RANdom SAmple Consensus (RANSAC)

Pablo Mesejo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

RANSAC

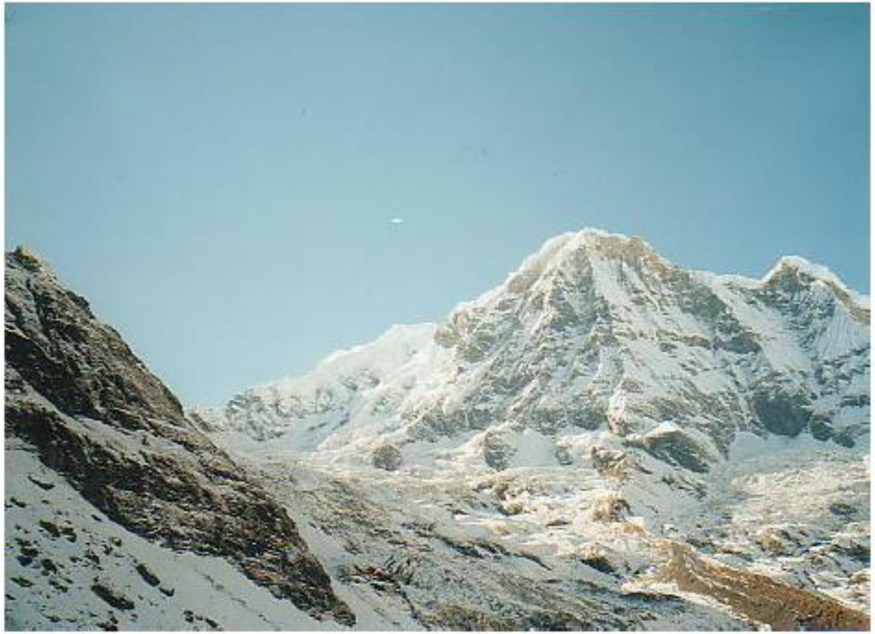
- Many of the slides and explanations in this seminar come from
  - Cyrill Stachniss (University of Bonn):
    - https://www.ipb.uni-bonn.de/html/teaching/msr2-2020/sse2-11-ransac.pdf
    - https://www.youtube.com/watch?v=oT9c_LlFBqs
  - Silvio Savarese (University of Stanford):
    - https://cvgl.stanford.edu/teaching/cs231a_winter1415/lecture/lecture9_fitting_matching.pdf
- Other interesting references:
  - Daniel Huttenlocher (University of Cornell): http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/cs664-20-robust-fitting.pdf
  - Robert Collins (The Pennsylvania State University): http://www.cse.psu.edu/~rtc12/CSE486/lecture15.pdf

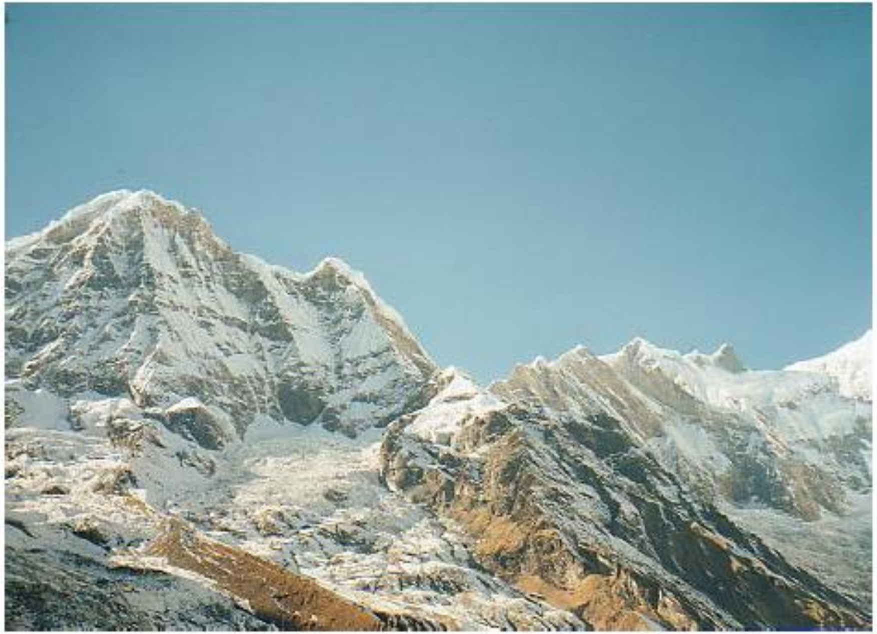
RANSAC

- Somewhat related to Hough Transform
  - In Forsyth and Ponce (2012), both are explained within the same chapter: 10. Grouping and Model Fitting
    - 10.1 The Hough Transform
    - 10.4 Robustness
      - 10.4.2 RANSAC: Searching for Good Points

How do we build a panorama?

