Robust Fitting in Computer Vision (and elsewhere)

Maximum Consensus - MaxCon

Already seen in course...

The CVPR2023 prize winning paper was essentially on this topic..

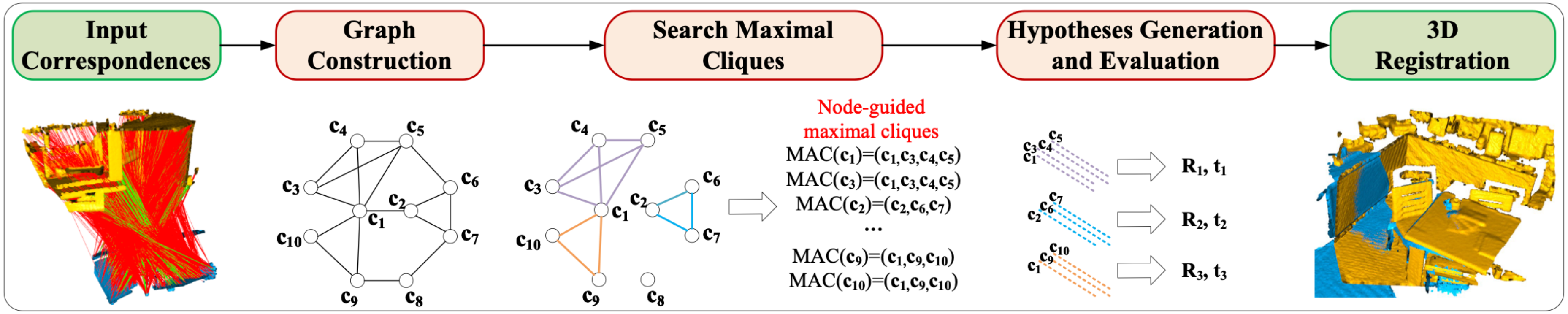


Figure 2. **Pipeline of MAC. 1.** Construct a graph for the initial correspondence set. **2.** Select a set of maximal cliques from the graph as the consistent sets. **3.** Generate and evaluate the hypotheses according to the consistent sets. **4.** Select the best hypothesis to perform 3D registration.

Robust Fitting - MaxCon

(Permission to use photo via Daniel Barath)

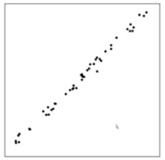
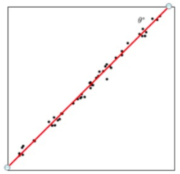
As long ago as the 1970’s,early 80’s , image Analysis and computer vision people realized that need robust fitting methods: Hough Transform and RANSAC were algorithms devised for this and are still used today…

manually aligned picture with scene. CVPR2023 best paper was automatic alignment of point clouds.
Computer vision/Image analysis/Robotics etc….need *automatic methods* for many tasks – alignment of images and pointclouds, extraction of lines/planes/cylinders from the data, 3-d reconstruction from multiple images etc. etc.

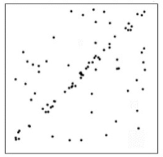
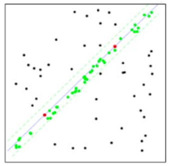
These tasks are *fitting a model/equation* do data *when there are outliers* - data points that don’t belong to your model.

Taxonomy of Geometric Estimation Problems

- Standard Single Class Single Instance Fitting Problem (SCSI)

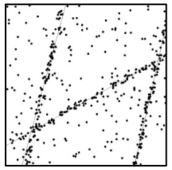
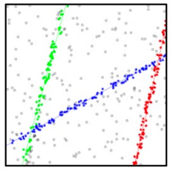
 ⇒ 

- **Robust Single Class Single Instance Fitting Problem (R-SCSI)**

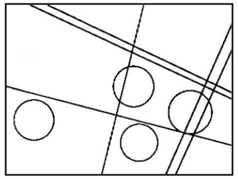
 ⇒ 

These are "traditional robust fitting" - statisticians rarely go beyond- and even here <50% outliers

- Single Class Multiple Instance Fitting Problem (SCMI)

 ⇒ 

- Multiple Class Multiple Instance Fitting Problem (MCMI)

 ⇒ 

We are never going to be concerned with this – can solve by least squares

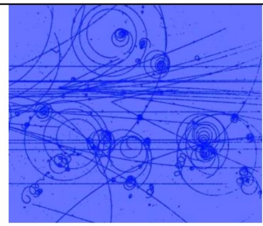
We are going to mostly talk about this problem – Least squares won't work because of outliers.

These problems are also a major motivation – EVEN IF we only want to extract one of the structures.

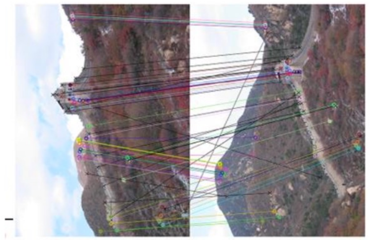
This becomes VERY hard – need model selection – a notoriously difficult topic ON TOP OF ROBUST fitting.

Single/Multi-Class S/M-Instance Fitting Applications

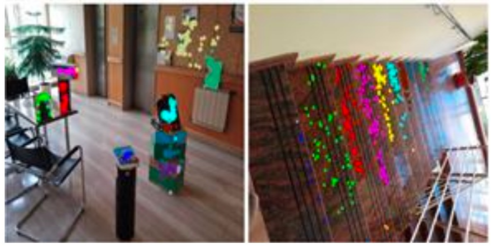
- detection of geometric primitives



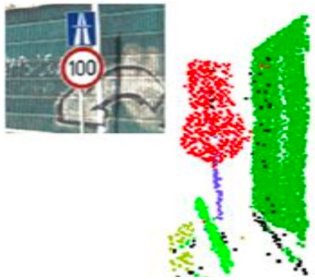
- epipolar geometry estimation
- detection of planar surfaces



- multiple motion segmentation

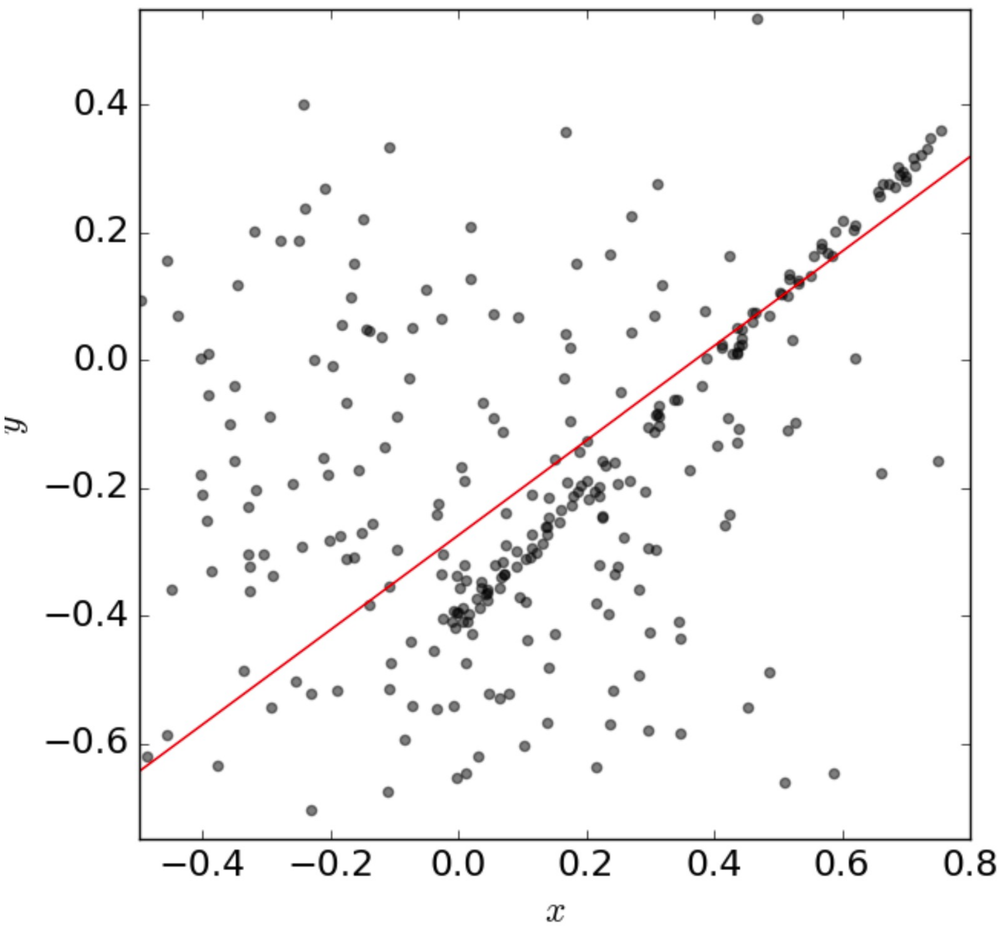


- Interpretation of lidar scans



Line Fitting with Outliers

Example 1



Least squares fit

Old studied problem…..without invoking

MACHINE ANALYSIS OF BUBBLE CHAMBER PICTURES

P. V. C. Hough

The University of Michigan, Ann Arbor, Mich.

Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959(!!!!) – though Duda and Hart generalised and popularised for the PR community in early 70's….

