وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار - عنابة

Faculté des sciences de l'ingénieur **Année :** 2011

Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme de **MAGISTER**

Règles de calcul pour une recomposition d'images par apprentissage

Option

TIC & Ingénierie du document

Par

Nadia GUERROUI

DIRECTEUR DE MEMOIRE : Hamid SERIDI Professeur Université de Guelma

DEVANT LE JURY

PRESIDENT :	Mohamed Tarek KHADIR	Pr	Université d'Annaba
EXAMINATEURS:	Hayet Farida MEROUANI	MC	Université d'Annaba
	Hassina SERIDI	MC	Université d'Annaba

LISTE DES FIGURES

وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار - عنابة

Faculté des sciences de l'ingénieur **Année :** 2011

Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme de **MAGISTER**

Règles de calcul pour une recomposition d'image par apprentissage

Option

TIC & Ingénierie du document

Par

Nadia GUERROUI

DIRECTEUR DE MEMOIRE : Hamid SERIDI Professeur Université de Guelma

DEVANT LE JURY

PRESIDENT :	Mohamed Tarek KHADIR	Pr	Université d'Annaba
EXAMINATEURS:	Hayet Farida MEROUANI	MC	Université d'Annaba
	Hassina SERIDI	MC	Université d'Annaba

ملخص

يمكن للصور البانورامية أن تشمل توسيع مجال الرؤية المحدود هناك طريقتين للحصول على الصور البانورامية :إما مباشرة باستخدام نظم شاملة لكل الاتجاهات ، حيث يتم التقاط صورة بانورامية مباشرة ، أو بإعادة بناء المشهد بواسطة سلسلة من الصور المتعاقبة زمنيا مع مراعاة حركة الكاميرا. الحقل التجريبي هنا عبارة عن بناء مشاهد من خلال صور لمنتجعات التزلج على الجليد . في الواقع الخوارزميات المستعملة تعمل بشكل جيد جدا في الظروف الطقس المثالي. ومع ذلك ، فإن أدوات البناء في هذه الخوارزميات تطرح مشاكل حقيقية لمعاودة بناء الصور بطريقة لا تشوبها شائبة خاصة بسبب سوء الأحوال الجوية الذي يعتبر مصدر تشويش و عيوب لإعادة البناء ، ويمكن أن نستحضر أحد هذه العيوب المعروفة باسم الشبح الناتج عن الآثار الناجمة عن تحركات المتزلجين مثلا ويمكن حل هذه المشكلة أن ينظر إليه من خلال طريقة لتكون طموحة تعطي نتائج مقبول إلى حد ما ، نقترح حلا حيث يتم استخدام تسجيل الأسلوب في البداية ، قبل تطبيق خوارزمية الرؤية الحاسوبية على بيانات الصورة من أجل الحصول على معلومات مفيدة، فإنه من الضروري إجراء عمليات مسبقة على البيانات من أجل تأكيد أن البيانات تحقق افتراضات محددة تابعة للخوارزمية. لجعل هذا ممكنا ، فمن الضروري استخدام المعالم المحلية للصورة كالزوايا و تحويل صفة صورة غير مرتبط بمقياس حيث يتم استحصال معالم الصورة على مستويات دقة مختلفة من بيانات الصورة ذاتها, أين يتم تحديد أي نقاط أو مناطق من الصورة هي المناطق الهامة من أجل العمليات اللاحقة.

- كاكتشاف النقاط المميزة في الصورة و التي ستشكل المراكز للصفات المرشحة.
- إيجاد المواقع الدقيقة للصفات و حذف النقاط غير المستقرة
- حساب اتجاه محلي لكل نقطة متبقية من المرحلة السابقة
- توصيف النقطة بشعاع مؤلف من 128 مركبة من خلال المنطقة الدائرية المحيطة بها.

بعد مطابقة النقاط الميزة يمكن الحصول على مصفوفة التحول باستخدام عينة عشوائية وهو أسلوب تكراري لتقدير معالم نموذج رياضي من مجموعة من البيانات الملحوظة ،وذلك باختيار أربع نقاط مهمة تكون كافية لحساب مصفوفة التحول التي تسمح لنا ببساطة محاذاة الصور معا .
الصورة المشيدة تحمل بعض التشويش والملاحظ مباشرة هو الفرق في الكثافة الذي يمكن معالجته بكسب التعويض و هو وسيلة أساسية للقضاء على شدة الاختلاف.

كلمات مفاتيح: النقاط المهمة ، معايرة ، إعادة بناء المشهد البانورامي ،تحويل صفة صورة غير مرتبط **بمقياس** ،مصفوفة التحويل.

ABSTRACT

A panorama is a picture that is made by combining a series of photos into one large picture. By combining a series of photos, it is possible to provide a complete view of an area or location that cannot fit in a single shot. Obtaining images panoramas can be summarized into two groups: systems omnidirectional, which directly capture a panoramic image, and systems by mosaic. The mosaic case is to deform and align a series of images, obtained for example with a rotating camera around its optical axis. When it is not known, it is find the geometric transformation which connects the coordinates of two consecutive frames in order to align them. The purpose of this study .This work is the reconstruction of a mosaic of images, a problem important field of computer vision.

Here, the experimental field for the reconstruction of images scenic for ski resorts. Algorithms redial known literary work very well in the conditions ideal weather. However, tools based on these algorithms pose real problems for a reconstruction without artifacts in this context of use. Indeed, bad weather is a source defects redial. We could also mention the failure known the name ghost effects caused by the movement of skiers. Solving this problem can be seen through a method that will be ambitious and gives acceptable results in some degree. We propose a solution where registration is used first sparse method to approach the solution and a dense method to refine the result.

In order to create our panorama, the general idea will consist in identifying common points between the two images and then projecting one of the images on top of the other in an effort to match those points. In order to identify those points, which from now on we will be calling interest points, we will be using a simple interest point detector known as the Harris Corners Detector. Which is robust to rotations, variations in brightness and change of scale. To make this possible, it is necessary that the descriptors follow a certain number of assumptions, so we chose as the SIFT descriptor features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. After our interest points have been detected, we need to correlate them somehow. Now that we have two sets of correlated points, all we have to do is define a model which can translate points from one set to the other. What we are looking for is some kind of image transformation which can be used to project one of the two images on top of the other while matching most of the correlated feature points - we need an homography matrix matching the two images.

By using homogeneous coordinates, one can represent an Homography matrix as a 3x3 matrix with 8 degrees of freedom. We can create this homography matrix from our set of correlated points. To estimate a robust model from the data, we will be using a method known as RANSAC. The name RANSAC is actually an abbreviation for "RANdom SAmple Consensus". It is an iterative method for robust parameter estimation to fit mathematical models from sets of observed data points which may contain outliers.

After our homography matrix has been computed, all that is left for us is to blend the two images together. The stitched image here has obvious artifact. The most direct one is the difference of intensity. The Gain Compensation is a fundamental way to eliminate intensity difference.

Keywords: interest points, matching, calibration, homography,SIFT, panoramic reconstruction.

RESUME

Les images panoramiques permettent notamment d'élargir le champ de vision restreint des systèmes de captures d'images. L'obtention d'images panoramiques peut se résumer en deux groupes : les systèmes omnidirectionnels, qui capturent directement une image panoramique, et les systèmes par mosaïque. Ce dernier cas consiste à déformer et aligner les images d'une série, obtenues par exemple avec une caméra en rotation autour de son axe optique. Lorsque celle-ci n'est pas connue, il faut retrouver la transformation géométrique qui relie les coordonnées de deux images consécutives afin de pouvoir les aligner. L'objet d'étude de ce présent travail est la reconstruction d'une mosaïque d'images, problème important du domaine de la vision par ordinateur.

Ici, Le champ expérimental concerne la recomposition d'images panoramiques pour des stations de ski. Les algorithmes de recomposition connus de la littérature fonctionnent très bien dans les conditions météorologiques idéales. Cependant, les outils basés sur ces algorithmes posent de réels problèmes pour une recomposition sans défauts dans ce contexte d'utilisation. En effet, une mauvaise météo est source de défauts de recomposition .On peut aussi évoquer le défaut connu sous le nom fantôme (ghost) causé par effets de mouvements des skieurs.

La résolution de ce problème peut être vue à travers une méthode qui sera ambitieuse et qui donne résultats acceptables à certain degré.

Nous proposons une solution de recalage où on utilise une première méthode éparse pour s'approcher de la solution puis une méthode dense pour affiner le résultat.

Dans un premier temps, l'utilisation d'un détecteur de points d'intérêts permet d'extraire d'une image les coordonnées de points caractéristiques Le détecteur présenté ici est une version améliorée du détecteur de coins de Harris, le détecteur de Harris-Laplace, qui est robuste aux rotations, aux variations de luminosité et aux changements d'échelle.

Dans un deuxième temps, les points ainsi localisés sont décrits à l'aide de descripteurs de vecteurs de caractéristiques qui permettent ensuite d'appareiller les points d'intérêts de plusieurs prises de vue distinctes d'une même scène. *Toutefois, cette correspondance n'est possible que lorsque le centre optique des deux images est confondu. C'est-à-dire que lors de la prise de vue des deux images la caméra (ou l'appareil photo) n'a subit que des rotations mais pas de translation. Pour rendre cela possible, il est nécessaire que les descripteurs vérifient un certain nombre d'hypothèses alors on a choisit SIFT comme le descripteur de caractéristiques le plus connu grâce de sa stabilité et sa précision hautement performant aux autres descripteurs. Après la mise en correspondance des vecteurs de caractéristique. Nous obtenons la matrice de transformation en utilisant RANSAC "RANdom SAmple Consensus". Il s'agit d'une méthode itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées, la transformation entre les deux images est une homographie 2D qui ce calcul à partir d'une matrice 3x3 avec 8 coefficients indépendants que vous pouvez estimer à partir d'au moins 4 points en correspondances (et en coordonnée homogène). Depuis la matrice de transformation est connu, les deux images peuvent être simplement alignées ensemble. . L'image construite a un évident artefact. Le plus observer directement est la différence d'intensité. Le gain de compensation est une voie fondamentale à éliminer différence d'intensité.*

***Mots-clés** : points d'intérêt, mise en correspondance, calibration, homographie,SIFT, la reconstruction panoramique.*

REMERCIEMENTS

Je voudrais tout d'abord remercier chaleureusement mon directeur de recherche, le professeur Hamid SERIDI, pour m'avoir accueillie au sein de son équipe. Sa très grande disponibilité, sa patience et ses conseils judicieux m'ont permis de terminer mes travaux dans les temps impartis, merci pour tout ! Je remercie également mon co-directeur de recherche, Sid Ahmed LAMROUS pour ses remarques pertinentes.

TABLE DES MATIERES

LISTE DES FIGURES

INTRODUCTION GENERALE

De nombreuses applications en vision par ordinateur nécessitent un champ de vision large, comme par exemple la construction de cartes aériennes ou satellites ou encore la vidéosurveillance. Cependant, de nombreux systèmes conventionnels de capture d'images sont limités par leur champ de vision, souvent plus petit que celui de l'humain. La création d'images panoramiques, qui sont des images à champ visuel très large, est un moyen permettant de compenser ce champ visuel limité. Les images panoramiques sont tout d'abord apparues dans le domaine de la peinture, à la fin du 18e siècle, dans le but de donner l'illusion de regarder une scène réelle, avant de gagner le domaine de la photographie.

L'utilisation des images panoramiques s'est très vite étendue à de nombreux domaines. Par exemple en robotique, l'usage d'un robot pour visualiser une scène inaccessible à un humain est désormais fréquente. Le robot reconstitue l'environnement dans lequel il se déplace en prenant une série de photos et en les assemblant pour obtenir une image continue. En compression vidéo, le décor fixe, sous forme de panorama, peut être compressé indépendamment des objets mobiles, permettant ainsi de gagner de l'espace.

Dans le cas de la construction d'environnements virtuels pour la réalisation de décors 3D, on peut citer les visites virtuelles de musées, ou de maisons pour la vente immobilière.
Il existe deux types de méthodes pour obtenir un panorama : les méthodes visant à augmenter le champ visuel de la caméra à l'aide d'un système optique (systèmes omnidirectionnels) et les méthodes visant à reconstruire le panorama à partir d'une série d'images. Les systèmes omnidirectionnels permettent d'obtenir une vue panoramique en une seule image, tandis que les mosaïques d'images nécessitent d'aligner les différentes images de la série, prises à chaque position de la caméra, pour reconstruire le panorama.

Dans le présent mémoire, nous nous intéressons à la fabrication de mosaïques d'images.

La construction d'une mosaïque à partir d'une série d'images nécessite un recalage géométrique approprié des repères des images, afin de les mettre dans le même système de coordonnées. Dans le cas où le centre optique est immobile, deux images sont reliées par une seule transformation homographique 2D (transformation projective). Il est donc nécessaire d'estimer cette transformation pour aligner les deux repères des images données.
La transformation est ensuite utilisée pour déformer une image afin de l'amener dans le même système de coordonnées que l'autre. En fusionnant les régions de recouvrement de l'image déformée avec celles de l'autre image, il est possible de construire la mosaïque.

Pour reconstituer un panorama à partir d'un grand nombre d'images, étudions tout d'abord, l'association de deux images qui se recouvrent partiellement et ensuite, appliquons les principes en question de plus près pour l'ensemble des images d'origine.

Pour la construction automatique de la mosaïque, les étapes suivantes sont nécessaires :

- Détection des points d'intérêt
- Corrélation automatique
- La mise en correspondance
- Approximation de l'homographie correspondante
- Formation du panorama.

Nous avons scindé notre travail de recherche en 5 chapitres.
Le chapitre 1 est consacré à la représentation de grandes étapes pour la construction et l'exploitation d'une image panoramique : l'acquisition, la projection des prises de vue dans l'image panoramique, l'amélioration du rendu et la visualisation.

.Le chapitre 2 est consacré aux différentes représentations possibles des mosaïques d'image. Dans ce chapitre nous nous placerons dans le cas idéal où la caméra correspond exactement au modèle et où il n'y a aucune distorsion de quelque nature que ce soit. Bien évidemment, nous serons rapidement confrontés aux limites de ce cas idéal. Dans ce chapitre, nous introduisons également les principes mathématiques de base permettant de calculer une mosaïque d'images. Une description plus complète est donnée en annexe. Enfin, dans ce chapitre nous présentons une étude théorique permettant un pavage optimal de la sphère à partir de surfaces rectangulaires en limitant les zones de recouvrement. Cette étude nous permet de calculer une trajectoire optimale de la caméra et de limiter le nombre de prises de vues nécessaire à la représentation de la scène.
Dans le chapitre 3, nous allons aborder les principales distorsions que nous rencontrons lors de la création d'une mosaïque d'image. Le but de ce chapitre est double. Le premier est de vérifier l'écart des caméras par rapport au modèle que nous utilisons et surtout d'en définir les limites de validité. Le deuxième est de proposer des outils permettant de réaliser des panoramas robustes en temps réel. La contrainte forte du temps réel, nous oblige à faire des choix entre la qualité de la construction et la rapidité de sa mise en œuvre. Nous évoquons essentiellement les problèmes d'alignement photométrique et la suppression des fantômes.

Le chapitre 4 est consacré lui aussi à l'amélioration de la qualité du panorama. Cependant, ce chapitre se focalise exclusivement sur le problème de recalage d'images auquel nous sommes invariablement confrontés lorsque les paramètres de prise de vue ne sont pas disponibles ou ne sont pas suffisamment précis. Une bonne part de notre travail de recherche s'est attachée à résoudre ce problème en temps réel. Le recalage d'images n'est pas un problème récent et plusieurs solutions ont déjà été apportées. Cependant, les expérimentations que nous avons effectuées ne se sont pas révélées concluantes en termes de qualité du recalage mais surtout en temps de calcul. Pour les applications que nous envisageons, le recalage d'image n'est qu'une étape intermédiaire. L'ensemble du processus, recalage compris, devant s'exécuter en temps réel. Ensuite nous passons en revues les principales méthodes de recalage denses et éparses décrites dans la littérature.

Le chapitre 5 est consacré travail réalisé .L'application mise en œuvre est basée sur les travaux de David G.lowe pour la détection des points d'intérêts.
Nous proposons une solution de recalage où on utilise une première méthode éparse pour s'approcher de la solution puis une méthode dense pour affiner le résultat.

Dans un premier temps, l'utilisation d'un détecteur de points d'intérêts permet d'extraire d'une image les coordonnées de points caractéristiques Le détecteur présenté ici est une version améliorée du détecteur de coins de Harris, le détecteur de Harris-Laplace, qui est robuste aux rotations, aux variations de luminosité et aux changements d'échelle.
Dans un deuxième temps, les points ainsi localisés sont décrits à l'aide de descripteurs de vecteurs de caractéristiques qui permettent ensuite d'appareiller les points d'intérêts de plusieurs prises de vue distinctes d'une même scène. Pour rendre cela possible, il est nécessaire que les descripteurs vérifient un certain nombre d'hypothèses alors on a choisit SIFT (The Scale Invariant Feature Transform) comme le descripteur de caractéristiques le plus connu grâce de sa stabilité et sa précision hautement performant par rapport aux autres descripteurs. Après la mise en correspondance (*Matching*) des vecteurs de caractéristique. Nous obtenons la matrice de transformation en utilisant RANSAC "RANdom SAmple Consensus". Il s'agit d'une méthode itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées. Depuis la matrice de transformation est connu, les deux images peut être simplement alignées ensemble. L'image construite a un évident artefact. Le plus observer directement est la différence d'intensité. Le gain de compensation est une voie fondamentale à éliminer différence d'intensité. Dans ce chapitre les résultats obtenus sont discutés en dernier lieu.
Pour visualiser du panorama on utilise Live Picture browser. Ce browser s'exécute en Java, sans nécessité de plugin spécial n'importe qui avec un navigateur compatible Java Web devrait être en mesure d'afficher notre panorama.

Nous terminerons notre travail par une conclusion générale et des perspectives.

Chapitre 1

Mosaïque d'images

1.1 Présentation

Il est manifeste aujourd'hui que la surveillance des lieux et des personnes se généralise actuellement, que ce soit dans des zones du domaine public ou chez des particuliers. Si cette tendance se poursuit, il est bien évident que les besoins d'une analyse automatique des séquences vidéo obtenues seront de plus en plus importants.
La nécessité de combiner des images dans un panorama existe depuis le début de la photographie .La création d'images panoramiques, qui sont des images à champ visuel très large, est un moyen permettant de compenser ce champ visuel limité.
De nombreuses applications en vision par ordinateur nécessitent un champ de vision large, comme par exemple la construction de cartes aériennes ou satellites ou encore la vidéosurveillance. Cependant, de nombreux systèmes conventionnels de capture d'images sont limités par leur champ de vision, souvent plus petit que celui de l'humain. L'utilisation des images panoramiques s'est très vite étendue à de nombreux domaines. Par exemple en robotique, l'usage d'un robot pour visualiser une scène inaccessible à un humain est désormais fréquente. Le robot reconstitue l'environnement dans lequel il se déplace en prenant une série de photos et en les assemblant pour obtenir une image continue. En compression vidéo, le décor fixe, sous forme de panorama, peut être compressé indépendamment des objets mobiles, permettant ainsi de gagner de l'espace.
Dans le cas de la construction d'environnements virtuels pour la réalisation de décors 3D, on peut citer les visites virtuelles de musées, ou de maisons pour la vente immobilière.
Ces 15 dernières années, les systèmes panoramiques de formation d'image ont sensiblement progressé. Non seulement les professionnels peuvent créer et afficher des panoramas, mais il existe une grande quantité de logiciels disponibles, dont certains gratuits[1]. Chacun d'entre nous, avec un ordinateur et simple appareil photo numérique, peut créer des panoramas. Même les grandes compagnies sont présentes sur ce marché comme Apple avec son logiciel QuickTime[2] ou Canon avec le logiciel PhotoStitch[3] qui permettent un accès facile à la création d'image panoramique. Il existe également plusieurs méthodes pour capturer des panoramas. La solution la plus simple consiste à utiliser un appareil photo classique monté sur un trépied et en faisant tourner manuellement l'appareil photo autour de son centre optique.

1 http://user.cs.tu-berlin.de/~nowozin/autopano-sift/
http://autostitch.softonic.fr/
http://www.easypano.com/
2 http://www.apple.com/fr/quicktime/
3 http://www.canon.fr/

Il existe deux types de méthodes pour obtenir un panorama : les méthodes visant à augmenter le champ visuel de la caméra à l'aide d'un système optique (systèmes omnidirectionnels) et les méthodes visant à reconstruire le panorama à partir d'une série d'images. Les systèmes omnidirectionnels permettent d'obtenir une vue panoramique en une seule image, tandis que les mosaïques d'images nécessitent d'aligner les différentes images de la série, prises à chaque position de la caméra, pour reconstruire le panorama.

Dans le présent travail, nous nous intéressons à la fabrication de mosaïques d'images.
La construction d'une mosaïque à partir d'une série d'images nécessite un recalage géométrique approprié des repères des images, afin de les mettre dans le même système de coordonnées. Dans le cas où le centre optique est immobile, deux images sont reliées par une seule transformation homographique 2D (transformation projective). Il est donc nécessaire d'estimer cette transformation pour aligner les deux repères des images données.
La transformation est ensuite utilisée pour déformer une image afin de l'amener dans le même système de coordonnées que l'autre. En fusionnant les régions de recouvrement de l'image déformée avec celles de l'autre image, il est possible de construire la mosaïque.

Figure 1.1:*Exemple de représentation panoramique (15 images)*

La construction et l'exploitation d'une image panoramique nécessitent 4 grandes étapes : l'acquisition, la projection des prises de vue dans l'image panoramique, l'amélioration du rendu et la visualisation. Nous allons présenter rapidement ces quatre points.

1.2 Acquisition

La formation d'image panoramique est un exercice particulier qui requiert un nombre d'images différent selon la technologie employée et le but final de l'application. Dans tous les cas, la méthode qui doit être utilisée pour la construction de l'image panoramique est fonction des paramètres suivants : résolution, couverture visuelle et temps d'acquisition.
L'utilisation de caméra omnidirectionnelle est fréquemment utilisée dans la robotique. Le but ici est d'obtenir une vue la plus large possible dans le périmètre immédiat du capteur. Les caméras omnidirectionnelles sont à privilégier lorsque le temps d'acquisition est inférieur à la seconde et que la résolution ou plus précisément la profondeur de champ ne sont pas très importants. Avec ces modèles de caméras, sur le plan horizontal, le champ couvert peut aller sans grande difficulté jusqu'à 360°. Par contre, sur le plan vertical, le champ couvert dépasse plus rarement 90°. Avec une caméra unique en rotation autour de son centre optique le champ couvert peut aller jusqu'à 360° sur le plan horizontal et 180° sur le plan vertical. Il dépend essentiellement de la conception mécanique de la mise en rotation de la caméra. La résolution peut être très importante en fonction de la distance focale de l'objectif. Par contre le temps d'acquisition est alors très long et dépend finalement du nombre d'images nécessaires pour une couverture optimale de la scène à observer. Une solution intermédiaire consiste à utiliser ce que l'on appelle traditionnellement un

« bouquet » de caméra. Il s'agit d'un ensemble de caméras installées autour d'un axe. Le bouquet de caméra est un compromis intéressant puisqu'il permet d'augmenter la résolution tout en gardant l'aspect temps réel de l'acquisition. Même si le champ couvert peut être de 360° sur le plan horizontal et 180° sur le plan vertical, des contraintes mécaniques limitent généralement celui ci. De plus cette solution est relativement coûteuse. Outre le nombre plus important de caméras, cette solution nécessite des ressources en matériels plus importantes (carte d'acquisition, processeur, mémoire …). Le système GeoView 3000[4] de la société iMove utilise 6 appareils photo, quatre sur le plan horizontal, un vers le bas et un vers le haut.

1.3 Traitement de l'image

Les images capturées par les dispositifs panoramiques vont devoir subir un certain nombre de traitements avant de pouvoir être exploitées. Dans le cas des dispositifs omnidirectionnels, l'image est très déformée. Dans les autres dispositifs, une image ne contient qu'une partie de la scène. La représentation complète de la scène nécessite donc plusieurs images qui vont être projetées sur une image panoramique. Les images ne sont donc pas forcement acquises au même moment ou avec le même capteur. Si les informations de prise de vue ne sont pas précises ou si elles ne sont pas connues, une étape de mise en correspondance sera également nécessaire. Dans le cas où une seule caméra est utilisée, les différentes images formant le panorama ne sont pas prises en même temps. Les conditions d'illumination peuvent être différentes. De même, la caméra peut disposer d'une correction automatique la luminosité de façon à ajuster le gain du capteur. Les images devront alors subir un traitement pour supprimer les effets de couture[5]. Ce cas se retrouve également avec les bouquets de caméra. Pour simplifier, les traitements sont : étalonnage du dispositif, correction des aberrations de l'objectif, réduction du bruit, recalage des images, correction du gain et projection. Dans le travail de thèse que nous présentons, nous n'aborderons pas la phase d'étalonnage et de correction des aberrations de l'objectif de notre matériel. Nous avons repris des techniques classiques décrites dans la littérature [LUO97, BEN01,HAR04]. Nous allons par contre étudier longuement la phase de mise en correspondance, la correction du gain et la projection.

1.3.1 Mise en correspondance

Une large part de notre travail a été consacré à ce problème de mise en correspondance. Initialement, nous n'avions pas prévu d'aborder ce sujet. Dans notre cas, le modèle de transformation mathématique était clairement identifié et les paramètres de prise de vue donnés par notre caméra devaient nous permettre de calculer cette transformation.
Cependant, la précision de ces paramètres ne s'est pas révélée suffisante et ne nous a pas permis de réaliser des panoramas robustes.
La mise en correspondance est un problème difficile qui continue à mobiliser la communauté scientifique. Plusieurs solutions sont proposées dans la littérature et de nombreux articles proposent une synthèse de ces méthodes [BRO92, ZIT05]. Le principe de base consiste à mettre en correspondance des zones de l'image en fonction de leurs propriétés radiométriques ou géométrique. Les méthodes sont généralement classées en deux catégories : denses ou éparses.

Les méthodes denses cherchent à estimer les paramètres de transformation en exploitant l'intensité de l'ensemble des pixels contenu dans la zone de recouvrement. Les méthodes éparses

4 http://www.imoveinc.com
5 Couture : variation brusque de la luminosité le long de la frontière entre des images adjacentes

n'utilisent pas tous les points mais seulement quelques points particuliers (coins, extrema locaux) ou des primitives géométriques (lignes, cercles…).

Dans le cas où il n'y a pas de déformation des images, les méthodes denses sont réputées plus précises, mais elles sont souvent moins rapides, moins robustes aux changements de luminosité et à la présence d'objet en mouvements.

1.3.2 Correction de gain

Le contrôle de la luminosité n'est pas un problème aussi simple qu'il y paraît. Si la luminosité est relativement constante dans toutes les directions alors il suffit de fixer manuellement le gain de la caméra pendant toute la durée de la prise de vue. Par contre, il ne faut pas que cette luminosité évolue pendant cette prise de vue (lors du passage d'un nuage par exemple).
Cependant, même si la luminosité générale n'évolue pas au cours de la prise de vue, si les écarts de luminosité dans la scène sont trop important, les zones sombres apparaîtront sous exposées dans la mosaïque d'image. A l'inverse, d'autres zones beaucoup plus lumineuses risquent de se trouver sur-exposées. La solution pourrait être de laisser la caméra gérer automatiquement le gain en fonction de la luminosité de la portion de scène visée. Cette solution offre bien sûr l'avantage que chaque image est acquise en optimisant la dynamique du capteur. Le problème est qu'une même portion de la scène prise avec deux gains différents n'a plus la même valeur de luminosité. L'exemple suivant est un cas d'école. La caméra est placée à l'intérieur d'une pièce. Cette pièce est éclairée par une lumière artificielle et par une lumière naturelle à travers une fenêtre. La luminosité à l'intérieur est faible par rapport à celle de l'extérieur. Si le gain de la caméra est fixé manuellement pour obtenir une luminosité correcte à l'intérieur de la pièce, la fenêtre apparaîtra sur-exposée et il ne sera pas possible de visualiser l'extérieur. Dans le cas contraire, si le gain est piloté automatiquement par la caméra, alors l'extérieur devient visible, mais le phénomène de « couture » apparaît.

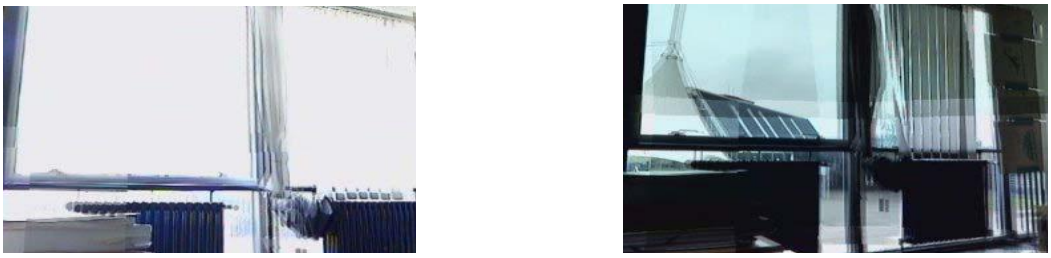


Figure 1.2:Image de gauche : Gain fixe, l'extérieur apparaît très largement saturée. Image de droite : Gain automatique, l'extérieur est visible mais on observe des « coutures »

1.4 Assemblage

La mise en correspondance des images est utilisée pour déterminer avec la meilleure précision possible les paramètres de la prise de vue. Cette information permet de projeter correctement l'image dans le panorama. Dans le cas général, une homographie est calculée à partir de surfaces planes contenues dans la zone de recouvrement des deux images [KAN99,SUG04]. Nous verrons que dans le cas particulier des caméras PTZ où le centre de projection est confondu avec

le centre des rotations, tous les points de la zone de recouvrement peuvent être utilisés pour calculer l'homographie, même s'ils n'appartiennent pas à des surfaces planes.
Avant toute chose, nous devons définir le modèle de représentation que nous allons utiliser.
Lorsqu'il s'agit de fusionner quelques images de façon à obtenir une image globale plus large, il suffit de prendre une image de référence et de projeter toutes les images dans le plan cette image de référence. Lorsque l'angle solide devient trop important, les déformations inhérentes à la projection sont importantes. Au delà d'une certaine limite, l'homographie ne peut plus être calculée où du moins ne permet pas une projection conforme. Il est donc nécessaire d'utiliser d'autres modes de projection. Nous classons ces projections en deux catégories. Les projections planes et la projection sur polyèdre. Les projections planes permettent de visualiser l'ensemble de la scène sur une seule image mais elles entraînent des déformations importantes. Parmi les projections planes, la plus utilisée est la projection cylindrique. Un autre mode de représentation est la projection de la scène sur un polyèdre. Ces projections ne permettent pas la représentation de la scène en une seule image mais elles limitent les déformations ainsi que le coût du stockage.

1.5 Immersion et visualisation

La visualisation est souvent l'étape ultime du mosaïquage. Bien que les techniques de base soient connues depuis très longtemps, ce n'est que récemment que les mosaïques d'images servent à autre chose qu'à la visualisation. Cependant, cela reste pour beaucoup, le seul intérêt de ces constructions. Comme nous venons de le voir, la projection d'une image sur un cylindre permet de visualiser l'image à partir d'une seule vue avec une limitation au niveau des pôles mais surtout en déformant la scène. Une fois le mosaïquage réalisé, plusieurs logiciels permettent de visualiser une portion de la scène en redressant l'image dans les mêmes conditions que si elle avait été acquise par la caméra. Il est alors possible de se déplacer dans la scène et d'effectuer des changements de focale afin d'augmenter ou diminuer l'angle de visualisation. Le logiciel le plus répandu est QuickTimeVr. L'inconvénient de la plupart des logiciels est qu'ils ne gèrent pas l'aspect multi-résolution ou alors indirectement. Ce que nous entendons par « gestion de la multi résolution » est la possibilité de ne pas stocker l'ensemble du panorama avec la même résolution. En effet, l'ensemble de la scène n'est pas forcément d'un intérêt égal. Lorsque nous réalisons un panorama, nous aimerions avoir plus de détail sur certaines zones de l'image et moins sur d'autre (le ciel par exemple). Avec la plupart des logiciels disponibles, c'est le niveau de détail le plus fin qui conditionne la résolution du panorama tout entier. Le temps de calcul et surtout la taille du panorama ne sont pas optimisés. Cela peut poser un problème lorsque l'on transmet un panorama sur le réseau. C'est la raison pour laquelle, certains logiciels gèrent la multi-résolution en utilisant plusieurs panoramas complets de la même scène avec des niveaux de résolutions différents. Lors d'une transmission sur Internet, cette solution permet rapidement d'obtenir une première représentation de la scène à basse résolution de façon à ce que l'utilisateur puisse naviguer. Les résolutions plus élevées arrivant ensuite au gré du débit de la connexion, mais uniquement pour les zones observées.

1.6 Caméras et applications

Pendant longtemps, les caméras IP ont eu une mauvaise réputation. Il est vrai que sur les premières caméras IP la qualité de la vidéo n'avait rien à voir avec les caméras analogiques.

Les progrès réalisés sur la taille des capteurs, la qualité de la compression et la bande passante ont grandement amélioré le rendu des caméras IP. Ces dernières années ont vu l'essor d'une nouvelle classe de caméras IP, les caméras PTZ. Des caméras PTZ analogiques existent depuis presque aussi longtemps que les caméras analogiques fixes, mais la baisse des coûts de fabrication à permis une démocratisation de ces caméras. Du coup, les applications se sont multipliées. L'un des objectifs de ce travail de thèse, en accord avec la société **SatXpro**, est d'étudier les applications de détection et de suivi automatique des objets en mouvement dans une scène à partir d'une caméra PTZ. L'intérêt étant de sécuriser une zone étendue en limitant le nombre de caméras.

