

Bundle adjustment with known positions Ajustement de faisceaux avec positions de caméras connues

Soutenance de stage de L3

Antoine BARRIER | Ségolène MARTIN

antoine.barrier@ens-paris-saclay.fr | segolene.martin@ens-paris-saclay.fr

Encadrés par Carlo DE FRANCHIS et Jean-Michel MOREL

école
normale
supérieure
paris—saclay



29 juin 2017

OBJECTIF : modéliser des scènes en 3D à partir d'images prises par des caméras

- modélisation de scènes terrestres fixes (drones, caméras au sol, ...),
- modélisation du *relief de la Terre* à partir d'images satellites.

- 1 Présentation du problème et rappels
- 2 Données réelles
- 3 Fiabilité de nos résultats : validation croisée
- 4 Comparaison avec d'autres méthodes

Présentation du problème et rappels

Reconstruction 3D

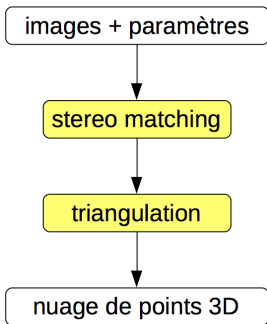


FIGURE – Reconstruction 3D



FIGURE – Stereo matching

Présentation du problème et rappels

Reconstruction 3D

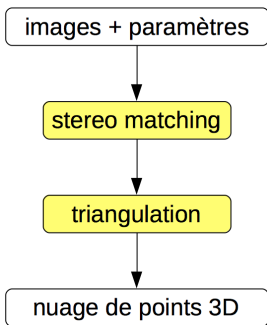


FIGURE – Reconstruction 3D

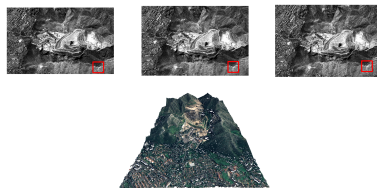


FIGURE – Triangulation

Présentation du problème et rappels

Reconstruction 3D

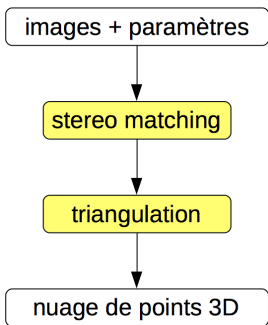


FIGURE – Reconstruction 3D

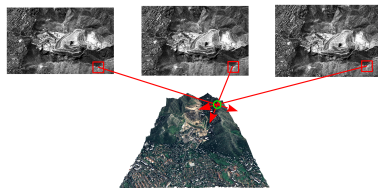


FIGURE – Triangulation

Présentation du problème et rappels

Reconstruction 3D

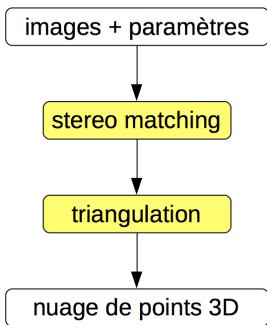


FIGURE – Reconstruction 3D

DIFFÉRENTS TYPES D'ERREURS :

- position des caméras,
- rotation des caméras,
- coordonnées des points lors du *stereo matching*.

Présentation du problème et rappels

Reconstruction 3D et *bundle adjustment*

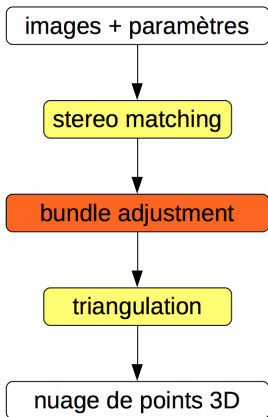


FIGURE – Reconstruction 3D avec *bundle adjustment*

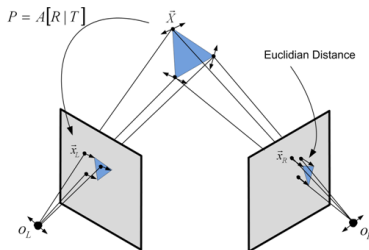


FIGURE – Principe du *bundle adjustment*¹

1. Richard HARTLEY & Andrew ZISSERMAN, *Multiple view geometry in computer vision* (Cambridge University Press) – 2003

Présentation du problème et rappels

Les méthodes actuelles ne sont pas adaptées

- Méthodes *fonctionnelles*, mais qui ne donnent pas accès aux *paramètres réels*.
→ **impossible de faire du recollement de scènes et des comparaisons de scènes**
- Dans le *cas satellitaire*, l'erreur de position sur les caméras est très faible et ne nécessite donc pas d'être corrigée.
→ **toutes les méthodes essaient de réestimer les positions**

