Optimization in Computer Vision

Quasiconvex Optimization Problems

- PART I: Fast algorithms
- PART II: Outliers

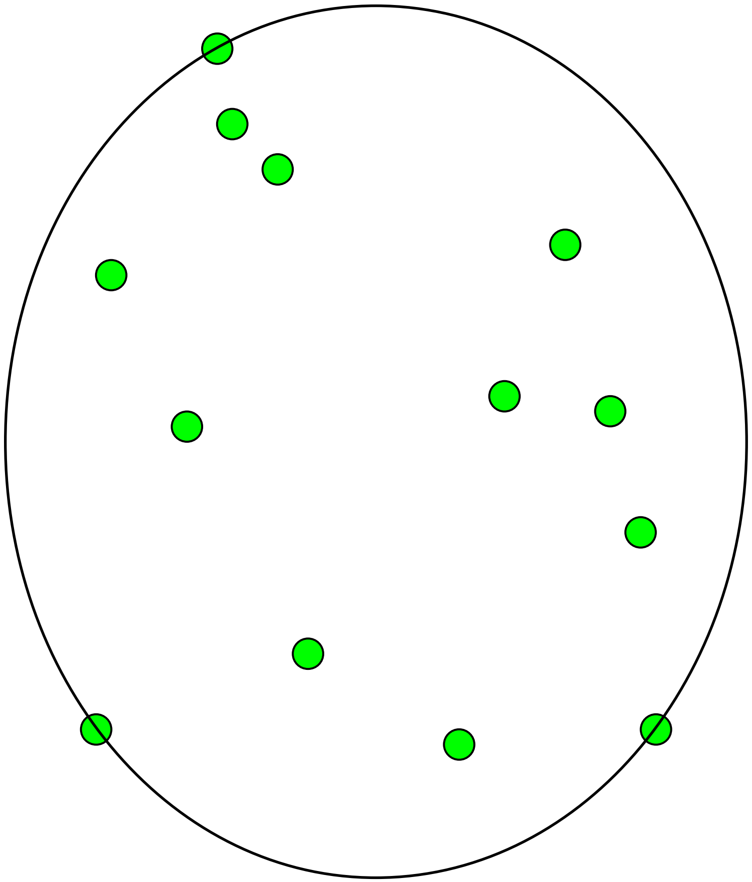
Fredrik KAHL
Lund University, Sweden
fredrik@maths.lth.se

Fast Algorithms

— based on ideas from Computational Geometry

Computational Geometry

Example: Find smallest enclosing circle.



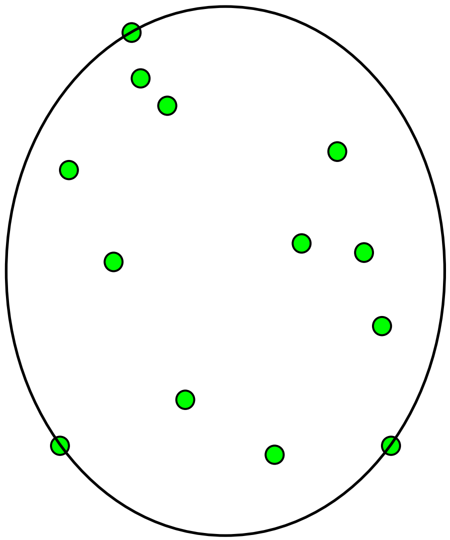
Smallest Enclosing Circle

Consider n points (x_i, y_i).
Find centre (x_0, y_0) and minimum radius r.

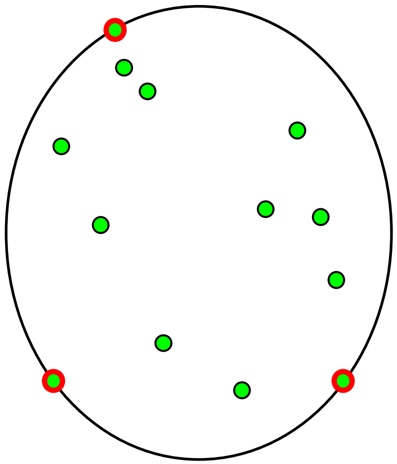
$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & ||(x_0 - x_i, y_0 - y_i)|| \leq r \quad i = 1, \dots, n \end{aligned}$$

SOCP!
What about time complexity?

Not so good...



Smallest Enclosing Circle



Notice that 3 points normally touch the circle.

Let's call the 3 supporting points a basis.

We will say a point x violates basis B if x is not enclosed by the circle corresponding to basis B.

Problem formulation:
- Find basis which is not violated by any other point.

Given $3 + 1 = 4$ points, we can compute an optimal basis in fixed time.

A Randomized Algorithm

H - set of points. B - some initial basis. Call: $LP(H, B)$.

```
function LP(H, B)
  n := |H \ B|; %cardinality of H \ B
  π := randperm(n); %random order
  i := 0;
  while i++ < n,
    h := element π(i) in H \ B;
    if h violates B then
       B := basis(B ∪ {h}); %update basis
       i := 0; %start all over
    end
  end
  return B;
```

Note: Randomization independent of input data.

Time complexity is $O(n)$!

Abstract Linear Programming

Consider optimization problem specified by pair (H, w).

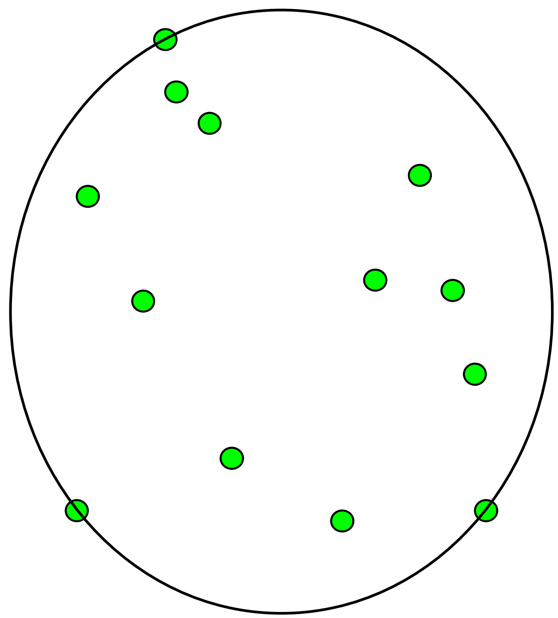
H - set of constraints
w - objective function, $w : 2^H \mapsto \mathbb{R} \cup \{-\infty\}$

For $G \subseteq H$, $w(G)$ means smallest value while satisfying constraints of G.

Examples:

- Smallest enclosing circle,
- Linear Programming,

...



Basis

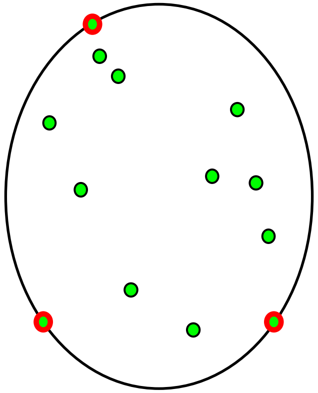
A basis is a set $B \subseteq H$ (with $w(B) > -\infty$) for which all proper subsets B' of B imply $w(B') < w(B)$.

A basis of G is a minimal subset B of G with $w(B) = w(G)$.

The combinatorial dimension is the maximum cardinality of any basis.

Goal: Compute basis B_H of H with $w(B_H) = w(H)$.

Examples:
- Smallest enclosing circle,
- Linear Programming.



Abstract Linear Programming

We assume the following three primitive operations:

Violation test: for a constraint h and a basis B, test whether h is violated by B.

Basis computation: for a constraint h and a basis B, compute basis of $B \cup \{h\}$.

Initial basis: An initial basis B_0.

Examples:

- Smallest enclosing circle,
- Linear Programming.

Abstract Linear Programming

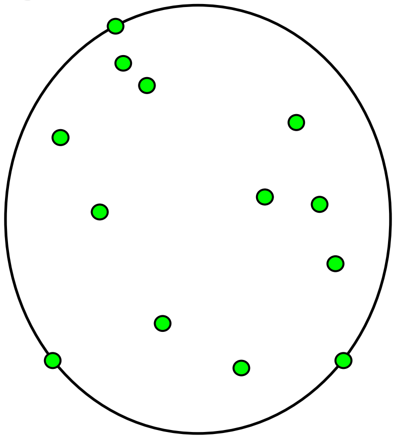
Called LP-type if the following axioms satisfied:

Axiom 1. (Monotonicity) For any $F \subseteq G \subseteq H$, we have

$$w(F) \leq w(G).$$

Axiom 2. (Locality) For any $F \subseteq G \subseteq H$ with $w(F) = w(G) > -\infty$ and any $h \in H$, we have

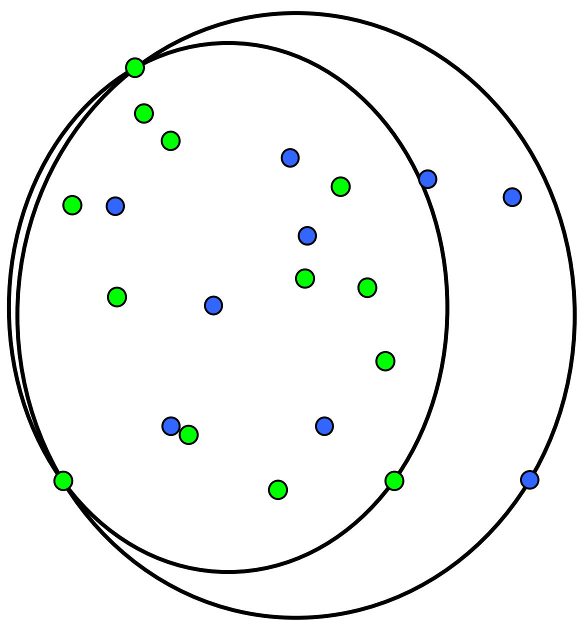
$$w(G) < w(G \cup \{h\}) \Rightarrow w(F) < w(F \cup \{h\}).$$



Example: Smallest Enclosing Circle

Axiom 1. (Monotonicity) For any $F \subseteq G \subseteq H$, we have

$$w(F) \leq w(G).$$



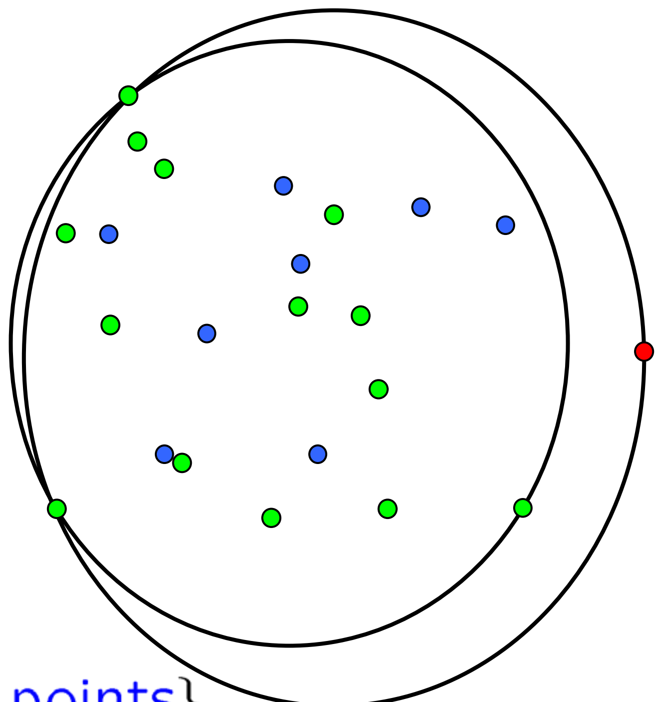
$F = \{$green points$\}$

$G = \{$green points$\} \cup \{$blue points$\}$

Example: Smallest Enclosing Circle

Axiom 2. (Locality) For any $F \subseteq G \subseteq H$ with $w(F) = w(G) > -\infty$ and any $h \in H$, we have

$$w(G) < w(G \cup \{h\}) \Rightarrow w(F) < w(F \cup \{h\}).$$



$F = \{$green points$\}$

$G = \{$green points$\} \cup \{$blue points$\}$

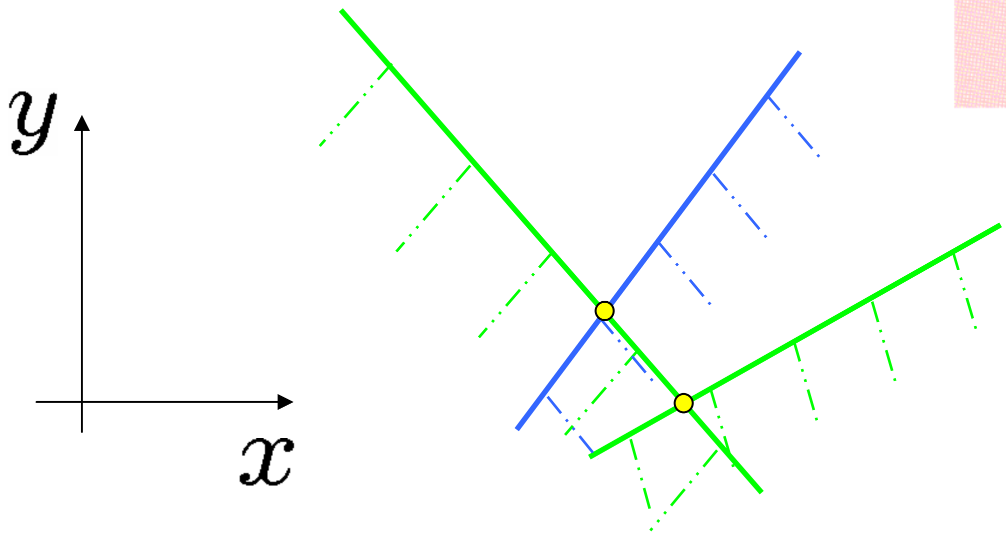
$h =$ red point

Example: Linear Programming

Axiom 1. (Monotonicity) For any $F \subseteq G \subseteq H$, we have

$$w(F) \leq w(G).$$

min	y
s.t.	$a_i x + b_i y \leq c_i$



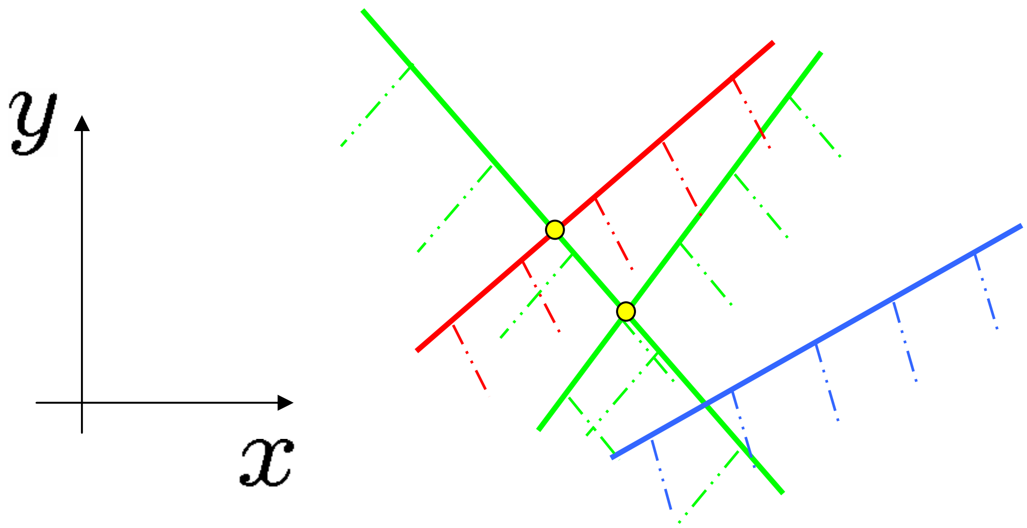
$F = \{$green half planes$\}$

$G = \{$green half planes$\} \cup \{$blue half planes$\}$

Example: Linear Programming

Axiom 2. (Locality) For any $F \subseteq G \subseteq H$ with $w(F) = w(G) > -\infty$ and any $h \in H$, we have

$$w(G) < w(G \cup \{h\}) \Rightarrow w(F) < w(F \cup \{h\}).$$



$F = \{$green half planes$\}$

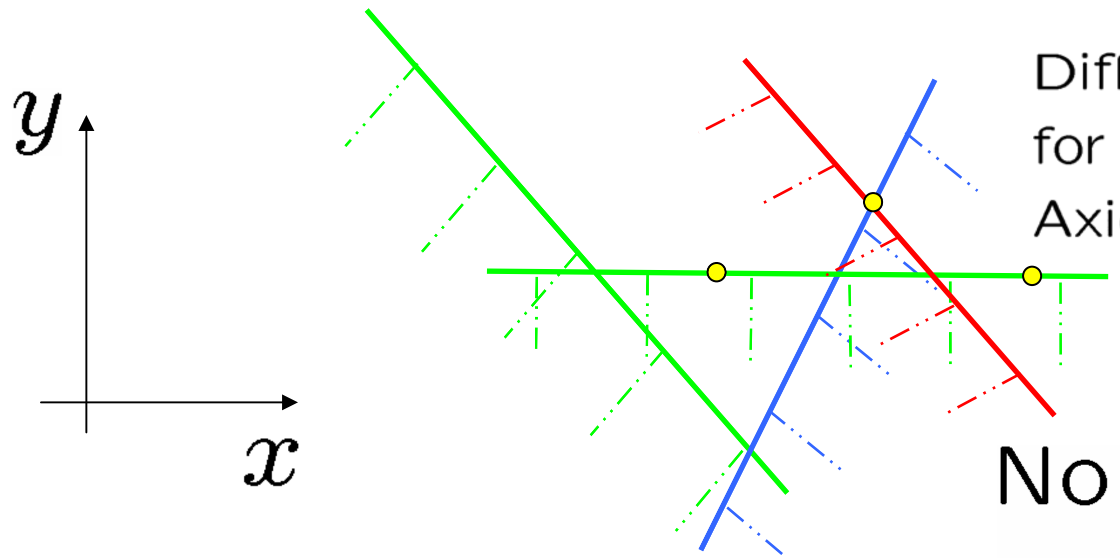
$G = \{$green half planes$\} \cup \{$blue half planes$\}$