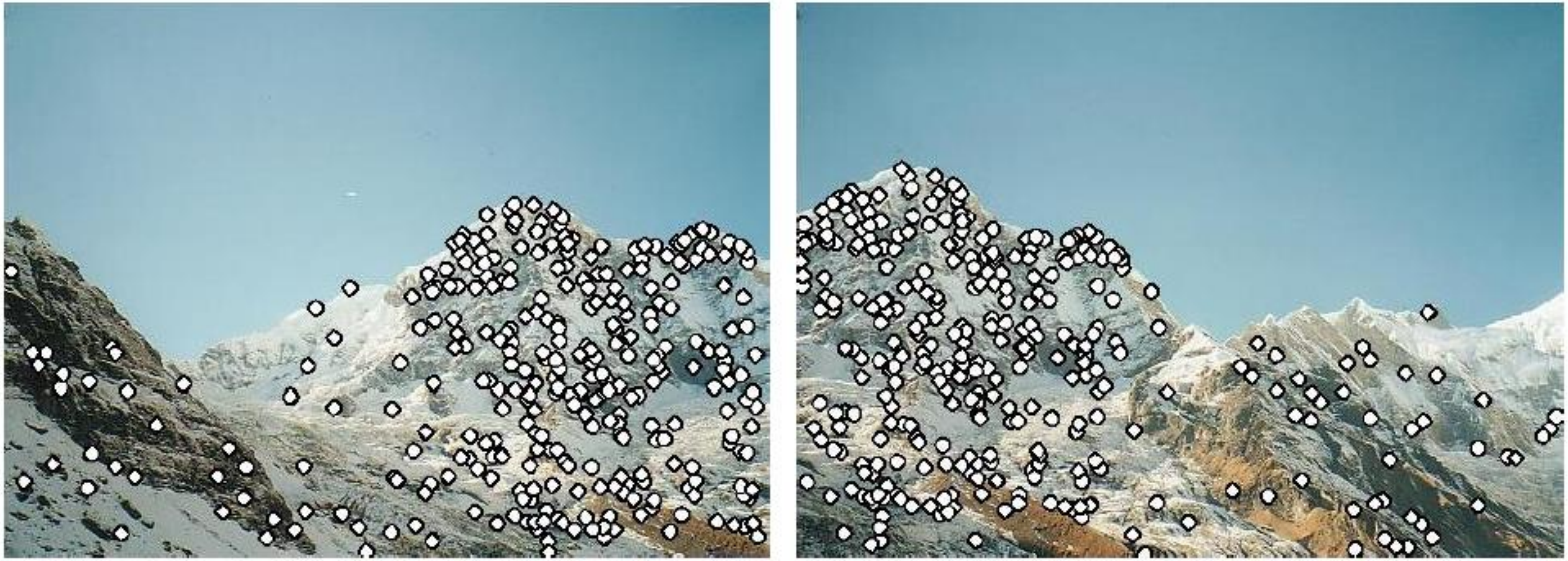
- We need to align images





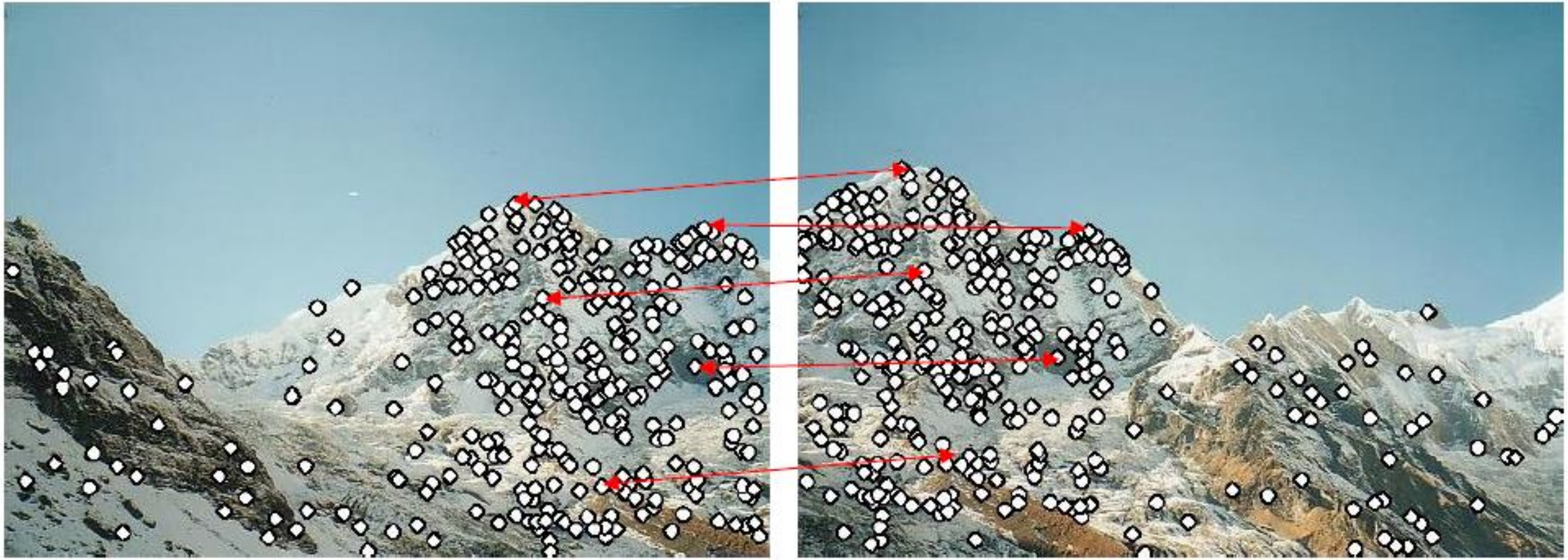
How do we build a panorama?

- We detect keypoints in both images



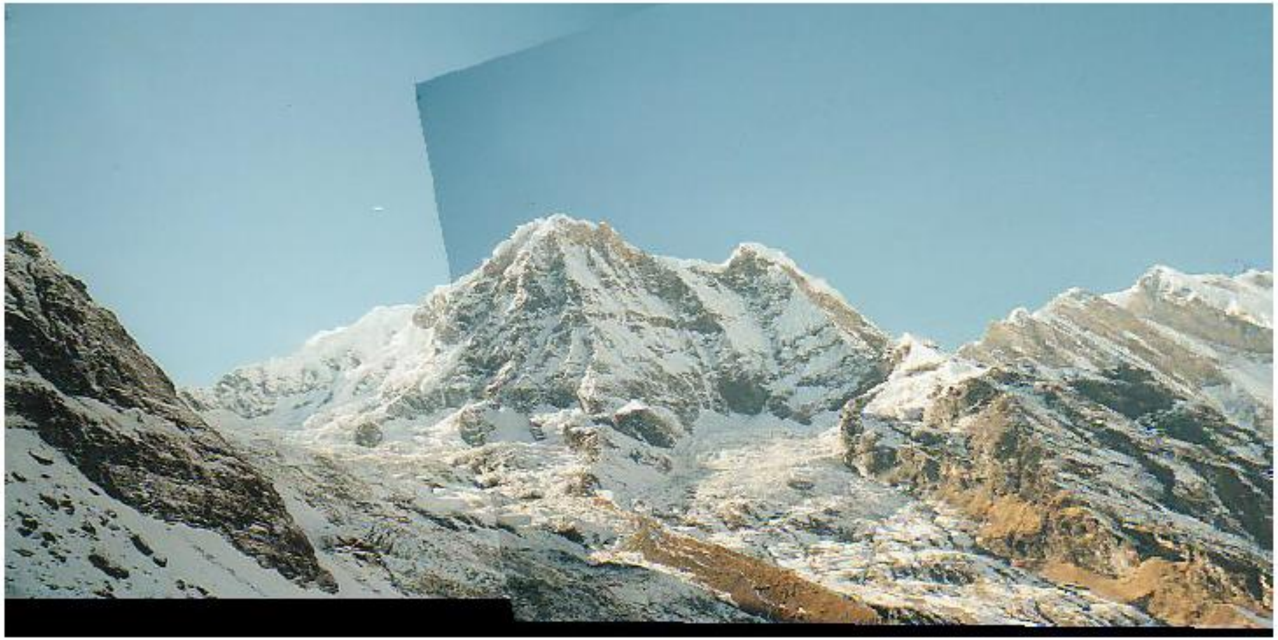
How do we build a panorama?

