

PROSAC

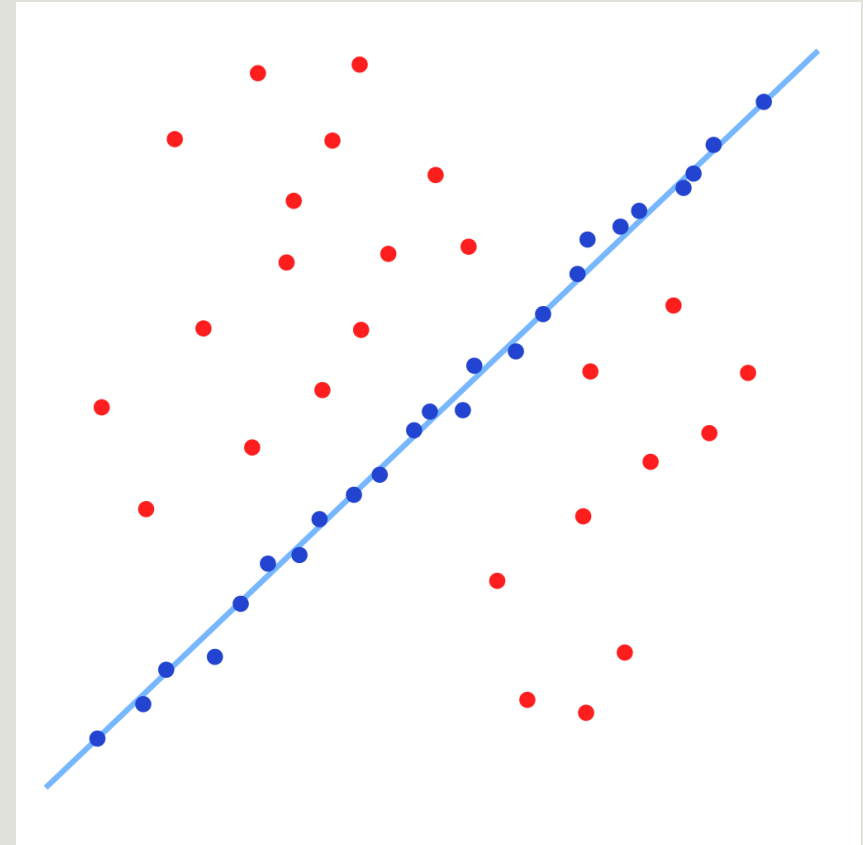
Progressive Sample Consensus

CHUM & MATAS

Problème

Estimation de modèles

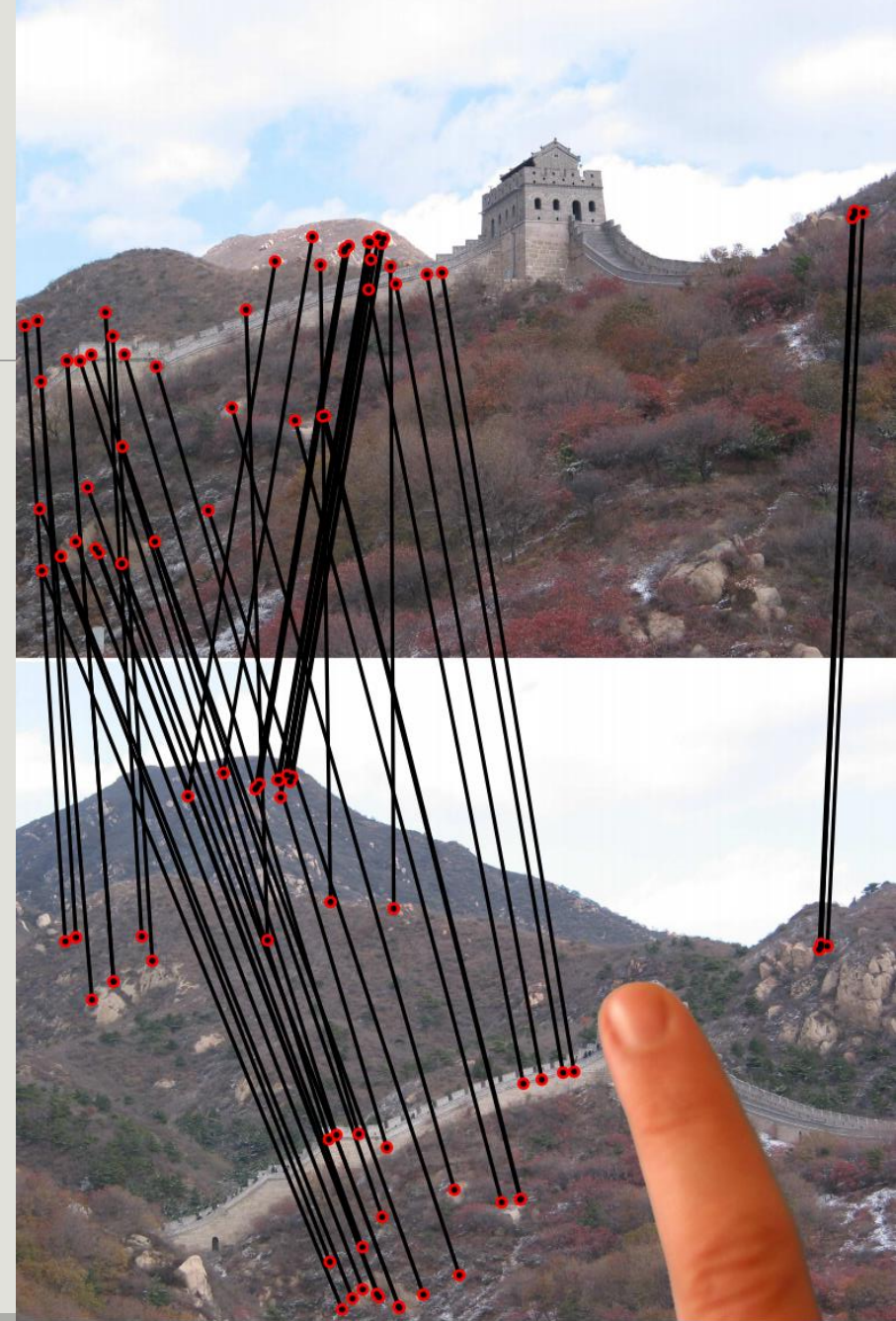
- Estimer modèles
- Beaucoup de bruit
- Robustesse



Problème

Correspondance entre images

- Occlusion
- Points de vues différents
