C.D.T.A

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAITRE

MINISTÈRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة سعد دحلب البليدة 1

UNIVERSITÉ SAAD DAHLEB DE BLIDA 1

كلية العلوم - دائرة الإعلام الآلي

FACULTÉ DES SCIENCES

DÉPARTEMENT D'INFORMATIQUE

MÉMOIRE DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLÔME

MASTER EN INFORMATIQUE

OPTION : INGÉNIERIE LOGICIEL

THÈME :

SUIVI D'OBJET RIGIDE BASÉ SUR LES POINTS D'INTERÉT : APPLICATION À LA RÉALITE AUGMENTÉE

enté par : DAHMANI Abdelmalek
KORICHI Nassiba

Promotrice : Mlle BENBLIDIA Nadjia
Encadreur : Mr. HAMIDIA Mahfoud
Mme ZENATI-HENDA Nadia

Promotion 2013/2014

Remerciements

Nous tenons tout d'abord à remercier le bon dieu de nous avoir donné la santé, le courage et la volonté pour réaliser ce travail.

Nous adressons nos plus vifs remerciements à notre promotrice Mlle BENBLIDIA Nadjia, Maître de Conférences à l'Université de Blida 1 pour la confiance qu'il nous a accordée, pour ses précieux conseils et pour ses remarques pertinentes.

On tient également à exprimer notre gratitude à nos encadreurs Mr. HAMIDIA Mahfoud, Attaché de Recherche et Mme ZENATI-HENDA Nadia, Maître de Recherche au Centre de Développement des Technologies Avancées (CDTA) pour les efforts fournis, les conseils prodigués et la compréhension durant la période du stage.

Nous aimerions aussi à remercier Mr. BELLARBI Abdelkader, Ingénieur de Recherche au CDTA pour ses encouragements, ses conseils et son soutien technique.

On remercie très sincèrement, les membres du jury d'avoir bien voulu accepter d'examiner et d'évaluer ce travail et apporter une contribution critique à celui-ci et aussi toutes les remarques positives et constructives.

Enfin, nous tenons à exprimer toute notre gratitude à tous ceux qui ont contribué, de près ou de loin, à la concrétisation de ce travail

ملخص

الحقيقة المعززة تتمثل في إضافة معلومات افتراضية ينتجها الحاسوب إلى مشهد فيديو حقيقي . في السنوات الأخيرة ، صارت تطبق هذه التكنولوجيا في العديد من المجالات على غرار: الطب ،الروبوتيك ، الصيانة ، الهندسة المعمارية، الخ.
من بين المشاكل في الحقيقة المعززة المحاذاة افتراضي-حقيقي . يجب ضمان التلاحم المكاني والزماني من أجل تثبيت الشيء الافتراضي بالشكل صحيح بالنسبة إلى الكاميرا .تستخدم علامات بأهداف مرمزة لضمان هذا التوافق ، ولكن فعاليتها محدودة .لهذا السبب، يتم استخراج مصادر بصرية مثل نقاط الاهتمام مباشر من المشهد الحقيقي ، وتستخدم كعلامات طبيعية.
يتمثل هذا العمل في استخدام واصفات جديدة لنقاط الاهتمام لتحسين أداء نظام تتبع الشيء الصلب و إضافة الشيء الافتراضي بالشكل صحيح في المشهد الحقيقي.
الكلمات المفتاحية : الحقيقة المعززة ، المحاذاة، نقاط الاهتمام ، وصف ، التوافق ، تتبع الشيء.

Résumé

La réalité augmentée consiste à ajouter des informations virtuelles générées par ordinateur à une scène vidéo réelle. Dans les années récentes, cette technologie est appliquée dans plusieurs domaines tel que : la médecine, la robotique, la maintenance, l'architecture, etc.
L'un des problèmes dans la réalité augmentée c'est alignement virtuelle-réelle. Une cohérence spatiale et temporelle doit être assurée pour recaler correctement l'objet virtuel par rapport à la caméra. Des marqueurs à cible codée sont utilisés pour assurer cet alignement, mais ils ont certaines limitations. Pour ce raison, des primitives visuelles comme les points d'intérêt sont extrait directement à partir de la scène réelle, utilisées comme des marqueurs naturels.
Ce travail consiste à utiliser des nouveaux descripteurs des points d'intérêt pour améliorer les performances d'un système de suivi d'objet rigide et incruster correctement l'objet virtuel dans la scène réel.
Mots clés: Réalité augmentée, alignement, points d'intérêt, description, mise en correspondance, suivi d'objet.

Abstract

Augmented reality is to add virtual information generated by computer to a real video scene. In recent years, this technology is applied in several fields such as: medicine, robotics, maintenance, architecture, etc.
One of the problems in augmented reality is virtual-real alignment. Spatial and temporal coherence must be sure to correctly reposition the virtual object from the camera. Tag coded markers are used to ensure this alignment, but they have some limitations. For this reason, visual primitives such as interest points are extracted directly from the real scene, used as natural markers.
This work consist the use of new descriptors of interest points to improve the performance of rigid object tracking system and embed correctly the virtual object in the real scene.
Key words: Augmented reality, alignment, interest points, description, matching, object tracking.

Table des matières

Introduction générale 1

Chapitre 1 : La réalité augmentée, concept et définitions 3

1.1. Introduction 3

1.2. La réalité augmentée 3

1.2.1. Augmentation de l'utilisateur 3

1.2.2. Augmentation des objets physiques 3

1.2.3. Augmentation de l'environnement des utilisateurs et des objets 3

1.3. La réalité virtuelle 4

1.3.1. La réalité virtuelle et la réalité augmentée 5

1.3.2. Continuum réel – virtuel 'La réalité mixte' 5

1.4. Principe et problématique de l'augmentation 6

1.4.1. Alignement des caméras réelle et virtuelle 6

1.4.2. Cohérence spatio-temporelle 6

1.4.3. Cohérence photométrique 7

1.5. Les applications de réalité augmentée 7

1.5.1. La médecine 7

1.5.2. La maintenance 8

1.5.3. La robotique 8

1.5.4. L'architecture 9

1.6. Les composantes d'un système de réalité augmentée 9

1.6.1. Affichage basé sur la configuration moniteur 10

1.6.2. Affichage basé sur la configuration vue à travers vidéo 10

1.6.3. Affichage basé sur la configuration vue à optique 11

1.7. Approches de la réalité augmentée 12

1.7.1. Réalité augmentée à partir de marqueurs 12

1.7.2. Réalité augmentée à partir de primitives naturelles 13

1.8. Schéma global du système de réalité augmentée à développer 14

1.9. Conclusion 15
Chapitre 2 : Suivi d'objet rigide basé sur les points d'intérêt 16
2.1. Introduction 16
2.2. Techniques de reconnaissance existantes : 16
2.2.1. Techniques basées sur les caractéristiques géométriques 16
2.2.2. Techniques basées sur les caractéristiques de luminance 17
2.2.3. Techniques basées sur les points d'intérêt 17
2.3. Définition d'un point d'intérêt 17
2.4. Les différents types de transformations d'image 18
2.5. Détection de points d'intérêt 19
2.5.1. Détecteur de Moravec 19
2.5.2. Détecteur de Harris 20
2.5.3. Détecteur FAST 23
2.5.4. Détecteur SURF 24
2.6. Description de points d'intérêt 26
2.6.1. Détecteur SIFT 27
2.6.2. Descripteur SURF 30
2.6.3. Descripteur BRIEF 32
2.6.4. Descripteur ORB 34
2.7. Suivi d'objet par la mise en correspondance 35
2.7.1. La mise en correspondance 36
2.7.2. L'algorithme RANSAC 37
2.7.3. Estimation de l'homographie 38
2.8. Conclusion 39
Chapitre 3 : Conception 40
3.1. Introduction 40
3.2. Le choix de processus de conception 40
3.3. Présentation du projet 42

3.3.1. Analyse et spécification des besoins 42
3.4. Le choix des méthodes et algorithmes utilisés 43
3.4.1. Le choix d'une méthode appropriée à notre cas d'étude 43
3.4.2. Architecture global du système de recalage à développer 44
3.4.3. Choix de la méthode ORB pour l'extraction des points d'intérêt 44
3.4.4. La mise en correspondance 45
3.5. Conception du système 45
3.5.1. Diagrammes de cas d'utilisation 45
3.5.2. Diagrammes de classes 49
3.5.3. Diagramme de séquence 50
3.6. Conclusion 54
Chapitre 4 : Mise en œuvre du système, application et résultats 55
4.1. Introduction 55
4.2. Environnement et outils de développement 55
4.2.1. Visual Studio 2013 55
4.2.2. Langage C# 55
4.2.3. Librairies de vision par ordinateur et 3D 56
4.3. Implémentation et étapes de réalisation 57
4.3.1. La modélisation 3D 57
4.3.2. Extraction des paramètres (points d'intérêt) 57
4.3.3. Le suivi d'objet basé sur la mise en correspondance 58
4.3.4. L'augmentation de la scène réelle (Insertion des objets 3D) 59
4.4. Tests et évaluation 59
4.4.1. Le temps de traitement 60
4.4.2. Répétabilité 61
4.4.3. La précision et l'exactitude 63
4.5. Conclusion 64
Conclusion général 65
Bibliographie 67

Liste des figures

Fig. 1.1. Exemple de la réalité augmentée. 4
Fig. 1.2. Le "Continuum réel – virtuel" de Milgram . 5
Fig. 1.3. Augmentations par le véhicule rouge 7
Fig. 1.4. Visualisation 3D utilisant la RA dans le domaine médicale 8
Fig. 1.5. Application de la RA pour la maintenance d'une imprimante 8
Fig. 1.6. Les lignes virtuelles montrent le mouvement planifié du bras du robot. 9
Fig. 1.7. La réalité augmentée pour la conception et l'urbanisme . 9
Fig. 1.8. Vue à travers HMD 10
Fig. 1.9. Système de réalité augmentée basé sur vue à travers vidéo. 11
Fig. 1.10. Google glass 11
Fig. 1.11. Système de réalité augmentée basé sur vue à travers optique. 12
Fig. 1.12. Exemple des marqueurs : (a) ARtoolKit, (b) ARTag. 12
Fig. 1.13. Réalité augmentée basé sur des marqueurs (ARtoolKit) 13
Fig. 1.14. Réalité augmentée basée sur des primitive visuelles 13
Fig. 1.15. Le schéma global du système à développer. 14
Fig. 2.1. Différents types de coins 17
Fig. 2.2. Les transformations euclidiennes 18
Fig. 2.3. Transformations de type similitude 18
Fig. 2.4. Les transformations affines 19
Fig. 2.5. Exemple changement de projective de perspective 19
Fig. 2.6. Les différentes situations considérées par le détecteur de Moravec 20
Fig. 2.7. Schéma simplifié de l'analyse des valeurs propres. 22
Fig. 2.8. Points d'intérêt détectés par Harris. 23
Fig. 2.9. Détecteur FAST 23
Fig. 2.10. Principe de l'image intégrale 24
Fig. 2.11. Dérivées partielles secondes 25
Fig. 2.12. Détection multi-échelles 25
Fig. 2.13. Passage du filtre 9x9 au filtre 15x15. 26
Fig. 2.14. Points d'intérêt détectés par la méthode SURF. 26

Fig. 2.15. Construction des vecteurs descripteurs. 27
Fig. 2.16.Images Gaussiennes groupées par octaves 28
Fig. 2.17. Différences de gaussiennes 28
Fig. 2.18. Recherche des extrema 29
Fig. 2.19. Vecteur de descripteur des points d'intérêt 29
Fig. 2.20. Ondelettes de Haar suivant les directions x et y (zone noire: -1, zone blanche: +1). 30
Fig.2.21. Détection de l'orientation principale 30
Fig. 2.22. Fenêtre de calcul du descripteur 31
Fig. 2.23. Calcul des éléments du descripteur 31
Fig. 2.24. Différents composants du vecteur de paramètres 32
Fig. 2.25. Illustration du motif du cinq échantillonnage 33
Fig. 2.26. La performance des cinq modèles d'échantillonnages 34
Fig. 2.27. Illustration de calcul d'angle de rotation 35
Fig. 2.28. Schéma global un système de suivi d'objet 36
Fig. 2.29. Exemple de l'appariement entre deux images utilisant ORB 37
Fig. 2.30. Ajustement d'une ligne avec les données correctes en utilisant RANSAC 38
Fig.3.1. Cycle de développement en « V » 41
Fig. 3.2. Diagramme de cas d'utilisation global 47
Fig. 3.3. Diagramme de cas d'utilisation augmentation d'une scène. 47
Fig. 3.4. Diagramme des cas d'utilisation de détection et de reconnaissance. 48
Fig. 3.5. Diagramme des cas d'utilisation de configuration et de calibrage de caméra. 49
Fig. 3.6. Diagramme de classe globale. 50
Fig. 3.7. Diagramme de séquence globale. 51
Fig. 3.8. Diagramme de séquence choix et configuration de caméra. 52
Fig. 3.9. Diagramme de séquence détection et suivi d'objet. 53
Fig. 3.10. Diagramme de séquence augmentation de la scène. 54
Fig. 4.1. Modèle du logo de l'Université de Blida sous 3DsMax. 57
Fig. 4.2. Extraction des points d'intérêt d'une scène réelle utilisant ORB. 57
Fig. 4.3. Reconnaissance dans la scène réelle. 58
Fig. 4.4. Le suivi d'objet basé sur la mise en correspondance. 58
Fig. 4.5. Augmentation de scène réelle par l'insertion d'un objet virtuel. 59

Fig. 4.6. Images de référence utilisées 60
Fig. 4.7. Temps de traitement en fonction de nombre de points. 60
Fig. 4.8. Score de répétabilité pour différentes séquences d'images. 62
Fig. 4.9. Evaluation d'exactitude et de précision. 63

Liste des tableaux

Tableau 3.1 Identification des cas d'utilisation..... 46

Tableau 4.1 Comparaison entre SIFT et SURF et ORB en termes de temps de traitement.... 61

Liste des abréviations

BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CDTA	Centre de Développement des Technologies Avancées
DLL	Dynamic Link Library
FAST	Features from Accelerated Segment Test
FREAK	Fast REtinA Keypoint
HMD	Head Mounted Display
IDE	Integrated Development Environment
IRM	Imagerie par Résonance Magnétique
IRVA	Interaction homme système pour la Réalité Virtuelle/Augmentée
KARMA	Knowledge-based Augmented Reality for Maintenance Assistance
ORB	Oriented FAST and Rotated BRIEF
PC	Personnel Computer
PI	Point d'Intérêt
RA	Réalité Augmentée
RANSAC	RANdom SAmple Consensus
RM	Réalité Mixte
RV	Réalité Virtuelle
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
UML	Unified Modeling Language
VA	Virtualité Augmentée

Introduction générale

La convergence de la recherche vers la réalité augmentée paraît logique, suite à la puissance croissante des ordinateurs et la maîtrise presque totale du domaine de traitement d'images et des séquences vidéo. Dans ce sens, l'être humain s'intéresse toujours à enrichir son environnement d'action par des informations supplémentaires avec lesquelles il peut être plus réactif et plus efficace.

Par l'intermédiaire des outils informatiques, l'utilisateur vise à combiner les informations issues de l'environnement informatique et celles issues de l'environnement physique [1]. La combinaison de ces deux environnements a permis la naissance d'un nouvel axe de recherche appelé « Réalité Augmentée ». Cette technologie est basée sur la combinaison d'un monde réel perçu par l'utilisateur et d'une scène virtuelle générée par ordinateur. Elle est employée dans plusieurs domaines par exemple : médecine, architecture, maintenance, robotique, etc.

L'objectif de la réalité augmentée est d'apporter un réalisme et une cohérence visuelle entre le flux réel et virtuel. Afin de maintenir le recalage dynamique des objets de synthèse sur le monde réel, les systèmes d'acquisition doivent en permanence calculer le point de vue de l'opérateur pour donner l'illusion que ces objets virtuels appartiennent au monde réel. Le positionnement de ces objets dans la scène nécessite de connaître la position et l'orientation de la caméra par rapport à un repère lié au monde [2]. Ainsi, un bon système de réalité augmentée est un système qui permet de garder à tout moment un alignement correct entre les objets virtuels et les objets réels. Ceci est possible, grâce à un bon suivi de la position et de l'orientation des objets réels de la scène.

De nombreuses approches ont été proposées pour la reconnaissance et le suivi dans les applications de réalité augmentée elles sont regroupées en deux catégories :

- Les approches basées sur marqueurs, utilisant des cibles codées qui sont insérées dans la scène réelle.
- Les approches sans marqueur par l'extraction des primitives visuelles à partir de la scène réelle.

Les points d'intérêt sont des primitives visuelles pertinentes pour détecter et reconnaitre un objet dans une séquence vidéo, en introduisant des caractéristiques visuelles certes moins intuitives que les régions ou les segments de droites. Plusieurs méthodes de suivi d'objet rigide basées sur les

points d'intérêt ont été proposées comme les méthodes : SIFT (Scale Invariant Feature Transform) et SURF (Speeded Up Robust Features), mais elles souffrent la complexité de calcul et certaines limitations dans ses performances.

Notre Projet de Fin d'Etude s'inscrit dans le cadre du projet «Interaction 3D multimodale et collaborative dans un environnement de réalité virtuelle et augmentée» initié par l'équipe IRVA (Interaction homme système Réalité Virtuelle et Augmentée) au sein de la Division Robotique et Productique du Centre de Développement des Technologies Avancées (CDTA).

L'objectif de ce travail c'est d'améliorer les performances de système de reconnaissance et suivi d'objet rigide par l'utilisation des nouveaux descripteurs des points d'intérêt comme ORB (Oriented FAST and Rotated BRIEF). Ce dernier est appliqué à la réalité augmentée pour recaler correctement les objets virtuels dans la scène réelle.

Ce mémoire sera présenté comme suit :

Dans le premier chapitre, nous présentons un état de l'art sur : la réalité augmentée, et leur différentes applications, les différentes dispositifs d'un système de réalité augmentée et leur problématique.

Le deuxième chapitre fait une étude théorique sur : les différentes méthodes de détection des points d'intérêt, la description et la mise en correspondance pour la reconnaissance et le suivi d'objet.

Dans le troisième chapitre nous présentons la conception et la mise en place de la solution, avec le détail des différentes étapes nécessitent pour la fonctionnalité d'un système de réalité augmentée.