- We find corresponding pairs



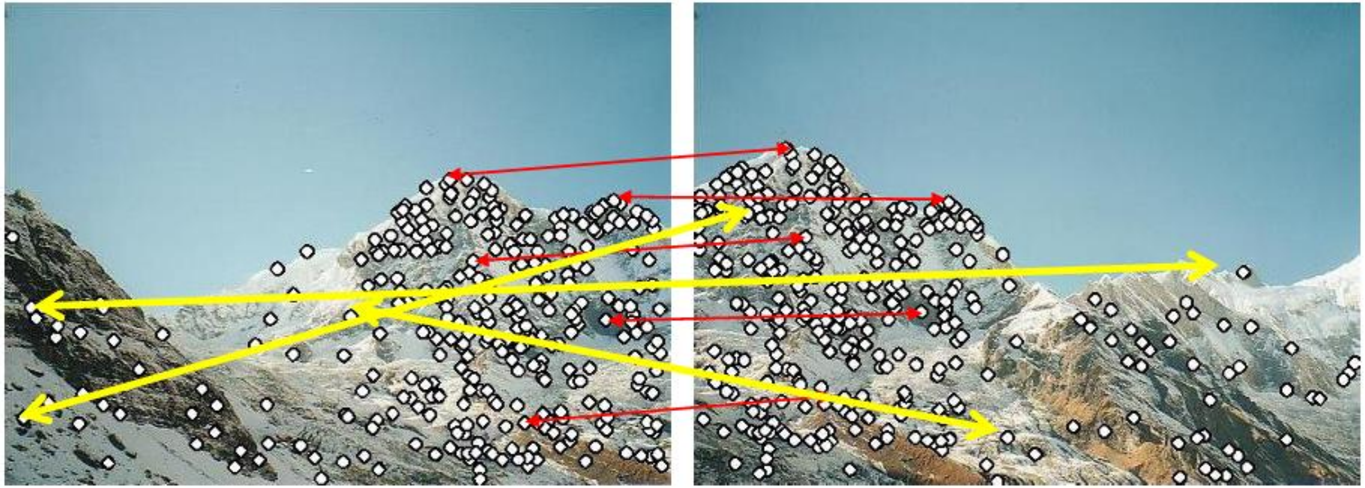
How do we build a panorama?

