



École des Ponts

ParisTech

Ecole des Ponts Paristech

2015

Rapport de stage scientifique

Leman FENG

Élève ingénieur, 1ère année

Reconstruction 3D par capteur laser et stéréovision

Calibration et estimation de position d'appareil photo

Stage réalisé au sein de **Laboratoire IMAGINE**
École des Ponts ParisTech 6-8, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne 77455 Marne-la-Vallée cedex 2 - France

Maître de stage : M.Pascal MONASSE
Tuteur École : M.Pascal MONASSE

Fiche de Synthèse

Dans le cadre de la gestion informatisée du référencement des rapports de stage, nous vous demandons de renseigner impérativement les items suivants ; ce qui facilitera l'intégration des références dans le catalogue de la bibliothèque de l'École :

- Type de stage : stage scientifique
- Année académique : 2014/2015
- Auteur : Leman FENG
- Formation (1ère année) : Ingénieur
- Titre du rapport : Reconstruction 3D par capteur laser et stéréovision
- Organisme d'accueil : Laboratoire IMAGINE
- Pays d'accueil : France
- Maître de stage : M.Pascal MONASSE
- Tuteur de stage : M.Pascal MONASSE
- Mots-clés caractérisant votre rapport (4 à 5 mots maximum) : stéréovision, étalonnage de caméra, distorsion, calibration, LIDAR
- Thème École : 38 INFORMATIQUE, RECHERCHE OPERATIONNELLE

Remerciements

Au terme de ce travail, je remercie le personnel du Laboratoire IMAGINE dans son ensemble pour m'avoir accueilli dans ses locaux et m'avoir renseigné sur différents points techniques.

Je tiens également à adresser mes remerciements les plus sincères à mon tuteur de stage M.Pascal MONASSE pour son encadrement, son soutien, ses précieux conseils tout au long de la période du stage, son excellent suivi et ses remarques très enrichissantes qui m'ont été d'un appui considérable dans mon travail.

Je remercie mon binôme Laila ZOUAKI pour son assiduité et sa contribution à tester et à comprendre les codes des algorithmes intégrés dans notre logiciel, pour la gentillesse et la patience dont elle a fait preuve pour comprendre mon français et pour ses conseils pertinents et importants à notre logiciel.

Je remercie aussi M. Pierre Moulon, ancien doctorant à IMAGINE et auteur d'une bibliothèque excellente en C++ « OpenMVG » [1], pour sa sollicitude, son intérêt pour notre travail.

Je remercie M.Martin DE LA GORCE pour ses explications techniques de mode d'emploi sur l'utilisation du système LIDAR.

Résumé

La stéréovision est une technique de reconstruction 3D d'une scène à partir de photos prises sous différents angles. Une étape très importante de cette technique est d'estimer précisément la position et l'orientation de caméra correspondant à chaque image. Il faut tout d'abord étalonner l'appareil : (1) corriger les photos afin de compenser la distorsion optique de la lentille ; (2) calculer la matrice intrinsèque de l'appareil photo. Une méthode « harpe » est employée pour (1) et une autre méthode « point de contrôle circulaire » pour (2). Ensuite, deux méthodes de stéréovision : « Incremental » et « Global », sont utilisées pour estimer les positions d'appareil photo. Enfin, cette estimation est comparée avec la vérité terrain. Pour obtenir cette dernière, des damiers sont placés dans la scène en tant que points de contrôle. Puis le système LIDAR scanne la scène par laser et obtient les coordonnées 3D des damiers. Avec les positions 2D des cible dans les photos, la vérité terrain peut être trouvée par la methode « resection ».

Afin de simplifier les opérations des différentes étapes, nous avons développé un logiciel pendant notre stage scientifique. Il comporte différentes fonctionnalités, dont l'étalonnage, la méthode « resection » et l'évaluation d'erreur.

Mots clés : stéréovision, étalonnage de caméra, distorsion, calibration, LIDAR

Abstract

Stereovision is a technique to reconstruct a 3D model of a scene from photos taken from different angles. A very important step of this technique is to estimate the position and orientation of the camera precisely for each image. First, before using those images, you must calibrate the camera : (1) correct photos to compensate the optical lens distortion ; (2) then calculate the intrinsic matrix of camera. One method called "harp" is used for (1) and another "circular checkpoint" method for (2). Secondly, two methods of stereovision : "Incremental" and "Global", are used to estimate the camera positions. Finally, this estimation is compared with the ground truth. To obtain the latter, some targets are put in the scene as control point. Then the LIDAR system use laser to scan the scene and gets the target coordinate in 3D. Together with the target positions in the photos, the camera position can be found by a "resection" method.

To simplify the different stages of operations, a software is developed in this scientific internship. It can do camera calibration, the "resection" method and error evaluation.

Keywords : stereovision, camera calibration, distortion, LIDAR

Table des matières

Title page	1
Fiche de Synthèse	3
Remerciements	5
Résumé	7
Abstract	9
Table des matières	12
Liste des tableaux	13
Liste des figures	15
1 Présentation de l'organisme d'accueil et du maître de stage	1
1.1 Organisme d'accueil	1
1.2 Maître de stage	1
2 Introduction	3
3 Présentation du problème	5
3.1 Etalonnage de caméra	5
3.2 Stéréovision	7
3.3 LIDAR et vérité terrain	8
4 Revue de la littérature sur le sujet	9
4.1 On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery [C. Strecha, W. von Hansen, L. Van Gool, P. Fua, U. Thoennessen]	9
	11

4.2	High-precision camera distortion measurements with a "calibration harp" [Zhongwei Tang, Pascal Monasse]	10
4.3	Camera Matrix Calibration Using Circular Control Points and Separate Correction of the Geometric Distortion Field [Victoria Rudakova, Pascal Monasse]	11
5	Méthode pour aborder le problème	13
5.1	Etalonnage de caméra	13
5.1.1	Correction de distorsion	13
5.1.2	Paramètres internes de caméra	15
5.2	Stéréovision	16
5.3	LIDAR et vérité terrain	17
5.4	Comparaison de résultat	18
6	Résultats	21
6.1	Développement de logiciel	21
6.2	Scène et appareil	22
6.3	Correction de distorsion	23
6.4	Paramètres internes de caméra	24
6.5	Stéréovision	26
6.6	LIDAR et vérité terrain	26
6.7	Comparaison de résultat	27
6.8	Analyse d'erreur	27
7	Retour d'expérience	31
8	Bilan personnel	33
	Bibliography	35
9	Annexe 1 : Mode d'emploi de Calib3DTools	39
10	Annexe 2 : Résultats et format de fichier	43
10.1	Polynôme de distortion	43
10.2	La matrice K	43
10.3	Les points 2D	44
10.4	Les point 3D	45
10.5	Les positions de la vérité terrain	46
10.6	Les positions calculées par la stéréovision	47

Liste des tableaux

6.1	Les paramètres internes de K de la caméra (pixel)	24
6.2	La comparaison des paramètres internes (pixel)	25

Table des figures

3.1	Un sténopé	6
3.2	Trois types de distorsion	6
3.3	L'ajustement de faisceaux	8
5.1	Les étapes pour résoudre le problème	14
5.2	La harpe	14
5.3	La mire de points de contrôle circulaires	16
5.4	Une cible de type damier	17
5.5	La détection du centre de cible dans le nuage de points	18
6.1	Notre logiciel Calib3D Tools	22
6.2	La scène dans une salle	23
6.3	Les courbes détectées	24
6.4	Les ellipses détectées	25
6.5	Zoomer et sélectionner	28
9.1	Les widgets dans Calib3D Tools	40

Présentation de l'organisme d'accueil et du maître de stage

1.1 Organisme d'accueil

IMAGINE (précédemment CERTIS) est un groupe de recherche conjoint entre l'Ecole des Ponts ParisTech et le Centre Scientifique et Technique du Bâtiment. Il y a aussi une partie du laboratoire Université Paris Est Computer Science (l'LIGM, Laboratoire d'informatique Gaspard-Monge). IMAGINE se situe à Paris. Les domaines de IMAGINE sont :

- Vision par ordinateur (en particulier la reconstruction 3D et la compréhension de la scène)
- Apprentissage automatique
- Optimisation

1.2 Maître de stage

Après avoir obtenu son diplôme d'ingénieur à l'Ecole Nationale des Ponts et Chaussées et soutenu sa thèse de mathématiques à l'Université Paris IX, Pascal MONASSE s'est lancé dans la recherche dans le domaine du traitement de l'image et de la vision par ordinateur. C'est d'ailleurs le domaine dans lequel il travaille encore aujourd'hui au sein du laboratoire IMAGINE en tant que chercheur.

Ses travaux dans la vision par ordinateur sont reconnus à l'international dans la communauté scientifique. Avec ses élèves-doctorants, avec le laboratoire ou encore à titre individuel, M. Monasse a remporté de nombreux prix en récompense de ceux-ci.

