# An Exact Penalty Method for Locally Convergent Maximum Consensus

Huu Le, Tat-Jun Chin and David Suter

School of Computer Science, The University of Adelaide

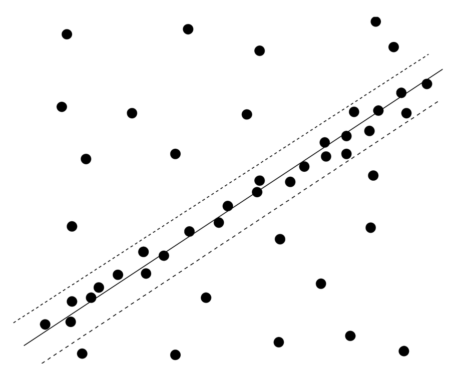
IEEE 2017 Conference on Computer Vision and Pattern Recognition

## Maximum Consensus:

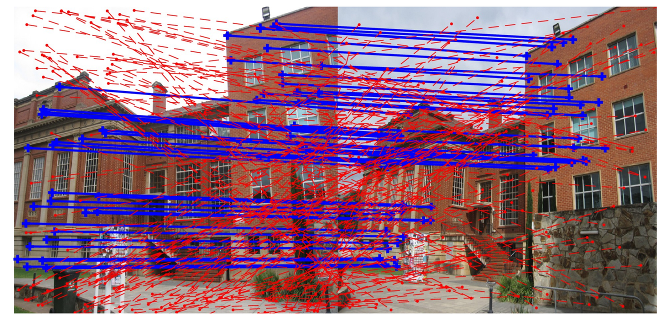
- Estimate a model that is consistent with as many of the data as possible:

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d,\ \mathcal{I}\in\mathcal{P}(N)} \quad |\mathcal{I}| \qquad \text{subject to} \quad |\mathbf{x}_j^T\boldsymbol{\theta} - y_j| \le \epsilon \quad \forall j \in \mathcal{I}, \tag{1}$$

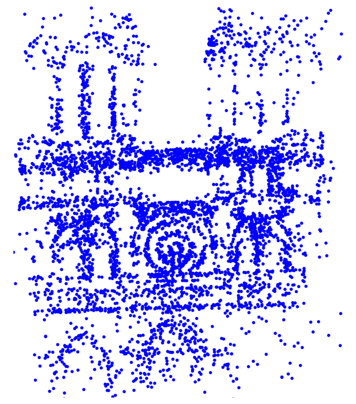
Applications:



Line/Plane fitting



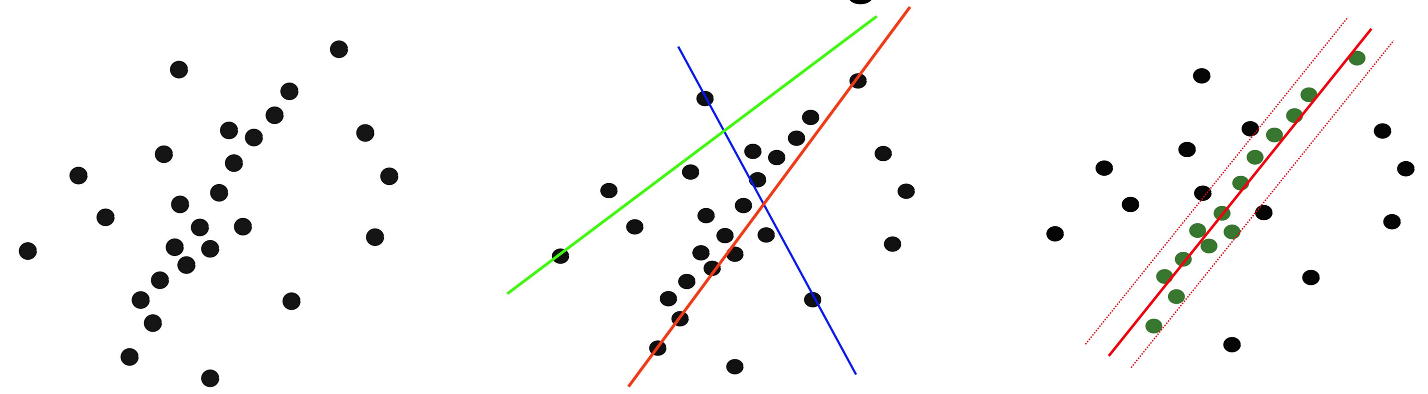
Homography Estimation



Triangulation

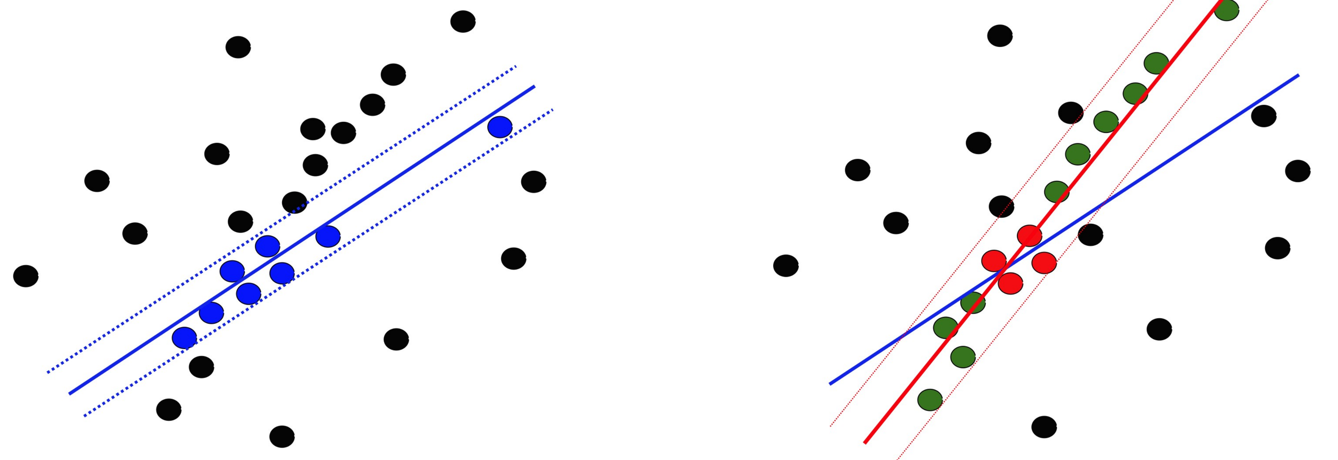
## RANSAC

- Fit a model onto randomly sampled minimal subsets.
- Return the model with the largest consensus size.



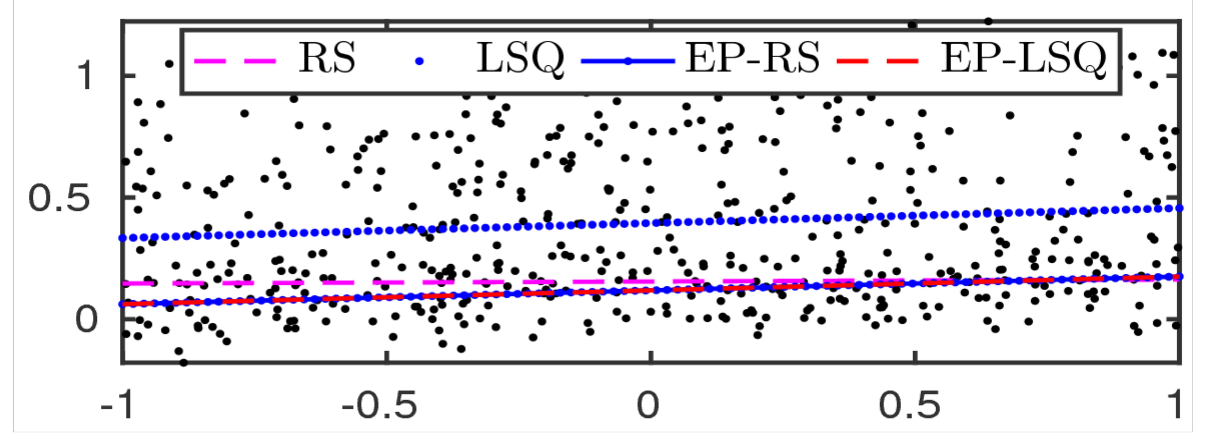
## Locally optimized RANSAC (LO-RANSAC)