- Motif répétitif
- Appariements incorrects durs à éviter
- Mauvais modèle



RANSAC

Rappel

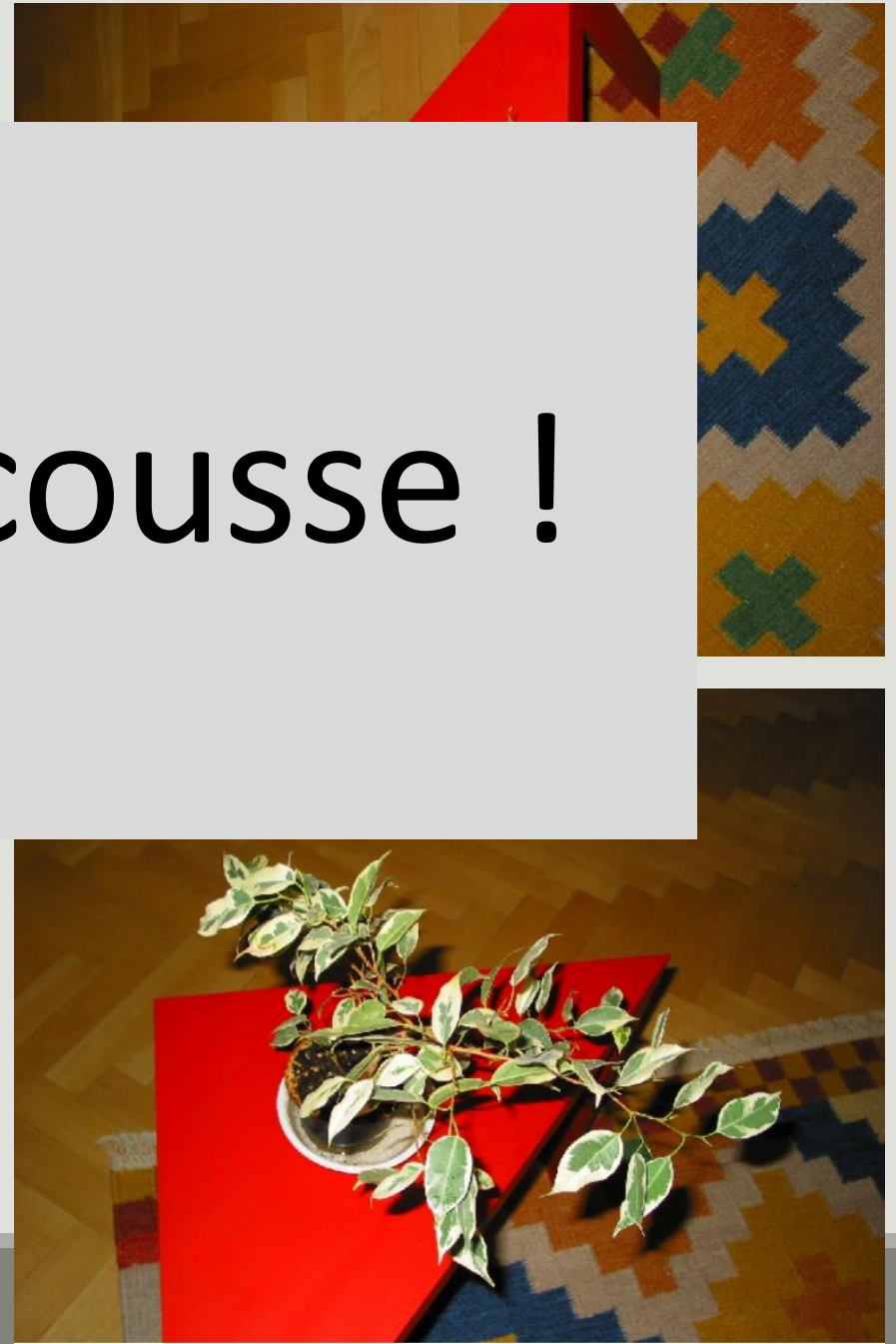
1. Piger m points
2. Calculer modèle
3. I_N : nombre inliers
4. Si $\frac{I_N}{N} > \tau \rightarrow$ termine
5. Répéter 1 à 4, max $\approx \frac{1}{w^m}$ fois