- We use those pairs to align the images

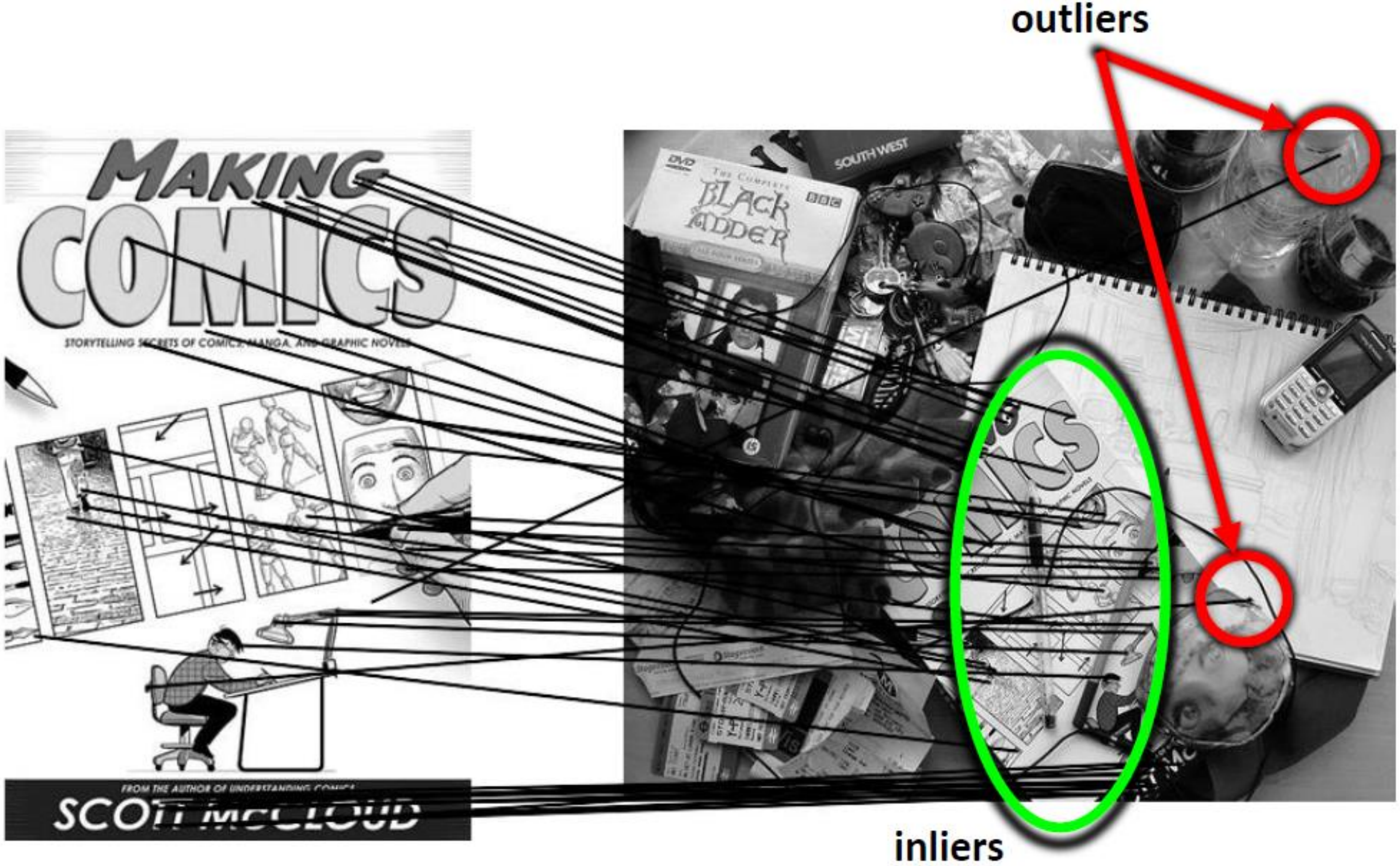


Remember Problem 3

- Need to estimate transformation between images, despite erroneous correspondences



Inliers vs Outliers


outliers
MAKING
COMICS
STORYTELLING SECRETS OF COMICS, MANGA, AND GRAPHIC NOVELS
FROM THE AUTHOR OF UNDERSTANDING COMICS
SCOTT McCLOUD
DVD
THE COMPLETE
BLACK ADDER
SOUTH WEST
inliers

How do we know if the correspondences are correct?

RANdom SAmple Consensus (**RANSAC**) (Fischler & Bolles, 1981)

- Trial-and-error method
- **Key idea:** find the best partition of keypoints in the set of *inliers* and *outliers*, and estimate the model from the *inliers*
- Standard approach to deal with *outliers* (robust statistics).

Fischler, M. A., and R. C. Bolles. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Communications of the Association for Computing Machinery 24 (1981): 381-395.

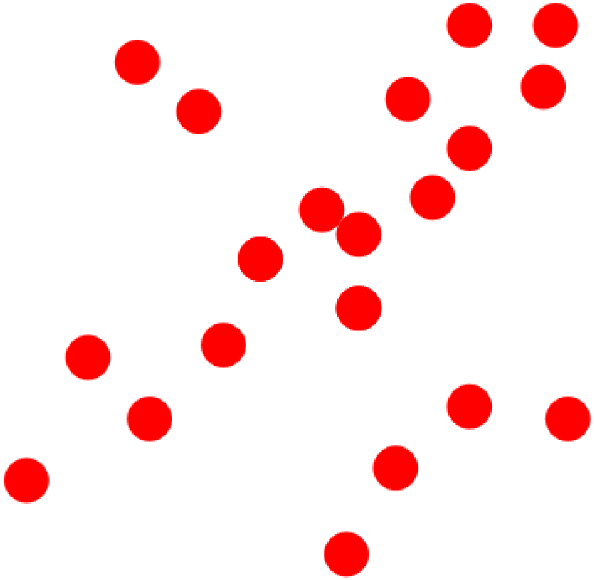
RANSAC

1. **Sample** the number of points our model requires.
2. **Calculate the parameters** of my model using the sampled points.
3. **Calculate the support** our model has based on the number of *inliers* that support it.

Repeat 1-3 until we have found the best model with high confidence.

RANSAC

- Simple example with linear regression. We want to fit a line to these points



Our hypothesis is a line.

How many points de we need to fit a line?

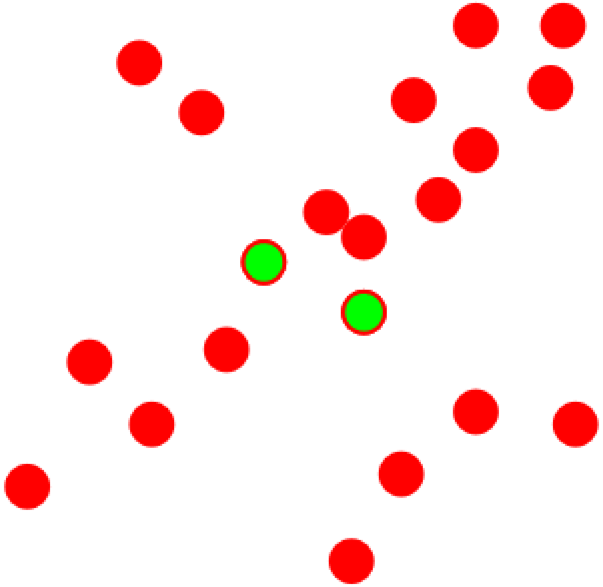
2

In the case of homography, how many matching points do we need?

4 pairs of matching points

RANSAC

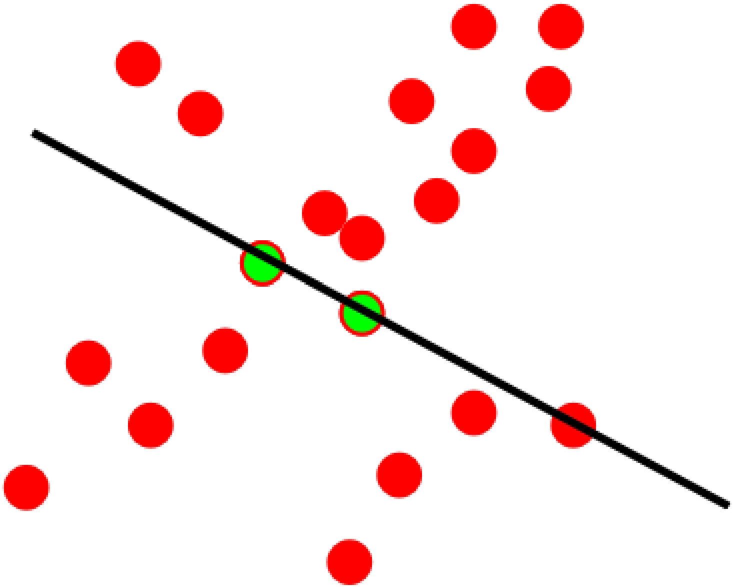
- Simple example with linear regression. We want to fit a line to these points



