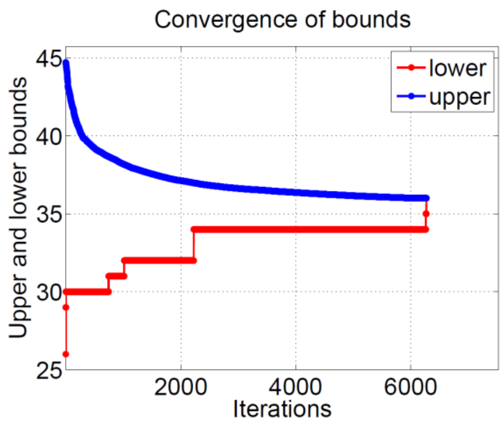
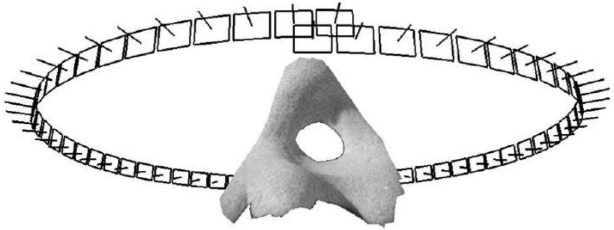
# Global Optimization





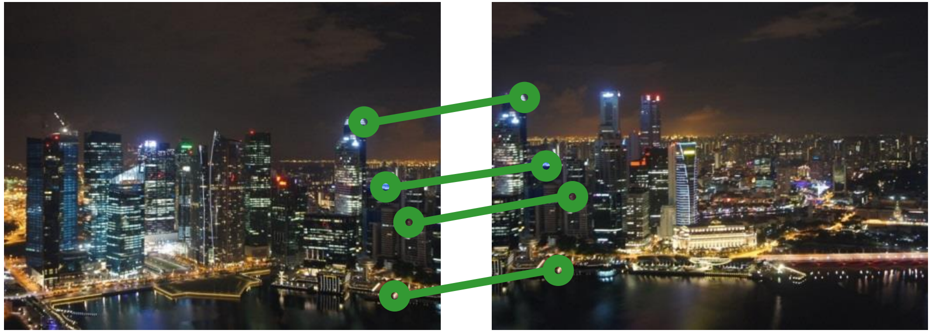
**Dr. Jean-Charles Bazin**

Disney Research Zürich

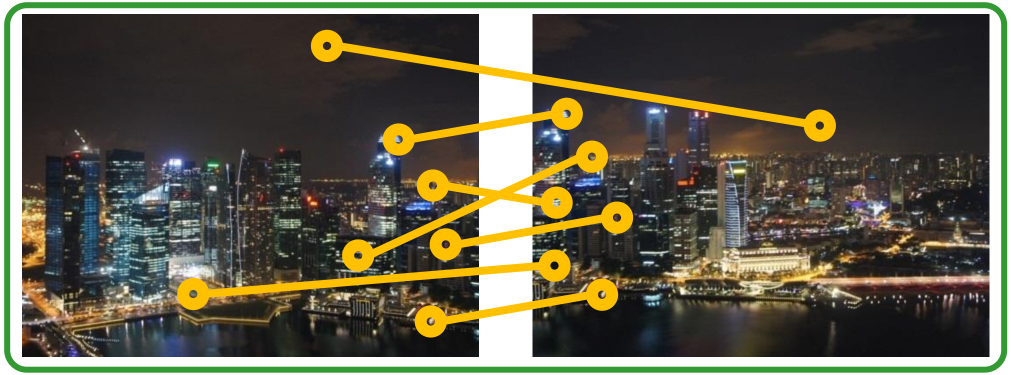

ETH zürich

# Summary of the previous lecture and additional content

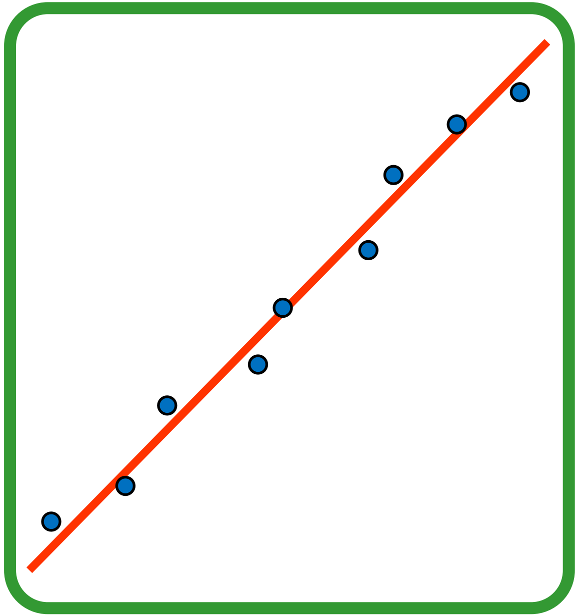
# Outlier - Motivation



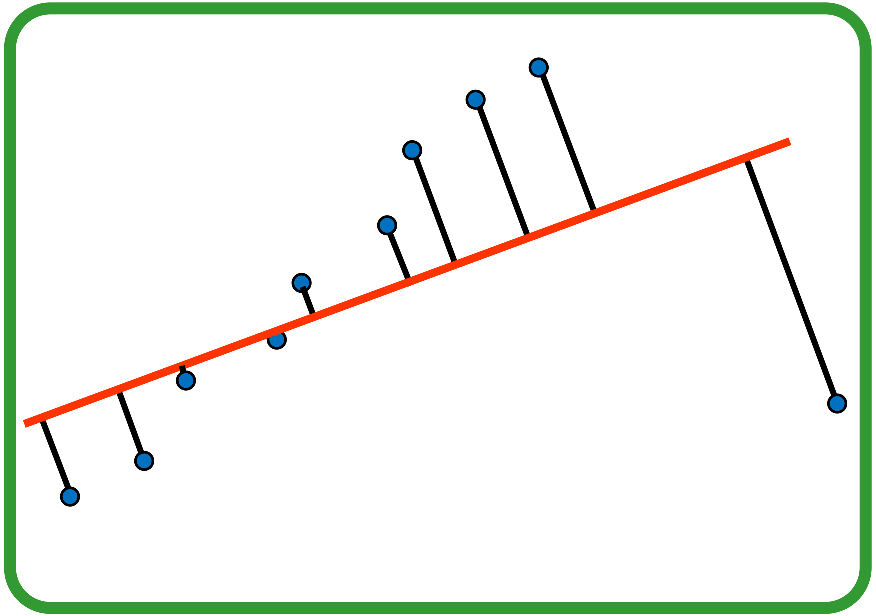
# Outlier - Motivation



# Outlier - a motivation example



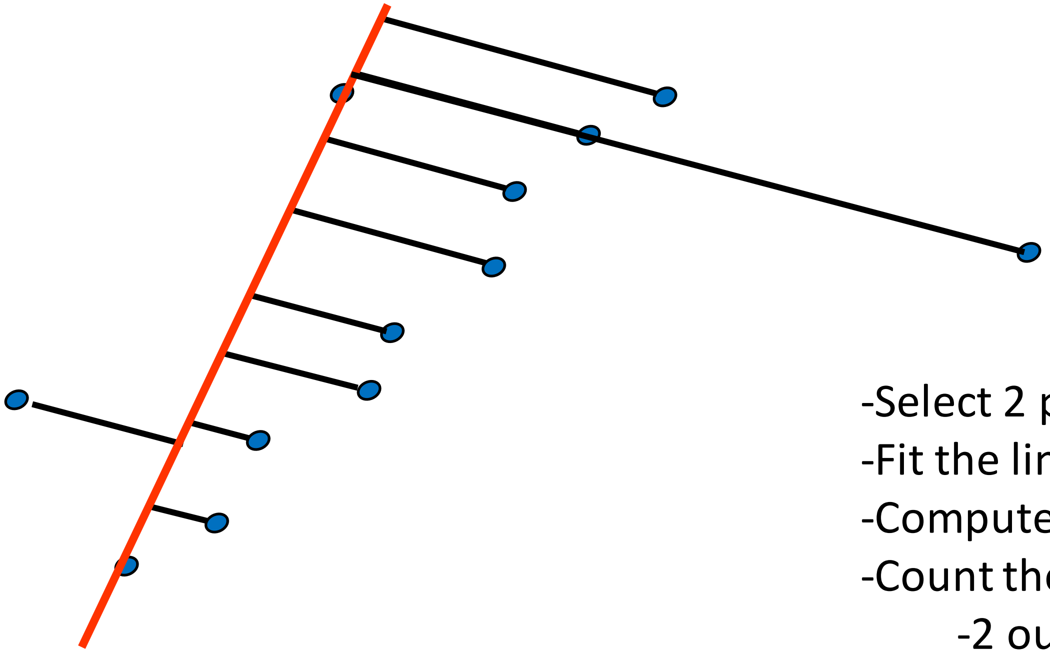
Only **inliers**



Effects due to a single **outlier**

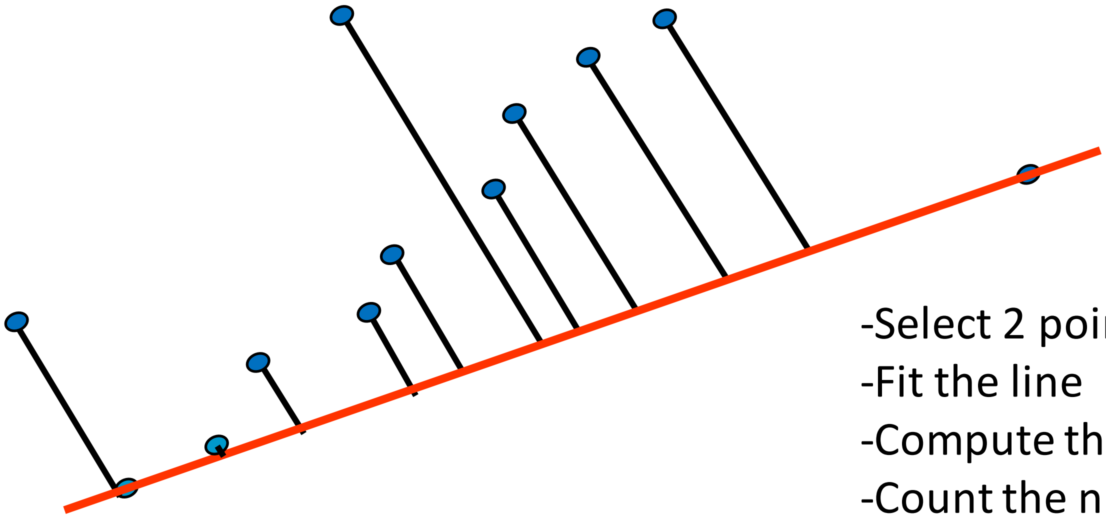
Detecting outliers is very important!

# RANSAC



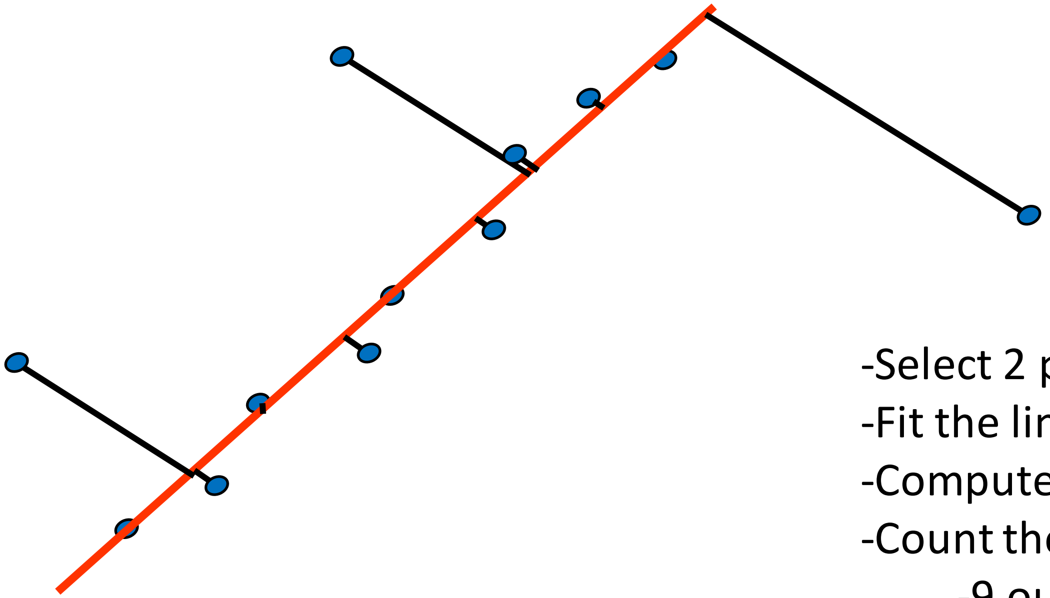
-Select 2 points
-Fit the line
-Compute the distances
-Count the nb of inliers
    -2 out of 13

# RANSAC



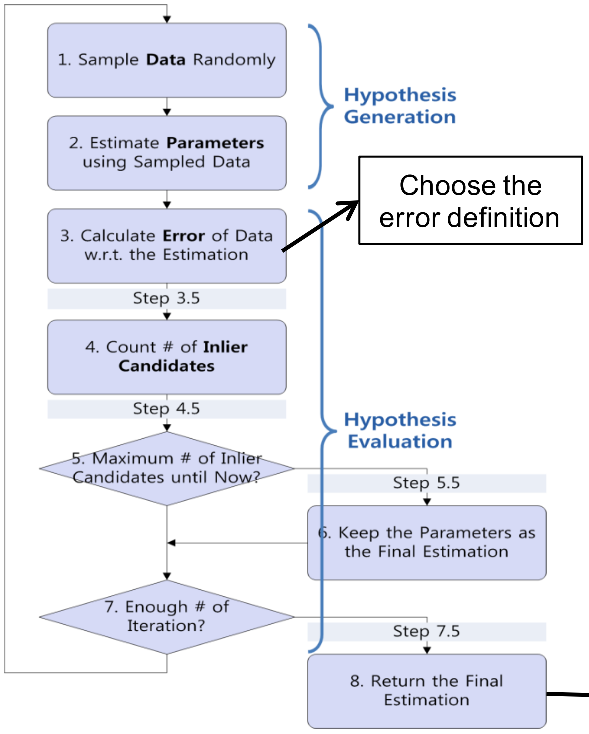
- -Select 2 points
- -Fit the line
- -Compute the distances
- -Count the nb of inliers
  - -3 out of 13

# RANSAC

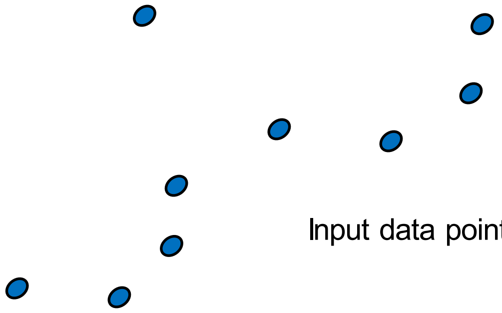


-Select 2 points
-Fit the line
-Compute the distances
-Count the nb of inliers
-9 out of 13

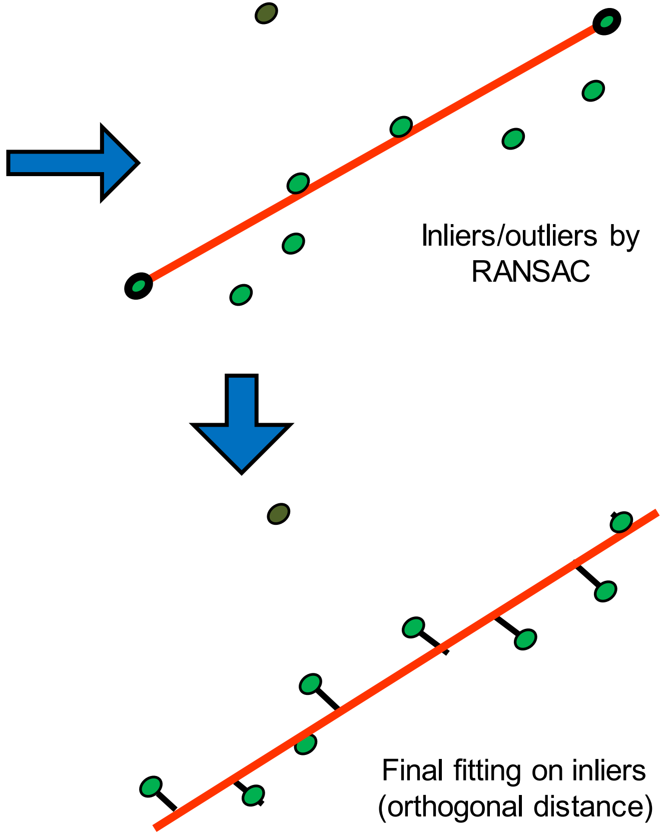
# RANSAC




Fitting **with the inliers** (L2, $L_\infty$, etc)



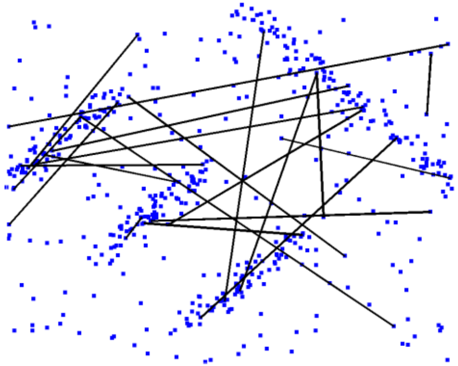
Input data points

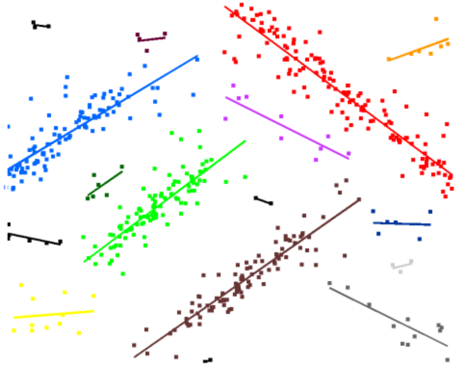


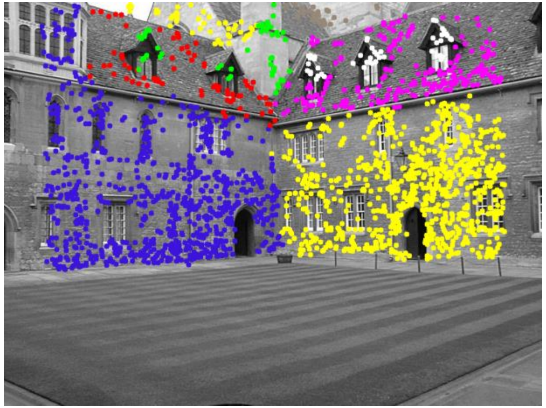

From “Performance Evaluation of RANSAC Family”, by Choi et al., BMVC, 2009

# RANSAC

- Multi-structure → Sequential RANSAC
- See also J-linkage, multi-label graph-cut



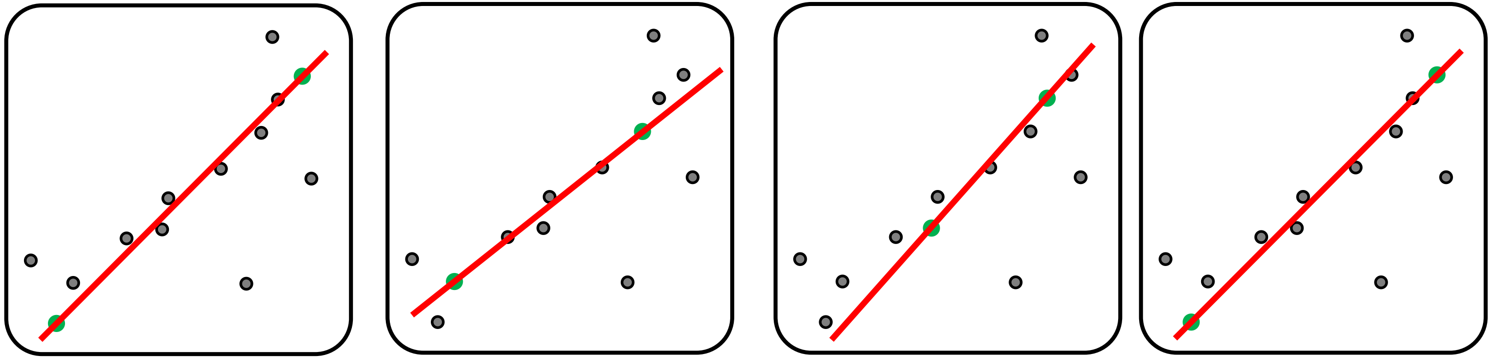




“Energy-based Geometric Multi-Model Fitting”, by Hossam Isack, Yuri Boykov, IJCV, 2012

# Exhaustive search

- On data points
  - e.g. every pair of points for line detection
  - Finite nb. of combinations (but maybe many!)



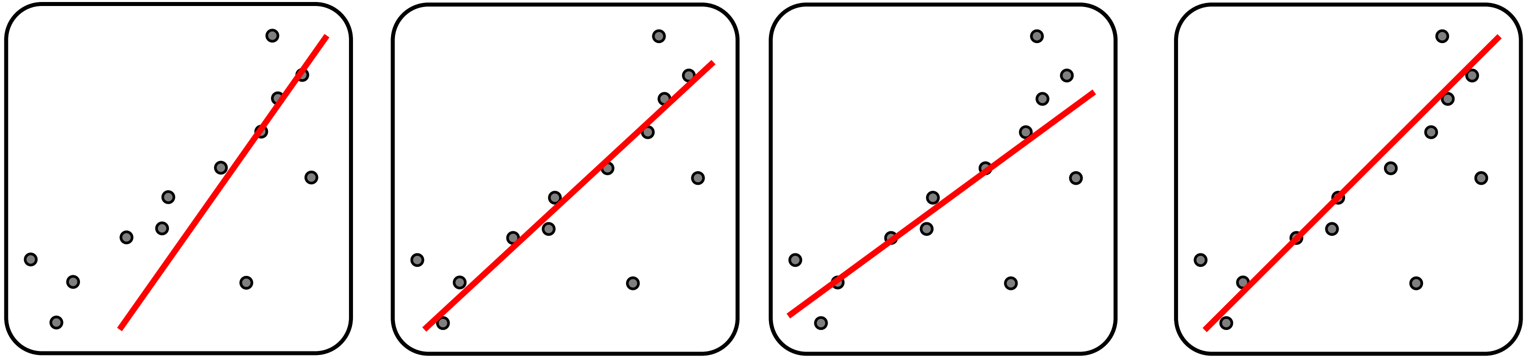
Etc etc…

How many combinations?

- $N*(N-1)$
- $(p_i,p_j)$ provides the same line as $(p_j,p_i)$ so $N*(N-1)/2$ $\binom{N}{2}$

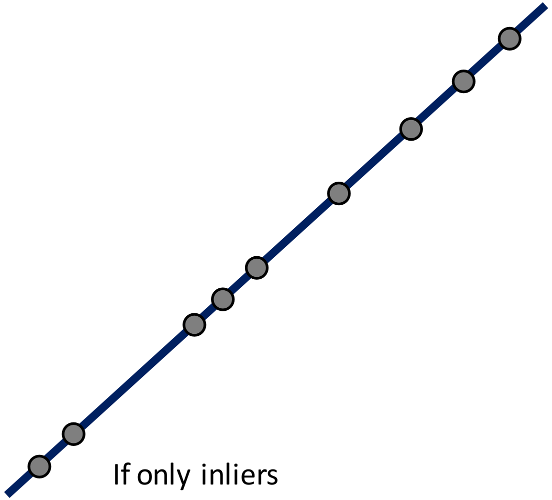
# Exhaustive search

- On model search space
  - E.g. every 2D line
  - Will give the optimal result but infinite nb. of models
    - Set of real numbers, in multi-dimensions
    - generally impossible in practice



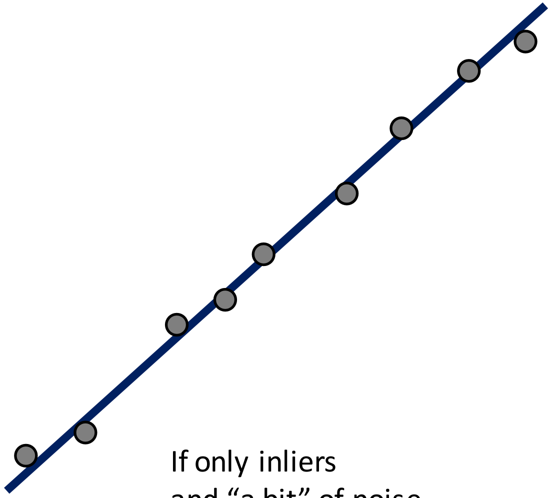
Etc etc...