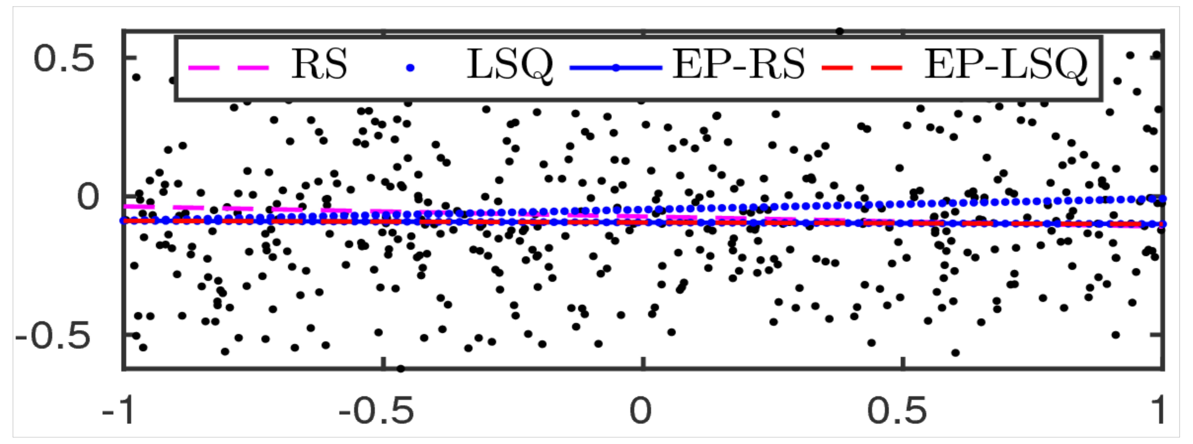
- Execute an inner sampling routine whenever the solution is updated by RANSAC.
- **Non-deterministic.**



## Deterministic local refinement

- Improve an initial estimate (RANSAC, LSQ,...) to a solution with larger consensus size.




## Complementarity constraints

- Rewrite (1) as:

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d,\ \mathcal{I}\in\mathcal{P}(M)} \quad |\mathcal{I}| \qquad \text{subject to} \quad \mathbf{a}_i^T\boldsymbol{\theta} - b_i \le 0 \quad \forall i \in \mathcal{I}$$

$$\mathbf{A} = [\mathbf{x}_1, -\mathbf{x}_1, \dots, \mathbf{x}_N, -\mathbf{x}_N],$$

$$\mathbf{b} = [\epsilon + y_1, \epsilon - y_1, \dots, \epsilon + y_N, \epsilon - y_N]^T$$

- Outlier minimization

$$\min_{\mathbf{u}\in\{0,1\}^M,\ \mathbf{s}\in\mathbb{R}^M,\ \boldsymbol{\theta}\in\mathbb{R}^d} \quad \sum_i u_i$$

$$\text{subject to} \quad s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \ge 0,\quad u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0,\quad s_i(1-u_i) = 0,\quad s_i \ge 0.$$