Introduction

La reconstruction 3D est un processus de capturer une présentation en trois dimensions d'un objet ou d'une scène, et de la sauvegarder sous un format numérique pour la représenter. C'est une technique très utile, qui est appliquée dans de nombreux domaines comme l'infographie, CAO, l'imagerie médicale, la réalité virtuelle, la réalité augmentée, la protection de patrimoine, etc.

Pour reconstruire une scène réelle ou un objet en modèle 3D, plusieurs méthodes sont possibles. On peut employer le LIDAR, c'est-à-dire scanner une scène par laser et puis obtenir la distance de chaque point au laser en mesurant le délai entre l'émission et la réflexion du signal. On connaît alors les coordonnées de l'ensemble des points, qui est appelé « le nuage de points », correspondant aux points de réflexion dans la scène. En se basant sur le nuage de points, le modèle numérique peut être créé.

La méthode LIDAR est vraiment précise. Mais le défaut est que l'appareil de laser est lourd, coûteux en temps et onéreux. Par exemple, l'appareil que nous utilisons pendant le stage coûte 50,000 euros. De plus, le temps qu'il prend pour scanner une salle est 10 à 30 min, durée qui varie selon la précision souhaitée.

Heureusement, il existe d'autres options. Pourquoi l'humain peut voir en 3D ? Parce que nous possédons deux yeux, et non pas qu'un. Chaque œil a une vue légèrement différente de l'autre. Le cerveau analyse les deux images, détecte les points correspondants parmi les deux images, et donne une perception de profondeur grâce aux différences de positions des points correspondant. C'est la stéréovision.

En suivant la même idée, on peut prendre deux ou plusieurs photos sous différents angles. Il existe beaucoup d'algorithmes pour résoudre ce problème et trouver un nuage de points. La stéréovision est beaucoup plus pratique et

facile que le LIDAR. Mais nous devons nous demander si la stéréovision est assez précise pour remplacer le LIDAR.

Donc, le sujet de notre stage est d'évaluer la précision de la stéréovision. Nous évaluons la position de l'appareil photo de chaque image car le nuage de point est engendré dans le repère de l'appareil. Mais la position de prise de photo est souvent inconnue, et est aussi difficile à mesurer. Donc on compte sur l'algorithme pour trouver les positions de l'appareil. La précision de la position de l'appareil influence directement la précision du nuage de points.

Levons tout de suite une impasse qui n'en est pas une : comment évaluer la précision de la position estimée par l'algorithme si on ne connaît pas la position réelle ? En fait, c'est l'utilisation du LIDAR qui va nous permettre cette comparaison. Sur la scène que le laser scanne, des damiers ont été placés. Le scan du LIDAR donne avec précision leur position dans l'espace, et nous pouvons alors utiliser un algorithme qui peut estimer la position de l'appareil dans chaque image avec une grande précision, qui servira ensuite de référence ou de vérité terrain.

Présentation du problème

Le problème est d'évaluer la précision de la position de caméra trouvée par la stéréovision. Il y a principalement trois étapes : l'étalonnage de caméra, la stéréovision et l'acquisition de la vérité terrain.

3.1 Etalonnage de caméra

Un appareil photo ou une caméra peut être considéré comme un instrument qui définit une application d'un espace à trois dimensions (scène) à un plan à deux dimensions (photo). Dans le cas idéal, un appareil photo est un sténopé (figure 3.1). Donc on peut exprimer cette application mathématiquement (3.1).

$$\lambda m = \overbrace{K}^{\text{Intrinsic Matrix}} \times \overbrace{[R \mid \mathbf{t}]}^{\text{Extrinsic Matrix}} \times M$$

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \overbrace{\begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}^{\text{Intrinsic Matrix}} \times \overbrace{\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & | & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & | & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & | & t_3 \end{pmatrix}}^{\text{Extrinsic Matrix}} \times \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.1)$$

f_x, f_y : la focale

(x_0, y_0) : le point principal

λ : une scalaire

R : une matrice de rotation

\mathbf{t} : l'origine du repère global

En considérant un point dont la coordonnée globale est M , on doit d'abord connaître la position et l'orientation de l'appareil pour en connaître l'image

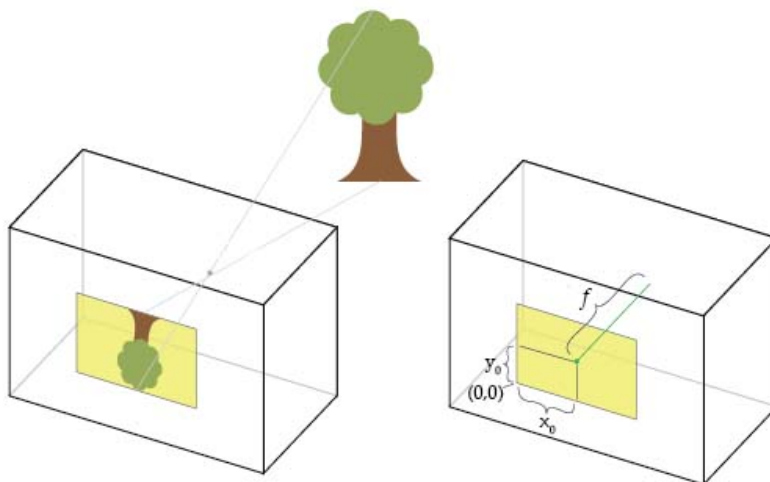


FIGURE 3.1: *Un sténopé*

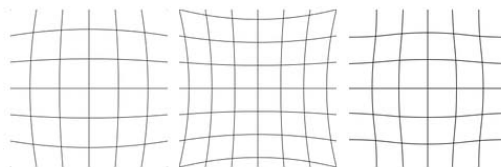


FIGURE 3.2: *Trois types de distorsion*

m dans la photographie. La position et l'orientation de la caméra sont représentées par la matrice extrinsèque $[R \mid \mathbf{t}]$. Les coordonnées de ce point sont passées dans le repère local de l'appareil par $[R \mid \mathbf{t}] \times M$. Ensuite, il faut connaître les paramètres internes de l'appareil, qui sont représentés par la matrice intrinsèque K . La matrice K contient la focale utilisée ainsi que le point principal qui correspond au centre de l'optique de lentille.

En pratique, un appareil photo n'est pas aussi simple qu'un sténopé parce qu'il est équipé d'une lentille au lieu d'un trou. La lentille engendre souvent une distorsion sur la photo. Dans le cas idéal, les images de toutes les lignes droites dans l'espace restent toujours droites dans la photo. A cause de la distorsion, les lignes sont parfois courbées dans la photo (figure 3.2). Cette distorsion optique peut être visible à l'œil nu, il faut donc la corriger. De plus, les paramètres internes, ou la matrice K , sont aussi inconnus. L'appareil photo peut enregistrer la focale utilisée de chaque photo. D'ailleurs, la focale est indiquée sur la lentille. Mais ces données sont imprécises. En effet, la taille du capteur est de 5 mm–30 mm et la précision de focale est de 1 mm, ce qui est trop élevé par rapport à la précision recherchée. Il faut alors trouver une

autre méthode pour estimer K .

3.2 Stéréovision

La reconstruction 3D à partir des photos prises sous différents angles de vue peut être considérée comme un processus inverse de prise de photo sous différents angles d'une scène 3D. Ce n'est pas simple parce que quand on prend la photo, on réalise une projection de l'espace au plan. Une dimension est donc perdue : la profondeur. Pour retrouver la profondeur, il faut utiliser plusieurs photos, chercher les points communs entre eux et les analyser.

Il faut donc chercher les points communs aux différentes vues de la même scène. Cela revient à résoudre deux problèmes : le choix des points à chercher, et la correspondance de ces points sur les différentes photos. En fait, la deuxième étape a créé une restriction sur le choix de point dans la première. Théoriquement, chaque pixel dans une photo correspond à un point dans l'espace, donc son image projetée dans le plan peut aussi être trouvée comme un pixel dans une autre photo. Mais la correspondance entre les deux pixels ne sont souvent pas identifiable. Par exemple, on peut imaginer que deux photos sont prises devant un mur blanc. Les deux photos sont probablement complètement blanches donc il n'y a aucune information permettant de réaliser une correspondance entre les deux car le mur manque de détails.

Donc, on doit chercher dans la photo les points qui sont identifiables dans une autre. On les appelle points d'intérêt. L'algorithme pour trouver le point d'intérêt est appelé « le détecteur ». De plus, comme demandé dans la deuxième étape, on a aussi « le descripteur » qui décrit les caractéristiques d'un point d'intérêt. Les caractéristiques nous permettent de trouver la correspondance des points entre les photos. Il est crucial que les caractéristiques soient indépendantes de toute rotation et de l'échelle de l'image.