# Additional comments



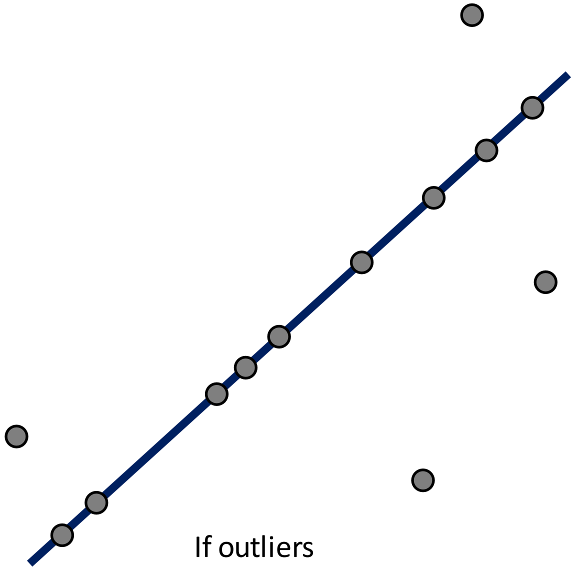
If only inliers
and no noise

Least squares fitting OK
Or even fitting by 2 points



If only inliers
and “a bit” of noise

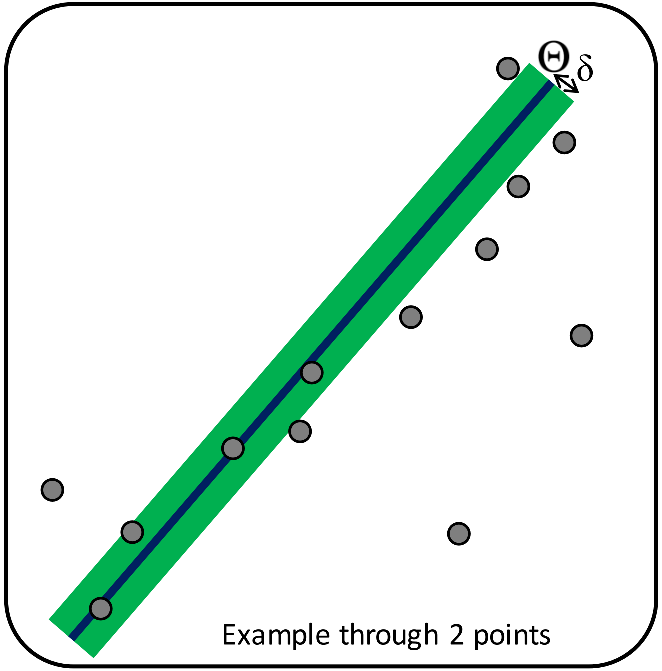
Least squares fitting OK



If outliers
and no noise

The optimal line model passes through 2 points.
- So exhaustive search on the data points will return the **optimal** line model
- RANSAC will **likely return** the optimal line model (but not guaranteed)

# Additional comments

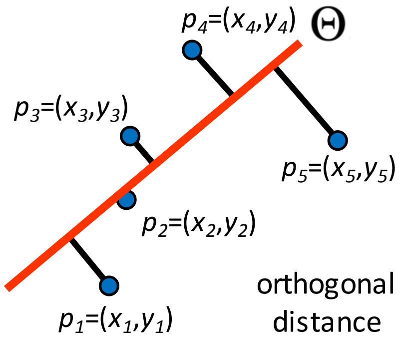



If outliers and noise

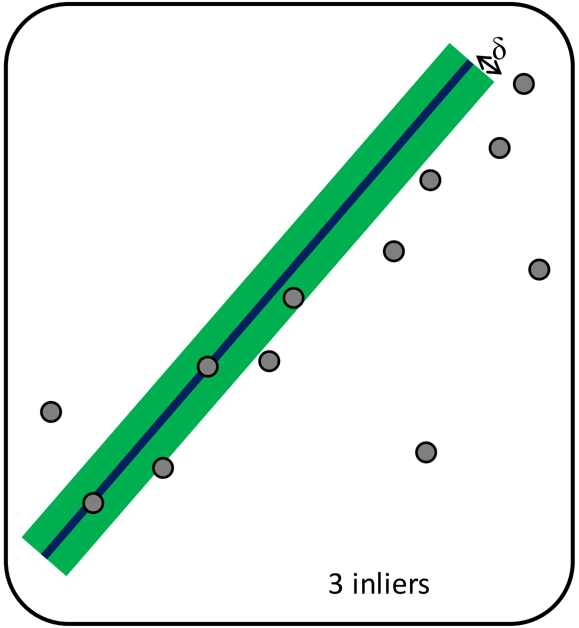
Inlier definition:

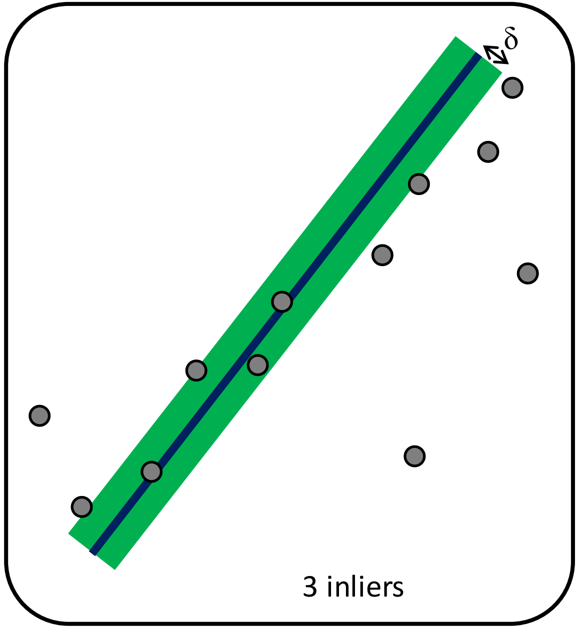
$$d(\Theta, p_i) \leq \delta$$

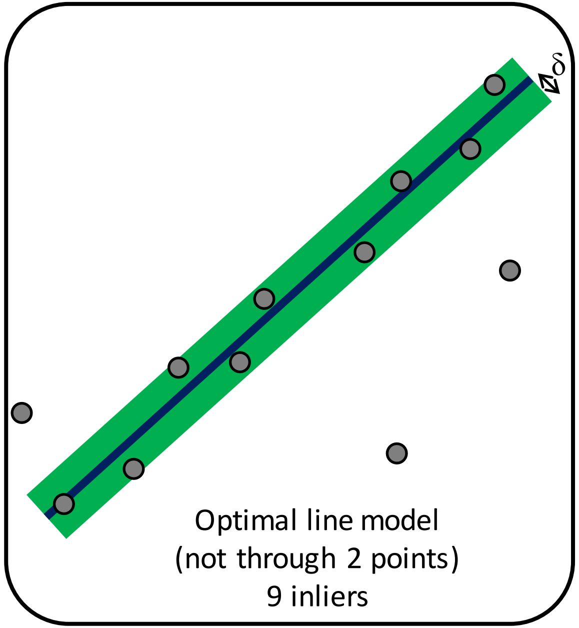
threshold



# Additional comments



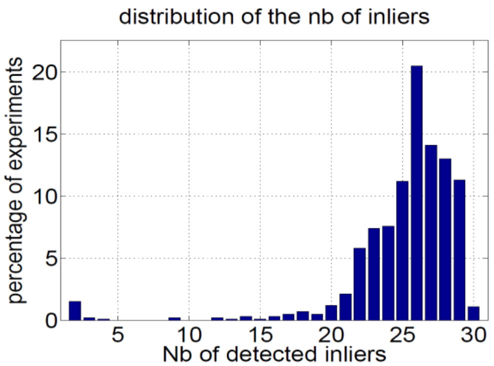

If outliers
and noise

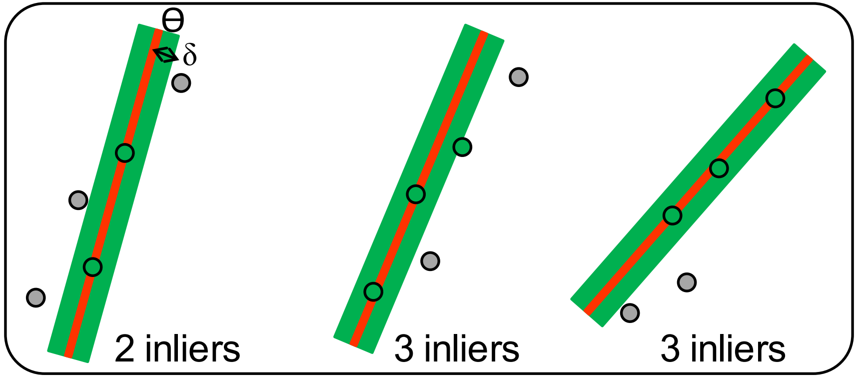
Exhaustive search on the data points and RANSAC will miss the optimal line model if it does not go exactly through two data points

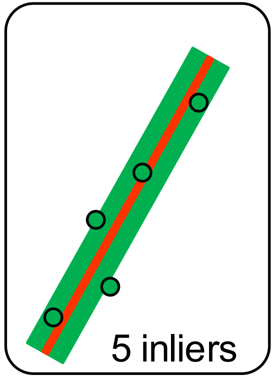
RANSAC stuck 在 noisy 的 data上,
miss optimal model

# RANSAC - limitations

- Works great but….
- Probabilistic behavior
  - Different runs might lead to different results
- How many iterations?
- Hypothesizes only models directly supported by the samples




Models hypothesized from samples "continuous" models

RANSAC 非要从 sample 来

→ No guarantee to obtain the **optimal** solution (i.e. maximize the nb of inliers)

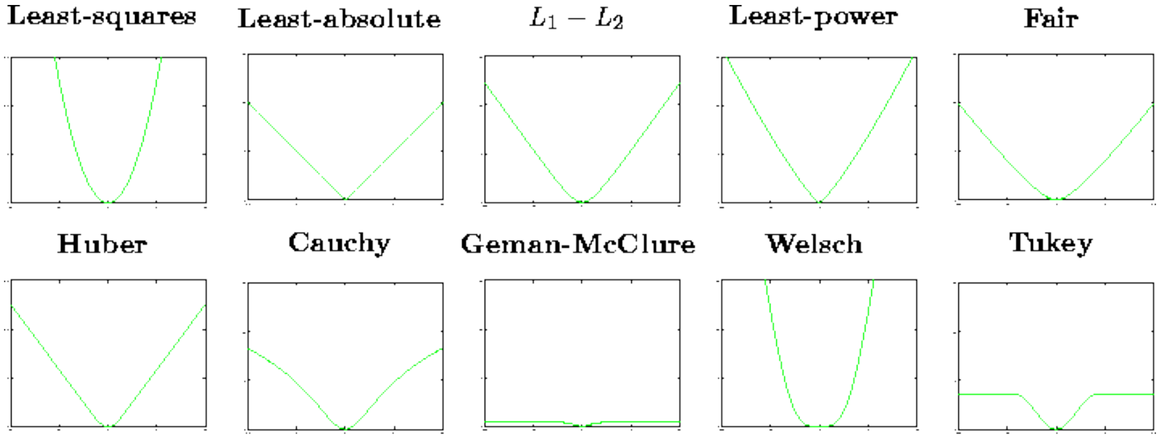
# Norms and Loss functions

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p} \qquad \begin{aligned} \|\mathbf{x}\|_\infty &= \max_i |x_i| = \max\left(|x_1|, \ldots, |x_n|\right) \\ \|\mathbf{x}\|_0 &= \operatorname{card}\left(i | x_i \neq 0\right) \end{aligned}$$

$$\arg\min_{\mathbf{x}} \|A\mathbf{x} - b\|_2 = \arg\min_{\mathbf{x}} \|A\mathbf{x} - b\|_2^2 = \arg\min_{\mathbf{x}} \sum_{i=1}^{m} (A_i\mathbf{x} - b_i)^2 = \arg\min_{\mathbf{x}} \sum_{i=1}^{m} e_i^2$$

$$\arg\min_{\mathbf{x}} \|A\mathbf{x} - b\|_1 = \arg\min_{\mathbf{x}} \sum_{i=1}^{m} |A_i\mathbf{x} - b_i| = \arg\min_{x} \sum_{i=1}^{m} |e_i|$$

$$\arg\min_{\mathbf{x}} \|A\mathbf{x} - b\|_\infty = \arg\min_{\mathbf{x}} \max_{i=1,\ldots,m} |A_i\mathbf{x} - b_i| = \arg\min_{\mathbf{x}} \max_{i=1,\ldots,m} |e_i|$$



# Inlier-outlier example

- Motion model (1D translation)

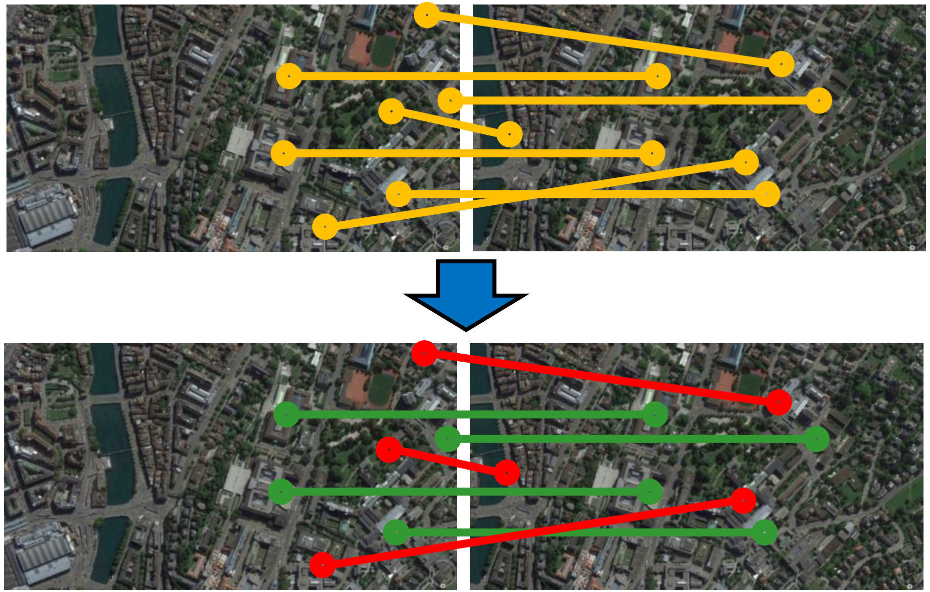
$$x_i' = x_i + T$$

- Inlier definition

$$d(x_i, x_i') \leq \varepsilon$$

$$|x_i' - (x_i + T)| \leq \varepsilon$$

- How to find inliers?
  - RANSAC
  - Exhaustive search on the data points
  - Exhaustive search on the model space (generally impossible)
  - Other approaches: e.g. BnB, etc



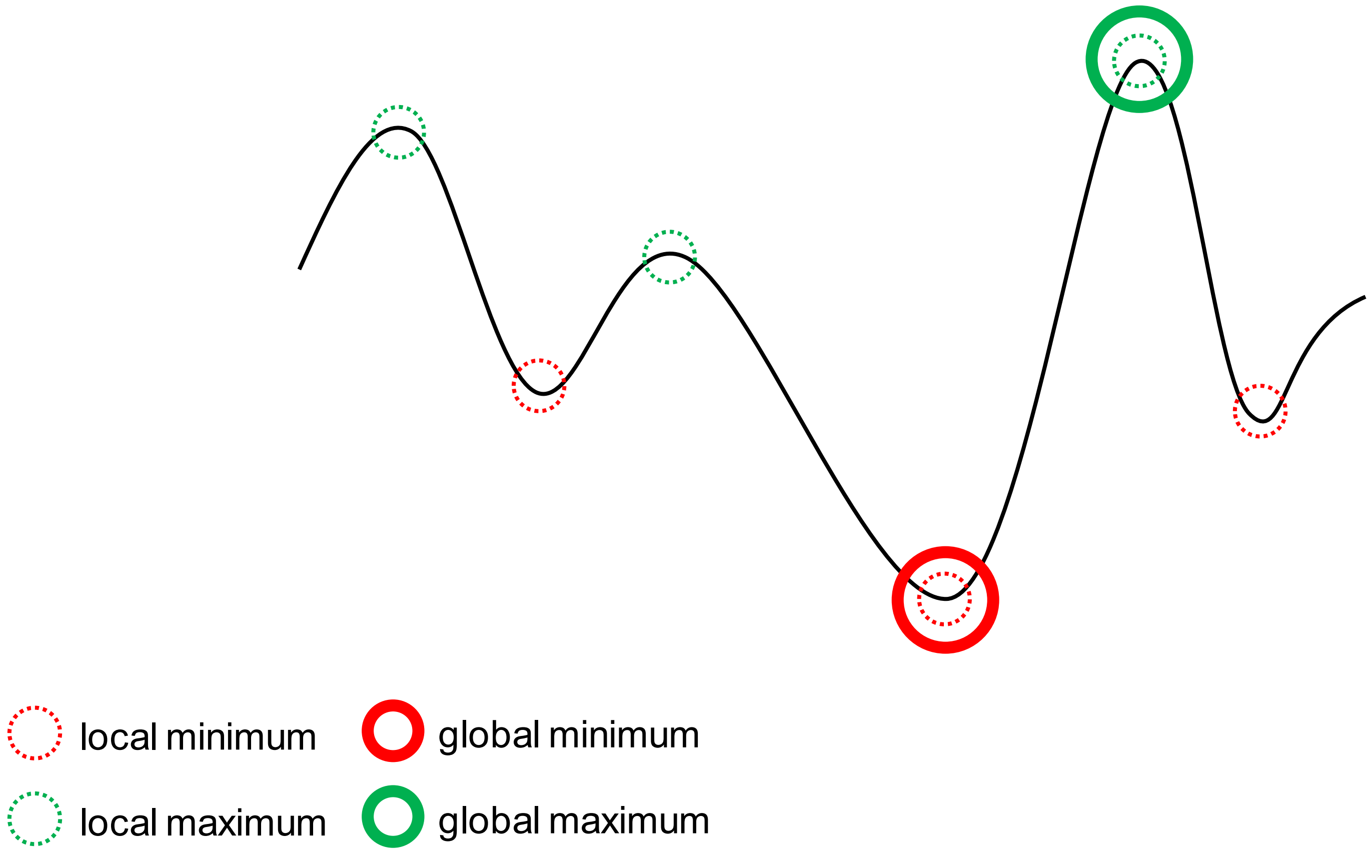
From Google Earth views

# Contents

- Part I (easy)
  - Convex
  - Linear programming
  - “Degenerate” linear system
  - (univariate) polynomial

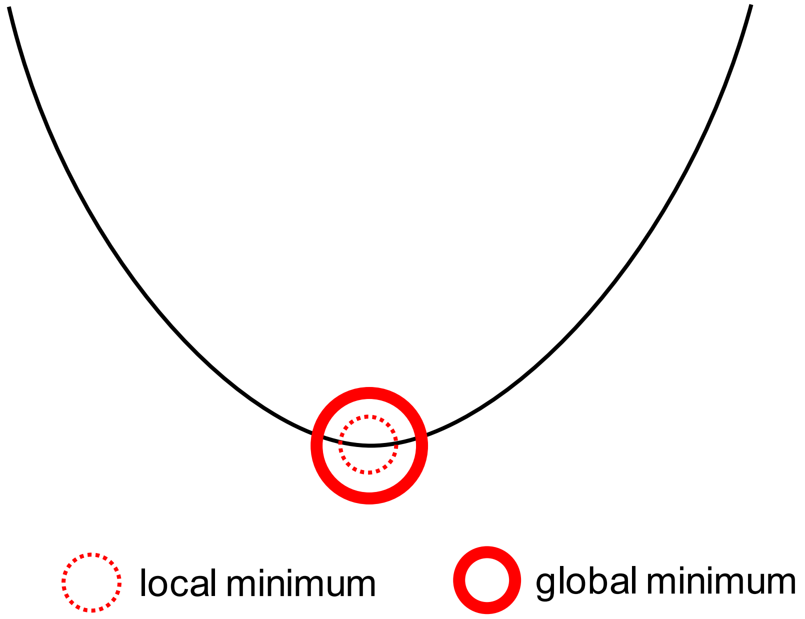
- Part II (harder)
  - System of polynomials
  - Branch and bound

# Local vs global optimization



# A simple case

- Convex system

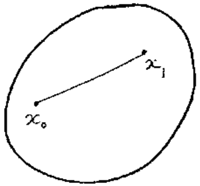


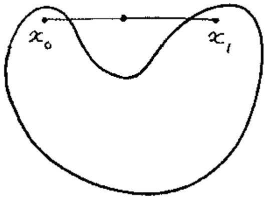
# Convex

- Convex domain:

A set $D \subset \mathbf{R}$ is convex if the line joining points $x_0$ and $x_1$ lies inside $D$.

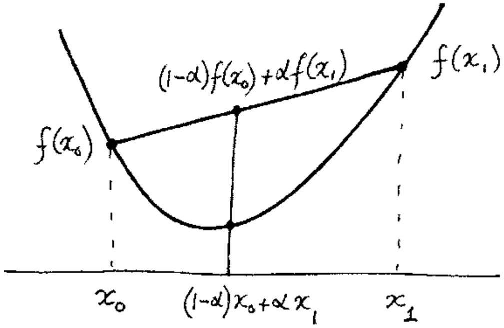
$$(1-t)x_0 + tx_1 \in D, \forall x_0, x_1 \in D \text{ and } \forall t \in [0,1]$$




- Convex function:

$$f((1-\alpha)x_0 + \alpha x_1) \leq (1-\alpha)f(x_0) + \alpha f(x_1)$$




- Solving:
  - e.g. gradient descent from anywhere
  - If high dimension: see “rank minimization” lecture

ECCV Course on “Continuous Optimization Methods in Computer Vision”, F. Kahl and R. Hartley, 2006

# Linear Programming

- Canonical form:

$$\max_x c^T x$$
$$\text{s.t. } Ax \leq b$$
$$\text{and } x \geq 0$$

- Linear objective function
- Linear equality/inequalities
- Convex set

- Easy to convert: min/max, ≤, ≥, >, <, =
- Solving algorithms:
  - Simplex, interior point

- Matlab's linprog

$$\min_x c^T x \text{ such that } \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ l_b \leq x \leq u_b \end{cases}$$

# Linear system

$$\boxed{Ax = b}$$

*mxn* *nx1* *mx1*

$$7x + 5y + 3z = 9$$
$$2x + 6y + 8z = 12$$
$$4x + y + 5z = 7$$

- *m*>*n*: over-determined → least-square $\min_x \|Ax - b\|$
- *m*<*n*: under-determined
  - Infinite number of solutions
  - "family" of solutions

$$Ax = b$$
$$\Rightarrow A^T A x = A^T b$$
$$\Rightarrow x = (A^T A)^{-1} A^T b$$

- Algorithms:
  - Inverse (square *A*,*m*=*n*), pseudo-inverse (*m*≥*n*)
  - Gaussian elimination, Cramer's rule (small matrix)
  - Iterative:
    - Jacobi, Gauss-Seidel, etc…
    - Conjugate gradient, etc..

$$x_i = \det(\widetilde{A}_i)/\det(A)$$

$i^{th}$ column of *A* replaced by *b*

# Linear system

- Example of Gaussian elimination

$$
\begin{aligned}
4x+2y-2z&=16\\
-6x-2y+4z&=-22\\
-4x+2y+4z&=-6
\end{aligned}
\qquad
\left[\begin{array}{ccc|c}
4 & 2 & -2 & 16\\
-6 & -2 & 4 & -22\\
-4 & 2 & 4 & -6
\end{array}\right]
$$

Eliminate *x*

$$
\begin{aligned}
L_2 &= L_2 + 6/4L_1\\
L_3 &= L_3 + L_1
\end{aligned}
$$

$$
\begin{aligned}
4x+2y-2z&=16\\
y+z&=2\\
4y+2z&=10
\end{aligned}
\qquad
\left[\begin{array}{ccc|c}
4 & 2 & -2 & 16\\
0 & 1 & 1 & 2\\
0 & 4 & 2 & 10
\end{array}\right]
$$

Eliminate *y*

$$L_3 = L_3 - 4L_2$$

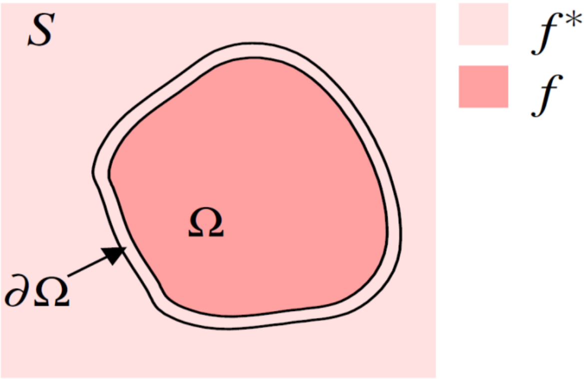
$$
\begin{aligned}
4x+2y-2z&=16\\
y+z&=2\\
-2z&=2
\end{aligned}
\qquad
\left[\begin{array}{ccc|c}
4 & 2 & -2 & 16\\
0 & 1 & 1 & 2\\
0 & 0 & -2 & 2
\end{array}\right]
$$

$$
\begin{aligned}
&z=-1\\
&y=2-z=3\\
&x=(16-2y+2z)/4=(16-2*3+2*(-1))/4=2
\end{aligned}
$$

# Linear System $\quad Ax = b$

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

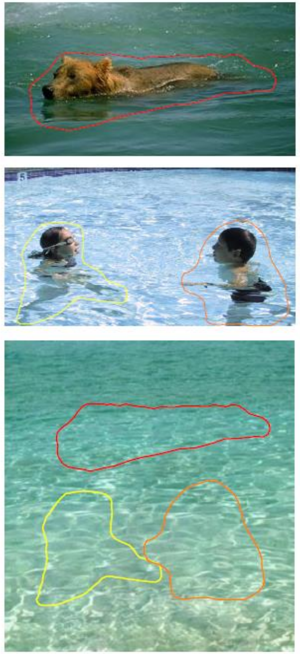
$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$




source/destination | cloning | seamless cloning

Perez et al., “Poisson image editing”, SIGGRAPH, 2003

# Linear System $Ax = b$



sources/destinations

cloning

seamless cloning

Perez et al., “Poisson image editing”, SIGGRAPH, 2003

# "Linear" system $Ax = 0$

- Not by pseudo-inverse:

$$\begin{aligned} x &= (A^T A)^{-1} A^T b \\ &= (A^T A)^{-1} A^T \mathbf{0} \\ &= \mathbf{0} \end{aligned}$$

- Degenerate solution is 0 (and up-to-scale)
- Solve by **SVD** (Singular Value Decomposition)
  - eigenvector associated with the smallest eigenvalue

$$[U, S, V] = svd(M)$$

*m*x*n* input matrix

*m*x*m*. Contains the eigenvectors of $MM^T$

*n*x*n*. Contains the eigenvectors of $M^TM$

$$M = USV^T$$

*m*x*n* diagonal matrix.
Contains the singular values of *M*, i.e. square roots of the eigenvalues of $MM^T$ and $M^TM$

$$\min_x ||Ax||$$

Since Ax=0

$$\text{s.t. } ||x|| = 1$$

***Solution:*** *column of V associated to the smallest eigenvalue!*

# Proof #1/2

- Eigenvalue/vector of *M*: $M\mathbf{e}_i = \lambda_i \mathbf{e}_i$ for $i = 1, 2, 3$
- The eigenvectors span the space so any ***v*** can be written as:

$$\mathbf{v} = \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \alpha_3 \mathbf{e}_3$$

- Since the eigenvectors are orthonormal:

$$\mathbf{v} \cdot \mathbf{v} = \alpha_1^2 + \alpha_2^2 + \alpha_3^2$$

- Derivations:

$$\begin{aligned} M\mathbf{v} &= M\left(\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \alpha_3 \mathbf{e}_3\right) \\ &= \alpha_1 M\mathbf{e}_1 + \alpha_2 M\mathbf{e}_2 + \alpha_3 M\mathbf{e}_3 \\ &= \alpha_1 \lambda_1 \mathbf{e}_1 + \alpha_2 \lambda_2 \mathbf{e}_2 + \alpha_3 \lambda_3 \mathbf{e}_3 \end{aligned}$$

$$\mathbf{v}M\mathbf{v} = \mathbf{v} \cdot (M\mathbf{v}) = \alpha_1^2 \lambda_1 + \alpha_2^2 \lambda_2 + \alpha_3^2 \lambda_3$$

Proof from A3 of "Closed-form solution of absolute orientation using unit quaternions", B. Horn, Journal of the Optical Society of America A, 1987

# Proof #2/2

- Eigenvalues ordering: $\lambda_3 \leq \lambda_2 \leq \lambda_1$
- Thus

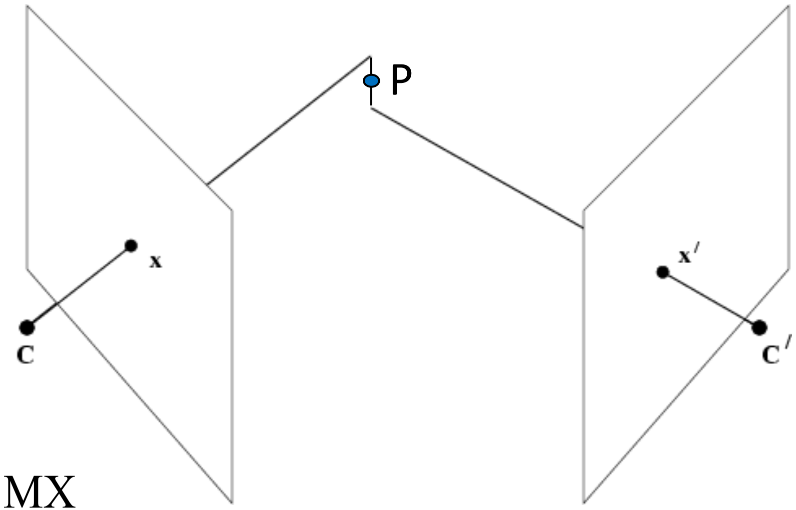
$$
\begin{aligned}
\mathbf{v}^T M \mathbf{v} &= \alpha_1^2 \lambda_1 + \alpha_2^2 \lambda_2 + \alpha_3^2 \lambda_3 \\
&\leq \alpha_1^2 \lambda_1 + \alpha_2^2 \lambda_1 + \alpha_3^2 \lambda_1 \\
&= \lambda_1 (\alpha_1^2 + \alpha_2^2 + \alpha_3^2) \\
&= \lambda_1
\end{aligned}
$$

- The max value is $\lambda_1$. How to get it? $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$
- Finally: $\mathbf{v} = \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \alpha_3 \mathbf{e}_3 = \mathbf{e}_1$

$$\max_{\mathbf{v}, \|\mathbf{v}\|=1} \mathbf{v}^T M \mathbf{v} \Rightarrow$$ Eigenvector associated to the largest eigenvalue!

$$\min_{\mathbf{v}, \|\mathbf{v}\|=1} \mathbf{v}^T M \mathbf{v} \Rightarrow$$ Eigenvector associated to the smallest eigenvalue!