- Let $\mathbf{v} = \begin{bmatrix}\boldsymbol{\theta} + \gamma\mathbf{1} \\ \gamma\end{bmatrix}, \quad \mathbf{c}_i = \begin{bmatrix}\mathbf{a}_i^T & -\mathbf{a}_i^T\mathbf{1}\end{bmatrix}^T$

$$\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\ \mathbf{v}\in\mathbb{R}^{d+1}} \quad \sum_i u_i$$

$$\text{subject to} \quad s_i - \mathbf{c}_i^T\mathbf{v} + b_i \ge 0,\quad u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) = 0,\quad s_i(1-u_i) = 0,\quad 1 - u_i \ge 0,\quad s_i, u_i, v_i \ge 0. \tag{2}$$

## Penalty Method

- Incorporate the complementarity constraints to the cost function:

$$\min_{\mathbf{u},\mathbf{s},\mathbf{v}} \quad \sum_i u_i + \alpha\left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right]$$

$$\text{s.t.} \quad s_i - \mathbf{c}_i^T\mathbf{v} + b_i \ge 0,\quad 1 - u_i \ge 0,\quad s_i, u_i, v_i \ge 0. \tag{3}$$

## Frank-Wolfe method for solving (3)

- Fix **s, v** → (2) becomes LP with respect to **u**
- Fix **u** → (2) becomes LP with respect to **s,v**

**Algorithm 1** Frank-Wolfe method for ( 3 ).

**Require:** Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, penalty value $\alpha$, initial solution $\mathbf{u}^{(0)}, \mathbf{v}^{(0)}, \mathbf{s}^{(0)}$, threshold $\delta$.

```
P^(0) ← P(u^(0), s^(0), v^(0) | α).
t ← 0.
while true do
    t ← t + 1.
    s^(t), v^(t) ← arg min_{s,v} P(u^(t-1), s, v | α) s.t. 𝒫.
    u^(t) ← arg min_u P(u, s^(t), v^(t) | α) s.t. 𝒫.
    P^(t) ← P(u^(t), s^(t), v^(t) | α).
    if |P^(t-1) − P^(t)| ≤ δ then
        Break.
    end if
end while
return u^(t), v^(t), s^(t).
```

* **Algorithm 1 converges to a KKT point of (3)**

## Main algorithm

- **Theorem:** There exists α* such that for all α ≥ α*, a KKT point of (3) is also a KKT point of (2)

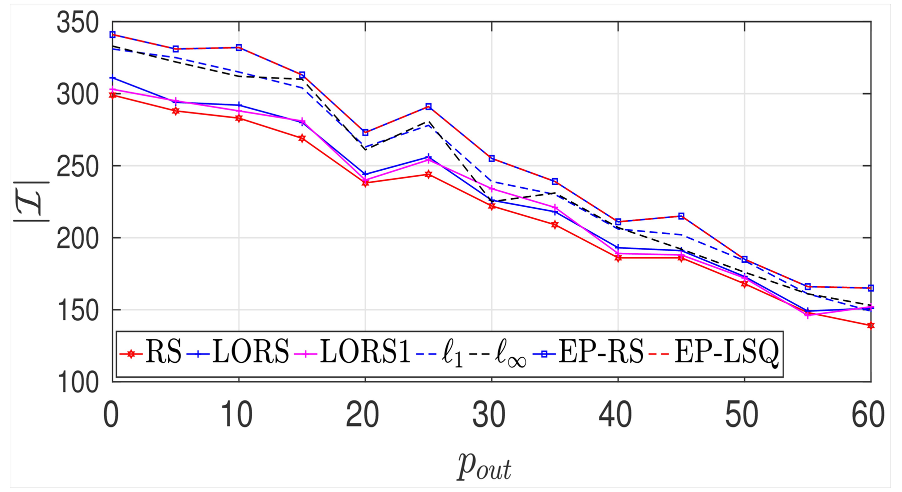
**Algorithm 2** Main algorithm for solving (2).

