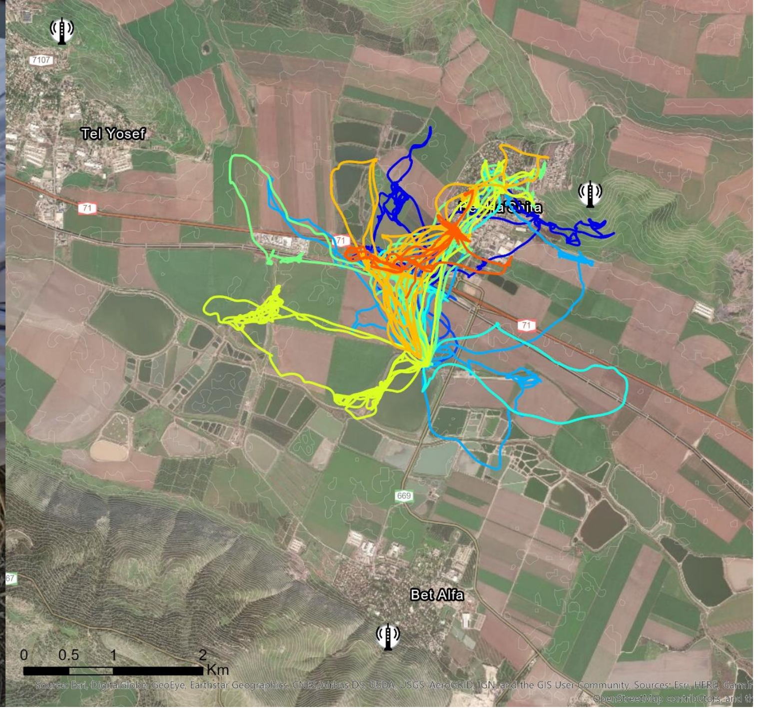
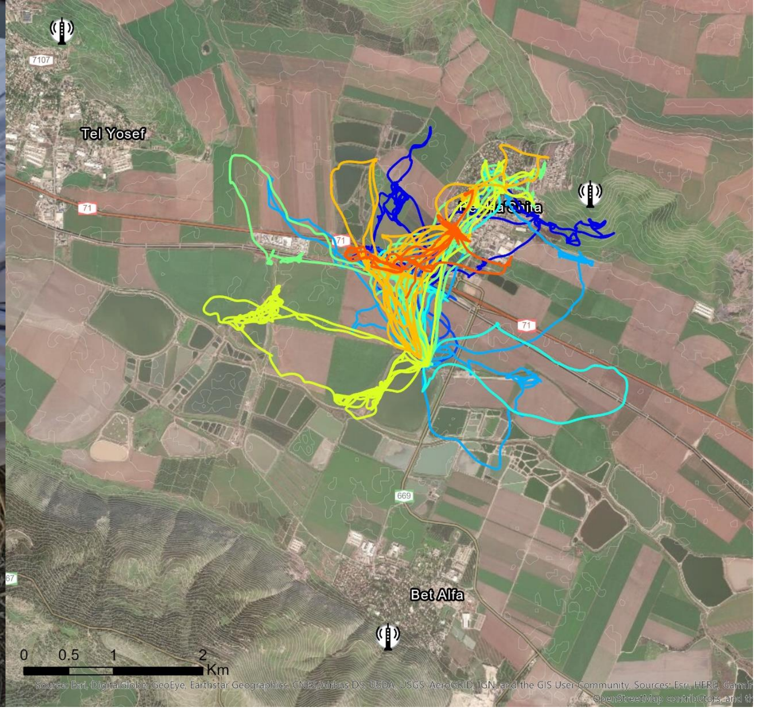
Combinatorial Problems and Algorithms in Robust Estimation