Après avoir trouvé les points d'intérêt et les correspondances entre photos, avec les paramètres intrinsèques dans K , on peut calculer les positions/orientations de caméra et les coordonnées des points d'intérêt dans 3D. Ce processus est appelé « l'ajustement de faisceaux » (figure 3.3). C'est la dernière étape commune à toutes les méthodes de reconstruction 3D. C'est un problème d'optimisation. L'algorithme propose les positions de caméra et les positions des points 3D, selon la relation 3.1, avec le K fourni, il peut projeter les points 3D sur les photos. Il minimise alors la différence entre la position réelle du point d'intérêt et la projection. Ce qu'on va évaluer dans ce rapport est les positions de caméra calculées par différentes méthodes de stéréovision.

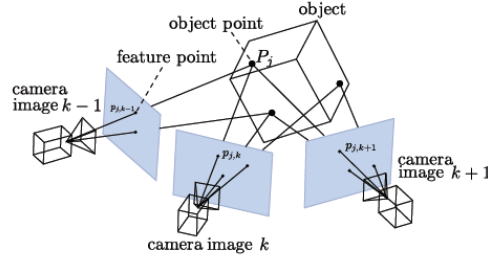


FIGURE 3.3: L'ajustement de faisceaux

3.3 LIDAR et vérité terrain

Afin d'évaluer les positions de caméra calculées, il faut connaître la vérité terrain des positions. Le problème est qu'il est difficile de connaître les positions exactes de la caméra lorsqu'on prend la photo. En fait, la position dont on a besoin est la position de la focale. Même si l'on peut estimer la position de la caméra, la structure interne de caméra est cachée à l'intérieur. Donc une autre méthode est adoptée.

A l'aide de LIDAR, la scène est scannée et un nuage de points de haute précision est engendré. Ainsi, on peut obtenir la position exacte de chaque point de l'objet ou de la scène. Mais LIDAR ne peut pas trouver les position de la caméra de cette façon. En effet, (1) le laser ne pénètre pas dans la caméra (2) le point focale est un point imaginé (3) la caméra doit prendre plusieurs photo. Mais elle ne peut pas apparaître aux différentes positions en même temps pour attendre le scan. Par conséquent, il nous force à chercher une méthode indirecte.

L'idée est que, les positions exactes des points sur objet sont connues. Et en même temps, les points sur objet sont capturés dans les photos. C'est-à-dire dans la relation 3.1, le m et M sont connu. D'ailleurs, K est obtenu par l'étalonnage. Donc s'il y a plusieurs paires de (m, M) , ou pair des coordonnées d'un point 3D et sa projection en 2D, la matrice extrinsèque $[R | \mathbf{t}]$ peut être trouvée.

Revue de la littérature sur le sujet

4.1 On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery [C. Strecha, W. von Hansen, L. Van Gool, P. Fua, U. Thoennessen]

L'idée principale de notre stage vient de cet article. Strecha et son groupe veut savoir si la technique de reconstruction 3D basée sur la stéréovision passive peut remplacer le système LIDAR pour des scènes à l'extérieur. Ils ont comparé les positions de caméra calculées par la stéréovision à celle de la vérité terrain acquise par le LIDAR.

Ils ont réalisé des tests à l'extérieur. La taille de scène est de l'ordre d'une dizaine mètre. D'abord, Ils ont placé des cibles de type damier dans la scène. Ensuite le LIDAR scanne la scène pour acquérir les positions 3D des cibles. Les cibles sont aussi capturées dans les photos. De plus, la distorsion de lentille est corrigée selon un modèle de distorsion radiale et la matrice K est calculée par une méthode expliquée dans l'article. Puis, les positions 2D de cible sont acquises. Enfin, les positions des caméra peuvent être calculées, et serviront de vérité terrain pour évaluer la stéréovision.

Ce qui est remarquable, c'est qu'ils ont analysé l'erreur induite par cette méthode. Deux sources d'erreur sont prises en compte : la précision de mesure par laser et l'étalonnage (distorsion et K). La propagation d'erreur est calculée. La précision finale de la vérité terrain est de 5 mm–20 mm. Pour une scène de dizaine mètre, cette erreur est faible et confirme que cette vérité terrain peut être considérée comme une référence.

Du côté de la stéréovision, trois groupes de photos sont testés. Les photos du premier groupe ne sont pas étalonnées, C'est-à-dire qu'on ne corrige pas la

distorsion et que les paramètres intrinsèque restent inconnus. Le deuxième groupe est étalonné. C'est ce groupe qu'ils souhaitent comparer avec LIDAR. Pour le troisième groupe, tous les paramètres intrinsèques et extrinsèques de la caméra sont connus.

Les erreurs de position de caméra varient selon les scènes et les algorithmes adoptés. Pour une scène donnée, la précision est la même que celle de la vérité terrain. Pour les autres, on peut constater une erreur dix fois plus grande. Le nuage de points calculé par la stéréovision est aussi comparé avec le scan de LIDAR. Ils concluent qu'il y a toujours un écart entre l'acquisition LIDAR et la stéréovision.

4.2 High-precision camera distortion measurements with a "calibration harp" [Zhongwei Tang, Pascal Monasse]

Cet article nous présente une nouvelle méthode pour mesurer et corriger la distorsion de lentille à partir de photos d'un motif particulier. Traditionnellement, ce motif est placé sur une surface plate. Mais la planéité de cette surface doit être assurée avec précision, ce qui est souvent difficile. Donc, les auteurs de l'article ont du chercher une alternative. Au lieu de créer une surface plate, ils considèrent que c'est simple de créer des lignes droites en tendant des fils de bonne qualité.

L'instrument qu'ils ont créé est appelé « harpe », parce que sa forme ressemble à une harpe avec un cadre rectangle. C'est un outil simple à fabriquer, et peu coûteux. Ensuite, la harpe est prise en photo selon différentes directions avec la même focale. Les photos sont traitées par un algorithme et les lignes sont détectées. En analysant les lignes courbées par la distorsion, un polynôme est construit pour définir une application entre deux images pour réaliser la correction.

Il s'avère que la précision de cette nouvelle méthode « harpe » est de 0.08 pixel. Ce résultat est meilleur que les autres méthodes comme la grille. De plus, le traitement d'image est réalisé automatiquement par algorithme, alors le temps qu'il prend est négligeable.

4.3 Camera Matrix Calibration Using Circular Control Points and Separate Correction of the Geometric Distortion Field [Victoria Rudakova, Pascal Monasse]

Cet article décrit une méthode d'estimation des paramètres internes d'une caméra à partir de photos de points de contrôle circulaires. Cette méthode est plus stable que les autres de l'état de l'art.

Une mire de point de contrôle circulaire est dessinée sur une surface de haute planéité. Puis cette mire est prise en photo sous différents angles. Avant d'utiliser les photos, la distorsion est analysée et corrigée par la méthode « harpe ». Ensuite les centres des points circulaires sont détectés par un algorithme. L'image d'un cercle dans la photo est une ellipse. Mais l'image du centre d'un cercle ne coïncide pas avec le centre d'ellipse. Donc une transformation affine conique est introduite pour résoudre cette disparité.

Quand les coordonnées 2D de points de contrôle sont détectées, il reste à résoudre un problème d'homographie. Les relations des points dans les différentes photos peuvent être exprimées par une matrice d'homographie. Enfin, les paramètres internes de caméra sont extraits de cette matrice.

Méthode pour aborder le problème

5.1 Etalonnage de caméra

5.1.1 Correction de distorsion

La méthode adoptée pour la correction de distorsion est la « harpe ». La « harpe » est un instrument simple constitué de fils tendus et fixés sur un cadre rectangle en bois (figure 5.2). Les fils sont quasiment parallèles et équidistants. Théoriquement, ce n'est pas nécessaire de placer les cordes de cette façon, ni qu'elles soient coplanaires. En fait, La seule condition à satisfaire est la rectitude des cordes, car la propriété de la projection de sténopé assure que L'image d'une ligne droite en 3D est une ligne droite en 2D.

Nous prenons les photos de « harpe » selon différentes directions en tournant la caméra. Comme les images des cordes vont être traitées par l'algorithme, nous voulons diminuer le bruit et éviter les autres objets parasites. Pour masquer les objets derrière les cordes, un papier semi-translucide est collé sur le cadre. De plus, le côté de papier est tourné vers le soleil pour éviter que l'ombres des cordes ne soit projetée sur le papier.

Les images des cordes sont courbées par la lentille. Cette déformation est plus grave au bord de la photo qu'au centre. L'algorithme va détecter les lignes courbées dans toutes les photos prises avec la même focale car la distorsion reste la même pour une focale fixée. Concrètement, grâce à l'algorithme Devernay, le gradient d'intensité de lumière est calculé pour chaque pixel. Les différences de couleur de papier blanc et de corde noir créent un gradient de valeur élevée qui réussit à décrire correctement le bord de corde. L'algorithme Devernay peut donner une précision de 0.05 pixel [2].