Figure 1.3: Caméra Sony

Les principales caractéristiques de cette caméra sont :

- Caméra réseau
- Optique SONY SUPERHAD 1/6
- Caméra motorisée Pan 360° Tilt +30° à -90°
- Caisson thermostaté (-45° à +50°)
- Affichage 480 lignes
- Zoom optique X22
- 16 Preset

1.7 Conclusion

Nous avons présenté les principaux étapes qui contribués à La construction et l'exploitation d'une image panoramique. Dans le chapitre suivant nous présenterons les différentes techniques qui vont nous permettre de construire une mosaïque d'images.

Chapitre 2

Construction d'image mosaïque

2.1 Présentation

Une mosaïque d'images est une collection d'images prises suivant des angles de vue différents et ramenées à un même repère. Un panorama est une représentation d'une mosaïque d'images, permettant de voir l'intégralité d'une scène à 360° voir à 4π stéradians. Le terme panorama est dérivée du grec est signifie « tout-voir ». L'un des premiers panoramas répertoriés a été réalisé par Robert Barker. Il fit d'ailleurs breveter en 1787 un dispositif qu'il nomma «La nature d'un coup d'oeil» où les spectateurs, placés sur une estrade, se trouvent au coeur d'un paysage ou d'un champ de bataille.
En décembre 1900, Louis Lumière dépose le brevet du Photorama. Il s'agit d'un procédé photographique permettant de prendre, en une seule prise de vue, une scène sur 360°. Le brevet inclut la projection intégrale de ce panorama sur un cylindre. Les spectateurs sont, là aussi, placés au centre de l'estrade.

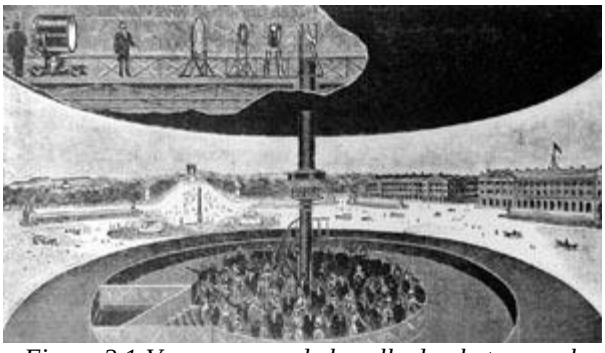


Figure 2.1:Vue en coupe de la salle du photorama lumière.

Crédit photo : http://www.institut-lumiere.org

Les développements des techniques de visualisation de ces dix dernières années et notamment les travaux de Chen [CHE95] et de Szeliski [SZE94] ont permis la création et l'exploitation de panoramas virtuels à partir d'ordinateurs personnels. Il existe aujourd'hui plusieurs logiciels commerciaux permettant de créer un panorama à partir de simples photos en quelques clics de souris. Les utilisateurs de matériel Canon connaissent le logiciel PhotoStitch. Mais il existe bien d'autres logiciels disponibles sur internet comme PanoramaFactory, PanoramaComposer, AutoStitch ou PanoramaTools pour les utilisateurs avancés.
La création d'un panorama ou d'une mosaïque d'images de façon automatique et en contrôlant éventuellement les conditions de prise de vue est un défi qui motive les industriels comme le monde de la recherche. Comme il est précisé dans [TAN04], nous pouvons classer les méthodes

de construction d'une mosaïque d'images en deux catégories : dioptrique et catadioptrique. Dans les méthodes dioptriques seuls des éléments réfractifs (lentilles) sont utilisés. Dans les méthodes catadioptriques, des éléments réflectifs (miroirs) vont être ajoutés. Parmi les méthodes dioptriques, nous allons retrouver les ensembles de caméras (bouquet), les lentilles panoramiques [NAY97], les lentilles « fish-eyes » [XIO97], les caméras linéaires [DOU03] ou plus classiquement les caméras rotatives. Dans les méthodes catadioptriques, nous trouvons généralement une caméra associée à un miroir conique [YAG90], sphérique [HON91], parabolique [STU02] ou a double courbure [SOU96, FIA02], ou alors plusieurs caméras associées à des miroirs plans [TAN04].

Les applications sont diverses. Chen [CHE95] est l'un des premiers à utiliser un panorama cylindrique à 360° pour représenter une scène et permettre la navigation en temps réel dans une application de réalité virtuelle. Son approche a été utilisée dans le produit commercial QuickTimeVR.

Pour Lee [LEE99], la construction du panorama sert de support à la compression et à la transmission d'un flux vidéo. Le panorama correspondant aux composantes statiques de la scène est transmis une fois. Ensuite, les objets en mouvement sont extraits de la vidéo et transmis séparément. La première étape de leur approche consiste à créer le panorama en projetant les images sur un cylindre. Les auteurs utilisent pour cela le logiciel PhotoVista™.

Une importante limitation de leur approche est que dans cette première étape, les images acquises ne doivent comporter que les composantes statiques du fond. Une fois cette étape réalisée, ils utilisent la mosaïque d'image pour segmenter les objets en mouvement. Lors de cette deuxième phase, les auteurs doivent estimer la position de la camera de façon à reconstruire l'image de fond utilisée pour la segmentation. Pour réaliser ce calcul, leur méthode nécessite de sélectionner manuellement et préalablement dans le panorama des petits blocs carrés qui ne devront pas être occultés pendant le mouvement. Ces blocs sont alors recherchés dans l'image courante.

Hsu et Anandan [HSU96] décrivent ce qu'ils appellent la compression basée sur la mosaïque d'image (MBC pour Mosaic-Based Compression) et décrivent plusieurs types de représentations permettant d'éviter la redondance d'informations dans les données vidéo.

Douze et al. [DOU02, DOU03] proposent une méthode de suivi robuste dans une séquence vidéo en utilisant un panorama. L'originalité de leur approche repose sur l'utilisation d'une caméra linéaire de leur fabrication. Cette caméra leur permet rapidement, et sans calcul particulier, de construire le panorama. Une caméra classique est ensuite utilisée pour le suivi.

Les auteurs proposent une solution de suivi à base de point contrôle et d'un modèle homographique.

Kang et al [KAN03] utilisent une caméra PTZ pour réaliser le suivi des personnes. La première étape de leur algorithme consiste à construire un panorama sans les objets en mouvement et à sélectionner des points statiques de la scène de façon à ce que, quel que soit le déplacement de la caméra et les objets en mouvement dans la scène, ils puissent en retrouver au moins quatre dans l'image et ainsi calculer correctement l'homographie. Dans la phase d'exploitation, ils recherchent les points clés dans l'image, calculent l'homographie et par simple soustraction de l'image courante et de l'image du fond, ils déterminent les pixels en mouvement. Bartoli et all. [BAR03] utilisent une technique similaire pour créer le panorama du mouvement d'une séquence d'images. Le principe réside dans la construction d'une couche statique et d'une couche dynamique et ceci en deux étapes. La première étape consiste à créer un modèle de fond sous la forme d'un panorama. Ce fond ne contient alors que les composantes statiques de la scène. Dans une

deuxième phase, chaque image de la séquence est projetée dans le panorama statique et comparée avec le modèle de fond de façon à extraire les objets en mouvement.
D'autres applications sont plus marginales et sortent du cadre de la visualisation. Peer et al. [PEE02] présentent un système permettant de déterminer la carte de profondeur d'une image panoramique. Ils utilisent pour cela une caméra placée sur un bras tournant. En raison du décalage entre le centre optique de la caméra et l'axe de rotation du bras, les auteurs peuvent ainsi estimer l'effet de parallaxe sur deux images prises suivant deux angles différents. Ceci rend donc possible la reconstruction stéréo. Pour Sato [SAT04] la mosaïque d'images peut servir de support à l'analyse de document.
Si le centre optique de la caméra est confondu avec les axes de rotations, le monde vu par la caméra peut être considéré comme une sphère dont le centre est justement le centre optique.
Si deux points sont alignés avec le centre de rotation de la caméra et que le centre optique est confondu avec le centre de rotation, quel que soit l'angle de rotation que l'on fait subir à la caméra les deux points restent alignés avec le centre optique puisque ce dernier ne se déplace pas. Ceci implique que quel que soit l'angle de rotation, un seul rayon passe par ces deux points et que donc ces deux points sont projetés sur un point unique dans l'image. En conséquence, lorsque le centre optique de la caméra est confondu avec le (ou les) centre(s) de rotation, il n'est pas possible d'obtenir une mesure de la profondeur par triangulation même en multipliant les prises de vue pour des angles différents. Nous pouvons donc considérer que tous les points sont à la même distance du centre c'est à dire sur la surface d'une sphère de rayon fixé.

2.2 Construction d'une mosaïque d'images

Nous allons aborder dans ce chapitre les différentes techniques qui vont nous permettre de construire une mosaïque d'images dans le cas idéal. C'est à dire que la caméra peut être modélisée avec le modèle sténopé que nous présentons en annexe et que le centre de projection est confondu avec le centre des rotations, que les paramètres de la prise de vue sont connus avec précision, qu'il n'y a pas de distorsion dans l'image et que la luminosité est constante dans toute la scène. Nous pouvons déjà imaginer que la réalité sera quelque peu éloignée de ce cas idéal.

2.2.1 Projection d'un point dans l'image

Si l'on se place dans les conditions idéales décrites ci-dessous, nous pouvons établir les équations qui relient un point quelconque de la scène et sa projection dans le plan image. Soit un point *P* quelconque du monde dans le repère de la caméra qui se projette en un point *p* sur le plan image. En fonction de la distance *f* entre le centre optique Oc et le plan image, les coordonnées 2D (u, v) de ce point de l'image peuvent s'exprimer à partir de deux angles (θ_P et ϕ_P) comme représenté sur la figure ci-dessous :

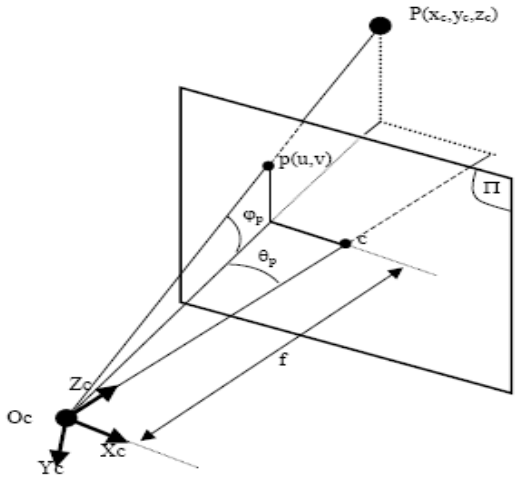


Figure 2.2:Schéma de la projection d'un point du monde dans l'image

Les relations qui relient les coordonnées (u, v) d'un point aux angles (θ_p et ϕ_p) en fonction de f sont données par :

$$\theta_p = \tan^{-1}\left(\frac{u}{f}\right) \text{ et } \varphi_p = \tan^{-1}\left(\frac{v}{\sqrt{v^2 + f^2}}\right) \qquad (2.1)$$

Dans la pratique, f représente la distance focale exprimée en unités pixels (up).

2.2.2 Unité pixel

L'unité pixel (notée up) que nous utilisons n'a pas de réalité physique. Cependant, nous l'utilisons pour faciliter l'écriture des équations. Nous partons de l'hypothèse que les cellules composant le capteur CCD sont rangées sous la forme d'une matrice rectangulaire comme les pixels de l'image et qu'il y a le même nombre de pixels que de cellules. Dans ce cas, l'unité pixel correspond à la taille d'une cellule sur le capteur.

$$1up(mm/pixel) = \frac{\text{largeur du capteur (mm)}}{\text{Nombre de coloones de l'image}}$$

Dans la plupart des applications nous faisons également l'hypothèse que les pixels sont carrés. Dans la réalité, les cellules composants le capteur ne sont pas carrées mais ont plutôt une forme rectangulaire. L'unité pixel n'a donc pas la même valeur suivant les deux axes.
Dans le reste de ce manuscrit, l'unité pixel que nous utilisons est identique selon les deux axes.

2.2.3 Relation entre deux images

Nous pouvons également utiliser cette expression pour représenter une autre image prise avec un angle de vue différent. Soit deux plans images π_1 et π_2 et un rayon passant par le centre optique Oc qui intersecte ces deux plans respectivement en deux points p_1 et p_2. θ_1 et θ_2 représentent les

angles de rotation panoramique autour de l'axe Y des axes optiques des deux images et ϕ_1 et ϕ_2 les angles d'inclinations des deux axes.

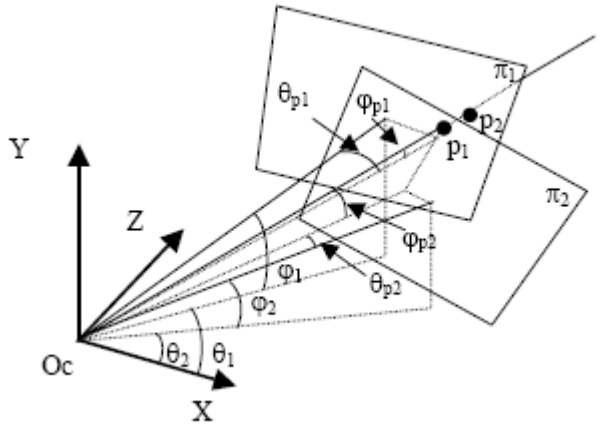


Figure 2.3:Schéma de la relation entre deux images

Nous avons donc :

$X_1 = R_{\phi 1}.R_{\theta 1}.X$ et $X_2 = R_{\phi 2}.R_{\theta 2}.X$ (2.2)

Nous en déduisons que :

$$X_1 = R_{\varphi 1}.R_{\theta 1}.R_{\theta 2}^{-1}.R_{\theta 2}^{-1}X_2 \qquad (2.3)$$

Les matrices $R_{\theta 2}$ et $R_{\phi 2}$ sont orthogonales, donc $R_{\varphi 2}^{-1} = R_{\varphi 2}^{T}$ et $R_{\theta 2}^{-1} = R_{\theta 2}^{T}$

La relation qui relie les angles θ_{p1} et ϕ_{p1} de l'image 1 aux angles θ_{p2} et ϕ_{p2} de l'image 2 d'un point P s'exprime de la façon suivante :

$$\begin{bmatrix} \cos\varphi_{p1}.\sin\theta_{p1} \\ \sin\varphi_{p1} \\ \cos\varphi_{p1}.\cos\theta_{p1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_1 & -\sin\varphi_1 \\ 0 & \sin\varphi_1 & \cos\varphi_1 \end{bmatrix} * \begin{bmatrix} \cos(\theta_1-\theta_2) & 0 & -\sin(\theta_1-\theta_2) \\ 0 & 1 & 0 \\ \sin(\theta_1-\theta_2) & 0 & \cos(\theta_1-\theta_2) \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_2 & \sin\varphi_2 \\ 0 & -\sin\varphi_2 & \cos\varphi_2 \end{bmatrix} * \begin{bmatrix} \cos\varphi_{p2}.\sin\theta_{p2} \\ \sin\varphi_{p1} \\ \cos\varphi_{p2}.\cos\theta_{p2} \end{bmatrix}$$

Cette expression peut se résumer de la façon suivante :

$$\begin{bmatrix} \cos\varphi_{p1}.\sin\theta_{p1} \\ \sin\varphi_{p1} \\ \cos\varphi_{p1}.\cos\theta_{p1} \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \qquad (2.4)$$

Ou A, B et C sont les coefficients de la matrice ligne résultant du calcul numérique de l'expression précédente. A partir de cette relation, il est facile de retrouver les angles θ_{p1} et φ_{p1}

$\varphi_{p1} = \sin^{-1}(B)$ et $\theta_{p1} = \tan^{-1}\left(\frac{A}{C}\right)$ pour $C \neq 0$

Ces deux angles permettent à présent de calculer les coordonnées (u_1, v_1) du point p dans l'image 1 :

$$u_1 = f.\tan\theta_{p1}, \quad v_1 = \tan\varphi_{p1}.\sqrt{u_1^2 + f^2}$$

2.2.4 Homographie entre deux images

Dans le paragraphe précédent, nous avons présenté une relation permettant de calculer la projection d'un pixel d'une image 1 dans une image 2. Cependant, si le centre des rotations de la caméra est confondu avec le centre optique (i.e. pas de translation de la caméra) et que le centre optique correspond à l'origine du repère de la caméra, alors, la relation qui relie deux images peut s'exprimer sous la forme d'une transformation homographique H
[MAN94]. Une homographie 2D s'exprime avec 8 coefficients puisqu'elle est définie à un facteur d'échelle près (cf. annexe).

$$X_1 \sim \mathrm{H}\ .\ X_2 = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \bullet \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \qquad (2.5)$$

où X_1 est un point de l'image défini par ses coordonnées homogènes (u_1, v_1, 1) dans l'image et X_2 sa projection homographique dans l'image 2 est défini par ses coordonnées homogènes
(u_2, v_2, 1). Le signe ~ indique la relation d'équivalence, à un facteur d'échelle près, entre le point X_1 et la transformation du point X_2.

$$u_1 = \frac{m_0.u_2 + m_1.v_2 + m_2}{m_{.6}u_2 + m_7.v_2 + 1},\ u_1 = \frac{m_3.u_2 + m_4.v_2 + m_5}{m_6.u_2 + m_7.v_2 + 1}, \qquad (2.6)$$

Les coordonnées (u_1, v_1) du point X_1 dans l'image 1 se déduisent simplement :
Il existe plusieurs solutions permettant de déterminer la matrice H en fonction des données que l'on possède. La matrice H contenant 8 coefficients, cela signifie que seulement 4 points sont nécessaires pour résoudre le système linéaire. Ces quatre points minimum pourront être saisis par l'utilisateur ou obtenus de façon automatique avec un détecteur de Harris ou d'autres détecteurs plus robustes [TOR96, LOW04]. Lorsque le nombre d'équations est supérieur au nombre d'inconnues, le système n'a en général par de solution exacte et une solution approchée peut être obtenue au sens des moindres carrés.

Il est également possible de déterminer les coefficients de la matrice de transformation à partir des paramètres intrinsèques et extrinsèques des prises de vue des deux images. Dans ce cas, la relation entre un point P de coordonnées (x, y, z) dans le repère de la caméra et un point X de coordonnée homogène (u, v, 1) dans le plan image, s'écrit :

$$X \sim T.K.R.P \qquad (2.7)$$

où

$T=\begin{bmatrix} 1 & 0 & u_{10} \\ 0 & 1 & v_{10} \\ 0 & 0 & 1 \end{bmatrix}$ est la matrice de changement d'unité et de translation du repère image,

$K=\begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ est la matrice de changement d'échelle,

et $R=\begin{bmatrix} r_p \end{bmatrix}$ la matrice de rotation 3D.

Cette relation entre X et P correspond à un modèle simplifié de la projection perspective pour lequel nous considérons que d'une part le repère image est orthogonal (i.e. les pixels du capteur CCD sont rectangulaires) et que d'autre part, il n'y a pas de distorsion de l'image.
Pour simplifier encore les équations, Szeliski [SZE97] fait l'hypothèse que le centre optique de la caméra passe par l'origine de l'image (i.e. $u_0 = v_0 = 0$) et que l'origine de l'image correspond au centre de l'image. Il fait en outre remarquer qu'un léger décalage entre l'axe optique et le centre de l'image n'a que peu d'incidence sur le résultat final. La relation simplifiée s'écrit alors :

$$X \sim K.R.p.$$

La projection perspective d'un même point p du repère caméra donne $Xl \sim K_l.R_l.p$ dans une image l et $X_2 \sim V_2.R_2.p$ dans une image 2. On en déduit rapidement la relation homographique entre les deux images :

$$H \sim V_2.R_2.R_1^{-1}.V_1^{-1} \qquad (2.8)$$

2.3 Visualisation des mosaïques d'images

La création d'une mosaïque d'images consiste à représenter sur une seule structure de données la vue d'ensemble de la caméra. A partir d'une mosaïque d'images, les applications sont nombreuses. Quelle que soit l'application, l'une des finalités les plus courantes reste la restitution d'une prise de vue particulière qui n'aurait pas été directement acquise. Plus concrètement, à partir de deux images prises avec des angles de vue différents, il doit être possible de calculer l'image équivalente à une prise de vue intermédiaire, pour peu qu'il y ait recouvrement partiel entre les deux images de départ.
Dans le cas d'une projection centrale et où le centre optique de la caméra est confondu avec les axes de rotations, le monde vu par la caméra peut être considéré comme une sphère.
Nous cherchons donc à exprimer le nombre de pixels optimal permettant la représentation de cette sphère qui limite la perte d'information en fonction de la résolution du capteur et de la distance focale. Donc en fonction de la taille du capteur et de la distance focale, un pixel de l'image couvre un certain angle solide. Une approximation de cet angle solide est donnée par :

$$\Omega = \frac{l.h}{f^2}$$

où l est la largeur du pixel, h sa hauteur et f la distance focale. Cette approximation est valable si l et h sont très inférieurs à f. Pour un capteur 1/3", l et h sont de l'ordre du micromètre alors que f est de l'ordre du millimètre. Il y a donc un rapport 1000 entre la taille du capteur et la distance focale. Exprimé en unité pixel, l'angle solide est donné par :

$$\Omega(up) = \frac{1}{f^2}$$

L'angle solide d'une sphère étant de 4πsr, ceci implique que le nombre de pixels nécessaire à la représentation de la sphère est donnée par :

$$nb\ pixels = 4.\pi\ .\ f^2$$

Ce nombre de pixel est une approximation dans laquelle nous considérons que chaque pixel représente une unité de surface. Nous pouvons donc en déduire que pour représenter un panorama en limitant les pertes d'information, nous devons utiliser une sphère de rayon égale à la distance focale exprimée en pixels. Si nous utilisons une sphère plus petite, nous allons concentrer l'information. Si nous utilisons une sphère plus grande, nous n'aurons pas assez de résolution pour un pavage complet.

Pour simplifier le problème, nous avons calculé l'angle solide couvert par le pixel du centre de l'image ou plus exactement du pixel situé à l'intersection du plan image et de l'axe optique.
L'angle solide de n'importe quel pixel du plan image perpendiculaire à l'axe optique est donnée par :

$$\Omega = \cos\theta.\sin\varphi\frac{l.h}{f^2}$$

où θ et ϕ sont les angles entre l'axe optique et la droite passant par le centre optique et le centre du pixel. En définitive, ce sont donc les pixels des coins de l'image qui couvrent l'angle solide le plus faible. En toute rigueur nous devrions donc utiliser cette expression.
Avec un capteur 1/3", une focale de 4.1mm et une résolution d'image de 640 x 480, l'erreur sur le calcul de l'angle solide est de l'ordre de 11%. Elle n'est plus que de 3% dans les mêmes conditions mais avec une distance focale deux fois plus grande.
La résolution optimale de la représentation du monde vue par la caméra qui limite autant que faire se peut les pertes d'informations, est une sphère dont le rayon est de l'ordre de la distance focale exprimée en pixels. Les deux défis à relever sont d'une part la complexité algorithmique nécessaire au calcul de cette représentation et d'autre part la minimisation de l'espace de stockage. L'espace de stockage optimal est simple à déterminer. Il correspond simplement, comme nous l'avons vu, à la surface de la sphère exprimée en nombre de pixels.

$$s = 4.\pi\ .\ f^2$$

Le mode de représentation le plus classique est la projection sur un cylindre. Cette projection offre notamment la possibilité de visualiser en une seule image l'ensemble de la scène. Par contre la représentation des pôles est impossible et la complexité algorithmique est la même que la sphère puisqu'il est nécessaire de calculer la projection de chaque pixel.
Afin d'optimiser le temps de calcul, nous recherchons une représentation à base de faces planes de façon à pouvoir calculer une matrice de projection pour un ensemble de pixels.

2.4 Représentation plane

Nous regroupons sous le terme « représentation plane », différentes représentations d'un panorama permettant de visualiser l'ensemble de la scène en une seule image. Ce type de représentation est réalisé à partir d'une projection par développement. Comme il n'est pas possible d'étendre la surface d'une sphère, ou d'un ellipsoïde en général, sur un plan sans

déchirure ni déformation, la représentation ne pourra pas être rigoureuse. La représentation la plus utilisée est la projection de la scène sur un cylindre que l'on va dérouler. Nous allons rapidement présenter quelques unes des ces projections ayant un intérêt dans le cas d'une mosaïque d'images.

2.4.1 Projection cylindrique

La projection cylindrique est le mode de représentation le plus couramment utilisé pour la visualisation. Les auteurs préfèrent utiliser une projection sur les faces d'un cube. Cependant, pour la visualisation, la projection cylindrique permet de visualiser l'ensemble de la scène en une seule image. Les déformations sont bien sûr importantes mais l'œil humain et surtout le cerveau arrivent à reconstituer mentalement la scène. Le principe de ce type de projection consiste à projeter la sphère sur un cylindre tangent à la sphère.

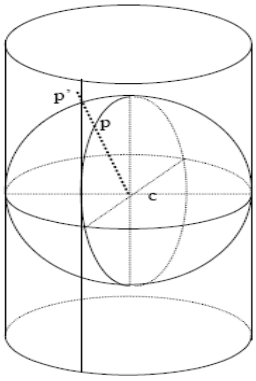


Figure 2.4:Principe de la projection cylindrique

L'équation de la projection cylindrique classique est la suivante :

$$\begin{cases} X = \theta \\ Y = \tan\varphi \end{cases}$$

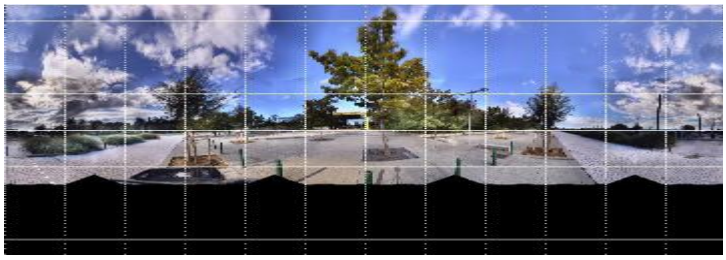


Figure 2.5: Projection cylindrique avec –63° < ϕ < 63°

Visuellement cette solution est intéressante puisqu'elle permet en une seule image de représenter l'ensemble de la scène. Cependant, elle engendre deux inconvénients principaux.
Le premier est qu'elle déforme les lignes droites. Le deuxième est qu'elle ne permet pas une définition optimale pour des angles de tangage proche des pôles. Au delà de 60°, le coût de stockage d'un stéradian devient trop important.

2.4.1.1 Projection plate carré cylindrique équidistante (PPCE)

Cette projection a été utilisée pour la première fois par Anaximandre vers 550 avant notre ère. L'équation utilisée est donnée ci-dessous. Comme nous pouvons le remarquer elle est extrêmement simple à mettre en œuvre :

$$\begin{cases} X = \theta \\ Y = \varphi \end{cases}$$

Par rapport à la projection cylindrique classique, celle-ci permet une représentation des pôles pour un coût de stockage raisonnable. Cependant, elle déforme également les droites horizontales. Parmi les représentations cylindriques, c'est la représentation qui est généralement utilisée pour représenter une mosaïque d'images.

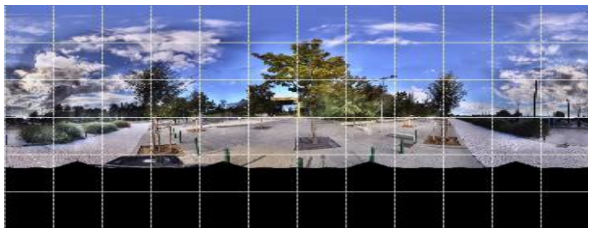


Figure 2.6 : Représentation cylindrique équidistante

2.4.2 Projection azimutales

Les projections azimutales sont des projections au sens mathématique du terme, qui transforment les méridiens en rayons également espacés. La projection s'opère à partir d'un centre de projection sur un plan tangent à la sphère.

2.4.2.1 Projection gnomonique

Cette projection n'est ni conforme, ni équivalente, c'est à dire qu'elle ne conserve respectivement ni les angles ni les surfaces. Elle est réalisée en projetant la sphère sur un plan **Π** tangent en un point p de la sphère et en utilisant la projection radiale centrée sur le centre de la sphère c. Dans le cas d'une représentation de la terre, si le pôle nord est le point de tangence alors l'équateur est renvoyé à l'infini. Cette représentation ne pourra pas être utilisée pour représenter la scène complète en une seule image. Par contre, elle est souvent utilisée dans la navigation parce que l'une des ses propriétés intéressante est qu'elle conserve les droites. Une droite sur la carte correspond à une droite sur le terrain.

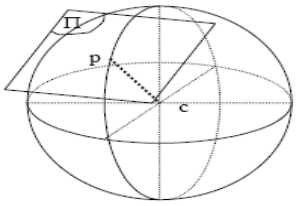


Figure 2.7:Principe de la projection gnomonique

La formule de la projection gnomonique est donnée ci dessous :

$$\begin{cases} X = \dfrac{\cos\varphi.\sin(\theta-\theta_0)}{\sin\varphi_0.\sin\varphi+\cos\varphi_0.\cos\varphi.\cos(\theta-\theta_0)} \\ Y = \dfrac{\cos\varphi_0.\sin\varphi-\sin\varphi_0.\cos\varphi.\cos(\theta-\theta_0)}{\sin\varphi_0.\sin\varphi+\cos\varphi_0.\cos\varphi.\cos(\theta-\theta_0)} \end{cases}$$

où (θ_0, ϕ_0) sont les coordonnées du point p de tangence du plan et la de sphère. La figure suivante donne une représentation de la projection gnomonique pour ($\theta_0 = 0°$, $\phi_0 = 0°$) :

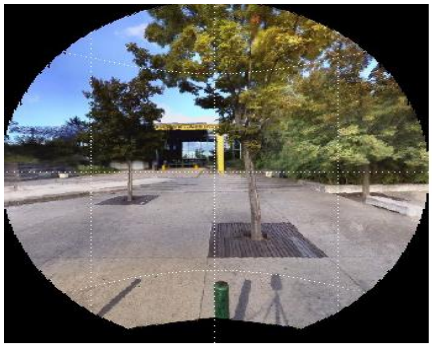


Figure 2.8: Projection gnomonique avec (θ0 = 0°, ϕ0 = 0°)

2.4.2.2 Projection stéréographique

Le principe de la projection stéréographique est le même que celui de la projection gnomonique. Dans le cas de la projection stéréographique, le centre c de projection est aux antipodes du point p de tangence. De plus c'est une transformation conforme, c'est-à-dire qu'elle conserve les angles.

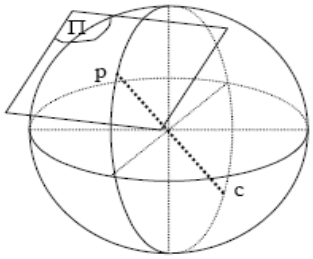


Figure 2.9: Principe de la projection stéréographique

La formule de la projection stéréographique est donnée ci dessous :

$$\begin{cases} X = 2.\dfrac{\cos\varphi.\sin(\theta-\theta_0)}{1+\sin\varphi_0.\sin\varphi+\cos\varphi_0.\cos\varphi.\cos(\theta-\theta_0)} \\ Y = 2.\dfrac{\cos\varphi_0.\sin\varphi-\sin\varphi_0.\cos\varphi.\cos(\theta-\theta_0)}{1+\sin\varphi_0.\sin\varphi+\cos\varphi_0.\cos\varphi.\cos(\theta-\theta_0)} \end{cases}$$

où (θ_0, ϕ_0) sont les coordonnées du point p de tangence du plan et de la sphère. Dans le cas d'une représentation de la scène vue par la caméra, l'intérêt de cette représentation est qu'elle permet de

visualiser l'ensemble de la scène à partir de 2 images. La figure suivante donne une représentation de la projection stéréographique pour ($\theta_0 = 0°$, $\phi_0 = 0°$) et ($\theta_0 = 180°$, $\phi_0 = 0°$).

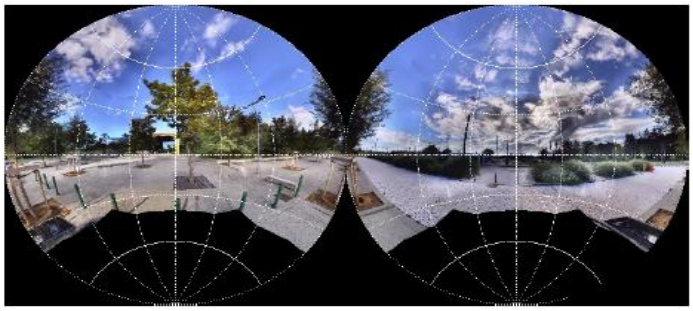


Figure 2.10: Projection stéréographique pour (θ0 = 0°, ϕ0 = 0°) et (θ0 = 180°,ϕ0 =0°)

La projection stéréographique a été imaginée 130 ans avant J.C. par Hipparque qui lui donna le nom de ***planisphère***. C'est en 1643 que le jésuite François Aiguillon la renomma ***projection stéréographique***.

2.4.2.3 Projection orthographique

Le principe de la projection orthographique est le même que les deux précédentes. Dans ce cas, le centre de projection est rejeté à l'infini. Cette projection n'est ni conforme, ni équivalente. L'équation de la projection orthographique est la suivante :

$$\begin{cases} X = \cos\varphi.\sin(\theta - \theta_0) \\ Y = \cos\varphi_0.\sin\varphi - \sin\varphi_0.\cos\varphi.\cos(\theta - \theta_0) \end{cases}$$

Cette représentation est souvent utilisée pour plaquer une texture sur une sphère. Elle permet de donner un « ***effet 3D*** » à l'image.

