Application of Search to Computer Vision: (Robust) Model Fitting

David Suter

References

- CVPR2020 just had a tutorial on “RANSAC” (this section is about “better RANSAC”)

http://cmp.felk.cvut.cz/cvpr2020-ransac-tutorial/

- Book

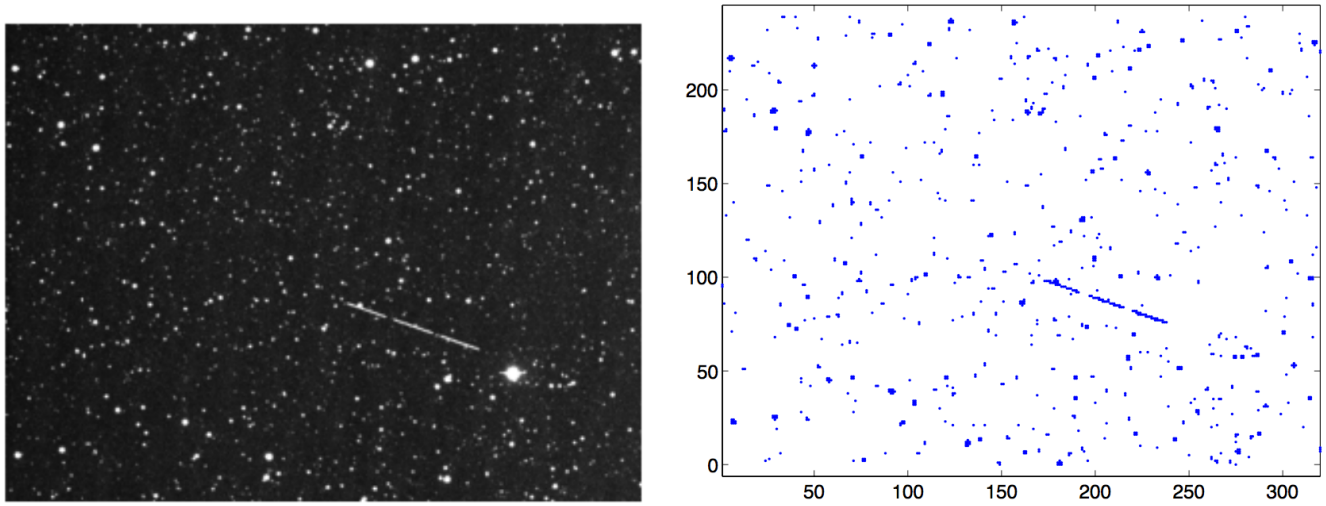
Chin, Tat-Jun and David Suter (2017). The Maximum Consensus Problem: Recent Algorithmic Advances. Synthesis Lectures on Computer Vision (Eds. Gerard Medioni and Sven Dickinson). Morgan & Claypool, pp. 194–. 194 pp. isbn: 9781627052863. doi: 10.2200/S00757ED1V01Y201702COV011.

Which covers the CVPR paper..

Chin, T. J., P. Purkait, A. Eriksson, and D. Suter (2015). “Efficient globally optimal consensus maximisation with tree search”. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR Best Paper Honourable Mention Award, pp. 2413–2421. doi: 10.1109/CVPR.2015.7298855

- And of course the papers the above reference + search web….

Best fitting line Y=mx+b



(a) (b)

Figure 1.1: Estimating satellite trajectory by finding linear streaks. (a) Input image containing an orbiting satellite obtained via telescope (courtesy of Defence Science Technology Group, Australia's primary defence science research organisation). (b) A set of points obtained by intensity thresholding, removal of large blobs and centroiding. Observe that there exist a significant number of outliers, *i.e.*, points not lying on the target line.

Fitting Homography X'=HX

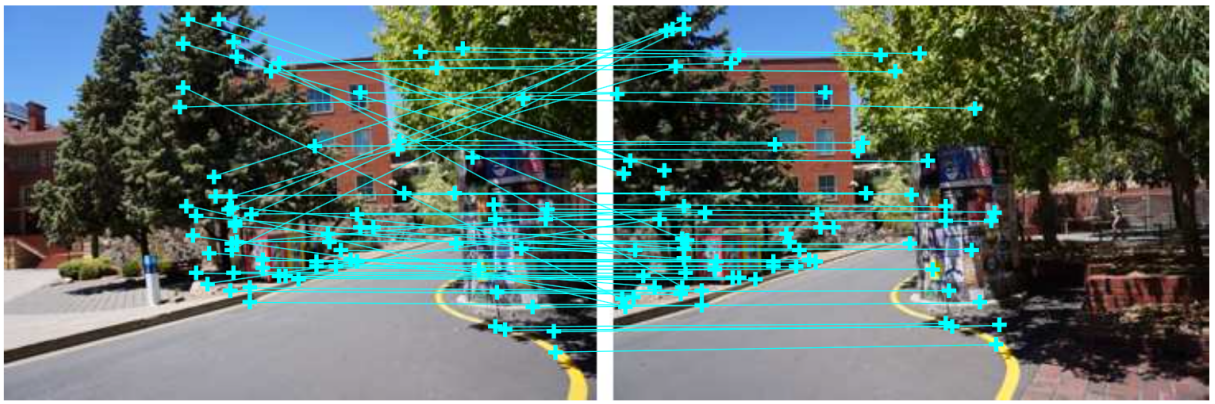
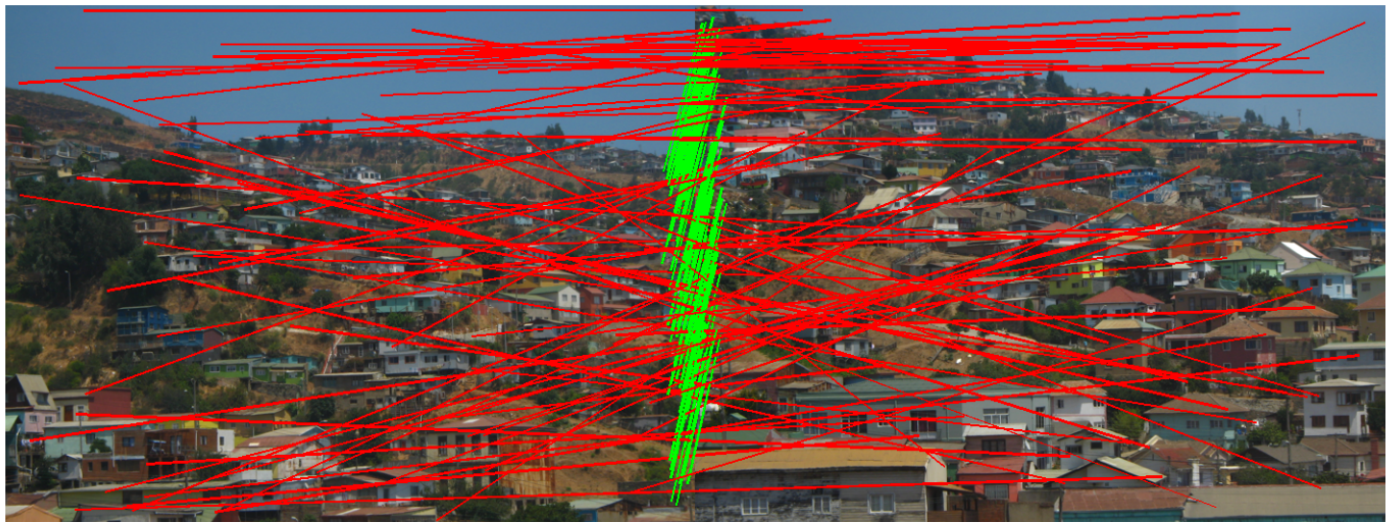


Figure 1.2: To estimate a warping function that aligns two images, an algorithm such as SIFT [Lowe, 2004] is first used to extract putative feature correspondences. Such automatic methods invariably produce mistakes, *e.g.*, observe that parts of one of the trees were wrongfully matched. These incorrect correspondences represent outliers to the correct warping function.