# Application - triangulation




$\mathbf{x} = \mathbf{M}\mathbf{X}$

$\mathbf{x}' = \mathbf{M}'\mathbf{X}$

Given:
- intrinsically and extrinsically calibrated cameras
- point correspondences

Find: *P* as the midpoint

"Multiple View Geometry", R. Hartley and A. Zisserman, 2004

# Application - triangulation

$$\mathrm{x} = \mathrm{MX} \text{ and } \mathrm{x}' = \mathrm{M}'\mathrm{X}$$

$$\mathrm{x} \times \mathrm{MX} = 0$$

$$\mathrm{x}' \times \mathrm{M}'\mathrm{X} = 0$$

Linear combination of 2 other equations

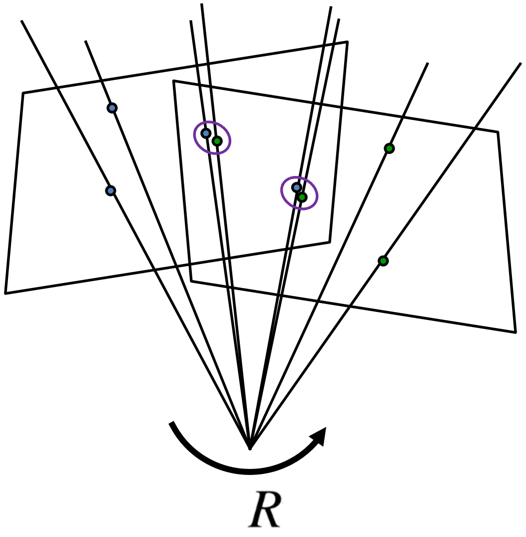
$$x\left(\mathrm{m}_3{}^{\mathrm{T}}\mathrm{X}\right) - \left(\mathrm{m}_1{}^{\mathrm{T}}\mathrm{X}\right) = 0$$
$$y\left(\mathrm{m}_3{}^{\mathrm{T}}\mathrm{X}\right) - \left(\mathrm{m}_2{}^{\mathrm{T}}\mathrm{X}\right) = 0$$
$$x\left(\mathrm{m}_2{}^{\mathrm{T}}\mathrm{X}\right) - y\left(\mathrm{m}_1{}^{\mathrm{T}}\mathrm{X}\right) = 0$$

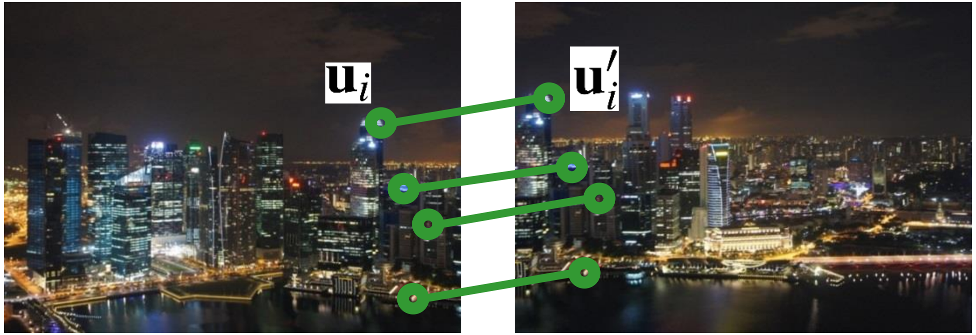
$$\Rightarrow \quad \mathrm{AX} = 0 \qquad \mathrm{A} = \begin{bmatrix} x\mathrm{m}_3{}^{\mathrm{T}} - \mathrm{m}_1{}^{\mathrm{T}} \\ y\mathrm{m}_3{}^{\mathrm{T}} - \mathrm{m}_2{}^{\mathrm{T}} \\ x'\mathrm{m}_3{}'^{\mathrm{T}} - \mathrm{m}_1{}'^{\mathrm{T}} \\ y'\mathrm{m}'_3{}^{\mathrm{T}} - \mathrm{m}'_2{}^{\mathrm{T}} \end{bmatrix}$$

⇒ *X* is last column of *V* in the SVD of *A*= $USV^T$

"Multiple View Geometry", R. Hartley and A. Zisserman, 2004

# Application – rotation estimation




$$R\mathbf{u}_1 = \mathbf{u}'_1$$
$$R\mathbf{u}_i = \mathbf{u}'_i$$
$$R\mathbf{u}_N = \mathbf{u}'_N$$

$$\max_{R \in SO(3)} \sum_{i=1}^{n} R\mathbf{u}_i \cdot \mathbf{u}'_i$$

$$\Downarrow$$

$$\max_{\mathbf{q}, ||\mathbf{q}||=1} \mathbf{q}^T M \mathbf{q}$$

"Closed-form solution of absolute orientation using unit quaternions", B. Horn, Journal of the Optical Society of America A, 1987

# Introduction – other problems

$$\min_{x} 5x+7$$
$$w.r.t \quad 2x+5 \geq 0$$
$$and \quad 3x-4 \geq 0$$

$$\min_{x,y} 8x+3y+7$$
$$w.r.t \quad 3x+5y \geq 0$$
$$and \quad x+3y-2 \geq 0$$

Linear Programming (easy), cf text book

$$\min_{x,y} 8x+3y+2xy+7$$
$$w.r.t \quad 3x+5y \geq 0$$
$$and \quad x+3y-2 \geq 0$$

$$\min_{x,y} 8x^2+3y^3+2xy+7$$
$$w.r.t \quad 4x^2+3xy+5y \geq 0$$
$$and \quad x+3y^2-2 \geq 0$$

Not linear: bilinearities (and higher). Much harder

# Never forget the basics!

- Equation: $ax^2 + bx + c = 0$

- Solving:

$$x = \frac{-b+\sqrt{b^2-4ac}}{2a} \text{ and } x = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

# Polynomial – degree 3

- Equation: $ax^3 + bx^2 + cx + d = 0$

- Solving:

$$x_1 = -\frac{b}{3a} - \frac{1}{3a}\sqrt[3]{\frac{1}{2}\left[2b^3 - 9abc + 27a^2d + \sqrt{(2b^3 - 9abc + 27a^2d)^2 - 4\left(b^2 - 3ac\right)^3}\right]} - \frac{1}{3a}\sqrt[3]{\frac{1}{2}\left[2b^3 - 9abc + 27a^2d - \sqrt{(2b^3 - 9abc + 27a^2d)^2 - 4\left(b^2 - 3ac\right)^3}\right]}$$

$x_2$, $x_3$, …

# Polynomial – degree 4

- Equation: $ax^4 + bx^3 + cx^2 + dx + e = 0$
- Solving:

$$r_1 = \frac{-a}{4} - \frac{1}{2}\sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{2^{\frac{1}{3}}\left(b^2 - 3ac + 12d\right)}{3\left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}\right)^{\frac{1}{3}}} + \left(\frac{2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}}{54}\right)^{\frac{1}{3}}} -$$

$$- \frac{-a^3 + 4ab - 8c}{4\sqrt{\frac{a^2}{4} - \frac{2b}{3} + \frac{2^{\frac{1}{3}}\left(b^2 - 3ac + 12d\right)}{3\left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}\right)^{\frac{1}{3}}} + \left(\frac{2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}}{54}\right)^{\frac{1}{3}}}}$$

$$-\frac{1}{2}\sqrt{\frac{a^2}{2} - \frac{4b}{3} - \frac{2^{\frac{1}{3}}\left(b^2 - 3ac + 12d\right)}{3\left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}\right)^{\frac{1}{3}}} - \left(\frac{2b^3 - 9abc + 27c^2 + 27a^2d - 72bd + \sqrt{-4\left(b^2 - 3ac + 12d\right)^3 + \left(2b^3 - 9abc + 27c^2 + 27a^2d - 72bd\right)^2}}{54}\right)^{\frac{1}{3}}} -$$

- Depressed quartic, Ferrari's solution, etc..

# Polynomial – in practice

$$p(t) = t^n + c_{n-1}t^{n-1} + \ldots + c_1 t + c_0$$

$$p(t) = \det(tI - A)$$

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & -c_o \\ 1 & 0 & \ldots & 0 & -c_1 \\ 0 & 1 & \ldots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & 1 & -c_{n-1} \end{bmatrix}$$

**Companion matrix**

Remember eigenvalues/vectors?

$$Av - \lambda v = 0$$

$$(A - \lambda I)v = 0$$

v≠0 so (A- λI) is singular so det = 0

Eigenvalues λ of A are the solutions of: $\det(A - \lambda I) = 0$

i.e. the roots of *p* are the eigenvalues of the companion matrix *A*

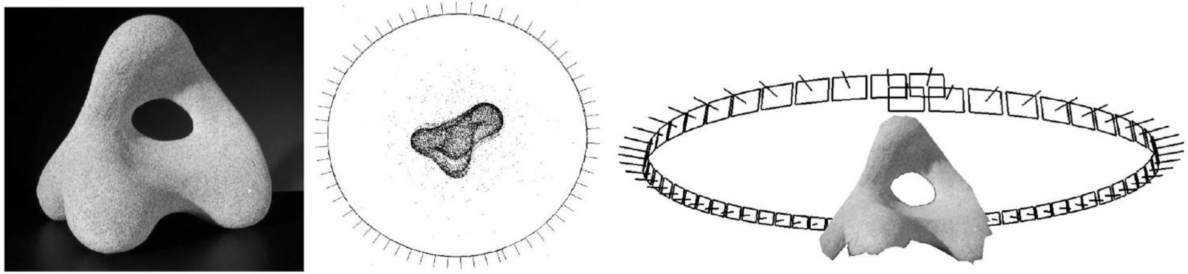


Fig. 16. Result from the turntable sequence "Stone." No prior knowledge about the motion or that it closes on itself was used in the estimation. The circular shape of the estimated trajectory is a verification of the correctness of the reconstruction that was made with low delay and bundle adjustment only for groups of three views.

$$q'^{\top} E q = 0$$

$$E E^{\top} E - \frac{1}{2} trace(E E^{\top}) E = 0$$

→ Polynomial of degree 10

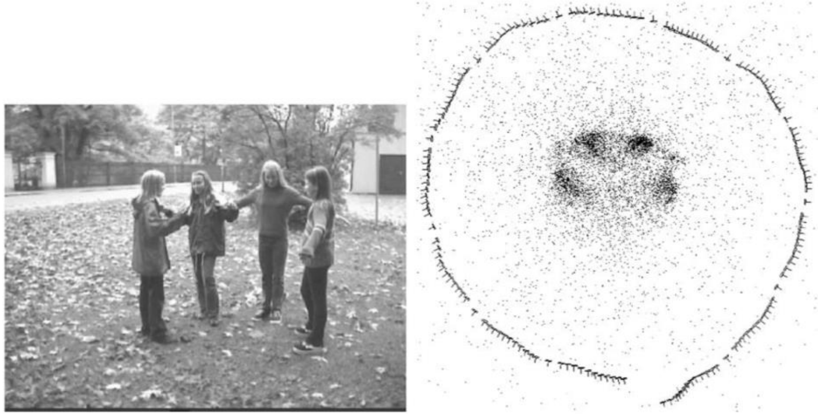


Fig. 19. Reconstruction from the sequence "Girlsstatue" that was acquired with a handheld camera. Only approximate intrinsic parameters were used and no global bundle adjustment was performed.

D. Nister, "An Efficient Solution to the Five-Point Relative Pose Problem", PAMI, 2004

# Gröbner basis

- Motivation example:

$$8x^2 + 3y^3 + 2xy + 7 = 0$$

$$4x^2 + 3xy + 5y = 0$$

- Gröbner basis:
  - Provides a list of polynomials that are easier to solve
  - At least one univariate polynomial
- Buchberger was the first to give an algorithm for computing Gröbner bases, 1965.
- Gröbner: name of his PhD advisor.
- A generalization of three techniques:
  - *Gaussian elimination* for solving linear systems of equations
  - the *Euclidean algorithm* for computing the greatest common divisor of two univariate polynomials
  - the *Simplex Algorithm* for linear programming

# Polynomial Division

- Remember (univariate) polynomial division?
  - repeatedly eliminating the term of highest degree
- Example: $f(x) = x^4 + x^3 + x^2 + x + 1$

  $g(x) = x^2 - x - 2$

$$
\begin{array}{r|l}
x^4 + x^3 + x^2 + x + 1 & x^2 - x - 2 \\ \hline
2x^3 + 3x^2 + x + 1 & x^2 \quad +2x \quad +5 \\
5x^2 + 5x + 1 & \\
10x + 11 &
\end{array}
$$

$$f(x) = (x^2 + 2x + 5)g(x) + 10x + 11$$

# Polynomial Division

- What about multivariate polynomials?
  - Univariate: eliminating the term of "highest degree"
  - What is the "term of highest degree"? $x^3y^2 + x^2y^3 - y^2$
  - → need to define an ordering $xy^2z, z^2, x^3, x^2z^2$
- Example of ordering:
  - Lex: Lexicographic order
    - "Power of *x* dominates"

$$x^3 > x^2z^2 > xy^2z > z^2$$

  - Grlex: Graded lexicographic order
    - "total degree dominates; then higher power of x"

$$x^2z^2 > xy^2z > x^3 > z^2$$

  - Grevlex: Graded reverse lexicographic order
    - "total degree dominates; then lower power of z" $xy^2z > x^2z^2 > x^3 > z^2$

# Polynomial Division

$$f(x) = x^3y^2 + x^2y^3 - y^2$$
$$g_1(x) = xy - 1$$
$$g_2(x) = y^2$$

- Dividing a given polynomial $f$ by a sequence of polynomials $g_1$, ..., $g_m$, given a term order:
- Find the leading term $t$ of $p$

  $$x^3 > x^2z^2 > xy^2z > z^2$$
- Eliminate it by
  - finding the smallest $i$ for which the leading term of $g_i$ divides $t$
  - and subtracting the appropriate multiple of $g_i$ from $f$.
- When no such $i$ exists, $t$ is eliminated by putting it in the remainder
- Stop when $f = 0$.

$$f(x,y) = \boxed{(x^2y + xy^2 + x + y)}\, g_1(x,y) \boxed{-1}\, g_2(x,y) + \boxed{x + y}$$

remainder

$$\boxed{x^3y^2} + x^2y^3 - y^2$$
$$-(x^3y^2 - x^2y)$$
$$\boxed{x^2y^3} - y^2 + x^2y$$
$$-(x^2y^3 - xy^2)$$
$$-y^2 + \boxed{x^2y} + xy^2$$
$$-(x^2y - x)$$
$$-y^2 + \boxed{xy^2} + x$$
$$-(xy^2 - y)$$
$$-y^2 + \boxed{x} + y$$
$$-\boxed{y^2} + y$$
$$-y^2$$
$$y$$
$$0$$

$xy - 1$ or $y^2$

$x^2y + xy^2 + x + y$

$-1$

remainder:

$$\boxed{x + y}$$

# S-polynomial

- S-polynomial: ("S" for subtractions)
  - eliminates the leading terms of multivariate polynomials (i.e. of *f* and *g*)

$$\mathbf{x}^{\gamma} = LCM(LM(f), LM(g))$$

Given $deg(f) = \alpha$ and $deg(g) = \beta$,
then $\gamma = (\gamma_1, \dots, \gamma_n)$ where $\gamma_i = \max(\alpha_i, \beta_i) \forall i$

$$S(f,g) = \frac{\mathbf{x}^{\gamma}}{LT(f)} \cdot f - \frac{\mathbf{x}^{\gamma}}{LT(g)} \cdot g$$

- Example
  - Input polynomials: $f(x,y) = x^3y^2 - x^2y^3 + x$ $\quad g(x,y) = 3x^4y + y^2$
  - Grlex: $x^2z^2 > xy^2z > x^3 > z^2$
  - LT and LCM

"total degree dominates; then higher power of x"

$LT(f) = x^3y^2$ $\quad deg(f) = (3,2)$ $\quad \gamma = (\max(3,4), \max(2,1)) = (4,2)$

$LT(g) = 3x^4y$ $\quad deg(g) = (4,1)$

# S-polynomial

$$f(x,y) = x^3y^2 - x^2y^3 + x \qquad g(x,y) = 3x^4y + y^2$$

$$LT(f) = 3x^2 \qquad LT(g) = 3x^4y \qquad \gamma = (\max(3,4), \max(2,1)) = (4,2)$$

$$\begin{aligned}
S(f,g) &= \frac{(x,y)^\gamma}{LT(f)}.f - \frac{(x,y)^\gamma}{LT(g)}.g \\
&= \frac{x^4y^2}{x^3y^2}\left(x^3y^2 - x^2y^3 + x\right) - \frac{x^4y^2}{3x^4y}\left(3x^4y + y^2\right) \\
&= x\left(x^3y^2 - x^2y^3 + x\right) - \frac{y}{3}\left(3x^4y + y^2\right) \\
&= x^4y^2 - x^3y^3 + x^2 - x^4y^2 - \frac{y^3}{3} \\
&= -x^3y^3 + x^2 - \frac{y^3}{3}
\end{aligned}$$

Ensures to have no variables in the denominator

Both LTs have been canceled!

# Buchberger's algorithm

**Theorem 2 (Buchberger's Algorithm)** *Let $I = \langle f_1, \ldots, f_s \rangle \neq \{0\}$ be a polynomial ideal. Then a Gröbner basis for $I$ can be constructed in a finite number of steps by the following algorithm:*

```
INPUT: F = (f_1, ..., f_s)
OUTPUT: Gröbner basis G = (g_1, ..., g_t) for I

LET G := F
REPEAT
    LET G' := G
    FOR each pair {p, q}, p ≠ q in G'
    DO
        LET S := \overline{S(p, q)}^{G'}
        IF S ≠ 0
            THEN G := G ∪ {S}
UNTIL G = G'
```

# An example

$$f_1(x,y) = x^2y - 2xy + x = 0$$

$$f_2(x,y) = 3x^2y - y^2 = 0$$

$$G = \{\underbrace{3x^4 - 6x^3 + x}_{\text{univariate}}, -18x^3 + 3x + y^2, xy - 3x^3\}$$

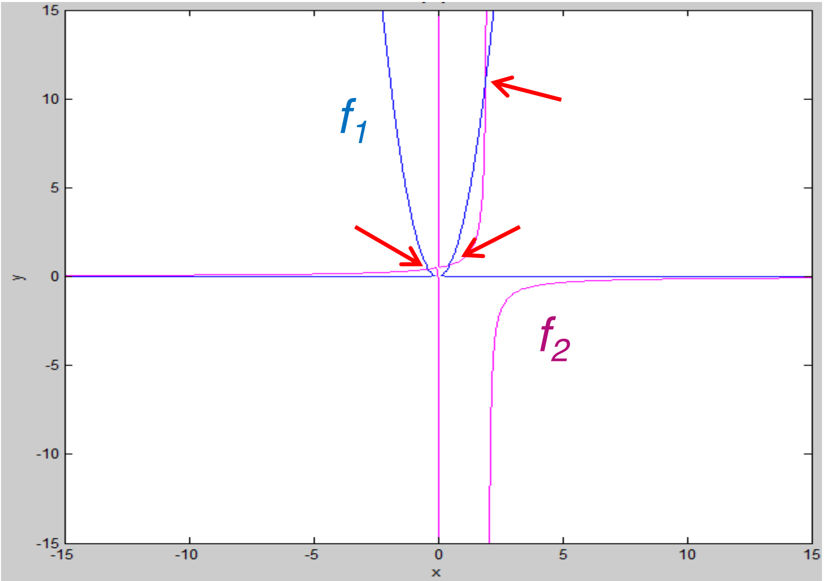
roots: (0,1.90,0.46,-0.37)

x=0 → y=0

x=1.90 → y=10.92

x=0.46 → y=0.65

x=-0.37 → y=0.42




f1=0,f2=0=>各grobner basis 都=0

# Applications

- **Minimal problems in computer vision**
  - Works of Thomas Pajdla, Zuzana Kukelova and Marting Bujnak
  - http://cmp.felk.cvut.cz/minimal/

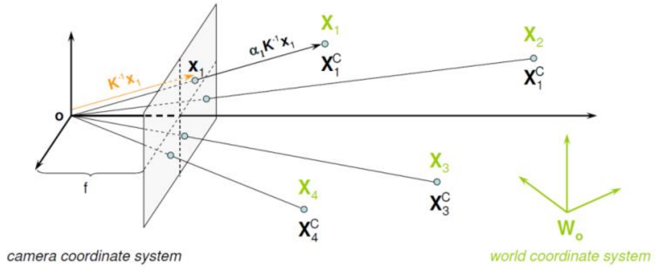



Figure 2. The camera and the world coordinate systems, image measurements ($\mathbf{x}$), ray direction vectors ($\mathtt{K}^{-1}\mathbf{x}$), and 3D points in the world coordinate system ($\mathbf{X}$) and in the camera coordinate system ($\alpha\,\mathtt{K}^{-1}\mathbf{x}_1$).

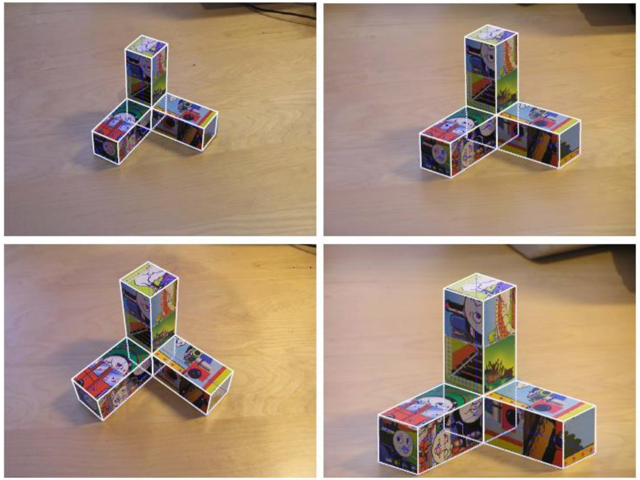


Figure 1. Camera pose and focal length are estimated automatically using four 2D-to-3D point correspondences. The accuracy of the estimate is demonstrated by predicting the projections of the edges of known 3D model of the structure.