Le quatrième chapitre concerne l'évaluation expérimentale et les résultats des différents tests. Nous comparons entre la méthode de suivi utilisant le descripteur ORB et les méthodes SIFT, SURF, avec l'insertion des objets virtuels pour la réalisation d'un système de réalité augmentée complet.

Enfin, notre mémoire sera terminé par une conclusion générale et des perspectives pour les travaux futurs, ainsi que deux annexes.

La RA est vue selon Robinett [5] comme un moyen d'augmenter les sens de l'utilisateur, de transformer des événements imperceptibles en phénomènes visibles, audibles ou touchables. C'est vu comme une symbiose entre l'homme et la machine comme jamais auparavant.

De son côté, Milgram définit la RA comme l'ensemble des cas où un environnement réel est Augmenté au moyen d'objets virtuels (graphiques générés par ordinateur) [6].

Pour Azuma [7] un système de RA complète le monde réel avec des objets virtuels (générés par ordinateur) qui semblent coexister dans le même espace que le monde réel comme illustre la Figure 1.1. Un système de RA a les propriétés suivantes :

- Il combine des objets réels et virtuels dans un environnement réel.
- Il fonctionne de manière interactive, en temps réel.
- il fait coïncider les objets réels avec les objets virtuels.

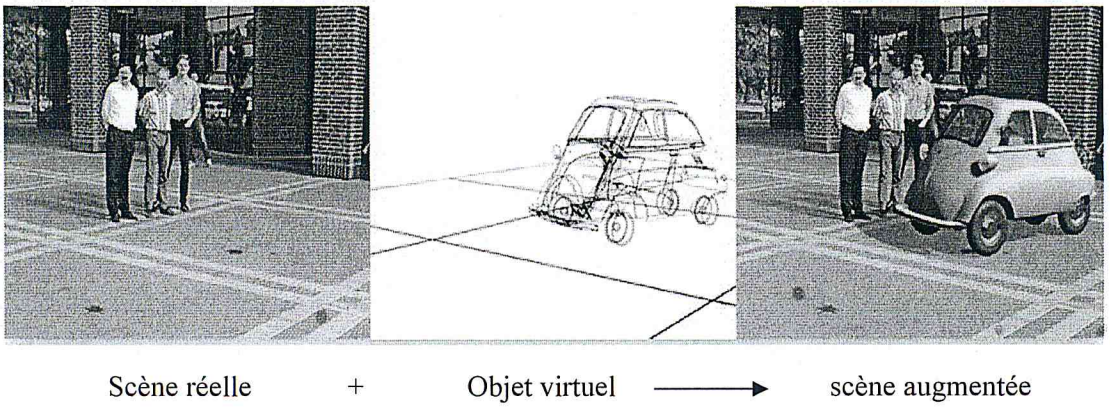


Fig. 1.1. Exemple de la réalité augmentée.

1.3. La réalité virtuelle

La réalité virtuelle (RV) est une façon pour l'humain de visualiser, manipuler, interagir avec des données complexes à l'aide d'un ordinateur. La RV joue un rôle primordial dans la simulation d'interactions 3D avancées entre l'homme et l'ordinateur [8].

Le terme réalité virtuelle a été introduit par " Jaron Lanier " pour définir :"Un environnement 3D, interactif, généré par ordinateur dans lequel l'utilisateur est Complètement immergé". Cette définition comporte deux points principaux :

- L'environnement virtuel est un environnement tridimensionnel, généré par ordinateur, ce qui nécessite une performance élevée pour assurer le réalisme de la scène.
- On parle d'un environnement interactif par l'utilisation, car l'utilisateur a besoin de réponses en temps réel.

1.3.1. La réalité virtuelle et la réalité augmentée

Par opposition à la RV où une scène est entièrement représentée en images de synthèse, la RA consiste à faire coexister des entités virtuelles du monde numérique avec des scènes du monde réel de manière homogène et consistante [9].

La réalité augmente complète le monde réel au lieu de le remplacer, par opposition à la réalité virtuel qui le synthétise complètement. En général, la RA hérite plusieurs caractéristiques de RV donc elles possèdent des points communs, par exemple, elles ont la même architecture du système (générateur de scènes, dispositifs d'affichage, etc.) ainsi que des caractéristiques tel que interactivité en temps réel.

1.3.2. Continuum réel – virtuel 'La réalité mixte'

La réalité mixte (RM) qualifie un dispositif interactif dans lequel des objets réels et des données informatiques sont mêlées. L'ensemble est perçu de manière cohérente par l'utilisateur par un ou plusieurs de ses sens ce qui lui permet d'interagir avec les différents objets réels ou virtuels de manière cohérente.

Milgram [6], [10] décrit un espace, appelé Continuum de réalité-virtualité présenté dans la Figure 1.2, dont les deux extrémités correspondent à la réalité et à la virtualité pures, et au milieu on y trouve la réalité mixte qui englobe la réalité augmentée et la virtualité augmentée, telle que ces dernières se distinguent par leurs environnements dominants. Dans le cas de la réalité augmentée c'est le réel et dans le cas de la virtualité augmentée c'est le virtuel qui domine :

- Virtualité augmentée : elle inclue des éléments de l'environnement réel dans une interface purement virtuelle dans le but de rehausser l'interaction de l'utilisateur avec le monde virtuelle grâce à ces entités réelles.
- Réalité augmentée : contrairement à la virtualité augmentée, la RA permet d'introduire des éléments virtuels dans une scène réelle afin de mieux interagir l'utilisateur avec le monde réel grâce à ces entités virtuelles.

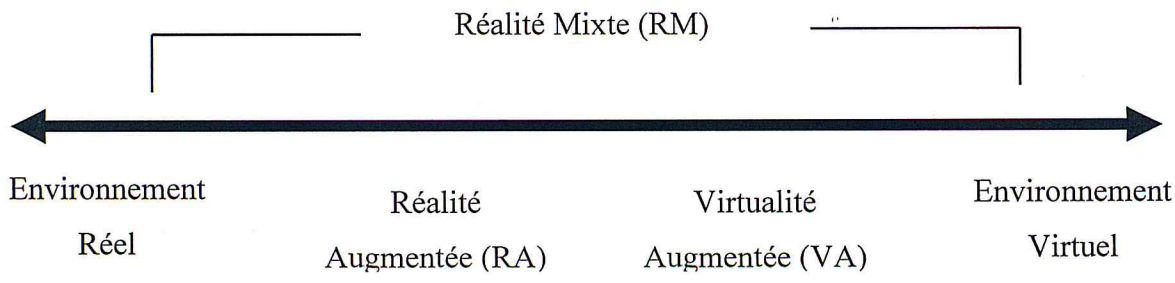


Fig. 1.2. Le "Continuum réel – virtuel" de Milgram [6].

1.4. Principe et problématique de l'augmentation

D'une manière générale, pour augmenter une scène, on doit disposer d'une caméra (ou plus, en stéréovision) aménagée relativement à un repère.

Sa calibration consiste à déterminer, Géométriquement, ses propriétés optiques ainsi que sa position et son orientation. La scène filmée dispose également de son propre repère, où devront être connues les positions de certains objets réels. Les traitements d'augmentation doivent prendre en compte les temps de latence [11], variables et grands, des outils qui composent le système.

Leurs dissemblances (fiabilité, fréquence, nature, etc.) Nécessitent des corrections spatiales et temporelles. En plus, L'imprécision inévitable des parties mécaniques est un autre point à considérer.

C'est pour ces raisons que d'autres sondes, tels que des télémètres laser ou des capteurs magnétiques et même des interventions humaines sont parfois nécessaires. Ils servent à fixer ou connaître la position 'absolue' de la caméra et des objets réels [12].

L'incrustation des objets virtuels se fait alors selon des principes de projections géométriques.

Le calcul de la matrice H (homographie planaire) est fait pour chaque image de la séquence.

Elle correspond à la solution de l'équation $(x\ y\ w)^t = H\ (x'\ y'\ w)^t$, dans le cas d'une projection plane 2D, qui peut être résolue par la méthode des valeurs singulières. Dans cette équation $(x'\ y'\ w)$ sont les coordonnées homogènes d'un point X de l'objet virtuel relativement à son repère et $(x\ y\ w)$ ses coordonnées homogènes dans le repère de l'image.

La résolution de cette équation nécessite la connaissance de quatre points dans le repère de l'image.

Ainsi, plusieurs problèmes se posent lorsqu'on tente d'incruster des objets virtuels dans des images réelles :

1.4.1. Alignement des caméras réelle et virtuelle

Le premier problème est de faire correspondre la perspective de l'objet virtuel avec celle de la scène réelle. Ce problème est connu sous le nom d'alignement des caméras réelle et virtuelle.

Pour le résoudre, il faut d'abord retrouver les propriétés de la caméra réelle ayant donné lieu à L'observation, ensuite à calculer les images synthétiques en utilisant une caméra virtuelle Reprenant ces propriétés. La Figure 1.3.b illustre un exemple de résultat obtenu lorsque l'image de la voiture respecte la perspective réelle. Par contre, la Figure 1.3.a montré le résultat lors d'une intégration quelconque de cette image. Le résultat obtenu en Figure 1.3.b ne suffit pas à obtenir une image réaliste : la voiture n'est pas correctement éclairée et l'arrière du véhicule devrait être occulté par le bâtiment photographié et non pas être projetée par-dessus l'édifice.

1.4.2. Cohérence spatio-temporelle

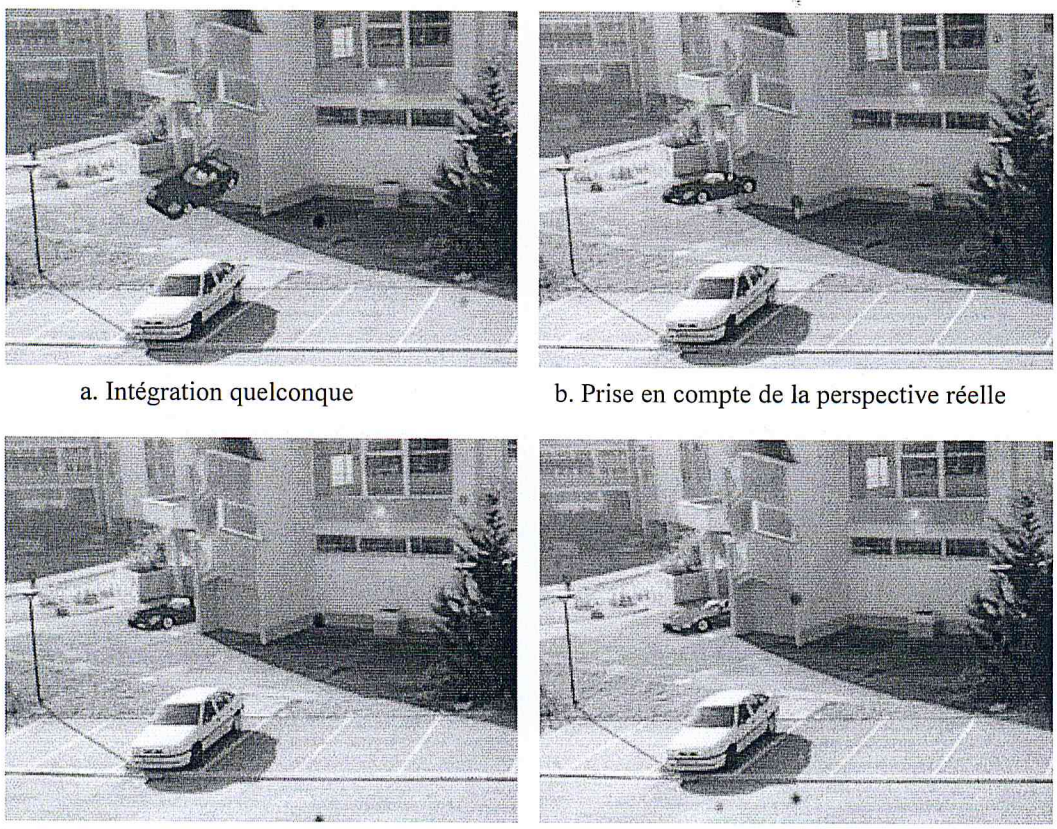
Les déplacements des objets virtuels dans la scène réelle et les occultations qui peuvent se

Produire entre objets virtuels et réels constitue le problème de cohérence spatio-temporelle.

La Figure 1.3.c montre un résultat de composition où ce problème est pris en charge.

1.4.3. Cohérence photométrique

Enfin, la prise en compte des inter-réflexions lumineuses (ombres, reflets) entre les objets réels et virtuels est du ressort de la cohérence photométrique. Le résultat obtenu en Figure1.3.d tient compte de ce problème.



a. Intégration quelconque b. Prise en compte de la perspective réelle

c. Prise en compte des contraintes spatio-temporel d. Prise en compte des contraintes photométriques

Fig. 1.3. Augmentations par le véhicule rouge [13].

1.5. Les applications de réalité augmentée

Les applications potentielles de la RA sont multiples : nous trouvons en particulier des applications dans le domaine de la médecine, de la robotique et la télé-robotique, de l'assemblage et la maintenance des objets complexes, de la conception (engineering design), des systèmes mobiles et des applications d'extérieurs et des développements commerciaux [14], [15], nous avons cité quelques applications :

1.5.1. La médecine

L'une des applications primaires de la réalité augmentée est dans le domaine médical appelé Chirurgie guidée par l'image (image guided surgery) comme est illustré dans la Figure 1.5. La RA

peut être utilisé par les médecins pour visualiser les données 3D extraites chez un patient par-dessus le corps du malade (images ultrasonores, tomographie 3D, Imagerie par Résonance Magnétique (IRM) [16].

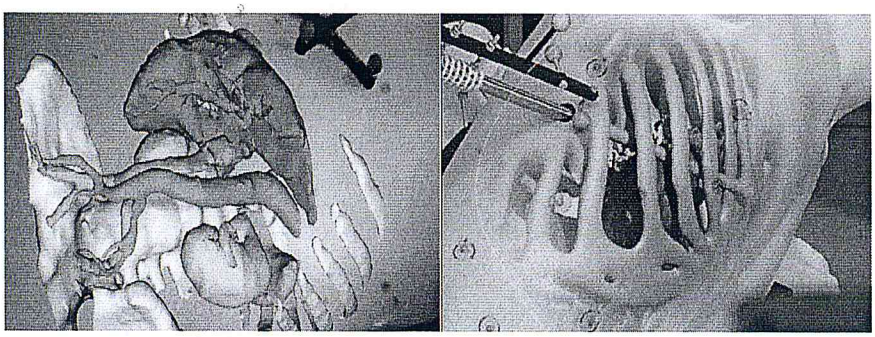


Fig. 1.4. Visualisation 3D utilisant la RA dans le domaine médicale [17].

1.5.2. La maintenance

L'assistance pour la fabrication, la maintenance ou la réparation est un autre domaine où la réalité augmentée est appréciée. Elle consiste à fournir des animations qui peuvent être superposées aux équipements, montrant leur manipulation. L'objectif, dans ce cadre, étant celui de remplacer les manuels (documentation) [11].

Au début des années 1990, Steven Feiner travaillait sur le projet KARMA (Knowledge-based Augmented Reality for Maintenance Assistance) [19], c'est le premier projet de maintenance basé sur la RA, ce dernier avait pour but de guider un opérateur dans le cadre de la réalisation de tâches simples de maintenance d'une imprimante laser comme montre la Figure 1.5.

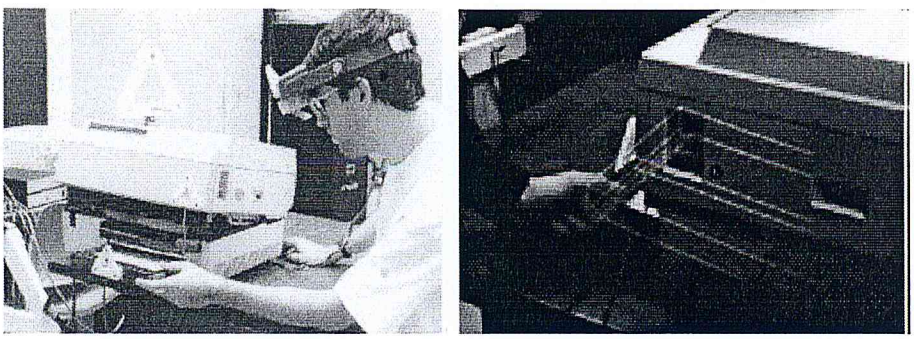


Fig. 1.5. Application de la RA pour la maintenance d'une imprimante [19].

1.5.3. La robotique

La télé-opération d'un robot est difficile surtout lorsqu'il est éloigné. Les retards de communication causés par la distance nécessitent l'exécution des actions demandées sur une version virtuelle avant de les contrôler directement sur un robot réel. Cette augmentation permet de corriger les erreurs des

actions demandées afin d'éviter des perturbations du comportement du robot comme est indiqué dans la Figure 1.6 [20].

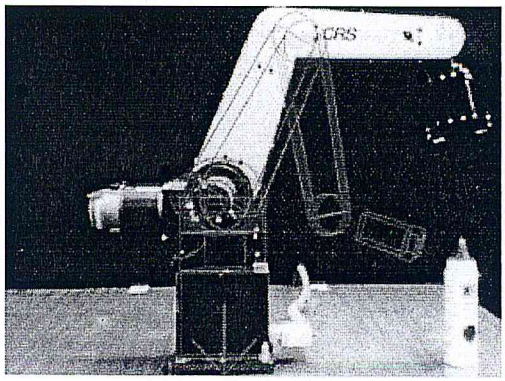


Fig. 1.6. Les lignes virtuelles montrent le mouvement planifié du bras du robot [20].

1.5.4. L'architecture

L'architecture est particulièrement concernée par les applications de la réalité augmentée. En effet, les produits mis au point permettent de visualiser les maquettes à des échelles diverses et de voir aussi à l'intérieur et à l'extérieur des bâtiments. La réalité augmentée offre la possibilité de visualiser la conception d'une construction dans son futur contexte réel comme est montré dans la Figure1.7 [21]. Elle permet d'évaluer l'esthétique et les effets des changements apportés aux contextes réels. La réalité augmentée permet de visualiser les futurs travaux de restauration et de rénovation.