(a)

(b)

Figure 4.8: (a) Input image pair with $N = 154$ feature matches. Green lines are true inliers, whilst red lines are true outliers. (b) Stitched image using the rotation output by Algorithm 20.

The Problem Class

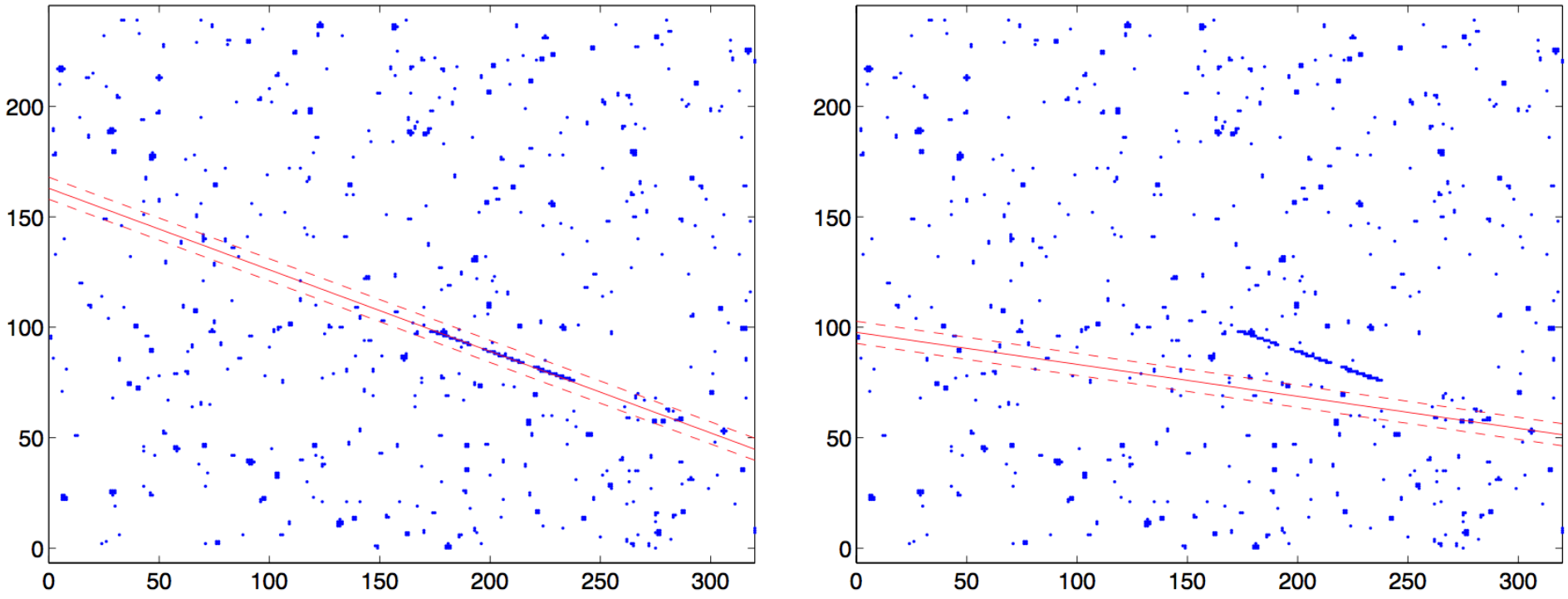
- Have some mathematical model (e.g., linear) that the "(ideal) data of interest" follows (the measured data is this ideal data *plus some small noise*)
- The actual data is the (noisy) measured data *plus some outliers*
- The Problem is to find the parameters of the model and/or what data is "of interest" and what is clutter"
- These are chicken and egg problems – if have one then the other is easy – what you actually want to find (model parameters or inlier/outlier classification may be subject to the application).

Robust Statistics?

- This falls within the subject “robust statistics” and this is has been studied by decades (if not centuries) by statisticians.
- BUT
- We want a fully automated solution (no humans “in the loop” to assess whether it worked or to adjust and re-run…)
- We deal with more outliers than is typical of what statisticians studied
- We often deal with multiple structures in the data – rarely do statisticians consider problems of that complexity – here, though, we will restrict to only one structure in the data

How are such problems commonly solved today?

- Hough Transform (discretize the parameter search space into bins) – data votes for bins compatible with them – highest vote (maximum agreement/consensus) wins
- Standard robust method (m-estimator, Least median of Squares - LMedS….)
- RANSAC (like LMedS but invented by computer vision people). RANSAC solves MaxCon – see next.



(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.

Formal Definition – MaxConc (linear model – relatively easy to generalise to other models)

$$b = \mathbf{a}^T \mathbf{x}, \tag{1.5}$$

$$\left| \mathbf{a}_i^T \mathbf{x} - b_i \right|, \tag{1.6}$$

"Residual" – how close to model

$$\Psi(\mathbf{x}) = \sum_{i=1}^{N} \mathbb{I}\left(\left| \mathbf{a}_i^T \mathbf{x} - b_i \right| \leq \epsilon \right). \tag{1.7}$$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{maximise}} \quad \Psi(\mathbf{x}). \tag{1.8}$$

Formal Definition – MaxConc (generalised to nonlinear models)

Whilst the robust fitting of linear models is fundamentally important to many scientific disciplines, including computer vision, we often need to deal with more complex problems. To generalise (1.8), let ω be a set of parameters from domain Ω (not necessarily Euclidean) that defines the model of interest, and $r_i : \Omega \mapsto \mathbb{R}^+$ be a non-negative function that computes the residual of the i-th datum. The consensus of ω is

$$\Psi(\omega) = \sum_{i=1}^{N} \mathbb{I}\left(r_i(\omega) \leq \epsilon\right), \tag{1.9}$$

and the general maximum consensus problem is defined as

$$\underset{\omega \in \Omega}{\text{maximise}} \quad \Psi(\omega). \tag{1.10}$$

Example 1.1 (Affine registration) Invoking the image alignment problem (Figure 1.2), our goal is to estimate a warping function $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ that aligns two images I and I' from outlier-contaminated point correspondences $\mathcal{D} = \{\mathbf{p}_i, \mathbf{p}'_i\}_{i=1}^N$. We may choose to model f as a affine transformation

$$f(\mathbf{p} \mid \mathbf{\Lambda}) = \mathbf{\Lambda}\tilde{\mathbf{p}} \tag{1.11}$$

where $\tilde{\mathbf{p}} = [\mathbf{p}\ 1]^T$ is $\mathbf{p}$ in homogeneous coordinates, and $\mathbf{\Lambda} \in \mathbb{R}^{2\times 3}$ is a matrix that defines the affine transformation. Given $\mathbf{\Lambda}$, the residual at the i-th "datum" $(\mathbf{p}_i, \mathbf{p}'_i)$ can be defined as the *transfer error*

$$r_i(\mathbf{\Lambda}) = \|f(\mathbf{p}_i \mid \mathbf{\Lambda}) - \mathbf{p}'_i\|_2 \,, \tag{1.12}$$

which is simply the Euclidean distance between the warped version of point $\mathbf{p}_i$ and the observed point $\mathbf{p}'_i$. Identifying $\mathbf{\Lambda}$ as ω and defining Ω as the set of all real matrices of size 2×3, we have thus formulated an instance of (1.10) for robust affine registration.