Limitations de RANSAC

- Nombre d'essaies en $O\left(\frac{1}{w^m}\right)$
- $w \approx 9.2\%$
- $m = 7$ (géométrie épipolaire)
- $8.43 \cdot 10^7$ piges !!!



PROSAC à la rescousse !



PROSAC

Progressive Sample Consensus

- Version plus rapide de RANSAC
- Même robustesse
- Algorithme relativement simple

PROSAC

Intuition

- RANSAC avec notion de qualité
- Meilleure qualité \Rightarrow plus de chance d'être correct
- On commence par piger les meilleurs
- Ajout progressif de points de moins bonne qualité
- Modèle rapidement

PROSAC

Intuition

- Plus grande qualité
⇒ Plus grande probabilité d'être correct

RANSAC	PROSAC
p_1	p_2
p_2	p_4
p_3	p_6
p_4	p_3
p_5	p_5
p_6	p_1

PROSAC

Progressive Sample Consensus

1. Trier les points par qualité
2. Pige m points dans le top qualité
3. Calculer modèle
4. Vérifier le modèle sur tous les points
5. Répéter 1 à 4 en incorporant **progressivement** des points de moins bonne qualité, jusqu'aux critères de d'arrêt

Qualité

- Corrélation de l'intensité autour des points
- Distance de Lowe
 - s_1 distance plus similaire
 - s_2 distance second plus similaire
 - $p = \frac{s_1}{s_2}$

PROSAC

Progressive Sample Consensus

1. Trier les points par qualité
2. Pige m points dans le top qualité
3. Calculer modèle
4. Vérifier le modèle sur tous les points
5. Répéter 1 à 4 en incorporant **progressivement** des points de moins bonne qualité, jusqu'aux critères d'arrêt

Pige efficace

- T_n nombre de piges possibles
- T_{n+1} après avoir ajouté un point
- $a = T_{n+1} - T_n$ nouvelles piges
- Seulement besoin de faire a piges
 - p_{n+1} et $m - 1$ points dans top n

