L_∞ Optimization and Quasi-convex Optimization

Fredrik Kahl

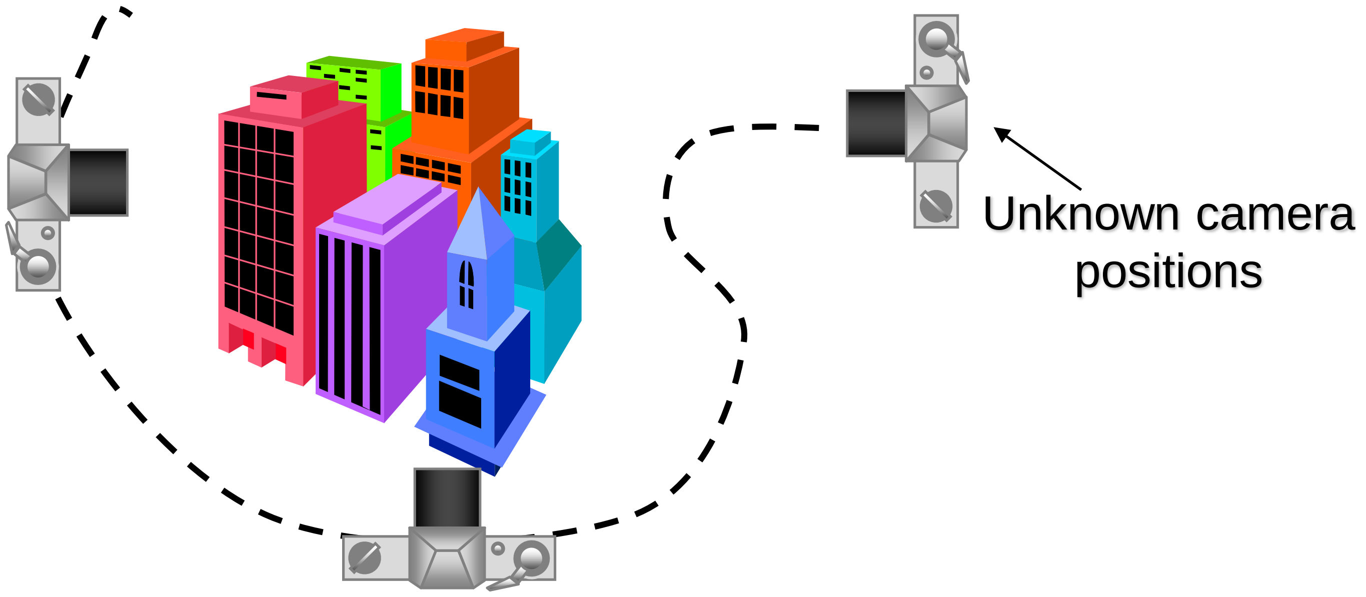
Chalmers, Sweden

L_∞ Optimization and Quasi-convex Optimization

PART 1: Outline

- Introduction: why L_∞ optimization?
- Convexity and quasi-convexity
- Examples
- Globally optimal algorithms
- Outliers and quasi-convexity

Multiple View Geometry and Geometric Reconstruction Problems



- Given images, reconstruct:
  - Scene geometry (structure)
  - Camera positions (motion)

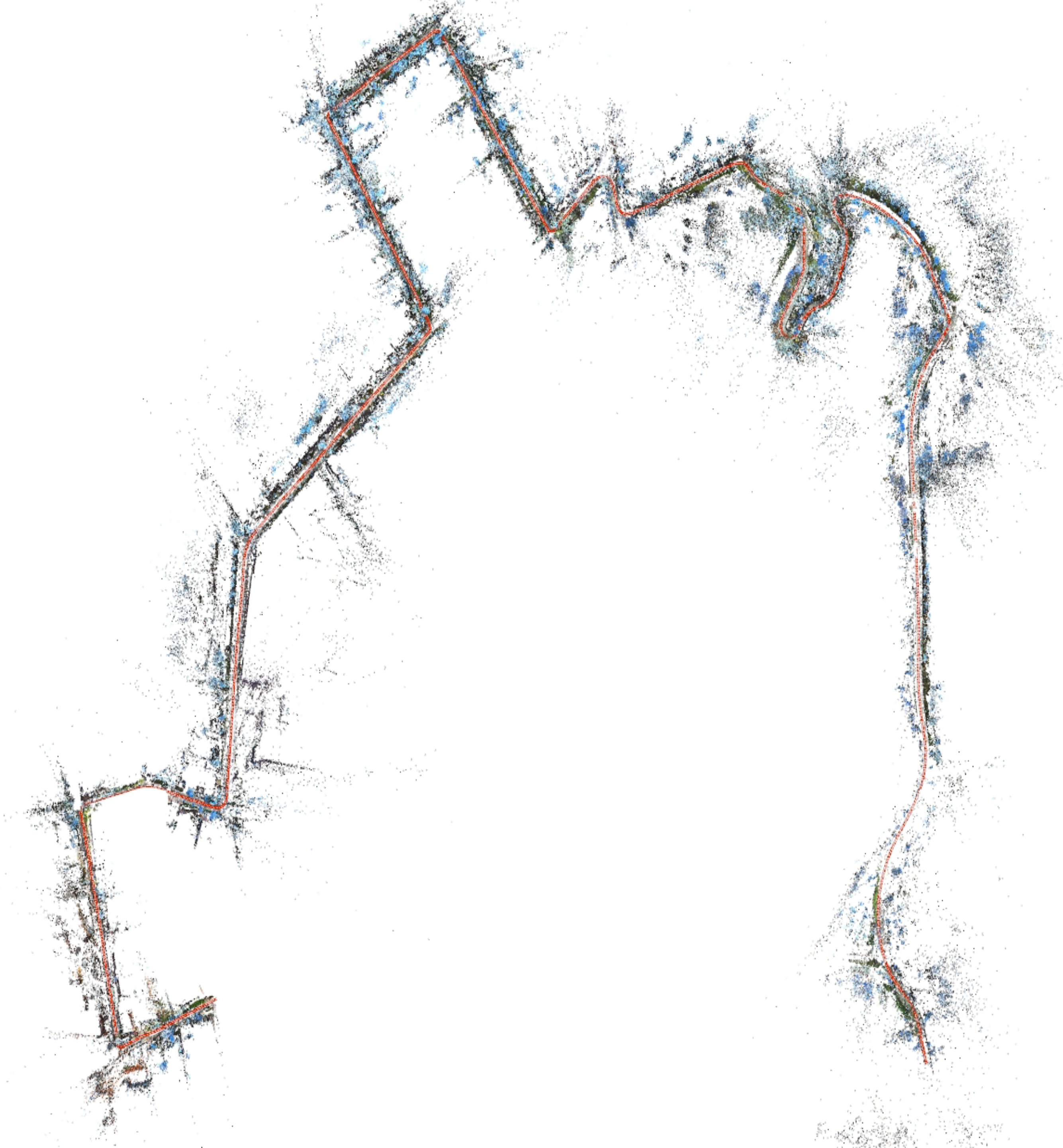
Benchmarking Long-Term Localization

High-quality night-time images

Seasonal changes, urban; Low-quality night-time images

Seasonal changes, (sub)urban

Sattler, Maddern, Toft, Torii, Hammarstrand, Stenborg, Safari, Okutomi, Pollefeys, Sivic, Kahl, Pajdla
Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, **CVPR 2018**



Multi-view optimization methods

- Algebraic cost-functions
  - Example: 8-point algorithm by Longuet-Higgins
  - Example: DLT for absolute pose

  \+ Simple and fast

  \- Unstable. Cost function makes no sense.

- Minimal solvers
  - Example: 5-point algorithm by Nistér

  \+ Fast and good for robust estimation (RANSAC)

  \- Only small dimensional problems

Multi-view optimization methods, cont'd

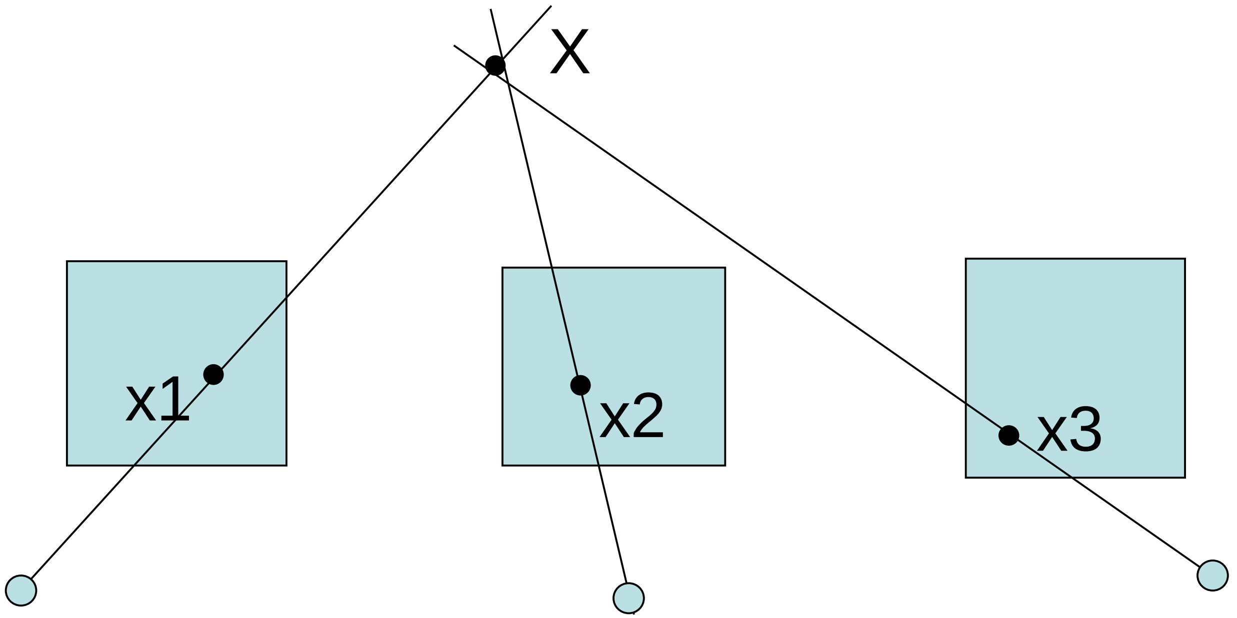
- Bundle adjustment
  - \+ Good for refinement. ML estimate.
  - \- Requires good initial solution.

- L_∞-optimization and convex optimization
  - \+ Computes globally optimal solutions
  - \+ Cost function based on reprojection errors
  - \+ Good for detecting outliers
  - \- Bad with outliers

L_∞ Optimization
and
Quasi-convex Optimization

The triangulation problem

- Given known camera positions and matched points $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2 \leftrightarrow \ldots \leftrightarrow \mathbf{x}_n$
- Find the 3D point $\mathbf{X}$ that maps to these points.

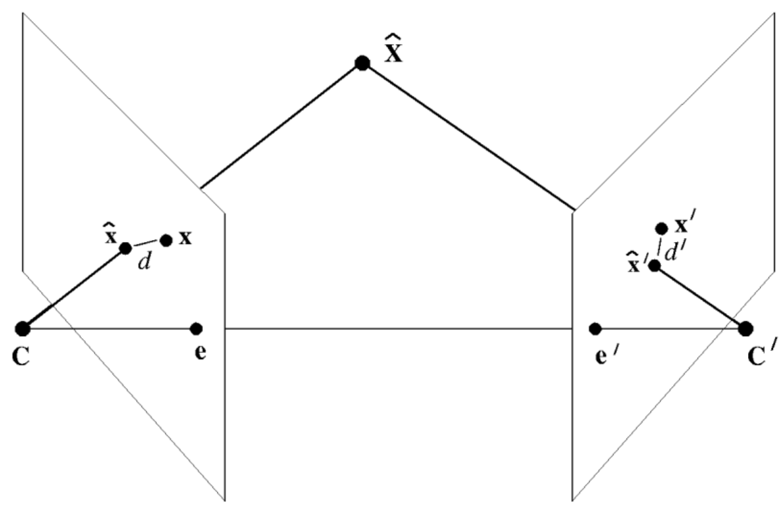


Triangulation

Triangulation :

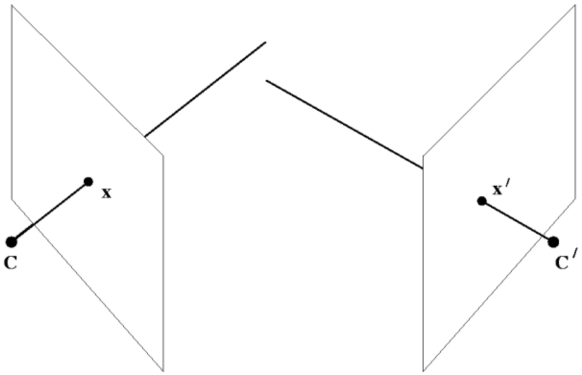
- Knowing $\mathbf{P}$ and $\mathbf{P}'$
- Knowing $\mathbf{x}$ and $\mathbf{x}'$
- Compute $\widehat{\mathrm{X}}$ such that

$$\mathbf{x} = \mathbf{P}\mathrm{X} \quad ; \quad \mathbf{x}' = \mathbf{P}'\mathrm{X}$$

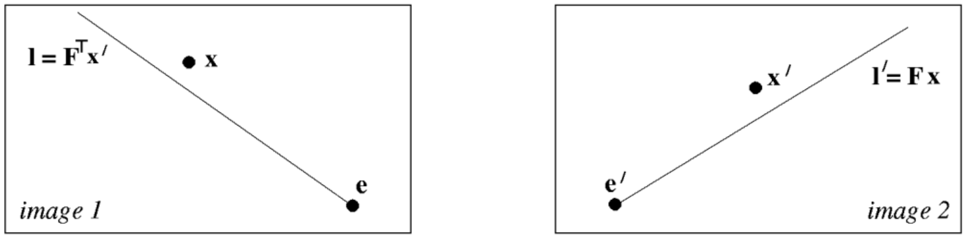


Triangulation in presence of noise

- In the presence of noise, back-projected lines do not intersect.