$h =$ red half plane

Counter-Example: Linear Programming

Axiom 2. (Locality) For any $F \subseteq G \subseteq H$ with $w(F) = w(G) > -\infty$ and any $h \in H$, we have

$$w(G) < w(G \cup \{h\}) \Rightarrow w(F) < w(F \cup \{h\}).$$



Different minima for F and G! Axiom not satisfied.

$F = \{\text{green half planes}\}$

$G = \{\text{green half planes}\} \cup \{\text{blue half planes}\}$

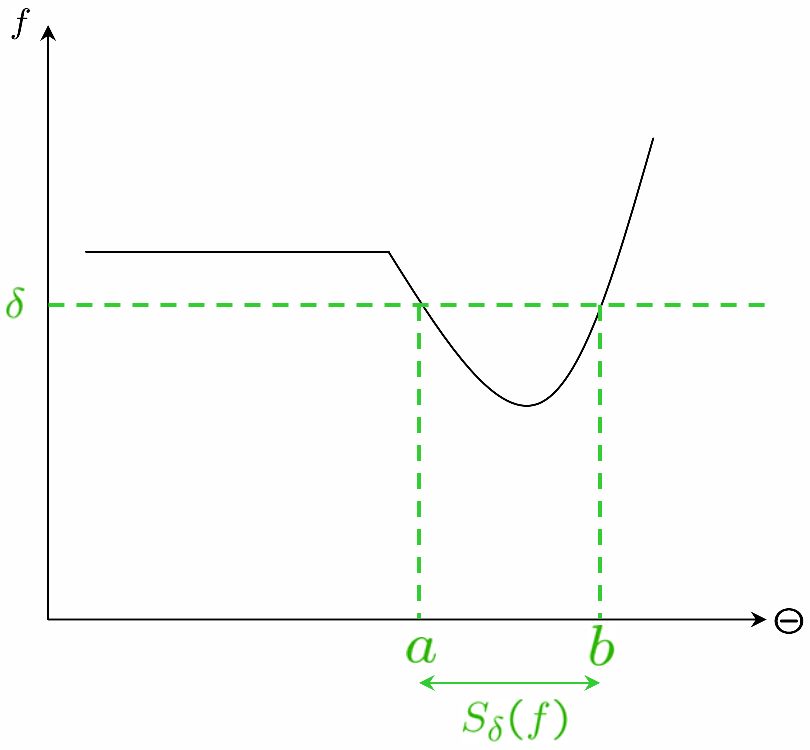
$h =$ red half plane

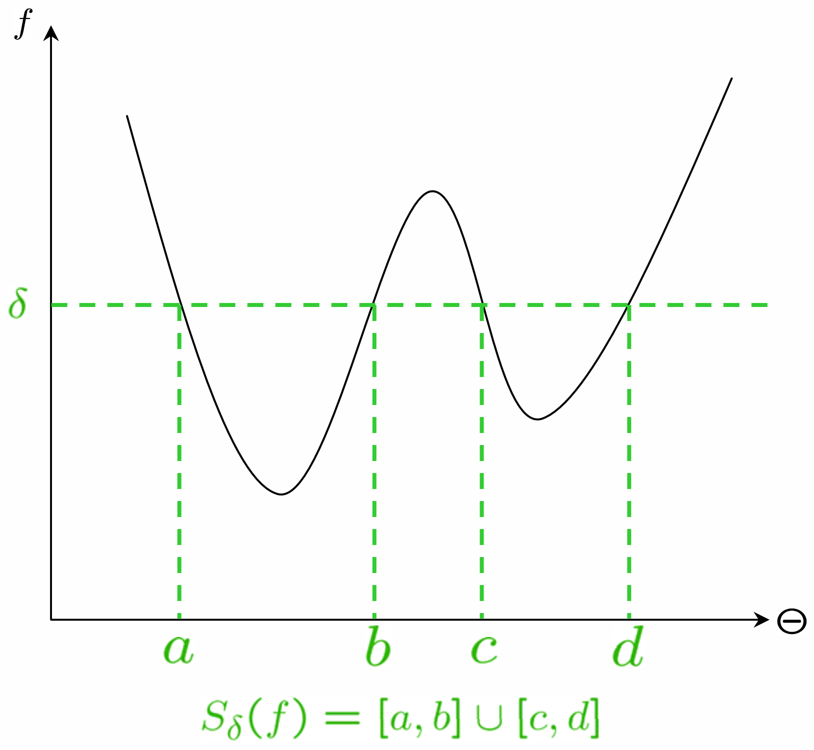
No plateaus! Degenerate in this case.

Recall the Definition of ...

Quasiconvex Functions

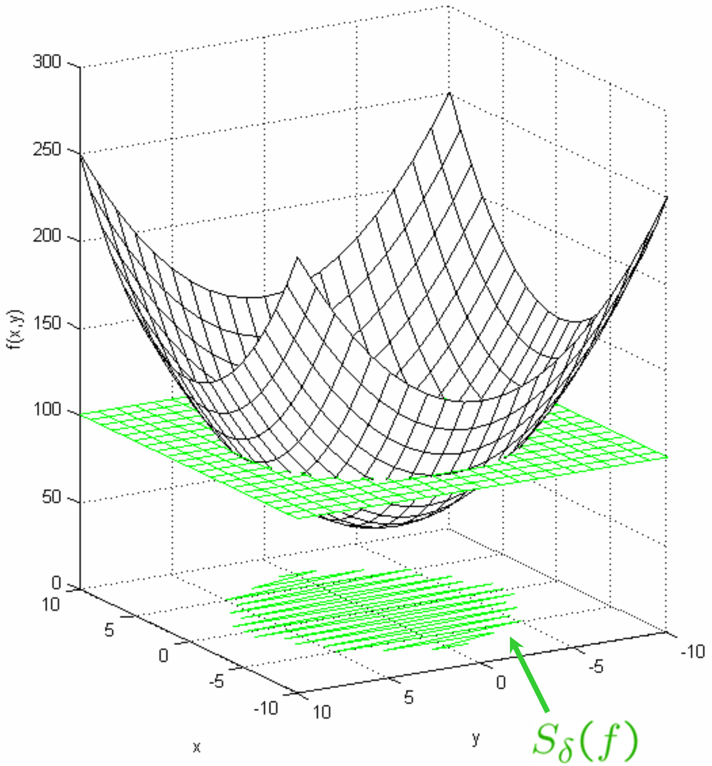
Sublevel Sets: $S_\delta(f) = \{\Theta | f(\Theta) \leq \delta\}$

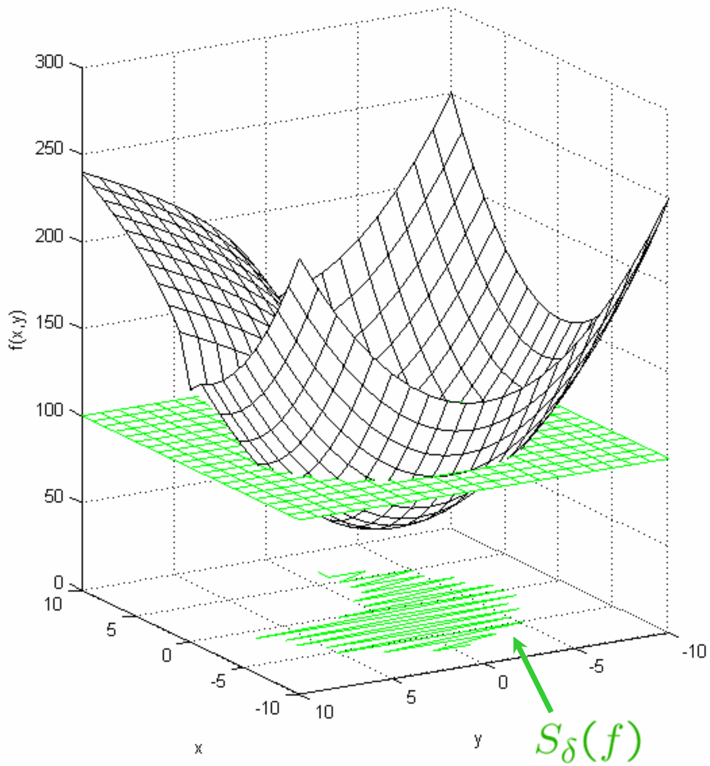




Quasiconvex Functions

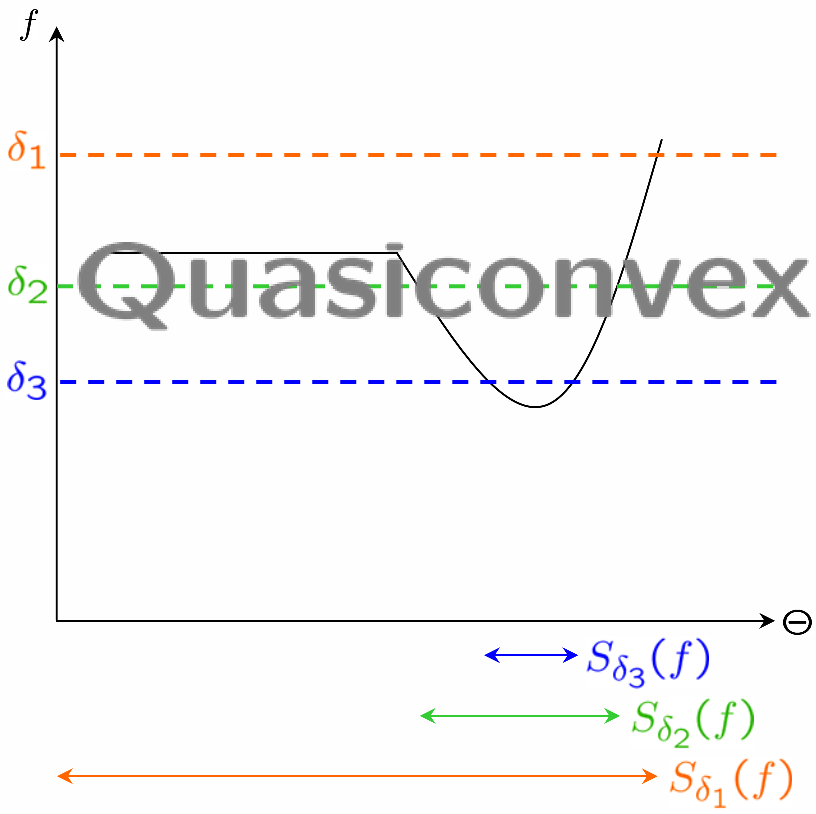
Sublevel Sets: $S_\delta(f) = \{\Theta | f(\Theta) \leq \delta\}$

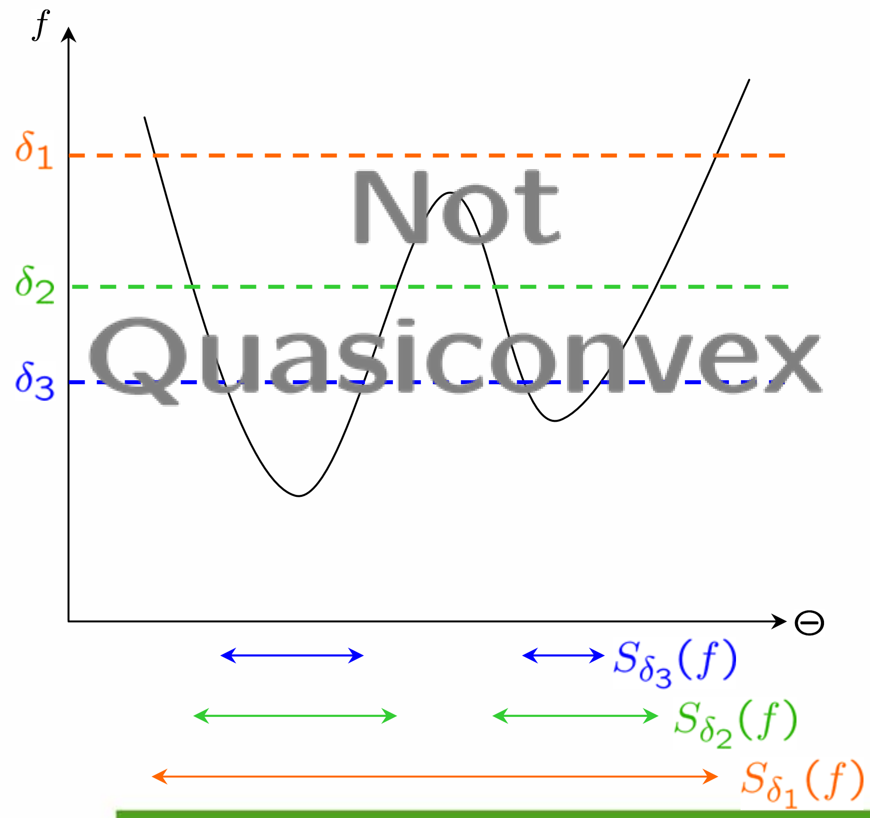




Quasiconvex Functions

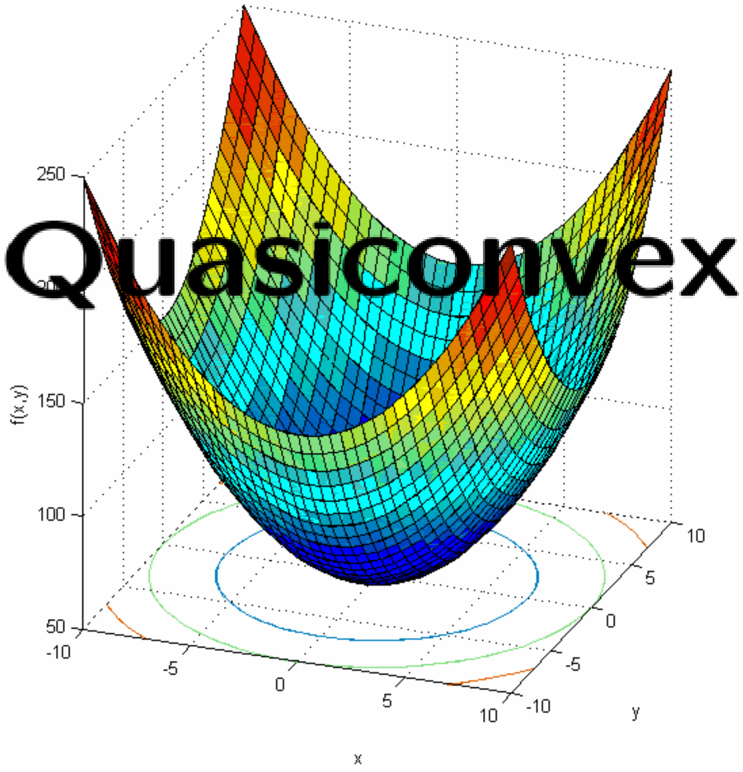
f is **quasiconvex** if its sublevel sets $S_\delta(f)$ are convex $\forall\, \delta$.