Présentation du problème et rappels

Les méthodes actuelles ne sont pas adaptées

- Méthodes *fonctionnelles*, mais qui ne donnent pas accès aux *paramètres réels*.
→ **impossible de faire du recollement de scènes et des comparaisons de scènes**
- Dans le *cas satellitaire*, l'erreur de position sur les caméras est très faible et ne nécessite donc pas d'être corrigée.
→ **toutes les méthodes essaient de réestimer les positions**

NOTRE APPROCHE

On suppose que les positions des caméras sont parfaitement connues.

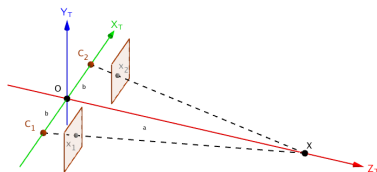
*On espère que cette simplification permettra d'estimer les paramètres **réels** de chaque situation.*

Présentation du problème et rappels

Fonction cible du problème

- Avec 2 caméras et 1 point de correspondance, on veut minimiser la quantité

$$\det \left(R_1^{-1} K^{-1} \mathbf{x}_1 \mid R_2^{-1} K^{-1} \mathbf{x}_2 \mid \tilde{\mathbf{C}}_2 - \tilde{\mathbf{C}}_1 \right) \quad (1)$$



- Avec K caméras et N points, on minimise :

$$\begin{aligned} F : \quad \mathbb{R}^{3K} \times \mathbb{R}^{2NK} &\longrightarrow \mathbb{R}^{\binom{K}{2}N} \\ (\theta, \mathbf{X}) &\longmapsto (\det_{i_1, i_2}^j)_{\substack{1 \leq i_1 < i_2 \leq K \\ 0 \leq j \leq N-1}} \end{aligned}$$

où $\det_{i_1, i_2}^j$ est le déterminant (1) associé aux caméras i_1, i_2 et au point j .
Typiquement, $K \simeq 6$ et N vaut quelques milliers de points.

Présentation du problème et rappels

Stratégie

Soit $(\theta_0, \mathbf{X}_0)$ les paramètres en notre possession. On veut trouver une meilleure estimation des paramètres réels $(\theta_*, \mathbf{X}_*)$.

On applique à F une formule de Taylor à l'ordre 1 :

$$F(\theta, \mathbf{X}) = F(\theta_0, \mathbf{X}_0) + dF_{\theta}(\theta_0, \mathbf{X}_0) d\theta + dF_{\mathbf{X}}(\theta_0, \mathbf{X}_0) d\mathbf{X} + o((d\theta, d\mathbf{X}))$$

où $d\theta = \theta - \theta_0$ et $d\mathbf{X} = \mathbf{X} - \mathbf{X}_0$.

Puis on résout $F(\theta, \mathbf{X}) = 0$ (puisque $F(\theta_*, \mathbf{X}_*) = 0$) à l'ordre 1, en essayant de trouver $(d\theta, d\mathbf{X})$ tel que :

$$\underbrace{\left(\frac{\partial F}{\partial \theta}(\theta_0, \mathbf{X}_0), \frac{\partial F}{\partial \mathbf{X}}(\theta_0, \mathbf{X}_0) \right)}_M \begin{pmatrix} d\theta \\ d\mathbf{X} \end{pmatrix} = -F(\theta_0, \mathbf{X}_0)$$

Présentation du problème et rappels

Simulations

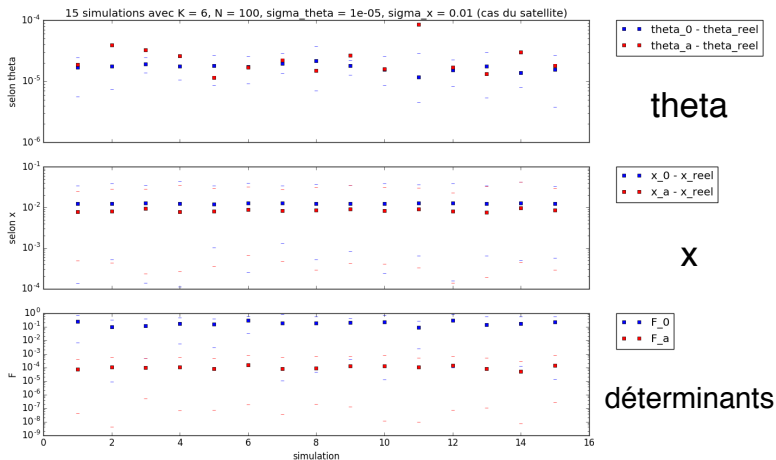


FIGURE – Simulations dans le cas général

Le problème semble mal posé !
Comment modifier le problème ?

Le problème semble mal posé ! Comment modifier le problème ?

DEUX SOLUTIONS :

- *fixer certains paramètres* (en général les coordonnées des points de correspondances) et n'essayer de réestimer que les autres (en général les angles de rotation),
- *ajouter un terme forçant*, c'est-à-dire minimiser :

$$\left\| M \begin{pmatrix} d\theta \\ d\mathbf{X} \end{pmatrix} + F(\theta_0, \mathbf{X}_0) \right\|^2 + \frac{\lambda^2}{\sigma^2} \left\| \begin{pmatrix} d\theta \\ d\mathbf{X} \end{pmatrix} \right\|^2$$

où λ est à déterminer (à optimiser).

Présentation du problème et rappels

Premiers résultats

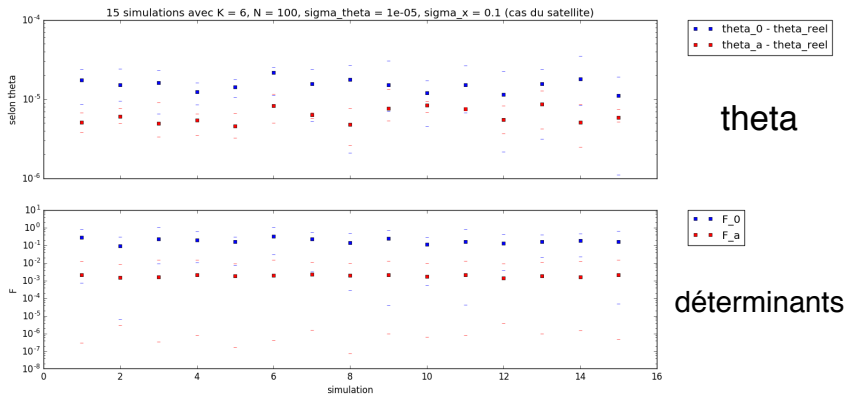
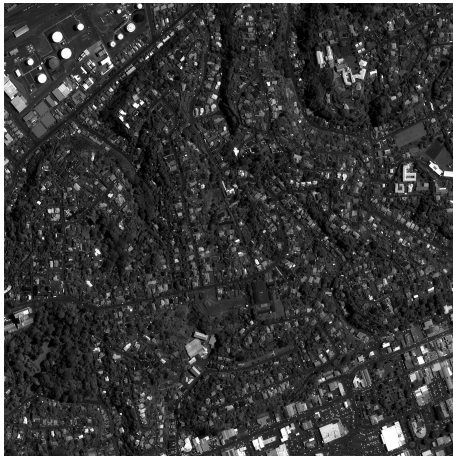
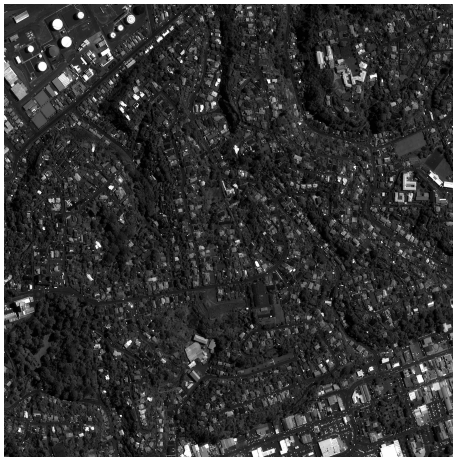


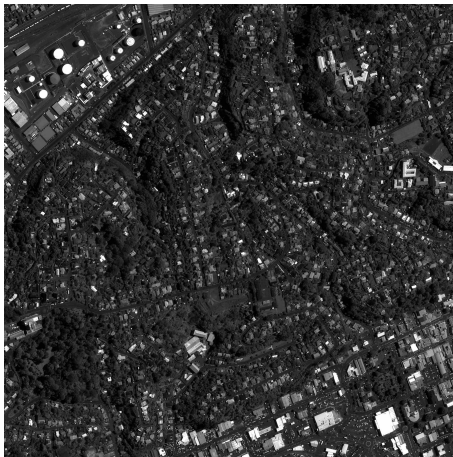
FIGURE – Simulations avec terme forçant



Quelques images de Napier prises par Pléiades.



Quelques images de Napier prises par Pléiades.



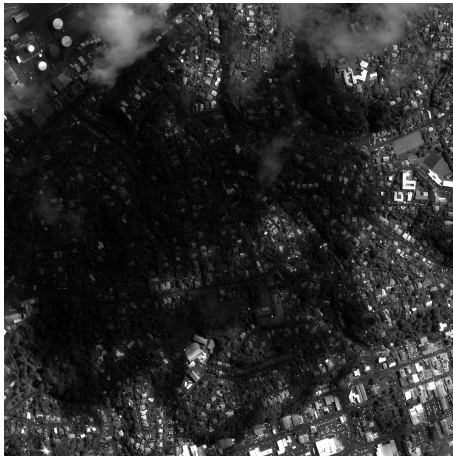
Quelques images de Napier prises par Pléiades.



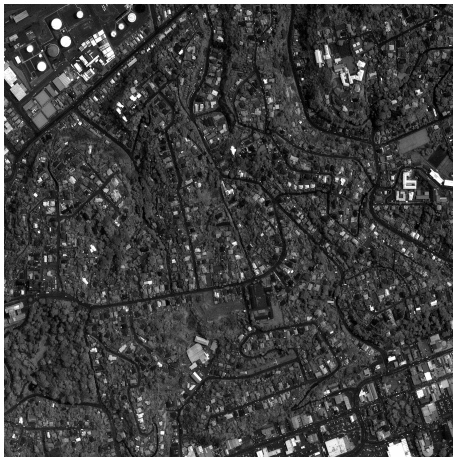
Quelques images de Napier prises par Pléiades.



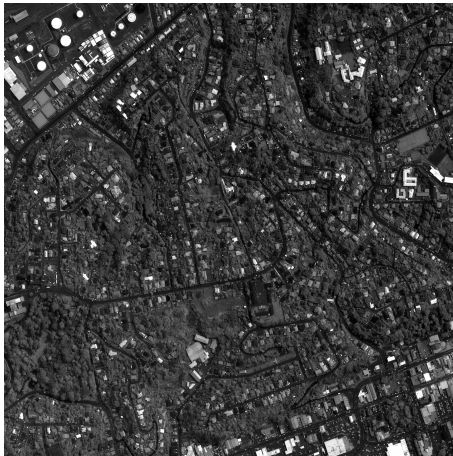
Quelques images de Napier prises par Pléiades.



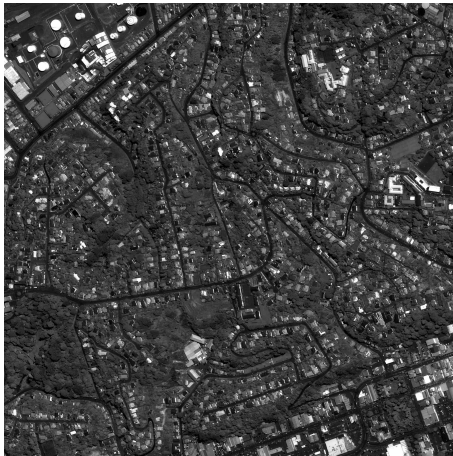
Quelques images de Napier prises par Pléiades.



Quelques images de Napier prises par Pléiades.



Quelques images de Napier prises par Pléiades.



Quelques images de Napier prises par Pléiades.



Quelques images de Napier prises par Pléiades.

À NOTRE DISPOSITION :

- 20 images de la ville de Napier,
- une *mesure* très précise des 20 matrices de projection des caméras au moment des prises de vues,
- pour chaque paire de caméra, un fichier de correspondances contenant sur chaque ligne un *matching* entre un point (x_1, y_1) de la première caméra et un point (x_2, y_2) de la deuxième, *les coordonnées des points étant inexactes*.

À NOTRE DISPOSITION :

- 20 images de la ville de Napier,
- une *mesure* très précise des 20 matrices de projection des caméras au moment des prises de vues,
- pour chaque paire de caméra, un fichier de correspondances contenant sur chaque ligne un *matching* entre un point (x_1, y_1) de la première caméra et un point (x_2, y_2) de la deuxième, *les coordonnées des points étant inexactes*.

PROBLÈME : comment, à partir des fichiers de correspondances deux à deux, obtenir *rapidement* des points de correspondances sur K caméras?

PROBLÈME : comment, à partir des fichiers de correspondances deux à deux, obtenir *rapidement* des points de correspondances sur K caméras ?

2. NetworkX Developers, *package Networkx*
<http://networkx.github.io/>

PROBLÈME : comment, à partir des fichiers de correspondances deux à deux, obtenir *rapidement* des points de correspondances sur K caméras ?

→ on se ramène à un problème de graphes !

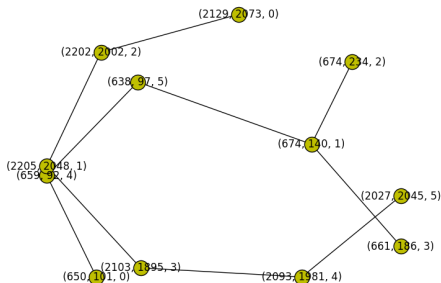


FIGURE – Sous-graphe associé au problème²

2. NetworkX Developers, *package Networkx*
<http://networkx.github.io/>

LE PROBLÈME PRINCIPAL :

on n'a pas accès à la vérité de terrain

(c'est-à-dire aux paramètres réels de la situation)

→ comment s'assurer que notre algorithme permet d'obtenir une meilleure estimation que les données initiales ?

PREMIÈRE MÉTHODE

On choisit K caméras puis on sépare les points de correspondance en plusieurs groupes d'intersection un unique point ($K = 6$).

On réestime alors les paramètres avec chaque groupe et on vérifie qu'ils sont proches.

Fiabilité de nos résultats : validation croisée

Première méthode

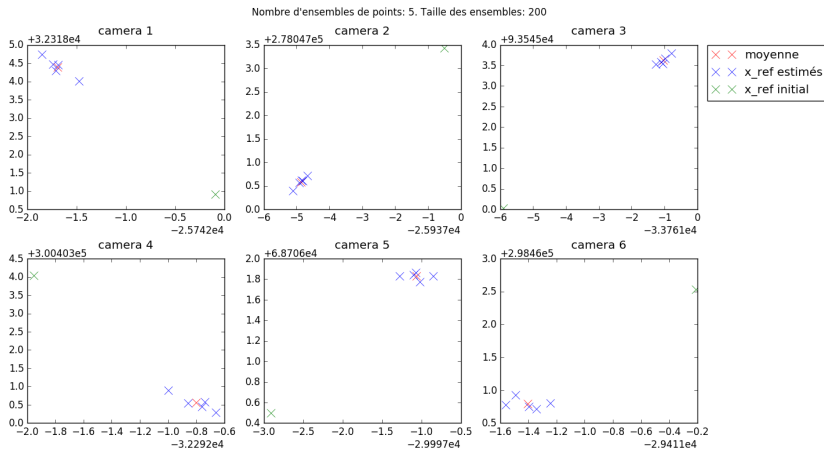


FIGURE – Un bon x_{ref}

Fiabilité de nos résultats : validation croisée

Première méthode

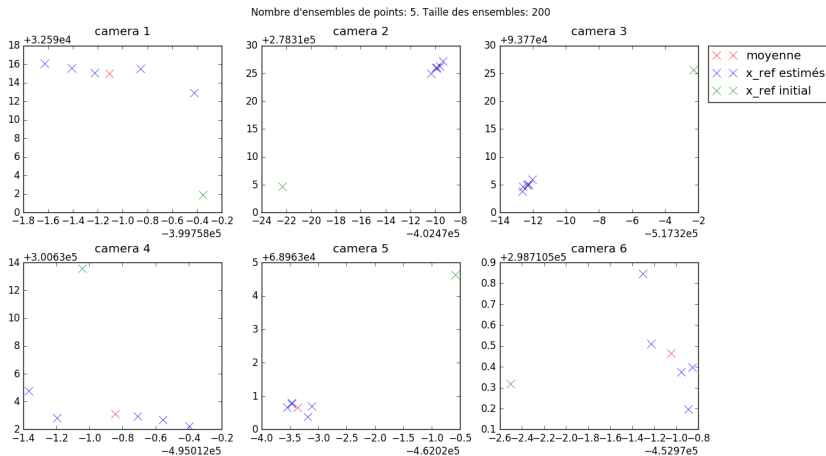


FIGURE – Un mauvais x_{ref}

Fiabilité de nos résultats : validation croisée

Première méthode

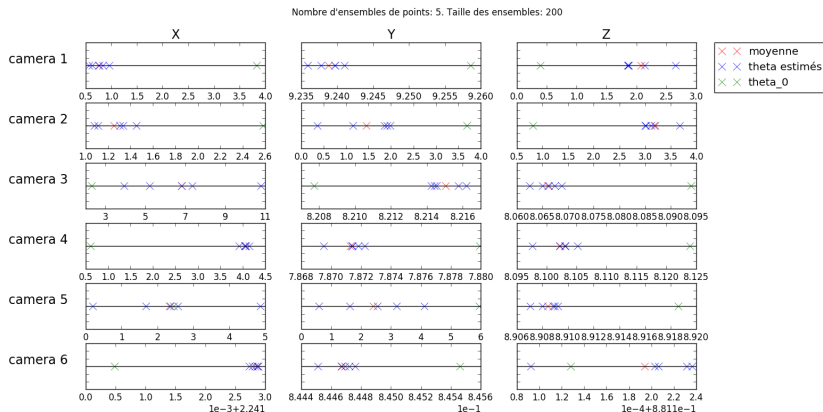
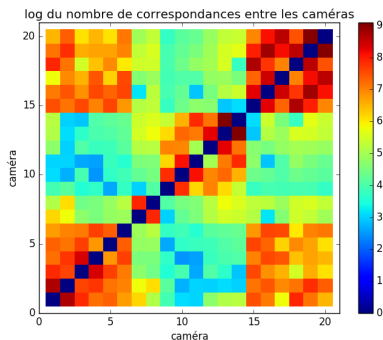
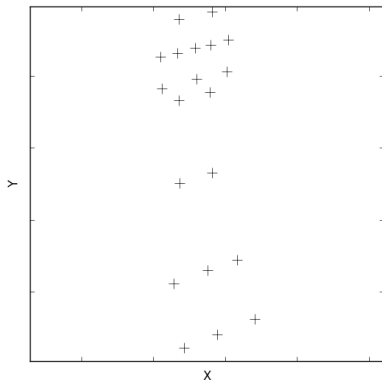


FIGURE – Les angles

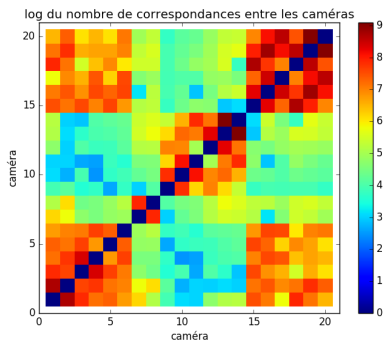
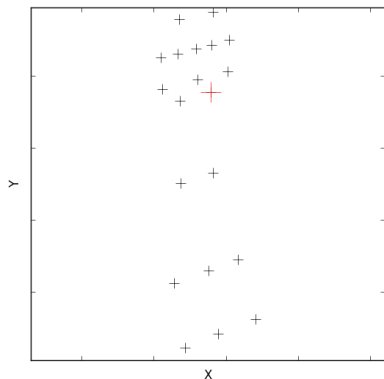
DEUXIÈME MÉTHODE

On cherche deux groupes de K caméras d'intersection une unique caméra.
On réestime alors les paramètres angulaires avec chaque groupe et on vérifie qu'ils sont proches.