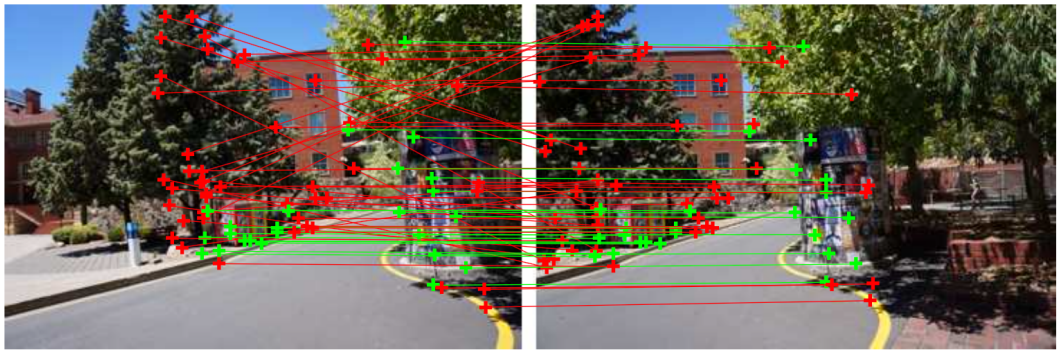


Figure 2.9: ℓ_1 approximation of the maximum consensus affine transformation obtained by solving (2.46). With inlier threshold $\epsilon = 3$ pixels, the SOCP solution has a consensus of 24.

Example 1.2 (Homography fitting) We may also choose to model the warping function f for image alignment as a planar perspective warp or a *homography*

$$f(\mathbf{p} \mid \mathbf{H}) = \frac{\mathbf{H}^{(1:2)}\tilde{\mathbf{p}}}{\mathbf{H}^{(3)}\tilde{\mathbf{p}}}, \tag{1.13}$$

where $\mathbf{H}$ is a 3×3 homogeneous matrix that defines the homography, $\mathbf{H}^{(1:2)}$ is the first-two rows of $\mathbf{H}$, and $\mathbf{H}^{(3)}$ is the third row of $\mathbf{H}$. Given $\mathbf{H}$, the i-th residual can be defined using the transfer error again as

$$r_i(\mathbf{H}) = \|f(\mathbf{p}_i \mid \mathbf{H}) - \mathbf{p}'_i\|_2 \,. \tag{1.14}$$

Identifying $\mathbf{H}$ as ω and defining Ω as the set of all homogeneous 3×3 matrix, we have thus formulated robust homography fitting as an instance of (1.10).

Figure 2.10: ℓ_1 approximation of the maximum consensus homography obtained by solving (2.46). With inlier threshold $\epsilon = 3$ pixels, the SOCP solution has a consensus of 3.

Image “Understanding” –Geometry and Semantics

- Homographies relate planar regions seen in multiple images – the homography encodes geometry between cameras (R,t) and Geometry of plane (d,n)
- So finding the homographies allows us to segment, and to understand the geometry (how many planar surfaces) and to understand some semantics – where are the walls, the floor, the ceiling etc.

Example 1.3 (Triangulation) The 3×4 camera matrix $\mathbf{P}$ underlying an image encodes the pose and internal parameters of the camera that captured the image. A point $\mathbf{x} \in \mathbb{R}^3$ in the scene is projected onto the image according to the function

$$f(\mathbf{P} \mid \mathbf{x}) = \frac{\mathbf{P}^{(1:2)}\tilde{\mathbf{x}}}{\mathbf{P}^{(3)}\tilde{\mathbf{x}}}. \tag{1.15}$$

In the problem of triangulation, we are given N image observations $\mathcal{D} = \{\mathbf{p}_i\}_{i=1}^N$ of the same scene point $\mathbf{x}$, and we wish to estimate $\mathbf{x}$. Let $\mathbf{P}_i$ be the camera matrix of the i-th image. Since a subset of the observations $\mathcal{D}$ may be erroneous (*e.g.*, due to mistakes in feature association), $\mathbf{x}$ must be estimated robustly. Given $\mathbf{x}$, the residual in the i-th view can be taken as the *reprojection error*

$$r_i(\mathbf{x}) = \|f(\mathbf{P}_i \mid \mathbf{x}) - \mathbf{p}_i\|_2 \,, \tag{1.16}$$

which is the Euclidean distance between the projected and observed points.

In this example, the "parameter" ω is the 3D point $\mathbf{x}$. Additional constraints must be placed on $\mathbf{x}$ such that only points that lie in front of all the cameras are allowed. This can be done by forcing the denominators $\mathbf{P}_i^{(3)}\tilde{\mathbf{x}}$ for all i to be strictly positive. Hence, $\Omega = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^3, \mathbf{P}_i^{(3)}\tilde{\mathbf{x}} > 0, i = 1, \ldots, N\}$.

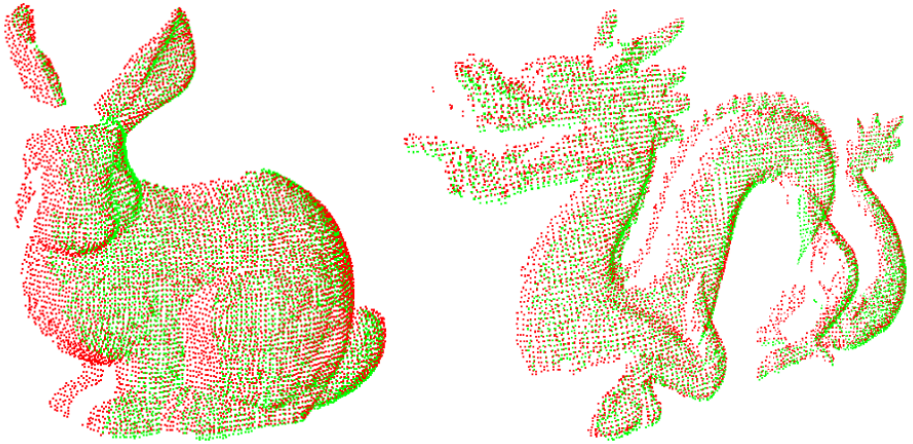
Example 1.4 (Point set registration) Given N corresponding points $\mathcal{D} = \{\mathbf{p}_i, \mathbf{p}'_i\}_{i=1}^{N}$ from two 3D point sets, we wish to estimate the rigid transformation

$$f(\mathbf{p} \mid \mathbf{R}, \mathbf{t}) = \mathbf{R}\mathbf{p} + \mathbf{t} \tag{1.17}$$

that aligns the point sets. Here, $\mathbf{R}$ is a 3×3 rotation matrix, and $\mathbf{t}$ is a translation vector. Since a subset of the correspondences $\mathcal{D}$ may be incorrect (*i.e.*, the outliers), the rigid transform must be estimated robustly. Given candidate parameters $(\mathbf{R}, \mathbf{t})$, the residual at the i-th datum $(\mathbf{p}_i, \mathbf{p}'_i)$ can be defined as

$$r_i(\mathbf{R}, \mathbf{t}) = \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{p}'_i\|_2 \,, \tag{1.18}$$

which is simply the Euclidean distance between the rigidly transformed version of $\mathbf{p}_i$ and the matching point $\mathbf{p}'_i$. In this example, $\omega = (\mathbf{R}, \mathbf{t})$, and Ω is the space of all rigid transformations, also known as the special Euclidean group $SE(3)$.



(a) (b)

Why are these search problems?

- Well intuition (especially for the easy to visualise line fitting problem) tells you it can be formulated as a search – but that is in an infinite search space of all orientations of the line and all translations of the line).
- One can of course discretize the search (e.g., Hough Transform) to get an approximate solution over a finite search BUT…
- it takes a lot more work to realise/explain that it can be formulated as an EXACT solution of a FINITE TREE SEARCH (and therefore can be solved by A* if you have good heuristics).

