

2013-2014

# Chef D'œuvre M2 IM

Calibrage automatique de captures vidéo en caméra HD + Kinect ©

Mathieu Bérengère  
Tardy Benjamin  
Vilardell Alexandre

## Revue de recette

Clients:  
Christophe Collet  
Alain Crouzil  
Equipe TCI



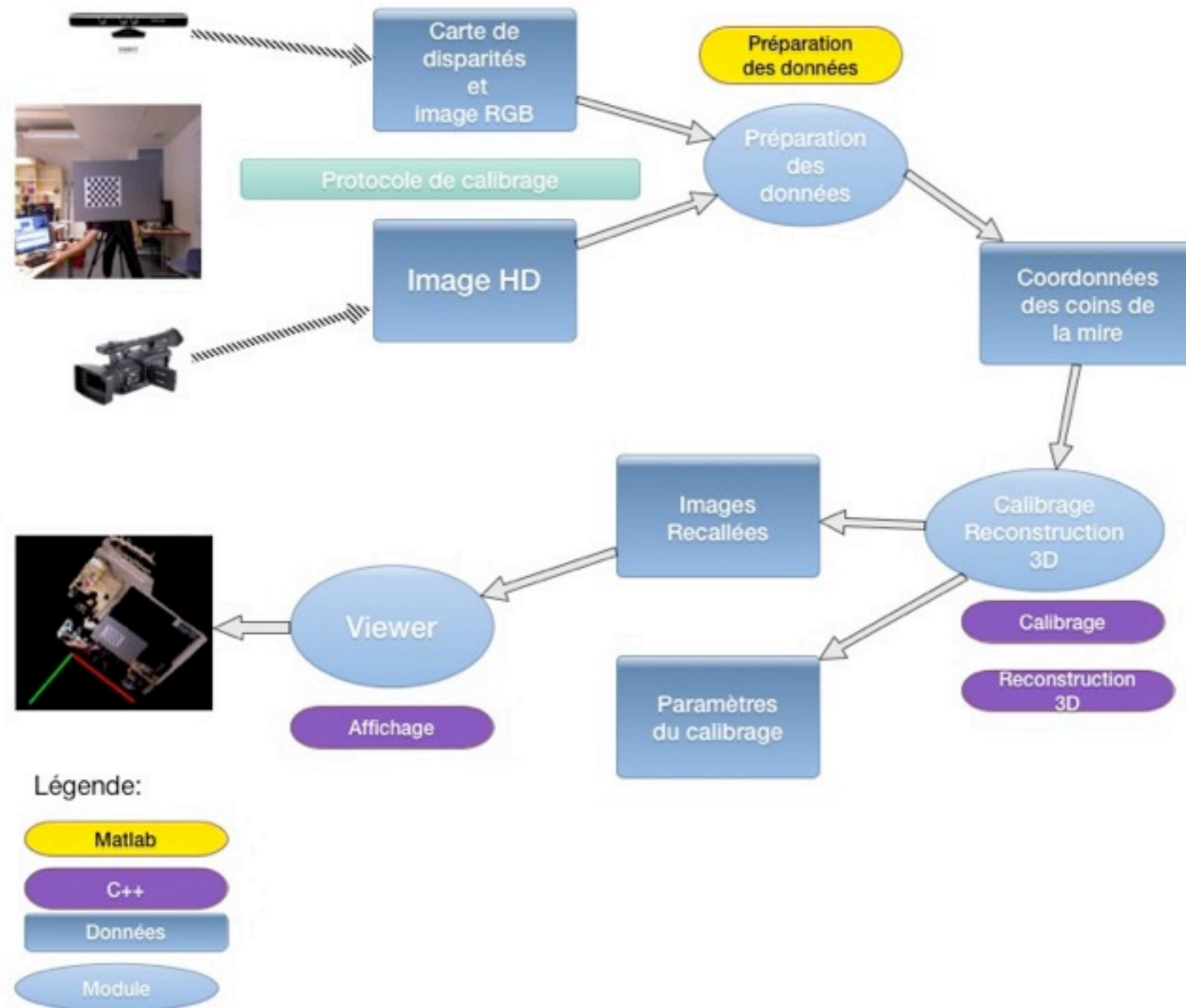
# Plan

- Rappels
- Présentation du travail effectué
- Bilan

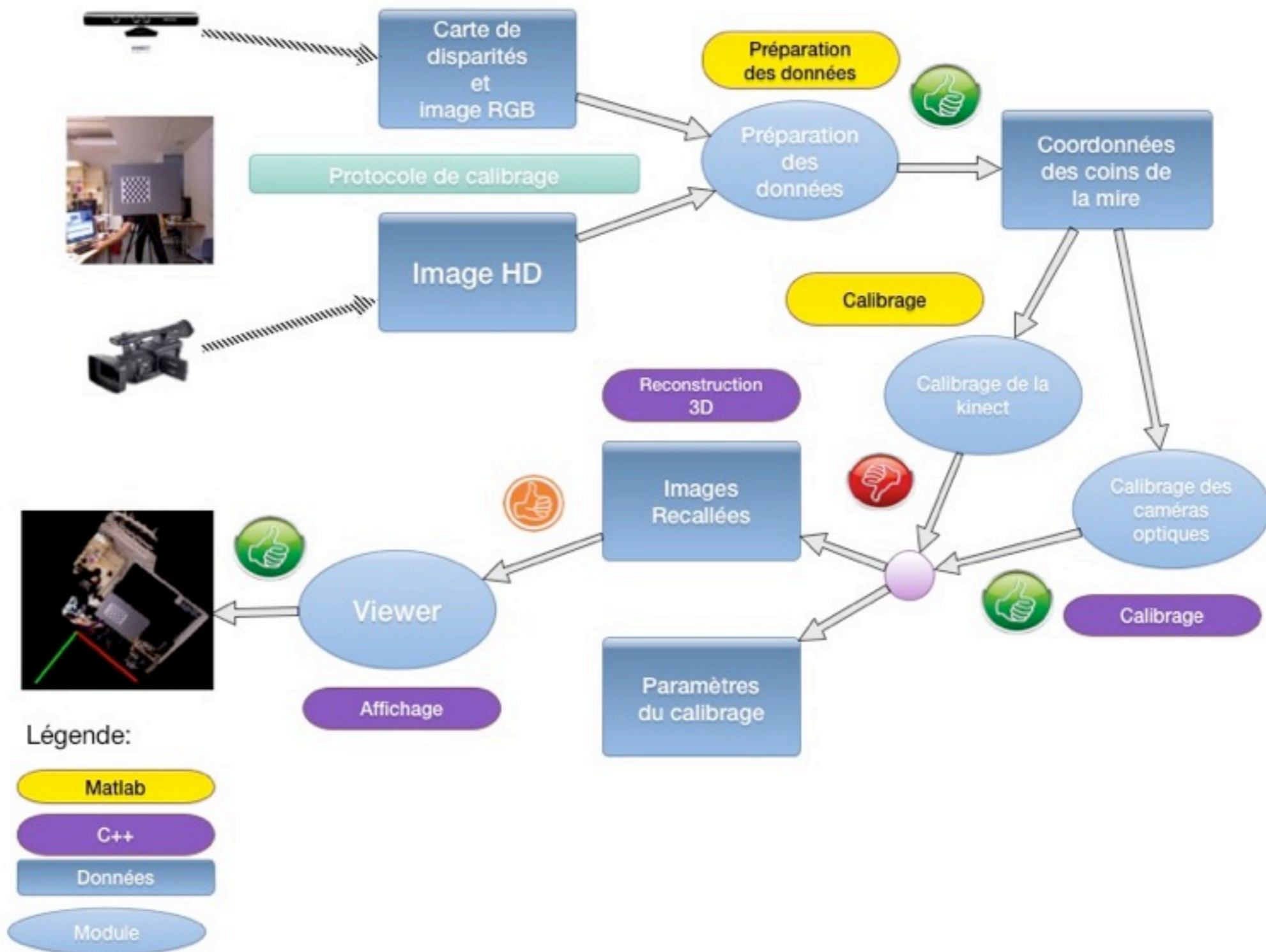
# Rappel des objectifs

- Un protocole d'acquisition
  - Extraction des points caractéristiques
  - Un programme de calibrage du système
  - Un programme de reconstruction 3D
  - Un programme d'affichage
- ➔ Une chaîne de traitement fonctionnelle

# Aperçu



# Systeme final



# Protocole d'acquisition

- Validé pendant la séance d'acquisition
- Problème de récupération des données
- Aucun test de calibrage effectué
- Mise à jour du protocole pour faciliter la constitution du jeu de données

# Protocole d'acquisition



# Perspectives

- Faire un guide sous forme de vidéo
- Faire une application guide pour le positionnement de la mire
- Capturer les images directement