Fig. 1.7. La réalité augmentée pour la conception et l'urbanisme [21].

1.6. Les composantes d'un système de réalité augmentée

Il existe de nombreux types de périphériques d'entrée pour les systèmes de RA. Certains systèmes, tels que le système mobile augmenté de Reitmayr utilise des gants. D'autres, comme ReachMedia utilisent un bracelet sans fil. Dans le cas des Smartphones, le téléphone lui-même peut être utilisé comme un dispositif de pointage, par exemple, Google SkyMap sur téléphone Android, l'utilisateur doit diriger son téléphone dans le sens des étoiles ou des planètes pour connaître leurs nom. Les dispositifs d'entrée choisie dépendent en grande partie du type d'application du système en cours d'élaboration et du type d'affichage choisir [14].

Plusieurs dispositifs et capteurs sont utilisés pour réaliser un système de réalité augmentée. Néanmoins, les dispositifs les plus employés sont : les systèmes d'affichage qui jouent un rôle primordial pour le rendu visuel et la génération des scènes virtuelles et les systèmes de suivi nécessaires à la localisation dans le repère du monde. La plupart des capteurs utilisés en réalité augmentée sont dédiés pour des applications en intérieur. Cependant, de nouvelles technologies nécessitant l'utilisation de nouveaux dispositifs sont destinées pour des applications en extérieur ou sans fil.

Les systèmes d'affichage sont utilisés pour visualiser des objets virtuels. Des dispositifs d'affichage dédiés permettent alors de mixer le réel et le virtuel. On peut distinguer différentes classes de systèmes :

1.6.1. Affichage basé sur la configuration moniteur

Les systèmes basés moniteurs offrent à l'opérateur la possibilité d'observer le monde réel et les objets virtuels superposés à sa vue sans être équipé pour autant de lunettes spéciales. La visualisation est donc faite directement à travers un écran, à l'aide d'une caméra couplée à cet élément.

L'avantage de cette méthode est sa simplicité et son accessibilité, car elle requiert un PC (Personnel Computer) et une caméra. En plus, traiter chaque image individuellement, permet au système d'augmentation d'utiliser des approches basées vision pour extraire des informations associées à l'utilisateur (position et orientation), et ce pour accomplir le processus de recalage. Cependant cette simplicité est acquise au dépriment du réalisme de la scène du monde. Il est clair que voir le monde réel à travers un petit moniteur, limite le réalisme et la mobilité du monde augmenté. En plus, chaque image capturée avec la camera doit être traitée individuellement par le système d'augmentation, donc il y a un délai potentiel entre le moment ou l'image a été capturée par la camera et le moment ou cette même image augmentée est affichée sur le moniteur.

1.6.2. Affichage basé sur la configuration vue à travers vidéo

Dans le but d'augmenter le sens d'immersion dans les systèmes de réalité virtuelle, les HMDs (Head Mounted Display) qui couvrent complètement la vue de l'utilisateur sont développés par Sutherland en 1965 [22], la Figure 1.8 montre un exemple de HMD (5DT HMD 800).

Fig. 1.8. Vue à travers HMD [23].

La Figure 1.9 montre un schéma représentant un système de réalité augmentée basé sur vue à travers vidéo. Dans cette configuration, l'utilisateur ne voit pas directement le monde réel, mais il voit ce que le système affiche dans des moniteurs minuscules à l'intérieur du HMD.

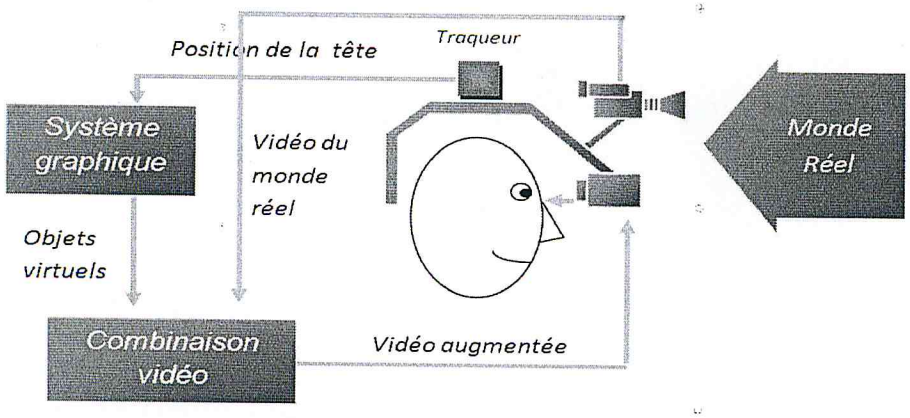


Fig. 1.9. Système de réalité augmentée basé sur vue à travers vidéo.

1.6.3. Affichage basé sur la configuration vue à optique

Il existe un autre type d'HMD utilisé pour la réalité augmentée qui est un système d'affichage basé sur vue à travers optique qui est représenté par la Figure 1.10, un exemple de la lunette Google.
Dans cette configuration l'utilisation de verres semi transparents permet d'une part de voir directement le monde réel et d'autre part de réfléchir les objets virtuels générés par le système d'augmentation.

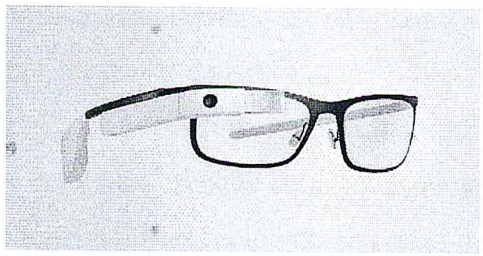


Fig. 1.10. Google glass [24].

Donc, quand l'utilisateur bouge la tête, les objets virtuels maintiennent leurs positions comme s'ils étaient une part du monde réel. Contrairement aux HMDs vidéo vue à travers, ces HMDs ne limitent pas la résolution d'affichage et ne consomment pas de temps pour accomplir l'augmentation car le monde réel est vu directement à travers les verres. Cependant, la qualité des objets virtuels est limitée par la vitesse du traitement et les capacités graphiques du système d'augmentation. Par conséquent, la création d'augmentations convaincantes est difficile car le monde réel paraîtra naturel alors que les objets virtuels paraîtront pixélisés.

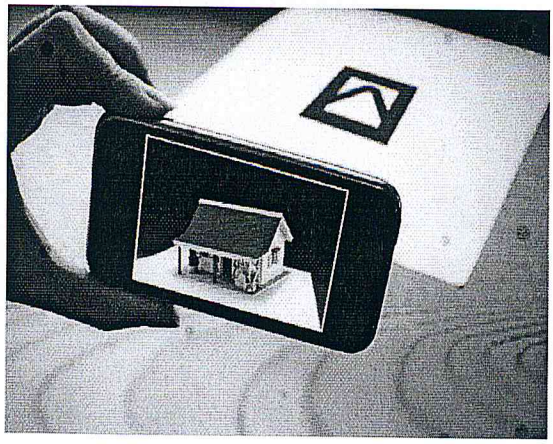


Fig. 1.13. Réalité augmentée basé sur des marqueurs (ARtoolKit) [26].

1.7.2. Réalité augmentée à partir de primitives naturelles

Cette approche se base sur la reconnaissance d'entités (cibles) naturelles (points d'intérêt, segments, cercles, droites, etc.) de la scène réelle, par différentes techniques. A partir des coordonnées 2D de la cible reconnue, la position et orientation de la camera est déterminée par rapport à un repère dans la scène. Par la suite, l'objet virtuel est recalé par rapport à la cible reconnue.

L'utilisation de marqueurs en environnement naturel est cependant, en pratique, peu réaliste. Les méthodes de calcul de point de vue ont donc été adaptées au cas d'environnements quelconques.

L'un des avantages de ces méthodes est leur robustesse face aux occultations partielles, aux variations d'éclairage, aux mouvements relativement importants de la caméra, etc. Ceci est dû d'une part à des algorithmes efficaces pour gérer les appariements locaux et d'autre part à l'utilisation d'estimateurs robustes dans le processus de minimisation. La Figure 1.14 montre une augmentation de la scène réelle utilisant des primitives visuelles.

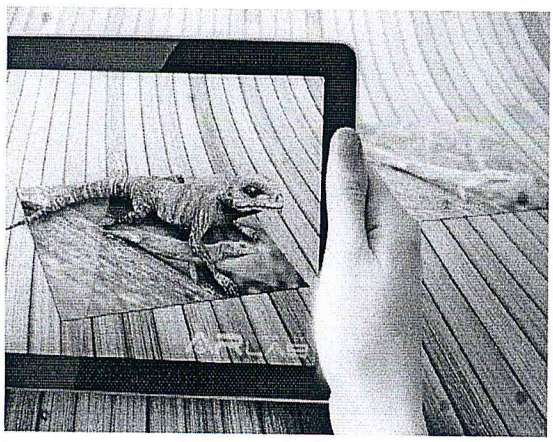


Fig. 1.14. Réalité augmentée basée sur des primitive visuelles [26].

1.8. Schéma global du système de réalité augmentée à développer

Notre travail consiste à développer un système de recalage pour une application de réalité augmentée basée sur les points d'intérêts (PI) (RA sans marqueurs). A cause de certaines limitations de régalage à base des marqueurs, comme par exemple : la sensibilité des marqueurs au changement de lumination, difficile à placer dans les endroits complexes, comme les applications de l'architecture, la médecine, etc. Ainsi qu'ils sont sensibles aux occlusions partielles.
Pour réaliser un système de recalage à base des primitive visuelles, deux parties seront développées : la reconnaissance d'une cible naturelle dans une scène réelle en temps réel, ainsi que calibrage de camera et de suivi d'objets. L'augmentation des objets virtuels se fait par le calcule de la matrice de transformation. La Figure 1.15 montre le schéma global du système de RA à développer.

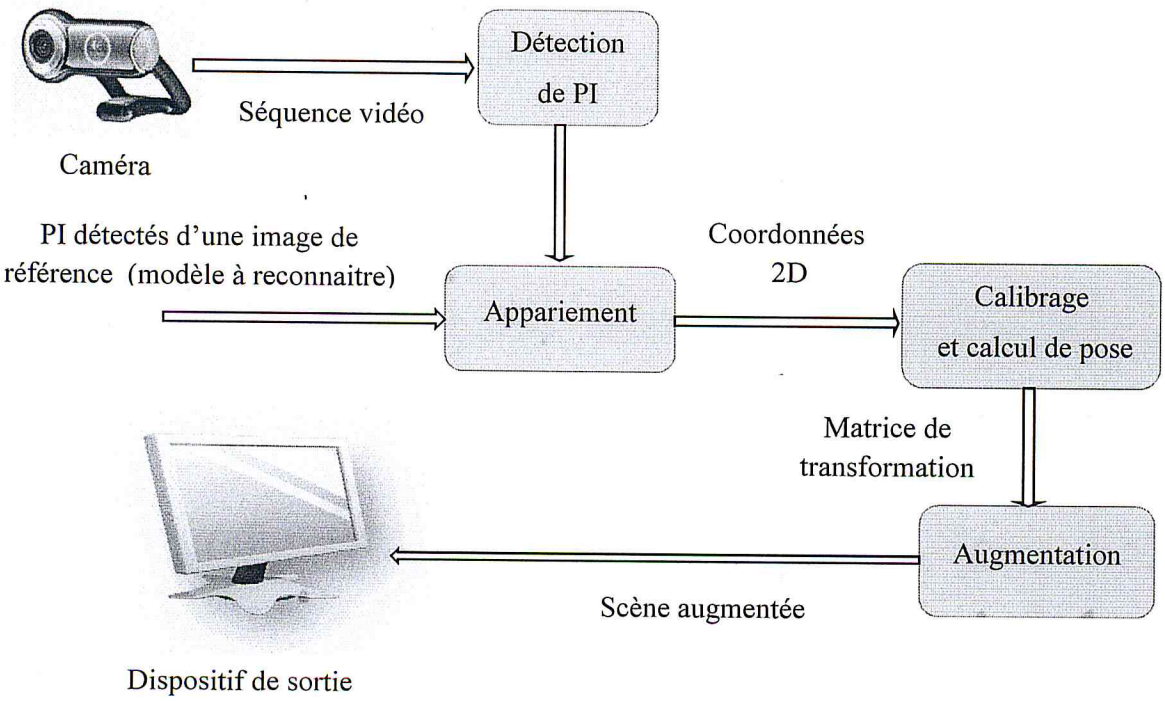


Fig. 1.15. Le schéma global du système à développer.

Un dispositif d'entré (camera) fournit un flux vidéo en temps réel. Chaque image de la séquence vidéo est traitée par un algorithme de détection de points d'intérêt en vue de reconnaitre des objets naturels après appariement des points détectés avec ceux des modèles préenregistrés. La reconnaissance et le suivi de ces objets fournie les coordonnées des points nécessaires pour le calibrage de camera et le calcul de pose, en sortie de ce dernier, nous aurons la matrice de transformation utilisée pour augmenter la scène qui sera transmise au dispositif de sortie dédié à cela, voir le détail sur le calibrage de caméra dans l'Annexe B.

1.9. Conclusion

Nous avons abordé dans ce chapitre les systèmes de réalité augmentée d'une manière globale, les différents domaines, les dispositifs nécessaires pour les réaliser un système de RA, et on a terminé avec le schéma global du système de réalité augmentée. Nous allons construire à partir de ce schéma, la suite de notre étude. Dans ce sens, nous allons traiter essentiellement les méthodes de la détection et la description des points d'intérêt, la mise en correspondance de ces points, ainsi que le suivi d'objet basé sur les PI.

Chapitre 2

Suivi d'objet rigide basé sur les points d'intérêt

2.1. Introduction

Le suivi des objets une étape préalable dans différents application, réalité augmentée, robotique, vision par ordinateur, etc. Différentes primitives visuelles sont utilisées (points, lignes, régions, etc.). Dans ces dernières années, la reconnaissance et le suivi d'objet à base des points d'intérêt, prends un part d'importante de l'étude, a cause des performances de ce dernier pour reconnaitre un objet. Ce chapitre consiste à introduire la notion de points et citer les différentes méthodes de la détection et description des points d'intérêt. Ainsi que la description d'un système de suivi d'objet rigide à base de la mise en correspondance des points d'intérêt.

2.2. Techniques de reconnaissance existantes :

Trois grandes techniques de reconnaissance Résumées par Schmid et Mohr [27], chacune différente de l'autre par les caractéristiques utilisées pour la reconnaissance et le suivi.
Plusieurs méthodes sont proposées dans la littérature comme nous allons le voir. Celle qui nous intéresse est la méthode de suivi d'objets naturels basée sur la détection de points d'intérêts.

2.2.1. Techniques basées sur les caractéristiques géométriques

Cette approche se base sur l'extraction de caractéristiques géométriques de l'image comme les lignes, les cercles et les rectangles pour reconnaitre les objets. Elle comprend l'appariement (mise en correspondance entre les caractéristiques d'un objet sur différentes images), le calcul de la position de l'objet et vérification. Cette approche n'est pas fiable lorsque l'objet en question n'a pas de forme géométrique spéciale tel un arbre et elle pose problème pour différentier entre plusieurs objets (appariement non discriminatoire).

2.2.2. Techniques basées sur les caractéristiques de luminance

Une alternative est proposée pour répondre aux manques constatés dans la première méthode, elle se base sur l'utilisation des informations sur la luminance d'un objet. L'idée est de ne pas imposer ce qui devrait être reconnu (cercles, triangles, etc.) mais plutôt d'exploiter les caractéristiques d'un objet dans l'image, ceci en utilisant un histogramme de couleurs. Des améliorations ont été ajoutées à cet histogramme pour éliminer la sensibilité aux changements de luminosité.

2.2.3. Techniques basées sur les points d'intérêt

Cette approche se base sur l'utilisation des points d'intérêts qui sont des caractéristiques locales (contours, région, coin, etc.) de l'image ayant une grande quantité d'information. Ils assurent l'invariance contre plusieurs changements dans l'image. C'est le sujet de ce qui va suivre.

2.3. Définition d'un point d'intérêt

La notion de points d'intérêt a été introduite pour la première fois par Moravec [28]. Pour lui, les points d'intérêts correspondent à un changement bidimensionnel du signal comme par exemple les coins, les jonctions, etc., comme illustre la Figure 2.1. Toutefois, un point d'intérêt est plus général qu'un coin, pourtant ils sont utilisés dans la littérature comme équivalents .

Les points référençant ces zones sont dénommés points d'intérêt. De nombreux travaux présentés dans la section 3.2 concernent la détection des points d'intérêt.

Informellement, un point d'intérêt est associé à une discontinuité des niveaux de gris (voir des couleurs), de la texture, de la géométrie, etc., de l'image. Il est souvent assimilé.

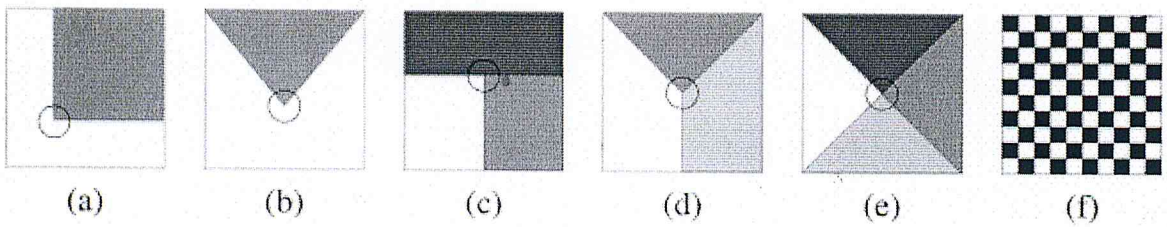


Fig. 2.1. Différents types de coins : (a) jonction en "L", (b) jonction en "V", (c) jonction en " T", (d) jonction en "Y ", (e) jonction en "X" et (f) jonction en "damier ".

Parmi les avantages d'utilisation des points d'intérêt, on trouve :

- Sources d'informations plus fiables que les contours car plus de contraintes sur la fonction d'intensité.
- Robuste aux occultations (soit occulté complètement, soit visible).
- Pas d'opérations de chainage.
- Présents dans une grande majorité d'images.