Approx. Max Con

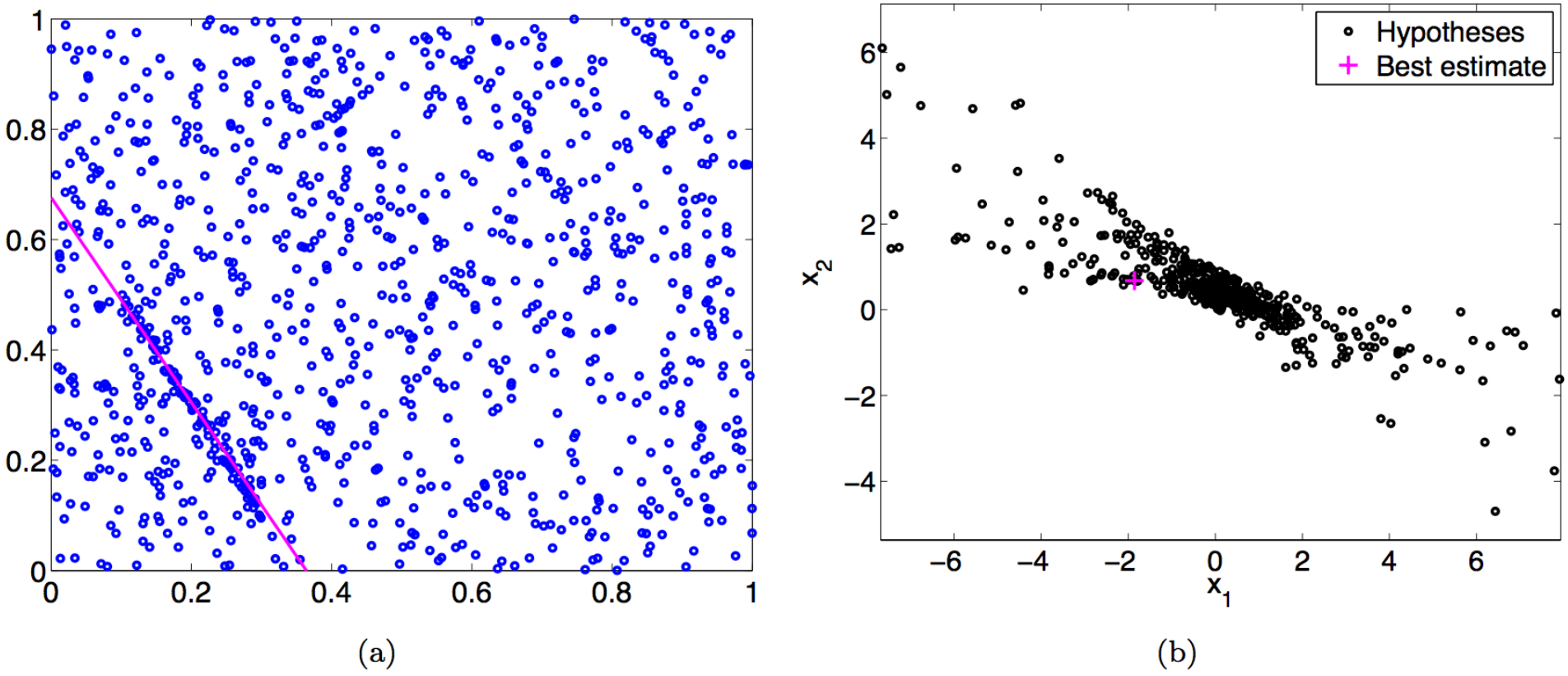
(and starting journey from continuous search problem to discrete search problem)

Hough Transform

- Invented in 1959 by Hough to detect tracks in cloud chambers – particle physics (other end of scale to astronomical data showed earlier!). Patented in 1961. Generalised and popularised by Duda and Hart 1979/81.
- Simple idea – discetize parameter space and count votes (each data point votes for bins consistent with it)

RANSAC – Fischler and Bolles 1981

- Basic idea
  - sample enough data to determine a candidate model (fit) – d-tuple (for line fitting in 2D d=2)
  - This d-tuple might have outliers – so generate lots of d-tuples in the hope that at least one d-tuple is "clean"
  - Count the number of data that "agree" with the hypothesised fit. Keep the best scoring one.



(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 before termination in a matter of seconds. (b) A plot of a subset of the 498 hypotheses generated in parameter space $\mathbf{x} = [x_1 \; x_2]^T \in \mathbb{R}^2$.

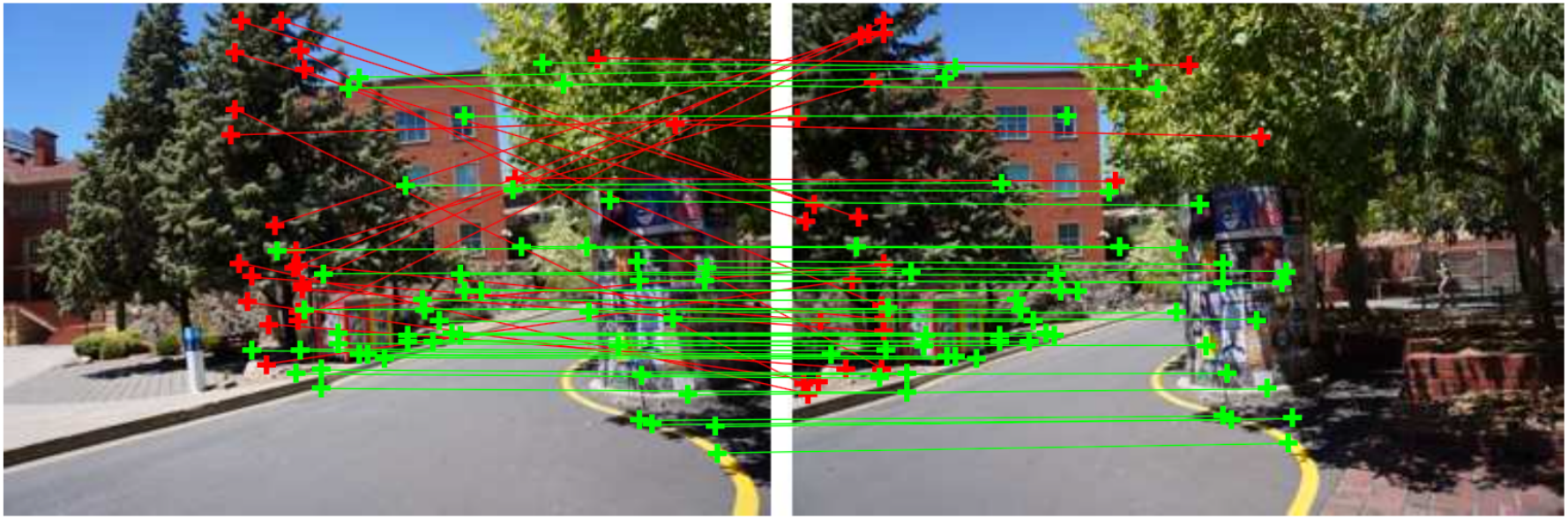


Figure 2.3: Applying RANSAC with an inlier threshold of $\epsilon = 3$ pixels for robust homography fitting on the data in Figure 1.2, where there are a total of 70 feature matches. RANSAC found a consensus set with 47 inlying feature matches (plotted in green) in a matter of seconds. Observe that the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

Pro's and Con's

- RANSAC is simple to implement
- If you don't want "the best" solution it generally gives a good enough solution reasonably quickly
- Can be improved (see references for some pointers)
- BUT…
- Is not optimal (or even locally optimal – see references for definition)
- Comes with NO GAURANTEES – no guarantee of good result nor even bounds on how bad result can be (even if knew outlier rate and tolerance – both of which don't know anyway!!)
- That is, you get an answer: but you have no sure way to even know how good that answer was (!!)