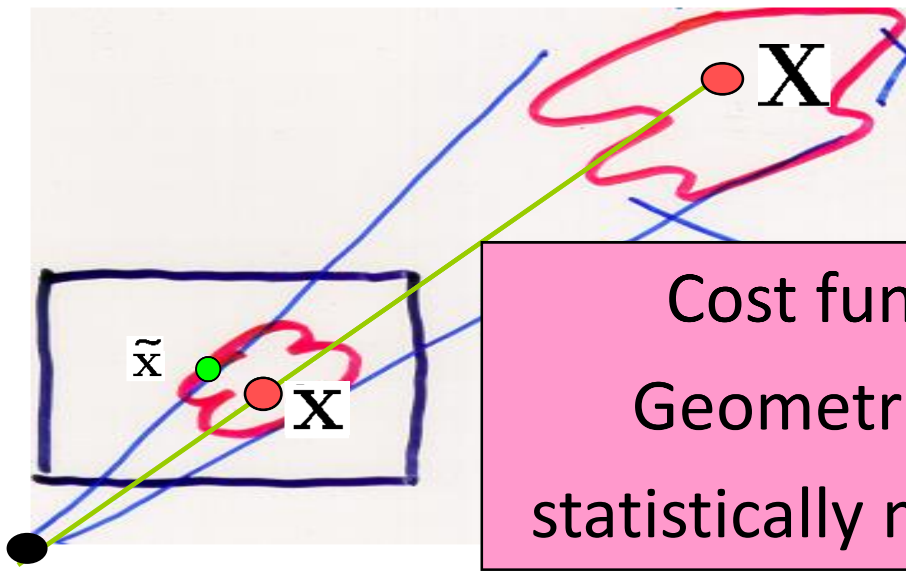


Rays do not intersect in space



Measured points do not lie on corresponding epipolar lines

Problem formulation



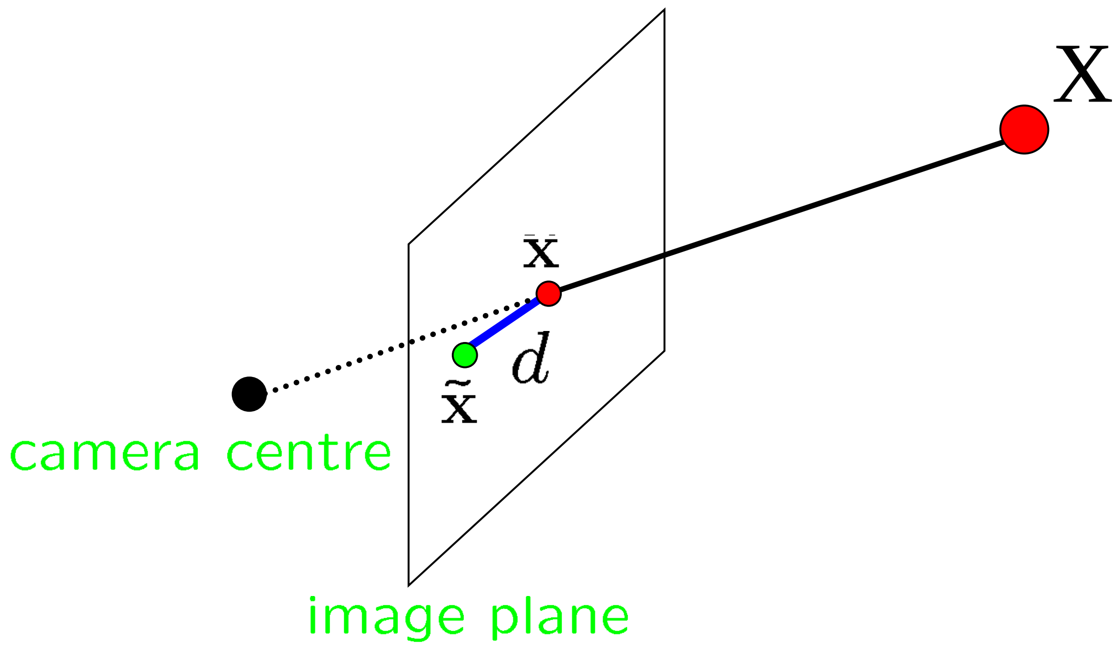
Cost function:
Geometrically &
statistically meaningful

$$\min \left\| \begin{bmatrix} \vdots \\ d(\tilde{\mathbf{x}}, \mathbf{x}) \\ \vdots \end{bmatrix} \right\|_{L_p}$$

measured image point

reprojected image point

Perspective cameras



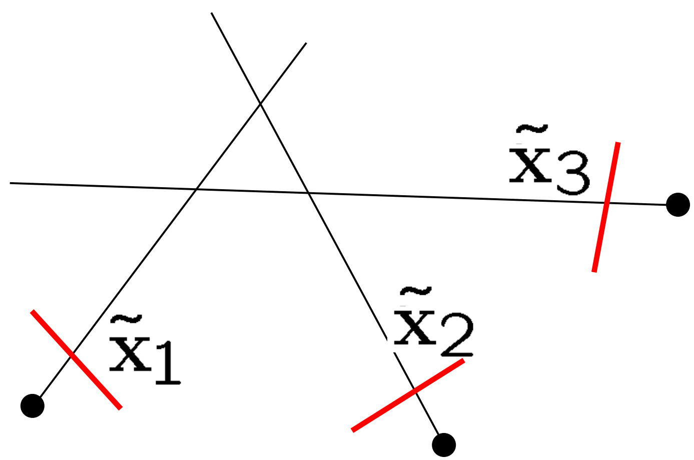
$d(\tilde{\mathbf{x}}, \mathbf{x})^2$ can be written as rational function

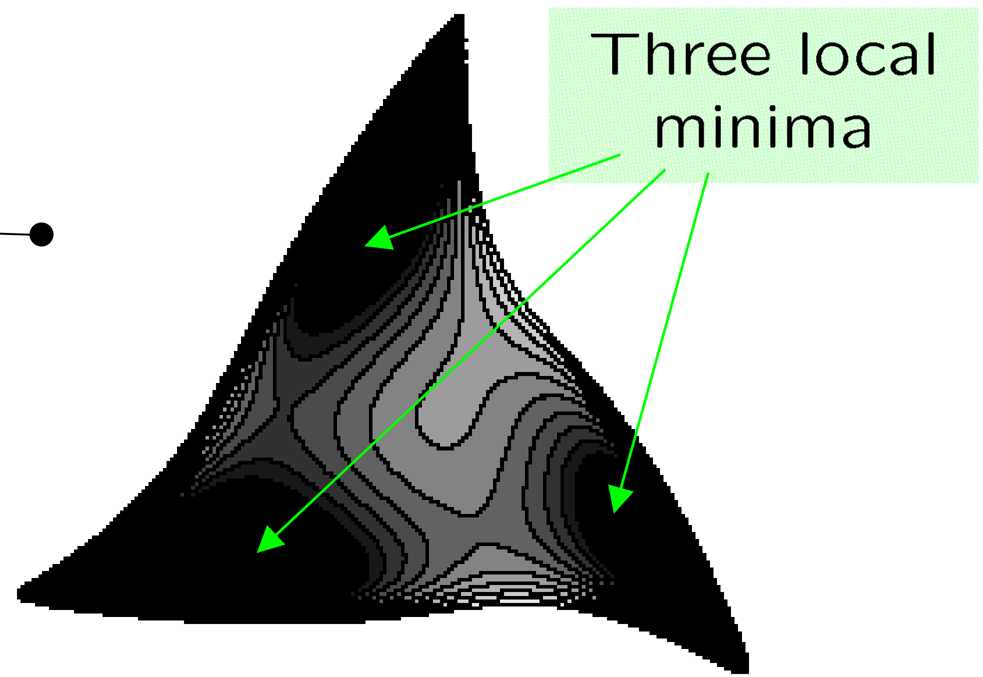
reprojected image point

measured image point

Do local minima occur?

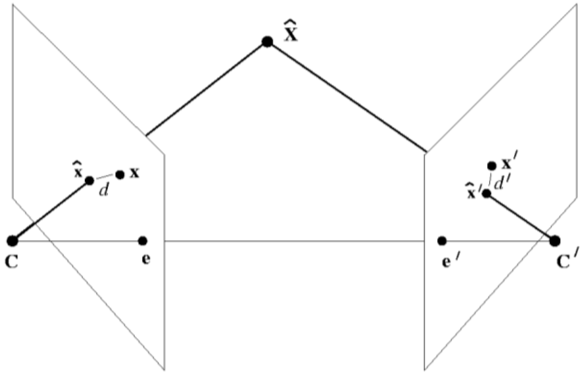
Consider the following three-view triangulation problem *in the plane*.





Contour plot of the L_2-error function

Two-view triangulation

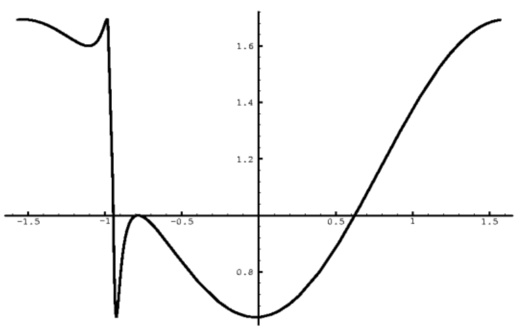


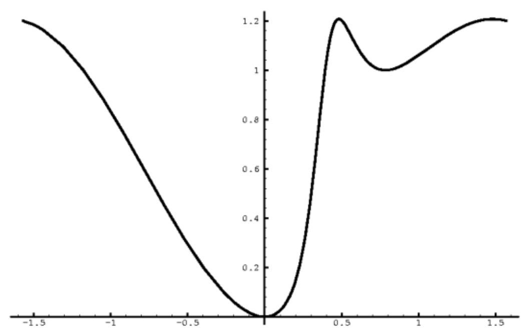
Cost function

$$\mathcal{C}(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

Multiple local minima

- Cost function may have local minima.
- Shows that gradient-descent minimization may fail.





Left : Example of a cost function with three minima.

Right : Cost function for a perfect point match with two minima.

Projective Projection

Camera matrix – 3×4 matrix

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}^{1\top} \\ \mathbf{p}^{2\top} \\ \mathbf{p}^{3\top} \end{pmatrix}$$

3D point represented in homogeneous coordinates

$$\mathbf{X} = (\mathsf{x}, \mathsf{y}, \mathsf{z}, 1)^\top$$

Projected image point is given by

$$\mathbf{x} = \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{P}\mathbf{X} = \begin{pmatrix} \mathbf{p}^{1\top}\mathbf{X}/\mathbf{p}^{3\top}\mathbf{X} \\ \mathbf{p}^{2\top}\mathbf{X}/\mathbf{p}^{3\top}\mathbf{X} \end{pmatrix}$$

Geometric Projection Error

Given point $\mathbf{X}$ that should map to image point $\mathbf{x}$, error is

$$
\begin{aligned}
d(\mathbf{x}, \mathbf{P}\mathbf{X})^2 &= \left(u - \frac{\mathbf{p}^{1\top}\mathbf{X}}{\mathbf{p}^{3\top}\mathbf{X}}\right)^2 + \left(v - \frac{\mathbf{p}^{2\top}\mathbf{X}}{\mathbf{p}^{3\top}\mathbf{X}}\right)^2 \\
&= \left(\frac{u(\mathbf{p}^{3\top}\mathbf{X}) - \mathbf{p}^{1\top}\mathbf{X}}{\mathbf{p}^{3\top}\mathbf{X}}\right)^2 + \left(\frac{v(\mathbf{p}^{3\top}\mathbf{X}) - \mathbf{p}^{2\top}\mathbf{X}}{\mathbf{p}^{3\top}\mathbf{X}}\right)^2 \\
&= \frac{f(\mathbf{X})^2 + g(\mathbf{X})^2}{\lambda(\mathbf{X})^2}
\end{aligned}
$$

- All of f, g and λ are linear in (x, y, z)

Error to be minimized is

$$
\sum_{i=1}^{N} d(\mathbf{x}, \mathbf{P}_i\mathbf{X})^2 = \sum_{i=1}^{N} \frac{f_i(\mathbf{X})^2 + g_i(\mathbf{X})^2}{\lambda_i(\mathbf{X})^2}
$$

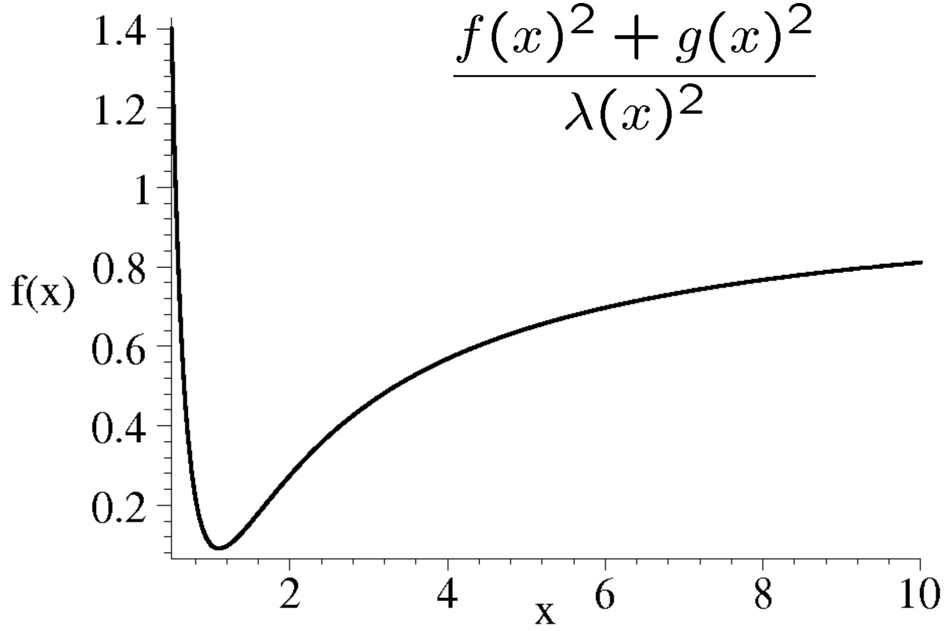
The main difficulty

- Function

$$\boxed{\frac{f_i(\mathbf{X})^2 + g_i(\mathbf{X})^2}{\lambda_i(\mathbf{X})^2}}$$

  is not convex.
- Sum of non-convex functions can have several minima.