“A general solution to the P4P problem for camera with unknown focal length”, M. Bujnak, Z. Kukelova, T. Pajdla, CVPR, 2008

# Application

- Given:
  - m=2 cameras, all calibrated except *f*
  - *n=6* corresponding points
- Compute the motion (*E* matrix) and *f*

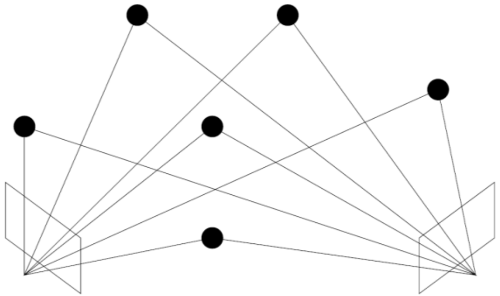


Fig. 1. The problem solved here: Relative orientation for two cameras, with a common but unknown focal length *f*, that see six unknown points.

$$2PFPPF^{\mathrm{T}}PPFP - \mathrm{tr}(PFPPF^{\mathrm{T}}P)PFP = 0.$$
$$\iff 2FP^2F^{\mathrm{T}}P^2F - \mathrm{tr}(FP^2F^{\mathrm{T}}P^2)F = 0.$$

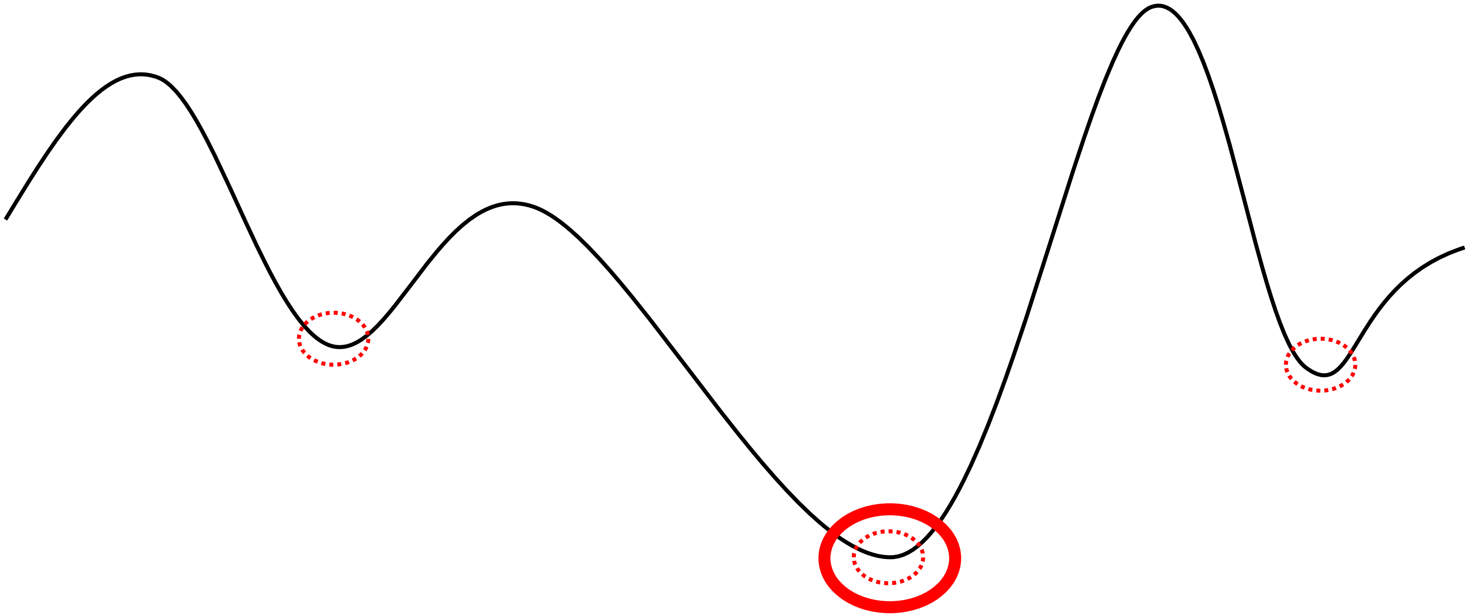
$$\det(F) = \det(F_0 + F_1 l_1 + F_2 l_2) = 0$$

10 equations of degree 5

$$\begin{aligned} X = [&l_1^3p^2, l_1^3p^1, l_1^3, l_1^2l_2^1p^2, l_1^1l_2^2p^2, l_2^3p^2, l_1^2l_2^1p^1, l_1^1l_2^2p^1, l_2^3p^1, \\ \times\ &l_1^2l_2^1, l_1^1l_2^1p^2, l_1^2p^2, l_2^2p^2, l_1^1p^3, l_1^1l_2^2, l_2^3, l_1^2p^1, l_2^1p^3, l_1^1l_2^1p^1, \\ \times\ &l_2^2p^1, l_1^1p^2, l_2^1p^2, p^3, l_1^2, l_1^1l_2^1, l_2^2, l_1^1p^1, l_2^1p^1, p^2, l_1^1, l_2^1, p^1, 1]^{\mathrm{T}}. \end{aligned}$$

“A minimal solution for relative pose with unknown focal length”, H. Stewenius, D. Nister, F. Kahl, F. Schaffalitzky, IVC, 2007

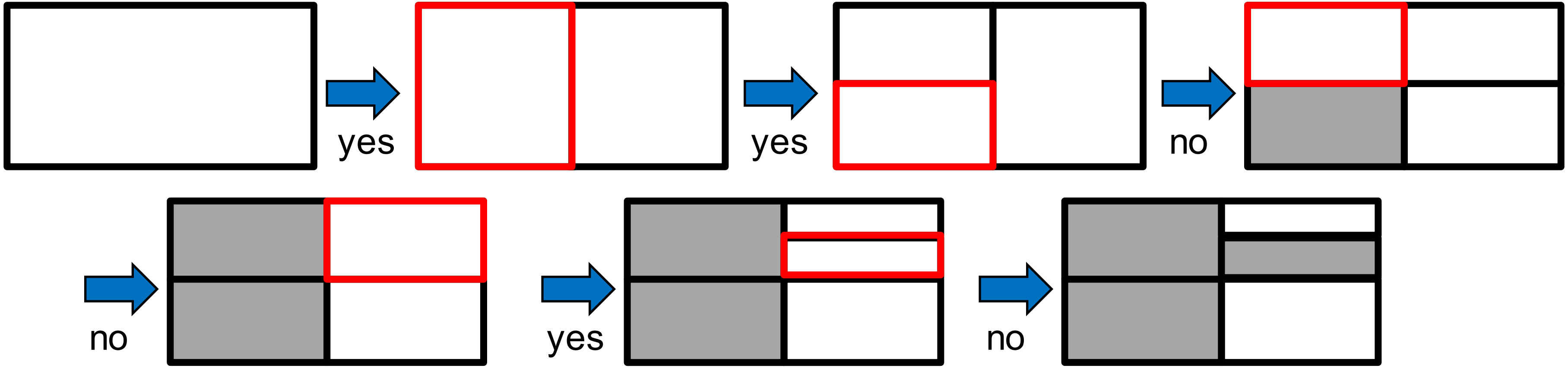
# Branch-and-bound - motivation



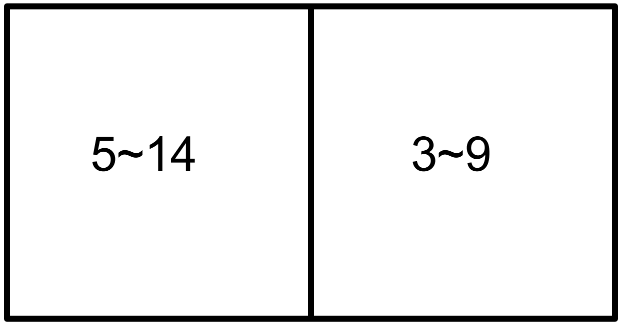
local minimum global minimum

# Branch-and-bound

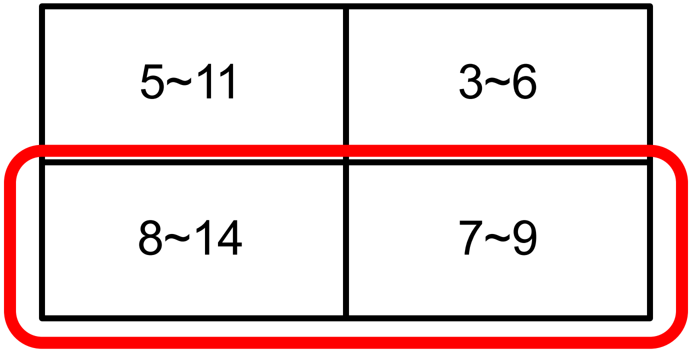
- BnB main idea:
  - Divide the search space into smaller search spaces
  - For each small search space
    - decide if it might contain an optimal (or satisfying) solution
    - Using an upper bound (and/or lower bound)
  - Iterate by splitting the remaining spaces



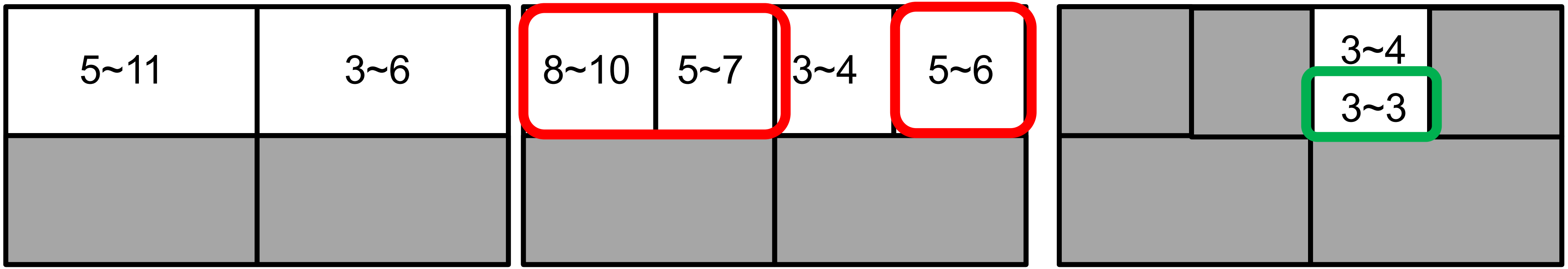
# BnB – a simple example



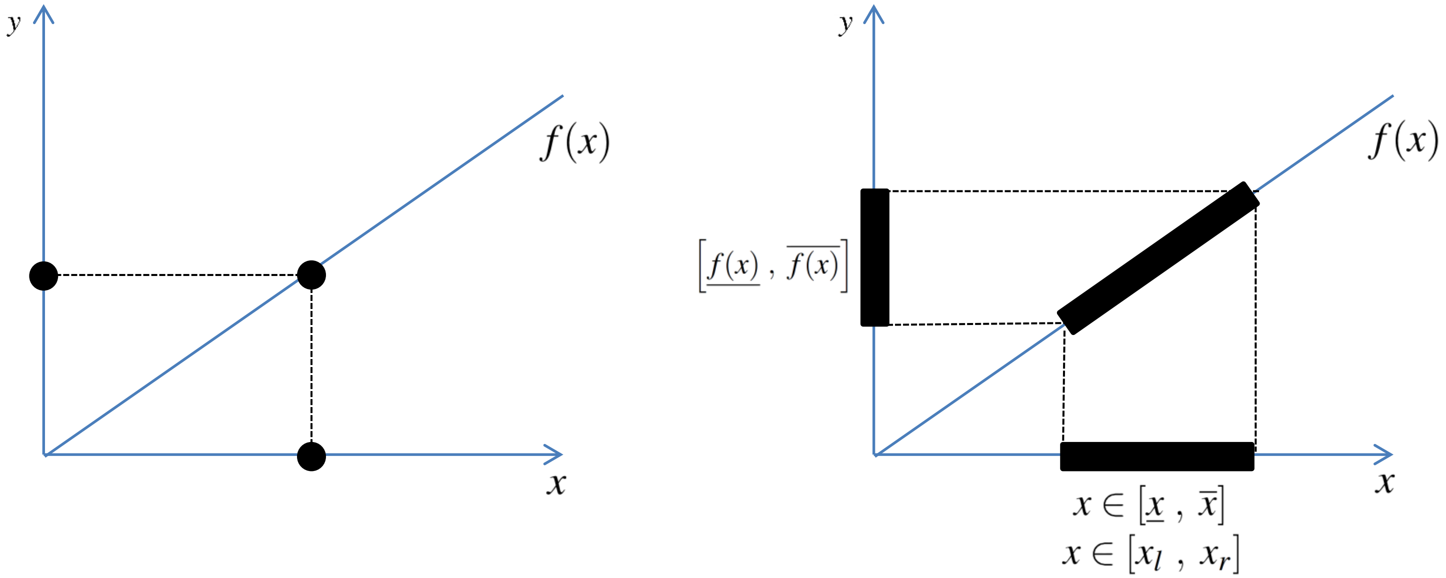

Goal: find the youngest person
Note: the bounds might not be strict!!
Ex: 3~9: the <u>true</u> extremes can be 4 and 5




We are sure the global min is not here (less than 6), so we **discard** these boxes



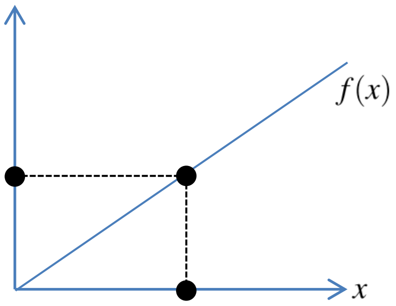
# Intervals

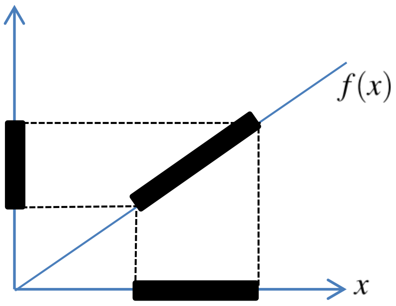


$y$
$f(x)$
$x$
$y$
$f(x)$
$[\underline{f(x)}\ ,\ \overline{f(x)}]$
$x$
$x \in [\underline{x}\ ,\ \overline{x}]$
$x \in [x_l\ ,\ x_r]$

# Intervals

- "Easy" examples
- $f(x)=2x$
  - $x=5$ ➔ $f(x)=2*5=10$
  - $x=[3,6]$ ➔ $f(x)=2*[3,6]=[6,12]$




- $f(x)=2x+3$
  - $x=5$ ➔ $f(x)=2*5+3=13$
  - $x=[3,6]$ ➔ $f(x)=2*[3,6]+3=[9,15]$

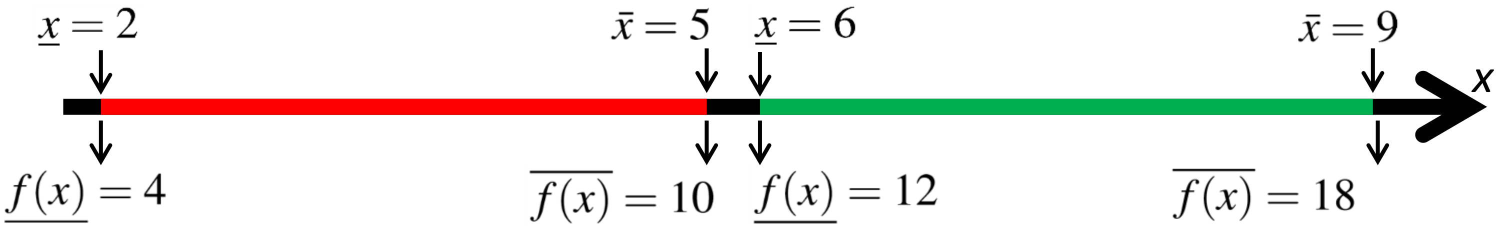
- $f(x,y)=x+y$
  - $x=2$ and $y=3$, $f(x+y)=2+3=5$
  - $x=[2,4]$ and $y=[3,5]$, $f(x+y)=[2,4]+[3,5]=[5,9]$

# BnB – Interval analysis

- Interval analysis


$$[a,b] + [c,d] = [a+c, b+d]$$
$$[a,b] - [c,d] = [a-d, b-c]$$
$$[a,b] \times [c,d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$
$$[a,b] / [c,d] = [\min(\tfrac{a}{c}, \tfrac{a}{d}, \tfrac{b}{c}, \tfrac{b}{d}), \max(\tfrac{a}{c}, \tfrac{a}{d}, \tfrac{b}{c}, \tfrac{b}{d})]$$

$$f(x) = 2x$$



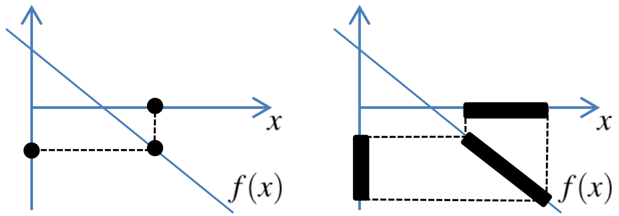
# Intervals

- Examples

- f(x)=-4x
  - x=5 ➔ f(x)=-4*5=-20
  - x=[3,6] ➔ f(x)=-4*[3,6]=[-24,-12]

- f(x)=-4x+10
  - x=5 ➔ f(x)=-4*5+10=-10
  - x=[3,6] ➔ f(x)=-4*[3,6]+10=[-14,-2]

- f(x,y)=x-y
  - x=2 and y=3, f(x-y)=2-3=-1
  - x=[2,4] and y=[3,5], f(x-y)=[2,4]-[3,5]=[-3,1]

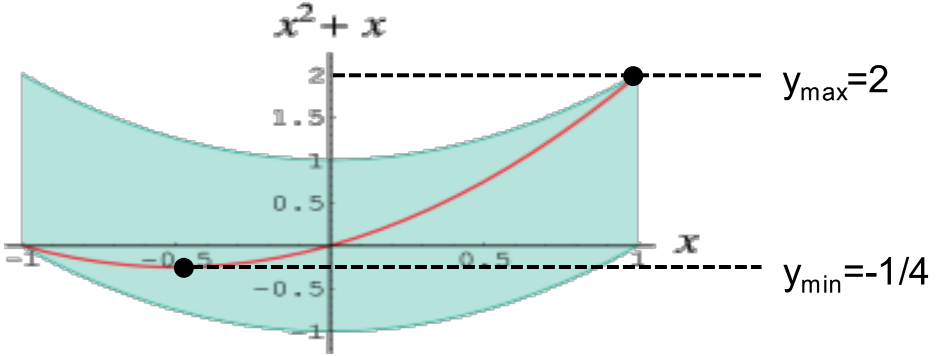



$$[a,b]+[c,d]=[a+c,b+d]$$
$$[a,b]-[c,d]=[a-d,b-c]$$
$$[a,b]\times[c,d]=[\min(ac,ad,bc,bd),\max(ac,ad,bc,bd)]$$
$$[a,b]/[c,d]=[\min(\tfrac{a}{c},\tfrac{a}{d},\tfrac{b}{c},\tfrac{b}{d}),\max(\tfrac{a}{c},\tfrac{a}{d},\tfrac{b}{c},\tfrac{b}{d})]$$

# BnB – Interval analysis

For even $n$

$$[a,b]^n = \begin{cases} [a^n, b^n] \text{ if } a \geq 0 \\ [b^n, a^n] \text{ if } b < 0 \\ [0, \max(a^n, b^n)] \text{ otherwise} \end{cases}$$




$$f(x) = x^2 + x$$

$$x \in [a,b] = [-1,1]$$

$$[-1,1]^2 + [-1,1] = [0, \max(a^n, b^n)] + [-1,1] = [0,1] + [-1,1] = [-1,2]$$

As if $f(x) = x^2 + y$ ➡ $f(x) = (x + \frac{1}{2})^2 - \frac{1}{4}$

$$([-1,1] + \tfrac{1}{2})^2 - \tfrac{1}{4} = [-\tfrac{1}{2}, \tfrac{3}{2}]^2 - \tfrac{1}{4} = [0, \tfrac{9}{4}] - \tfrac{1}{4} = [-\tfrac{1}{4}, 2]$$

# Branch and bound algorithm for IP

$$
\begin{aligned}
\max_{\mathbf{x}}\ & 2x_1 + 3x_2 + x_3 + 2x_4 \\
\text{s.t. } & 5x_1 + 2x_2 + x_3 + x_4 \leq 15 \\
\text{and } & 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60 \\
\text{and } & x_1 + x_2 + x_3 + x_4 \leq 8 \\
\text{and } & 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16 \\
\text{and } & 0 \leq x_1 \leq 3,\ 0 \leq x_2 \leq 7,\ 0 \leq x_3 \leq 5,\ 0 \leq x_4 \leq 5 \\
\text{and } & x_i \in \mathbb{R},\ \text{for all } i = 1, \dots, 4
\end{aligned}
$$

Easy
(Linear Programming)

$$
\begin{aligned}
\max_{\mathbf{x}}\ & 2x_1 + 3x_2 + x_3 + 2x_4 \\
\text{s.t. } & 5x_1 + 2x_2 + x_3 + x_4 \leq 15 \\
\text{and } & 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60 \\
\text{and } & x_1 + x_2 + x_3 + x_4 \leq 8 \\
\text{and } & 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16 \\
\text{and } & x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5] \\
\text{and } & \underline{x_i \in \mathbb{Z},}\ \text{for all } i = 1, \dots, 4
\end{aligned}
$$

More difficult
(Integer Linear Programming)

# Branch and bound algorithm for IP

- Naïve approach
  - Exhaustive search

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$
$$\text{and } x_i \in \mathbb{Z}, \text{ for all } i = 1, \ldots, 4$$

$$[0,3] \times [0,7] \times [0,5] \times [0,5]$$

$$\left.\begin{array}{l} x_1=0, x_2=0, x_3=0, x_4=0 \\ x_1=1, x_2=0, x_3=0, x_4=0 \\ x_1=2, x_2=0, x_3=0, x_4=0 \\ x_1=3, x_2=0, x_3=0, x_4=0 \\ x_1=0, x_2=1, x_3=0, x_4=0 \\ x_1=1, x_2=1, x_3=0, x_4=0 \\ x_1=2, x_2=1, x_3=0, x_4=0 \\ x_1=3, x_2=1, x_3=0, x_4=0 \\ \ldots \\ \ldots \\ x_1=3, x_2=7, x_3=5, x_4=5 \end{array}\right\} \text{4*8*6*6=1152 combinations}$$

$$\boxed{\begin{array}{l} x^*=(0,7,0,0) \\ c^*=21 \end{array}}$$

$$x_i \in \{0,1\}, \text{ for all } i = 1, \ldots, n$$

$2^n$ combinations
n=5 => 32 combinations
n=10=> 1024
n=20 => 1,048,576
n=30 => 1.0737e+09
n=100 => 1.2677e+30

$$x_i \in \{0, \ldots, K\}, \text{ for all } i = 1, \ldots, n$$

$(K+1)^n$ combinations
With K=10
n=5 => 161051 combinations
n=10=> 2.5937e+10
n=20 => 6.7275e+20
n=30 => 1.7449e+31
n=100 => 1.3781e+104

# Branch and bound algorithm for IP

- Naïve approach
  - Rounding of the LP solution

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$
$$\text{and } \underline{x_i \in \mathbb{Z}}, \text{ for all } i = 1, \ldots, 4$$

Integer Linear Programming

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } 0 \leq x_1 \leq 3,\; 0 \leq x_2 \leq 7,\; 0 \leq x_3 \leq 5,\; 0 \leq x_4 \leq 5$$
$$\text{and } \underline{x_i \in \mathbb{R}}, \text{ for all } i = 1, \ldots, 4$$

Relaxation to Linear Programming

LP

x*=(0.08,7,0,0.62)
c*=22.4

rounding

x*=(0,7,0,1)
➔ c*=23

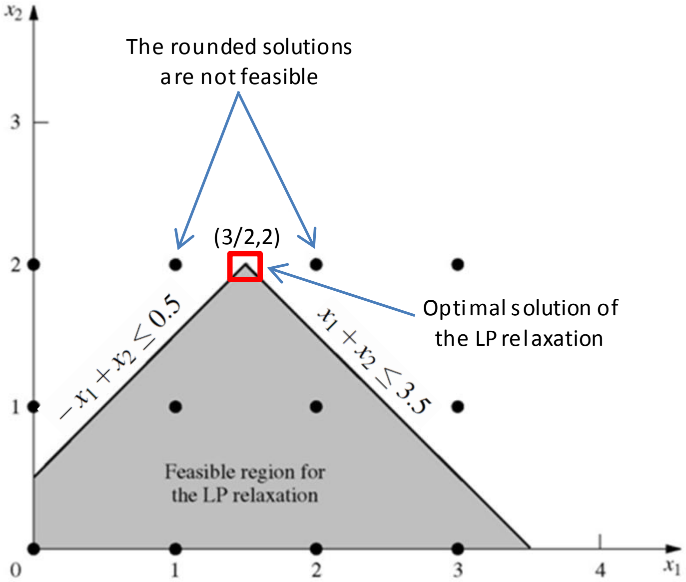
Comparison to the solution of the ILP:

x*=(0,7,0,0)
c*=21

No too far from the optimal solution for this example.
But sometimes, it can be very far from it.

# Branch and bound algorithm for IP

- Naïve approach
  - Rounding of the LP solution



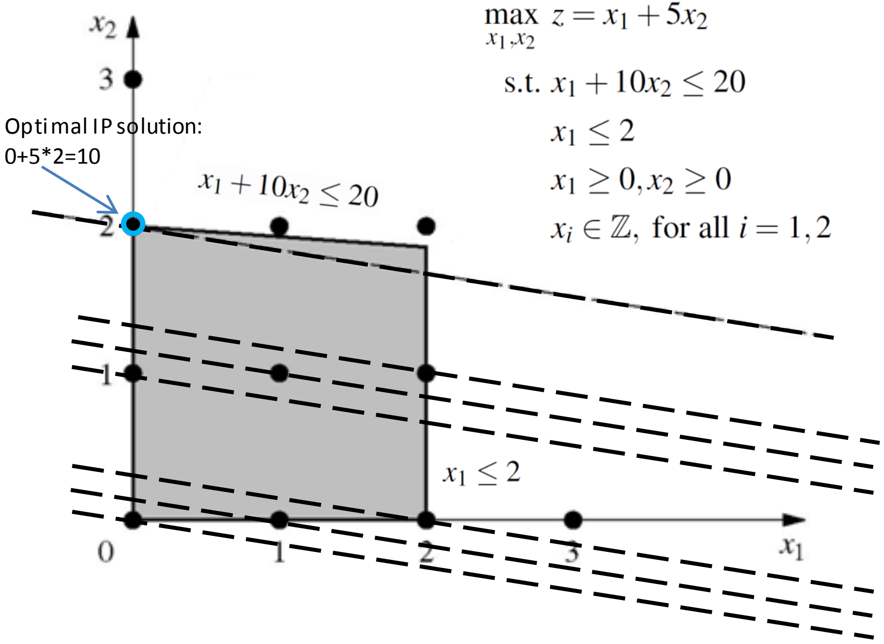

$$\max_{x_1,x_2} \; z = x_2$$
$$\text{s.t.} \quad -x_1 + x_2 \leq 0.5$$
$$x_1 + x_2 \leq 3.5$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$

$$\max_{x_1,x_2} \; z = x_2$$
$$\text{s.t.} \quad -x_1 + x_2 \leq 0.5$$
$$x_1 + x_2 \leq 3.5$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{R}, \text{ for all } i = 1,2$$

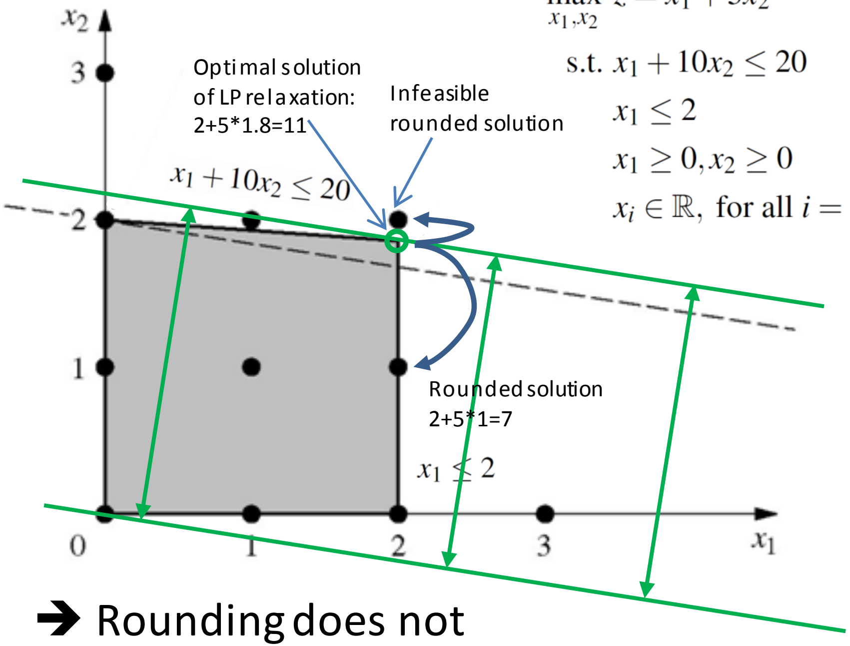
➔ Rounding does not guarantee the **feasibility**

# Branch and bound algorithm for IP

- Naïve approach
  - Rounding of the LP solution




$$\max_{x_1,x_2} \ z = x_1 + 5x_2$$
$$\text{s.t. } x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$

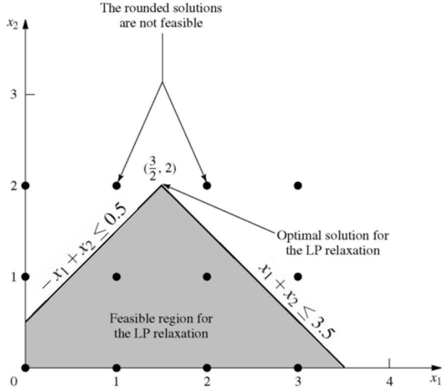



$$\max_{x_1,x_2} \ z = x_1 + 5x_2$$
$$\text{s.t. } x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{R}, \text{ for all } i = 1,2$$

➔ Rounding does not guarantee the **optimality**

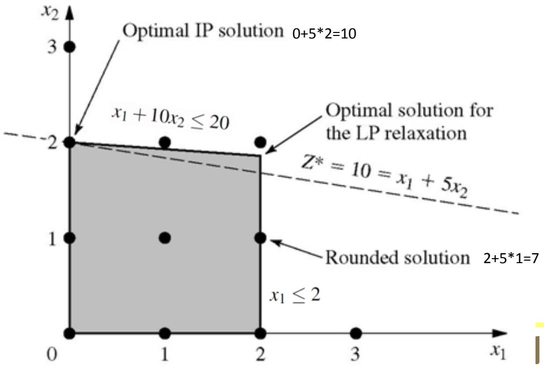
# Branch and bound algorithm for IP

- Naïve approach
  - Rounding of the LP solution




$$\max_{x_1,x_2} z = x_2$$
$$\text{s.t. } -x_1 + x_2 \leq 0.5$$
$$x_1 + x_2 \leq 3.5$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$

➔ Rounding does not guarantee the **feasibility**




$$\max_{x_1,x_2} z = x_1 + 5x_2$$
$$\text{s.t. } x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$

➔ Rounding does not guarantee the **optimality**

# Branch and bound algorithm for IP

For minimization

- **Algorithm:**
- **#1: Initialization.**
  - Set list of problems to $\{P_0\}$.
  - Initialize OPT.
  - Solve LP relaxation of $P_0 \Rightarrow x^*(P_0)$.
  - If $x^*(P_0)$ feasible for $P_0$, OPT = $c^T x^*(P_0)$ and stop.
- **#2: Problem Selection**
  - Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$<OPT . If no such P exists, stop.
- **#3: Variable Selection**
  - Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.
- **#4: Branching**
  - Create two new problems P′ and P″ with $x_p \leq \mathrm{intinf}[x^*_p(P)]$ and $x_p \geq \mathrm{intsup}[x^*_p(P)]$, respectively.
  - Solve continuous relaxations of P′ and P″ $\Rightarrow x^*(P')$, $x^*(P'')$.
  - **Update OPT**
    - If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') <$ OPT $\Rightarrow$ OPT = $c^T x^*(P')$. Same for P ″ .
  - **Further Inspection**
    - If P′ feasible and $c^T x^*(P') <$ OPT $\Rightarrow$ add P′ to list of problems. Same for P″.
- Afterwards, go back to (2).