RANDOM SAMPLE CONSENSUS: A PARADIGM FOR MODEL FITTING WITH APPLICATIONS TO IMAGE ANALYSIS AND AUTOMATED CARTOGRAPHY

Technical Note 213

March 1980

By: Martin A. Fischler, Senior Computer Scientist
Robert C. Bolles, Computer Scientist

Artificial Intelligence Center
SRI International
Menlo Park, California 94025

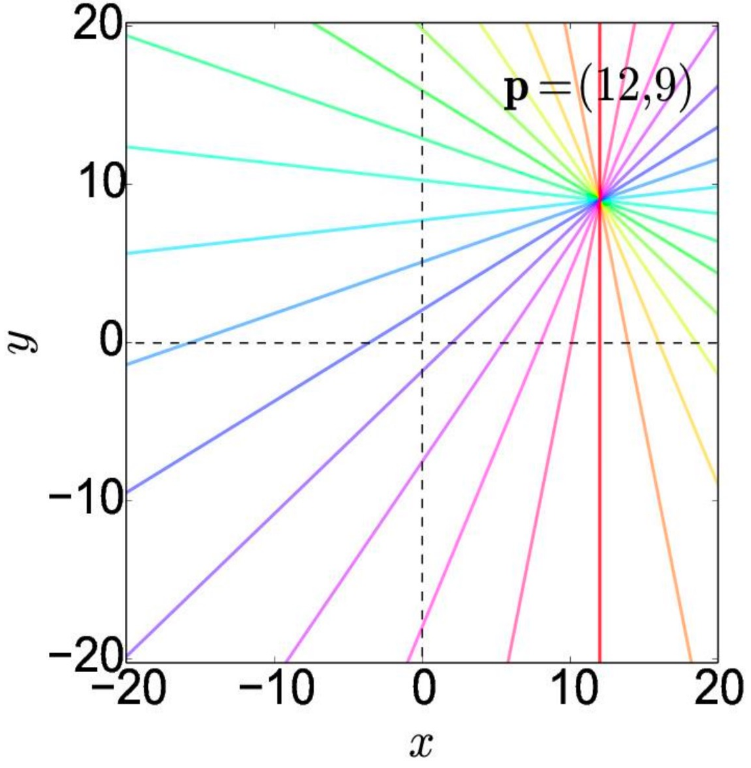
RANSAC - A contemporary of LMedS algorithm in Stats community

Hough works directly in "parameter space" - and is more or less inherently multi-structural, RANSAC works in "data subset space" and is more or less single structural in nature.

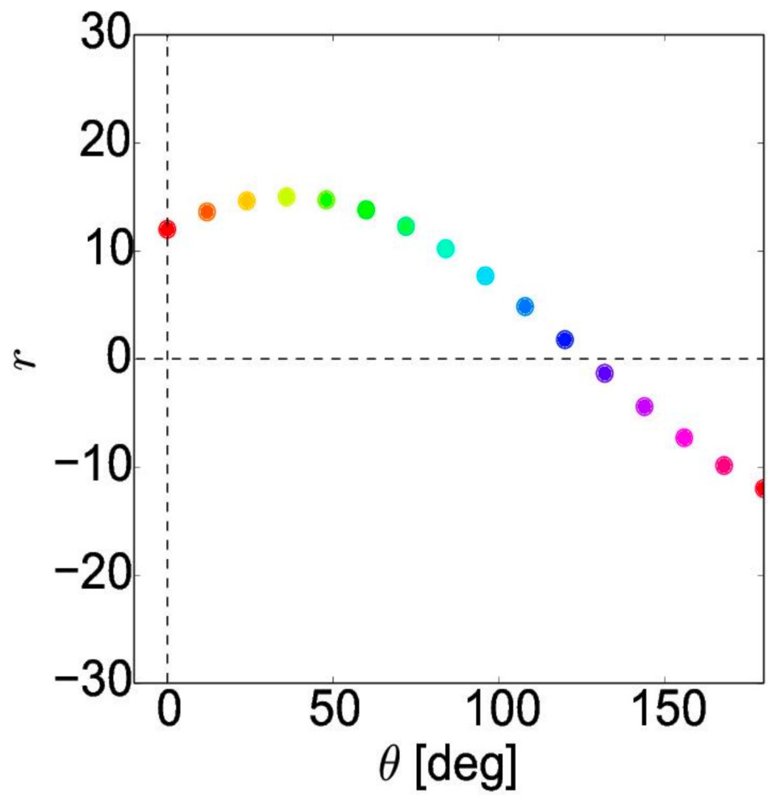
Technically they don't solve the same problem – they solve related problems – due to not optimizing the same criteria of model fit…but……….

Hough Transform Intro, Finding a Line

A point **p** votes for all lines it can be incident with.



A subset of lines incident to **p**



Corresponding line parameters

There are an infinite number of lines through a single point. Can parameterize these lines by theta and r (r perpendicular distance from origin to the line).

The infinite # lines trace a sinusoid curve in r,theta space

These curves intersect in r,theta corresponding to COMMON line between points.

Quantize r,theta space into voting "bins" – helps deal with small noise: but more importantly makes it simple to implement

Hough Transform, Finding a Line

A point **p** votes for all lines it can be incident with.

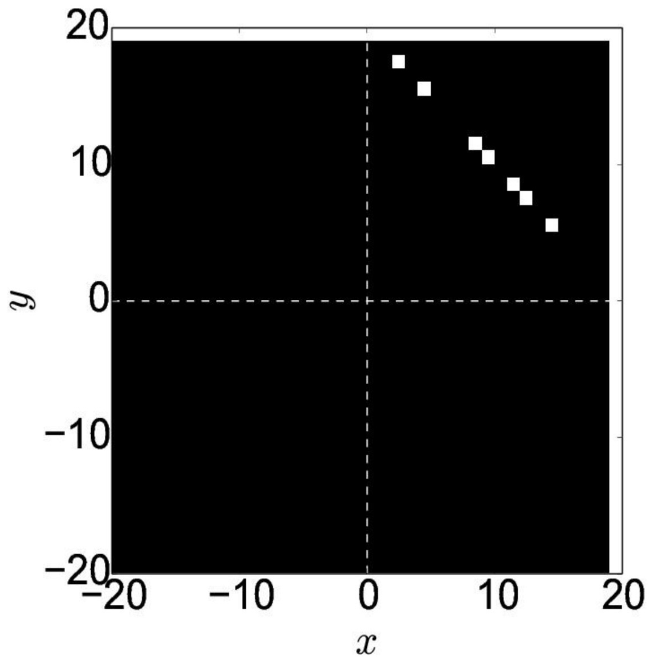
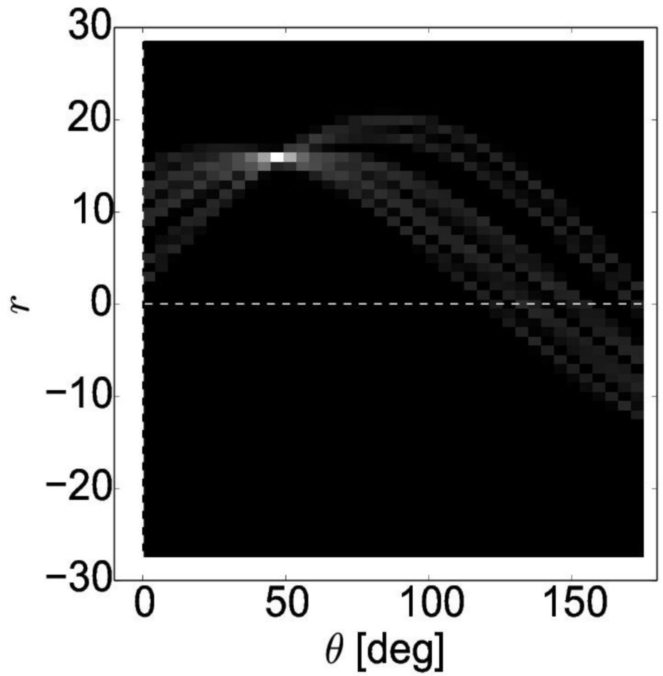


Image with a multiple points



Accumulated votes

Ransac – most popular algorithm in computer vision for robust fitting

Set number of iterations (usually on some “back of the envelope calulations of how many iterations to get at least one clean *minimal subset* for the model (e.g., 2 points for a line) with some number of anticipated possible outliers…)