Triangulation cost – cross-section

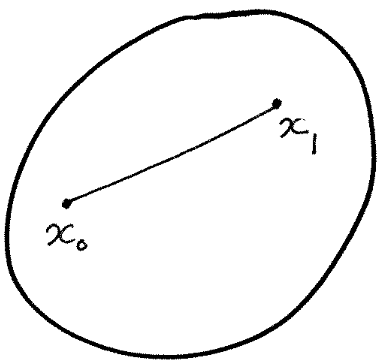


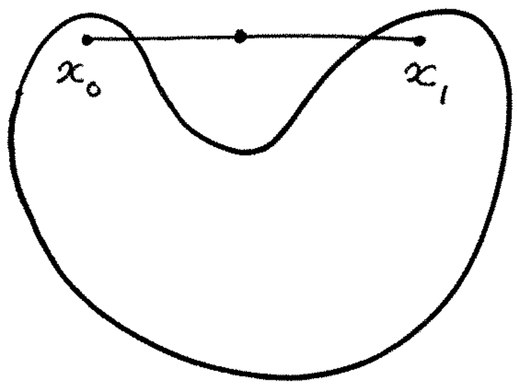
Convex Optimization

— convex and quasi-convex functions, convex cones

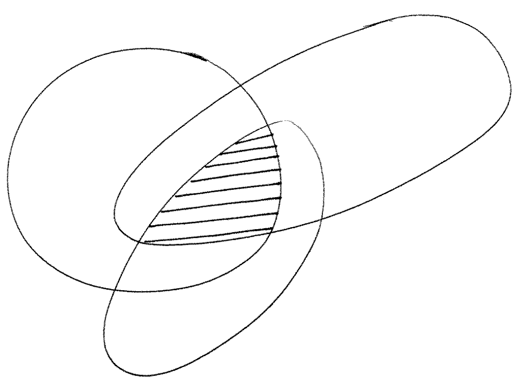
Convex set

A set $D \subset R^n$ is convex if the line joining points $\mathbf{x}_0$ and $\mathbf{x}_1$ lies inside D.





Intersection of convex sets in convex.



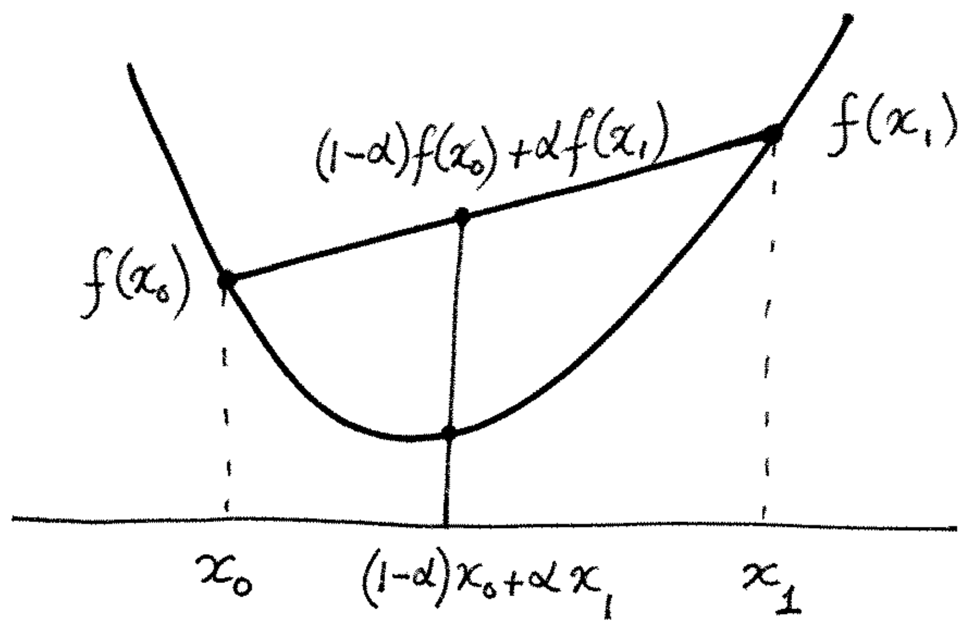
Convex function

D – a domain in $\mathbb{R}^n$.

A convex function $f : D \to \mathbb{R}$ is one that satisfies, for any $\mathbf{x}_0$ and $\mathbf{x}_1$ in D:

$$f((1-\lambda)\mathbf{x}_0 + \lambda\mathbf{x}_1) \leq (1-\lambda)f(\mathbf{x}_0) + \lambda f(\mathbf{x}_1) \quad .$$

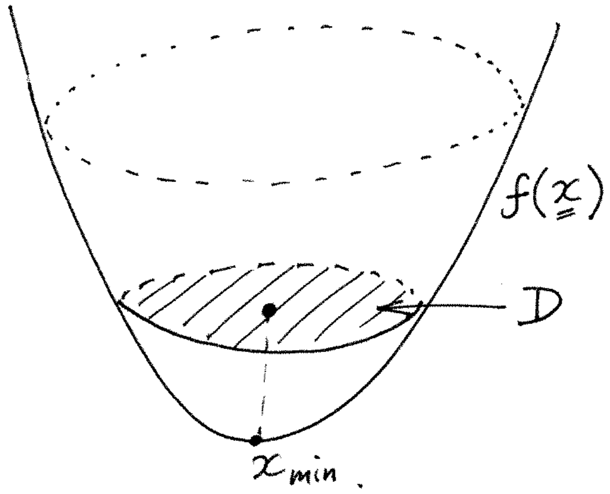
Line joining $(\mathbf{x}_0, f(\mathbf{x}_0))$ and $(\mathbf{x}_1, f(\mathbf{x}_1))$ lies above the function graph.



Convex optimization

The generic convex optimization problem is:

Minimize the convex function $f(\mathbf{x})$ over a convex set D.



Properties:

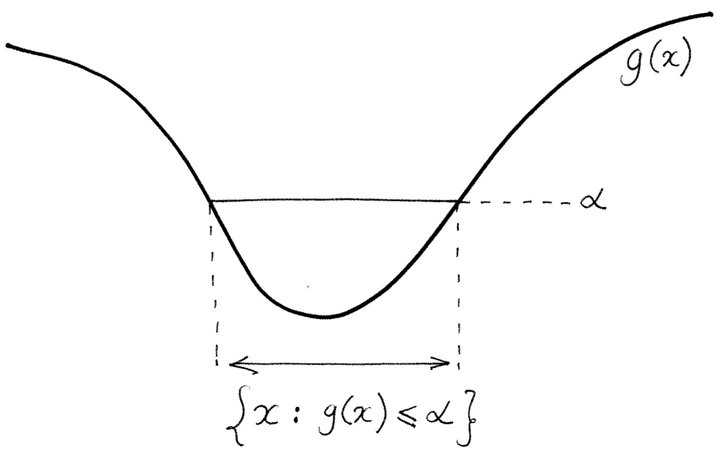
1. A local minimum is also a global minimum

2. The sum of convex functions is also convex

Quasi-convex function

Definition: Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the α-sublevel set of f is the set

$$S_\alpha(f) = \{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) \leq \alpha\}$$



A function is quasi-convex if all its sublevel sets are convex

- A convex function is quasi-convex, but not conversely.

Quasi-convex optimization problem

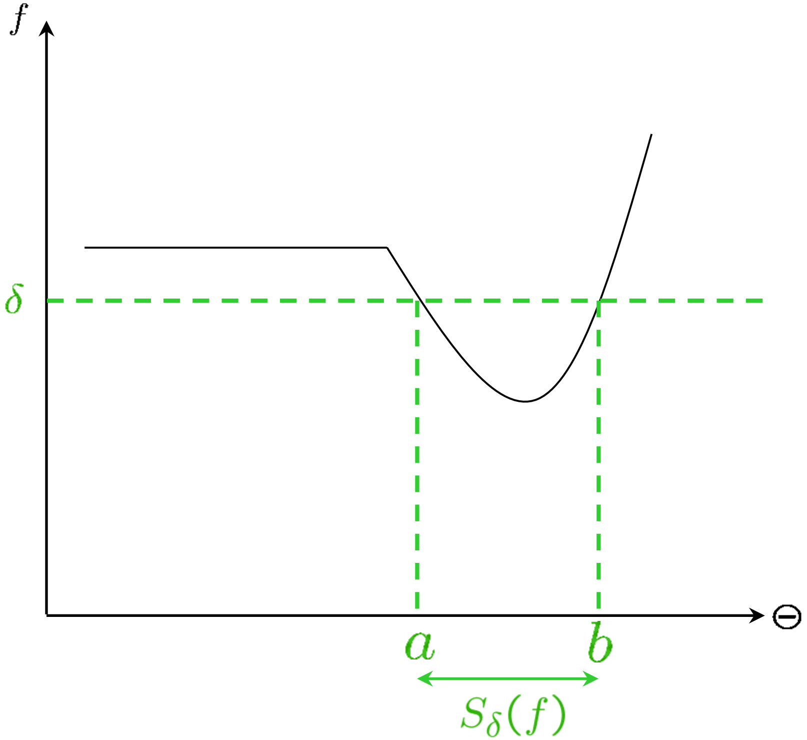
$$\begin{array}{rl} \text{Minimize} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 \end{array}$$

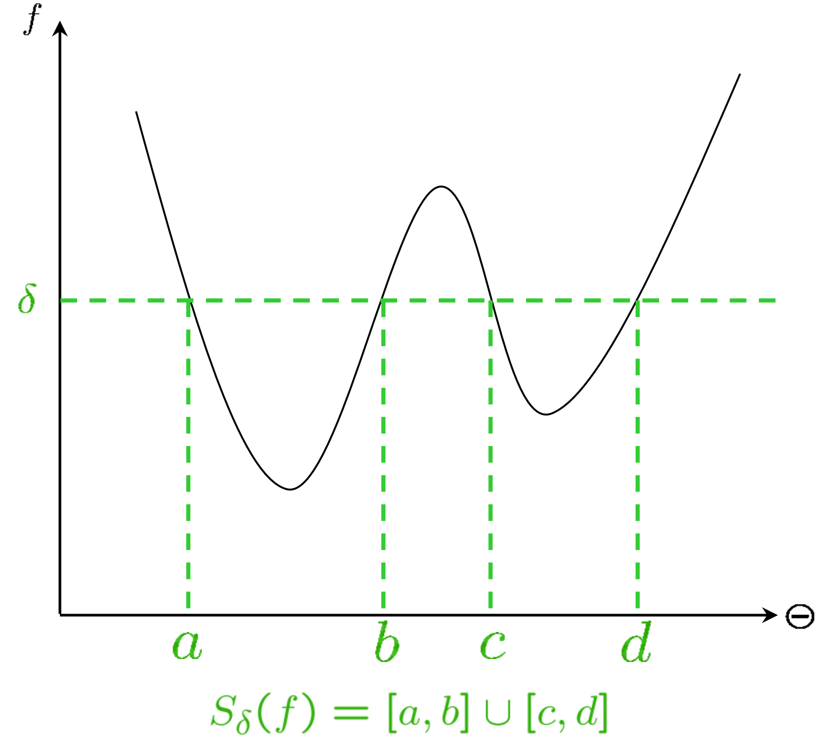
where

- f is quasi-convex.
- g_i is convex for all i.