# Branch and bound algorithm for IP

For maximization

- **Algorithm:**
- **#1: Initialization.**
  - Set list of problems to $\{P_0\}$.
  - Initialize OPT.
  - Solve LP relaxation of $P_0 \Rightarrow x^*(P_0)$.
  - If $x^*(P_0)$ feasible for $P_0$, OPT = $c^T x^*(P_0)$ and stop.
- **#2: Problem Selection**
  - Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0) > \text{OPT}$ . If no such P exists, stop.
- **#3: Variable Selection**
  - Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.
- **#4: Branching**
  - Create two new problems P′ and P″ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.
  - Solve continuous relaxations of P′ and P″ $\Rightarrow x^*(P'), x^*(P'')$.
  - **Update OPT**
    - If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') > \text{OPT} \Rightarrow \text{OPT} = c^T x^*(P')$. Same for P ″ .
  - **Further Inspection**
    - If P′ feasible and $c^T x^*(P') > \text{OPT} \Rightarrow$ add P′ to list of problems. Same for P″.
- Afterwards, go back to (2).

# B&B example

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$
$$\text{and } \underline{x_i \in \mathbb{Z},} \text{ for all } i = 1, \ldots, 4$$

Original (integer) system

$\Rightarrow$

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } 0 \leq x_1 \leq 3, \; 0 \leq x_2 \leq 7, \; 0 \leq x_3 \leq 5, \; 0 \leq x_4 \leq 5$$
$$\text{and } \underline{x_i \in \mathbb{R},} \text{ for all } i = 1, \ldots, 4$$

Relaxed system

# B&B example

**#1: Initialization.**

Set list of problems to $\{P_0\}$.
Initialize OPT.
- objective function value of best

Solve LP relaxation of $P_0 \Rightarrow x^*(P_0)$.
If $x^*(P_0)$ feasible for $P_0$, OPT = $c^T x^*(P_0)$ and stop.

$x^*(P_0)=(0.08,7,0,0.62)$
$c^*(P_0)=22.4$

$P_0$

$$
\begin{aligned}
\max_{\mathbf{x}} \ & 2x_1 + 3x_2 + x_3 + 2x_4 \\
\text{s.t. } & 5x_1 + 2x_2 + x_3 + x_4 \leq 15 \\
\text{and } & 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60 \\
\text{and } & x_1 + x_2 + x_3 + x_4 \leq 8 \\
\text{and } & 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16 \\
\text{and } & 0 \leq x_1 \leq 3,\ 0 \leq x_2 \leq 7,\ 0 \leq x_3 \leq 5,\ 0 \leq x_4 \leq 5 \\
\text{and } & x_i \in \mathbb{R},\ \text{for all } i = 1, \ldots, 4
\end{aligned}
$$

Problem list = $\{P_0\}$
Opt=$-\infty$

# B&B example

**#2: Problem Selection**

Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$<OPT.

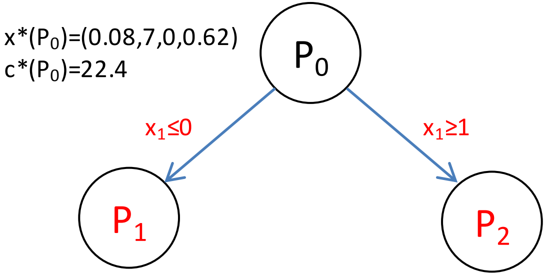
If no such P exists, stop.

**#3: Variable Selection**

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

**#4: Branching**

Create two new problems P' and P'' with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.




Problem list = $\{P_1,P_2\}$

Opt=$-\infty$

# B&B example

**#4: Branching**

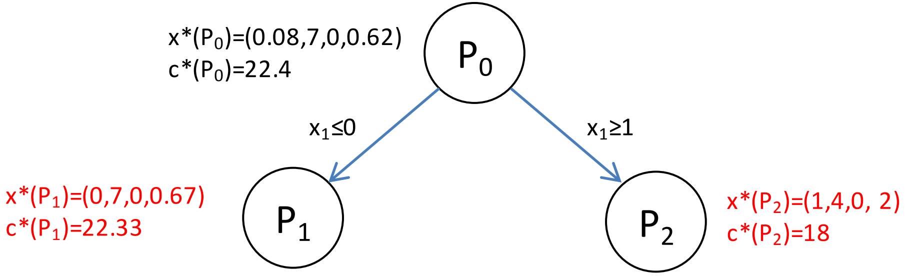
Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.
Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

**Update OPT**

If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') <$ OPT ⇒ OPT = $c^T x^*(P')$. Same for P′′.

**Further Inspection**

If P′ feasible and $c^T x^*(P') <$ OPT ⇒ add P′ to list of problems. Same for P′′.




Problem list = $\{P_1\}$
Opt=18

# B&B example

**#2: Problem Selection**

Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$<OPT.

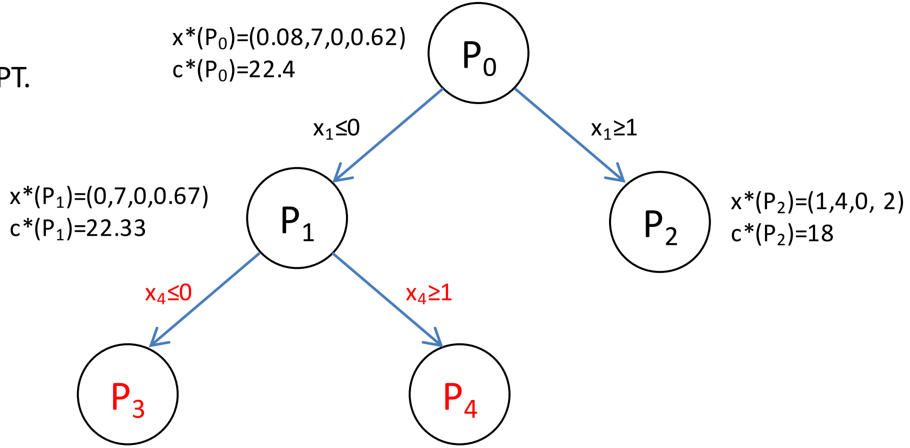
If no such P exists, stop.

**#3: Variable Selection**

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

**#4: Branching**

Create two new problems P' and P'' with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.




Problem list = $\{P_3, P_4\}$

Opt=18

# B&B example

**#4: Branching**

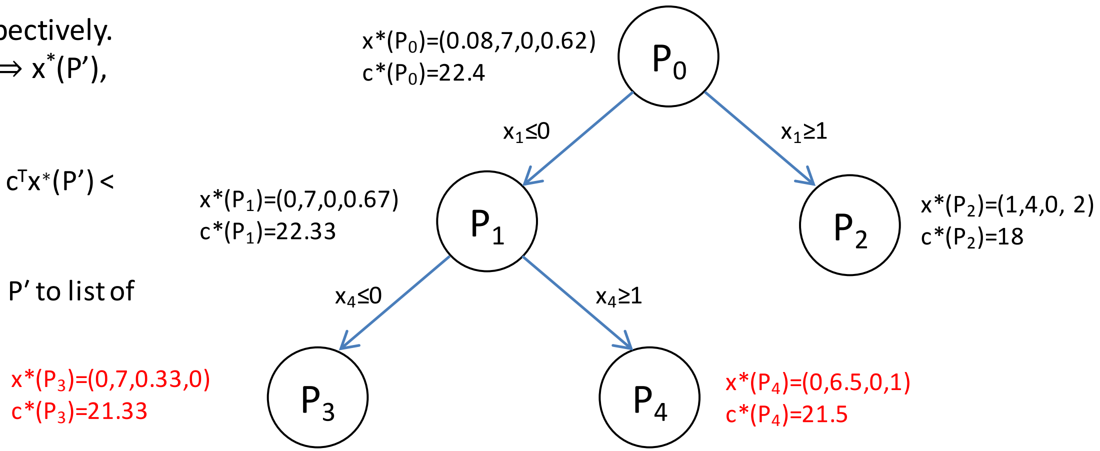
Create two new problems P′ and P′′ with $x_p \le \text{intinf}[x^*_p(P)]$ and $x_p \ge \text{intsup}[x^*_p(P)]$, respectively. Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

**Update OPT**

If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') <$ OPT ⇒ OPT = $c^T x^*(P')$. Same for P′′.

**Further Inspection**

If P′ feasible and $c^T x^*(P') <$ OPT ⇒ add P′ to list of problems. Same for P′′.




Problem list = $\{P_3, P_4\}$

Opt=18

# B&B example

**#2: Problem Selection**

Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$<OPT.

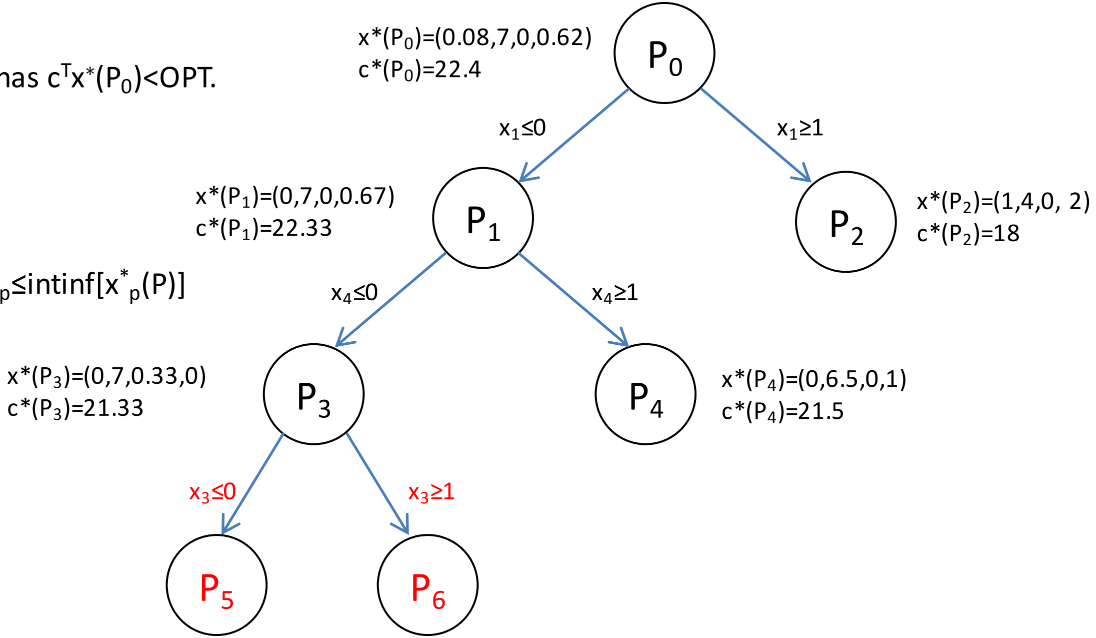
If no such P exists, stop.

**#3: Variable Selection**

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

**#4: Branching**

Create two new problems P' and P'' with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.




Problem list = $\{P_4, P_5, P_6\}$

Opt=18

# B&B example

**#4: Branching**

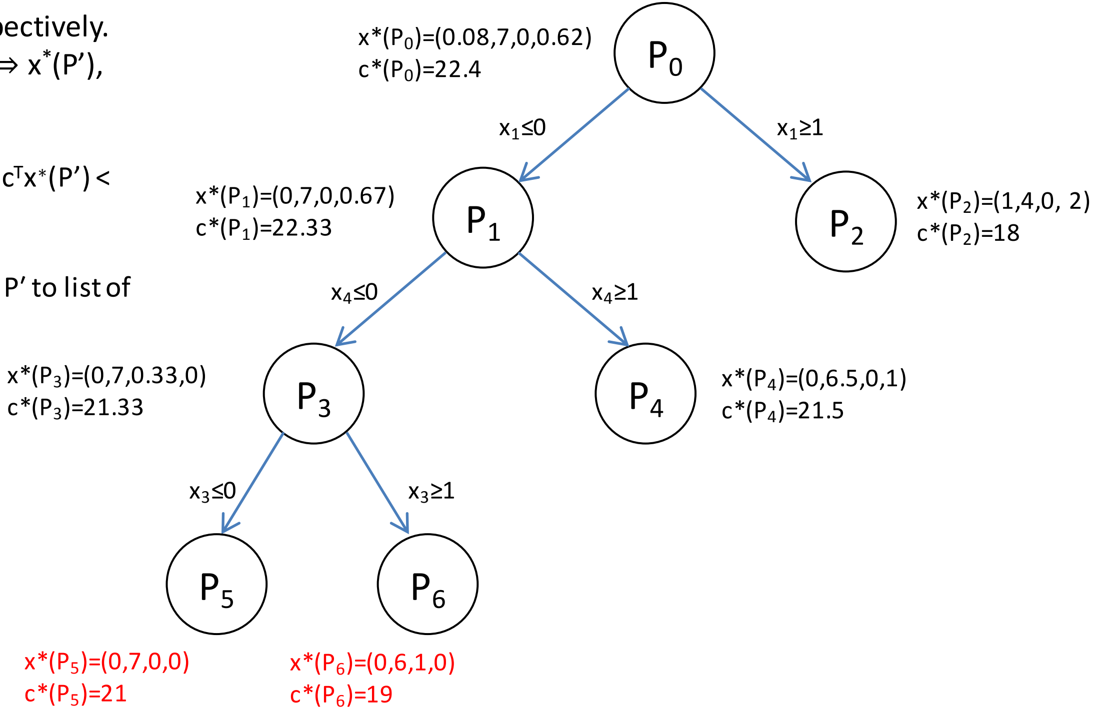
Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively. Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

**Update OPT**

If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') <$ OPT ⇒ OPT = $c^T x^*(P')$. Same for P′′.

**Further Inspection**

If P′ feasible and $c^T x^*(P') <$ OPT ⇒ add P′ to list of problems. Same for P′′.




Problem list = $\{P_4\}$

Opt=21

# B&B example

**#2: Problem Selection**

Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0) < OPT$.

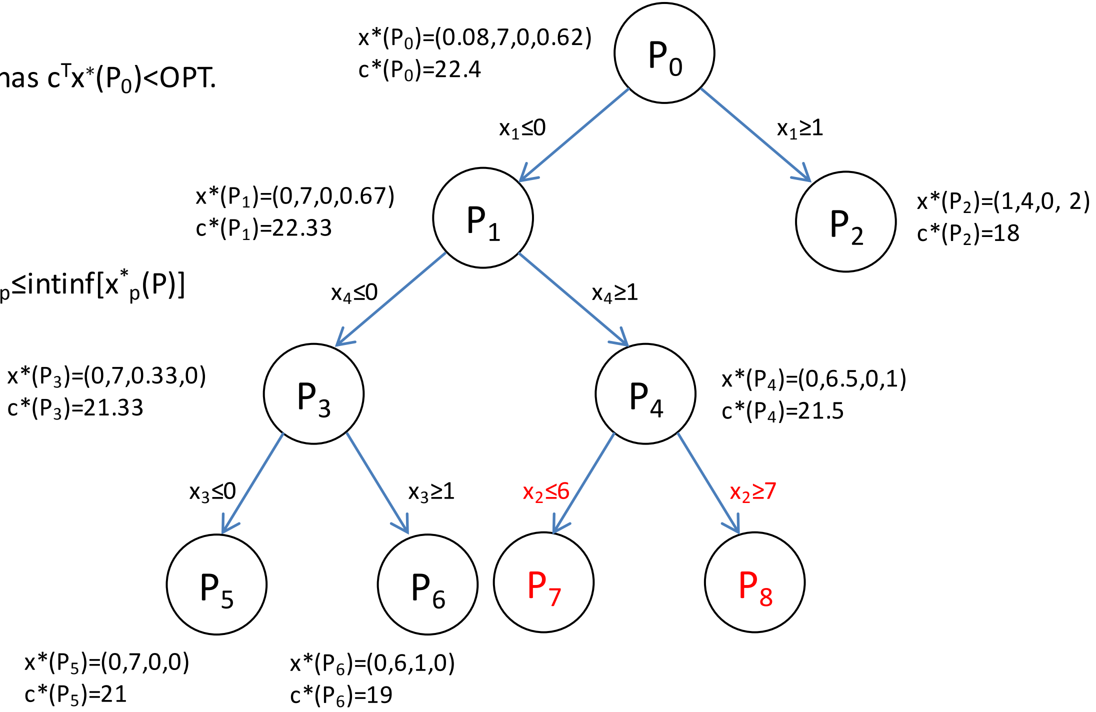
If no such P exists, stop.

**#3: Variable Selection**

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

**#4: Branching**

Create two new problems P' and P'' with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.




Problem list = $\{P_7, P_8\}$

Opt=21

# B&B example

**#4: Branching**

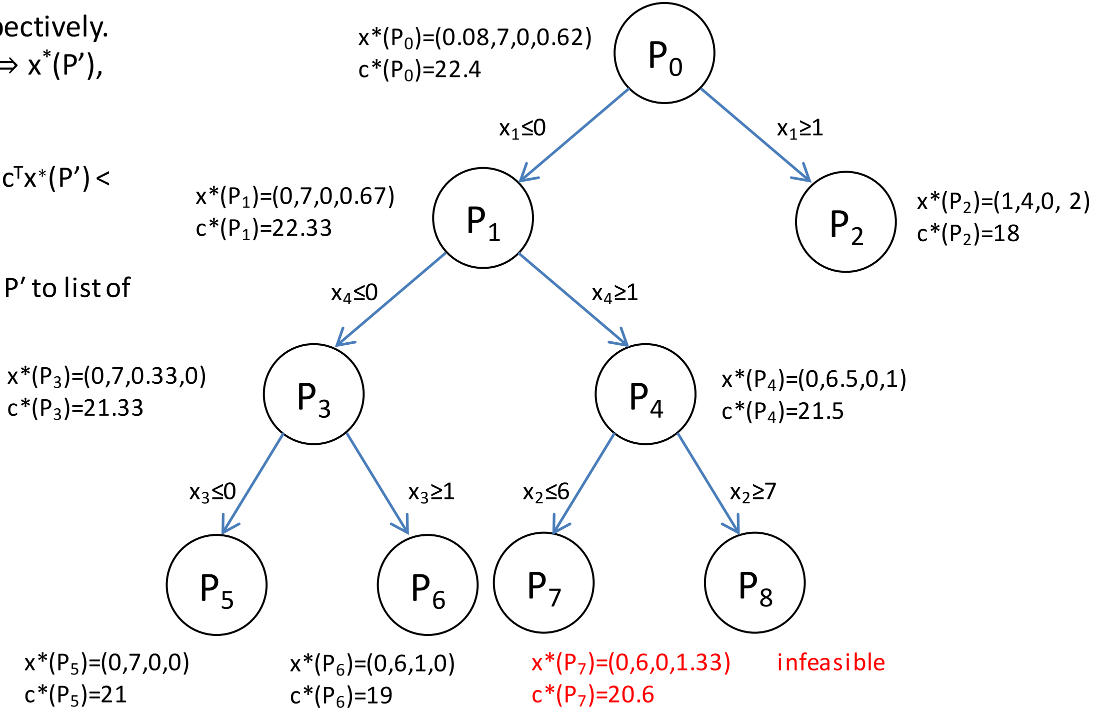
Create two new problems P′ and P′′ with $x_p \le \text{intinf}[x^*_p(P)]$ and $x_p \ge \text{intsup}[x^*_p(P)]$, respectively. Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

**Update OPT**

If P′ feasible, $x^*(P')$ feasible for $P_0$ and $c^T x^*(P') <$ OPT ⇒ OPT = $c^T x^*(P')$. Same for P′′.

**Further Inspection**

If P′ feasible and $c^T x^*(P') <$ OPT ⇒ add P′ to list of problems. Same for P′′.

