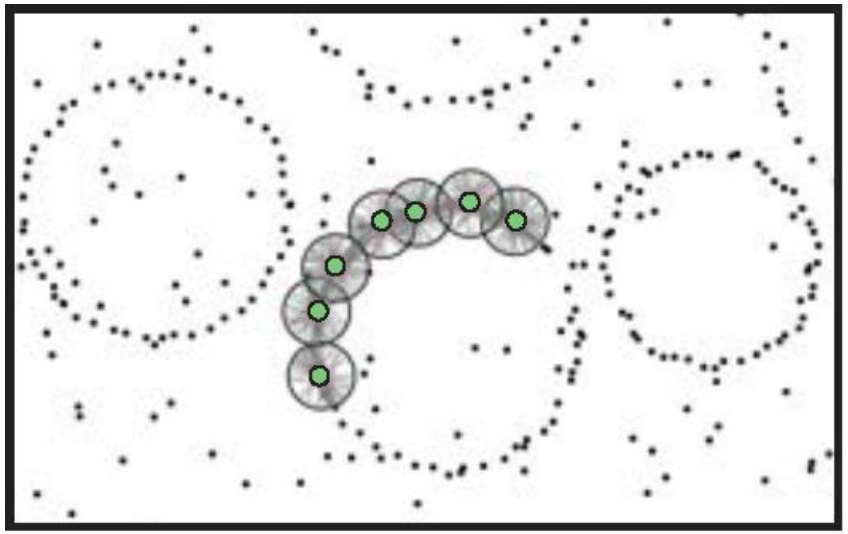
Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

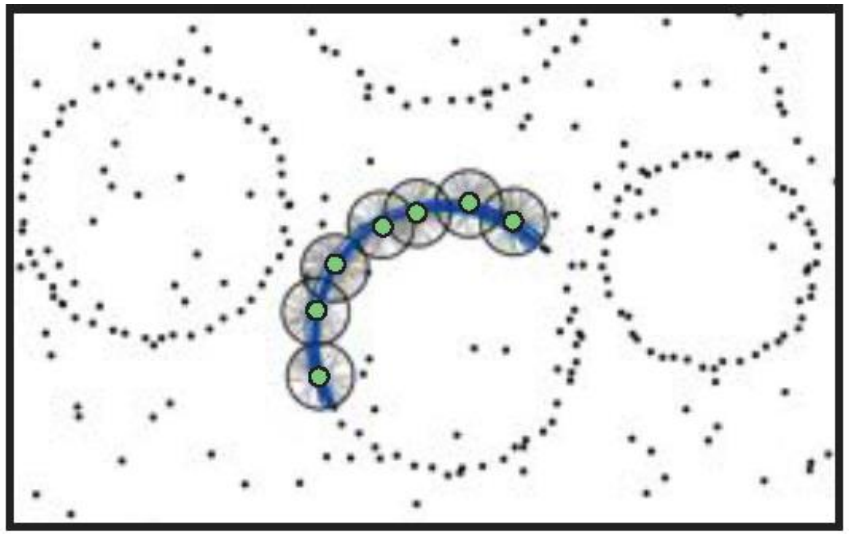
The representation of a cluster $U \subset X$ in the preference space is the intersection of the PS of its points

$$\mathrm{U} \subseteq \mathrm{X}, \mathrm{PS}(\mathrm{U}) = \bigcap_{\mathrm{x} \in \mathrm{U}} \mathrm{PS}(\mathrm{x})$$

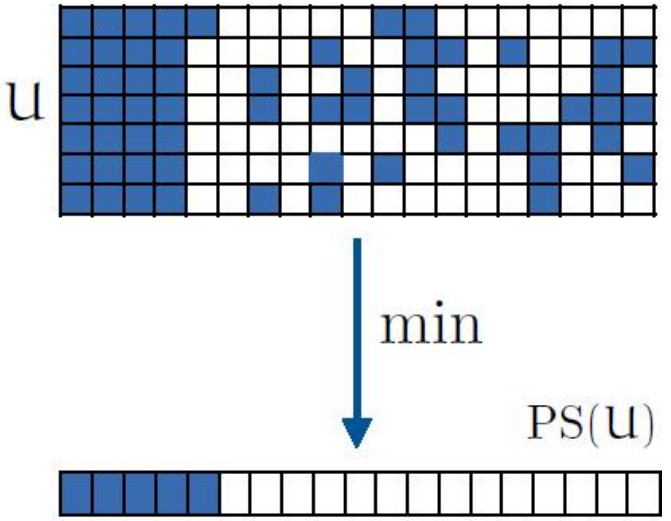
This is the component-wise min of rows in the preference matrix.



$\mathrm{U} \subseteq \mathrm{X}$



$\mathrm{PS}(\mathrm{U})$



Credits Luca Magri

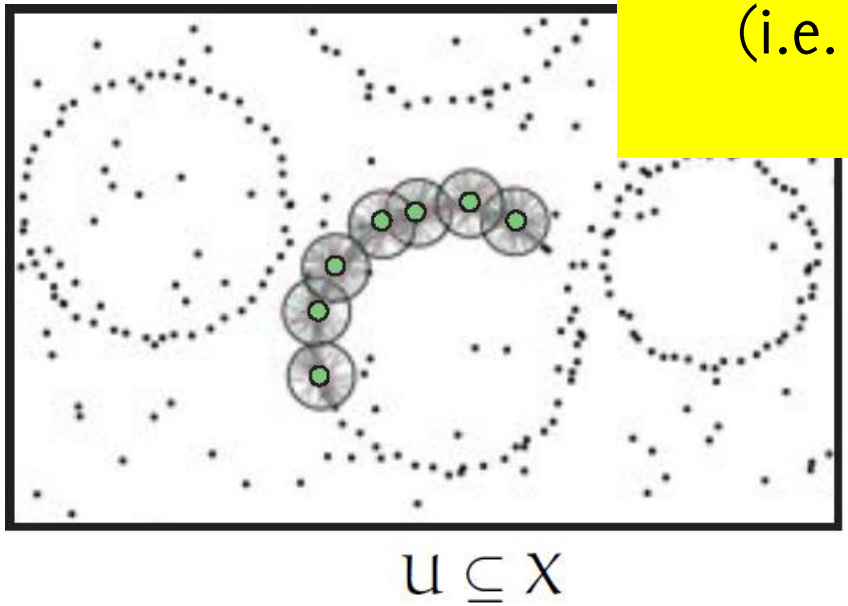
J-linkage clustering [Toldo and Fusiello, ECCV 08]

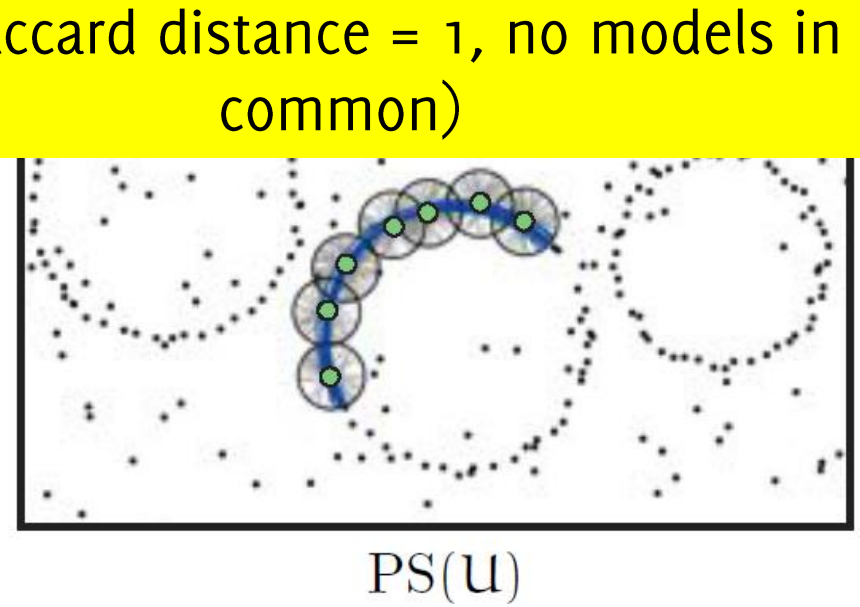
The representation of a cluster $U \subset X$ in the preference space is the intersection of the PS of its points

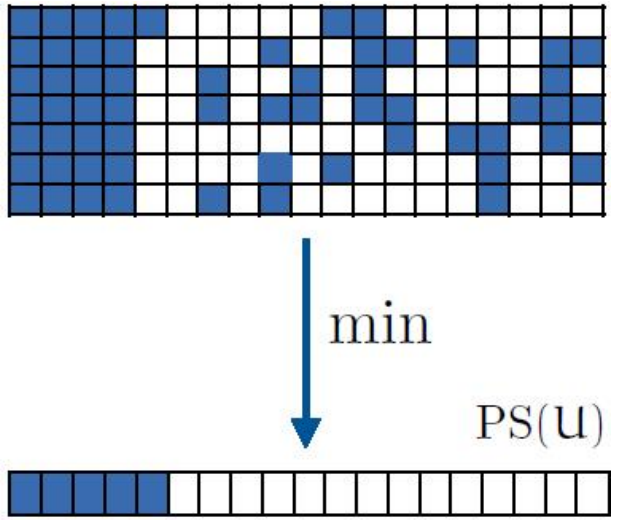
$$U \subset X, \mathrm{PS}(U) = \bigcap_{x} \mathrm{PS}(x)$$

This is the compor natrix.