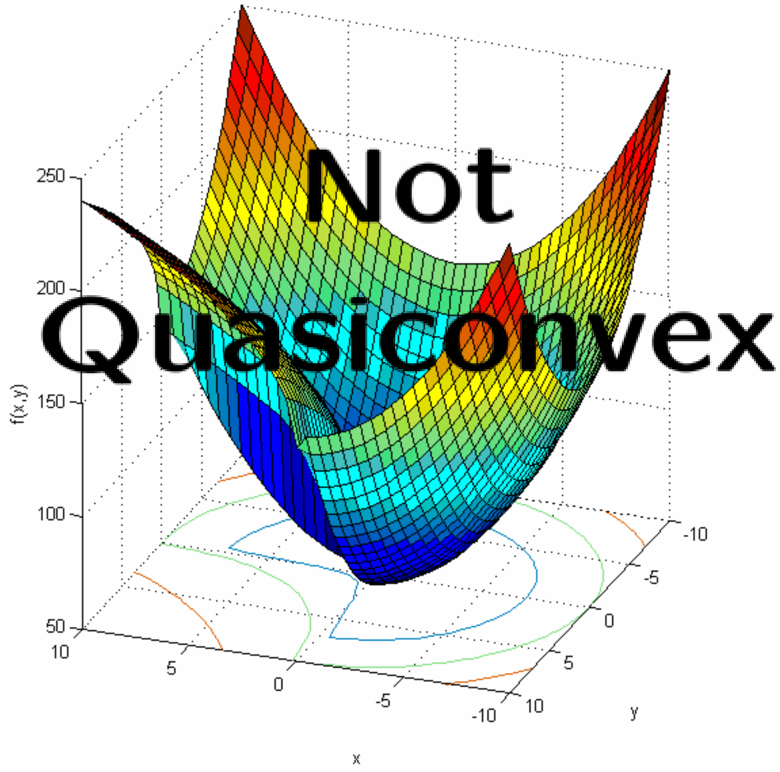




Quasiconvex Functions

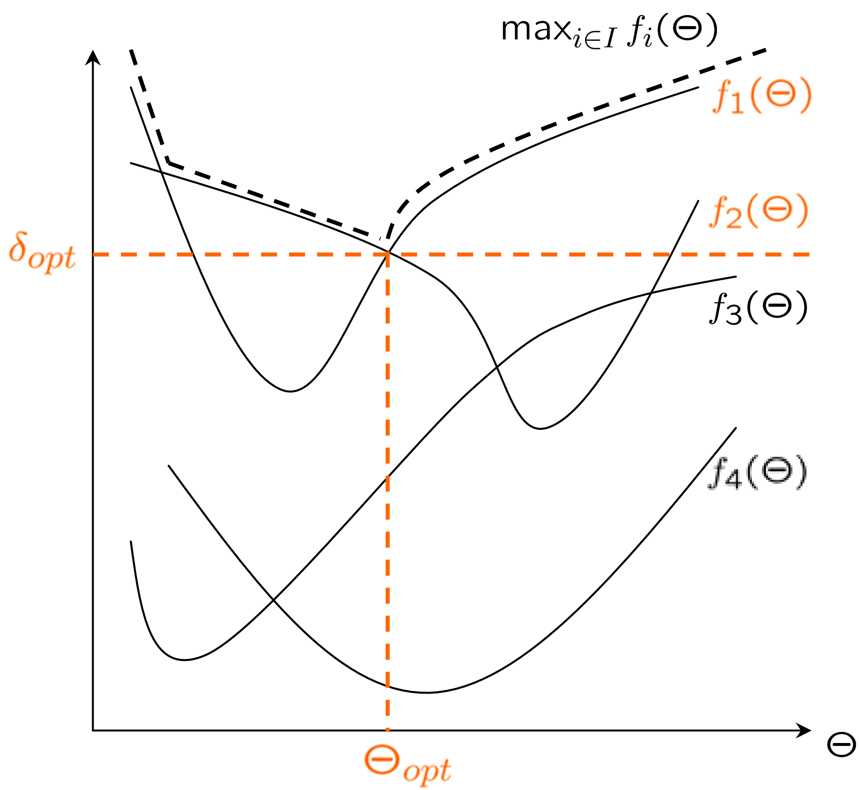
f is **quasiconvex** if its sublevel sets $S_\delta(f)$ are convex $\forall\, \delta$.





Back to Quasiconvex Problems

$$\delta_{opt} = \min_{\Theta} \max_{i \in I} f_i(\Theta)$$



$B = \{f_i(\Theta) | f_i(\Theta_{opt}) = \delta_{opt}\} = \{f_1(\Theta), f_2(\Theta)\}$

B is a basis for $H = \{f_1(\Theta), f_2(\Theta), f_3(\Theta), f_4(\Theta)\}$

Quasiconvex Problems

Are quasiconvex problems LP-type problems?

Let H be set of quasiconvex constraints. Set $w(G)$ to be the objective function. We need to show monotonicity and locality.

Axiom 1. (Monotonicity) For any $F \subseteq G \subseteq H$, we have

$$w(F) \leq w(G).$$

This is easy to show: Adding more constraints can never decrease the objective function, cf. Linear Programming.

Locality is harder. This needs some more work.

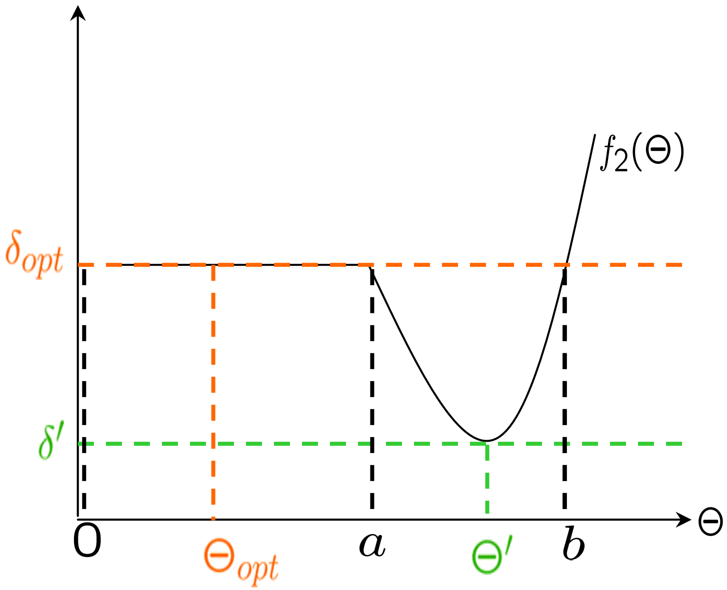
If so, what is the combinatorial dimension?

We will get back to this question as well.

Strict Quasiconvexity

Strict quasiconvexity: As δ decreases, the sublevel sets $S_\delta(f)$ must shrink smoothly.
That is, no plateaus allowed.

Definition: f is strictly QC if $\cup_{\mu<\delta} S_\mu(f) = \text{Int } S_\delta(f) \quad \forall\ \delta$



For function f_2 at δ_{opt}:

- $\cup_{\mu<\delta_{opt}} S_\mu(f_2) = (a, b)$
- Int $S_{\delta_{opt}}(f_2) = (0, b)$
- $\cup_{\mu<\delta_{opt}} S_\mu(f_2) \neq$ Int $S_{\delta_{opt}}(f_2)$
- f_2 is NOT strictly quasiconvex

Strict Quasiconvexity

SOCP problems have error functions $f_i(\Theta)$ that are strictly QC. Proved in [Sim-Hartley-CVPR-2006].

- Two view triangulation *[Nister 2001]*;
- Multiview triangulation *[Hartley & Schaffalitzky CVPR04]*;
- Multiview SFM, known rotations *[Hartley CVPR04, Kahl ICCV05, Ke ICCV05]*;
- Reconstruction with plane-induced homographies *[Kahl ICCV05, Ke ICCV05]*;
- Homography estimation *[Kahl ICCV05, Ke ICCV05]*;
- Camera resectioning *[Kahl ICCV05, Ke ICCV05]*;
- Camera motion recovery *[Sim & Hartley CVPR06]*;
- Vanishing point computation in images *[Hartley 2006]*;

Strict quasiconvexity implies locality, cf. Linear Programming. Hence, our min-max problems are LP-type.

Quasiconvex Optimization

We assume the following three primitive operations:

Violation test: for a constraint h and a basis B, test whether h is violated by B.

Just check $h(\Theta_B) \leq \delta_B$.

Basis computation: for a constraint h and a basis B, compute basis of $B \cup \{h\}$.

Use, for example, bisection.

Initial basis: An initial basis B_0.

Pick random constraints according to combinatorial dimension.

Combinatorial Dimension

Applications:

- $d = 3$ for smallest enclosing circle.
- $d = m$ for Linear Programming in R^m.
- $d = m + 1$ for quasiconvex problems with m degrees of freedom. In particular:
  – $d = 4$ for L_∞-triangulation.
  – $d = 9$ for L_∞-homography estimation.
  – etc.

For a proof for quasiconvex problems, see

- C. Olsson, O. Enqvist, F. Kahl. A Polynomial-Time Bound for Matching and Registration with Outliers. *CVPR*. 2008.

Time complexity is $O(K_d n)$!

Literature

Computational Geometry:

- P.K. Agarwal, M. Sharir. Efficient Algorithms for Geometric Optimization. *ACM Computing Surveys*. 1998.

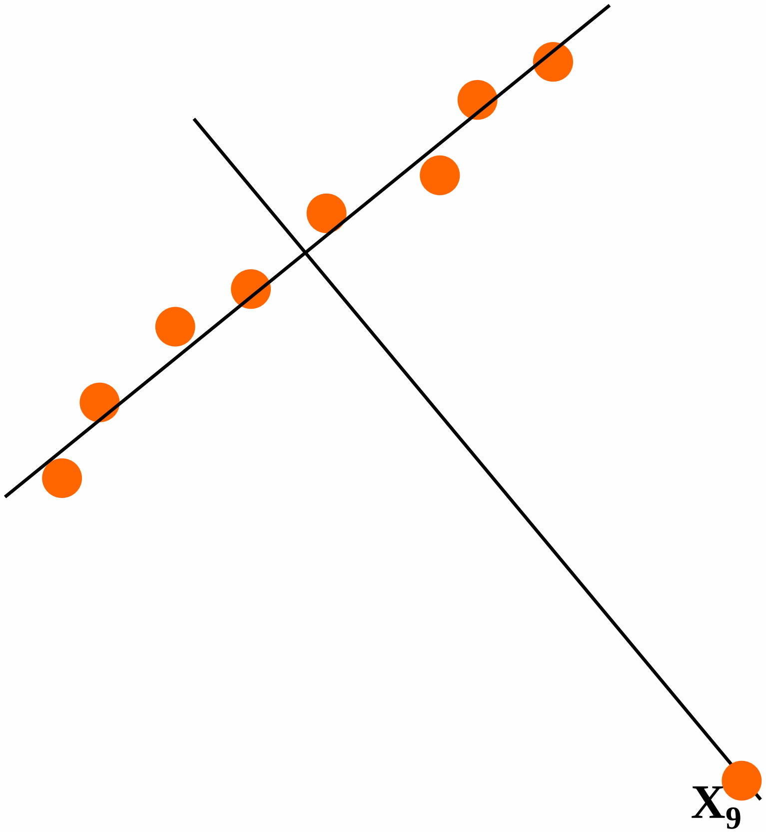
Quasiconvex Problems:

- Y. Seo, R. Hartley. A Fast Method to Minimize L_∞ Error Norm for Geometric Vision Problems *ICCV*. 2007.
- C. Olsson, A. Eriksson, F. Kahl. Efficient Optimization for L_∞ problems using Pseudoconvexity. *ICCV*. 2007.
- S. Agarwal, N. Snavely and S. M. Seitz. Fast Algorithms for L_∞ Problems in Multiview Geometry. *CVPR*. 2008.

Outliers

— detection/removal of outliers

Why are Outliers a Problem?



Problem: Find line of best fit

Measurements: $\mathbf{X}_i = (x_i, y_i)$
Parameters: $\Theta = \{a, b\}$
Error functions: $f_i(\Theta) = (y_i - ax_i - b)^2$

L_2 **optimization:**
$\min_{a,b} \sum_i (y_i - ax_i - b)^2$
L_∞ **optimization:**
$\min_{a,b} \max_i (y_i - ax_i - b)^2$

X_9 is an OUTLIER.
We need to remove it!

Overview

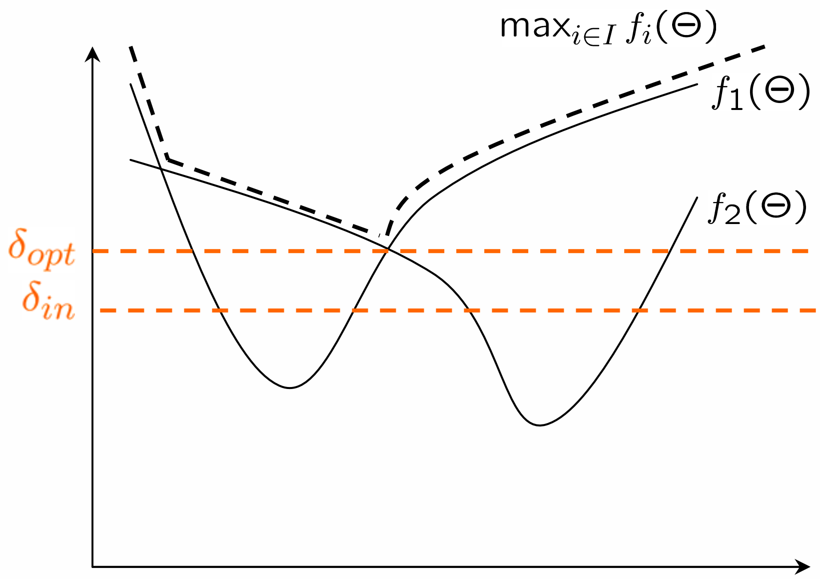
When the L_∞-idea was first introduced, it was considered a major drawback its sensitivity to outliers.

Now, one of its strengths.

Many different ideas and approaches for detection and removal introduced last few years.

- Outlier detection [Sim-Hartley].
- Abstract LP-approach [Li].
- Minimize infeasibility [Seo and Ke-Kanade].
- Verification strategy [Olsson-Enqvist-Kahl].

How to define an outlier?



Suppose only two error functions $f_1(\Theta)$ and $f_2(\Theta)$.

Choose a threshold δ_{in}.