Sivan Toledo
Blavatnik School of Computer Science
Tel Aviv University

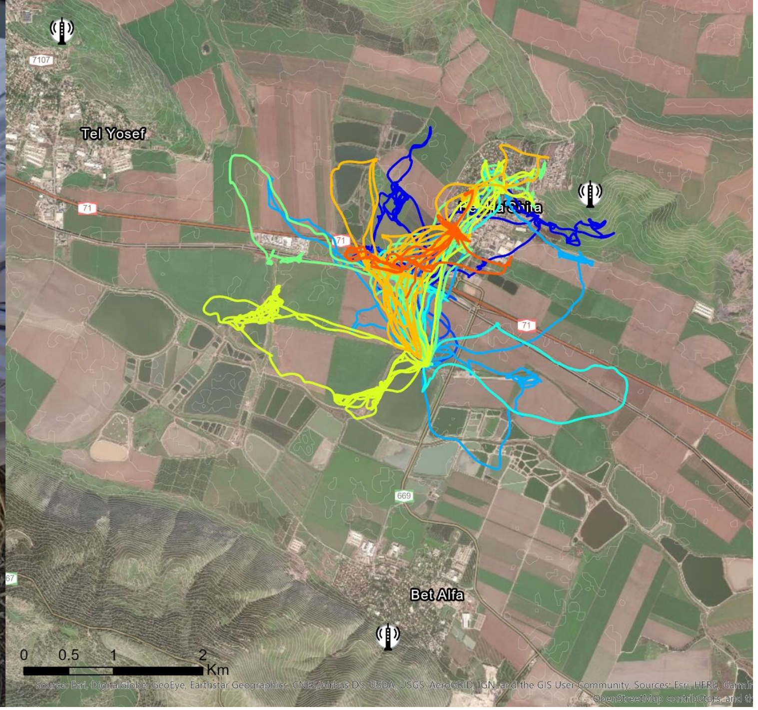
Eitam Arnon
Tel Aviv University


Tagging & image credit:
Shlomo Cain & Orr Spiegel

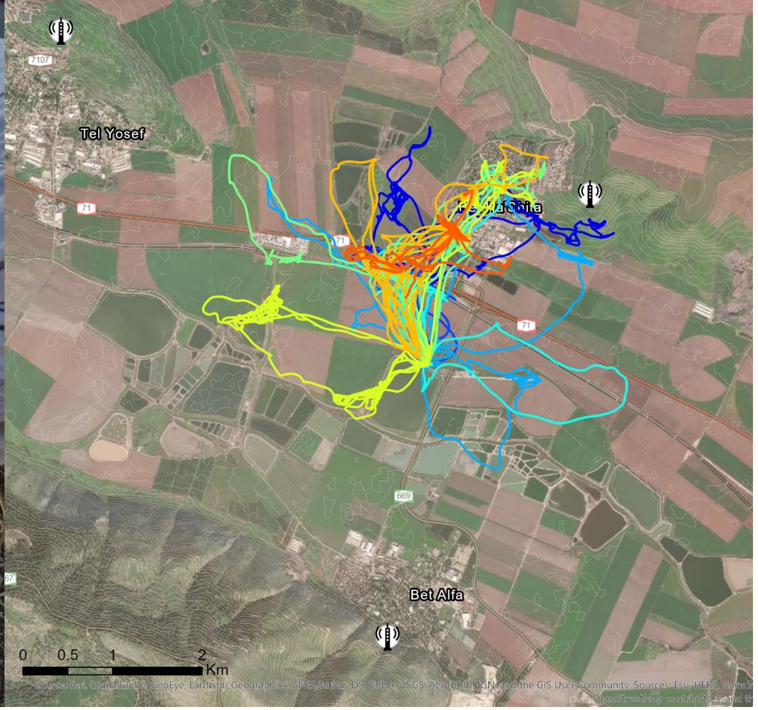

Tel Yosef
Bet Alfa
7107
71
669
0
0.5
1
2
Km



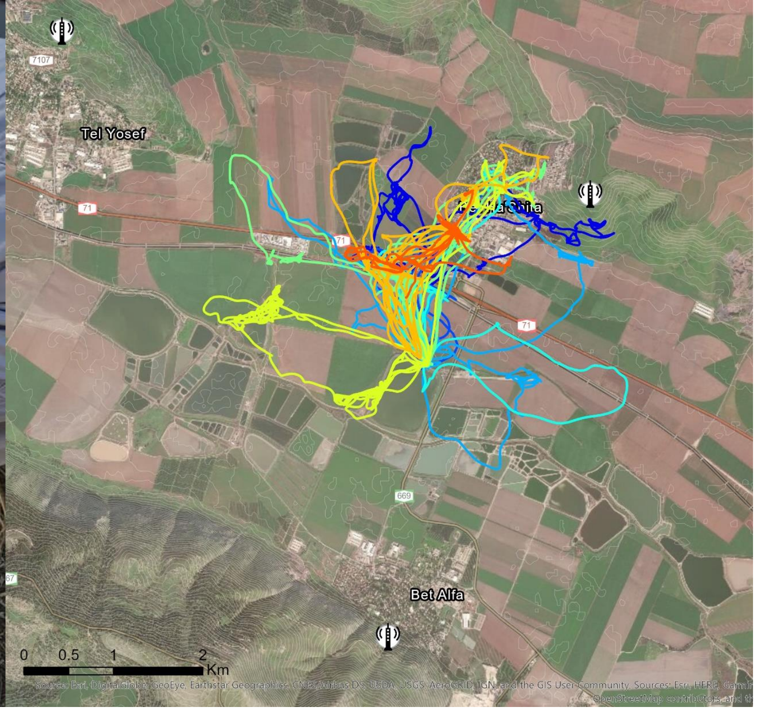
$$\hat{x} = \arg\min \|M(a, x) - b\|_2$$



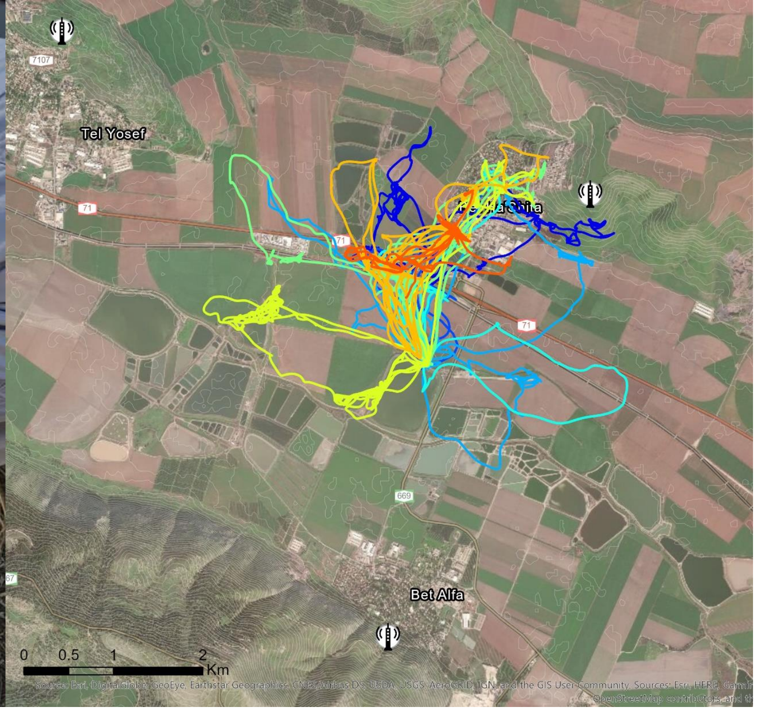
$$\hat{x} = \arg\min \|Ax - b\|_2$$



$$\hat{x} = \arg\min \|M(a, x) - b\|_2$$

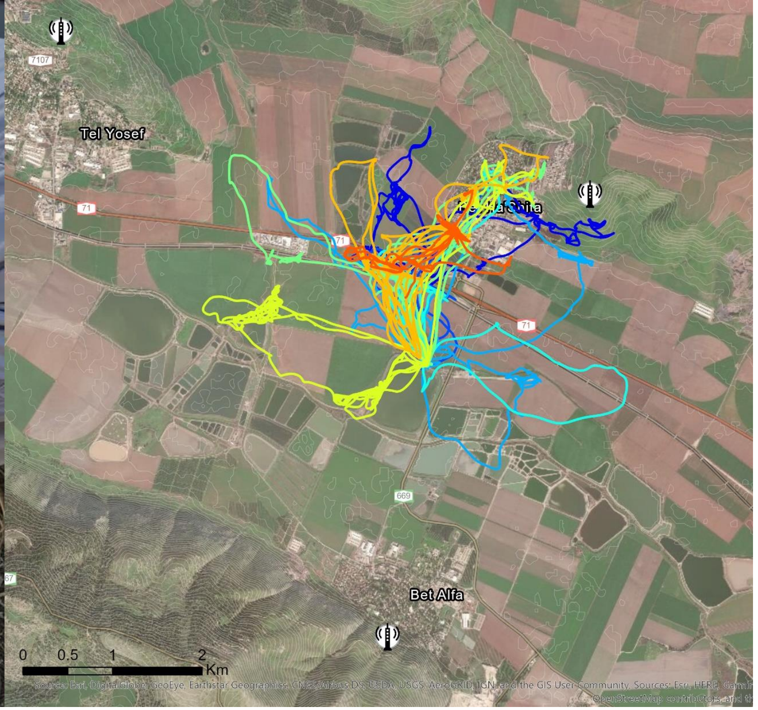


$$\hat{x} = \arg\min \| M(a, x) - b \|_2$$



Why?

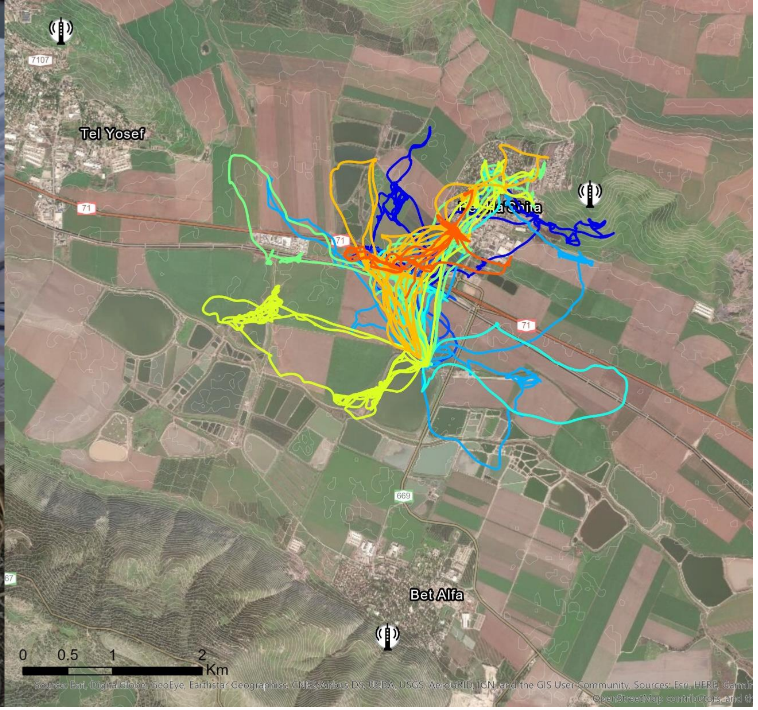
$$\hat{x} = \arg\min \| M(a, x) - b \|_2$$



$$\hat{x} = \arg\min \|M(a, x) - b\|_2$$

Why?

Some obserservations (sensor data) b_i (or even some elements of a) are bad



$$\hat{x} = \arg\min \|M(a,x) - b\|_2$$

Why?

Some obeservations (sensor data) b_i (or even some elements of a) are bad

(slightly) over determined, so hopefully can remove bad observations

Estimation 101

- An observation vector b
- Generated by a parameterized system; some of the parameters, denoted x, are not known
- The system translates x to the observed quantities through a (possibly nonlinear) function $M(a, x) = M(x), M: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- Inaccurate observation and/or imperfect model → $b = M(x) + \epsilon$ where ϵ is a noise vector; simplest possible form (additive)
- Our task is to estimate x from b

Estimation 102: Least Squares

- $\epsilon \sim N(0, I)$ implies $\hat{x} = \arg\min \|M(x) - b\|_2$ is optimal (max likelihood)
- $\epsilon \sim N(0, C)$ implies $\hat{x} = \arg\min \|W(M(x) - b)\|_2$ where $W^T W = C^{-1}$
- (Basically same thing)
- Distributions with heavier tails lead to other norms, like $\|\cdot\|_1$
- Least squares minimization is very common in practice

Robust Estimation

- The assumption that we know the distribution of **all** the elements of ϵ is often **too simplistic**

- What to assume? Many possible answers
  - A mixture of two distributions (e.g., small & large Gaussian errors)
  - Most ϵ_i's from a Gaussian distribution, the rest are worst case (statistically or computationally)
  - Many outliers or just a few
  - …

Approaches to Robust Estimation

1. Identify outliers and remove them (a combinatorial problem)
2. Limit the influence (leverage) of outliers on the solution
3. Combinations

Norms and Other Penalties

- Given a *hypothesis* x, form the residual $r = M(x) - b$

- $\|\cdot\|_2^2 = \sum_i r_i^2$ all i, breakdown of 0 (1 bad outlier is ruinous)
- $\|\cdot\|_1 = \sum_i |r_i|$ all i, breakdown of 0

Norms and Other Penalties

- Given a *hypothesis* x, form the residual $r = M(x) - b$

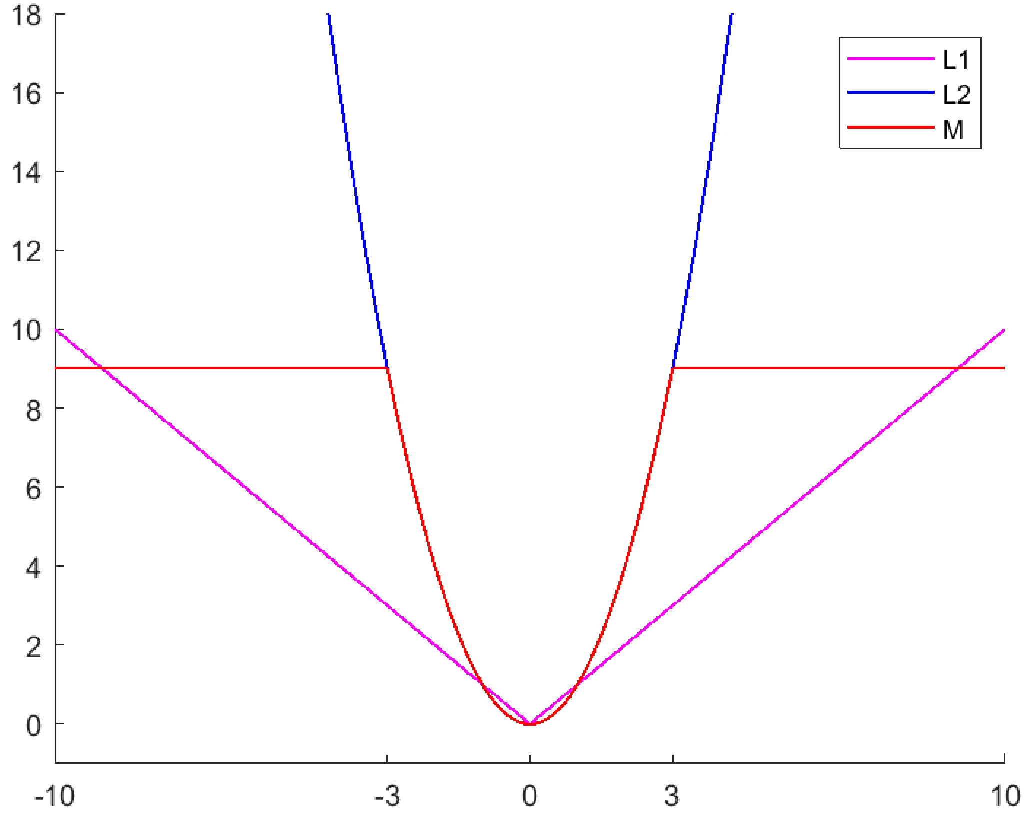
- $\|\cdot\|_2^2 = \sum_i r_i^2$ all i, breakdown of 0 (1 bad outlier is ruinous)
- $\|\cdot\|_1 = \sum_i |r_i|$ all i, breakdown of 0

- $\sum_i \xi(r_i)$ ξ bounded, zero at zero (bounded influence)

- $\sum_{i \in S} r_i^2$ S may depend on r, e.g., smallest $|r_i|$'s; LTS
- $\text{median}(|r_i|)$ LMS; similar, but less efficient (statistically)

M-Measures and M-Estimators

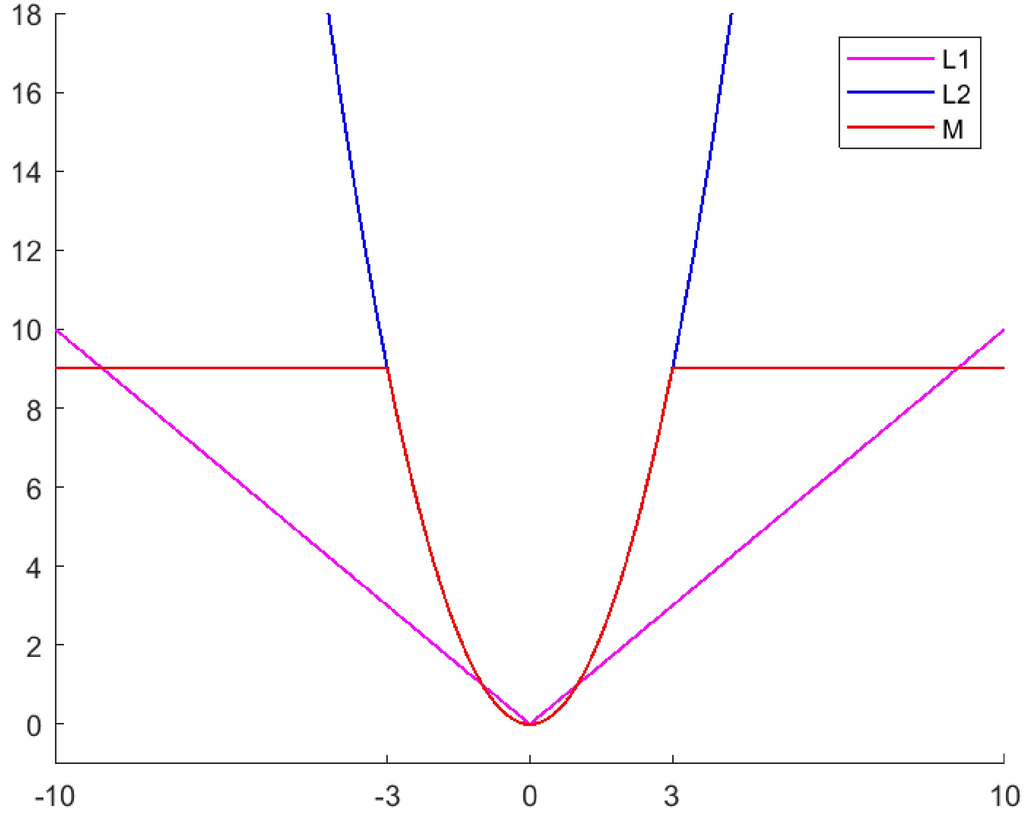
- Assume that for inliers $\epsilon \sim N(0, I)$
- → if x is exact, then , $\Pr(|r_i| > 3) \approx 0.01$
- So the meaning of $|r_i| = 20$ or $|r_i| = 30$ is the same: an outlier
- $\sum_i \xi(r_i)$, ξ bounded, zero at zero



M-Measures and M-Estimators

- Assume that for inliers $\epsilon \sim N(0, I)$
- → if x is exact, then , $\Pr(|r_i| > 3) \approx 0.01$
- So the meaning of $|r_i| = 20$ or $|r_i| = 30$ is the same: an outlier
- $\sum_i \xi(r_i)$, ξ bounded, zero at zero

- A miminization problem, but clearly nonconvex
- And (for this ξ) no incentive to reduce number of outliers



Now You Known What Robust Estimation Is

Beware of Nonexperts

- Like me; I am not a statistician
- But experts also sometimes have weird views, as when they promote LTS disregarding computational efficiency

Are These Problems Hard?
Focusing for now on Linear Problems

- $M(x) = Ax$
- $A \in \mathbb{R}^{m \times n}$

Can We Find the Outliers?

Can We Find the Outliers? Probably Not

- That is, not if they are hiding

Can We Find the Outliers? Probably Not

- That is, not if they are hiding
- *Hardness of Solving Sparse Overdetermined Linear Systems: A 3-Query PCP over Integers*, Guruswami & Raghavendra, ACM Trans. on Computation Theory, 2009

- $S(Ax = b)$, S only selects rows; 3 nonzeros per row; real or integer
- Is there an x that satisfies a $1 - \delta$ fraction of the equations?
- Does every x, possibly real, violate at least $1 - \epsilon$ of the equations?

- NP hard to distinguish for any $\delta, \epsilon > 0$

Approximate Solutions Easier? Probably Not

- *NP-Hardness of Approximately Solving Linear Equations over the Reals*, Khot and Moskovich, SIAM J. on Computing 2013

- $S(Ax \approx 0)$, S only selects rows; only 3 bounded nonzeros per row
- Is there a nontrivial x that satisfies a $1 - \delta$ fraction of the equations?
- Does every nontrivial x lead to residuals larger than $\sqrt{\delta}$ in a constant fraction of the equation?

- NP-hard to classify

Open Problems for Theoreticians

- $S(Ax \approx b)$
- Find outliers when A is a Laplacian
- Even the case of a bipartite (weighted) Laplacian is interesting

- Probably also hard, but the structure probably precludes many useful reductions

Are These Problems Hard in Practice?

- Not necessarily
- ***Ra****ndom* ***sa****mple* ***c****onsensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Fischler & Bolles, CACM 1981 (~29K citations)

- $S_j(Ax_j = b)$ select many random samples S_j of size n (exactly determined)
- (also works in the nonlinear case)
- Consensus set of x_j is rows for which $|Ax_j - b|$ is small
- Largest consensus set → inliers, solve using inliers using least squares

RANSAC Details

- Admission threshold (x_j are very approximate)
- How many subsets to test
- Obviously, lots of variants

- Original motivation came from highly overdetermined problems in low dimensions (e.g., 6); can afford to throw away lots of suspects

Why Does it Work: Theory and Practice

- Suppose that an adversary gets to choose the indexes of the outliers
- But not their value; they will come from some oblivious process

- Given a reasonably accurate **hypothesis** x_j, an outlier b_i will be far from $\left(Ax_j\right)_i$ and hence easy to detect

Adaptations to Location Estimation

- ToA equation with an imperfect clock

$$t_{ir} = \tau_i + {}^{1}/_{c}\,\|\rho_r - \ell_i\|_2 + o_r + \epsilon_{ir}$$

- ToA for a beacon at a known location

$$t_{br} = \tau_b + {}^{1}/_{c}\,\|\rho_r - \ell_b\|_2 + o_r + \epsilon_{br}$$

- We use difference equations for outlier classification

$$(t_{ir} - t_{br}) = (\tau_i - \tau_b) + {}^{1}/_{c}\,\|\rho_r - \ell_i\|_2 - {}^{1}/_{c}\,\|\rho_r - \ell_b\|_2 + (\epsilon_{ir} - \epsilon_{br})$$

- Three (nonlinear) equations in 3 unknowns → 0, 1, 2 *analytical* solutions

Overall Setup

- For every tag transmission, we have many admissible beacon tx's
- Each beacon tx generates a set of difference equations
- Each triplet generates 0, 1, or 2 *hypotheses*
- The aim is to first find a good hypothesis

- We can generate random triplets, but in practice we rank them by SNR (related to standard deviation of ϵ_{ir})

Approach 1: a la RANSAC

- For each triplet, classify difference equations from the same beacon tx as inliers or outliers by substituting a hypothesis

$$(\epsilon_{ir} - \epsilon_{br}) = (t_{ir} - t_{br}) - (\tau_i - \tau_b) + {}^1/_c \, \|\rho_r - \ell_i\|_2 - {}^1/_c \, \|\rho_r - \ell_b\|_2$$

- Rank hypotheses by *consensus set*, then max residual & distance from a previous location
- Pick the best hypothesis, filter inliers for the same beacon tx, solve
- Optional: classify and add inliers from other beacon tx's, solve again

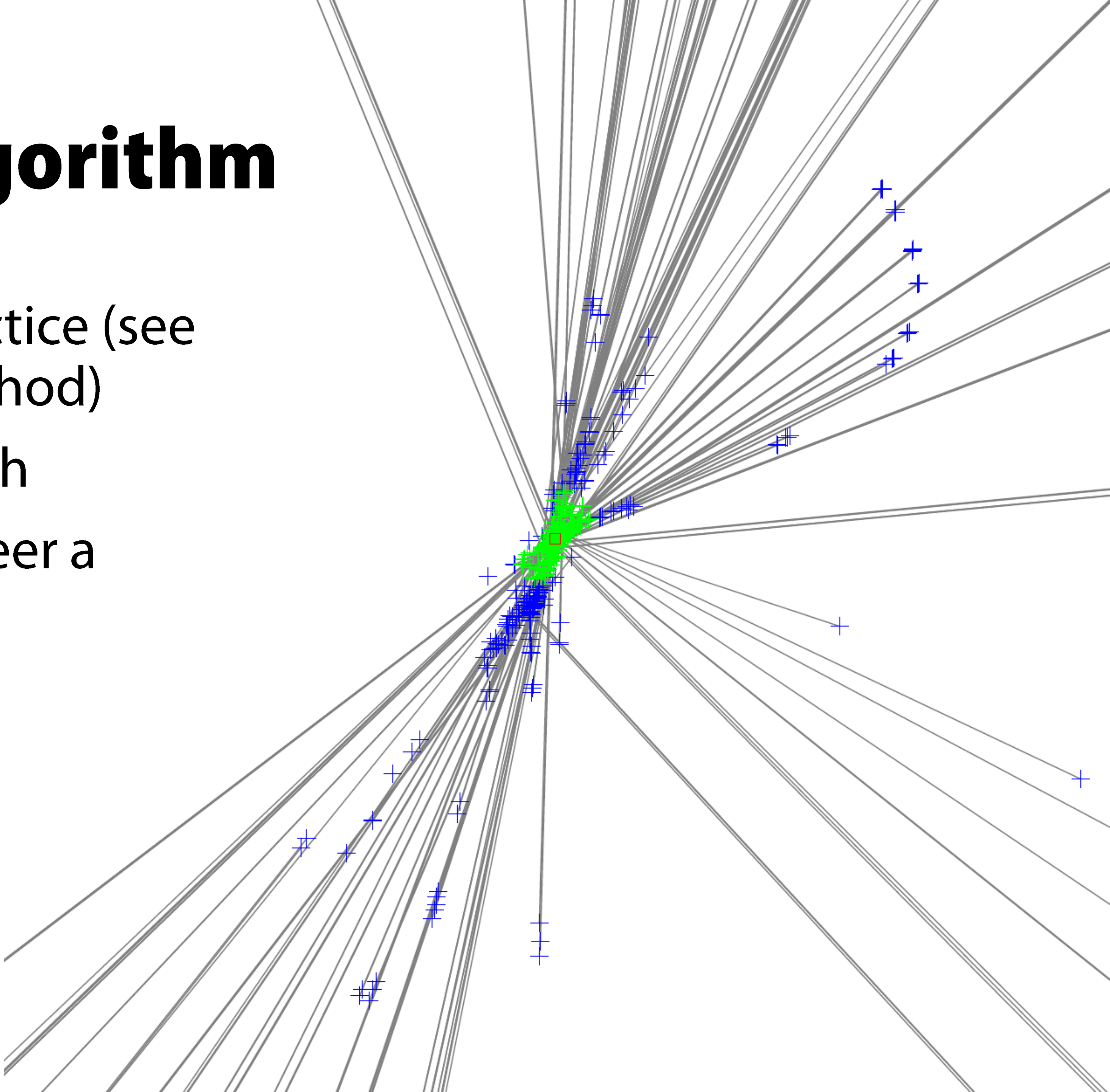
+

Approach 2: Hypotheses Clustering

- Collect a large set of hypothetical geometric solutions (from many beacon tx's, many triplets)
- For inliers, at least one of up to 2 is near ℓ_i, so run a clustering algorithm (we use HDBSCAN) to find the largest cluster, take its medians as a hypothesis
- Use that location to filter and solve inlier ToA constraints

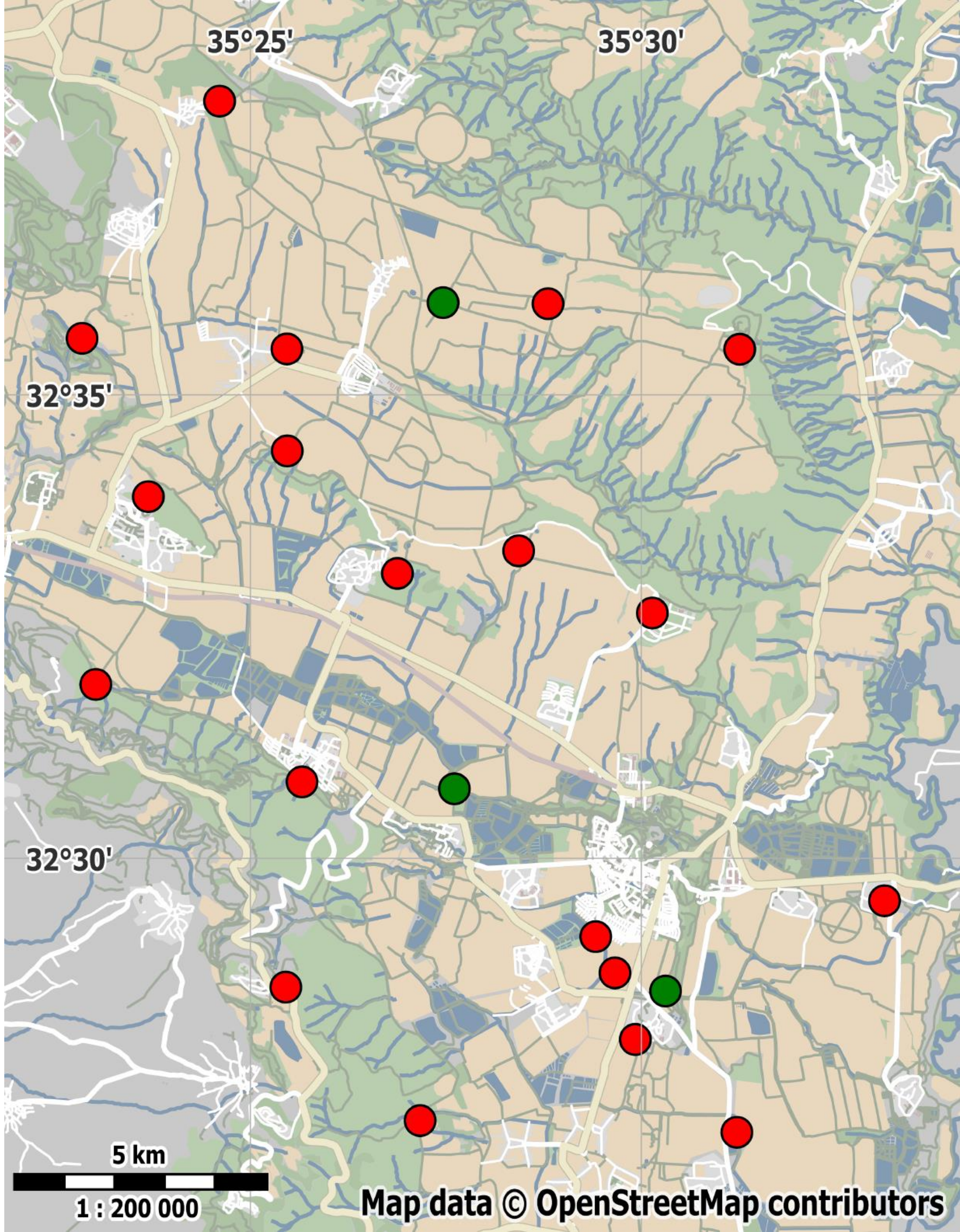
The Clustering Algorithm

- Did not work so well in practice (see later for the evaluation method)
- We did not pursue this much
- Might be possible to engineer a good algorithm



Real-World Results from an ATLAS System

- Joint work with **Eitam Arnon**, Shlomo Cain, Assaf Uzan, Ran Nathan, and Orr Spiegel

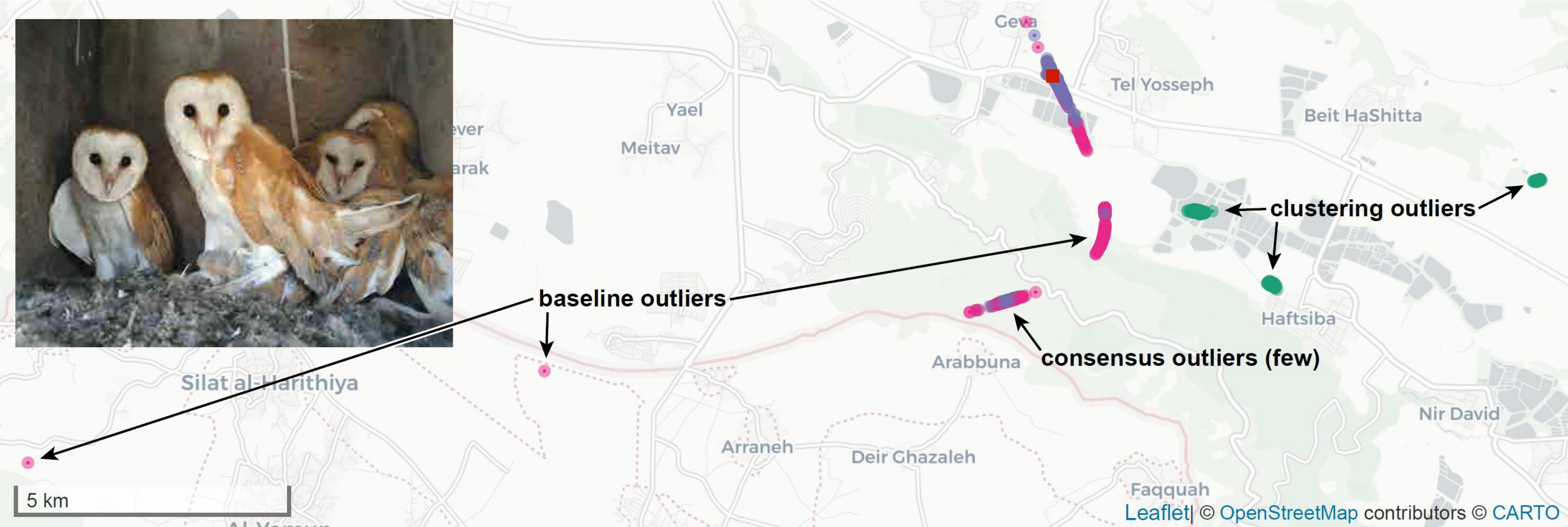


An Easy Case

- Tracks with few outliers (a flying owl) are almost the same with or without outlier detection and rejection (orange and green, respectively)

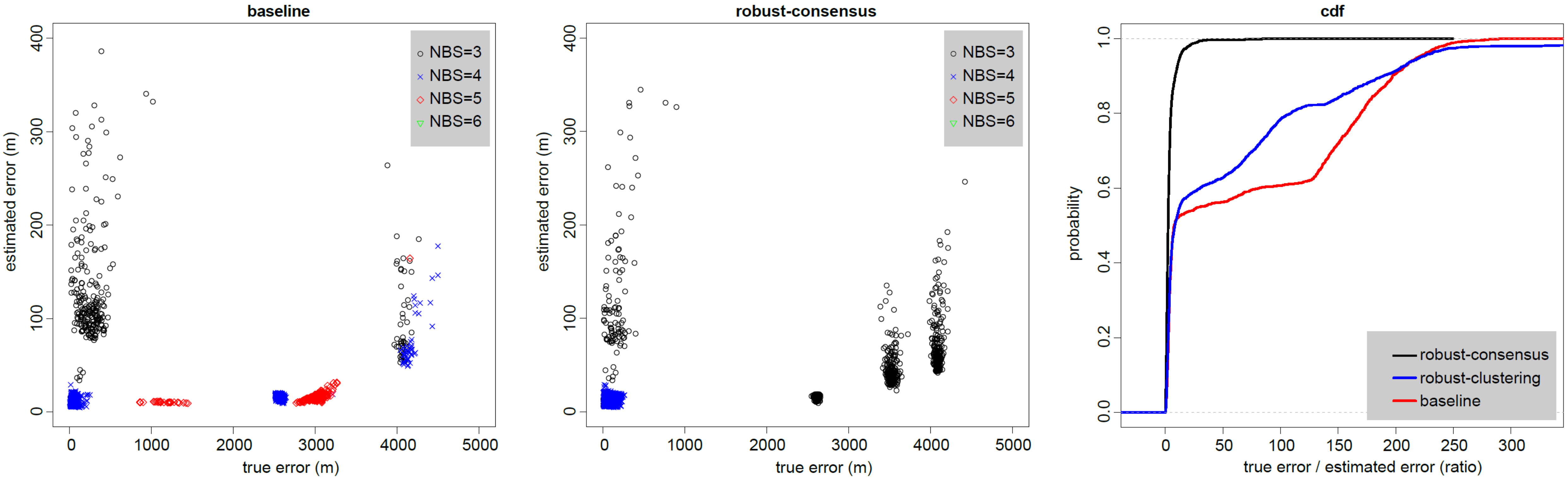
A Harder Case

- An owl in a next box generates a lots of outliers; many fewer with the robust algorithm than without outlier detection and rejection



Statistics

- The robust algorithm produces outliers almost exclusively with exactly-determined problems; the old algorithm also with overdetermined



Summary

- Robust estimation is a combinatorial problem in statistics
- Combinatorial solutions proposed by statisticians appear to be computationally inefficient
- Minimization solutions by statisticians are nonconvex and probably also hard
- Natural simplifications are hard even to approximate
- In practice the problem is usually easy in low dimensions
- An example from location estimation; two kinds of heuristics
- And the next steps are…

Next Challenges

- (Complexity of robust overdetermined Laplacians)

- Practical robust estimation in high dimensions
  - Identify inliers in small seperable sub-problems, stitch together
  - Might allow a RANSAC-like strategy to work in high dimensions
- Why solve a large unified problem if we can split it?
  - Better statistical performance; but only if we removed the outliers!
- Examples of such problems:
  - Overdetermined Laplacians (clock synchronization problems; bipartite)
  - Kalman smoothing

That’s It