J-linkage iterates a hierarchical clustering scheme based on this distance until all the representatives of the clusters are disjoint (i.e. Jaccard distance = 1, no models in common)

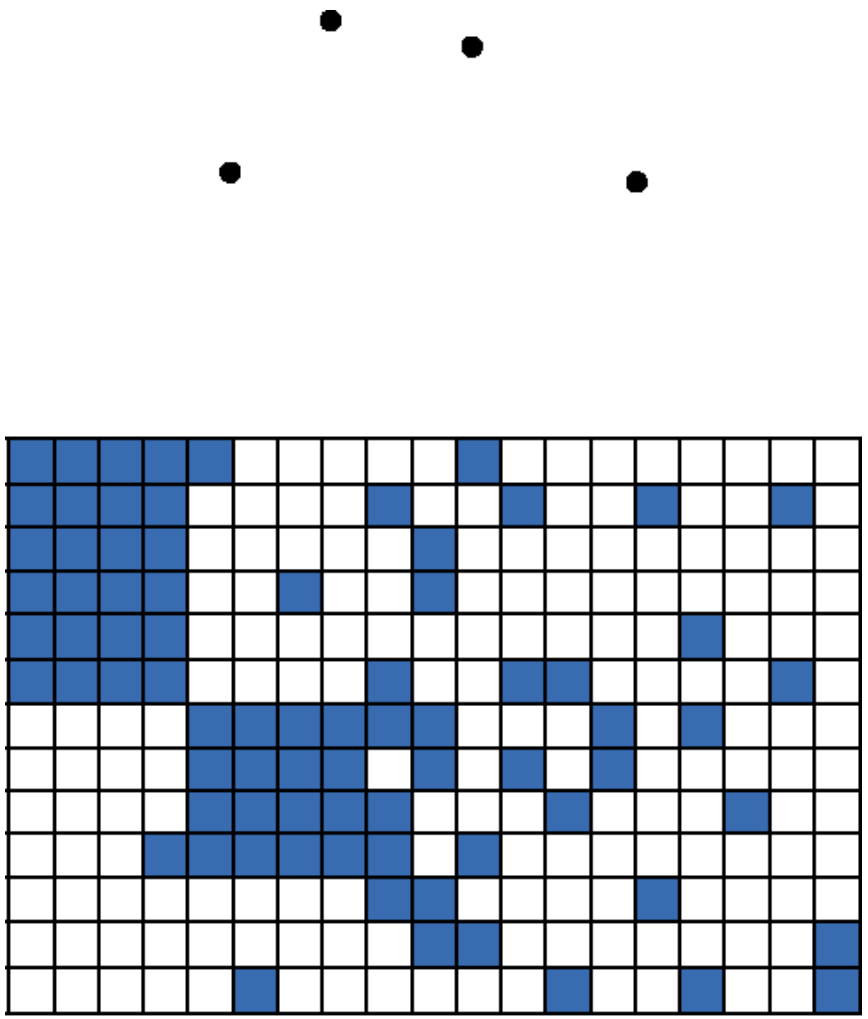






Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]



Preference matrix

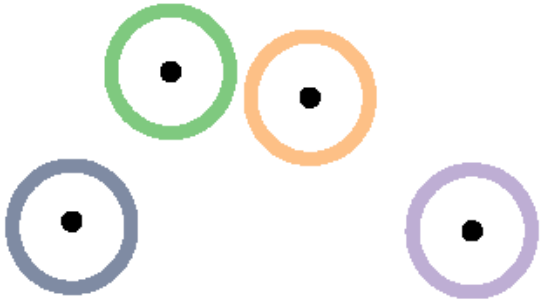
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

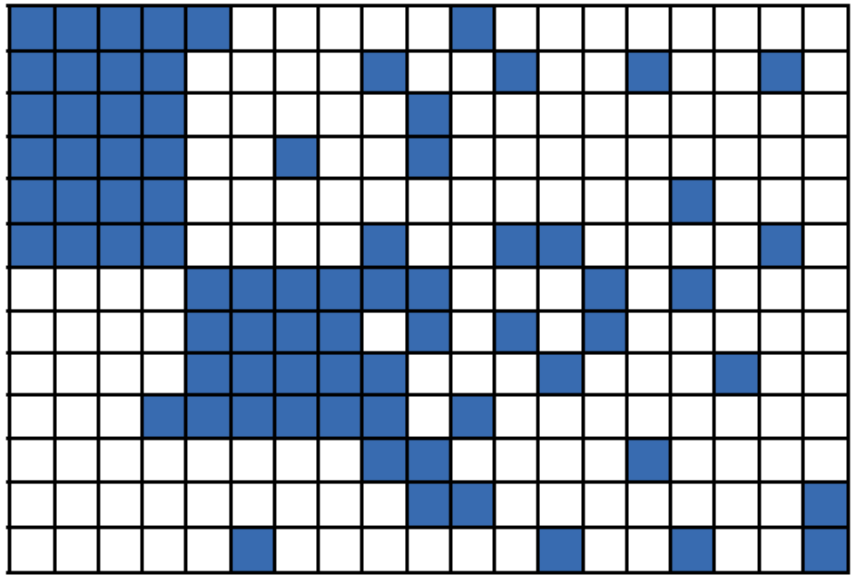
Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Put each point in its own cluster





Preference matrix

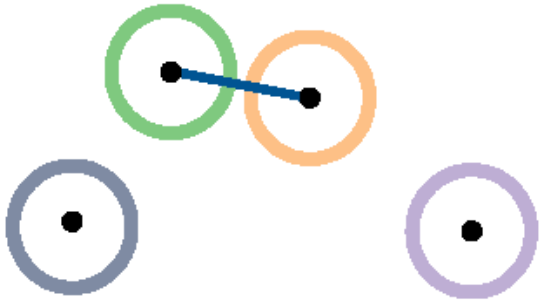
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

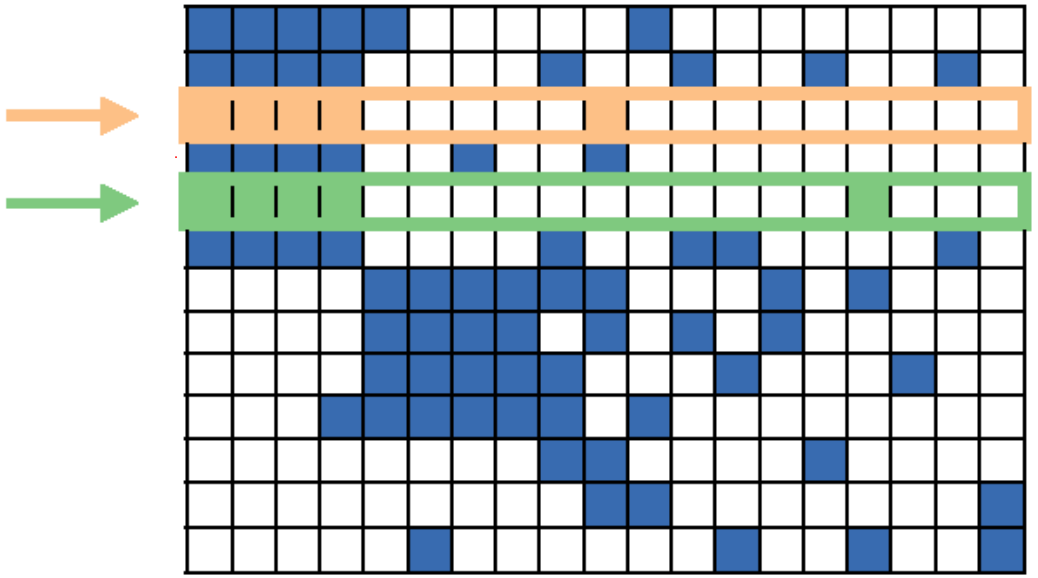
Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Find the closest points in Preference Space