Key Idea to Discretize Search

- For a model fit (within tolerance) there are always “extreme” points that determine how small that tolerance can be.
- These can be found by “infinity norm” solution – see references if interested
- Related to computational geometry notion/fact – LP-type problems have a “basis” (a fixed number of points that ”tell the whole” story (other data “don’t matter”). In our case the “whole story” is “the smallest tolerance to fit all the data”

The infinity norm solution is completely determined by the worst of the outliers(!!!)

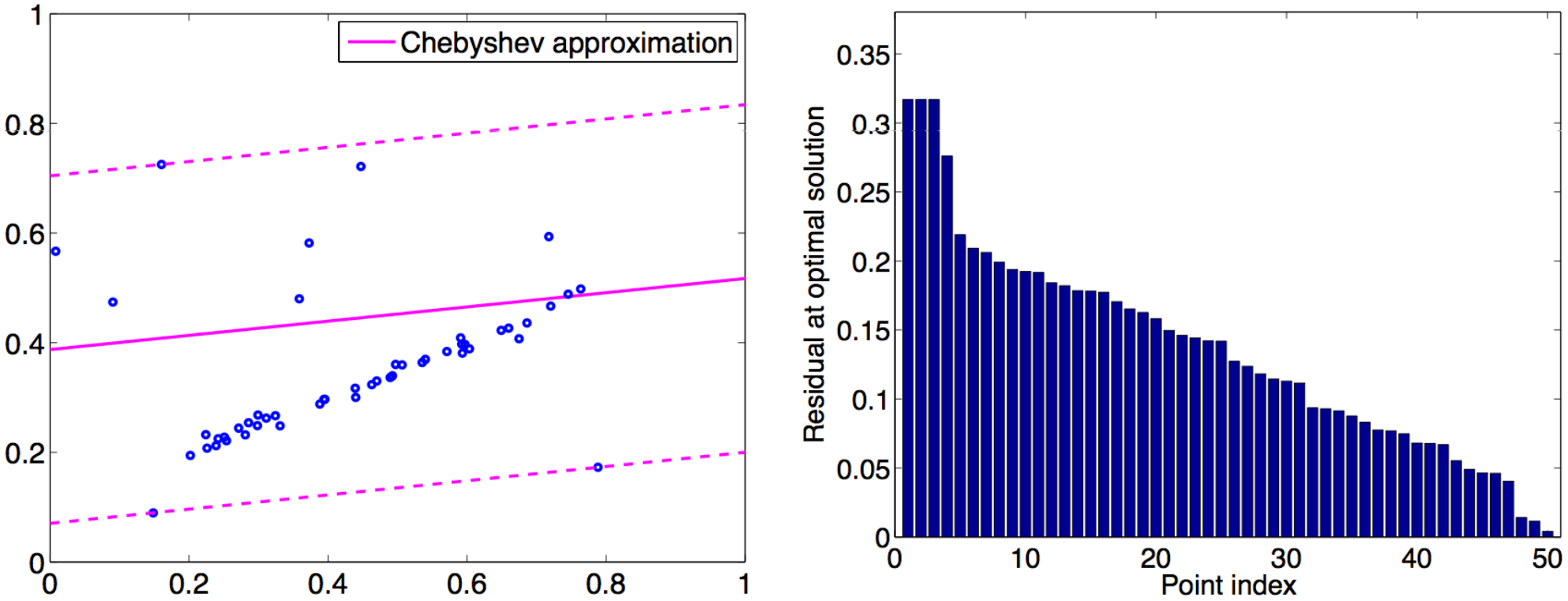
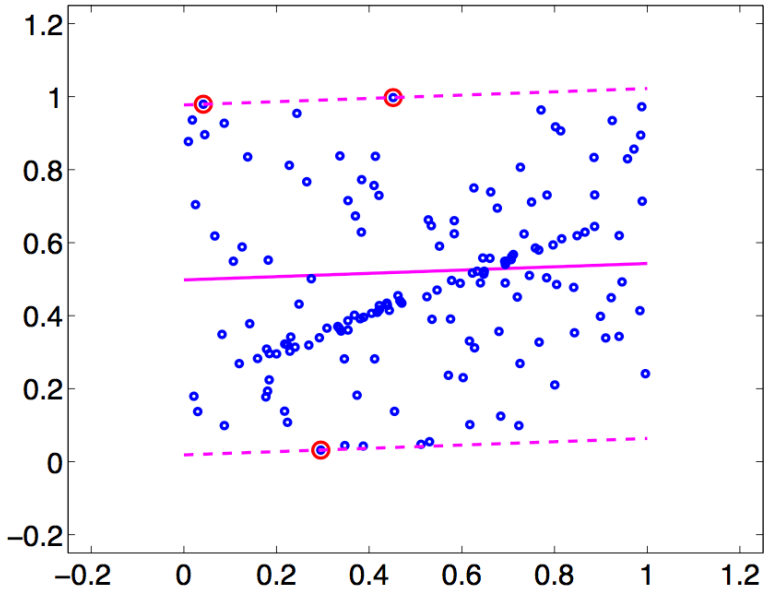
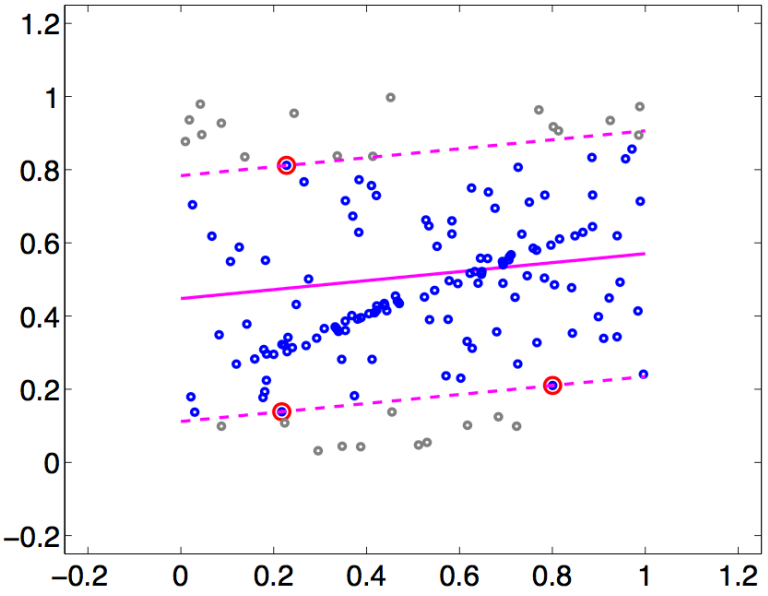


Figure 2.11: Chebyshev approximation (2.47) on line fitting with outliers. In conjunction to the Chebyshev estimate, the dashed lines indicate the optimised minimax residual value. The bar chart on the right shows the sorted residuals at the optimal solution.

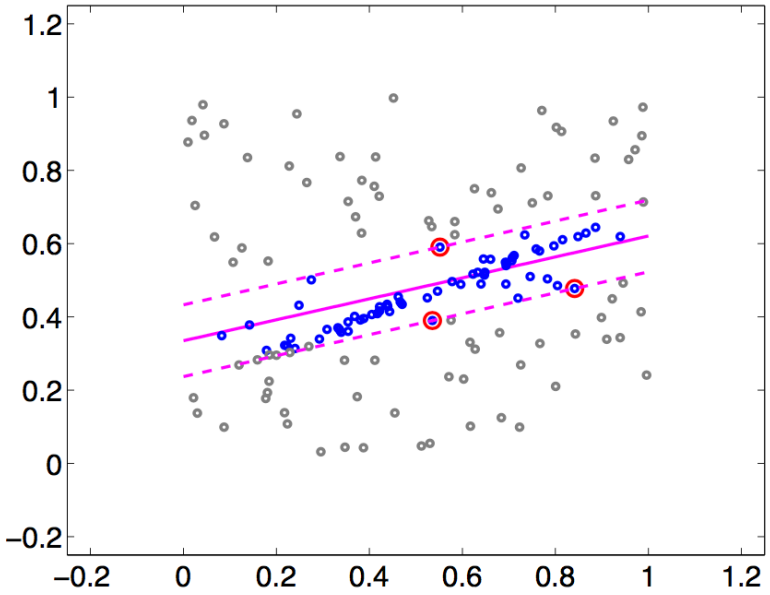
Just throw away d+1 points until fits??