Loop

randomly pick minimal subset

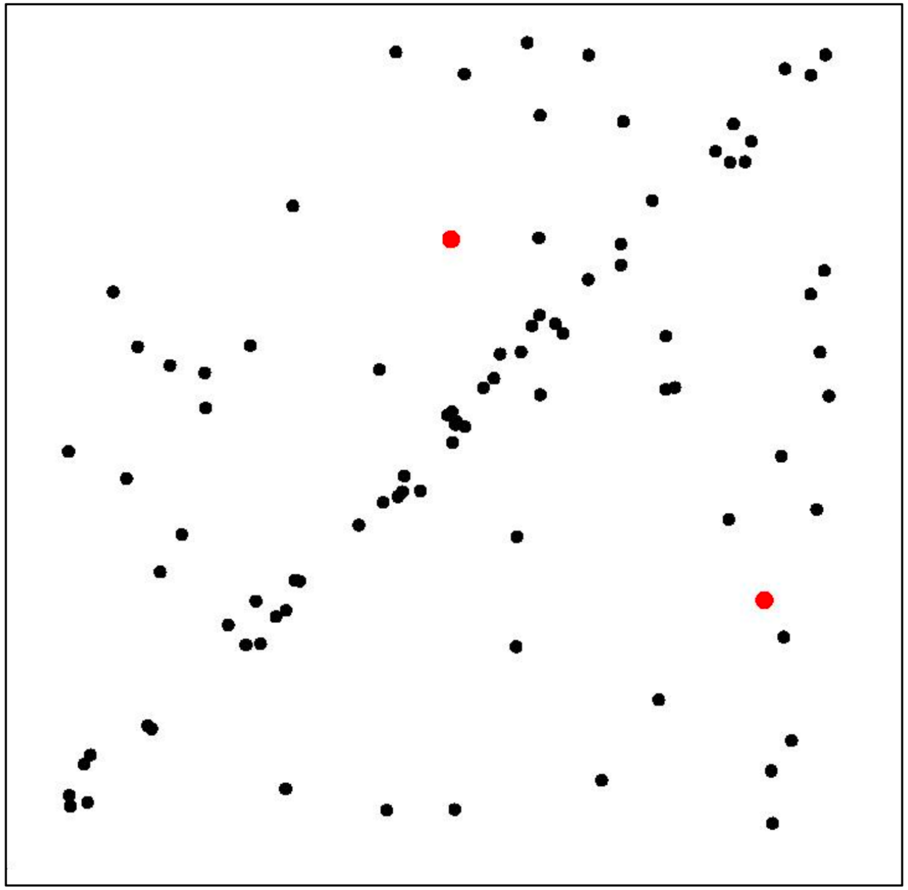
”fit” model to subset

count how many data points are within tolerance of model fit

remember the model with the largest number of data points within tolerance

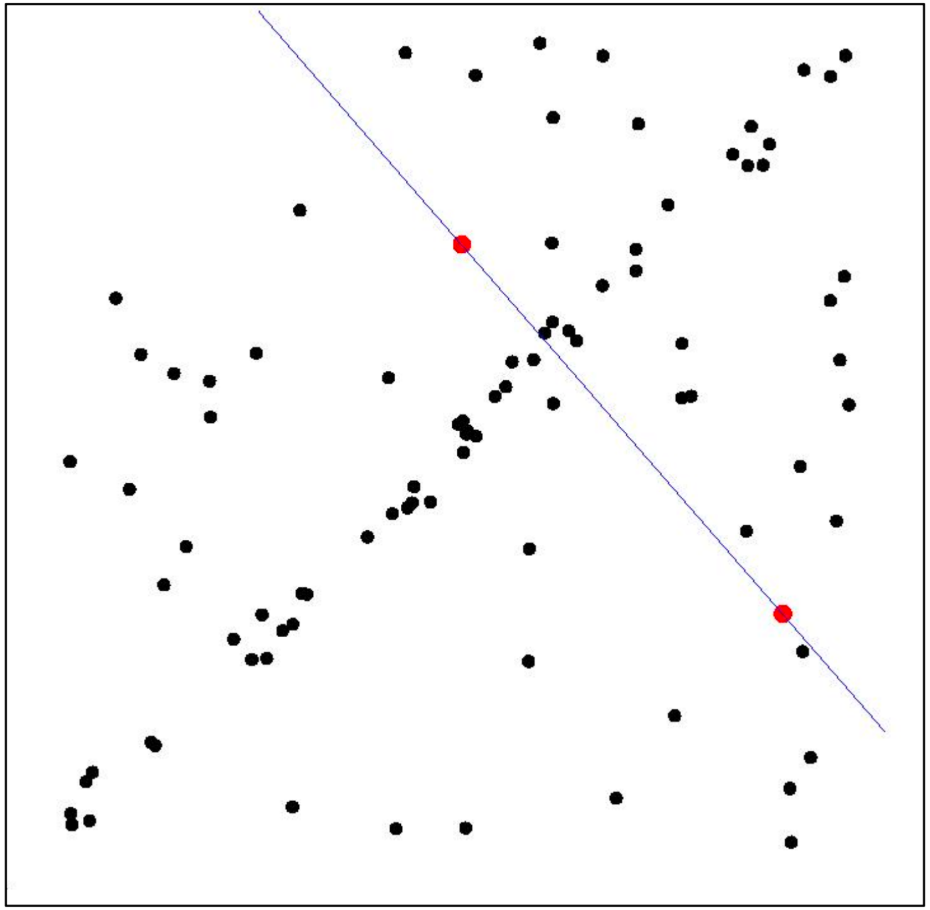
EndLoop

Random Sample Consensus - RANSAC



Select sample of m points at random

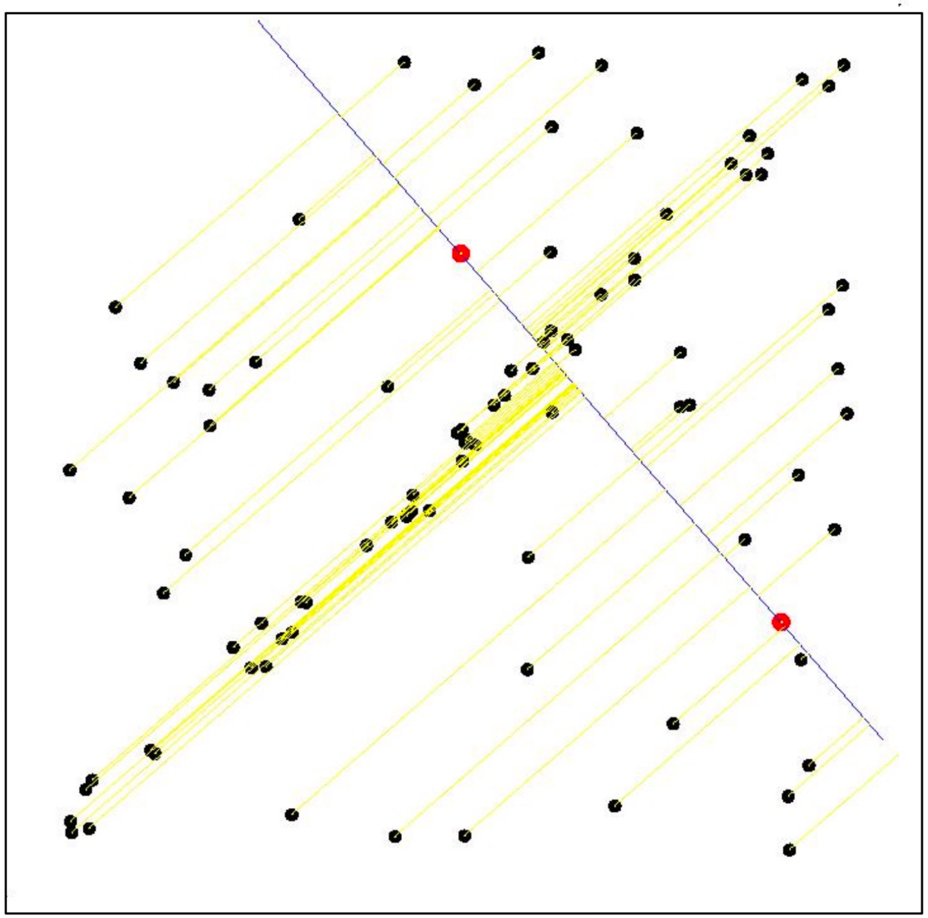
RANSAC



Select sample of m points at random

Estimate model parameters from the data in the sample

RANSAC

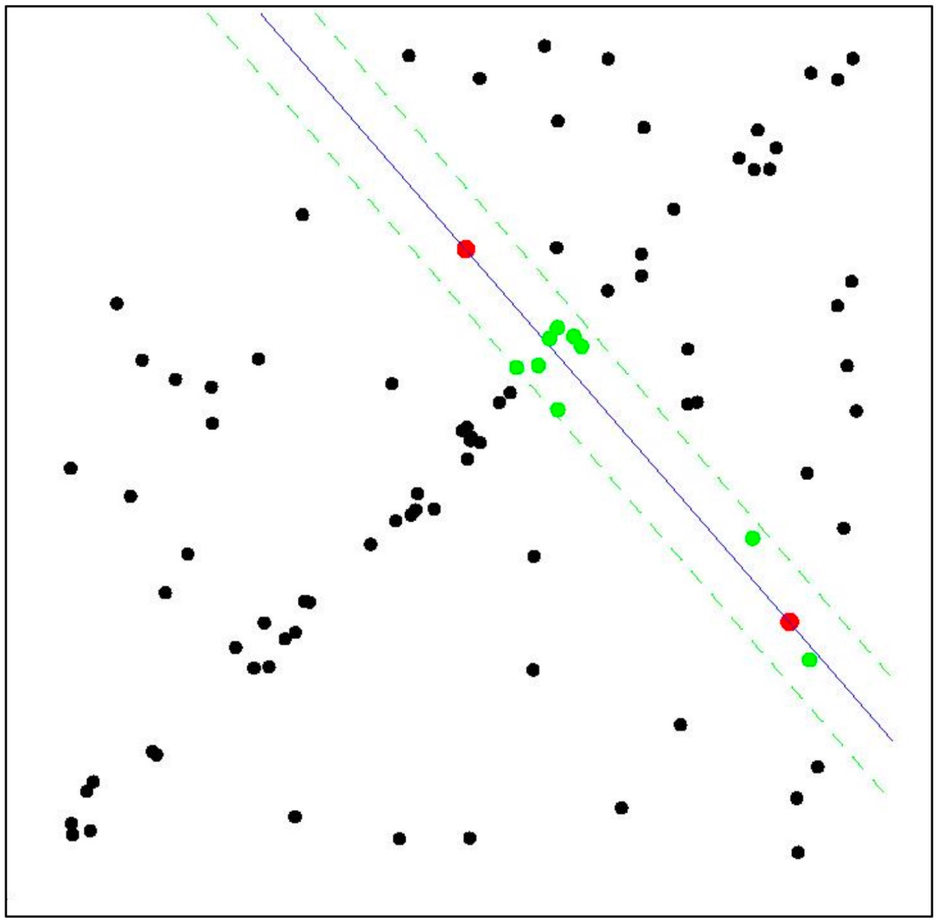


Select sample of m points at random

Estimate model parameters from the data in the sample

Evaluate the error (residual) for each data point

RANSAC



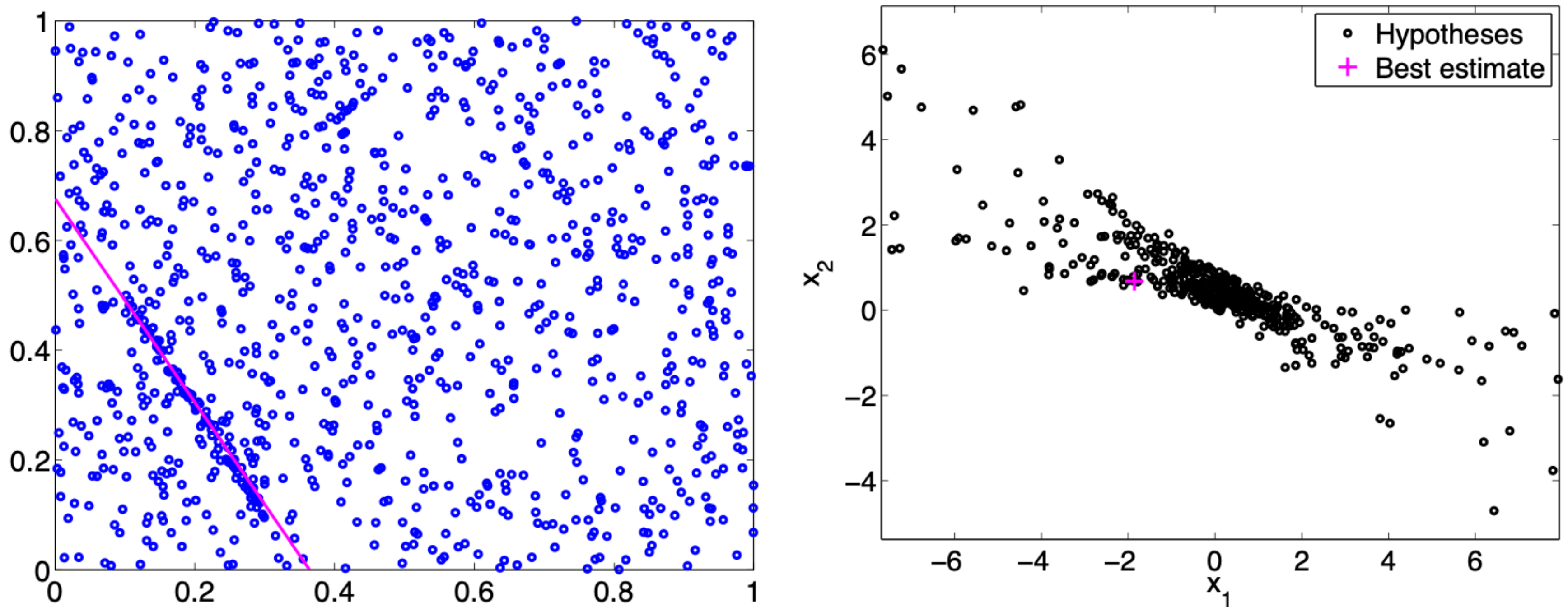
Select sample of m points at random

