REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IVA4/M2/2018

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Images et vie artificielle**

Reconstruction d'environnement 3D pour la réalité augmentée sur les dispositifs mobiles

---

Par :

KIDOUS ISMAIL

Soutenu le 24/06/2018, devant le jury composé de :

Chighoub Rabia	MAA	Président
Babahenini Med Chaouki	Professeur	Rapporteur
Boucetta Mebarek	MAA	Examinateur

Remerciements

En tout premier lieu, je remercie **Allah** *de m'avoir donné la force et l'audace pour dépasser toutes les difficultés.*

Je remercie Monsieur **Med Chaouki BABAHENINI**, *Professeur à l'université de Biskra, en tant que Directeur de mon mémoire, il m'a guidé dans mon travail et aidé à trouver des solutions pour avancer.*

Et je remercie ma Famille et surtout ma mère.

Résumé

La démocratisation des smartphones et leur montée en puissance a permis de les rendre plus utiles pour l'homme que pour faire des appels uniquement. La réalité augmentée a su profiter de cette puissance et est devenue plus qu'un phénomène de mode ces derniers temps, en effet les applications ne cessent d'affluer tous les jours et ce presque dans tous les domaines.

Ceci est spécialement dû au fait que la réalité augmentée de par son principe permet d'améliorée la perception de réalité par le biais de contenu additionnel tel que du texte, image ou objets 3D, ce qui nous amène a la problématique traité dans ce document, en effet ce continu additionnel peut être le résultat interne de l'application ou stocké au préalable pendant l'installation ou téléchargé après depuis l'internet par exemple.

Dans ce mémoire, nous avons focalisé sur le continu 3D et sur la combinaison entre la réalité augmentée et la reconstruction 3D à partir d'images pour créer de nouveaux objets 3D. Cette dernière permet à l'utilisateur de s'affranchir des difficultés de modélisation et apprentissage de logiciels de modélisation professionnels pour ajouter de nouveaux objets 3D et les manipuler.

Mots clés :

Réalité Augmentée, Android, SLAM, Reconstruction 3D, RGB-D.

Table des matières

Introduction générale

Les avancées technologiques en électronique et informatique ont toujours ouvert la porte à de nouvelles méthodes pour l'humanité de percevoir la réalité. Avec l'arrivée de l'internet le traitement et la diffusion de l'information ont connu une progression fulgurante.

Dans ce contexte l'arrivée des smartphones et leur démocratisation a permis de générer de nouvelles méthodes pour la création, manipulation et diffusion de l'information, une de ces méthodes qui a su exploiter ce condensé technologique est la réalité augmentée.

Par son principe la réalité augmentée permet d'enrichir notre perception de la réalité en lui ajoutant de l'information complémentaire sous forme de texte, image, son et surement sous forme de contenu 3D.

Insérer un objet 3D pourrait constituer un réel défi si l'objet est très complexe à modéliser. Bien sûr, on peut toujours stocker autant d'objets 3D qu'on veut dans l'application, mais on aura toujours envie d'avoir encore un peu plus, on se pose alors des questions et on propose d'autres défis comme :

Et si ça affecterait aussi l'interaction temps réel de l'application ?

Et si l'utilisateur souhaitait ajouter un objet réel ?

Et si l'utilisateur n'est pas habitué aux logiciels de modélisation ?

Pourquoi ne pas offrir à l'utilisateur la possibilité d'ajouter de nouveaux objets en les filmant ?

Surtout si le mobile dispose déjà de l'équipement nécessaire à la reconstruction 3D à partir d'images ?

Manque plus que l'application pour le faire...

Ce mémoire essai de répondre à toutes ces interrogations, il sera organisé en 4 chapitres. Le premier sera consacré à la réalité augmentée en essayant d'en explorer l'étendue des possibilités offertes par cette technologie en remontant depuis ces origines aux labos de recherche jusqu'à son explosion avec l'arrivé des smartphones.

Le deuxième chapitre quant à lui se focalisera sur la reconstruction 3D à partir d'images comme méthode alternative des méthodes classiques de modélisation, bien que ce domaine est toujours très actif et les articles continuent d'affluer sur le sujet, des résultats assez satisfaisants sont obtenues par les chercheurs et donc la technique est parfaitement exploitable, c'est ce qui nous a encourager a vouloir intégrer cette méthode à la réalité augmentée. Nous allons explorer les principes de cette méthode et son état de l'art ainsi qu'au différentes approches utilisées.

Le troisième chapitre se consacré à la conception de notre application et le dernier chapitre présentera notre implémentation agrémentée des résultats obtenus.

Enfin nous allons clôturer ce mémoire par une conclusion générale où nous discuterons des résultats obtenus et des difficultés rencontrés au cours de notre projet ainsi que les perspectives à voir pour le futur.

CHAPITRE I :

REALITE AUGMENTEE

1.Définition

La réalité augmentée désigne selon **Ronald T Azuma,** chercheur à l'Université de Caroline du Nord et auteur d'une des premières études sur la réalité augmentée intitulée *"**A survey of Augmented reality**"*, publié en 1997, la réalité augmentée peut se définir comme une interface entre des données "virtuelles" et le monde réel [7].

Concrètement, la réalité augmentée combine le monde réel et les éléments numériques en temps réel, offre à l'utilisateur des possibilités d'interaction en temps réel, et repose généralement sur un environnement 3D.

Initialement dérivée de la notion de réalité virtuelle, le terme de réalité augmentée est toutefois de plus en plus remis en question [6]. En plus d'être peu compréhensible, ce terme est inexact. Techniquement, ce n'est pas la réalité qui est augmentée, mais bien la perception de l'utilisateur.

La plupart du temps, la réalité augmentée permet d'altérer la vision de l'usager. Toutefois, les cinq sens peuvent être affectés par cette technologie. Il est par exemple possible d'ajouter des sons artificiels à un environnement sonore.

La réalité augmentée désigne toutes les interactions entre une situation réelle et des éléments virtuels tels que de la 3D, des images 2D ou de la géolocalisation. Cette interaction est rendue possible par un **"device"**, à savoir un appareil qui va faire office d'unité de calcul. Cet appareil va permettre de positionner et de suivre les éléments numériques en temps réel [6].

Selon **Fabrice Arsicot,** Directeur du Pôle Digital, de la société **Publicorp,** la réalité augmentée se compose de plusieurs éléments :

- **Un *device*** : du type smartphone, tablette ou ordinateur équipé a minima d'une webcam ou capteur caméra, ainsi que d'une application. Le but sera de mixer virtuel et réel en superposant l'image captée par l'objectif et un contenu généré.

- **Un *marqueur* ou *déclencheur*** : un symbole, une image, voire même un logo si le niveau de contraste est suffisant.
- **Le *contenu généré par l'application*** : 2D, 3D, vidéo, jeu, etc.

Pour définir la réalité augmentée, **on évoque bien souvent les travaux de Milgram et Kishino sur le continuum entre réel et virtuel publiés en 1994**. Selon ces deux experts, la réalité mixte va de l'environnement réel à l'environnement virtuel en passant par la réalité augmentée et la virtualité augmentée [2].

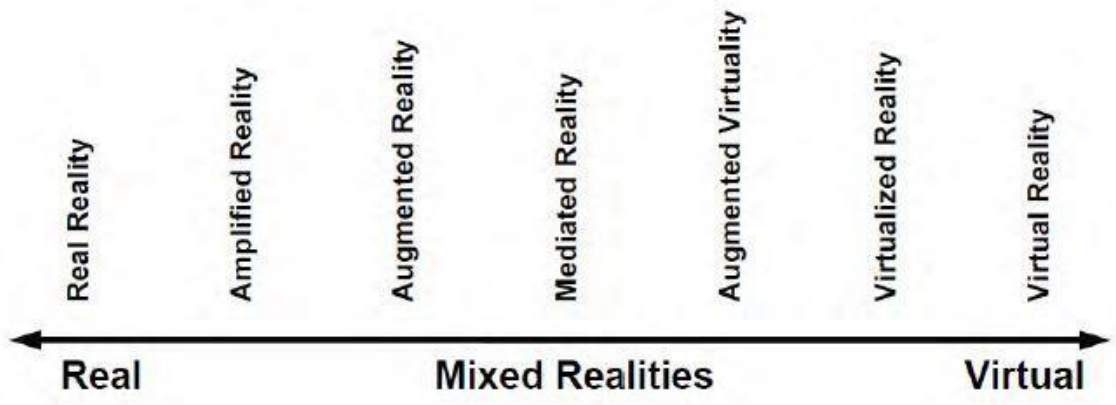


Figure 1.1 : les différents niveaux de réalités mixées

2.L'Histoire de la réalité augmentée

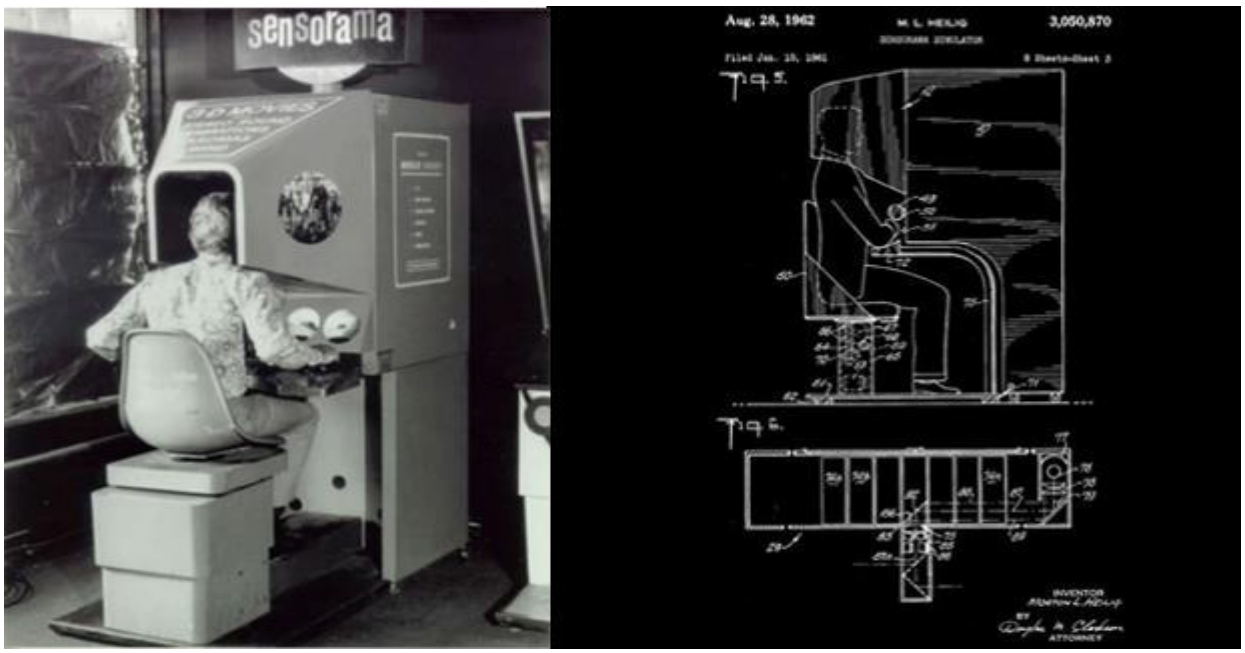


Figure 1.2 : le sensorama

La réalité augmentée a pour la première fois été conceptualisée par **Horton Heilig**, qui en **1962 crée le "Sensorama"**. Il s'agit d'un casque équipé de capteurs permettant de simuler une scène telle qu'une balade à moto dans New York par exemple. Ce **concept se rapproche davantage de la réalité virtuelle, mais pose également les bases de la réalité augmentée**.

En 1968**, Ivan sutherland invente le premier casque HUD à vision transparente réagissant aux mouvements de tête**. L'appareil est baptisé « Épée de Damoclès », car il se porte directement sur la tête.

Dans les années 1980, **Steve Mann invente le EyeTap, un casque permettant d'afficher des informations virtuelles devant les yeux de son utilisateur**. Il s'agit du premier véritable casque de réalité augmentée, un concept qui se développera jusqu'à devenir aussi léger qu'une paire de lunettes [6].

3. Comment fonctionne la réalité augmentée ?

Il existe plusieurs techniques de création de la réalité augmentée. Elle consiste à construire une représentation se superposant au monde réel. Pour cela, les créateurs de réalité augmentée utilisent des coordonnées géographiques qui permettent une localisation. Sont donc utilisées des données GPS auxquelles doivent être associés des éléments précisant la direction de la vision. Si l'on prend l'exemple d'un smartphone, ce sont la boussole et les accéléromètres qui fournissent les informations nécessaires pour permettre au logiciel de savoir où se trouve l'usager et la direction vers laquelle il est tourné.

Il est **également possible de s'en remettre à la reconnaissance d'une image ou d'un motif lié au sens augmenté pour déterminer la position de l'utilisateur**. Dans le cas de la vision, c'est la reconnaissance d'image qui permet cette prouesse.

Aux débuts de la réalité augmentée, les marqueurs étaient monochromes et dissymétriques. Ces marqueurs simples sont encore utilisés aujourd'hui par certaines applications. Désormais toutefois, **les ordinateurs et smartphones sont suffisamment puissants pour reconnaître des images plus complexes grâce à des algorithmes prévus à cet effet**. [3]

Les **applications de réalité augmentée peuvent également analyser des flux vidéo ou des éléments du corps humain, les mains ou l'ensemble des membres, comme le fait le Kinect de Microsoft**. Ce procédé est par exemple utilisé par l'application de test de maquillage L'Oréal Makeup Genious.

Dans un avenir proche, **il est probable que tous les équipements portables disposent d'un capteur de reconnaissance du corps humain permettant de proposer davantage d'interactions entre le réel et le virtuel**. C'est déjà le cas du smartphone Asus Zenfone AR, reposant sur la technologie Google Tango, présenté lors du CES 2017. Selon les rumeurs, le prochain iPhone attendu pour la fin de l'année 2017 proposerait des fonctionnalités similaires.

4. Quels outils pour représenter la réalité augmentée ?

Concrètement, **la réalité augmentée nécessite un ensemble de capteurs permettant de localiser l'utilisateur** (GPC, caméra, accéléromètre, hygromètre, hydromètre…), un ordinateur pour interpréter l'environnement et le mélanger aux éléments virtuels, et un écran permettant d'afficher le résultat de ce mélange.

À l'origine, **les premiers logiciels de réalité augmentée reposaient sur une caméra, un ordinateur et un écran**. Depuis 2009 [7], suite à l'essor des smartphones et du réseau 3G/4G, le téléphone est devenu la principale interface de visualisation de la réalité augmentée. Cette caractéristique distingue la réalité augmentée de la réalité virtuelle, reposant principalement sur les casques VR comme **l'Oculus Rift**. Le principal avantage du smartphone est de permettre d'exploiter les données de géolocalisation grâce à sa portabilité. De plus, ces appareils à la pointe de la technologie peuvent se charger des calculs, au même titre qu'un ordinateur.

Depuis quelques années toutefois, d'autres outils ont fait leur apparition. Les **lunettes Google Glasses ont été pour la première fois présentées en 2012, puis commercialisées en 2013, et retirées du marché l'année suivante**. Depuis lors, les lunettes de réalité augmentée prolifèrent. Ces appareils sont particulièrement pratiques pour les sportifs, puisqu'ils permettent d'afficher des informations utiles sur la visière tout en laissant les mains libres à l'usager.

Les **constructeurs automobiles, tel que PSA, travaillent également actuellement à la construction d'un pare-brise équipée d'un outil de réalité augmentée**. Les premiers véhicules de la marque qui en seraient équipés pourraient en être équipés en 2020. Enfin, certains constructeurs envisagent d'utiliser les montres connectées comme interfaces de réalité augmentée, même si aucun projet concret n'a été dévoilé jusqu'à présent.

Les composants matériels pour la réalité augmentée sont : le processeur, l'affichage, les capteurs et les dispositifs d'entrée. Les appareils informatiques mobiles modernes tels que les smartphones et les tablettes électroniques contiennent ces éléments qui comprennent souvent une

caméra et des capteurs MEMS tels qu'un accéléromètre, un GPS et une_boussole à semi-conducteurs, ce qui en fait des plates-formes AR appropriées [1].

4.1 Affichage :

Divers technologies sont utilisées dans le rendu en réalité augmentée, y compris les systèmes de projection optique, les moniteurs, les dispositifs portatifs et les systèmes d'affichage portés sur le corps humain.

Un visiocasque (HMD) est un dispositif d'affichage porté sur le front, tel qu'un harnais ou un casque. Les HMD placent des images du monde physique et des objets virtuels sur le champ de vision de l'utilisateur. Les HMD modernes utilisent souvent des capteurs pour une surveillance à six degrés de liberté, ce qui permet au système d'aligner les informations virtuelles sur le monde physique et de s'ajuster en conséquence aux mouvements de la tête de l'utilisateur. Les HMD peuvent fournir aux utilisateurs de réalité virtuelle des expériences mobiles et collaboratives. Les fournisseurs spécifiques, tels que **uSens** et **Gestigon**, incluent des contrôles gestuels pour une immersion virtuelle complète.

En janvier 2015, Meta a lancé un projet dirigé par **Horizons Ventures**, **Tim Draper**, **Alexis Ohanian**, **BOE Optoélectronique** et **Garry Tan** . Le 17 février 2016, Meta a annoncé leur produit de deuxième génération chez **TED**, **Meta 2**. Le casque d'affichage monté sur la tête de Meta 2 utilise un tableau sensoriel pour les interactions de main et le suivi de position, avec un champ visuel de 90 degrés (diagonale), et l'affichage de la résolution de 2560 x 1440 (20 pixels par degré), qui est considéré comme le plus grand champ de vision (FOV) actuellement disponible.

4.2 Lunettes de vue :

Figure 1.3 :Vuzix AR3000 Augmented Reality Smart Glasses

Les écrans AR peuvent être rendus sur des appareils ressemblant à des lunettes. Les versions comprennent des lunettes qui utilisent des caméras pour intercepter la vision du monde réel et réafficher sa vue augmentée à travers les oculaires et les dispositifs dans lesquels les images AR sont projetées ou réfléchies sur les surfaces des lentilles des lunettes.

4.3 HUD

Figure 1.4 : Casque Ordinateur

Un **affichage tête haute** (**HUD**) est un affichage transparent qui présente des données sans obliger les utilisateurs à détourner le regard de leurs points de vue habituels. Une technologie précurseur de la réalité augmentée, les affichages tête haute ont d'abord été développés pour les pilotes dans les années 1950 [6], projetant des données de vol simples dans leur ligne de vue, leur permettant ainsi de garder leur tête et ne pas regarder les instruments. Les dispositifs de réalité augmentée à proximité des yeux peuvent être utilisés comme des affichages tête haute portables, car ils peuvent afficher des données, des informations et des images pendant que l'utilisateur visualise le monde réel. De nombreuses définitions de la réalité augmentée la définissent seulement comme recouvrant l'information. C'est essentiellement ce que fait un affichage tête-haute; cependant, en pratique, la réalité augmentée devrait inclure l'enregistrement et le suivi entre les perceptions, les sensations, les informations, les données et les images superposées et une partie du monde réel.

CrowdOptic, une application existante pour smartphones, applique des algorithmes et des techniques de triangulation aux métadonnées de la photo, y compris la position GPS, le cap de la boussole et un horodatage pour arriver à une valeur de signification relative pour les objets photo. La technologie **CrowdOptic** peut être utilisée par les utilisateurs de Google Glass pour apprendre à regarder à un moment donné.

Un certain nombre de "**smartglasses**" ont été lancés pour la réalité augmentée. En raison du contrôle encombré, les lunettes intelligentes sont principalement conçues pour la micro-interaction comme la lecture d'un message texte, mais encore loin des applications plus complètes de la réalité augmentée. En janvier 2015, **Microsoft** a présenté **HoloLens**, une unité de lunettes intelligentes

indépendante. Brian Blau, directeur de la recherche sur les technologies grand public et les marchés chez Gartner, a déclaré : «Parmi tous les visiocasques que j'ai essayés au cours des deux dernières décennies, le HoloLens était le meilleur de sa catégorie. Les premières impressions et opinions étaient généralement que HoloLens est un appareil supérieur à **Google Glass**, et parvient à faire plusieurs choses "correctes" dans lesquelles Glass a échoué.

4.4 Lentilles de contact :

Les lentilles de contact qui affichent l'imagerie AR sont en cours de développement. Ces lentilles de contact bioniques peuvent contenir les éléments d'affichage intégrés dans l'objectif, y compris les circuits intégrés, les DEL et une antenne pour la communication sans fil. Le premier affichage de lentilles de contact a été rapporté en 1999, puis 11 ans plus tard en 2010-2011. Une autre version de lentilles de contact, en développement pour l'armée américaine, est conçue pour fonctionner avec des lunettes AR, permettant aux soldats de se concentrer sur les images AR proches des lunettes. Et les objets du monde réel lointain en même temps. Le court métrage futuriste "***Sight***" propose des appareils de réalité augmentée à lentilles de contact.

De nombreux scientifiques ont travaillé sur des lentilles de contact capables de nombreuses prouesses technologiques. La société Samsung travaille également sur une lentille de contact. Cet objectif, une fois terminé, est destiné à avoir un appareil photo intégré sur l'objectif lui-même. La conception est destinée à vous faire cligner des yeux pour contrôler son interface pour l'enregistrement prévu. Il est également destiné à être relié à votre smartphone pour visionner des vidéos et à les contrôler séparément. En cas de succès, l'objectif comprendrait une caméra ou un capteur à l'intérieur. On dit que cela peut être quelque chose d'un capteur de lumière, à un capteur de température.

En Réalité Augmentée, la distinction est faite entre deux modes distincts de repérage, connus sous le nom de «marqueur» et «sans marqueur». Les marqueurs sont des indices visuels qui déclenchent l'affichage de l'information virtuelle. Un morceau de papier avec des géométries distinctes peut être utilisé. La caméra reconnaît les géométries en identifiant des points spécifiques dans le dessin. "Markerless" également appelé suivi instantané n'utilise pas de marqueur. Au lieu de cela, l'utilisateur place l'objet dans la vue de la caméra de préférence dans un plan horizontal. Il utilise des capteurs dans les dispositifs mobiles pour détecter avec précision l'environnement réel, comme les emplacements des murs et des points d'intersection.

4.5 Affichage rétinien virtuel :

Un dispositif d'affichage rétinien virtuel (VRD) est un dispositif d'affichage personnel en cours d'élaboration au Human Interface Technology Laboratory de l'Université de Washington sous la direction du Dr Thomas A. Furness III [6]. Avec cette technologie, un affichage est scanné directement sur la rétine de l'œil d'un spectateur. Cela donne des images lumineuses avec une haute révolution et un contraste élevé. Le spectateur voit ce qui semble être un affichage conventionnel flottant dans l'espace.

Plusieurs tests ont été effectués afin d'analyser la sécurité du VRD. Dans un test, les patients avec une perte partielle de la vision ont été sélectionnés pour voir des images en utilisant la technologie ayant une dégénérescence maculaire (une maladie qui dégénère la rétine) ou kératocône. Dans le groupe de dégénérescence maculaire, 5 sujets sur 8 préféraient les images VRD aux images CRT ou papier et pensaient qu'ils étaient meilleurs et plus brillants et qu'ils pouvaient voir des niveaux de résolution égaux ou supérieurs. Les patients de Kerocunus peuvent tous résoudre des lignes plus petites dans plusieurs tests de ligne en utilisant le VDR par opposition à leur propre correction. Ils ont également trouvé les images VDR pour être plus facile à voir et plus nette. À la suite de ces différents tests, l'affichage rétinien virtuel est considéré comme une technologie sûre.

L'affichage rétinien virtuel crée des images visibles à la lumière ambiante et ambiante. Le VRD est considéré comme un candidat préféré à utiliser dans un affichage chirurgical en raison de sa combinaison de haute résolution et de contraste et de luminosité élevés. Des tests supplémentaires montrent que VRD peut être utilisé comme technologie d'affichage pour les patients ayant une basse vision.

4.6 EyeTap :

Le EyeTap [7] (également connu sous le nom de Generation-2 Glass) capture les rayons de lumière qui traversent le centre de la lentille de l'utilisateur et substitue la lumière synthétique contrôlée par ordinateur à chaque rayon de lumière réel.

Le verre Generation-4 (Laser EyeTap) est similaire au VRD (c'est-à-dire qu'il utilise une source de lumière laser commandée par ordinateur), à la différence près qu'il a une profondeur de champ infinie et que l'œil fonctionne à la fois une caméra et un affichage par alignement exact avec l'œil et resynthèse (en lumière laser) des rayons de lumière entrant dans l'œil.

Ordinateur de poche

Un écran de poche utilise un petit écran qui tient dans la main d'un utilisateur. Toutes les solutions AR portatives à ce jour optent pour la transparence visuelle. Initialement, l'AR portable utilisait des repères fiduciaires et plus tard des unités GPS et des capteurs MEMS tels que des compas numériques et un accéléromètre à six degrés de liberté - gyroscope . Aujourd'hui, les trackers sans marqueur SLAM tels que PTAM commencent à être utilisés. L'affichage portable AR promet d'être le premier succès commercial pour les technologies AR. Les deux principaux avantages de l'AR portable sont la nature portable des appareils portatifs et la nature omniprésente des téléphones-appareils photo. Les inconvénients sont les contraintes physiques de l'utilisateur qui doit constamment tenir le dispositif de poche devant lui, ainsi que l'effet de distorsion des caméras de téléphone mobile grand angle classiques par rapport au monde réel vu à travers l'œil. Les problèmes découlant du fait que l'utilisateur doit tenir le dispositif portatif (manipulabilité) et percevoir la visualisation correctement (compréhensibilité) ont été résumés dans le questionnaire d'utilisation de HARUS.

Des jeux tels que ***Pokémon Go*** et ***Ingress*** utilisent une interface ILM (Image Linked Map), où les emplacements géomarqués approuvés apparaissent sur une carte stylisée avec laquelle l'utilisateur peut interagir.

4.7 Réalité augmentée Spatiale :

La réalité augmentée spatiale (SAR) augmente les objets et les scènes du monde réel sans l'utilisation d'écrans spéciaux tels que des moniteurs , des visiocasques ou des dispositifs portatifs. SAR utilise des projecteurs numériques pour afficher des informations graphiques sur des objets physiques. La principale différence dans SAR est que l'affichage est séparé des utilisateurs du système. Parce que les affichages ne sont pas associés à chaque utilisateur, le SAR échelonne naturellement à des groupes d'utilisateurs, permettant ainsi une collaboration colocalisée entre les utilisateurs.