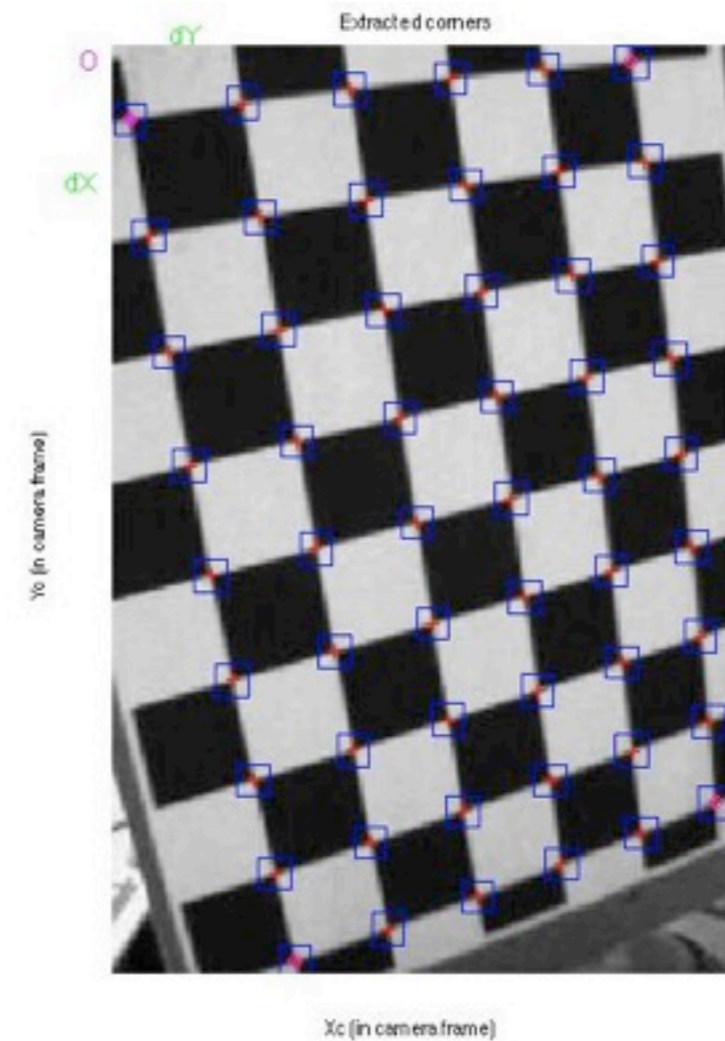
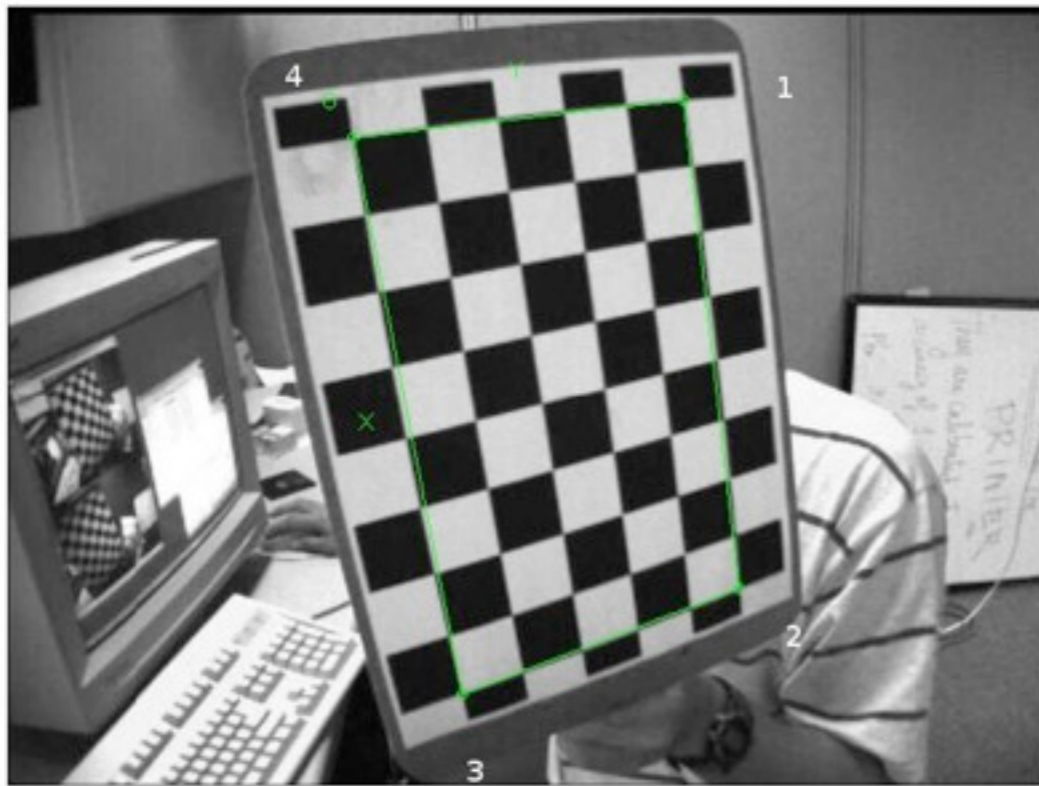
# Préparation des données



# Préparation des données

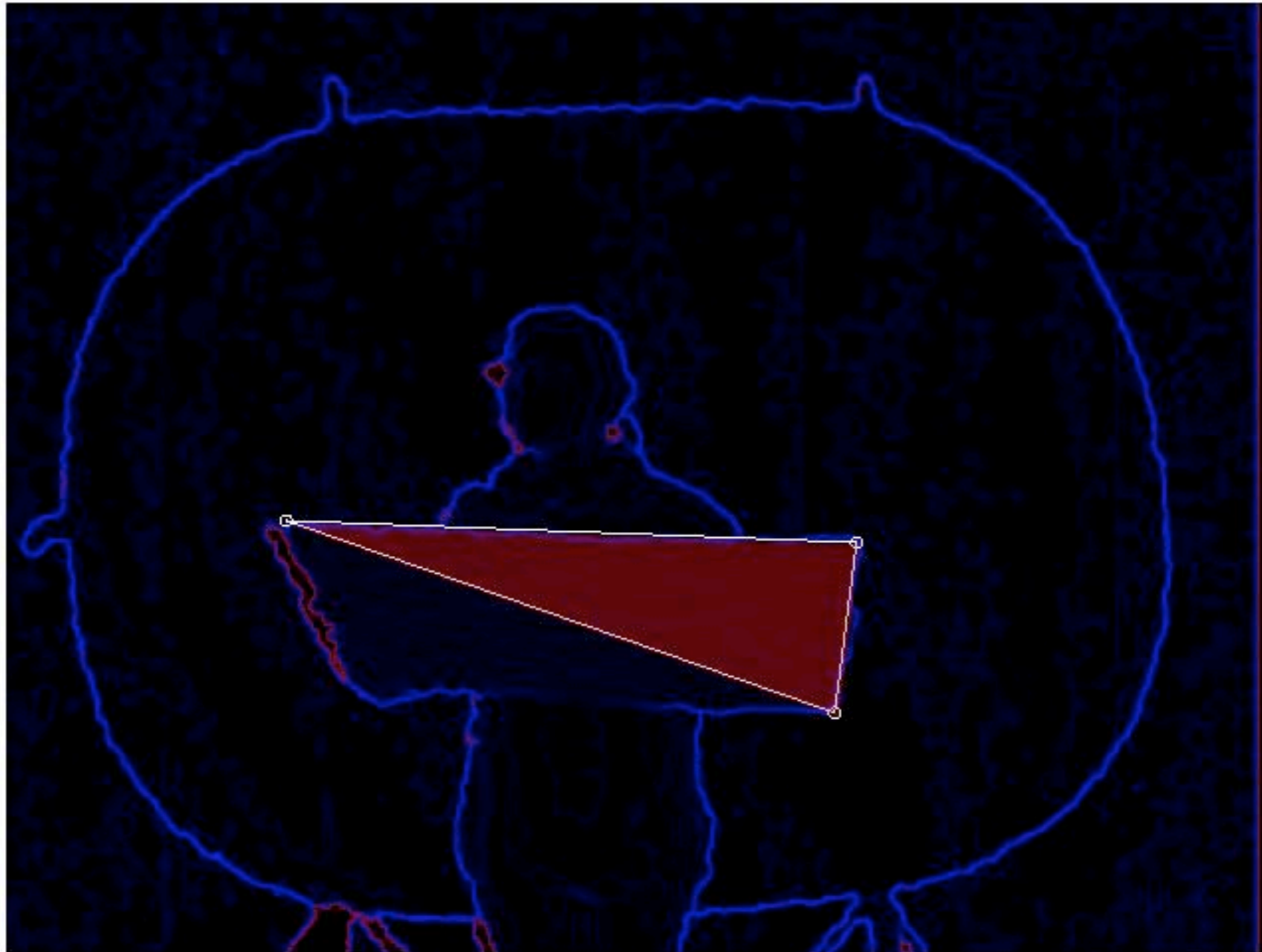
- Basé sur le code de D.Herrera
- Modifié pour s'adapter à notre application
- Programme matlab indépendant qui stocke les coordonnées des points dans un fichier xml

# Préparation des données



# Préparation des données

Select corners of polygon enclosing plane points [left=add corner,ESC=remove corner,right=end]