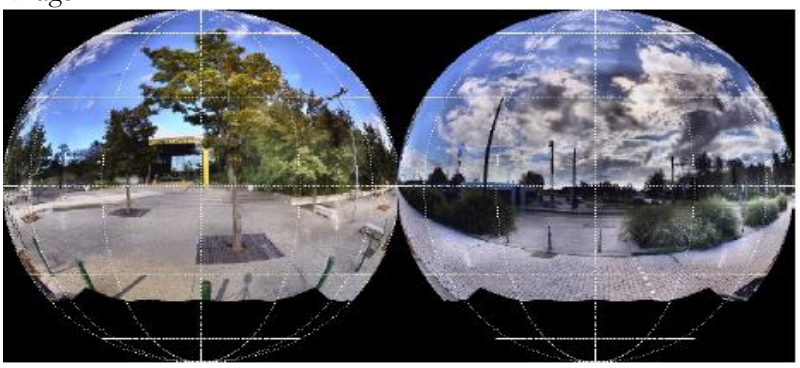


Figure 2.11: Projection orthographique pour (θ0 = 0°, ϕ0 = 0°) et (θ0 = 180°,ϕ0= 0°)

2.4.3 Conclusion

Les projections planes apportent une solution élégante et pertinente pour représenter une sphère sur un plan. Cependant, pour représenter une mosaïque d'images, les projections utilisées habituellement se limitent à la projection cylindrique classique avec le problème de la représentation des pôles et la projection plate carré cylindrique équidistante (PPCCE) proposée par Anaximandre. Ces projections présentent néanmoins quelques inconvénients.

Tout d'abord le coût de stockage pour une représentation sans perte d'informations est relativement important. Le stockage en mémoire vive n'est aujourd'hui plus vraiment un problème. Par contre le transfert du panorama à travers le réseau internet ou autre peut être un frein. De même, il n'en reste pas moins que la quantité d'informations à traiter est importante ce qui peut être pénalisant pour le traitement en temps réel. Nous pouvons calculer rapidement l'espace mémoire nécessaire pour stocker un panorama complet sur une PPCCE. Dans le cas d'une PPCCE, le cylindre est tangent à la sphère, c'est à dire que les pixels de l'équateur sont représentés sans déformation. Il en va de même pour les pixels le long du méridien de référence. Tous les autres pixels sont dilatés par la transformation mathématique. Donc les incréments $\Delta\theta$ et $\Delta\phi$ des angles respectivement θ et ϕ permettant une représentation limitant les pertes d'informations correspondent à l'angle solide du pixel situé sur l'axe de projection. Une approximation de ces angles est donnée par :

$$\theta = \varphi = 2.\tan^{-1}\left(\frac{1}{2.f}\right)$$

Une PPCE couvrant une scène complète de 2π x π nécessite une matrice rectangulaire de résolution $\frac{2.\pi}{\Delta\theta} \times \frac{\pi}{\Delta\varphi}$.

L'autre inconvénient des représentations sur forme de planisphère est le temps de calcul.
Pour chaque pixel d'une image il faut tout d'abord calculer la valeur des angles θ_{pi} et ϕ_{pi} en fonction des conditions de prise de vue puis de calculer la projection du pixel dans l'image panoramique. Toutes ces transformations sont à base de fonctions trigonométriques assez lourdes en temps de calcul. Comme nous l'avons vu précédemment, lorsque le centre de projection est confondu avec le centre des rotations, la transformation qui relie deux images est une homographie.

2.5 Multi résolution

2.5.1 Avant propos

L'aspect multi-résolution est un point que nous n'avons pas encore abordé. Cependant il peut être considéré comme un paramètre intrinsèque de la construction d'un panorama comme de la visualisation. Nous avons vu dans le chapitre précédent que sous certaines conditions, nous pouvons déterminer l'homographie entre deux plans correspondant à deux prises de vue. Cette homographie peut être calculée à partir des paramètres intrinsèques et extrinsèques des deux prises de vue. Parmi les paramètres intrinsèques, la distance focale, exprimée en pixel, correspond justement à la résolution. Nous pouvons donc calculer l'homographie entre deux images dont la longueur focale, c'est à dire la résolution, n'est pas la même. Nous traitons donc implicitement l'aspect multi-résolution. Ce même constat s'applique lors de la visualisation. La limite étant, cette fois ci, la résolution avec laquelle est construit le panorama. Si la distance focale de l'image à construire est plus faible que celle du panorama, l'image pourra être construite sans perte d'information. En revanche, si la distance focale de l'image à construire est plus grande que celle du panorama, l'image ne pourra être construite que par interpolation.

2.5.2 Problématique

Nous avons présenté plusieurs modes de représentation possibles d'une mosaïque d'images.
Toutes ces représentations ont leurs avantages et leurs inconvénients. Cependant, elles sont toutes basées sur le principe d'une focale fixe. C'est à dire que l'ensemble de la scène est stockée avec la même résolution. Le logiciel QuickTime VR ainsi que d'autres logiciels du commerce permettent de créer un panorama avec plusieurs résolutions. Cependant, la résolution la plus haute est utilisée pour créer plusieurs panoramas de résolution plus basse.
L'intérêt est d'optimiser les calculs lors des visualisations à basse résolution et d'optimiser les transferts sur le réseau.

Figure 2.12: Plusieurs résolutions d'un même panorama

Or, nous pouvons imaginer que sur l'ensemble de la scène toutes les zones ne présentent pas le même intérêt.

2.6 Conclusion

Dans ce chapitre, nous avons présenté plusieurs méthodes permettant de représenter un panorama. Toutes ces approches ont leurs avantages et leurs inconvénients et finalement seront choisies en fonction du besoin. Pour une visualisation de l'ensemble de la scène en une seule vue, nous choisirons une projection cylindrique. Habituellement, c'est la projection plate carré cylindrique équidistante qui est utilisée.
Dans ce chapitre, nous avons fait l'hypothèse que les paramètres de la projection sont parfaitement connus. De même, nous avons considéré que les conditions de luminosité ou que le gain de la caméra n'ont pas varié au cours de la prise de vue des différentes images.
Nous nous sommes placés dans un cas idéal auquel il est possible de se ramener. Il suffit que la caméra soit suffisamment bien instrumentée (précision et temps de réponse) pour obtenir les paramètres précis de la prise de vue. De même, le gain de la caméra peut être fixé. Nous allons à présent aborder le cas un peu moins idéal et présenter différentes méthodes permettant d'obtenir un panorama robuste.

Chapitre 3

Création d'un panorama robuste en temps réel

3.1 Objectif

Nous avons présenté dans le chapitre précédent les principes mathématiques de base permettant de calculer et de représenter une mosaïque d'images. Nous avons alors fait l'hypothèse que la caméra correspond exactement au modèle sténopé, que les paramètres intrinsèques et extrinsèques de la caméra sont parfaitement connus et qu'il n'y a pas de changement de luminosité de la scène au cours de l'acquisition. Dans la pratique, nous allons être confrontés à des situations où nous ne sommes pas dans le cas « idéal » . Nous allons donc avoir à résoudre un certain nombre de difficultés. Bien qu'il existe plusieurs solutions à ces différents problèmes, elles sont bien souvent incompatibles avec le temps réel. La définition du temps réel reste une notion subjective. Notre définition du temps réel est le délai qui sépare deux acquisitions. Cela va donc dépendre des applications visées. Dans la mesure où la fréquence d'acquisition classique des caméras est de 25 ou 30 images par seconde, le temps de calcul ne devrait pas excéder respectivement 40ms ou 30ms.
Cependant, dans certaines applications, une fréquence de 5 images par seconde suffit, ce qui porte ce délai à 200ms. Si le but final de la création du panorama se limite à la visualisation *a posteriori* de la scène, de nombreux logiciels sont disponibles.
Comme nous l'avons indiqué en introduction, certain d'entre eux sont même gratuits. Le logiciel AutoStitch™ que l'on peut se procurer gratuitement sur Internet, permet un recalage précis des images et donne un rendu excellent. Cependant, pour créer un panorama à partir de 72 images couleur de taille 640x480, le temps de calcul est d'environ 65s soit près d'une seconde par image. Pour une application de visualisation, ce délai est très honorable.
Nous avons globalement, 3 défis à relever : corriger le défaut de positionnement de la caméra, compenser les changements de luminosité et supprimer ce que les auteurs appellent les « fantômes ». Pour chacune de ces opérations, nous allons présenter un état de l'art des solutions existantes et dans la mesure du possible.

3.2 Interpolation

Avant toute chose, nous devons résoudre une première difficulté liée à la discrétisation des images numériques. La définition classique d'une image numérique donnée par la géométrie discrète est la suivante. Une image numérique est constituée d'un ensemble de cellules dénombrables appelées pixels dans le cas des images en 2 dimensions. Ces cellules forment une partition de la portion de plan que représente l'image. Ce pavage de l'image est appelé espace discret ou grille discrète. Le centre de gravité de chaque cellule est appelé point discret.

Or, il est peu probable qu'un pixel x_c de l'image courante C se projette, après transformation, exactement sur la grille de l'image panoramique. Il est donc nécessaire de diffuser la valeur du pixel vers les plus proches voisin sur la grille discrète. Classiquement ce problème est résolu par une interpolation bilinéaire en « backward mapping ».

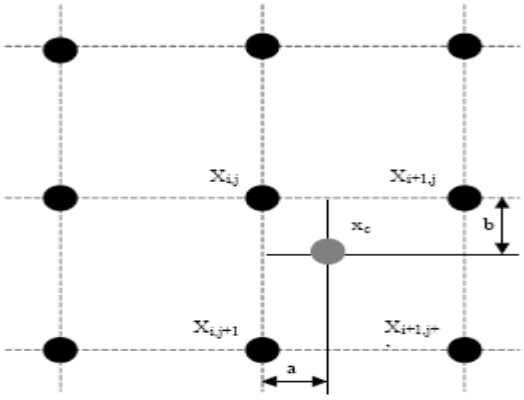


Figure 3.1:principe de l'interpolation linéaire

La valeur du pixel x_c est une somme pondéré des quatre pixels les plus proches :

$$xc = (1-b)\cdot(1-a)\cdot Xi,j + a\cdot(1-b)\cdot Xi,+1j + b\cdot(1-a)\cdot Xi,j+1 + a\cdot b\cdot Xi+1,j+1$$

Figure 3.2:Projection d'une image sans interpolation

Figure 3.3:Projection de la même image avec interpolation bilinéaire

La première image correspond à une projection sans interpolation. Nous pouvons remarquer la pixélisation qui apparaît sur l'ensemble de l'image mais dont l'effet est plus sensible au niveau des gradients forts. Par contre, sur la deuxième image réalisée avec une interpolation bilinéaire, les gradients forts sont légèrement lissés.

Il existe d'autre type d'interpolation comme l'interpolation bicubique ou l'interpolation au plus proche voisin. L'intérêt de l'interpolation bilinéaire est qu'elle est simple à calculer et donne de bons résultats dans l'ensemble, même si elle laisse apparaître des discontinuités au niveau des dérivées de l'image obtenue.

3.3 Défaut d'alignement

Avant de réaliser notre premier panorama, nous avons encore à vérifier le domaine de validité du modèle mathématique que nous utilisons. Notre but ici n'est pas de remettre en cause le modèle sténopé. Le domaine de validité de ce modèle a été abondamment commenté dans la littérature, notamment dans l'ouvrage de Faugeras [FAU93] et dans celui de Hartley [HAR04]. Cependant, nous faisons souvent l'hypothèse que l'axe de projection coupe le plan focal au centre de l'image. Dans [SZE94] Szeliski montre que si le point d'intersection est peu éloigné du centre de l'image l'impact de ce décalage n'a pas d'effet significatif. D'autre part, pour justifier l'utilisation des matrices d'homographie 2D entre deux images, nous avons fait également l'hypothèse que le centre de projection est confondu avec le centre des rotations. C'est à dire que le centre de projection ne se déplace pas avec le mouvement de la caméra.

3.3.1 Etude théorique

Dans le cas idéal, le centre de projection Oc de la caméra est confondu avec le centre O des rotations, comme indiqué dans la figure ci-dessous. Pour illustrer notre propos nous étudions simplement le décalage suivant l'axe z et nous considérons que l'angle $\phi = 0$. Le plan image Π est à une distance f correspondant à la distance focale.

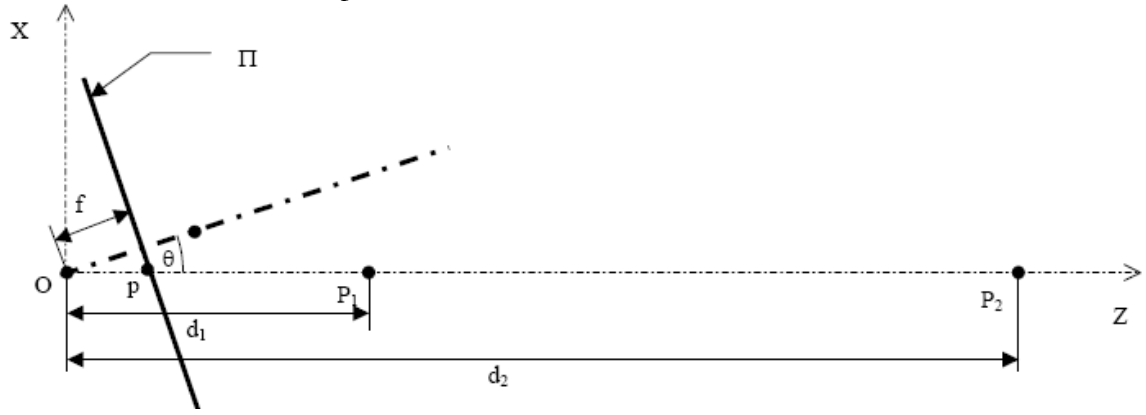


Figure 3.4:Centre de projection de la caméra confondu avec le centre des rotations

Si deux points P_1, P_2 ainsi que le centre O sont alignés, quelle que soit la rotation θ (à l'exception des cas dégénérés $\theta = \pi/2$ et $\theta = -\pi/2$) les deux points se projettent en un point p unique sur le plan image.
Maintenant, si le centre de projection Oc est à une distance d du centre O des rotations, pour $\theta = 0$, les points P_1 et P_2 se projettent toujours en un point p du plan image. Par contre, pour une rotation $\theta \neq 0$, les points P_1 et P_2 se projettent respectivement en deux points p_1 et p_2 sur le plan image. L'angle θ_1 est l'angle que fait le rayon $[OcP_1]$ avec l'axe optique et l'angle θ_2 est l'angle que fait le rayon $[OcP_2]$ avec l'axe optique.

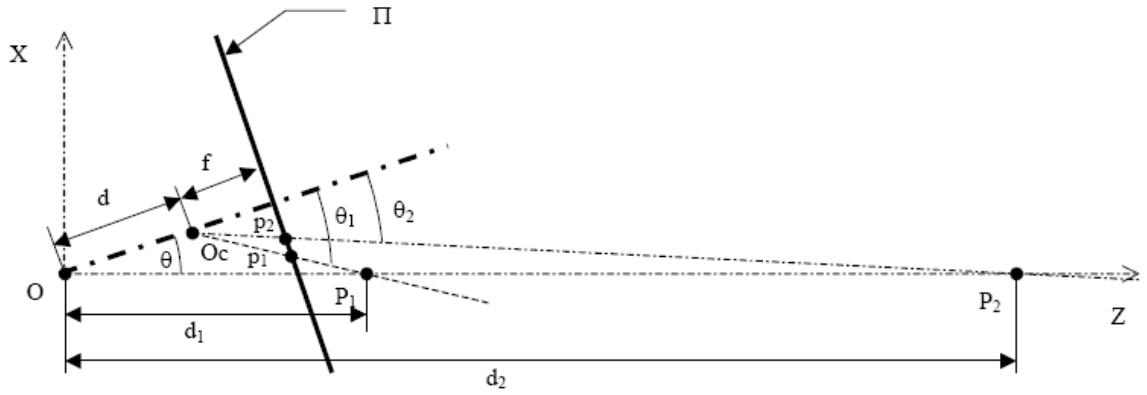


Figure 3.5:Décalage positif du centre optique de la caméra par rapport au centre des rotations

Dans le schéma ci-dessus, nous avons arbitrairement placé le centre Oc entre le centre O et le plan focale. Le même schéma s'applique si Oc est placé en arrière du centre O.

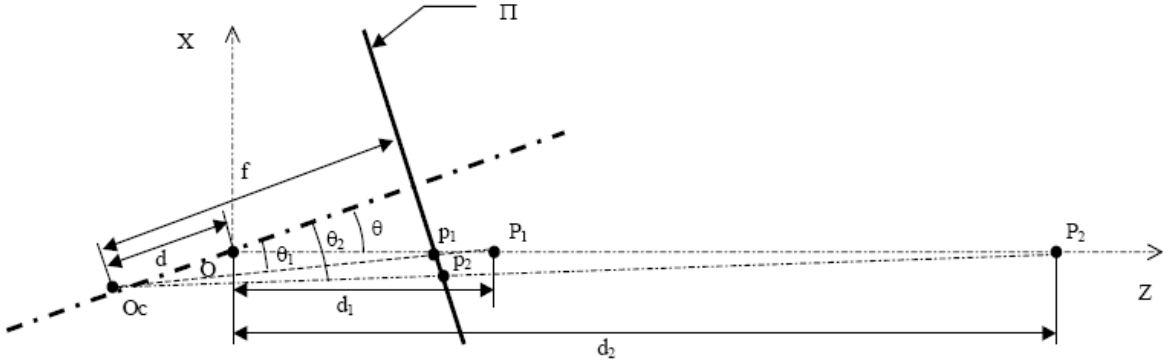


Figure 3.6:Décalage négatif du centre optique de la caméra par rapport au centre des rotations

La différence est que si le centre de projection est entre le centre des rotations et le plan image, un point proche de la camera aura tendance à être décalé vers l'extérieur de l'image.
A l'inverse, si le centre de projection est placé derrière le centre des rotations, un point proche de la camera aura tendance à être décalé vers le centre de l'image. Dans les deux cas, le décalage observé est un décalage relatif par rapport au cas où O et Oc sont confondus.
A partir des équations présentées au chapitre précédent, il est possible de déterminer l'angle θ_1 d'un point p_1 de coordonnée (u_1,v_1) dans le plan image. Pour rappel :

$$\tan\theta_1 = \frac{u_1}{f}$$

Nous cherchons à présent à déterminer la distance d correspondant à la distance entre le centre O des rotations et le centre Oc de projection. En fonction des données du problème nous pouvons reconstituer le schéma simplifié suivant :

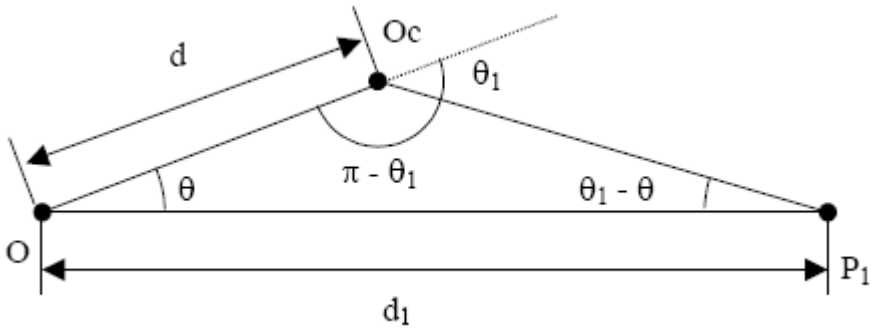


Figure 3.7:Schéma simplifié du décalage positif entre le centre optique et le centre des rotations

De ce schéma, nous pouvons écrire :

$$d.\sin\theta_1 = d_1.\sin(\theta_1 - \theta)$$

D'où :

$$d = d_1.\frac{\sin(\theta_1 - \theta)}{\sin\theta_1} \qquad (3.1)$$

3.3.2 Conclusion

En conclusion, pour réaliser un panorama robuste, nous devons prendre un certain nombre de précautions. Il n'y a pas de solution simple pour corriger le défaut d'alignement. Si le centre de projection se déplace le long de l'axe optique en fonction du facteur de zoom, le mouvement de rotation de la caméra entraîne un décalage des pixels vers la droite ou vers la gauche en fonction de l'angle et de la profondeur (dans le repère du monde) du point visé.
Pour Peer [PEE02] ce défaut est une qualité. Il utilise justement cette excentricité pour calculer la carte de profondeur d'un panorama. Pour réaliser un panorama robuste, nous devons nous mettre dans des conditions qui minimisent la portée de cette excentricité. Dans la mesure du possible, afin d'éviter les distorsions dues au décalage entre le centre de projection et le centre des rotations, nous utiliserons une distance focale de 8mm. Un autre intérêt est que pour cette distance focale, les distorsions radiales sont également négligeables. Pour les distances focales inférieures, si les obstacles sont à une distance supérieure à 7m (pour la focale la plus petite) et que la résolution de l'image est de 640 x 480 pixels, les distorsions dues au décalage ne sont pas significatives. Par contre, nous devrons tenir compte des distorsions radiales. Pour des distances focales supérieures à 8 mm, la distance minimale devient rapidement importante. Elle est de 55m pour une distance focale de 40mm. C'est une contrainte importante lorsque l'on souhaite réaliser un panorama dans un espace relativement réduit. Par contre cela ne constitue plus un problème lorsque pour nous car on cherche à réaliser un panorama d'un paysage.

3.4 Défaut de positionnement

Un autre défaut que nous pouvons qualifier également de défaut intrinsèque est le défaut de positionnement. En effet, si la commande de la caméra n'est pas suffisamment précise, le

plaquage des images sur le panorama ne sera pas correct. En prenant la distance focale la plus faible (i.e. la moins sensible au défaut de positionnement), une erreur de 0,5° sur la position de l'angle panoramique engendre un décalage de près de 7 pixels sur l'axe horizontal de l'image. Pour réaliser un panorama robuste, nous avons donc à résoudre un problème de recalage d'image avec comme première conséquence, la nécessité de garantir un certain recouvrement entre les images et donc d'augmenter le nombre de prises de vue. Le recalage d'image est un problème complexe qui motive énormément la communauté scientifique. Nous l'évoquons simplement ici et nous lui consacrons le chapitre suivant.

3.5 Alignement photométrique

3.5.1 Problématique

Le contrôle de la luminosité n'est pas un problème aussi simple qu'il y paraît. Si la luminosité est relativement constante dans toutes les directions alors il suffit de fixer manuellement le gain de la caméra pendant toute la durée de la prise de vue. Par contre, il ne faut pas que cette luminosité évolue pendant cette prise de vue (lors du passage d'un nuage par exemple). Si il y a de trop gros écarts de luminosité dans la scène alors des zones risquent de se trouver sous-exposées. Elles apparaîtront donc très sombres dans la mosaïque d'image. A l'inverse, d'autres zones beaucoup plus lumineuses risquent de se trouver sur-exposées. La solution est donc de laisser la caméra gérer automatiquement le gain en fonction de la luminosité de la portion de scène visée. Cette solution offre bien sûr l'avantage que chaque image est acquise en optimisant la dynamique du capteur. Le problème est qu'une même portion de la scène prise avec deux gains différents n'a plus la même valeur de luminosité. L'exemple suivant est un cas d'école. La caméra est placée à l'intérieur d'une pièce. Cette pièce est éclairée par une lumière artificielle et par une lumière naturelle à travers une fenêtre. La luminosité à l'intérieur est faible par rapport à celle de l'extérieur. Si le gain de la caméra est fixé manuellement pour obtenir une luminosité correcte à l'intérieur de la pièce, la fenêtre apparaîtra sur-exposée et il ne sera pas possible de visualiser l'extérieur. Dans le cas contraire, si le gain est piloté automatiquement par la caméra, alors l'extérieur devient visible, mais le phénomène de couture apparaît.

Figure 3.8:Gain automatique, l'extérieur est visible mais présence de couture

3.5.2 Etat de l'art

Pour résoudre cet alignement photométrique et éviter ce problème de couture, plusieurs solutions sont proposées dans la littérature. Nous pouvons les classer en 2 catégories en fonction de la portée de la correction dans le panorama. Une première classe d'algorithme proposée dans

la littérature s'attache à appliquer une correction du gain sur l'ensemble du panorama. Dans ce cas, deux solutions se détachent. La première consiste à normaliser le gain des différentes images constituant le panorama. Huang et al [HUA98] proposent une normalisation du gain de la camera à partir de la moyenne et de l'écart type des pixels présents dans la zone de recouvrement. La correction du gain est appliquée à l'ensemble de l'image. Leur approche a finalement pour effet de fixer le gain de toute la construction panoramique à la valeur d'une image de référence. On se retrouve donc dans le cas du gain fixe à la différence qu'il est calculé automatiquement et non fixé manuellement par l'utilisateur. Dans [CAN03], Candocia utilise une technique similaire mais appliquée localement de façon à améliorer la mise en correspondance des images.

Figure 3.9:Normalisation du gain à partir de la valeur moyenne et de l'écart type

Dans cette première approche, il y a une perte d'information lié à la contraction de la dynamique des pixels. Pour éviter ces désagréments, plusieurs auteurs dont Mann [MAN95], proposent d'étendre la dynamique des pixels. Les composantes RGB des pixels ne sont plus codées sur 8 bits mais sur 16 ou 32 bits. Cette solution offre l'avantage de tirer profit de la dynamique totale du capteur. Les zones sous exposées et sur exposées sont acquises en optimisant la définition. Le problème se pose lors de la restitution du panorama. En fonction de l'utilisation demandée, une contraction ou un glissement de la dynamique du panorama doit être opéré.
La deuxième catégorie d'algorithme a une portée plus locale et s'attache à améliorer la transition entre deux images. Une première solution consiste à réaliser une simple moyenne entre les pixels des deux images. Si l'écart de gain entre les deux images est faible, cette solution donne des résultats acceptables mais dès que l'écart est important, la présence de couture réapparaît.

Figure 3.10:Correction du gain par calcul de la moyenne entre deux pixels commun

Uyttendaele and al [UYT02] proposent une solution pour éliminer ce qu'ils appellent les artefacts de luminosité. Ils découpent chaque image en blocs de 32x32 et calculent pour chaque bloc une fonction de transfert permettant de lisser la luminosité.
Dans [LEV04], les auteurs proposent une solution localisée uniquement sur la jonction entre deux images. Ils lissent la couture de façon à limiter le gradient entre les deux images. Le reste de l'image comme la zone de recouvrement, ne sont pas modifié. Utilisée seule, cette solution peut s'avérer très utile lorsque la zone de recouvrement entre les images est très faible. Une autre solution proposée par [GRA09, LEM07] consiste à utiliser un algorithme de type Graph-Cut. Le principe de cet algorithme est de trouver la jointure qui donne la meilleure délimitation dans la zone de recouvrement.

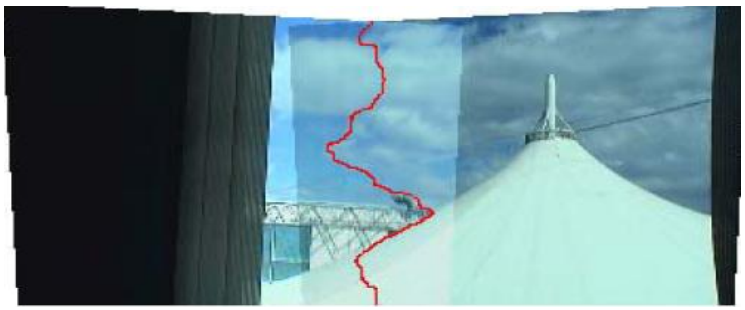


Figure 3.11:Exemple de délimitation avec l'algorithme « Graph Cut »

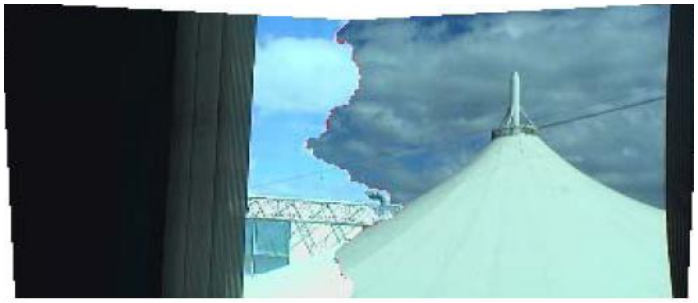
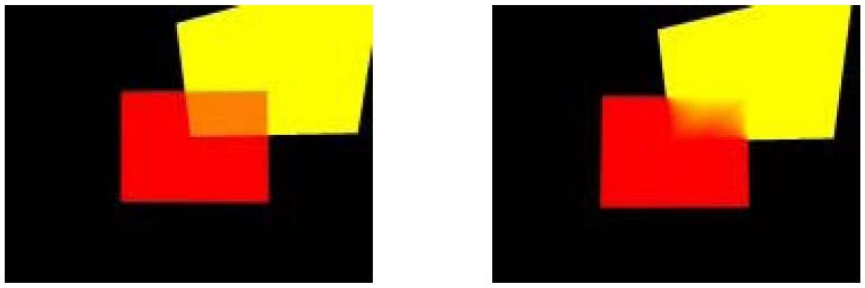


Figure 3.12:Résultat du plaquage des deux images avec l'algorithme « Graph Cut »

Comme pour la méthode de diffusion par la moyenne, cette solution donne d'assez bons résultats lorsque la différence de gain est faible. Comme nous pouvons le voir sur l'image ci-dessus, le résultat est visuellement moins bon avec un écart de gain important.

3.5.3 Solution proposée

Par contre ces solutions ne s'exécutent pas en temps réel. La méthode temps réel que nous proposons consiste à lisser par un dégradé les zones des images qui se recoupent. Traditionnellement la zone de recouvrement entre deux images est calculée en effectuant la moyenne des pixels issus des deux images. Dans l'exemple ci-dessous, nous avons plaqué deux images l'une rouge et l'autre jaune sur une face de cube avec une zone de recouvrement. La méthode simple basée sur la moyenne fait apparaître cette zone de recouvrement en orange avec des coupures franches.



(a) (b)

Figure 3.13*:Exemple de rendu du calcul de la zone de recouvrement entre deux images*

(a) : calcul de la valeur moyenne dans la zone de recouvrement
(b) : pondération proportionnelle à la distance du point au contour

Nous avons introduit un facteur de pondération proportionnel à la distance du pixel par rapport à un point du contour le plus proche. Concrètement, nous commençons par déterminer la zone de recouvrement entre les deux images. Cette zone est délimitée par un contour. Pour chaque point de ce contour nous recherchons l'image à laquelle il appartient.
En toute objectivité, il appartient aux deux, mais nous considérons qu'il appartient à l'image I si les 8 pixels qui l'entourent appartiennent aussi à l'image I. Puis, pour chaque pixel p de la zone de recouvrement, nous calculons la distance minimale d^1 qui sépare ce point d'un point du contour appartenant à l'image 1 et la distance d^2 le séparant d'un point du contour de l'image 2. La valeur d'un pixel p_k dans la zone de recouvrement est donnée par :

$$p_k = p_k^1 \frac{d_k^2}{d_k^1 + d_k^2} + p_k^2 \frac{d_k^1}{d_k^1 + d_k^2} \qquad (3.2)$$

Le résultat final est le suivant :

Figure 3.14:Correction du gain par diffusion

La correction que nous apportons n'est pas parfaite, mais elle permet de lisser le changement de luminosité et ne laisse pas apparaître de discontinuité.

3.6 Suppression des « fantômes »

3.6.1 Problématique

Le dernier problème que l'on rencontre lors de la construction d'une mosaïque d'images est que pendant le processus d'acquisition, les objets contenus dans la scène ne sont pas forcément immobiles. Entre deux prises de vue, il arrive qu'un objet (ou plusieurs objets), situé(s) dans la zone de recouvrement de deux images, soi(en)t en mouvement. Lors de la projection des images dans le panorama, les objets en mouvement ne pourront pas être alignés correctement. Si nous appliquons les différents algorithmes d'alignement photométrique que nous avons présentés précédemment, les objets en mouvement vont apparaître flous ou plus ou moins fondus dans la scène, d'où l'apparition de « fantômes ».

Figure 3.15:Exemple de séquence d'images avec un personnage en mouvement dans la scène

Figure 3.16:Projection des images dans le même plan en utilisant la méthode de la moyenne.

Dans l'exemple ci-dessous, les trois images ont été acquises à deux secondes d'intervalle. Le décalage entre deux images successives est d'environ 2° selon l'angle de panorama. Dans cette scène un personnage est en mouvement. Après projection des trois images dans le plan de l'image t+1, où une simple moyenne des pixels est calculée dans la zone de recouvrement, les composantes statiques de la scène sont correctement alignées et apparaissent nettes. Par contre, les pixels correspondant au personnage en mouvement sont moyennés avec des pixels du fond de la scène. Le personnage semble donc se « dématérialiser ».

3.6.2 Cas général

Les solutions proposées dans la littérature afin de supprimer ces « fantômes » dépendent du nombre d'images mises à contribution pour reconstruire une portion de la scène. A partir de

3 images, la plupart des auteurs dont [IRA96] proposent l'utilisation d'un filtre médian. C'est une méthode simple à mettre en oeuvre et qui donne des résultats relativement satisfaisants, comme nous pouvons le voir dans l'image suivante.

Figure 3.17:projection des images dans le même plan en utilisant un filtre médian

Pour calculer cette image, nous avons repris les images de l'exemple précédent. Comme nous pouvons le constater, le personnage n'apparaît plus. Seules les composantes statiques sont conservées.

Dans [SHU00], les auteurs proposent une méthode basée sur le calcul du flot optique. Dans [UYT02], les auteurs identifient et isolent les régions en mouvement dans chacune des images. Chacune de ces régions correspond à une partie plus ou moins importante de l'objet en mouvement. A partir d'un graphe de mise en correspondance de ces régions, ils recherchent la région qui contribue le plus à la représentation de l'objet. Seule cette région est maintenue dans l'image correspondante et les autres sont supprimées.

Figure 3.18:projection des images dans le même plan en utilisant la méthode décrite dans [UYT02]

Lorsque le panorama est réalisé à partir d'un flux vidéo, nous pouvons disposer d'un nombre beaucoup plus important d'images pour restituer une portion de la scène. Nous pouvons alors, après projection de chaque image dans un plan de référence, utiliser des méthodes de modélisation des composantes statiques du fond. Ces méthodes sont traditionnellement utilisées pour segmenter les objets en mouvement dans le cas des caméras fixes. Dans le cas d'une caméra PTZ en rotation, nous pouvons considérer qu'après projection de chaque image dans un plan de référence l'image correspondant à ce plan est une image issue d'une caméra fixe. De même,

chaque prise de vue peut être réalisée à partir d'une collection d'image pendant laquelle la caméra est fixe. Nous allons donc nous intéresser aux différents algorithmes permettant cette modélisation.