Ensuite les lignes courbées sont décrites par une série de points. Le but est de chercher une application de 2D à 2D pour rendre les points alignés

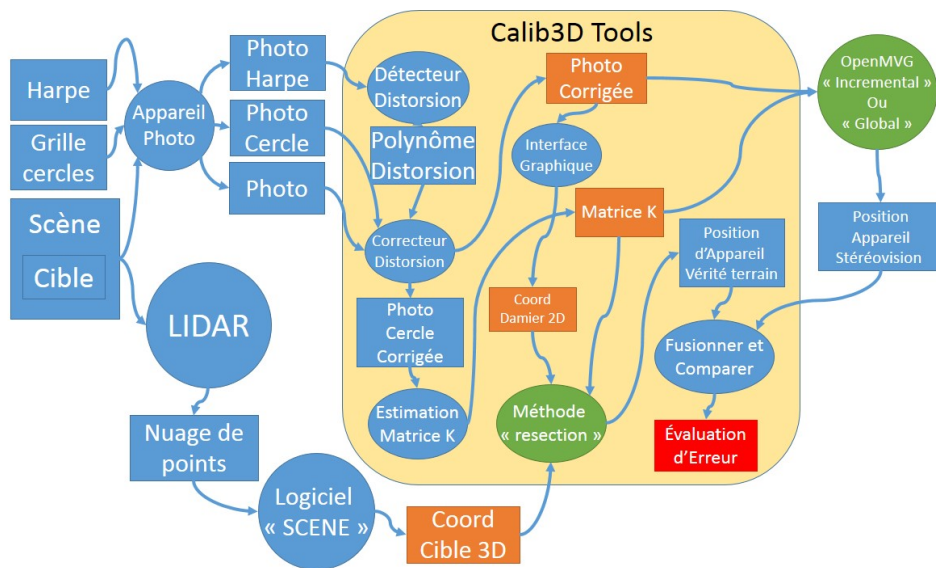


FIGURE 5.1: Les étapes pour résoudre le problème



FIGURE 5.2: La harpe

sur une ligne droite. Il y a plusieurs façons d'exprimer cette application mathématiquement. Ici, nous utilisons deux polynômes :

$$\begin{aligned}
 x_u &= b_0 x_d^p + b_1 x_d^{p-1} + b_2 x_d^{p-2} y_d^2 + \dots + b_p y_d^p \\
 &\quad + b_{p+1} x_d^{p-1} + b_{p+2} x_d^{p-2} y_d + \dots + b_{2p} y_d^{p-1} \\
 &\quad + \dots \\
 &\quad + b_{\frac{(p+1)(p+2)}{2}-3} x_d + b_{\frac{(p+1)(p+2)}{2}-2} y_d + b_{\frac{(p+1)(p+2)}{2}-1} \\
 y_u &= c_0 x_d^q + c_1 x_d^{q-1} + c_2 x_d^{q-2} y_d^2 + \dots + c_q y_d^q \\
 &\quad + c_{q+1} x_d^{q-1} + c_{q+2} x_d^{q-2} y_d + \dots + c_{2q} y_d^{q-1} \\
 &\quad + \dots \\
 &\quad + c_{\frac{(q+1)(q+2)}{2}-3} x_d + c_{\frac{(q+1)(q+2)}{2}-2} y_d + c_{\frac{(q+1)(q+2)}{2}-1}
 \end{aligned} \tag{5.1}$$

(x_d, y_d) est le point original et (x_u, y_u) est le point après la correction. p est l'ordre de polynôme.

Quand le polynôme est calculé, il peut s'appliquer aux photos de même focale et de même caméra. Nous allons l'utiliser pour corriger la photo de la scène et la photo de points de contrôle circulaire.

5.1.2 Paramètres internes de caméra

Les paramètres internes de caméra, ou la matrice K , sont demandé par la stéréovision et aussi le calcul de la vérité terrain. C'est difficile de mesurer les paramètres directement. Donc une mire de point de contrôle circulaire est utilisée (figure 5.3). Il y a $12 * 15 = 180$ points circulaire. Le rayon de point est un tiers de la distance entre les points. L'algorithme est indépendant de la taille absolue de cette mire. Les photos de cette mire sont d'abord corrigées par la méthode « harpe ». Un point est transformé en une ellipse dans la photo. La différence entre le centre d'ellipse et le centre du point est traitée. Les centres des points dans toutes les photos sont liés aux centres des mires par une homographie représentée par une matrice notée H .

La matrice H est liée à la matrice K . La matrice K est la même pour toutes les photos. En donnant différents H de différentes photo, la matrice K peut être trouvée.

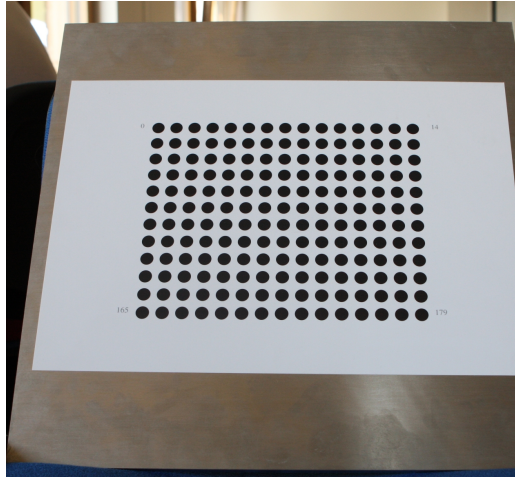


FIGURE 5.3: La mire de points de contrôle circulaires

5.2 Stéréovision

Après l'étalonnage, les photos de la scène corrigées et la matrice K sont utilisés par l'algorithme de stéréovision.

Tous les algorithmes utilisés dans la stéréovision sont réalisés par une bibliothèque de C++ « OpenMVG » [1].

Un détecteur et descripteur de points d'intérêt appelé « SIFT » est utilisé. C'est un algorithme protégé par brevet aux États-Unis. SIFT va chercher les points d'intérêt dans toutes les photos. Ensuite les points sont comparés entre les photos pour chercher les correspondances.

Pour faire « l'ajustement de faisceaux », deux méthodes de stéréovision sont testées : « IncrementalSfM » [3] et « GlobalSfM » [4].

« IncrementalSfM » commence par une paire de photo initiale. Il fait la triangulation sur la correspondance des points dans cette paire pour résoudre un problème de « l'ajustement de faisceaux » de deux photos. C'est une reconstruction 3D à partir de la paire initiale. Puis il ajoute les autres photos une par une et répète la triangulation d'où le nom « Incremental ». Enfin il donne un nuage de points et les positions de la caméra.

Mais, un défaut de « Incremental » est que la qualité du résultat final dépend du choix de la paire initiale et aussi l'ordre d'ajout des photos. Normalement on préfère la paire avec le plus de correspondances de points. Mais ce n'est pas toujours le meilleur choix. Malheureusement, il n'y a pas une stratégie

optimale pour choisir cette paire initiale. C'est Pour cette raison qu'une autre méthode « Global » a été mise en place.

« Global » sépare l'estimation de la rotation et de la position de caméra en deux étapes. Dans la première étape, pour toutes les paires de photos qui ont une correspondance assez forte entre elles, la rotation relative entre deux caméras peut être calculée par triangulation. Puis la rotation globale de chaque caméra peut être trouvée. « Global » cherche à contrôler l'erreur de clôture. Par exemple si les rotations relatives entre photo 1 et 2, 2 et 3, 3 et 4 sont connues, et qu'on fixe la rotation de photo 1, on peut calculer la rotation de photo 2, 3 et 4 successivement. Mais l'erreur va s'accumuler. Si la rotation relative entre 1 et 4 est aussi donnée, on peut probablement constater un écart entre le résultat expérimental et la rotation réelle?. Donc « Global » va ré-ajuster les rotations pour distribuer uniformément les erreurs parmi toutes les caméras.

On peut donc s'attendre à obtenir un meilleur résultat par « Global » que « Incremental ».

5.3 LIDAR et vérité terrain

Comme c'est difficile de mesurer directement la position de caméra, nous avons utilisé une méthode indirecte « resection ». Cette méthode est aussi utilisée par le groupe de Strecha [5].

Il faut placer des cibles identifiables par caméra et aussi par laser dans la scène. Nous utilisons les damiers et les collons sur des surface planes (figure 5.4). Le LIDAR scanne la scène et donne un nuage de points avec l'information de l'intensité de réflexion. Le nuage de points est ensuite importé dans un logiciel « SCENE » [6]. « SCENE » peut détecter le centre de chaque damier dans le nuage de point (figure 5.5). Les coordonnées sont exprimées dans le repère du LIDAR. Nous utilisons ce repère comme référentiel.