Les exemples incluent des lampes de shader, des projecteurs mobiles, des tables virtuelles et des projecteurs intelligents. Les lampes Shader imitent et augmentent la réalité en projetant des images sur des objets neutres, offrant l'opportunité d'améliorer l'apparence de l'objet avec des matériaux d'une unité simple - un projecteur, une caméra et un capteur.

D'autres applications incluent des projections de table et de mur. Une innovation, la table virtuelle étendue, sépare le virtuel du réel en incluant des miroirs de diviseur de faisceau attachés

au plafond à un angle réglable. Les vitrines virtuelles, qui emploient des miroirs de diviseur de faisceau avec de multiples affichages graphiques, fournissent un moyen interactif de s'engager simultanément avec le virtuel et le réel. Beaucoup plus d'implémentations et de configurations font de l'affichage spatial à réalité augmentée une alternative interactive de plus en plus attrayante.

Un système SAR peut s'afficher sur n'importe quel nombre de surfaces d'un réglage d'intérieur à la fois. SAR supporte à la fois une visualisation graphique et une sensation haptique passive pour les utilisateurs finaux. Les utilisateurs sont capables de toucher des objets physiques dans un processus qui procure une sensation haptique passive.

4.8 Suivi :

Les systèmes de réalité augmentée mobiles modernes utilisent une ou plusieurs des technologies de suivi suivantes : caméras numériques et / ou autres capteurs optiques , accéléromètres , GPS , gyroscopes , compas à semi-conducteurs , RFID . Ces technologies offrent différents niveaux de précision et de précision. Le plus important est la position et l'orientation de la tête de l'utilisateur. Le suivi des mains de l'utilisateur ou d'un périphérique d'entrée portable peut fournir une technique d'interaction 6DOF.

4.9 Réseaux :

Les applications mobiles de réalité augmentée gagnent en popularité en raison de l'adoption généralisée des appareils mobiles et surtout portables. Cependant, ils s'appuient souvent sur des algorithmes de vision par ordinateur intensifs en calcul avec des exigences de latence extrêmes. Pour compenser le manque de puissance de calcul, le déchargement du traitement des données vers une machine distante est souvent souhaité. Le déchargement par calcul introduit de nouvelles contraintes dans les applications, notamment en termes de latence et de bande passante. Bien qu'il existe une pléthore de protocoles de transport multimédias en temps réel, il existe également un besoin de support de l'infrastructure réseau.

4.10 Périphériques d'entrée :

Les techniques comprennent des systèmes de reconnaissance vocale qui traduisent les mots parlés d'un utilisateur en instructions informatiques et des systèmes de reconnaissance gestuelle qui interprètent les mouvements du corps par détection visuelle ou à partir de capteurs intégrés dans un périphérique tel qu'une baguette, un stylet, un gant ou autre. les produits qui essayent de

servir de contrôleur des casques d'AR incluent Wave par Seebright Inc. et Nimble par Intugine Technologies.

4.11 Ordinateur :

L'ordinateur analyse les données visuelles et autres détectées pour synthétiser et positionner les augmentations. Les ordinateurs sont responsables des graphiques qui vont avec la réalité augmentée. La réalité augmentée utilise une image générée par ordinateur et elle a un effet frappant sur la façon dont le monde réel est montré. Avec l'amélioration de la technologie et des ordinateurs, la réalité augmentée va changer radicalement notre perspective du monde réel. Selon Time Magazine, dans environ 15-20 ans, il est prédit que la réalité augmentée et la réalité virtuelle vont devenir la principale utilisation pour les interactions informatiques. Les ordinateurs s'améliorent très rapidement, ce qui signifie que nous cherchons de nouvelles façons d'améliorer d'autres technologies. Plus les ordinateurs progressent, plus la réalité augmentée devient plus flexible et plus commune dans notre société. Les ordinateurs sont le noyau de la réalité augmentée.

L'ordinateur reçoit des données provenant des capteurs qui déterminent la position relative de la surface des objets. Cela se traduit par une entrée à l'ordinateur qui produit ensuite aux utilisateurs en ajoutant quelque chose qui ne serait pas autrement là. L'ordinateur comprend une mémoire et un processeur. L'ordinateur prend l'environnement scanné puis génère des images ou une vidéo et le place sur le récepteur pour que l'observateur puisse le voir. Les marques fixes sur la surface d'un objet sont stockées dans la mémoire d'un ordinateur. L'ordinateur retire également de sa mémoire pour présenter des images de façon réaliste au spectateur. Le meilleur exemple est celui de l'abribus Pepsi Max AR.

5. Les limites de la réalité augmentée :

À l'aube de l'an 2017, la réalité augmentée est encore confrontée à de nombreux problèmes. Les **applications de réalité augmentée utilisables en extérieur, doivent faire face au soleil à la température extérieure qui peuvent gêner la perception**. Dans le cas des applications de tourisme AR, par exemple, qui nécessitent de se rendre dans un lieu touristique pour profiter d'éléments virtuels superposés à la réalité, ce désagrément peut ruiner l'expérience [3].

La réalité augmentée peut également poser un problème de sécurité. Par exemple**, un cycliste ou un skieur muni de lunettes de réalité augmentée peut être perturbé par les informations qui s'affichent à l'écran et risquer un accident**. De même, certains n'hésitent pas à jouer à

Pokémon Go en conduisant, provoquant un réel danger pour la sécurité routière. Toutefois, c'est ici le comportement des utilisateurs qui doit être incriminé et non la technologie qui leur est proposée.

Autre problème de la réalité augmentée, celui de la confidentialité. Les **lunettes de réalité augmentée nécessitent une caméra pour fonctionner. Or, cette caméra peut très bien permettre de filmer des personnes à leur insu**. C'est l'un des problèmes qui a provoqué l'échec des Google Glass. Les lunettes AR provoquaient des rixes dans les lieux publics, et certains les utilisaient pour filmer des événements privés.

6. Applications de la réalité augmentée

Avec l'avènement et l'utilisation universelle des dispositifs mobiles, de nombreuses applications intégrant la réalité augmentée se sont développé, nous citons dans ce paragraphe les principales :

6.1 Éducation

La technique de la réalité augmentée permet de créer des manuels numériques artisanaux. L'association d'un contenu à des portions d'un livre permet de dépasser simplement le contenu du texte ou des illustrations. Les applications simples sur tablettes ou smartphones permettent d'envisager son utilisation par les élèves eux-mêmes dans le cadre de la co-construction des outils d'apprentissage.

6.2 Patrimoine

La réalité augmentée est également un atout pour les sites patrimoniaux ; l'abbaye de **Cluny** a ainsi mis en place des bornes qui permettent de voir l'état de l'abbaye au XVe siècle , et à **Cherbourg**, il est possible de visualiser le château fort disparu en 3D grâce à une application téléchargeable sur un smartphone ou une tablette tactile.

Une expérience dans ce domaine a également été menée par la ville d'**Amiens** avec la possibilité de télécharger une application permettant de visualiser la cathédrale en 3D et en couleurs. Dans le cadre d'un projet de maquette virtuelle représentant la ville de **Caen** à la fin des années 1930, une application a été produite pour visualiser le séminaire des **Eudistes de Caen**, détruit en 1944.

Fin 2012 à Bordeaux, le projet **Imayana** propose plusieurs concepts (anima mori, portes temporelles, passe-murailles...) au service de la narration.

6.3 Sciences

La réalité augmentée devient un outil pour les scientifiques et notamment dans le domaine médical, ou elle peut être un outil d'apprentissage. L'usage peut aussi être ludique, dans une approche de vulgarisation, par exemple pour expliquer le fonctionnement du cerveau au grand public.

6.4 Politique

Sébastien Nadot, candidat à l'élection présidentielle de 2017, est le premier homme politique français à utiliser une affiche de campagne à réalité augmentée.

6.5 Presse écrite

Les journaux ***Le Parisien***. Aujourd'hui lus par 4 millions de personnes en France chaque jour, utilisent ce principe afin de rendre leurs quotidiens interactifs et permettre aux lecteurs de réagir aux articles. En mars 2010, *Télé 7 Jours* présente pour ses 50 ans une couverture en réalité augmentée avec Johnny Hallyday et deux publicités interactives.

6.6 Édition

Orange, Robert Laffont et Jacques Attali éditent le premier livre « hyperactif » (**Hyperlivre** - *Le sens des choses*). Édité à 50 000 exemplaires, le livre est parsemé de codes barres 2D permettant d'accéder à des contenus complémentaires. En 2012, la maison d'édition Casterman a publié une bande dessinée (*12 - la douce*) en réalité augmentée. L'histoire porte sur une locomotive mythique en Europe dans les années 1930.

6.7 Musique

Le dernier album du rappeur français **SINIK** (Ballon d'or, 2009) utilise ce principe, en tant que Bonus.

L'album Live! **d'Alexx & MoOonshiners** propose une eBoite à musique en réalité augmentée . Un tag à l'ouverture de l'album permet l'affichage d'un jukebox animé en 3D proposant l'écoute des morceaux de l'album, chacun étant associé à une animation propre.

6.8 Publicité

La réalité augmentée sert par exemple à insérer des encarts publicitaires dans des images vidéo tournées sur des terrains de sport : les logos des entreprises partenaires d'un événement peuvent ainsi être toujours visibles quel que soit l'angle de vue choisi par le réalisateur, il est de plus possible d'afficher différents messages sur un même emplacement. En 2011, Volkswagen réfléchit à un moyen de faire découvrir la Nouvelle Golf Cabriolet dans ses concessions alors que le modèle n'est pas encore sur place. De cette réflexion est venue l'idée de se servir de la réalité augmentée pour voir la voiture à taille réelle. Jusqu'alors, plusieurs marques utilisaient la réalité augmentée pour montrer un produit via un ordinateur ; Volkswagen a innové en le faisant pour la première fois via un smartphone. Présent en presse, sur carte postale et disponible sur le web, le marqueur (image permettant au smartphone de faire apparaître la voiture) relaye l'opération sur un maximum de supports pour faire vivre l'expérience au plus grand nombre.

6.9 Loisirs

La réalité augmentée permet aussi de plonger le spectateur au coeur d'un monde partiellement réel (des décors réels) et partiellement virtuel (des objets, des animaux...). Le spectateur devient acteur en interagissant avec les objets virtuels au moyen de capteurs. Une première application accessible au grand public existe au Futuroscope depuis avril 2008, dans l'attraction *Les Animaux du Futur*, réalisée par la société française Total Immersion.

À l'occasion de la sortie du film ***Arthur et la Vengeance de Maltazard*** en décembre 2009, Dassault Systèmes a conçu un jeu en réalité augmentée sur les paquets de **Chocapic**.

La réalité augmentée est aussi utilisée dans certains jeux vidéo notamment sur la PlayStation Vita de Sony et la 3DS de Nintendo, ainsi que sur smartphone comme le jeu ***Pokémon Go*** sorti en juillet 2016.

6.10 Industrielle

La réalité augmentée permet d'insérer des automobiles n'existant que dans un ordinateur dans des décors réels filmés (villes par exemple) et de les faire interagir avec les éléments réels afin de visualiser l'intégration du prototype dans le paysage réel.

6.11 Commerce

La réalité augmentée devient un outil de vente incontournable et permet de rendre la marque plus attractive et haut de gamme. Elle offre une grande qualité promotionnelle et constitue une vraie aide à la décision d'achat.

Elle fait désormais son entrée dans l'univers du vin. Le consommateur peut scanner l'étiquette d'une bouteille et voir le vigneron en sortir virtuellement afin de lui présenter son domaine et son vin.

De nombreux catalogues, notamment de jouets (LEGO, Playmobil...) intègrent dorénavant la réalité augmentée sous forme de 3D interactive, permettant de visualiser les objets animés et d'interagir avec eux.

6.12 E-commerce

La réalité augmentée est un élément d'aide à la prise de décision dans l'acte d'achat pour l'e-commerce. Cela permet, par exemple dans le secteur du mobilier, de visualiser des meubles dans son propre intérieur grâce à une photo. Le mobilier modélisé en 3D est ainsi représenté dans ses proportions réelles, chez soi ou dans quelque visuel que ce soit, ce qui peut rassurer le cyberacheteur lors de son choix de mobilier (taille, couleur, placement dans la pièce, etc.). Les premiers catalogues en réalité augmentée ont été créés pour **Brisach** ou **Ikea** en 2012.

Les acteurs de l'e-commerce sont de plus en plus nombreux à proposer ce service à leurs clients : les sites de **But**, **Made In Design**, **La Redoute**, par exemple, se sont dotés en 2010 d'une « cabine d'essayage » pour le mobilier et les objets de décoration de leur catalogue.

Dans le domaine de l'optique, les technologies d'essayage virtuel permettent de faciliter l'achat de lunettes. L'internaute était jusqu'alors freiné par l'impossibilité de voir le rendu des lunettes sur son visage ce qui provoquait un faible taux de conversion et un nombre de retours important. Il est possible d'essayer ses lunettes sur photo ou webcam avec les technologies développées par **FittingBox**, une startup toulousaine.

6.13 Tourisme

Le principe de réalité augmentée apparaît dans plusieurs applications sur smartphone (iPhone, Blackberry, Android). La réalité augmentée permet d'enrichir l'expérience du visiteur en proposant des contenus associés à ce qu'il est en train de regarder.

6.14 Cuisine

La réalité augmentée peut permettre d'apporter une aide à la découpe des mets. Comme avec l'application **Slice** pour IPhone qui superpose un patron lorsque l'on filme sa pizza ou son gâteau.

7. Conclusion :

Nous avons présenté dans ce chapitre les grandes lignes de la réalité augmentée et son évolution, de ses applications ainsi que de ses accessoires, ce qui nous a permis de définir notre domaine d'intervention. Après nous allons essayer de répondre à la question concernant l'utilisation du mobile pour créer de nouveaux objets 3D en utilisant simplement la caméra du mobile.

CHAPITRE II :

RECONSTRUCTION DU MODELE 3D A PARTIR D'IMAGES

1. Introduction

Ces dernières années l'infographie a fait d'énormes progrès dans la visualisation de modèles 3D. Beaucoup de techniques ont atteint la maturité et sont actuellement transférés vers le matériel. C'est ce qui explique que dans le domaine de la 3D la performance de visualisation augmente encore plus vite que la Loi de Moore. Ce qui exigeait un ordinateur d'un million de dollar il y a quelques années peut maintenant être réalisé par un ordinateur de jeu qui coûte quelques centaines de dollars. Il est maintenant possible de visualiser des scènes 3D complexes en temps réel.

Cette évolution entraîne une demande importante pour des modèles plus complexes et plus réalistes. Le problème est que même si les outils qui sont disponibles pour la modélisation en trois dimensions deviennent plus puissants, la synthèse de modèles réalistes est difficile et long et donc très coûteuse. Nombre d'objets virtuel s'inspirent d'objets réels et il serait donc intéressant de pouvoir acquérir les modèles directement à partir de l'objet réel.

Les chercheurs ont étudié des méthodes pour acquérir des informations 3D des objets et des scènes pendant de nombreuses années. Dans le passé, les principales applications étaient inspection visuelle et guidage de robot. De nos jours, cependant la motivation a changé. Il y a une demande de plus en plus de contenu 3D pour l'infographie, réalité virtuelle et de la communication. Cela se traduit par un changement d'orientation pour les besoins. La qualité visuelle devient l'un des principaux points d'attention. C'est pourquoi non seulement la position d'un petit nombre de points doit être mesurées avec une grande précision, mais la géométrie et l'aspect de tous les points de la surface doivent être mesurées.

Les conditions d'acquisition et de l'expertise technique des utilisateurs dans ces nouveaux domaines d'application ne peuvent souvent pas être associées avec les exigences des systèmes existants. Ceux-ci exigent des procédures complexes de calibration chaque fois que le système est

utilisé. Il y a une demande importante de souplesse dans l'acquisition. Les procédures de calibration doivent être absentes ou réduites au minimum.

Dans ce chapitre, il est expliqué comment obtenir un modèle 3D d'une séquence d'images prises avec des appareils photo ordinaires. L'utilisateur acquiert les images en déplaçant librement la caméra autour de l'objet. Ni les mouvements de caméra ni les réglages de l'appareil doivent être connus. Le modèle 3D obtenu est une version à l'échelle de l'objet original (c'est-à-dire une reconstruction *métrique*), et l'albédo de surface est obtenu à partir de la séquence d'images aussi bien.

2. La 3D à partir d'images

Dans cette section, nous allons essayer de formuler une réponse aux questions suivantes. Que nous disent des images sur une scène 3D ? Comment pouvons-nous obtenir des informations 3D de ces images ? Que devons-nous savoir au préalable ? Quelques problèmes et les difficultés seront également présentés.

Une image comme sur la Figure 2.1_nous parle beaucoup de la scène observée. Il n'y a cependant pas suffisamment d'informations pour reconstituer la scène 3D (du moins ne pas sans faire un nombre important d'hypothèses sur la structure de la scène). Cela est dû à la nature du processus de formation image qui consiste en une projection d'une scène en trois dimensions sur une image en deux dimensions. Au cours de ce processus, la profondeur est perdue.

Figure 2.1 : Une image d'une scène

Figure 2.2 illustre cette situation. Le point en 3D correspondant à un point d'image spécifique est contraint d'être sur la ligne de mire associé. Partir d'une image unique, il n'est pas possible de déterminer quel point de cette ligne correspond au point de l'image.

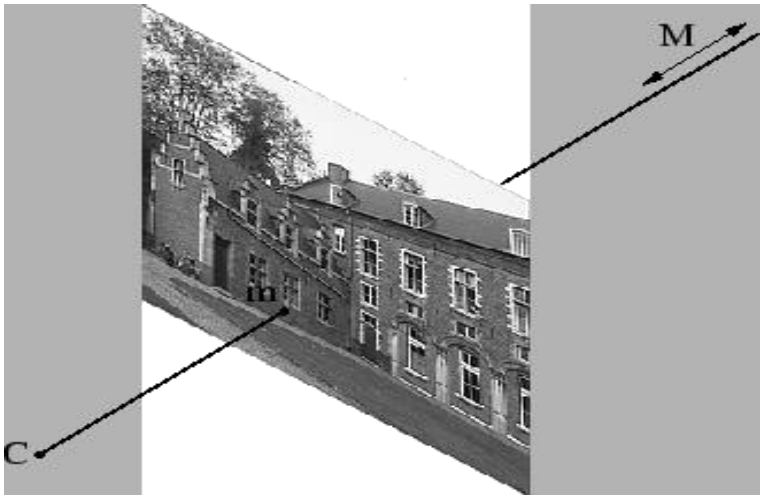


Figure 2.2 : reprojection d'un point le long de la ligne de mire.

Si deux (ou plusieurs) images sont disponibles, alors – comme illustré sur la Figure 2.3 - le point en 3d peut être obtenu comme l'intersection de la ligne de deux sites. Ce processus s'appelle triangulation. Notez, cependant, qu'un certain nombre de choses est nécessaires pour cela :

- Points de l'image correspondante
- Position relative de la caméra pour les différentes vues
- Relation entre les points de l'image et le correspondant de la ligne de mire

La relation entre un point de l'image et sa ligne de mire est donnée par le modèle d'appareil photo (par exemple, le sténopé) et les paramètres de calibration. Ces paramètres sont souvent appelés les paramètres *intrinsèques* de la caméra tandis que la position et l'orientation de la caméra sont en général appelés paramètres *extrinsèques.*

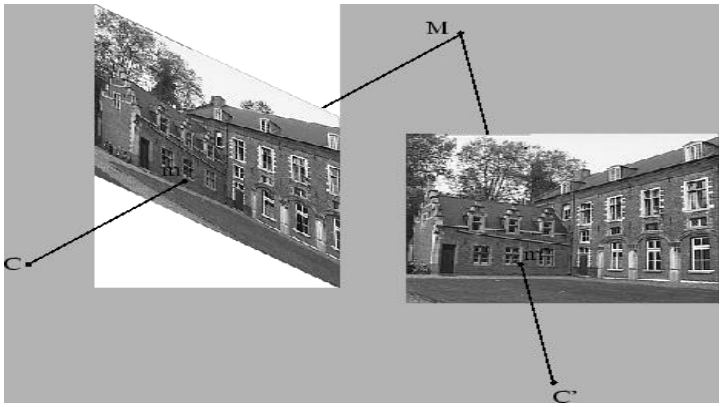


Figure 2.3 : Reconstruction tridimensionnelle point par triangulation.

3. Relier les images

A partir d'une collection d'images ou d'une séquence vidéo, la première étape consiste à relier les différentes images entre elles. Ce n'est pas un problème facile. Un nombre restreint de points correspondants est suffisant pour déterminer la relation géométrique ou *les contraintes multi-vues* entre les images. Comme tous les points ne sont pas également adaptés pour l'appariement ou le suivi (par exemple un pixel dans une région homogène), la première étape consiste à sélectionner un nombre de points intéressants ou de *points caractéristiques*. Certaines approches utilisent également d'autres fonctionnalités, telles que des lignes ou des courbes, mais elles ne seront pas abordées ici. En fonction du type de données d'image (vidéo ou images fixes), les points caractéristiques sont suivis ou appariés et un certain nombre de correspondances potentielles sont obtenues. À partir de ceux-ci, les contraintes multi-vues peuvent être calculées. Cependant, comme le problème de la correspondance est un problème mal posé, l'ensemble des points correspondants peut être contaminé par un nombre important de correspondances erronées ou *aberrantes*. Dans ce cas, une approche traditionnelle des moindres carrés échouera et, par conséquent, une méthode robuste est nécessaire. Une fois les contraintes multi-vues obtenues, elles peuvent être utilisées pour guider la recherche de correspondances supplémentaires. Ceux-ci peuvent ensuite être utilisés pour affiner les résultats des contraintes multi-vues.

3.1 Extraction de points et correspondance

Avant de discuter de l'extraction des points caractéristiques, il est nécessaire d'avoir une mesure pour comparer des parties d'images. L'extraction et la mise en correspondance des fonctionnalités sont basées sur ces mesures. Outre la fonction ponctuelle simple, un type de fonction plus avancé est également présenté.

3.1.1 Comparaison des régions d'image

Les régions d'image sont généralement comparées en utilisant des différences de somme de carrés (SSD) ou une corrélation croisée normalisée (NCC). Considérons une fenêtre $\boldsymbol{W}$ dans l'image $\boldsymbol{I}$ et une région correspondante $\mathbf{T(W)}$ dans l'image $\boldsymbol{J}$. La *dissemblance* entre deux régions d'image basées sur SSD est donnée par

$$D = \int \int_W [J(\mathbf{T}(x,y)) - I(x,y)]^2 \, w(x,y) dx dy \qquad \text{(D1)}$$

où $\boldsymbol{w(x,y)}$ est une fonction de pondération définie sur $\boldsymbol{W}$. Généralement, $\boldsymbol{w(x,y) = 1}$ ou c'est un Gaussien. La mesure de *similarité* entre deux régions d'image basée sur NCC est donnée par

$$S = \frac{\int\int_W (J(\mathbf{T}(x,y)) - \bar{J}).(I(x,y) - \bar{I})w(x,y)dxdy}{\sqrt{\int\int_W (J(\mathbf{T}(x,y)) - \bar{J})w(x,y)dxdy}.\sqrt{\int\int_W (I(x,y) - \bar{I})w(x,y)dxdy}} \quad \text{(D2)}$$

avec $\bar{I} = \int\int_W I(x,y)dxdy$ et $\bar{J} = \int\int_W J(T(x,y))dxdy$ l'intensité moyenne de l'image dans la région considérée. Notez que cette dernière mesure est invariante aux changements globaux d'intensité et de contraste sur les régions considérées.

3.1.2 Extraction de points caractéristiques

L'une des exigences les plus importantes pour un point caractéristique est qu'il peut être différencié de ses points d'image voisins. Si ce n'était pas le cas, il ne serait pas possible de le faire correspondre uniquement avec un point correspondant d'une autre image. Par conséquent, le voisinage d'une entité devrait être suffisamment différent des quartiers obtenus après un petit déplacement.

Une approximation du second ordre de la dissemblance, telle que définie dans l'équation. (D1), entre une fenêtre d'image ***W*** et une fenêtre d'image légèrement traduite est donnée par

$$D(\Delta x, \Delta y) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \mathbf{M} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \text{ with } \mathbf{M} = \int\int_W \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} w(x,y)dxdy \quad \text{(D3)}$$

Pour s'assurer qu'il n'existe aucun déplacement pour lequel **D** est petit, les valeurs propres de **M** devraient être grands. Cela peut être réalisé en imposant une valeur minimale pour la plus petite valeur propre ou alternativement pour la fonction de réponse de coin suivante $R = \det \mathbf{M} - k(\text{trace } \mathbf{M})^2$ [11] où k est un paramètre mis à 0,04 (une suggestion de Harris). Dans le cas d'un suivi, cela suffit pour assurer le suivi des fonctionnalités d'une image vidéo à l'autre. Dans ce cas, il est naturel d'utiliser le voisinage de suivi pour évaluer la qualité d'une caractéristique (par ex. 7 x7 fenêtre avec ***w(x,y)* = 1**). Le suivi se fait en minimisant l'Eq D1 sur les paramètres de **T**. Pour les petites étapes, une traduction est suffisante pour **T**. Pour évaluer la différence accumulée depuis le début de la piste, il est conseillé d'utiliser un modèle de mouvement affine.

Dans le cas de trames séparées obtenues avec une caméra fixe, il est également nécessaire d'extraire autant de points d'image provenant des mêmes points 3D que possible. Par conséquent, seuls les maximas locaux de la fonction de réponse de coin sont considérés comme des entités. La précision du sous-pixel peut être obtenue grâce à une approximation quadratique du voisinage des

maxima locaux. Un choix typique pour ***w(x)*** dans ce cas est un gaussien avec $\sigma = 0.7$. L'appariement est généralement effectué en comparant petit, par exemple 7*7, fenêtres centrées autour du point d'intérêt via SSD ou NCC. Cette mesure est seulement invariante aux translations d'images et ne peut donc pas faire face à de trop grandes variations dans la pose de la caméra.

Pour faire correspondre les images qui sont largement séparées, il est nécessaire de faire face à un plus grand ensemble de variations d'images. La recherche exhaustive sur toutes les variations possibles est informatiquement non traçable. Une approche plus intéressante consiste à extraire une caractéristique plus complexe qui détermine non seulement la position, mais aussi les autres inconnues d'une transformation affine locale.