Estimate model parameters from the data in the sample

Evaluate the error (residual) for each data point

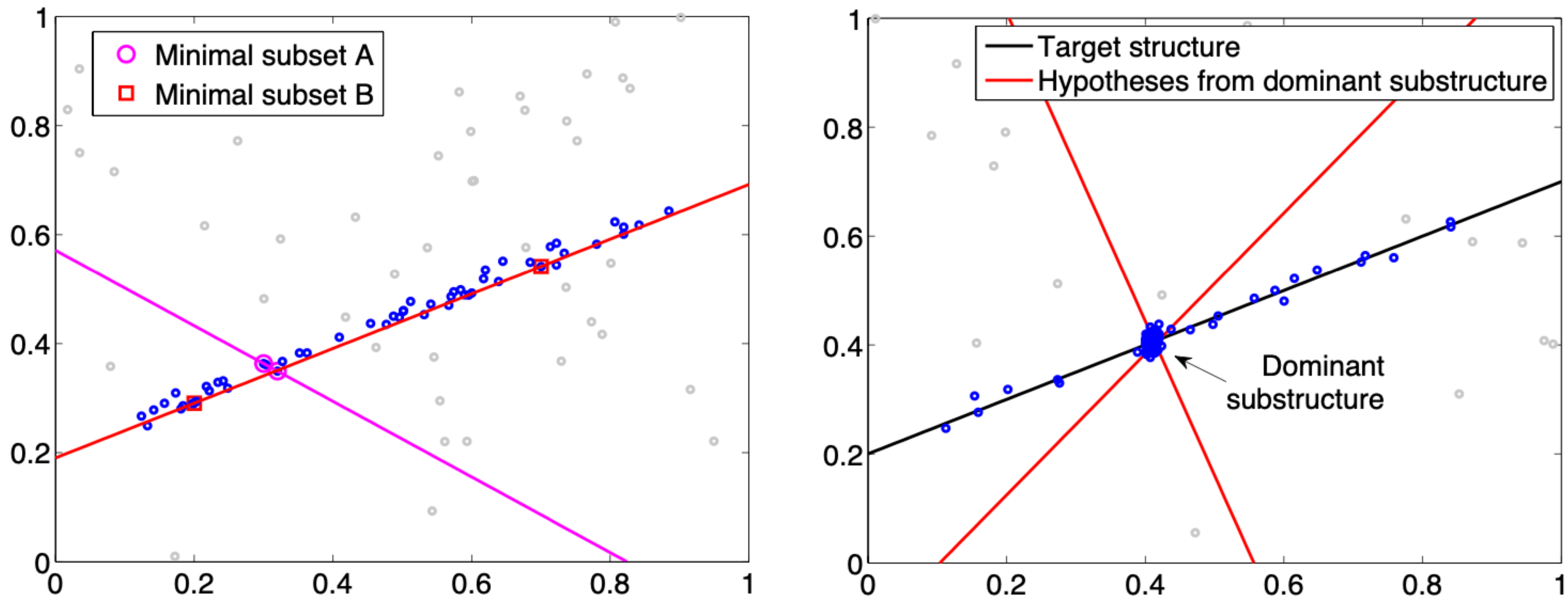
Select data that support the current hypothesis

And the COUNT of the data within tolerance is the score of the hypothesis – repeat – keeping always the currently highest “consensus” count.



(a) (b)

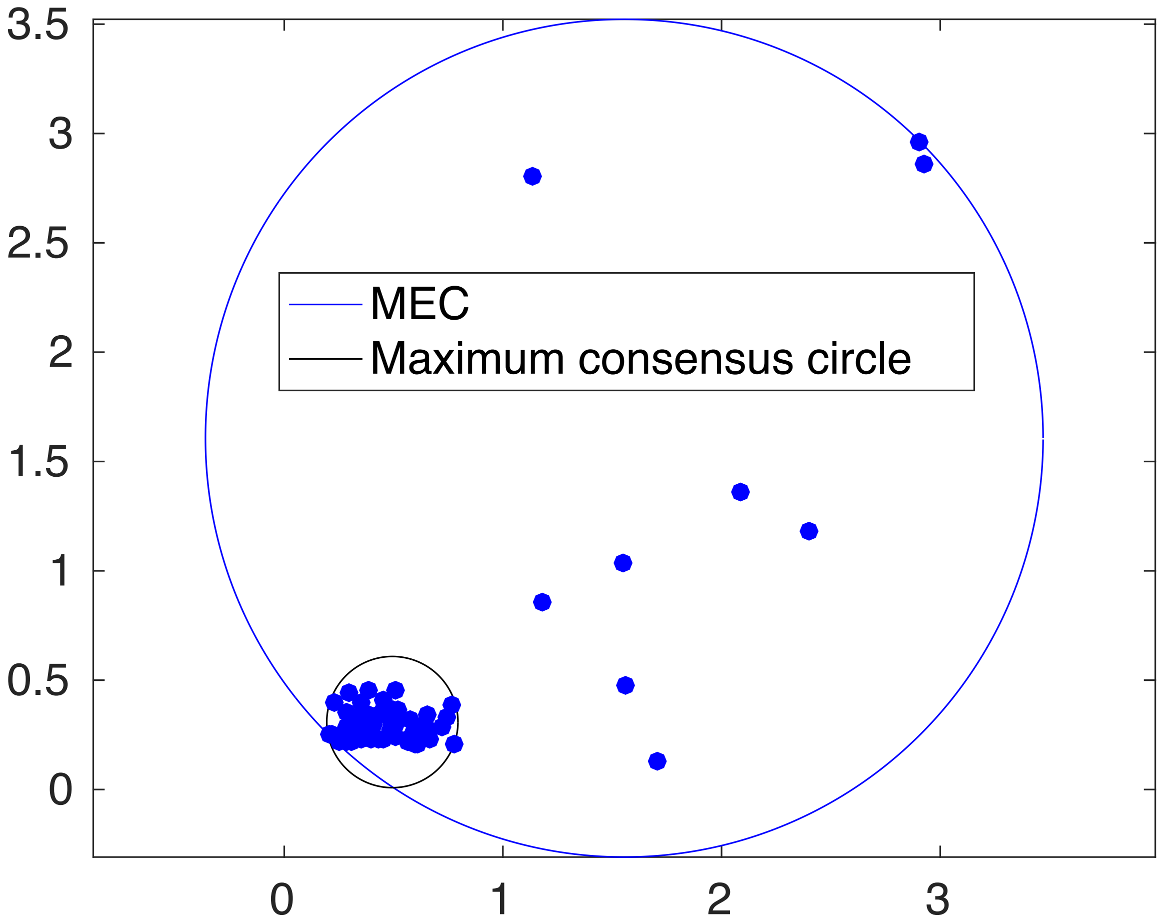
Figure 2.2: (a) An application of Algorithm 3 (RANSAC) on a line fitting problem with $N = 1000$ points and 90% outliers ($\eta = 0.1$). By setting confidence $\gamma = 0.99$, a total of 498 hypotheses were generated in this particular run before terminating in a matter of seconds. (b) A plot of a subset of the hypotheses generated in parameter space $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$.