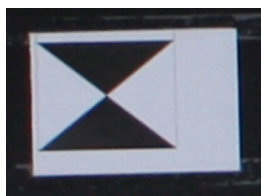


FIGURE 5.4: Une cible de type damier

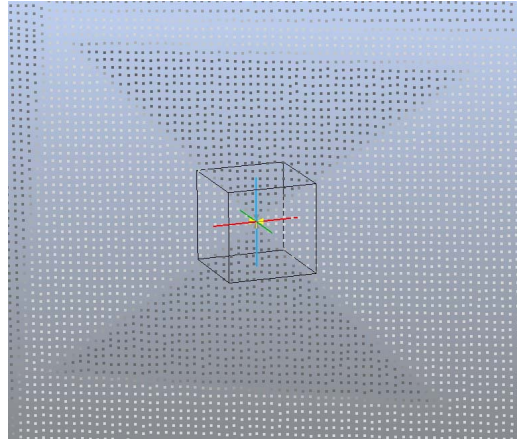


FIGURE 5.5: *La détection du centre de cible dans le nuage de points*

D'autre part, les cibles sont aussi dans les photos. Pour trouver les coordonnées 2D des cibles, nous avons créé une interface graphique qui permet de cliquer sur l'image pour indiquer le centre de cible précisément (figure 6.1).

Une fois que les coordonnées 3D des cibles dans l'espace et les coordonnées 2D dans la photo sont connues, avec la matrice K , on peut estimer la position et l'orientation de la caméra de cette photo. Ce problème est appelé « PnP ». Le n est le nombre de points dont les coordonnées 2D-3D sont connues. Notons que lorsque n est inférieur à 6, il est possible d'obtenir plusieurs solutions au problème. En effet, nous avons rencontré le problème des solutions multiples quand nous avons étudié le rapport d'un stage de l'année précédente écrit par Maryam MARIH. Dans ce rapport, une déviation de 18 cm est observée à cause du choix d'une solution incorrecte.

L'algorithme qu'on utilise est « P3P intrinsèque », accompagné de « AC-Ransac » [7], un algorithme d'estimation robuste créé par IMAGINE. Nous les avons intégrés dans notre logiciel.

5.4 Comparaison de résultat

Les positions de la vérité terrain sont données dans le repère du laser, ou le repère global, dont l'unité est le mètre. Les positions données par la stéréovision sont exprimées dans un repère créé par l'algorithme. Comme tous les calculs de stéréovision utilisent le pixel comme unité, les deux repères ne sont pas dans la même échelle. Pour fusionner deux groupes de points, il faut d'abord unifier l'unité, et ensuite traduire et tourner les points.

L'algorithme qu'on utilise est « Umeyama », qui est disponible dans Eigen, une bibliothèque C++ de mathématique linéaire.

Résultats

6.1 Développement de logiciel

La tâche principale de notre stage est de développer un logiciel avec une interface graphique pour combiner les algorithmes et les fonctionnalités utilisés fréquemment dans la stéréovision. Ce logiciel nous a aidé à trouver le résultat de nos expériences. De plus, il va par la suite être utile aux chercheurs pour faire des calculs. Notre logiciel, Calib3DTools [8] est basé sur Qt, une API orientée objet et développée en C++ qui offre des composants d'interface graphique, d'accès aux données, de gestion des fils d'exécution, etc. Il permet la portabilité des applications dans les environnements comme Linux, Windows et Mac OS. Notre logiciel a aussi été testé sous les trois systèmes.

De plus, dans ce logiciel, les algorithmes scientifiques sont donnés par M. Monasse. Ils sont tous inventés et écrits par les chercheurs du laboratoire IMAGINE, comme la méthode « harpe » et la méthode « point contrôle circulaire ». Ce logiciel utilise aussi une partie de code issu de la bibliothèque OpenMVG [1]. La lecture et l'essai des codes nous ont pris beaucoup de temps. Cela a été une bonne opportunité de comprendre profondément les techniques utilisées dans le domaine de vision par ordinateur. Et en même temps, nous avons trouvé certains bogues dans les codes existants. Nous citons ci-dessous quelques bogues graves pour information. Ils sont maintenant tous réparés.

1. L'équation de jacobian utilisé pour résoudre le problème d'homographie est incorrecte. Cela influence le raffinement de solution. Elle a ensuite été remplacée par une équation correcte.
2. Dans la classe de l'algorithme Levenberg–Marquardt, la fonction « mi-

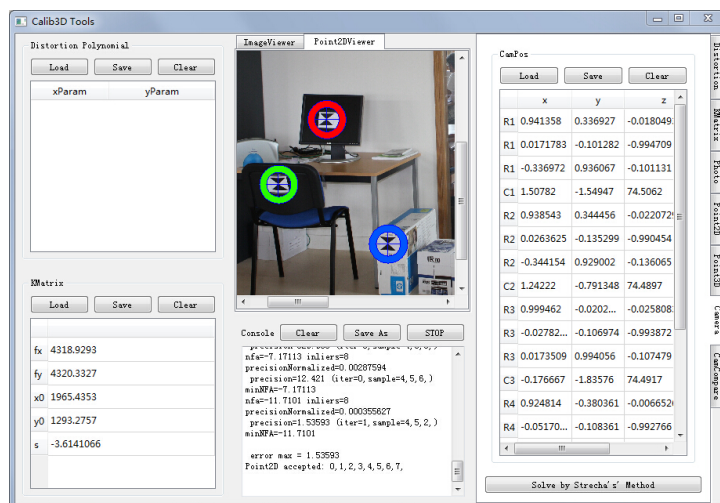


FIGURE 6.1: Notre logiciel Calib3D Tools

nimize » calcule la valeur de l'erreur de manière erronée. La valeur de retour est incorrecte et le nombre d'itération est changé.

3. Dans le code de détection d'ellipse, un cercle très régulier peut être refusé par l'algorithme à cause d'une condition sur le ratio de compacité.

Certains codes existants sont en format de langage C, par exemple la bibliothèque d'image. Nous les avons transformés en C++ pour diminuer le risque de fuite de mémoire et ajouter le mécanisme d'exception car le logiciel doit être stable. En effet, une grande différence entre un logiciel et un morceau de code utilisable est qu'un logiciel ne peut pas crasher devant un utilisateur quand il n'y a pas assez de mémoire ou quand l'utilisateur donne une mauvaise instruction.

Nous allons présenter le logiciel avec les méthodes suivant.

6.2 Scène et appareil

La scène que nous avons choisie est une petite salle de réunion (figure 6.2). La taille de la scène et la distance entre la caméra et les objets sont d'environ 3 à 4 mètres. La caméra utilisée est Canon 40D, qui possède une résolution de 3888 * 2592 pixels. La focale de la lentille est toujours réglée à 24mm. C'est la focale minimale de la lentille qui peut produire une distorsion évidente.



FIGURE 6.2: *La scène dans une salle*

L'appareil LIDAR utilisé est FARO Laser Scanner Focus3D X 130. Sa précision de mesure est 2mm. La densité de scan qu'on choisit est représentée par un écart de 7 mm entre deux points scannés à une distance de 10 m.

6.3 Correction de distorsion

Nous avons pris 25 photos de « harpe ». La distorsion est calculée à partir de ces photos et est représentée par deux polynômes d'ordre 11. Pour vérifier la rectitude des lignes corrigées, les 25 photos de « harpe » sont corrigées par ce polynôme. Dans les photos sans distorsion, les lignes sont ensuite détectées et représentées par une série de points. L'algorithme mesure la distance entre chaque point et une droite de régression. Cette distance est l'erreur résiduelle.

Dans notre logiciel, nous chargeons les photos et puis lançons le calcul. Les points rouges et les lignes vertes indiquent les courbes détectées.

La distance maximum est 0.7 pixel. L'écart de racine carré moyenne (RMSE) est 0.07 pixel.

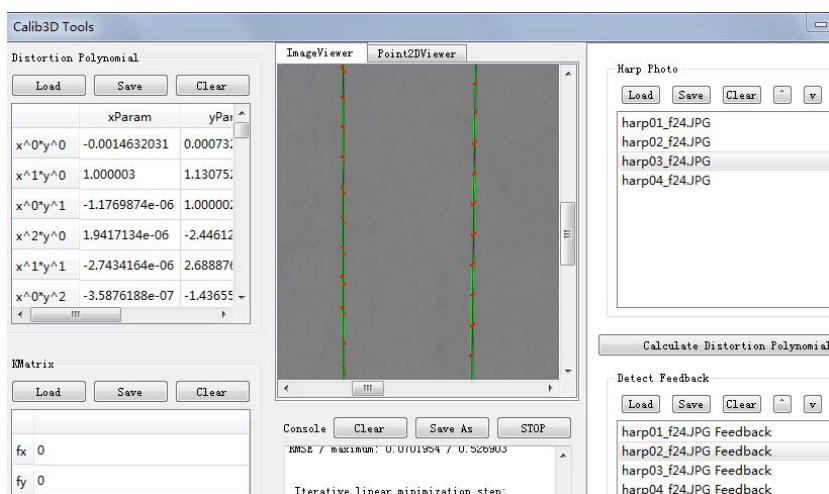


FIGURE 6.3: Les courbes détectées

6.4 Paramètres internes de caméra

Ensuite, 14 photos de mire sont corrigées pour supprimer la distorsion. Les points de contrôle circulaires dans les photos sont détectés. Quand la matrice d'homographie H est trouvée pour chaque photo, on peut connaître les positions théoriques des points et les comparer ensuite avec les points observés dans les photos.