3.6.3 Modélisation des composantes statiques du fond

Plusieurs stratégies différentes sont proposées dans la littérature pour créer et mettre à jour un modèle de fond. La première approche consiste à réaliser une moyenne glissante à chaque nouvelle acquisition d'image. Cette solution simpliste peut donner de bons résultats lorsqu'il n'y a pas de changement de luminosité et que la scène ne subit pas de modification. Suivant le temps d'intégration de l'image de référence et la vitesse de déplacement des objets, cette technique permet d'extraire la forme complète contrairement à l'approche précédente où seuls les pixels ayant changés de luminosité entre deux images sont détectés. Il n'en reste pas moins que cette approche présente, comme la précédente, l'inconvénient d'avoir à déterminer, souvent empiriquement, le seuil de la différence à fixer pour étiqueter les pixels en mouvement. Ce seuil, qui est généralement le même pour chacun des pixels de l'image, est sensible aux contrastes de l'image. Il s'agit alors de trouver un compromis entre les zones ensoleillées et les zones de l'image à l'ombre, à moins d'analyser localement le contraste des différentes zones de l'image de façon à adapter le seuil.

Cette solution est bien trop restrictive pour s'appliquer à l'extérieur. Bertolino et al. [BER01] ont développé une méthode basée sur le calcul d'une moyenne glissante en considérant deux phases : initialisation et mise à jour. Dans la phase d'initialisation, l'image de référence est construite à partir du calcul d'une moyenne glissante où chaque pixel est pondéré par un terme α_p sur n images successives d'une séquence.

$$Iref(p,t+1)=\alpha p \cdot I(p,t)+(1-\alpha p)\cdot Iref(p,t)$$

La valeur du terme α fixe le temps de réponse du filtre. Pour ne prendre en compte dans l'initialisation de cette image de référence que les pixels fixes, ils créent une carte de stabilité en calculant la différence de trois images successives. Cette carte leur permet donc de distinguer les pixels fixes des pixels mobiles. Pour chaque pixel p, si p appartient au fond,
alors $\alpha_p \in]0,1]$ sinon $\alpha_p = 0$. Une fois l'image de référence construite, ils passent dans une deuxième phase où l'image de référence est mise à jour en utilisant le même principe de la moyenne glissante pondérée par α_p sans utiliser de carte de stabilité. Les changements brutaux de la luminosité globale de l'image sont détectés en calculant simplement la différence cumulée de chaque pixel entre deux images successives divisée par le nombre de pixels dans l'image. Si cette valeur atteint un certain seuil, cette valeur est ajoutée à tous les pixels de l'image de référence.
Dans un article très complet sur la surveillance en temps réel des personnes, Haritaoglu et al [HAR00] proposent également une solution permettant de modéliser le comportement de chaque pixel de l'image dans une fenêtre glissante. Après l'application d'un filtre médian sur l'image courante, ils déterminent pour chaque pixel, la valeur minimum d'illumination, la valeur maximum ainsi que le gradient maximum entre deux images successives. Ces trois informations pour chaque pixel leur permettent de déterminer les seuils pour l'étiquetage des pixels de l'image suivante. L'inconvénient majeur de cette approche, comme pour l'algorithme de Perner, est qu'il est nécessaire de garder en mémoire un buffer de plusieurs images.
Afin de palier ce défaut, plusieurs auteurs [STA99,PAV01], modélisent la variation de chaque pixel de l'image au cours du temps par plusieurs distributions gaussiennes représentées par une moyenne et un écart type. Cette méthode est communément appelée « mélange de gaussienne ».

Le nombre de distributions utilisées pour la modélisation, dépend de la complexité des mouvements du fond. Si le fond reste fixe, deux distributions suffisent.
Si le fond varie, ce qui est notamment le cas dans les scènes en extérieur avec du vent dans les arbres par exemple, il peut être nécessaire d'utiliser plusieurs distributions. Dans [DAR05], Dar-Shyang propose une solution basée sur l'algorithme de Stauffer et al. [STA99] permettant d'améliorer la stabilité et la rapidité de la convergence. Enfin, dans [KIM04] les auteurs proposent une méthode utilisant ce qu'ils appellent des « codebook ». Le principe consiste à isoler la variation de chaque pixel dans une ou plusieurs zones colorimétrique en mémorisant un certain nombre de caractéristiques comme les dates de début et de fin d'apparition du pixel dans la zone, le nombre d'occurrence, etc. Les auteurs proposent des stratégies pour fusionner ou supprimer des blocs. Cette méthode semble donner de bons résultats. Cependant, le temps d'intégration pour stabiliser le modèle est relativement long.

Figure 3.19:Extrait du panorama réalisé à partir d'une séquence vidéo en utilisant les mélanges de gaussienne.

3.7 Conclusion

Notre première préoccupation était de déterminer l'influence de l'écart entre la caméra et les modèles mathématiques que nous utilisons. Au regard des résultats, nous pouvons considérer que les modèles utilisés sont valables dans des conditions normales d'utilisation.
Nous avons également présenté les problèmes liés à l'alignement photométrique et à la suppression des « fantômes ». Plusieurs solutions existent mais elles sont utilisées essentiellement lorsque la finalité de la construction du panorama est la visualisation de la scène. Lorsque la caméra est utilisée pour de la détection en temps réel ou pour piloter un robot, ces traitements ne sont pas appliqués. Le défaut de positionnement que nous allons aborder maintenant est plus significatif. C'est la raison pour laquelle nous lui consacrons le chapitre suivant.

Chapitre 4

Recalage d'images

4.1 Problématique

Au cours du chapitre précédent nous avons abordé un certain nombre de problèmes et apporté des solutions permettant d'obtenir un panorama robuste aux conditions de prise de vue et aux objets en mouvement dans la scène. Ces différents points sont essentiellement gênants pour la visualisation *a posteriori* de la scène. Dans les applications de détection ou de pilotage d'un robot, ces défauts ne sont pas traités de la même façon dans la mesure où ce qui importe ce n'est pas l'ensemble de la scène mais principalement ce qui est vu à l'instant présent par la caméra. Pour la visualisation d'un panorama, nous cherchons à supprimer les objets en mouvement alors que pour la détection nous cherchons justement à les détecter.
Cependant, dans les deux cas, il y a un défaut qu'il est absolument impératif de corriger. C'est le défaut de positionnement. Pour que la visualisation de la scène soit parfaite, il est nécessaire que les images acquises soient correctement plaquées sur le panorama. De même pour le suivi automatique des objets en mouvement, il est indispensable de connaître les paramètres de prise de vue de façon à extraire de l'image les bonnes informations de pilotage. La solution la plus simple est d'instrumenter correctement la caméra. Si ce n'est pas le cas, nous devons donc extraire les paramètres de prises de vue à partir de l'image elle-même et des images acquises précédemment. Nous avons donc un problème de recalage d'images à résoudre.

Le recalage d'images est un problème particulièrement intéressant. Cependant, au vu des problèmes rencontrés, notamment en ce qui concerne le temps de calcul, nous avons dû y consacrer une partie de notre travail. Bien que de nombreuses solutions aient été proposées pour la construction de panoramas, la réalisation de mosaïques de haute qualité en temps réel reste une tâche très difficile. Le recalage d'images n'est pas un problème spécifique au mosaïquage.
Dans certaines applications, le recalage d'images est l'objectif final, alors que dans d'autres applications, c'est un lien exigé pour accomplir des tâches d'un niveau plus élevé. Le but du recalage d'images est d'aligner géométriquement deux images ou plus de sorte que des pixels respectifs ou leurs dérivés (bords, coin, etc..), représentant la même structure fondamentale, puissent être mis en correspondance. L'objectif de la plupart des algorithmes décrits dans la littérature est de mettre en correspondance les images selon leurs propriétés radiométriques ou géométriques en utilisant une fonction spécifique pour évaluer la qualité de la mise en correspondance. Quelles que soient les méthodes utilisées et comme précisé dans [BRO92], le recalage entre deux images I et J peut se formaliser de la façon suivante :

$$\hat{T} = \arg\min_{T \in E} C(I, J \circ T) \qquad (4.1)$$

où T est la transformation qui appartient à un espace de recherche E de transformations, $J \circ T$ est l'application de la transformation T sur l'image J et C est une mesure de similaritéque l'on va chercher à minimiser.

Pour rappel (cf. Chapitre 2), dans le cas de notre étude, T est la transformation homographique H qui relie deux images. Dans la suite de cette étude, nous continuerons à utiliser la lettre T pour exprimer la transformation de façon à généraliser notre propos, tout en gardant à l'esprit que la transformation que l'on recherche est une homographie 2D qui s'exprime avec 8 coefficients et qui est définie à un facteur d'échelle près :

$$X' \sim \mathrm{H}.x = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \bullet \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

où x est un point de l'image I1 défini par ses coordonnées homogènes (u, v, 1) dans l'image et x' sa projection homographique dans l'image I0 et défini par ses coordonnées homogènes (u', v', 1). Le signe ~ indique la relation d'équivalence, à un facteur d'échelle près, entre le point x' et la transformation du point x.
Comme nous l'avons vu dans le chapitre 2, les paramètres de cette transformation peuvent être calculés directement à partir des informations des prises de vue. Cependant, nous pouvons également la calculer à partir des coordonnées des points dans les deux images. En effet, les coordonnées (u', v') du point x' s'expriment en fonction des coordonnées (u, v, 1) de la façon suivante :

$$u' = \frac{m_0.u + m_1.v + m_2}{m_6.u + m_7.v + 1}, \qquad v' = \frac{m_3.u + m_4.v + m_5}{m_6.u + m_7.v + 1},$$

Soit sous forme matricielle :

$$\begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -u.u' & -v.u' \\ 0 & 0 & 0 & u & v & 1 & -u.v' & -v.v' \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \end{bmatrix} = \begin{bmatrix} u' \\ v' \end{bmatrix} \qquad (4.2)$$

L'appariement de n points entre deux images permet de disposer de 2n équations. Ce système peut alors se résoudre par la méthode des moindres carrés par exemple.

Comme nous l'avons évoqué, le recalage d'images n'est pas un problème spécifique à la construction d'un panorama. Nous n'allons pas présenter un état de l'art exhaustif des méthodes de recalage et nous limitons notre étude aux méthodes qui peuvent être appliquées aux caméras PTZ.

4.2 Etat de l'art du recalage appliqué aux caméras PTZ

Il y a plusieurs façons de classer les différentes approches. Elles peuvent tout d'abord être classées en fonction du type de transformation que nous allons faire subir à l'image ou de la complexité du modèle utilisé. Une autre méthode de classification consiste à considérer d'une

part les approches locales qui visent à déterminer la transformation mathématique entre deux images seulement et les approches globales qui prennent en compte la totalité du panorama.
Enfin, nous pouvons également les classer en fonction de l'utilisation d'une partie ou de la totalité des pixels des deux images.
La première classification que nous avons évoquée concerne le type de transformation que nous devons faire subir aux images. Dans ce cas, les transformations sont généralement classées en deux catégories : les transformations rigides et les transformations non rigides ou élastiques. Nous aurons une transformation rigide lorsqu'il n'y a pas de déformation de l'image et où nous n'utilisons que des rotations, des translations et des facteurs d'échelles. A l'inverse, s'il est nécessaire de faire subir une transformation inhomogène à l'image avec éventuellement changement de la topologie, nous aurons une transformation non rigide. Les transformations homographiques sont un cas particulier. Comme elles déforment l'image, nous avons une tendance naturelle à les classer dans les transformations non rigides.
Cependant nous les classerons dans la catégorie des transformations rigides puisqu'en coordonnées homogènes la transformation est linéaire et bijective. Bhat et al [BHA00] utilisent un modèle basé uniquement sur un mouvement de translation de la caméra PTZ. Ce modèle simple est utilisé par les auteurs pour segmenter les objets en mouvement entre deux prises de vue. Ils considèrent donc que le mouvement de la caméra peut être approximé par une translation. Toutefois, cette hypothèse n'est vérifiée que pour les petits angles d'inclinaison. L'estimation de la translation est obtenue par la détection et le suivi de points d'intérêts, nous y reviendrons par la suite. Des modèles de transformations plus complexes sont généralement proposés, telles que les transformations rigides ou affines [SZE97, BRO03], ou les transformations projectives [BEV05, BEV06]. Toutefois, la plupart des caméras s'écartent du modèle sténopé, généralement en raison des distorsions radiales que l'on observe pour les distances focales faibles. Dans leur approche, Sinha et al. [SIN04] proposent une solution pour compenser ces distorsions. Leur modèle prend également en compte le rapport entre la hauteur et la largeur des pixels. En effet, comme nous l'avons évoqué dans le paragraphe de la définition de l'unité pixélique , les pixels ne sont généralement pas carrés mais rectangulaires. Cependant, cette information est un paramètre intrinsèque à la caméra qui n'évolue pas avec le temps. Il peut donc être déterminé à partir d'un calibrage de la caméra.

Le deuxième mode de classification consiste à séparer les approches locales des approches globales. Les approches locales, qui sont les plus courantes, visent à déterminer les paramètres du modèle utilisé pour chaque couple d'images successives. Ces approches sont généralement efficaces et rapides, mais les petites erreurs d'alignement successives ont tendances à s'accumuler avec le temps. Ces erreurs sont d'autant plus visibles lorsque la vidéo renvoie à une zone du panorama déjà capturée (problème connu sous le nom de "looping path"). Les approches globales [SZE97, BRO03] formulent le problème du recalage d'images de façon à résoudre l'ensemble des paramètres et contraintes du système ; dans le cas de mosaïque d'images, une de ces contraintes est que les extrémités d'un panorama se rejoignent. Ces types d'approches d'optimisation exacte sont la plupart du temps incompatibles avec le temps réel.
De façon plus générale, les techniques de recalage sont classées en fonction de la contribution des pixels de l'image. Traditionnellement, deux types d'approches sont considérées : denses (dites aussi iconique) et éparses (géométriques). En simplifiant, nous pouvons dire que les méthodes denses mettent à contribution l'ensemble des pixels présents dans la zone de recouvrement des deux images alors que les méthodes éparses n'utilisent que certains pixels aux caractéristiques particulières. Les méthodes denses [IRA96, SZE97, SIN04] cherchent à estimer itérativement les

paramètres de la caméra par la minimisation d'une fonction de coût basée la plupart du temps sur la différence d'intensité entre les zones de recouvrement. Nous verrons par la suite qu'il existe plusieurs méthodes de calcul pour estimer cette différence. La plus commune est la somme des différences au carré (Sun Square Difference SSD). Szeliski et Shum [SZE97] proposent une mise à jour itérative des paramètres de la matrice d'homographie basée sur l'utilisation de la mesure SSD. Les méthodes denses sont réputées plus précises mais ne sont pas robustes lorsqu'il y a beaucoup d'objets en mouvements dans les images. Elles sont donc particulièrement efficaces dans le cas de scènes statiques. Les méthodes éparses [BAR03, BEV05, BEV06, BRO03] utilisent la mise en correspondance de points d'intérêts, de lignes ou d'autres primitives géométriques pour déterminer les paramètres de la caméra. Plusieurs solutions proposées sont basées sur le détecteur de coins de Harris [HAR88] ou sur les SIFT décrit par Lowe [LOW04]. De façon plus marginale, d'autres solutions sont basées sur les appariements de régions [COH89, LEE93]. Plus classiquement, Bevilacqua et. al [BEV05] proposent de mettre en correspondance des points caractéristiques de l'image. Ils utilisent un modèle de projection homographique et proposent une solution pour la fermeture du panorama. Ils améliorent leur approche dans [BEV06] en prenant en compte les changements d'illuminations. Brown et Lowe [BRO03] proposent de mettre en correspondance des points d'intérêts à partir de l'algorithme SIFT qui sera décrit par Lowe dans [LOW04]. Ils utilisent une transformation affine en justifiant que le descripteur SIFT est invariant aux changements affines, ainsi qu'un algorithme de type RANSAC. Cet algorithme probabiliste permet de minimiser la complexité de la mise en correspondance et permet de supprimer les correspondances aberrantes. Enfin, ils utilisent un ajustement décrit dans [TRI00] permettant un recalage global d'un ensemble d'images. Leur approche permet la génération du panorama particulièrement robuste. Cependant leur algorithme nécessite 83 secondes pour recaler 8 images avec un PC équipé d'un processeur 2GHZ.

Nous avons essentiellement axé notre travail de recherche sur l'optimisation du temps de calcul. Dans un premier temps, nous avons étudié deux méthodes décrites dans la littérature.
Une méthode dense proposée par Szeliski [SZE94] et une méthode éparse, à partir des SIFT, décrit par Lowe [LOW04]. Ces deux méthodes se sont révélées particulièrement efficaces mais nécessitent des temps d'exécution beaucoup trop longs. Dans le cadre des applications
visées, le recalage n'est qu'une étape intermédiaire, mais nécessaire, avant de réaliser des opérations de plus haut niveau. Nous avons donc étudié plusieurs approches différentes, denses ou éparses, que nous avons adaptées à notre cas particulier de façon à optimiser le temps de calcul.

4.3 Limitation de l'espace de recherche

Pour aborder le problème du recalage, nous nous plaçons dans le cas d'une projection centrale pour laquelle le centre de projection est confondu avec le centre des rotations.

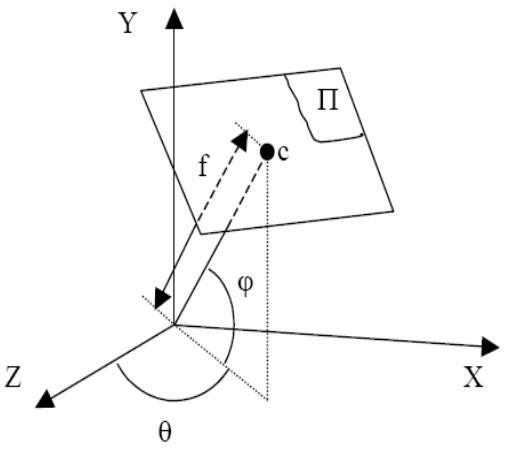


Figure 4.1:Schéma de la projection centrale

Nous avons montré que dans ce cas, la transformation mathématique qui relie deux images est une homographie (i.e. Chapitre 2). Pour rappel, soit θ l'angle de panorama, ϕ l'angle de tangage et f la distance focale. c, représente le centre optique de l'image[6] et Π le plan image. Nous avons vu qu'une transformation homographique entre deux images s'exprime à partir d'une matrice H à 8 paramètres. Dans le cas d'une projection centrale on montre que cette matrice peut être calculée à partir des paramètres intrinsèque et extrinsèque du modèle sténopé [FAU93]. En utilisant un modèle simplifié et sans tenir compte des distorsions géométriques, chromatiques ou autre, la projection s'exprime de la façon suivante :

$$H = K_I . R_{\varphi I} . R_{\theta I} . R_{\theta J}^{-1} . R_{\varphi J}^{-1} . K_J^{-1} \qquad (4.3)$$

où K est la matrice simplifiée des paramètres intrinsèques du modèle, R_θ et R_ϕ les matrices de rotations suivant les angles de panorama et de tangage. La matrice K s'exprime de la façon suivante :

$$K = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.4)$$

où fu et fv correspondent à la distance focale exprimée en unité pixélique suivant l'axe u et l'axe v, et (u0, v0) correspond au centre de projection de l'image.

Généralement, le modèle utilisé est encore plus simplifié. Dans ce cas, nous considérons que le repère $(c, \vec{u}, \vec{v})$ est orthonormé, c'est à dire que fu = fv = f. S'il est vrai que les axes u et v sont orthogonaux, les pixels ne sont pas forcément carrés mais plutôt rectangulaires. La distance focale exprimée en unité pixélique n'est donc pas forcément identique suivant les axes.
Cependant, Szeliski a montré dans [SZE94] qu'un faible décalage du centre optique de l'image n'a que peu d'influence sur le recalage. Comme données variables au cours du temps, il reste à estimer les paramètres de prise de vue des deux images, c'est-à-dire la distance focale et les deux rotations.

Nous pouvons éventuellement poser des limites en fonction de connaissances *a priori*. Par exemple, en fonction de la vitesse d'acquisition, nous pouvons déterminer le déplacement

[6] Pour rappel, nous appelons centre optique de l'image, le point d'intersection entre l'axe optique et le plan image

maximum de la caméra. De même, si l'information de position de la caméra est disponible, il est possible que nous connaissions l'imprécision maximale de ces données. Il est utile de pouvoir clairement poser les limites de l'espace de recherche de façon à éviter la divergence des méthodes de recalage. Surtout en ce qui concerne les méthodes itératives.

Maintenant que nous avons clairement posé les limites de notre espace de recherche, nous allons définir des métriques pour mesurer la qualité de la mise en correspondance.

4.4 Algorithmes de recalage

Pour la présentation des méthodes de recalage, nous avons repris la classification traditionnelle qui consiste à définir deux catégories : dense et éparse. Nous abordons en premier le cas des méthodes denses. La présentation des méthodes éparses suit une progression un peu différente. Ces méthodes sont, pour la plupart, une succession d'étapes intermédiaires (recherche des points d'intérêt, appariement, optimisation …), pour lesquelles il peut y avoir différentes solutions.

4.4.1 Méthodes denses

Le principe général des méthodes denses consiste à explorer l'espace des transformations possibles de façon à minimiser la mesure de (dis)similarité. Nous avons alors à faire à un problème d'optimisation globale qui se décompose en deux grandes approches : déterministe et aléatoire. Dans la première approche, les algorithmes utilisent toujours le même cheminement en fonction des conditions initiales. Parmi ces méthodes nous pouvons citer : la descente de gradient, la minimisation de Gauss-Newton ou encore la minimisation de Levenberg-Maquardt. Si nous faisons abstraction de la phase d'initialisation, la méthode du simplexe, que nous allons voir en détail (cf annexe), se classe également dans cette catégorie. Dans le cas des méthodes dites aléatoire, le cheminement est dicté, en partie, par des lois stochastiques. A partir des mêmes conditions initiales, le cheminement et surtout le résultat final n'est pas forcément le même. Par contre ces méthodes sont généralement réputées pour être moins sensibles aux minima locaux. Nous allons tout d'abord présenter la méthode déterministe de Szeliski [SZE94]. Cette méthode consiste à affiner itérativement les paramètres de la matrice d'homographie. Dans la classe des méthodes déterministes, nous verrons également la méthode du simplexe (cf annexe). Nous définissons la méthode du simplexe comme étant déterministe parce que pour une initialisation donnée du simplexe, qui peut être aléatoire, l'évolution du simplexe est déterministe. Nous verrons la méthode du recuit simulé ainsi qu'une adaptation des algorithmes génétiques (cf annexe).

4.4.1.1 Méthode de Szeliski

Une méthode de recalage classique utilisée pour la construction d'une mosaïque d'images est proposée pour la première fois par Szeliski dans [SZE94]. Cette méthode permet de calculer la matrice d'homographie H d'une image I1 dans I0 par itérations successives. L'algorithme fait intervenir une matrice de correction D permettant la mise à jour de la matrice H. Le calcul des paramètres de la matrice de correction s'effectue en recherchant la minimisation d'une fonction de coût. Szeliski propose d'utiliser la mesure SSD.

L'algorithme est finalement assez simple. La matrice H de départ est initialisée soit avec des paramètres approchés de l'homographie par une autre méthode ou par des informations issues de la caméra soit avec la matrice identité. L'image du gradient, la matrice jacobienne et la matrice de l'erreur d'intensité sont ensuite calculées à partir de l'image I1 transformée par la matrice H de départ. A partir de ces données, la matrice de correction D est évaluée en utilisant la méthode des moindres carré. Cette matrice est ensuite appliquée à la matrice H.
Par itérations successives, on affine les valeurs de la matrice H. L'algorithme s'arrête sur un critère d'arrêt qui peut être un nombre maximum d'itérations atteint, un calcul sur la somme ou la somme au carré des valeurs de la matrice d'erreur d'intensité ou encore sur la somme ou la somme au carré des paramètres de la matrice D. Une présentation plus complète de cet algorithme est donnée en annexe.
La méthode de Szeliski donne de très bons résultats lorsque l'écart de position entre les images est très faible. Cependant, les résultats sont nettement moins bons lorsque les écarts sont importants. De plus, c'est une méthode particulièrement lente.

4.4.2 Méthodes éparses

Le principe des méthodes éparses consiste généralement à mettre en correspondance des points particuliers ou des structures géométriques de façon à disposer de suffisamment d'équations pour résoudre le système linéaire et déterminer la transformation mathématique entre les deux images.

Les méthodes éparses que nous allons présenter sont donc basées sur ce principe et reprennent, pour la plupart, le schéma suivant :

- Détection des points d'intérêts dans les deux images. Cette première étape consiste à sélectionner des points suffisamment caractéristiques pour être susceptibles d'être appariés sans ambiguïté. En général, les points à fort gradient sont privilégies. Dans notre étude nous nous sommes limités au cas des points d'intérêts. C'est la solution la plus courante, mais certains auteurs utilisent d'autres types de primitives géométriques (ligne, cercle).
- Mise en correspondance des points. Cette étape consiste à déterminer des couples de points entre les deux images et dont les caractéristiques sont proches.
- Première estimation des paramètres de la transformation et suppression de mauvais appariement (outliers). Cette étape est relativement classique lorsque l'on résout un système de plusieurs équations par la méthode des moindres carrés. Elle consiste à calculer une première solution avec l'ensemble des équations à disposition puis à appliquer la transformation sur les données. Les équations dont les résultats s'écartent trop de la mesure sont alors rejetées.
- Estimation de la solution finale après suppression des « outliers ».

Une variante de cette approche consiste non pas à détecter les points d'intérêts dans la deuxième image, mais de suivre ceux de la première image au cours du temps. Cette approche est essentiellement utilisée dans les séquences vidéo pour lesquelles le déplacement de la caméra entre deux images successive est relativement faible. Nous allons voir l'ensemble de ces différentes étapes de façon plus approfondie.

4.4.2.1 Détection des points d'intérêts

La première étape consiste donc à sélectionner des points caractéristiques dans les deux images. La détection de points d'intérêts n'est pas un problème récent et plusieurs solutions existent déjà. Nous nous sommes limités à étudier uniquement le détecteur de Harris. Dans [SCH00], le lecteur trouvera la comparaison de plusieurs détecteurs réalisée par Schimd et Mohr. Cependant, l'algorithme le plus cité dans la littérature est celui de Harris [HAR88] qui découle des travaux de Moravec [MOR77]. Ce détecteur s'attache à localiser des points où les variations différentielles sont importantes dans plus d'une direction de façon à ne pas inclure les contours.

Ce détecteur, bien qu'étant le plus utilisé, n'est pas toujours facile à mettre en oeuvre, puisque cinq paramètres sont utilisés (cf annexe) : la taille du filtre gaussien, le filtre dérivatif, le paramètre k, le voisinage des maxima locaux et le seuil final. Une adaptation de ce détecteur est proposée par Zheng [ZHE99].
A l'issue de cette première étape, nous disposons de deux ensembles de points.

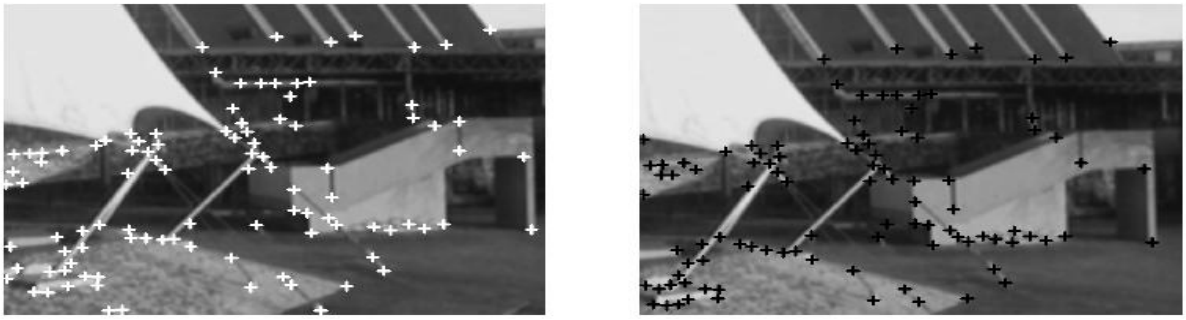


Figure 4.2*:Détection des points d'intérêts dans deux images avec le détecteur de Harris*

4.4.2.2 Mise en correspondance

L'étape suivante est la mise en correspondance. Cette mise en correspondance consiste à déterminer les couples de points dont les caractéristiques sont communes. Les caractéristiques de chaque point peuvent être calculées en tenant compte de la couleur ou de la texture au voisinage des points. Certains auteurs utilisent également le résultat du détecteur de Harris. Les résultats de la comparaison entre tous les points sont donc placés dans un tableau de (n x m) éléments, où n est le nombre de points retenus dans l'image I et m le nombre de points retenus dans l'image J. Ce tableau est parcouru de façon à trouver la valeur maximum. Lorsque la valeur maximum est déterminée, les deux points sont appariés et la ligne et la colonne correspondant aux deux points sont effacées. Ce processus se poursuit jusqu'à ce que le min(n, m) points soit atteint où si le maximum trouvé n'atteint pas un seuil fixé à l'avance. En supposant que m est du même ordre de grandeur que n, cet algorithme de recherche est d'une complexité en $O(n^3)$ en l'état et pourrait aisément voir sa complexité ramenée à $O(n^2\log(n))$ si un tri préalable des n^2 valeurs était effectué. Ce problème de couplage bipartie pondéré pourrait être résolu plus efficacement en utilisant les tas de Fibonacci. Le nombre « n » de points à traiter étant relativement faible (inférieur à 100), ces optimisations algorithmiques auraient présenté un intérêt essentiellement théorique.
Une autre solution assez classique consiste à rechercher pour tous les points de la première image le point le plus semblable dans la deuxième image puis d'intervertir le processus. Les appariements retenus sont les paires de points qui se sont choisi mutuellement.

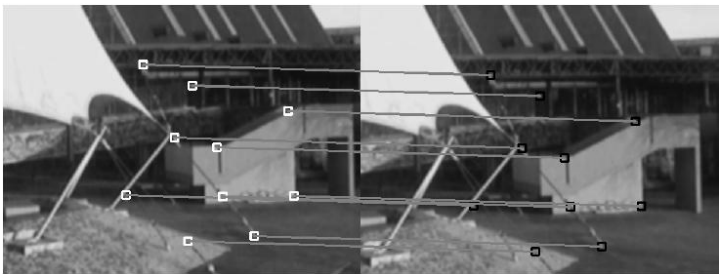


Figure 4.3*:Mise en correspondance des points d'intérêts dans deux images*

Le fait d'utiliser la liste des candidats possibles que nous avons évoquée dans le paragraphe précédent permet de limiter le risque de faux appariement. De plus, comme tous les candidats possibles d'un même point sont tous dans un même voisinage cela évite de calculer des solutions complètements aberrantes. Le risque évidemment est que des mauvais appariements peuvent conduire à des solutions acceptables sans pour autant être optimales.
La mise en correspondance nécessite de disposer d'un descripteur local robuste aux transformations que l'on peut faire subir à l'image. Parmi les descripteurs locaux, les plus utilisés sont :

• Les invariants différentiels dont le *jet local* qui caractérise la surface locale en niveau de gris par un développement de Taylor [KOE87]. Dans [SCH97] Schmid utilise ce descripteur calculé jusqu'à l'ordre 3 pour l'indexation d'image. Les invariants différentiels ont été étendus aux images couleur par Gouet dans [GOU00]

• Le détecteur SIFT (*Scale Invariant Feature Transform*) basé sur le calcul un histogramme des orientations. Ce détecteur proposé par Lowe [LOW04] est considéré aujourd'hui comme l'un des plus performants. L'intérêt de ce détecteur est qu'il est invariant aux transformations affines (changement d'échelle, rotation et translation) et qu'il permet de calculer un descripteur robuste. Ceci le rend particulièrement efficace dans le cas de recalage rigide.

• Les invariants fréquentiels comme la transformée de Fourier-Mellin, la transformée de Gabor ou la transformée en ondelettes.

Avec la suppression des « outliers » que nous avons évoquée plus haut, ces trois étapes constituent l'ossature des méthodes de recalage éparse que nous avons étudiées. Afin de déterminer la qualité de ces méthodes, nous avons implémenté la solution en utilisant les détecteurs SIFT décrit par Lowe [LOW04]. L'algorithme SIFT est très efficace cependant, le temps de calcul ainsi que l'espace mémoire nécessaire à son exécution sont assez important.

4.4.2.3 RANSAC

Une alternative à l'approche qui consiste à constituer des paires de points à partir de la similarité de leurs caractéristiques locales est de définir aléatoirement un sous ensemble de paires de points et de tester plusieurs solutions pour ne garder que la meilleure. Cette approche s'inspire de l'algorithme RANSAC (*RANdom SAMple Consensu*) exposé par Martin A. Fischler et Robert C. Bolles dans [FIS81]. A l'origine cet algorithme a été développé pour résoudre des problèmes liés à la cartographie.
Dans notre cas, la solution que nous recherchons correspond à l'homographie entre les deux images qui s'exprime à partir de 8 paramètres indépendants. Nous avons donc besoins de 4

paires de points pour résoudre le système linéaire. Le principe de l'algorithme est alors le suivant. A partir de deux ensembles de points, nous définissons aléatoirement un sous ensemble de 4 paires de points. Ces 4 paires de points nous permettent de calculer une matrice de transformation qui est appliquée à l'image J. Si cette solution minimise la fonction de coût, elle est préservée, sinon elle est rejetée. Dans les deux cas, une nouvelle solution est calculée. Lorsqu'un certain nombre d'itérations est atteint, la meilleure solution intermédiaire
H', est utilisée pour déterminer précisément l'ensemble des paires de points. C'est-à-dire les points pour lesquels la distance entre $p_{i,I}$ et $H'.p_{j,J}$ est inférieure à un seuil.

$$\left\| p_{i,I} - H^{'}.p_{i,j} \right\| < \varepsilon \qquad (4.5)$$

Cet ensemble de paire de points plus complet est utilisé pour calculer la matrice H définitive.
Cependant, dans [LOW04] l'auteur indique que l'algorithme RANSAC ne donne pas de bons résultats lorsque la proportion de faux appariements est supérieure à 50%.
La méthode RANSAC donne de meilleurs résultats que la méthode « local jet » mais elle est un peu moins rapide. La difficulté de cette approche est qu'il faut réaliser suffisamment de tirage aléatoire de sous ensemble pour augmenter la probabilité de réaliser un bon appariement.