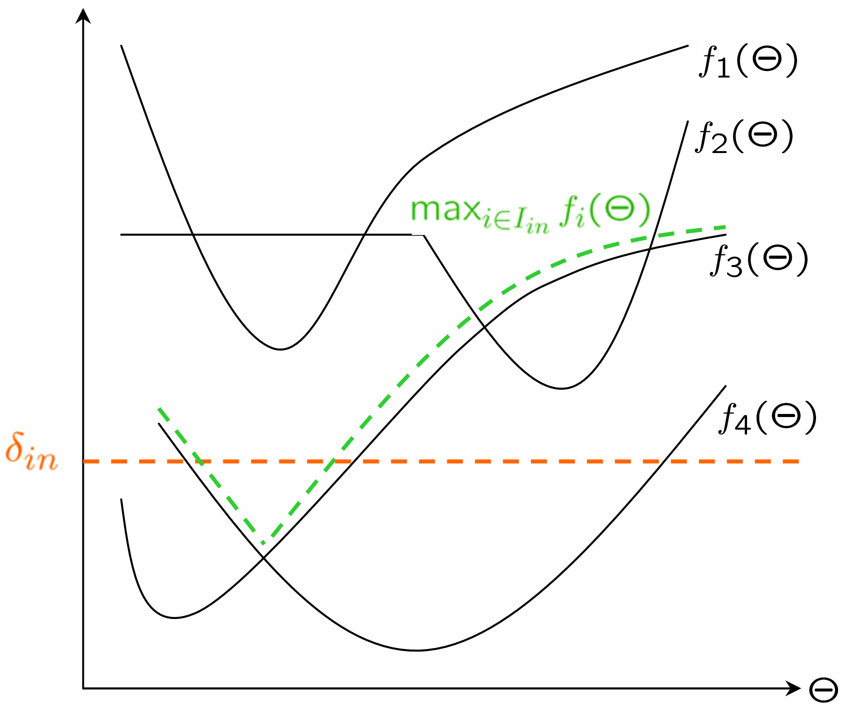
Then either $f_1(\Theta)$ or $f_2(\Theta)$ has to be removed such that

$$\min_{\Theta} \max_{i \in I_{in}} f_i(\Theta) < \delta_{in}$$

where $I_{in} = \{1\}$ or $I_{in} = \{2\}$. But which one?

It is inherently AMBIGUOUS.

Definition of an Outlier



$I_{in} = \{3, 4\}$ and $I_{out} = \{1, 2\}$

We have error functions $f_i(\Theta)$ indexed by i in an index set I.

Choose a threshold δ_{in}.

Choose largest subset I_{in} (the inlier set) that satisfies

$$\min_{\Theta} \max_{i \in I_{in}} f_i(\Theta) < \delta_{in}$$

An **inlier** is any measurement in I_{in}
An **outlier** is any measurement not in I_{in}.

Index set I is made up of two subsets - I_{in} (inlier set) and I_{out} (outlier set).

$$I = I_{in} \cup I_{out}$$

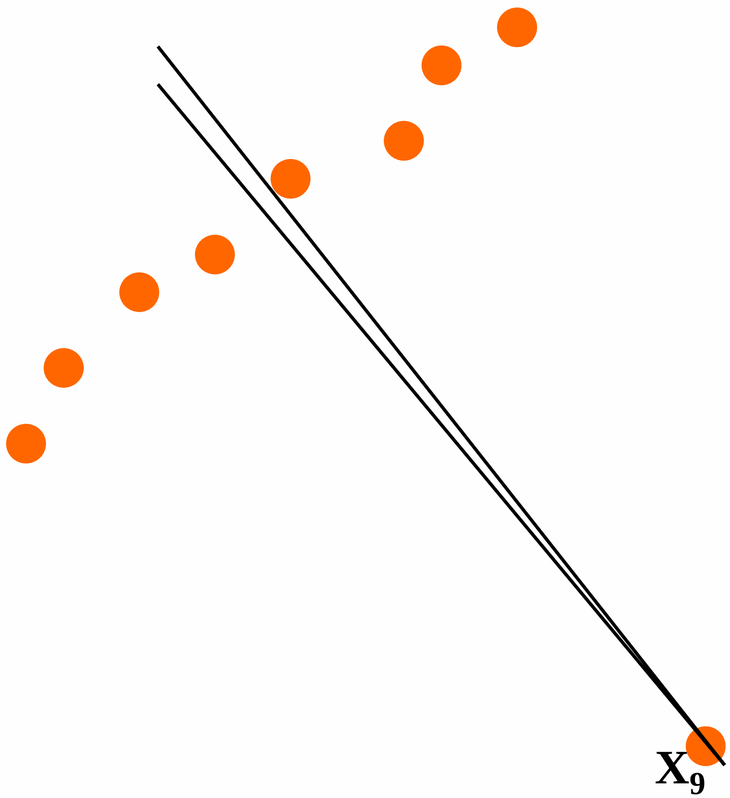
How Do We Remove Outliers?

- **Method 1:** RANSAC
  - Relies on random sampling to find a set of measurements containing only inliers.
  - Can only be used on problems where solution can be computed quickly and from only a small number of measurements.
  - Some outliers may be missed because they happen to fit the model used.

- **Method 2:** Throw out measurements with largest residual
  - Solve optimization problem.
  - Remove measurements with largest residual.
  - Repeat first two steps until an acceptable max residual is achieved.

For this to work, the set of measurements with largest residual must contain outliers. BUT THIS IS NOT ALWAYS THE CASE!

Outlier Removal Strategy

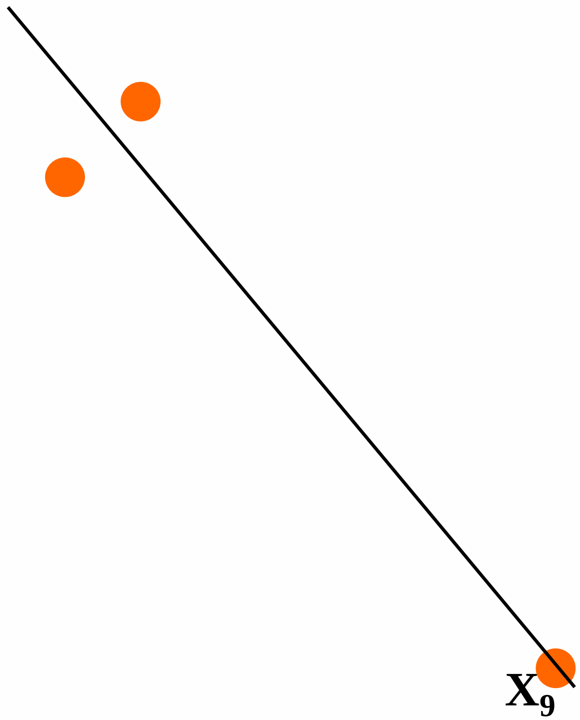


Outlier removal strategy:

– Solve optimization problem

– Remove measurements with largest residual

Outlier Removal Strategy



Outlier removal strategy:

– Solve optimization problem

– Remove measurements with largest residual

Why does strategy fail for general L_2 or L_∞ problems?

For general L_2 or L_∞ problems, the set of measurements with largest residual does not necessarily contain outliers.

BUT strategy works for certain L_∞ problems!

We show that, under certain conditions, the measurements with largest residual are guaranteed to contain outliers.

What Conditions Are Needed?

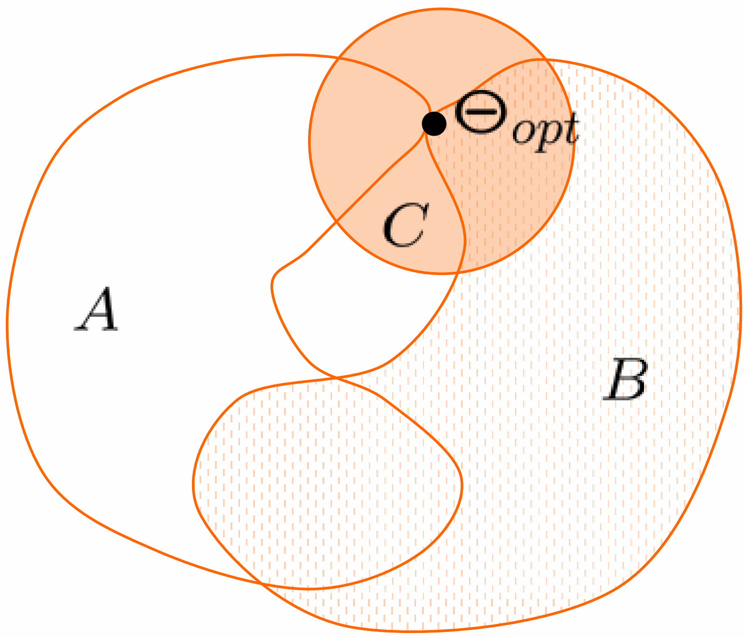
Theorem: (Under certain conditions)
Consider a minimax problem with solution $\min_\Theta \max_{i \in I} f_i(\Theta) = \delta_{opt}$.
Suppose there exists $I_{in} \subset I$ for which $\min_\Theta \max_{i \in I_{in}} f_i(\Theta) < \delta_{in} < \delta_{opt}$.
Then I_{supp} must contain at least one index i not in I_{in}.

In English: The support set must contain at least one outlier.

Condition A: (Under certain conditions)
If f_0 is a function not in the support set for a minimax problem,
then we can remove f_0 without decreasing the L_∞ error δ_{opt}.
That is, if $0 \notin I_{supp}$, then $\min_\Theta \max_{i \in I - \{0\}} f_i(\Theta) = \min_\Theta \max_{i \in I} f_i(\Theta) = \delta_{opt}$.

In English: If $f_0 \notin I_{supp}$, then f_0 should not be constraining our solution.
So we can remove f_0 without affecting the L_∞ error δ_{opt}.

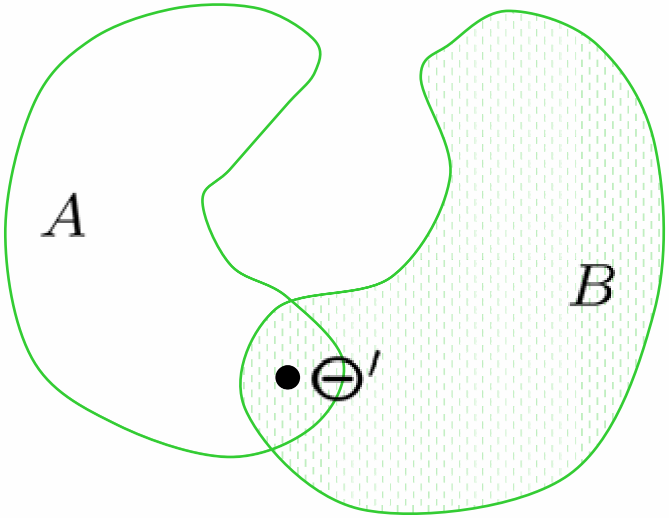
Quasiconvexity Is Needed For Condition A To Hold



- A, B, C are the sublevel sets of 3 error functions $f_{i_A}, f_{i_B}, f_{i_C}$.
- f_{i_C} is QC $\Rightarrow$ C is a convex set
  f_{i_A}, f_{i_B} are not QC $\Rightarrow$ A, B are nonconvex sets
- $\Theta_{opt} = A \cap B \cap C$

- $\Theta_{opt} \notin bd(C) \Rightarrow f_{i_C} \notin I_{supp} = \{i_A, i_B\}$
- Suppose we remove f_{i_C}.

Quasiconvexity Is Needed For Condition A To Hold

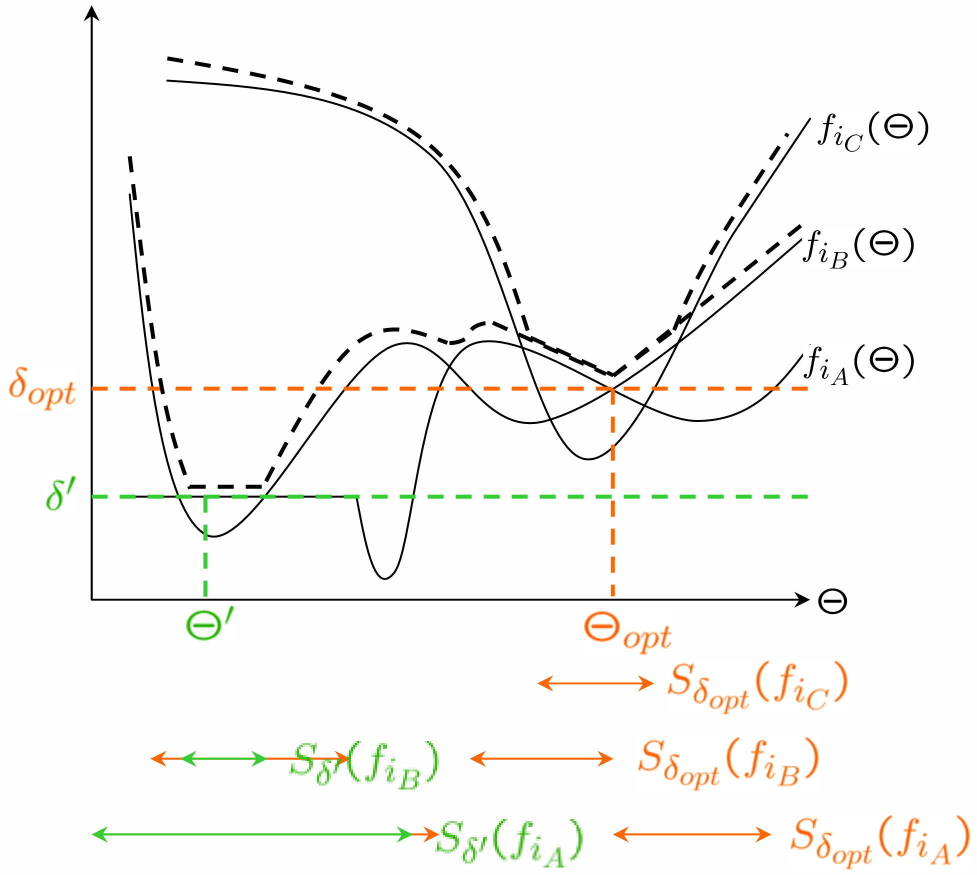


- A, B, C are the sublevel sets of 3 error functions $f_{i_A}, f_{i_B}, f_{i_C}$.
- f_{i_C} is QC $\Rightarrow C$ is a convex set
  f_{i_A}, f_{i_B} are not QC $\Rightarrow A, B$ are nonconvex sets
- $\Theta_{opt} = A \cap B \cap C$
- $\Theta_{opt} \notin bd(C) \Rightarrow f_{i_C} \notin I_{supp} = \{i_A, i_B\}$
- Suppose we remove f_{i_C}.
- Since A, B are not convex, the solution may jump to Θ' where $f_{i_A}(\Theta') < \delta_{opt}$ and $f_{i_B}(\Theta') < \delta_{opt}$.
- That is, because A, B are not convex, it is possible to remove $f_{i_C} \notin I_{supp}$ and obtain a lower L_∞ error δ' at Θ'.

We need convex sublevel sets.
Quasiconvexity is needed!

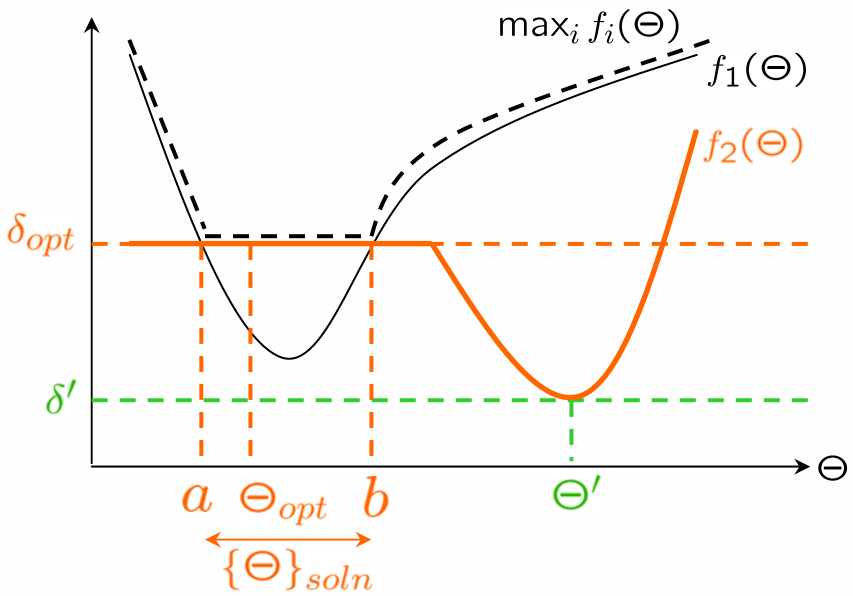
Quasiconvexity Is Needed For Condition A To Hold

We need convex sublevel sets.
Quasiconvexity is needed!



. . . but Quasiconvexity Is Insufficient

If $\{\Theta\}_{soln}$ is a single point, then QC is necessary and sufficient.
If $\{\Theta\}_{soln}$ contains more than a single point, then QC is necessary but insufficient.