Quasiconvex Functions

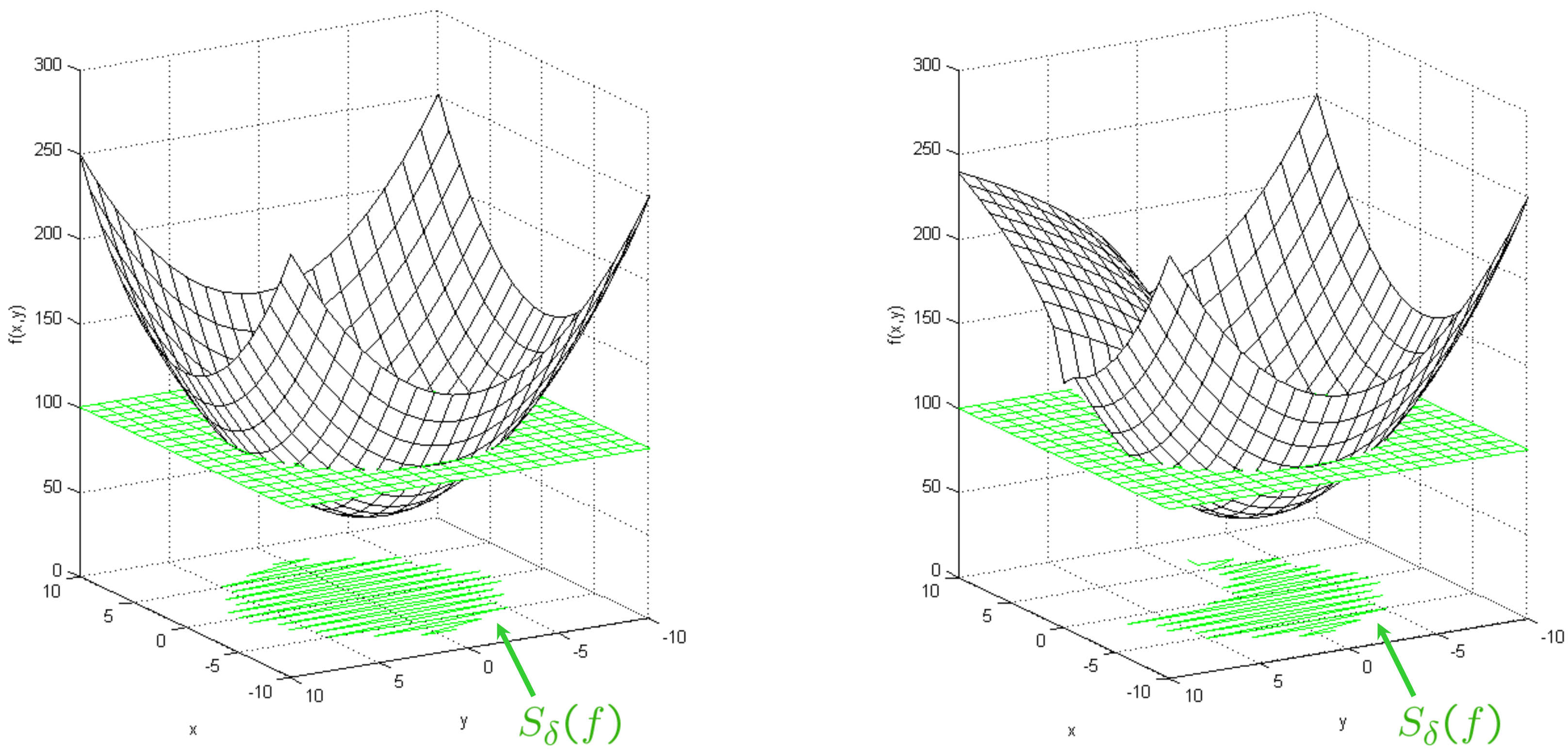
Sublevel Sets: $S_\delta(f) = \{\Theta | f(\Theta) \leq \delta\}$





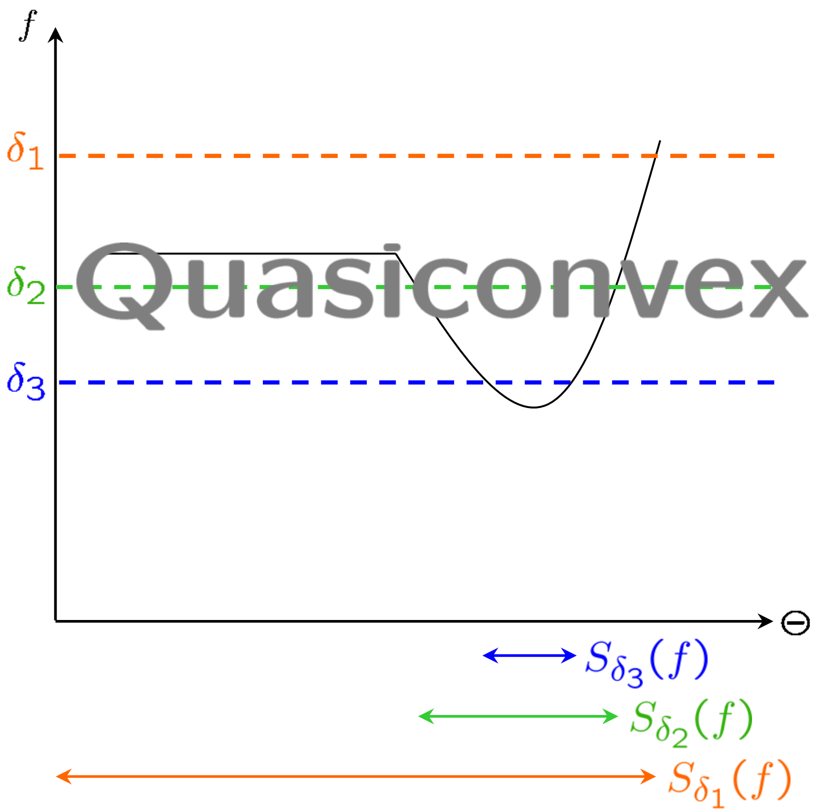
Quasiconvex Functions

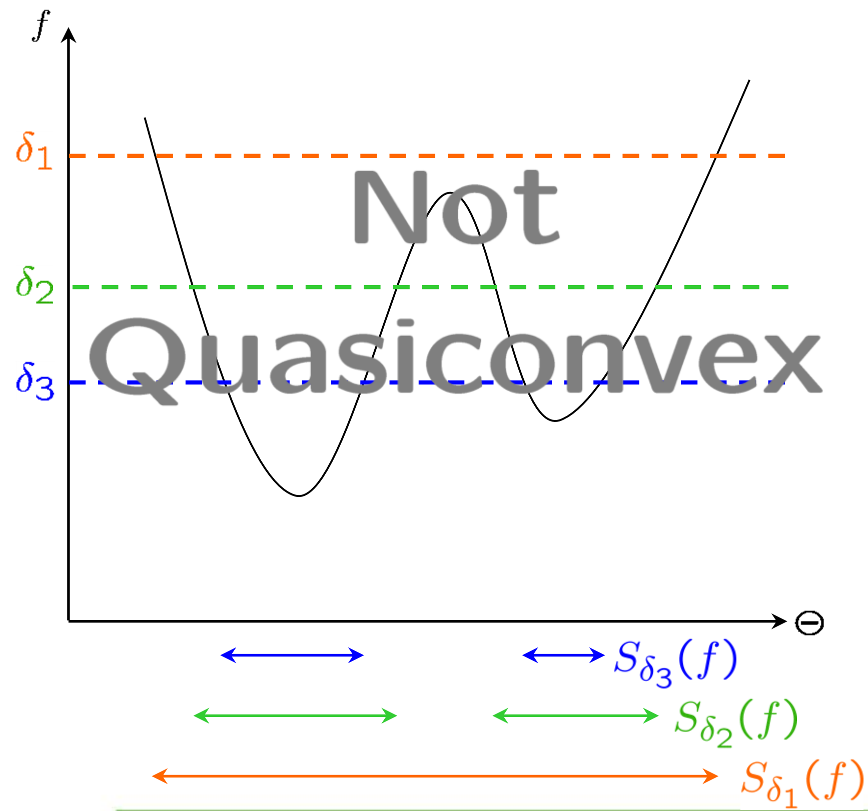
Sublevel Sets: $S_\delta(f) = \{\Theta | f(\Theta) \leq \delta\}$



Quasiconvex Functions

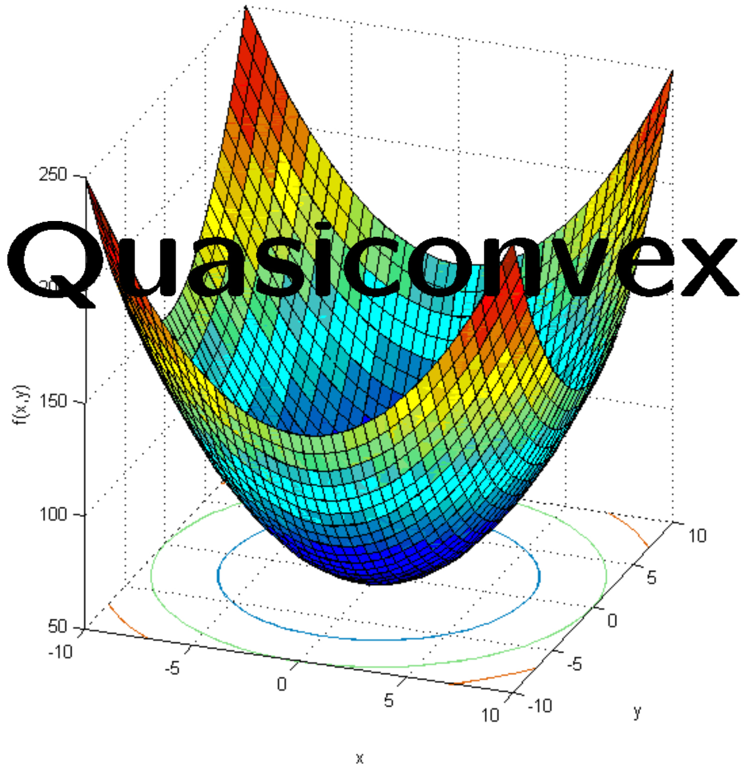
f is **quasiconvex** if its sublevel sets $S_\delta(f)$ are convex $\forall\, \delta$.