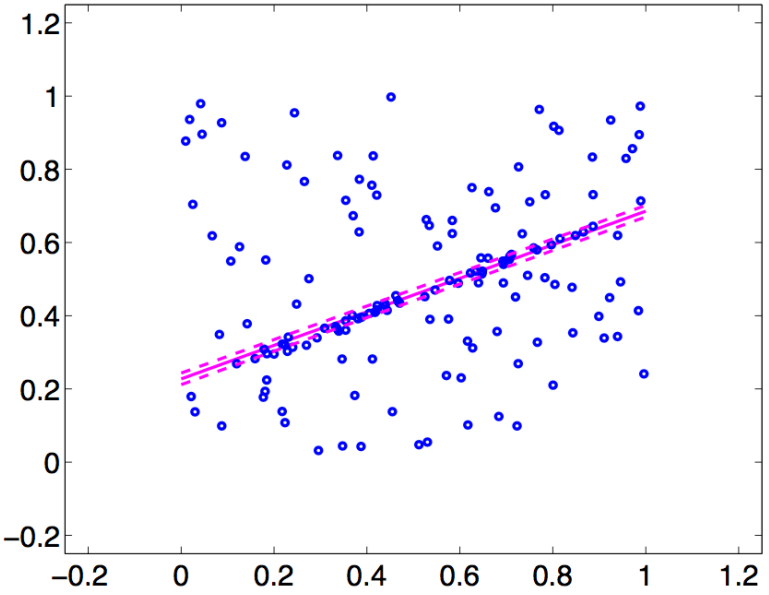
(a) At iteration 1.



(b) At iteration 10.



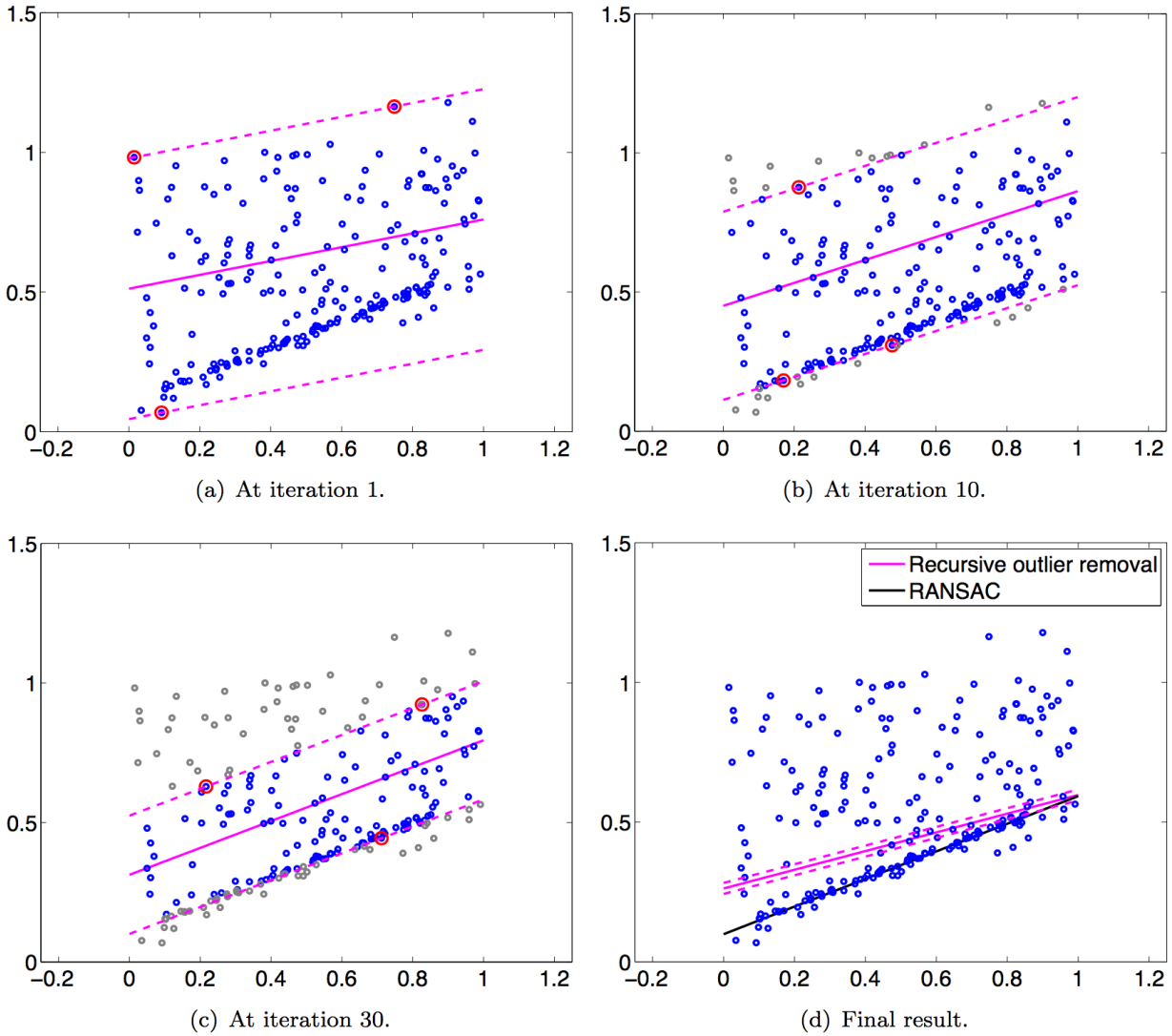
(c) At iteration 30.



(d) Final result.

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.

But.....



(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.

Only ONE of the d+1 points is guaranteed to be an outlier! But which one? See the connection between "which road to take" in path planning?

Exact Max Consensus

Maximum consensus as tree search

Taking advantage of the tree structure, we redefine problem (3.59) as

$$\begin{aligned} \underset{\mathcal{B},\ \mathcal{B} \text{ is a basis}}{\text{minimise}} \quad & l(\mathcal{B}) \\ \text{subject to} \quad & f(\mathcal{B}) \leq \epsilon. \end{aligned} \tag{3.62}$$

Here, instead of maximising its coverage, we minimise the level of the basis. Formulation (3.62) thus seeks the *shallowest* basis that is feasible. Figure 3.17 illustrates this version of maximum consensus. The set of feasible bases are the nodes situated below the horizontal line of height ϵ. Amongst the feasible bases, the ones with the lowest level (level 4 in this example) are coloured in green - these are the global minimisers of problem (3.62). Intuitively, for this example, the maximum consensus set must exclude exactly four constraints.