Points	Qualité
p_1	1
p_2	0.9
p_3	0.8
...	...
p_n	0.6
p_{n+1}	0.55

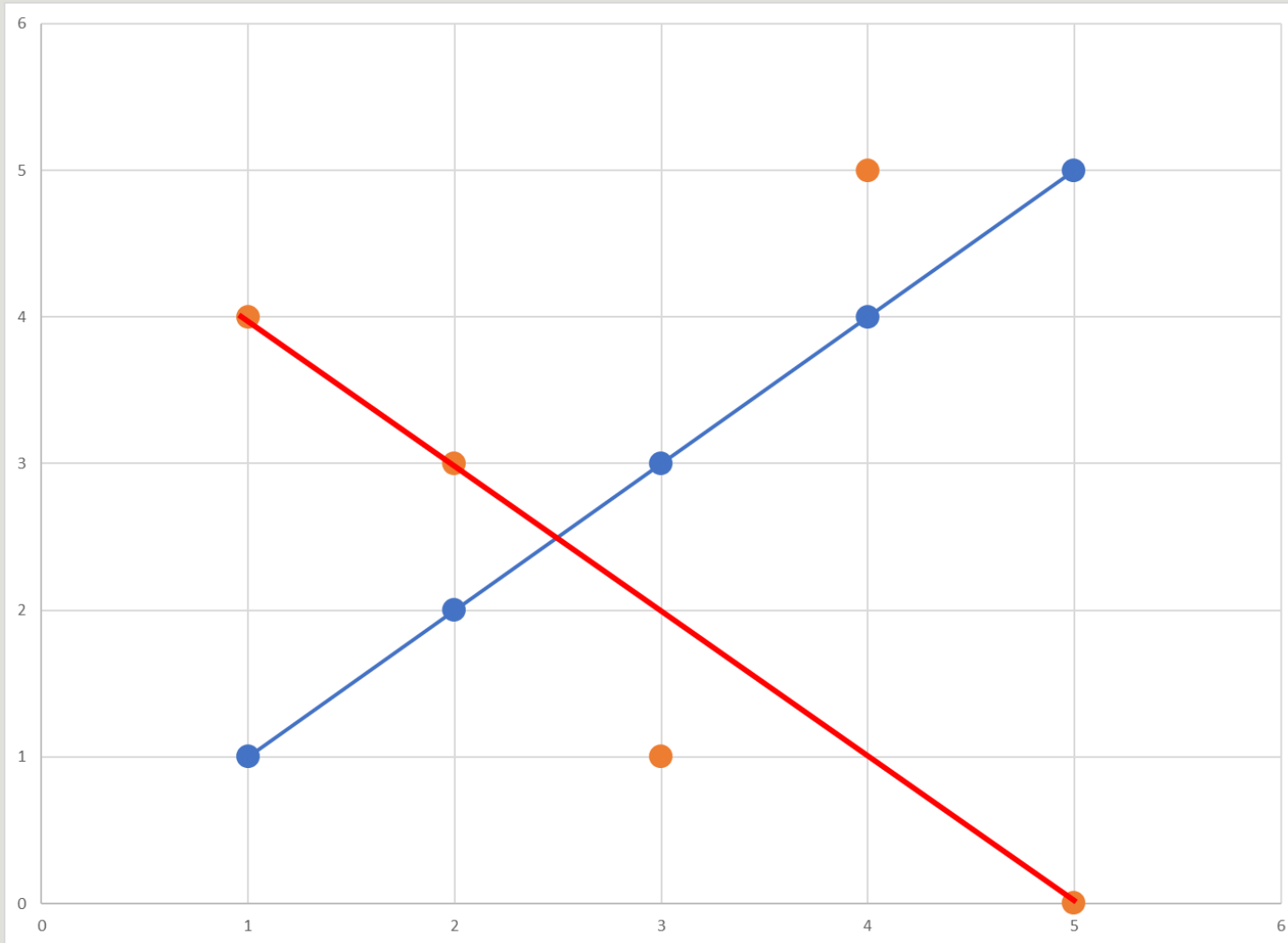
Pige efficace

- Paramètre de l'algorithme
- Devient équivalent à RANSAC en pire cas

PROSAC

Progressive Sample Consensus

1. Trier les points par qualité
2. Pige m points dans le top qualité
3. Calculer modèle
4. Vérifier le modèle sur tous les points
5. Répéter 1 à 4 en incorporant **progressivement** des points de moins bonne qualité, jusqu'aux critères d'arrêt



Critères d'arrêt

- **Solution non aléatoire**
- **Estimé pessimiste**

Critères d'arrêt

- Probabilité de trouver un meilleur modèle trop faible

$t := 0, n := m, n^* := N$

Repeat until a solution satisfying eqs. (12), (9) is found.