We select the number of necessary points to fit our model (in this case, 2 points)

RANSAC

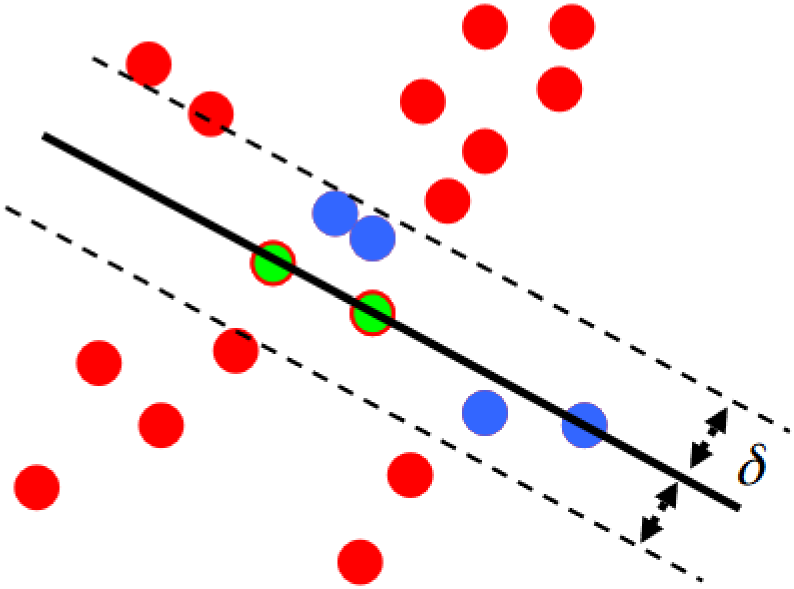
- Simple example with linear regression. We want to fit a line to these points



We calculate the line that passes through these two points.

RANSAC

- Simple example with linear regression. We want to fit a line to these points

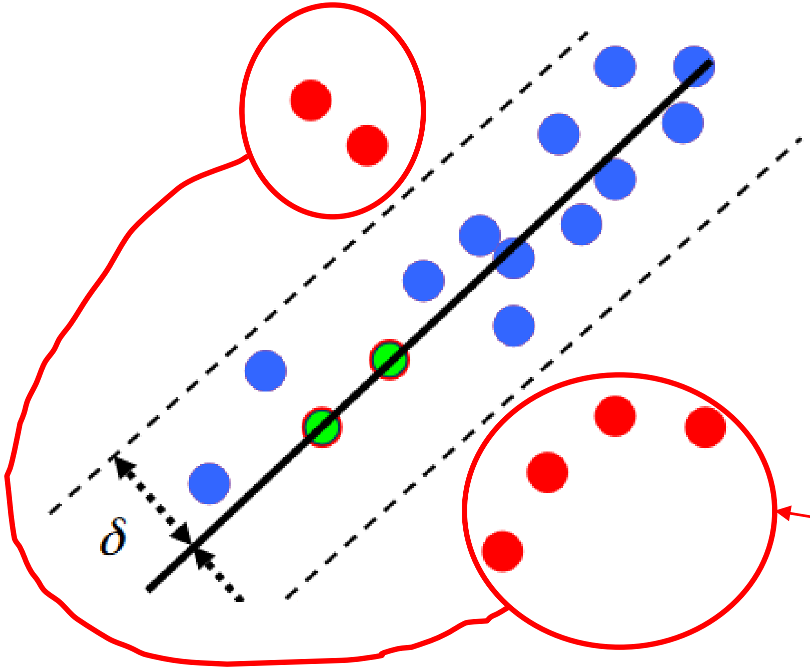


We calculate the number of inliers with respect to our model using some predefined threshold.

#inliers: 4

RANSAC

- Simple example with linear regression. We want to fit a line to these points



We try different subsets of two points until we find the “best” model.

#inliers: 12

These points are considered *outliers*, and are not taken into account for the computation of the straight line → **robust method**!

RANSAC

- Ok, very easy, but…
  - How many sampling repetitions should I carry out?

- Number of sampled points **s** (minumum number of points needed to fit our model)
  - Determined by the model we want to use
- Percentage of outliers **e** (e=#outliers/#datapoints)
  - We do not need the exact number (that we commonly do not know), an approximation is enough.
- Number of rounds (rounds/trials) **T**
  - We must choose **T**, such that, with probability **p**, at least one random set is free of *outliers*

RANSAC

- Number of sampled points **s** (minimum number of points needed to fit our model)
- Percentage of outliers **e** (e=#outliers/#datapoints)
- Number of rounds (rounds/trials) **T**
  - We must choose **T**, such that, with probability **p**, at least one random set is free of *outliers*

Probability of being an *outlier*?

RANSAC

- Number of sampled points **s** (minimum number of points needed to fit our model)
- Percentage of outliers **e** (e=#outliers/#datapoints)
- Number of rounds (rounds/trials) **T**
    - We must choose **T**, such that, with probability **p**, at least one random set is free of *outliers*

Probability of being an *outlier*: **e**

Probability of being an *inlier*?

RANSAC

- Number of sampled points **s** (minimum number of points needed to fit our model)
- Percentage of outliers **e** (e=#outliers/#datapoints)
- Number of rounds (rounds/trials) **T**
  - We must choose **T**, such that, with probability **p**, at least one random set is free of *outliers*

Probability of being an *outlier*: **e**

Probability of being an *inlier* (s=1): **(1-e)**

Probability of extracting s>1 *inliers*?