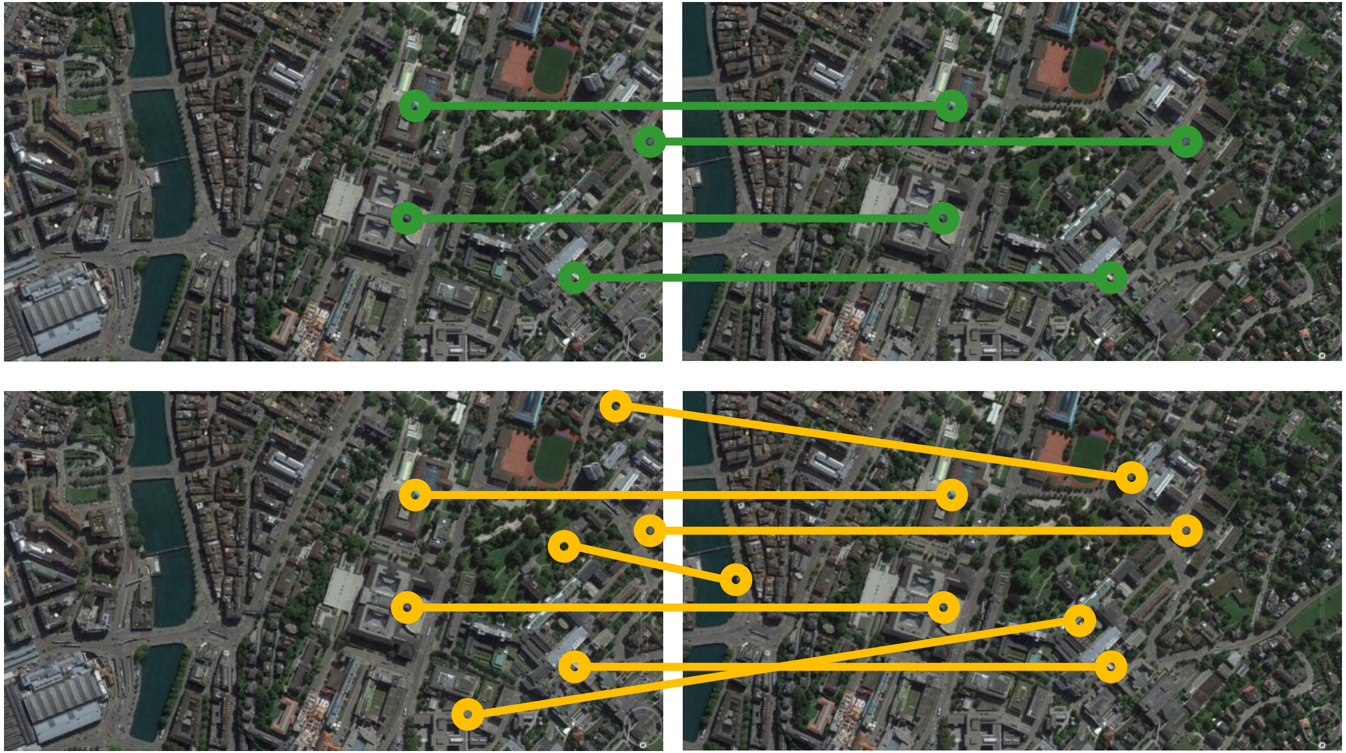



Problem list = {}

Opt=21

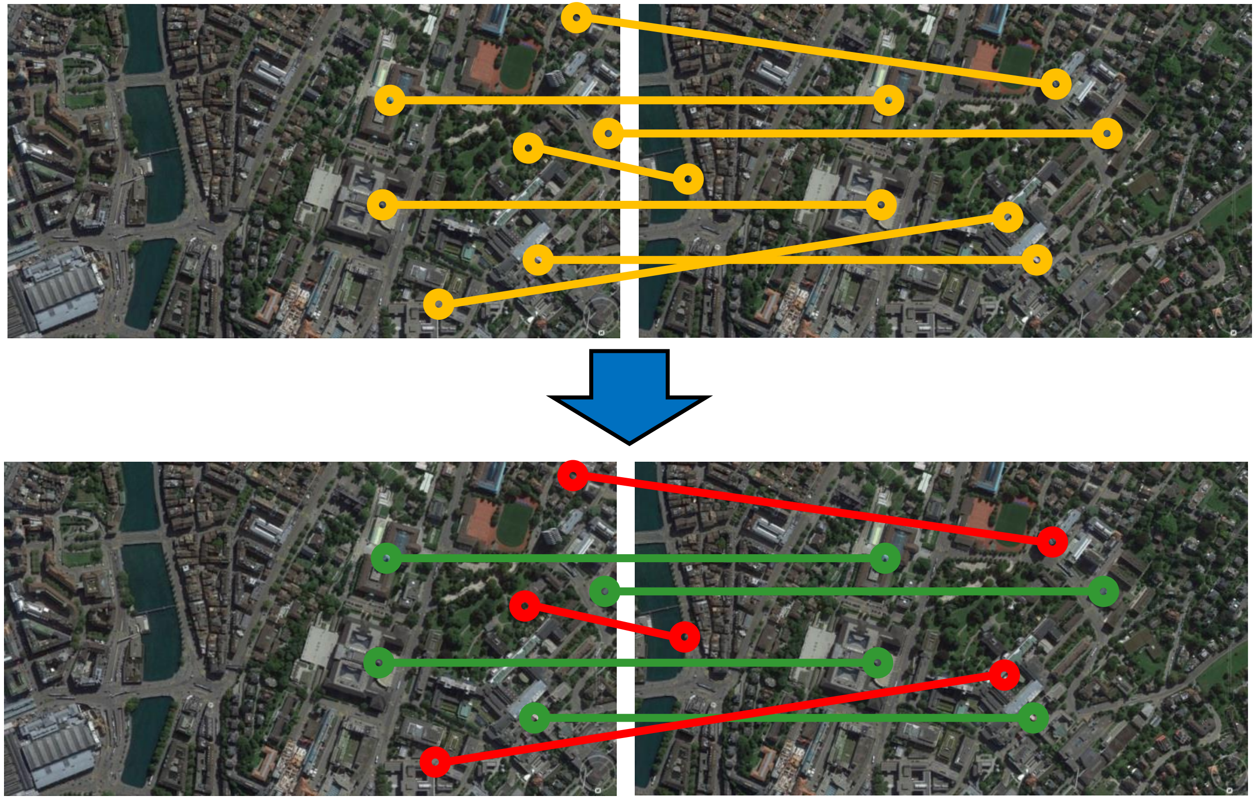
$x^*=(0,7,0,0)$

# BnB – simple bounds



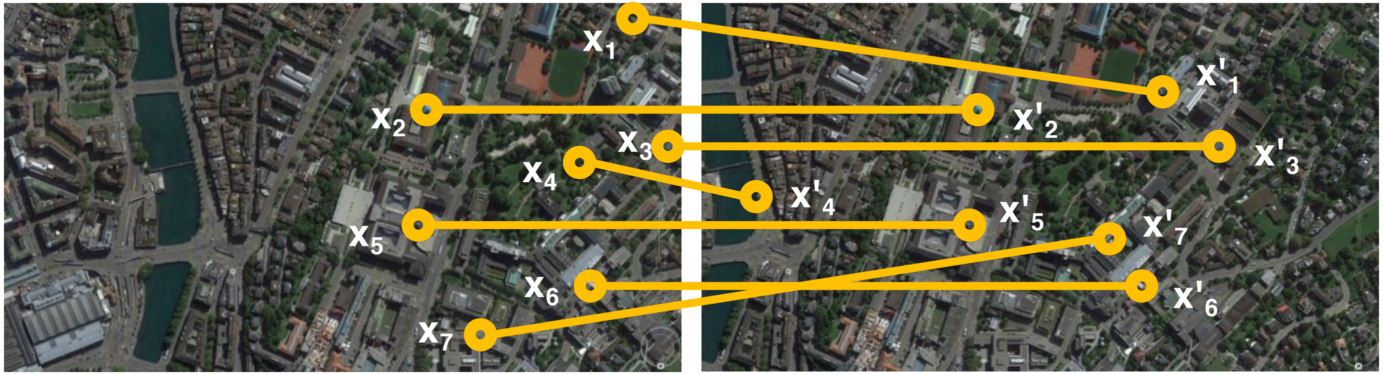
From Google Earth views

# BnB – simple bounds



From Google Earth views

# BnB – simple bounds




Motion model (1D translation) $x'_i = x_i + T$

Inlier definition $d(x_i, x'_i) \leq \varepsilon$

Model bounds $T_l \leq T \leq T_u$ $\quad -1 \leq T \leq 3$

$-1 \leq T \leq 3$ FAIL!

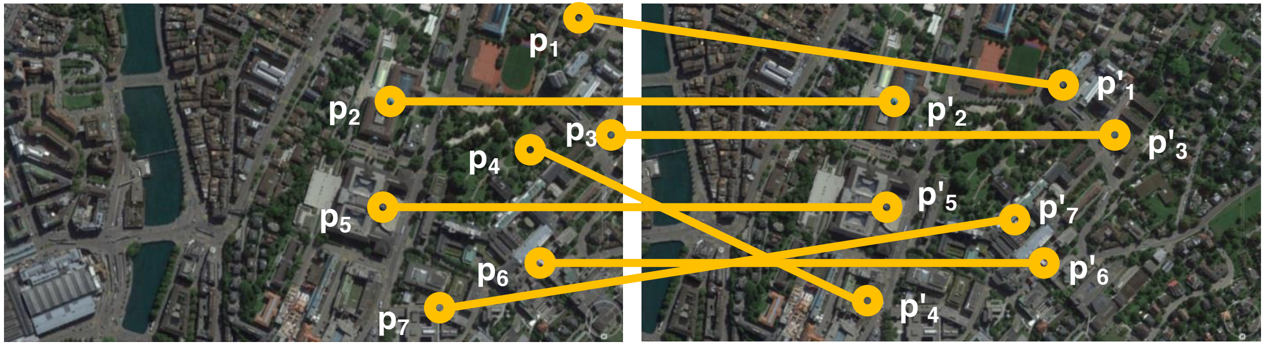
Example $|x'_i - (x_i + T)| \leq \varepsilon$ $\quad |15 - (9 + T)| \leq 2$

$-1 \leq T \leq 6$

$-1 \leq T \leq 4$

# BnB – simple bounds

$p_1=(950, 20)$ $p'_1=(700, 130)$
$p_2=(640, 160)$ $p'_2=(420, 160)$
$p_3=(1000, 210)$ $p'_3=(780, 210)$
$p_4=(875, 245)$ $p'_4=(400, 480)$
$p_5=(630, 330)$ $p'_5=(410, 330)$
$p_6=(890, 430)$ $p'_6=(670, 430)$
$p_7=(725, 500)$ $p'_7=(620, 350)$



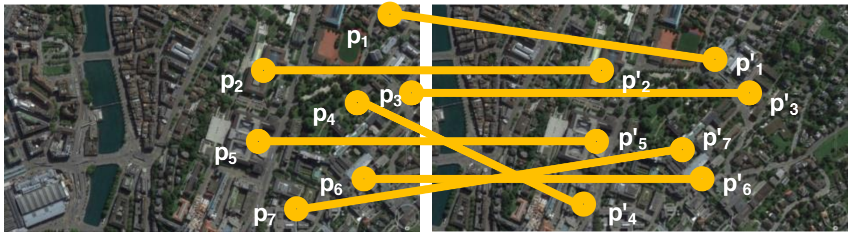

Motion model (2D translation) $(x'_i, y'_i) = (x_i, y_i) + (T_x, T_y)$

Inlier definition $d(x_i, x'_i) = |x'_i - (x_i + T_x)| \leq \varepsilon$

$$d(y_i, y'_i) = |y'_i - (y_i + T_y)| \leq \varepsilon$$

$$\varepsilon = 10 \text{ pixels}$$

# BnB – simple bounds




Motion model (2D translation) $(x'_i, y'_i) = (x_i, y_i) + (T_x, T_y)$

Inlier definition $d(x_i, x'_i) = |x'_i - (x_i + T_x)| \le \varepsilon$

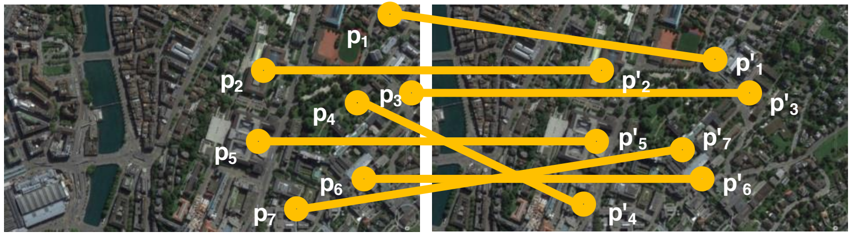
$d(y_i, y'_i) = |y'_i - (y_i + T_y)| \le \varepsilon$

$\varepsilon = 10$ pixels

| | | $T_x$ | $T_y$ | $-1000 \le T_x \le 1000$<br>$-1000 \le T_y \le 1000$ | $T_x=0$ in [-1000 , 1000]<br>$T_y=0$ in [-1000 , 1000] |
|---|---|---|---|---|---|
| $p_1$=(950 , 20) | $p'_1$=(700 , 130) | -250<br>[-260,-240] | +110<br>[+100,+120] | ✓ | FAIL! |
| $p_2$=(640 , 160) | $p'_2$=(420 , 160) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | FAIL! |
| $p_3$=(1000 , 210) | $p'_3$=(780 , 210) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | FAIL! |
| $p_4$=(875 , 245) | $p'_4$=(400 , 480) | -475<br>[-485,-465] | 235<br>[225,245] | ✓ | FAIL! |
| $p_5$=(630 , 330) | $p'_5$=(410 , 330) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | FAIL! |
| $p_6$=(890 , 430) | $p'_6$=(670 , 430) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | FAIL! |
| $p_7$=(725 , 500) | $p'_7$=(620 , 350) | -105<br>[-115,-95] | -150<br>[-160,-140] | ✓ | FAIL! |

➔ between 0 and 7 inliers

# BnB – simple bounds




Motion model (2D translation) $(x'_i, y'_i) = (x_i, y_i) + (T_x, T_y)$

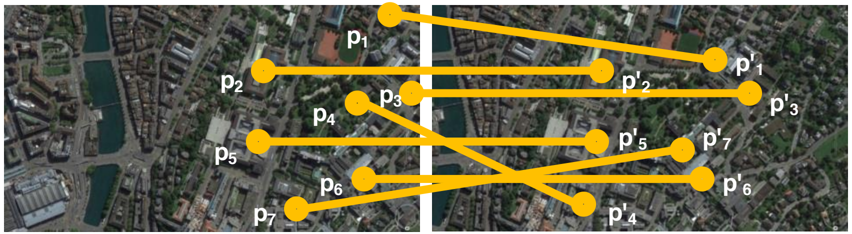
Inlier definition $d(x_i, x'_i) = |x'_i - (x_i + T_x)| \le \varepsilon$

$d(y_i, y'_i) = |y'_i - (y_i + T_y)| \le \varepsilon$

| | | $T_x$ | $T_y$ | $-450 \le T_x \le 0$<br>$-1000 \le T_y \le 1000$ | $T_x$=-225 in [-450, 0]<br>$T_y$=0 in [-1000 , 1000] |
|---|---|---|---|---|---|
| $p_1$=(950 , 20) | $p'_1$=(700 , 130) | -250<br>[-260,-240] | +110<br>[+100,+120] | ✓ | FAIL! |
| $p_2$=(640 , 160) | $p'_2$=(420 , 160) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | ✓ |
| $p_3$=(1000 , 210) | $p'_3$=(780 , 210) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | ✓ |
| $p_4$=(875 , 245) | $p'_4$=(400 , 480) | -475<br>[-485,-465] | 235<br>[225,245] | FAIL! | FAIL! |
| $p_5$=(630 , 330) | $p'_5$=(410 , 330) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | ✓ |
| $p_6$=(890 , 430) | $p'_6$=(670 , 430) | -220<br>[-230,-210] | 0<br>[-10,10] | ✓ | ✓ |
| $p_7$=(725 , 500) | $p'_7$=(620 , 350) | -105<br>[-115,-95] | -150<br>[-160,-140] | ✓ | FAIL! |

➔ between 4 and 6 inliers

# BnB – simple bounds




Motion model (2D translation) $(x'_i, y'_i) = (x_i, y_i) + (T_x, T_y)$

Inlier definition $d(x_i, x'_i) = |x'_i - (x_i + T_x)| \leq \varepsilon$

$d(y_i, y'_i) = |y'_i - (y_i + T_y)| \leq \varepsilon$

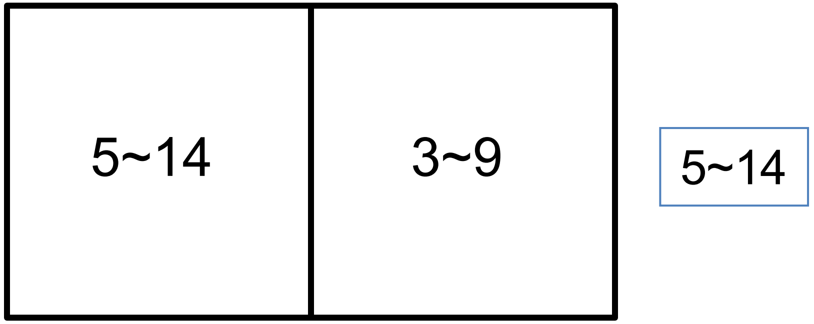
| | | $T_x$ | $T_y$ |
|---|---|---|---|
| $p_1$=(950 , 20) | $p'_1$=(700 , 130) | -250<br>[-260,-240] | +110<br>[+100,+120] |
| $p_2$=(640 , 160) | $p'_2$=(420 , 160) | -220<br>[-230,-210] | 0<br>[-10,10] |
| $p_3$=(1000 , 210) | $p'_3$=(780 , 210) | -220<br>[-230,-210] | 0<br>[-10,10] |
| $p_4$=(875 , 245) | $p'_4$=(400 , 480) | -475<br>[-485,-465] | 235<br>[225,245] |
| $p_5$=(630 , 330) | $p'_5$=(410 , 330) | -220<br>[-230,-210] | 0<br>[-10,10] |
| $p_6$=(890 , 430) | $p'_6$=(670 , 430) | -220<br>[-230,-210] | 0<br>[-10,10] |
| $p_7$=(725 , 500) | $p'_7$=(620 , 350) | -105<br>[-115,-95] | -150<br>[-160,-140] |

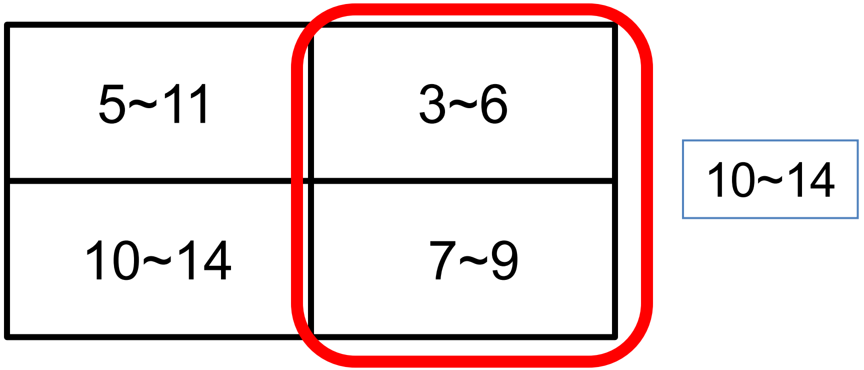
$-450 \leq T_x \leq 0$
$-30 \leq T_y \leq 30$


$T_x$=-225 in [-450, 0]
$T_y$=0 in [-30 , 30]


➔ between 4 and 4 inliers

# BnB – max cardinality



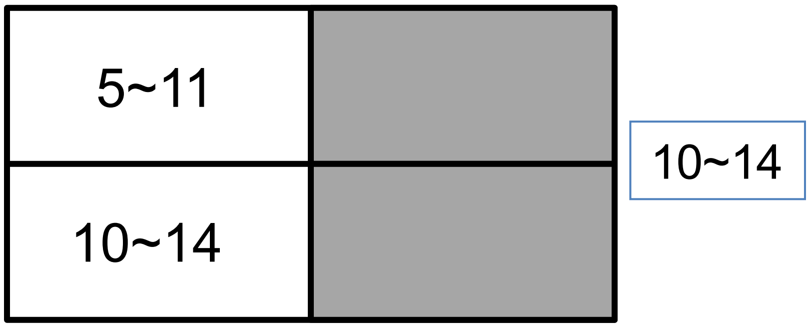



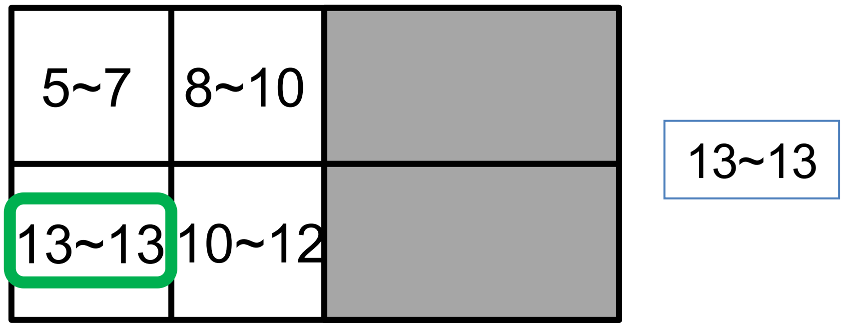

Goal: find the max cardinality

Note: the bounds might not be strict!!

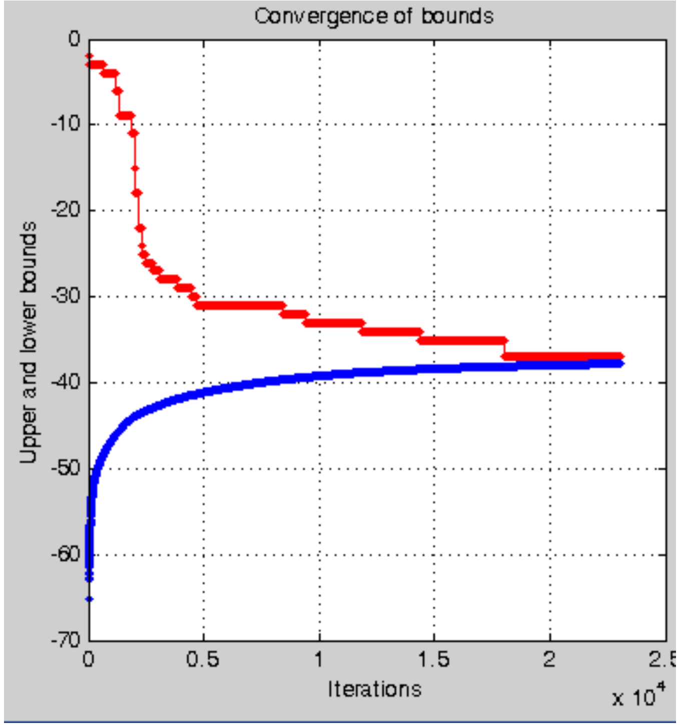
Ex: 3~9: the true extremes can be 4 and 5

We are sure the global max is not here, so we **discard** these boxes

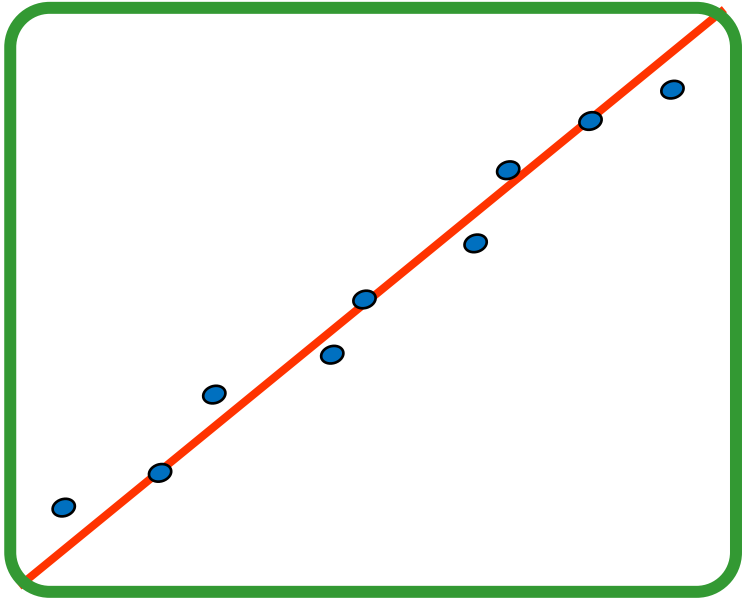


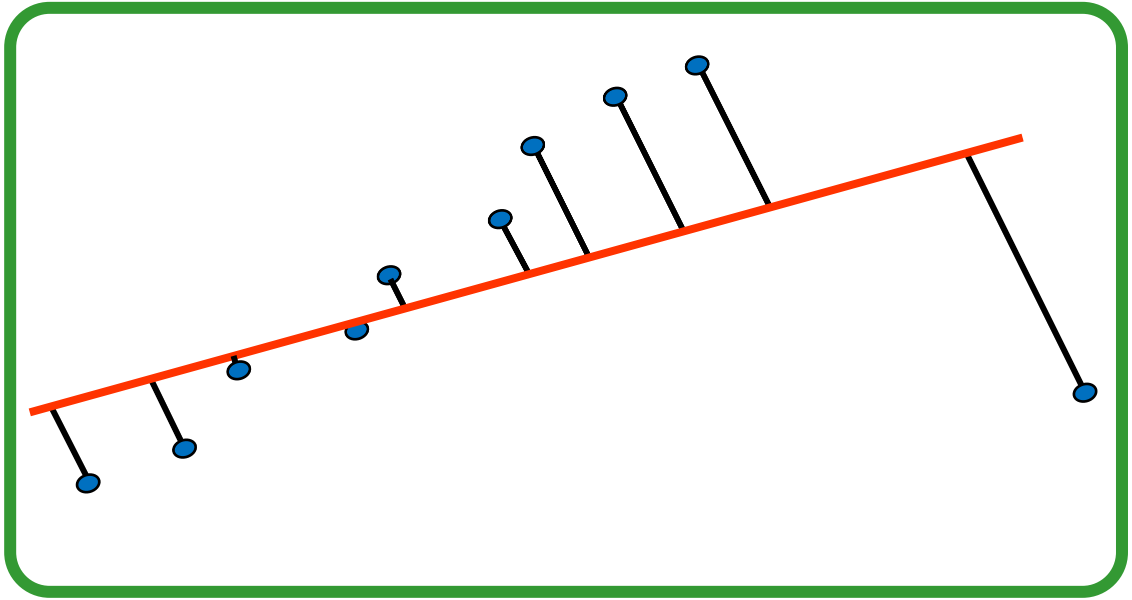
# BnB – max cardinality



# BnB - a motivation example



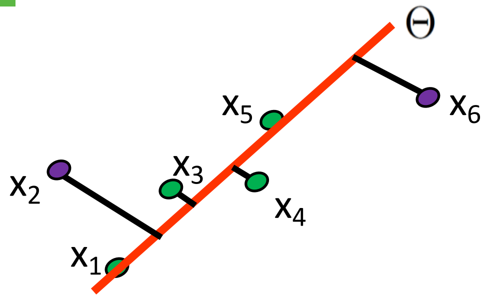
Only **inliers**



Effects due to a single **outlier**

# Mathematical formulation


- Input: data $\mathcal{S}$
  - We note $x_i$ the *i*-th data point
- Output: model $\Theta$ and inliers/outliers ($\mathcal{S}_I$ , $\mathcal{S}_O$ )
  - More precisely, the “model leading to the highest nb of inliers”
- What are the unknows?
  - The model $\Theta$
  - The set of inliers/outliers ($\mathcal{S}_I$ , $\mathcal{S}_O$)
- Cost: to distinguish inliers/outliers
  - An inlier has a “small” cost, *i.e.* $f(\Theta, x_i) \leq T$



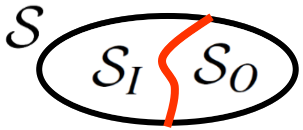

$$\max_{\mathcal{S}_\mathrm{I},\Theta} \quad \mathbf{card}\ (\mathcal{S}_\mathrm{I})$$

maximize the nb of inliers

$$\text{s.t.} \quad f(\Theta, x_i) \leq T, \forall i \in \mathcal{S}_\mathrm{I}$$

all the inliers must have a “small” error

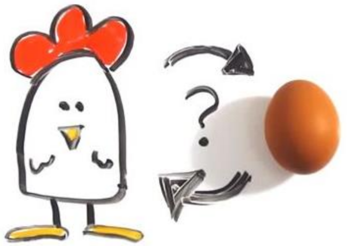
# Chicken-and-egg problem




$$\max_{\mathcal{S}_I,\theta} \quad card(\mathcal{S}_I)$$
$$\text{s.t.} \quad g(x_i,\theta) \leq T, \forall i \in \mathcal{S}_I$$

nb of inliers

all the inliers must have a "small" error

$\mathcal{S}_I$  $\theta$

Test **all** $\mathcal{S}_I$

➔ Finite ($2^N$) but intractable

Test **all** $\theta$

➔ Non-finite (continuous space) and intractable

A "counter-intuitive" approach

$$[\, \underline{\theta} \,,\, \overline{\theta} \,] \rightarrow \begin{matrix} \mathcal{S}_I \\ \theta \end{matrix}$$

# Mathematical formulation

- Initial formulation

$$\max_{\mathcal{S}_{\mathcal{I}},\Theta} \quad \mathbf{card}\left(\mathcal{S}_{\mathcal{I}}\right)$$
$$\text{s.t.} \qquad f(\Theta, x_i) \leq T, \forall i \in \mathcal{S}_{\mathcal{I}}$$

- Alternative formulation

$$\max_{\mathbf{z},\Theta} \quad \sum_{i=1}^{N} z_i$$
$$\text{s.t.} \quad z_i f(\Theta, x_i) \leq z_i T, \forall i = 1..N$$
$$\text{and} \quad z_i \in \{0, 1\}, \forall i = 1..N$$

Let $z_i = 1$ if the i[th] data point is an inlier → $f(\Theta, x_i) \leq T$
$= 0$ else → $0 \leq 0$