Dans la pratique souvent beaucoup trop de coins sont extraits. Dans ce cas, il est souvent intéressant de limiter d'abord le nombre de coins avant d'essayer de les faire correspondre. Une possibilité consiste à ne sélectionner que les coins avec une valeur ***R*** au-dessus d'un certain seuil. Ce seuil peut être réglé pour obtenir le nombre de caractéristiques souhaité. Puisque pour certaines scènes, la plupart des coins les plus forts sont situés dans la même zone, il peut être intéressant d'affiner ce schéma pour s'assurer que dans chaque partie de l'image on trouve un nombre suffisant de coins.

En figure 2.5 les parties correspondantes de deux images sont montrées. Dans chacun la position de 5 coins est indiquée. En figure 2.6 le voisinage de chacun de ces coins est montré. L'inter corrélation d'intensité a été calculée pour chaque combinaison possible. Ceci est illustré dans le tableau 2.7. On peut voir que dans ce cas, la paire correcte correspond à toutes les valeurs de corrélation croisée les plus élevées (c'est-à-dire les valeurs les plus élevées sur la diagonale). Cependant, la combinaison 2-5, par exemple, est très proche de 2-2. En pratique, on ne peut certainement pas compter sur le fait que tous les appariements seront corrects et que les procédures d'appariement automatique devraient donc pouvoir traiter une fraction importante des valeurs aberrantes. Par conséquent, d'autres procédures d'appariement robustes seront introduites.

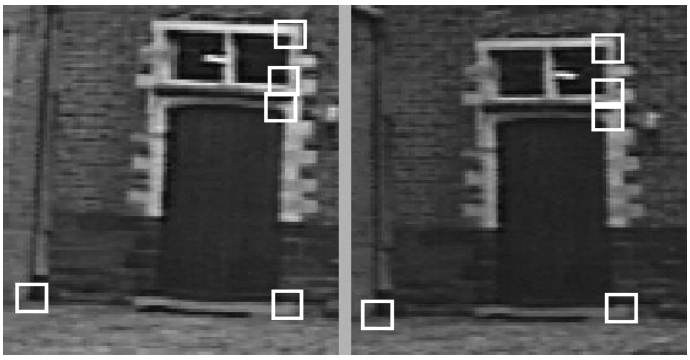


Figure 2.5: Détail des images de la figure 2.4 avec 5 coins correspondants.

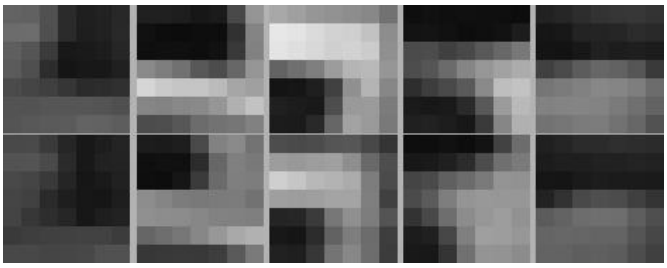


Figure 2.6: voisinage local des 5 coins de la figure 2.5.

Si l'on peut supposer que le mouvement entre deux images est petit (ce qui est nécessaire de toute façon pour que la mesure d'intercorrélation d'intensité donne de bons résultats), l'emplacement de l'entité ne peut pas changer largement entre deux vues consécutives. Cela peut donc être utilisé pour réduire la complexité combinatoire de l'appariement. Seules les entités ayant des coordonnées similaires dans les deux images seront comparées. Pour un coin situé à ***(x,y)***, seuls les coins de l'autre image avec les coordonnées situées dans l'intervalle$[x-w_i,\ x+\ w_i]\times[y-h_i,\ y+h_i]$. w_i et h_i sont typiquement 10% ou 20% de l'image

3.1.3 Correspondance en utilisant des régions affines invariantes

On peut noter que la mesure de similarité présentée dans la section précédente est seulement invariante à la translation et aux offsets dans les valeurs d'intensité. Si une rotation ou une mise à l'échelle importante a lieu, la mesure de similarité n'est pas appropriée. La même chose est vraie lorsque les conditions d'éclairage diffèrent trop. Par conséquent, l'approche basée sur la corrélation croisée ne peut être utilisée qu'entre des images pour lesquelles la caméra n'est pas trop éloignée.

Dans cette section, une procédure d'appariement plus avancée est présentée, qui peut traiter des changements beaucoup plus importants du point de vue et de l'éclairage. Comme cela devrait être clair d'après la discussion précédente, il est important que les pixels correspondant à la même partie de la surface soient utilisés pour la comparaison pendant l'appariement. En supposant que la surface est localement plane et qu'il n'y a pas de distorsion de perspective, les transformations locales de pixels d'une image à l'autre sont décrites par des transformations affines 2D. Une telle transformation est définie par trois points. A ce niveau, nous en avons seulement un, c'est-à-dire le coin considéré, et en avons donc besoin de deux de plus. L'idée est d'aller les chercher le long des bords qui traversent le point d'intérêt. Il est proposé d'utiliser uniquement des coins ayant deux bords connectés à eux, comme dans la figure 2.8.

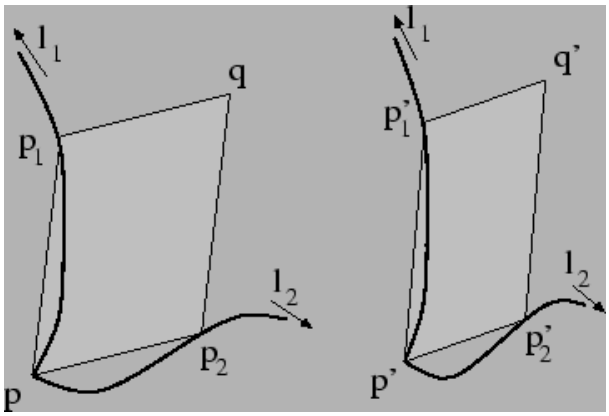


Figure 2.8: Basé sur les bords au voisinage d'un point d'angle p une région affinement invariante est déterminée jusqu'à deux paramètres l_1 et l_2 .

Pour les arêtes courbes, il est possible de relier de façon unique un point sur un bord avec un point sur l'autre arête (en utilisant un paramétrage invariant affine l_1 peut être lié à l_2), ne donnant qu'un seul degré de liberté. Pour les bords droits, deux degrés de liberté sont laissés. Sur la région en forme de parallélogramme (voir figure 2.8) fonctions qui atteignent leurs extrema d'une manière invariante pour les deux changements géométriques et photométriques, sont évalués. Deux fonctions possibles sont :

$$\frac{\int I(x,y)dxdy}{\int dxdy} \text{ and } \frac{|\mathbf{p}-\mathbf{q}\ \ \mathbf{p}-\mathbf{g}|}{|\mathbf{p}-\mathbf{p}_1\ \ \mathbf{p}-\mathbf{p}_2|} \tag{D4}$$

avec ***I (x,y)*** l'intensité de l'image, ***g*** le centre de gravité de la région, pondéré avec l'intensité de l'image et les autres points définis dans la figure 2.8.

$$\mathbf{g} = \left(\frac{\int I(x,y)\,x\,dxdy}{\int I(x,y)\,dxdy}, \frac{\int I(x,y)\,y\,dxdy}{\int I(x,y)\,dxdy}\right) \tag{D5}$$

Les régions pour lesquelles un tel extremum est atteint seront ainsi également déterminées de manière invariable. En pratique, il s'avère que les extrema ne sont souvent pas très bien définis lorsque deux degrés de liberté sont laissés (c'est-à-dire pour des bords droits), mais se produisent dans des « vallées » peu profondes. Dans ces cas, plus d'une fonction est utilisée en même temps et les intersections de ces « vallées » sont utilisées pour déterminer les régions invariantes.

Maintenant que nous avons une méthode à portée de main pour l'extraction automatique de régions d'images locales, invariablement invariantes, celles-ci peuvent facilement être décrites d'une manière affinement invariante en utilisant des invariants de moments. Comme dans la région, les étapes de recherche, l'invariance à la fois sous les changements géométriques affines et les changements photométriques linéaires, avec différents décalages et différents facteurs d'échelle pour chacune des trois bandes de couleur, est considérée.

Pour chaque région, un vecteur caractéristique de l'invariance du moment est composé. Ceux-ci peuvent être comparés assez efficacement avec les vecteurs invariants calculés pour d'autres régions, en utilisant une technique de hachage. Il peut être intéressant de prendre en compte le type de région (courbes ou droites ? Extrema de quelle fonction ?). Une fois que les régions correspondantes ont été identifiées, la corrélation croisée entre elles (après normalisation à une région de référence carrée) est calculée comme une vérification finale pour rejeter les fausses concordances.

3.2 Calcul de la géométrie à deux vues

Comme on l'a vu dans la section précédente, même pour une structure géométrique arbitraire, les projections de points dans deux vues contiennent une certaine structure. Retrouver cette structure n'est pas seulement très intéressant puisqu'elle équivaut à la calibration projective de la caméra pour les deux vues, mais permet aussi de simplifier la recherche de plus de correspondances puisque celles-ci doivent satisfaire la contrainte épipolaire. Cela permet également d'éliminer la plupart des valeurs aberrantes des matchs. Plusieurs algorithmes qui se basent sur la résolution d'un système d'équations linéaires ont été proposés.

Le problème le plus important avec les approches proposées est qu'elles ne peuvent pas faire face aux valeurs aberrantes. Si l'ensemble de correspondance est contaminé par un petit nombre de valeurs aberrantes, le résultat sera probablement inutilisable.

Le problème est qu'il est très difficile de segmenter l'ensemble des correspondances dans les inliers et les outliers avant d'avoir la bonne solution. Les valeurs aberrantes pourraient avoir un effet si désastreux sur les calculs des moindres carrés que presque aucun point ne serait classé comme inliers.

Une solution à ce problème a été proposée par Fischler et Bolles [47]. Leur algorithme est appelé RANSAC (**RAN**dom **SA**mpling **C**onsensus) et peut être appliqué à toutes sortes de problèmes.

3.3 Calcul de géométrie à trois et quatre vues

Il est possible de déterminer la géométrie à trois ou quatre vues de la même manière que le calcul de la géométrie à deux vues expliquée dans la section précédente. Plus de détails sur ces concepts peuvent être trouvés dans [12]. Puisque les points satisfaisant la géométrie à trois ou quatre vues doivent certainement satisfaire la géométrie à deux vues, il est souvent intéressant d'avoir une approche hiérarchique. Dans ce cas, la géométrie des deux vues est estimée d'abord à partir des vues consécutives. Ensuite, des correspondances de triplets sont déduites en comparant deux ensembles consécutifs de paires-correspondances. Ces triplets sont ensuite utilisés dans une

approche robuste similaire à la méthode présentée dans la section 3.2.4. Dans ce cas, seulement 6 triplets de points sont nécessaires. Une approche similaire est possible pour la géométrie à quatre vues.

La méthode de récupération de la structure et du mouvement présentée dans le chapitre suivant ne repose que sur la géométrie des deux vues. Par conséquent, le lecteur intéressé est renvoyé à la littérature pour plus de détails sur le calcul direct de trois et quatre vues des relations géométriques. De nombreux auteurs ont étudié différentes approches pour calculer des relations multi-vues (par ex. [13]).

4. Structure et mouvement

Dans cette section, la relation entre les vues et les correspondances entre les entités sera utilisée pour récupérer la structure de la scène et le mouvement de la caméra. Ce problème s'appelle *Structure et mouvement*.

L'approche proposée ici s'étend en étant entièrement projectif et donc non dépendant de l'initialisation quasi-euclidienne. Ceci a été réalisé en effectuant toutes les mesures dans les images. Cette approche offre une alternative à l'approche triplet proposée dans [14]. Une mesure basée sur l'image qui est capable d'obtenir une distance qualitative entre les points de vue est également proposée pour soutenir l'initialisation et la détermination des vues rapprochées (indépendamment de la trame projective réelle).

Au début, deux images sont sélectionnées et un cadre de reconstruction initial est mis en place. Ensuite, la pose de la caméra pour les autres vues est déterminée dans ce cadre et chaque fois que la reconstruction initiale est affinée et étendue. De cette manière, l'estimation de pose de vues qui n'ont pas de caractéristiques communes avec les vues de référence devient également possible. Généralement, une vue est uniquement associée à son prédécesseur dans la séquence. Dans la plupart des cas, cela fonctionne très bien, mais dans certains cas (par exemple lorsque la caméra se déplace d'avant en arrière), il peut être intéressant de relier une nouvelle vue à un certain nombre de vues supplémentaires. Une fois que la structure et le mouvement ont été déterminés pour toute la séquence, les résultats peuvent être affinés grâce à un ajustement de faisceau projectif. Ensuite, l'ambiguïté sera limitée à la métrique grâce à l'auto-calibration. Enfin, un ajustement du faisceau de métriques est effectué pour obtenir une estimation optimale de la structure et du mouvement.

4.1 Structure initiale et mouvement

La première étape consiste à sélectionner deux vues adaptées à l'initialisation de la structure séquentielle et au calcul de mouvement. D'une part, il est important que des caractéristiques suffisantes soient appariées entre ces vues, d'autre part les vues ne doivent pas être trop proches

l'une de l'autre pour que la structure initiale soit bien conditionnée. Le premier de ces critères est facile à vérifier, le second est plus difficile dans le cas non calibré. La distance basée sur l'image que nous proposons est la distance médiane entre les points transférés à travers une homographie planaire moyenne et les points correspondants dans l'image cible :

$$\text{median}\{D(\mathbf{H}\mathtt{m}_i, \mathtt{m}'_i)\} \tag{E1}$$

Cette homographie planaire **H** est déterminée comme suit à partir des correspondances entre les deux vues :

$$\mathbf{H} = [\mathbf{e}]_\times \mathbf{F} + \mathbf{e}\mathbf{a}^\top_{min} \text{ with } \mathbf{a}_{min} = \text{argmin}_\mathbf{a} \sum_i D(([\mathbf{e}]_\times \mathbf{F} + \mathbf{e}\mathbf{a}^\top)\mathtt{m}_i, \mathtt{m}'_i)^2 \tag{E2}$$

4.1.1 Cadre initial

Deux images de la séquence sont utilisées pour déterminer un cadre de référence. Le cadre du monde est aligné avec la première caméra. La deuxième caméra est choisie pour que la géométrie épipolaire corresponde à la $\mathbf{F}_{12}$:

$$\begin{array}{rcl} \mathbf{P}_1 & = & [\ \mathbf{I}_{3\times 3} \ | \ 0_3\ ] \\ \mathbf{P}_2 & = & [\ [\mathbf{e}_{12}]_\times \mathbf{F}_{12} + \mathbf{e}_{12}\mathbf{a}^\top \ | \ \sigma \mathbf{e}_{12}\ ] \end{array} \tag{E3}$$

Équation E3 n'est pas complètement déterminé par la géométrie épipolaire (ie $\mathbf{F}_{12}$ et $\mathbf{e}_{12}$), mais a 4 autres degrés de liberté (c.-à-d. **a** and **σ**), **a** détermine la position du plan de référence (c'est-à-dire le plan à l'infini dans un cadre affine ou métrique) et **σ** détermine l'échelle globale de la reconstruction. Le paramètre **σ** peut simplement être mis à un ou alternativement la ligne de base entre les deux vues initiales peut être réduite à un. Fixer le coefficient de **a** assurer une trame quasi-euclidienne, pour éviter des distorsions projectives trop importantes. Cela était nécessaire car toutes les parties des algorithmes n'étaient pas strictement projectives.

4.1.2 Structure d'initialisation

Une fois que deux matrices de projection ont été entièrement déterminées, les correspondances peuvent être reconstruites par triangulation. En raison du bruit, les lignes de vue ne se croisent pas parfaitement. Dans le cas non calibré, les minimisations doivent être effectuées dans les images et non dans l'espace 3D projectif. Par conséquent, la distance entre le point 3D reprojeté et les points de l'image doit être minimisée :

$$D(\mathtt{m}_1, \mathbf{P}_1\mathtt{M})^2 + D(\mathtt{m}_2, \mathbf{P}_2\mathtt{M})^2 \tag{E4}$$

Il a été noté par Hartley et Sturm [15] que le seul choix important est de choisir dans quel plan épipolaire le point est reconstruit. Une fois ce choix effectué, il est trivial de sélectionner le point optimal dans le plan. Un faisceau de plans épipolaires n'a qu'un seul paramètre. Dans ce cas,

la dimension du problème est réduite de 3 dimensions à 1 dimension. Minimiser l'équation suivante équivaut donc à minimiser l'équation E4.

$$D(\mathbf{m}_1, \mathbf{l}_1(\alpha))^2 + D(\mathbf{m}_2, \mathbf{l}_2(\alpha))^2 \quad \text{(E5)}$$

avec $l_1(\alpha)$ et $l_2(\alpha)$ les lignes épipolaires obtenues en fonction du paramètre α décrivant l'ensemble de plans épipolaires. Il se trouve que cette équation est un polynôme de degré 6 en α. Le minimum global d'équation E5 peut donc facilement être calculé. Dans les deux images, le point sur la ligne épipolaire $\mathbf{l_1(\alpha)}$ et $\mathbf{l_2(\alpha)}$ le plus proche des points $\mathbf{m_1}$ resp. $\mathbf{m_2}$ est sélectionné. Puisque ces points sont en correspondance épipolaire, leurs lignes de vue se rencontrent en un point 3D.

4.2 Mise à jour de la structure et du mouvement

La section précédente traitait de l'obtention d'une reconstruction initiale à partir de deux vues. Cette section explique comment ajouter une vue à une reconstruction existante. D'abord la pose de la caméra est déterminée, puis la structure est mise à jour en fonction de la vue ajoutée et enfin de nouveaux points sont initialisés.

4.2.1 Estimation de pose projective

Pour chaque vue supplémentaire, la pose vers la reconstruction préexistante est déterminée, puis la reconstruction est mise à jour. Ceci est illustré à la figure 2.11.

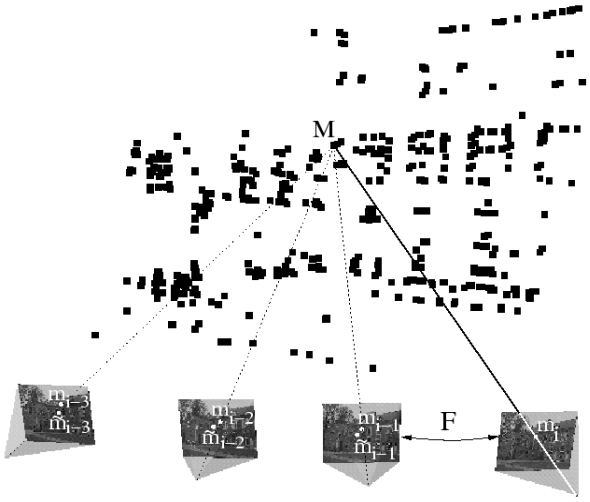


Figure 2.11: Les correspondances d'images .

La première étape consiste à trouver la géométrie épipolaire décrite à la section 3.2 . Ensuite, les correspondances qui correspondent à des points déjà reconstruits sont utilisées pour déduire les correspondances entre 2D et 3D. Sur cette base, la matrice de projection $\mathbf{P_k}$ est calculée à l'aide d'une procédure robuste similaire à celle présentée dans le tableau 2.10. Dans ce cas, un échantillon minimal de 6 correspondances est nécessaire pour calculer $\mathbf{P_k}$. A Un point est considéré comme inlier s'il est possible de reconstruire un point 3D pour lequel l'erreur de reprojection

maximale pour toutes les vues (y compris la nouvelle vue) est inférieure à un seuil prédéfini. Une fois que $\mathbf{P_k}$ a été déterminé, la projection de points déjà reconstruits peut-être prédite. Cela permet de trouver des correspondances supplémentaires pour affiner l'estimation de $\mathbf{P_k}$. Cela signifie que l'espace de recherche est progressivement réduit de l'image complète à la ligne épipolaire à la projection prédite du point.

4.2.2 Relatif à d'autres points de vue

La procédure proposée dans la section précédente ne concerne que l'image à l'image précédente. En fait, on suppose implicitement qu'une fois qu'un point est hors de vue, il ne reviendra pas. Bien que cela soit vrai pour de nombreuses séquences, cette hypothèse ne tient pas toujours. Supposons qu'un point 3D spécifique soit hors de vue, mais qu'il soit à nouveau visible dans les deux dernières vues. Dans ce cas, un nouveau point 3D sera instancié. Cela ne causera pas immédiatement de problèmes, mais puisque ces deux points 3D ne sont pas liés au système, rien n'oblige leur position à correspondre. Pour les séquences plus longues où la caméra est déplacée d'avant en arrière sur la scène, cela peut conduire à des résultats médiocres en raison d'erreurs accumulées. Le problème est illustré à la figure 2.12.

La solution que nous proposons est de faire correspondre toutes les vues qui sont proches de la vue réelle (comme décrit dans la section 3.2). Pour chaque vue rapprochée, un ensemble de correspondances 2D-3D potentielles est obtenu. Ces ensembles sont fusionnés et la matrice de projection de caméra est estimée en utilisant la même procédure robuste que celle décrite ci-dessus, mais sur l'ensemble fusionné de correspondances 2D-3D.

Les vues rapprochées sont déterminées comme suit. D'abord une homographie planaire qui explique le mieux le mouvement d'image des points caractéristiques entre la vue réelle et la vue précédente est déterminée (en utilisant l'équation E2). Ensuite, le résidu médian pour le transfert de ces caractéristiques à d'autres vues en utilisant des homographies correspondant au même plan est calculé (voir l'équation E2). Comme la direction du mouvement de la caméra est donnée par les épipôles, il est possible de limiter la sélection aux vues les plus proches dans chaque direction. Dans ce cas, il est préférable de prendre en compte l'orientation pour différencier les directions opposées.

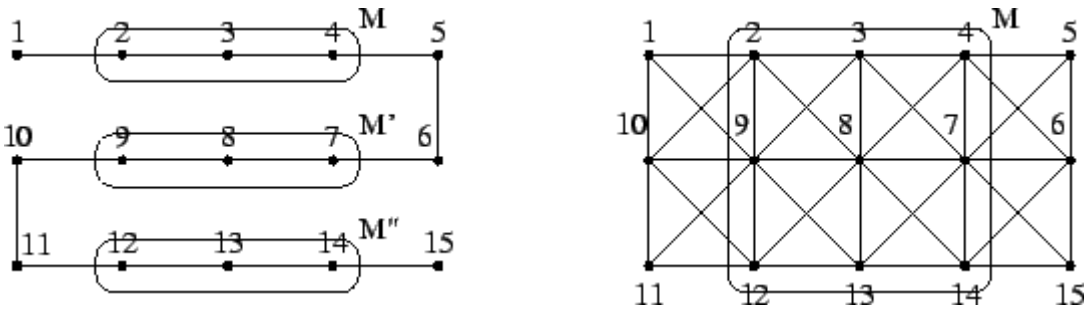


Figure 2.12: *Approche séquentielle (à gauche) et approche étendue (à droite). Dans le schéma traditionnel, la vue 8 serait assortie aux vues 7 et 9 seulement. Un point qui serait visible dans les vues 2,3,4,7,8,9,12,13 et 14 aboutirait donc à 3 points reconstruits indépendamment. Avec l'approche étendue, un seul point sera instancié. Il est clair que ceci entraîne une plus grande précision pour le point reconstruit, tout en réduisant considérablement l'accumulation d'erreurs de calibration.*

4.2.3 Raffinage et extension de la structure

La structure est affinée en utilisant un algorithme de reconstruction linéaire itéré sur chaque point. Comme suit :

$$\begin{aligned} \mathbf{P}_3\mathbf{M}x - \mathbf{P}_1\mathbf{M} &= 0 \\ \mathbf{P}_3\mathbf{M}y - \mathbf{P}_2\mathbf{M} &= 0 \end{aligned} \qquad \text{(E6)}$$

avec $\mathbf{P}_i$ la i-ième rangée de **P** et (x,y) étant les coordonnées de l'image du point. Une estimation de **M** est calculée en résolvant le système d'équations linéaires obtenu à partir de toutes les vues où un point d'image correspondant est disponible. Pour obtenir une meilleure solution, le critère $\sum D(PM, m)$ devrait être minimisé. Ceci peut être approximativement obtenu en résolvant de manière itérative les équations linéaires pondérées suivantes (sous forme matricielle) :

$$\frac{1}{\mathbf{P}_3\tilde{\mathbf{M}}} \begin{bmatrix} \mathbf{P}_3x - \mathbf{P}_1 \\ \mathbf{P}_3y - \mathbf{P}_2 \end{bmatrix} \mathbf{M} = 0 \qquad \text{(E7)}$$

Où $\tilde{\boldsymbol{M}}$ est la solution précédente pour $\boldsymbol{M}$. Cette procédure peut être répétée plusieurs fois. En résolvant ce système d'équations par SVD, un point homogène normalisé est automatiquement obtenu. Si un point 3D n'est pas observé, la position n'est pas mise à jour. Dans ce cas, on peut vérifier si le point a été vu dans un nombre suffisant de vues à conserver dans la reconstruction finale. Ce nombre minimum de vues peut par exemple être mis à trois. Cela évite d'avoir un nombre important de valeurs aberrantes dues à des correspondances parasites.

Bien sûr, dans une séquence d'images, de nouvelles fonctionnalités apparaîtront dans chaque nouvelle image. Si des correspondances de points qui ne sont pas liées à un point existant dans la structure sont disponibles, un nouveau point peut être initialisé comme dans la section 4.1.2.

Après que cette procédure a été répétée pour toutes les images, on dispose des poses de caméra pour toutes les vues et la reconstruction des points d'intérêt. Dans les autres modules, la

calibration de la caméra est principalement utilisé. La reconstruction elle-même est utilisée pour obtenir une estimation de la plage de disparité pour la correspondance stéréo dense.

4.3 Raffinage de la structure et du mouvement

Une fois que la structure et le mouvement ont été obtenus pour toute la séquence, il est recommandé de l'affiner au moyen d'une étape de minimisation globale. Une estimation du maximum de vraisemblance peut être obtenue par l'*ajustement du faisceau* [10]. L'objectif est de trouver les paramètres de la vue caméra $\mathbf{P_k}$ et les points 3D $\mathbf{M_i}$ pour lesquels les distances quadratiques moyennes entre les points image observés $\boldsymbol{M_{ki}}$ et les points image reprojetés $\mathbf{P_k(M_i)}$ sont minimisées. Le modèle de projection de la caméra doit également tenir compte de la distorsion radiale. Pour $\boldsymbol{m}$ vues et $\boldsymbol{n}$ points le critère suivant doit être minimisé :

$$\min_{\mathbf{P}_k, \mathtt{M}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D(\mathtt{m}_{ki}, \mathbf{P}_k(\mathtt{M}_i))^2 \qquad \text{(E8)}$$

4.4 Traiter avec des plans dominants

La structure projective et l'approche du mouvement décrites dans la section précédente supposent que le mouvement et la structure sont généraux. Lorsque ce n'est pas le cas, l'approche peut échouer. Dans le cas d'un mouvement, cela se produit lorsque l'appareil photo est en rotation pure. Une solution à ce problème a été proposée dans [48]. Ici, nous supposerons que le soin est pris lors de l'acquisition de ne pas prendre plusieurs images de la même position afin que ce problème ne se produise pas.