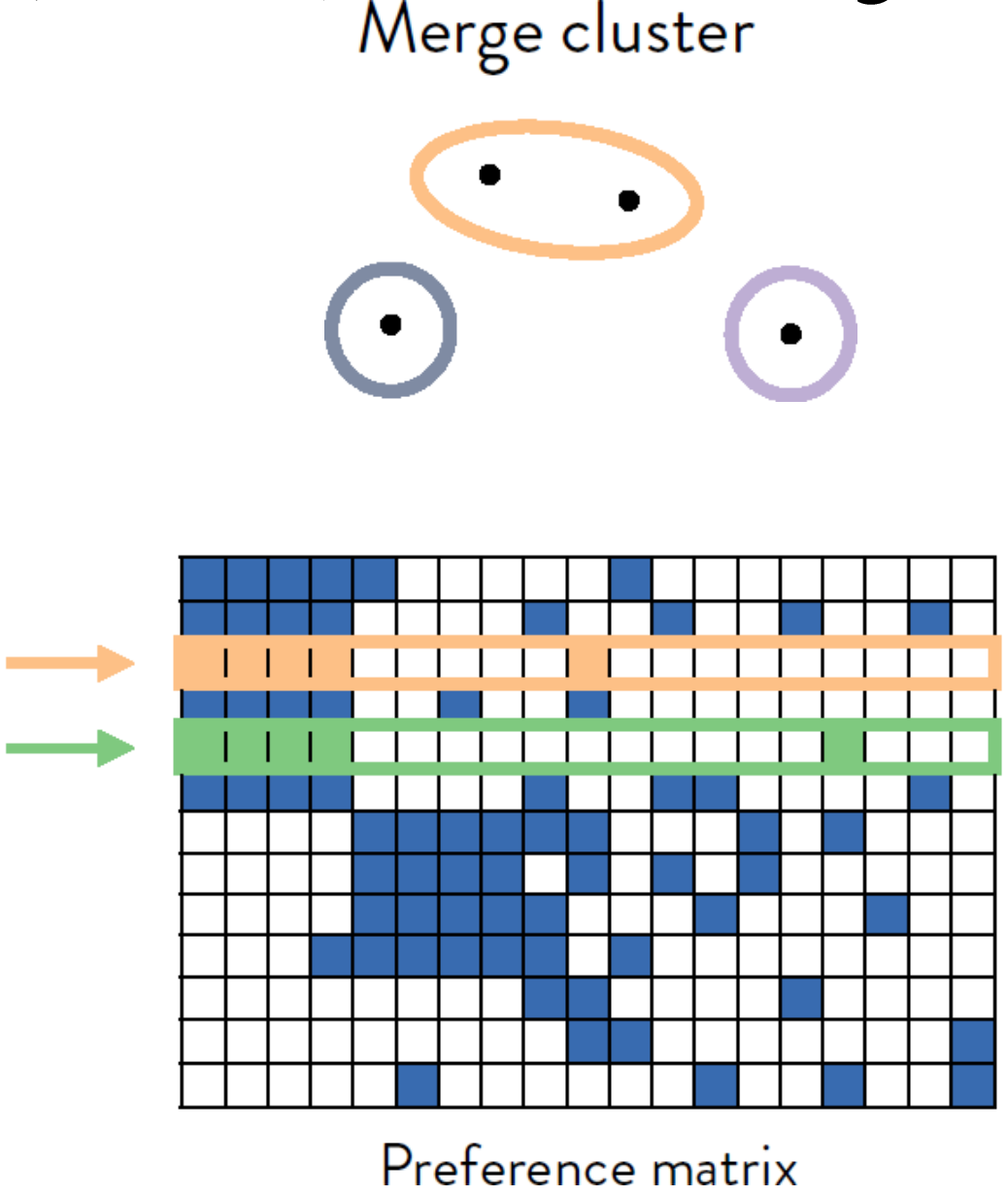
Preference matrix

Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
  Find pair (C_i, C_j) of clusters with the min d_J
  Replace the clusters with their union;
  Compute the PS of $C_i \cup C_j$;
  Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

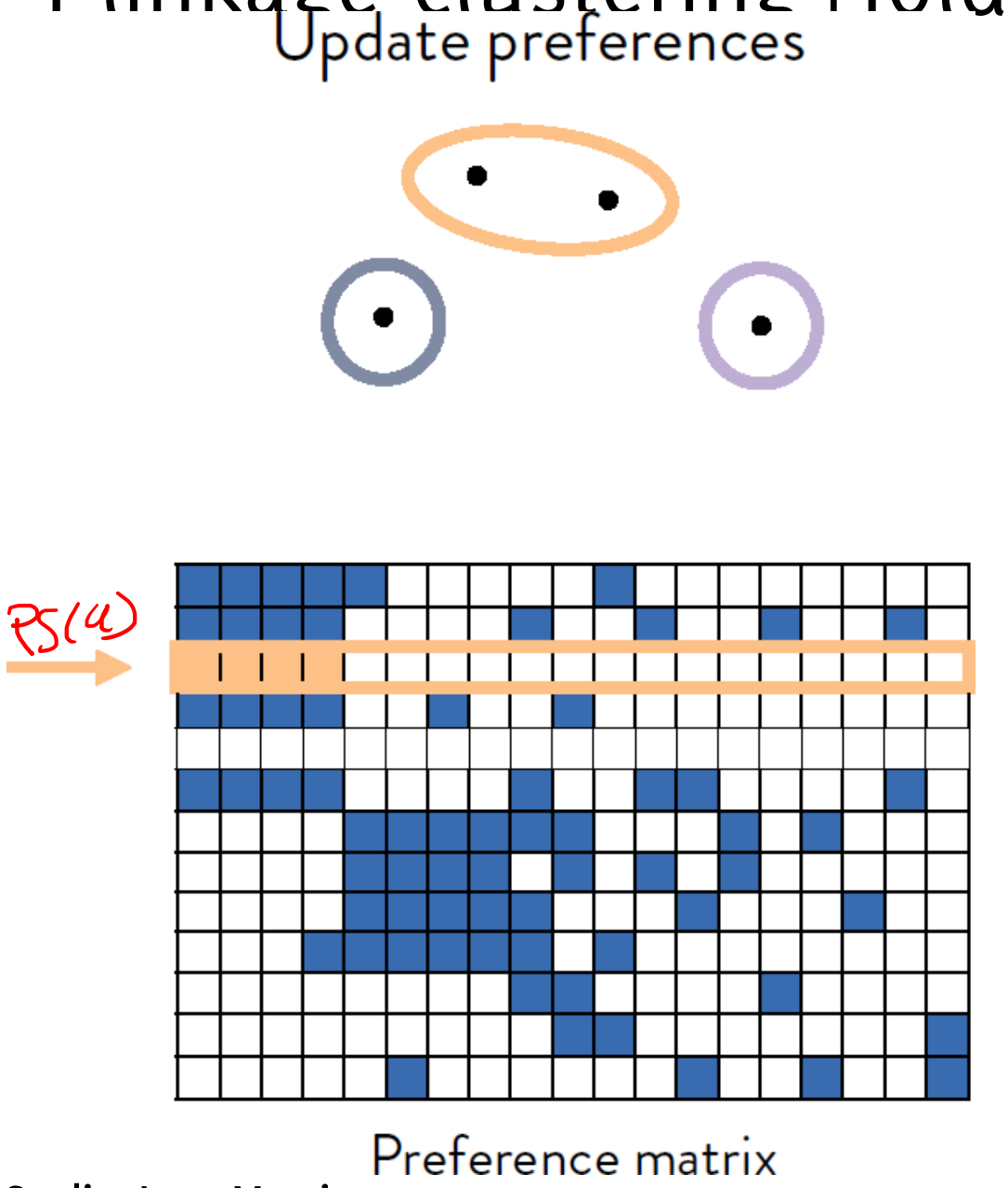


Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]



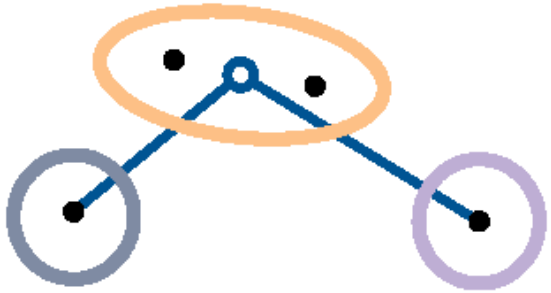
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