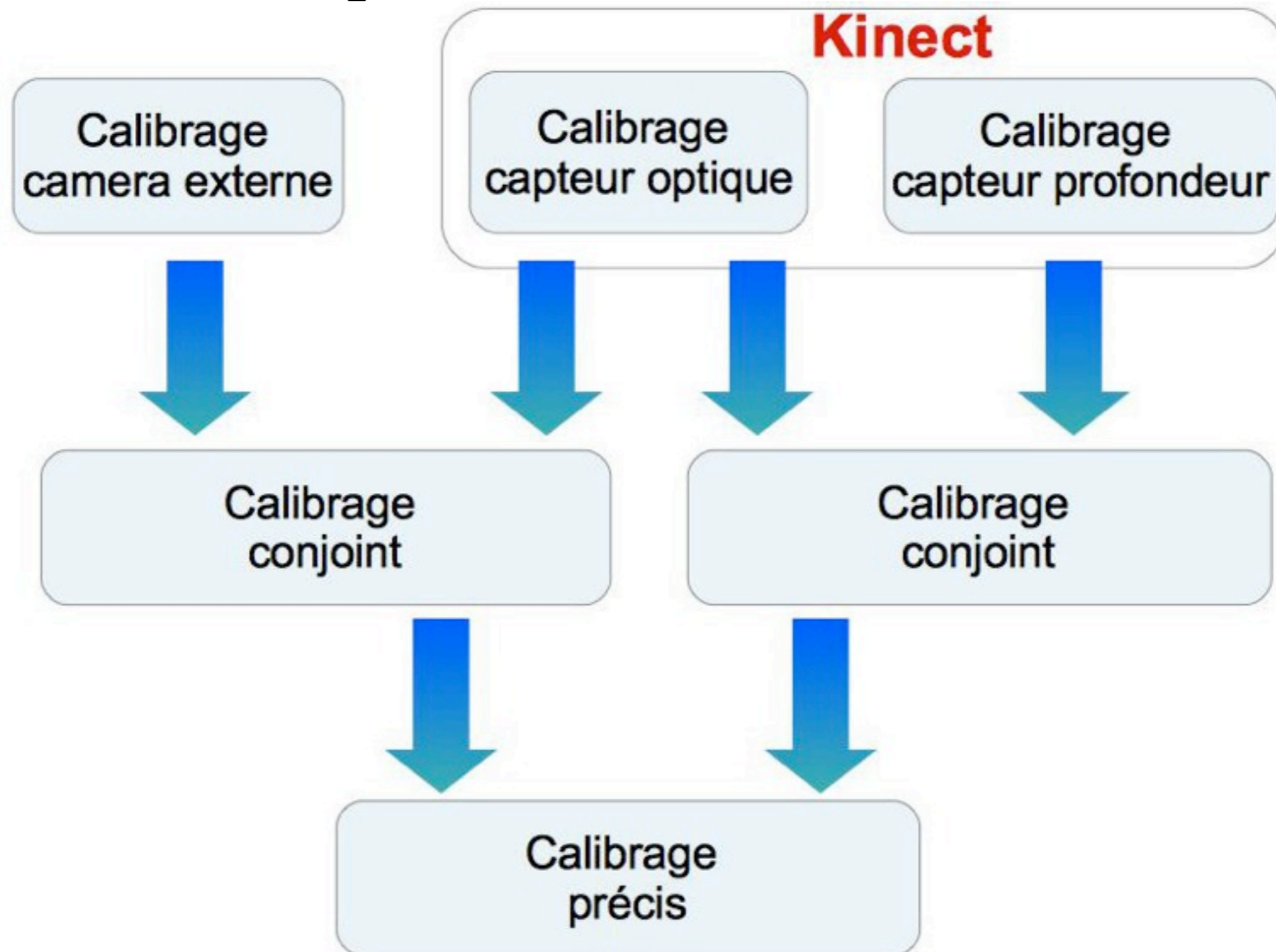
# Perspectives

- Implémenter une méthode de sélection automatique des points caractéristiques
- Conversion du programme Matlab en C++