# Mathematical formulation

- Line fitting
  - Model: $\Theta = (a, b, c)^T$
  - Cost function: $|\mathbf{A}_i^T\Theta| = |ax_i + by_i + c|$

$$\max_{\mathbf{z},\Theta} \sum_{i=1}^{N} z_i$$

$$\text{s.t.} \quad z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N$$ geometric

$$\text{and} \quad ||\Theta|| = 1$$ scale

$$\text{and} \quad z_i \in \{0, 1\}, \forall i = 1..N$$ binary

Hard to solve: binary/continuous unknowns and bilinearities

# Mathematical formulation

- Interesting observation
  - Relaxation is equivalent!
  - Due to the total unimodularity [Berclaz PAMI'11]

$$
\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & \|\Theta\| = 1 \\
\text{and} \quad & \boxed{z_i \in \{0,1\}}, \forall i = 1..N
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & \|\Theta\| = 1 \\
\text{and} \quad & \boxed{0 \leq z_i \leq 1}, \forall i = 1..N
\end{aligned}
$$

Still hard to solve: bilinear, non-convex, etc

"Multiple Object Tracking using K-Shortest Paths Optimization" (see Appendix A), J. Berclaz, F. Fleuret, E. Turetken, P. Fua, PAMI,2011

# Envelopes for bilinearities

$$\begin{aligned} \max_{x,y} \quad & x + 2y \\ \text{s.t.} \quad & 2x + 3y \leq 20 \\ & 3 \leq x \leq 5 \\ & 2 \leq y \leq 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

Linear Programming (simple)

$$\begin{aligned} \max_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ \text{and} \quad & x \geq 0 \end{aligned}$$

$$\begin{aligned} \max_{x,y} \quad & x + 2y \\ \text{s.t.} \quad & 2\underline{xy} + 3y \leq 20 \\ & 3 \leq x \leq 5 \\ & 2 \leq y \leq 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

With bilinearities $xy$

# Envelopes for bilinearities

- Bilinear relaxation: by concave and convex envelopes

bilinear equality $\gamma = \alpha\beta$ with the bounding-boxes $[\underline{\alpha}, \bar{\alpha}]$ and $[\underline{\beta}, \bar{\beta}]$ relaxed by the following envelopes:

$$\gamma \geq \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$

$$\gamma \leq \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

We collectively represent them as $conv(\alpha, \beta) \leq \gamma \leq conc(\alpha, \beta)$

- Example:

$$\alpha \in [3,5], \beta \in [2,7]$$

$$\gamma \geq \max(3\beta + 2\alpha - 3*2, 5\beta + 7\alpha - 5*7)$$
$$\gamma \leq \min(5\beta + 2\alpha - 5*2, 3\beta + 7\alpha - 3*7)$$

$$\gamma \geq 3\beta + 2\alpha - 3*2$$
$$\gamma \geq 5\beta + 7\alpha - 5*7$$
$$\gamma \leq 5\beta + 2\alpha - 5*2$$
$$\gamma \leq 3\beta + 7\alpha - 3*7$$

# Envelopes for bilinearities

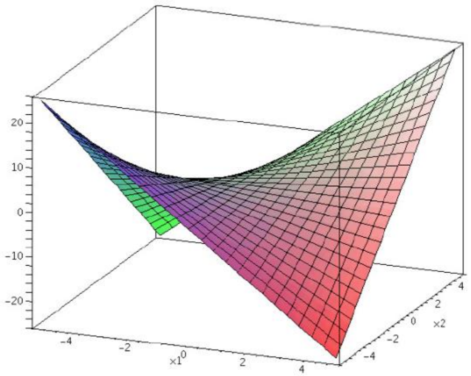
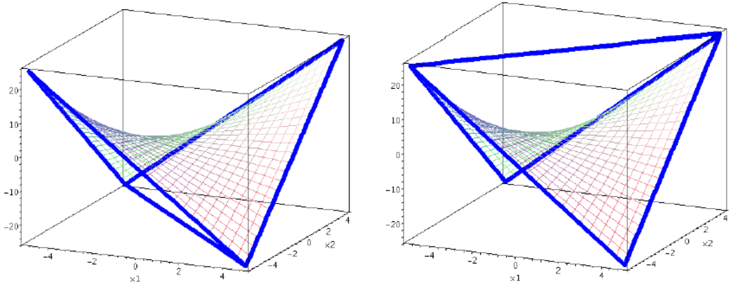


Fig. 1. The bilinear surface $w(x_1, x_2) = x_1 x_2$.



"above" these two planes "below" these two planes

Fig. 2. Lower convex (left) and upper concave (right) envelopes for the bilinear term.

$$
\begin{aligned}
w_{12} &\geq x_1^L x_2 + x_2^L x_1 - x_1^L x_2^L \\
w_{12} &\geq x_1^U x_2 + x_2^U x_1 - x_1^U x_2^U \\
w_{12} &\leq x_1^L x_2 + x_2^U x_1 - x_1^L x_2^U \\
w_{12} &\leq x_1^U x_2 + x_2^L x_1 - x_1^U x_2^L
\end{aligned}
$$


G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. Mathematical Programming, 10:146–175, 1976.

F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. Math- ematics of Operations Research, 8(2):273–286, 1983.

"Relaxations of multilinear convex envelopes: dual is better than primal", Alberto Costa, Leo Liberti, International conference on Experimental Algorithms, 2012

# Envelopes for bilinearities

$$\boxed{\begin{aligned} \max_{x,y} \;& x + 2y \\ \text{s.t. } & 2xy + 3y \le 20 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}}$$

$$\Rightarrow$$

$$x \in [3,5],\; y \in [2,7]$$

$$\boxed{w = xy}$$

$$\gamma \ge \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$
$$\gamma \le \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

$$\boxed{\begin{aligned} w &\ge 3y + 2x - 6 \\ w &\ge 5y + 7x - 35 \\ w &\le 5y + 2x - 10 \\ w &\le 3y + 7x - 21 \end{aligned}}$$

$$\Rightarrow$$

$$\boxed{\begin{aligned} \max_{x,y,w} \;& x + 2y \\ \text{s.t. } & 2w + 3y \le 20 \\ & w \ge 3y + 2x - 6 \\ & w \ge 5y + 7x - 35 \\ & w \le 5y + 2x - 10 \\ & w \le 3y + 7x - 21 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}}$$

# Envelopes for bilinearities

Canonical form
$$\max_x c^T x \quad \text{s.t. } Ax \le b \quad \text{and } x \ge 0$$

Matlab's linprog
$$\min_x c^T x \text{ such that } \begin{cases} Ax \le b \\ A_{eq}x = b_{eq} \\ l_b \le x \le u_b \end{cases}$$

$$\begin{aligned} \max_{x,y,w}\ & x + 2y \\ \text{s.t. } & 2w + 3y \le 20 \\ & w \ge 3y + 2x - 6 \\ & w \ge 5y + 7x - 35 \\ & w \le 5y + 2x - 10 \\ & w \le 3y + 7x - 21 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

$$\begin{aligned} \max_{x,y,w}\ & x + 2y + 0w \\ \text{s.t. } & 0x + 3y + 2w \le 20 \\ & 2x + 3y - w \le 6 \\ & 7x + 5y - w \le 35 \\ & -2x - 5y + w \le -10 \\ & -7x - 3y + w \le -21 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & -\infty \le w \le +\infty \\ & x \in \mathbb{R}, y \in \mathbb{R}, w \in \mathbb{R} \end{aligned}$$

```
c=-[1 2 0]
A=[0 3 2;
    2 3 -1;
    7 5 -1;
   -2 -5 1;
   -7 -3 1]
b=[20 6 35 -10 -21]
lb=[3 2 -inf]
ub=[5 7 inf]
[sol,val]=linprog(c,A,b,[],[],lb,ub)
```

```
sol=[3.5 2 7]
val=-7.5
```

Here x*y=w
But not in general

# Envelopes for bilinearities

$$\max_{x,y} \; x+2y$$
$$\text{s.t. } 2xy+3y \le \boxed{20}$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$x \in \mathbb{R}, y \in \mathbb{R}$$

$$\max_{x,y} \; x+2y$$
$$\text{s.t. } 2xy+3y \le \boxed{50}$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$x \in \mathbb{R}, y \in \mathbb{R}$$

$$\max_{x,y,w} \; x+2y+0w$$
$$\text{s.t. } 0x+3y+2w \le 50$$
$$2x+3y-w \le 6$$
$$7x+5y-w \le 35$$
$$-2x-5y+w \le -10$$
$$-7x-3y+w \le -21$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$-\infty \le w \le +\infty$$
$$x \in \mathbb{R}, y \in \mathbb{R}, w \in \mathbb{R}$$

Matlab's linprog

$$\min_x c^T x \text{ such that } \begin{cases} Ax \le b \\ A_{eq}x = b_{eq} \\ l_b \le x \le u_b \end{cases}$$

```
c=-[1 2 0]
A=[0 3 2;
    2 3 -1;
    7 5 -1;
   -2 -5 1;
   -7 -3 1]
b=[50 6 35 -10 -21]
lb=[3 2 -inf]
ub=[5 7 inf]
[sol,val]=linprog(c,A,b,[],[],lb,ub)
```

sol=[3.70 5.24 17.13]
val=-14.19

x*y=19.41 ≠ w=17.13
In general, not equal

# Envelopes for square terms

$$||\mathbf{x}||_2 = 1 \quad \Rightarrow \quad \sum_{j=1}^{n} x_j^2 = 1$$

$$-1 \leq x_j \leq 1, j = 1, 2, \cdots, n$$

$$\Rightarrow \quad \sum_{j=1}^{n} w_j = 1, w_j = x_j^2,$$

$$-1 \leq x_j \leq 1, j = 1, 2, \cdots, n$$

Envelopes:

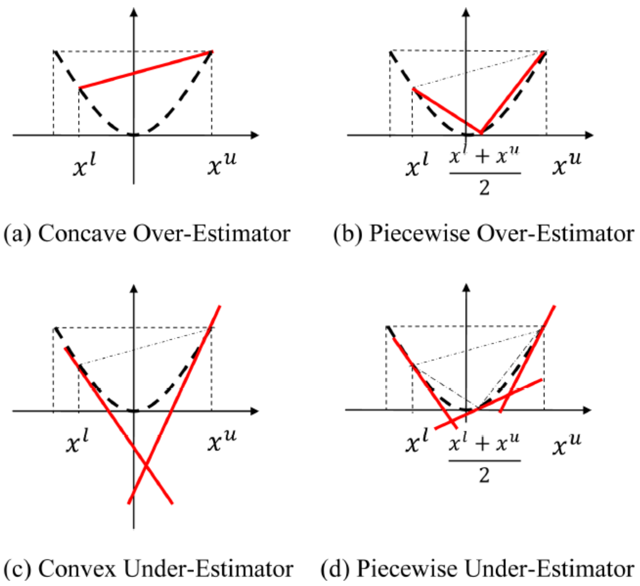
$$\gamma \geq \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$

$$\gamma \leq \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

$$w \geq 2x^l x - (x^l)^2, w \geq 2x^u x - (x^u)^2$$

$$w \leq (x^l + x^u)x - x^l x^u$$

Too loose ==> use piecewise estimator instead




(a) Concave Over-Estimator (b) Piecewise Over-Estimator

(c) Convex Under-Estimator (d) Piecewise Under-Estimator

Figure 1. Concave and convex relaxations v.s. piecewise linear relaxations for a square term. (a) and (c) are the concave and convex linear over- and under-estimator, respectively. (b) and (d) are the piecewise linear over- and under-estimator using $K$=2 segments, respectively.


"Deterministically Maximizing Feasible Subsystem for Robust Model Fitting with Unit Norm Constraint", Yinqiang Zheng, Shigeki Sugimoto and Masatoshi Okutomi, CVPR2011

# Envelopes for trilinearites

**Case 1:** $\underline{x} \geq 0,\ \underline{y} \geq 0,\ \underline{z} \geq 0$

Convex envelopes

$$
\begin{aligned}
w &= \underline{y}\underline{z}x + \underline{x}\underline{z}y + \underline{x}\underline{y}z - 2\underline{x}\underline{y}\underline{z} && (3.17)\\
w &= \overline{y}\overline{z}x + \overline{x}\overline{z}y + \overline{x}\overline{y}z - 2\overline{x}\overline{y}\overline{z} && (3.18)\\
w &= \underline{y}\overline{z}x + \underline{x}\overline{z}y + \overline{x}\underline{y}z - \underline{x}\underline{y}\overline{z} - \overline{x}\underline{y}\overline{z} && (3.19)\\
w &= \overline{y}\underline{z}x + \overline{x}\underline{z}y + \underline{x}\overline{y}z - \overline{x}\overline{y}\underline{z} - \underline{x}\overline{y}\underline{z} && (3.20)\\
w &= \frac{\theta}{\overline{x}-\underline{x}}x + \overline{x}\underline{z}y + \overline{x}\underline{y}z + \left(-\frac{\theta\underline{x}}{\overline{x}-\underline{x}} - \overline{x}\overline{y}\underline{z} - \overline{x}\underline{y}\overline{z} + \underline{x}\overline{y}\overline{z}\right), && (3.21)
\end{aligned}
$$

where $\theta = \overline{x}\overline{y}\underline{z} - \underline{x}\overline{y}\overline{z} - \overline{x}\underline{y}\underline{z} + \overline{x}\underline{y}\overline{z}$

$$
w = \frac{\theta}{\underline{x}-\overline{x}}x + \underline{x}\overline{z}y + \underline{x}\overline{y}z + \left(-\frac{\theta\overline{x}}{\underline{x}-\overline{x}} - \underline{x}\underline{y}\overline{z} - \underline{x}\overline{y}\underline{z} + \overline{x}\underline{y}\underline{z}\right), \quad (3.22)
$$

where $\theta = \underline{x}\underline{y}\overline{z} - \overline{x}\underline{y}\underline{z} - \underline{x}\overline{y}\overline{z} + \underline{x}\overline{y}\underline{z}$.

Concave envelopes

$$
\begin{aligned}
w &= \underline{y}\underline{z}x + \overline{x}\underline{z}y + \overline{x}\overline{y}z - \overline{x}\overline{y}\underline{z} - \overline{x}\underline{y}\underline{z}\\
w &= \overline{y}\underline{z}x + \underline{x}\underline{z}y + \overline{x}\overline{y}z - \overline{x}\overline{y}\underline{z} - \underline{x}\overline{y}\underline{z}\\
w &= \underline{y}\underline{z}x + \overline{x}\overline{z}y + \overline{x}\underline{y}z - \overline{x}\underline{y}\overline{z} - \overline{x}\underline{y}\underline{z}\\
w &= \overline{y}\overline{z}x + \underline{x}\underline{z}y + \underline{x}\overline{y}z - \underline{x}\overline{y}\overline{z} - \underline{x}\overline{y}\underline{z}\\
w &= \underline{y}\overline{z}x + \overline{x}\overline{z}y + \underline{x}\underline{y}z - \overline{x}\underline{y}\overline{z} - \underline{x}\underline{y}\overline{z}\\
w &= \overline{y}\overline{z}x + \underline{x}\overline{z}y + \underline{x}\underline{y}z - \underline{x}\overline{y}\overline{z} - \underline{x}\underline{y}\overline{z}.
\end{aligned}
$$

$$
\begin{aligned}
\min_{\mathbf{f}} \quad & E_{samp}\\
s.t., \quad & det(\mathbf{F}) = 0, ||\mathbf{f}|| = 1,\\
& -1 \leq f_n \leq 1, n = 1, 2, \cdots, 8, 0 \leq f_9 \leq 1
\end{aligned}
$$

$$f_1 f_5 f_9 - f_1 f_6 f_8 - f_2 f_4 f_9 + f_2 f_6 f_7 + f_3 f_4 f_8 - f_3 f_5 f_7 = 0$$

$$w_{159} - w_{168} - w_{249} + w_{267} + w_{348} - w_{357} = 0$$

$$w_{rst} = f_r f_s f_t$$


C. Meyer and C. Floudas. Trilinear monomials with positive or negative domains: facets of the convex and concave envelopes. Frontiers in Global Optimization, 2003

C. Meyer and C. Floudas. Trilinear monomials with mixed sign domains: facets of the convex and concave envelopes. Journal of Global Optimization, 2004

A Branch and Contract Algorithm For Globally Optimal Fundamental Matrix Estimation, Yinqiang Zheng, Shigeki Sugimoto and Masatoshi Okutomi, CVPR 2011

# Mathematical formulation

- Bilinear relaxation: by concave and convex envelopes

bilinear equality $\gamma = \alpha\beta$ with the bounding-boxes $[\underline{\alpha}, \bar{\alpha}]$ and $[\underline{\beta}, \bar{\beta}]$ is relaxed by the following envelopes:

$$\gamma \geq \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$

$$\gamma \leq \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

We collectively represent them as $conv(\alpha, \beta) \leq \gamma \leq conc(\alpha, \beta)$

$$\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & \|\Theta\| = 1 \\
\text{and} \quad & 0 \leq z_i \leq 1, \forall i = 1..N
\end{aligned}$$

$$\begin{aligned}
\max_{\mathbf{z},\Theta,w,\mu} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & |A_i^T w_i| \leq z_i T, \forall i \quad \longrightarrow \quad \boxed{z_i|\mathbf{A}_i^T\Theta|} \ \text{and} \ \boxed{w_i = z_i\Theta} \\
\text{and} \quad & \sum_{j=1}^{D} \mu_j = 1 \ \text{and} \ \underline{\Theta} \leq \Theta \leq \overline{\Theta} \quad \longrightarrow \quad \boxed{\mu_j = \Theta_j^2} \\
\text{and} \quad & 0 \leq z_i \leq 1, \forall i \\
\text{and} \quad & conv(z_i, \Theta_j) \leq w_{ij} \leq conc(z_i, \Theta_j) \\
& i = 1 \ldots N, j = 1 \ldots D \\
\text{and} \quad & conv(\Theta_j) \leq \mu_j \leq conc(\Theta_j), \forall j = 1 \ldots D
\end{aligned}$$

Bilinearities have been removed
→ All **linear!!**

But how to get bounding boxes of Θ?

# Bounds

- **Upper bound** ("too good")
  - Obtained by LP via the envelope relaxation

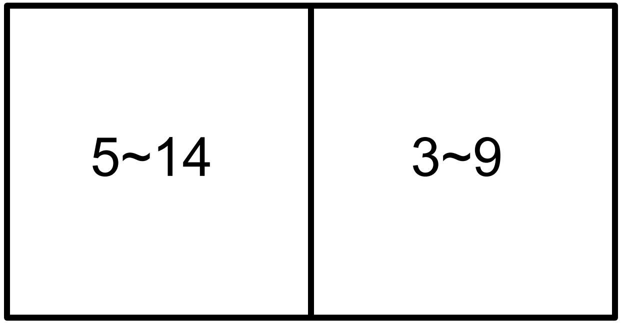
$$
\begin{aligned}
\max_{\mathbf{z},\Theta,w,\mu} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & |A_i^T w_i| \leq z_i T, \forall i \\
\text{and} \quad & \sum_{j=1}^{D} \mu_j = 1 \text{ and } \underline{\Theta} \leq \Theta \leq \overline{\Theta} \\
\text{and} \quad & 0 \leq z_i \leq 1, \forall i \\
\text{and} \quad & conv(z_i, \Theta_j) \leq w_{ij} \leq conc(z_i, \Theta_j) \\
& i = 1 \dots N, j = 1 \dots D \\
\text{and} \quad & conv(\Theta_j) \leq \mu_j \leq conc(\Theta_j), \forall j = 1 \dots D
\end{aligned}
$$

- **Lower bound**
  - Given $\Theta$ obtained by LP, simple test on $z$

If $f(\Theta, x_i) \leq T$ then $z_i = 1$

else $z_i = 0$

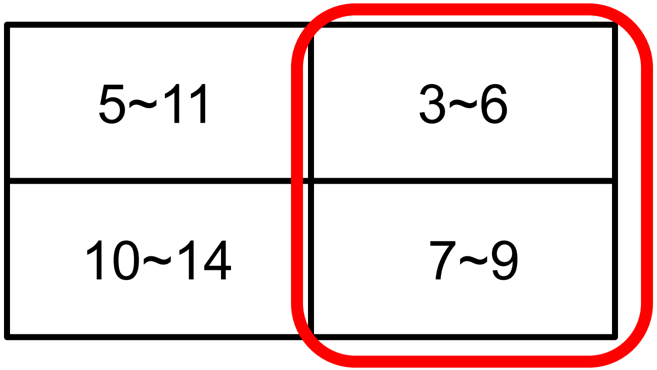
# BnB – max cardinality



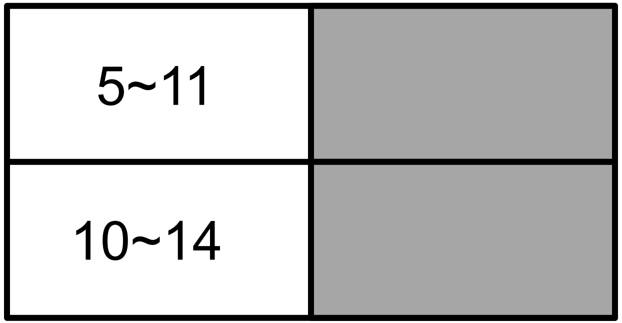

Goal: find the max cardinality

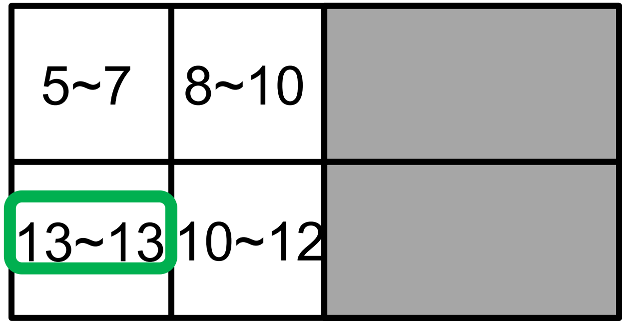
Note: the bounds might not be strict!!

Ex: 3~9: the true extremes can be 4 and 5



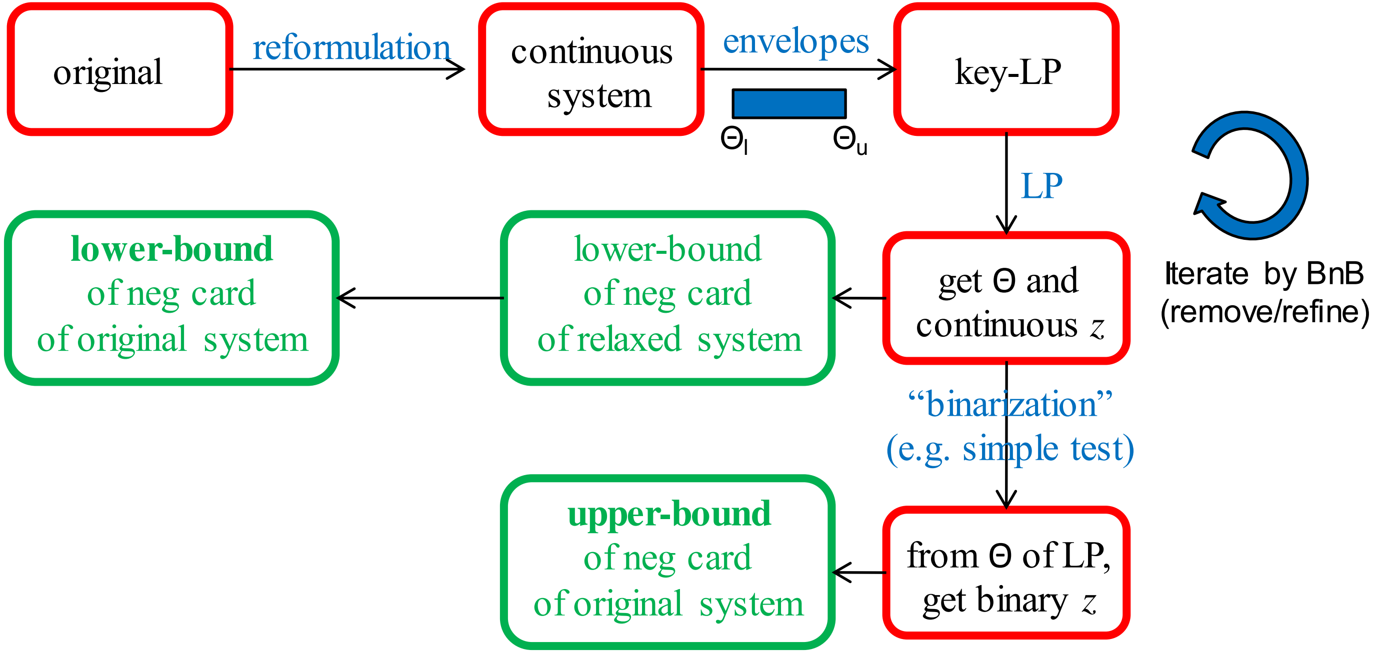

We are sure the global max is not here, so we **discard** these boxes



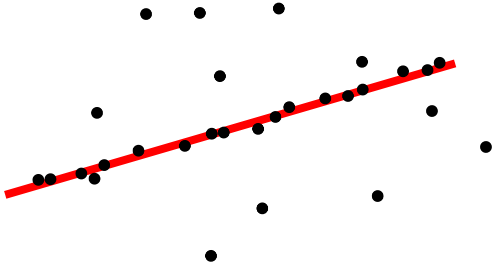
# Algorithm

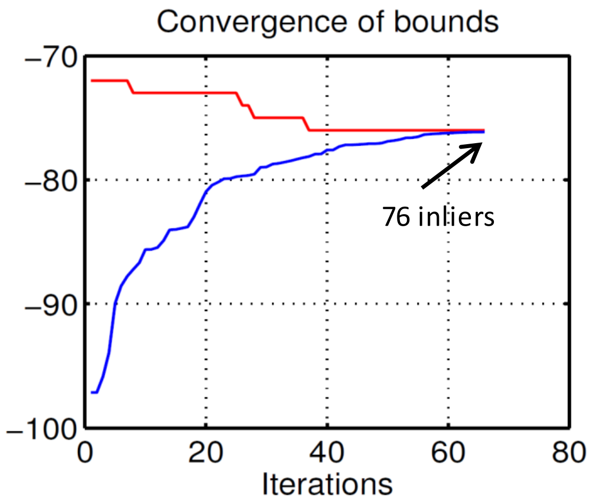
- Branch-and-bound
  - Used to get bounding boxes of the model Θ
  - Permits to reduce the relaxation gap: lower and upper bounds