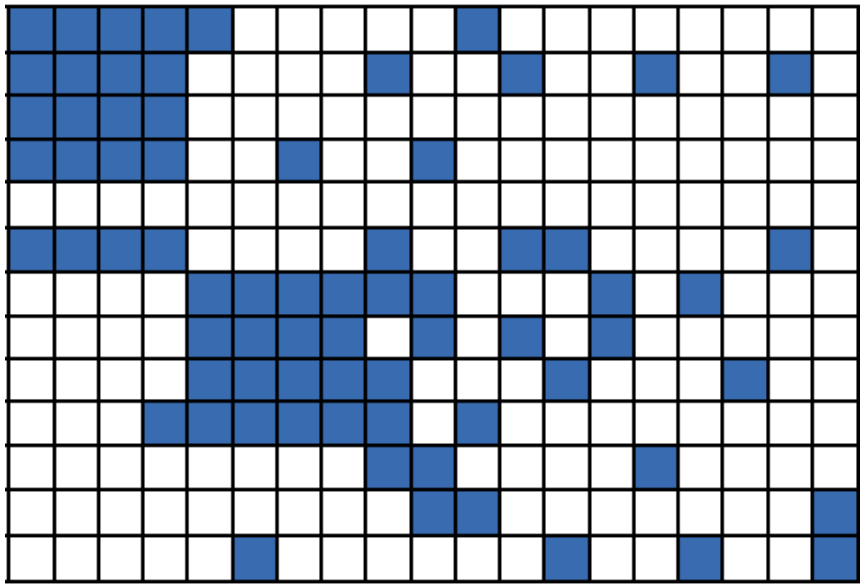
Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Update distances





Preference matrix

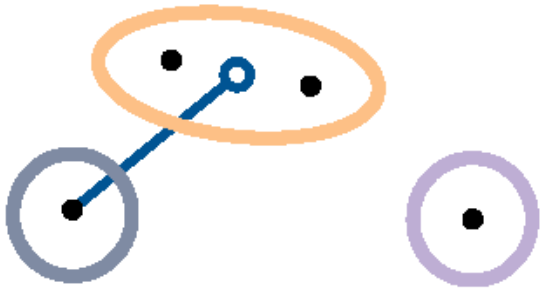
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

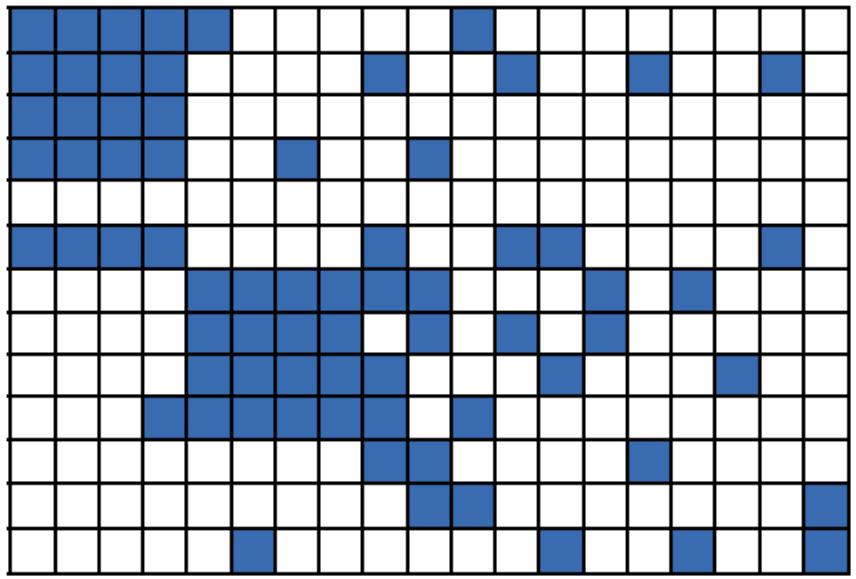
Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Continue until all PS are disjoint...





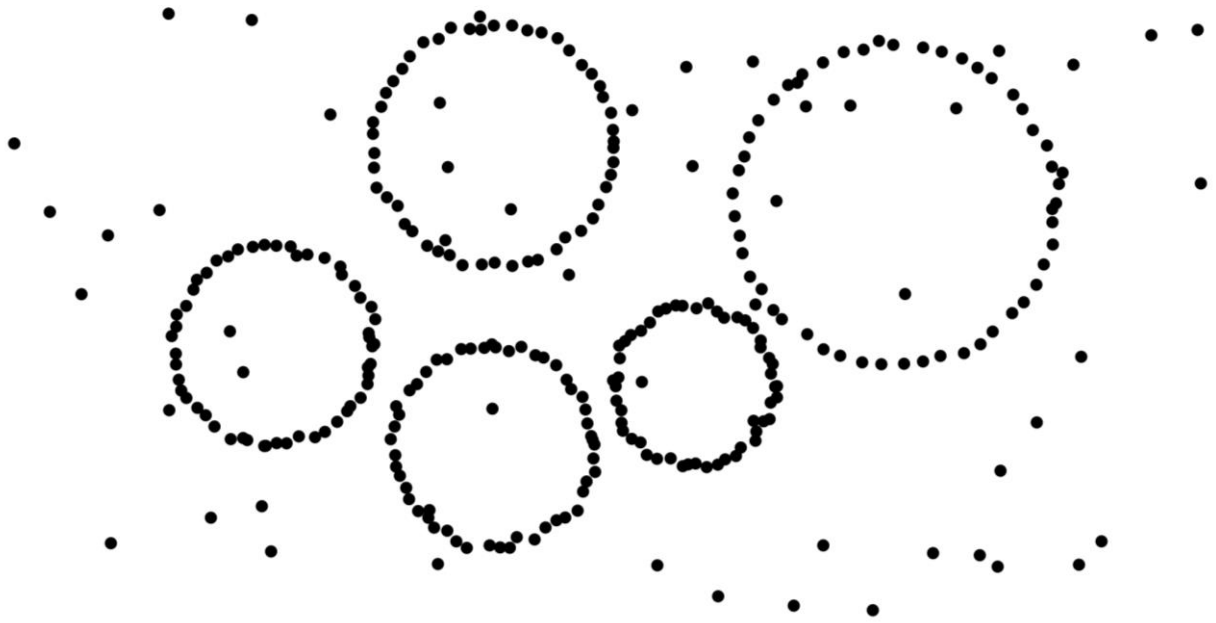
Preference matrix

Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

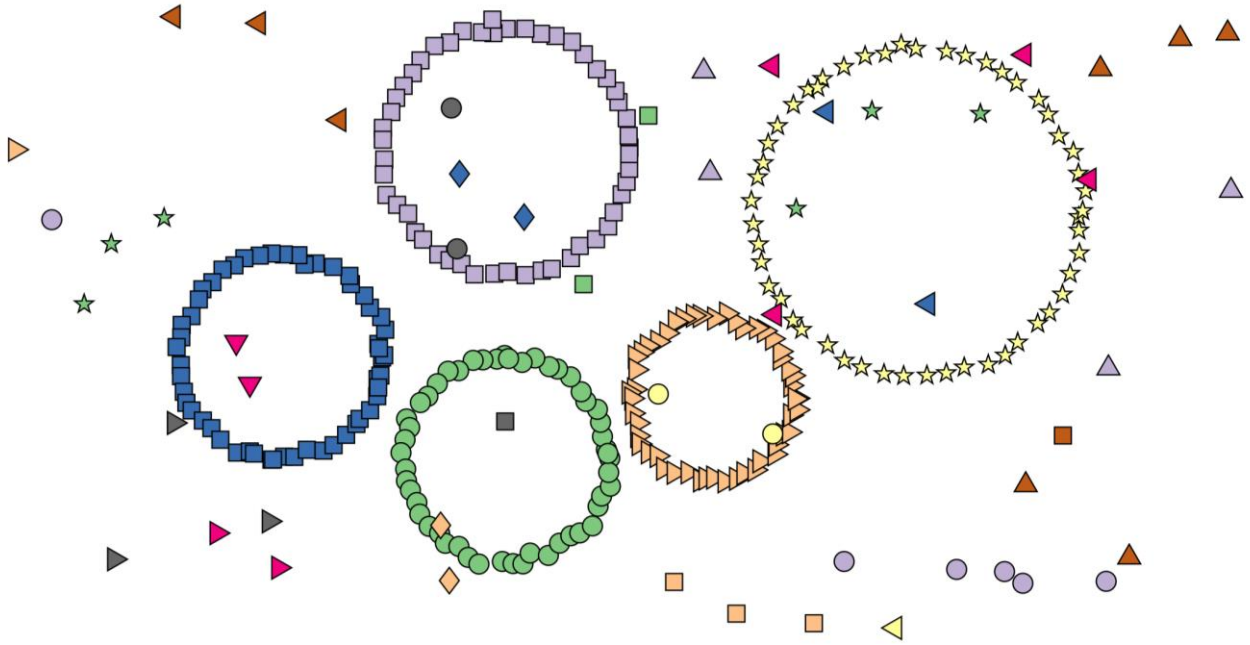


Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]



Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

Credits Luca Magri

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Pro:

- The number of structures is automatically determined
- For each **cluster there exists at least one model** that fits all the points of the cluster
- **Clusters are "maximal"** in the sense that does not exist a model that explain all the points of two distinct clusters

Cons:

- It still relies on a pre-defined inlier threshold ϵ
- **Outliers have to be filtered out a posteriori** (they will always end up in small clusters)