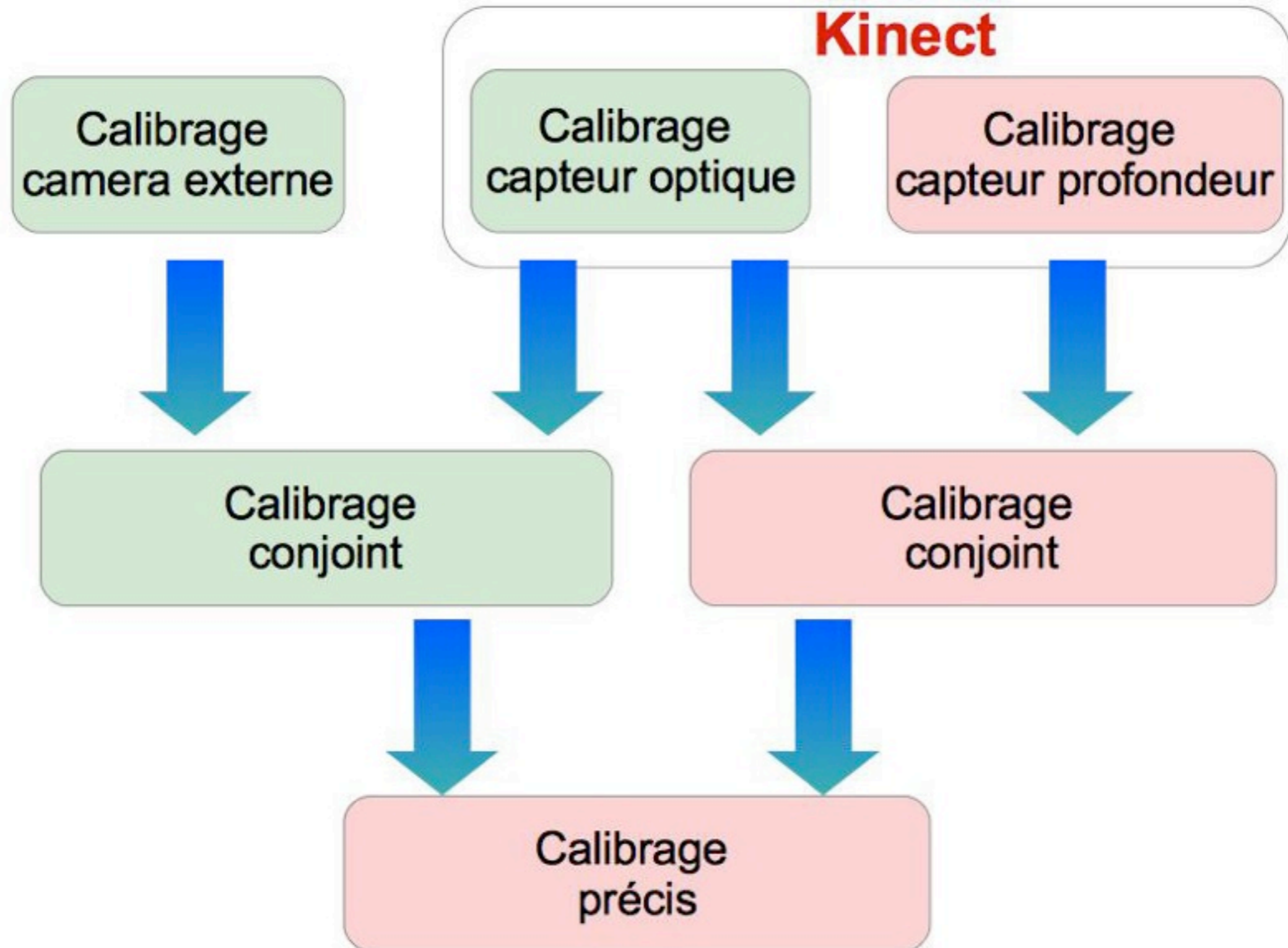
# Calibrage



# Objectifs initiaux



# Objectifs atteints





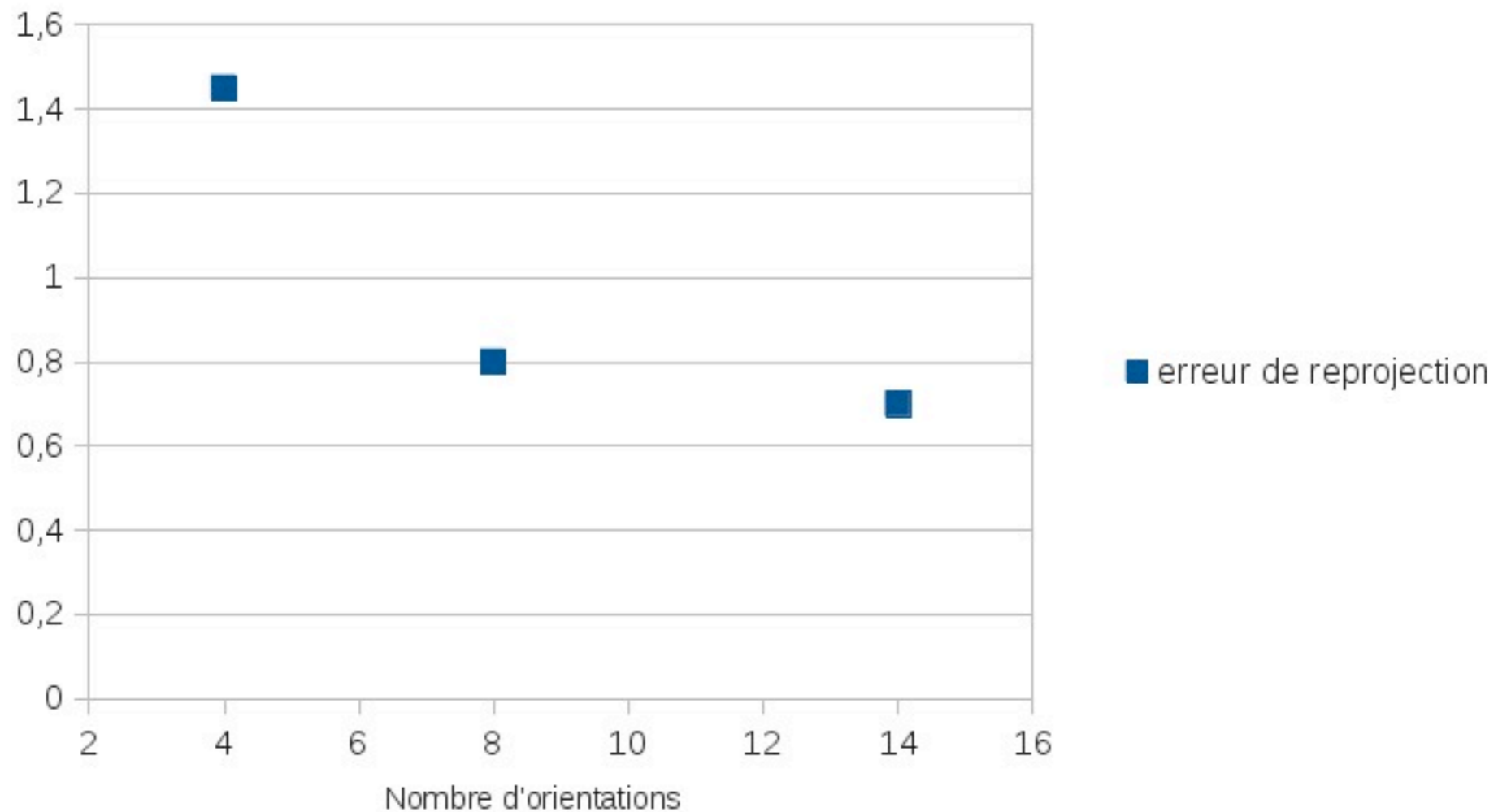
# Problèmes rencontrés

- Les méthodes en C++ plus sensibles aux erreurs numériques
- Le jeu données de D.Herrera pas suffisants pour réaliser l'ensemble des tests

# Calibrage de 2 caméras optiques

- Problème très classique
  - Une application de référence, développée en Matlab par J.Y.Bouguet
  - Réutilisation de méthodes fournies par OpenCV
- Test sur un jeu de données fourni par J.Y.Bouguet: concluant
- Test avec les images de D.Herrera : echec

# Calibrage de 2 caméras optiques



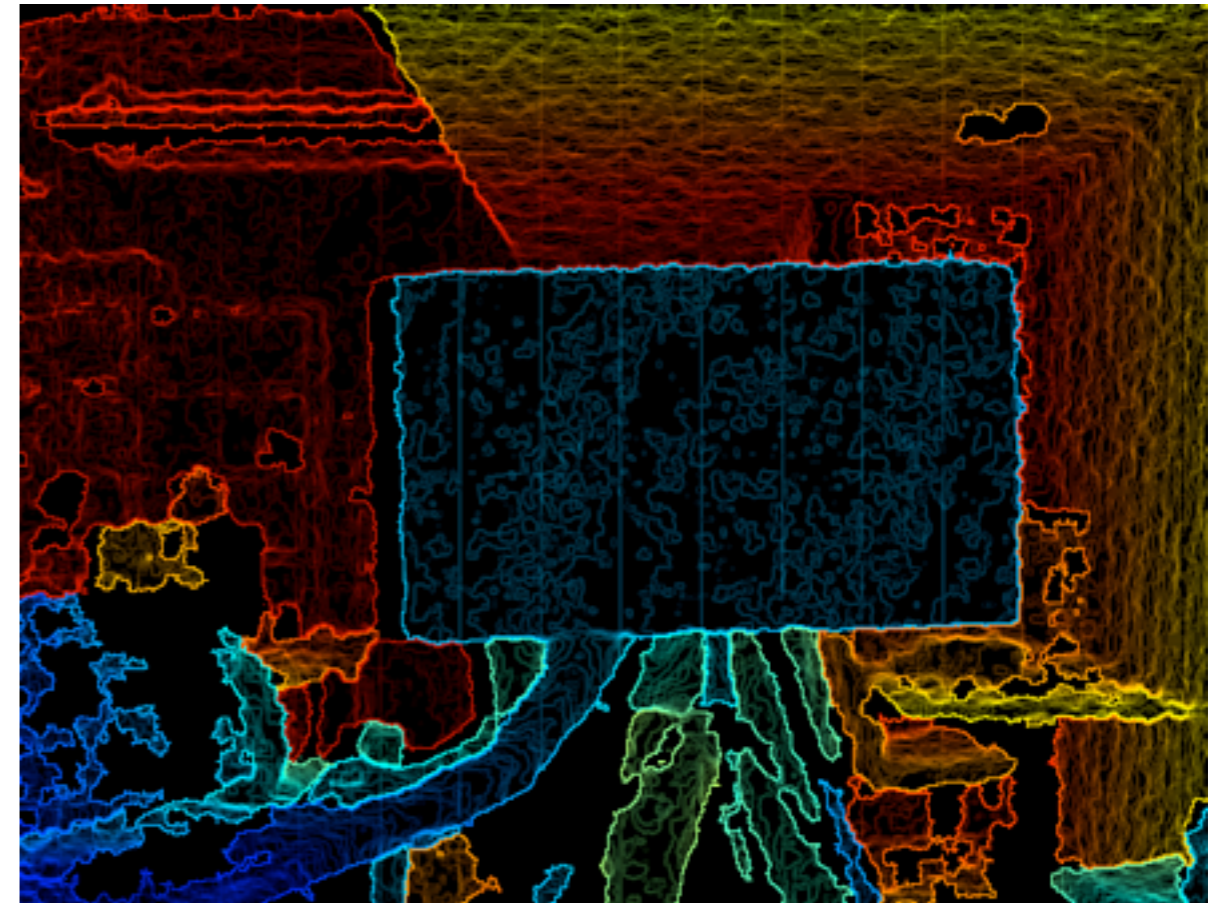
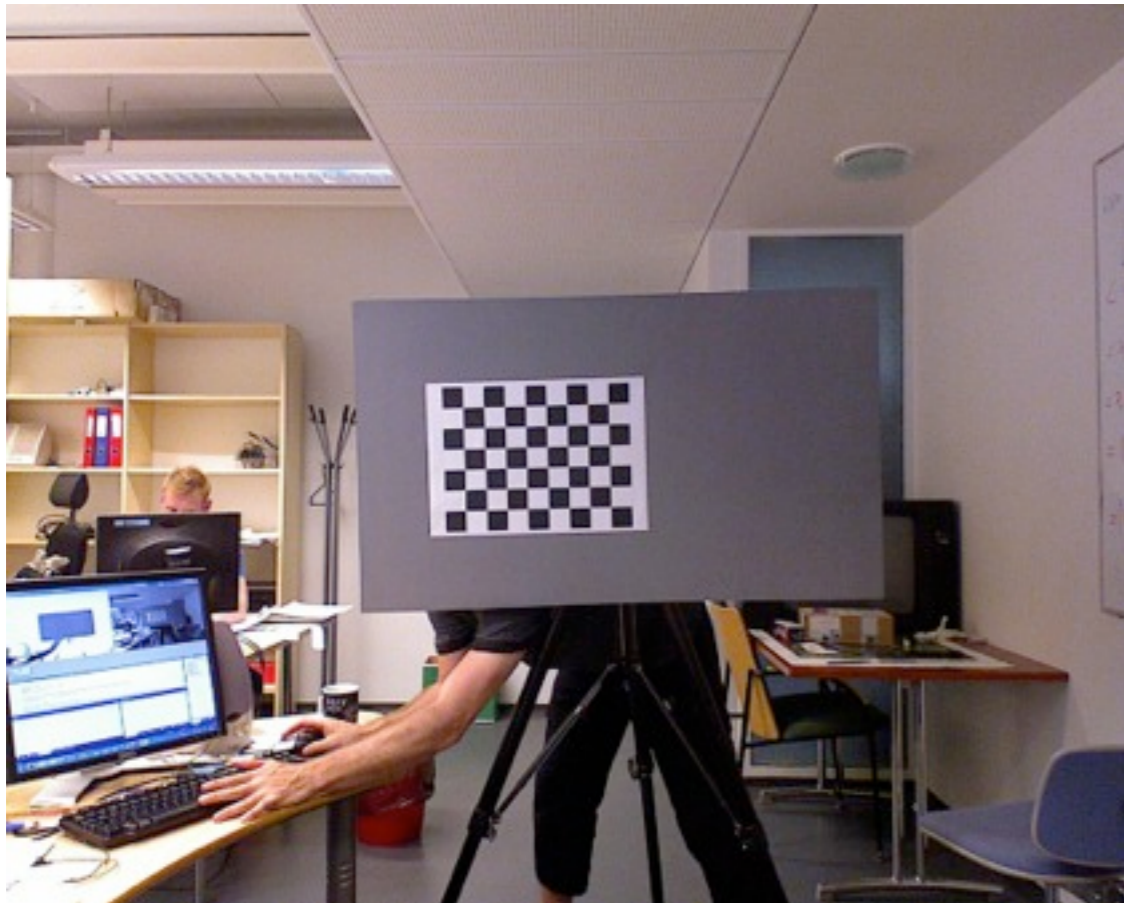
# Calibrage de 2 caméras optiques