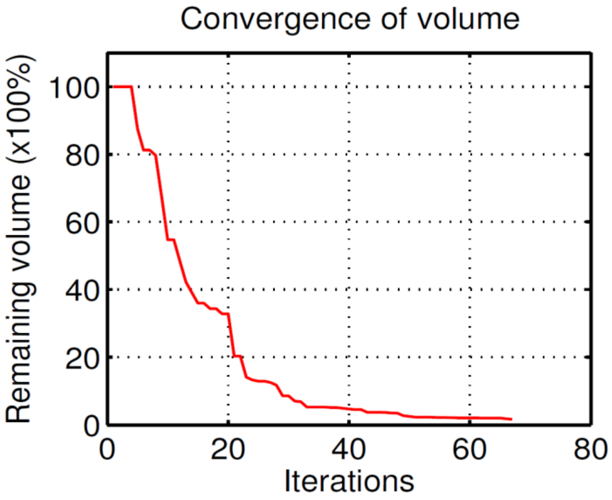



"Consensus Set Maximization with Guaranteed Global Optimality for Robust Geometry Estimation", Hongdong Li, ICCV'09

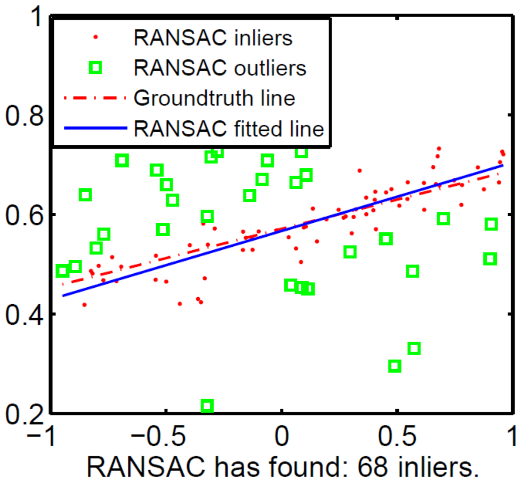
# Line fitting

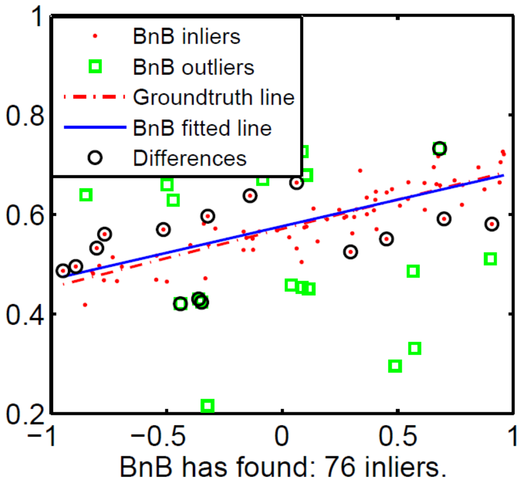


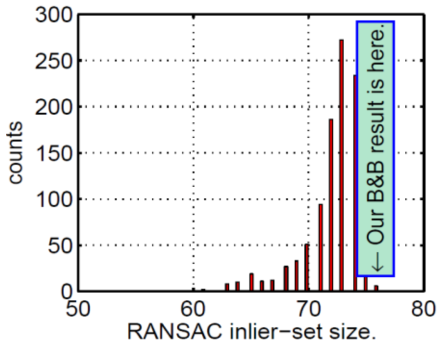


# Line Fitting

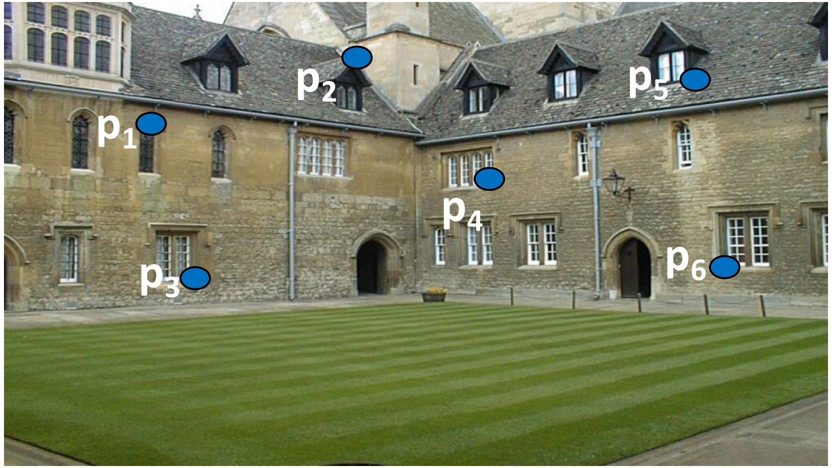


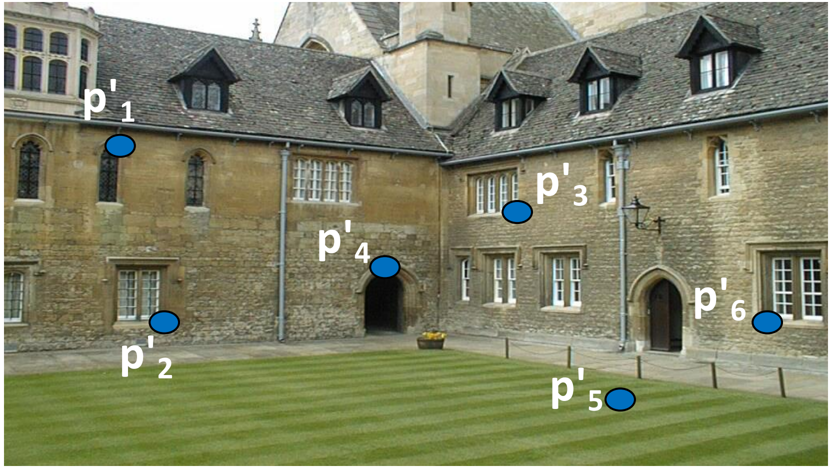




ETH zürich

# Extension to correspondence



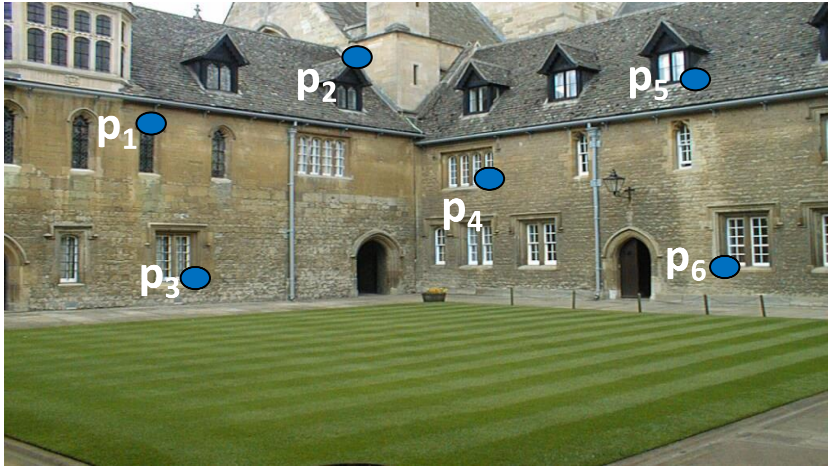

Let $z_{ij} = 1$ if the $i^{th}$ left point matches to the $j^{th}$ right point
$= 0$ else

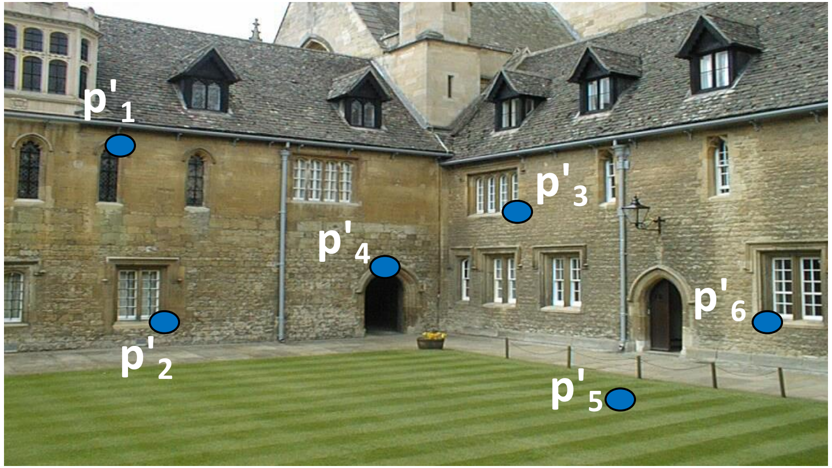
For the above example: $z_{11}=1$, $z_{43}=1$, $z_{32}=1$, etc…
$z_{35}=0$, $z_{61}=0$, etc…

"A Branch and Bound Approach to Correspondence and Grouping Problems" by J.C. Bazin, H. Li, I. Kweon, C. Demonceaux, P. Vasseur and K. Ikeuchi, PAMI, 2013

# Notations and constraints

- What is a “good correspondence”?
  - They must look the same, i.e. have similar descriptors:
    - $h(f_i, f'_j) \leq T_a$
  - They must verify the (motion) model:
    - $|A_{ij}^T \theta| \leq T_g$



# Mathematical Formulation

- Main goal: maximize the nb of inlier matches while verifying appearance and motion constraints

$$\max_{Z,\theta} \sum_{i=1}^{N_l} \sum_{j=1}^{N_r} z_{ij}$$

$$\text{s.t. } z_{ij} h(f_i, f'_j) \leq z_{ij} T_a, \forall i, j$$ appearance

$$\text{and } z_{ij} |A_{ij}^T \theta| \leq z_{ij} T_g, \forall i, j$$ motion

$$\text{and } \|\theta\| = 1$$ scale

$$\text{and } z_{ij} \in \{0, 1\}, \forall i, j$$ binary

$$\text{and } 0 \leq \sum_{j}^{N_r} z_{ij} \leq 1, \forall i \text{ and } 0 \leq \sum_{i}^{N_l} z_{ij} \leq 1, \forall j$$ At most 1

  - $A_{ij}$ is the row of the data matrix corresponding to the $i^{th}$ point of the left image and the $j^{th}$ point of the right image
  - $\theta$ is the motion model
  - $T_a$ and $T_g$ are respectively the appearance and geometric thresholds

Hard to solve: binary/continuous unknowns and bilinearities

# Solving

- Bilinear relaxation using envelopes (see previous slide)

$$
\begin{aligned}
\max_{Z,\theta,w,\mu} & \sum_{i=1} \sum_{j=1} z_{ij} \\
\text{s.t. } & z_{ij} h(f_i, f'_j) \le z_{ij} T_a, \forall i,j \\
\text{and } & |A_{ij}^T w_{ij}| \le z_{ij} T_g, \forall i,j \\
\text{and } & \sum_{k=1}^{D} \mu_k = 1 \text{ and } \underline{\theta} \le \theta \le \overline{\theta} \\
\text{and } & 0 \le z_{ij} \le 1, \forall i,j \\
\text{and } & 0 \le \sum_{j}^{N_r} z_{ij} \le 1, \forall i \text{ and } 0 \le \sum_{i}^{N_l} z_{ij} \le 1, \forall j \\
\text{and } & conv(z_{ij}, \theta_k) \le w_{ijk} \le conc(z_{ij}, \theta_k), \\
& i = 1 \dots N_l, j = 1 \dots N_r, k = 1 \dots D \\
\text{and } & conv(\theta_k) \le \mu_k \le conc(\theta_k), \forall k = 1 \dots D
\end{aligned}
$$

$\boxed{z_{ij} |A_{ij}^T \theta|}$ → $w_{ij}$

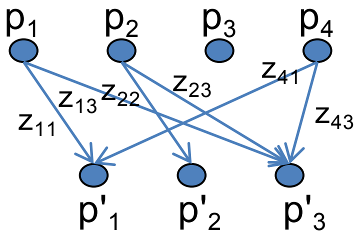
Bilinearities have been removed → All **linear** → solvable by **LP**

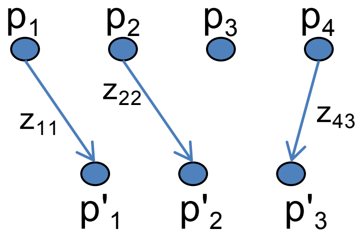
→ Provide **upper bound** (“too good”)

# Solving

- Lower bound
  - Relaxation of *z* is not equivalent (because of the sum)
  - Given *Θ*, how to find optimal binary *z*?
    - by **maximum bipartite matching**

$$
\begin{aligned}
&\max_{Z,\theta} \sum_{i=1}^{N_l} \sum_{j=1}^{N_r} z_{ij} \\
&\text{s.t. } z_{ij} h(f_i, f'_j) \leq z_{ij} T_a, \forall i,j && \text{appearance} \\
&\text{and } z_{ij} |A_{ij}^T \theta| \leq z_{ij} T_g, \forall i,j && \text{motion} \\
&\text{and } \|\theta\| = 1 && \text{scale} \\
&\text{and } z_{ij} \in \{0,1\}, \forall i,j && \text{binary} \\
&\text{and } 0 \leq \sum_{j}^{N_r} z_{ij} \leq 1, \forall i \text{ and } 0 \leq \sum_{i}^{N_l} z_{ij} \leq 1, \forall j && \text{At most 1}
\end{aligned}
$$



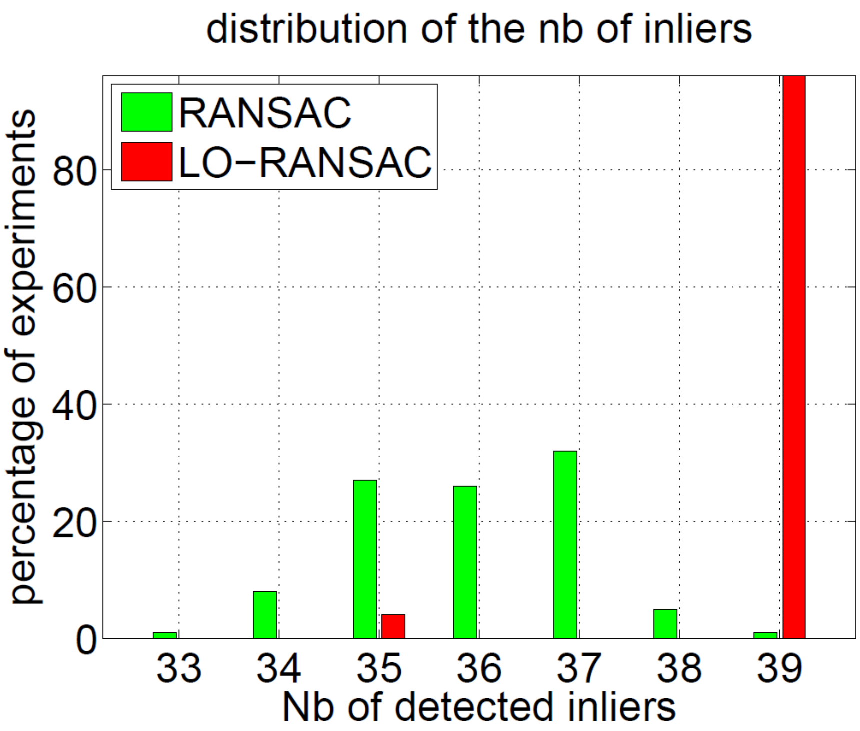



# Experimental results

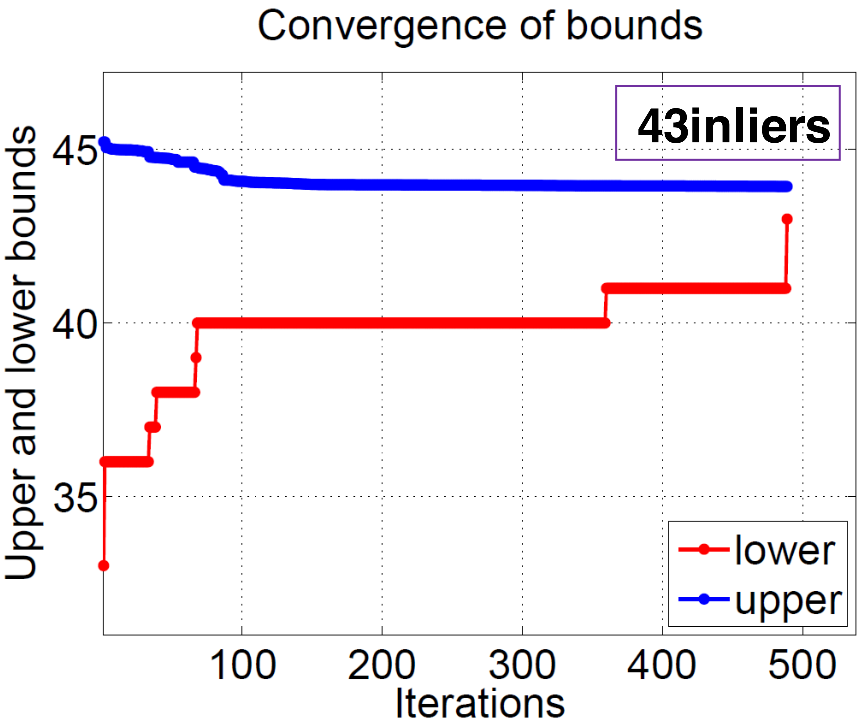


Urban scene with small overlap and many repetitive textures (issued from ICCV’05 contest) - Homography model

# Experimental results




**RANSAC**

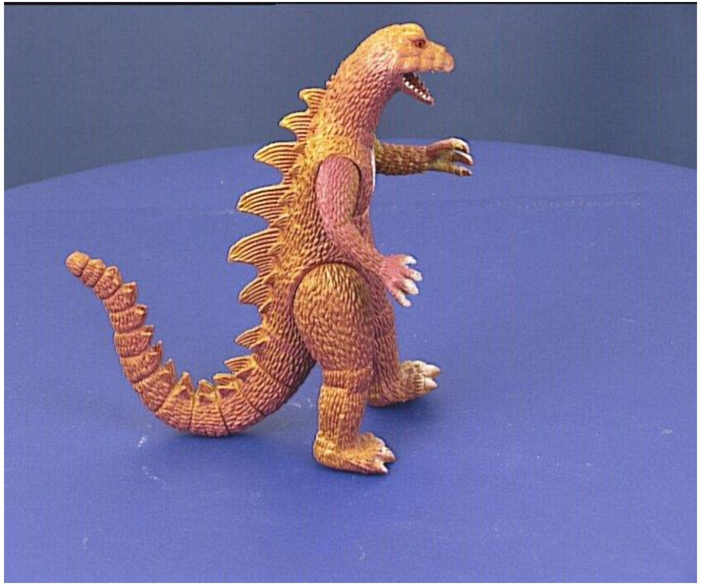



**BNB-LP**


Discovery
CHANNEL
STORE
GODIVA

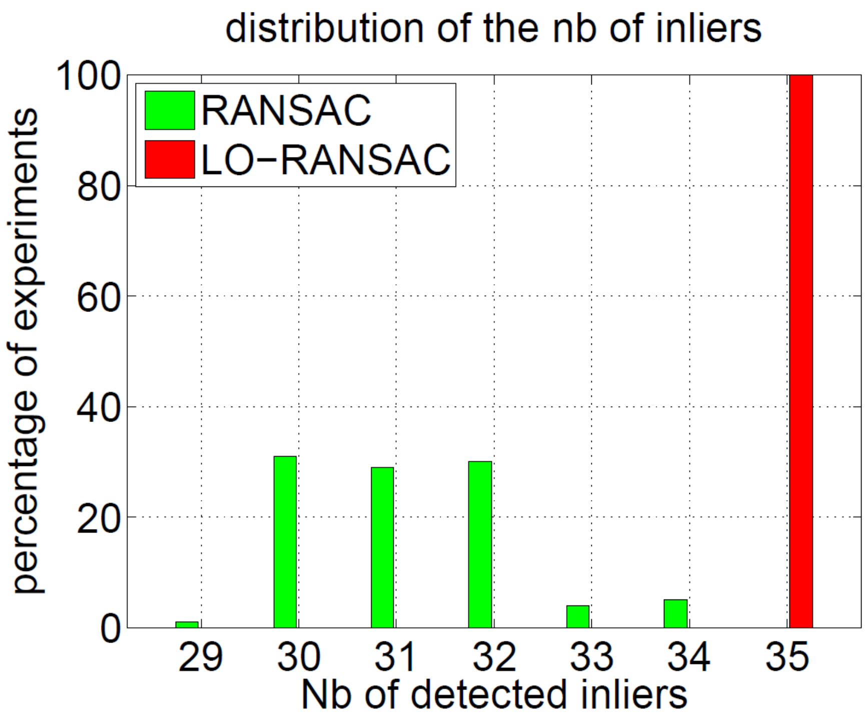
Discovery
CHANNEL
STORE
GODIVA

# Experimental results

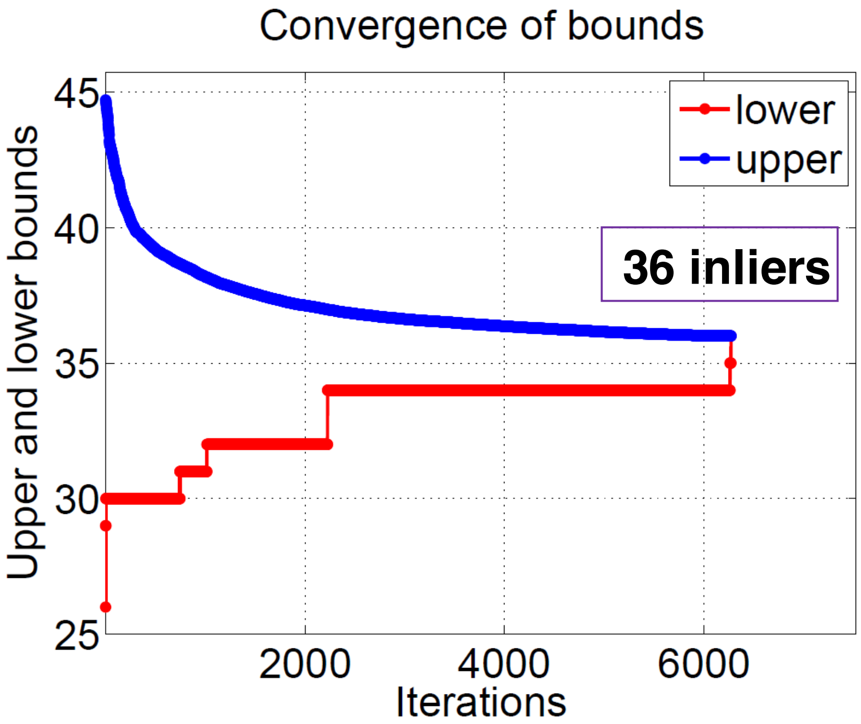


Dinosaur sequence (Oxford) – Affine fundamental matrix

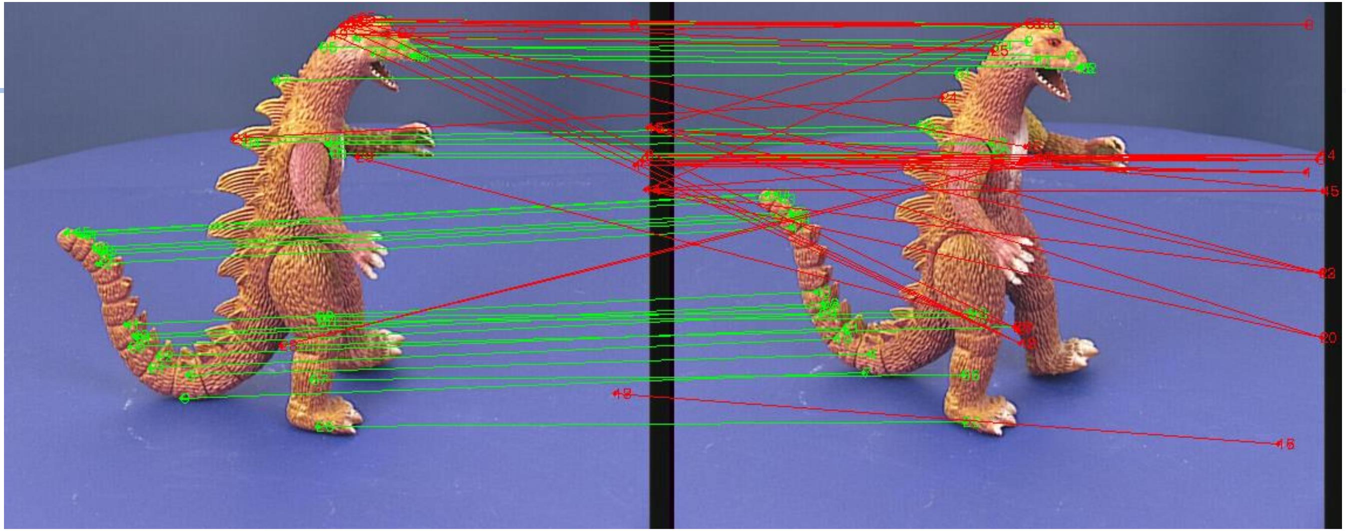
# Experimental results

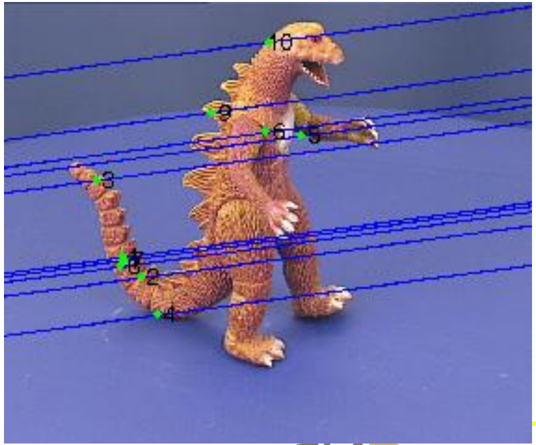



**RANSAC**




**BNB-LP**





Multi potential matching and inliers/outliers

Epipolar lines (only 10 are displayed)

# Summary

- (univariate) polynomial system $ax^4 + bx^3 + cx^2 + dx + e = 0$
  - Companion matrix $p(t) = t^n + c_{n-1}t^{n-1} + \ldots + c_1 t + c_0$
- "Degenerate" linear system $Ax = 0$
  - By SVD
- System of polynomial (Grobner basis)

$$8x^2 + 3y^3 + 2xy + 7 = 0$$
$$4x^2 + 3xy + 5y = 0$$

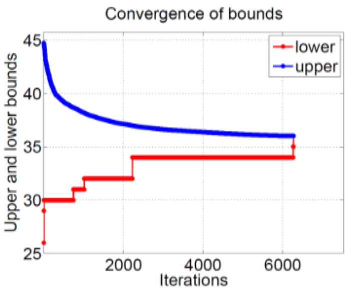
- Concave and convex envelopes

bilinear equality $\gamma = \alpha\beta$ with the bounding-boxes $[\underline{\alpha}, \bar{\alpha}]$ and $[\underline{\beta}, \bar{\beta}]$ is relaxed by the following envelopes:

$$\gamma \geq \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$
$$\gamma \leq \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

We collectively represent them as $conv(\alpha, \beta) \leq \gamma \leq conc(\alpha, \beta)$

- Branch-and-bound



# Questions?