2.4. Les différents types de transformations d'image

Le choix d'un détecteur de points d'intérêt repose essentiellement sur l'utilisation souhaitée. Il sera en plus aisé de sélectionner le détecteur approprié afin de pallier aux différentes transformations de l'image. Les transformations étudiées sont réparties en quatre catégories [29] :

- Les transformations euclidiennes (ou rigides) : se composent de l'identité, de la rotation

Et de la translation. Elles préservent les angles, les distances et sont inversibles. La Figure 2.2 donne un aperçu des modifications de l'image obtenues par le biais de ces transformations.

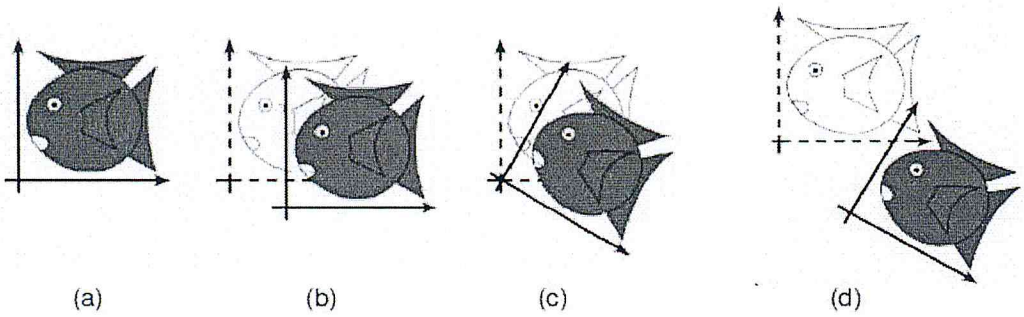


Fig. 2.2. Les transformations euclidiennes : (a) identité, (b) translation, (c) rotation, (d) exemple de transformation euclidienne [29].

- L'ajout du changement d'échelle isotrope aux précédentes transformations, permet d'obtenir Les similitudes présentées dans la Figure 2.3. Ces dernières préservent les angles, le rapport des longueurs et sont inversibles.

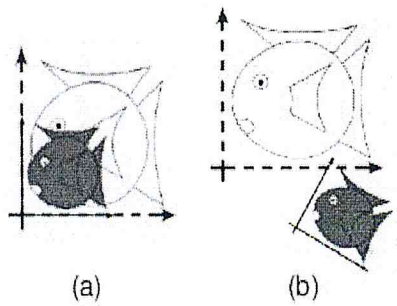


Fig. 2.3. Transformations de type similitude : (a) changement d'échelle isotrope, (b) exemple de similitude [29].

- La troisième catégorie correspond aux transformations affines. Ces dernières englobent les deux premiers types de modifications de l'image auxquels s'ajoutent, la réflexion, le changement d'échelle anisotrope et le "Shear". Cette catégorie conserve les parallèles et est inversible également. La Figure 2.4 représente les différentes transformations ajoutées, et la résultante qui en découle.

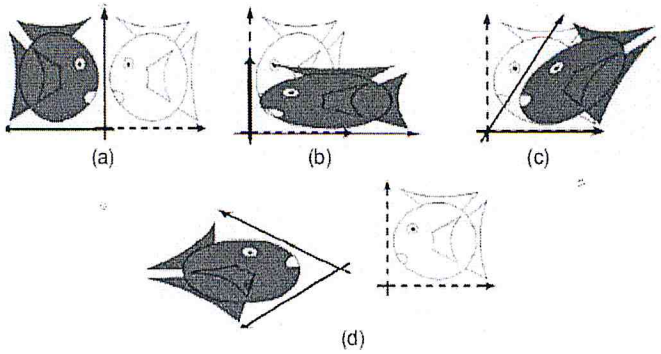


Fig. 2.4. Les transformations affines : (a) réflexion, (b) changement d'échelle anisotrope, (c) "Shear", (d) exemple de transformation affine [29].

- Une dernière catégorie, nommée transformations projectives, est obtenue en couplant l'ensemble des transformations précédentes avec une modification de la perspective de l'image. Elles préservent les droites mais ne conservent pas le barycentre. La Figure 2.5 représente une transformation perspective de l'image.

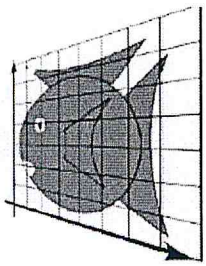


Fig. 2.5. Exemple changement de projective de perspective [29].

2.5. Détection de points d'intérêt

Les points d'intérêts sont des caractéristiques locales d'une image dans lesquelles le signal (en niveau de gris) est en deux dimensions [29]. Les points d'intérêts présentent beaucoup d'avantages par rapport aux autres caractéristiques comme les contours et les régions qui peuvent être exploitées dans une image. En particulier d'être robustes aux problèmes de visibilité et la quantité d'informations représentée, mais elle diffère d'une méthode à une autre.

Nous comptons plusieurs méthodes de détection de points d'intérêt, chronologiquement on peut citer les méthodes les plus importantes et qui peuvent être exploitée dans le domaine de la réalité.

2.5.1. Détecteur de Moravec

Elle consiste à quantifier la variation d'intensité en un point donné dans toutes les directions [29]. Pour chaque point de l'image on effectue un test en utilisant une fenêtre centrée autour (voisinage rectangulaire du point). Cette fenêtre est ensuite décalée dans les 8 directions à partir du point considéré. On calcule une corrélation sur les points entre la fenêtre initiale et la fenêtre décalée. On affecte au point la plus petite corrélation parmi les huit calculées. Un point est classé point d'intérêt

si la corrélation qui lui a été affectée est élevée, c'est-à-dire que la variation dans chacune des directions à partir de ce point est importante. La corrélation se calcule par l'équation suivante :

$$E(u,v) = \sum_{u,v} \omega(x,y)|I(x+u,y+v) - I(x,y)|^2 \qquad (2.1)$$

où :

$\omega(x,y)$ spécifie la fenêtre/voisinage considérée (valeur 1 à l'intérieur de la fenêtre et 0 à l'extérieur);

I est l'intensité au pixel (x,y);

$E(u,v)$ représente la moyenne du changement d'intensité lorsque la fenêtre est déplacée de (u,v).

En appliquant cette fonction dans les trois situations principales suivantes (voir la Figure 2.6), on obtient :

a. L'intensité est approximativement constante dans la zone image 1 considérée : la fonction E prendra alors de faibles valeurs dans toutes les directions (u,v).
b. La zone image 2 considérée contient un contour rectiligne : la E fonction prendra alors de faibles valeurs pour des deplacements (u,v) le long du contour et de fortes valeurs pour des déplacements perpendiculaires au contour.
c. La zone image 3 considérée contient un coin ou un point isolé : la E fonction prendra de fortes valeurs dans toutes les directions (u,v).

En conséquence, le principe du détecteur de Moravec est donc de rechercher les maxima locaux de la valeur minimale de E en chaque pixel (au dessus d'un certain seuil).

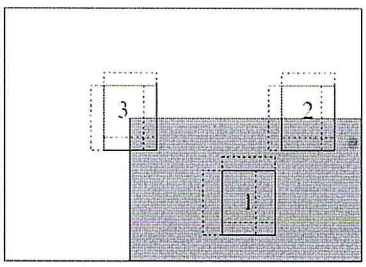


Fig. 2.6. Les différentes situations considérées par le détecteur de Moravec.

2.5.2. Détecteur de Harris

Le détecteur de Moravec fonctionne dans un contexte limité. Il souffre en effet de nombreuses limitations. Harris et Stephen [30] ont identifié certaines limitations et, en les corrigeant, en ont déduit un détecteur de coins très populaire : le *détecteur de Harris*. Les limitations du détecteur de Moravec prises en compte sont :

1. La réponse du détecteur est anisotropique en raison du caractère discret des directions de changement que l'on peut effectuer (des pas de 45 degrés). Pour améliorer cet aspect, il suffit de considérer le développement de Taylor de la fonction d'intensité au voisinage du pixel (x,y):

$$I(x+u,y+v) = I(x,y) + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + o(u^2,v^2) \quad (2.2)$$

D'ou

$$E(u,v) = \sum_{u,v} \omega(x,y)\left|u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + o(u^2,v^2))\right|^2 \quad (2.3)$$

En négligeant le terme $o(u^2,v^2)$ (valide pour les petits déplacements), on obtient l'expression analytique suivante:

$$E(u,v) = Au^2 + 2Cuv + Bv^2 \quad (2.4)$$

avec :

$$A = \frac{\partial I^2}{\partial x} * \omega(x,y) \quad (2.5)$$

$$B = \frac{\partial I^2}{\partial y} * \omega(x,y) \quad (2.6)$$

$$C = \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} * \omega(x,y) \quad (2.7)$$

où $*$ définie l'opérateur de convolution.

La réponse du détecteur de Moravec est bruitée en raison du voisinage considéré. Le filtre ω utilisé est en effet binaire (valeur 1 ou 0) et est appliqué sur un voisinage rectangulaire. Pour améliorer cela, Harris et Stephen propose d'utiliser un filtre Gaussien :

$$\omega(x,y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.7)$$

où σ^2 est la variance de la fenêtre gaussienne $\omega(x,y)$.

Enfin, le détecteur de Moravec répond de manière trop forte aux contours en raison du fait que seul le minimum de E est pris en compte en chaque pixel. Pour prendre en compte le comportement général de la fonction E localement, on écrit :

$$E(u,v) = (u,v).M.(u,v)^T \quad (2.8)$$

avec

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2.9)$$

La matrice M caractérise le comportement local de la fonction E, les valeurs propres de cette matrice correspondent en effet aux courbures principales associées à E:

- Si les deux courbures sont de faibles valeurs, alors la région considérée a une intensité approximativement constante.
- Si une des courbures est de forte valeur alors que l'autre est de faible valeur alors la région contient un contour.
- Si les deux courbures sont de fortes valeurs alors l'intensité varie fortement dans toutes les directions, ce qui caractérise un coin.

Par voie de conséquence, Harris et Stephen propose l'opérateur suivant pour détecter les coins dans une image :

$$R = det(M) - k.trace\,(M)^2 \tag{2.10}$$

avec

$$det(A) = AB - C^2 \tag{2.11}$$

$$trace(M) = A + B \tag{2.12}$$

le réel k est un constant de régulation, empiriquement k entre 0.04 et 0.06

Les valeurs propres de M correspondent aux courbures principales de la fonction E. Le détecteur de Harris se base sur l'hypothèse qu'un point est dit d'intérêt si les valeurs des deux courbures sont élevées. Ceci peut se caractériser par l'analyse des valeurs propres de M comme montre la Figure 2. 7, notées λ_1 et λ_2 avec $\lambda_1 \geq \lambda_2$:

- si $\lambda_1 = \lambda_2 = 0$: la zone sélectionnée est complètement uniforme (zone homogène).
- si $\lambda_1 > \lambda_2 = 0$: la zone correspond à un contour et le vecteur propre associé à λ_1 lui est perpendiculaire.
- si $\lambda_1 > \lambda_2 = \varepsilon$ (avec ε étant un seuil) : la zone caractérise un coin.

La figure 3 représente l'influence des valeurs propres de M sur le voisinage d'un point.

La Figure 2. 8 présente un exemple de détection des points d'intérêt (en rouge) par la méthode de Harris.

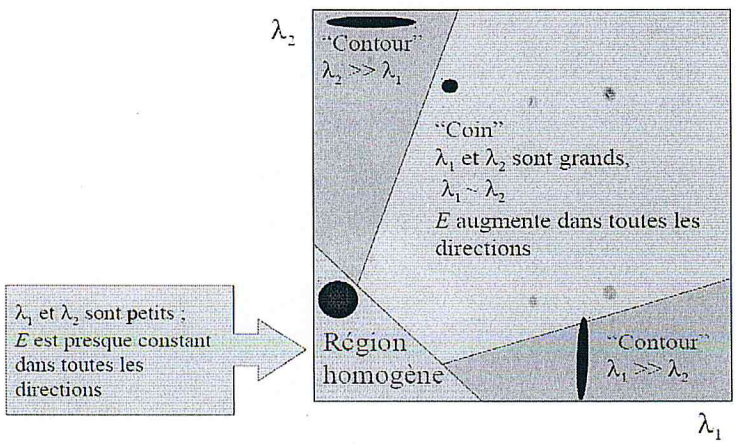


Fig. 2.7. Schéma simplifié de l'analyse des valeurs propres.

Fig. 2.8. Points d'intérêts détectés par Harris [31].

2.5.3. Détecteur FAST

Rosten et Drummond ont développé un détecteur très rapide appelé FAST (Features from Accelerated Segment Test) [32]. Cet algorithme opère en considérant un cercle de seize pixels autour du point candidat $P(x, y)$ comme il est illustré dans la Figure 2.9. Les pixels P_i à l'intérieur du cercle sont classés de la manière suivante :

- Foncée : si $I_{P_i} \leq I_P - T$
- Similaire : si $I_P - T < I_{P_i} \leq I_P + T$
- Brillante : si $I_P + T \leq I_{P_i}$

Où I_P est l'intensité du point candidat, I_{P_i} est l'intensité d'un pixel du cercle et T est un seuil utilisé pour la classification.

On considère un point P comme étant un coin s'il existe un ensemble de n pixels contigus à l'intérieur du cercle qui vérifie la première ou la dernière de conditions précédentes. L'algorithme a été encore accéléré en utilisant l'arbre de décision pour minimiser le nombre des tests en vue de la classification du pixel candidate [33].

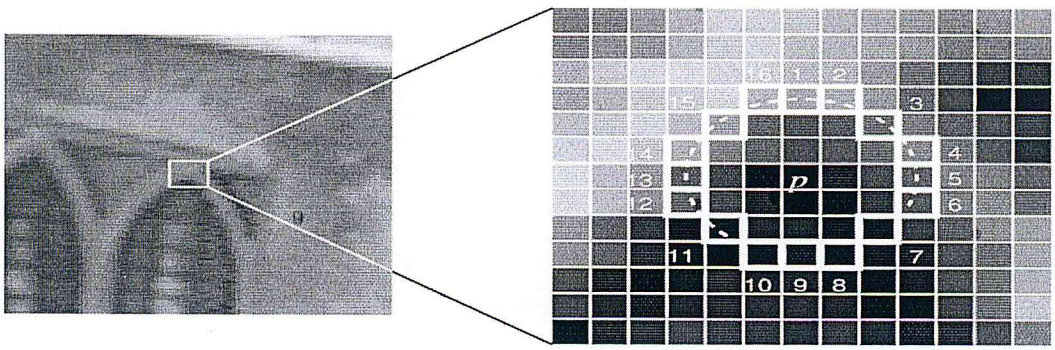


Fig. 2.9. Détecteur FAST [32].

2.5.4. Détecteur SURF

L'approche proposée par SURF (Speeded Up Robust Features) [34] utilise une approximation de la matrice Hessienne afin de détecter les structures de types « blobs». Elle utilise des images intégrales afin de diminuer fortement les temps de calculs car elles permettent le calcul rapide des convolutions avec les approximations de types « box-filters ».

La valeur d'une image intégrale en un point donné représente la somme de tous les pixels de l'image d'origine situés dans le rectangle formé par l'origine et le point considéré :

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{2.13}$$

Une fois que l'image intégrale a été calculée, il suffit de trois additions pour calculer la somme des intensités des pixels de n'importe quelle région rectangulaire de l'image d'origine, quelle que soit sa taille comme illustre la Figure 2.10.

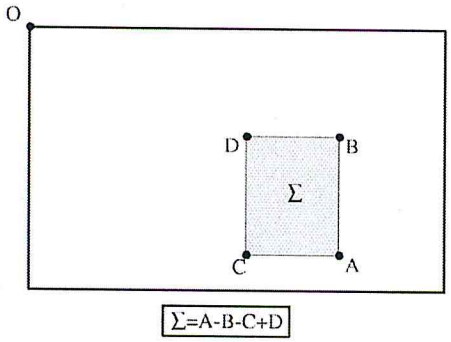


Fig. 2.10. Principe de l'image intégrale [34].

Le détecteur localise les blobs là où le déterminant de la matrice Hessienne atteint un maximum. Pour rappel, la matrice Hessienne (ou simplement la Hessienne) d'une fonction numérique f est la matrice carrée, notée $H(f)$, de ses dérivées partielles secondes.

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{2.14}$$

Dans le contexte du détecteur de point, la matrice Hessienne en un point $\mathrm{x} = (x, y)$ et à l'échelle σ est définie comme suit :

$$H(\mathrm{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathrm{x}, \sigma) & L_{xy}(\mathrm{x}, \sigma) \\ L_{xy}(\mathrm{x}, \sigma) & L_{yy}(\mathrm{x}, \sigma) \end{bmatrix} \tag{2.15}$$

Avec $L_{xx}(\mathrm{x}, \sigma)$ qui est le résultat de la convolution de la dérivée seconde de la gaussienne $\frac{\partial}{\partial x_1^2} g(\sigma)$ avec l'image au point x.

En pratique, la gaussienne doit être finie et discrétisée. Pour pouvoir tirer parti des images intégrales, Bays construise une approximation de type «*box*» des dérivées secondes de la gaussienne. Grâce aux images intégrales, le temps de calcul est indépendant de la taille du filtre. Sur la Figure 2.12, on peut voir les dérivées partielles de la gaussienne. D'abord finies et discrétisées (les deux images de gauche) et puis approximées par un *box filter* suivant les directions x et y. Les zones grises sont égales à zéro.

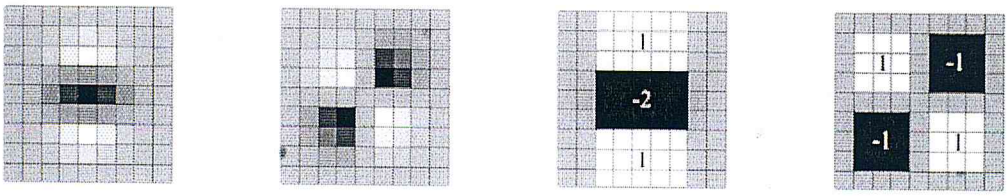


Fig. 2.11. Dérivées partielles secondes [34].

L'approximation du déterminant de la matrice hessienne calculée en un point x de l'image est stockée dans une «*blob response map*», puis les maxima locaux sont recherchés, afin de trouver des blobs.