Des problèmes liés à une scène se produisent lorsque (une partie de) la scène est purement planaire. Dans ce cas, il n'est plus possible de déterminer la géométrie épipolaire de façon unique. Si la scène est plane, le mouvement de l'image peut être entièrement décrit par une homographie. Depuis $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}$ (avec $[\mathbf{e}']_\times$ le produit vecteur avec l'épipole $\mathbf{e}'$), il existe une famille de solutions à 2 paramètres pour la géométrie épipolaire. En pratique, les techniques robustes choisiraient une solution aléatoire basée sur l'inclusion de valeurs aberrantes.

4.4.1 Détection des plans dominants

La première partie de la solution consiste à détecter les cas où seules les entités planaires sont appariées. L'approche de sélection du modèle géométrique robuste d'information (GRIC) proposée dans [16] est brièvement passée en revue. Le GRIC sélectionne le modèle avec le score le plus bas. Le score d'un modèle est obtenu en additionnant deux contributions. Le premier est lié à la qualité de l'ajustement et le second est lié à la parcimonie du modèle. Il est important d'utiliser un estimateur de vraisemblance maximale (MLE) robuste pour estimer les différents modèles de

structure et de mouvement comparés à travers le GRIC. GRIC prend en compte le nombre **n** d'inliers plus les valeurs aberrantes, les résidus $\mathbf{e_i}$, l'écart-type de l'erreur de mesure $\boldsymbol{\sigma}$, la dimension des données **r** , le nombre **k** de paramètres du modèle de mouvement et la dimension **d** de la structure:

$$\mathrm{GRIC} = \sum \rho(e_i^2) + (nd\ln(r) + k\ln(rn)) \quad . \tag{E9}$$

où

$$\rho(e^2) = \min\left(\frac{e^2}{\sigma^2}, 2(r-d)\right) \quad . \tag{E10}$$

Dans l'équation ci-dessus ***nd*ln(r)** représente le terme de pénalité pour la structure ayant ***n*** paramètres ***d*** chacun estimé à partir des **r** observations et **kln(rn)** représente le terme de pénalité pour le modèle de mouvement ayant **k** paramètres estimés à partir des **rn** observations.

4.4.2 Structure projective partielle et récupération de mouvement

La séquence est d'abord traversée et séparée en sous-séquences. Pour les sous-séquences avec une structure 3D suffisante (cas A), l'approche décrite dans la section 4 est suivie de sorte que la structure projective et le mouvement sont récupérés. Lorsqu'un triplet correspond au cas B, seules les entités planaires sont suivies et reconstruites (en 2D). Un partitionnement possible d'une séquence d'images est donné dans le Tableau suivant.

Tableau: Exemple sur la façon dont une séquence serait partitionnée en fonction des différents cas obtenus dans l'étape de sélection du modèle. Les F soulignés correspondent à des cas qui ne seraient pas traités de manière appropriée en utilisant une analyse par paire.

Cas	AABAABBBBBAAA
3D	PPPP PPPPP
2D	HH HHHHHH
3D	PPPP
	FF FF FFHHHHHFFFF

Supposons que le plan Π soit étiqueté comme un plan dominant ***i*** en fonction des caractéristiques suivies dans les vues (***i***-1,***i***,***i***+1). En général, certains points caractéristiques $\mathbf{M_\Pi}$ situés sur **Π** auront été reconstruits en 3D à partir de vues précédentes (par ex. ***i*** et (***i***-1)). Par conséquent, les coefficients de **Π** peuvent être calculés à partir de $\mathbf{M_\Pi^T \Pi = 0}$. Définir $\mathbf{M_\Pi}$ comme

l'espace nul de Π^T (matrice 4×3). $\mathbf{M_{\Pi}}$ représente 3 points d'appui pour le plan $\mathbf{\Pi}$ et $\mathbf{m_{\Pi i}=P_i M_{\Pi}}$ soit les projections d'image correspondantes. Définir l'homographie $\mathbf{H_{i\Pi}=m_{\Pi i}^{-1}}$, puis la reconstruction 3D des points d'image situés dans le plan sont obtenus comme suit :

$$\mathbf{M}_i = \mathbf{M}_{\Pi}\mathbf{H}_{i\Pi}\mathbf{m}_i \quad \text{(E11)}$$

De même, une caractéristique $\mathbf{m}_j$ vue en j(>i) peut être reconstruite comme :

$$\mathbf{M}_j = \mathbf{M}_{\Pi}\mathbf{H}_{i\Pi}(\mathbf{H}_{ij}^{\Pi})^{-1}\mathbf{m}_j \quad \text{(E12)}$$

où $\mathbf{H}_{ij}^{\Pi} = \mathbf{H}_{i(i+1)}^{\Pi} \cdots \mathbf{H}_{(j-1)j}^{\Pi}$.

4.4.3 Structure métrique et récupération de mouvement combinée

En utilisant l'algorithme d'auto-calibration couplé décrit au paragraphe 5.3.1 il est possible de récupérer la structure métrique des différentes sous-séquences.

Une fois que la structure métrique des sous-séquences a été récupérée, la pose de la caméra peut également être déterminée pour les points de vue observant uniquement les points planaires. Puisque les intrinsèques ont été calculées, un algorithme d'estimation de position standard peut être utilisé. Nous utilisons l'algorithme de Grunert comme décrit dans [17]. Pour faire face aux valeurs aberrantes.

Finalement, il devient possible d'aligner la structure et le mouvement récupérés pour les sous-séquences séparées en fonction de points communs. Notez que ces points sont tous situés dans un plan et par conséquent certaines précautions doivent être prises pour obtenir des résultats en utilisant des équations linéaires. Cependant, comme 3 points forment une base dans un espace 3D métrique, des points supplémentaires hors du plan peuvent facilement être générés (c'est-à-dire en utilisant le produit vectoriel) et utilisés pour calculer la transformation relative en utilisant des équations linéaires. Ici encore, une approche robuste est utilisée.

Maintenant que tous les paramètres de structure et de mouvement ont été estimés pour toute la séquence. Un dernier ajustement de l'ensemble est effectué pour obtenir une solution globalement optimale.

5. L'Auto-calibration

La reconstruction obtenue comme décrit dans les chapitres précédents n'est déterminée que jusqu'à une transformation projective arbitraire. Cela peut suffire pour certaines applications de robotique ou d'inspection, mais certainement pas pour la visualisation. Par conséquent, nous avons besoin d'une méthode pour mettre à niveau la reconstruction vers une reconstruction métrique (c'est-à-dire déterminée jusqu'à une transformation euclidienne arbitraire et un facteur d'échelle).

En général, trois types de contraintes peuvent être appliquées pour y parvenir : contraintes de scène, contraintes de mouvement de caméra et contraintes sur les intrinsèques de la caméra. Tous ont été essayés séparément ou conjointement. Dans le cas d'une caméra portative et d'une scène inconnue, seul le dernier type de contraintes peut être utilisé. La réduction de l'ambiguïté sur la reconstruction en imposant des restrictions sur les paramètres intrinsèques de la caméra est appelée *auto-Calibration* (dans le domaine de la vision par ordinateur). Au cours des dernières années, de nombreux chercheurs ont travaillé sur ce sujet. La plupart des algorithmes d'auto-Calibration concernent des paramètres de caméra intrinsèques inconnus mais constants (voir par exemple Faugeras et al. [18].

5.1 Calibration

Dans cette section, quelques approches de Calibration existantes sont brièvement discutées. Ceux-ci peuvent être basés sur des connaissances euclidiennes ou métriques sur la scène, la caméra ou son mouvement. Une approche consiste à calculer d'abord une reconstruction projective puis à la mettre à niveau a posteriori vers une reconstruction métrique (ou euclidienne) en imposant certaines contraintes. Les approches traditionnelles vont cependant immédiatement à une reconstruction métrique (ou euclidienne).

5.1.1 Connaissance de la scène

La connaissance des distances (relatives) ou des angles dans la scène peut être utilisée pour obtenir des informations sur la structure métrique. L'un des moyens les plus faciles de calibrer la scène au niveau métrique est la connaissance de la position relative de 5 points ou plus en position générale. Supposer les points $\mathbf{M}_i'$ sont les coordonnées métriques des points reconstruits de manière projective $\mathbf{M}_i$, puis la transformation $\mathbf{T}$ qui met à jour la reconstruction de projective à métrique peut être obtenue à partir des équations suivantes

$$\mathbf{M}_i' \sim \mathbf{T}\mathbf{M}_i \text{ or } \lambda_i \mathbf{M}_i' = \mathbf{T}\mathbf{M}_i \tag{F1}$$

qui peut être réécrit comme des équations linéaires en éliminant λ_i . Boufama et al. [20] ont étudié comment certaines contraintes euclidiennes pouvaient être imposées à une reconstruction non calibrée. Les contraintes auxquelles ils ont été confrontés sont des points 3D connus, des points sur un plan de masse, un alignement vertical et des distances connues entre les points. Bondyfalat et Bougnoux [21] ont récemment proposé une méthode dans laquelle les contraintes sont d'abord traitées par un système de raisonnement géométrique afin d'obtenir une représentation minimale de la scène. Ces contraintes peuvent être l'incidence, le parallélisme et l'orthogonalité. Cette représentation minimale est ensuite appliquée à un ajustement de faisceau contraint.

L'approche traditionnelle des photogrammétristes [10] consiste à imposer immédiatement la position des points de contrôle connus lors de la reconstruction. Ces méthodes utilisent l'ajustement du bundle [22] qui est une minimisation globale de l'erreur de reprojection. Cela peut être exprimé par le critère suivant :

$$C_{bundle} = \sum_{i=1}^{n} \sum_{l \in I_i} \left((x_{li} - \mathbf{P}_i(\mathbf{M}_l))^2 + (y_{li} - \mathbf{P}_i(\mathbf{M}_l))^2 \right) \quad \text{(F2)}$$

où $\mathbf{I_i}$ est l'ensemble des indices correspondant aux points vus $\boldsymbol{i}$ et $\mathbf{P_i(M_l)}$ décrit la projection d'un point $\mathbf{M_l}$ avec un appareil photo $\mathbf{P_i}$ en prenant en compte toutes les distorsions.

Notez que $\mathbf{M_l}$ est connu pour les points de contrôle et inconnu pour d'autres points. Il est clair que cette approche entraîne un énorme problème de minimisation et que, même si la structure particulière du jacobien est prise en compte, il est très coûteux en termes de calcul.

5.1.1.1 Objet de la calibration

Dans le cas d'un objet de calibration, les paramètres de la caméra sont estimés à l'aide d'un objet de géométrie connue. La calibration connue peut ensuite être utilisé pour obtenir des reconstructions immédiatement métriques.

De nombreuses approches existent pour ce type de calibration. La plupart de ces méthodes consistent en une procédure en deux étapes où une Calibration est obtenu d'abord pour un modèle simplifié (linéaire), puis un modèle plus complexe, tenant compte des distorsions, est ajusté aux mesures. La différence entre les méthodes réside principalement dans le type d'objet de calibration attendu (par exemple planaire ou non) ou la complexité du modèle de caméra utilisé.

5.1.2 Connaissance de la caméra

La connaissance de la caméra peut également être utilisée pour limiter l'ambiguïté sur la reconstruction de projective à métrique ou même au-delà. Différents paramètres de la caméra peuvent être connus. Les deux connaissances sur les paramètres extrinsèques (à savoir la position et l'orientation) que les paramètres intrinsèques peuvent être utilisés pour la Calibration.

5.1.2.1 Paramètres extrinsèques

Connaître la position relative des points de vue équivaut à connaître la position relative des points 3D. Par conséquent, la position relative de 5 points de vue en position générale suffit pour obtenir une reconstruction métrique. C'est le principe derrière l'Omni-rig [35] proposé par Shashua. Il est moins évident de traiter les paramètres d'orientation, sauf lorsque les paramètres intrinsèques sont également connus (voir ci-dessous).

5.1.2.2 Paramètres intrinsèques

Si les paramètres intrinsèques de la caméra sont connus, il est possible d'obtenir une reconstruction métrique. Par exemple, cette calibration peut être obtenu par calibration hors ligne avec un objet de calibration. Dans le cas minimal de 2 vues et 5 points, des solutions multiples peuvent exister, mais en général une solution unique est facilement trouvée. La structure traditionnelle des algorithmes de mouvement suppose des paramètres intrinsèques connus et en obtient des reconstructions métriques.

5.1.2.3 Paramètres intrinsèques et extrinsèques

Lorsque les paramètres de caméra intrinsèque et extrinsèque sont connus, la matrice de projection complète de la caméra est déterminée. Dans ce cas, une reconstruction euclidienne est obtenue immédiatement en rétroprojetant les points.

Dans le cas de la position relative et de l'orientation connues des caméras, la première vue peut être alignée sur la trame du monde sans perte de généralité. Si seulement l'orientation (relative) et les paramètres intrinsèques sont connus, le premier 3×3 des matrices de projection de la caméra est connue et il est toujours possible d'obtenir linéairement la transformation qui met à jour la reconstruction projective en métrique.

5.2 Auto Calibration

Dans cette section, quelques concepts importants pour l'auto Calibration sont introduits. Ils sont ensuite utilisés pour décrire brièvement certaines des méthodes d'auto-calibration existantes.

5.2.1 Un argument à prendre en compte

Pour limiter l'ambiguïté projective (15 degrés de liberté) à une métrique (3 degrés de liberté pour la rotation, 3 pour la translation et 1 pour l'échelle), au moins 8 contraintes sont nécessaires. Ceci détermine ainsi la longueur minimale d'une séquence à partir de laquelle l'auto-calibration peut être obtenu, en fonction du type de contraintes disponibles pour chaque vue. *Connaître* un paramètre de caméra intrinsèque pour ***n*** vues donne ***n*** contraintes, *fixant* un seul donne ***n-1*** contraintes.

$$n \times (\#known) + (n-1) \times (\#fixed) \geq 8$$

Bien sûr, cet argument de comptage n'est valable que si toutes les contraintes sont indépendantes. Dans ce contexte, les séquences de mouvement critiques sont d'une importance particulière (voir la section 5.2.5).

Par conséquent, l'absence de biais (1 contrainte par vue) devrait en général être suffisante pour permettre l'auto-calibration sur une séquence de 8 images ou plus. Si en plus le rapport

d'aspect est connu (par ex. $\mathbf{f_x = f_y}$) alors 4 vues devraient suffire. Lorsque le point principal est connu, une paire d'images est suffisante.

5.2.2 Interprétation géométrique des contraintes

Dans cette section, une interprétation géométrique d'une matrice de projection de caméra est donnée. On voit que les contraintes sur les paramètres internes de la caméra peuvent facilement être interprétées géométriquement dans l'espace.

Un plan de projection de caméra définit un ensemble de trois plans. Le premier est parallèle à l'image et passe par le centre de projection. Ce plan peut être obtenu en rétroprojetant la ligne à l'infini de l'image (ie $[001]^T$). Les deux autres correspondent respectivement à la rétroprojection de l'image x- et y-axes (ie $[010]^T$ et $[100]^T$ resp.). Une ligne peut être rétroprojetée par l'équation (C10):

$$\Pi \sim \mathbf{P}^\top \mathbf{l} \sim \begin{bmatrix} \mathbf{R} \\ -\mathbf{t}^\top \mathbf{R} \end{bmatrix} \mathbf{K}^\top \mathbf{l} \tag{F3}$$

Regardons l'orientation relative de ces plans. Par conséquent, la rotation et la translation peuvent être omises sans perte de généralité (c'est-à-dire qu'une représentation centrée sur la caméra est utilisée). Laissez-nous alors définir les vecteurs $\mathbf{v_x}$, $\mathbf{v_y}$ et $\mathbf{v_i}$ comme les trois premiers coefficients de ces plans. Cela donne alors les trois vecteurs suivants :

$$\mathbf{v}_x = \begin{bmatrix} 0 \\ f_y \\ c_y \end{bmatrix}, \mathbf{v}_y = \begin{bmatrix} f_x \\ s \\ c_x \end{bmatrix}, \mathbf{v}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{F4}$$

Les vecteurs coïncidant avec la direction des axes x et y peut être obtenu par des intersections de ces plans:

$$\mathbf{l}_x = \mathbf{v}_x \times \mathbf{v}_i = \begin{bmatrix} f_y \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathbf{l}_y = \mathbf{v}_y \times \mathbf{v}_i = \begin{bmatrix} s \\ -f_x \\ 0 \end{bmatrix} . \tag{F5}$$

Les produits scalaires des points suivants peuvent maintenant être obtenue :

$$\mathbf{l}_x.\mathbf{l}_y = sf_y \,,\ \mathbf{v}_x.\mathbf{v}_i = c_y \text{ and } \mathbf{v}_y.\mathbf{v}_i = c_x \tag{F6}$$

L'équation (F6) prouve que la contrainte pour les pixels rectangulaires s=0), et les coordonnées nulles du point principal $\mathbf{c_x = 0}$ et $\mathbf{c_y = 0}$) peuvent tous être exprimés en termes d'orthogonalité entre vecteurs dans l'espace. Notez en outre qu'il est possible de pré-warp l'image de sorte qu'un biais connu ou les paramètres du point principal connus coïncident avec zéro. De même, une longueur focale ou un rapport d'aspect connu peut être réduit à un.

5.2.4 Méthodes d'auto-Calibration

Dans cette section, certaines approches d'auto-Calibration sont brièvement discutées. Combinaison d'équations (F8) et (F9) on obtient l'équation suivante :

$$\mathbf{K}_i\mathbf{K}_i^\top \sim \mathbf{P}_i\Omega^*\mathbf{P}_i^\top \quad \text{(F11)}$$

Plusieurs méthodes sont basées sur cette équation. Pour les paramètres intrinsèques constants Triggs [41] a proposé de minimiser l'écart par rapport à l'équation (F11). Pollefeys et Van Gool [19] ont proposé une approche connexe basée sur l'équation de transfert (c.-à-d. (F9)) plutôt que l'équation de projection. Ces différentes approches sont très similaires, comme l'a montré [19]. La méthode d'auto-Calibration plus flexible qui permet de modifier les paramètres intrinsèques de la caméra [25] est également basé sur l'équation (F11).

5.2.5 Séquences de mouvement critiques

On a remarqué très vite que toutes les séquences de mouvement ne sont pas adaptées à l'auto-calibration. Certains cas évidents sont les mouvements restreints décrits dans la section précédente (c'est-à-dire la translation pure, la rotation pure et le mouvement planaire). Cependant, il y a plus de séquences de mouvement qui ne conduisent pas à des solutions uniques pour le problème d'auto-calibration. Cela signifie qu'au moins deux reconstructions sont possibles qui satisfont toutes les contraintes sur les paramètres de la caméra pour toutes les images de la séquence et qui ne sont pas liées par une transformation de similarité.

Plusieurs chercheurs ont réalisé ce problème et ont mentionné quelques cas spécifiques ou ont fait une analyse partielle du problème. Sturm [26] ont fourni un catalogue complet des séquences de mouvement critique (CMS) pour les paramètres intrinsèques constants. De plus, il a identifié des dégénérescences spécifiques pour certains algorithmes.

Cependant, il est très important de noter que les classes de CMS existantes dépendent des contraintes qui sont appliquées lors de l'auto-calibration. Les extrêmes étant tous les paramètres connus, dans ce cas il n'y a presque pas de dégénérescence, et pas de contraintes du tout, dans ce cas toutes les séquences de mouvement sont critiques.

6. Estimation de la profondeur dense

Avec la calibration de la caméra donné pour tous les points de vue de la séquence, nous pouvons procéder avec des méthodes développées pour la structure calibrée à partir d'algorithmes de mouvement. L'algorithme de suivi des caractéristiques délivre déjà un modèle de surface clairsemé basé sur des points caractéristiques distincts. Cependant, ceci n'est pas suffisant pour reconstruire des modèles de surface géométriquement corrects et visuellement agréables. Cette

tâche est accomplie par une disparité dense correspondant qui estime les correspondances à partir des images de niveau de gris directement en exploitant des contraintes géométriques supplémentaires.

Ce chapitre est organisé comme suit. Dans une première section, la rectification est discutée. Cela permet d'utiliser des techniques de couplage stéréo standard sur des paires d'images. La correspondance stéréo est abordée dans une deuxième section. Enfin, une approche multi-vues permettant d'intégrer les résultats obtenus à partir de plusieurs paires est présentée.

6.1 Rectification de la paire d'images

Le problème de correspondance stéréo peut être résolu beaucoup plus efficacement si les images sont rectifiées. Cette étape consiste à transformer les images pour que les lignes épipolaires soient alignées horizontalement. Dans ce cas, les algorithmes de couplage stéréo peuvent facilement tirer parti de la contrainte épipolaire et réduire l'espace de recherche à une dimension (c'est-à-dire les rangées correspondantes des images rectifiées).

Le schéma de rectification traditionnel consiste à transformer les plans d'image de sorte que les plans spatiaux correspondants coïncident. Il existe de nombreuses variantes de cette approche traditionnelle. Cette approche échoue lorsque les épipoles sont situées dans les images car cela devrait aboutir à des images infiniment grandes. Même lorsque ce n'est pas le cas, l'image peut devenir très grande (c'est-à-dire si l'épipole est proche de l'image).

Roy et al. [27] ont proposé une méthode pour éviter ce problème, mais leur approche est relativement complexe et présente quelques problèmes. Récemment Pollefeys et al [28] ont proposé une méthode simple qui garantit une taille d'image minimale et fonctionne pour toutes les configurations possibles. Cette méthode sera présentée en détail plus loin, mais d'abord la rectification planaire standard est brièvement discutée.

6.1.1 Rectification planaire

L'approche de rectification standard est relativement simple. Il consiste à sélectionner un plan parallèle à la ligne de base. Les deux images sont ensuite reprojetées dans ce plan. Ceci est illustré sur la figure 2.21. Ces nouvelles images satisfont la configuration stéréo standard.

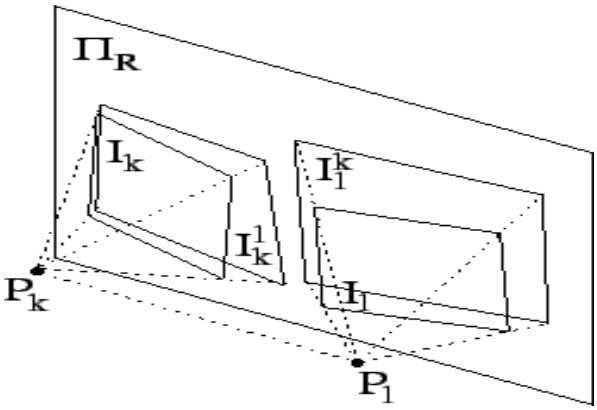


Figure 2.21: Rectification planaire :

Les différentes méthodes de rectification diffèrent principalement dans la façon dont les degrés de liberté restants sont choisis. Dans le cas calibré, on peut choisir la distance du plan à la ligne de base de sorte qu'aucun pixel ne soit comprimé pendant le gauchissement des images vers les images rectifiées et que la normale sur le plan puisse être choisie au milieu des deux plans épipolaires contenant les axes optiques. Dans le cas non calibré, le choix est moins évident. Plusieurs approches ont été proposées.

6.1.2 Rectification polaire

Nous présentons ici un algorithme simple de rectification pouvant traiter toutes les géométries de caméra possibles. Seule la matrice fondamentale orientée est requise. Toutes les transformations sont effectuées dans les images. La taille de l'image est aussi petite que possible sans compression des parties des images. Ceci est obtenu en préservant la longueur des lignes épipolaires et en déterminant la largeur indépendamment pour chaque ligne demi-épipolaire.

Pour les applications stéréo traditionnelles, les limitations des algorithmes de rectification standard ne sont pas si importantes. Le composant principal du déplacement de la caméra est parallèle aux images pour les configurations stéréo classiques. La vergence limitée garde les épipoles loin des images. De nouvelles approches dans la structure et le mouvement non calibrés présentées dans ce texte permettent cependant de récupérer des modèles 3D de scènes acquises avec des caméras portatives. Dans ce cas, le mouvement vers l'avant ne peut plus être exclu. Surtout quand une rue ou un genre de scène similaire est considéré.

6.2 Correspondance stéréo

Le couplage stéréo est un problème qui a été étudié pendant plusieurs décennies en vision par ordinateur et de nombreux chercheurs ont travaillé à le résoudre. Les approches proposées peuvent être globalement classées en approches basées sur les caractéristiques et les corrélations [29]. Marr et Poggio ont proposé des approches basées sur des caractéristiques importantes [30],

(toutes les méthodes basées sur la relaxation), Gimmel'Farb [31] (en utilisant la programmation dynamique).

Des approches basées sur des corrélations réussies ont par exemple été proposées par Cox et al. [32]. Ce dernier a été raffiné par Falkenhagen [33]. C'est ce dernier algorithme qui sera présenté dans cette section.

6.2.1 Exploiter les contraintes de scène

La contrainte épipolaire restreint la plage de recherche pour un point correspondant $\boldsymbol{m}_k$ dans une image à la ligne épipolaire dans l'autre image. Il n'impose aucune restriction sur la géométrie de l'objet autre que le point d'objet reconstruit $\boldsymbol{M}$ repose sur la ligne de vue $\boldsymbol{L_k}$ du centre de projection de $\boldsymbol{P_k}$ et à travers le point correspondant $\boldsymbol{m_k}$ comme vu dans la figure 2.28 (gauche). La recherche du point correspondant $\boldsymbol{m_l}$ est limité à la ligne épipolaire mais aucune restriction n'est imposée le long de la ligne de recherche.

Si nous considérons maintenant la contrainte épipolaire comme étant un plan enjambé par la ligne de visée $\boldsymbol{L_k}$ et la ligne de base reliant les centres de projection de la caméra, alors nous allons trouver la ligne épipolaire en coupant le plan de l'image $\boldsymbol{I_l}$ avec ce plan épipolaire.

Ce plan intersecte également le plan de l'image $\boldsymbol{I_k}$ et il coupe un profil 3D hors de la surface des objets de la scène. Le profil projette sur les lignes épipolaires correspondantes $\boldsymbol{I_k}$ et $\boldsymbol{I_l}$ où il forme un ensemble ordonné de correspondances voisines, comme indiqué sur la figure 2.28 (droite).