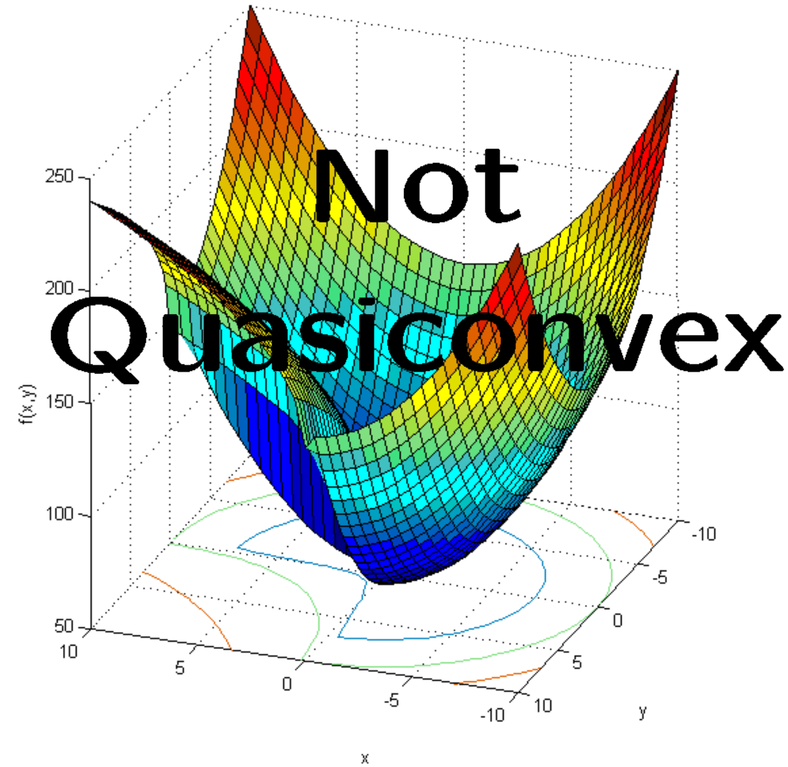




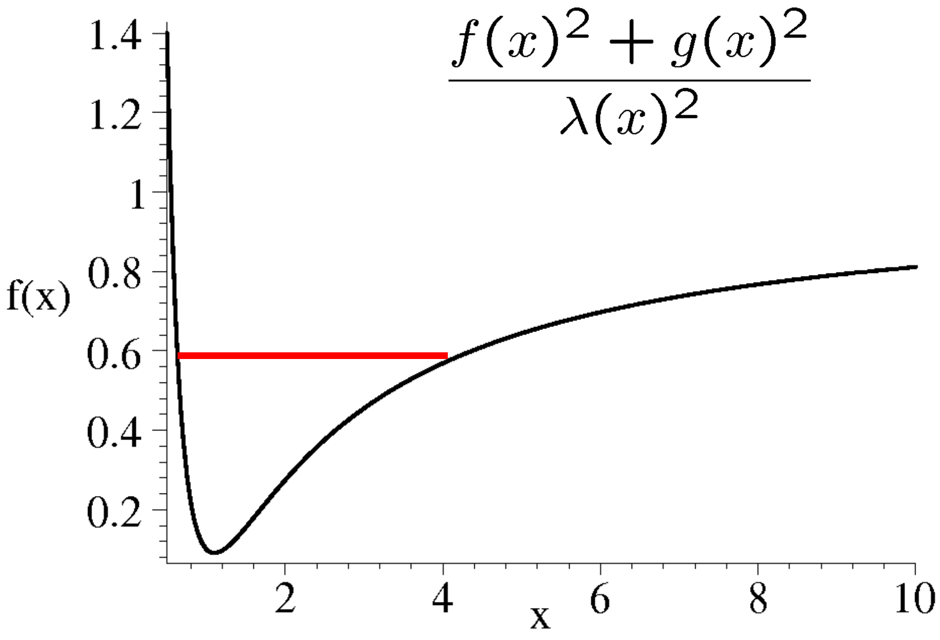
Quasiconvex Functions

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





Quasiconvex



Sublevel set is convex – function is quasi-convex

Maximum of quasi-convex functions

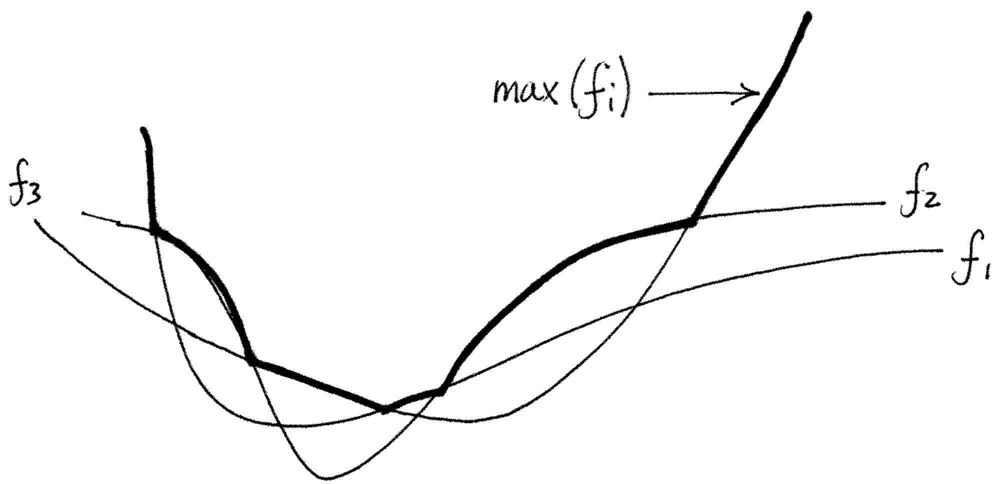
If functions f_i, $i = 1, \ldots, r$ are quasi-convex, then the function

$$f(\mathbf{x}) = \max_i f_i(\mathbf{x})$$

is quasi-convex.

Proof: $S_\alpha(f) = \bigcap_i S_\alpha(f_i)$.

Intersection of convex sets is convex.



Minimax Optimization

$$\text{Find} \quad \min_{\mathbf{x}} \max_{i} f_i(\mathbf{x})$$

$$\text{Such that} \quad \mathbf{x} \in D$$

Alternatively:

$$\text{Find} \quad \min_{\mathbf{x},s} \; s$$

$$\boxed{\begin{array}{rl} \text{Subject to} & f_i(\mathbf{x}) \leq s \quad \forall i \\ \text{and} & \mathbf{x} \in D \end{array}}$$

Second Order Cone Constraints

Consider the function $C(\mathbf{x})$:

$$C(\mathbf{x}) = \frac{\|(f(\mathbf{x}), g(\mathbf{x}))\|}{\lambda(\mathbf{x})}$$

with

$$\lambda(\mathbf{x}) \geq 0$$

Quasi-convex function: Arises naturally in Vision problems.

Second Order Cone Programs (SOCP)

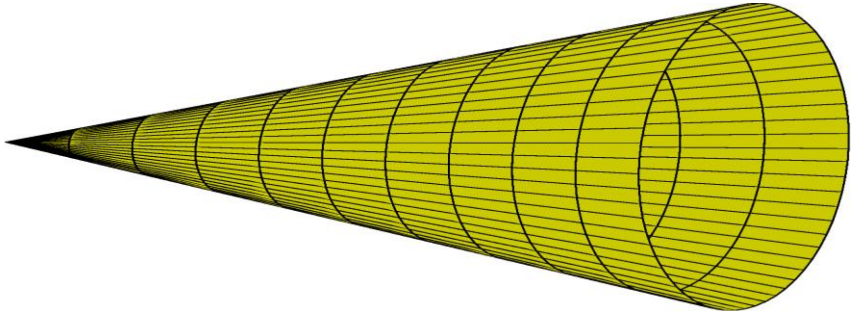
Convex problems with constraint functions

$$||A_i\mathbf{x} + \mathbf{b}_i|| \leq {c_i}^\top \mathbf{x} + d_i$$

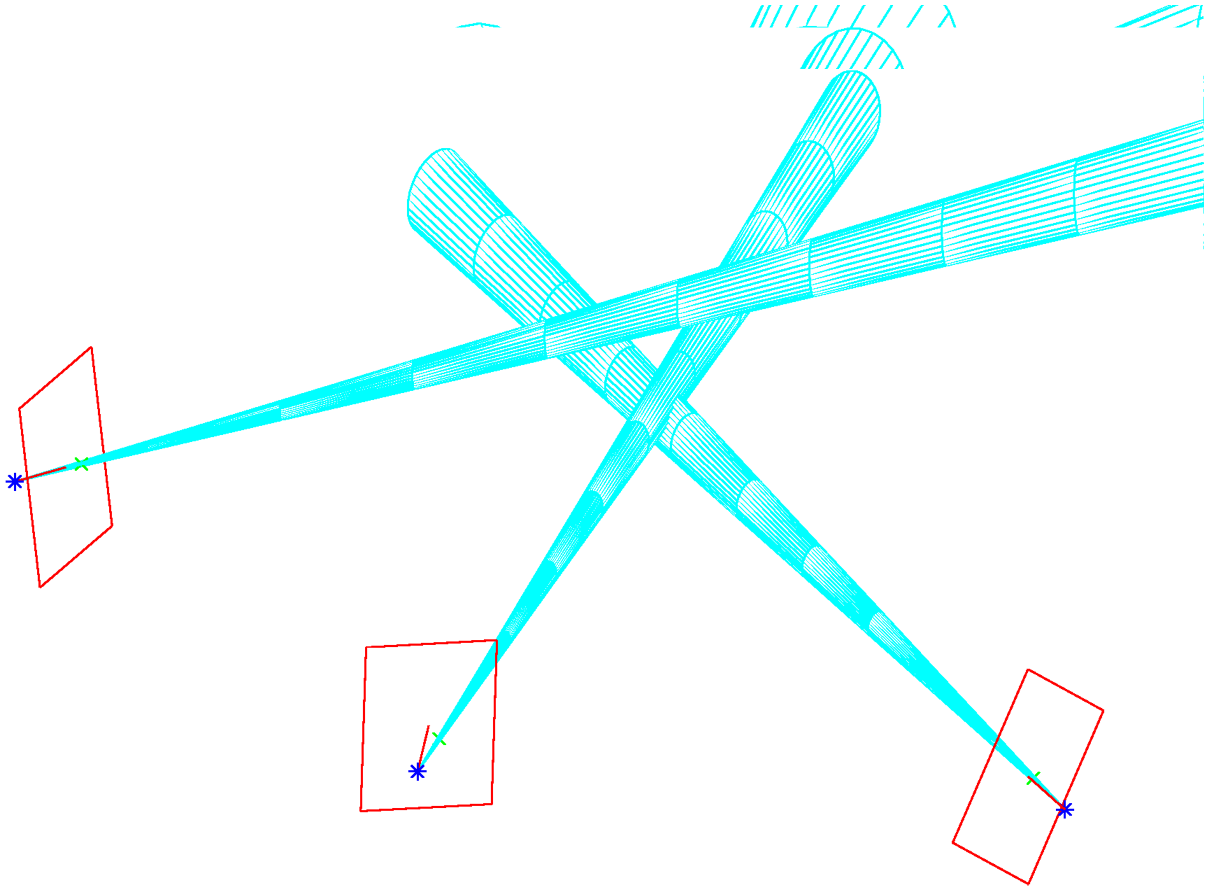
are called SOCP.

More general than LP, but less general than SDP.

SOCP is easily solvable by off-the-shelf software.



Bisection



Finding the minimum s so that all cone constraints are satisfied.

The minimax solution

1. The L_∞ solution to the triangulation problem is to find

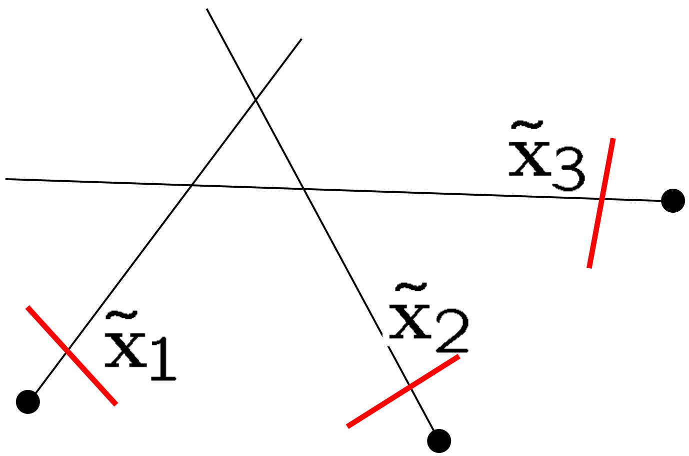
$$\min_{\mathbf{X}} \max_i d(\mathbf{x}_i, \mathbf{P}_i\mathbf{X})$$

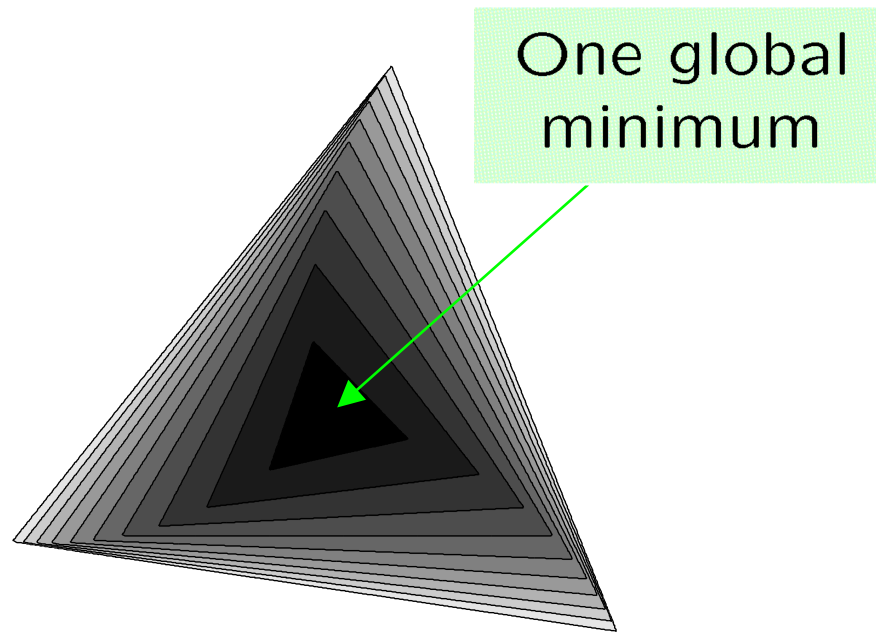
2. Function is quasi-convex on domain in front of cameras.

3. Method of solution: Second Order Cone Programming and bisection.

Do local minima occur?

Consider the following three-view triangulation problem *in the plane*.