$I_{supp} = \{i | f_i(\Theta_{opt}) = \delta_{opt}\} = \{2\}$

- f_1, f_2 are quasiconvex
- $\min_\Theta \max_{i=1,2} f_i(\Theta) = \delta_{opt}$
- $\{\Theta\}_{soln} = \cap_{i=1,2} S_{\delta_{opt}}(f_i) = [a, b]$
- But bisection algorithm only returns a single point $\Theta_{opt} \in \{\Theta\}_{soln}$

- $f_1(\Theta_{opt}) < \delta_{opt} \Rightarrow f_1 \notin I_{supp} = \{2\}$
- Suppose we remove f_1.

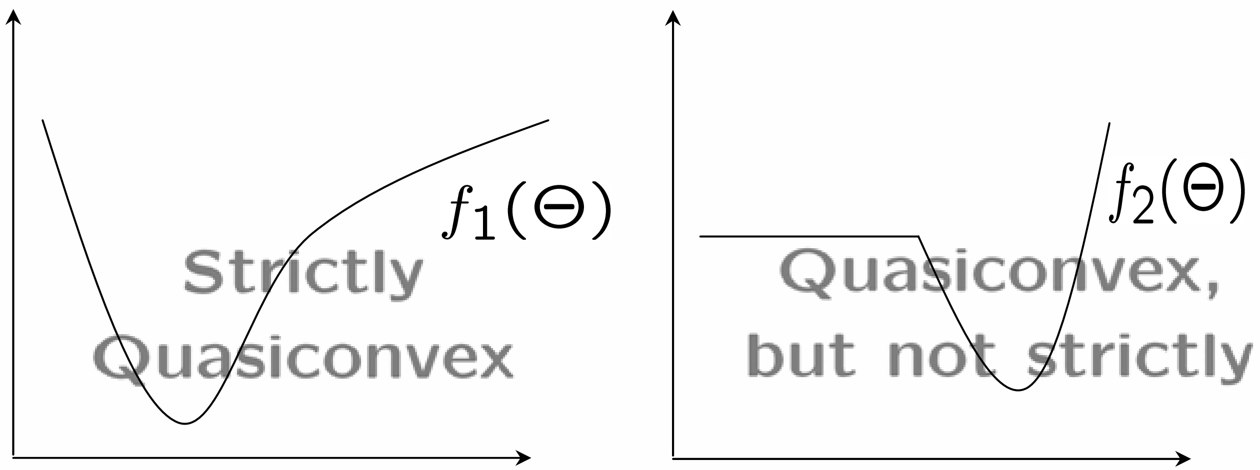
- Bisection algorithm will find a new solution Θ' with a lower L_∞ error δ'.
  $\Rightarrow$ Quasiconvexity is insufficient

Need smoothness condition on sublevel sets.
Strict Quasiconvexity is needed!

Strict Quasiconvexity

Strict quasiconvexity: As δ decreases, the sublevel sets $S_\delta(f)$ must shrink smoothly. That is, no plateaus allowed.

Definition: f is strictly QC if $\cup_{\mu<\delta} S_\mu(f) = \text{Int } S_\delta(f) \quad \forall \delta$



Strict QC is sufficient

Theorem:
Consider a minimax problem with solution $\min_\Theta \max_{i \in I} f_i(\Theta) = \delta_{opt}$ where error functions $f_i(\Theta)$ are all strictly quasiconvex.
Suppose there exists $I_{in} \subset I$ for which $\min_\Theta \max_{i \in I_{in}} f_i(\Theta) < \delta_{in} < \delta_{opt}$.
Then I_{supp} must contain at least one index i not in I_{in}.

In English: If our error functions $f_i(\Theta)$ are all strictly quasiconvex, then the support set must contain at least one outlier.

In Abstract LP-language: Let B_{in} be a basis for error functions in I_{in} and B_H a basis for all error functions.
Hence $w(B_{in}) < \delta_{in} < \delta_{opt} = w(B_H)$.
Then B_H must contain at least one outlier.
For a detailed proof, see:

- K. Sim, R. Hartley. Removing Outliers Using the L_∞ Norm. *CVPR*. 2006.

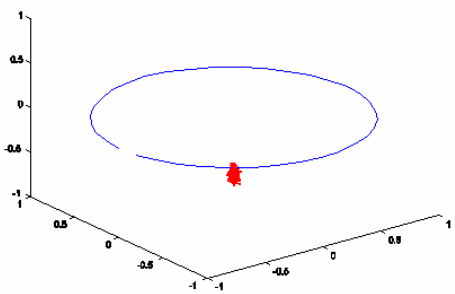
What Does This All Mean?

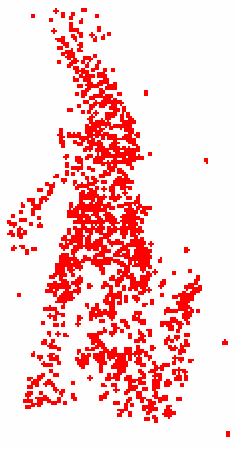
If we can write a geometric vision problem as an L_∞ optimization problem where the error functions $f_i(\Theta)$ are strictly quasiconvex then I_{supp} must contain at least one outlier.

So by repeatedly throwing out part or all of I_{supp}, it should be possible to eventually remove outliers from a given problem.

Results - Reconstruction

- 4402 image points $\mathbf{x}_{ij}$ used to recover 36 camera locations $\mathbf{C}_i$ and 1381 scene points $\mathbf{X}_j$.
- Gaussian noise added to 5% of the 4402 image points $\mathbf{x}_{ij}$ (i.e. 220 outliers).

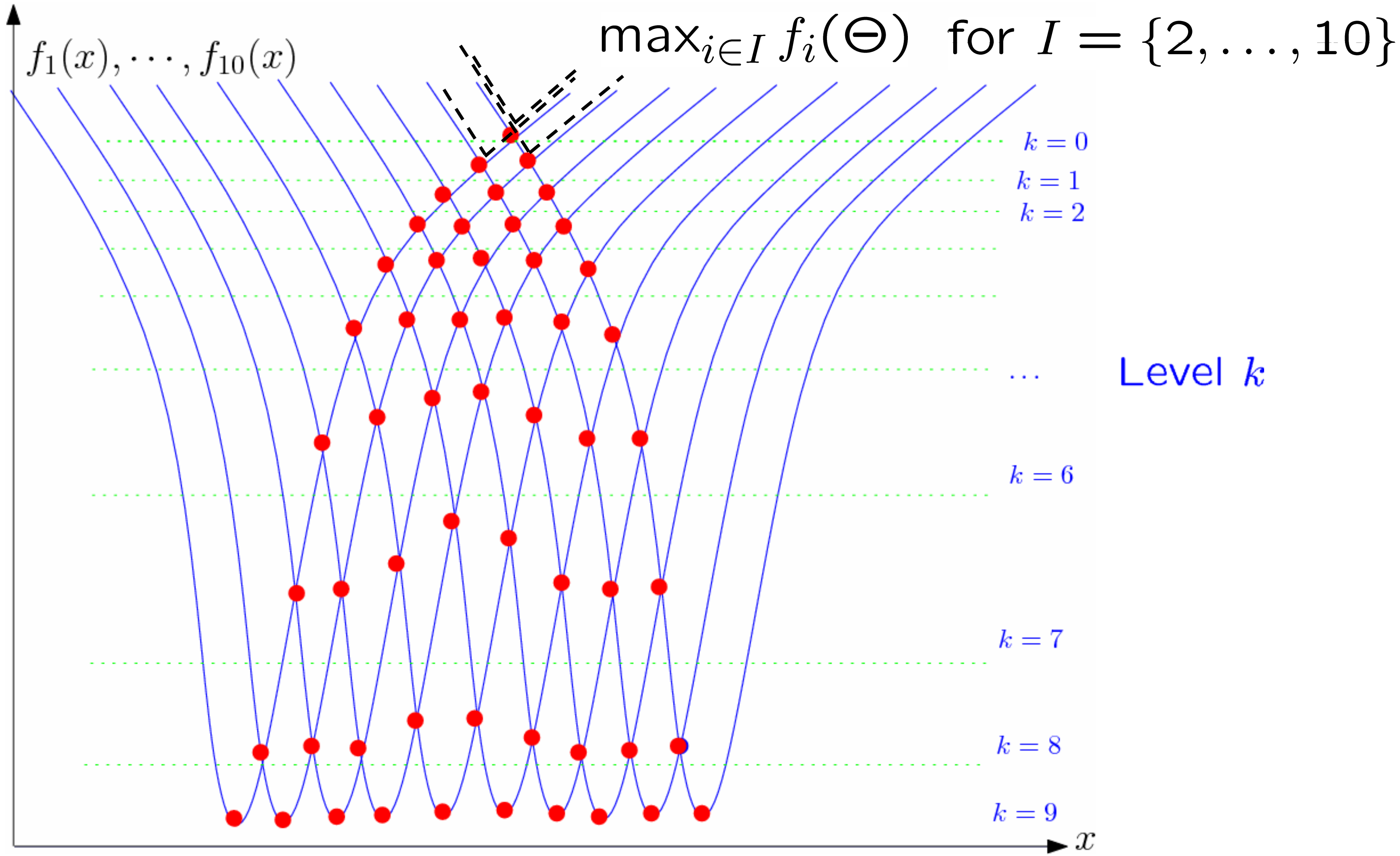




Cycle	Max Residual	Size of I_{supp}	Remaining Outliers
1	0.0390	10	210
2	0.0277	43	168
3	0.0196	54	123
4	0.0140	100	57
5	0.0080	72	23
6	0.0035	60	7
7	0.0019	36	4

Another approach using Abstract LP

- Removing the whole support set is rather crude.
- We know our min-max problems are LP-type.
- Exhaustive search too slow, but using properties of LP-type problems we can do it more efficiently.
- Suppose we seek to remove at most k outliers where k is a small number.



Observation 1: If we are to remove one outlier, it is enough to consider elements in the basis.

Observation 2: Satisfying all but k constraints can be obtained from a path from from $k-1$ constraints.

Outlier removal using Abstract LP

- Exhaustive search but not all subsets need to be investigated.

- More formally, the previous two observations can be stated as follows.

Theorem. (upper bound of cardinality) For a non-degenerate LP-type problem (H, w) of combinatorial dimension d with $w(G) > -\infty$ for any $G \subset H$, the number of bases of level at most k is bounded from above by $|B_{\leq k}| = O((k+1)^d)$.

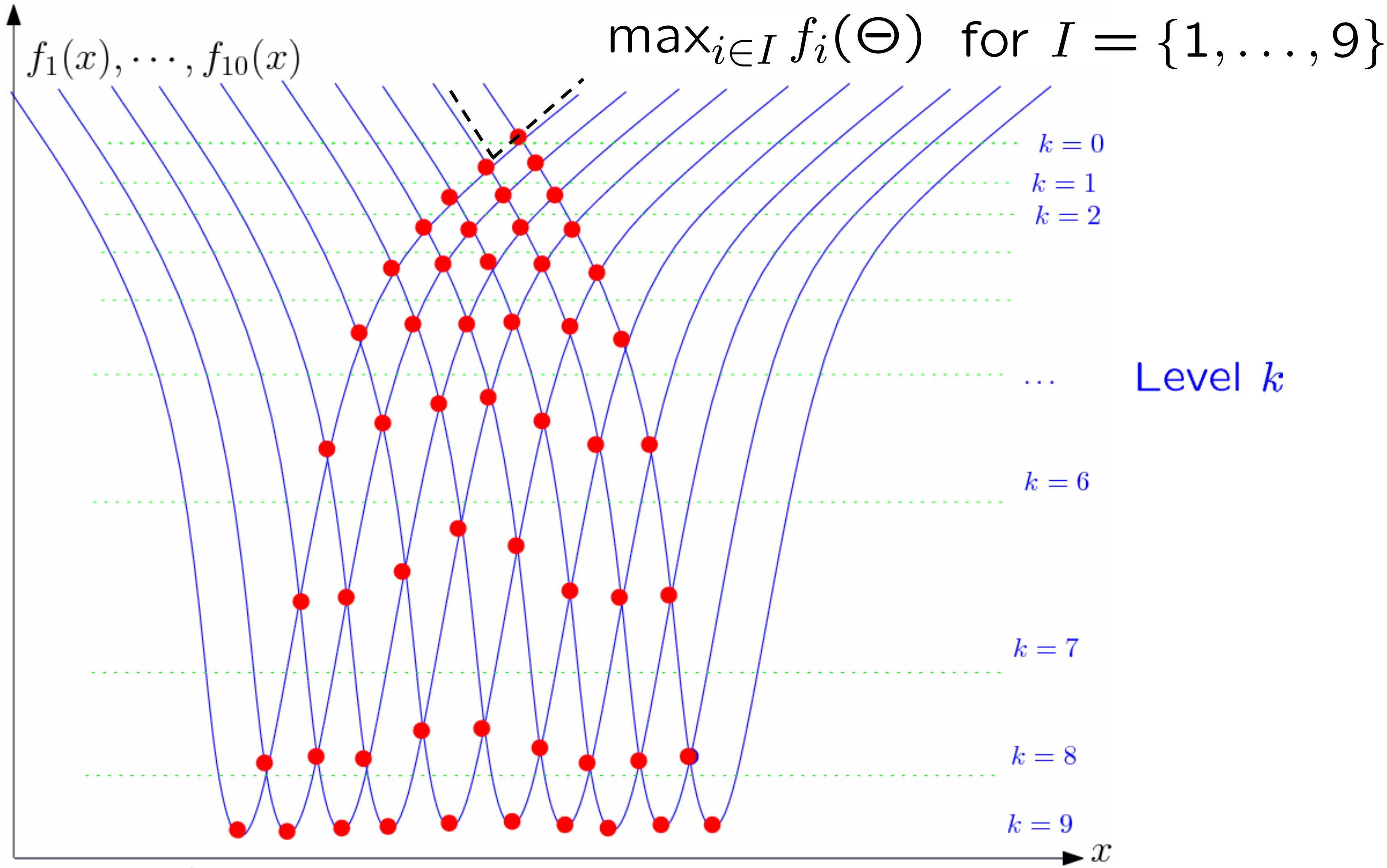
Theorem. (basis reachability) Every basis of level k can be reached from the basis of level $k-1$ through a direct path. Consequently, all bases are connected through a tree structure.

Outlier Removal Using Abstract LP

- A deterministic algorithm for LP-type problems:

Input: an LP-type problem (H, w), a given maximal level K.
Output: all the bases B_k at each level $0 \leq k \leq K$.

1. (Initial basis finding.) Find the root basis set for the universe set H, i.e., $B_0 = B_H$. Let $k = 0$.
2. (Basis change.) Generate all bases at level $k+1$ by performing a series of basis-change operations. Specifically, for every $b \in B_k$, do the following: generate a basis at level $k+1$ for $H \backslash V(B_k) \backslash b$, where $V(B_k)$ is the violation set of B_k.
3. If $k < K$ then $k = k+1$, go back to 2.
4. Output all the bases, i.e., B_0, $B_{1,0}$, $B_{1,1}$, $\cdots$,$B_{K,1}$, $\cdots$.



Example:

First basis of level 1: $B_{1,1} = \{f_1(\Theta), f_9(\Theta)\}$

First basis of level 2: Take $b \in B_{1,1}$, say $b = f_9(\Theta)$.
Then $B_{2,1}$ = basis of $H \backslash V(B_{1,1}) \backslash \{f_9(\Theta)\}$, where $V(B_{1,1}) = \{f_{10}(\Theta)\}$.