La RMSE de la déviation entre les deux groupes de points est 0.12-0.3 pixel. Cette erreur est normale. En effet, l'algorithme [9] s'avère très précis. Il donne une précision de 0.01 pixel quand on le teste avec les images synthétiques.

Les cinq paramètres de la matrice K calculés sont renseignés dans la tableau 6.1 :

Nous pouvons constater que le point principal est (1964, 1295). Le centre

	K
f_x	4318.32
f_y	4319.50
x_0	1963.92
y_0	1294.52
s	-3.79

TABLE 6.1: Les paramètres internes de K de la caméra (pixel)

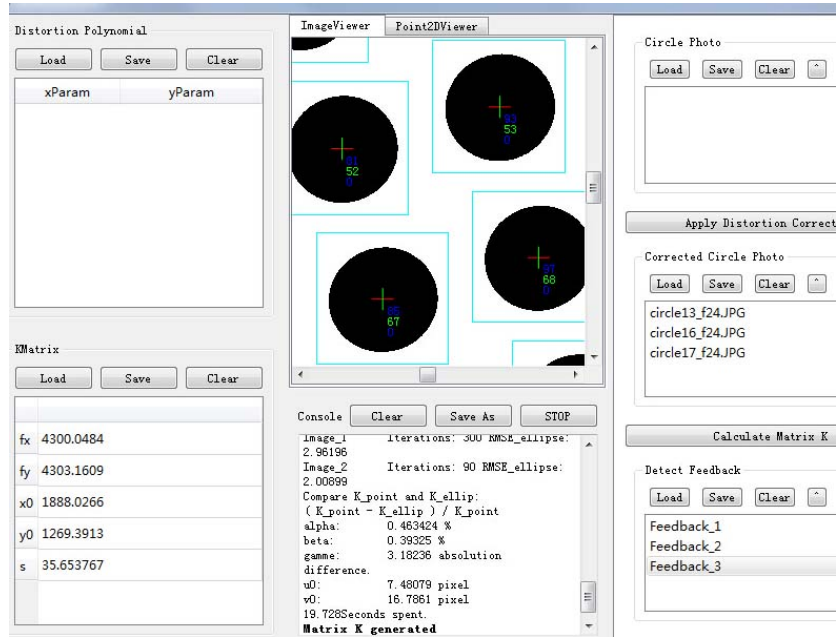


FIGURE 6.4: Les ellipses détectées

	K_1	K_2
f_x	4315.37	4315.33
f_y	4316.76	4315.59
x_0	1964.47	1965.16
y_0	1294.74	1294.65
s	-3.82	-3.94

TABLE 6.2: La comparaison des paramètres internes (pixel)

de photo est (1943, 1295). Il y a un écart de 20 pixels. Nous voulons savoir la raison pour cet écart. Donc deux matrices K_1 , K_2 sont calculées selon respectivement les photos 1-7 et 8-14 (tableau 6.2) :

On constate que les coordonnées du point principal calculées sont presque identiques. Cela montre que l'écart entre le point principal et le centre de la photo est probablement induit par les erreurs de montage. D'ailleurs, la focale varie selon les différents choix de photos. Ce problème sera discuté dans la section « analyse d'erreur ».

6.5 Stéréovision

Nous avons pris 5 photos de la scène. Nous utilisons le logiciel? « SfM : Structure-from-Motion » dans « OpenMVG ». Nous suivons les indications et lançons les quatres étapes successivement :

1. Image listing
Nous donnons les 5 photos corrigées de la distorsion et la matrice K .
2. Image description computation
Le logiciel détecte les points d'intérêt.
3. Corresponding images and correspondences computation
Le logiciel cherche les correspondances des points d'intérêt entre les photos.
4. SfM solving (2 méthodes)
Dans les deux algorithmes testés, nous avons désactivé deux fonctionnalités optionnelles. L'un est la correction de distorsion automatique parce que les photos sont déjà corrigées précisément. L'autre est le raffinement de la matrice K .
 - (a) La méthode « Incremental SfM » :
Il faut choisir une paire initiale de photos. Nous avons choisi la paire $(0, 1)$, qui a le plus de points de correspondance.
 - (b) La méthode « Global SfM » :
Dans cet algorithme, nous avons deux choix : calculer une matrice E (appelée matrice essentielle) en utilisant le repère local de la caméra dont l'unité est le mètre, ou calculer une matrice F (dite fondamentale) en utilisant le repère associé à l'image dont l'unité est le pixel. Cette dernière est plus simple à calculer. Mais la matrice K est demandée par F pour trouver E . Mathématiquement, on a $E = K^\top F K$.

6.6 LIDAR et vérité terrain

Pour éviter la situation de solutions multiples, il y a au total 8 cibles placées dans la scène. Mais après avoir pris toutes les photos, nous nous sommes rendu compte que la cible sur la chaise est invalide car la surface de la chaise est courbée. La représentation de cette cible dans le nuage de point est tordue, ce qui va induire une erreur supplémentaire dans les coordonnées 3D de ce

point. Donc, nous avons abandonné cette cible et nous avons observé une amélioration dans le résultat final.

Les positions de caméra sont trouvées. L'erreur maximum de la distance résiduelle entre la projection de cible réelle et la projection estimée varie de 0.7 à 1.4 pixels pour les photos différentes.

6.7 Comparaison de résultat

La déviation entre « Incremental » et la vérité terrain (millimètre) :

	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5
Distance	1.35	0.84	2.41	1.25	1.70

La déviation entre « Global » et la vérité terrain (millimètre) :

	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5
Distance	0.80	0.59	1.93	1.27	1.44

6.8 Analyse d'erreur

Nous constatons une déviation de 2 mm à la fin. Vu que c'est une expérience en intérieur d'une scène de 4 m, un résultat avec une erreur de $1/2000$ est correct. Mais il n'est tout de même pas précis. Dans l'article de M. Monasse et M. Moulon [4], ils ont eu un résultat avec une erreur de 2.5 mm dans une scène « Fountain-P11 » publiée par Strecha, dont la largeur est de 9 mètres. La distance entre l'objet et la caméra est de 10 m. Alors la précision est de $1/4000$.

Pour avoir une compréhension intuitive, on peut analyser la relation entre la caméra et les objets. La focale de notre caméra est d'environ 4000 pixels. La distance entre la caméra et les objets est de 4 m. Alors si la caméra se déplace dans le sens horizontal d'1 mm, l'image des objets va être décalée d'un pixel sur la photo. Autrement dit, pour atteindre une précision de 1 mm de l'estimation de position, il faut examiner toutes les étapes de traitement d'image et contrôler les erreurs de moins d'un pixel. La caméra est en effet très sensible.

Pour revenir sur les étapes précédentes, l'erreur maximale de la correction de distorsion est de 0.4 pixel. Cependant, nous avons constaté que la correction de

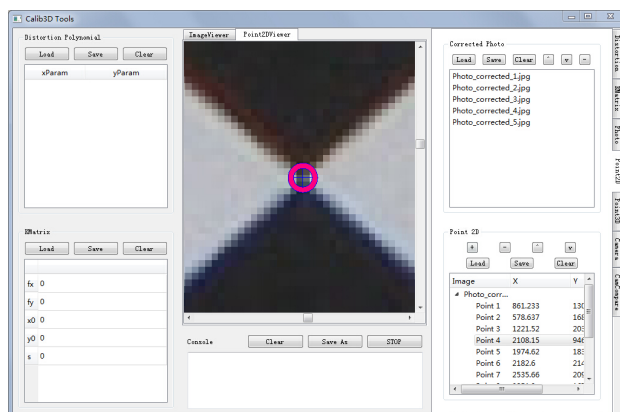


FIGURE 6.5: Zoomer et sélectionner

distorsion n'est pas toujours stable. Nous avons divisé les photos de la « harpe » en deux groupes et ainsi engendré deux polynômes différents. Une photo de la « harpe » est ensuite corrigée respectivement par les deux polynômes. Enfin, nous avons obtenu deux photos corrigées. Elles sont légèrement différentes. La différence est évidente au bord de la photo, qui montre un écart de 2 pixels. Mais quand nous évaluons la rectitude des lignes corrigées dans les deux photos, nous trouvons que c'est toujours très précis. Nous avons discuté de ce phénomène avec M. Monasse. En fait, il est possible pour l'algorithme d'introduire une transformation d'homographie parasite. Mais une fois que toutes les photos sont corrigées par le même polynôme, il n'y a plus d'influence sur le résultat.