(a) (b)

Figure 2.4: Illustrating the effects of data span and data degeneracy towards hypothesis generation from minimal subsets. (a) Not all minimal subsets that are composed purely of inliers give good estimates of the model. In particular, a minimal subset with a small span (*e.g.*, minimal subset A in the figure) can lead to a model hypothesis that is arbitrarily far from the target estimate. (b) A quasi-degenerate line fitting problem instance, where most of the inliers lie on a dominant substructure. If the minimal subsets are sampled randomly, most of the all-inlier minimal subsets will be retrieved from the dominant substructure.

Can fit any model by this criterion…



Aligning Point Clouds

Home / News / Maptek expertise helps with Chile rescue

Maptek expertise helps with Chile rescue

Thursday, October 14th, 2010

On 11 August, 2010, Maptek received a call from the team that was working on the rescue in San José mine, asking for help on the drilling control and 3D display to map the drillholes.

We immediately sent one of our Mine Engineers, Alvaro Quezada, to the site to help. Following this, Estíbaliz Echevarría from the Maptek I-Site team travelled to the site to conduct surveys of the region.

This data was taken into Maptek Vulcan 3D software to create an accurate topographic model and 3D representation of the complex underground workings to understand where the 33 miners were trapped

Alvaro then helped to design the direction and orientation of the drillhole which targeted the tunnel, named the Esperanza Drillhole – Hope Drillhole. Maptek Geologist, Sandra Jara, working the shift on 22 August, witnessed the first contact with the miners when she heard the men tapping on the drill as it entered the chamber where they were trapped.

She immediately telephoned Maptek South American Vice-President Marcelo Arancibia and exclaimed “Marcelo, there is life down there!”

Sandra then helped to design and control the orientation of the next two drillholes that also successfully made contact with the 33 miners down below. This final drillhole was the one used for the Plan B access shaft that was used to rescue the men.

Sources of Outliers…

- Imperfections in sensors (multiple reflections of laser, noise in electrical wires, faulty equipment,….)
- Imperfections in preprocessing steps – a “matcher” matching parts of an image to parts of another may give wrong matches (similar image texture around the points even though entirely different points)
- Multiple structures – trying to find a line in an image when there are multiple lines…the data to one line are outliers to the other lines…

(multiple moving objects etc etc)

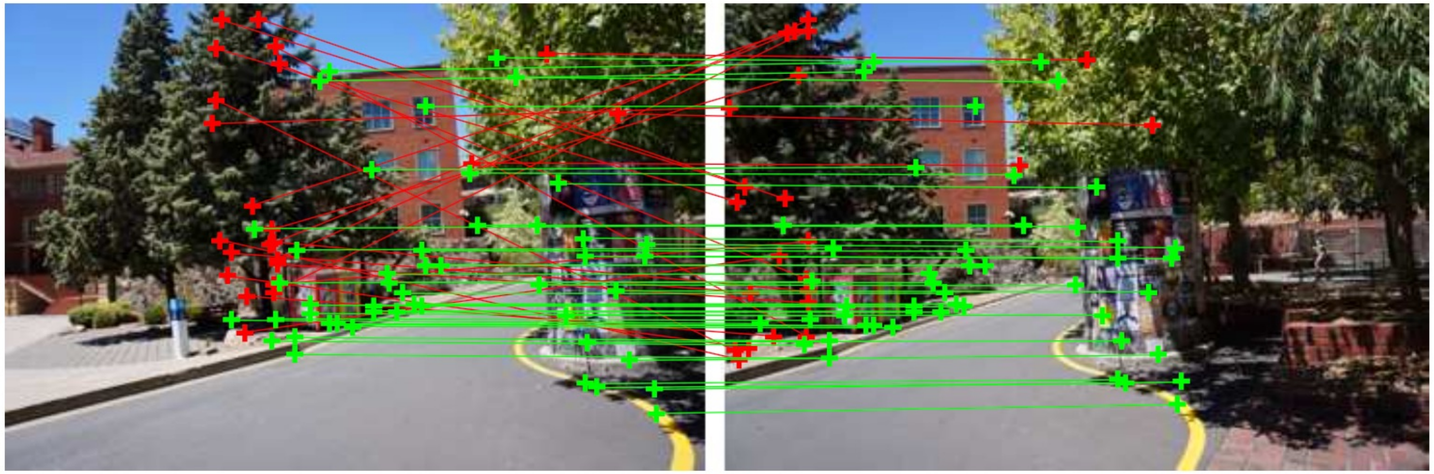


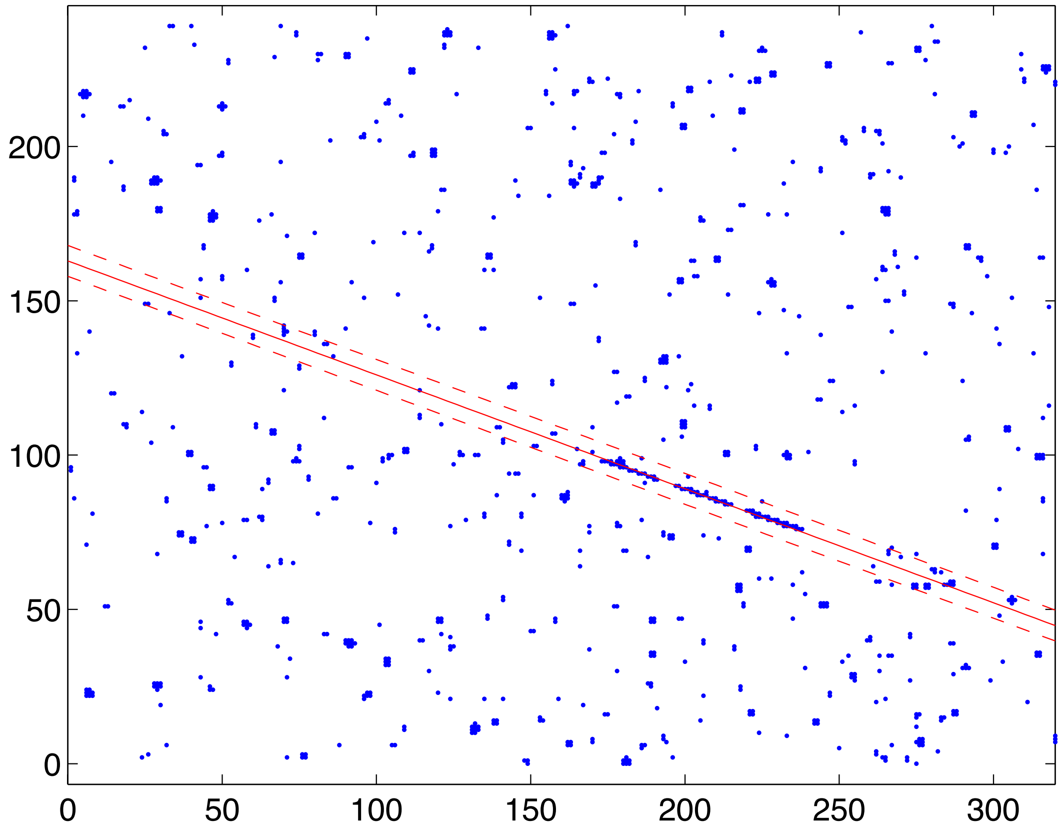
Figure 2.3: Applying RANSAC with an inlier threshold of $\epsilon = 3$ pixels for robust homography fitting on the data in Figure 1.2, where there are a total of 70 feature matches. RANSAC found a consensus set with 47 inlying feature matches (plotted in green) in a matter of seconds. Observe that the data identified as outliers (plotted in red) do correspond to incorrect feature matches.

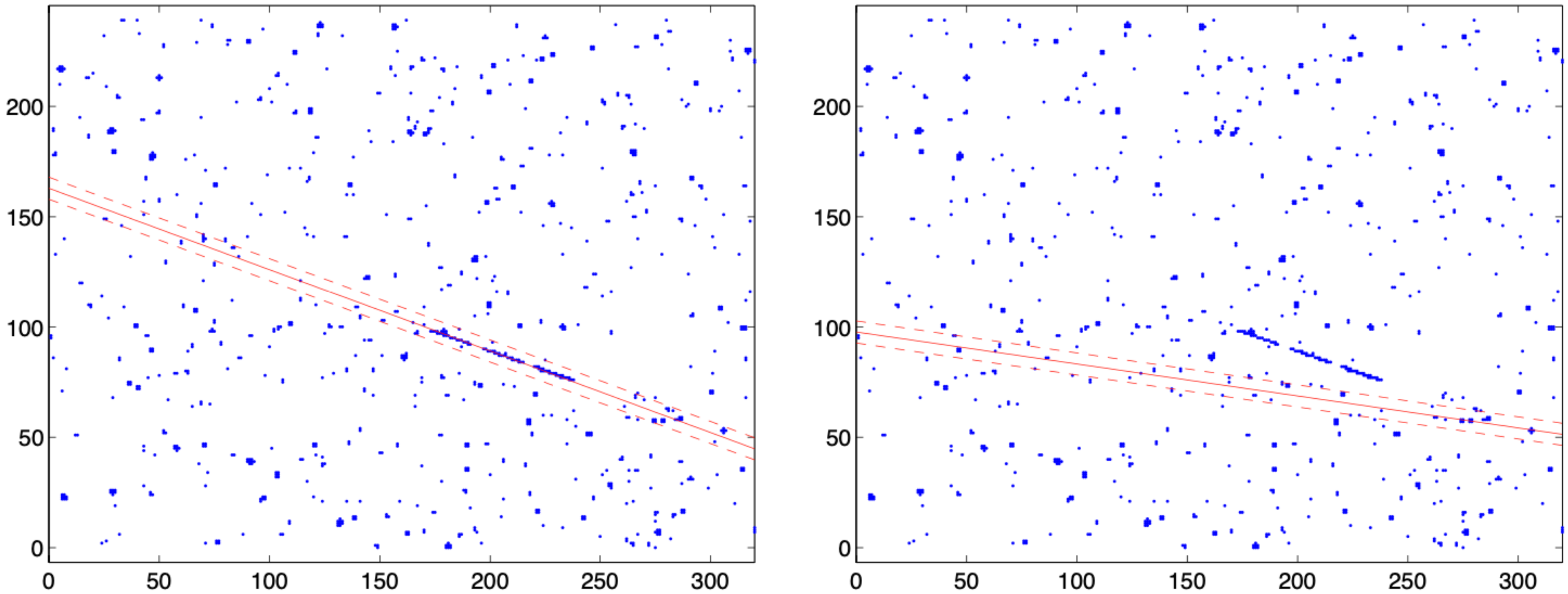
Toy problem….