Outlier Removal Using Abstract LP

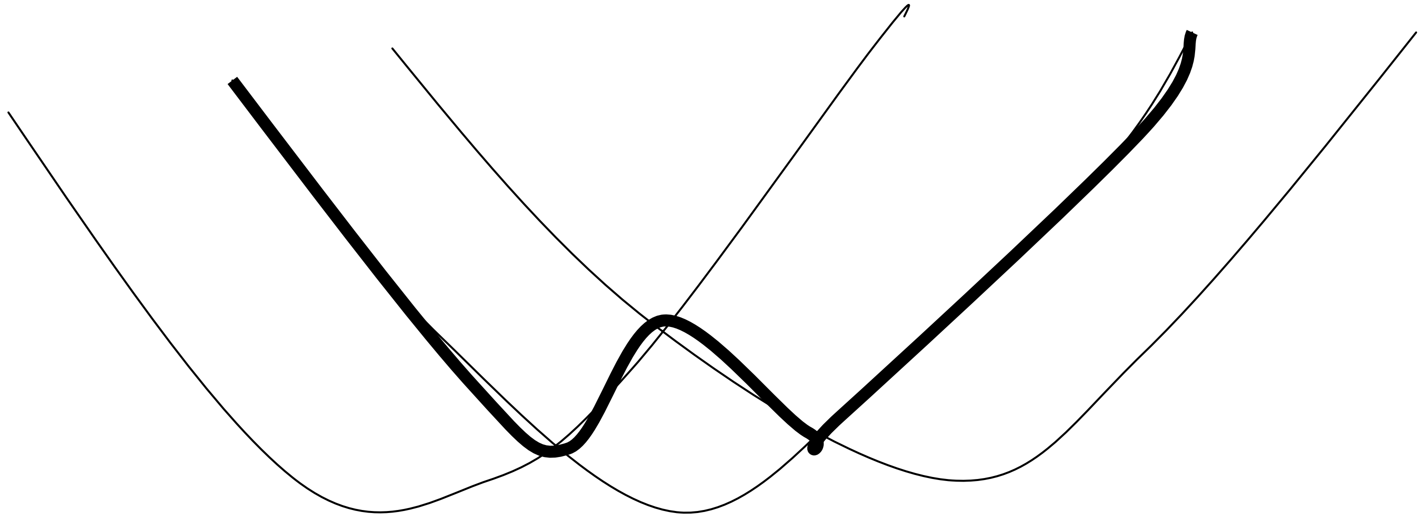
- Pros: Deterministic, guaranteed to get optimal level k (kth median) solution. Relatively fast compared to exhaustive search.

- Cons: Becomes very slow for large k or large d (combinatorial dimension).

Least Median Optimization

Could we minimize the median measurement?

1. k-th largest error is not a convex function.

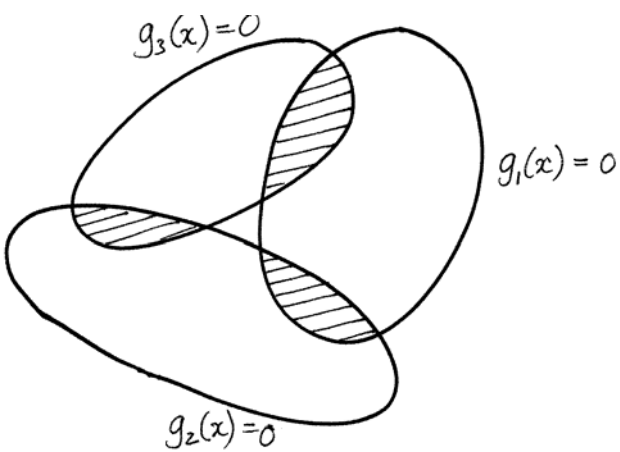


Minimizing the median

Work of Ke and Kanade (ICCV 2005)

$$\begin{array}{rl} \min\limits_{s_i, \mathbf{x}} & s_1 + s_2 + \ldots + s_n \\ \text{subject to} & g_i(\mathbf{x}) \leq s_i \\ \text{and} & s_i \geq 0 \end{array}$$

Use bisection: Feasible if less than k of the s_i are non-zero.



Problem Formulation

Given a set of hypothetic correspondences $\{(x_i, y_i)\}_{i=1}^m$
find the largest consistent set I.

$$\max |I|$$

such that for some transformation $T \in \mathcal{T}$

$$d(T(x_i), y_i) \leq \delta_{in} \text{ for all } i \in I$$

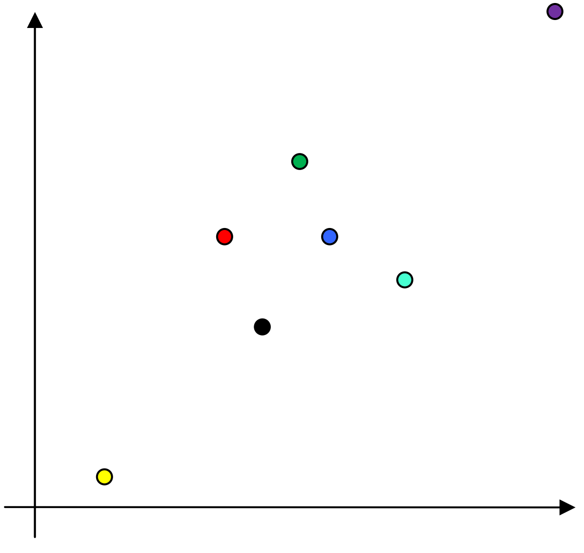
δ_{in} error tolerance
$d(\cdot,\cdot)$ metric

Applications:

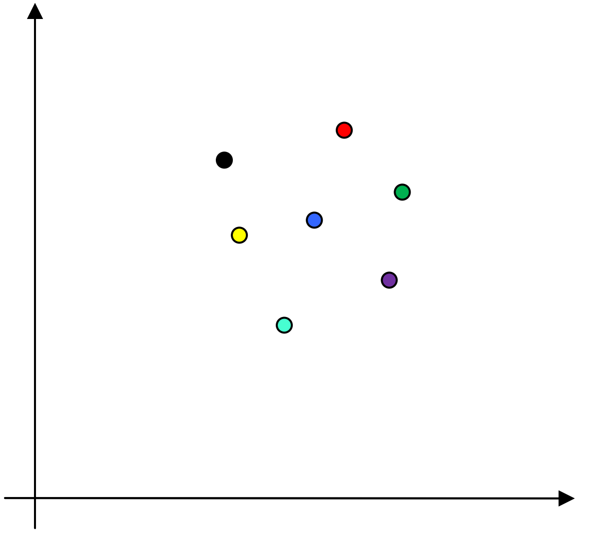
- Multiview geometry problems:
  - Triangulation
  - Uncalibrated camera pose
  - Etc.
- Matching problems
- Registration problems
- Etc.

Registration Problems

Model:

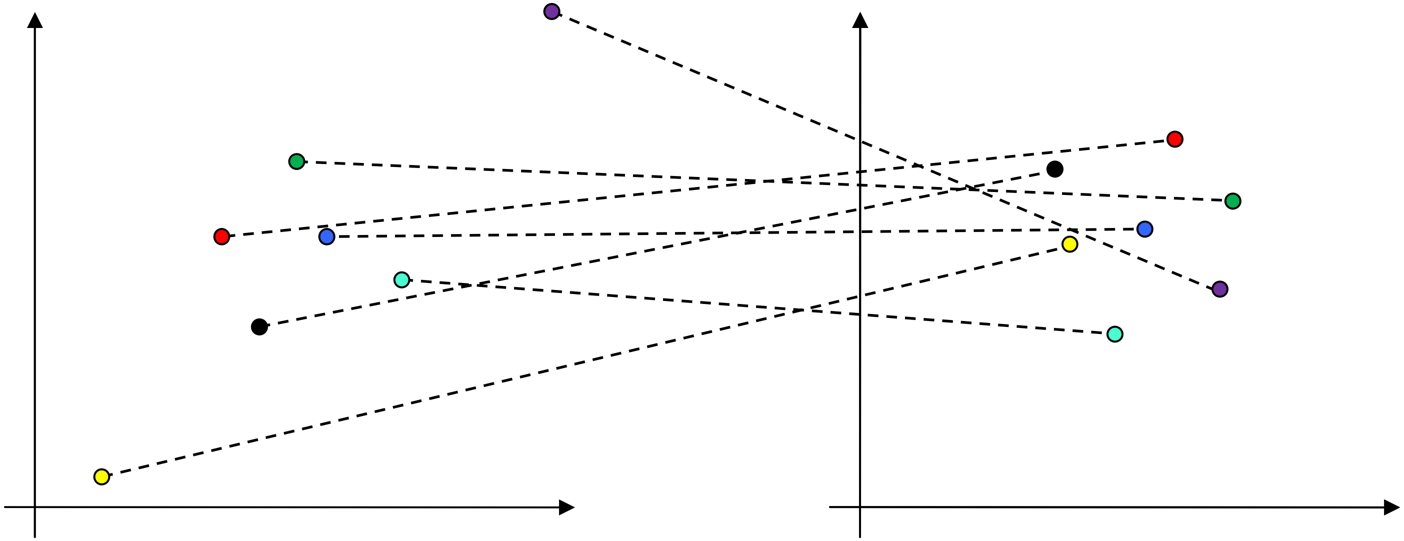


Measurements:

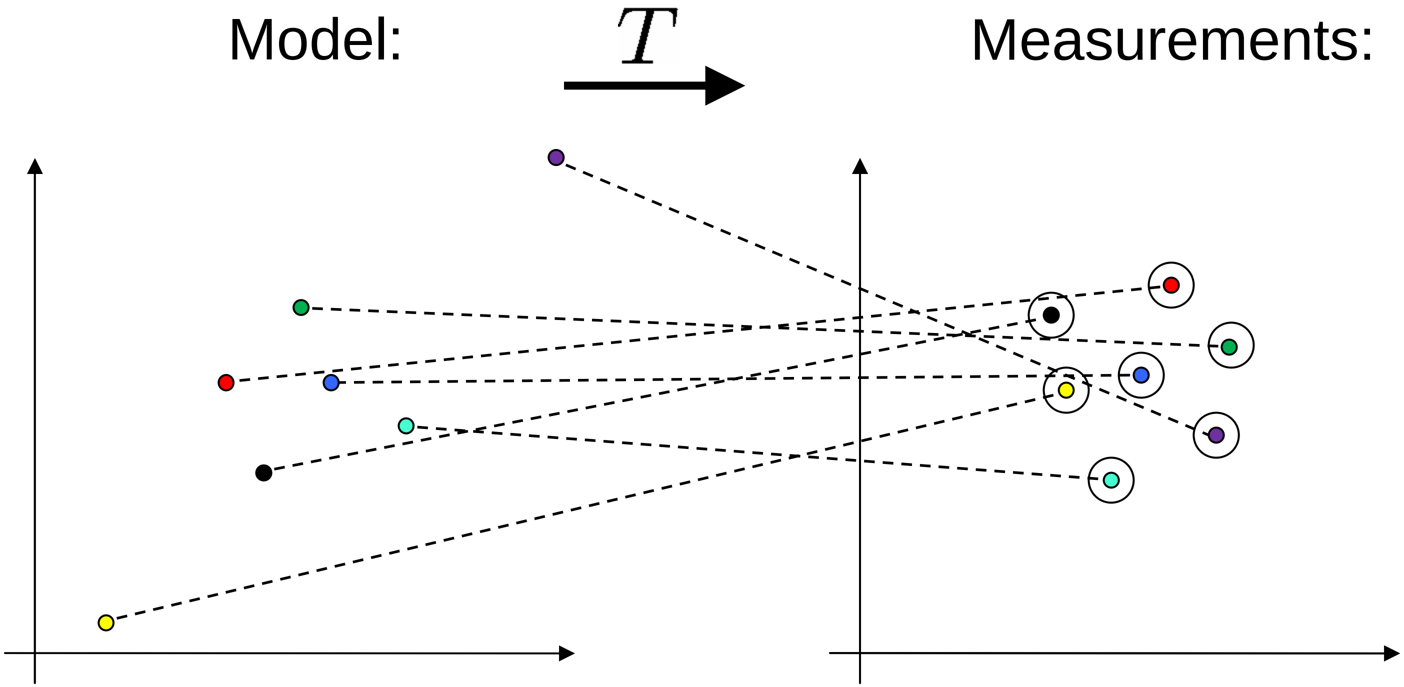


Registration Problems

Model: Measurements:



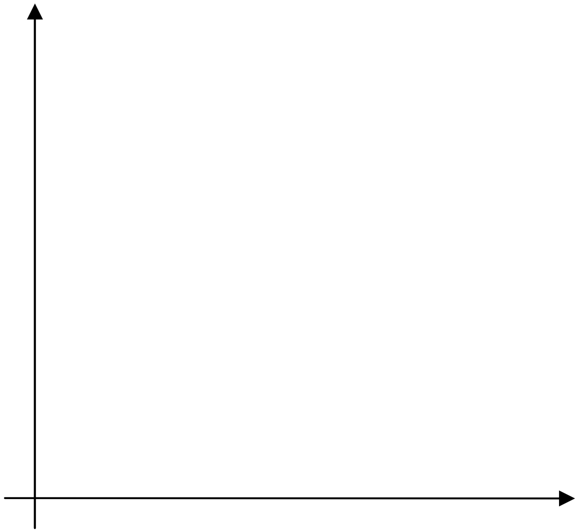
Registration Problems



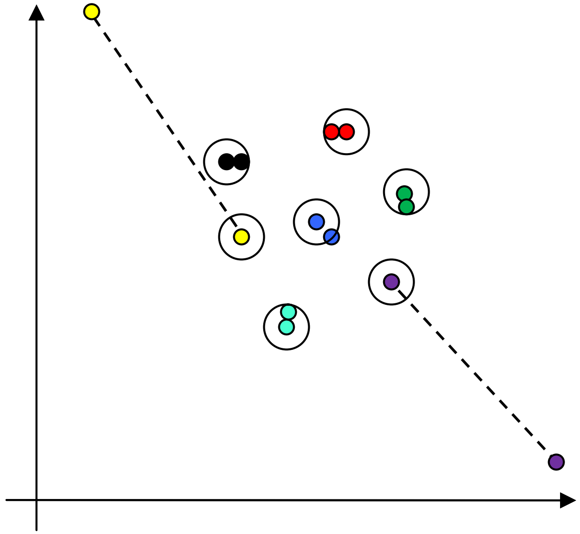
T

Registration Problems

Model:



Measurements:



Registration Problems

Source points: x_i

Target points: y_i, $i = 1, ..., m$

Registration Problems

Source points: x_i

Target points: y_i, $i = 1, ..., m$

$$\begin{array}{ll} \max_{I,T} & |I| \\ \text{s.t.} & d(T(x_i), y_i) \leq \delta_{in}, \quad \forall i \in I \end{array}$$

δ_{in} error tolerance

$d(\cdot,\cdot)$ metric

Registration Problems

$\mathbb{R}^n \ni t \mapsto T_t$ parametrisation of the set of feasible transformations.

Registration Problems

$\mathbb{R}^n \ni t \mapsto T_t$ parametrisation of the set of feasible transformations.

Assumption:

The reprojection errors

$$r_i(t) = d(T_t(x_i), y_i)$$

are quasi/pseudo-convex functions.