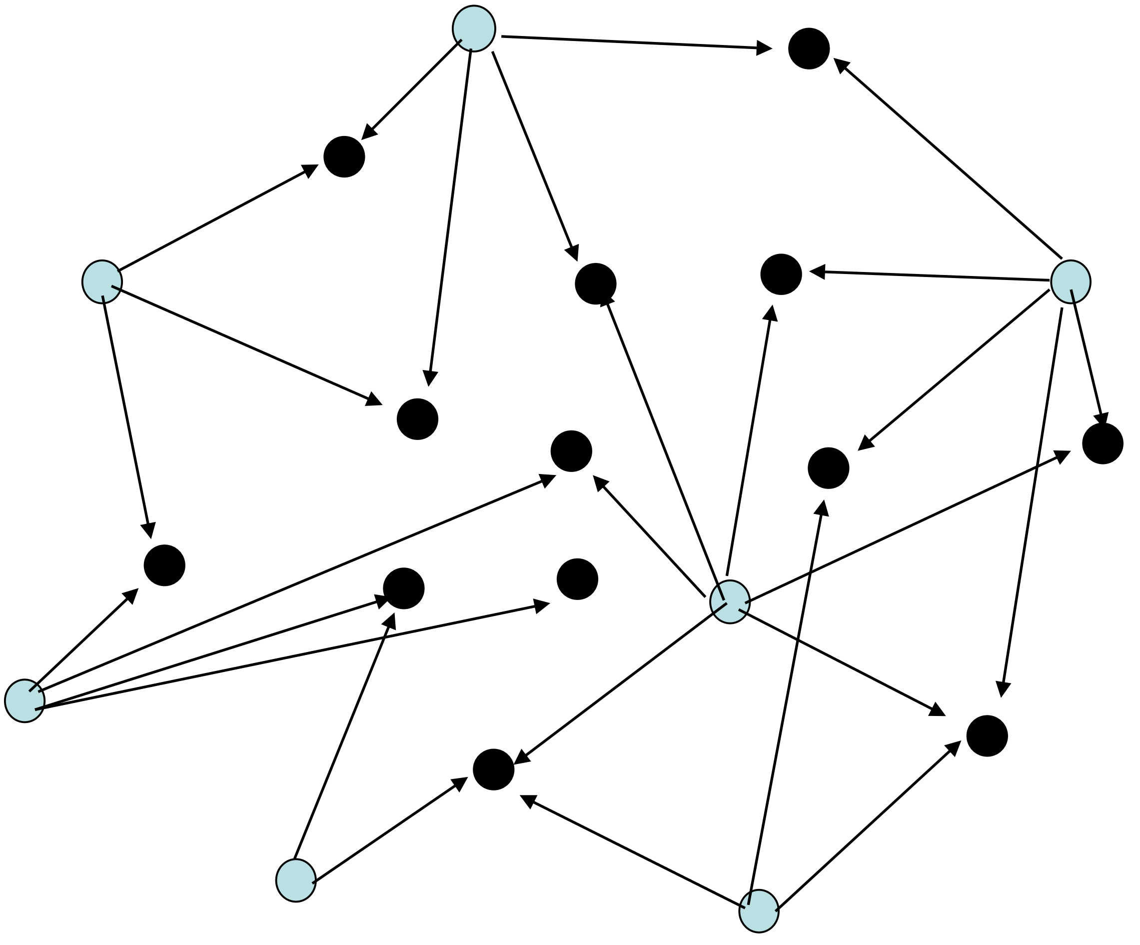




Contour plot of the L_∞-error function

Problem 2: partial structure and motion

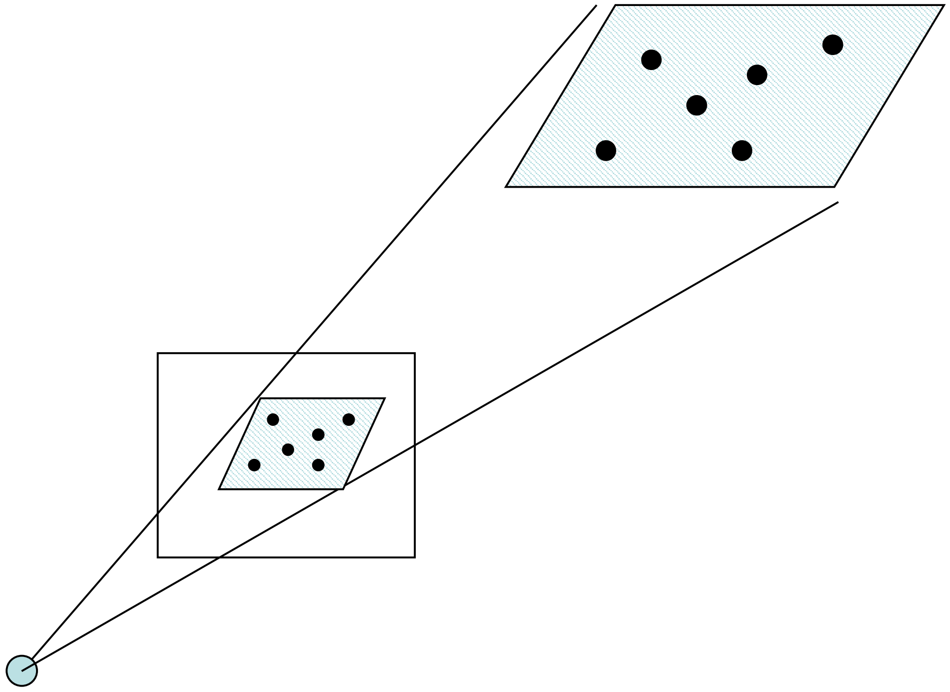
- Assume calibrated cameras.
- Assume rotations are known.
- Given points $\mathbf{x}_{ij}$ in several views, find the positions of points $\mathbf{X}$ and cameras $\mathbf{C}$ that minimize the projection errors.



Direction (unit) vectors from cameras (blue) to points (black) are given : Find the positions of the cameras and points.

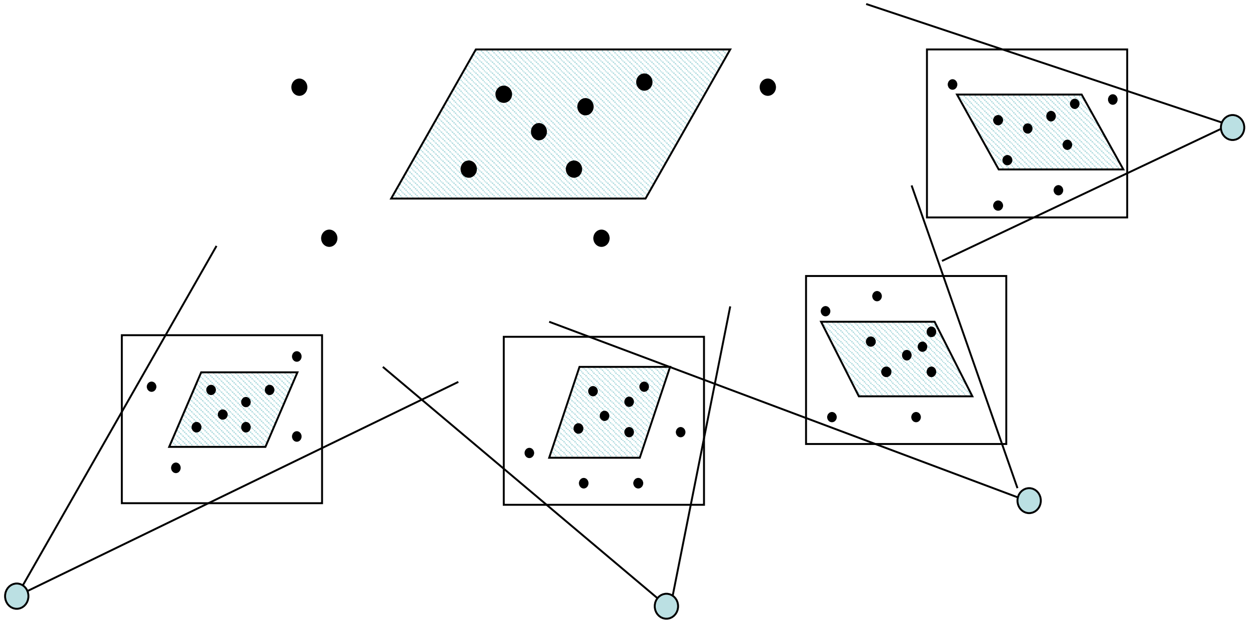
Problem 3: Homographies

- Given known 3D points $\mathbf{u}_i$ on a plane and corresponding image points $\mathbf{u}'_i$.
- Find the homography that maps corresponding points, i.e., $\mathbf{u}'_i \simeq \mathbf{H}\mathbf{u}_i$.

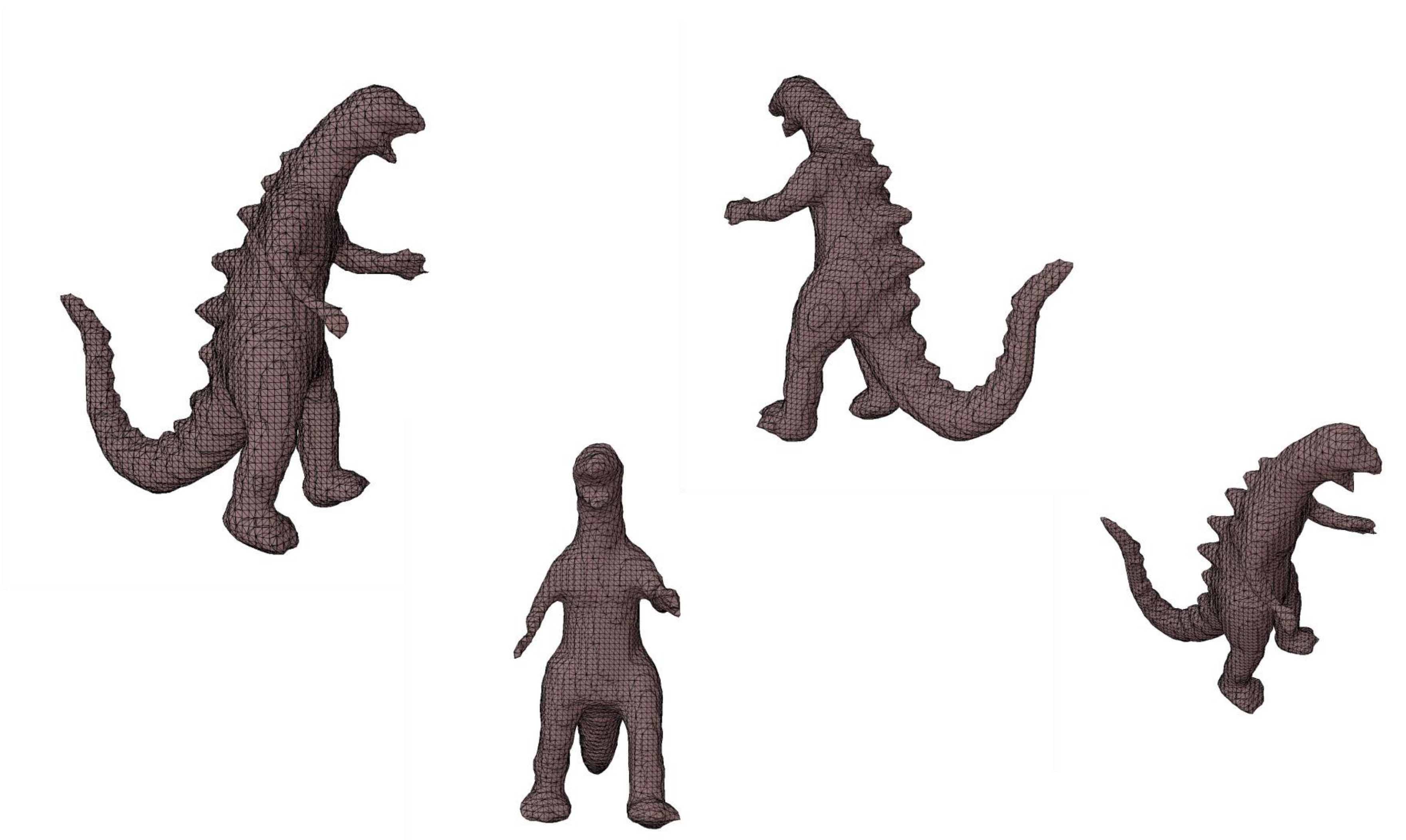


Problem 4: SfM using a Reference Plane

- Given a reference plane and corresponding image points.
- Find
  1. Interimage homographies
  2. Cameras and 3D points that map to corresponding image points.