- The models/equations for most computer vision problems are a distraction to those not versed in the subject….so let's take a simple example that everyone can understand…fitting the line $y=mx+b$ (i.e., finding m and b) to points in 2-D

Maximum Consensus line fitting – the image at left is made by a streak from a moving spaced object against a stationary background of stars…
Find the moving object…

Find the line that has more points within some specified tolerance of that line….





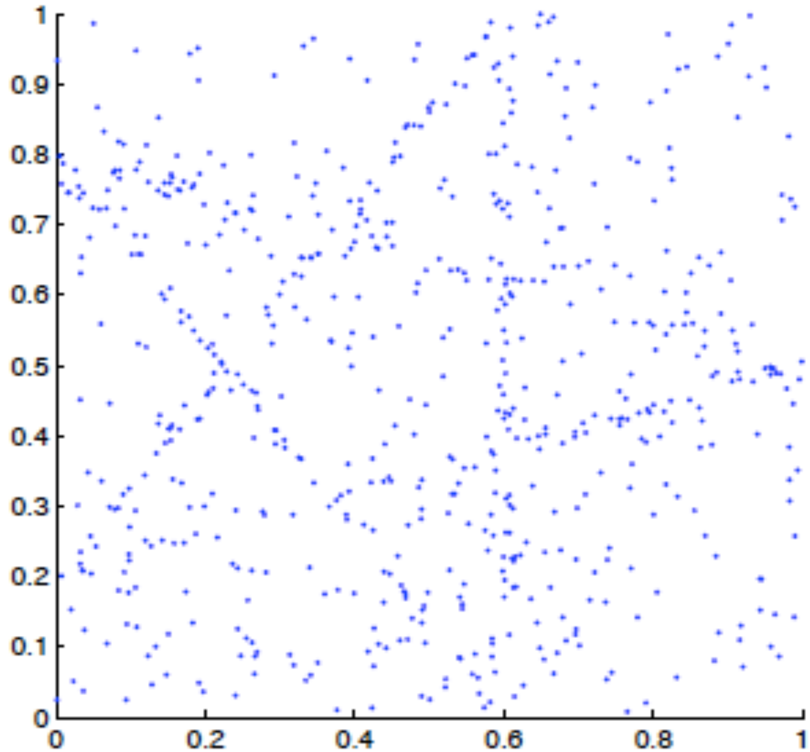
(a) Consensus = 131.

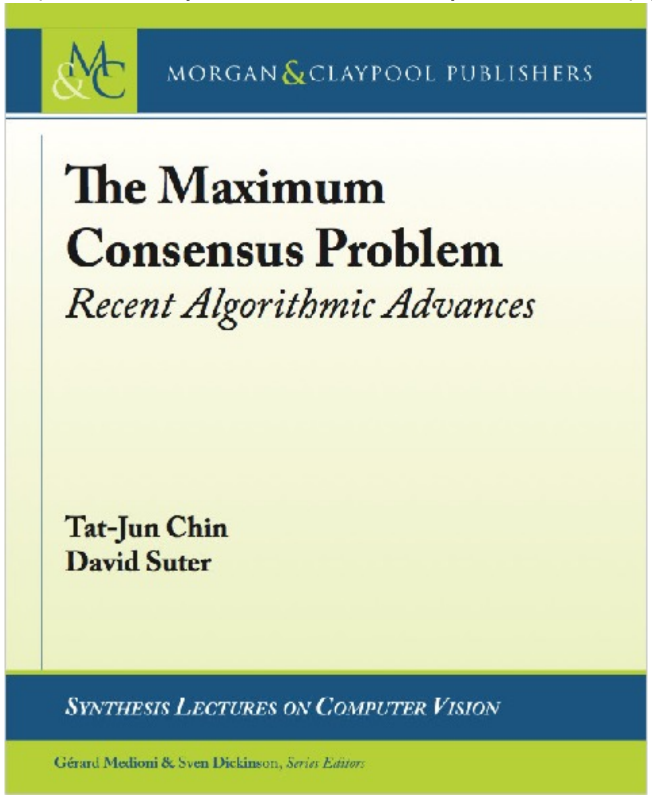
(b) Consensus = 61.

Figure 1.3: Lines with respectively 131 and 61 consensus with $\epsilon = 5$ pixels. The dashed lines indicate the boundary of the inlier threshold ϵ.

So – what do you do? Tracking, Face Recognition, Activity Detection, Human Motion Modelling, Medical Image Analysis, Film Restoration, Forensic Image Analysis, 3D Shape Extraction,…but……

Surprisingly (even to me..☺) a lot of what I have done "boils down" to something akin to line fitting to noisy points (with outliers) – if you can do that **fully automatically** and without knowing things in advance (how many lines?, how many outliers?...) you can do many things…



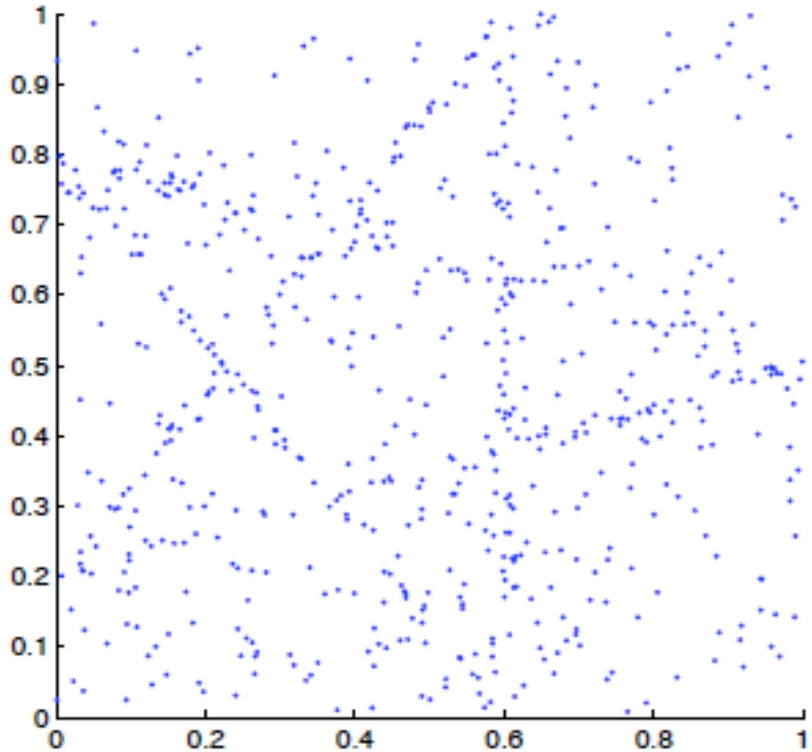


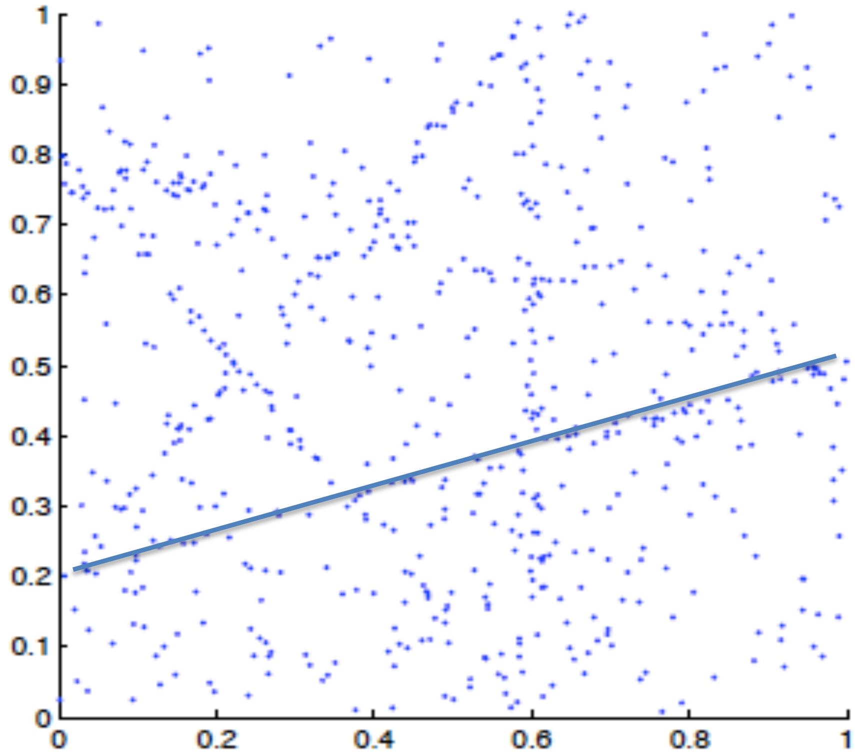
(published Feb 2017)

Strictly speaking, this is about single structure fitting….and with a particular measure of "fit"….