Credits Luca Magri

Object Detection by Computer Vision Features

Object Detection: Keypoint Extraction

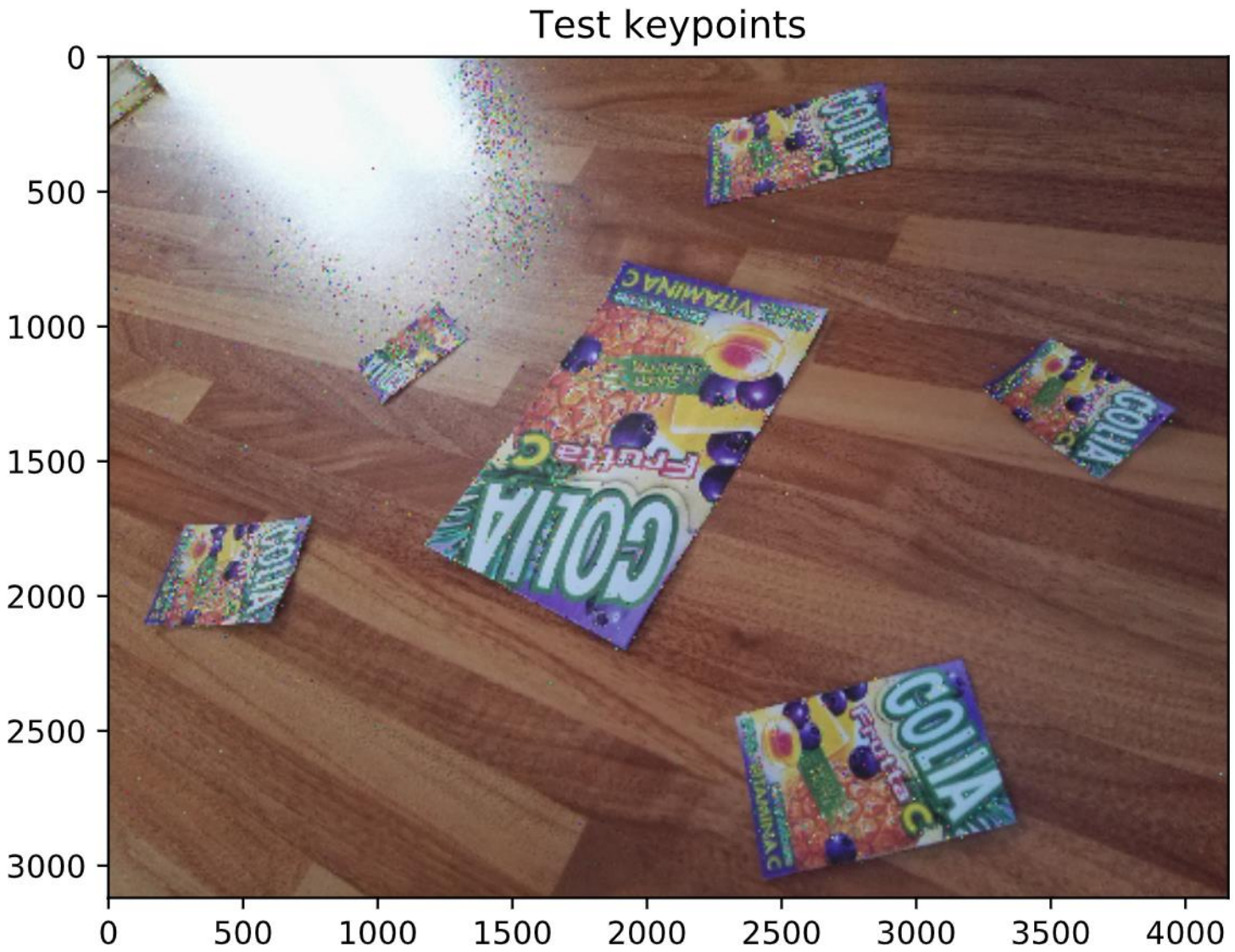


Image Courtesy of Filippo Leveni

Object Detection: Keypoint Extraction

Image Courtesy of Filippo Leveni

Keypoint Matching

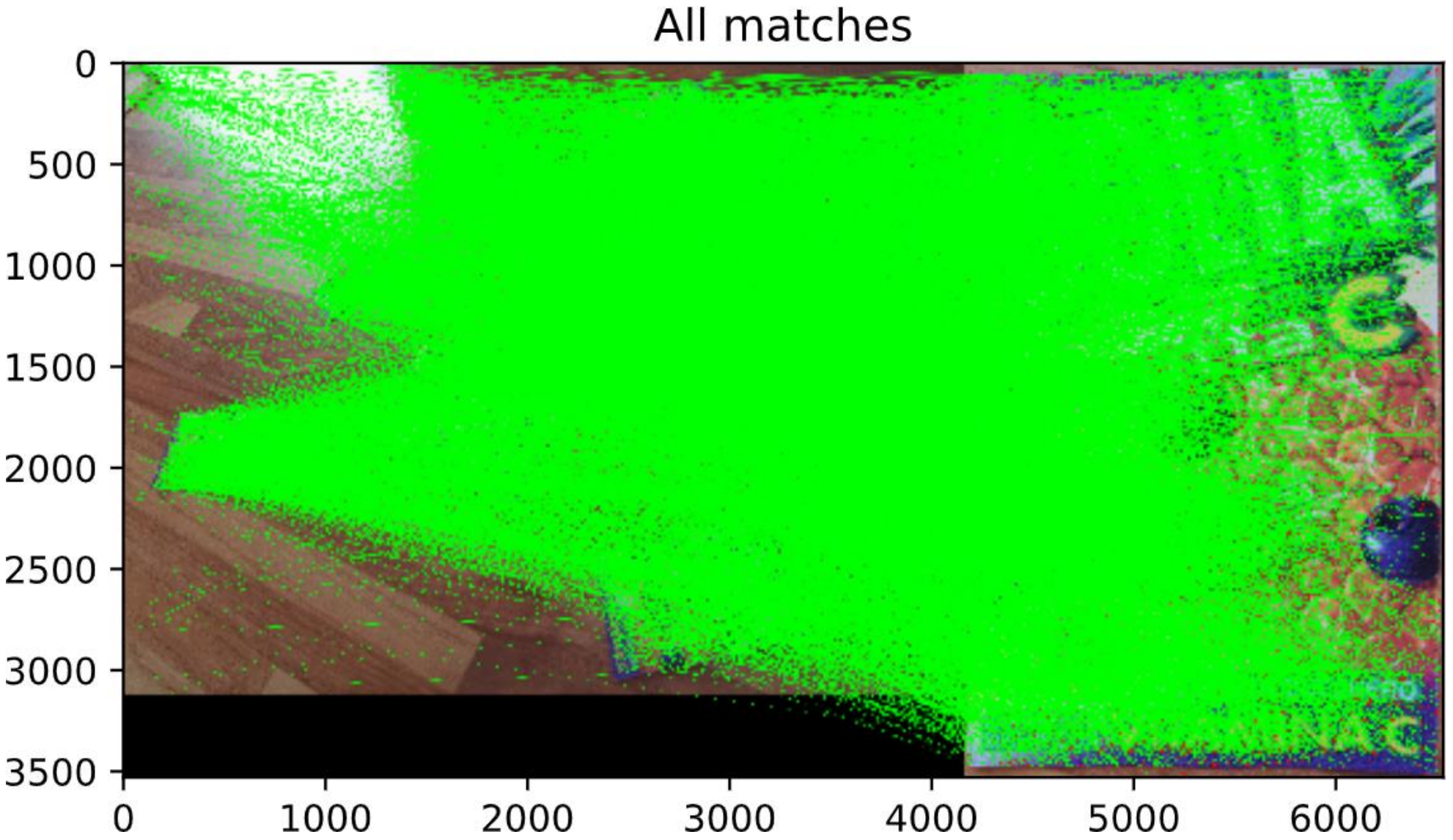


Image Courtesy of Filippo Leveni

Keypoint Matching: Ratio Test

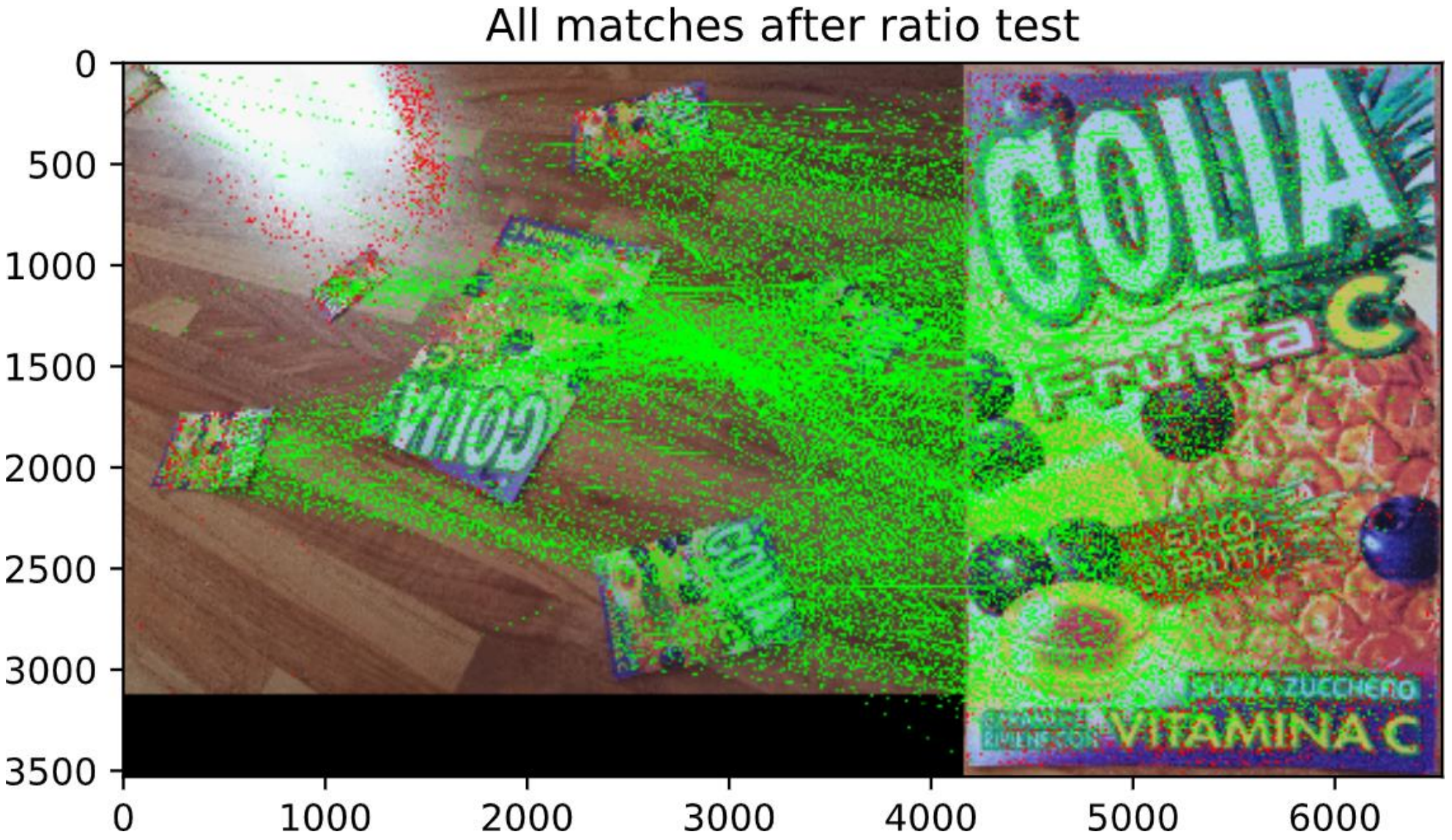


Image Courtesy of Filippo Leveni

Sequential Ransac Iterations

Image Courtesy of Filippo Leveni

Sequential Ransac Image Rectification

Image Courtesy of Filippo Leveni

Sequential Ransac: Histogram Matching

Image Courtesy of Filippo Leveni