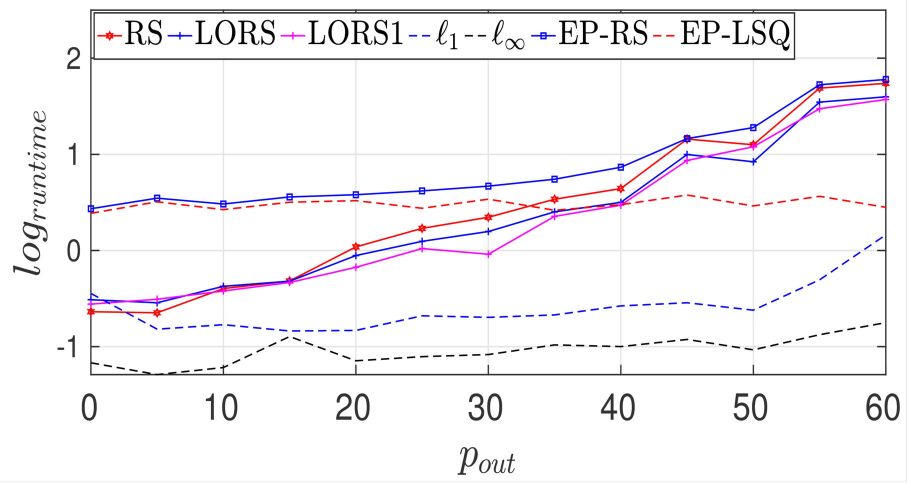
**Require:** Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, initial model parameter $\boldsymbol{\theta}$, initial penalty value $\alpha$, increment rate $\kappa$, threshold $\delta$.

```
v ← [θ^T  1^T θ]^T.
s ← Cv − b.
u ← 𝕀(s > 0).
while true do
    u, s, v ← FW({c_i, b_i}_{i=1}^M, α, u, s, v). /*Algo. 1.*/
    if Q(z) ≤ δ then
        Break.
    end if
    α ← κ · α.
end while
return u, s, v.
```