So – what <u>do</u> you do? Tracking, Face Recognition, Activity Detection, Human Motion Modelling, Medical Image Analysis, Film Restoration, Forensic Image Analysis, 3D Shape Extraction,...but......

Surprisingly (even to me..☺) a lot of what I have done “boils down” to something akin to line fitting to noisy points (with outliers) – if you can do that **fully automatically** and without knowing things in advance (how many lines?, how many outliers?...) you can do many things...





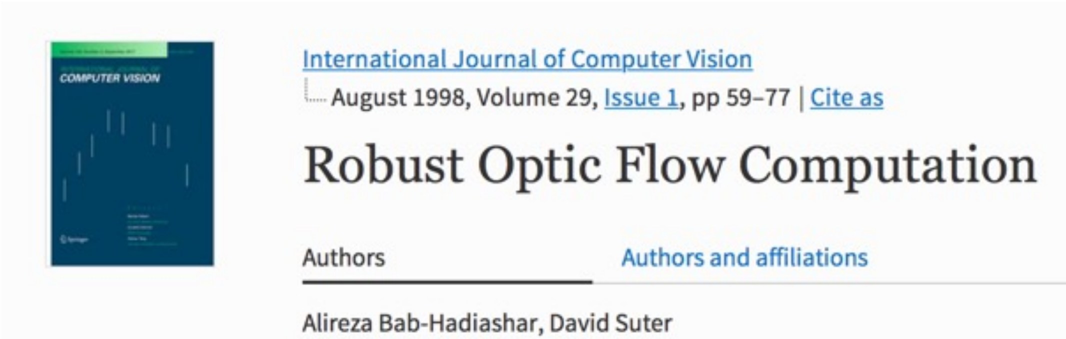
Maximum Consensus

When solving a problem one generally formulates an objective function (measuring how well a proposed solution does), and then finds a way to optimize the objective function.

The objective function of RANSAC is “how many data points are within some specified tolerance distance away from a proposed model/solution”. This is called consensus as it is like these data points agreeing/voting for the proposed solution.

Been studying Maxcon (well, more generally, robust fitting) for a while…..

It all “started” a long time ago…



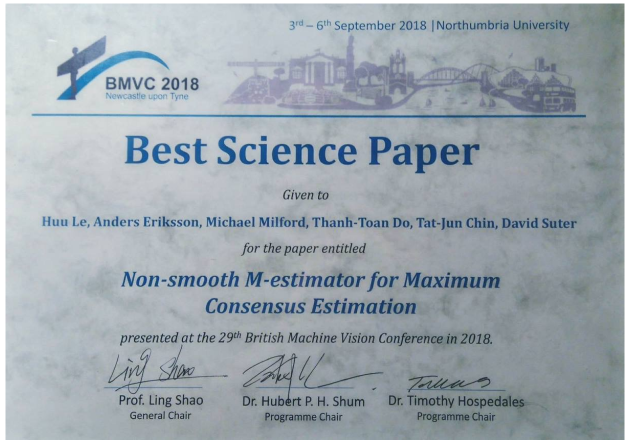
Although that worked with LMedianSquares which has similar complexity issues….

Along the journey CVPR2015 best paper honorable mention award for A* optimal MaxCon

“The CVPR2015 paper”

And BMVC 2018 Best Science Paper award

“The BMVC paper”



Why MaxCon is so hard..EVEN FOR SINGLE STRUCTURE AND EVEN IN LOW DIM?

$$\max_{\theta,\, \mathcal{I} \subseteq \mathcal{X}} \quad |\mathcal{I}|$$

$$\text{subject to} \quad r_i(\theta) \leq \epsilon \;\; \forall \mathbf{x}_i \in \mathcal{I},$$

Usually ϑ is continuous – "so looks like" an optimization/search over a continuous space (albeit with "nasty" local optima structure of objective)

But actually, can also be seen/shown to be a DISCRETE optimization (albeit over a combinatorially huge discrete space).

"Two ways" to "see" this – Theory of LP-type problems (see "our" 2017 book/CVPR 2015 paper…..)
- "Chicken and egg" type problem: if know ϑ then finding and eliminating outliers is trivial, if know outliers/inliers then finding ϑ is trivial.

One key concept – for every model, there is a number p such that p data point can always be fit by that model. For line fitting in 2D – p=2. For ANY two points there is always a line that fits them perfectly. Hence, MaxCon solution >=2.

Moreover, for p+1 points there may be NO model that fits all p+1 points within the tolerance. So p+1 is the size of the smallest possible infeasible (can't be fit by the model within tolerance) subset.

L-infinity fitting a.k.a Chebyshev

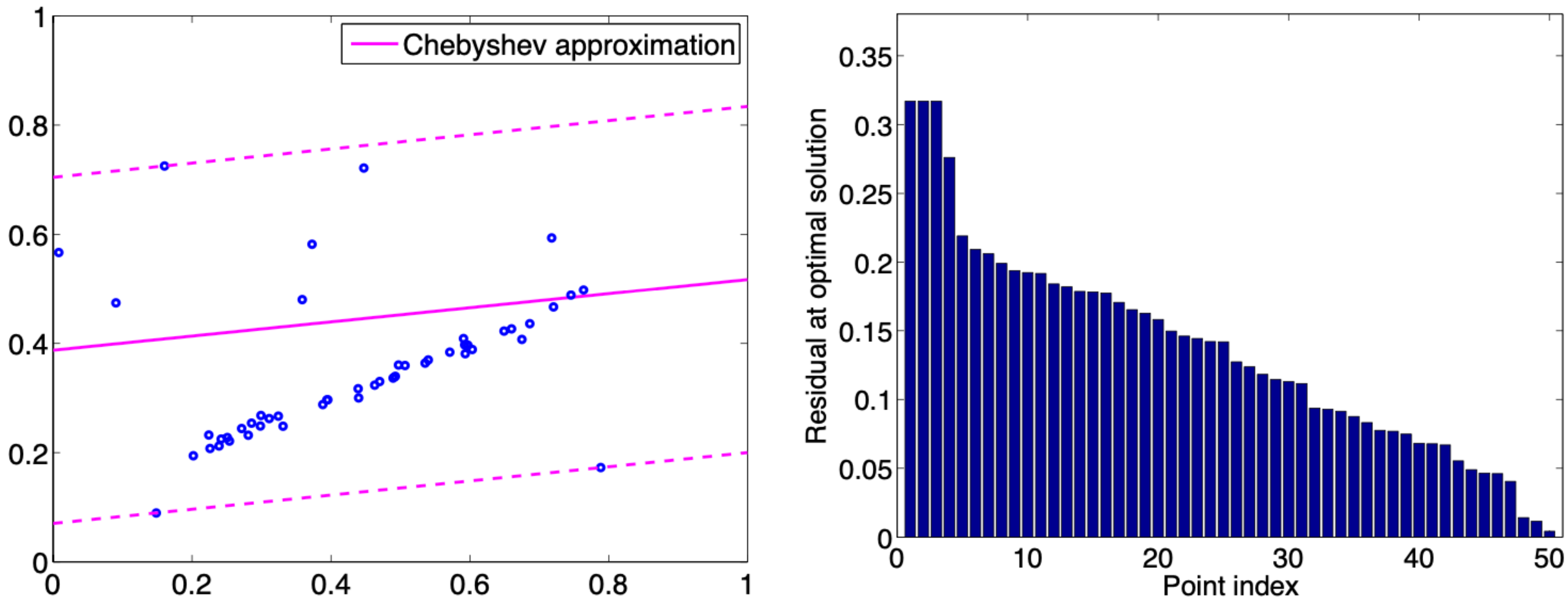


Figure 2.11: Chebyshev approximation (2.46) on line fitting with outliers. In conjunction to the Chebyshev estimate, the dashed lines indicate the optimised minimax residual value. The bar chart on the right shows the sorted residuals at the optimal solution.

When a subset is infeasible it is clear *at least one of the basis points must go…but which?