La méthode « harpe » est valide. En revanche, dans la méthode « point de contrôle circulaire », nous avons observé une erreur de plus de 10 pixels sur la focale en testant différentes photos de la mire. Une raison possible est que la mise au point automatique réalisée par la caméra peut changer légèrement la focale. Pour notre caméra, une différence de 10 pixels correspond à 0.06 mm. Comparé à la focale de 24 mm, cette hypothèse est vraisemblable. De plus, nous avons divisé les photos selon la distance entre la caméra et la mire et la focale varie de 4315 – 4330 pixels. Pour supprimer cette erreur, il faut placer la mire loin de la caméra. Au contraire, la caméra doit se rapprocher de la mire pour rendre les points de contrôle clairs. C'est un paradoxe. Une solution possible est que l'on fabrique une mire plus grande.

Les coordonnées des cibles dans les photos sont indiquées à la main. L'utilisateur peut agrandir l'image pour sélectionner (figure 6.5). Nous pensons que l'erreur de sélection est moins de 0.5 pixel.

De plus, l'appareil LIDAR a déjà une précision de 2 mm. D'ailleurs, à cause de certaines limites, nous n'avons pas fixé la caméra à un trépied. Cela diminuer la qualité de photos, mais nous n'avons pas assez de temps de comparer et analyser ce facteur.

Retour d'expérience

Le stage est terminé. Si nous avions plus de temps, nous voudrions encore faire plus d'expériences pour trouver les sources d'erreurs. D'ailleurs, comme nous n'avons pas analysé mathématiquement la propagation d'erreur, nous ne connaissons pas la précision de la vérité terrain.

Il faut encore faire attention aux paramètres des appareils. Notre appareil LIDAR assure une précision de 2mm. Pour une scène en intérieur c'est déjà très grand. Mais il peut scanner jusqu'à 130m. Donc nous pourrions avoir un meilleur résultat si nous considérons une scène à l'extérieur car l'erreur relative va diminuer. Aussi, il est mieux de trouver les paramètres de performance de la lentille et la caméra utilisées. Nous avons utilisé la mise au point automatique de la caméra. En fait, cela fait varier la focale et induit des erreurs.

Notre logiciel Calib3DTools est maintenant utilisable. Mais nous ne pouvons pas dire qu'il est vraiment très stable. Il existe encore des bogues cachés dans le code. Nous espérons que ce logiciel sera utile pour les chercheurs. Dans ce cas, nous pourrions continuer de le faire évoluer.

Bilan personnel

Ce stage a vraiment été une expérience enrichissante pour moi. Bien qu'il ait seulement duré trois mois, j'ai appris de nouvelles connaissances et des techniques pratiques chaque jour. J'ai beaucoup acquis de compétences, principalement dans deux domaines : les techniques de développement de logiciel et la science de vision par ordinateur.

C'est la première fois que je développe un logiciel sérieux. M.Monasse nous a conseillé d'utiliser la bibliothèque Qt pour développer l'interface graphique. Du coup, je me suis d'abord familiarisé avec cette bibliothèque. Qt est une bibliothèque puissante, et en même temps très conviviale pour un débutant, mais j'ai tout de même eu besoin d'un mois pour en comprendre le fonctionnement . En outre, les liaisons entre différents modules, la conception de fenêtre sur l'écran, et tous ces mécanismes derrière l'interface graphique me donnent une vision profonde aux des logiciels qu'on utilise tous les jours.

Le langage utilisé est C++. Lors de la phase de développement, j'ai rencontré beaucoup de problèmes complexes que j'ai résolus successivement. C'est ainsi que j'ai appris de nombreuses caractéristiques du C++, qui est une langage puissant mais difficile à maîtriser.

Le traitement d'image consomme souvent beaucoup de temps à cause de la complexité et la quantité des calculs . Donc j'ai essayé d'utiliser le multithread pour accélérer les calculs. Et je trouve que c'est vraiment un sujet passionnant car il faut faire attention aux problèmes comme la synchronisation des tâches, le partage des données, etc. De plus, un bogue peut survenir très facilement mais être difficile à trouver par la suite. Néanmoins, en étant patient, on a réussi à améliorer certains algorithmes.

Nous avons utilisé Git pour gérer la version de logiciel. Les codes sont mis en ligne sur GitHub. Cela prend un peu de temps pour en comprendre le

fonctionnement et les subtilités . Mais il s'avère qu'il peut vraiment rendre le travail plus efficace et simplifier la vie de l'informaticien.

D'un autre côté, ce stage m'a guidé vers un nouveau domaine : la vision par ordinateur. Cela a beaucoup influencé mon choix de parcours pour ma deuxième année : vision et apprentissage automatique. Cela a été une bonne occasion pour moi pour aborder les sujets en avance.

J'ai aussi appris que le principe de la formation d'image de caméra peut être décrit par un modèle mathématique de caméra et que basé sur ce modèle, les images sont liées par les équations linéaires. D'ailleurs, en lisant les codes donnés par notre tuteur, et aussi les thèses concernant ce sujet, j'ai connu les détails de ces techniques dans le domaine. Je m'aperçois que les méthodes adoptées dans notre logiciel sont inventées et ensuite améliorées graduellement par les chercheurs. Il nous prend trois mois pour développer un logiciel simple avec quelques fonctions intéressantes. En effet, nous sommes sur les épaules des géants.

En plus de la théorie, j'ai vu aussi les applications dans la réalité. L'expérimentation de Strecha et son groupe est déjà impressionnant pour moi. Encore, j'ai trouvé sur Internet certain technique merveilleuse qui peut analyser une vidéo et le reconstruire en 3D simultanément. Bien sûr nous avons aussi appliqué les théories dans notre expérience. Au début nous avons seulement la théorie et les algorithmes. Par le processus d'étudier, comprendre et appliquer, nous avons reconstruit notre modèle 3D. C'est absolument une expérience excellente.

Une autre récolte est la connaissance sur l'environnement de travail dans laboratoire. Au laboratoire d'accueil, je sens une atmosphère académique forte. La réunion régulière du laboratoire est une activité intéressante. Les chercheurs de nationalités différentes partagent leurs idées dans une salle. J'en ai profité un peu quand je travaille dans le coin, même si je comprends qu'un peu la plupart de temps. De plus, quand je passe par le couloir, je peux parfois apercevoir quelque chercheurs discutent entre eux devant un tableau.

Pour conclure, dans ce stage il y a encore ces erreurs dont sources que nous n'avons pas tous trouvées. Si le temps le permet, je voudrais faire encore plus des expériences pour les analyser. Le temps passe vite. Néanmoins, nous avons développé un logiciel pour faciliter le travail de chercheur. En effet, un doctorat l'a déjà utilisé une fois pour calibrer son appareil photo. Ça me fait plaisir de savoir que nos efforts ont sa valeur pour les autres. Je vais continuer à maintenir ce logiciel dans le futur. Ce stage a augmenté mon intérêt pour

l'informatique et pour la vision par ordinateur. C'est une expérience précieuse pour moi. Je ne peux pas attendre pour entrer dans la deuxième année.

Bibliographie

- [1] P. MOULON. *OpenMVG*. 2015. URL : <https://github.com/openMVG/openMVG> (visit  le 16/07/2015).
- [2] Z. TANG et al. « High-precision camera distortion measurements with a “calibration harp” ». In : *JOSA A* 29.10 (2012), p. 2134–2143.
- [3] P. MOULON, P. MONASSE et R. MARLET. « Adaptive structure from motion with a contrario model estimation ». In : *Computer Vision–ACCV 2012*. Springer, 2013, p. 257–270.
- [4] P. MOULON, P. MONASSE et R. MARLET. « Global fusion of relative motions for robust, accurate and scalable structure from motion ». In : *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE. 2013, p. 3248–3255.
- [5] C. STRECHA et al. « On benchmarking camera calibration and multi-view stereo for high resolution imagery ». In : *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, p. 1–8.
- [6] *SCENE*. 2015. URL : <http://www.faro.com/fr-fr/produits/logiciels-faro/scene/apercu> (visit  le 16/07/2015).
- [7] L. MOISAN, P. MOULON et P. MONASSE. « Automatic homographic registration of a pair of images, with a contrario elimination of outliers ». In : *Image Processing On Line* 2 (2012), p. 56–73.
- [8] L. FENG et L. ZOUAKI. *Calib3DTools*. 2015. URL : <https://github.com/flm8620/Calib3DTools> (visit  le 16/07/2015).
- [9] V. RUDAKOVA et P. MONASSE. « Camera matrix calibration using circular control points and separate correction of the geometric distortion field ». In : *Computer and Robot Vision (CRV), 2014 Canadian Conference on*. IEEE. 2014, p. 195–202.