1. Choice of the hypothesis generation set

$t := t + 1$

if $(t = T'_n) \ \& \ (n < n^*)$ then $n := n + 1$ (see eqn. 4)

2. Semi-random sample \mathcal{M}_t of size m

if $T'_n < t$ then

The sample contains $m - 1$ points selected from \mathcal{U}_{n-1} at random and \mathbf{u}_n

else

Select m points form \mathcal{U}_n at random

3. Model parameter estimation

Compute model parameters p_t from the sample \mathcal{M}_t

4. Model verification

Find support (i.e. consistent data points) of the model with parameters p_t

Select termination length n^* if possible according to Section 2.2

Incorporation progressive des points

Pige

Calculer le modèle

Vérifier et critères d'arrêt

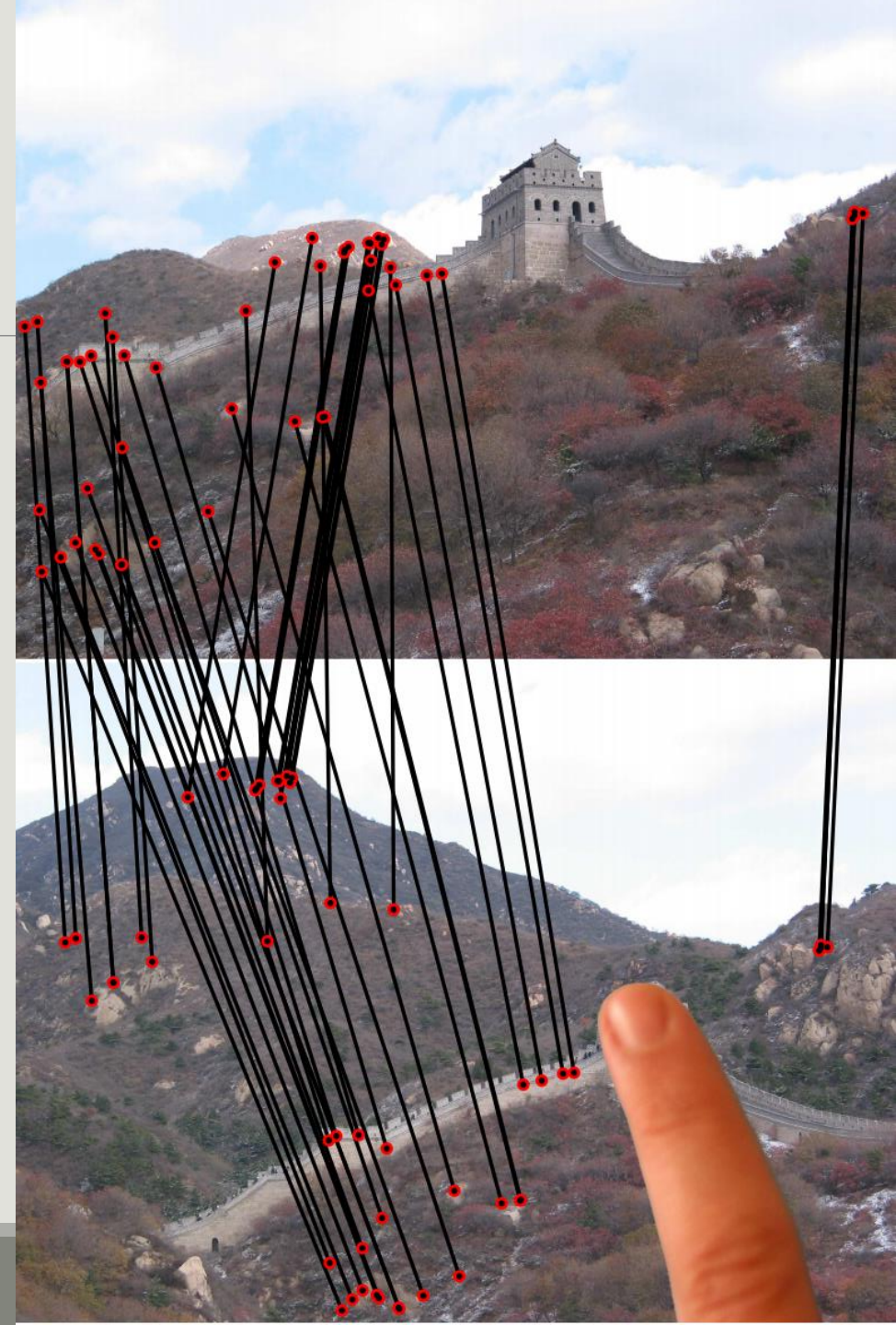
PROSAC

Code

- Python
- Sera disponible ici après nettoyage et optimisation :
 - <https://github.com/willGuimont/PROSAC>