Pour les surfaces bien agencées, cet ordre est préservé et délivre une contrainte supplémentaire, appelée «contrainte d'ordre». Les contraintes de scène comme celle-ci peuvent être appliquées en faisant de fausses hypothèses sur la géométrie de l'objet. Dans de nombreuses applications réelles, les objets observés seront opaques et composés de surfaces continues par morceaux. Si cette restriction est vérifiée, des contraintes supplémentaires peuvent être imposées à l'estimation de la correspondance. Koschan [34] a énuméré jusqu'à 12 contraintes différentes pour l'estimation de la correspondance en paires stéréo. Parmi eux, les plus importants en dehors de la contrainte épipolaire sont :

1. Contrainte de commande : Pour les surfaces opaques, l'ordre des correspondances voisines sur les lignes épipolaires correspondantes est toujours conservé. Cette commande permet la construction d'un schéma de programmation dynamique qui est utilisé par de nombreux algorithmes d'estimation de disparités denses.

2. Contrainte d' unicité : La correspondance entre deux points correspondants est bidirectionnelle tant qu'il n'y a pas d'occlusion dans l'une des images. Un vecteur de correspondance pointant d'un point d'image à son point correspondant dans l'autre image a

toujours un vecteur inverse correspondant pointant vers l'arrière. Ce test est utilisé pour détecter les valeurs aberrantes et les occlusions.

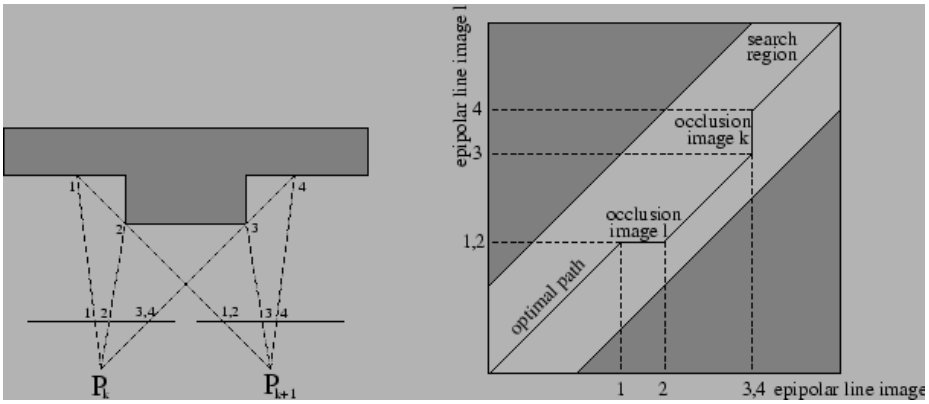


Figure 2.28: Triangle de profil d'objet à partir des correspondances voisines ordonnées (à gauche). Rectification et correspondance entre points de vue ***k*** et ***l*** (droite).

3. Limite de disparité : La bande de recherche est limitée le long de la ligne épipolaire car la scène observée n'a qu'une plage de profondeur limitée (voir la figure 2.28 , à droite).

4. Contrainte de continuité de disparité : Les disparités des correspondances varient le plus souvent de façon continue et les fronts d'étape ne se produisent que sur les discontinuités de surface. Cette contrainte concerne l'hypothèse de surfaces continues par morceaux. Il fournit des moyens pour restreindre davantage la plage de recherche. Pour les pixels d'image voisins le long de la ligne épipolaire, on peut même imposer une limite supérieure au changement de disparité possible. Les changements de disparité au-dessus de la limite indiquent une discontinuité de surface.

6.2.2 Correspondance avec contrainte

Pour une correspondance dense correspondant à un estimateur de disparité basé sur le schéma de programmation dynamique de Cox *et al* [32], est employé qui incorpore les contraintes mentionnées ci-dessus. Il fonctionne sur des paires d'images rectifiées où les lignes épipolaires coïncident avec les lignes de balayage d'image. Le comparateur recherche chaque pixel dans l'image $\boldsymbol{I_k^l}$pour la corrélation croisée normalisée maximale dans $\boldsymbol{I_l^k}$ en déplaçant une petite fenêtre de mesure (taille du noyau 5x5 ou 7x7) le long de la ligne de balayage correspondante. La taille de l'étape de recherche sélectionnée ΔD (généralement 1 pixel) détermine la résolution de recherche et les valeurs de disparité minimum et maximum déterminent la région de recherche. Ceci est illustré sur la figure 2.30.

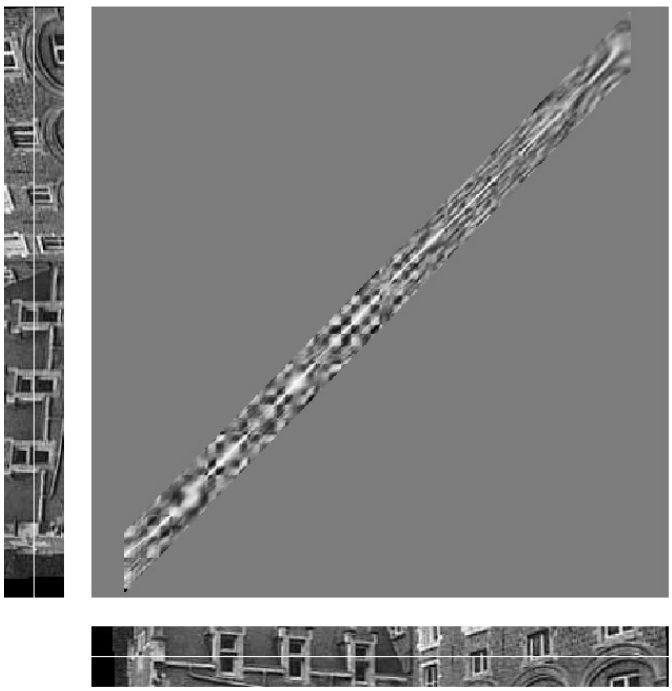


Figure 2.30: Corrélations croisées pour deux lignes épipolaires correspondantes (lumière signifie intercorrélation élevée). Une approche de programmation dynamique est utilisée pour estimer le chemin optimal.

Les ambiguïtés correspondantes sont résolues en exploitant la contrainte d'ordre dans l'approche de programmation dynamique [35].

6.3 Multi-vue stéréo

L'estimation de disparité par paire permet de calculer des correspondances d'image à image entre des paires d'images rectifiées adjacentes, et des estimations de profondeur indépendantes pour chaque point de vue de caméra. Une estimation conjointe optimale sera obtenue en fusionnant toutes les estimations indépendantes dans un modèle 3D commun. La fusion peut être effectuée de manière économique par liaison de correspondance contrôlée comme décrit dans cette section. L'approche utilise un schéma de points de vue multiples flexible en combinant les avantages de la petite ligne de base et de la large ligne de base stéréo.

En tant que **petite stéréo de base,** nous définissons des points de vue où la ligne de base est beaucoup plus petite que la profondeur de scène moyenne observée. Cette configuration est généralement valable pour les séquences d'images où les images sont prises comme une séquence spatiale à partir de nombreux points de vue légèrement différents. Les avantages (+) et les inconvénients (-) sont

+ estimation de correspondance facile, puisque les vues sont similaires,

+ petites régions d'occlusions liées au point de vue,

- petit angle de triangulation, d'où une grande incertitude de profondeur.

La **stéréo de base large** en contraste est utilisée principalement avec des photos d'une scène où quelques images sont prises d'un point de vue très différent. Ici, la résolution de profondeur est supérieure mais des problèmes de correspondance et d'occlusion apparaissent:

- l'estimation de correspondance dure, puisque les vues ne sont pas similaires,

- grandes régions d'occlusions liées au point de vue,

+ grand angle de triangulation, donc grande précision de profondeur.

Le **lien multi-points de vue** combine les vertus des deux approches. De plus, il produira des cartes de profondeur plus denses que n'importe laquelle des autres techniques, et permet des fonctions supplémentaires pour la fusion de profondeur et de texture. Les avantages sont:

+ cartes de profondeur très denses pour chaque point de vue,

+ pas d'occlusions dépendantes du point de vue,

+ résolution maximale en profondeur grâce à la fusion des points de vue,

+ amélioration de texture (texture moyenne, suppression de surbrillance, texture super-résolution).

Algorithme de liaison de correspondance

Le lien de correspondance est décrit dans cette section. Il concatène les points d'image correspondants sur plusieurs points de vue par un suivi de correspondance sur des paires d'images adjacentes. Cela implique bien entendu que les appariements de paires mesurés individuellement sont précis. Pour tenir compte des valeurs aberrantes dans les correspondances par paires, certaines stratégies de contrôle robustes doivent être utilisées pour vérifier la validité de la liaison de correspondance. Considérons une séquence d'images tirée de **k=[1,N]** points de vue. Supposons que la séquence est prise par une caméra se déplaçant latéralement tout en gardant l'objet en vue. Pour tout point de vue ***k*** considérons l'image triple **[$\mathbf{I_{k-1}}$, $\mathbf{I_k}$, $\mathbf{I_{k+1}}$].** Les paires d'images ($\mathbf{I_{k-1}}$, $\mathbf{I_k}$) et ($\mathbf{I_k}$, $\mathbf{I_{k+1}}$) forment deux paires d'images stéréoscopiques avec des estimations de correspondance comme décrit ci-dessus. Nous avons maintenant défini 3 représentations de matrices d'images et de caméras pour chaque point de vue: l'image originale $\mathbf{I_k}$ et matrice de projection $\mathbf{P_k}$, leurs versions transformées $\mathbf{I_k^{k-1}}, \mathbf{P_k^{k-1}}$ rectifié vers le point de vue ***k-1*** avec transformation $\mathbf{R_k^{k-1}}$ et le transformé $\mathbf{I_k^{k+1}}, \mathbf{P_k^{k+1}}$ rectifié vers le point de vue ***k+1*** avec cartographie $\mathbf{R_k^{k+1}}$. La carte de disparité ***D(k,k-1)*** détient les correspondances à la baisse de $\mathbf{I_k^{k-1}}$ à $\mathbf{I_{k-1}^k}$ alors que la carte ***D(k,k+1)*** contient les correspondances ascendantes de $\mathbf{I_k^{k+1}}$ à $\mathbf{I_{k+1}^k}$. Nous pouvons maintenant créer deux chaînes de liens de correspondance pour un point d'image $\boldsymbol{m_k}$, un en haut et en bas de l'index d'image ***k***.

Liens vers le haut : $\mathbf{m}_{k+1} = (\mathbf{R}_{k+1}^{k})^{-1} D_{(k,k+1)}[\mathbf{R}_{k}^{k+1}\mathbf{m}_{k}]$

Liaison vers le bas : $\mathbf{m}_{k-1} = (\mathbf{R}_{k-1}^{k})^{-1} D_{(k,k-1)}[\mathbf{R}_{k}^{k-1}\mathbf{m}_{k}]$

Ce processus de liaison est répété le long de la séquence d'images pour créer une chaîne de correspondances vers le haut et vers le bas. Chaque lien de correspondance nécessite 2 mappages et 1 recherche de disparité. Tout au long de la séquence de N images, 2(N-1) les cartes de disparité sont calculées. La liaison à plusieurs points de vue est ensuite effectuée efficacement via des fonctions de recherche rapide sur les estimations précalculées.

7. Conclusion :

Nous avons vu dans ce chapitre les différentes étapes menants à la construction d'objets 3D à partir d'images en commençant par l'extraction des points d'intérêts jusqu'à obtention du modèle finale tout en essayant d'énumérer les différentes approches déjà existantes.

CHAPITRE III:

MODELISATION, RENDU ET INTEGRATION VOLUMIQUE DU MODELE 3D

1. Introduction :

Dans le chapitre précédent, nous avons vu comment les informations nécessaires pour construire un modèle 3D pouvaient être obtenues automatiquement à partir d'images. Ce chapitre explique comment combiner ces informations pour construire des représentations réalistes de la scène. Il n'est pas seulement possible de générer un modèle de surface ou un modèle volumique facilement, mais toutes les informations nécessaires sont disponibles pour construire des modèles de champs lumineux ou même des séquences vidéo augmentées. Ces différents cas seront maintenant discutés plus en détail.

2. La modélisation

2.1 Modèle de surface

La surface 3D est approximée par un maillage triangulaire pour réduire la complexité géométrique et adapter le modèle aux exigences des systèmes de visualisation par ordinateur. Une approche simple consiste à superposer un maillage triangulaire 2D sur le dessus de l'image puis à construire un maillage 3D correspondant en plaçant les sommets des triangles dans l'espace 3D en fonction des valeurs trouvées dans la carte de profondeur. Pour réduire le bruit, il est recommandé de lisser d'abord l'image de profondeur (le noyau peut être choisi de la même taille que les triangles de maille). L'image elle-même peut être utilisée comme carte de texture (les coordonnées de texture sont trivialement obtenues en tant que coordonnées 2D des sommets).

Il peut arriver que pour certains sommets, aucune valeur de profondeur ne soit disponible ou que la confiance soit trop faible (voir la section 6.2.2). Dans ces cas, les triangles correspondants ne sont pas reconstruits. La même chose se produit lorsque des triangles sont placés sur des discontinuités. Ceci est réalisé en sélectionnant un angle maximum entre la normale d'un triangle et la ligne de visée à travers son centre (par exemple 85 degrés).

Cette approche simple fonctionne très bien sur les cartes de profondeur obtenues après la liaison multi-vue. Sur les cartes de profondeur stéréo simples, il est recommandé

d'utiliser une technique plus avancée décrite dans [36]. Dans ce cas, les limites des objets à modéliser sont calculées par segmentation en profondeur. Dans un premier temps, un objet est défini comme une région connectée dans l'espace. Un filtrage morphologique simple élimine les régions parasites et très petites. Ensuite, un modèle de plaque mince bornée est utilisé avec une spline du second ordre pour lisser la surface et interpoler de petits espaces de surface dans des régions qui n'ont pas pu être mesurées.

L'approche de reconstruction de surface est illustrée sur la figure 2.32. Le modèle de surface 3D obtenu est représenté sur la figure 2.33 avec ombrage et avec texture. Notez que ce modèle de surface est reconstruit du point de vue d'une image de référence. Si l'ensemble de la scène ne peut pas être vu d'une image, il est nécessaire d'appliquer une technique pour fusionner différentes surfaces ensemble (par ex. [37]).

Figure 2.32: Approche de reconstruction de surface: Un maillage triangulaire (à gauche) est superposé au-dessus de l'image (au milieu). Les sommets sont rétroprojetés dans l'espace en fonction de la valeur trouvée dans la carte de profondeur (à droite).

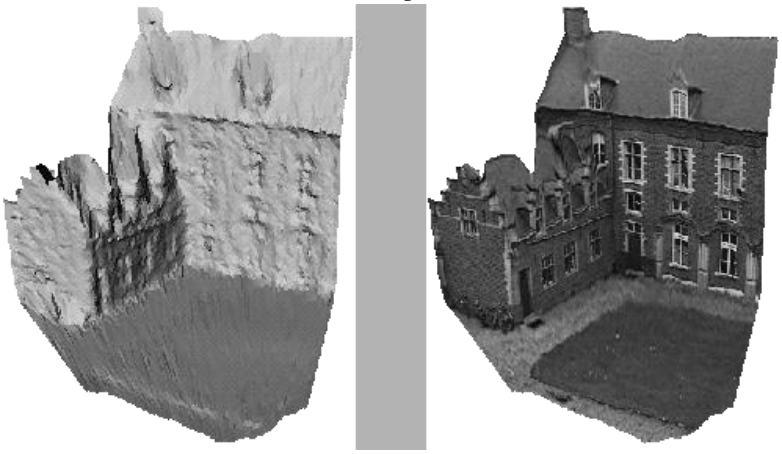


Figure 2.33: *Modèle de surface 3D obtenu automatiquement à partir d'une séquence d'image non étalonnée, ombrée (à gauche), texturée (à droite).*

2.1.1 Amélioration de la texture

Le lien de correspondance construit une chaîne de correspondance contrôlée qui peut également être utilisée pour améliorer la texture. A chaque pixel de référence, on peut collecter une liste triée de valeurs de couleurs d'image à partir des positions

d'image correspondantes. Cela permet d'améliorer la texture originale de plusieurs façons en accédant aux statistiques de couleurs. Certaines fonctionnalités dérivées naturellement de l'algorithme de liaison sont :

2.1.1.1 Suppression des hautes lumières et des reflets

Une moyenne médiane ou robuste des valeurs de texture correspondantes est calculée pour ignorer les artefacts d'imagerie tels que le bruit du capteur, les réflexions spéculaires et les hautes lumières [38]. Un exemple de suppression de surbrillance est représenté sur la figure 2.34.

2.1.1.2 Texture super-résolution

La liaison de correspondance n'est pas limitée à la résolution de pixel, puisque chaque position de sous-pixel dans l'image de référence peut être utilisée pour démarrer une chaîne de correspondance. Les valeurs de correspondance sont interrogées à partir de la carte de disparité par interpolation. L'objet est vu par de nombreuses caméras avec une résolution de pixel limitée, mais chaque grille de pixels d'image sera en général légèrement déplacée. Cela peut être exploité pour créer une texture super-résolution en fusionnant toutes les images sur une grille de rééchantillonnage plus fine [39].

2.1.1.3 Meilleure sélection de vue pour une résolution de texture maximale

Pour chaque région de surface autour d'un pixel, l'image ayant la résolution de texture la plus élevée possible est sélectionnée, en fonction de la distance de l'objet et de l'angle de vue. L'image composite prend en compte la plus haute résolution possible de toutes les images.

Figure 2.34: vue en gros plan (à gauche), région originale zoomée 4x (en haut à droite), génération de texture super-résolution médian-filtrée (en bas à droite).

2.1.2 Intégration volumique

Pour générer un modèle 3D complet à partir de différentes cartes de profondeur, nous proposons d'utiliser l'approche d'intégration volumétrique de Curless et Levoy [40]. Cette approche est décrite dans cette section.

L'algorithme utilise une fonction implicite continue, ***D(M),*** représenté par des échantillons. La fonction que nous représentons est la distance pondérée entre chaque point et la surface de portée la plus proche le long de la ligne de visée du capteur. Nous construisons cette fonction en combinant des fonctions de distance signées, $d_1(M)$, $d_2(M)$ … $d_n(M)$ et les fonctions de poids $w_1(M)$, $w_2(M)$ … $w_n(M)$ obtenu à partir des cartes de profondeur pour les différentes images. Les règles de combinaison donnent une fonction de distance signée cumulée pour chaque voxel, ***D(M)*** et un poids cumulé ***W(M)***. Ces fonctions sont représentées sur une grille de voxel discrète et une isosurface correspondant à $D(x) = 0$ est extrait. Sous un certain ensemble d'hypothèses, cette isosurface est optimale au sens des moindres carrés.

Figure 2.35 illustre le principe de combiner des distances signées non pondérées pour le cas simple de deux surfaces de distance échantillonnées dans la même direction. Notez que l'isosurface résultante serait la surface créée en faisant la moyenne des deux surfaces de gamme le long des lignes de visée du capteur. En général, cependant, les pondérations sont nécessaires pour représenter les variations de certitude sur les surfaces de portée. Le choix des poids devrait être spécifique à la procédure d'estimation de la profondeur. Il est proposé de faire dépendre les poids du produit scalaire entre chaque vertex normal et la direction de visualisation, reflétant une plus grande incertitude lorsque l'observation est à l'angle rasant de la surface, comme Soucy [41] proposé pour les scanners à triangulation optique. Les valeurs de profondeur aux limites d'un maillage ont généralement une plus grande incertitude, nécessitant plus de pondération.

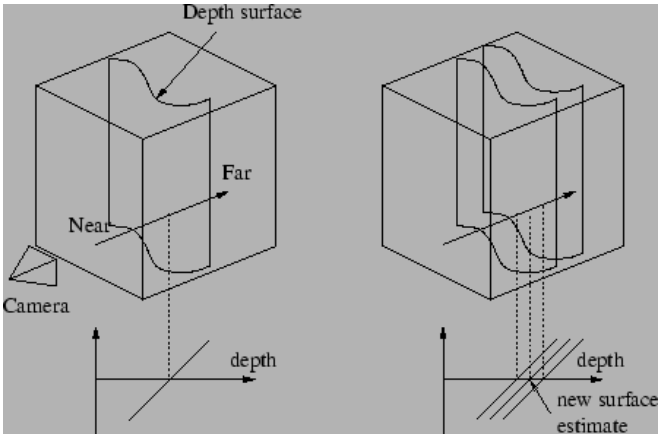


Figure 2.35: Fonctions de distance signée non pondérée en 3D. (a) Une caméra regardant vers le bas de l'axe des x observe une image en profondeur, représentée ici comme une surface reconstruite. En suivant une ligne de visée sur l'axe des x, une fonction de distance signée peut être générée. Le passage par zéro de cette fonction est un point sur la surface. (b) Une autre carte de profondeur donne une surface légèrement différente en raison du bruit. En suivant la même ligne de vue que précédemment, nous obtenons une autre fonction de distance signée. En additionnant ces fonctions, nous arrivons à une fonction cumulative avec un nouveau passage à zéro positionné à mi-chemin entre les mesures de distance d'origine.

Figure 2.36 illustre la construction et l'utilisation des fonctions de distance et de poids signées dans 1D. en 2.36a, le capteur est positionné à l'origine en regardant l'axe + x et a pris deux mesures, r_1 et r_2. Les profils de distance signés, $d_1(x)$ et $d_2(x)$ peut s'étendre indéfiniment dans les deux sens, mais les fonctions de poids, $w_1(x)$ et $w_2(x)$, se rétrécissent derrière les points de gamme pour les raisons discutées ci-dessous.

Figure 2.36b est la combinaison pondérée des deux profils. Les règles de combinaison sont simples:

$$D(x) = \frac{\sum w_i(x) d_i(x)}{\sum w_i(x)} \quad \text{(H1)}$$

$$W(x) = \sum w_i(x) \quad \text{(H2)}$$

où, $d_i(x)$ et $w_i(x)$ sont les fonctions de distance et de poids signées à partir de l'image de plage ith. Exprimé sous forme de calcul incrémental, les règles sont:

$$D_{i+1}(x) = \frac{W_i(x) D_i(x) + w_i(x) d_i(x)}{\sum W_i(x) + w_{i+1}(x)} \quad \text{(H3)}$$

$$W_{i+1}(x) = W_i(x) + w_i(x) \quad \text{(H4)}$$

où $D_i(x)$ et $W_i(x)$ sont les fonctions de distance et de poids signées cumulatives après intégration de l'image de i-ème plage. Dans le cas particulier d'une dimension, le passage par zéro de la fonction cumulative est à une distance, R donnée par:

$$R = \frac{\sum w_i r_i}{\sum w_i} \qquad \text{(H5)}$$

c'est-à-dire une combinaison pondérée des valeurs de gamme acquises, ce qui est ce à quoi on s'attendrait pour une minimisation par les moindres carrés.

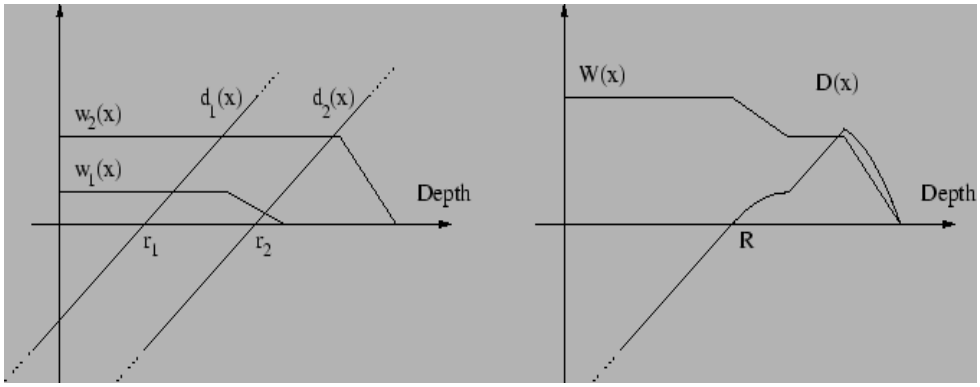


Figure 2.36: Fonctions de distance et de poids signées dans une dimension.

En principe, les fonctions de distance et de pondération doivent s'étendre indéfiniment dans les deux sens. Cependant, pour éviter que les surfaces des côtés opposés de l'objet n'interfèrent les unes avec les autres, nous forçons la fonction de pondération à se rétrécir derrière la surface. Il y a un compromis à faire pour choisir où la fonction de poids se rétrécit. Il devrait persister assez loin derrière la surface pour s'assurer que toutes les rampes de distance contribueront au voisinage du passage à zéro final, mais il devrait aussi être aussi étroit que possible pour éviter d'influencer les surfaces de l'autre côté. Pour répondre à ces exigences, on force les poids à tomber à une distance égale à la moitié de l'intervalle d'incertitude maximum des mesures de profondeur. De même, les fonctions de distance et de poids signées ne doivent pas s'étendre loin devant la surface. Restreindre les fonctions au voisinage de la surface donne une représentation plus compacte et réduit la dépense de calcul de la mise à jour du volume.

Dans les deux et trois dimensions, les mesures de profondeur correspondent à des courbes ou des surfaces avec des fonctions de poids, et les rampes de distance signées ont des directions qui sont cohérentes avec les directions primaires de l'incertitude du capteur.

Pour trois dimensions, nous pouvons résumer l'algorithme entier comme suit. Tout d'abord, nous définissons tous les poids de voxel à zéro, de sorte que les nouvelles données écraseront les valeurs de grille initiales. La contribution de distance signée est

calculée en faisant la différence entre la profondeur lue lors de la projection du point de la grille dans la carte de profondeur et la distance réelle entre le point et le centre de projection de la caméra. Le poids est obtenu à partir d'une carte de poids qui a été précalculée. Après avoir déterminé la distance et le poids signés, nous pouvons appliquer les formules de mise à jour décrites dans les équations (H3) et (H4).

À tout moment de la fusion des images de distance, nous pouvons extraire l'isosurface du passage par zéro à partir de la grille volumétrique. Nous limitons cette procédure d'extraction pour sauter des échantillons avec un poids nul, générant des triangles uniquement dans les régions de données observées. La procédure utilisée pour cela est **Marching cubes** [42].

2.1.2.1 Marching cubes

Marching Cubes est un algorithme de génération d'isosurfaces à partir de données volumétriques. Si un ou plusieurs voxels d'un cube ont des valeurs inférieures à l'isovalue ciblée, et qu'un ou plusieurs ont des valeurs supérieures à cette valeur, nous savons que le voxel doit contribuer à un composant de l'isosurface. En déterminant quelles arêtes du cube sont intersectées par l'isosurface, on peut créer des patches triangulaires qui divisent le cube entre les régions à l'intérieur de l'isosurface et les régions à l'extérieur. En connectant les patchs de tous les cubes sur la limite d'isosurface, une représentation de surface est obtenue.