RANSAC

- Number of sampled points **s** (minimum number of points needed to fit our model)
- Percentage of outliers **e** (e=#outliers/#datapoints)
- Number of rounds (rounds/trials) **T**
  - We must choose **T**, such that, with probability **p**, at least one random set is free of *outliers*

Probability of being an *outlier*: **e**

Probability of being an *inlier* (s=1): **(1-e)**

Probability of extracting s>1 *inliers*: **$(1-e)^s$**

p: Probability of extracting s points and none of them being an *outlier*

Probability of s samples all being inliers

$$\prod_{i=0}^{s-1}\frac{I-i}{D-i}$$

D data points and *I* inliers

For *s<<D* approximate by $(I/D)^s$ or $(1-e)^s$

RANSAC

p: Probability of extracting s points and none being an *outlier*

Probability of being an *outlier*: **e**

Probability of being an *inlier* (s=1): **(1-e)**

Probability of extracting s>1 *inliers*, i.e., probability of success **p = (1-e)^s**

Probability of extracting s>1 points and that, at least, one of them is an *outlier*? That is, what is our probability of failure?

RANSAC

p: Probability of extracting s points and none being an *outlier*

Probability of being an *outlier*: **e**

Probability of being an *inlier* (s=1): **(1-e)**

Probability of extracting s>1 *inliers*, i.e., probability of success **p = (1-e)^s**

Probability of extracting s>1 points and that, at least, one of them is an *outlier*, i.e. our probability of failure **1-p = 1-(1-e)^s**

What is the probability of failing T times?

Probability of failing **once**. That is, do not select only *inliers*. At least one *outlier* is there.

RANSAC

p: Probability of extracting s points and none being an *outlier*

Probability of being an *outlier*: **e**

Probability of being an *inlier* (s=1): **(1-e)**

Probability of extracting s>1 *inliers*, i.e., probability of success **p = (1-e)^s**

Probability of extracting s>1 points and that, at least, one of them is an *outlier* , i.e. our probability of failure **1-p = 1-(1-e)^s**

Probability of failing **<u>once</u>**. That is, do not select only *inliers*.

Probability of failing T times: **1-p = (1-(1-e)^s)^T**

Probability of **<u>selecting at least one *outlier* in each of the T rounds</u>**.

RANSAC

e: Probability of being an *outlier*, i.e. estimated percentage of *outliers* in our dataset.

We want to solve for T!

How do we do it?

$$1-p = (1-(1-e)^{s})^{T}$$

p: Probability of extracting s points and none of them being an *outlier*. We select this value. For example, 99%

s is determined by our model (one line, s=2; one homography, s=4 points in correspondence)

RANSAC

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

$$\log(1-p) = T \cdot \log(1-(1-e)^s)$$

$$\boxed{T = \frac{log(1-p)}{log(1-(1-e)^s)}}$$

With this we can answer the question: how many repetitions of sampling should I carry out?

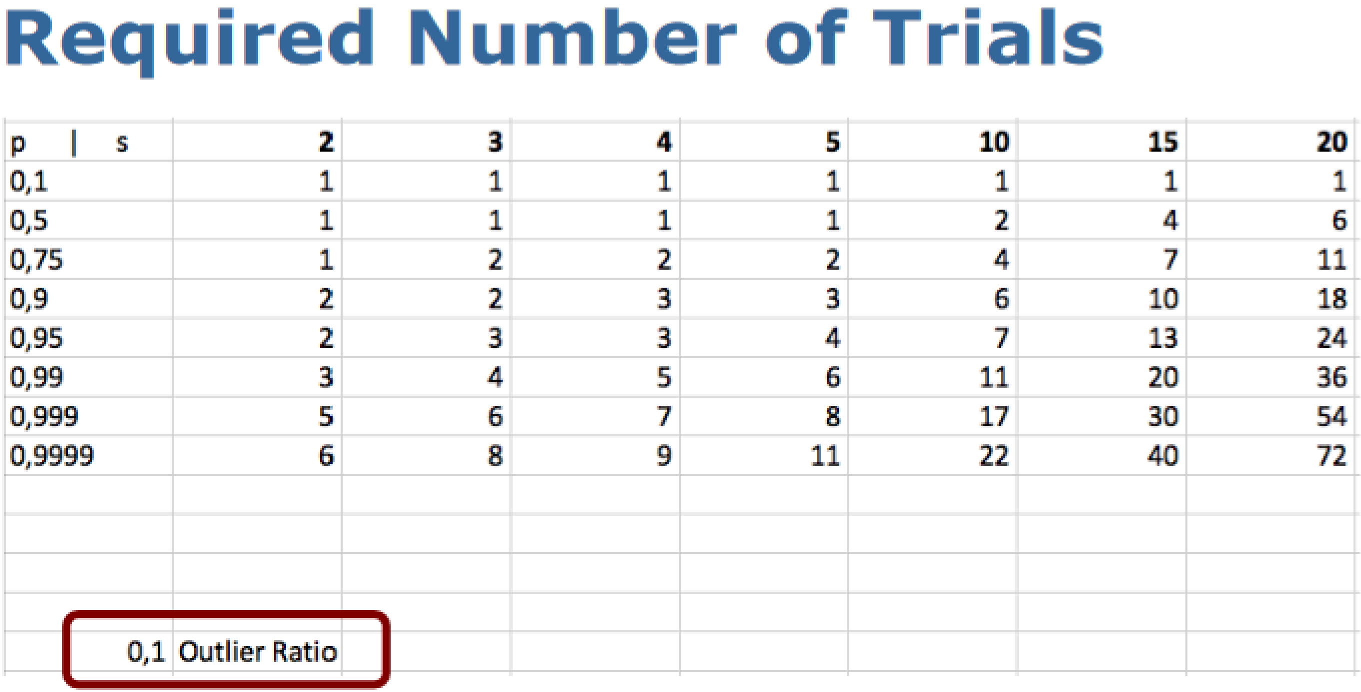
RANSAC

$$\boxed{T = \frac{log(1-p)}{log(1-(1-e)^s)}}$$

Critical element: **s** → If it grows a lot, i.e. **if our model needs a lot of points, I have to sample a lot**

RANSAC works well with simple models!