Sequential Ransac: Image Differences

Image Courtesy of Filippo Leveni

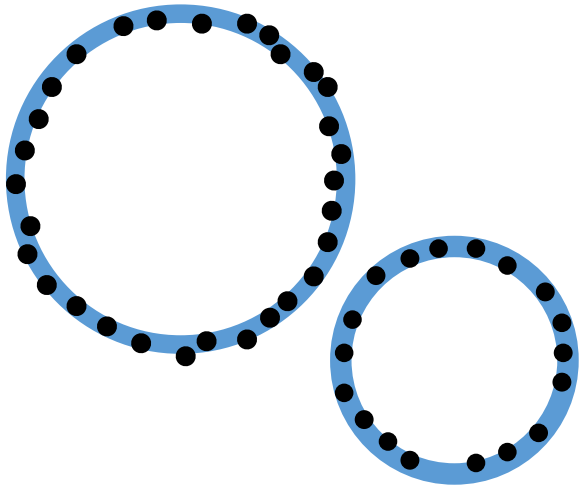
Sequential Ransac

Image Courtesy of Filippo Leveni

Tempalte matching limitation (it is not cherry creme)

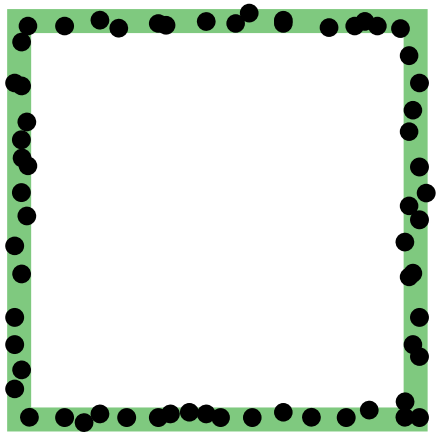
Towards multi-class model fitting

By using T-linkage we are able to detect separately



Multiple circles

or

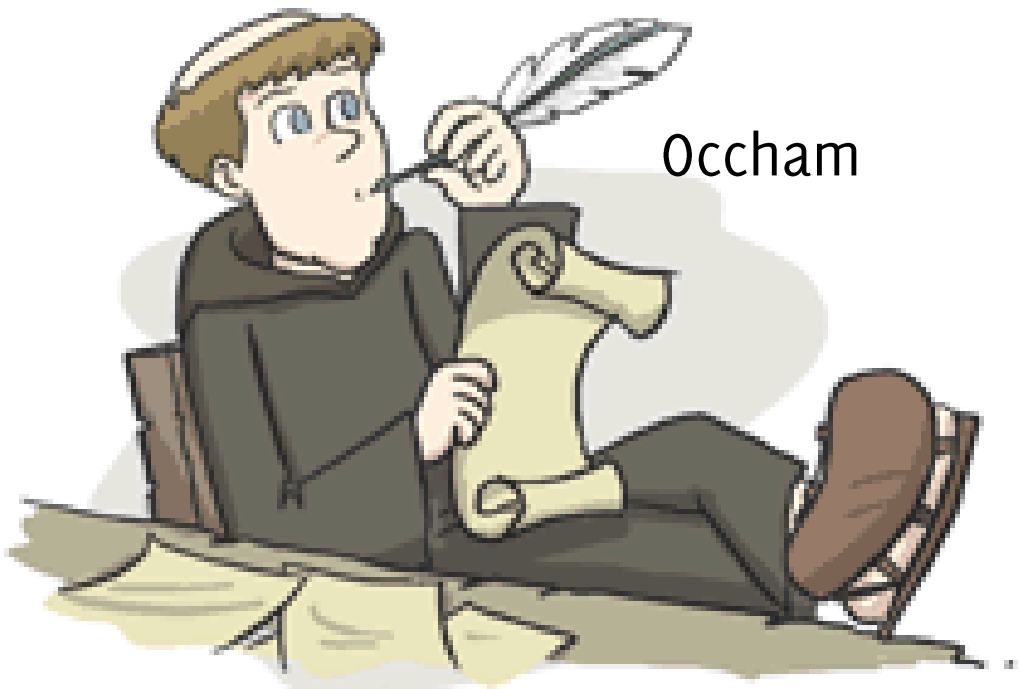


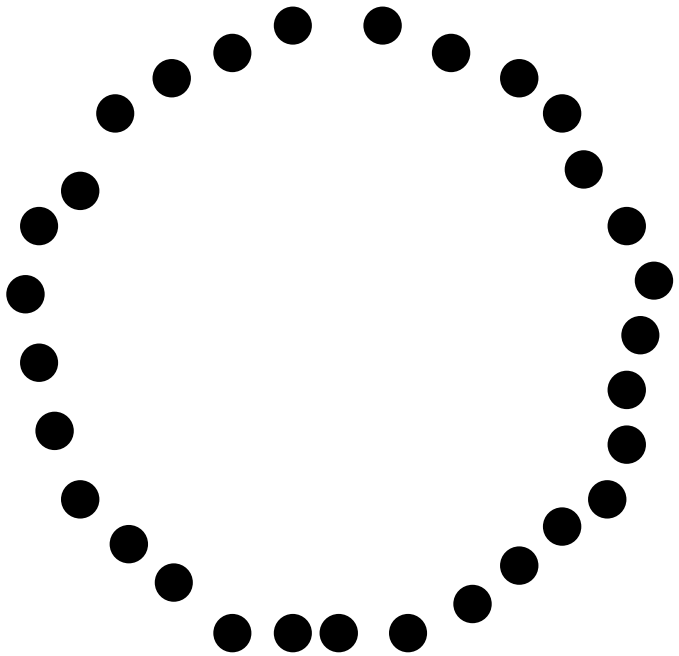
Multiple lines

Multi-class model fitting

If we want to describes data with multiple classes of models (*e.g.*, lines and circles), at some points we must face a model selection problem.

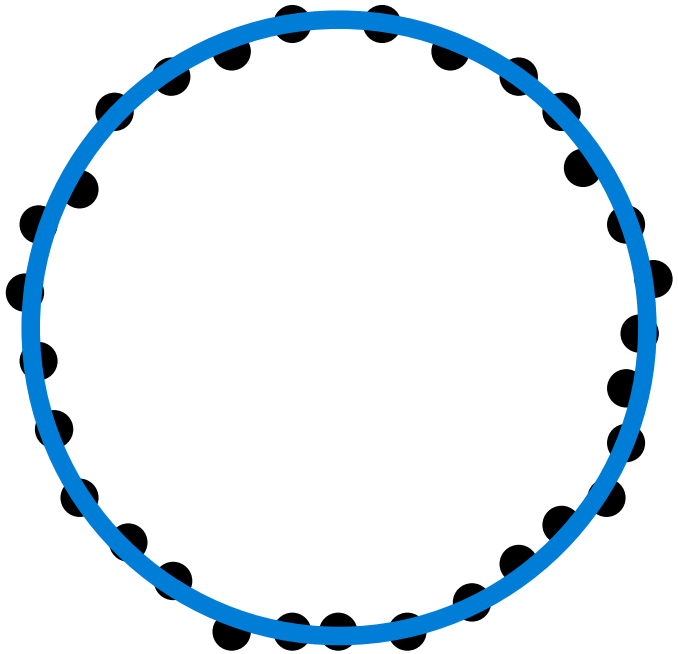
circle or a polygon?

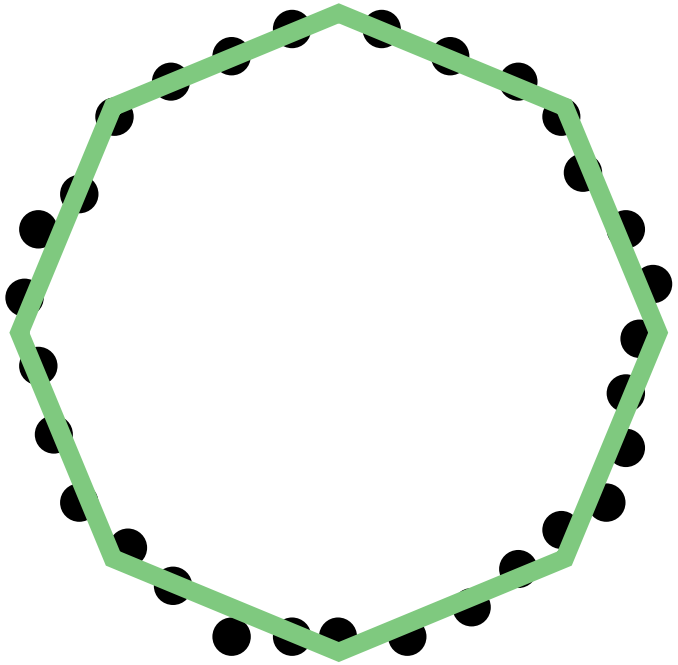




Multi-class model fitting

Different interpretations of the data are possible





1 complex instance

(cirlce/cylinder/fundamental matrix)

vs

many simpler instances

(lines/palnes/homographies)

Geometric Robust Information Criterion (GRIC)

GRIC cost: $-\underbrace{\frac{1}{2}\sum\left(\frac{r_i}{\sigma}\right)^2}_{\text{Geometric residuals}} + \underbrace{\lambda_1 dn + \lambda_2 k}_{\text{Model complexity}}$

AIC	$-2L + 2p$
BIC	$-2L + p\log(rn)$
MDL	$-2L + p/2\log_2(rn)$
GBIC	$-2L + \log(r)dn + \log(rn)k$
GRIC	$-2L + \lambda_1 dn + \lambda_2 k$

r_i residuals
σ noise estimate

d dim. of model manifold
n # points
k # model parameters

P. H.S. Torr. *Model Selection for Two View Geometry:A Review*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999

Geometric Robust Information Criterion (GRIC)

GRIC cost:

$$-\underbrace{\frac{1}{2}\sum\left(\frac{r_i}{\sigma}\right)^2}_{\$} + \underbrace{\lambda_1 dn + \lambda_2 k}_{€}$$

Geometric residual and model complexity are expressed in different "currency"

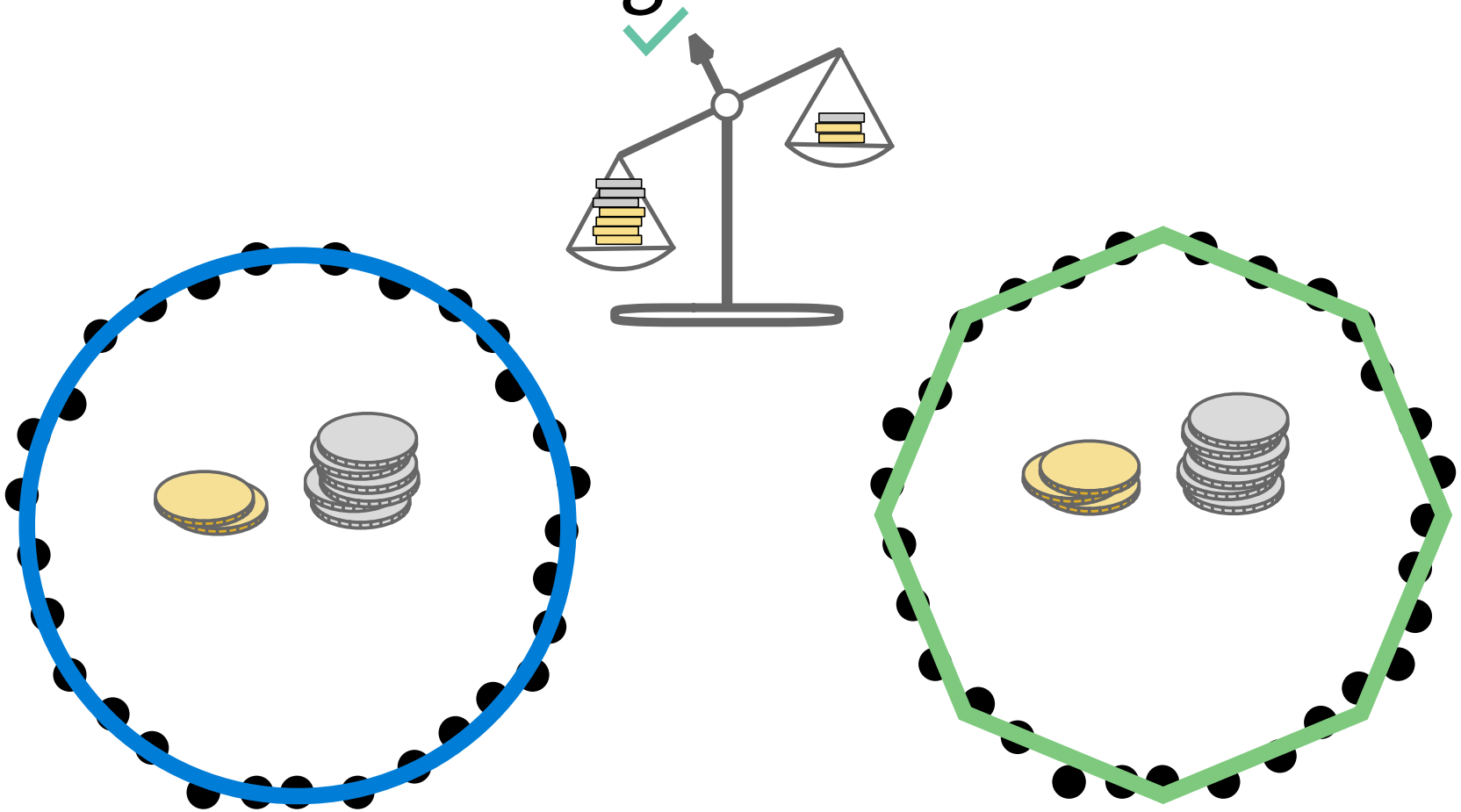
AIC	$-2L + 2p$
BIC	$-2L + p\log(rn)$
MDL	$-2L + p/2\log_2(rn)$
GBIC	$-2L + \log(r)dn + \log(rn)k$
GRIC	$-2L + \lambda_1 dn + \lambda_2 k$

r_i residuals
σ noise estimate

d dim. of model manifold
n # points
k # model parameters

P. H.S. Torr. *Model Selection for Two View Geometry:A Review*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999

Multi-class model fitting



1 complex instance vs many simpler instances

(cirlce/cylinder/fundamental matrix) (lines/palnes/homographies)

And if you want to investigate further in a reseach thesis...

Thesis on Multi model fitting: beyond analytical models

*Classical multi-model fitting methods leverage on **analytical models** with **very few** parameters.*
These can be easily integrated in robust fitting framework to gain resilience against outliers, but, due to their simplicitly, they are not able to encode higher level semantic information.

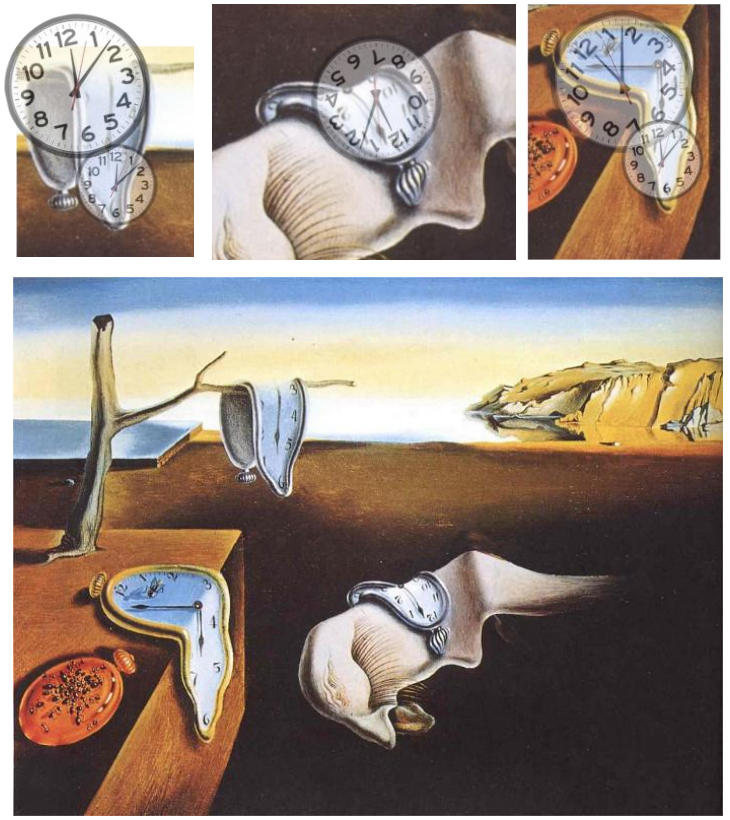
Research challenge:

- Go beyond analytical models: use more general models to foster the extraction of higher-level information from data.
- Non-analytical models can be:
  - Parameter-less (partition of data)
  - Neural models (exorbitant numbers of parameters)

- How non-analytical models can be fitted in a robust way (resamping)?

Applications:

- Ensamble clustering
- Instance Segmentation Networks



It is possible to use Preference Analysis to deal with non-rigid models ?

Luca Magri, Giacomo Boracchi

Thesis on Learning for 3D vision

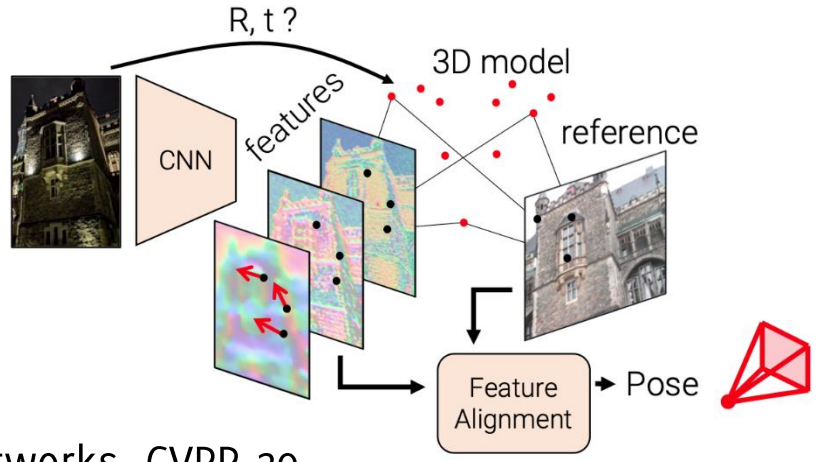
Combine the metrological accuracy of traditional model-fitting methods with the high-level understanding capabilities of deep learning models:.

Research challenges:

- 3D scene are complex, this make the training unstable
- Robustness across viewpoint changes and illumination condition is required

Aim:

- Conceive new algorithm able to exploit deep networks to learn robust and invariant visual features, and leverage on principled algorithm to geometric estimation.



- **Application:**
  3D reconstruction (Structure from Motion)
  Deep trifocal tensor estimation

References to start with:

[1] Sarlin et Al, SuperGlue: Learning Feature Matching with Graph Neural Networks. CVPR 20
[2] Sarlin et Al, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. CVPR 21
[3] Lindenberger et Al, Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. ICCV 21
[4] Zhang et Al. Content Aware Unsupervised Deep Homography Estimation

L. Magri, A. M. Rizzzo

Thesis on Neural Sampling for 3D data

Indoor modelling aims at recovering structural elements such as rooms, doors, and walls, then reconstructs a structured 3D model consisting of such elements. Instance Segmentation architectures can provide higher-level information to enable semantic-aware multi-model fitting

Research goals:

- Investigate how Instance Segmentation Networks can be integrated to steer the sampling towards promising model
- Enforce robustness through resampling scheme
- Test on 3D datasets for indoor classification

Materials and Methods:

- 3D datasets
- Access to a server mounting **GPUs will be provided**

http://www.scan-net.org

Some reference to start:

[1] Jiang, Li, et al. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In CVPR. 2020.
[2] Yang, Bo, et al. Learning object bounding boxes for 3d instance segmentation on point clouds. In NIPS. 2019.
[3] Yi, Li, et al. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In CVPR. 2019.

L. Magri, A. M. Rizzzo

Or in a research internship…

drop me an email for further information

giacomo.boracchi@polimi.it

Collaboration with Antares Vision Group

Antares Vision is currently sponsoring a PhD Scholarship under our supervision

We are planning to expand collaboration with an MSc internship in their research labs

Topics for a research thesis covers:

- Computer-Vision for industrial applications
- Deep Learning models for industrial applications

And in particular for visual inspection, recognition, tracking, anomaly detection,…

Collaboration with Antares Vision Group

Antares Vision Group

- 900+ employies worldwide
- 26% of workforce devoted to R&D activities
- Headquarter in Travagliato (Brescia)
- Includes Orobix, an AI service company

Collaboration with Antares Vision Group

ANTARES VISION GROUP

INSPECTION

Camera Based system
Visual Inspection for product and packaging
Smart Camera
Embedded Vision
SWIR camera
Hyperspectral NIR/VIS

Laser Absorbtion Spectroscopy
Pressure/Vacuum measurement
Head Space Gas Analysis
Leak detection [CCIT]
Laboratory Instruments

High Voltage
Leak detection [CCIT]
Laboratory Instruments

Sensor Based
Vacuum/Pressure Decay
Leak Detection
X-Ray and HF
CheckWeigh
Metal Detector
Event Based

TRACEABILITY

Serialization
Aggregation
Mobile Traceability
Real-time IOT

AUTHENTICATION

Visible and Invisible Tag
Serial Number Management
Visual Inspection Systems

DATA MANAGEMENT

Business Intelligence
Data Analysis
Advanced Analytics
Artificial Intelligence
Blockchain
Big Data
Cloud Computing
Integrated Platforms

PEOPLE
PROFIT
PRODUCT
PLANET

ONE ECOSYSTEM OF **TECHNOLOGIES**

Collaboration with Antares Vision Group

ORÒBIX
WE MAKE AI HAPPEN

AI solutions

Ready-to-use AI solutions with a shorter and consolidated time-to value.
Flexible, but focused on solving well-identified problems.

detectiv.ai
Real-time AI solution for anomaly detection

AI-go
Vision inspection platform: classification, segmentation, OCR

AI engineering

Tailor-made AI models, solving a wider variety of problems.
Leverage a team of experts in cross-industry AI applications. Wider range of applications (e.g. Vision inspection and Deep learning, Time series, Bayesian models, Reinforcement learning).

AI managed services*

We take care of all the AI of a company ensuring trust, compliance and reliability once in production.

invariant.ai®
Platform for the deployment, governance, monitoring and lifecycle management of AI systems in critical processes.

invariant.ai® EDGE

invariant.ai® CLOUD

*Oròbix named in the **2021 Gartner "Market Guide for Artificial Intelligence Service Providers"**