Il y a deux composants majeurs de cet algorithme. Le premier est de décider comment définir la ou les sections de surface qui découpent un cube individuel. Si nous classons chaque coin comme étant soit au-dessous ou au-dessus de l'isovalue, il y a 256 configurations possibles de classifications d'angle. Deux d'entre eux sont triviaux ; où tous les points sont à l'intérieur ou à l'extérieur du cube ne contribue pas à l'isosurface. Pour toutes les autres configurations, nous devons déterminer où, le long de chaque bord du cube, l'isosurface croise et utiliser ces points d'intersection de bord pour créer un ou plusieurs patchs triangulaires pour l'isosurface.

Si vous tenez compte des symétries, il n'y a vraiment que 14 configurations uniques dans les 254 possibilités restantes. Quand il n'y a qu'un angle inférieur à l'isovalue, cela forme un seul triangle qui croise les arêtes qui se rencontrent à cet angle, avec la normale perpendiculaire à l'angle. De toute évidence, il existe 8 configurations connexes de ce type. En inversant la normale, nous obtenons 8 configurations qui ont 7 coins de moins que l'isovalue. Cependant, nous ne les considérons pas comme

vraiment uniques. Pour les configurations avec 2 coins de moins que l'isovalue, il y a 3 configurations uniques, selon que les coins appartiennent au même bord, appartiennent à la même face du cube, ou sont positionnés diagonalement l'un par rapport à l'autre. Pour les configurations avec 3 coins de moins que l'isovaleur, il y a encore 3 configurations uniques, selon qu'il y a 0, 1 ou 2 bords partagés (2 bords partagés vous donne une forme en "L"). Il y a 7 configurations uniques quand vous avez 4 coins de moins que l'isovalue, selon qu'il y a 0, 2, 3 (3 variantes sur celui-ci), ou 4 bords partagés. Les différents cas sont illustrés sur la figure 2.37.

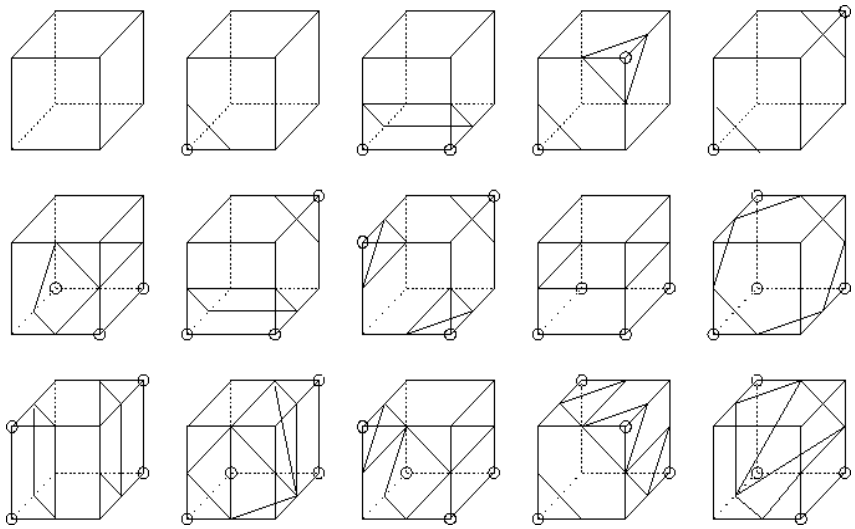


Figure 2.37: Les 14 configurations différentes pour les cubes de marche.

Chacune des configurations non triviales entraîne l'ajout de 1 à 4 triangles à l'isosurface. Les sommets eux-mêmes peuvent être calculés par interpolation le long des bords.

Maintenant que nous pouvons créer des correctifs de surface pour un seul voxel, nous pouvons appliquer ce processus à l'ensemble du volume. Nous pouvons traiter le volume dans les dalles, où chaque dalle est composée de 2 tranches de pixels. Nous pouvons traiter chaque cube indépendamment, ou nous pouvons proposer des intersections entre les cubes qui partagent les bords. Ce partage peut également se faire entre des dalles adjacentes, ce qui augmente un peu le stockage et la complexité, mais permet d'économiser du temps de calcul. Le partage des informations de bord / sommet produit également un modèle plus compact, et plus adapté à l'ombrage interpolé.

2.2 Modèle Lightfield

Dans cette section, notre objectif est de créer un modèle lightfield à partir d'une scène pour rendre de nouvelles vues de manière interactive. Notre approche a été présentée dans un certain nombre de documents consécutifs [43, 44]. Pour rendre de

nouvelles vues, deux concepts majeurs sont connus dans la littérature. Le premier est le concept basé sur la géométrie. La géométrie de la scène est reconstruite à partir d'un flux d'images et une texture unique est synthétisée mappée sur cette géométrie. Pour cette approche, un ensemble limité de vues de caméra est suffisant, mais les effets spéculaires ne peuvent pas être gérés de manière appropriée. Cette approche a été largement discutée dans ce texte.

Le deuxième concept majeur est le rendu basé sur l'image. Cette approche modélise la scène comme une collection de vues tout autour de la scène sans représentation géométrique exacte [45]. Les nouvelles vues (virtuelles) sont rendues à partir des vues enregistrées par interpolation en temps réel. Des informations géométriques approximatives peuvent être utilisées pour améliorer les résultats [46]. Nous nous concentrons ici sur cette deuxième approche. Jusqu'à présent, la représentation de la scène connue a une structure régulière fixe. Si la source est un flux d'images prises avec une caméra portative, cette structure régulière doit être rééchantillonnée.

Notre but est d'utiliser les images enregistrées elles-mêmes comme représentation de scène et de rendre directement de nouvelles vues à partir d'elles. Les informations géométriques sont considérées dans la mesure où elles sont connues et aussi détaillées que le temps de rendu le permet. L'approche est conçue de telle sorte que les opérations consistent uniquement en des cartographies projectives pouvant être efficacement exécutées par le matériel graphique.

Pour chacune de ces techniques de modélisation de scène, les paramètres de la caméra pour les vues originales sont censés être connus. Nous les récupérons en appliquant les techniques connues de structure et de mouvement décrites dans les chapitres précédents. Les cartes de profondeur locales sont calculées en appliquant des techniques stéréo sur des paires d'images rectifiées comme expliqué précédemment.

2.2.1 Structure et mouvement

Pour faire une modélisation de lightfield dense comme décrit ci-dessous, nous avons besoin de beaucoup de vues d'une scène de plusieurs directions. Pour cela, nous pouvons enregistrer une séquence d'images étendue en déplaçant la caméra de manière zigzag. La caméra peut traverser son propre chemin de déplacement plusieurs fois ou au moins s'en approche. Les méthodes de calibration connues ne considèrent généralement que les voisinages du flux d'images.

Généralement, aucune liaison n'est effectuée entre des vues dont la position est proche l'une de l'autre dans l'espace 3D mais qui ont une grande distance dans la séquence. Pour résoudre ce problème, nous exploitons donc la topologie 2D des points de vue de la caméra pour stabiliser davantage la Calibration. Nous traitons non seulement l'image séquentielle suivante, mais recherchons également les images les plus proches dans la topologie du point de vue actuel. Typiquement, nous pouvons établir une correspondance fiable à 3-4 images voisines, ce qui améliore considérablement la Calibration.

Les détails ont été décrits précédemment. Nous montrerons également comment utiliser les cartes de profondeur locales pour améliorer les résultats de rendu. A cette fin, des cartes de correspondance dense sont calculées pour des paires d'images adjacentes de la séquence.

2.2.2 Modélisation et rendu de Lightfield

Dans [45] l'apparition d'une scène est décrite à travers tous les rayons lumineux (2D) qui sont émis à partir de chaque point de la scène 3D, générant une fonction de radiance 5D. Par la suite deux réalisations équivalentes de la fonction plénière ont été proposées sous forme de champ lumineux [45], et le lumigraph [46]. Ils manipulent le cas lorsque l'observateur et la scène peuvent être séparés par une surface. D'où la fonction plénière est réduite à quatre dimensions. Le rayonnement est représenté en fonction des rayons lumineux qui traversent la surface de séparation.

Pour créer un tel modèle plenoptique pour des scènes réelles, un grand nombre de vues est pris. Ces vues peuvent être considérées comme une collection de rayons lumineux avec des valeurs de couleurs appropriées. Ce sont des échantillons discrets de la fonction plénière. Les rayons lumineux qui ne sont pas représentés doivent être interpolés à partir des signaux enregistrés en tenant compte des informations supplémentaires sur les restrictions physiques. Souvent, les objets réels sont supposés être lambertiens, ce qui signifie qu'un point de l'objet a la même valeur de rayonnement dans toutes les directions possibles.

Ceci implique que deux rayons de visualisation ont la même valeur de couleur s'ils s'intersectent en un point de surface. Si des effets spéculaires se produisent, ce n'est plus vrai. Deux rayons de visualisation ont alors des valeurs de couleur similaires si leur direction est similaire et si leur point d'intersection est proche du point de scène réel qui est à l'origine de leur valeur de couleur. Pour rendre une nouvelle vue, nous

supposons avoir une caméra virtuelle qui regarde la scène. Nous déterminons les rayons les plus proches de ceux de cette caméra. Plus un rayon est proche d'un rayon donné, plus son support à la valeur de couleur est grand.

Le champ lumineux original 4D [45] la structure de données utilise un paramétrage à deux plans. Chaque rayon lumineux traverse deux plans parallèles avec des coordonnées planes (s,t) et (u,v). le (u,v)-plan est le *plan* du *point de vue* dans lequel tous les points focaux de la caméra sont placés sur des points de grille réguliers. Le (s,t)-plan est le *plan focal* .

De nouvelles vues peuvent être rendues en croisant chaque rayon d'une caméra virtuelle avec les deux plans à (s,t,u,v). L'éclat résultant est une recherche dans la grille régulière. Pour les rayons qui passent entre les (s,t) et (u,v) coordonnées de grille une interpolation est appliquée qui dégrade la qualité de rendu en fonction de la géométrie de la scène. En fait, le champ de lumière contient une hypothèse géométrique implicite, c'est-à-dire que la géométrie de la scène est plane et coïncide avec le plan focal.

La déviation de la géométrie de la scène par rapport au plan focal entraîne une dégradation de l'image (c'est-à-dire un flou ou une image fantôme). Pour utiliser des images de caméra portatives, la solution proposée dans [46] consiste à *réintégrer* les images dans la grille régulière. L'inconvénient de cette étape de rebasement est que la structure régulière interpolée contient déjà des incohérences et des artefacts fantômes en raison d'erreurs dans la géométrie approximative. Pendant le rendu, l'effet des artefacts fantômes est répété, ce qui produit des effets fantômes en double.

2.2.2.1 Rendu à partir d'images enregistrées

Notre but est de surmonter les problèmes décrits dans la dernière section en relâchant les restrictions imposées par la structure régulière du champ de lumière et de rendre les vues directement à partir de la séquence calibrée d'images enregistrées en utilisant des cartes de profondeur locales. Sans perte de performance, les images originales sont directement mappées sur un ou plusieurs plans visualisés par une caméra virtuelle.

Pour obtenir une représentation de scène basée sur l'image de haute qualité, nous avons besoin de nombreuses vues provenant d'une scène de plusieurs directions. Pour cela, nous pouvons enregistrer une séquence d'images étendue en déplaçant la caméra de manière zigzag. La caméra peut traverser son propre chemin de déplacement plusieurs fois ou au moins s'en approche. Pour obtenir une bonne estimation de la

structure et du mouvement de ce type de séquence, il est important d'utiliser les extensions proposées dans la section 4.2 pour faire correspondre des vues proches qui ne sont pas des prédécesseurs ou des successeurs dans le flux d'images. Pour permettre de construire les cartes de profondeur d'approximation géométrique locales, il faut également calculer comme décrit dans la section précédente.

2.2.2.2 Approximation de plan fixe

Dans une première approche, nous approximons la géométrie de la scène par un seul plan en minimisant l'erreur des moindres carrés. Nous cartographions toutes les images de caméra données sur un plan et voir à travers une caméra virtuelle. Cela peut être réalisé en cartographiant directement les coordonnées x_i , y_i de l'image i sur les coordonnées de la caméra virtuelle $[x_V\, y_V\, 1]^\top = \mathbf{H}_{iV}[x_i\, y_i\, 1]^\top$. Par conséquent, nous pouvons effectuer une recherche directe dans les images enregistrées à l'origine et déterminer la radiance en interpolant les valeurs de pixels voisins enregistrés. Cette technique est similaire à l'approche lightfield [45] qui assume implicitement le plan focal comme plan de géométrie. Ainsi, pour construire une vue spécifique, nous devons interpoler entre les vues voisines. Ces vues donnent le plus de soutien à la valeur de couleur d'un pixel particulier dont le centre de projection est proche du rayon de vision de ce pixel. Cela équivaut au fait que les vues dont les centres de caméra projetés sont proches de la coordonnée de l'image donnent le plus de soutien à un pixel spécifié. Nous limitons le support aux trois caméras les plus proches (voir Figure 2.38).

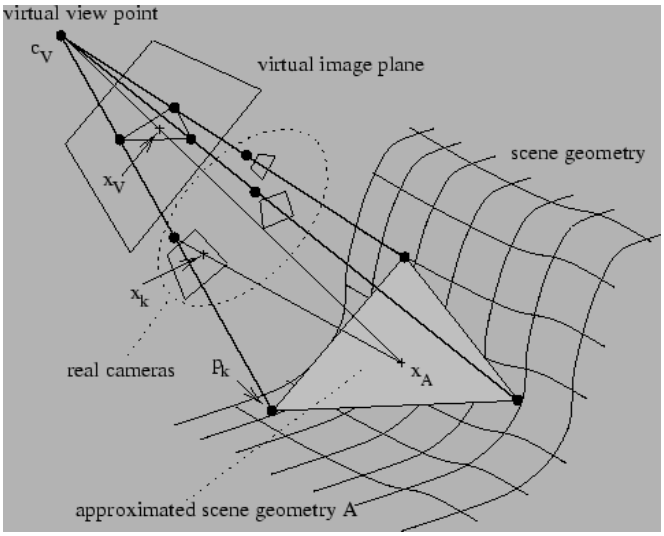


Figure 2.38: Dessin des triangles des centres de la caméra projetés voisins et approximation de la géométrie par un plan pour l'ensemble de la scène, pour une caméra triple ou par plusieurs plans pour une caméra triple.

Nous projetons tous les centres de la caméra dans l'image virtuelle et effectuons une triangulation 2D. Ensuite, les caméras voisines d'un pixel sont déterminées par les coins du triangle auquel ce pixel appartient. Chaque triangle est dessiné comme une somme de trois triangles. Pour chaque caméra, nous recherchons les valeurs de couleurs dans l'image originale comme décrit ci-dessus et les multiplions par le poids 1 au sommet correspondant et par le poids 0 aux deux autres sommets. Entre les deux, les poids sont interpolés de manière linéaire similaire à l'ombrage Gouraud. Dans le triangle, la somme des poids est de 1 à chaque point. L'image totale est construite comme une mosaïque de ces triangles. Bien que cette technique suppose une approximation très approximative de la géométrie, les résultats de rendu ne montrent que de petits artefacts fantômes.

2.2.2.3 Approximation de la géométrie dépendante de la vue

Les résultats peuvent être améliorés en considérant les cartes de profondeur locales. En dépensant plus de temps pour chaque vue, nous pouvons calculer le plan d'approximation de la géométrie pour chaque triangle en fonction de la vue réelle. Cela améliore encore la précision car l'approximation n'est pas faite pour toute la scène mais seulement pour la partie de l'image qui est vue à travers le triangle réel. Les valeurs de profondeur sont données en tant que fonctions D_i des coordonnées dans les images enregistrées D_i (x,y). Ils décrivent la distance d'un point au centre de projection. En utilisant cette fonction de profondeur, nous calculons les coordonnées 3D de ces points de scène qui ont les mêmes coordonnées d'image 2D dans la vue virtuelle que les centres de caméra projetés des vues réelles. Le point 3D $\mathbf{M_i}$ ce qui correspond à $\boldsymbol{i}$ vues peut être calculé comme

$$\mathbf{M}_i = sD_i(\mathbf{P}_k\mathbf{C}_V)n(\mathbf{C}_k - \mathbf{C}_V) + \mathbf{C}_k \qquad \text{(H6)}$$

où $n(\mathbf{A}) = \frac{\mathbf{A}}{\|\mathbf{A}\|}$ et $s = \text{sign}\,(\mathbf{P}_{3i}.(\mathbf{C}_k - \mathbf{C}_V))$ avec $\boldsymbol{P_{3i}}$ la troisième rangée de $\boldsymbol{P_i}$ est nécessaire pour une orientation correcte. Nous pouvons interpréter les points $\boldsymbol{M_i}$ comme l'intersection de la ligne $\overline{\boldsymbol{CvCk}}$ avec la géométrie de la scène. Connaissant les coordonnées 3D des coins du triangle, nous pouvons définir un plan à travers eux et appliquer la même technique de rendu que celle décrite ci-dessus.

Enfin, si les triangles dépassent une taille donnée, ils peuvent être subdivisés en quatre sous-triangles en divisant les trois côtés en deux parties, chacune. Pour chacun de ces sous-triangles, un plan approximatif distinct est calculé de la manière ci-dessus.

Nous déterminons le point médian du côté et utilisons la même méthode de recherche que celle utilisée pour les valeurs de luminance pour trouver la profondeur correspondante. Après cela, nous reconstruisons le point 3D et le projetons dans la caméra virtuelle résultant en un point près du côté du triangle.

CHAPITRE IV :

Conception d'une application de reconstruction d'environnement 3D pour la réalité augmentée.

1. Introduction et objectifs :

Dans ce chapitre nous allons détailler la démarche qui va nous aider a concevoir notre system en partant de tous ce qui a été montré au cours des deux chapitres précédents, le but étant de fusionner les 2 concepts en une seule application mobile.

Notre objectif est de concevoir une application mobile de réalité augmentée avec la possibilité de créer de nouveaux objets 3D à partir d'images prises par la caméra du mobile.

Pour cela, nous allons évoquer dans ce chapitre la Conception d'une application de reconstruction d'environnement 3D pour la réalité augmentée.

2. Motivation :

Comme réponse a la question évoquée dans l'introduction générale on veut offrir à l'utilisateur ordinaire ne possédant pas de connaissances préalables en modélisation 3D la possibilité de créer sa propre bibliothèque d'objets 3D et de pouvoir les visualiser et manipuler juste en utilisant intuitivement son mobile.

L'application proposée se décompose en deux parties, la première est dédiée à création d'objets 3D à partir d'images et la seconde partie concerne la visualisation et manipulation d'objets en réalité augmentée.

3. Conception globale de l'application

De par leur similitude de fonctionnement, les applications AR mobile peuvent avoir un schéma conceptuel généralisé comme celui proposé dans [1] illustré dans la figure suivante :

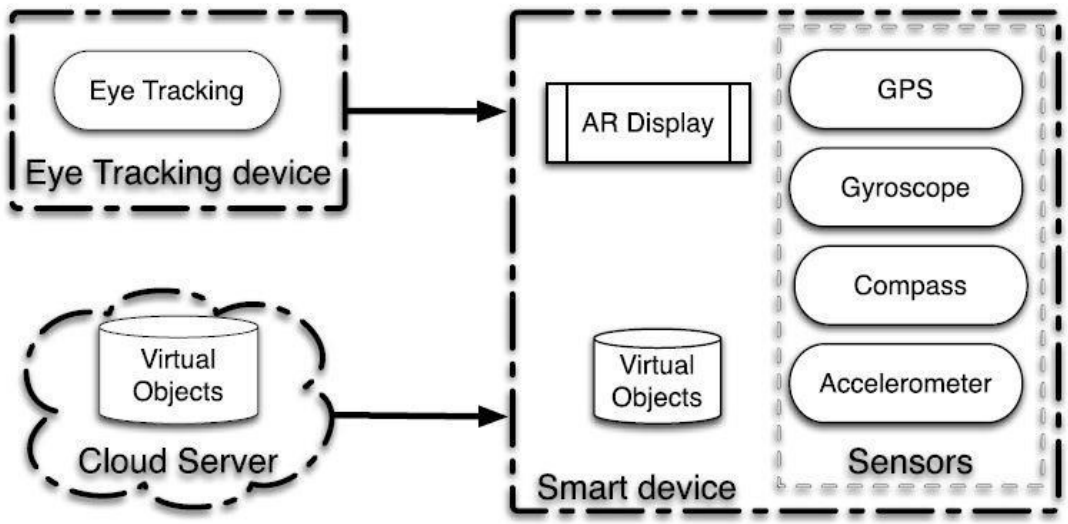


Figure 3.1: schéma conceptuel généralisé d'une application AR mobile

A noter que les 2 composants (eye tracking et cloud server) sont optionnels et ne font pas partie intégrantes de toutes les applications MAR (mobile augmented reality).

A partir de cela et en prenant en compte les objectifs fixés pour notre application, nous avons proposé le schéma conceptuel comme illustré sur la figure suivante :

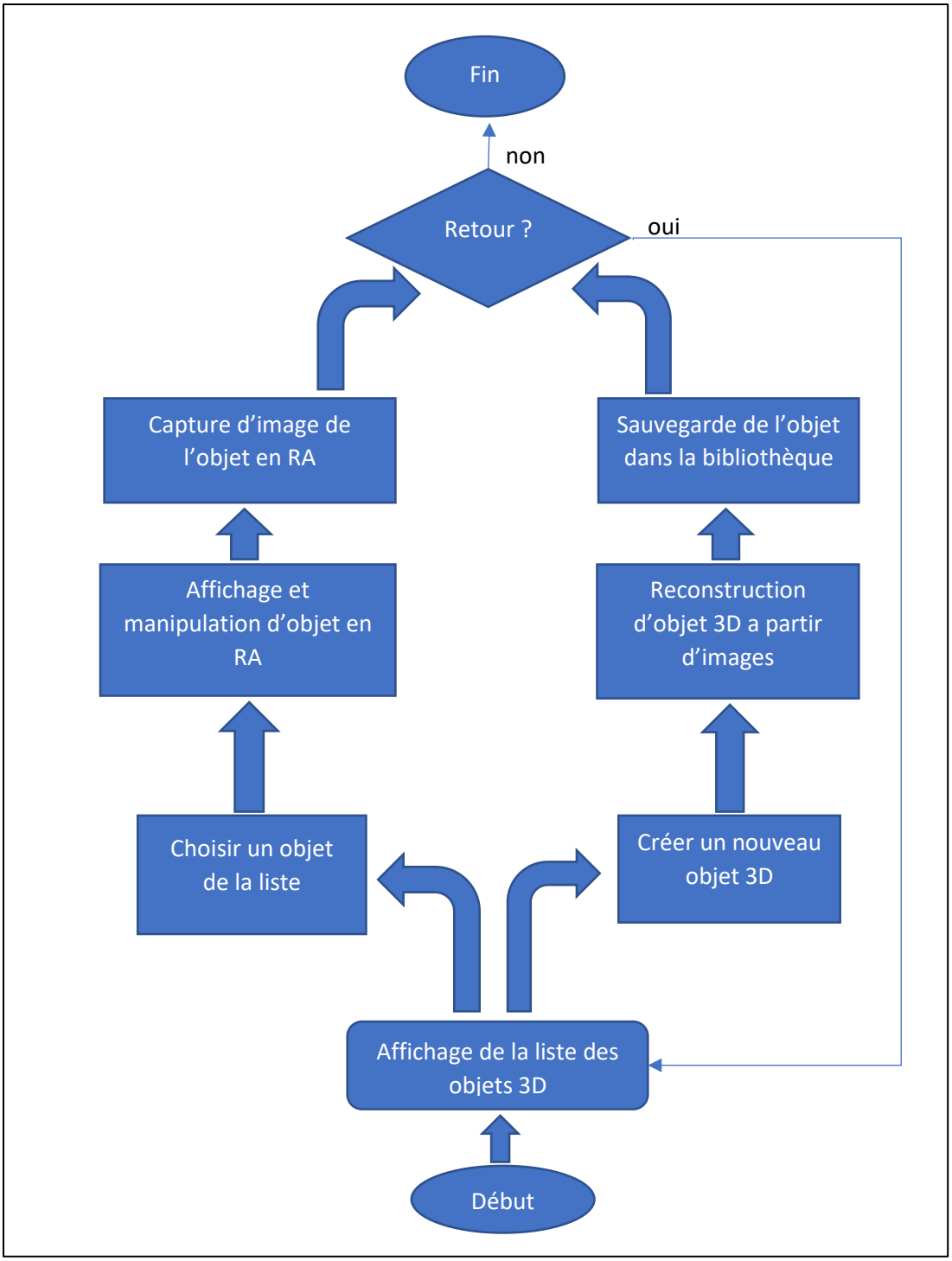


Figure 3.2 : schéma conceptuel généralisé de notre application

Notre application étant divisée en deux parties principales ajoutés a la bibliothèque d'objets 3D nous allons proposer leur schémas conceptuel comme suit :

Partie 1 : visualisation et manipulation d'objets en AR

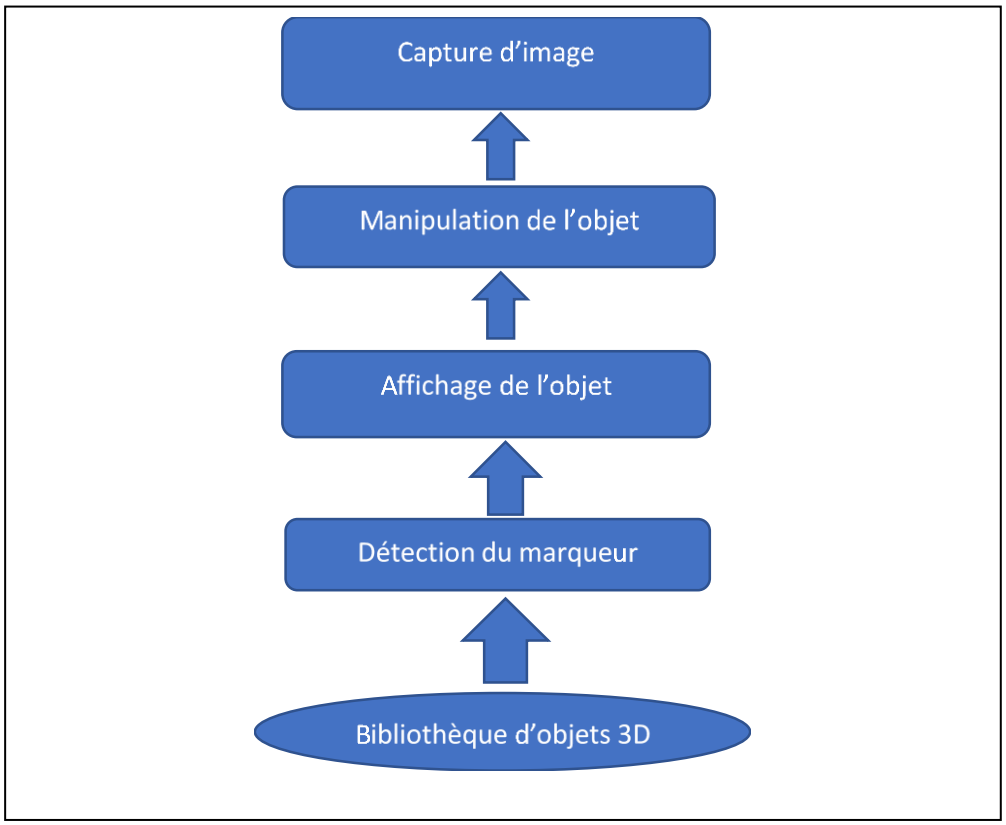


Figure 3.3 : schéma conceptuel généralisé de partie visualisation en AR

Partie 2 : création d'objets 3D à partir d'images

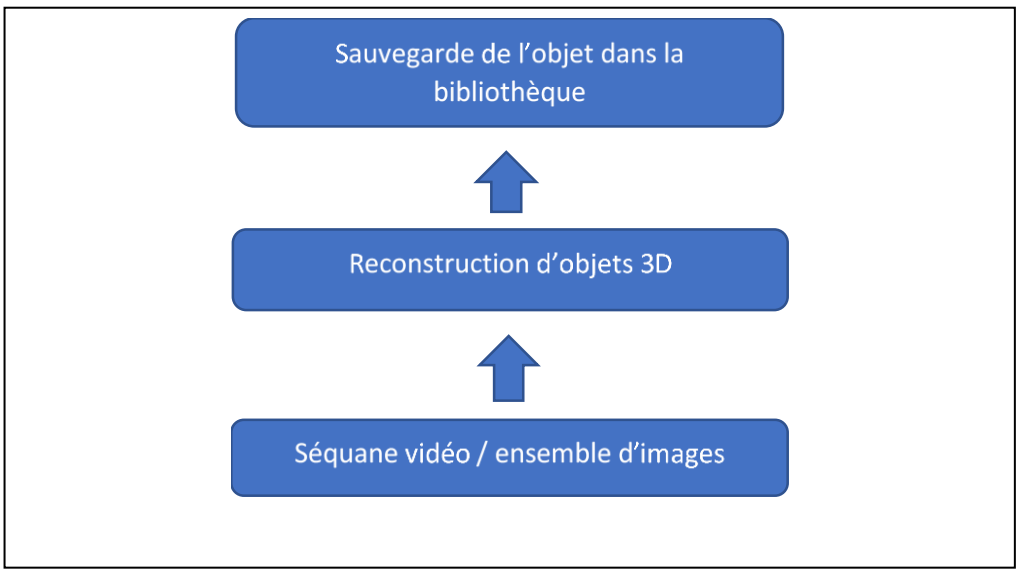


Figure 3.4 : schéma conceptuel généralisé de la partie création d'objets 3D

Nous allons maintenant détailler les différentes étapes de fonctionnement de notre système.