Il est intéressant de pouvoir retrouver des points d'intérêt à différentes échelles afin de rendre le détecteur invariant aux changements d'échelle (le même objet peut être représenté en tailles différentes sur deux images). Cet aspect est souvent pris en compte en créant une pyramide d'images. Les images sont répétitivement filtrées avec une gaussienne puis sous-échantillonnées afin d'obtenir une image de plus petite taille. Chaque niveau de la pyramide représente une échelle différente. SURF peut procéder différemment grâce aux *box filters* et aux images intégrales. Au lieu d'appliquer successivement le même filtre à la sortie d'une image filtrée et sous-échantillonnée, on peut utiliser des *box filters* de diverses tailles directement sur l'image d'origine. Les « *blob response maps* » à différentes échelles sont donc construites en agrandissant le filtre plutôt qu'en réduisant itérativement la taille de l'image. Ceci permet d'une part de réduire le temps de calcul et d'autre part d'éviter l'aliasing dû au sous-échantillonnage de l'image. L'image de gauche de la Figure 2.12 représente la méthode classique avec sous-échantillonnage et filtre de taille constante. Sur l'image de droite les filtres sont de tailles variables.

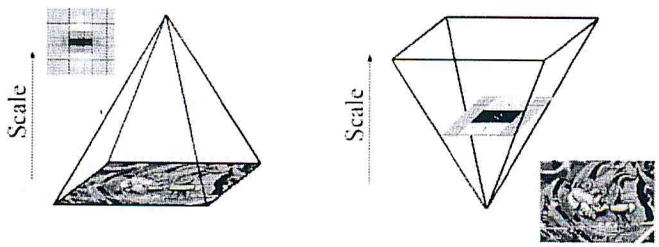


Fig. 2.12. Détection multi-échelles [34].

Les filtres représentés sur la Figure 2.11 ont une taille de 9x9. La taille des filtres est progressivement augmentée. Typiquement, on utilisera les filtres 9x9, 15x15, 21x21, 27x27, 39x39, 51x51, 75x75 et 99x99. On pourrait utiliser des filtres encore plus grands mais, en pratique, le nombre de points détectés décroit rapidement avec la taille du filtre.

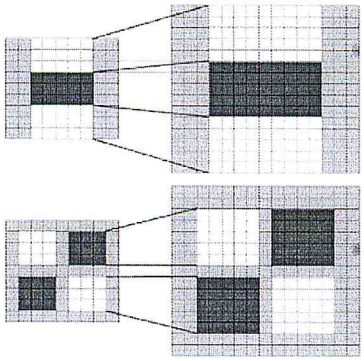


Fig. 2.13. Passage du filtre 9x9 au filtre 15x15.

En recherchant les maxima de la «*blob response map*» aux différentes échelles, on peut maintenant extraire la position et la taille des blobs dans l'image. Un exemple des points d'intérêt détectés est montré sur la Figure 2. 14, les cercles sont les échelles caractéristiques des points d'intérêt.

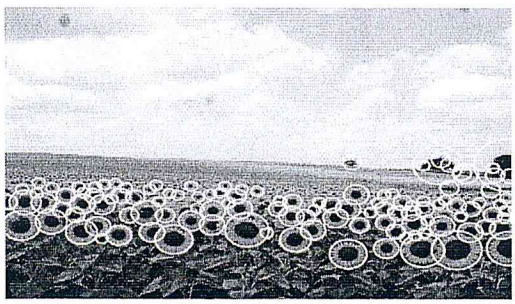


Fig. 2.14. Points d'intérêt détectés par la méthode SURF.

2.6. Description de points d'intérêt

Le rôle du descripteur est d'extraire à partir des données brutes de l'image, des informations exploitables par le système. Il existe plusieurs types de descripteurs. Le choix est conditionné par l'application visée puisque c'est elle qui détermine les informations utiles [34].

Un des problèmes centraux en interprétation d'images concerne le choix d'une représentation pertinente permettant d'accéder à des primitives significatives et fiables traduisant le contenu de l'image. En général, ces descripteurs sont regroupés en 3 classes : les descripteurs liés à la couleur (histogramme), les descripteurs de texture (matrice de cooccurrence, indices de direction principale et de rugosité, filtres de Gabor et ondelettes), et les descripteurs de formes (descripteurs de Fourier et moments invariants, points d'intérêt) [35].

Donc, un descripteur est une fonction qui est appliquée sur le patch (région autour d'un point-clé) afin de le décrire, et d'une manière qui est invariante pour tous changements sur l'image (par exemple, les changements indiquées dans la section 2.4, l'éclairage, le bruit, etc.) comme montre la Figure 2.15.

Dans notre travail nous intéressons par les descripteurs de formes de type points d'intérêt, nous citons quelques descripteurs qui basé sur les ondelettes de Haar comme exemple SURF, et les descripteurs binaires comme exemple ORB.

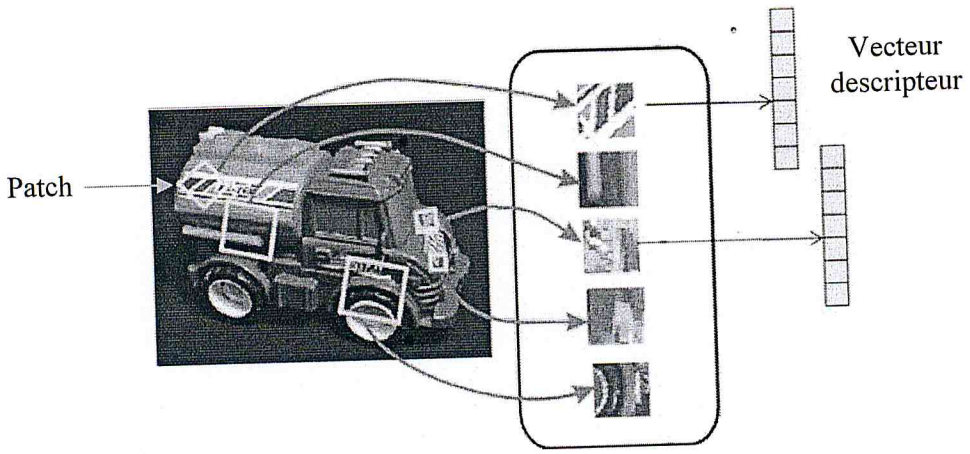


Fig. 2.15. Construction des vecteurs descripteurs.

2.6.1. Détecteur SIFT

L'algorithme SIFT (Scale Invariant Feature Transform) a été proposé par David Lowe [36], université de British Columbia, en 1999 pour détecter et décrire des zones d'intérêts (local features) dans une image. A noter qu'il s'agit ici non seulement de détecter mais aussi de caractériser, par des valeurs, pour pouvoir reconnaitre (mettre en correspondance) par la suite ces zones ou points d'intérêts dans d'autres images de la même scène. Cet algorithme a eu un succès très important au sein de la communauté vision, mais aussi en dehors de la communauté, et de nombreuses adaptations existent.

L'idée générale de SIFT est de trouver des paramètres (features) qui sont invariants à plusieurs transformations : rotation, échelle, illumination et changements mineurs du point de vue.

La détection et l'extraction de caractéristiques sur les points d'intérêt se déroulent en quatre étapes :

- détection d'extrema d'espace-échelle ("scale-space"),
- localisation des points d'intérêt,
- choix de l'orientation des descripteurs,
- calcul des descripteurs.

Pour la première étape de détection d'extrema d'espace-échelle, l'image est convoluée avec un noyau gaussien. L'espace-échelle d'une image est donc défini par la fonction :

$$L(x, y, s) = G(x, y, s) * I(x, y) \tag{2.16}$$

où $I(x, y)$ est l'image originale et

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} e^{-(x^2+y^2)/2\sigma^2} \tag{2.17}$$

A une normalisation près, cela revient à résoudre $\partial I/\partial\sigma = \Delta I$, où ΔI représente le Laplacien de I.
La Figure 2.16 nous montre les images Gaussiennes groupées par quatre octaves. De gauche à droite, l'échelle augment. De haut à bas, la taille d'image divisée par deux.

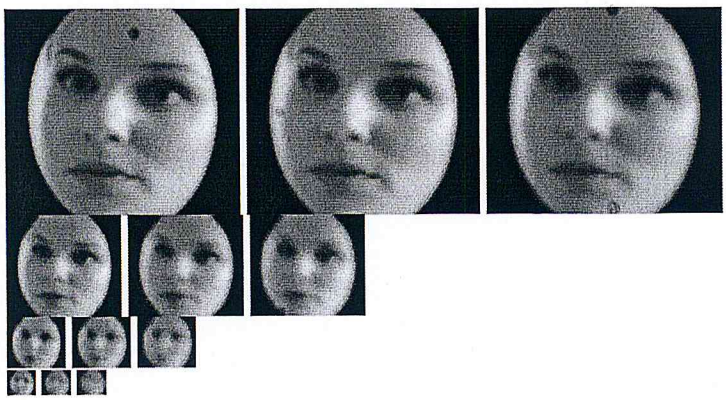


Fig. 2.16. Images Gaussiennes groupées par octaves [37].

La présélection des points d'intérêt et de leur échelle est faite en détectant les extrema locaux des différences de gaussiennes comme illustre la Figure 2.17.

$$\begin{aligned} D(x, y, s) &= (G(x, y, ks) - G(x, y, s)) * I(x, y) \\ &= L(x, y, ks) - L(x, y, s) \end{aligned} \tag{2.18}$$

Notons que $D(x, y, \sigma) \approx (k - 1)\Delta I$ lorsque $k \to 1$.

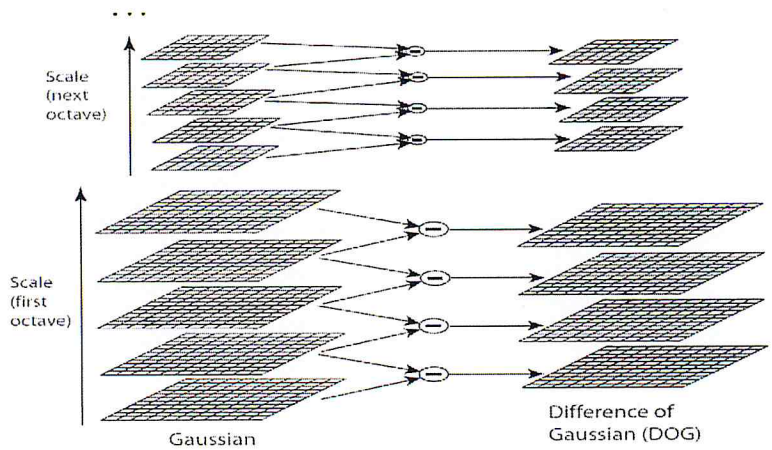


Fig. 2.17. Différences de gaussiennes [36].

Les extrema sont recherchés dans de petits voisinages en position et en échelle (typiquement 3x3x3) (voir la Figure 2.18).

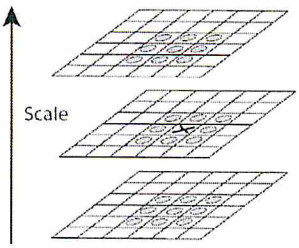


Fig. 2.18. Recherche des extrema [36].

Une étape d'interpolation a pour but d'améliorer la localisation des points d'intérêt en espace et en échelle. Puis une analyse des rapports des valeurs propres de la matrice Hessienne 2 x 2 permet d'éliminer les points d'intérêt situés dans des zones insuffisamment contrastées ou sur des bords présentant une courbure trop faible [37].

L'étape suivante consiste à assigner à chaque point une orientation. Cette orientation correspond à l'orientation majoritaire des gradients spatiaux d'intensité calculés dans un voisinage du point d'intérêt à l'échelle préalablement déterminée. Un point d'intérêt peut se voir associer plusieurs orientations. Cela entraîne par la suite une redondance des descripteurs.

Finalement, pour une position, une échelle et une orientation données, chaque point d'intérêt se voit associer un descripteur. Pour chaque image, la norme du gradient spatial $m(x, y)$ et l'orientation du gradient spatial $\theta(x, y)$ correspondants à cette échelle sont calculées :

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \tag{2.19}$$

$$\theta(x,y) = tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{2.20}$$

Le descripteur est constitué d'histogrammes d'orientation du gradient spatial d'intensité pondérés par la norme du gradient spatial (voir la Figure 2.19).

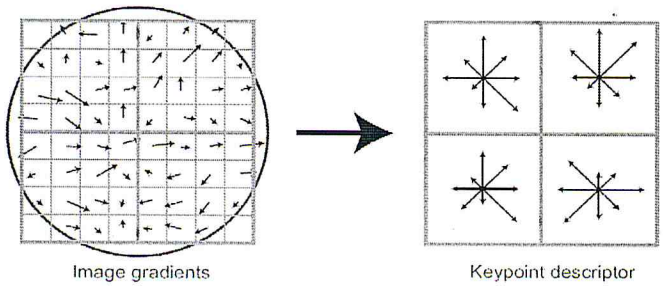


Fig. 2.19. Vecteur de descripteur des points d'intérêt [36].

En effet, le voisinage du point d'intérêt dont la taille dépend de l'échelle subit un découpage 4x4 en blocs. Pour chaque bloc, un histogramme à 8 niveaux de quantification résume les orientations du gradient spatial d'intensité à l'intérieur du bloc. Le descripteur SIFT est donc un vecteur à 4x4x8=128 coordonnées.

2.6.2. Descripteur SURF

Pour rendre le descripteur invariant à la rotation, il faut identifier de manière reproductible une direction principale dans la zone du point d'intérêt. Pour cela, on calcule la réponse à des ondelettes de Haar suivant les directions x et y dans le voisinage du point d'intérêt. Les ondelettes de Haar étant en fait des *box filters* (voir Figure 2.20), on peut de nouveau utiliser les images intégrales pour accélérer le calcul [38] [29].

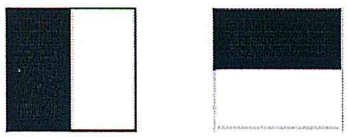


Fig. 2.20. Ondelettes de Haar suivant les directions x et y (zone noire: -1, zone blanche: +1).

Une fois les réponses calculées, elles sont pondérées par une fenêtre gaussienne centrée sur le point d'intérêt et représentées comme un point dans un espace dont l'abscisse représente la valeur de la réponse horizontale (axe x) et l'ordonnée représente la valeur de la réponse verticale (axe y). Une région qui répondrait mieux aux ondelettes orientées suivant la direction verticale verrait donc la majorité des réponses concentrées le long de l'axe y. Ensuite on calcule la somme de toutes les réponses situées dans une fenêtre de taille $\pi/3$ tournant autour du centre de la région d'intérêt (voir Figure 2.21), ce qui permet de définir la norme du vecteur d'orientation local. La direction du plus long vecteur définit l'orientation principale de la région d'intérêt.

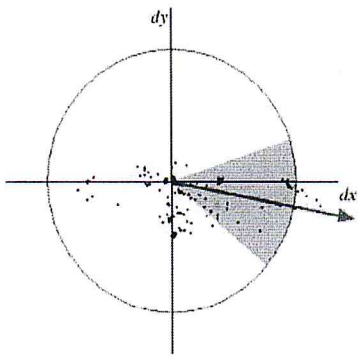


Fig. 2.21. Détection de l'orientation principale [38].

Dans la construction d'un descripteur basé sur la somme des réponses à une ondelette de Haar, La première étape pour extraire le descripteur est de construire une région rectangulaire centrée autour du point d'intérêt et orientée suivant la direction principale sélectionnée au point précédent. La

taille de cette fenêtre est déterminée par l'échelle à laquelle le point d'intérêt a été trouvé. Un exemple de ces fenêtres est donné à la Figure 2.22.

Ces régions sont ensuite découpées en 16 sous-régions (voir Figure 2.23). Dans chacune de ces sous-régions, les réponses à une ondelette de Haar sont calculées sur des échantillons régulièrement espacés. La réponse suivant la direction horizontale de la sous-région sélectionnée est notée dx, et dy suivant la direction verticale. Notons que les directions verticale et horizontale sont définies par rapport à l'orientation de la zone d'intérêt. Pour augmenter la robustesse par rapport à une erreur de localisation du point d'intérêt, les réponses dx et dy sont pondérées par une fenêtre gaussienne (cercle sur la Figure 2.14).

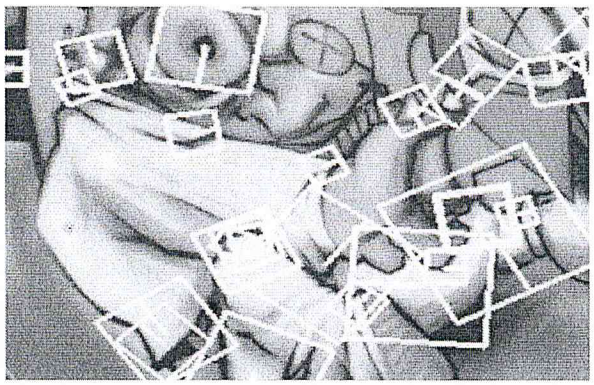


Fig. 2.22. Fenêtre de calcul du descripteur [38].

Ensuite, les réponses pondérées sont sommées sur chaque sous-région et forment les 2 premières entrées du vecteur de *paramètres* (Σdx et Σdy). Pour acheter une information supplémentaire à propos des changements d'intensité, les sommes des valeurs absolues des réponses est aussi extraite ($\Sigma|dx|$ et $\Sigma|dy|$) et constituent les deux entrées suivantes du vecteur de *paramètres*. En examinant la Figure 2.23, on voit mieux l'intérêt d'une telle description. En effet, dans les deux premiers cas, Σdx est très faible, alors que les motifs sont très différents, $\Sigma|dx|$ permet de rendre compte de cette différence.

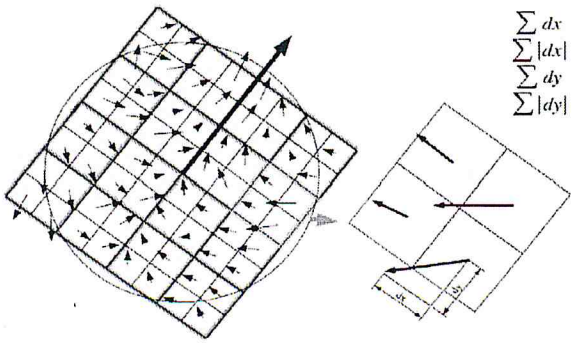


Fig. 2.23. Calcul des éléments du descripteur [38].