Dinosaur Reconstruction

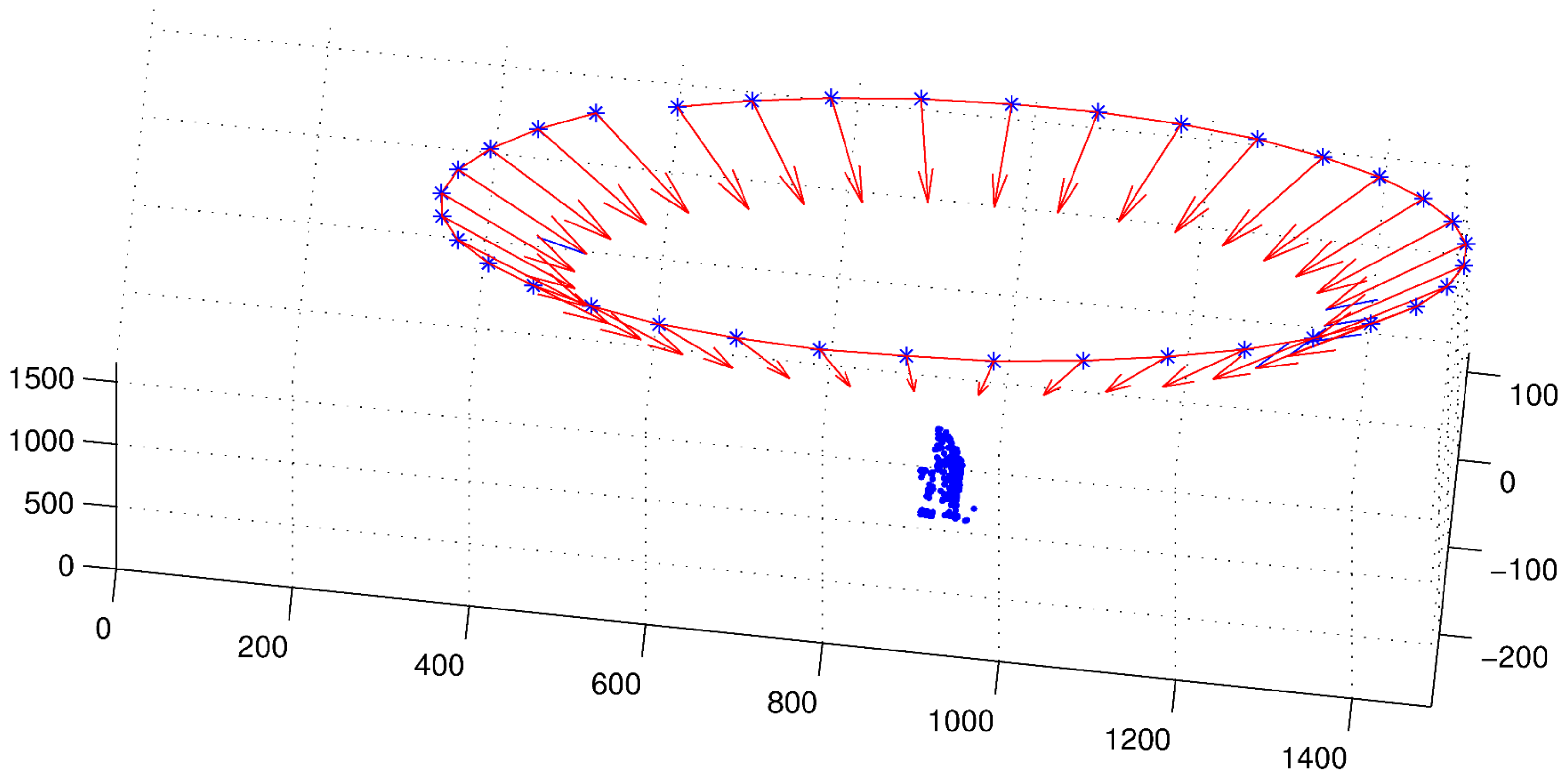


Example on a Real Sequence

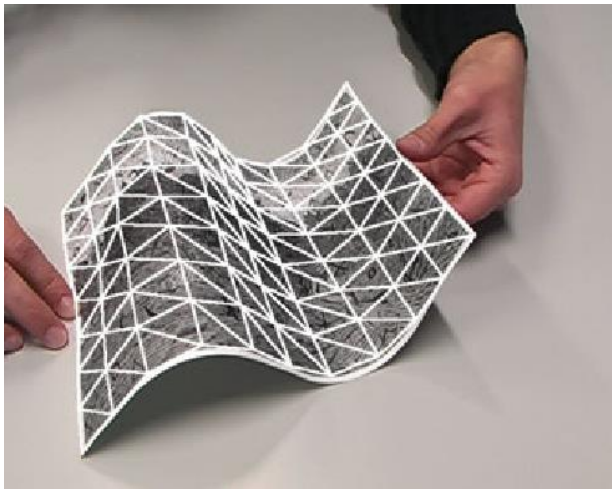
Algorithm (reference plane):

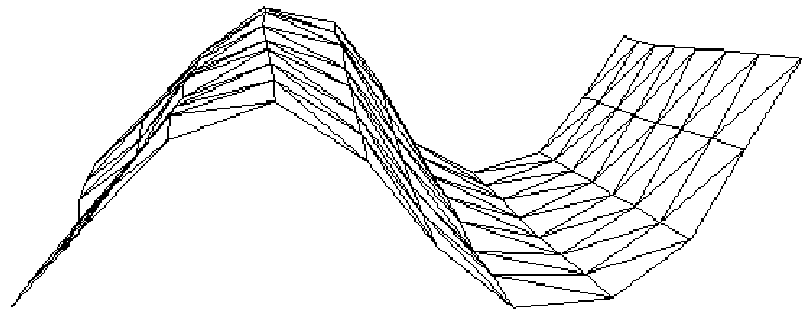
1. Track image points
2. Compute interimage homographies
3. Compute cameras and 3D points

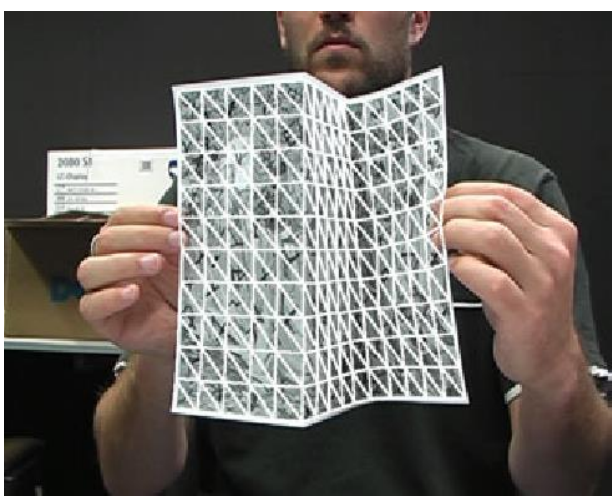
Dinosaur Reconstruction

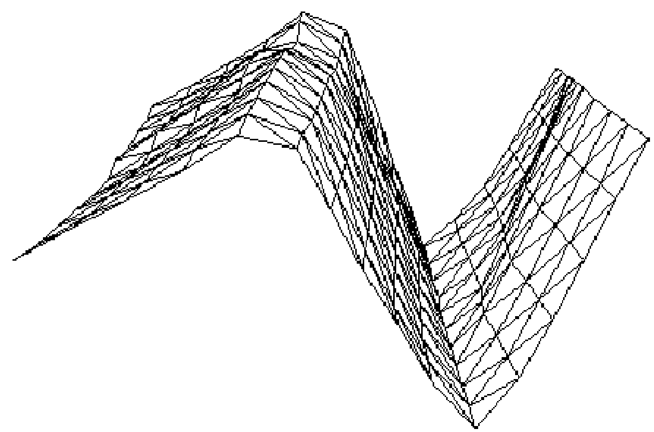


Flexible object tracking









M. Salzman, R. Hartley, P. Fua, ICCV 2007

Other Problems

1. Camera resection.
2. Projections from $\mathcal{P}^n \rightarrow \mathcal{P}^m$
3. Minimax vanishing point estimation.
4. Estimate with uncertainty (Sim-Hartley: CVPR 2006, Ke-Kanade: CVPR 2006).
5. Projective triangulation.
6. Projective SfM given plane correspondence. (Hartley-Kahl)

Problem: How to incorporate rotations into this methodology?

Implemention

- Convex optimization based on SeDuMi
- Convex feasibility problem:
  - 0.05s for 3-view triangulation
  - 1s for 2270 cone constraints with 36 views and 328 points
- Improved Bisection method
- Typically 5-10 iterations required to reach optimum within 10^{-5} pixels

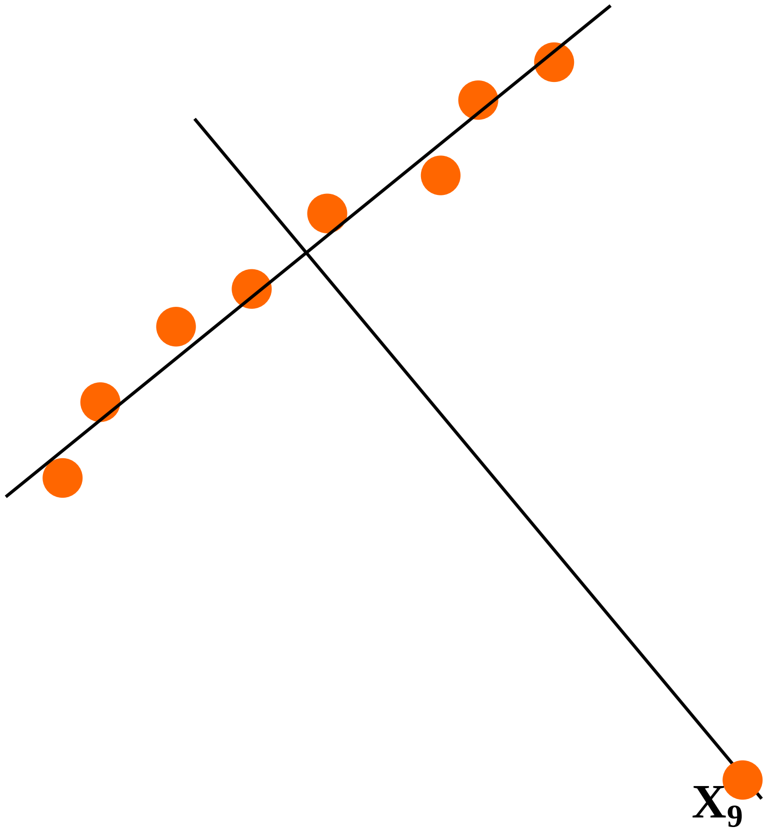
MATLAB Toolbox available on my homepage

See also: Pierre Moulon, PhD Thesis, 2013

Outliers

— detection/removal of outliers

Why are Outliers a Problem?



Problem: Find line of best fit

Measurements: $\mathrm{X}_i = (x_i, y_i)$
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

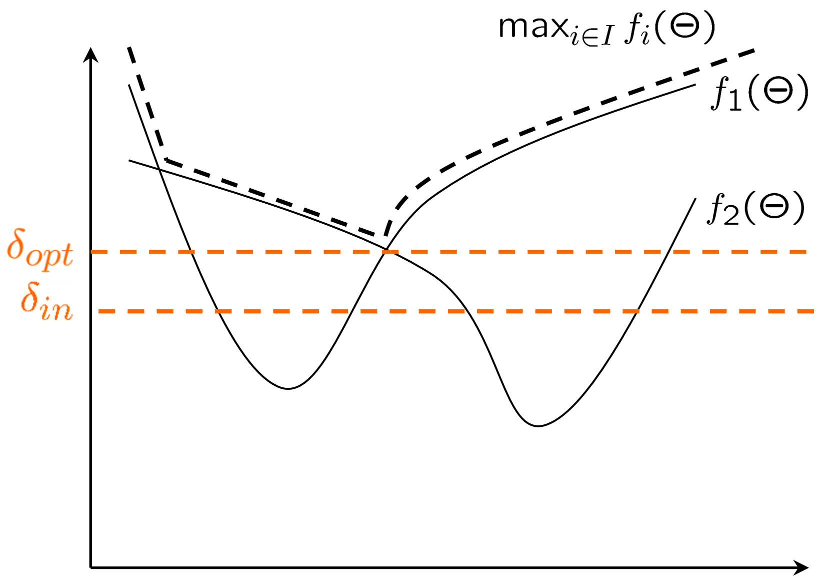
When the L_∞-idea was first introduced, it was considered a major drawback its sensitivity to outliers.

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}.

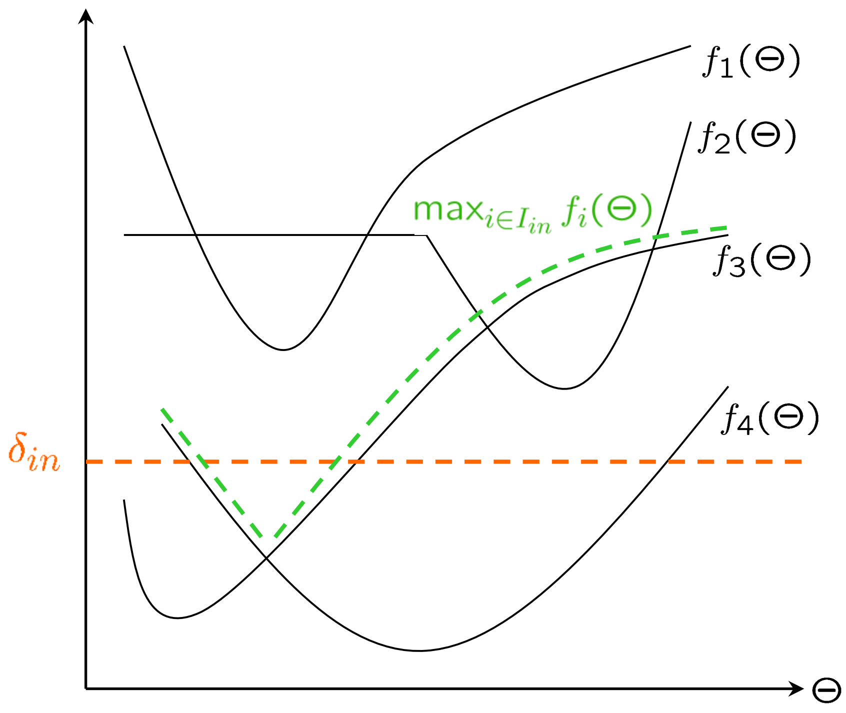
Then either $f_1(\Theta)$ or $f_2(\Theta)$ has to be removed such that

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

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).

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

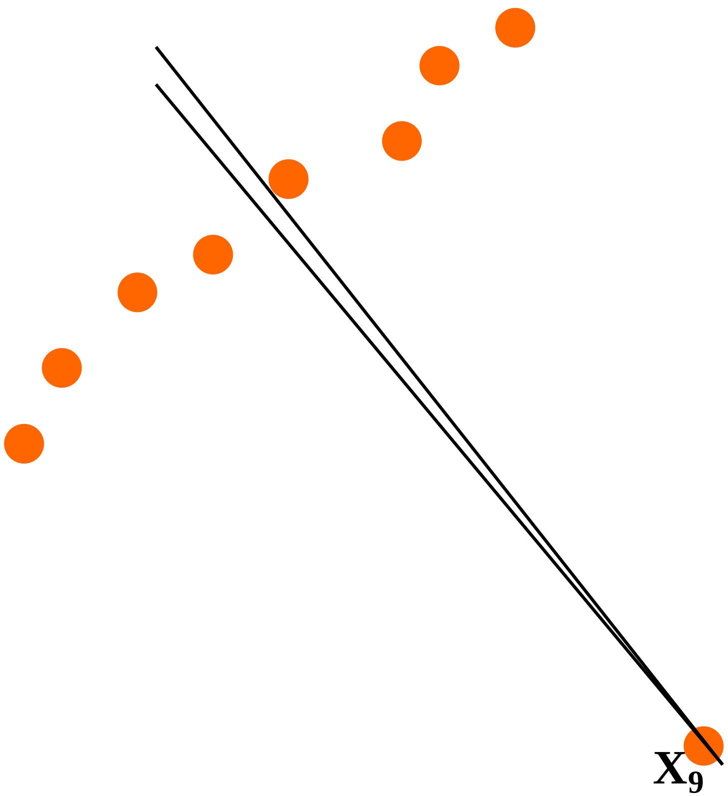
How Do We Remove Outliers?