4. Conception détaillée du system

5.1 Partie 1 réalité augmentée :

Pour la partie réalité augmentée on a adopté une architecture décrite dans [2] qui décompose l'application en 5 sous-systèmes définis comme suit :

Sous-système application : qui définis la partie du code spécifique a l'application selon son objectif, en plus des données de configuration, données utilisateur …etc. Peut être considéré comme le noyau même de l'application.

Sous-système de suivi : qui consiste en la partie consacré a la gestion des capteurs comme le GPS, l'accéléromètre …etc.

Sous-système d'entrée utilisateur : définis la partie consacrée a l'écoute des actions de l'utilisateur comme le geste, le mouvement, la voix …etc.

Sous-système de sortie utilisateur : consiste en gros en l'interface homme-machine donc tous ce que voit l'utilisateur de l'application.

Sous-système de contexte : joue le rôle d'intermédiaire entre les autres sous-systèmes en gérant la diffusion de données selon le besoin de chacun.

La figure suivante illustre l'architecture générale de la partie réalité augmentée avec les interactions entre les différents sous-systèmes comme définis dans l'article cité précédemment :

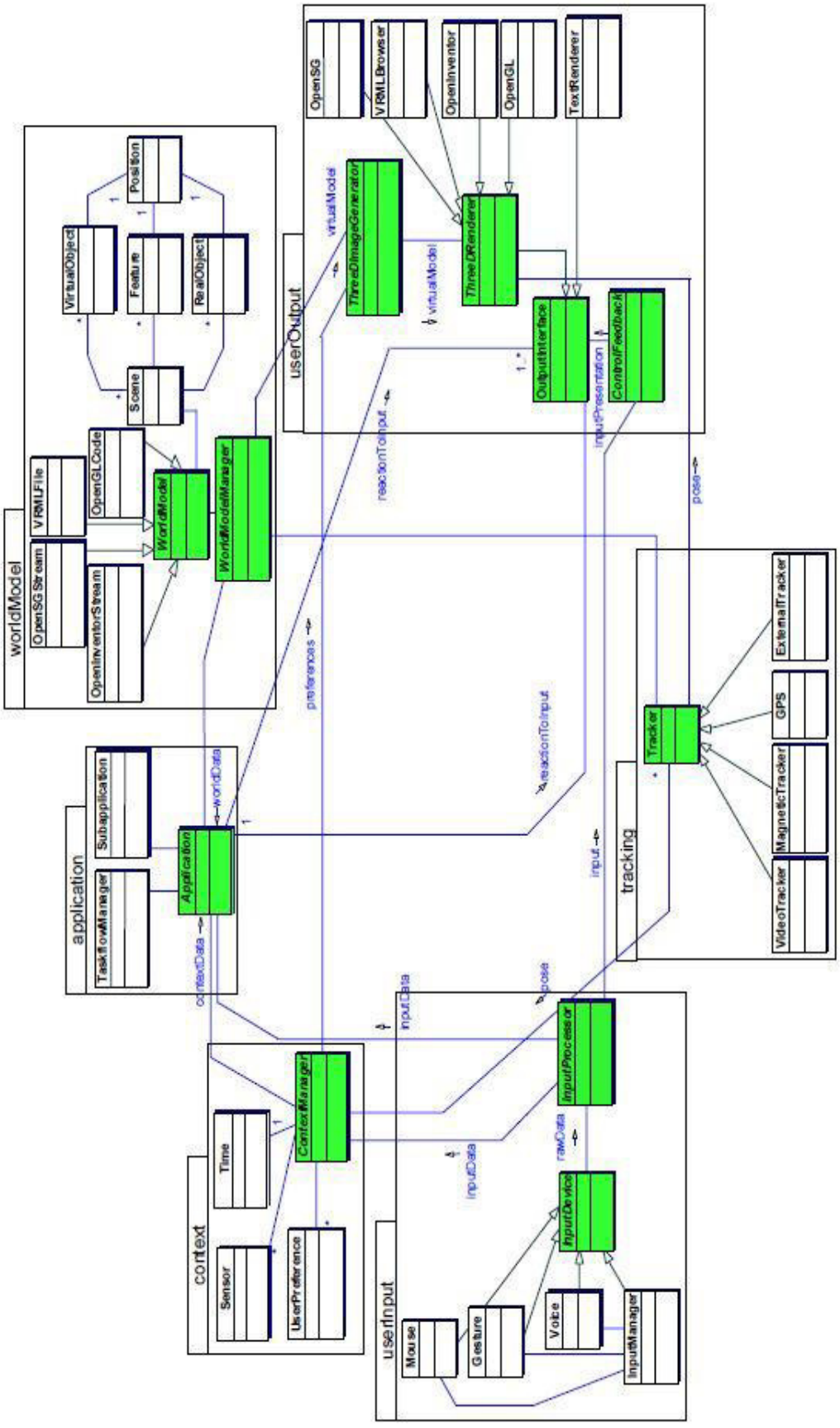


Figure 3.5 : architecture générale d'une application de RA

5.2 Partie 2 reconstruction 3D à partir d'images :

Comme décrit dans le chapitré II la reconstruction d'objets 3d a partir d'images passe par plusieurs étapes pour l'obtention de l'objet finale, ces étapes peuvent êtres résumées dans le schéma suivant :

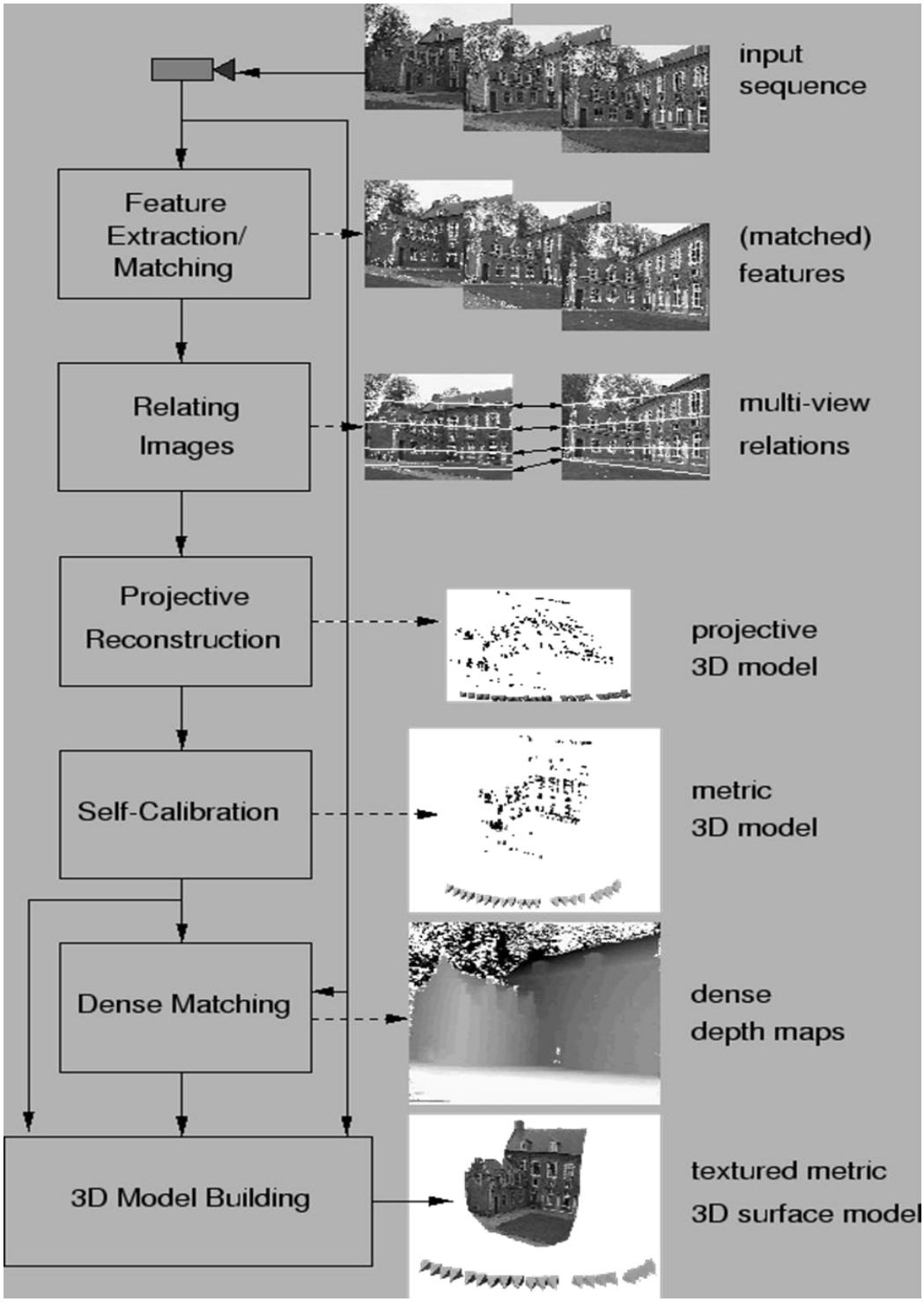


Figure 3.6 : schéma représentant les étapes de reconstruction 3D

A partir de ce schéma nous avons élaboré une architecture généralisée pour cette partie qui peut être illustré dans la figure suivante :

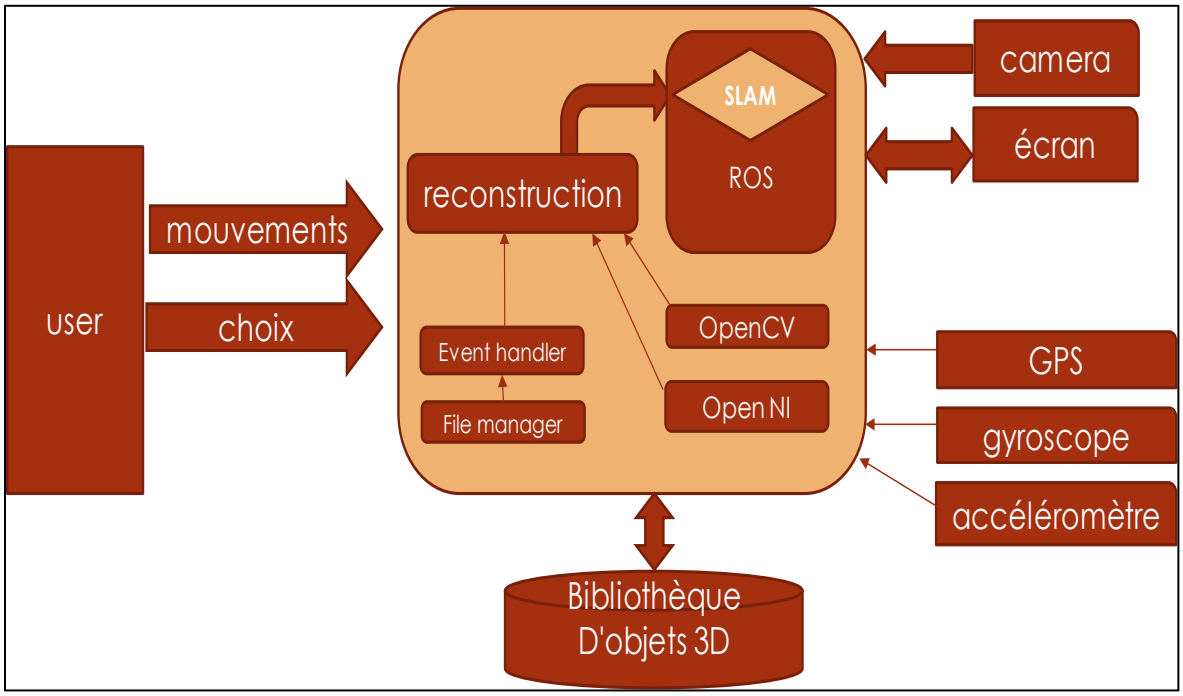


Figure 3.7 : architecture généralisée de la partie reconstruction 3D

Le cœur de cette architecture est le procédé SLAM (Simultaneous Localisation And Mapping) qui est un sous-système de la bibliothèque ROS (Robot Operating System). Une bibliothèque utilisée dans le guidage et la manœuvre de robots. Ici le robot est remplacé par le mobile.

Le SLAM permet d'extraire les points d'intérêt des images qu'il reçoit et en même temps détecter avec précision la position de la caméra (donc du mobile), deux paramètres essentiels pour la reconstruction. La figure suivante illustre le principe de fonctionnement du SLAM.

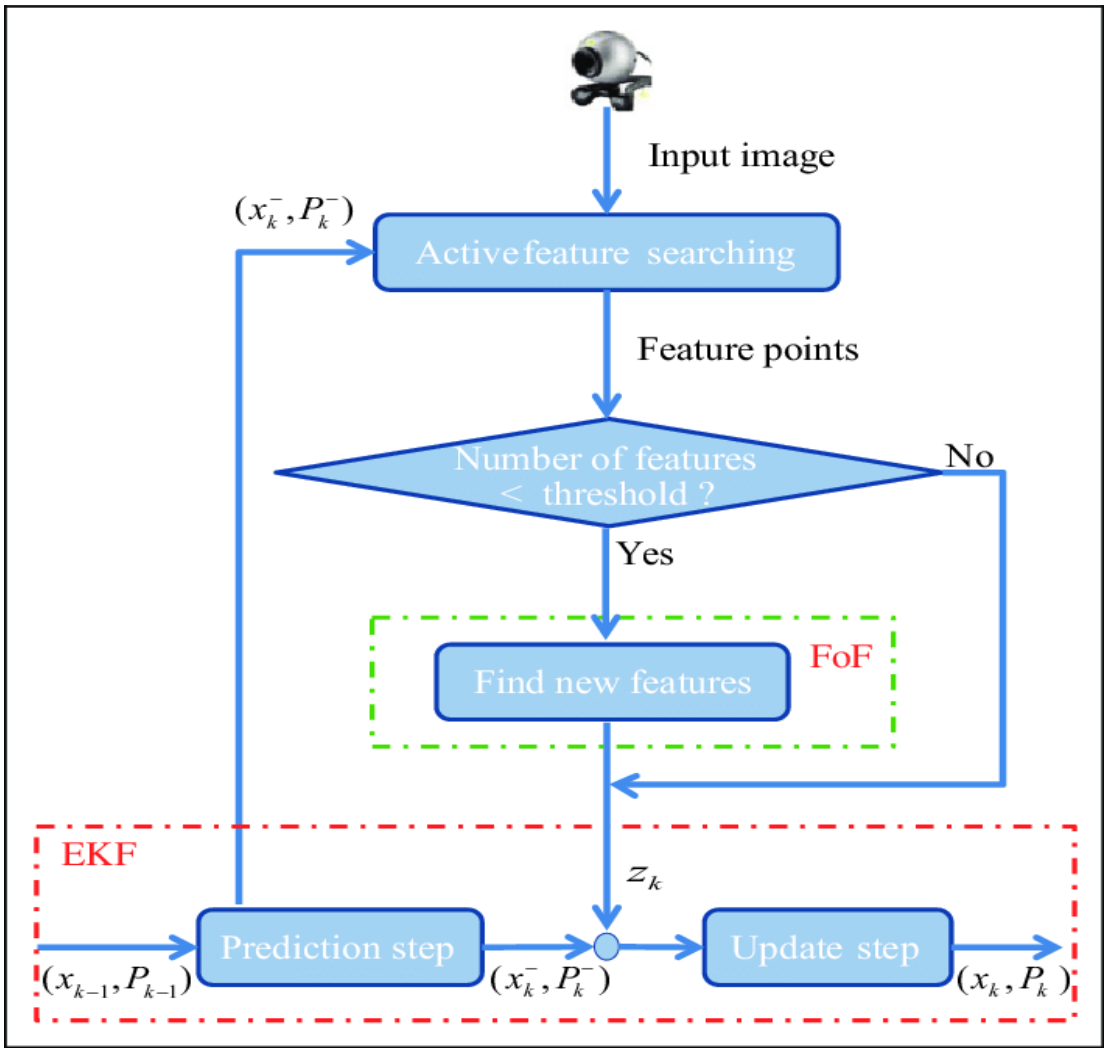


Figure 3.8 : principe du SLAM

5. Conclusion :

Dans ce chapitre nous avons donné une spécification sur la conception de l'application, cette phase représente l'une des phases les plus importantes dans le processus de développement d'un logiciel.

Elle décrit le système du point de vue général et détaillé pour comprendre et réussir la phase de programmation.

Nous avons détaillé l'architecture globale de l'application, et nous avons détaillé chaque composant de manière indépendante.

Dans le prochain chapitre, nous allons illustrer la réalisation de notre système en représentant quelques interfaces et quelques résultats obtenus, avec les structures des données qui sont choisies pour implémenter ce système.

CHAPITRE IV :

Réalisation de l'application

1. Introduction :

L'implémentation d'un logiciel est atteinte après un enchainement de plusieurs étapes dans le processus de développement, et son but principal est de réaliser un produit capable de résoudre les problèmes posés en utilisant des outils et des algorithmes.

Nous avons donné dans le précédent chapitre la conception de façon globale et détaillée avec les méthodes utilisées, le présent chapitre donne une vue sur le système et les outils afin d'arriver à un système fiable.

Dans ce chapitre nous allons présenter en premier lieu, l'environnement de développement avec les différentes bibliothèques utilisées, ainsi que les structures de données choisies pour implémenter ce type de système. Ensuite, nous allons présenter les algorithmes utilisés illustrés par quelques résultats obtenus, nous terminons ce chapitre par une conclusion.

2. Environnement du développement et Langage de la programmation JAVA :

Notre système est développé sur un ordinateur avec un processeur Intel I5 avec une RAM de 4 Go et une carte graphique GT820m 1GO sous Windows 7 64bits.

JAVA un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld [52].

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows,

Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java

4. Environnement de programmation (Android Studio) :

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA. Avant Android Studio, de 2009 à 2014, Google propose comme environnement de développement officiel une distribution spécifique de l'environnement **Eclipse**, contenant notamment le **SDK** d'Android.

Android Studio est annoncé le 15 mai 2013 lors du Google I/O et une version *Early Access Preview* est disponible le jour même[1].Le 8 décembre 2014, Android Studio passe de version bêta à version stable 1.0. L'environnement devient alors conseillé par Google, et Eclipse est délaissé[2].

Android Studio permet principalement d'éditer les fichiers **Java/Kotlin** et les fichiers de configuration **XML** d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.

5. Outils d'implémentation

5.1 OpenGL :

OpenGL (Open Graphics Library) est un ensemble normalisé de fonctions de calcul d'images 2D ou 3D lancé par Silicon Graphics en 1991. Cette interface de programmation est disponible sur de nombreuses plateformes où elle est utilisée pour des applications qui vont du jeu vidéo jusqu'à la CAO en passant par la modélisation.

OpenGL permet à un programme de déclarer la géométrie des objets sous la forme de points, de vecteurs, de polygones, de bitmaps et de textures. OpenGL effectue ensuite des calculs de projection en vue de déterminer l'image à l'écran, en tenant compte de la distance, de l'orientation, des ombres, de la transparence et du cadrage.

L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plates-formes,

elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D et certaines applications 2D vectorielles.

Cette bibliothèque est également utilisée dans l'industrie du jeu vidéo où elle est souvent en rivalité avec la bibliothèque de Microsoft : Direct3D.

Une version nommée OpenGL-ES a été conçue spécifiquement pour les applications embarquées (téléphones portables, agenda de poche, consoles de jeux…).

5.2 OpenCV

OpenCV (Open Source Computer Vision) est une librairie open source de traitement et analyse d'images et vidéos avec des interfaces pour les principaux langages de programmation C, C++, Java, C#, Python. Elle a l'avantage d'être optimisée pour les applications temps réelles, de fournir une API bas et haut niveau ainsi qu'une interface pour le langage de programmation parallèle IPP [54]. Parmi les fonctions les plus récurrentes dans le monde de l'entreprise ou dans les laboratoires on peut retrouver :

♣ La manipulation d'images (chargement, sauvegarde, copie, conversion...)

♣ Le traitement d'images (filtrage, détections de discontinuités, morphologie mathématique...)

♣ L'analyse d'images (composantes connexes, ajustement de primitives...)

♣ La manipulation et acquisition de vidéos

♣ La vision (calibration de caméra, stéréovision, recherche d'association...)

6. Les structures de données utilisées:

6.1 Les objets 3D au format « .obj » :

OBJ est un format de fichier contenant la description d'une géométrie 3D. Il a été défini par la société **Wavefront** Technologies dans le cadre du développement de son logiciel d'animation Advanced **Visualizer**. Ce format de fichier est ouvert et a été adopté par d'autres logiciels 3D (tels que **3D Turbo** de iluac software , **Poser** de e-frontier, **Maya** de Autodesk, **Blender**, **MeshLab**, **3D Studio Max**, **Lightwave** de Newtek, **GLC Player** …etc) pour des traitements d'import / export de données.

Les formes géométriques peuvent être définies par des polygones ou des surfaces lisses telles que des surfaces rationnelles et non rationnelles.

Structure du fichier

Les fichiers OBJ sont au format ASCII (une version binaire existe, identifiée par l'extension MOD).

Un commentaire peut être placé en faisant commencer la ligne par le caractère #.

Une surface polygonale est décrite par un ensemble de sommets (accompagné de coordonnées de texture et de normales en chaque sommet) et d'un ensemble de faces.

Un sommet est défini de la manière suivante:

```
v 1.0 0.0 0.0
```

Une coordonnée de texture est définie de la manière suivante:

```
vt 1.0 0.0
```

Une normale est définie de la manière suivante:

```
vn 0.0 1.0 0.0
```

Chaque face est ensuite définie par un ensemble d'indices faisant référence aux coordonnées des points, de texture et des normales définies précédemment.

Par exemple, la face suivante

```
f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3
```

définit un triangle constitué des sommets d'indices v1, v2 et v3 dans la liste des sommets v. Chacun de ces sommets possède une coordonnée de texture identifiée par son indice dans la liste des coordonnées de texture vt et une normale identifiée dans la liste des normales vn.

Lorsque plusieurs objets cohabitent dans le même fichier, la section définissant l'objet est définie par

```
o [nom de l'objet]
```

Lorsque plusieurs groupes de faces cohabitent dans le même objet, la section définissant chaque groupe est définie par

```
g [nom du groupe]
```

Des matériaux peuvent être référencés en important des fichiers .mtl (Material Template Library)

```
usemtl [nom de matériau]
```

7. Présentation de l'application :

7.1 La partie réalité augmentée :

Elle est composée des classes suivantes :

AssetsFileUtility.java : permet de gérer les fichiers RAW contenus dans /assets

BaseFileUtil.java : permet de gérer le répertoire de base où sont stockés nos objets 3D.

CheckFileManagerActivity.java : permet a l'utilisateur d'indiquer un autre emplacement pour les objets 3d .obj.

Config.java : fichier de configuration de l'application.

FixedPointUtilities.java : regroupe quelques fonctions mathématiques concernant les tableaux.

Group.java : contient quelques fonctions utiles pour le parsing des fichiers .obj.

Instructions.java : permet de gérer le fichier help pour manipuler l'application.

LightingRenderer.java : pour le rendu de la Lumière avec OpenGL.

Material.java : pour gérer les matériaux des objets 3d.

MemUtil.java : pour la gestion de la mémoire.