Pour bien comprendre, il faut imaginer que l'on fait glisser l'ondelette de Haar (représentée à la Figure 2.24) le long de la direction horizontale. Dans le premier cas (zone uniforme) la réponse sera toujours égale à zéro puisque la zone négative de l'ondelette sera exactement compensée par la zone positive. Nous aurons donc bien Σdx et $\Sigma|dx|$ égaux à zéro. Dans le second cas par contre, on aura deux types de réponses : soit très positive (transition du noir vers le blanc) ou très négative (transition du blanc vers le noir). Ces contributions se compensent, de sorte que Σdx est bien proche de zéro, mais ici $\Sigma|dx|$ prend une grande valeur. Dans le troisième cas, on a toujours une réponse positive (transition du noir vers le blanc) mais assez petite. On retrouve donc bien Σdx et $\Sigma|dx|$ égaux et assez importants.

Chaque sous-région est donc décrite par un vecteur de 4 éléments $(\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|)$. Comme il y a 16 sous-régions, on retrouve un vecteur de 64 dimensions, qui constitue la signature de la région d'intérêt.

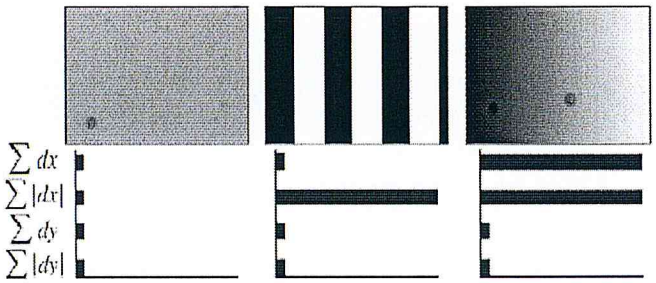


Fig. 2.24. Différents composants du vecteur de paramètres [38].

2.6.3. Descripteur BRIEF

Un descripteur binaire est composé de trois parties :

1. Un schéma d'échantillonnage : où déguster des points dans la région autour des points d'intérêt.
2. Compensation d'orientation : un mécanisme pour mesurer l'orientation du point intérêt et la tourner pour compenser les variations de rotation.
3. Échantillonnage paires : quelles paires à comparer lors de la construction du descripteur final.

Pour construire la chaîne binairė représentant une région autour d'un point intérêt que nous devons aller sur toutes les paires et pour chaque paire (P_1, P_2) et calcule le bit.

$$bit_i = \begin{cases} 1 & \text{si } P_1 < P_2 \\ 0 & \text{sinon} \end{cases} \tag{2.21}$$

BRIEF (Binary Robust Independent Elementary Features) a été le premier descripteur binaire présenté en 2010 [39]. Il ne dispose pas d'une méthode de modélisation du patch élaborée ni d'un mécanisme de compensation de l'orientation, ce qui le rend plus facile à comprendre, et à l'implémenter.

BRIEF ne prend que les informations à emplacement unique pixels pour construire le descripteur. Alors, les paires peuvent être choisies à tout moment d'une manière aléatoire sur le patch $S \times S$ comme illustre la Figure 2.25. Pour construire un descripteur BRIEF de longueur n, nous devons déterminer n paires (X_i, Y_i). Notons X et Y les vecteurs du point X_i et Y_i, respectivement.

Ainsi, afin de rendre le patch choisi moins sensible au bruit on applique un filtre gaussien.

Dans [39], les auteurs considèrent cinq méthodes pour déterminer les vecteurs X et Y:

1. X et Y sont choisis d'une manière purement aléatoire.
2. X et Y sont échantillonnés de façon aléatoire en utilisant une distribution gaussienne, ce qui signifie que des emplacements qui sont plus proches du centre du patch sont préférés.
3. X et Y sont prélevés au hasard en utilisant une distribution gaussienne où les X sont échantillonnés avec un écart type de $0{,}04 \times S^2$, et les Y_i sont prélevés en utilisant une distribution gaussienne par rapport à X_i et avec un écart type de $0{,}01 \times S^2$.
4. X et Y sont échantillonnés au hasard à partir d'un emplacement discret cerceau polaire grossier.
5. Pour chaque i, X_i est $(0, 0)$ et Y_i prend toutes les valeurs possibles sur une grille polaire.

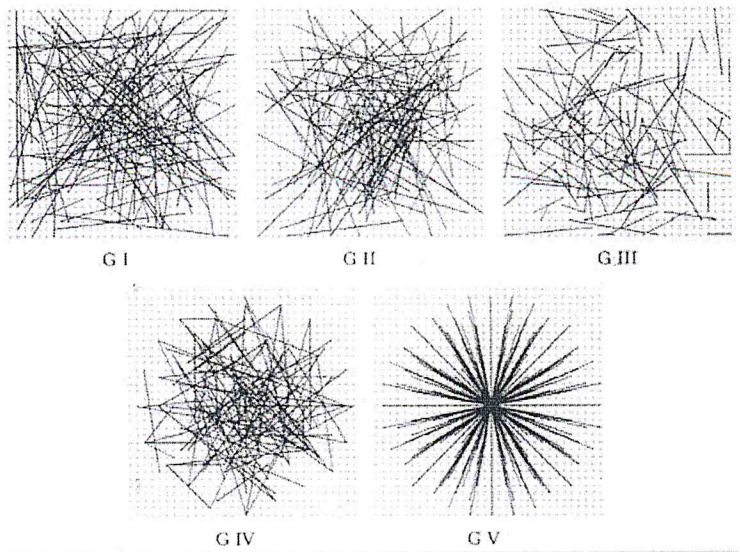


Fig. 2.25. Illustration du motif du cinq échantillonnage [39].

La Figure 2.26 présente les taux de reconnaissance en utilisant tous les cinq stratégies d'échantillonnage. Nous pouvons voir les taux de reconnaissance sont sur le même, attendons pour la cinquième stratégie d'échantillonnage qui indique le rendement pire.

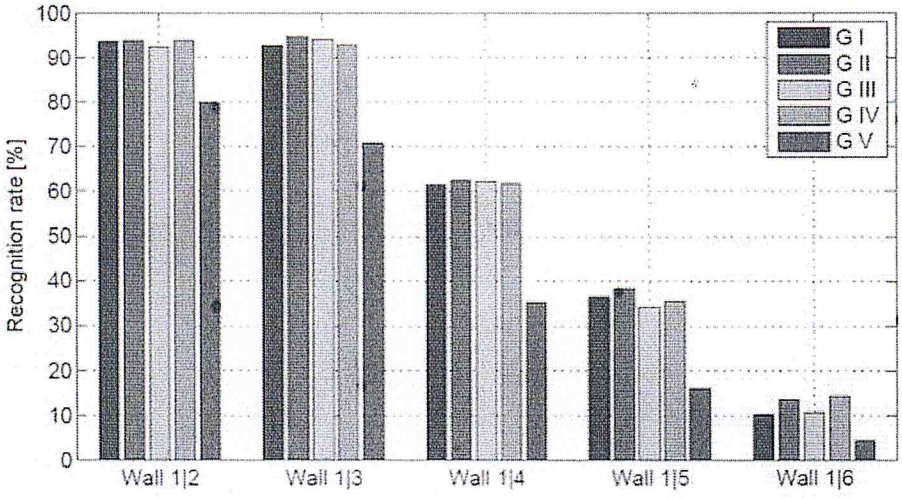


Fig. 2.26. La performance des cinq modèles d'échantillonnages [39].

Comme avec tous les descripteurs binaires, la mesure de la distance de BRIEF est le nombre de bits différents entre les deux chaînes binaires qui peut également être calculée comme la somme de l'opération [40] [41].

2.6.4. Descripteur ORB

Rublee et al. [42] Proposent un descripteur très rapide binaire basé sur le descripteur BRIEF (Binary Robust Independent Elementary Features), appelé ORB (Oriented FAST and Rotated BRIE), qui est invariante par rotation et robuste au bruit.

Le descripteur ORB est un peu similaire au BRIEF. Il ne dispose pas d'un modèle d'échantillonnage élaboré que BRISK (Binary Robust Invariant Scalable Keypoints) [43] ou FREAK (Fast REtinA Keypoint) [44]. Cependant, il ya deux différences principales entre ORB et BREF:

- ORB utilise un mécanisme de compensation de l'orientation, qui le rend invariant à la rotation.
- Les paires de points optimales sont apprises pour ORB, alors que dans BRIEF elles sont choisies au hasard.

ORB est essentiellement une fusion de détecteur de point intérêt FAST et le descripteur BRIEF, des nombreuses modifications ont été fait pour améliorer les performances de ce dernier.

ORB utilise la mesure du centre de gravité de l'intensité I afin de calculer la rotation du patch. Tout d'abord, les moments m_{pq} d'un patch sont définis par l'équation 2.17 comme suit [45]:

$$m_{pq} = \sum_{xy} x^p y^q I(x,y) \quad , \qquad (2.22)$$

avec $p = q = 1,2,3$.

Avec ces moments que nous pouvons trouver le centre de gravité C de la pièce en tant que :

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2.23}$$

En suite, la construction d'un vecteur du centre du patch "O", vers le centre de gravité "C" comme présenta la Figure 2.27. L'orientation du patch est donnée par l'équation :

$$\theta = \text{atan } 2\ (m_{01}, m_{10}) \tag{2.24}$$

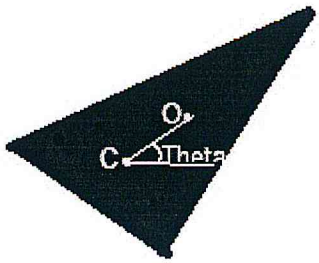


Fig. 2.27. Illustration de calcul d'angle de rotation [45].

2.7. Suivi d'objet par la mise en correspondance

Le suivi d'objet dans des séquences d'images reste est un thème de recherche très actif en vision par ordinateur dans ces dernières décennies, Le reconnaissance correspond à l'estimation de la localisation de l'objet dans chacune des images d'une séquence vidéo. Le procédé de localisation se fonde sur la reconnaissance d'objet d'intérêt à partir d'un ensemble de caractéristiques visuelles.

La Figure 2. 28 présente le schéma global un système de reconnaissance d'objet basé sur la mise en correspondance des points d'intérêt. En premier temps, paramètres (vecteurs descripteurs) sont extrait à partir d'un modèle de référence. En suite, pour chaque trame de la séquence vidéo (image en temps réel) en extraire les vecteurs caractéristiques et en fait la mise en correspondance entre les vecteurs de références et les vecteurs courants, par une mesure de similarité entre eux. Finalement pour reconnaitre en calcule la matrice d'homographie.

Cette technique se base sur l'appariement des descripteurs des images, il consiste à appliquer un algorithme de détection et description sur chaque frame a part. Après, une comparaison est faite entre ces descripteurs avec ceux de l'image de référence afin de reconnaitre et localiser l'objet ciblé. Dans notre cas, nous avons adopté l'algorithme ORB. C'est un algorithme robuste d'extraction de points d'intérêt qui calcule les descripteurs locaux invariants à différentes changements de ces points caractéristiques. En plus, ORB est aussi rapide par rapport à d'autres méthodes de détection-description comme SURF.

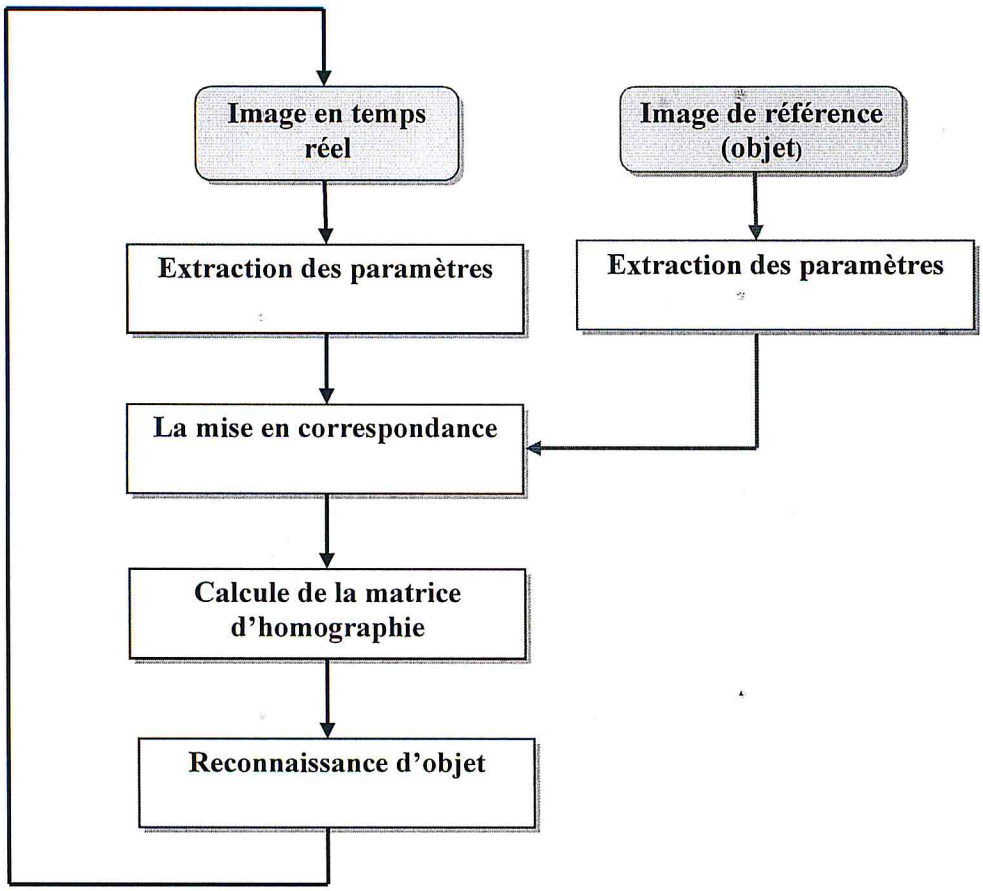


Fig. 2.28. Schéma global un système de suivi d'objet.

2.7.1. La mise en correspondance

Le problème de la mise en correspondance ou bien le matching d'images consiste à identifier dans deux ou plusieurs images d'une même scène, les primitives qui "se correspondent". Le terme de primitives désigne des points ou des régions particulières de l'image riches en information.

Les primitives utilisées sont les points d'intérêt extraites à partir de l'image modèle. On peut établir des correspondances entre des images ne représentant pas exactement la même scène.

C'est le cas, par exemple, dans les applications de reconnaissance et de suivi d'objet où l'on cherche à identifier une zone de l'image contenant l'objet en question [29].

L'objectif d'un appariement est de rechercher, dans plusieurs images, le couple de points ayant la meilleure similarité (ou ressemblance). Afin de mettre en correspondance un ensemble de points d'une image à une autre, une description locale est utilisée. Elle permet d'extraire l'information du voisinage de chaque point.

La mise en correspondance des points d'intérêt comme illustre la Figure 2.29 se base quant à elle sur une approche par corrélation à laquelle nous ajoutons un coefficient de sélection ainsi qu'une étape de suppression des doublons. Afin de trouver les appariements, une fonction de distance est appliquée entre les descripteurs des points d'intérêt des deux images. Il existe plusieurs distances qui peuvent être utilisées pour la mise en correspondance comme :

- ✓ La distance de Minkowsky : $d(X,Y) = (\sum_{i=1}^{n} |X_i - Y_i|^P)^{1/P}$
- ✓ La distance Euclidienne : $d(X,Y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$
- ✓ La distance des Cordes Carrées : $d(X,Y) = \sum_{i=1}^{n}(\sqrt{X_i} - \sqrt{Y_i})$
- ✓ La distance de Hamming : $d(X,Y) = \sum_{i=1}^{n}(X_i \oplus Y_i)$

avec X, Y sont deux vecteurs de description.

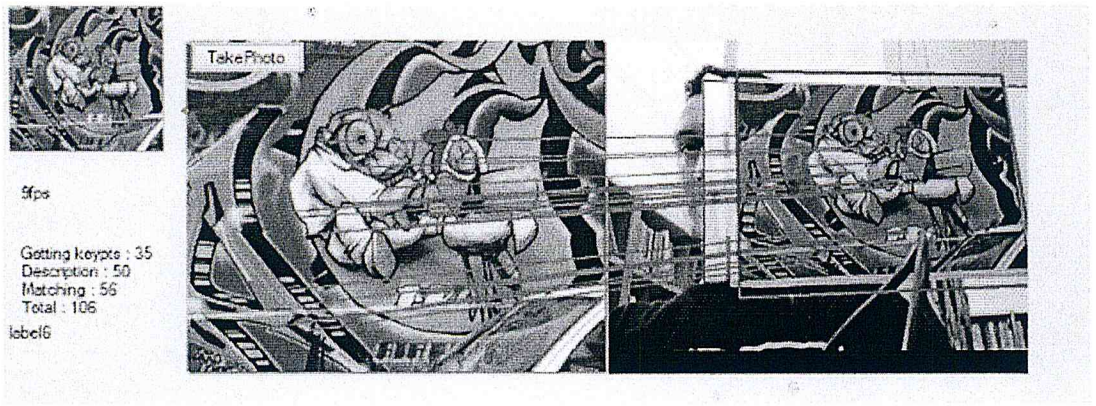


Fig. 2.29. Exemple de l'appariement entre deux images utilisant ORB, en vert : les lignes d'appariement, en rouge l'objet à suivre.

2.7.2. L'algorithme RANSAC

RANSAC est une abréviation de "RANdom SAmple Consensus" (le consensus d'échantillons aléatoires), proposé par Fischler et Bolles en 1981 [46]. C'est un algorithme d'estimation de paramètres d'un modèle mathématique pour un ensemble de données observées. Les données se composent de données correctes (inliers) ou les points sont exprimés en fonction des paramètres du modèle, et des données aberrantes (outliers) qui sont des points qui ne permettent pas l'ajustement du modèle comme illustre Figure 2.30. Trois paramètres apparaissent importants dans l'implémentation de RANSAC :

La distance-seuil qui détermine si une donnée est un inlier ou un outlier (elle est le plus souvent choisie de façon empirique)

Le nombre N d'échantillons testés avant de choisir celui qui a abouti à la meilleure estimation

La taille de support S considérée comme acceptable, qui est le deuxième critère de terminaison de l'algorithme.