4.4.3 Conclusion

Les méthodes denses donnent de bons résultats, mais le principal inconvénient de ces méthodes est que si elles ne sont pas correctement initialisées, elles risquent de ne pas tendre vers la solution optimale. Même si les méthodes stochastiques permettent de contourner cette difficulté, le nombre d'itérations n'est pas constant ce qui rend aléatoire le temps de calcul. Parmi les méthodes denses, la méthode du simplexe et à la fois la plus rapide et celle qui donne globalement les meilleurs résultats.
Les méthodes éparses sont dans l'ensemble beaucoup plus rapides que les méthodes denses mais elles donnent des résultats légèrement moins bons. Dans [BAR03-2], les auteurs proposent une solution de recalage où ils utilisent une première méthode éparse pour s'approcher de la solution puis une méthode dense pour affiner le résultat.

Chapitre 5

Travail effectue

5.1 Présentation

Il existe plusieurs types de reconstruction panoramique à partir d'images. Les mosaïques les plus simples sont créées à partir d'un ensemble d'images dont les déplacements mutuels sont des translations « image-plan ». C'est approximativement le cas avec les images satellites.
D'autres mosaïques sont créées en tournant la caméra autour d'un centre optique à l'aide d'un dispositif spécial et créant une image panoramique qui représente la projection de la scène sur un cylindre ou une sphère.

Ici, Le champ expérimental concerne la recomposition d'images panoramiques pour des stations de ski. Les algorithmes de recomposition connus de la littérature fonctionnent très bien dans les conditions météorologiques idéales. Cependant, les outils basés sur ces algorithmes posent de réels problèmes pour une recomposition sans défauts dans ce contexte d'utilisation. En effet, une mauvaise météo est source de défauts de recomposition .On peut aussi évoquer le défaut connu sous le nom *ghost* (fantôme) causé par effets de mouvements des skieurs.
La résolution de ce problème peut être vue à travers une méthode qui sera ambitieuse et qui donne résultats acceptables à certain degré.
L'application mise en œuvre est basée sur les travaux de David G.lowe pour la détection des points d'intérêts.
Nous proposons une solution de recalage où on utilise une première méthode éparse pour s'approcher de la solution puis une méthode dense pour affiner le résultat.
Dans un premier temps, l'utilisation d'un détecteur de points d'intérêts permet d'extraire d'une image les coordonnées de points caractéristiques Le détecteur présenté ici est une version améliorée du détecteur de coins de Harris, le détecteur de Harris-Laplace, qui est robuste aux rotations, aux variations de luminosité et aux changements d'échelle.
Dans un deuxième temps, les points ainsi localisés sont décrits à l'aide de descripteurs de vecteurs de caractéristiques qui permettent ensuite d'appareiller les points d'intérêts de plusieurs prises de vue distinctes d'une même scène. Pour rendre cela possible, il est nécessaire que les descripteurs vérifient un certain nombre d'hypothèses alors on a choisit SIFT (The Scale Invariant Feature Transform) comme le descripteur de caractéristiques le plus connu grâce de sa stabilité et sa précision hautement performant aux autres descripteurs. Après la mise en correspondance (*Matching*) des vecteurs de caractéristique. Nous obtenons la matrice de transformation en utilisant RANSAC "RANdom SAmple Consensus". Il s'agit d'une méthode itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées. Depuis la matrice de transformation est connu, les deux images peuvent être simplement alignées ensemble.

5.2 Diagramme global

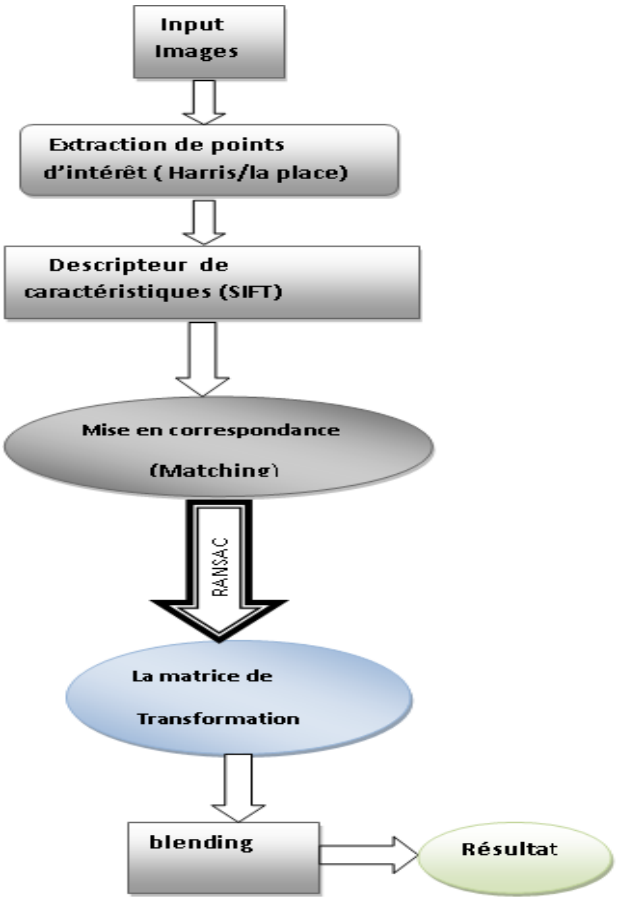


Figure 5.1: Diagramme global

5.2.1 Extraction de points d'intérêts.

Un des problèmes fondamentaux en vision par ordinateur et en traitement d'images concerne la détection de primitives (*features* en anglais). C'est une étape préalable à de nombreuses applications. Le principe est d'extraire d'une image des régions d'intérêt qui ont une certaine singularité et sont particulièrement porteuses d'information. Les primitives que l'on cherche à extraire peuvent être :

– des points d'intérêt (aussi appelés *points anguleux ou points saillants*) comme des coins ou des jonctions en T
– des contours (bordures des objets par exemple)
– des régions de l'image
– etc.

Ces primitives correspondent bien sûr à des objets présents dans la scène qui a été photographiée ou filmée (le mot *objet* est à prendre au sens large : ce peut être des personnes ou encore des lettres manuscrites dans le cas de la reconnaissance de caractères).

La difficulté du problème consiste à mettre au point une méthode qui soit robuste : en appliquant le même algorithme de détection de primitives à une autre image où figure le même objet, on souhaite maximiser la probabilité que les primitives détectées correspondent aux mêmes points de l'objet, et que les points détectés dans la première image le soient aussi dans la deuxième. On parle aussi souvent de la *répétabilité* du détecteur. Divers facteurs peuvent en effet modifier l'apparence de l'objet et donc perturber les résultats du détecteur de primitives :

– le bruit de l'image
– les conditions d'illumination
– la position de la caméra ou de l'appareil photo par rapport à l'objet

5.2.1.1 Le détecteur de Harris-Laplace

Pour comprendre plus précisément le principe de fonctionnement d'un détecteur de points d'intérêt et les difficultés mises en jeu, nous allons rapidement décrire la méthode du détecteur de Harris. Avec Stephens, Harris a proposé en 1988 un algorithme de détection qui est certainement aujourd'hui le plus connu et qui est toujours très utilisé car il donne de très bons résultats et est assez simple à mettre en œuvre.
Les points d'intérêt recherchés sont des points au voisinage desquels l'image varie significativement dans plusieurs directions. Ce peut être des coins, des jonctions en T, des jonctions en Y, etc. Comme souvent pour les algorithmes de détections de primitive, il faut éviter de retenir des points situés sur un contour d'un objet pour deux principales raisons :

1. Lorsque l'objet est observé d'un point de vue différent, le bord de l'objet sur l'image ne correspond plus à la même zone physique dans la scène.
2. Un point situé sur un contour n'est généralement pas stable car les points du même bord situés dans son voisinage ont souvent une apparence similaire.

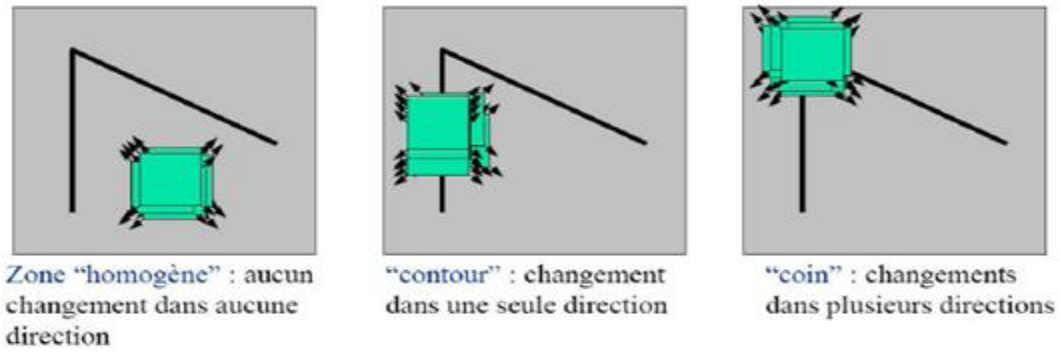


Figure 5.2:*Détecter les coins sans retenir les contours*

Le principe du détecteur de Harris est plus facilement accessible si l'on rappelle le fonctionnement du détecteur de Moravec : considérons une petite fenêtre de N ×N pixels que l'on centre sur un point (u, v) de l'image. Lorsque l'on décale légèrement cette fenêtre dans différentes directions et que l'on calcule la moyenne des différences d'intensité entre les pixels correspondants, plusieurs cas sont à envisager (voir la Figure 5.2) :
– Si l'intensité est globalement uniforme autour du point (u, v), les écarts d'intensités calculés seront tous assez faibles.

– Si la fenêtre est située au niveau d'un contour qui passe par le point (u, v), un décalage le long de ce bord engendrera de faibles écarts d'intensité, tandis qu'un décalage de la fenêtre perpendiculairement à la direction de ce bord provoquera des différences plus importantes.
– Si le point (u, v) est un coin, tout décalage de la fenêtre engendrera des écarts d'intensité importants.
Etant donné un déplacement (Δx, Δy), la fonction d'auto-corrélation au point *(u,v)* est définie comme suit :

$$c_{(u,v)}(\Delta x, \Delta y) = \sum_{(x_i, y_i) \in W} \left| I(x_i + \Delta x, y_i + \Delta y) - I(x_i, y_i) \right|^2$$

où :

– I(x, y) désigne l'intensité du pixel (x, y)
– W est la fenêtre centrée au point (u, v)

Ainsi, la méthode décrite ci-dessus revient formellement à retenir les maxima locaux dans l'image de cette quantité :

$$D(u,v) = \min_{(\Delta x, \Delta y) \in E} c_{(\Delta x, \Delta y)}(u,v)$$

Où : E = {(1, 0), (1, 1), (0, 1), (−1, 1)} désigne les différents déplacements considérés.
Comme l'ensemble E des décalages utilisés est discret, la réponse est anisotropique. En utilisant une approximation du premier ordre, on obtient :

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + (I_x(x_i, y_i) I_y(x_i, y_i)). \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Où $I_x = \frac{\partial I}{\partial x}$ désigne la dérivée partielle en x de la fonction intensité.
Ainsi :

$$c_{(u,v)}(\Delta x, \Delta y) \approx \sum_{(x_i, y_i) \in W} \left[(I_x(x_i, y_i) I_y(x_i, y_i)). \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right]^2$$

$$\approx (\Delta x \Delta y) \underbrace{\begin{pmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{pmatrix}}_{= M : \ matrice \ \ d'autocorrélation} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Une réponse trop bruitée Utiliser une fenêtre rectangulaire et binaire tend à renvoyer des résultats assez bruités. Harris et Stephens proposent plutôt de pondérer les valeurs dans la somme avec une fonction gaussienne 2D g_σ,d'écart-type σ, centrée au point (u, v). Ainsi, on réécrit la matrice d'auto-corrélation de la sorte :

$$M = \begin{pmatrix} \sum g_\sigma(x_i, y_i)(I_x(x_i, y_i))^2 & \sum g_\sigma(x_i, y_i) I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum g_\sigma(x_i, y_i) I_x(x_i, y_i) I_y(x_i, y_i) & \sum g_\sigma(x_i, y_i)(I_y(x_i, y_i))^2 \end{pmatrix}$$

Où pour chaque somme, (xi, yi) parcourt toute l'image (en pratique, on continue à ne prendre en compte qu'une région limitée autour du point (u, v), où le poids de la fonction gaussienne n'est pas négligeable).
Ceci peut être réécrit plus simplement grâce à l'opérateur de convolution :

$$M = g_\sigma \begin{pmatrix} I_x^2 & I_x.I_y \\ I_x.I_y & I_y^2 \end{pmatrix}$$

L'algorithme est trop sensible aux points situés sur des contours, dans la méthode originale de Moravec, seul le minimum de $c_{(\Delta x,\Delta y)}(u,v)$ pour les 4 déplacements considérés est pris en compte. On comprend facilement que l'algorithme retient souvent des points situés sur des contours alors que nous avons expliqué que l'on cherche à les éviter.
Harris et Stephens utilisent une méthode plus subtile grâce à la matrice d'auto-corrélation (aussi appelée matrice de moment du second ordre) qui représente les variations locales de l'image au point considéré. Comme nous avons vu que $c_{(u,v)}(\Delta x,\Delta y)$ est grand lorsque l'intensité varie fortement dans la direction perpendiculaire au déplacement $(\Delta x,\Delta y)$, on déduit que les points d'intérêt sont les points (u,v) pour lesquels la matrice d'auto-corrélation a deux valeurs propres grandes. Cela correspond aux points pour lesquels il existe localement une base de vecteurs propres décrivant des variations locales importantes de l'image.
Harris et Stephens ont proposé de calculer la fonction d'intérêt suivante :

$$\Theta(u,v) = \det(M) - \alpha trace(M)$$

Le premier terme correspond au produit des valeurs propres, alors que le second pénalise les points de contour qui n'ont qu'une seule forte valeur propre (typiquement, on choisit $\alpha = 0{,}06$).
Les points d'intérêt retenus sont ainsi ceux qui correspondent aux maxima locaux de cette fonction $\Theta(\cdot,\cdot)$ et qui sont au delà d'un certain seuil.

Mise en œuvre du détecteur de Harris

1. Construction des images correspondant aux dérivées premières de l'image initiale. Pour ce faire, on peut convoluer l'image par les filtres de dérivation basiques, tels que [-1 0 1] mais on peut aussi utiliser une convolution avec la dérivée première d'une gaussienne.
2. Evaluation de la matrice d'auto corrélation en tout point de l'image par calcul de moyennes pondérées par une gaussienne des valeurs issues des images calculées en 1.
3. Calcul de la fonction d'intérêt de Harris en tout point de l'image.
4. Recherche des maxima locaux supérieurs à un seuil fixé.

5.2.2 Le descripteur de point de caractéristiques (SIFT)

Scale-invariant feature transform (ou SIFT) est un algorithme pour détecter caractéristiques locales dans une image. SIFT caractéristiques sont très adaptés pour le problème de couture d'image, car ils sont invariants à l'échelle, l'orientation et distorsion affine.
Pour calculer les caractéristiques de chaque SIFT images d'entrée, on a directement utilisé une fonction MATLAB SIFT (ImageName) de détecteur de Lowe's SIFT Keypoint. Demo logiciel est fourni à [LOWE]

Au moment de calculer le descripteur, le point d'intérêt s'est déjà vu attribué une position, une orientation et même une échelle (nous verrons plus loin comment l'échelle peut être déterminée. Nous préférons dans un premier temps expliquer le principe d'un descripteur). Reste donc à définir le descripteur de son voisinage, qui doit être aussi invariant que possible aux variations restantes, telles que les changements d'illuminations ou de point de vue.

Dans un premier temps, en utilisant l'image $E(\cdot, \cdot, \sigma)$ de l'espace d'échelle correspondant à l´echelle σ déterminée préalablement (voir plus loin), on calcule la norme m et l'orientation θ des gradients d'intensité en chacun des pixels (x, y) du voisinage de taille 16 × 16 autour du point d'intérêt (u, v) :

$$m(x, y) = \sqrt{(E(x+1, y) - E(x-1, y))^2 + (E(x, y+1) - E(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \frac{E(x+1, y) - E(x-1, y)}{(E(x, y+1) - E(x, y-1)}$$

Ces calculs sont symbolisés par les petites flèches dans la partie gauche de la figure (Fig5.3). Afin d'assurer l'invariance à la rotation, les orientations des gradients sont ajustées pour prendre en compte l'orientation du point d'intérêt, attribuée précédemment.
Le descripteur est symbolisé dans la partie droite de la figure(Fig5.3). Il correspond à un ensemble de plusieurs histogrammes des orientations de gradient, chacun représentant une zone de 4 × 4 pixels. Comme le montre la figure, chaque histogramme est composé des huit classes.
En fait, afin de ne pas trop pénaliser un petit décalage dans la localisation du point d'intérêt et pour éviter les effets de bords, la contribution de chaque gradient—initialement égale à sa norme—est pondérée selon trois critères :
– chaque gradient calculé peut avoir un impact dans quatre histogrammes différents, car un gradient est pris en compte pour un histogramme si la distance entre le point où il est calculé et le centre de la zone de l'histogramme est inférieur à la « largeur » de quatre pixels. Sa contribution est pondérée en prenant en compte cette distance.
– De même, chaque gradient est pris en compte dans deux classes contiguës de chaque histogramme, correspondant aux deux orientations (parmi les huit) qui sont les plus proches de l'orientation du gradient étudié. Sa participation à chacune des deux classes est une nouvelle fois pondérée pour tenir compte de l'écart entre l'orientation du centre de la classe et la véritable orientation du gradient.
– La contribution des gradients dans les histogrammes est enfin pondérée selon une fonction gaussienne symbolisée, afin d'attribuer une importance moindre aux pixels situés plus loin du point d'intérêt détecté.
Le vecteur du descripteur est alors formé en concaténant les valeurs des entrées de chaque histogramme, correspondant aux longueurs des flèches sur la figure. Celle-ci ne montre qu'un tableau de 2 × 2 histogrammes calculés à partir d'un voisinage de 8 × 8 pixels alors que la méthode imaginée par Lowe propose en fait de construire 4 × 4 histogrammes en considérant un voisinage de 16 × 16 pixels. Comme chaque histogramme est constitué de huit classes, le vecteur caractérisant un point d'intérêt est donc composé de 4 × 4 × 8 = 128 valeurs.

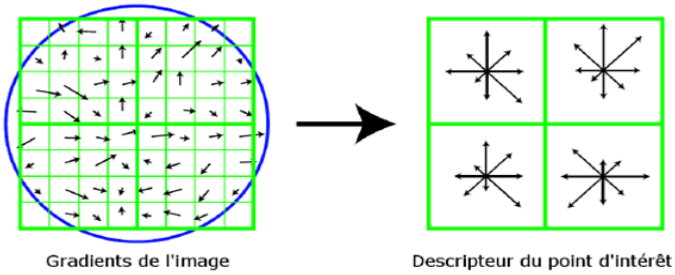


Figure 5.3:*Principe du descripteur SIFT*

Dans une dernière phase, ce vecteur est modifié afin de réduire les effets des variations d'illumination :
- Il est d'abord normalisé à l'unité : une modification du contraste de l'image — qui multiplie la valeur de chaque pixel par une constante — multiplierait tous les gradients par la même constante et n'aurait donc aucun impact sur le vecteur si celui-ci est normalisé. Notons qu'un changement de luminosité — qui ajoute une constante à chaque pixel de l'image — ne modifie pas la valeur des gradients qui sont calculés comme une différence d'intensité entre les pixels voisins. Ainsi, le descripteur est invariant à tout changement affine de l'illumination.

– Des modifications non linéaires des intensités peuvent aussi se produire (saturation, modification des conditions d'illumination de la scène, etc.). Généralement, les effets sont plus importants pour les normes des gradients que pour leurs orientations. L'influence des gradients dont la norme est trop importante est minimisée en seuillant le vecteur normé par une valeur de 0,2 puis en renormalisant le vecteur descripteur. Ceci revient à donner plus d'importance à la distribution des orientations des gradients et à restreindre l'impact des gradients dont la norme est élevée.

Points d'intérêt invariants à l'échelle
Les premières transformations géométriques étudiées dans ce cadre ont été les changements d'échelle. Généralement, on étudie ainsi deux photos prises avec des zooms différents. En considérant un voisinage de taille fixe autour des points d'intérêt, on comprend aisément qu'il soit quasiment impossible d'apparier les deux images.
Plusieurs approches ont été décrites pour obtenir l'invariance aux changements d'échelle. En 1998, les propriétés d'échelle caractéristiques ont été étudiées par Lindeberg. Il propose d'utiliser ce que l'on appelle un espace d'échelle, qui est en fait un ensemble d'images d'une scène représentée à plusieurs niveaux de résolution. Pour les générer, on convolue l'image initiale avec des gaussiennes ayant des écarts-type différents et un espace d'échelle

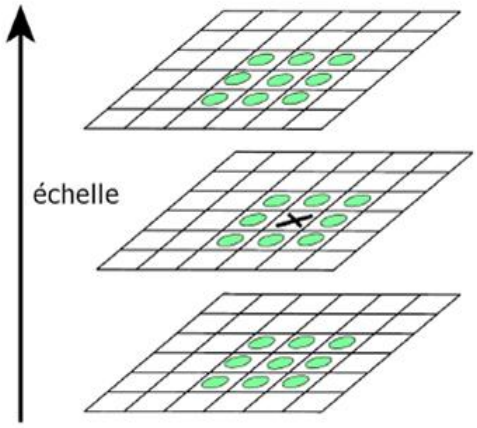


Figure 5.4*:Extrema dans un l'espace d'échelle*

La première étape consiste à détecter les points qui sont stables dans l'espace d'échelle. Pour cela Lowe calcul l'opérateur DoG (Difference of Gaussian) dans l'espace d'échelle puis extrait les extrema locaux. La représentation E dans l'espace d'échelle d'une image I est obtenue par convolution de l'image I avec une gaussienne G(x, y, σ).

$$E(x, y,\sigma)=G(x, y,\sigma)*I(x, y)$$

Où – σ est l'écart-type de la gaussienne G, que l'on fait varier
– * est l'opérateur de convolution en x, y
– I est la fonction intensité de l'image et

$$G(x, y, \sigma) = \frac{1}{2.\pi.\sigma^2}.e^{-(x^2+y^2)/2.\sigma^2}$$

L'opérateur DoG s'écrit alors :

$$DoG(x, y,\sigma)=E(x, y,s\cdot\sigma) - E(x, y,\sigma)$$

Où s est une constante multiplicative permettant de définir un nombre d'intervalle par octave (pour passer d'un octave à l'octave voisin la largeur et la hauteur de l'image sont multipliées 2).

La figure 5.5 donne un exemple d'échelle caractéristique sur deux photos prises avec un zoom différent. Le rapport des échelles correspond au facteur d'échelle entre les deux images.

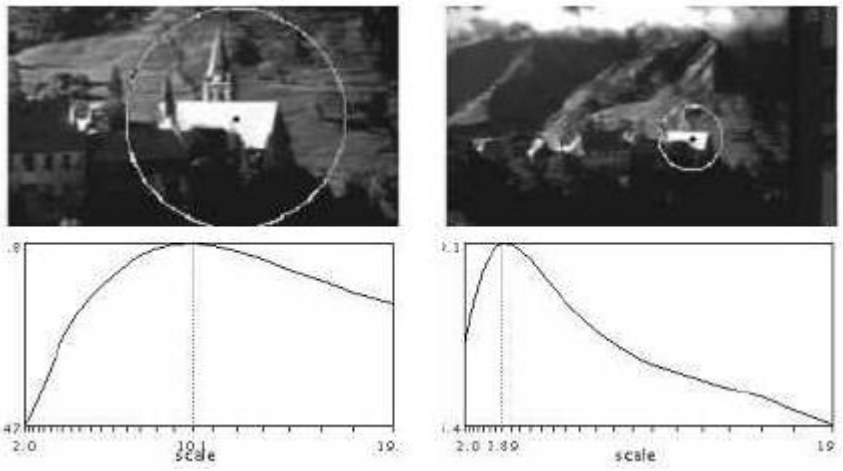


Figure 5.5:*Exemple d'échelle caractéristique*

Pour trouver ces extrema, au lieu de regarder la fonction LoG qui est couteuse à calculer, on approxime avec la fonction DoG (pour Difference of Gaussians)

Les extrema sont alors les pixels qui présentent une intensité maximum, ou minimum, par rapport à leurs voisins directs dans l'image (8 voisins) ainsi qu'à ceux dans l'espace-échelle (9 voisins dans l'échelle précédente et 9 voisins dans l'échelle suivante).

Les extrema de l'espace-échelle ainsi obtenus sont nombreux, pour les filtrer :
1. Les candidats avec peu de contraste sont éliminés.
2. Les réponses correspondants à des contours sont élimines en considérant le Hessien de l'image DoG et un opérateur proche de celui de Harris.

5.2.3 La mise en correspondance

Une fois les points d'intérêt extraits sur différentes images, le processus le plus courant concerne leurs mises en correspondance, ou appariement. Supposons être en présence de deux photos d'un même objet. Après avoir détecté les points d'intérêt sur chacune de ces images, nous souhaitons apparier ceux qui correspondent au même point physique de l'objet. Plusieurs difficultés apparaissent :

Occlusion : Si la position de la caméra par rapport à l'objet est différente, ou si un autre élément de la scène occulte une partie de l'objet qui nous intéresse, certains points détectés sur la première image peuvent ne pas être visibles sur la deuxième et inversement.

Répétabilité du détecteur : En admettant que le point physique soit apparent sur les deux images, comme son apparence est différente, il n'est pas nécessairement retenu par le détecteur de points d'intérêt sur chaque image. C'est la raison pour laquelle nous avons expliqué que le détecteur devait être le plus robuste possible.

Appariement : Enfin, même si le point a bien été détecté sur les deux images, les mettre en correspondance n'est pas toujours aisé.

La somme des différences d'intensité au carré

La plupart des méthodes utilisent des scores afin de déterminer pour un point d'intérêt de la première image, celui de la seconde qui lui « ressemble» le plus. Les scores permettent en outre de trier les pixels candidats dans l'ordre de préférence, ce qui rend possible des stratégies globales d'appariement des points des deux images.

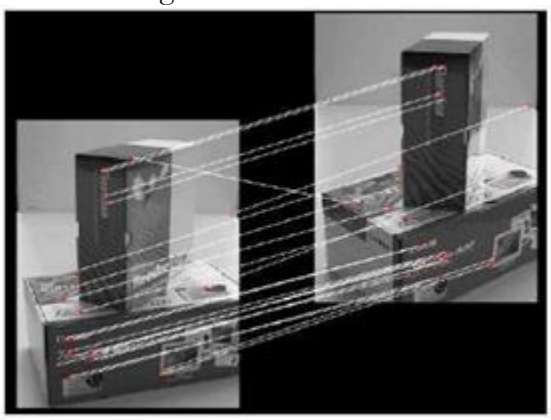


Figure 5.6*: appariement de points d'intérêt*

Il existe plusieurs méthodes de calcul de scores. Ils sont généralement calculés en examinant le voisinage des points d'intérêt.
La méthode la plus simple utilise la SSD (Sum of Square intensity Difference), c'est-à-dire la somme des différences d'intensité lumineuse des pixels situés au voisinage de chaque point au carré. Pour un point M1 de la première image et un point M2 de la seconde, cette quantité se calcule donc ainsi :

$$SSD = \sum_{i=-N}^{N} \sum_{j=-N}^{N} \left[I_1(M_1 + (i,j)) - I_2(M_2 + (i,j)) \right]^2$$

Où :

– N caractérise la taille des voisinages autour des points que l'on considère.
– I_1 et I_2 représente les fonctions intensités des deux images
Pour trouver le point homologue d'un point M_1, la méthode la plus basique consiste à calculer cette quantité pour tous les points d'intérêt M_2 détectés sur la seconde image, et à retenir celui qui minimise la SSD.

Implémentation

Après SIFT caractéristiques de deux images sont calculées, elles doivent être mis en correspondance.
Les mises en correspondance seront utilisées pour calculer la matrice homographie plus tard.

Pour faire correspondre les SIFT caractéristiques, on a adapté un code d'une fonction MATLAB match (imageName1, imageName2). Dans ce code, ils ont accepté deux points caractéristiques ont acceptés d'êtres une paire, si l'angle entre ces caractéristiques sont inférieures à un seuil.

- Chaque caractéristique est un vecteur avec une longueur de 128. On définit le meilleur correspond comme le minimum de la somme de la différence absolue.
- La complexité de la caractéristique de correspondance est très grande. Supposer qu'il y a 1000 points de caractéristique dans chaque image, alors il requiert 128*1000*1000 temps d'addition.

5.2.4 Estimation de la matrice de transformation (RANSAC)

Malheureusement, toutes les correspondances trouvées pendant la phase d'appariement ne sont pas correctes. La méthode pour calculer la position de la caméra doit donc être robuste, en ce sens qu'elle doit pouvoir gérer les mauvaises correspondances. Si le centre des rotations de la caméra est confondu avec le centre optique (i.e. pas de translation de la caméra) et que le centre optique correspond à l'origine du repère de la caméra, alors, la relation qui relie deux images peut s'exprimer sous la forme d'une transformation homographique H [MAN94]. Une homographie 2D s'exprime avec 8 coefficients puisqu'elle est définie à un facteur d'échelle près (cf. annexe).

----- transformation projective

$$X_1 \sim \mathrm{H}\ .\ X_2 = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \bullet \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

où X_1 est un point de l'image défini par ses coordonnées homogènes (u_1, v_1, 1) dans l'image et X_2 sa projection homographique dans l'image 2 est défini par ses coordonnées homogènes (u_2, v_2, 1). Le signe ~ indique la relation d'équivalence, à un facteur d'échelle près, entre le point X_1 et la transformation du point X_2.

$$u_1 = \frac{m_0.u_2 + m_1.v_2 + m_2}{m_{.6}u_2 + m_7.v_2 + 1},\ v_1 = \frac{m_3.u_2 + m_4.v_2 + m_5}{m_6.u_2 + m_7.v_2 + 1},$$

Selon la formule de transformation, nous pouvons obtenir la matrice de transformation [x,y] et [x',y']

$$m_0.u_1 + m_1v_1 + m_2 = u_2(m_6u_1 + m_7v_5 + 1)$$
$$m_3.u_1 + m_4v_1 + m_5 = v_2(m_6u_1 + m_7v_5 + 1)$$

- Les coordonnées (u_1, v_1) du point X_1 dans l'image 1 se déduisent simplement.

Il existe plusieurs solutions permettant de déterminer la matrice H en fonction des données que l'on possède. La matrice H contenant 8 coefficients, cela signifie que seulement 4 points sont nécessaires pour résoudre le système linéaire.
Donc, nous pouvons obtenir 2 équations d'un couple de point de correspondance. Alors il nous faut 4 couples pour avoir la matrice de transformation.

La somme de points de caractéristique de correspondance est relativement grandes, typiquement dans des centaines. Cependant, seulement quatre paires peuvent réparer une transformer la matrice. Le problème est que paire à choisir.
Dans tous les matches points , une grande somme est fausse. C'est surtout du fait que seulement partie de deux chevauchements d'image. Le vrai point de correspondance est appelée "inlier", et le faux point "outlier". Alors l'algorithme d'estimation de la matrice de transformation devrait avoir l'aptitude de pic dehors les "inlier" et utilise les "inlier" pour estimer le plus précis les paramètres

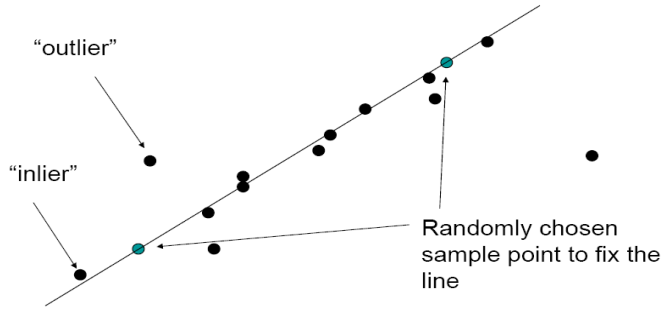


Figure 5.7*:Le principe fondamental de RANSAC*

Dans le cas d'estimation de ligne, nous choisissons au hasard deux points pour fixer une ligne. Alors nous définissons un seuil de distance à distinguer "inlier" et outlier". Le nombre de "inlier" décrit la précision de la ligne. Si l'expérience aléatoire est répétée plusieurs fois, la probabilité de trouve la vrai ligne est grande.

Dans notre cas, le nombre minimum de correspondance pour construire la matrice de transformation est 4. Si nous choisissons aléatoirement 4 paires de points de correspondance pour obtenir la matrice de transformations. Nous utilisons cette matrice de transformation et un seuil de distance pour trouver le nombre de "inlier". Après des milliers d'expériences, nous pouvons obtenir la vrai matrice de transformation.
Quand nous obtenons la matrice de transformation en utilisant RANSAC, nous trouvons que la précision est loin de parfaite. Les deux images alignées avec cette matrice ont le fantôme .

Une amélioration facile peut être faite sans trop beaucoup de coût. Nous emploierons simplement cette matrice de transformation et un seuil relativement grand pour obtenir les "inlier" et "outlier". Après on élimine les "outlier". Alors il y aura plus de proportion de "inlier" dans notre nouveau groupe de points de correspondance. Enfin, nous emploierons RANSAC encore un fois avec un très petit seuil, typiquement un pixel, pour trouver le nouveau nombre de "inlier". A ce moment, la matrice de transformation avec les plus "inlier" avoir une précision très grande.