- Can be applied to quasiconvex problems

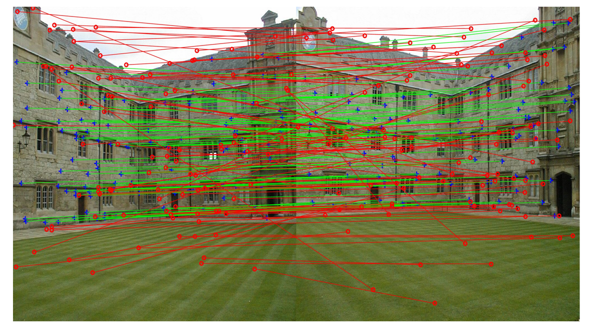
## Synthetic data results

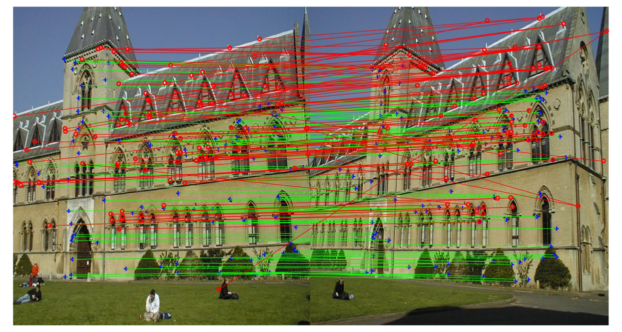


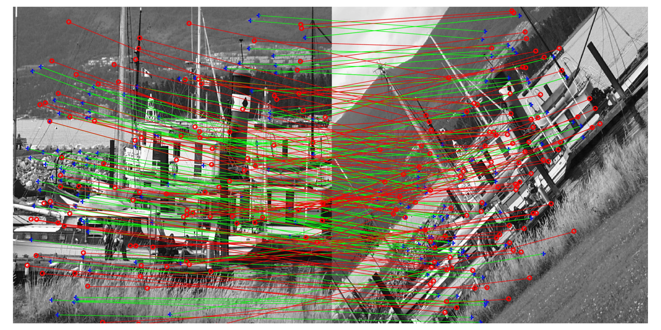




## Real data results

| | Methods | | RS | PS | GMLE | LORS | LORS1 | $\ell_1$ | $\ell_\infty$ | EP-RS | EP-$\ell_\infty$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Homography estimation | University Library | $\lvert\mathcal{I}\rvert$ | 251 | 269 | 251 | 294 | 294 | 120 | 53 | **301** | **301** |
| | N = 545 | time (s) | 0.73 | 0.62 | 0.69 | 1.90 | 1.89 | 3.10 | 2.49 | 12.76 | 14.49 |
| | Christ Church | $\lvert\mathcal{I}\rvert$ | 235 | 236 | 227 | 250 | 246 | 246 | 160 | **280** | **280** |
| | N = 445 | time (s) | 0.47 | 0.47 | 0.43 | 1.33 | 1.61 | 1.23 | 2.44 | 10.37 | 12.67 |
| | Valbonne | $\lvert\mathcal{I}\rvert$ | 131 | 134 | 117 | 156 | 136 | 24 | 22 | **158** | **158** |
| | N = 434 | time (s) | 3.17 | 2.39 | 5.76 | 3.04 | 5.80 | 1.36 | 1.27 | 17.20 | 14.84 |
| | Kapel | $\lvert\mathcal{I}\rvert$ | 163 | 167 | 130 | 167 | 168 | 28 | 161 | **170** | **170** |
| | N = 449 | time (s) | 1.19 | 1.15 | 9.89 | 2.18 | 2.70 | 1.62 | 1.16 | 8.46 | 8.68 |
| | Invalides | $\lvert\mathcal{I}\rvert$ | 144 | 159 | 140 | 149 | 156 | 84 | 142 | **178** | **178** |
| | N = 413 | time (s) | 1.36 | 0.90 | 1.60 | 2.17 | 2.94 | 1.04 | 0.71 | 10.20 | 9.15 |
| Affinity estimation | Bikes | $\lvert\mathcal{I}\rvert$ | 424 | 427 | 425 | 426 | 424 | 387 | 431 | **437** | **437** |
| | N = 557 | time (s) | 6.09 | 6.09 | 5.79 | 6.28 | 11.8 | 1.77 | 1.77 | 15.26 | 9.81 |
| | Graff | $\lvert\mathcal{I}\rvert$ | 126 | 129 | 127 | 134 | 126 | 147 | 274 | **276** | **276** |
| | N = 327 | time (s) | 3.51 | 3.35 | 3.14 | 4.07 | 6.61 | 0.99 | 0.23 | 5.94 | 2.70 |
| | Bark | $\lvert\mathcal{I}\rvert$ | 279 | 288 | 270 | 284 | 279 | 298 | 439 | **442** | **442** |
| | N = 458 | time (s) | 4.89 | 4.93 | 4.68 | 5.11 | 9.54 | 1.31 | 0.19 | 10.19 | 5.51 |
| | Tree | $\lvert\mathcal{I}\rvert$ | 372 | 367 | 371 | 372 | 372 | 377 | 370 | **396** | **396** |
| | N = 568 | time (s) | 5.70 | 6.01 | 5.73 | 6.93 | 11.50 | 4.81 | 0.81 | 15.96 | 11.82 |
| | Boat | $\lvert\mathcal{I}\rvert$ | 476 | 477 | 476 | 477 | 476 | 469 | 464 | **483** | **483** |
| | N = 574 | time (s) | 6.32 | 6.29 | 6.02 | 7.18 | 12.32 | 4.12 | 1.02 | 14.86 | 9.33 |







MATLAB code is available at: http://cs.adelaide.edu.au/~huu