- Utilisation de Levenverg-Marquardt pour améliorer les résultats: échec
- Caractéristiques du problème
  - Les données initiales sont très proches de la solution
  - Il y a un grand nombre d'inconnues à déterminer : 64 contre 5 dans le prototype
- Test avec deux implémentations de la fonction de coût

# Calibrage capteur de profondeur

- Etape 1 : déterminer les paramètres intrinsèques et extrinsèques propres à tout capteur
  - Utilisation des mêmes méthodes que pour les capteurs optiques
  - Beaucoup moins de données
  - Beaucoup plus de bruit sur les données
- Etape bloquante pour le reste du calibrage

# Calibrage capteur de profondeur



# Calibrage capteur de profondeur

- Utilisation du jeu de données de D.Herrera
  - D.Herrera donne les points caractéristiques et leurs positions dans le repère scène
  - Acquisition des points caractéristiques avec notre programme
  - D.Herrera donne les valeurs à obtenir pour chaque paramètres
  - A partir de ces données, notre application échoue à produire un résultat correct

# Calibrage capteur de profondeur

- Peut on améliorer les résultats avec Levenberg-Marquardt?
  - Peu d'inconnues (juste les 8 paramètres intrinsèques du capteur de profondeur)
  - Pour une inconnue la valeur initiale est très éloignée de la solution
  - Echec de l'implémentation de CMinpack => minimum local



# Calibrage capteur de profondeur

- D.Herrera fait-il vraiment mieux?
  - Inspection du code => la fonction pour réaliser le calibrage approximatif du capteur de profondeur n'est jamais appelée
  - Si on appelle cette fonction avec les données de D.Herrera, les résultats produits par son implémentation sont faux (point principal abbérant)

# Décisions prises avec le client

- Calibrage des deux capteurs de la Kinect : code de D.Herrera
- Calibrage des deux capteurs optiques : notre application
- Réaliser l'acquisition d'un nouveau jeu de données

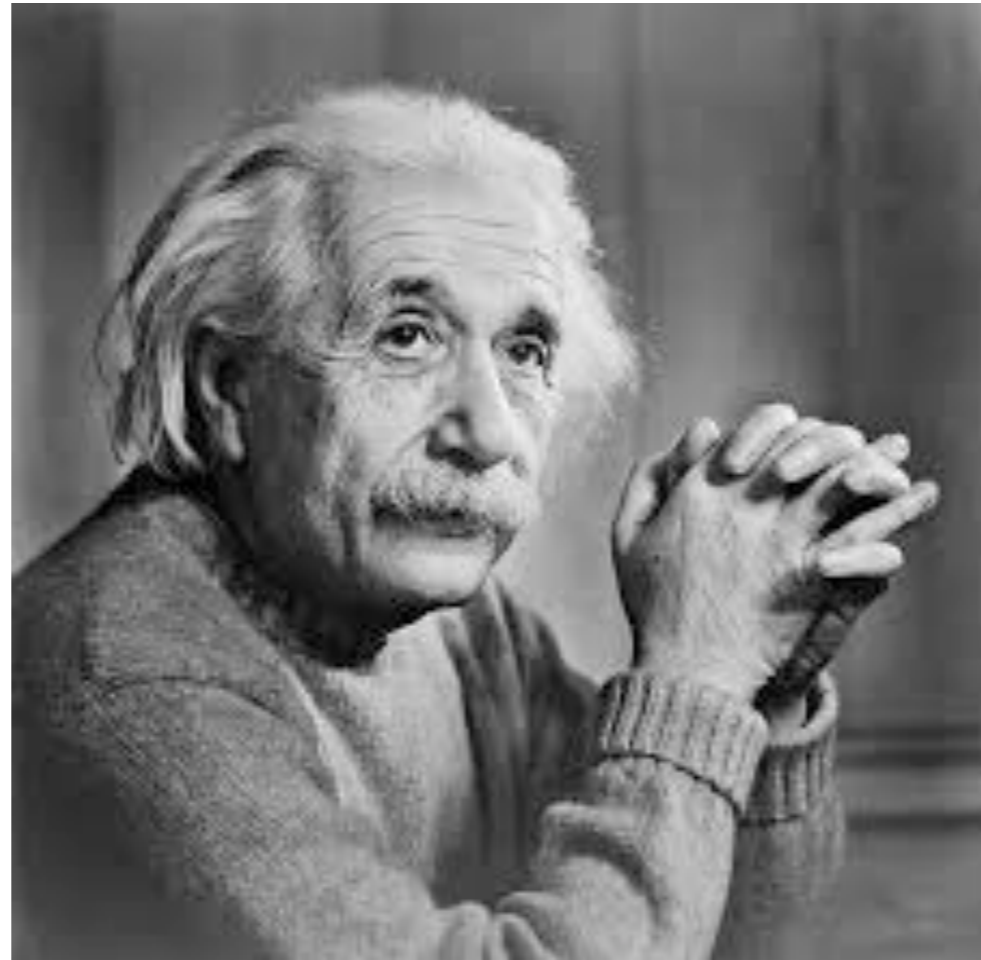
# Décisions prises avec le client

- Il faut pratiquer des tests plus poussés sur le code de D.Herrera
  - Acquisition d'un nouveau jeu de données
  - Test du code de D.Herrera avec ces images

# Perspectives

- Travail sur le code de D.Herrera
  - Tests
  - Mise au propre
- Travail sur les erreurs numériques
  - Tenter de corriger l'erreur numérique
  - Regarder les bibliothèques pour le calcul haute précision (gnu multiple précision)

# Reconstruction 3D



# Objectifs

- Corriger les distorsions sur les images couleurs
- Corriger les distorsions sur les cartes de disparité
- Générer un nuage de points :
  - Coordonnées dans la scène
  - Couleur dans l'image HD

# Objectifs

- Corriger les distorsions sur les images couleurs
- Corriger les distorsions sur les cartes de disparité
- Générer un nuage de points:
  - Coordonnées dans la scène
  - Couleur dans l'image HD

# Objectifs

- Corriger les distorsions sur les images couleurs
- Corriger les distorsions sur les cartes de disparité
- Générer un nuage de points:
  - Coordonnées dans la scène
  - Couleur dans l'image HD



# Objectifs

- Corriger les distorsions sur les images couleurs
- Corriger les distorsions sur les cartes de disparités
- Générer un nuage de points:
  - Coordonnées dans la scène
  - Couleur dans l'image HD

# Objectifs

- Corriger les distorsions sur les images couleurs
- Corriger les distorsions sur les cartes de disparité
- Générer un nuage de points:
  - Coordonnées dans la scène
  - Couleur dans l'image acquise par la Kinect