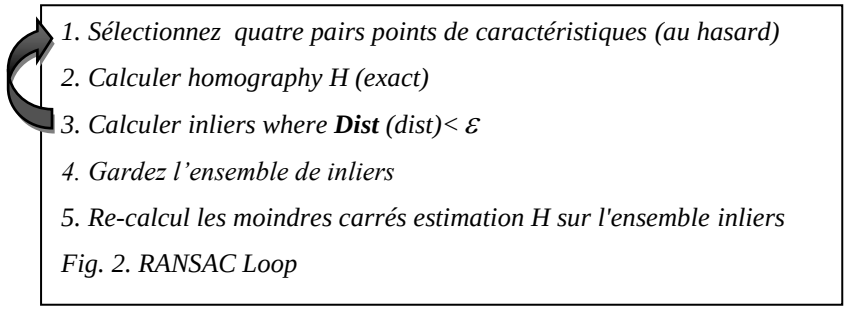


Figure 5.8:*La Boucle RANSAC*

5.2.5 Blending (mélange)

Depuis la matrice de transformation est connu, les deux images peut être simplement alignées ensemble. L'image construite a un évident artefact. Le plus observer directement est la différence d'intensité.
Le gain de compensation est une voie fondamentale à éliminer différence d'intensité.

Implémentation
1. Trouver la région de chevauchement des deux images alignée.
2. calculer l'intensité moyenne de la région de chevauchement des deux images respectivement.
3. Calculer la différence de l'intensité moyenne de la région chevauchée.
4. ajouter la différence de l'intensité moyenne à une de l'image intégralement, l'intensité moyenne des deux images dans la région chevauchée devrait être la même.

Tout d'abord, je crée un masque (la région chevauchée) pour intersection section entre les images. Puis à l'aide de ce masque, on fait le blending des images.

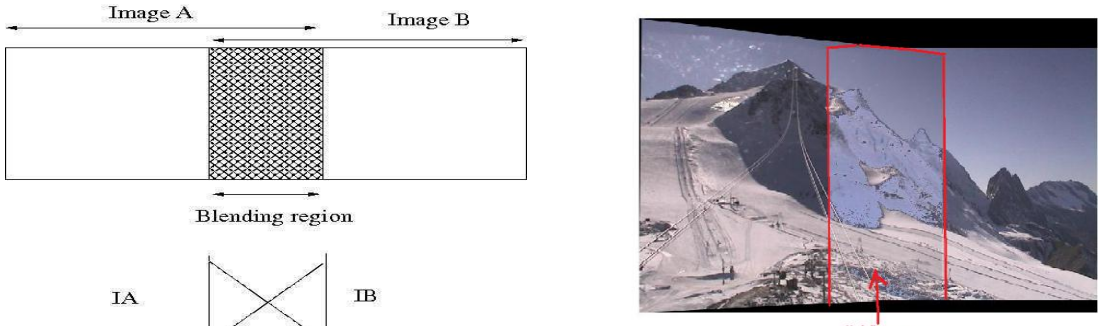


Figure 5.9:*Région de chevauchement*

5.2.6 Démonstration

Le travail réalisé sous Matlab R2007a :version 7.4 .0 .287(R2007a)
Matlab est un environnement de programmation mathématique de haut niveau. En d'autres termes, Matlab permet de faire des calculs, de la visualisation d'information et de la programmation dans un environnement mathématique familier. L'élément de base en MatLab est matrice, il inclut aussi un grand nombre de fonctions mathématiques qui facilitent la programmation. Les graphiques en deux ou trois dimensions représentent un des points forts de Matlab.
Disons que, nous avons donné deux images d'entrée « input images » (voir fig5.10).

Figure 5.10*: Inputs images*

La première étape de l'application calcule les SIFT caractéristiques des images d'entrée (voir Fig 5.11).

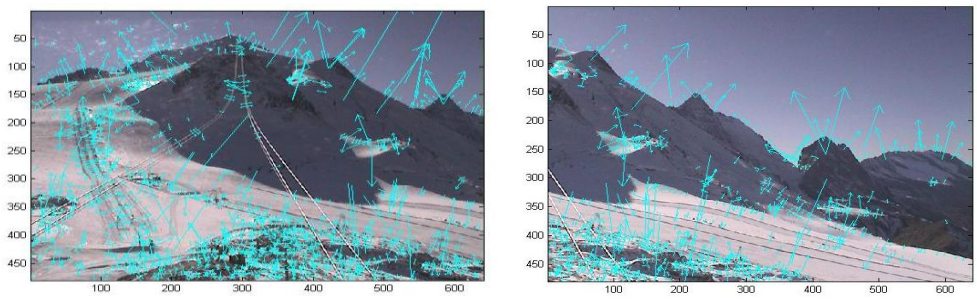


Figure 5.11*:SIFT Features*

Puis la mise en correspondance entre les SIFT caractéristiques extraits (voir Fig5.12).

- ✓ Input image 1 : 1754 points caractéristiques extraits
- ✓ Input image 2 : 916 points caractéristiques extraits

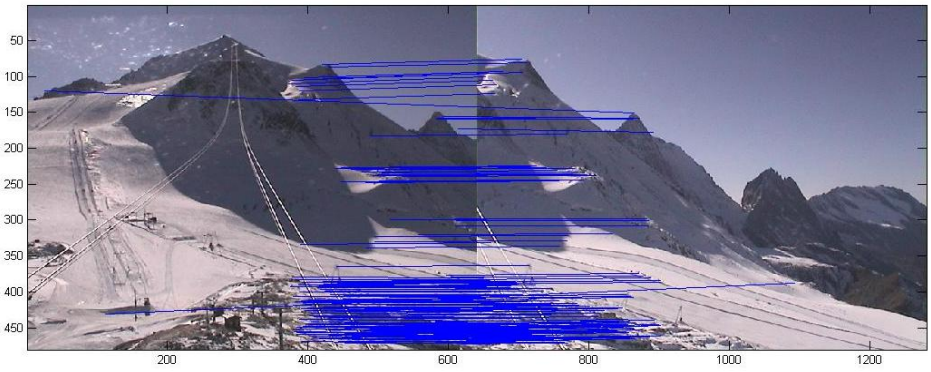


Figure 5.12*:Les points d'intérêts extraits*

- La prochaine étape dans l'application consiste à calculer une matrice homographie. On calcule tous les Inliers et les Outliers « Fig5.13 » (La couleur rouge sont Outliers et le vert sont Inliers.)
  On a trouvé : 147 inliers et 66 outliers.

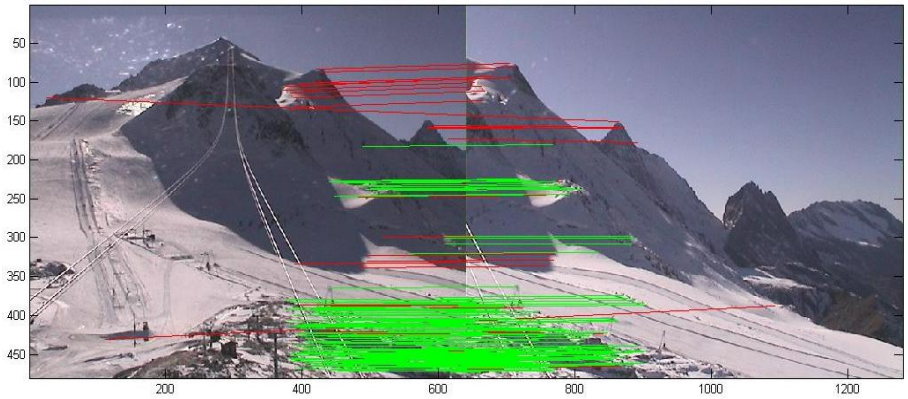


Figure 5.13*:Inliers and Outliers*

Les 4 paires de points de correspondance choisies sont utilisées pour obtenir la matrice de transformations.

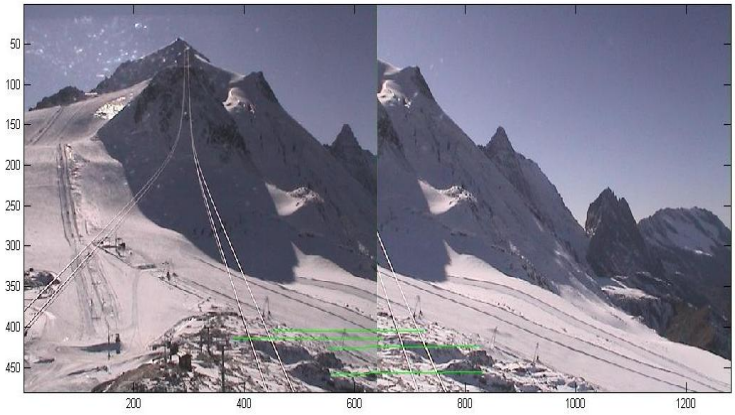


Figure 5.14: *Les 4 meilleurs matches (Best RANSAC*)

globalHomography =

-0.6023 0.0063 220.1320

-0.0549 -0.5789 28.3055

-0.0002 0.0000 -0.4968

Figure 5.15: *Matrice homographie*

Depuis la matrice de transformation est connu, les deux images peut être simplement alignées ensemble, dans un premier temps une création d'un mask d'intersection (voir la Fig 5.16) entre les images m'aide pour faire le blending .

Figure 5.16*:Paronama final*

Le temps total pour construire le panorama est : 12.714000s

Les figures suivantes (Fig5.17) et (Fig 5.18) illustrent un panorama sortie inspirer d'autres travaux.

Figure 5.17: Mosaïque obtenue par [LIS07]

Figure 5.18: Mosaïque obtenue par Autostitch

5.2.7 Résultats et discussion

Nous avons validé notre méthode de construction de mosaïque avec une série d'expérimentations sur des images réelles. Les cas étudiés sont :

➲ ***Autres essais avec une série de deux images***

- Séquence Source d'images N°1 : Date 20/10/2008 à 14h07

Notre Mosaique	Mosaique obtenue par lisa chan	Mosaique obtenue par Autostith
		

Figure 5.19 :Résultats de la série N°1

- Séquence Source d'images N°2: Date 21/10/2008 à 17h37


Notre Mosaique	Mosaique obtenue par lisa chan	Mosaique obtenue par Autostith
		

Figure 5.20: *Résultats de la série N°2*

- Séquence Source d'images N°3: Date 14/10/2008 à12h37


Notre Mosaique	Mosaique obtenue par lisa chan	Mosaique obtenue par Autostith

Figure 5.21 *: Résultats de la série N°3*

➲ ***Autres essais avec une série de trois images***

- Séquence Source d'images S°1(**Conditions météorologiques idéales**) : date 11/10/2008 à 15h37

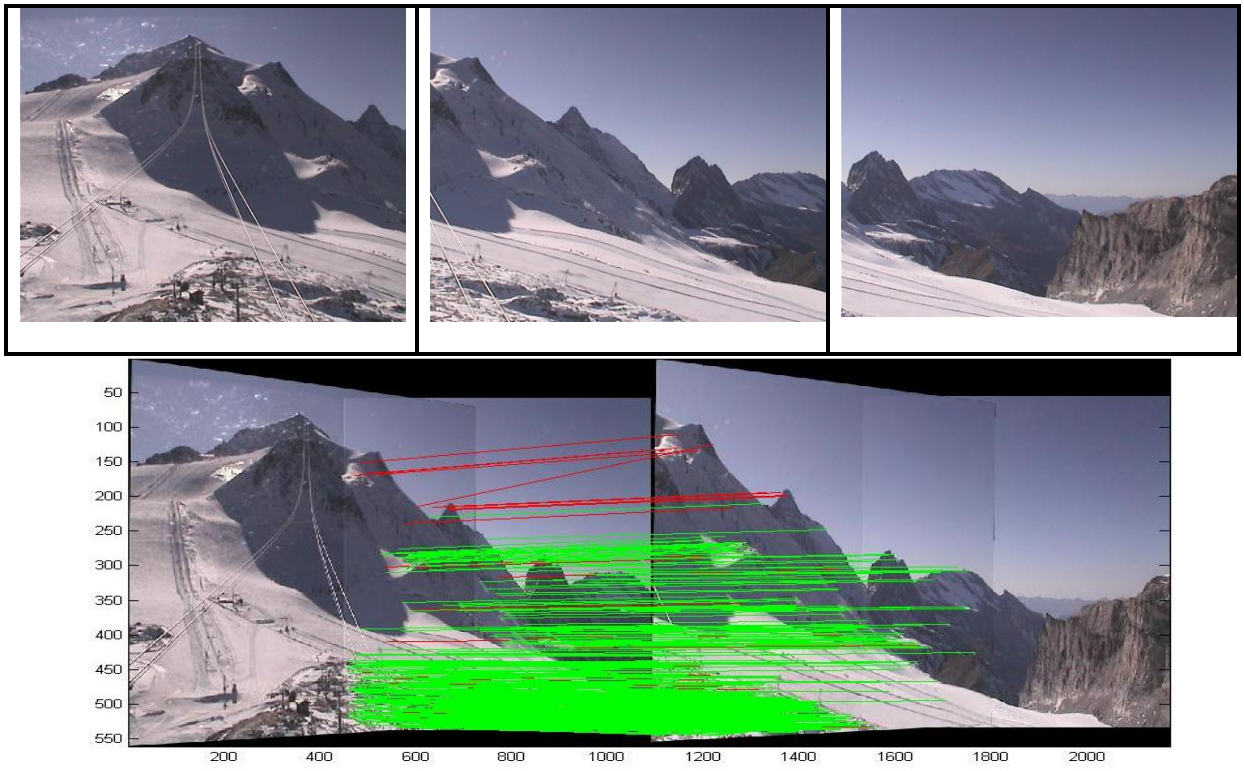


Figure 5.22*: Inliers and Outliers S°1*

- Nombre de Vrai appariements : 458 inliers.
- Nombre de Faux appariements : 9 outliers.

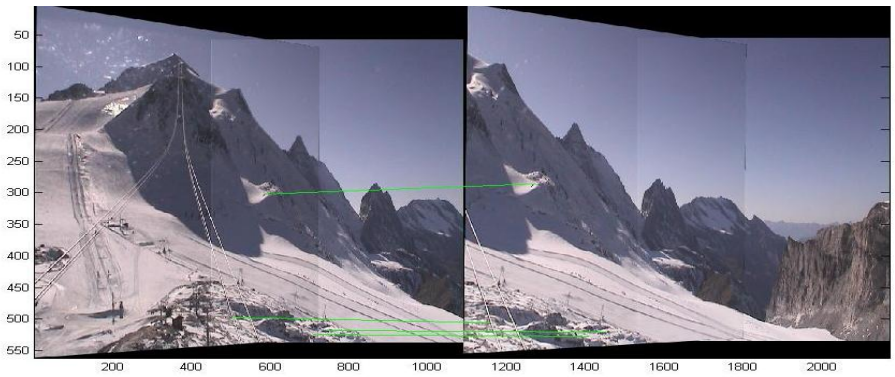


Figure 5.23*: Best RANSAC S°1*

Résultat S°1

Figure 5.24*: Résultats de la série S°1*

- Séquence Source d'images S°2 (**Mauvaise météo**) : date 20/10/2008 à 11h07

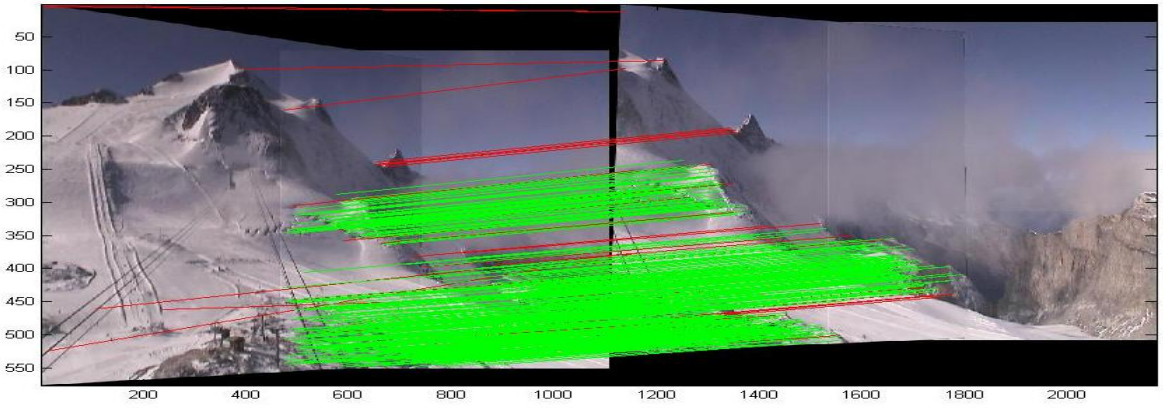


Figure 5.25 *Inliers and Outliers S°2*

- Nombre de Vrai appariements : 481 inliers.
- Nombre de Faux appariements : 47 outliers.
-

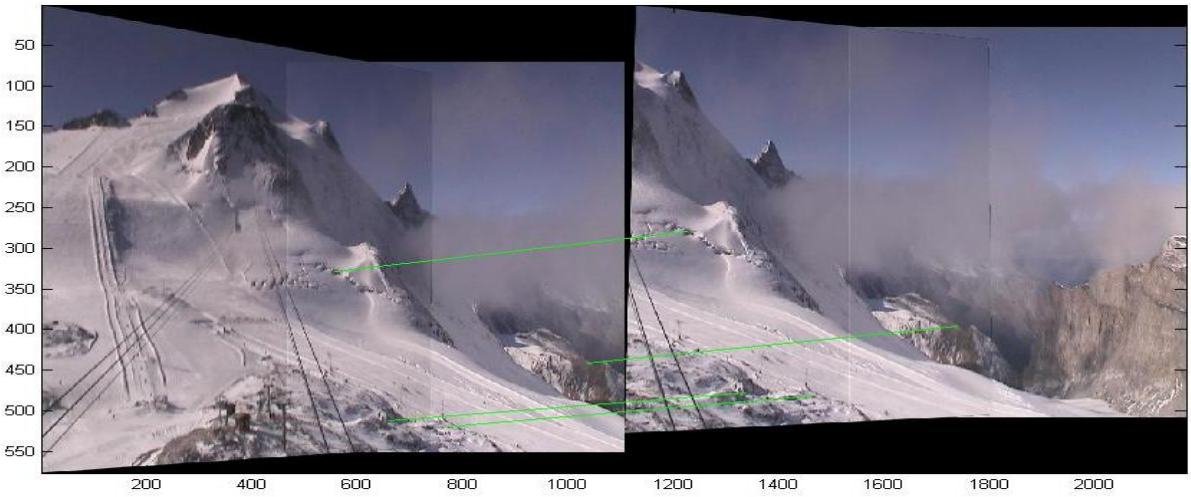


Figure 5.26*: Best RANSAC S°2*

Résultat S°2

Figure 5.27*: Résultats de la série S°2*

Remarque

Les résultats que nous avons présentés ci-dessus montrent la qualité des mosaïques obtenues à partir de plusieurs tests empiriques de notre système. Nous avons constatés que notre application donne des résultats excellents par rapport d'autres travaux ou des logiciels commerciaux.

Affichage des panoramas en utilisant l'image en direct

Pour visualiser du panorama on utilise Live Picture browser, version 1.0. Car ce browser s'exécute en Java, sans nécessité de plugin spécial n'importe qui avec un navigateur compatible Java Web devrait être en mesure d'afficher notre panorama. Il est assez facile à mettre en place.

Étape 1. Vous avez besoin de deux fichiers

pano.jpg image panorama cylindrique en format jpeg

IVR pano.ivr fichier en précisant le champ de vision et de l'emplacement de Java applets

Le fichier IVR devrait ressembler à:

```
#VRML V2.0 utf8
NavigationInfo {
    type "VISTA"
    headlight FALSE
}
Vista {
    texture ImageTexture { url "./pano.jpg" }
    type "CYLINDER"
    vFov -0.5 0.5
    pitchRange -0.5 0.5
}
```

Étape 2. Télécharger les deux fichiers[7] suivants

- ✓ lpjpano.zip
- ✓ lpjpano.cab

et les mettre dans le même répertoire avec pano.jpg et pano.ivr

Étape 3. Donnez un simple fichier html qui indique où est pano.ivr.
Ces instructions ont été testées

5.2.7.1 Observation

Nous avons utilisé plusieurs seuils pour améliorer la qualité de la distribution des points d'intérêt et ainsi obtenir le nombre de points nécessaires pour l'étape de la mise en correspondance
Pour observer l'effet de seuil, sur panorama d'images final. Avec des seuils de 1, 0.1 et 0.001, 0.0001.
On peut constater que le nombre des Inliers devient plus petit quand on diminuera le seuil, et aussi qu'un nombre inlier même faible n'affecte pas les résultats sont visuellement identiques. Seul le changement est observé, comme on l'ai dit, du nombre de inlier mais à partir de seuil de 0.0001 le nombre de inlier s'arrêt de se diminuer et le résultat final est inacceptable la même chose avec le seuil plus élevés le nombre de inliers devient plus grand jusqu'à, on nous obtient que 0 (zéro) Outlier c'est-à-dire que tous les matches points sont des inliers et ça aussi affecte le résultat final à partir un certain seuil élevé (100000).

Une chose à noter sur les seuils, c'est que, trop petites et trop élevé seuils, la probabilité d'obtenir le meilleur de quatre matches qui ne sont pas effectivement très approprié pour d'autres paires, on croit qu'il existe un seuil qui devrait être appelé approprié. Donc, en ayant quelques observations empiriques sur le mosaicing à partir de notre images, on décide que le «seuil » qui a un faible risque de hasard doit être de 1, dans notre cas.

Le second paramètre, le nombre de fois que s'exécute RANSAC, également il ne produit aucuns changements notables dans le résultat final. Tout ce que on peut dire que, un nombre raisonnable, comme 1.000, voire 100 suffit à RANSAC pour arriver aux meilleures quatre matches presque tout le temps. Et De même, si RANSAC ne peut pas trouver les quatre points de correspondance après la 1.000 loop, il ne peut pas trouver ces 04 matches à 10.000 boucle ou plus.
Le choix ces deux paramètres affecte aussi le temps d'exécution car un trop petit ou trop élevé seuil ou un nombre très grand de boucle RANSAC augmente le temps d'exécution du programme, mais avec un seuil de 1 est 1000 RANSAC loop le temps d'exécution ne dépasse pas de secondes.

L'application donne des résultats excellents pour presque panoramas de deux images. De même, il a également produit un bon panorama de quatre images par rapport les autres travaux. Un défaut de la mise en œuvre est cependant panoramas large. Si il ya beaucoup d'images qui sont reliés les uns aux autres, on ne peut les ensemble.

[7] http://pages.cs.wisc.edu

Conclusion et Perspectives

A conclusion is simply the place where someone got tired of thinking.
Arthur Block

La construction d'une mosaïque est un sujet important dans le domaine de la vision par ordinateur. Cette importance est vue clairement dans plusieurs situations où il est nécessaire d'avoir une image panoramique composée de plusieurs images distinctes. Une telle situation peut apparaître dans le cas où la taille d'un paysage dépasse le champ visuel de la caméra. Une mosaïque est construite en de quatre étapes que nous avons abordées dans notre mémoire. Ces étapes sont : la détection des points d'intérêt, la mise en correspondance, la calibration des caméras et la projection des images sur un plan cylindrique.
Dans le cas idéal, une caméra PTZ devrait nous permettre de disposer des paramètres exacts de la prise de vue. Dans la réalité ce n'est pas forcément le cas. Avec certains types de matériel, il n'est tout simplement pas possible de connaître ces paramètres. Avec d'autres caméras, comme celle que nous utilisons, la précision de la commande n'est pas suffisante pour obtenir une projection robuste avec simplement les données de la caméra. Nous devons donc déterminer les paramètres de la prise de vue à partir des données contenues dans les images. L'approche que nous avons retenue est d'utiliser le recalage d'images pour déterminer la matrice de projection. Le but du recalage d'images est d'aligner géométriquement deux images ou plus de sorte que des pixels respectifs ou leurs dérivés (bords, coin, etc..) représentant la même structure fondamentale puissent être mis en correspondance. L'idée fondamentale derrière la plupart des algorithmes décrits dans la littérature est de mettre en correspondance les images selon leurs propriétés radiométriques ou géométriques en utilisant une fonction spécifique pour évaluer la qualité de la mise en correspondance.

Nous avons cherché à améliorer et à corriger le problème lié à l'illumination de la scène. En fonction de l'angle de prise de vue, la scène peut être plus ou moins éclairée. Pour corriger ce changement de luminosité, nous pouvons soit fixer les paramètres d'ouverture et de gain de la caméra, soit laisser la caméra adapter automatiquement ces paramètres au mieux en fonction de la luminosité de la portion de scène visée. Dans le premier cas si la dynamique entre les zones sombres et les zones fortement éclairées est trop forte, les zones sombres risquent d'apparaître sous-exposées et les zones claires sur-exposées. Dans le second cas, nous allons voir apparaître des pavés de luminosités différentes avec ce que nous appelons communément des coutures. Il existe aujourd'hui des caméras qui adaptent automatiquement le gain localement en fonction de la luminosité. Ces caméras sont particulièrement bien adaptées pour gérer les contre-jours. Dans le cas, malheureusement le plus courant, où il n'est pas possible de fixer le gain de la caméra, Cependant, cette correction de luminosité n'est réellement utile que lorsque nous désirons construire une mosaïque d'images pour ensuite la visualiser. Dans le cas d'une application de détection et de suivi, la fréquence d'acquisition est plus rapide. De fait, le changement de

luminosité entre deux images successive est relativement faible et il peut être lissé par la modélisation du fond que nous avons mise en place. Un autre problème que l'on rencontre classiquement lors de la construction d'un panorama avec des images qui ne sont pas acquises dans le même temps, est la suppression des fantômes. A la différence d'un appareil photo, l'avantage d'utiliser une caméra est que nous pouvons traiter un flux vidéo plutôt qu'une image unique. Afin de supprimer les fantômes dans nos panoramas, nous avons contourné la difficulté. Pour chaque prise de vue, nous faisons l'acquisition d'une séquence d'images et nous calculons une modélisation des composantes statiques de la scène observée à partir d'un algorithme basé sur les mélanges de gaussienne

Nous avons traité la première étape qui consiste à détecter les points d'intérêt dans les différentes images. Cette détection est faite en utilisant le détecteur de Harris. Nous avons utilisé plusieurs seuils pour améliorer la qualité de la distribution des points d'intérêt et ainsi obtenir le nombre de points nécessaires pour l'étape suivante de la mise en correspondance.

Après avoir détecté les points d'intérêt, la deuxième étape étudiée sert à utiliser ces points pour faire la mise en correspondance. Cette partie est cruciale dans le domaine de la vision par ordinateur. Une détection erronée d'une paire de points parmi plusieurs paires valides conduit à un échec complet dans la construction de la mosaïque. Cette caractéristique nous force à concevoir une méthode plus précise pour assurer la fiabilité des paires mises en correspondants. La continuation de cette étape amène à la calibration des caméras. Dans cette partie, nous avons implanté l'algorithme RANSAC pour calculer la matrice de transformation qui est ensuite utilisée dans l'étape finale. Cette dernière consiste à projeter les images en les utilisant sur la projection cylindrique.

Nous constatons que les étapes de la construction d'une mosaïque sont interdépendantes, dans le sens que la réussite ou l'échec d'une étape induit respectivement une amélioration ou une dégradation dans l'étape suivante. C'est pour cette raison que notre travail vise à accomplir une construction correcte d'une mosaïque en réduisant au maximum les erreurs de chaque étape.

Rappelons que l'objectif initial de notre projet était de réaliser une construction automatique et correcte d'une mosaïque à partir d'un ensemble d'images données. Nous avons atteint ce but dans ce mémoire en implantant différentes techniques pour le choix des points d'intérêt ainsi que pour la mise en correspondance. Les résultats que nous avons présentés montrent la qualité des mosaïques obtenues à partir de plusieurs tests empiriques de notre système**.**

Même si l'ensemble de ces travaux de recherche a donné lieu à la fois à des publications dans les actes de conférences internationales mais aussi à des applications commerciales, il reste beaucoup de travail à faire. Parfois un panorama automatiquement généré peut ne pas être convaincante réalistes, les panoramas sont tout de même esthétiques.

Finalement, une direction pour des travaux futurs serait d'intégrer la possibilité de combiner une image d'entrée avec une banque d'images importante pour créer un panorama infini. L'algorithme effectue correspondant scène en utilisant l'image d'entrée d'origine pour trouver les meilleures scènes correspondant voisins et les compose ensuite ces images d'une manière transparente.

Bibliographie

[BAL07] Balland B., Optique géométrique : Imagerie et instruments, Brochè, ISBN : 978-2-88074- 689-6, 2007

[BAR03] Bartoli A., Dalal N., Horaud R., "Motion panoramas", Technical Report RR-4771, INRIA, 2003.

[BAR03-1] Bartoli A., Dalal N., Horaud R., Des séquences vidéo aux panoramas de mouvement, actes Coresa 2003

[BEN01] R. Benosman, S.B. Kang. Panoramic Vision: Sensors, Theory, Applications. Springer, 2001.

[BER01] P. Bertolino, G. Foret, D. Pellerin, Détection de personnes dans les vidéos pour leur immersion dans un espace virtuel, 18ème colloque sur le traitement du signal et des images (GRETSI'01), T2, pages 153-156, Toulouse, France, Sept. 2001.

[BEV05] Bevilacqua, A., di Stefano, L., Azzari, P., An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. (2005) 511-516

[BEV06] Bevilacqua, A., Azzari, P., High-quality real time motion detection using ptz cameras. (2006) 23-23

[BHA00] Bhat, K.S., Saptharishi, M., Khosla, P.K., Motion detection and segmentation using image mosaics, IEEE International Conference on Multimedia and Expo (III). (2000) 1577 1580

[BOS01] Lo Bosco, A genetic algorithm for image segmentation, Proceedings. 11th International Conference on Image Analysis and Processing, Volume , Issue , 26-28 Sep 2001 Page(s):262 – 266

[BRO92] L.G. Brown. A survey of image registration techniques. ACM Computing Surveys, 24(4):325–376, 1992.

[BRO03] Brown, M., Lowe, D.G., Recognising panoramas, Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003

[CAN03] Frank M. Candocia, Simultaneous Homographic and Comparametric Alignment of Multiple Exposure-Adjusted Pictures of the Same Scene, IEEE Transactions on Image Processing, vol. 12, n° 12, 2003

[CHE95] S.E. Chen, QuickTime VR : An Image-based Approach to Virtual Environment

[CHU04] Albert C. S. Chung, J. Alison Noble, and Paul Summers, Vascular Segmentation of Phase Contrast Magnetic Resonance Angiograms Based on Statistical Mixture Modeling and Local Phase Coherence, IEEE Transaction on Medical Imaging, vol. 23, No. 12, Dec. 2004.

[COH89] L.Cohen, L.Vinet, P.T. Sander et A. Gagalowicz, Hierarchical region based stereo matching in Proceedings of the International Conference on Computer Vision and Pattern Recognition, San Diego CA, pages 416-421, juin 1989

[DAR05] Dar-Shyang Lee, Effective Gaussain Mixture Learning for Video Background Substraction, IEEE Transaction On Pattern Analysis And Machine Intelligence, vol.27, no. 5, May 2005.

[DOU02] Matthijs Douze, Vincent Charvillat, Bernard Thiesse, Des mosaïques plus robustes, plus précises, RFIA'02, Angers, janvier 2002

[DOU03] Matthijs Douze, Philippe Puech, Vincent Charvillat, Jean Conter, Suivi temps-réel séquences vidéo dans un panoramique pour le codage par objets, Coresa 2003, Lyon janvier 2003

[FAU93] O. Faugeras, Three-Dimensional Computer Vision: a Geometric View-point, The MIT Press, 1993.

[FIA02] Mark Fiala, Anup Basu, Feature extraction and calibration for stereo reconstruction using non-svp optics in a panoramic stereovision sensor, Proceedings of the 3rd IEEE Workshop on Omnidirectionnal vision, pp 79-86, 2002.

[FIS81] M. A. Fischler and R. C. Bolles, Random sample consensus: a paradigm for model fittin with applications to image analysis and automated cartography, ACM, 24(6):381-395, 1981.

[GOL89] Goldberg D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Educational Publishers Inc, ISBN-10: 0201157675, 1989

[GOU00] V. Gouet, Mise en Correspondance d'Images en Couleur : Application à la synthèse de vues intermédiaires, PhD thesis, Université Montpellier II, 2000.

[GRA09] Gracias N., Mahoor M., Negahdaripour S., Gleason A., Fast image blending using watersheds and graph cuts, Image and Vision Computing, Volume 27, Issue 5, Pages 597- 607, 2009

[HAR88] Harris C., Stephens M., A Combined Corner and Edge Detector, Proceedings of The Fourth Alvey Vision Conference, Manchester, England (1988) 147-151

[HAR00] I. Haritaoglu, D. Harwood, L.S. Davis, real-time surveillance of people and their activities, IEEE Transaction On Pattern Analysis And Machine Intelligence Vol. 22, pages 809–830, 2000.

[HAR04] Hartley, R.I., Zisserman, A., Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)

[HON91] J. Hong, X. Tan, B. Pinette, R. Weiss, and E.M. Riseman, Image-based homing, IEEE International Conference on Robotics and Automation, pp 620 –625, vol.1, 9-11 April 1991.

[HSU96] S. Hsu, P. Anandan, Hierarchical Representations for Mosaic Based Video Compression, Proc. Picture Coding Symp. pp. 395-400, Mar. 1996.

[HUA98] H.-C. Huang, Y.-P. Hung, Panoramic stereo imaging system with automatic disparity warping and seaming, Graphical Models and Image Processing, 60(3):196–208, May 1998.

[IRA96] M. Irani, P. Anandan, J. Bergen, R. Kumar, et S. Hsu. Efficient representations of video sequences and their applications, Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, analysis, and Application, 8(4), 1996.