Registration Problems

2D Similarity Transformation:

$$\begin{aligned} d(Rx_i + t, y_i) &= ||Rx_i + t - y||_2 \\ R(a,b) &= \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \end{aligned}$$

Homography estimation:

$$d(y, Hx) = \left\| \left(y_1 - \frac{H_1 x}{H_3 x}, y_2 - \frac{H_2 x}{H_3 x} \right) \right\|_2$$

Triangulation/Resectioning:

$$d(u, PU) = \left\| \left(u_1 - \frac{P_1 U}{P_3 U}, u_2 - \frac{P_2 U}{P_3 U} \right) \right\|_2$$

Reformulation

Given a set of hypothetic correspondences $\{(x_i, y_i)\}_{i=1}^{m}$

For some pre-defined $\delta_{in} > 0$

$$\begin{aligned} \min_{s \in \mathbf{R}^m, T \in \mathcal{T}} \quad & ||s||_0 \\ \text{s.t.} \quad & d(T(x_i), y_i) \leq \delta_{in} + s_i \text{ for all } i \\ & s_i \geq 0. \end{aligned}$$

Hard, non-convex problem due to $|| \cdot ||_0$

Norms:

- $||x||_1 = \sum_i |x_i|$
- $||x||_2 = (\sum_i |x_i|^2)^{\frac{1}{2}}$
- $||x||_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$

Quasi-norm:

- $||x||_0 =$ sum of non-zero x_i

Convex L_1-relaxation (Y. Seo, 2008)

Given a set of hypothetic correspondences $\{(x_i, y_i)\}_{i=1}^m$

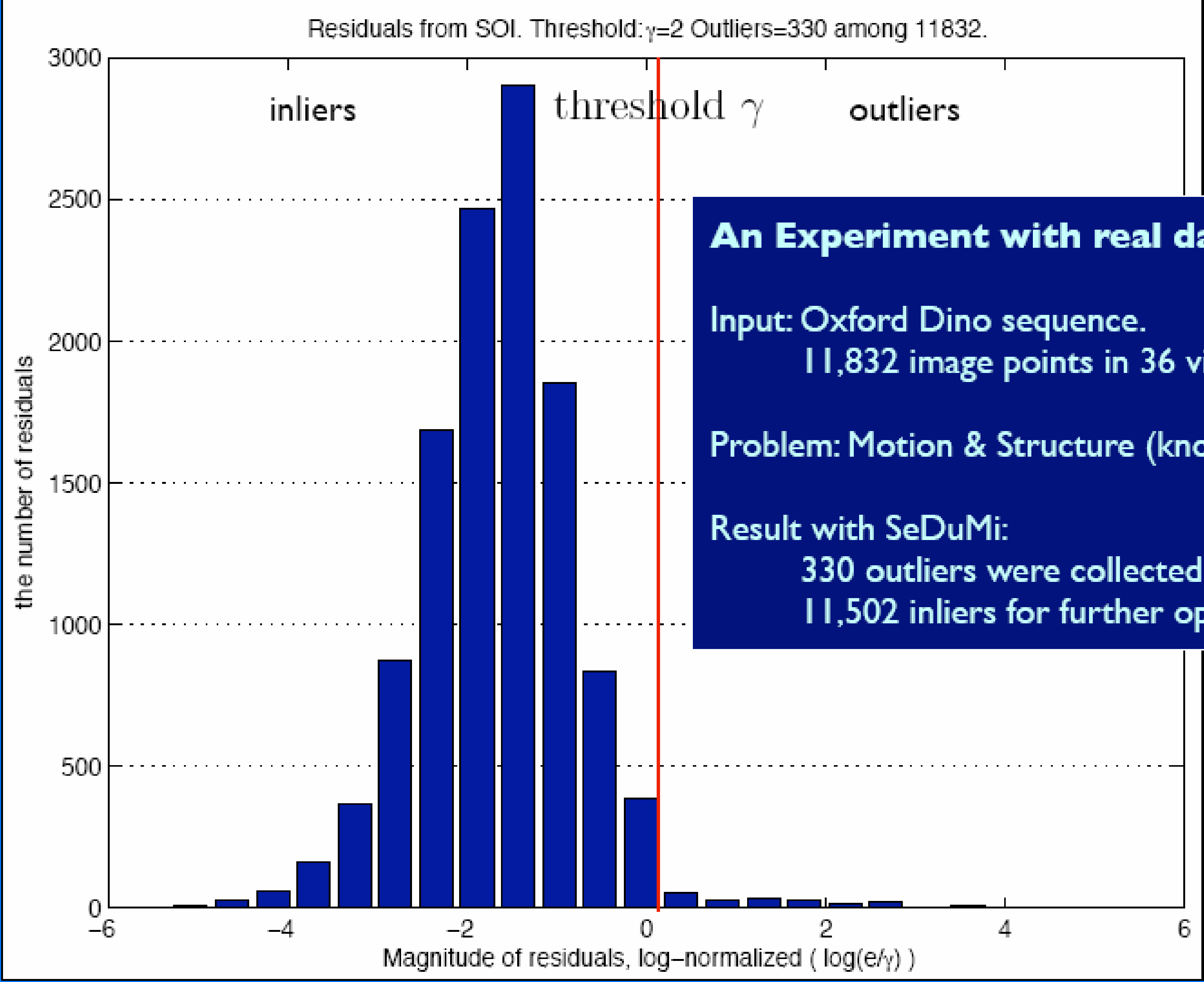
For some pre-defined $\delta_{in} > 0$

$$\begin{aligned} \min_{s \in \mathbf{R}^m, T \in \mathcal{T}} \quad & ||s||_1 \\ \text{s.t.} \quad & d(T(x_i), y_i) \leq \delta_{in} + s_i \text{ for all } i \\ & s_i \geq 0. \end{aligned}$$

Can be solved with LP or SOCP!

- The L_1-norm is known to produce sparse solutions.
- If $||s||_1 = 0$ at optimum, then there exists $T \in \mathcal{T}$ such that $d(T(x_i), y_i) \leq \delta_{in}$ - all correspondences are inliers.
- Any non-zero s_i implies existence of outliers.

Minimization of Sum of Infeasibilities for Outliers Removal



An Experiment with real data

Input: Oxford Dino sequence.
11,832 image points in 36 views.

Problem: Motion & Structure (known rotation).

Result with SeDuMi:
330 outliers were collected.
11,502 inliers for further optimization.

Verifying Optimality

Given a candidate solution I_0 obtained from your favourite heuristics (L_1-relaxation, RANSAC, etc.).

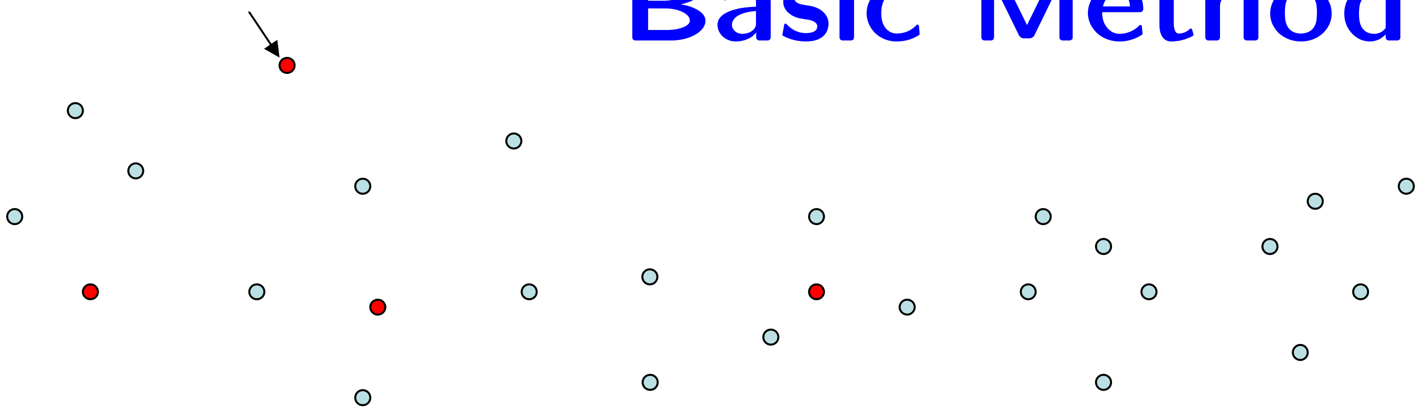
Try to reject the hypothesis that there exists $I_* \neq I_0$ with $|I_*| > |I_0|$.

Method: For each correspondence $h \notin I_0$. Show that it cannot be an element of I_* either.

To be presented this summer:

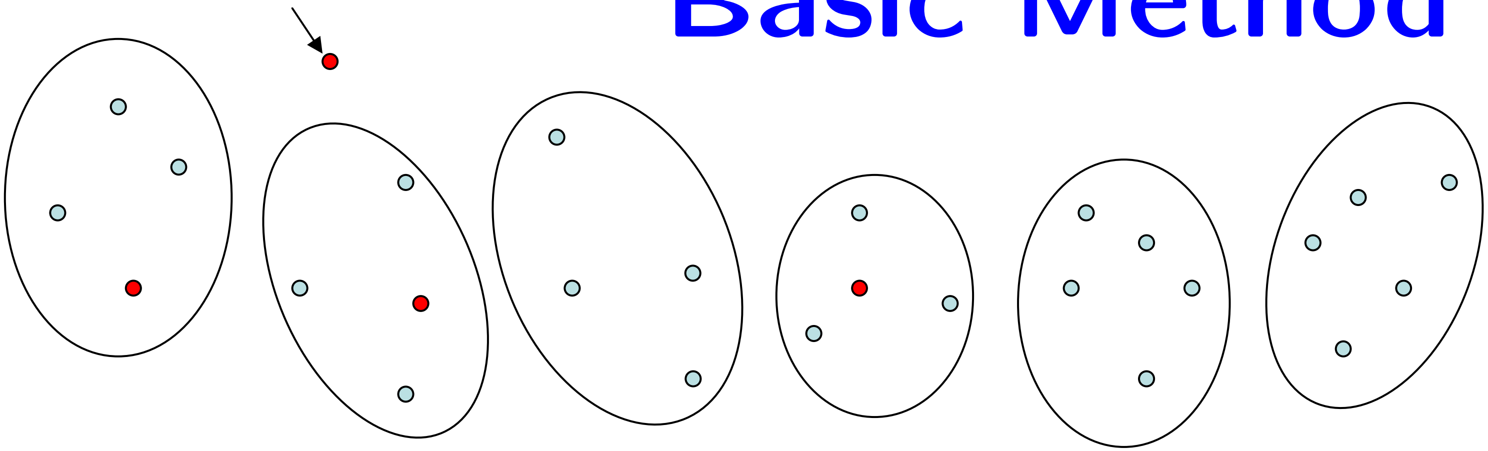
- C. Olsson, O. Enqvist, F. Kahl. A Polynomial-Time Bound for Matching and Registration with Outliers. *CVPR*. 2008.

Basic Method



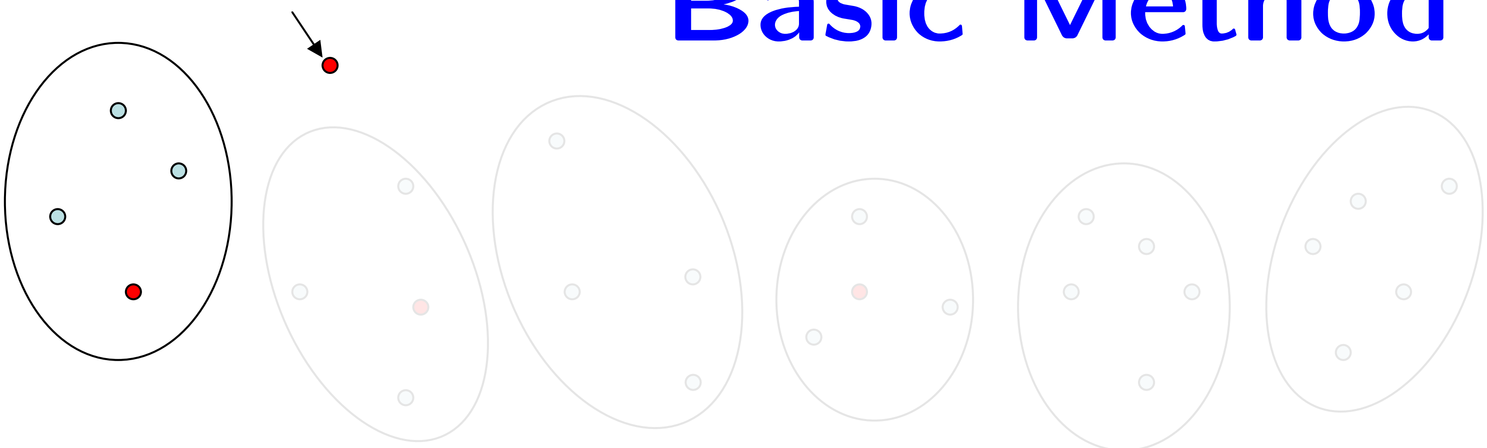
- Pick a $h \notin I_0$.

Basic Method



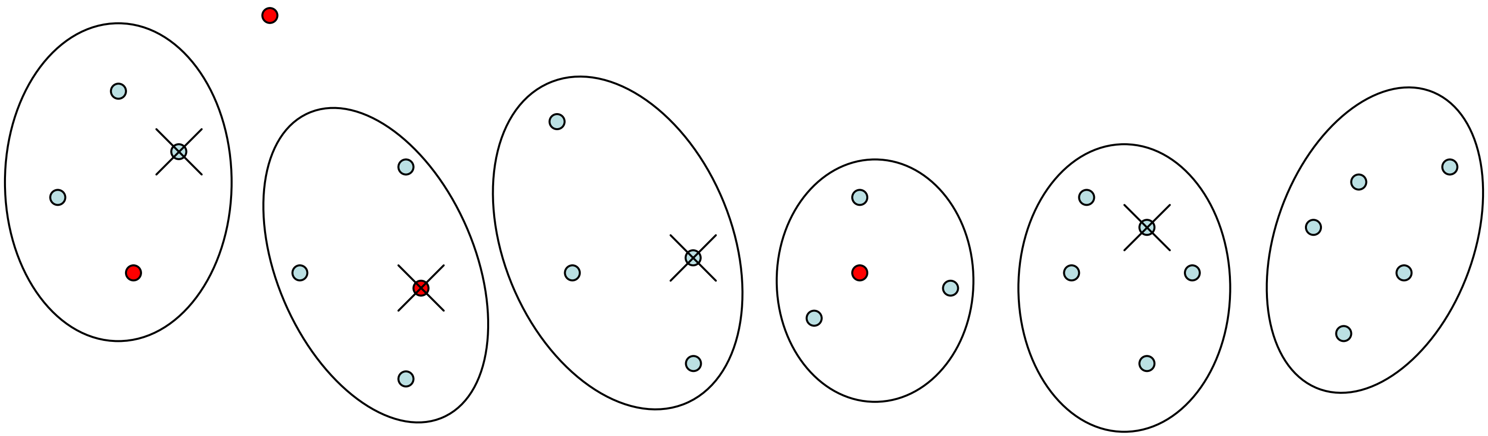
- Pick a $h \notin I_0$.
- Divide the remaining correspondences randomly into disjoint sets H_i.

Basic Method



- Pick a $h \notin I_0$.
- Divide the remaining correspondences randomly into disjoint sets H_i.
- Try consistency for sets $\{h\} \cup H_i$

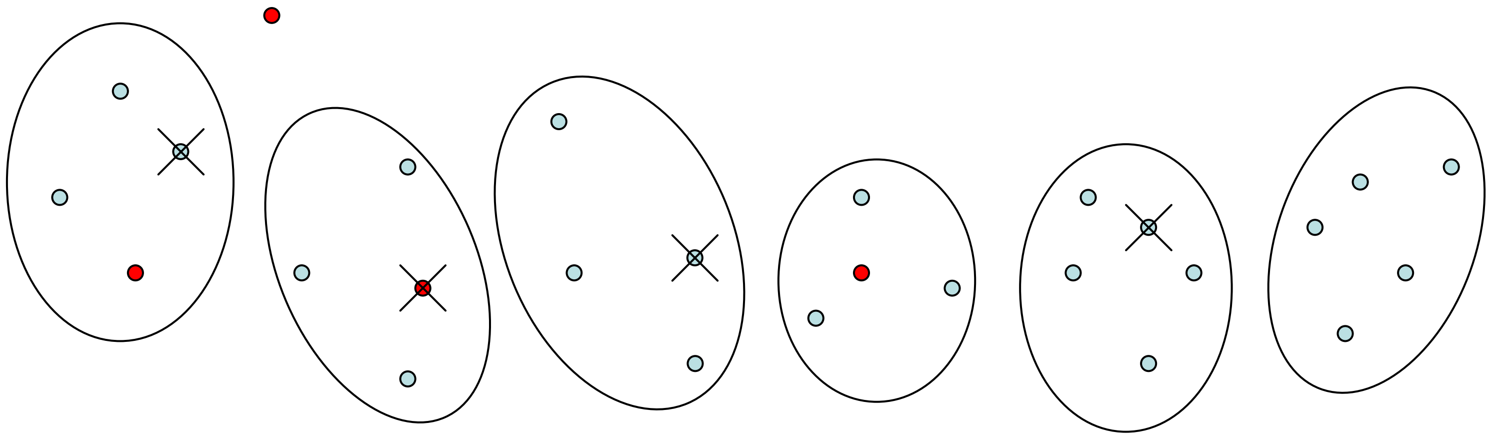
Basic Method



- Assume that K inconsistencies are found.
  Then $h \in I_* \Rightarrow |I_*| \leq N - K$

I_0 - candidate solution
I_* - hypothetic better solution
N - number of correspondences

Basic Method



- Assume that K inconsistencies are found. Then $h \in I_* \Rightarrow |I_*| \leq N - K$
- If $N - K < |I_0|$, remove h permanently and update N.