Annexe 1 : Mode d'emploi de Calib3DTools

Dans l'interface, il y a cinq sous-fenêtres (figure 9.1). (1) et (2) sont deux tableaux pour afficher respectivement le polynôme de distorsion et la matrice K. (3) contient deux visionneuses d'image. (4) est un console pour afficher les messages et les résultats.

Le (5) est composé de 7 étiquettes. C'est l'endroit où on lance les calculs différents.

Éléments de l'interface (figure 9.1) :

1. Deux visionneuses pour afficher les images et les points 2D
2. Une console pour afficher les messages et les résultats
3. « Clear » : supprimer tous les messages.
4. « Save As » : sauvegarder les messages affichés sous le format HTML
5. « Stop » : arrêter un calcul
6. Un tableau du polynôme de distorsion.
7. Un tableau pour afficher la matrice K.
8. Les étiquettes de différentes fonctionnalités. Les parties 6 et 7 changent selon le choix de l'étiquette.

Calib3DTools a principalement deux fonctionnalités : calibrer la caméra et évaluer une reconstruction 3D.

Calculer le polynôme de distorsion :

1. Choisir l'étiquette « Distortion ».
2. Charger les photos de « harpe » dans « harp photo »
3. Cliquer « Calcul Distortion Polynomial » et attendre le calcul.
4. Vérifier que les photos sont bien traitées dans « Detect Feedback »



FIGURE 9.1: Les widgets dans Calib3D Tools

5. Sauvegarder le polynôme dans « Distortion Polynomial »

Corriger les photos pour supprimer la distorsion :

1. Choisir l'étiquette « Photo »
2. Charger le polynôme dans « Distortion Polynomial »
3. Cliquer « Apply Distortion Correction » et attendre le calcul.
4. Sauvegarder les photos corrigées dans « Corrected Photo »

Calculer la matrice K :

1. Choisir l'étiquette « KMatrix »
2. (Optionnel) Charger les photos de mire avec distorsion et ensuite corriger.
3. Charger les photos de mire corrigées.
4. Cliquer « Calculate Matrix K » et attendre le calcul.
5. Sauvegarder la matrice K

Indiquer les points 2d :

1. Choisir l'étiquette « Point2D »
2. Charger les photos de la scène corrigées.

3. Cliquer « + » dans « Point2D » pour ajouter les points
4. Cliquer « Point2D Viewer », indiquer les positions des points

Indiquer les points 3d :

1. Choisir l'étiquette « Point3D »
2. Cliquer « + » dans « Point3D » pour ajouter les points
3. Remplir les coordonnées

Calculer la vérité terrain :

1. Vérifier la matrice K, les points 2D et les point3D sont chargés
2. Choisir l'étiquette « Camera »
3. Cliquer « Solve »
4. Sauvegarder les positions de la caméra.

Comparer les résultats de stéréovision avec la vérité terrain

1. Charger les positions de la caméra de la vérité terrain.
2. Choisir l'étiquette « CamCompare »
3. Charger les positions de la caméra trouvées par OpenMVG.
Cliquer « load from OpenMVG *.json » et
choisir le fichier sfm-data.json
4. Cliquer « Compare With CamPos »
5. Sauvegarder les résultats dans la console.

Annexe 2 : Résultats et format de fichier

10.1 Polynôme de distortion

```
Distortion.txt :
maxOrder:
11
Polynomial for X:
-0.0004919764715433032
0.9999996366896148
-6.461912625377179e-07
3.104362820559672e-07
-4.46940704664302e-07
...
Polynomial for Y:
0.0006698089693754963
-2.186716648151154e-07
0.9999982012685996
8.066661521951027e-08
4.392343232065116e-07
-1.383619885714114e-07
...
```

10.2 La matrice K

```
K.txt :
fx :
4318.33
```

fy :
4319.51
x0 :
1963.93
y0 :
1294.52
s :
-3.79

10.3 Les points 2D

ImageCount:
5
PointCount:
7
Image_1
861.47 1307.02
1221.43 2035.99
2108.05 946.58
1974.27 1835.93
2182.34 2148.04
2535.48 2094.66
3251.74 1672.82
Image_2
681.66 1228.81
1097.76 2118.80
2165.11 792.21
2008.50 1854.29
2262.35 2263.10
2717.87 2250.88
3572.63 1680.15
Image_3
770.07 1320.17
1381.69 2062.30
1984.82 908.48
1952.50 1795.92
2361.53 2066.44
2902.95 1959.08

```
3189.97 1538.83
Image_4
 867.15 1357.67
1764.44 2208.09
1987.13 864.20
2051.82 1845.60
2660.76 2106.14
3357.30 1949.48
3205.37 1505.73
Image_5
819.03 1180.05
1522.11 1854.20
1786.61 791.29
1824.50 1603.38
2297.81 1808.13
2838.26 1672.13
2831.40 1334.68
```

10.4 Les point 3D

```
3D.txt:
Point_1
-1.33889 2.63972 74.01543
Point_2
-0.69054 2.26496 73.31386
Point_3
-0.05161 3.30221 74.39467
Point_4
-0.12167 2.97639 73.39861
Point_5
0.281854 2.44885 73.22529
Point_6
0.733612 2.01352 73.43858
Point_7
1.270545 2.91551 73.66349
```

10.5 Les positions de la vérité terrain

CamPos.txt:

Camera 1

Matrix R:

```
0.941241 0.337212 -0.018772
0.016652 -0.101853 -0.994660
-0.337324 0.935902 -0.101483
```

Center:

```
1.50838 -1.54821 74.50629
```

Camera 2

Matrix R:

```
0.938206 0.345314 -0.022954
0.025873 -0.136130 -0.990352
-0.345108 0.928561 -0.136653
```

Center:

```
1.24506 -0.78883 74.49062
```

Camera 3

Matrix R:

```
0.999453 -0.020015 -0.026296
-0.028299 -0.107403 -0.993812
0.0170676 0.994014 -0.107910
```

Center:

```
-0.176779 -1.83531 74.492741
```

Camera 4

Matrix R:

```
0.925153 -0.379532 -0.006768
-0.051745 -0.108431 -0.992756
0.376049 0.918802 -0.119955
```

Center:

```
-1.842740 -0.999433 74.492048
```

Camera 5

Matrix R:

```
0.939079 -0.343575 -0.009214
-0.054590 -0.122630 -0.990949
0.339336 0.931084 -0.133915
```

Center:
-1.75627 -2.02313 74.49385

10.6 Les positions calculées par la stéréovision

CamPos_global.txt:
Camera 1
Matrix R:
0.999999 7.294496e-05 -1.529458e-05
-7.294541e-05 0.999999 -2.963142e-05
1.529242e-05 2.963254e-05 0.999999
Center:
-6.770762e-05 -4.142550e-06 7.192926e-05

Camera 2
Matrix R:
0.999954067 0.003305 0.008996
-0.00298107 0.999353 -0.035820
-0.00910912 0.035792 0.999317
Center:
0.002397 -0.019803 0.237511

Camera 3
Matrix R:
0.934549 0.044962 -0.352980
-0.044257 0.998969 0.010071
0.353069 0.006209 0.935576
Center:
-0.498606 0.004667 0.089467

Camera 4
Matrix R:
0.743244 0.061175 -0.666216
-0.067228 0.997599 0.016603
0.665633 0.032447 0.745573
Center:
-0.880508 -0.029612 0.486893

Camera 5

Matrix R:

```
0.768234 0.059993 -0.637351
-0.074282 0.997227 0.004330
0.635844 0.044017 0.770561
```

Center:

```
-0.959008 0.001887 0.194616
```

CamPos_incremental.txt:

Camera 1

Matrix R:

```
0.999999 2.227980e-05 -0.000643
-2.227824e-05 0.999999 2.437839
0.000643 -2.423511e-06 0.999999
```

Center:

```
-0.003755 0.000753 0.001043
```

Camera 2

Matrix R:

```
0.999957 0.0032935 0.008597
-0.002984 0.9993574 -0.035716
-0.008709 0.0356896 0.999324
```

Center:

```
0.008343 -0.082099 0.998041
```

Camera 3

Matrix R:

```
0.934458 0.044949 -0.353222
-0.044274 0.998969 0.009994
0.353307 0.006298 0.935486
```

Center:

```
-2.094847 0.019691 0.3767697
```

Camera 4

Matrix R:

```
0.742895 0.061058 -0.666616
-0.067332 0.997596 0.016338
0.666012 0.032746 0.745221
```

Center:

```
-3.699180 -0.125342 2.046810
```

Camera 5

Matrix R:

```
0.767890 0.059911 -0.637773
-0.074304 0.997226 0.0042132
0.636257 0.044154 0.770212
```

Center:

```
-4.029318 0.007596 0.819973
```