Lorsqu'on est capable d'estimer le pourcentage d'outliers ϵ, les paramètres N et S peuvent être choisis de façon à ce que la probabilité p qu'au moins un échantillon testé ne contiennent que des inliers soit de 0,99 : si la taille de l'échantillon est s ($s = 2$ dans le cas de l'estimation d'une droite, et $s = 3$ dans le cas de l'algorithme des trois points), alors la probabilité qu'un échantillon

contienne au moins un outlier est $1-(1-\epsilon)^s$. Lorsqu'on tire au sort N échantillons, la probabilité qu'au moins un ne contienne aucun outlier est donc $p = 1-[1-(1-\epsilon)^s]^N$, d'où:

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \tag{2.25}$$

où l'on choisit généralement $p = 0,99$.

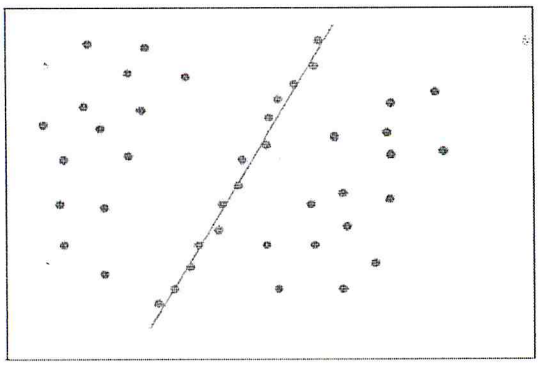


Fig. 2.30. Ajustement d'une ligne avec les données correctes en utilisant RANSAC [2].

Il est généralement possible de déterminer le seuil d'erreur tolérée expérimentalement. En perturbant les données, on calcule le modèle et on mesure l'erreur. Le seuil peut être pris égal à l'écart-type des erreurs. En supposant une distribution d'erreurs, ce seuil correspond à une certaine probabilité. Il dépend donc aussi de la distribution supposée des erreurs, mais l'hypothèse de la distribution gaussienne des erreurs est habituellement suffisante.

L'algorithme RANSAC est employé dans le domaine de reconnaissance d'objet en calculant des alignements des points par une projection perspective en utilisant un nombre minimum de correspondances entre les primitives du modèle et celles des données observées dans l'image.

2.7.3. Estimation de l'homographie

La transformation projective ou homographie, il faire correspondre les points dans deux images. Comme nous travaillons avec une image (un objet planaire) et nous considérons qu'il est rigide, c'est possible de trouver la transformation homographie entre points appariés de l'image de référence et l'image en cours. Cela revient a relié ces points par une relation linéaire :

$$H_0 x = x' \tag{2.26}$$

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{2.27}$$

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32} + 1} \tag{2.28}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32} + 1} \quad (2.29)$$

Dont H_0 est la matrice 3×3 dite d'homographie, x représente les coordonnées du point dans l'image de référence et x' dénote la nouvelle position de ce point dans l'autre image. Cette matrice à huit uniques nombres ils peuvent être estimés de 4 points d'une image et ses correspondants. Une fois cette matrice est estimée, tous les points dans une image, peuvent être transférés vers le second en utilisant cette relation.

L'étape de recherche de la matrice d'homographie est importante, car la transformation obtenue est l'élément clé pour trouver l'emplacement de l'objet dans l'autre image.

2.8. Conclusion

Le choix des primitives visuelles est un critère très important de la présentation d'un objet pour la reconnaissance et de suivi. Les points d'intérêt sont des primitives significatives très utilisées dans ces dernières années. Notre travail consiste à reconnaitre un objet basé sur la mise en correspondances des points d'intérêt à partir des vecteurs caractéristiques (descripteurs) calculés. Plusieurs méthodes de description des points d'intérêt sont proposées dans la littérature, les descripteurs binaires (ORB) sont présentés des performances importantes pour la reconnaissance. Dans ce chapitre nous avons présentés les différentes étapes de processus de suivi d'objet basé sue les points d'intérêt.

Dans le prochain chapitre, nous allons entamés la phase de conception et de mise en place de notre système.

Chapitre 3

Conception

3.1. Introduction

Le développement d'une application nécessite plusieurs phases. L'analyse des besoins étant la première phase, elle permet une meilleure compréhension de ce qu'on doit faire pour une meilleure organisation lors de la phase réalisation.

Dans ce chapitre nous allons commencer par l'analyse et la spécification des besoins puis nous allons définir les outils de développement à utiliser pour l'implémentation de notre application. Ensuite, nous présentons quelques-unes de ses fonctionnalités, puis nous terminons par quelques résultats et une conclusion.

3.2. Le choix de processus de conception

Le choix de la méthode de conception est crucial. Nous avons adopté le formalisme UML (Unified Modeling Language). Le choix a été guidé par le fait qu'UML permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation. Cette approche permet de passer du modèle au système de manière lisible.

UML est un langage de modélisation favorisant [47] :

- Une meilleure communication entre les intervenants dans un objet, puisque UML offre des moyens de capture des connaissances sur un sujet à travers divers points de vues (ces points de vues sont fournis par les différents diagrammes de UML).
- Une bonne compréhension du problème, en fait le système à étudié sera traité suivant différents angles et suivant les différents cas d'utilisation de ce système.
- UML incorpore les meilleures pratiques d'ingénierie dans les différents domaines qui ont abouti à son apparition, ce qui lui permet d'être adapté aux différents types de systèmes.

UML permet aussi de suivre un projet dans ses différentes étapes :

- Spécifier : UML s'adresse à la spécification du système il peut être utilisé pour modéliser les besoins(le quoi) et l'architecture(le comment).
- Visualiser : les différents diagrammes donnent aux concepteurs une vue précise sur le système avant sa réalisation.
- Construire : les différents diagrammes et modèles établis durant la phase de spécification et de conception servent de base pour la réalisation.
- Documenter : les diagrammes utilisés durant les différentes phases pour communiquer les connaissances entre les membres du projet, de la spécification des besoins jusqu'à la réalisation, présentent un document détaillé sur les diverses phases et modules du projet.

L'objectif de notre travail consiste à développer une technique de suivi sans marqueur pour le recalage virtuel/réel. Le suivi des indices visuels dans une séquence d'images vient de la caméra en temps réel pour incruster les objets virtuels d'une manière cohérente avec la scène réelle.

Dans notre cas, nous utilisons UML, qui est un langage qui fournit des outils permettant de modéliser le système à concevoir.

Pour définir les étapes de création du système connue sous l'appellation de cycle de vie du logiciel, nous avons opté pour le diagramme en "V", illustré dans la Figure 3.1. Il présente le cycle de vie d'un logiciel depuis l'analyse des besoins jusqu'à la validation.

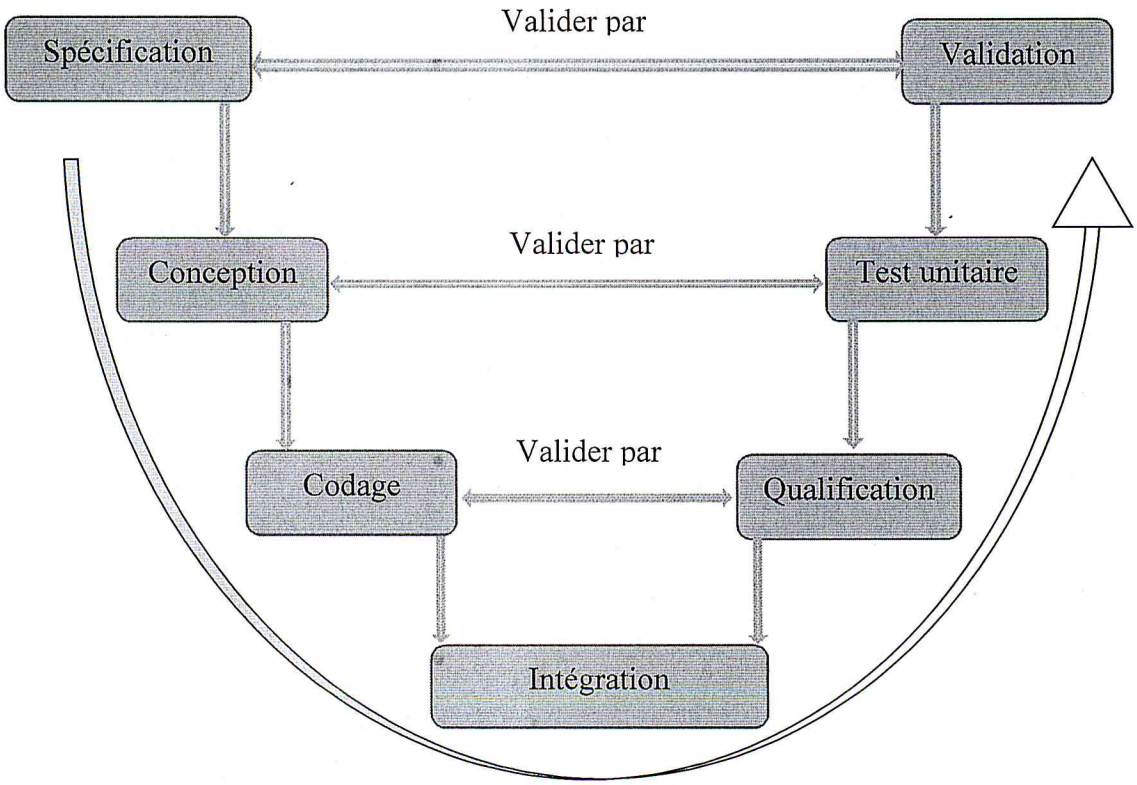


Fig. 3.1. Cycle de développement en « V » [47].

3.3. Présentation du projet

Notre projet rentre dans le cadre d'un projet de recherche initié par l'équipe IRVA (Interaction homme système Réalité Virtuelle et Augmentée) au sein de la Division Robotique et Productique du Centre de Développement des Technologies Avancées (CDTA). Le projet est intitulé «Interaction 3D multimodale et collaborative dans un environnement de réalité virtuelle et augmentée».

L'objectif scientifique du projet est l'étude et la mise en œuvre de nouveaux modèles et techniques logicielles pour l'assistance à l'interaction et à la collaboration dans des environnements de réalité virtuelle et augmentée utilisant ou simulant des systèmes complexes. Dans notre cas c'est l'étude et l'implémentation des techniques de suivi d'objet sans marqueurs utilisant des primitives visuelles pour la réalisation d'un système de réalité augmenté performant.

3.3.1. Analyse et spécification des besoins

D'une manière globale, tracer les objectifs, connaitre les acteurs et prévoir les fonctionnalités qu'assure notre système suffisent pour spécifier les besoins.

3.3.1.1. Présentation des objectifs de l'application

Le système que nous concevons contient deux principaux volets : (1) reconnaitre et suivre une cible dans une scène réel, (2) assurer l'insertion d'objets virtuels recalés par rapport aux objets réels reconnus dans la scène.

3.3.1.2. Identification des acteurs

Un acteur est une entité externe qui agit sur le système (opérateur, autre système, etc.) et qui peut consulter ou modifier l'état du système. Notre système et dédié à un simple utilisateur qui peut interagir avec les fonctionnalités présentées ci-après.

3.3.1.3. Établir les fonctionnalités

Parmi les fonctionnalités que notre système va assurer, nous citons :

- ✓ Choisir et configurer et calibrer la caméra à utiliser.
- ✓ Démarrer la caméra.
- ✓ Choisir une cible ou une scène à reconnaitre.
- ✓ Démarrer la détection et suivi d'objet et la reconnaissance d'objet.
- ✓ Visualiser les points d'intérêt de la scène réelle, les points d'intérêt de la cible et l'appariement entre les deux .
- ✓ Choisir et ajouter un objet virtuel 3D.

❖ **Identifications des besoins fonctionnels**

- ✓ **Configuration de la caméra :**
  - ➢ Le choix et l'activation de la caméra.
  - ➢ Procédure de calibrage interne automatique :

- Charger des paramètres intrinsèques.

✓ **Procédure du suivi :**

- Choisir de l'objet à suivre.
- Détection de l'objet dans la scène réelle.
- Suivi de l'objet détecté.
- Calcul des paramètres extrinsèques (calibrage externe de la caméra).

✓ **Augmentation de la scène réelle avec l'objet virtuel :**

- Sélection de l'objet virtuel.
- Insertion et recalage de l'objet virtuel.

3.4. Le choix des méthodes et algorithmes utilisés

3.4.1. Le choix d'une méthode appropriée à notre cas d'étude

Nous avons d'une part, les contraintes que doit satisfaire un système de réalité augmentée et d'une autre part, les performances et les caractéristiques des points d'intérêt qui varient selon le détecteur. Roland Azuma a défini trois critères que doit satisfaire un système de réalité augmentée [1], combiner le réel et le virtuel, le temps réel, l'alignement de l'objet virtuel sur l'objet de la scène réelle. Afin de satisfaire ces contraintes, une reconnaissance précise et rapide des objets d'une scène du monde réel est requise. Les caractéristiques que doivent posséder les points d'intérêt afin de satisfaire les besoins d'un système de réalité augmentée, sont comme suit :

- **Le temps de calcul réduit**

Le calcul et l'appariement des descripteurs ainsi que la détection d'un grand nombre de points d'intérêt doivent se faire en temps réel afin de permettre le traitement d'un nombre important d'images par seconde.

- **L'invariance au changement de luminance et au bruit**

Les descripteurs des points d'intérêt ne doivent pas varier significativement sous un changement de luminance ou augmentation du flou dans l'image.

- **L'invariance au changement de point de vue**

Les utilisateurs d'un système de réalité augmentée ne doivent pas être limités par des contraintes liées à l'angle de vue. Pour cela, les descripteurs des points d'intérêt ne doivent pas varier significativement avec un changement du point de vue ou de l'orientation du dispositif de capture.

- **L'invariance au changement d'échelle**

En réalité augmentée, les objets sont observés à différentes distances. Donc, leurs reconnaissances doit se faire à des intervalles de distances assez larges et dans aucun cas limitée à une distance

précise. Pour remédier à cette contrainte, les points d'intérêts doivent être invariants aux changements d'échelles. Cela signifie que les points détectés à une distance précise entre la caméra et l'objet ne doivent pas disparaitre en se rapprochant ou en s'éloignant de l'objet. En considérant la contrainte de temps réel que doivent satisfaire les systèmes de réalité augmentée (contraintes d'Azuma[1]) et les résultats des expériences décrites précédemment, notre choix se porte sur ORB. Cette méthode et robuste au bruit et stable par rapport aux changements d'échelle, de rotation, de points de vue et aux transformations affines. Néanmoins, on enregistre une supériorité d'ORB par rapport à SIFT et SURF sur la caractéristique d'invariance au changement de luminance et rotation et surtout par rapport au temps d'exécution.

3.4.2. Architecture global du système de recalage à développer

Notre travail consiste à développer un système de recalage pour une application de réalité augmentée basé sur le suivi d'objet (reconnaissance d'objet en temps réel) utilisant la mise en correspondance des points d'intérêt. Pour cela, deux parties seront développées : la reconnaissance d'une cible naturelle dans une scène réelle suivie du calibrage de camera et de suivi d'objets. Donc, Nous développerons le schéma global du système de RA représentée dans la Figure 1.15, Section 1.8 dans le Chapitre 1.

- Pour la calibration nous utiliserons la méthode Zhang.
- Pour ce qui est de l'estimation de la pose l'algorithme analytique reste très peu couteux et ne nécessite que 4 point pour estimer la pose il reste parfait pour son utilisation dans les applications de réalité augmentée, c'est pourquoi nous le retenant afin de l'exploiter dans notre système.
- Nous utilisons ORB pour la détection et description des points d'intérêt.

3.4.3. Choix de la méthode ORB pour l'extraction des points d'intérêt

Les performances des algorithmes de suivi haut-niveau tel que ceux cité dans le Chapitre 2 sont sensiblement liées aux performances des opérations bas-niveau (c'est-à-dire opérant directement sur les images) qui produisent les entrées de ces algorithmes. Dans notre cas le suivi des points d'intérêt fournit un échantillonnage du champ de mouvement 2D entre deux images.

Ils existent plusieurs méthodes d'extraction des points d'intérêt, ceux qui nous intéressent sont qui non seulement détecte les point d'intérêt mais aussi fournissent leur descripteurs qui vont être utilisé par la suite pour la comparaison et l'appareillement entres les deux images. La méthode la plus pertinente citée dans la littérature SIFT, SURF et ORB.

Nous avons choisi d'utiliser la méthode ORB à cause de certaines performances cité dans [42] en termes de rapidité, précision et l'invariance au quelques changement d'image.

3.4.4. La mise en correspondance

La mise en correspondance de points d'intérêt est un processus indispensable car l'appareillement sert de passerelle entre le haut niveau (reconnaissance) et le bas niveau (extraction d'informations). L'objectif d'un appariement est de rechercher, dans plusieurs images, le couple de points ayant la meilleure similarité (ou ressemblance).

L'appariement consiste à localiser, dans les images, les projections de la même entité du modèle de la scène. Il existe de nombreuses méthodes de mise en correspondance, cependant très peu sont applicable en temps réel.

Nous utilisons la méthode d'appareillement par le calcul de distance de Hamming (Section 2.7.1, Chapitre 2) qui est le type de mise en correspondance utilisé dans les descripteurs binaires. Ainsi que l'utilisation de l'algorithme RANSAC pour optimiser la mise en correspondance et éliminer les faux appariements.

Dans notre cas la méthode sera appliquée aux points d'intérêt résultant d'ORB sur deux images ; la première enregistrer au préalable et la deuxième de la scène réel provenant du flux vidéo de la caméra.

3.5. Conception du système

La conception de notre système s'est inspirée des diagrammes UML suivants : les diagrammes de cas d'utilisation, les diagrammes de classes et les diagrammes de séquences. A partir des fonctionnalités établies précédemment, nous dégageons les cas d'utilisation du système que nous représentons dans un diagramme de cas d'utilisation global. Ensuite nous établissons trois diagrammes de cas d'utilisation qui vont détailler le diagramme global, il s'agit du diagramme de configuration et de calibration de la caméra, le diagramme de détection et de reconnaissance d'une cible ainsi que le diagramme d'augmentation d'une scène. Par la suite, nous établissons un diagramme de classes global correspondant au diagramme de cas d'utilisation global .Ensuite, nous dégageons les classes participantes dans chaque diagramme de cas d'utilisation spécifique pour établir le diagramme de classe correspondant Une fois les diagrammes de cas d'utilisation et les diagrammes de classes établis, nous commençons par établir un diagramme de séquence global. Ce dernier sera détaillé dans les diagrammes de séquence correspondant à chaque diagramme de cas d'utilisation.