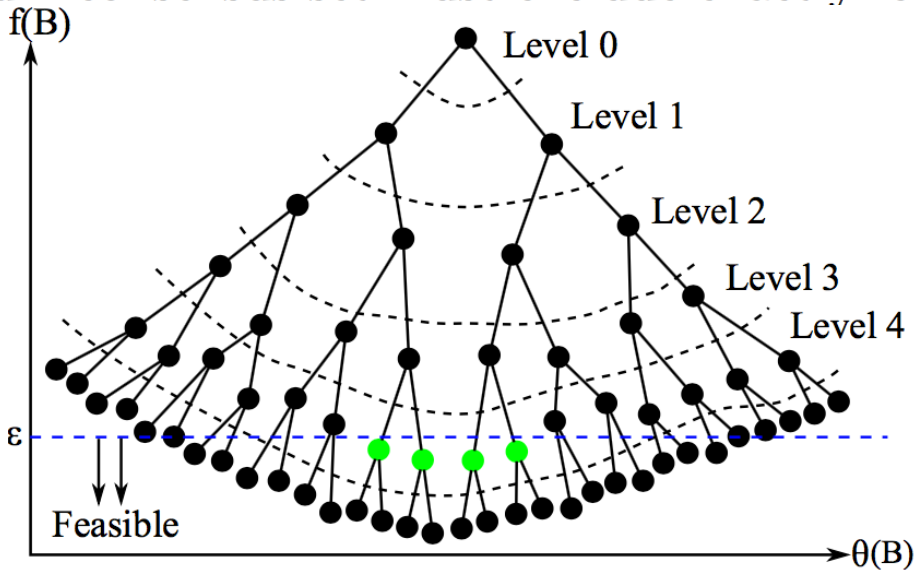


Figure 3.17: Maximum consensus as tree search. The set of feasible bases are situated below the horizontal line of height ϵ. The feasible bases with the lowest level are the nodes in green.

So tree search – but for A* we need heuristics!

- What is a heuristic in this setting?
- It must be greater than 0 but LESS THAN OR EQUAL to the TRUE number of constraints (data in our problems) we need to eliminate in order to get a feasible solution

With a heuristic h, the A* is simply a priority queue on the fringe of the tree search with priority given by (lowest) e

$$e(\mathcal{B}) = l(\mathcal{B}) + h(\mathcal{B}), \qquad (3.63)$$

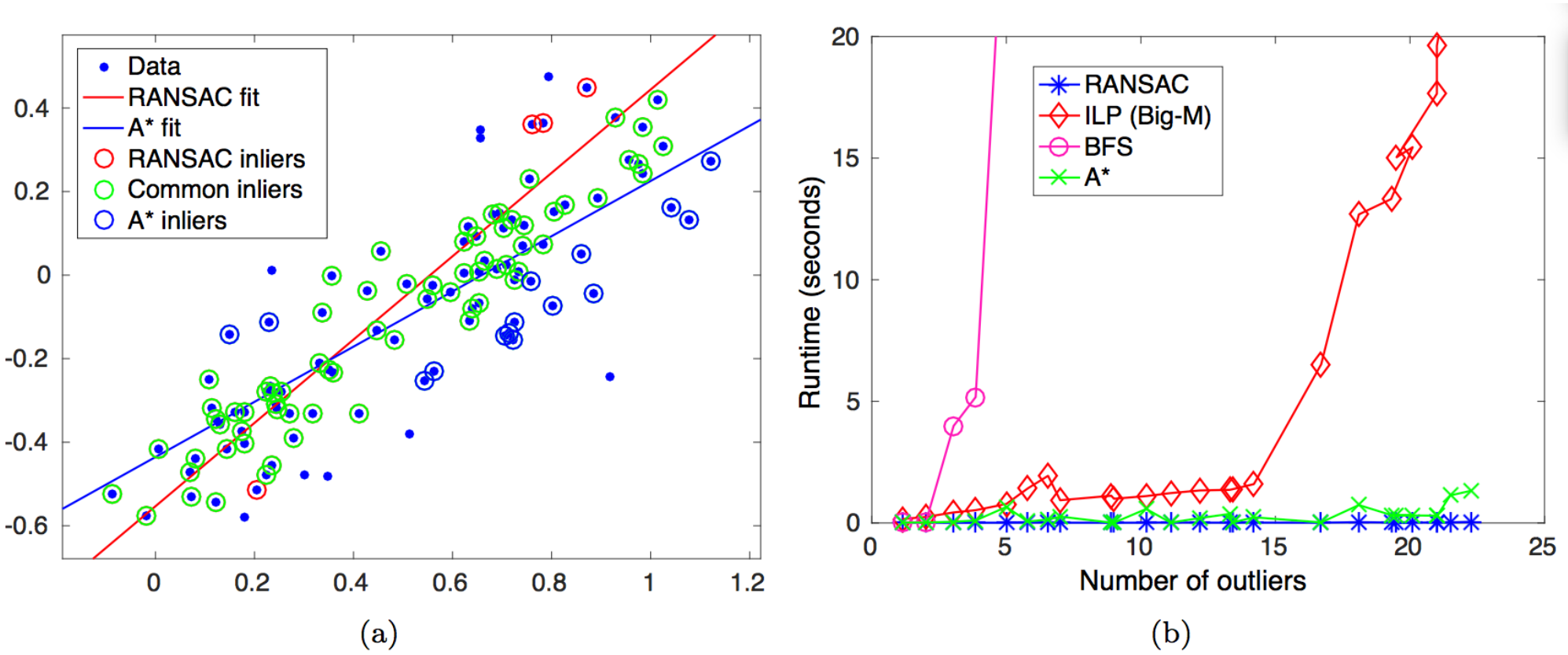
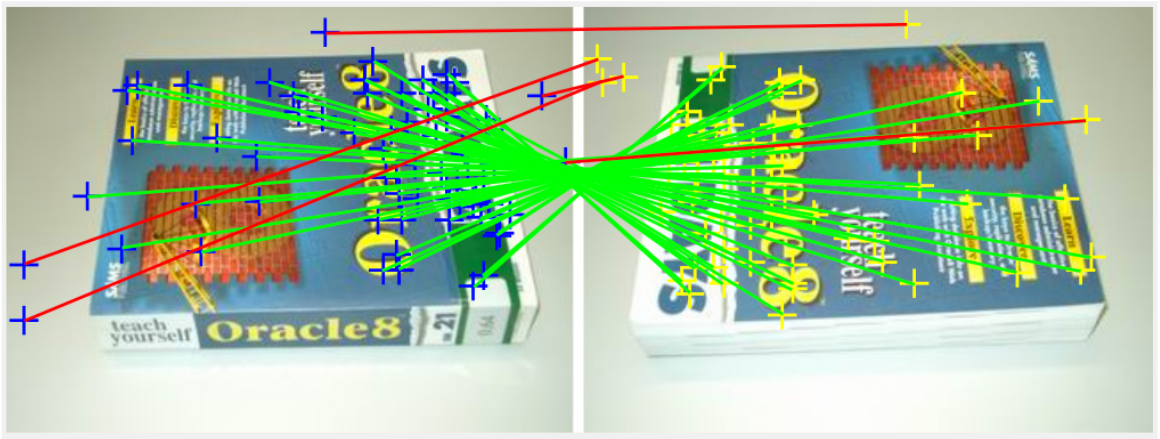
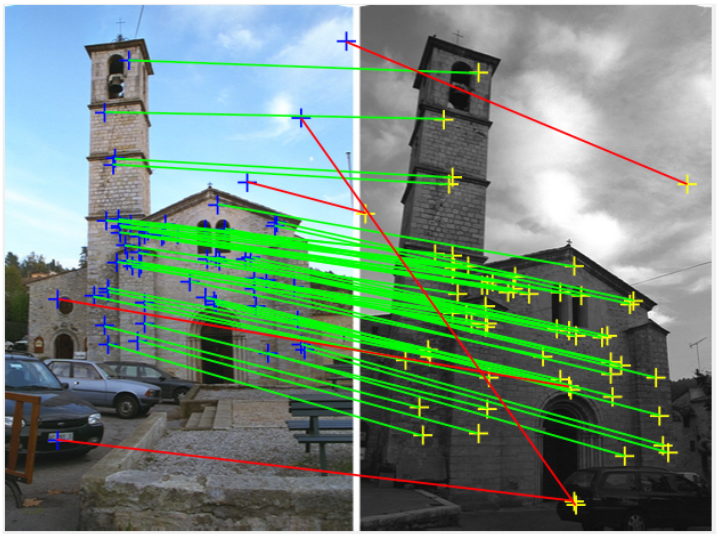


Figure 3.20: (a) Sample result of tree search (using the A* method) and RANSAC for robust line fitting on a randomly generated data instance. RANSAC found a suboptimal solution with consensus 76, while A* found the globally optimal solution with consensus 87. (b) Runtime of several methods as the number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



(a) Problem size $N = 70$, maximum tree depth to explore $\ell^* = 5$.