[KAN99] Kanatani K, Ohta N. Accuracy bounds and optimal computation of homograph for image mosaicing applications. The Seventh International Conference on Computer Vision, 1999.

[KAN03] S. Kang, J. Paik, A. Koschan, B. Abidi, M. A. Abidi, Real-time video tracking using PTZ cameras, Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, Vol. 5132, pp. 103-111, Gatlinburg, TN, May 2003.

[KIM04] Kyungnam Kim, TH Chalidabhongse, D Harwood, L Davis, Background modeling and subtraction by codebook construction, Image Processing, 2004. ICIP '04. 2004 International Conference on, Vol. 5 (2004), pp. 3061-3064 Vol. 5.

[KOE87] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. Biological Cybernetics, 1987.

[LEE93] C.Y. Lee, D.B. Cooper et D. Keren, Computing correspondence based on region and invariants without feature extraction and segmentation in Proceedings of the International Conference on Computer Vision and Pattern Recognition, pages 655-656, 1993

[LEE99] K.S. Lee, Y.F. Fung, K.H. Wong, S.H. Or, T.K. Lao, Panoramic Video Representation using Mosaic Image , Proc. of CISST'99, pp. 390-396, Las Vegas, USA, June 1999.

[LEM07] Victor Lempitsky, Denis Ivanov, Seamless Mosaicing of Image-Based Texture Maps, cvpr , pp. 1-6, 2007.

[LER95] I.C. Lerman, R.F. Ngouenet, Algorithmes génétiques séquentiels et parallèles pour une représentation affine des proximités, Rapport de recherche de l'INRIA RR2570, Janvier 1995

[LIS07] Lisa H. Chan, Alexei A. Efros , Automatic Generation of An Infinite Panorama ,*Carnegie Mellon University*,2007

[LOWE] D. Lowe. Demo Software: SIFT Keypoint Detector. *The University of British Columbia.* Available from: http://people.cs.ubc.ca/~lowe/keypoints/.

[LOW04] Lowe D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2) (2004) 91-110

[LOW07] Brown, M. and Lowe, D. G. 2007. Automatic Panoramic Image Stitching using Invariant Features. *Int. J. Comput. Vision* 74, 1 (Aug. 2007), 59-73.

[MAN94] S. Mann and R.W. Picard, Virtual Bellows: Constructing High Quality Stills from Video, ICIP, Vol. 1, pp. 363-367, 1994

[MAN95] S. Mann and R. W. Picard, On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures, in Proc. of IS&T 48th Annual Conference, May 1995, pp. 422–428.

[MIT96] Melanie Mitchell. An introduction to genetic algorithms. The MIT Press, 1996.

[MOR77] H. Moravec, Towards automatic visual obstacle avoidance, 5ème Procq. Joint Conf. Artificial Intell., Cambridge, page 584, 1977

[NAY97] S.K. Nayar. Omnidirectional video camera, In Proceedings of the 1997 DARPA Image Understanding Workshop, May 1997

[NEL65] J. A. Nelder and R. Mead, A simplex method for function minimization, Computer Journal 7 (1965), 308 - 313.

[PAV01] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, S. Harp, Urban Surveillance Systems: From the Laboratory to the Commercial World, IEEE Proceedings, Vol. 89, No. 10, pages 1478- 1497, Oct. 2001

[PEE02] P. Peer and F. Solina, Panoramic depth imaging: Single standard camera approach, International Journal of Computer Vision, vol. 47, pp. 149-- 160, 2002.

[SAT04] T. Sato, S. Ikeda, M. Kanbara, A. Iketani, N. Nakajima, N. Yokoya, and K. Yamada, Highresolution video mosaicing for documents and photos by estimating camera motion, Proc. SPIE Electronic Imaging, Vol. 5299, pp. 246-253, Jan. 2004

[SCH97] C. Schmid, R. Mohr. Local grayvalue invariants for image retrieval. PAMI, 19(5) :530-534, 1997.

[SCH00] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of Interest Points Detectors, International Journal of Computer Vision, vol 37(2), p151-172, 2000

[SHU00] H. Shum, R. Szeliski, Construction of Panoramic Image Mosaics with Global and Local Alignment, International Journal of Computer Vision, Vol. 36(2), pp. 101-130, Feb. 2000

[SIN04] Sinha, S., Pollefeys, M., Kim, S.: High-resolution multiscale panoramic mosaics from pantilt- zoom cameras. In: Proceedings of the 4th Indian Conference on Computer Vision, Graphics and Image Processing. pp: 28-33, 2004

[SOU96] D. Southwell, A. Basu, M. Fiala, J. Reyda. Panoramic stereo, International Conference on Pattern Recognition, 1996.

[STA99] Stauffer C., Grimson W., Adaptative background mixture models for real-time tracking, CVPR99, 1999

[STU02] Peter Sturm, Mixing catadioptric and perspective cameras, Proceedings of the 3rd IEEE Workshop on Omnidirectionnal Vision, pages 37-44, June 2002.

[SUG04] Y. Sugaya and K. Kanatani, Extracting moving objects from a moving camera video sequence, Symposium on Sensing via Image Information, pages 279–284, June 2004.

[SZE94] R. Szeliski Image Mosaicing for Tele-Reality Applications, Proc. IEEE Workshop on Applications of Computer Vision, IEEE CS Press, Los Alamitos, Calif., pp. 44-53, 1994.

[SZE97] R. Szeliski, H.-Y. Shum, Creating Full View Panoramic Mosaics and Environment Maps, Computer Graphics (SIGGRAPH 97), pp 251-258, 1997.

[TAN04] Tan K.H, Hua H., Ahuja N., Multiview Panoramic Cameras Using Mirror Pyramids. IEEE Trans. Pattern Anal. Mach. Intell. 26, 7 (Jul. 2004), 941-946, 2004

[TOR96] P. Torr and A. Zisserman, MLESAC: A new robust estimator with application to estimating image geometry, Computer Vision and Image Understanding, vol. 78, pp 138-156, 2000

[TRI00] Triggs B., McLauchlan P., Hartley R., Fitzgibbon A, Bundle Adjustment : A Modern Synthesis, Vision Algorithms: Theory and Practice, Springer-Verlag, pp 298-372, 2000

[UYT02] M. Uyttendaele, A. Eden, and R. Szeliski, Eliminating ghosting and exposure artifacts in image mosaics, In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pp 509-516, Kauai, Hawaii, December 2001.

[XIO97] Y. Xiong, K. Turkowski, Creating image based vr using a self-calibrating fisheye lens, In CVPR97, pages 237--243, 1997

[YAG90] Y. Yagi, S. Kawato, Panoramic scene analysis with conic projection In IROS90, 1990.

[ZAI01] M. Zaim, A. El ouaazizi, R. Benslimane, Genetic Algorithms Based Motion Estimation, 2001.

[ZHE99] Zhiqiang Zheng , Han Wang , Eam Khwang Teoh, Analysis of gray level corner detection, Pattern Recognition Letters, v.20 n.2, p.149-162, Feb. 1999

[ZIT05] Zitovà B., Flusser J., Sroubek P., Image Registration: A Survey and Recent Advances, ICIP 2005

[ZHE99] Zhiqiang Zheng , Han Wang , Eam Khwang Teoh, Analysis of gray level corner detection, Pattern Recognition Letters, v.20 n.2, p.149-162, Feb. 1999

Annexe

Annexe

« Si la géométrie oblige à contempler l'essence, elle nous convient ; si elle s'arrête au devenir, elle ne nous convient pas. (...) Elle a pour objet la connaissance de ce qui est toujours et non de ce qui naît et périt. Par suite, mon noble ami, elle attire l'âme vers la vérité, et développe en elle cet esprit philosophique qui élève vers les choses d'en haut les regards que nous abaissons à tort vers les choses d'ici-bas. Il faut donc, autant qu'il se peut, prescrire aux citoyens de ta Callipolis de ne point négliger la géométrie ; elle a d'ailleurs des avantages secondaires qui ne sont pas à mépriser. Ceux que tu as mentionnés, et qui concernent la guerre ; en outre, pour ce qui est de mieux comprendre les autres sciences, nous savons qu'il y a une différence du tout au tout entre celui qui est versé dans la géométrie et celui qui ne l'est pas. » **Platon** (428-347 av. J. C.), La République, Livre VII, 526

1.1 1.1 Modèles mathématiques

1.1.1 Avant propos

Nous rappelons dans ce chapitre quelques éléments fondamentaux de l'optique géométrique ainsi que quelques définitions. Après ces quelques notions simples, nous présentons un résumé du modèle sténopé. Pour plus d'information, le lecteur pourra se référer aux ouvrages de O. Faugeras, "Three-Dimensional Computer Vision: a Geometric View-point" [FAU93], de Hartley, "Multiple View Geometry in Computer Vision." [HAR04] et B. Balland, « Optique géométrique : Imagerie et instruments » [BAL07].

1.1.2 Eléments d'optique géométrique

Nous allons formaliser ici quelques concepts simples de l'optique géométrique.

1.1.2.1 Les bases de l'optique géométrique

Le concept de rayon de lumière introduit par Euclide est l'élément de base de l'optique géométrique. Ce rayon de lumière n'a pas d'existence physique, mais il indique la direction de propagation de la lumière. La propagation de la lumière est étudiée en lui associant une infinité de rayons de lumière indépendants les uns des autres. Un ensemble de rayons de lumière provenant d'une même source est appelé faisceau. Cette représentation a pour principal intérêt de dissocier le trajet de la lumière de l'onde électromagnétique dont elle est composée, ceci afin de permettre de traiter les problèmes d'optiques par de simples constructions géométriques. Avec le concept de rayon de lumière, l'optique géométrique repose sur quelques principes et lois simples. Tout d'abord, le principe de la propagation rectiligne dans un milieu transparent, homogène et isotrope. C'est l'un des premiers principes à avoir été énoncé. L'air, par exemple, est un milieu qui ne répond pas à cette définition puisque son indice absolu varie en fonction de la température et de la pression. Une autre limite du

domaine de validité de la propagation rectiligne de la lumière correspond au phénomène de diffraction que l'on peut observer lorsque l'on fait passer un faisceau de lumière à travers un trou mince. Un autre principe important est l'indépendance des rayons différences et dont les directions se croisent en un point, le point d'intersection ne provoque pas d'interférence particulière entre les rayons. Un dernier principe fondamental correspond au *retour inverse* de la lumière. Le trajet emprunté par la lumière dans un sens est le même dans l'autre sens (réciprocité du rayon incident). Parmi les lois de l'optique géométrique, celles de Snell-Descartes sont particulièrement importantes. Elles décrivent les phénomènes lumineux engendrés par l'impact de la lumière sur la matière : la réflexion et la réfraction. Pour reprendre la vision de Descartes, la réflexion correspond au « rebond » de la lumière sur une surface plane. De façon plus formelle, le rayon réfléchi est dans le plan d'incidence, plan défini par le rayon incident et la normale à la surface réfléchissante. La réfraction est la déviation des rayons lumineux passant obliquement d'un milieu transparent à un autre milieu transparent d'indice différent. Lors de la réfraction sur la surface dioptrique, le rayon de lumière coupe la normale et se réfracte selon une direction définie par l'angle *i2* lié à l'angle d'incidence *i1* par la relation des sinus :

$$n1 .\sin i_1 = n2 .\sin i_2$$

où n1 et n2 sont les indices de réfraction des deux milieux.

1.1.2.2 Systèmes optiques

Un système optique est un ensemble de milieux transparents, homogènes et isotropes d'indices de réfraction différents disposés les uns à la suite des autres. Ils sont séparés par des surfaces polies, appelées dioptres, de formes géométriques simples (plans, sphères) ou plus complexes (paraboloïdes, ellipsoïdes, hyperboloïdes). Ces surfaces peuvent être réfringentes (ou réfractantes). C'est à dire qu'une fraction plus ou moins importante de l'énergie incidente est transmise d'un milieu à l'autre. Dans le cas où la *quasi* totalité de l'énergie incidente est réfléchie, on parle alors de surface réfléchissante (miroir). Les systèmes optiques peuvent être classés en trois catégories :

- les systèmes dioptriques ne comportant que des surfaces réfringentes. Ces systèmes ont une face d'entrée et une face de sortie distincte.
- les systèmes catoptriques ne comportant que des surfaces réfléchissantes.
- les systèmes catadioptriques comportant à la fois des éléments réfringents et réfléchissants.

1.1.2.3 Conditions de Gauss

Afin de simplifier l'étude des phénomènes optiques, nous nous placons généralement dans les conditions de Gauss. C'est en publiant ses « Dioptrische Untersuchungen » entre 1838 et 1841 que Carl Friedrich Gauss (1777 – 1855) introduit les notions de plans principaux et points principaux. Son objectif est, sous réserve de certaines conditions (dites conditions de Gauss), de disposer d'un outil mathématique permettant justement de faciliter l'étude des instruments optiques (télescope, jumelle). L'approximation de Gauss consiste, en lumière monochromatique, à n'utiliser que les trajets des rayons pour lesquels le système fonctionne dans des conditions de stigmatisme approché. Le stigmatisme réel stipule que l'image d'un point est un point. Le stigmatisme est dit approché lorsque l'image d'un point est une tache suffisamment petite pour être considérée comme un point. Pour être dans les conditions de Gauss :

- les points objets doivent être voisins de l'axe optique,
- les rayons utilisés pour la formation des images sont très peu inclinés.

1.1.3 Le modèle Sténopé

1.1.3.1 Camera observa

L'acquisition d'une image à partir d'un capteur CCD est un phénomène complexe. Le phénomène optique est plus simple à comprendre puisqu'il s'agit d'un phénomène lumineux naturel que l'on peut expérimenter facilement. C'est l'expérience très simple de la chambre noire. Aristote est vraisemblablement le premier à avoir expérimenté ce phénomène. Le 25 janvier 1544 à Louvain, Reinerus Gemma-Frisius a utilisé le principe de la chambre noire ou « caméra observa » pour observer une éclipse du soleil. Il utilisera plus tard l'illustration suivante dans son livre « de radio astromomica et géométrico » publié en 1545. On pense que c'est la première illustration de la chambre noire.

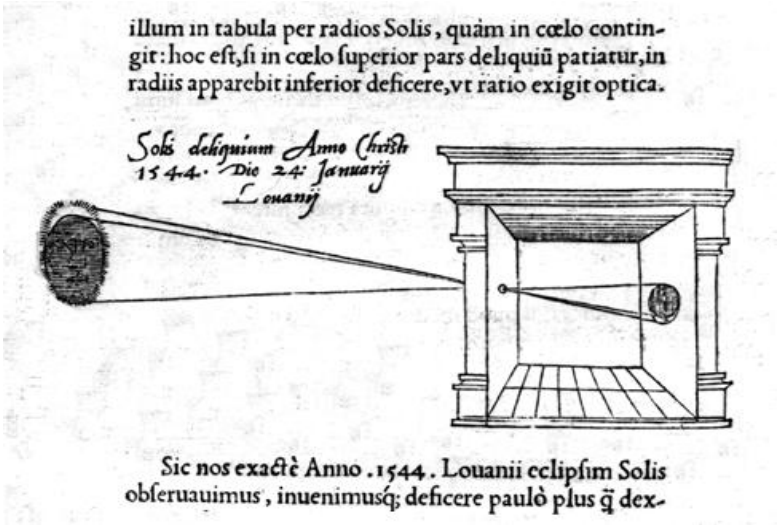


Illustration de la « Camera observa » - Reinerus Gemma-Frisius – 1545

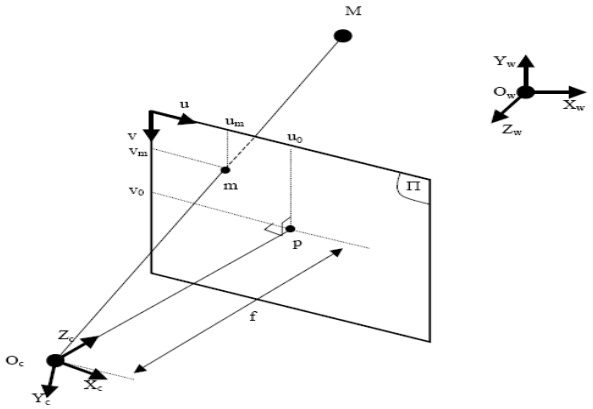
Les peintres du XVIième siècle connaissaient bien ce phénomène et comprirent très vite le bénéfice qu'ils pouvaient en retirer. Ils construisirent d'immenses chambres noires, parfois transportable, qui leur permettaient, comme avec un calque, de reproduire des paysages extérieurs. Le trou, appelé sténopé, ne devait pas être trop grand pour donner une projection nette, mais pas trop petit non plus pour laisser passer suffisamment de lumière. Il fut employé par de nombreux artistes, dont Giovanni Baptista della Porta, Vermeer, Guardi et Antonio Canal, dit Canaletto, qui l'utilisa notamment pour mettre en perspective ses célèbres paysages des canaux de Venise. Le dispositif fut amélioré en utilisant une lentille convergente qui donnait le même résultat avec une luminosité et une netteté meilleure. Le système se perfectionna en plaçant des boîtes coulissantes les unes dans les autres et par l'utilisation d'un miroir incliné à 45° pour redresser l'image. On essaya à maintes reprises de fixer l'image sur un support. En 1826, Nicéphore Niepce réussi à obtenir, après une exposition de 8 heures, une image stable sur une plaque d'étain enduite de bitume de Judée. La photographie venait de naître.

1.1.3.2 Définitions

Le modèle sténopé (pinhole) est le modèle projectif le plus souvent utilisé. Il permet un certain nombre de simplifications. Ce modèle est défini par deux éléments : un point, appelé centre optique et un plan ne contenant pas le point, appelé plan rétinien.
Dans ce modèle, le repère de la scène et le repère de la caméra sont composés d'axes orthogonaux. Le repère de la scène W est défini par son origine Ow et par ses trois axes (Xw, Yw, Zw). Le repère

de la caméra C est défini par son origine Oc et par ses trois axes (Xc, Yc, Zc). Le point origine Oc correspond également au centre optique de la caméra. L'axe optique est normal au plan rétinien appelé aussi plan image Π et coupe le plan au point p. A des fins de simplification, nous considèrons que l'axe optique est confondu avec l'axe Zc. Le plan Π est défini par deux vecteurs orthogonaux u et v. Dans ce repère, le point p a pour coordonnées (u_0, v_0). La distance O_cP correspond à la distance focale f.

La projection d'un point M de la scène de coordonnées (x_w, y_w, z_w) sur le plan image correspond au point m de coordonnée (u_m, v_m).

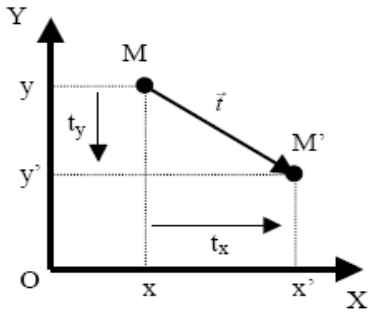


Dans cette représentation simplifiée du modèle sténopé, les paramètres extrinsèques sont les paramètres permettant le changement de repère de la scène vers le repère camera. Les paramètres intrinsèques sont la distance focale f et la position du point p dans le repère image. Il est à noter que le point p ne correspond pas forcement au centre de l'image. Dans la pratique, les constructeurs de caméras s'efforcent de faire correspondre ces deux points, mais des imperfections dans la réalisation de l'optique et le jeu mécanique nécessaire au montage engendrent un léger décalage dont il faudra tenir compte en fonction de la précision visée.

1.1.3.3 Paramètres extrinsèques

Le point M est défini par ses coordonnées (xw, yw, zw) dans le repère de la scène, mais il peut également être défini par ses coordonnées (xc, yc, zc) dans le repère de la caméra. Le plan image étant orthogonal à l'axe optique qui est lui-même confondu à l'axe Zc, un changement de repère est inévitable. Les paramètres extrinsèques sont donc les paramètres qui permettent le calcul du changement de repère entre le repère de la scène et le repère de la caméra.

1.1.3.3.1 Transformation rigide 2D

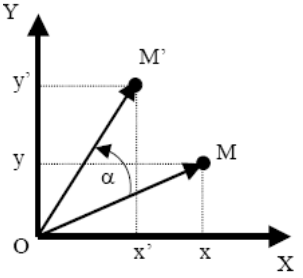


Dans le cas simple d'une transformation 2D, la translation de vecteur $\vec{t}$ d'un point M, s'écrit $\vec{OM'} = \vec{OM} + \vec{t}$,

soit en notation matricielle : $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$

En notation matricielle, la rotation α d'un point M, s'écrit :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix}$$



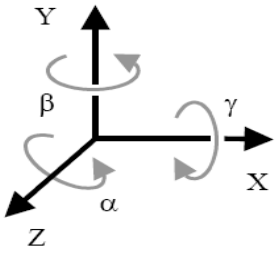
La transformation rigide 2D, composée d'une translation et d'une rotation, s'écrit donc :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

1.1.3.3.2 Transformation rigide 3D

Dans le cas 3D qui nous intéresse, la transformation rigide se compose de trois translations et d'une décomposition des rotations autour des trois axes.

- L'angle α autour de l'axe Z (de X vers Y) correspond au roulis (roll)
- L'angle β autour de l'axe Y (de Z vers X) correspond au lacet (pan)
- L'angle γ autour de l'axe X (de Y vers Z) correspond au tangage (tilt)



En notation matricielle, la décomposition des rotations s'écrit :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Le développement de l'expression ci-dessus donne le résultat suivant :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) & (\sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma) \\ \sin\alpha\cos\beta & (\cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma) & (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Cette matrice de transformation est notée R.

Nous pouvons remarquer que cette matrice R est une matrice orthogonale puisque quels que soient les angles α, β et γ, le produit :

$$R^T R = Id$$

où Id est la matrice identité.
Intuitivement, on peut formuler que quelle que soit la rotation que l'on fait subir au trièdre formant le repère, l'orthogonalité des axes formant ce repère est conservée après la transformation. Cette propriété d'orthogonalité de la matrice R, nous permet également d'écrire que la matrice inverse est égale à sa transposée :

$$R^{-1} = R^t$$

La transformation rigide 3D s'écrit donc :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) & (\sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma) \\ \sin\alpha\cos\beta & (\cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma) & (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_z \\ t_z \end{bmatrix}$$

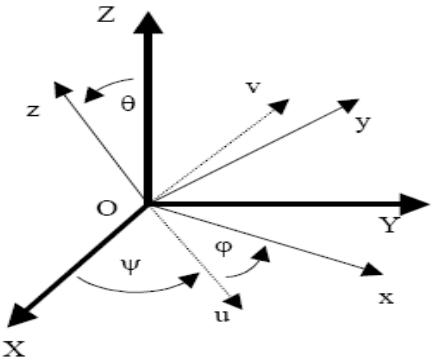
ou de manière simplifiée, en introduisant les coordonnées homogènes :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

En utilisant la notation $ri = (ri1 \;\; ri2 \;\; ri3)$, cette transformation s'écrit :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_1 & t_x \\ r_2 & t_y \\ r_3 & t_z \\ 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Les trois angles α, β et γ sont dit du cadran. Il ne faut pas les confondre avec les angles d'Euler. Les angles d'Euler permettent également la transformation d'un référentiel OXYZ en référentiel Oxyz. Cependant, les trois angles ψ, θ et ϕ portent des noms liés à leur application en astronomie.



- ψ est l'angle de précession
- θ est l'angle de nutation
- ϕ est l'angle de la rotation propre

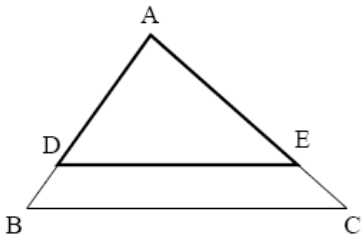
Cette décomposition fait intervenir un référentiel intermédiaire Ouvz.
Les paramètres extrinsèques sont donc au nombre de 6 : trois angles de rotation α, β, γ et trois translations t_x, t_y, t_z.

1.1.3.4 Paramètres intrinsèques

Une fois le changement de repère effectué, nous avons besoin de calculer la projection des points du repère caméra sur le plan image. Dans le cas du modèle simplifié, on rappelle que l'axe optique est confondu avec l'axe Z_c et que le plan image est orthogonal à cet axe, c'est-à-dire parallèle au plan X_cY_c. Un calcul supplémentaire permet de convertir les unités métriques du repère de la caméra en unités pixels de l'image. Les paramètres intrinsèques de la camera sont donc les paramètres qui permettent ces transformations.

1.1.3.4.1 Projection perspective

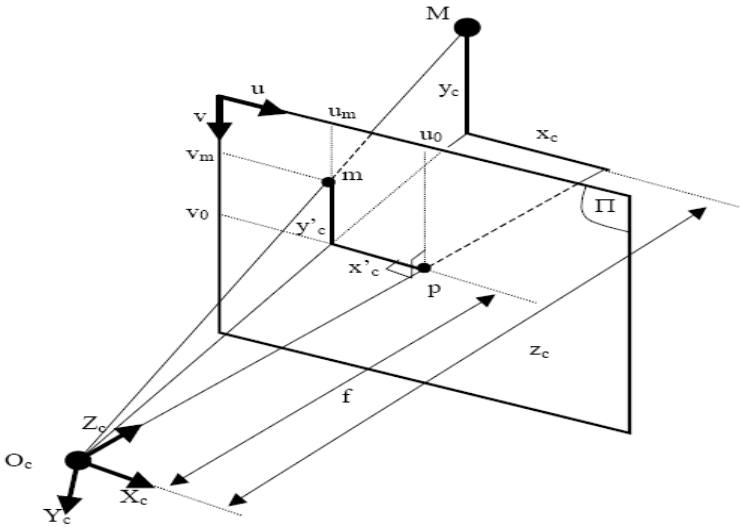
D'après le théorème de Thales, si dans un triangle ABC, D est un point de [AB], E est un point de [AC] et si les segments [BC] et [DE] sont parallèle alors on a la relation :



$$\frac{AD}{AB} = \frac{AE}{AC} = \frac{DE}{BC}$$

Donc d'après Thales :

$$\begin{cases} \frac{x_c}{f} = \frac{x_c}{z_c} \\ \frac{y_c}{f} = \frac{y_c}{y_c} \end{cases} \rightarrow \begin{cases} x_c = f\frac{x_c}{z_c} \\ y_c = f\frac{y_c}{z_c} \end{cases}$$

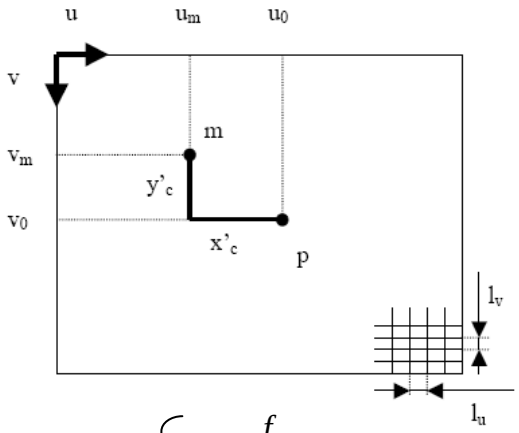


1.1.3.4.2 Changement d'unité

Le changement d'unité permet la transformation de l'unité métrique du repère caméra à l'unité pixelique du repère (u,v) de l'image. Contrairement au repère caméra, le repère de l'image n'est pas homogène du fait de la nature de même de la matrice CCD.

Cette transformation s'écrit simplement :

$$\begin{cases} u_m = u_0 + \frac{x'_c}{l_u} \\ v_m = v_0 + \frac{y'_c}{l_v} \end{cases}$$



où lu est la largeur d'un pixel, l_v la hauteur d'un pixel, u_0 et v_0 les coordonnées du point p, centre optique de l'image.
En regroupant les deux expressions, on obtient :

$$\begin{cases} u_m = u_0 + \frac{f.x_c}{l_u z_c} \\ v_m = v_0 + \frac{f.y_c}{l_v z_c} \end{cases} \rightarrow \begin{cases} u_m = u_0 + k_u \frac{x'_c}{z_c} \\ v_m = v_0 + k_v \frac{y'_c}{z_c} \end{cases} \quad \text{avec} \quad \begin{cases} k_u = \frac{f.}{l_u} \\ k_v = \frac{f}{l_v} \end{cases}$$

Les paramètres intrinsèques sont donc au nombre de 4 :
• k_u, la distance focale en pixels horizontaux,

• k_v, la distance focale en pixels verticaux,
• u_0, v_0, les coordonnées du centre optique de l'image.

De façon à rendre homogènes les matrices des paramètres intrinsèques et extrinsèques, les expressions des paramètres intrinsèques peuvent se mettre sous la forme :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad \text{avec} \quad \begin{cases} u = \dfrac{U}{W} \\ v = \dfrac{V}{W} \end{cases}$$

Soit, en notation simplifiée :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = [A] \bullet \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

où A est la matrice des paramètres intrinsèques.

1.1.3.4.3 Formulation complète

En regroupant les expressions des paramètres intrinsèques et extrinsèques, la formulation complète s'écrit :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

L'expression de la projection perspective s'écrit généralement sous la forme :

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

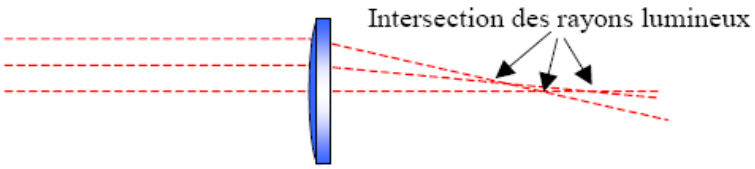
Avec :

$$
\begin{cases}
m_{11} = k_u.\cos\alpha.\cos\beta - u_0.\sin\beta \\
m_{12} = k_u(\cos\alpha.\sin\beta.\sin\gamma - \sin\alpha.\cos\gamma) + u_0.\cos\beta.\sin\gamma \\
m_{13} = k_u(\sin\alpha.\sin\gamma + \cos\alpha.\sin\beta.\cos\gamma) + u_0.\cos\beta.\cos\gamma \\
m_{14} = k_u.t_x + u_0.t_z \\
m_{21} = k_v.\sin\alpha.\cos\beta - v_0.\sin\beta \\
m_{22} = k_v(\cos\alpha.\cos\gamma + \sin\alpha.\sin\beta.\sin\gamma) + v_0.\cos\beta.\sin\gamma \\
m_{23} = k_v(\sin\alpha.\sin\beta.\cos\gamma - \cos\alpha.\sin\gamma) + v_0.\cos\beta.\cos\gamma \\
m_{24} = k_v.t_y + v_0.t_z \\
m_{31} = -\sin\beta \\
m_{32} = \cos\beta.\sin\gamma \\
m_{33} = \cos\beta.\cos\gamma \\
m_{34} = t_z
\end{cases}
$$

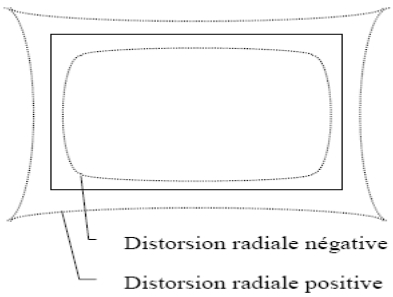
Dans cette expression, on remarque que les coefficients m_{31}, m_{32}, m_{33} et m_{34} ne dépendent pas des paramètres intrinsèques de la caméra.

1.1.3.5 Distorsion de l'image

Dans la plupart des applications, seule la distorsion radiale pose réellement un problème.
Cette distorsion de l'image se manifeste généralement par la courbure vers l'extérieur de l'image des droites verticales et horizontales. L'effet de la distorsion est d'autant plus visible que l'on s'éloigne du centre de l'image. L'exemple de distorsion radiale que l'on peut observer facilement est celui du juda des portes d'habitation. Lorsque nous regardons à travers, toutes les lignes droites apparaissent courbes. Ces aberrations sont essentiellement dues à la géométrie de la lentille. Les rayons de lumière qui passent par les bords de la lentille sphériques classiques convergent en un point légèrement décalé par rapport au rayon qui passe par le centre. Ce phénomène bien connu est également appelé « aberration sphérique ».

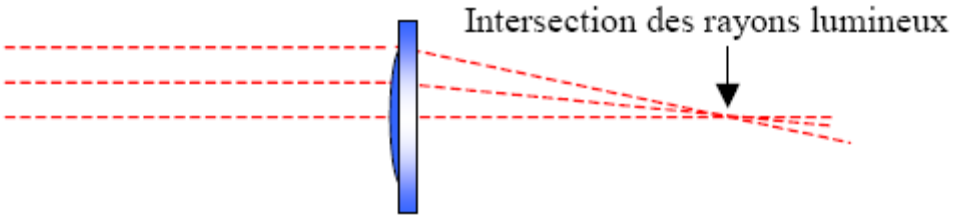


Distorsion radiale avec les lentilles sphériques classiques



Effet de la distorsion radiale sur l'image

Ce type d'aberration peut être corrigé analytiquement. Nous verrons par la suite un modèle mathématique permettant la correction. Cependant, cette correction se fait avec une perte d'information. Une autre solution consiste à diminuer l'angle d'ouverture des lentilles de façon à retomber dans le cas de la dioptrique de Gauss. L'inconvénient est que cette solution diminue la luminosité des images. La plupart des constructeurs proposent aujourd'hui des lentilles asphériques. Une surface non sphérique permet de faire converger les rayons lumineux du bord de la lentille et du centre vers un foyer unique.