RANSAC



Required Number of Trials

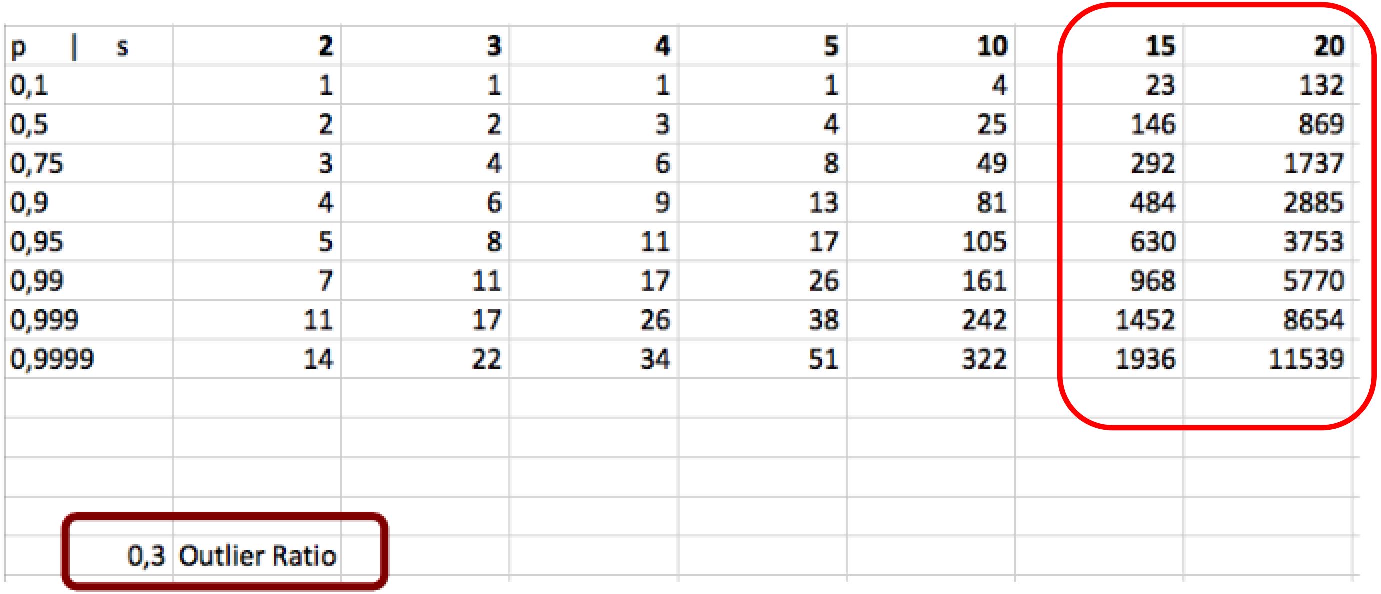
p | s	2	3	4	5	10	15	20
0,1	1	1	1	1	1	1	1
0,5	1	1	1	1	2	4	6
0,75	1	2	2	2	4	7	11
0,9	2	2	3	3	6	10	18
0,95	2	3	3	4	7	13	24
0,99	3	4	5	6	11	20	36
0,999	5	6	7	8	17	30	54
0,9999	6	8	9	11	22	40	72

0,1 Outlier Ratio

e

RANSAC

Required Number of Trials



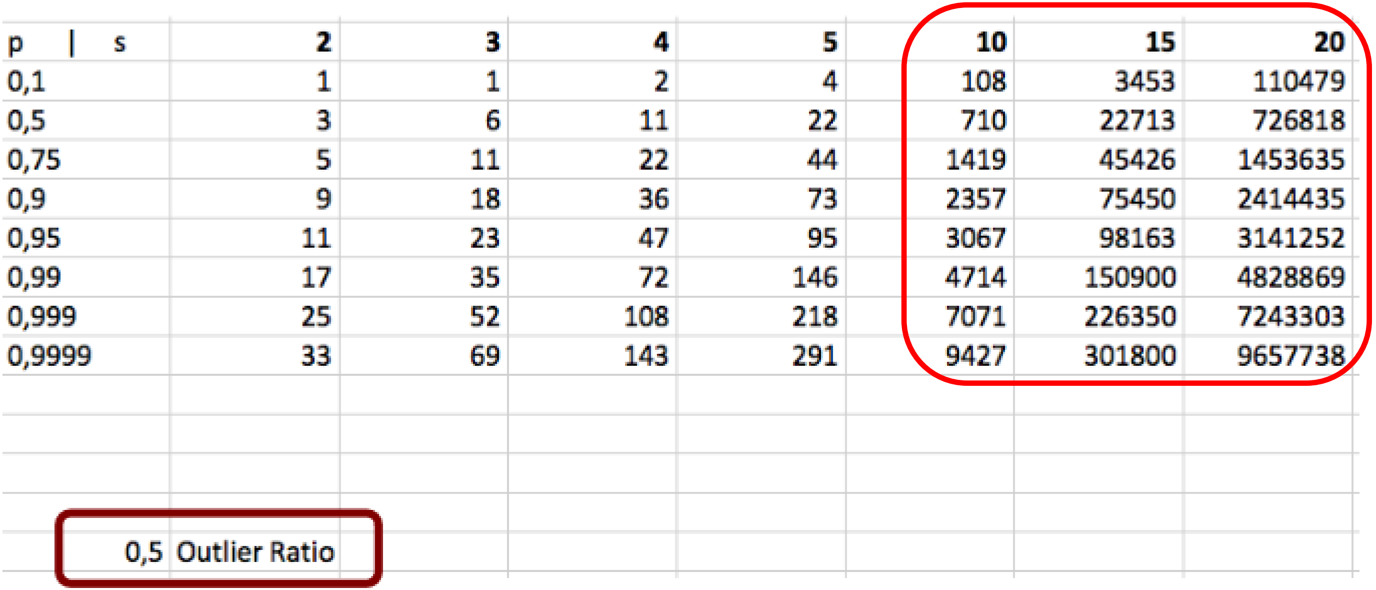
p | s	2	3	4	5	10	15	20
0,1	1	1	1	1	4	23	132
0,5	2	2	3	4	25	146	869
0,75	3	4	6	8	49	292	1737
0,9	4	6	9	13	81	484	2885
0,95	5	8	11	17	105	630	3753
0,99	7	11	17	26	161	968	5770
0,999	11	17	26	38	242	1452	8654
0,9999	14	22	34	51	322	1936	11539