Résultats

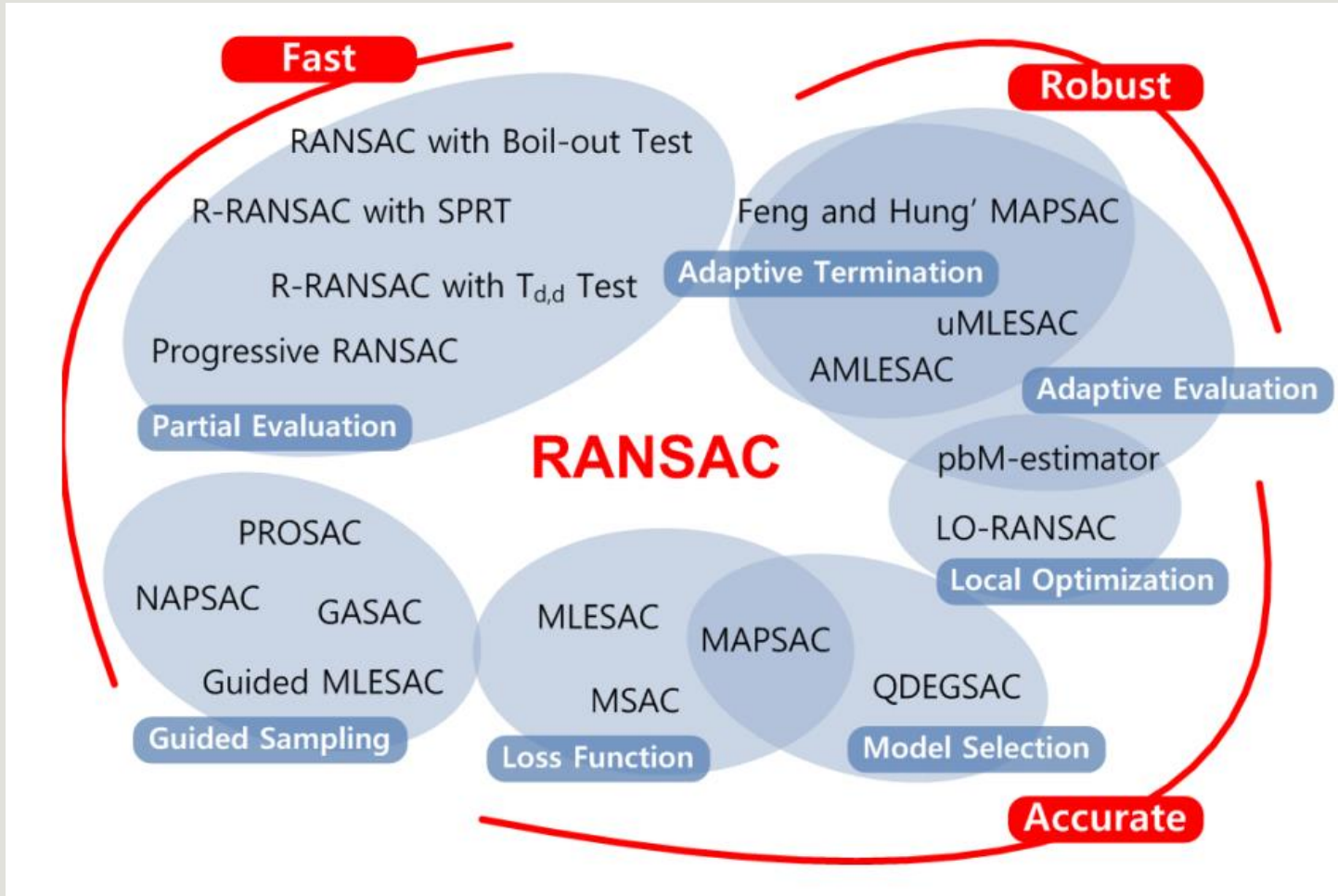
- RANSAC
 - 106 534 piges
 - 10.76 secondes
- PROSAC
 - 9 piges
 - 0.06 secondes



Résultats

- Pourcentage d'inliers: 9.2%
- RANSAC
 - $\approx 8.43 \cdot 10^7$ piges
 - Pas réaliste !
- PROSAC
 - 3 576 piges
 - 0.76 secondes





Conclusion



Questions?