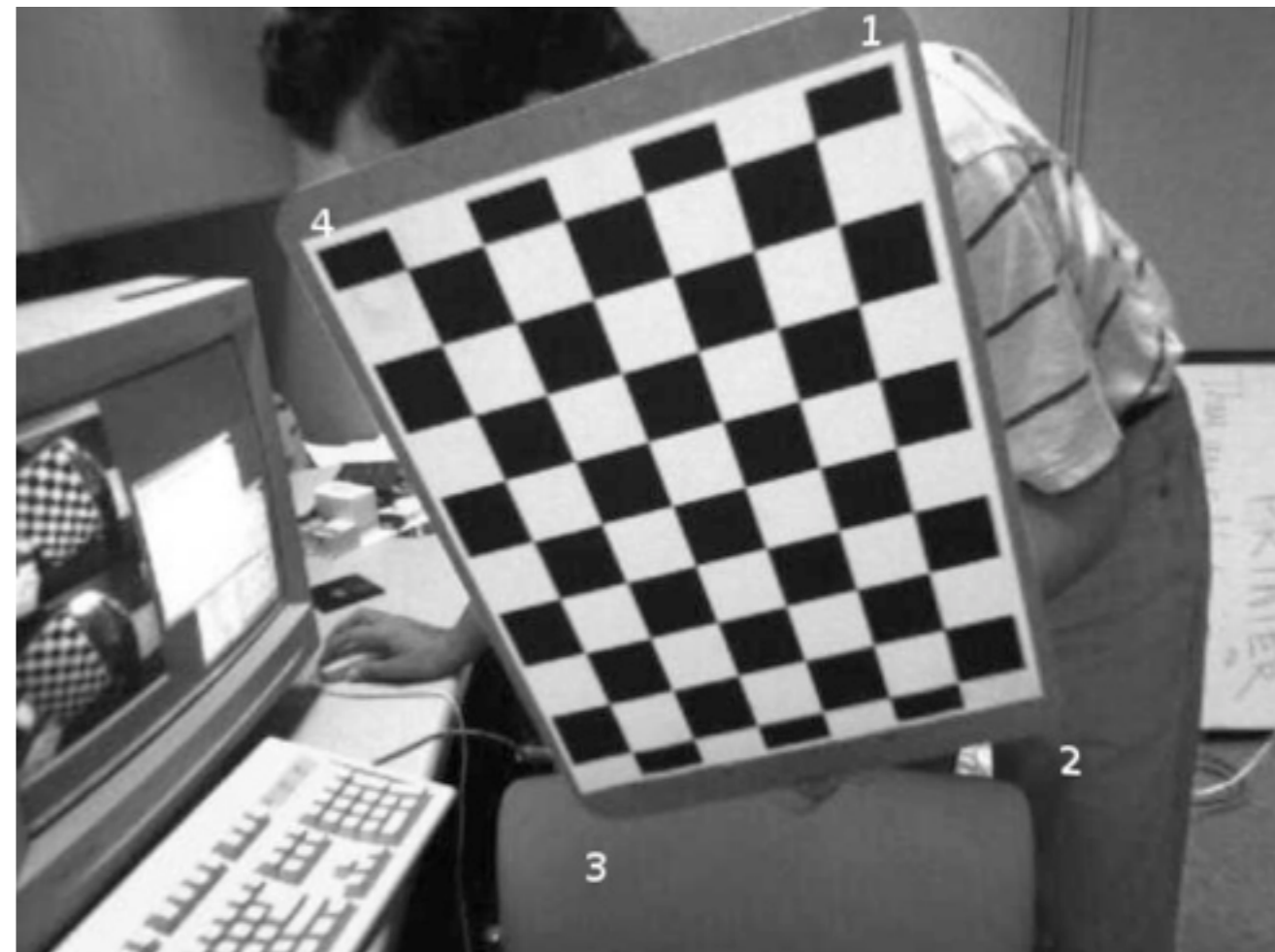
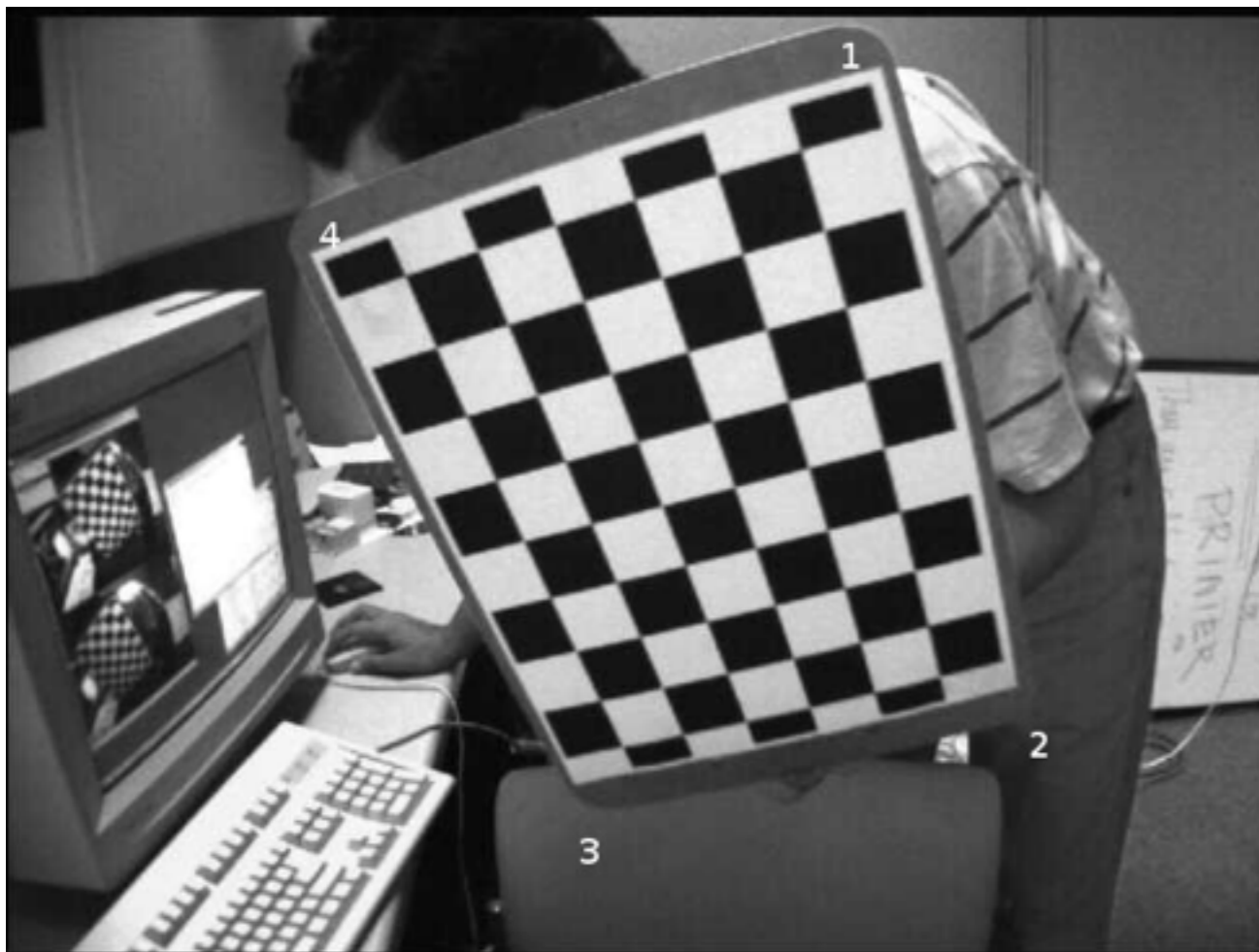
# Problèmes

- Retard important pris lors du développement du module de calibrage
- Test du module très fastidieux
  - Réaliser un prototype sous matlab
  - Pour chaque étape de l'algorithme
    - Exporter les valeurs
    - Calculer l'erreur entre les deux implémentations

# Problèmes

- Tolère mal les erreurs d'arrondis
  - Les résultats d'une étape sont réexploités par les suivantes
  - Les données de calibrage => 8 chiffres après la virgule

# Correction des distorsions



# Affichage



# Affichage

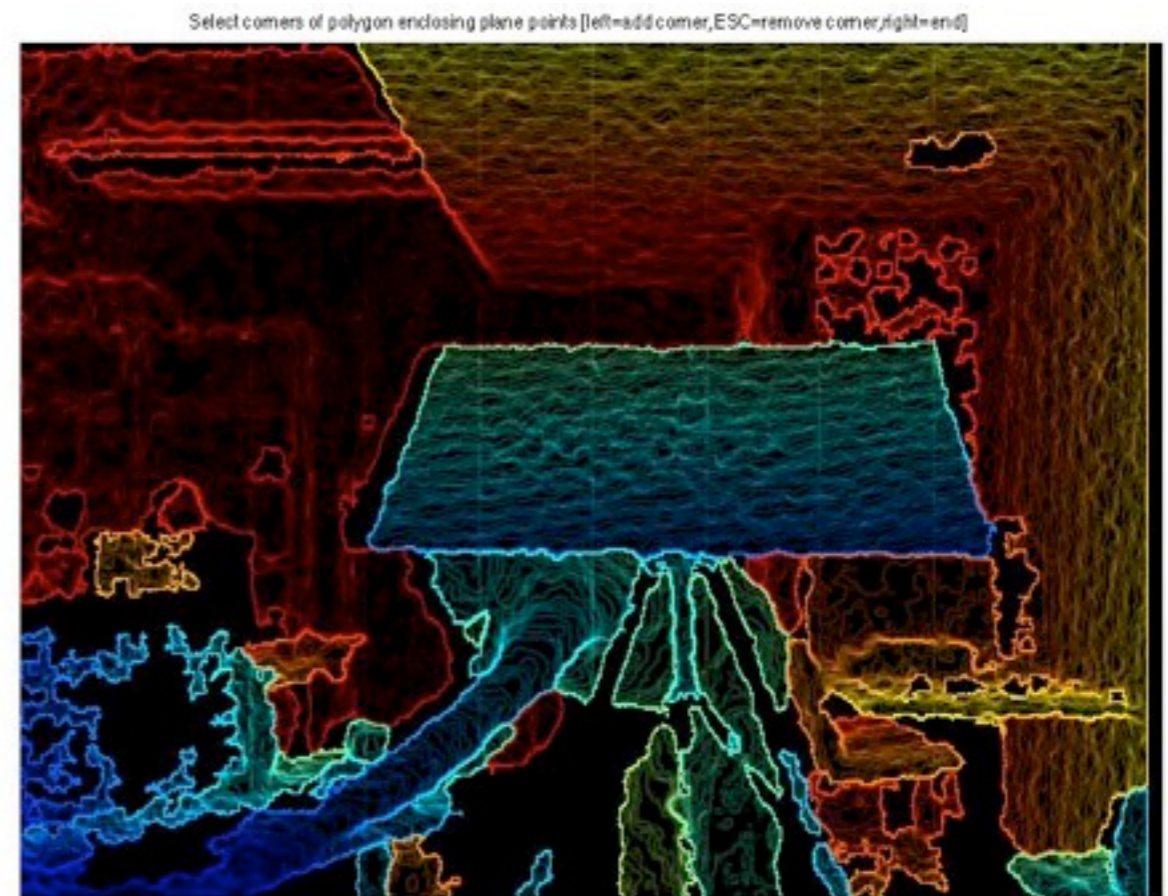
- Prototype réalisé durant la phase de conception
- Objectifs : afficher le nuage de point
- Le module fonctionne