0,3 Outlier Ratio

e

RANSAC

Required Number of Trials



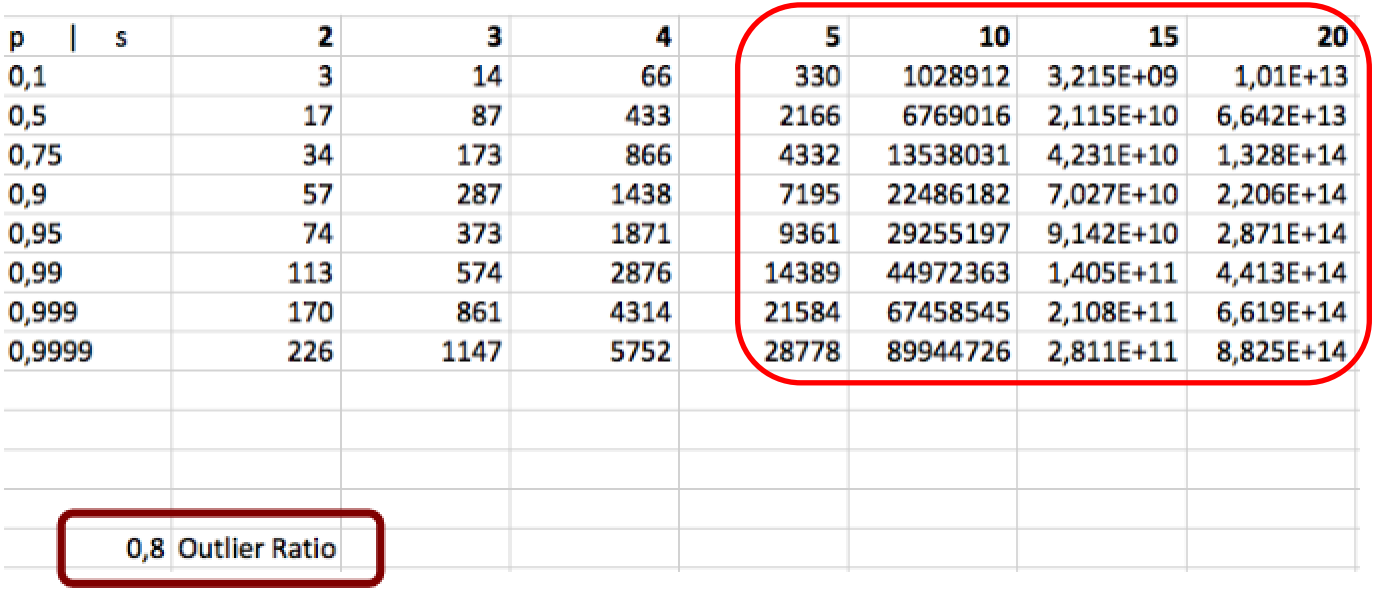
p | s	2	3	4	5	10	15	20
0,1	1	1	2	4	108	3453	110479
0,5	3	6	11	22	710	22713	726818
0,75	5	11	22	44	1419	45426	1453635
0,9	9	18	36	73	2357	75450	2414435
0,95	11	23	47	95	3067	98163	3141252
0,99	17	35	72	146	4714	150900	4828869
0,999	25	52	108	218	7071	226350	7243303
0,9999	33	69	143	291	9427	301800	9657738

0,5 Outlier Ratio

e

RANSAC

Required Number of Trials



p | s	2	3	4	5	10	15	20
0,1	3	14	66	330	1028912	3,215E+09	1,01E+13
0,5	17	87	433	2166	6769016	2,115E+10	6,642E+13
0,75	34	173	866	4332	13538031	4,231E+10	1,328E+14
0,9	57	287	1438	7195	22486182	7,027E+10	2,206E+14
0,95	74	373	1871	9361	29255197	9,142E+10	2,871E+14
0,99	113	574	2876	14389	44972363	1,405E+11	4,413E+14
0,999	170	861	4314	21584	67458545	2,108E+11	6,619E+14
0,9999	226	1147	5752	28778	89944726	2,811E+11	8,825E+14

0,8 Outlier Ratio

e

RANSAC and the calculation of homographies

- RANSAC loop:

  - Pick 4 points in correspondence randomly
  - Calculate H using DLT (*Direct Linear Transformation*)
  - Count *inliers*
- Keep largest set of *inliers*
- Re-compute least-squares H estimate using all inliers

RANSAC in OpenCV

- **`cv2.findHomography(srcPoints, dstPoints, cv2.RANSAC)`**

- https://docs.opencv.org/4.6.0/d9/d0c/group__calib3d.html#ga4abc2ece9fab9398f2e560d53c8c9780
- OpenCV default values
  - **p = 0.995**
  - **T = 2000**
  - «**RansacReprojThreshold** Maximum allowed reprojection error to treat a point pair as an inlier. That is, if

$$\|\mathtt{dstPoints}_i - \mathtt{convertPointsHomogeneous}(\mathtt{H} \cdot \mathtt{srcPoints}_i)\|_2 > \mathtt{ransacReprojThreshold}$$

then the point i is considered as an outlier.» **3 pixels**

Given a match **`m(srcPoint, dstPoint)`** and a homography **`H`**, if the distance between **`dstPoint`** and **`H*srcPoint`** (i.e., the homography applied to **`srcPoint`**) is greater than **`ransacReprojThreshold`**, then **`m`** is considered to be an *outlier*.

Conclusions

- RANSAC is
  - A very simple algorithm (easy to understand and implement)
  - Robust to *outliers*
  - Works well if your model needs up to 10 parameters
    - Otherwise the percentage of *outliers* should be low
  - Relatively sensitive to threshold selection
    - If it is too large, all hypotheses will be evaluated similarly
  - But, the computational time grows rapidly with the fraction of *outliers* (e) and the number of parameters needed to fit the model (s)!

RANdom SAmple Consensus (RANSAC)

Pablo Mesejo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial