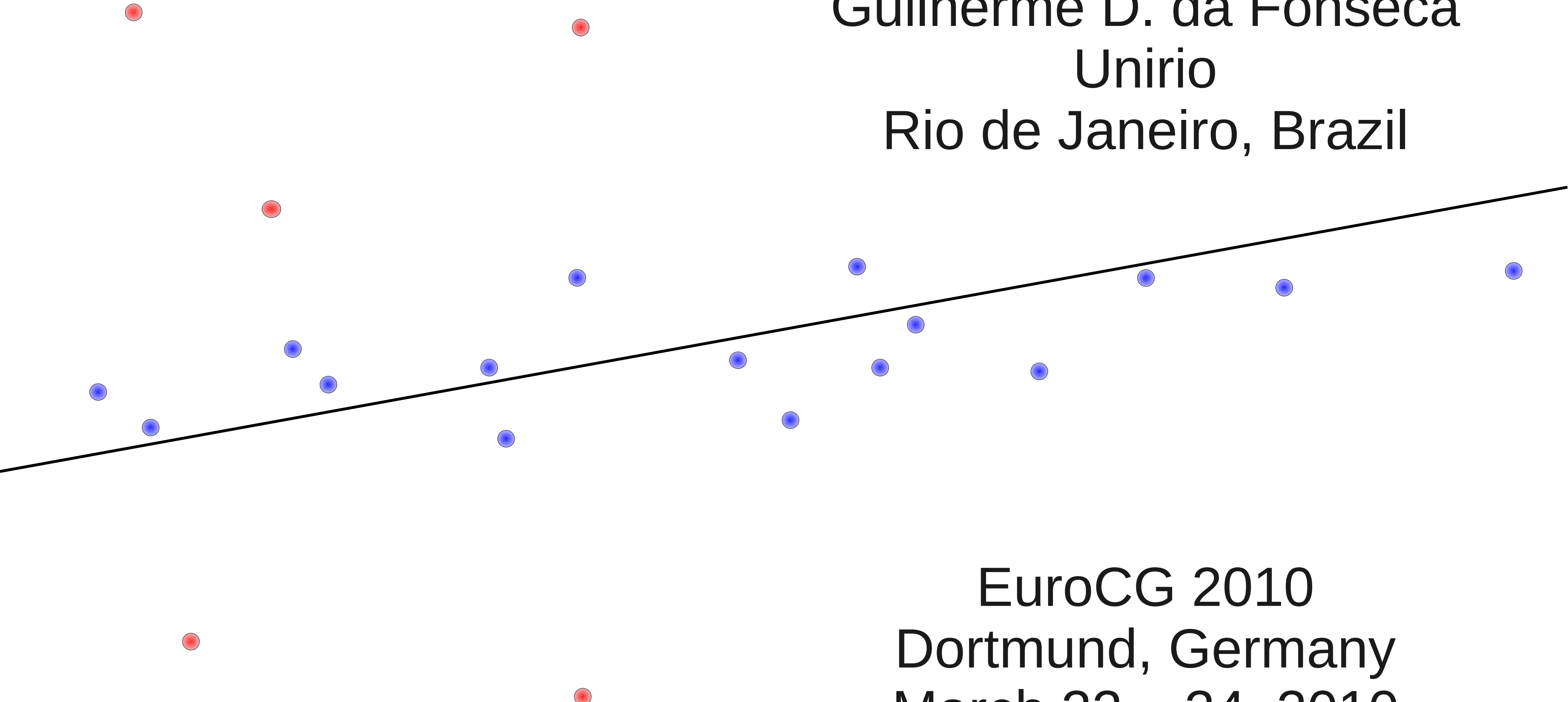
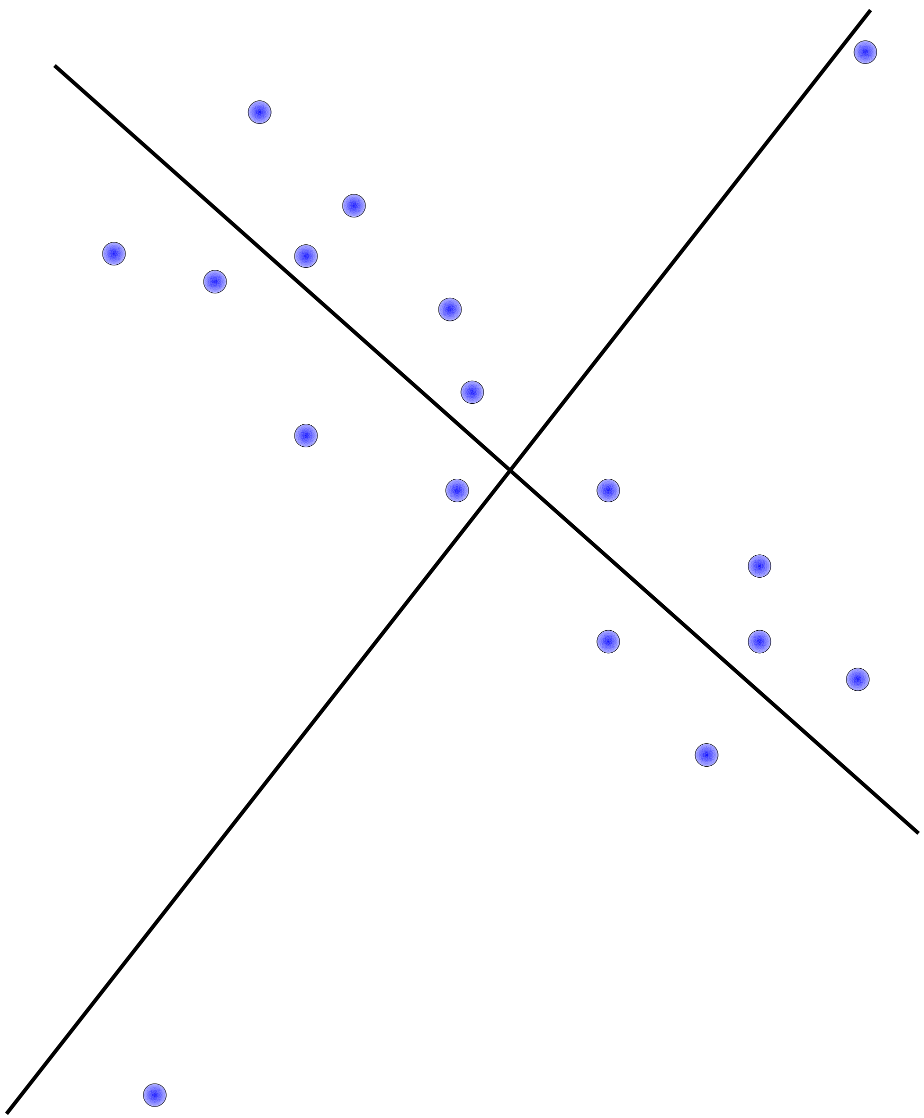
Fitting Flats to Points with Outliers

Guilherme D. da Fonseca
Unirio
Rio de Janeiro, Brazil



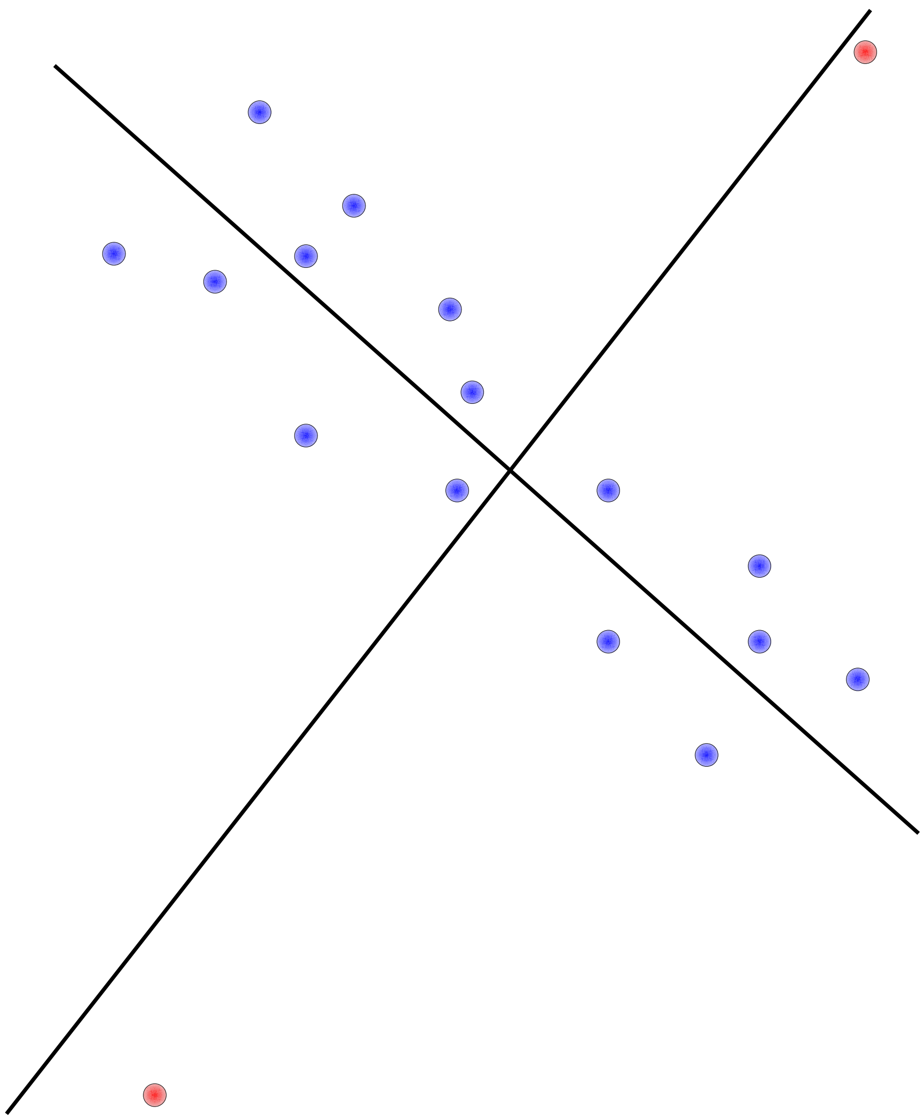
EuroCG 2010
Dortmund, Germany
March 22 – 24, 2010

Shape Fitting



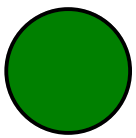
- Input: set *P* of *n* points in *d*-dimensional space.
- Place a given shape minimizing the distance between the shape and the farthest point.
- Dimension *d* is constant.
- Very sensitive to outliers!

Shape Fitting with Outliers

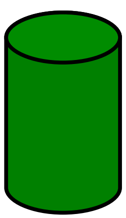


- We are also given a number m of inliers.
- Minimize the m-th smallest distance.
- The remaining $n - m$ points are called outliers.
- We focus on the approximate version, where the distance is (1+ε)-approximated.

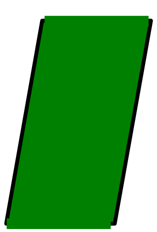
Fitting *k*-Flats



- $k = 0$ [Har-Peled and Mazumdar, 2005]
  - Smallest ball enclosing *m* points.
  - Linear time approximations.



- $k = 1$
  - Smallest infinite cylinder enclosing *m* points.
  - 3-SUM hard to approximate in the plane.



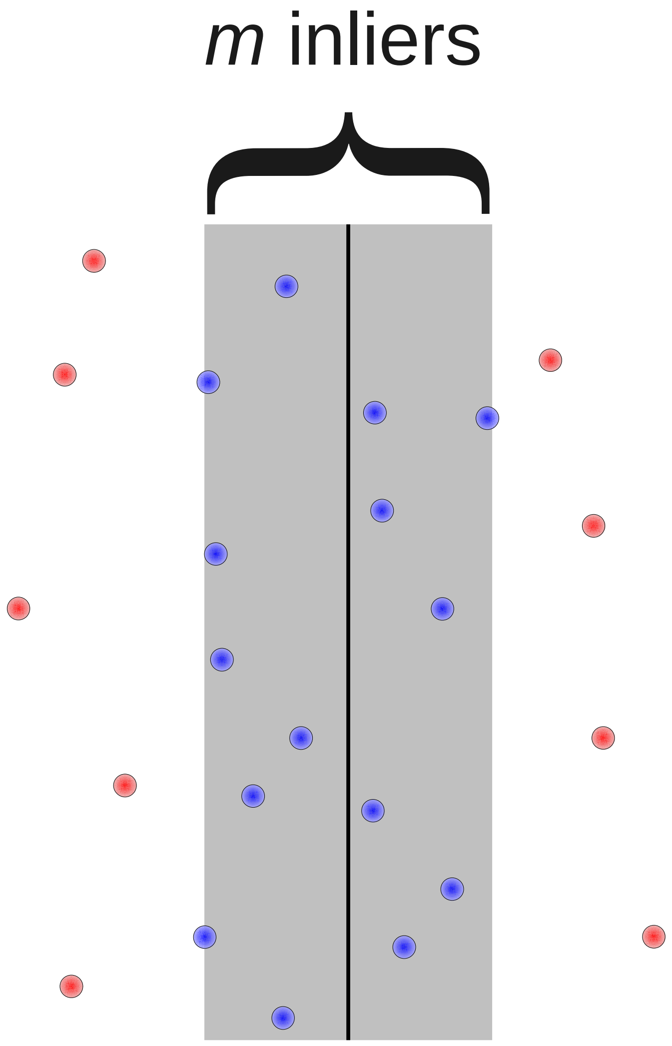
- $k = d - 1$ [Erickson, Har-Peled, Mount, 2006]
  - Smallest slab enclosing *m* points.
  - $\Omega((n - m)^{d-1} + (n / m)^d)$ and $\tilde{O}(n^d / m)$ bounds.

k-Flats for Arbitrary k

- Lower bound easily generalizes to
  $\Omega((n - m)^k + (n / m)^{k+1})$.
- There is a coreset with $O((n - m) / \varepsilon^{(d-1)/2})$ points. [Agarwal, Har-Peled, and Yu, 2008]
  - Useful when there are few outliers.

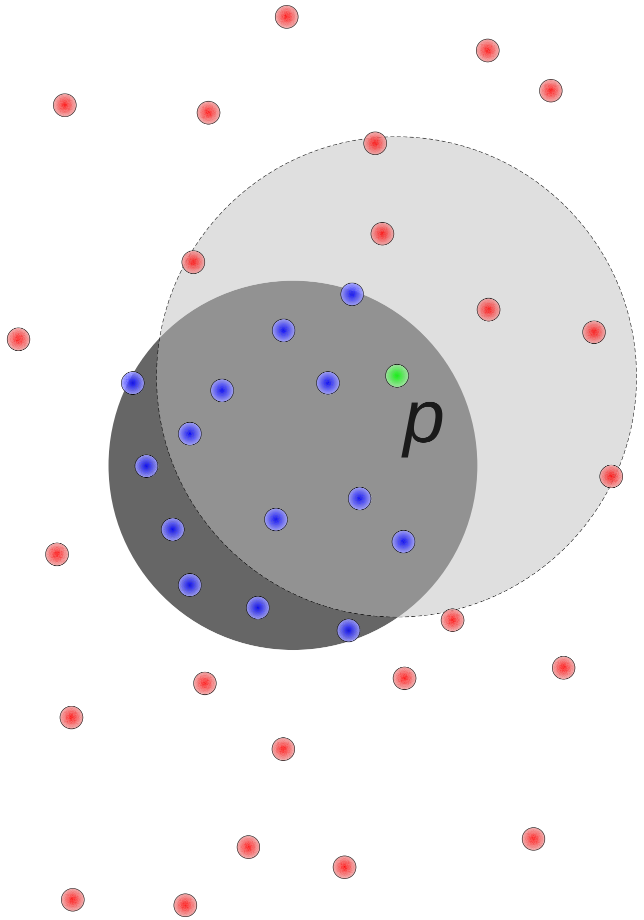
- Our focus: m is a constant fraction of n.
  - Lower bound becomes $\Omega(n^k)$.
  - Our Monte-Carlo upper bound is $O(n^{k+1})$.
  - For some data sets, the upper bound is $O(n)$.

Finding an Inlier



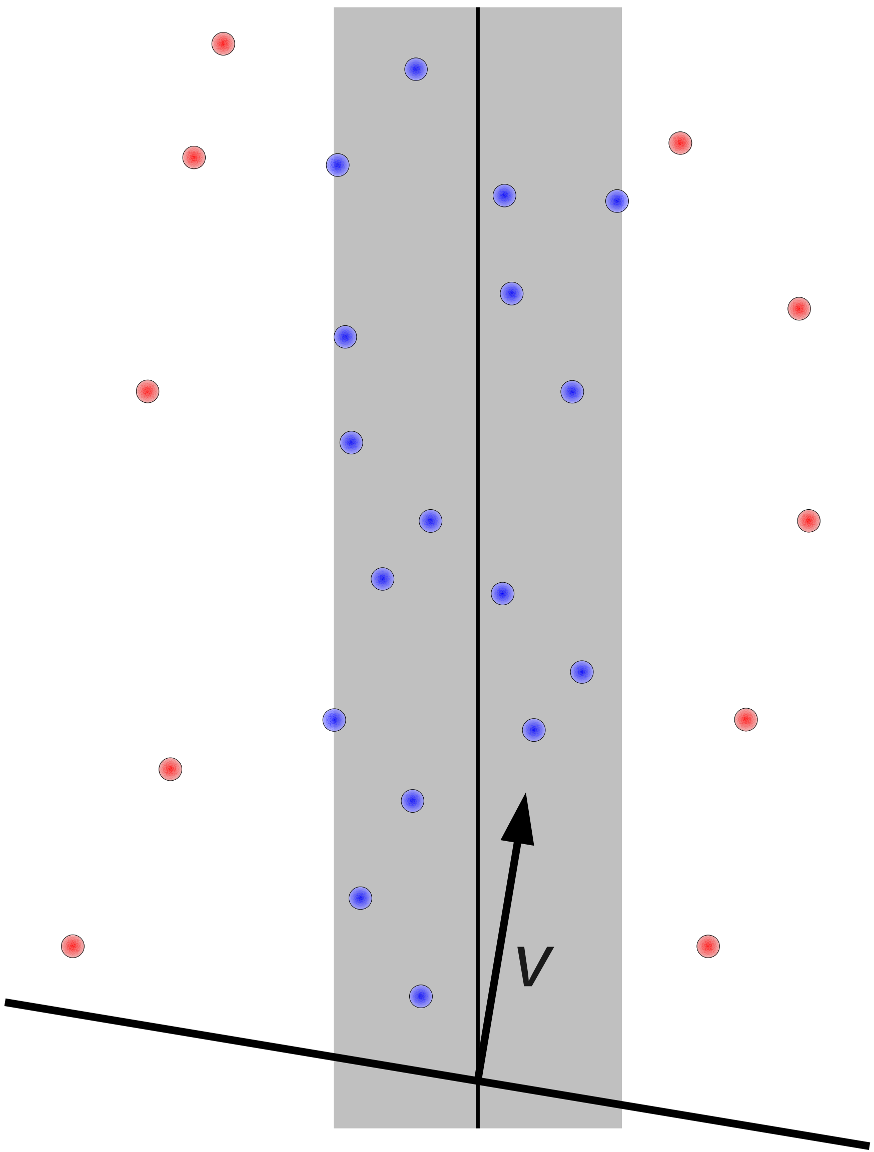
- More accurately: finding a set that contains an inlier.
- There are *m* inliers out of *n* data points.
- Monte Carlo: Random sample of *n* / *m* points contains an inlier with constant probability.
- Deterministic: Use all *n* points.

Base case: $k = 0$



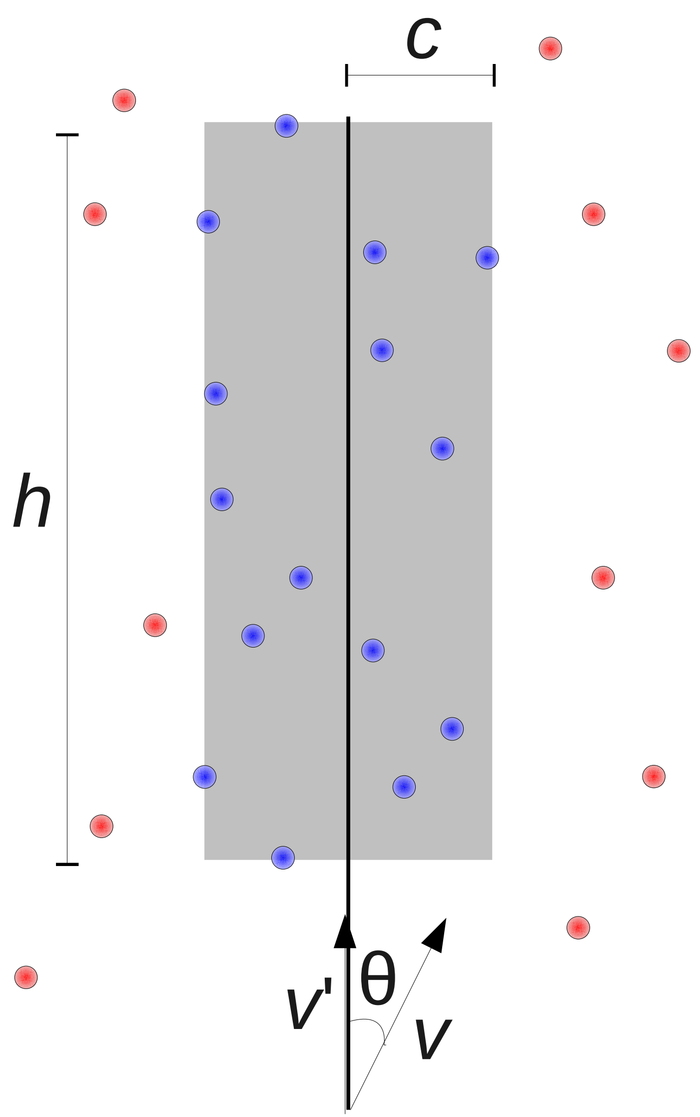
- We want to approximate the smallest ball enclosing m points given an inlier p.
- 2-approximation: select the m-th smallest distance to p.
- Takes $O(n)$ time.
- $(1+\varepsilon)$-approximation: use a grid around p.
- Takes $O(n + m / \varepsilon^d)$ time, but improvements are possible.

Reducing the Dimension



- Find a vector *v* approximately parallel to the optimal flat.
- Project the points onto a hyperplane perpendicular to *v*.
- Solve the problem recursively in lower dimension.
- We reduce dimensions (d,k) to (d–1, k–1).
- Base case: k = 0.

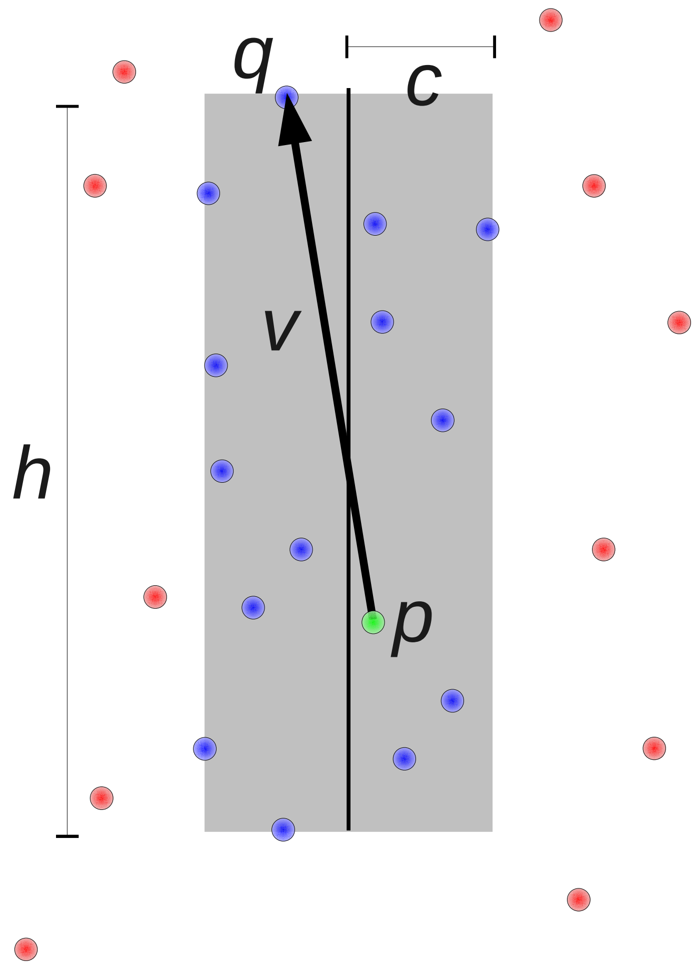
Approximately Parallel?



- "Find a vector *v* approximately parallel to the optimal flat."
- *c*: optimal cost.
- *v*': projection of *v* onto the optimal flat.
- *h*: directional width of the inliers in direction *v*'.
- θ: angle between *v* and *v*'.
- (1+ε)-approximation if

$$\theta \leq \varepsilon\, c / h.$$

Finding Such Vector v



- Lemma: For every inlier p there is an inlier q such that $v = q - p$ has

$$\theta \le 4\, c \,/\, h.$$

- To reduce the constant, use a grid of vectors near v.
- Given p, we can find a set of $O(n \,/\, \varepsilon^{d-k})$ vectors that contains a vector v with $\theta \le \varepsilon c/h$.
- Project and recurse for each vector in the set.

Running Time

- After finding an inlier, we take time

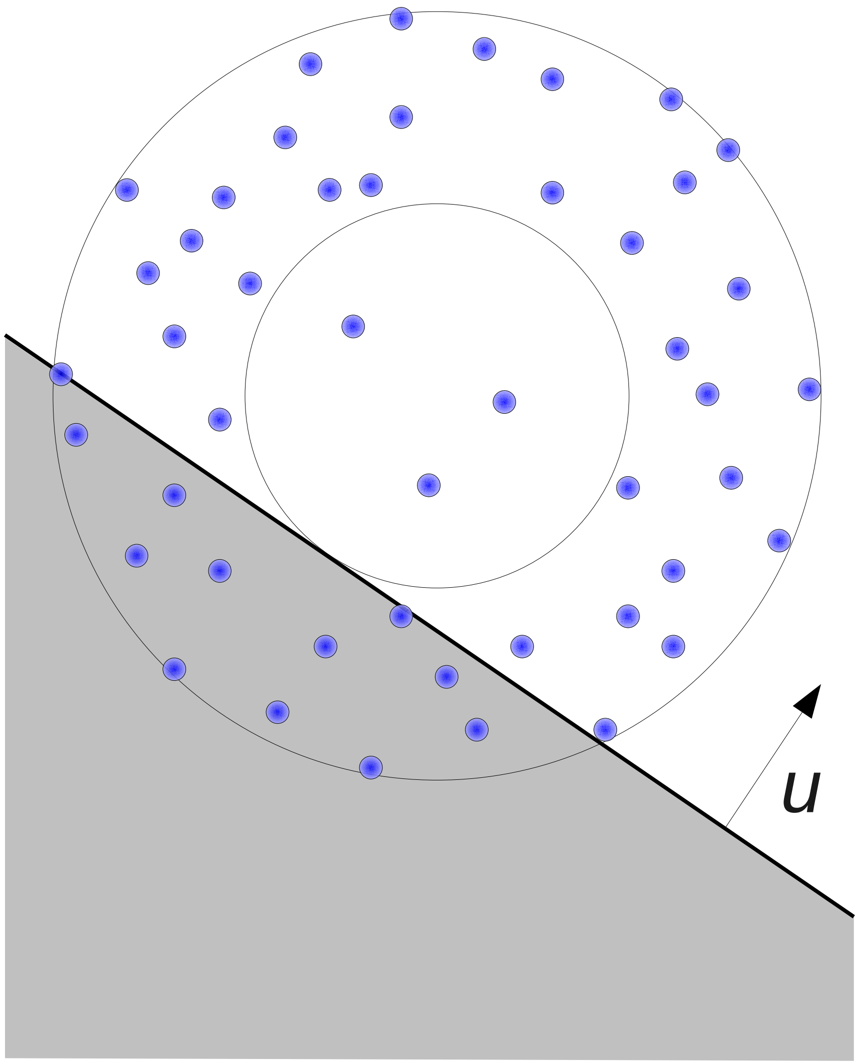
$$t_{k,d} = \begin{cases} O(n/\varepsilon^{d-k})t_{k-1,d-1} & \text{if } k > 0 \\ O(n + m/\varepsilon^{d}) & \text{if } k = 0. \end{cases}$$

- Which solves to

$$t_{k,d} = O\left(\frac{n^{k+1}}{\varepsilon^{k(d-k)}} + \frac{n^k m}{\varepsilon^{(k+1)(d-k)}}\right) = O_\varepsilon(n^{k+1})$$

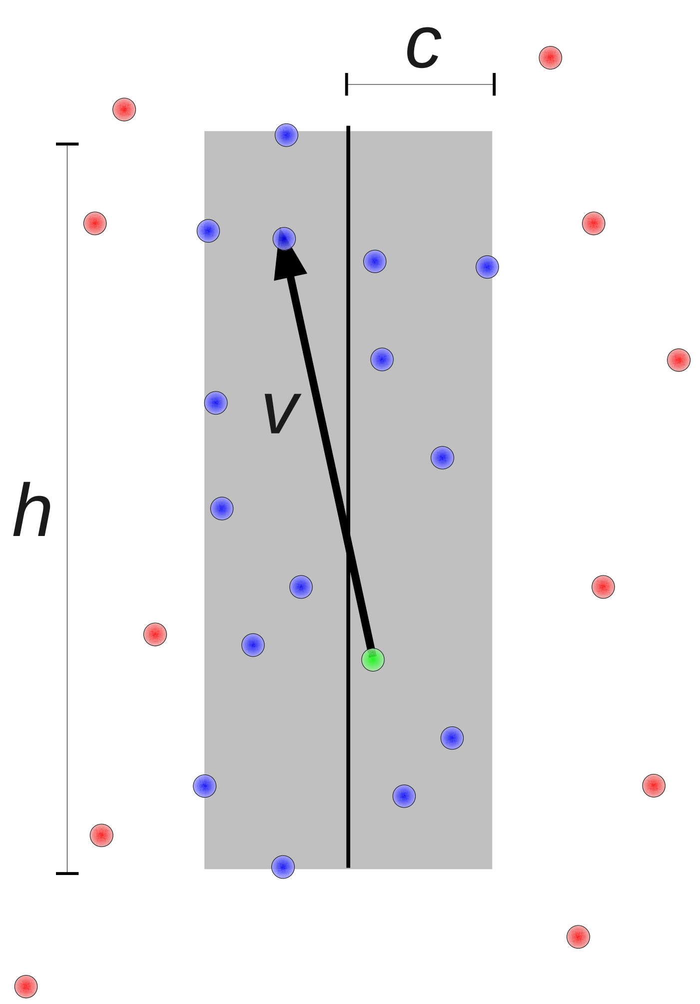
- The total time is $n\ t_{k,d} / m = O_\varepsilon(n^{k+2} / m)$ Monte Carlo and $n\ t_{k,d} = O_\varepsilon(n^{k+2})$ deterministic.

Outer-Dense



- A halfspace with normal vector *u* is *deep* if it contains 1/4 of the width in direction *u*.
- A set of points is *outer-dense* if every deep halfspace contains a constant fraction of the points.
- Points uniformly distributed in a convex region or on its boundary are outer-dense w.h.p.

Outer-Dense Inliers



- Lemma: If the set of inliers is outer-dense, then with constant probability a pair of inliers *p,q* defines a vector $v = q - p$ such that

$$\theta \leq 4\, c \,/\, h.$$

- We get a Monte Carlo algorithm with $O_\varepsilon(n^{k+2} / m^{k+1})$ running time for outer-dense sets of inliers.
- Linear for $m = \Omega(n)$.

Summary

- The running time of our Monte Carlo algorithm is

$$O\left(\frac{n^{k+2}}{m\varepsilon^{k(d-k)}} + \frac{n^{k+1}}{\varepsilon^{(k+1)(d-k)}}\right) = O_\varepsilon\left(\frac{n^{k+2}}{m}\right)$$

  which is close to the lower bound of

$$\Omega((n-m)^k + (n/m)^{k+1})$$

  for a constant approximation, especially when $m=n/2$.

- When the set of inliers is outer-dense, the upper bound becomes $O_\varepsilon(n^{k+2} / m^{k+1})$.

Open Problems

- Even when $m = n / 2$, there is a $\Theta(n)$ gap between the lower bound and our upper bound (except for k=0).
- A related problem consists of approximating the unit cylinder centered on the origin that contains the most points.
  - Easy in the plane.
  - Is it 3-sum hard in higher dimensions? Near-linear algorithms at least in 3d?

Thank you!

Questions???