Correction de la distorsion avec une lentille asphérique

La distorsion radiale peut être modélisée par un polynôme de degré n. Dans la pratique, on utilise un polynôme de degré 4 en ne gardant que les puissances paires. Cette simplification est réputée pour être suffisante. La formulation de la distorsion radiale est donc la suivante :

$$\hat{u} = u + (u - u_0).(k_1.r^2 + k_2.r^4)$$

$$\hat{v} = v + (v - v_0).(k_1.r^2 + k_2.r^4)$$

Où $r = \sqrt{(u-u_0)^2 + (v-v_0)^2}$

1.2 Recalage d'images appliqué aux caméras PTZ

1.2.1 Méthodes denses

1.2.1.1 Méthode de Szeliski

Une méthode de recalage classique utilisée pour la construction d'une mosaïque d'images est proposée pour la première fois par Szeliski dans [SZE94]. Cette méthode permet de calculer la matrice d'homographie H d'une image I_1 dans I_0 par itérations successives en recherchant la minimisation d'une fonction de coût. Lorsque certaines conditions sont respectées, la transformation homographique H qui relie une image I_1 à une image I_0 s'exprime avec 8 coefficients. On rappelle que la transformation est définie à un facteur près.

$$x' \sim \mathrm{H}.x = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \bullet \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

où x est un point de l'image I_1 défini par ces coordonnées homogènes (u, v, 1) dans l'image et x' sa projection homographique dans l'image I_0 et défini par ces coordonnées homogènes (u', v', 1). Le signe ~ indique la relation d'équivalence, à un facteur d'échelle près, entre le point x' et la transformation du point x.

Les coordonnées (u', v') du point x' dans l'image I_1 se déduisent simplement :

$$u' = \frac{m_0.u + m_1.v + m_2}{m_6.u + m_7.v + 1},\ v' = \frac{m_3.u + m_4.v + m_5}{m_6.u + m_7.v + 1},$$

Dans l'approche de Szeliski, la matrice H va être mise à jour itérativement à partir d'une matrice de correction D en utilisant l'équation suivante :

$$H \leftarrow (Id + D)H$$

où la matrice de correction D est définie comme suit :

$$D = \begin{bmatrix} d_0 & d_1 & d_2 \\ d_3 & d_4 & d_5 \\ d_6 & d_7 & 0 \end{bmatrix}$$

A chaque itération, Szeliski calcule une nouvelle transformation de l'image I_1 en fonction des paramètres précédemment calculés en utilisant la relation :

$$x' \sim (Id + D).H.x$$

où x est une pixel de I_1 original et x' le pixel de l'image transformée. Nous pouvons également calculer l'image transformée directement à partir de l'image précédente, c'est-à-dire calculer :

$$\tilde{I}_1(x_i) = I_1(x'_i)$$

à partir de :

$$x'' \sim (Id + D)x$$

en développant l'équation, cela revient à calculer :

$$u'' = \frac{(1 + d_0)u + d_1.v + d_2}{d_6.u + d_7.v + 1},\ v'' = \frac{d_3.u + (1 + d_4).v + d_5}{d_6.u + d_7.v + 1},$$

Le calcul des paramètres de la matrice de correction s'effectue en recherchant la minimisation d'une fonction de coût. Szeliski propose d'utiliser la fonction de coût suivante :

$$E(d) = \sum_i (\tilde{I}_1(x''_i) - I_1(x_i))^2$$

Cette fonction de coût est l'expression de la différence au carré de chaque pixel commun entre l'image I0 et l'image transformée de I_1 (SSD). Un développement de Taylor de cette expression donne :

$$E(d) = \sum_i \left[(\tilde{I}_1(x_i) - I_0(x_i) + \nabla(\tilde{I}_1(x_i) \frac{\partial x''_i}{\partial d} d \right]^2$$

On note :

- e_i, l'erreur d'intensité à l'ordre 0, soit $e_i = (\tilde{I}_1(x_i) - I_0(x_i)$
- d, le vecteur des coefficients de la matrice D mis en ligne, d = (d0, d1, d3, d4, d5, d6, d7) ^{T}
- $g_i^T = \nabla \tilde{I}_1(x_i)$, L'image du gradient de l'image transformée de I_1 à x_i,
- $J_i = \frac{\partial x''_i}{d} = \begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -u^2 & -uv \\ 0 & 0 & 0 & u & v & 1 & -uv & -v^2 \end{bmatrix}^T$, le jacobien de x_i" par rapport à d

L'expression de la fonction de coût peut donc se réécrire sous la forme :

$$E(d) = \sum_i \left[g_i^T . J_i^T . d + e_i \right]^2$$

La solution de ce système peut être déterminer à partir de la méthode des moindres carrés. L'expression est alors de la forme A.d = b avec :

$$A = \sum_i J_i g_i g_i^T J_i^T , et \qquad B = \sum_i e_i J_i g_i$$

L'algorithme est finalement assez simple. La matrice H de départ est initialisée soit avec des paramètres approchés de l'homographie par une autre méthode ou par des informations issues de la caméra soit avec la matrice identité. Nous calculons ensuite l'image du gradient, la matrice jacobienne et la matrice de l'erreur d'intensité à partir de l'image I_1 transformé par la matrice H de départ. Si la matrice H de départ est initialisée avec la matrice identité, il n'est bien évidement pas nécessaire de calculer l'image transformée. En utilisation la méthode des moindres carré, ces matrices permettent de déterminer les valeurs de la matrice de correction D que l'on peut appliquer à la matrice H. Par itérations successives, on affine les valeurs de la matrice H. L'algorithme s'arrête sur un critère d'arrêt qui peut être un nombre maximum d'itérations atteint, un calcul sur la somme ou la somme au carré des valeurs de la matrice d'erreur d'intensité ou encore sur la somme ou la somme au carré des paramètres de la matrice D.

1.2.1.2 Méthode du simplexe

Cette méthode est un grand classique la recherche d'une minimisation dans un espace à plusieurs dimensions. Elle a été introduite par Nelder et Mead en 1965 [NEL65]. Cette méthode repose sur l'utilisation d'un polyèdre de dimension n+1 que nous allons modifier à chaque itération pour se rapprocher du minimum d'une fonction de coût. Si les paramètres de prise de vue de l'une des deux images sont connus, l'espace de recherche n'a plus que 3 dimensions. En conséquence, le polyèdre du simplexe possède 4 sommets, c'est à dire un tétraèdre.

L'algorithme du simplexe présenté par Nelder et Mead fait intervenir 4 coefficients qui sont : le coefficient de réflexion ϱ, d'expansion χ, de contraction γ et de rétrécissement σ. Le domaine de validité de ces coefficients est donné par :

$$\varrho > 0, \chi > 1 , 0 < \gamma < 1 \text{ et } 0 < \sigma < 1$$

Traditionnellement, on donne implicitement à ces paramètres les valeurs suivantes :

$$\varrho = 1, \chi = 2 , \gamma = ½ \text{ et } \sigma = ½$$

Nous cherchons à recaler le mieux possible deux images. Il nous faut donc une mesure de coût exprimant la différence entre ces deux images. On peut faire le choix d'utiliser la mesure de corrélation croisée normalisée entre l'image I et l'image transformée de J. Ce que nous pouvons écrire :

$$C(I, J, T) = \frac{\sum_{x \in \Omega} I(x).J(T(x))}{\sqrt{\sum_{x \in \Omega} I(x)^2} . \sqrt{\sum_{x \in \Omega} J(T(x))^2}}$$

Dans la mesure où tous les pixels sont utilisés pour la fonction de coût, la méthode est bien dans la catégorie des méthodes denses.

La première étape de la méthode du simplexe consiste à initialiser les sommets du polyèdre.

Pour chaque sommet k, nous fixons arbitrairement les valeurs θ, ϕ, f de la transformation homographique, nous calculons les images transformées de J et nous calculons la fonction de coût C_{Sk}. Dans la pratique, les valeurs de θ, ϕ, f ne sont pas fixées totalement de façon arbitraire puisque nous connaissons une position approchée de la caméra. Nous choisissons donc des points au voisinage de celui donné par la caméra.

Ces points sont alors classés dans l'ordre croissant de leur fonction de coût.

$$C_{s0} \leq Cs_1 \leq \ldots \leq Cs_n$$

Partant du principe que le sommet dont la fonction de coût est la plus faible est plus proche du minimum que le point dont la fonction de coût est la plus forte, le principe de l'algorithme est de s'éloigner de ce dernier point. Pour cela les auteurs de la méthode proposent d'essayer le point p_0 dans la direction symétrique à Sn par rapport au centre c du polyèdre formé par les autres points. Ce point p_0 est donné par :

$$p_0 = c + \rho(c - S_n)$$

En fonction de la valeur de la fonction de coût C_{p0} de ce point, deux cas peuvent se présenter :

• $C_{p0} \leq C_{s0}$, c'est-à-dire que le point p0 est plus proche du minimum que le meilleur sommet du simplexe, on essaye donc d'aller plus loin dans cette direction en calculant la fonction de coût C_{p1} du point p_1. Ce point est donné par :

$$p_1 = c + \chi(c - S_n)$$

A nouveau, deux cas peuvent alors se présenter :

- $C_{p1} \leq C_{p0}$, on remplace le sommet S_n par le point p_1. On applique donc une expansion au simplexe
- $C_{p1} > C_{p0}$, on remplace le sommet S_n par le point p_0. On applique donc une réflexion au simplexe

• $C_{p0} > C_{s0}$, on essaie les points p_2 et p_3, placés de part et d'autre de c, dont les coordonnées sont données par :

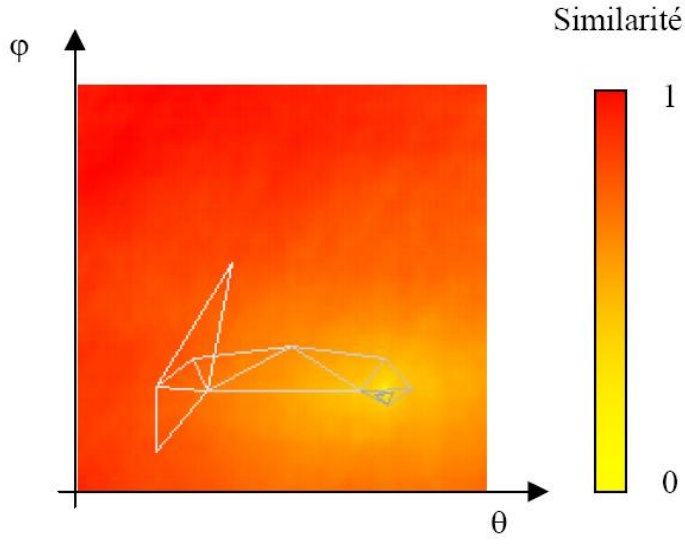
$$p_{2,3} = c \pm \gamma(c - S_n)$$

A nouveau, deux cas peuvent se présenter :

- si $C_{p2} \leq C_{sn-1}$ ou $C_{p3} \leq C_{sn-1}$, alors on remplace S_n par p_2 ou p_3. On applique alors une contraction au simplexe
- sinon, le point S_0 étant le point le plus proche du minimum, on rétrécit le simplexe autour de ce point. Les coordonnées de tous les autres points du simplexe sont données par :

$$S_k \rightarrow c - \sigma(c - S_k) \text{ pour } \quad k = 1 \text{ à } n$$

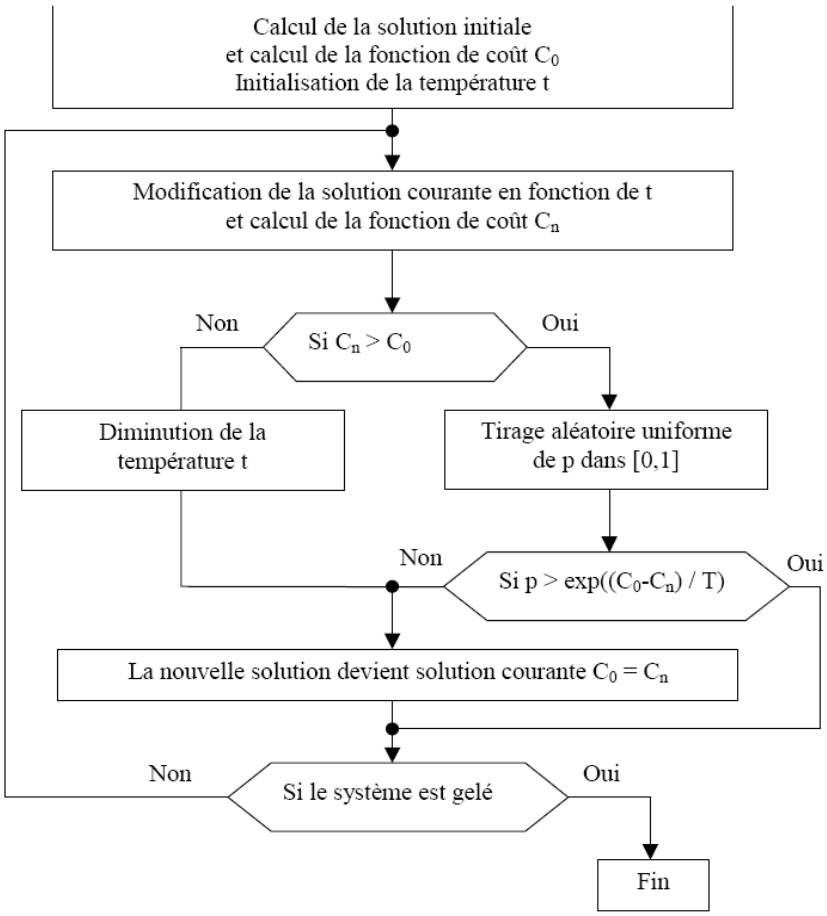
Une fois le (ou les) point(s) du sommet recalculé(s), on réordonne les points en fonction de la valeur du coût et on reprend l'algorithme. On arrête les itérations lorsque la taille du simplexe atteint un certain minimum fixé ou que la fonction de coût du sommet le plus faible atteint elle aussi un minimum ou encore lorsqu'un certain nombre d'itérations ont été effectuées.



Exemple de progression du simplexe

1.2.1.3 Recuit simulé

Le recuit simulé est une méthode de recherche stochastique qui s'inspire d'un procédé utilisé en métallurgie. Afin d'obtenir un arrangement régulier des atomes à l'état solide, on part de l'état liquide et on refroidit le métal très lentement. Si l'arrangement des atomes n'est pas satisfaisant, on réchauffe le métal sans aller jusqu'au point de fusion, mais suffisamment pour laisser une certaine liberté de mouvement aux atomes. Ce réchauffement est appelé recuit.
L'algorithme du recuit simulé s'inspire de cette approche. Le principe consiste à explorer aléatoirement l'espace de recherche autour d'une position courante en fonction d'une « température » t. Plus la température est élevée, plus on s'autorise à explorer loin de la solution courante. Plus la température est basse, plus l'espace de recherche est restreint. A partir d'une température élevée, on diminue progressivement la température à chaque fois que la fonction de coût de la nouvelle solution est meilleure que la solution courante.
Cependant, on s'autorise aléatoirement à accepter comme solution courante, une solution dont la fonction de coût est supérieure à la solution courante. Le but est de permettre à l'algorithme de franchir un minimum local. Cet algorithme ne garanti pas d'atteindre le minimum local, mais dans la mesure où la température est élevée, il peut permettre de s'en approcher.
L'organigramme de cet algorithme peut se résumer de la façon suivante :



Organigramme de l'algorithme du recuit simulé.

Comme pour la méthode du simplexe, la transformation T est calculée à partir des trois paramètres θ, ϕ, f dont nous cherchons l'estimation. Pour générer une transformation à l'itération n, nous modifions les paramètres à partir de l'expression suivante :

$$\theta n = \theta_{n-1} + a_{\theta n} . E_{\theta} . tn$$

où θ_{n-1} est la valeur de l'angle de panorama à l'itération précédente,
a_{0n} est une valeur aléatoire comprise entre –1 et 1,
E_θ est fonction de la dynamique de l'espace de recherche,
t_n est la température à l'itération n comprise entre 0 et 1.

Un calcul équivalent est réalisé pour f et ϕ. L'algorithme se termine lorsque la solution est gelée, c'est à dire lorsque la température atteint un minimum. La condition d'arrêt peut être également liée directement à la mesure de similarité.

.

1.2.1.4 Algorithme génétique

Les algorithmes génétiques sont des algorithmes d'optimisation stochastiques. A l'instar de l'algorithme du recuit simulé, les algorithmes génétiques s'inspirent d'un phénomène naturel.
Ici, ce sont les principes d'évolution et de sélection naturelle proposées par Darwin que l'on va chercher à reproduire. Dans la nature, les individus se croisent et se reproduisent en interagissant les uns avec les autres tout en s'adaptant à l'environnement. Les premiers travaux sur les algorithmes génétiques datent des années 1960-1970 avec John Holland. C'est en 1989 que D.E Golberg popularise ces algorithmes [GOL89] en décrivant leur utilisation dans le cadre de la résolution de problèmes concrets. Le champ d'application des algorithmes génétiques est très vaste. On les retrouve principalement dans le cas d'une recherche de minimisation pour laquelle les méthodes basées sur la descente de gradient sont inefficaces.
Mais ils sont également utilisés dans le cadre de l'apprentissage supervisé [MIT96], l'estimation du mouvement [ZAI01], la segmentation d'image [BOS01], la compression ou encore des domaines plus éloignés comme la programmation automatique, la théorie des graphes, l'économie ou la finance. Pour Lerman et Ngouenet [LER95], Les algorithmes génétiques diffèrent des algorithmes classiques d'optimisation et de recherche essentiellement en quatre points fondamentaux :

- Les algorithmes génétiques utilisent un codage des éléments de l'espace de recherche et non pas les éléments eux-mêmes.
- Les algorithmes génétiques recherchent une solution à partir d'une population de points et non pas à partir d'un seul point.
- Les algorithmes génétiques n'imposent aucune régularité sur la fonction étudiée (continuité, dérivabilité, convexité…). C'est un des gros atouts des algorithmes génétiques.
- Les algorithmes génétiques ne sont pas déterministes, ils utilisent des règles de transition probabilistes.

Le succès de ces algorithmes est justifié par leur simplicité de mise en œuvre et par leur performance. Ils ne fournissent pas nécessairement la solution globale mais permettent de généralement de s'approcher suffisamment prés de la solution optimale.
Les algorithmes génétiques cherchent donc à reproduire le processus de sélection naturelle à partir de deux constats. Le premier est que seul les individus les mieux adaptés perdure au cours du temps. Le deuxième est que le renouvellement des populations est assuré par ces éléments les plus adaptés par un processus de croisement et de mutation. Ils reposent donc sur 3 grands principes qui sont la sélection, le croisement et la mutation. L'implémentation de ces algorithmes nécessite de définir également un critère de performance (appelé fitness) ainsi que le codage du génotype et le nombre d'individu représentant la population. Le génotype caractérise chaque individu par l'intermédiaire des gènes. Un individu représente donc un point dans l'espace de recherche. Le critère de performance est utilisé pour mesurer l'adaptabilité de l'individu face à l'environnement dans lequel évolue le logiciel. Suivant le type de situation, un même codage du génotype ne donnera pas forcément les mêmes résultats. C'est en fonction de la valeur de ce critère de performance que l'on considère que l'individu est viable et qu'il peut ce reproduire.

Une fois que le critère de performance et que le codage du génotype sont définis, l'implémentation de l'algorithme est assez simple et suit les étapes suivantes :

- 1°) Initialisation de la population. Les gènes de chaque individu sont sélectionnés aléatoirement dans tout l'espace de recherche ou éventuellement dans une zone restreinte de l'espace de recherche en fonction des données dont on dispose. Le but est d'assurer une couverture la plus complète possible. A chaque individu est associé sa mesure du critère de performance.
- 2°) Sélection. A partir de la mesure du critère de performance, on sélectionne les individus qui maximisent la valeur du critère de façon à générer une population intermédiaire. Cependant, le processus de sélection ne doit pas « tomber » dans l'élitisme. Comme dans la nature, il arrive que des individus décrétés « mauvais » arrivent à survivre. Le processus de sélection doit donc introduire une part d'aléatoire. La solution régulièrement utilisée est la sélection par roulette de casino (wheel selection). Le principe consiste à associer à chaque individu un segment proportionnel à la valeur de son critère de performance. Le segment total est ensuite normalisé. Les individus sont alors sélectionnés par un tirage aléatoire de distribution uniforme compris entre 0 et 1. En règle générale, l'individu qui maximise le critère de performance est systématiquement sélectionné. C'est cette population intermédiaire qui va être utilisée dans les étapes suivantes.
- 3°) Croisement. Le croisement consiste à mélanger les génotypes de deux individus parents de façon à générer un individu enfant. Les parents sont sélectionnés aléatoirement parmi la population intermédiaire issue du processus de sélection. Dans le cas où les gènes sont constitués de valeur réelle, la combinaison des gènes parents est réalisée par interpolation linéaire à partir d'un poids fixé aléatoirement.
- 4°) Mutation. La mutation est appliquée aléatoirement sur certain enfant issu de l'étape croisement. Elle consiste à sélectionner un ou plusieurs gènes par un processus aléatoire et à leur appliquer un bruit gaussien dans la limite de l'espace de recherche. Ce processus de mutation permet d'assurer la propriété d'ergodicité de parcours de l'espace. Le processus ne garantit pas que tout l'espace de recherche sera exploré, mais il offre la possibilité à l'algorithme d'atteindre n'importe quel point de l'espace. Dans certaine configuration, il peut être intéressant d'appliquer une décroissance au bruit gaussien appliqué aux gènes et ainsi limiter l'exploration au cours des itérations successives.
- 5°) A partir de la nouvelle population issue de la sélection, du croisement et de la mutation, on réitère ces trois étapes jusqu'à ce qu'un critère d'arrêt soit satisfait.

Dans notre cas particulier, le génotype de chaque individu est composé de trois gènes. Ces trois gènes correspondent aux trois inconnues (f_J, ϕ_J et $\Delta\theta_{I,J}$) de la transformation homographique entre deux images. Nous pourrions très bien généraliser l'algorithme aux 6 paramètres de la transformation. Pour le critère de performance, nous utilisons la mesure de similarité et pour le critère d'arrêt nous fixons un seuil sur la mesure de similarité du meilleur individu de chaque génération. Comme évoqué à l'étape 2, nous préservons systématiquement le meilleur individu à chaque génération. Nous assurons ainsi une décroissance monotone de l'algorithme.

1.2.2 Méthodes éparses

1.2.2.1 Détecteur de Harris

La détection de points d'intérêts n'est pas un problème récent et plusieurs solutions existent déjà. Dans [SCH00], le lecteur trouvera la comparaison de plusieurs détecteurs réalisée par Schimd et Mohr. Cependant, l'algorithme le plus cité dans la littérature est celui de Harris [HAR88] qui découle

des travaux de Moravec [MOR77]. Ce détecteur s'attache à localiser des points où les variations différentielles sont importantes dans plus d'une direction de façon à ne pas inclure les contours.
L'algorithme de Harris s'appuie sur un filtrage gaussien de l'image et sur le calcul de la matrice M défini comme suit :

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}.\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}.\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

A partir de cette matrice M, Harris propose de calculer, en chaque point, le coefficient R donné par :

$$R = Det(M) - kTrace(M)^2 = (AB - C^2) - k(A+B)^2$$

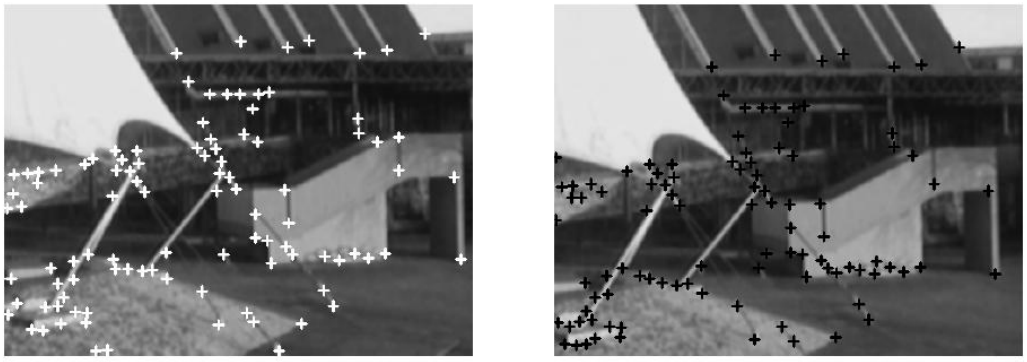
L'interprétation des valeurs des R est la suivante :

- R >> 0 → point d'intérêt
- R << 0 → contour
- R faible → région plate

Les points d'intérêts correspondent aux maxima locaux positifs de ce coefficient.

Cet algorithme fait intervenir le paramètre k. Des études théoriques ont montré que la valeur optimale de ce paramètre est 0.04. Le résultat de la mesure de Harris est cependant très sensible à une faible variation de ce paramètre. Ce détecteur, bien qu'étant le plus utilisé, n'est pas toujours facile à mettre en œuvre, puisque cinq paramètres sont utilisés : le filtre gaussien, le filtre dérivatif, le paramètre k, le voisinage des maxima locaux et le seuil final.
Une adaptation de ce détecteur est proposée par Zheng [ZHE99].

A l'issu de cette première étape, nous disposons de deux ensembles de points.



Détection des points d'intérêts dans deux images avec le détecteur de Harris

1.2.2.2 Le descripteur SIFT

Le détecteur SIFT (*Scale Invariant Feature Transform*) proposé par Lowe [LOW04] est considéré aujourd'hui comme l'un des plus performants. L'intérêt de ce détecteur est qu'il est invariant aux transformations affines (changement d'échelle, rotation et translation) et qu'il permet de calculer un descripteur robuste. Ceci le rend particulièrement efficace dans le cas de recalage rigide.
L'algorithme d'extraction de point d'intérêt et le calcul des descripteurs se déroulent en deux étapes :

• Détection de maximum ou minimum locaux dans l'espace d'échelle (scale – space) et localisation des points d'intérêts.
• Choix de l'orientation et calcul des descripteurs

La première étape consiste à détecter les points qui sont stables dans l'espace d'échelle. Pour cela Lowe calcul l'opérateur DoG (Difference of Gaussian) dans l'espace d'échelle puis extrait les extrema locaux. La représentation E dans l'espace d'échelle d'une image I est obtenue par convolution de l'image I avec une gaussienne G(x, y, σ).

$$E(x, y,\sigma)=G(x, y,\sigma)*I(x, y)$$

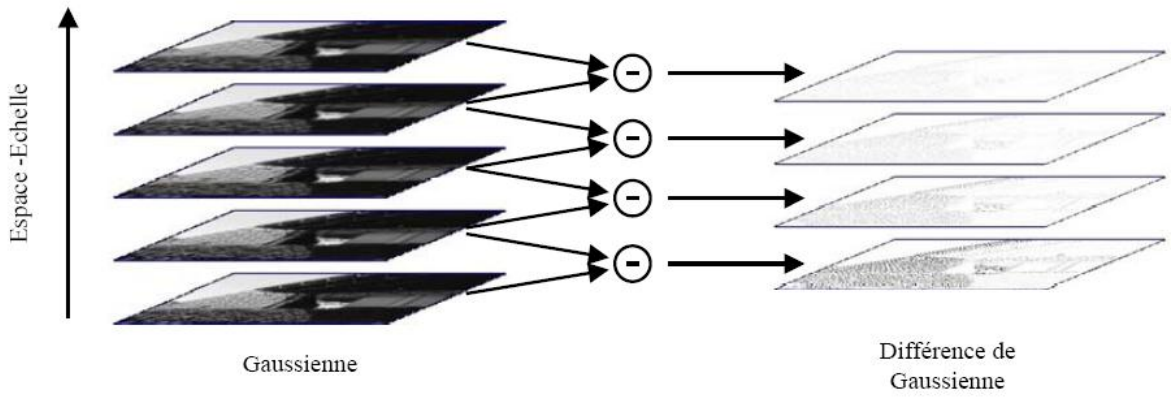
où * est l'opérateur de convolution en x, y et

$$G(x, y, \sigma) = \frac{1}{2.\pi.\sigma^2}.e^{-(x^2+y^2)/2.\sigma^2}$$

L'opérateur DoG s'écrit alors :

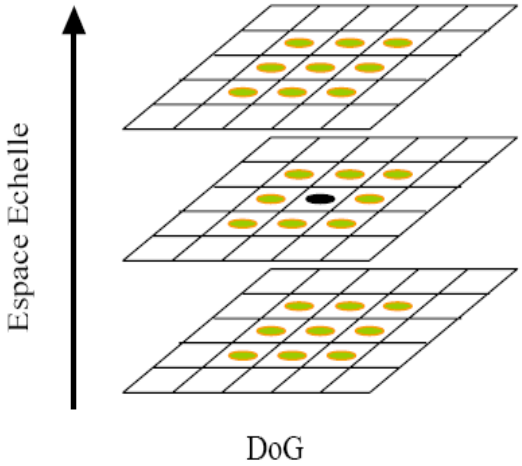
$$DoG(x, y,\sigma)=E(x, y,s\cdot\sigma) - E(x, y,\sigma)$$

où s est une constante multiplicative permettant de définir un nombre d'intervalle par octave (pour passer d'un octave à l'octave voisin la largeur et la hauteur de l'image sont multipliées 2).



Construction de l'opérateur DoG d'après l'article de Lowe [LOW04]

Lowe explique qu'il utilise cet opérateur parce qu'il existe des algorithmes efficaces pour réaliser l'opération de convolution et qu'ensuite il suffit de calculer une différence. Pour chaque octave de l'espace d'échelle, l'image initiale est convoluée par les masques gaussiens pour fournir les images qui seront ensuite soustraite les unes aux autres. La taille de l'image initiale est ensuite divisée par 4 (2x2) afin de calculer l'image initiale de l'octave suivant.
L'ensemble des images DoG ainsi calculé permet de constituer la base d'images utilisées pour la recherche des points d'intérêt. Chaque pixel d'une image DoG est comparée avec ces 8 voisins ainsi qu'avec les 9 pixels des images DoG immédiatement supérieure et inférieure dans l'espace d'échelle. Soit en tout 27 comparaisons. Si le pixel est à un minima ou un maxima local, il est un candidat possible. Une phase supplémentaire permet d'éliminer tous les candidats qui sont situé dans des zones insuffisamment contrastées.



Recherche des extrema dans l'espace d'échelle

L'étape suivante consiste à calculer un descripteur local. Le descripteur SIFT est particulièrement robuste aux petits changements de translation, d'illumination et de transformations affines. Le principe retenu par Lowe consiste à calculer un histogramme des orientations du gradient au voisinage du point d'intérêt. La région d'un point d'intérêt est décomposée en 4x4 parties et chacune des sous parties est elle-même décomposée en 8 secteurs angulaires. L'histogramme ainsi obtenu est pondéré par la norme du gradient en ce point. Un point est donc décrit par un vecteur de 4x4x8 soit 128 dimensions. La norme du gradient ainsi que son orientation sont calculées à partir de l'image convoluée par le masque gaussien de l'espace d'échelle le plus grand de façon à garantir l'invariance en échelle. Le calcul de la norme m(x, y) et de l'orientation θ(x, y) du gradient sont donnés par :

$$m(x,y) = \sqrt{(E(x+1,y) - E(x-1,y))^2 + (E(x,y+1) - E(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}\frac{E(x+1,y) - E(x-1,y)}{(E(x,y+1) - E(x,y-1)}$$

L'algorithme SIFT est très efficace cependant, le temps de calcul ainsi que l'espace mémoire nécessaire à son exécution sont assez important. Dans son article Lowe indique un temps de calcul de l'ordre de 300ms. L'implémentation que nous avons réalisé donne un temps de calcul de l'ordre la seconde. Cependant, cet algorithme reconnu comme efficace et précis par la communauté.

1.2.2.3 Le descripteur « Local jet »

Le descripteur « Local jet » est moins robuste que le descripteur SIFT. Il reste cependant le descripteur le plus utilisé, d'une part parce qu'il a été développé plus tôt et d'autre part parce qu'il est plus rapide à calculer et plus facile à mettre en œuvre. Le principe de ce descripteur est décrit par Koenderink et van Doorn dans [KOE87]. Il réside dans le fait qu'une fonction peut-être décrite localement par ses dérivées en une série de Taylor. Un point dans une image peut donc être décrit par ses dérivées au voisinage. Le descripteur « Local jet » $J_n(x, y, \sigma)$ est alors défini jusqu'à l'ordre n, pour le point (x, y) avec la taille de gaussienne σ, de la façon suivante :

$$j_n(x,y,\sigma) = \left\{L_{i_1,\ldots,i_k}(x,y,\sigma) / k = 0,\ldots.n\right\}$$

où $L_{i1\ldots ik}(x, y, \sigma)$ désigne la dérivée $k^{ième}$ de l'image I par rapport aux variables $i_1, \ldots, i_k$ En pratique, une solution simple pour stabiliser le calcul de dérivation consiste à utiliser les dérivées d'une fonction de lissage de type gaussien. La fonction de lissage est :

$$G(x, y, \sigma) = \frac{1}{2.\pi.\sigma^2}.e^{-(x^2+y^2)/2.\sigma^2}$$

et ses dérivées $G_{i1\ldots ik}(x, y, \sigma)$ sont données par :

$$G_{i1,\ldots..ik}(x, y, \sigma) = \frac{\partial^n}{\partial_{i1}\ldots\partial_{ik}} G(x, y, \sigma), k = 0\ldots.n$$

$L_{i1\ldots ik}(x, y, \sigma)$ est donc la convolution de l'image I avec la différentielle $G_{i1\ldots ik}(x, y, \sigma)$ de la gaussienne $G(x, y, \sigma)$.

$$L_{i_1},\ldots,_{i_k}(x, y, \sigma) = G_{i1,\ldots..ik}(x, y, \sigma) * I(x, y, \sigma)$$

Il est à noter que la taille σ de la gaussienne détermine la quantité de lissage effectué. Ce σ peut aussi être considéré comme un facteur d'échelle. Il est utilisé dans [KOE87] afin de définir un « local jet » multi-échelle (*multi-scale local jet*). Dans [SCH97] les auteurs appliquent une normalisation de la taille de l'image de façon à obtenir des dérivées invariantes en rotation. Leur principal argument est que le calcul des dérivées est influencé par le rapport largeur/hauteur des pixels de l'image. Ils appliquent donc un facteur de réduction correspondant à ce rapport sur les colonnes de l'image par interpolation linéaire. Dans le même article les auteurs proposent des invariants différentiels basés sur le calcul des « local jet » jusqu'à l'ordre 3.