I_0 - candidate solution
I_* - hypothetic better solution
N - number of correspondences

Limitations

- Test sets $\{h\} \cup H_i$'s must be large enough, otherwise exact solutions exist.

Solution. Divide into large sets H_i.
Perform several tests for each H_i.

Refined Method

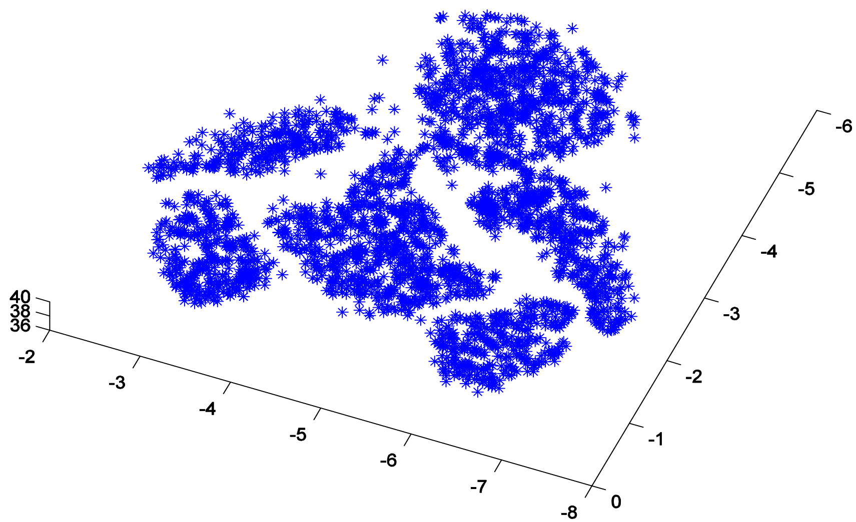
For a set H_i of size n form all subsets H_{ij} of size $k < n$.

Test consistency for sets $\{h\} \cup H_{ij}$.

- If all tested sets are inconsistent $h \in I_*$ implies that $|I_* \cap H_i| < k$

- If less than $\binom{k+f}{f}$ are consistent $h \in I_*$ implies that $|I_* \cap H_i| < k + f$

Results



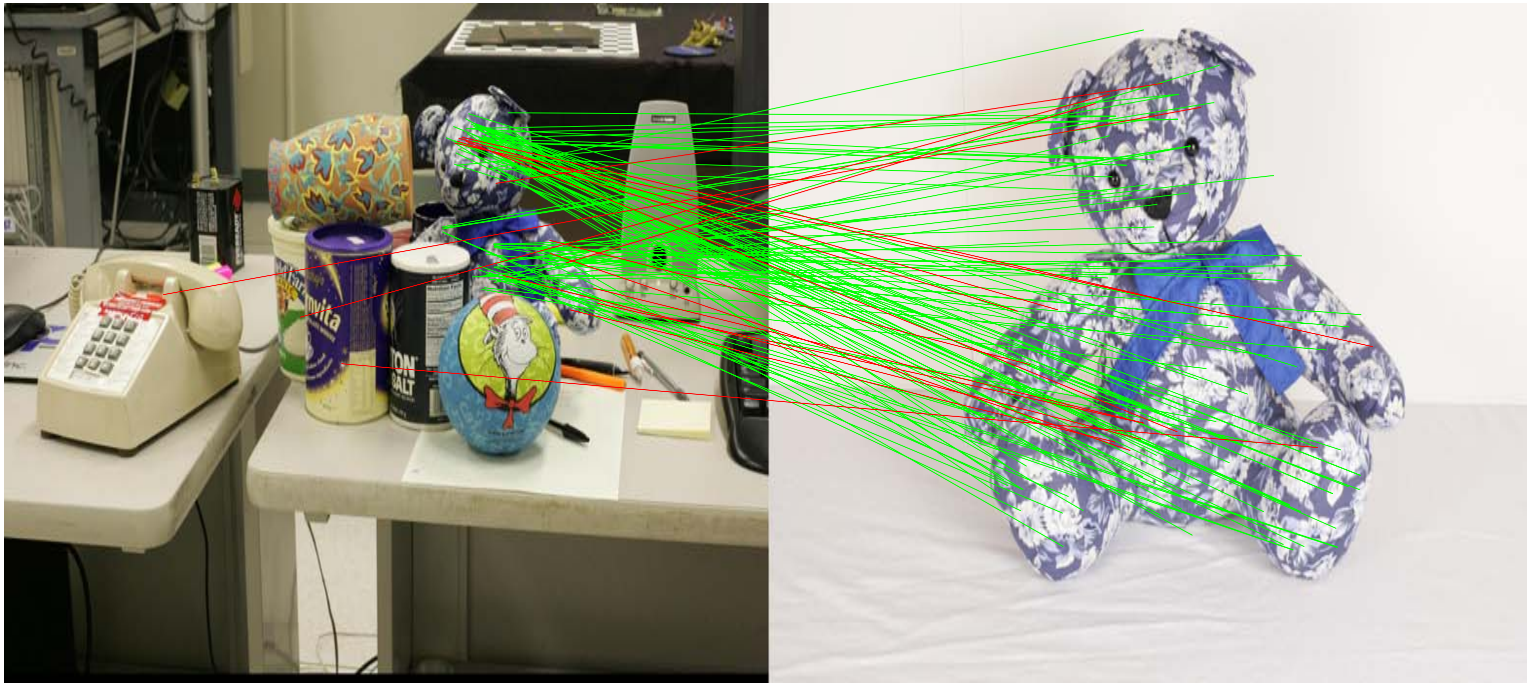
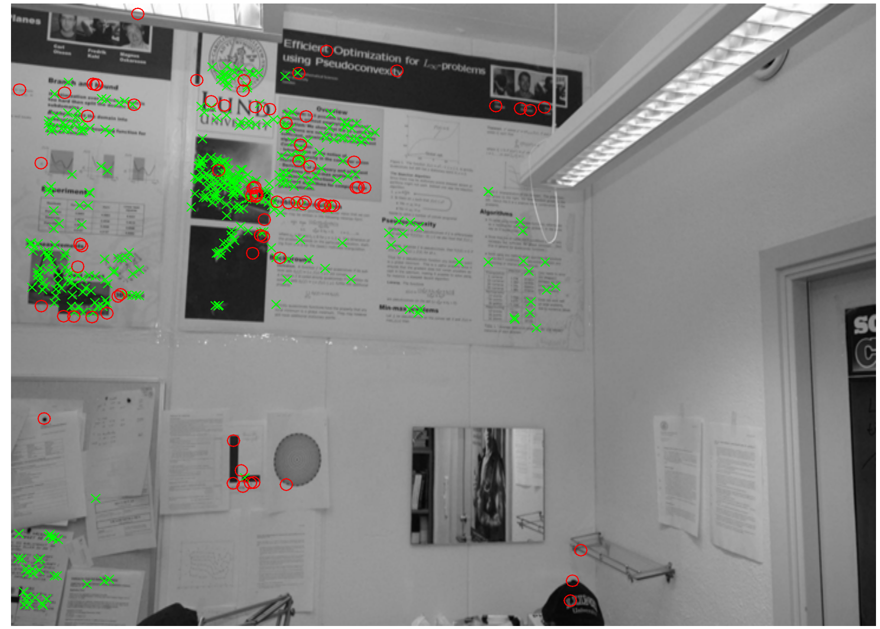


Image id	nr. corr $\|H\|$	Inliers RANSAC	Inliers Loc.	Number of tests to verify optimum	Number of tests in Li-CVPR2007 (* = worst case bound)
1	353	264	270	10276	$4.0 \cdot 10^{21*}$
2	121	104	105	420	$9.3 \cdot 10^{13*}$
3	69	49	50	1171	$5.6 \cdot 10^{14*}$
4	86	55	57	800	$5.6 \cdot 10^{14*}$
5	150	114	116	834	$6.5 \cdot 10^{15*}$
6	65	42	50	1228	$4.8 \cdot 10^{13*}$
7	14	9	9	1001	$2.7 \cdot 10^{5*}$
8	105	87	87	418	$3.2 \cdot 10^{14*}$
9	174	147	149	718	$1.0 \cdot 10^{16*}$
10	263	244	245	187	$3.2 \cdot 10^{14*}$



Stereo pair	nr. corr $\lvert H\rvert$	Inliers RANSAC	Inliers Loc.	Number of tests to verify optimum	Number of tests in Li-CVPR2007 (* = worst case bound)
1	513	430	432	5889	$6.7 \cdot 10^{15*}$
2	101	57	64	9468	$4.6 \cdot 10^{13*}$


ELEMENTARY THEORY
OF ANALYTIC FUNCTIONS
OF ONE OR SEVERAL
COMPLEX VARIABLES
HENRI CARTAN
KOSSAN

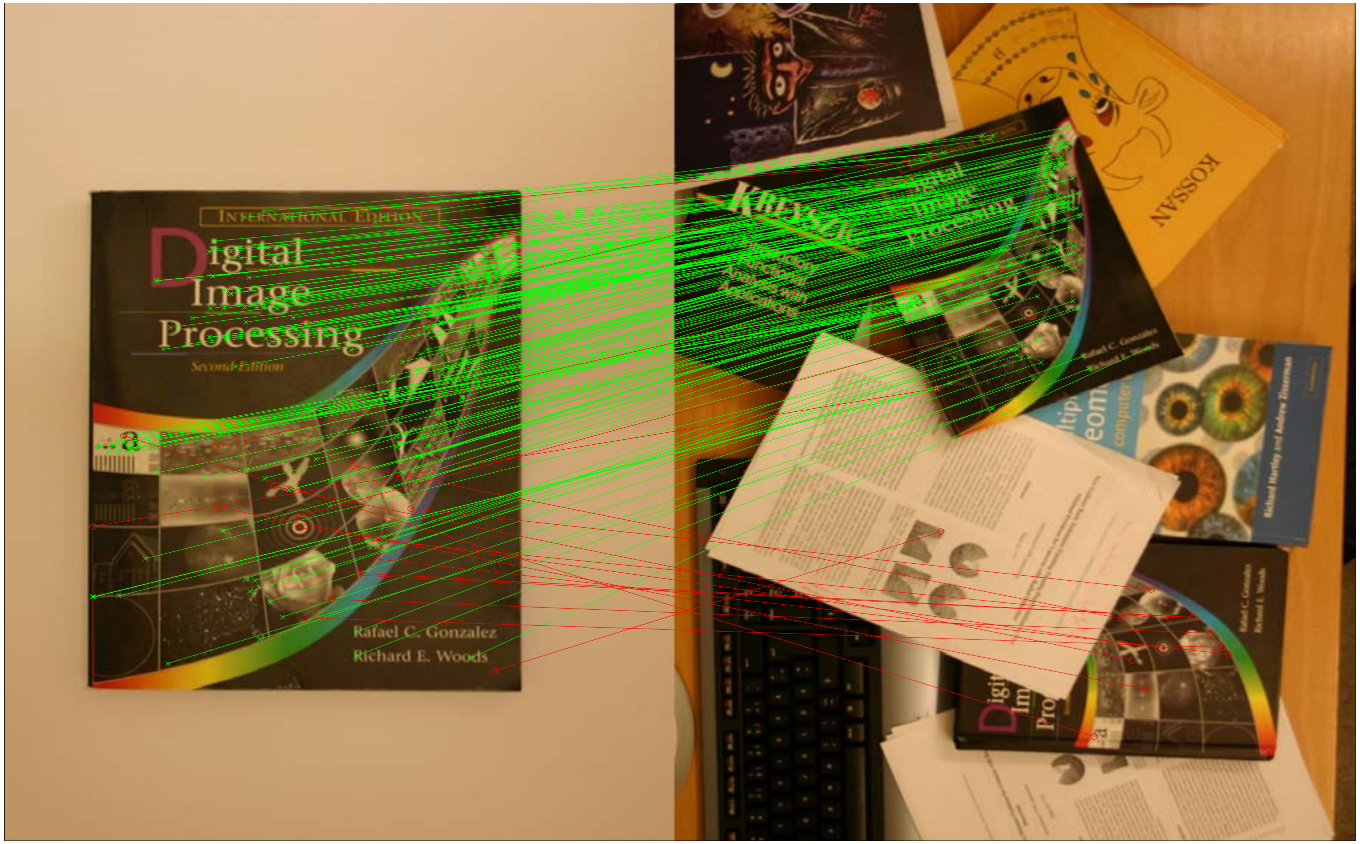
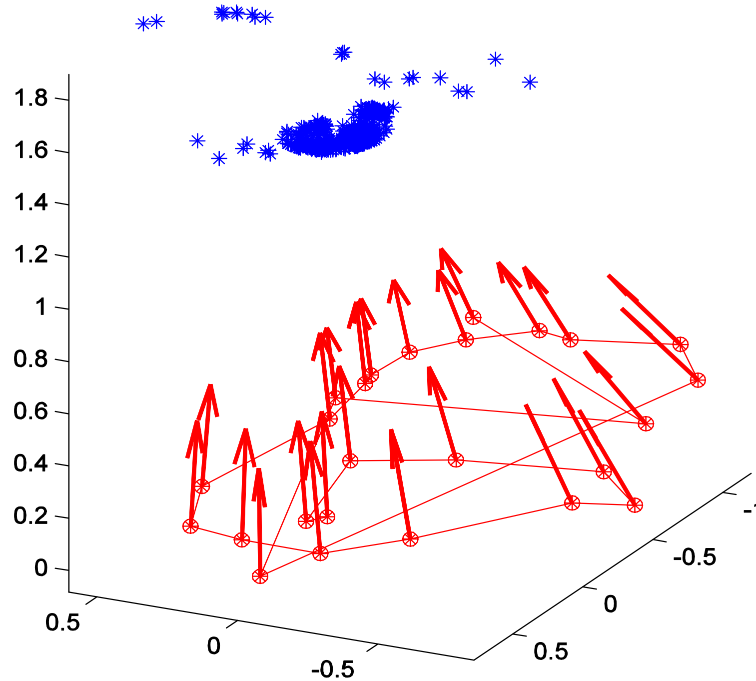

INTERNATIONAL EDITION
Digital Image Processing
Second Edition
Rafael C. Gonzalez
Richard E. Woods
KOSSAN

Image id	nr. corr $\|H\|$	Inliers RANSAC	Inliers Loc.	Number of tests to verify optimum	Number of tests in Li-CVPR2007 (* = worst case bound)
1	217	194	199	396	$3.5 \cdot 10^{5*}$
2	67	56	59	102	$4.9 \cdot 10^{5*}$
3	76	70	71	20	574
4	74	66	66	42	11177
5	77	46	46	4563	$2.8 \cdot 10^{6*}$
6	146	43	< 73	18335	$3.2 \cdot 10^{8*}$



1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.5
0
-0.5
-1
-0.5
0
0.5

Literature

Outliers for Quasiconvex Problems:

- K. Sim, R. Hartley. Removing Outliers Using the L_∞ Norm. *CVPR*. 2006.
- H. Li. A Practical Algorithm for L_∞ Triangulation with Outliers. *CVPR*. 2007.
- C. Olsson, O. Enqvist, F. Kahl. A Polynomial-Time Bound for Matching and Reconstruction with Outliers. *CVPR*. 2008.
- Q. Ke, T. Kanade. Quasiconvex Optimization for Robust Geometric Reconstruction. *PAMI*. 2008.
- Y. Seo. Non-Iterative Outlier Removal by Minimizing the Sum of Infeasibilities for L_∞ Approaches. *Unpublished*. 2008.

That's it

Thank you

A Randomized Algorithm (recursive version)

H - set of points. C - some basis. Call: $LP(H, C)$.

function $LP(H, C)$
  if $H = C$ then
    return C
  else
    choose random $h \in H \setminus C$
    $B := LP(H \setminus \{h\}, C)$
    if h violates B then
      return $LP(H, basis(B \cup h))$
    else
      return B

Note: Randomization independent of input data.

Time complexity is $O(n)$!