(b) Problem size $N = 58$, maximum tree depth to explore $\ell^* = 4$.

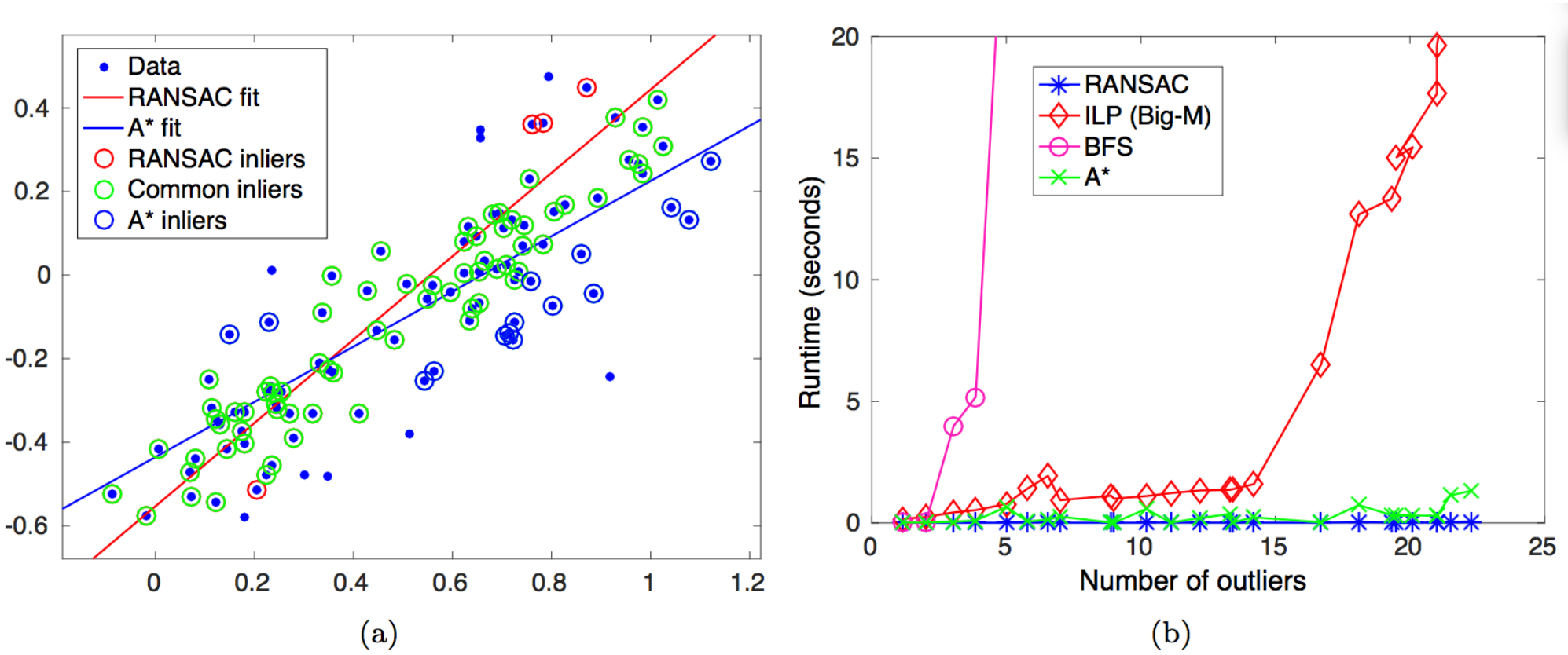
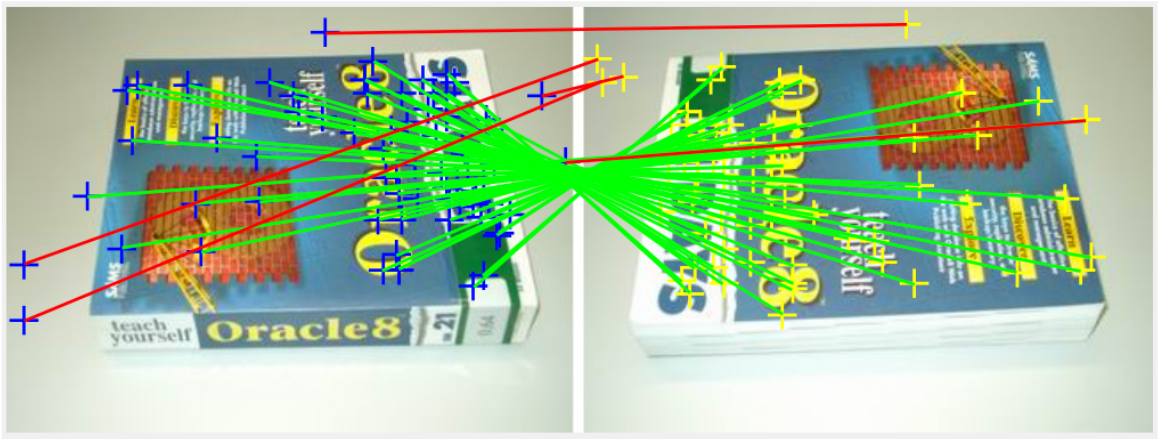
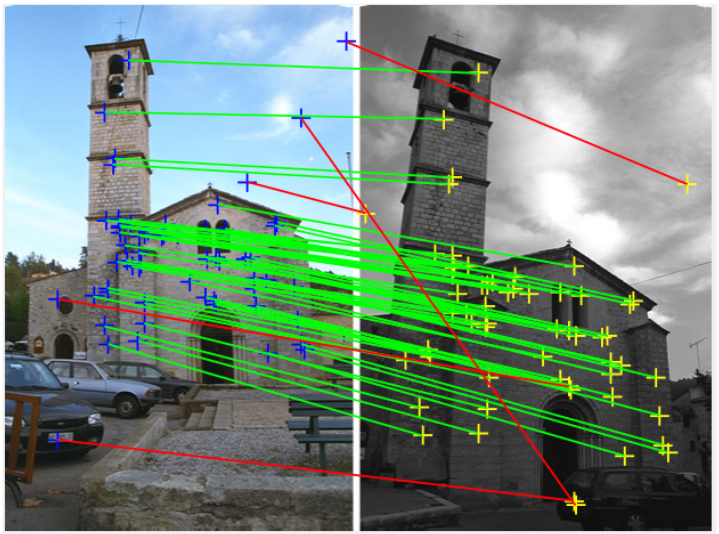


Figure 3.20: (a) Sample result of tree search (using the A* method) and RANSAC for robust line fitting on a randomly generated data instance. RANSAC found a suboptimal solution with consensus 76, while A* found the globally optimal solution with consensus 87. (b) Runtime of several methods as the number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



(a) Problem size $N = 70$, maximum tree depth to explore $\ell^* = 5$.



(b) Problem size $N = 58$, maximum tree depth to explore $\ell^* = 4$.