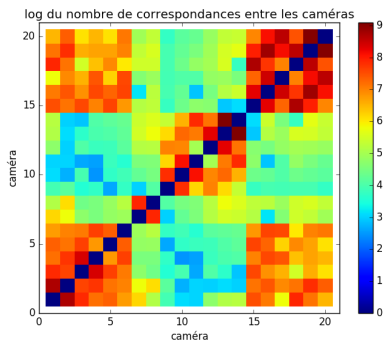
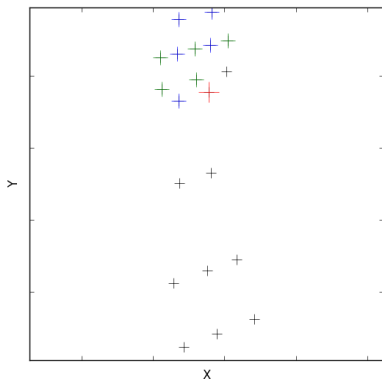
DEUXIÈME MÉTHODE

On cherche deux groupes de K caméras d'intersection une unique caméra.
On réestime alors les paramètres angulaires avec chaque groupe et on vérifie qu'ils sont proches.



DEUXIÈME MÉTHODE

On cherche deux groupes de K caméras d'intersection une unique caméra.
On réestime alors les paramètres angulaires avec chaque groupe et on vérifie qu'ils sont proches.



Fiabilité de nos résultats : validation croisée

Deuxième méthode

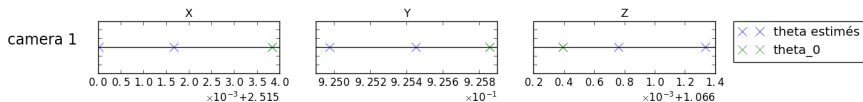


FIGURE – Comparaison des résultats à partir de différents ensembles de caméras

Deux bibliothèques à notre disposition : OpenMVG³ et le bundler⁴

3. Pierre MOULON, Pascal MONASSE, Renaud MARLET & Others, *OpenMVG*

<https://github.com/openMVG/openMVG>

4. Noah SNAVELY, *Bundler : Structure from Motion (SfM) for Unordered Image Collections*

<http://www.cs.cornell.edu/~snaveley/bundler>

Deux bibliothèques à notre disposition : OpenMVG³ et le bundler⁴

→ il est difficile de comparer précisément nos résultats avec ceux des méthodes classiques de *Bundle Adjustment*,

3. Pierre MOULON, Pascal MONASSE, Renaud MARLET & Others, *OpenMVG*
<https://github.com/openMVG/openMVG>

4. Noah SNAVELY, *Bundler : Structure from Motion (SfM) for Unordered Image Collections*
<http://www.cs.cornell.edu/~snave/bundler>

Deux bibliothèques à notre disposition : `OpenMVG`³ et le `bundler`⁴

- il est difficile de comparer précisément nos résultats avec ceux des méthodes classiques de *Bundle Adjustment*,
- cependant, nous sommes convaincus que les méthodes actuelles ne fonctionnent pas, notamment pour les situations satellitaires : placement des caméras aléatoires, nombreux points aberrants sur les scènes reconstruites.

3. Pierre MOULON, Pascal MONASSE, Renaud MARLET & Others, *OpenMVG*
<https://github.com/openMVG/openMVG>

4. Noah SNAVELY, *Bundler : Structure from Motion (SfM) for Unordered Image Collections*
<http://www.cs.cornell.edu/~snave/bundler>

Ce qui a été fait :

- ✓ construire une théorie et un algorithme de résolution,
- ✓ obtenir une première méthode de correction de l'erreur angulaire et de l'erreur sur les points dans les images : d'après les simulations, réduction d'un **facteur 10 sur les angles**, d'un **facteur 2 sur les points**,
- ✓ premiers tests sur des données réelles.

Ce qu'il resterait à faire :

- ✗ mise en pratique systématique sur des groupes d'images satellitaires,
- ✗ comparer plus précisément notre méthode avec des algorithmes de Bundle Adjustment classique,
- ✗ trouver des conditions d'unicité de la solution.

-  Richard HARTLEY & Andrew ZISSERMAN, *Multiple view geometry in computer vision* (Cambridge University Press) – 2003
-  Bill TRIGGS, Philip MCLAUCHLAN, Richard HARTLEY & Andrew FITZGIBBON, *Bundle Adjustment – A Modern Synthesis* (In Vision Algorithms : Theory and Practice, pp. 298–372) – 2000
-  Noah SNAVELY, *Bundler : Structure from Motion (SfM) for Unordered Image Collections*
<http://www.cs.cornell.edu/~snaveley/bundler>
-  Pierre MOULON, Pascal MONASSE, Renaud MARLET & Others, *OpenMVG*
<https://github.com/openMVG/openMVG>
-  NetworkX Developers, *package Networkx*
<http://networkx.github.io/>

Des questions?