Model.java : permet d'initialiser un modèle avant le parsing du fichier .obj.

Model3D.java : permet de créer l'objet en 3D avec OpenGL.

ModelChooser.java : c'est ce qui se lance au début de l'application permettant a l'utilisateur de choisir le modèle a afficher ou sinon ajouter un nouveau modèle via reconstruction 3D.

ModelViewer.java : s'occupe de l'affichage de l'objet en temps réel en RA.

MtlParser.java : pour le parsing des fichiers .mtl des objets 3D.

MyArrayIterator.java : gestion de tableaux.

ObjParser.java : pour le parsing des fichiers .obj.

ParseException.java : gestion des exceptions pendant le parsing.

Renderer.java : gérer le rendu sous OpenGL.

SDCardFileUtil.java : gestion des fichiers de la carte SD du mobile.

SimpleTokenizer.java : gestion des tokens.

Util.java : contient des fonctions d'optimisation des objets pour le parsing.

Vector3D.java : permet de gérer des opérations sur les vecteurs 3D.

7.2 La partie reconstruction 3D :

Cette partie comporte un nombre très important de classes qu'il serait un peu abusif de tous les listés ici, mais on va présenter les différents sous-systèmes dans lesquels ils sont groupés.

Appmanager : permet de lire les données du buffer lancer le SLAM et envoyer les resultats au renderer.

Buffers : permet des gérer les différents buffers d'entrée et de sortie.

Datatypes : contient les classes qui gèrent les différents structures de données utilisées.

Export : permet l'exportation du résultat sur un fichier (.obj).

Graph : pour gérer le scène-graph.

Gui : permet de gérer l'interface de l'application.

Input : permet de gérer les données en entrée des capteurs et des images.

Kalman : implémentation du filtre de kalman.

Main : ensemble de classes pour implémenter l'algorithme de SLAM.

Matcher : pour gérer la liaison des points d'intérêts.

Quadtree : pour créer et manipuler des quadtree.

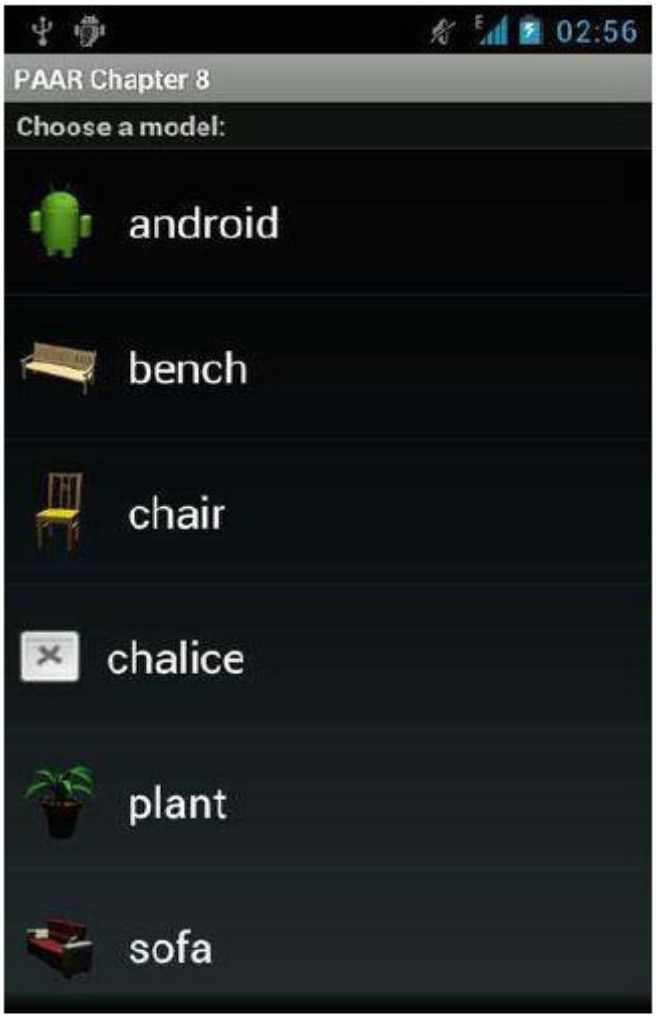
Traker : pour l'extraction des points d'intérêts.

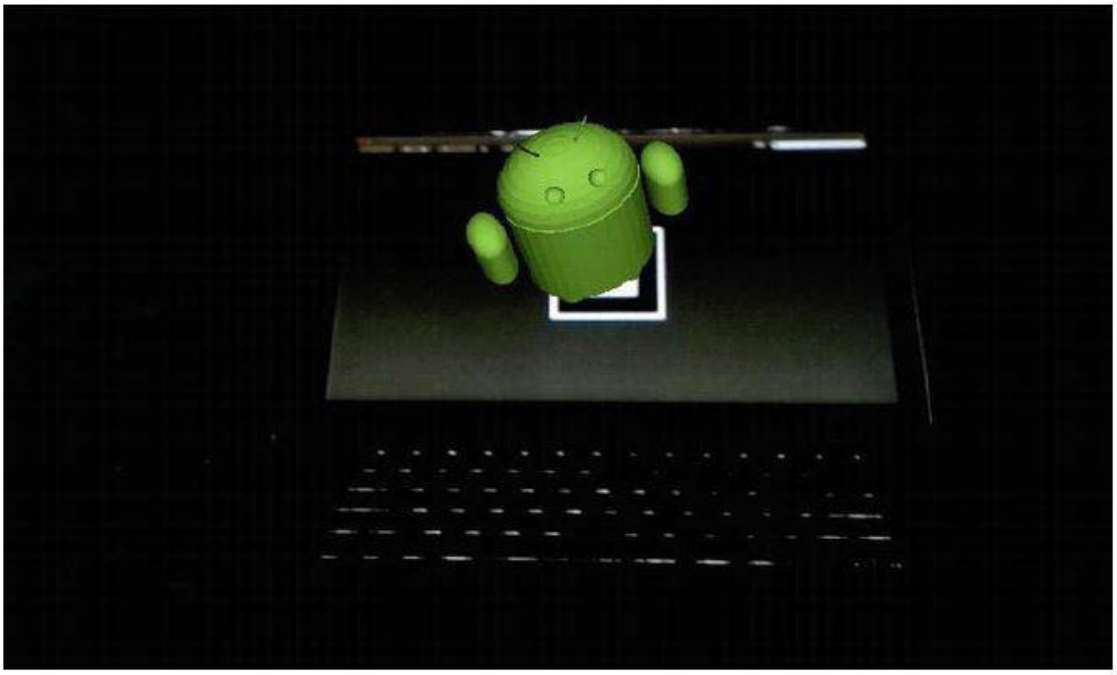
Sampleconsensus : pour l'implémentation de l'algorithme RANSAC .

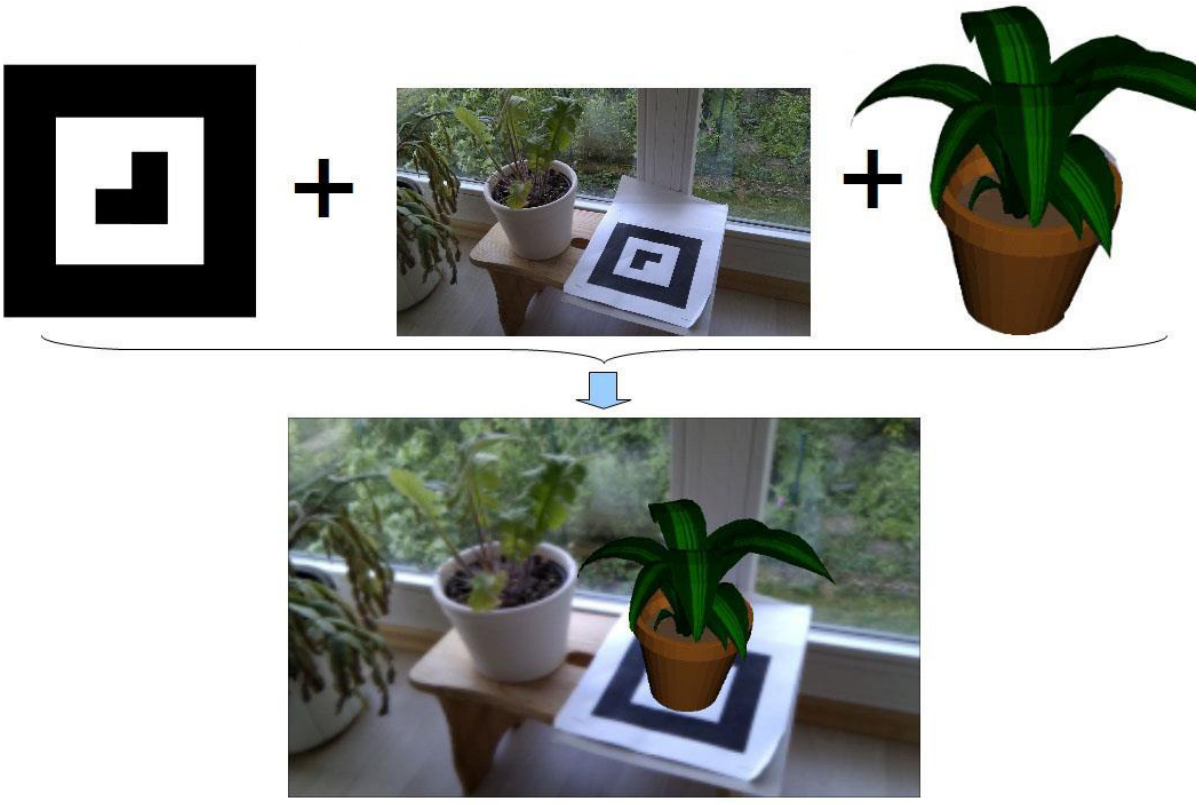
Utils : ensembles de classes pour effectuer certains calculs secondaires.

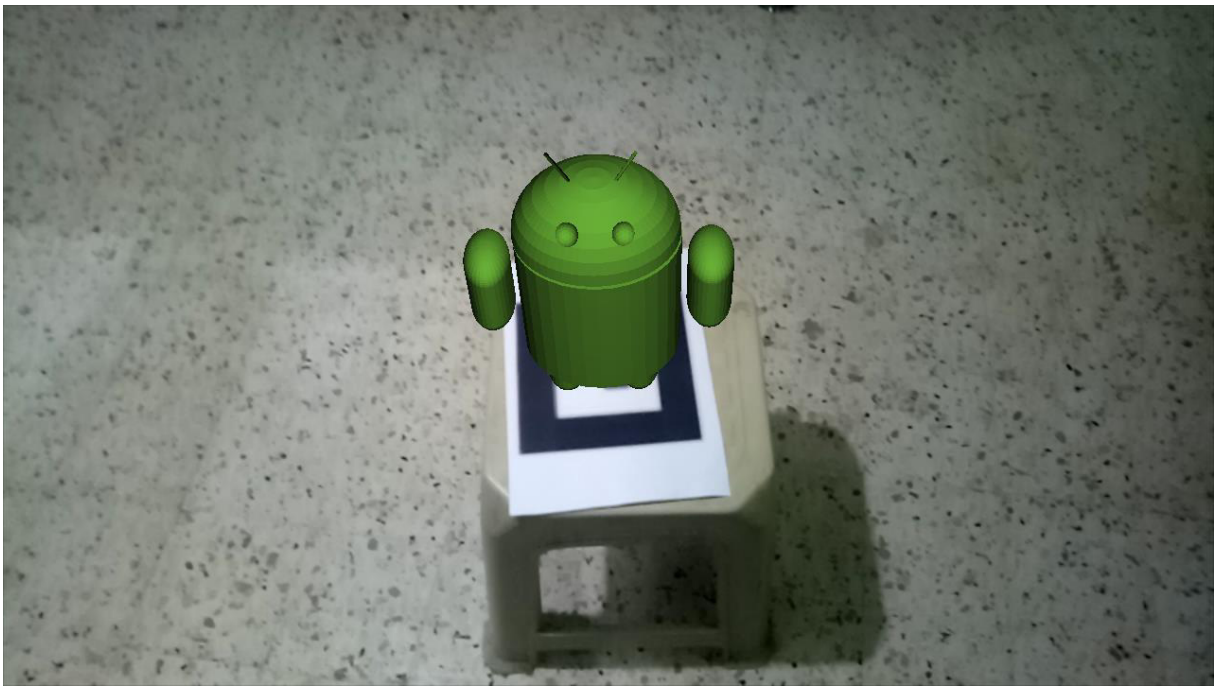
Ajouter a cela l'utilisation des bibliothèques suivantes : Boost, OpenCV, OpenNI, G2O, Eigen.

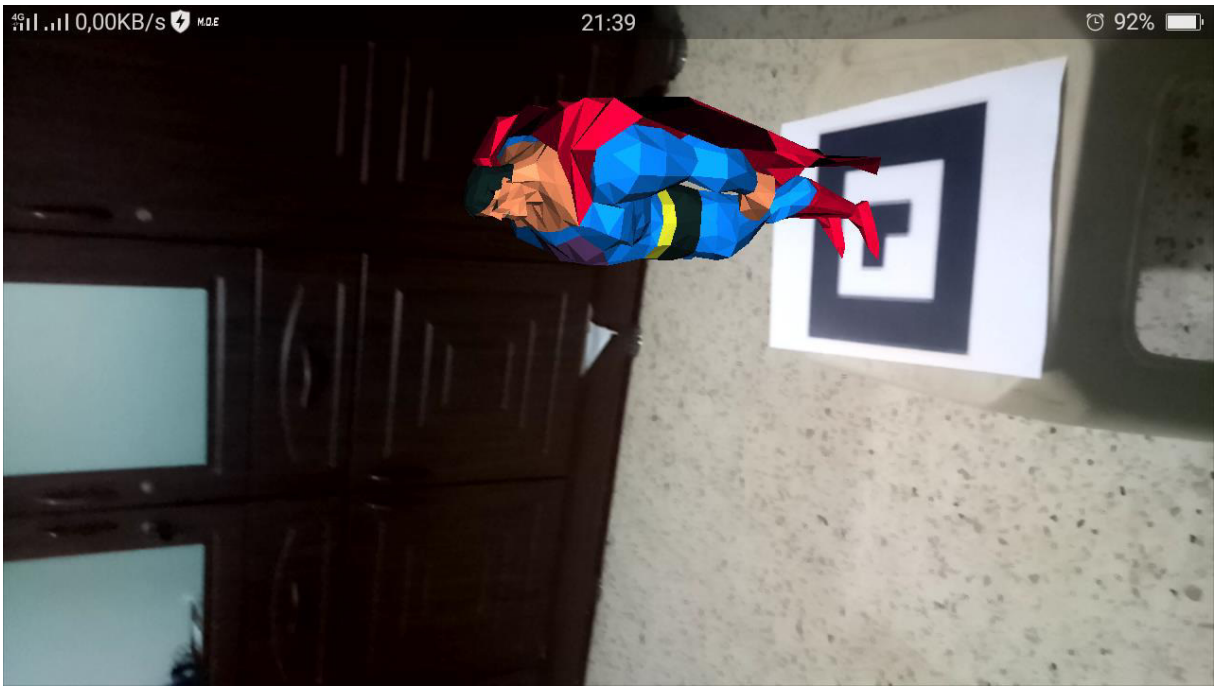
8. Interface du système et résultats :

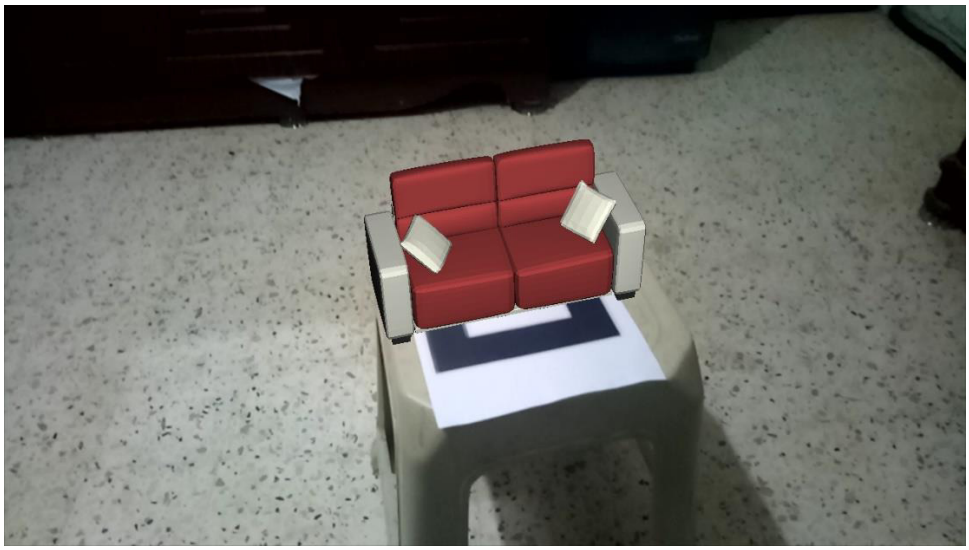









0,00KB/s
21:39
92%



Conclusion générale

Aux cours de ce projet nous avons essayé de répondre à la problématique évoquée dans l'introduction générale concernant la reconstruction 3D à partir d'images dans une application de réalité augmentée. Pour cela nous avons commencé par présenter la Réalité Augmentée et nous avons pu constater l'énorme étendu de ces capacités et ce qu'elle pourrait apporter à notre quotidien.

Après nous avons présenté en deux chapitres la théorie de la reconstruction 3D à partir d'images et sur les approches qui ont vu le jour jusqu'à maintenant. Nous avons consacré la première partie à la construction du modèle 3D à partir d'image, ce qui permet d'éviter les contraintes de la modélisation géométriques et même de générer des objets réel de façon interactive. La seconde partie concernait la modélisation, le rendu et l'intégration de l'objet construit.

Cette longue étude théorique nous a fourni les éléments essentiels sur la façon de créer de nouveaux objets 3D issues de la vie réelle par le simple maniement de la caméra du mobile, ce qui rend cette technologie accessible a presque tout le monde et ouvre de nouvelles perspectives quant à l'utilisation de cette technologie.

Enfin nous avons présenté notre application qui avait pour vocation de montrer ce que pourrait donner la fusion entre la réalité augmentée et la reconstruction 3D à base d'images, les résultats qu'on a obtenus donne une certaine réponse à la problématique mais pour être vraiment exploitable il faudrait introduire plus d'améliorations.

- Ces améliorations concernent donc les perspectives de ce travail et peuvent être résumées ainsi : Proposer la possibilité d'éditer les objets 3D après création pour pouvoir les affinés, ou supprimer l'utilisation des marqueurs pour l'affichage des objets en RA.

- On pourra aussi étendre l'application pour supporter d'autres types d'objets issue des logiciels de modélisation 3D professionnels,
- Permettre la prise en charge de modèles d'éclairement plus réalistes et pourquoi pas l'utilisation de l'intelligence artificielle pour la reconnaissance d'objets dans les images.

Liste des figures :

Bibliographie

1. R. Sood, " Pro Android Augmented Reality" Apress , 2012.
2. D. Chatzopoulos et al "Mobile Augmented Reality Survey: FromWhere We Are to Where We Go". IEEEaccess, April 2017.
3. Y. Marion et al "Augmented reality using android" costumer driven project, NTNU-Trondheim Norwegian university of science qnd technology, novembre 2010.
4. Ronald T. Azuma “A Survey of Augmented Reality”, In *Presence: Teleoperators and Virtual Environments 6*, 4 (August 1997).
5. Bernd Brügge, "Study on Software Architectures for Augmented Reality Systems", Report to the ARVIKA consortium, 2004.
6. https://en.wikipedia.org/wiki/Augmented_reality.
7. https://fr.wikipedia.org/wiki/Réalité_augmentée.
8. http://www.cs.unc.edu/~marc/tutorial/tutorial02.html.
9. K.Yaghmour, "Embedded Android", edition Oreilly , 2013.
10. C. Slama, Manual of Photogrammetry, American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.
11. C. Harris and M. Stephens, ``A combined corner and edge detector", Fourth Alvey Vision Conference, pp.147-151, 1988.
12. D. Bondyfalat and S. Bougnoux, ``Imposing Euclidean Constraints During Self-Calibration Processes", Proc. SMILE Workshop (post-ECCV'98), Lecture Notes in Computer Science, Vol. 1506, Springer-Verlag, pp.224-235, 1998.
13. A. Shashua, ``Trilinearity in visual recognition by alignment", Computer Vision - ECCV'94, Lecture Notes in Computer Science, Vol. 801, Springer-Verlag, pp. 479-484, 1994.
14. A. Fitzgibbon and A. Zisserman, ``Automatic camera recovery for closed or open image sequences", Computer Vision - ECCV'98, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, 1998. pp.311-326, 1998.
15. R. Hartley and P. Sturm, ``Triangulation", Computer Vision and Image Understanding, 68(2):146-157, 1997.
16. P. Torr. ``An assessment of information criteria for motion model selection". In CVPR97, pages 47-53, 1997.
17. R. Haralick, C. Lee, K. Ottenberg, and M. Nolle, ``Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", International Journal of Computer Vision, Vol.13, No.3, 1994, pp. 331-356.
18. O. Faugeras, Q.-T. Luong and S. Maybank. ``Camera self-calibration: Theory and experiments", Computer Vision - ECCV'92, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 321-334, 1992.
19. M. Pollefeys and L. Van Gool, ``Self-calibration from the absolute conic on the plane at infinity", Proc. Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, Vol. 1296, Springer-Verlag, pp. 175-182, 1997.
20. B. Boufama, R. Mohr and F. Veillon, ``Euclidian Constraints for Uncalibrated Reconstruction", Proc. International Conference on Computer Vision, pp. 466-470, 1993.
21. D. Bondyfalat and S. Bougnoux, ``Imposing Euclidean Constraints During Self-Calibration Processes", Proc. SMILE Workshop (post-ECCV'98), Lecture Notes in Computer Science, Vol. 1506, Springer-Verlag, pp.224-235, 1998.
22. D. Brown, ``The bundle adjustment - progress and prospect", XIII Congress of the ISPRS, Helsinki, 1976.
23. A. Shashua, ``Omni-Rig Sensors: What Can be Done With a Non-Rigid Vision Platform?" Proc. of the Workshop on Applications of Computer Vision (WACV), Princeton, Oct. 1998.
24. P. Courtney, N. Thacker and C. Brown, ``A hardware architecture for image rectification and ground plane obstacle detection", Proc. Intern. Conf. on Pattern Recognition, pp. 23-26, 1992.
25. M. Pollefeys, R. Koch and L. Van Gool. ``Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters", International Journal of Computer Vision.
26. P. Sturm, ``Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction", Proc. 1997 Conference on Computer Vision and Pattern Recognition, IEEE Computer Soc. Press, pp. 1100-1105, 1997.
27. S. Roy, J. Meunier and I. Cox, ``Cylindrical Rectification to Minimize Epipolar Distortion", Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.393-399, 1997.
28. M. Pollefeys, R. Koch and L. Van Gool, "A simple and efficient rectification method for general motion", Proc.ICCV'99 (international Conference on Computer Vision), pp.496-501, Corfu (Greece), 1999.
29. U. Dhond and J. Aggarwal, ``Structure from Stereo - A Review", IEEE Trans. Syst., Man and Cybern. 19, 1489-1510, 1989.
30. D. Marr and T. Poggio, ``A Computational Theory of Human Stereo Vision", *Proc. Royal Society of London*, Vol. 204 of B, pp. 301-328, 1979.
31. G. Gimel'farb, ``Symmetrical approach to the problem of automatic stereoscopic measurements in photogrammetry", Cybernetics, 1979, 15(20, 235-247; Consultants Bureau, N.Y.

32. I. Cox, S. Hingorani and S. Rao, ``A Maximum Likelihood Stereo Algorithm", Computer Vision and Image Understanding, Vol. 63, No. 3, May 1996.
33. L. Falkenhagen, ``Depth Estimation from Stereoscopic Image Pairs assuming Piecewise Continuous Surfaces", European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Productions, Hamburg, Germany, 1994.
34. A. Koschan, ``Eine MethodenbankzurEvaluierung von Stereo-Vision-Verfahren", Ph.D. Thesis, Technische Universität Berlin, Berlin, Germany, 1991.
35. R. Koch, AutomatischeOberflachenmodellierung starrer dreidimensionalerObjekteausstereoskopischenRundum-Ansichten, PhD thesis, University of Hannover, Germany, 1996 also published as Fortschritte-Berichte VDI, Reihe 10, Nr.499, VDI Verlag, 1997.
36. R. Koch, ``3-D Surface Reconstruction from Stereoscopic Image Sequences", Proc. Fifth International Conference on Computer Vision, IEEE Computer Soc. Press, pp. 109-114, 1995.
37. G. Turk and M. Levoy "Zippered Polygon Meshes from Range Images" Proceedings of SIGGRAPH '94 pp. 311-318.
38. E. Ofek, E. Shilat, A. Rappopport and M. Werman, ``Highlight and Reflection Independent Multiresolution Textures from Image Sequences", IEEE Computer Graphics and Applications, vol.17 (2), March-April 1997.
39. M. Irani and S. Peleg, Super resolution from image sequences, Proc. International Conference on Pattern Recognition, Atlantic City, NJ, 1990.
40. B. Curless and M. Levoy, ``A Volumetric Method for Building Complex Models from Range Images" Proc. SIGGRAPH '96, 1996.
41. M. Soucy and D. Laurendeau. ``A general surface approach to the integration of a set of range views". IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(4):344-358, April 1995.
42. W.E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Computer Graphics (SIGGRAPH '87 Proceedings), volume 21, pages 163-169, July 1987.
43. R. Koch, B. Heigl, M. Pollefeys, L. Van Gool and H. Niemann, ``A Geometric Approach to Lightfield Calibration", Proc. CAIP99, LNCS 1689, Springer-Verlag, pp.596-603, 1999.
44. B. Heigl, R. Koch, M. Pollefeys, J. Denzler and L. Van Gool, ``PlenopticModeling and Rendering from Image Sequences taken by Hand-held Camera", In Proc. DAGM'99, pp.94-101.
45. M. Levoy and P. Hanrahan, ``Lightfield Rendering", Proc. SIGGRAPH '96, pp 31-42, ACM Press, New York, 1996.
46. S. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, ``The Lumigraph", Proc. SIGGRAPH '96, pp 43-54, ACM Press, New York, 1996.
47. M. Fischler and R. Bolles, ``RANdomSAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography", Commun. Assoc. Comp. Mach., 24:381-95, 1981.
48. P. Torr, A. Fitzgibbon and A. Zisserman, ``Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure", Proc. International Conference on Computer Vision, Narosa Publishing house, pp 485-491, 1998.