- **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.

- **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

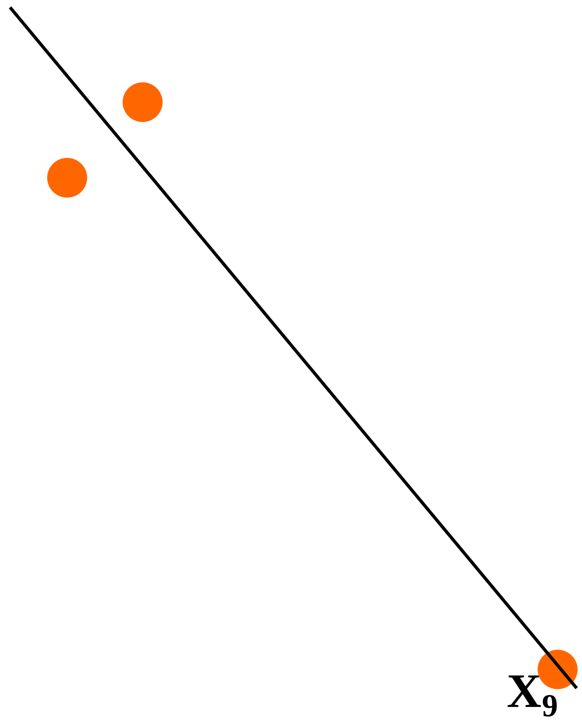


Outlier removal strategy:

– 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?

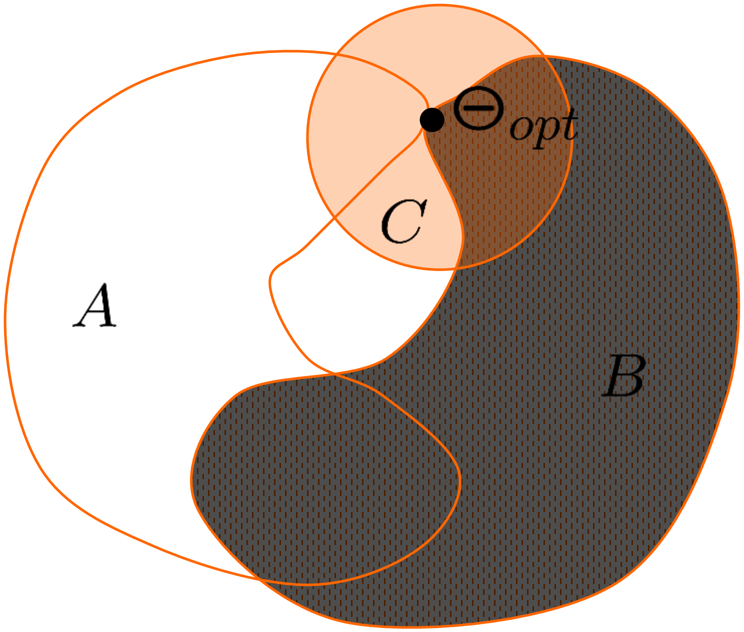
Theorem: **(Under certain conditions)**
Consider a minimax problem with solution $\min_\Theta \max_{i \in I} f_i(\Theta) = \delta_{opt}$.
Suppose there exists $I_{in} \subset I$ for which $\min_\Theta \max_{i \in I_{in}} f_i(\Theta) < \delta_{in} < \delta_{opt}$.
Then I_{supp} must contain at least one 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}.

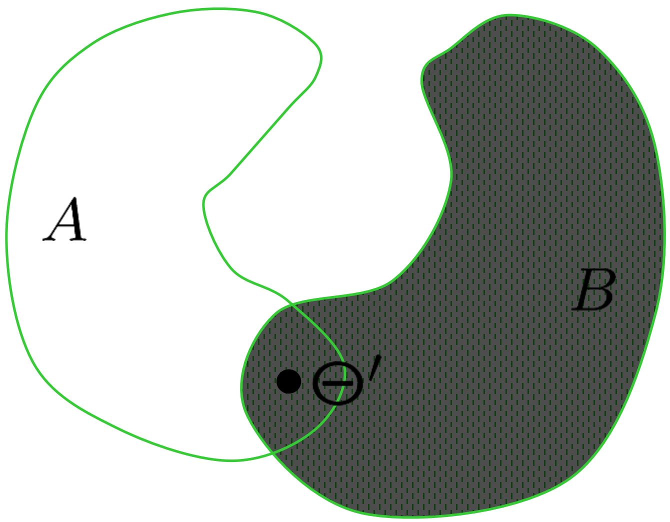
Quasiconvexity Is Needed For Condition A To Hold



- 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



We need convex sublevel sets.
Quasiconvexity is needed!

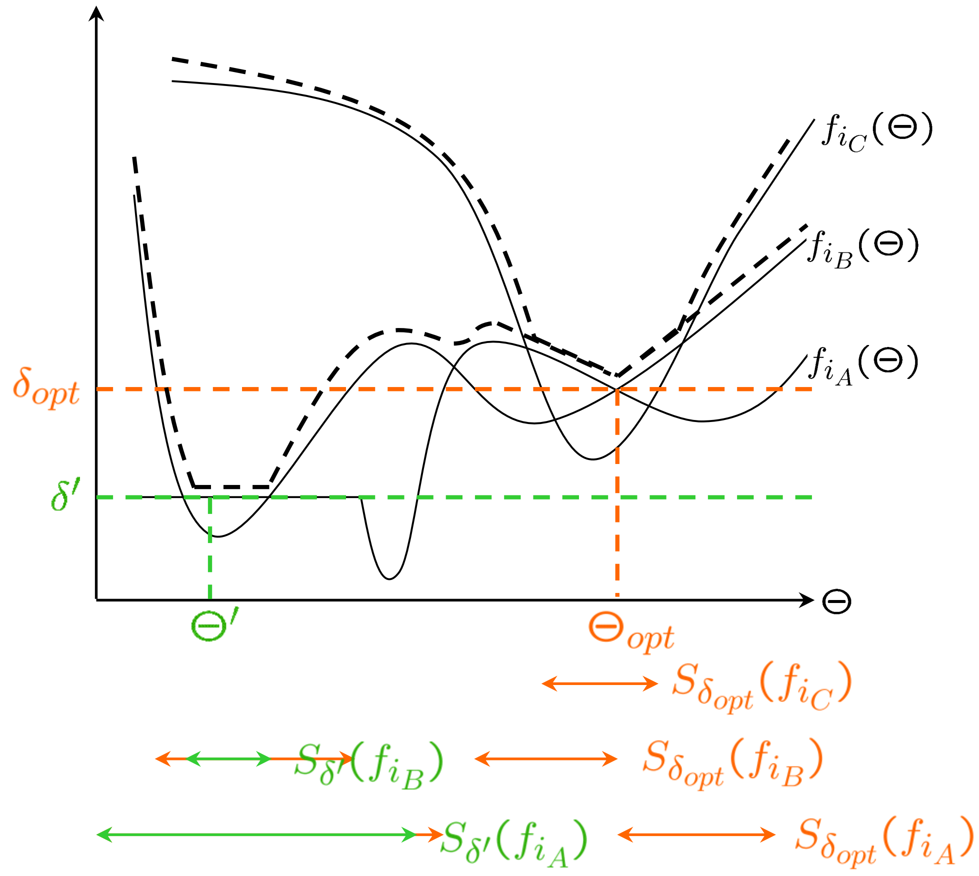
- A, B, C are the sublevel sets of 3 error functions $f_{i_A}, f_{i_B}, f_{i_C}$.
- f_{i_C} is QC $\Rightarrow$ C is a convex set
  f_{i_A}, f_{i_B} are not QC $\Rightarrow$ A, B are nonconvex sets
- $\Theta_{opt} = A \cap B \cap C$

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

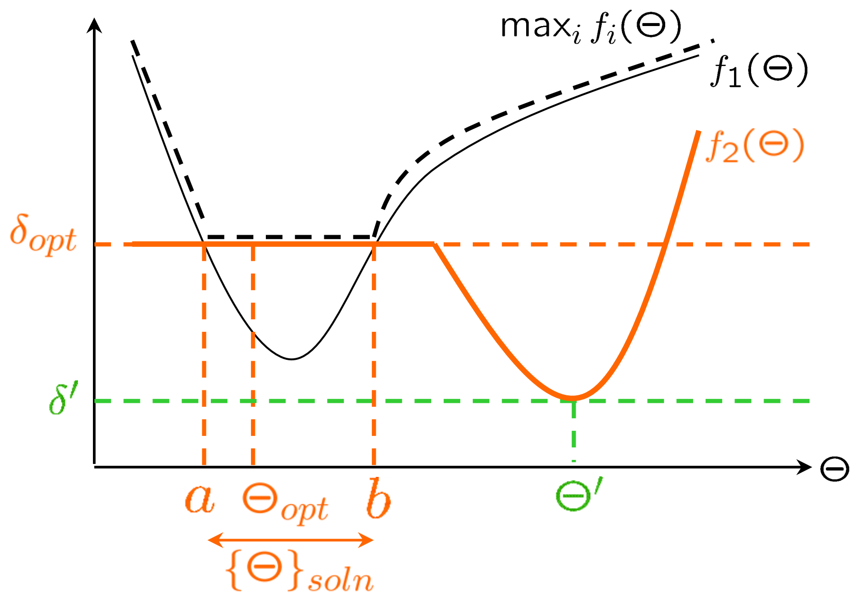
Quasiconvexity Is Needed For Condition A To Hold

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.

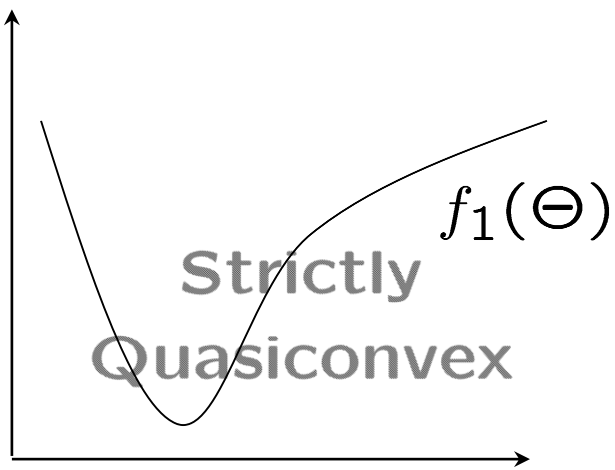
- Bisection algorithm will find a new solution Θ' with a lower L_∞ error δ'.
  $\Rightarrow$ Quasiconvexity is insufficient

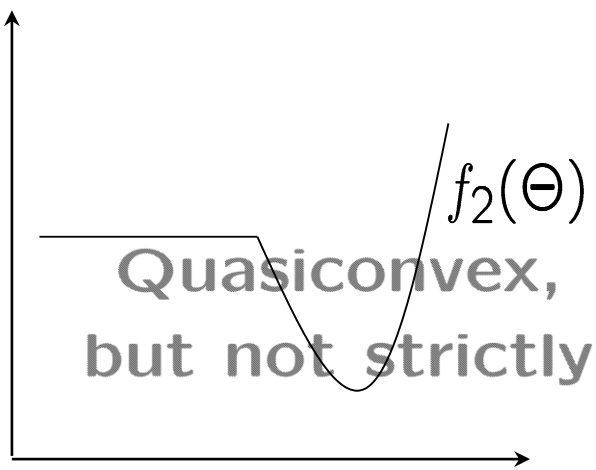
Need smoothness condition on sublevel sets.
Strict Quasiconvexity is needed!

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 $\cup_{\mu<\delta}\ S_\mu(f) = \text{Int}\ S_\delta(f)\ \ \forall\ \delta$





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.

For a detailed proof, see:

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

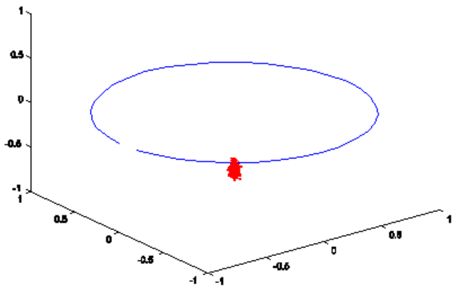
What Does This All Mean?

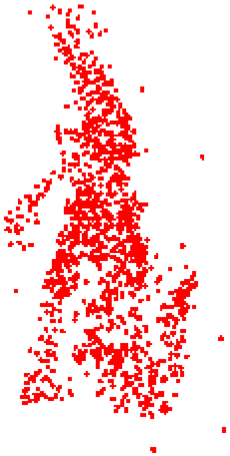
If we can write a geometric vision problem as an L_∞ optimization problem where the error functions $f_i(\Theta)$ are strictly quasiconvex then I_{supp} must contain at least one outlier.

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