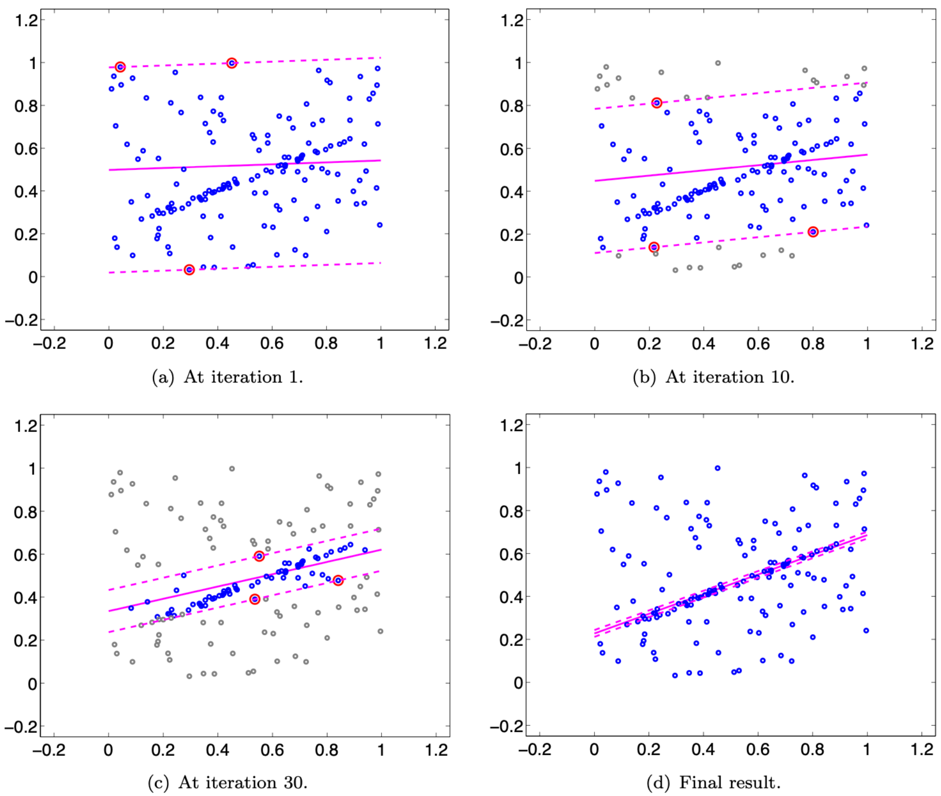
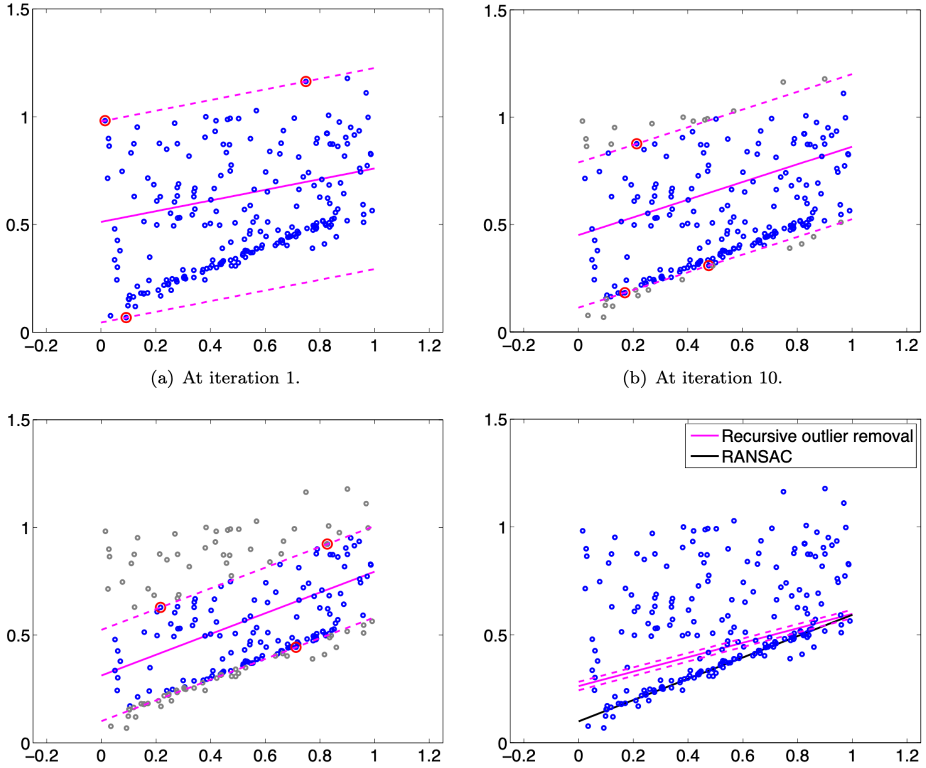


Figure 2.13: Demonstration of outlier removal with ℓ_∞ minimisation (Algorithm 4) on a line fitting problem. Since $d = 2$ for line fitting, at most three points are removed at each iteration.

If you are lucky – removing all basis points might work…

Removing all basis points can fail!



(a) At iteration 1. (b) At iteration 10. (c) At iteration 30. (d) Final result.

Figure 2.14: Applying the ℓ_∞ minimisation technique for outlier removal (Algorithm 4) on a line fitting problem with unbalanced data, *i.e.*, the structure of interest does not lie centrally in the spatial extent of the data. In such a data, the ℓ_∞ technique tends to remove a significant amount of inliers before finding a consensus set. Thus, the maximum consensus solution found is far from ideal. Contrast this to the RANSAC solution, which is much more acceptable.

If we had a test for in feasibility….

- If given a set of points, we could test whether those point can be fit by the model (feasible) then we could search over all subsets (Boolean cube!) – from (111…..1) down to subsets of size p+1, and take the first one we find (highest in the cube – largest subset)!
- There is such a test! L-infinity fitting. Given a set of points find the fit with *maximum* possible largest residual. If this is less than our tolerance, then the set is feasible, else – infeasible.
- Even though L-infinity fitting is reasonably fast, the search space is typically huge and so need to be more clever…
- The CVPR2015 best paper honourable mention work used A* search which is guaranteed to give the true answer, often runs quite fast, but could still be exponential in runtime.,..

Discrete Search - Search over subsets of the data – Search over hypercube/Hasse Diagram

Find the highest feasible set in the Hasse Diagram

All inliers, 0 outliers

(1,1,.....1)

N-1 inlier, 1 outliers

Lowest level that one needs to search downto….

P+1 inlier

p inlier, N-p outliers, p is minimal set to determine θ

(1,0,...0)

(0,...,1)

1 inlier, N-1 outliers

Empty set (0,0,……….0)
(no inliers)

Specifying A* search for MaxCon fitting

- Need to specify:
  - Start state – (1,1,1,…….1) (all the data)
  - Goal state test. Is_feasible(State). (L-infinity fitting)
  - Successor function – select one member from L-infinity fit basis and remove from subset/state. Produces p+1 “child” states .
  - Strategy for selecting state from fringe to expand
    - A* - so need to specify f
      - F(state)=g(state)+h(state)
        - g(state) is just how many data have been removed from start state to get to current state
        - h(state) – needs to be *optimistic* (admissible)
          - One possibility is to “throw” away all p+1 elements of basis and retest (continuing until feasible) and add ONE to h(state) for every iteration.
          - It is optimistic because it could be that each throwing away of a whole basis actually removes more than one true outlier (and thus the count of the number of bases thrown away is an under estimate of the actual number that need to be thrown away.
          - Another more complex (and dominating) heuristic is given in the paper – it *essentially* “reverses” the above heuristic by attempting to put back in the thrown away elements – testing if remains feasible. It is more tricky to understand and prove it is admissible
          - Also uses Max(h1,h2) to produce another heuristic where h2=h(parent_node)-1.

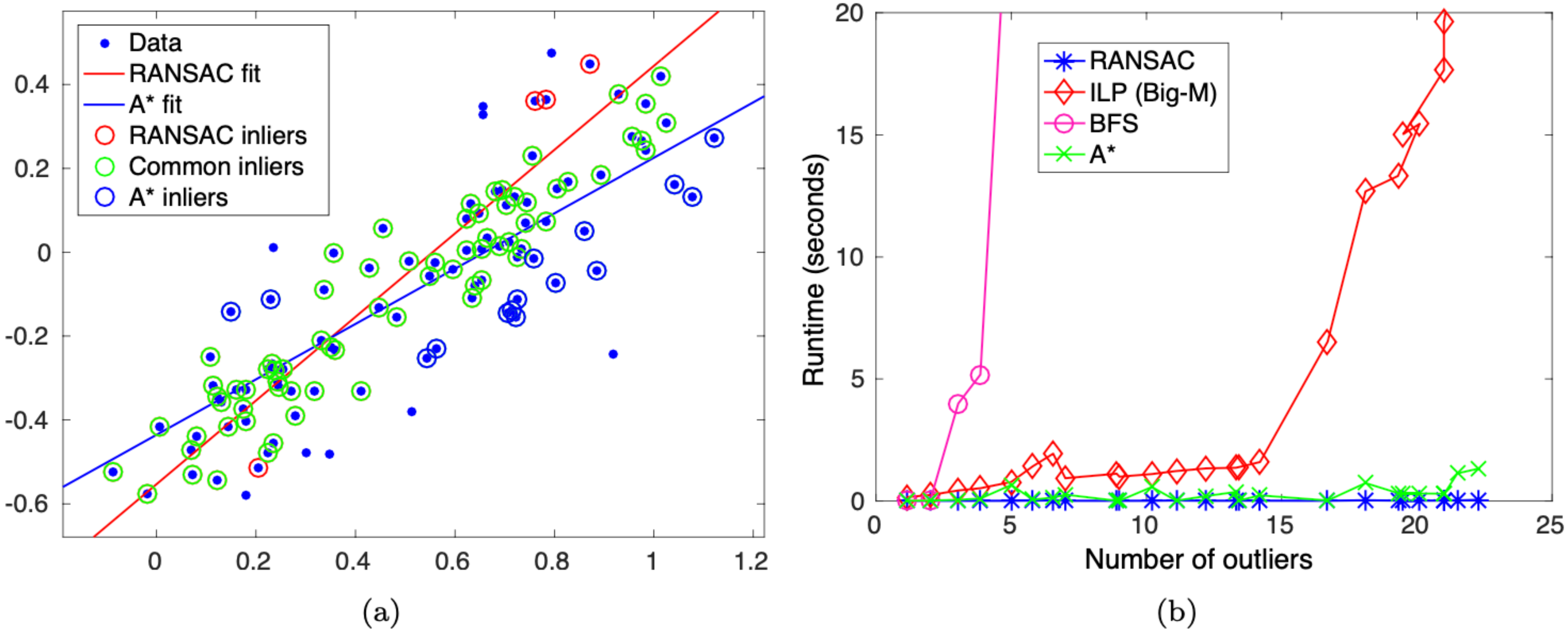


Figure 3.20: (a) Sample result of tree search (using the A* method) and RANSAC for robust line fitting on a randomly generated data instance. RANSAC found a suboptimal solution with consensus 76, while A* found the globally optimal solution with consensus 87. (b) Runtime of several methods as the number of outliers was increased.

BFS – breadth first search, ILP – integer linear programming. These will also find the *optimal solution* but obviously take much longer. For “higher order problems” (point cloud registration, etc) and for more data, A* is still much more efficient than BFS and ILP but nowhere bear as efficient as Ransac – but the latter does not return the optimal value (almost surely)

Other searches…

- We tried bisection searches…(ECCV2018 paper)….each level of the cube is still usually large so still need clever additions to the idea…