# Tests

- A partir des images et des résultats fournis par D.Herrera
- Uniquement appréciation visuelle

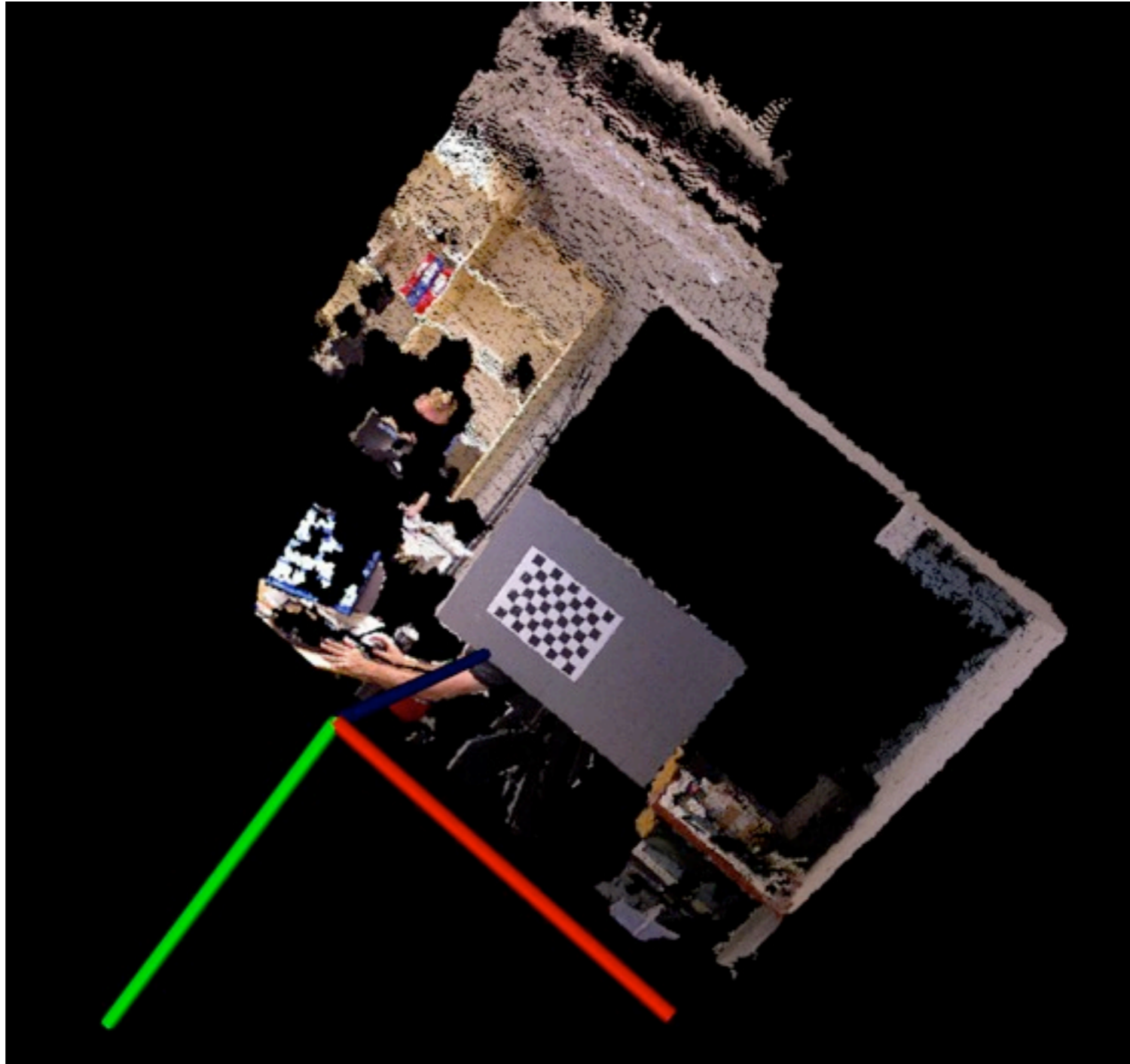


# Résultat

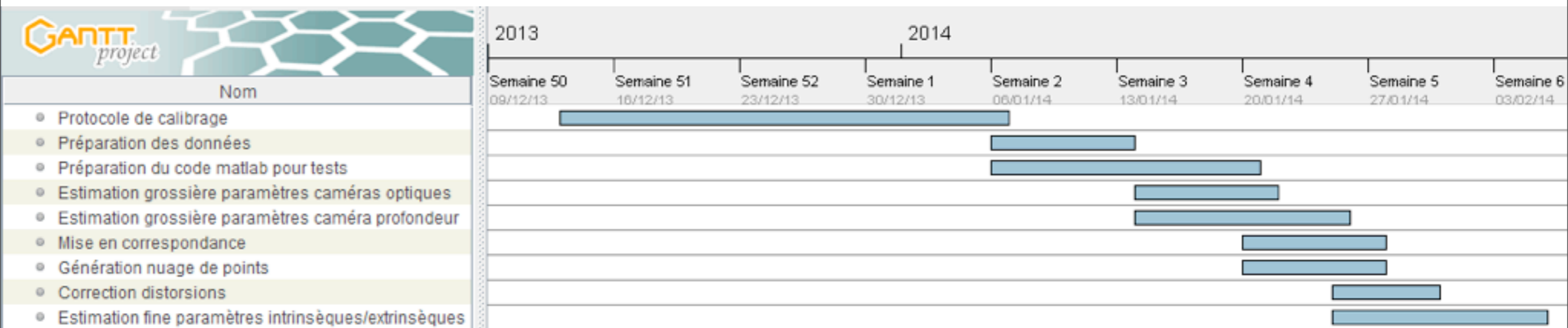


# Vidéo

# Résultat

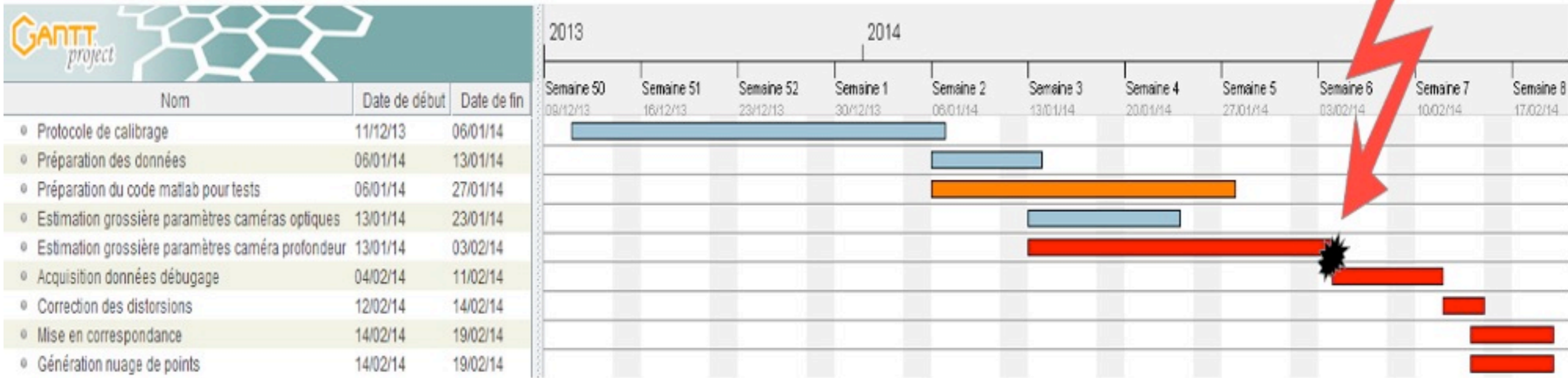


# Planning initial



# Planning effectif

Réunion de crise



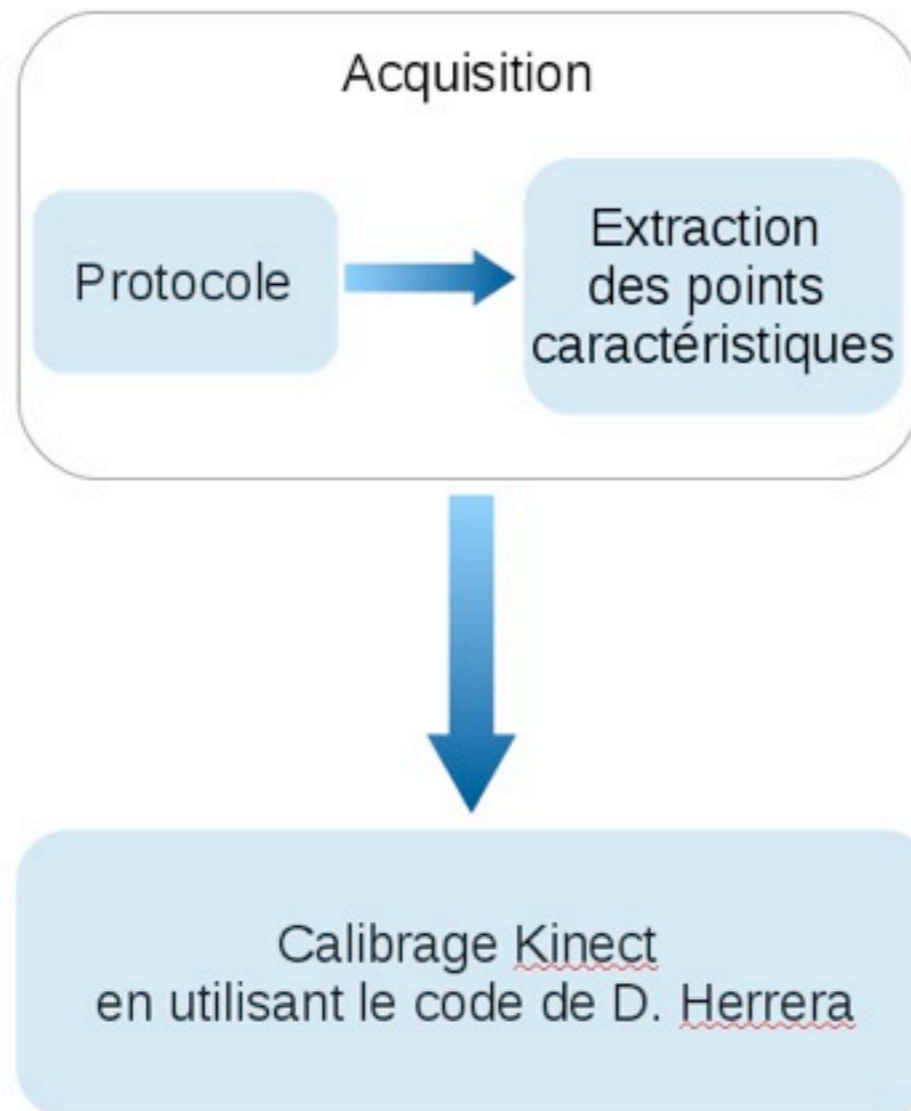
# Retour sur l'analyse de risque

- Analyse de risques pertinente, mais incomplète
- Réaction rapide face aux problèmes

# Bilan

- Pipeline d'applications
- Les livrables
- Expérience acquise pour le projet Kalibrage
- Expérience personnelle

# Pipeline Kinect

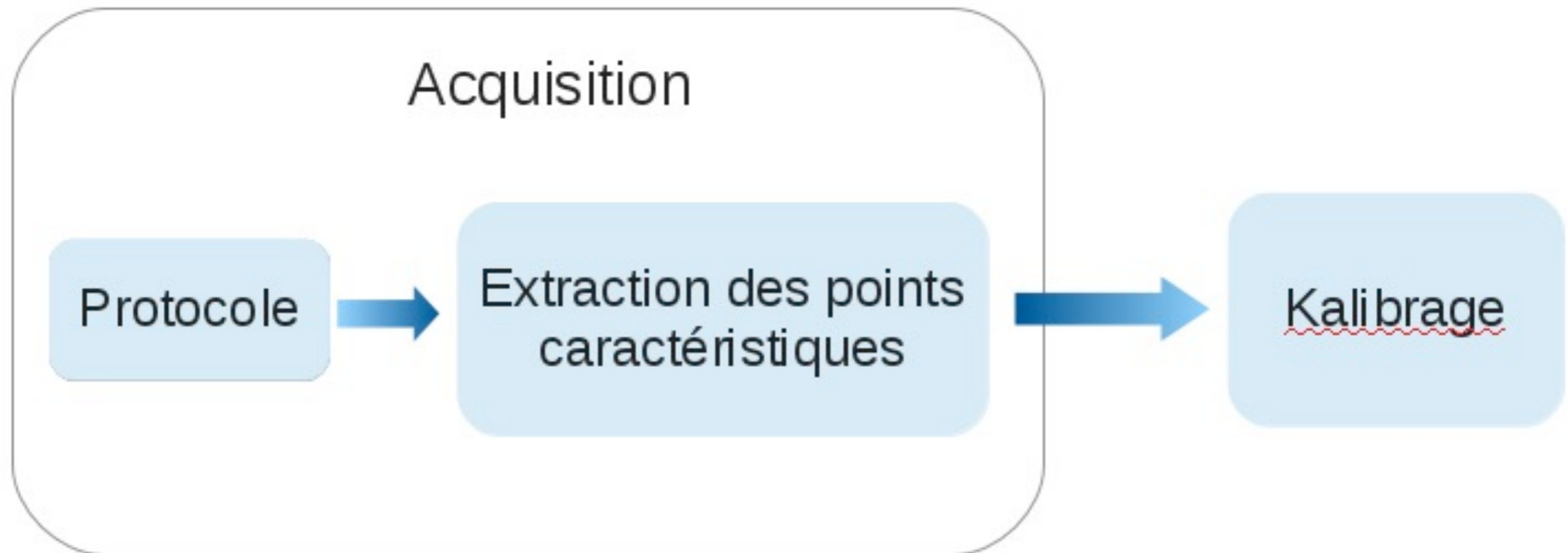


**OU**

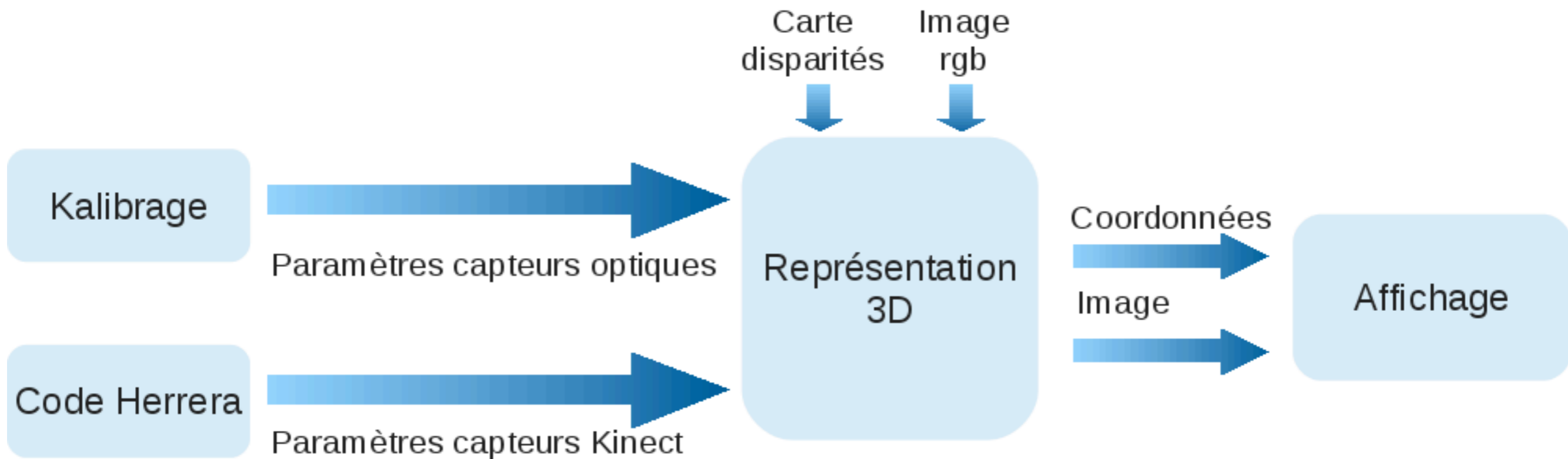
Réutilisation des données  
fournies par D. Herrera



# Pipeline calibrage caméras optiques



# Pipeline Final



# Livrables

- Un protocole d'acquisition au format pdf
- Un programme Matlab d'extraction des points caractéristiques
- Un programme C++ de calibrage des capteurs optiques
- Un programme C++ de reconstruction 3D
- Un programme C++ d'affichage

# Expérience acquise

- Sur le calibrage des 2 caméras optiques
  - 4 orientations de la mire : un peu juste
  - astuces pour faciliter le processus d'extractions des données
- Sur le calibrage d'un capteur de profondeur
  - mise en évidence problèmes liés au peu de données et à leur imprécisions
- Sur le passage Matlab → C++
  - méthodologie de développement
  - sensibilisation aux problèmes d'erreurs

# Acquis personnel

- Matlab
- C++
- Méthodes de calibrage
- Confrontation aux erreurs numériques

# Site Web

- <http://chefdoeuvreclibrage.dyndns.org/>

# Questions ?