3.5.1. Diagrammes de cas d'utilisation

Le diagramme des cas d'utilisation définit le comportement d'un système tel qu'un utilisateur extérieur (généralement non informaticien). Il divise les fonctionnalités d'un système en unités cohérentes, qui permettent l'expression des besoins des utilisateurs.

➢ Identification des cas d'utilisation

A partir des fonctionnalités établies du système, nous dégageons un ensemble de cas d'utilisation du système que nous présentons dans le Tableau suivant :

Tableau 3.1. Identification des cas d'utilisation.

Cas d'utilisation	Message (émis /reçu)
Choix et configuration et calibration de la caméra.	Emis : demande de choix et configuration et calibration de la caméra. Reçu : caméra sélectionner et configurer et calibrer.
Démarrer la caméra	Emis : démarrer la caméra Reçu : Flux vidéo de la scène
Choisir une scène	Emis : Sélection de la scène Reçu : scène sélectionner
Choisir une cible	Emis : Sélection d'une cible Reçu : cible sélectionnée
Choisir un objet 3D	Emis : Sélection d'un objet 3D Reçu : objet 3D sélectionnée
Détection et suivi d'objet	Emis : Demande de détection et suivi de la cible Reçu : Flux vidéo de la scène avec cible reconnu
Reconnaissance	Emis : Demande de reconnaissance Reçu : Flux vidéo de la scène avec cible reconnu
Augmentation	Emis : commencer l'augmentation Reçu : Flux vidéo de la scène augmentée

A partir des cas d'utilisation présentés dans le Tableau 3.1 nous allons construire un diagramme de cas d'utilisation global et trois diagrammes de cas d'utilisation spécifiques. Il s'agit du diagramme des cas d'utilisation de configuration et de calibrage de la caméra, du diagramme de cas d'utilisation de détection et de reconnaissance d'une cible et du diagramme de cas d'utilisation augmentation d'une scène.

3.5.1.1. Diagramme des cas d'utilisation global

C'est un diagramme qui résume les trois diagrammes de cas d'utilisations spécifiques qui viennent juste après, il est présenté dans la Figure 3.2.

3.5.1.2. Diagramme de cas d'utilisation augmentation d'une scène

L'augmentation d'une scène consiste à choisir une scène à reconnaitre et le choix d'un objet virtuel (entité virtuelle). L'ajout de l'objet virtuel se fait après calcul des matrices intrinsèque et extrinsèques à partir des cibles reconnues de la scène comme illustre la Figure 3.3.

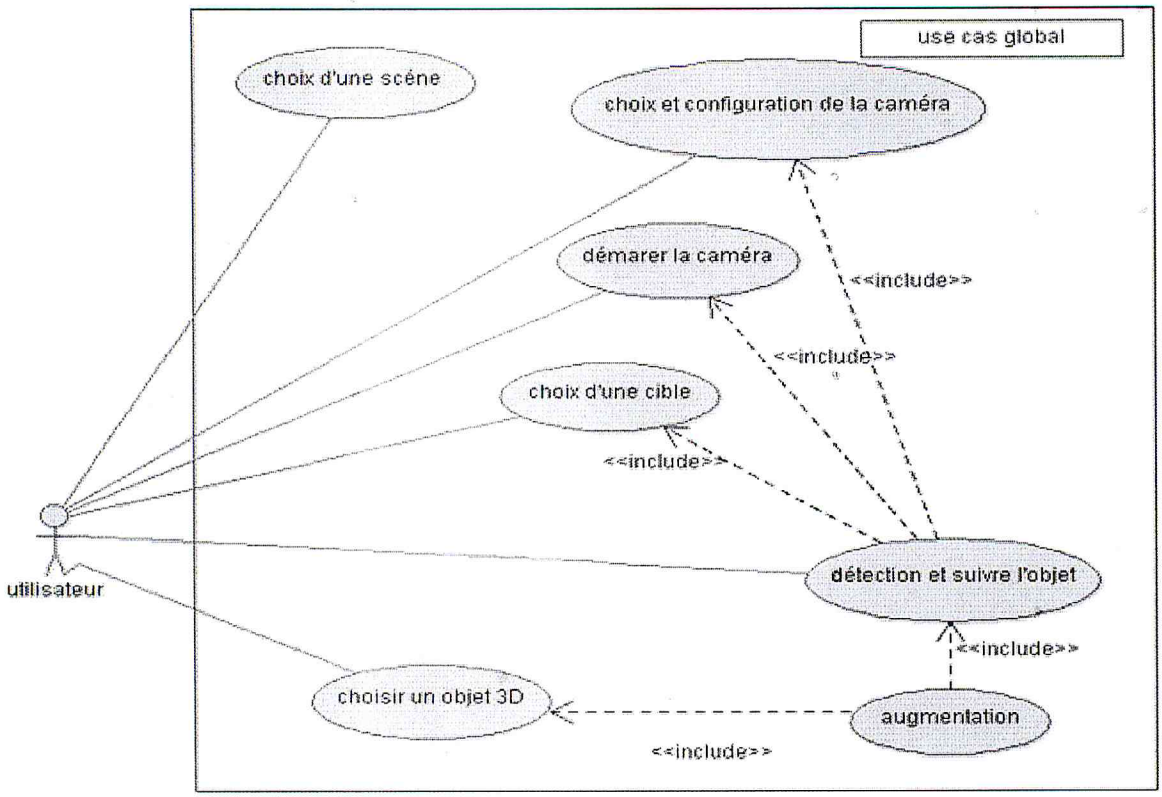


Fig. 3.2. Diagramme de cas d'utilisation global.

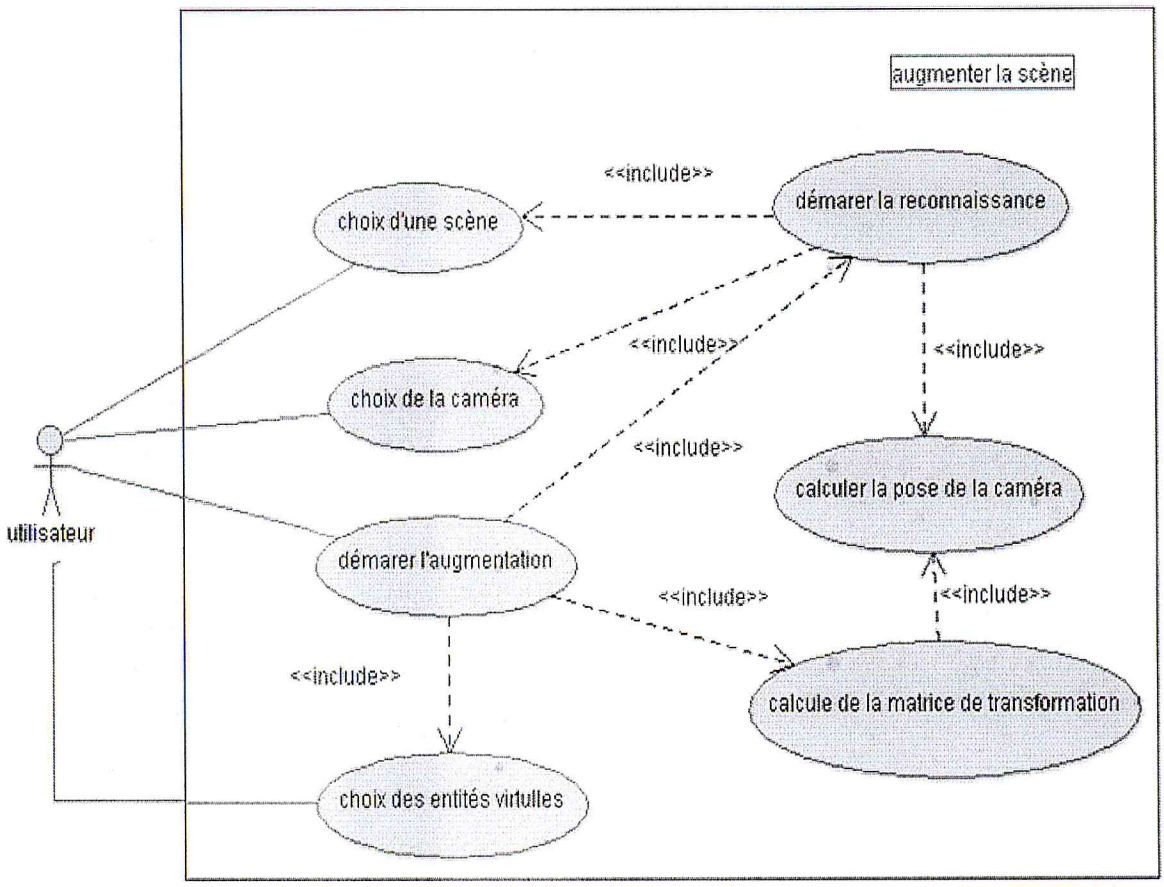


Fig. 3.3. Diagramme de cas d'utilisation augmentation d'une scène.

3.5.1.3. Diagramme des cas d'utilisation de détection et de reconnaissance d'une scène

Ce diagramme comprend la détection et la reconnaissance des points d'intérêt, donc la reconnaissance consiste à choisir une scène, l'extraction des points d'intérêt des cibles appartenant à la scène ainsi que ceux appartenant à l'image capturée de la scène réelle, enfin l'appariement des descripteurs des points d'intérêts de l'image de la scène réelle avec ceux des cibles reconnues. Si une correspondance est détectée alors l'appariement a réussi comme présente la Figure 3.4.

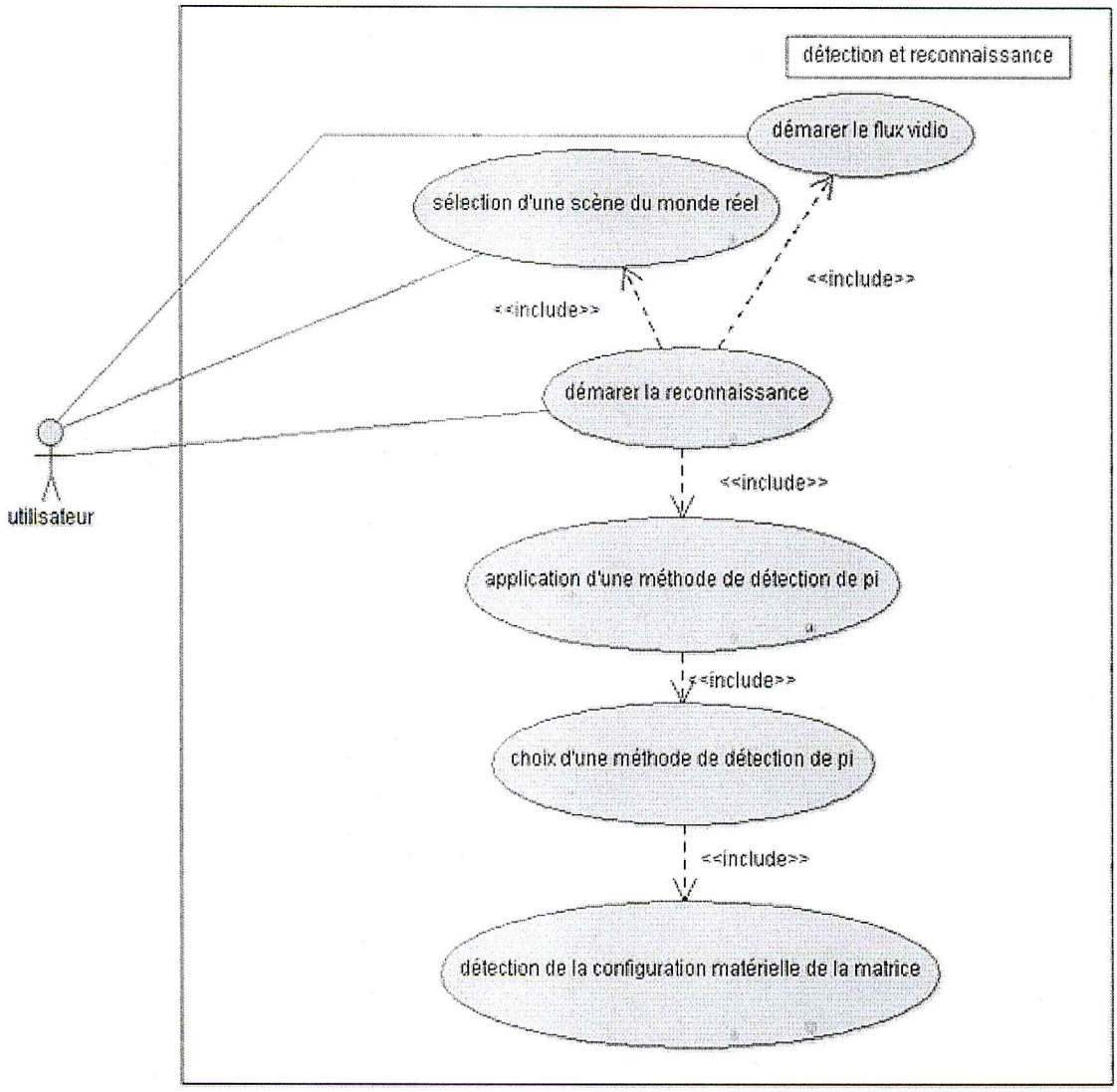


Fig. 3.4. Diagramme des cas d'utilisation de détection et de reconnaissance.

3.5.1.4. Diagramme des cas d'utilisation de configuration et de calibrage de caméra

Le diagramme des cas d'utilisation présenté ci-après comprend les manipulations qu'un utilisateur effectue pour configurer la caméra. Parmi lesquelles, nous citons le choix d'une caméra, l'activation de la caméra, et le calibrage de la caméra comme présente la Figure 3.5.

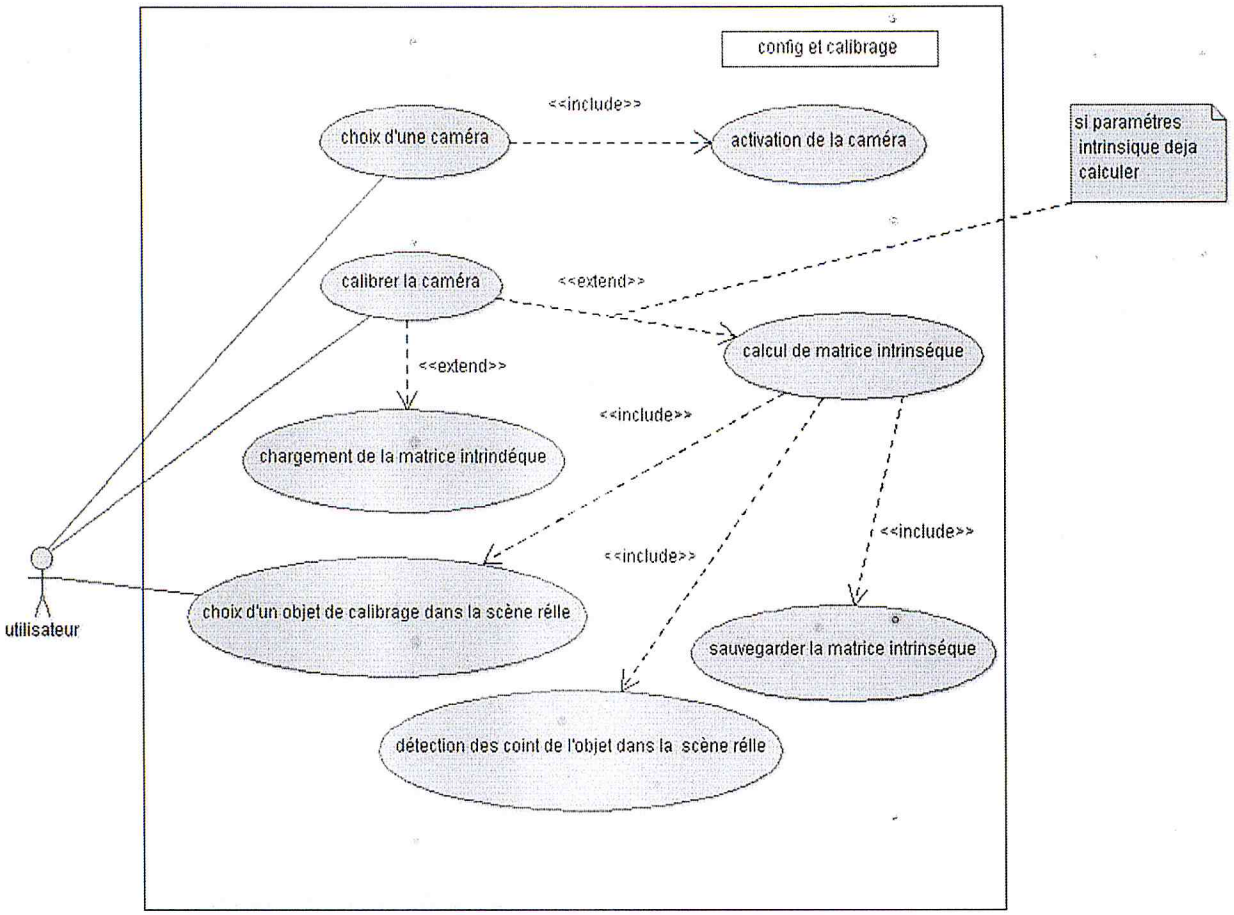


Fig. 3.5. Diagramme des cas d'utilisation de configuration et de calibrage de caméra.

3.5.2. Diagrammes de classes

Le diagramme de classes représente la structure statique d'un système, il montre les différentes classes et les relations qui existent entre elles. Il faut noter qu'il est indépendant de tout langage de programmation Orienté Objet.

Nous dégageons un diagramme de classe global à partir du diagramme de cas d'utilisation global.

3.5.2.1. Diagramme de classe globale

Dans ce diagramme figure toutes les classes participantes dans les diagrammes de cas d'utilisation, le diagramme est présenté dans la Figure 3.6.

- ✓ **Description Détaillée des classes**

Les détails des classes participantes dans les diagrammes précédents sont ci-après, quand à la discussion sur ces classes, elle vient juste après :

- ✓ **La classe augmentation**

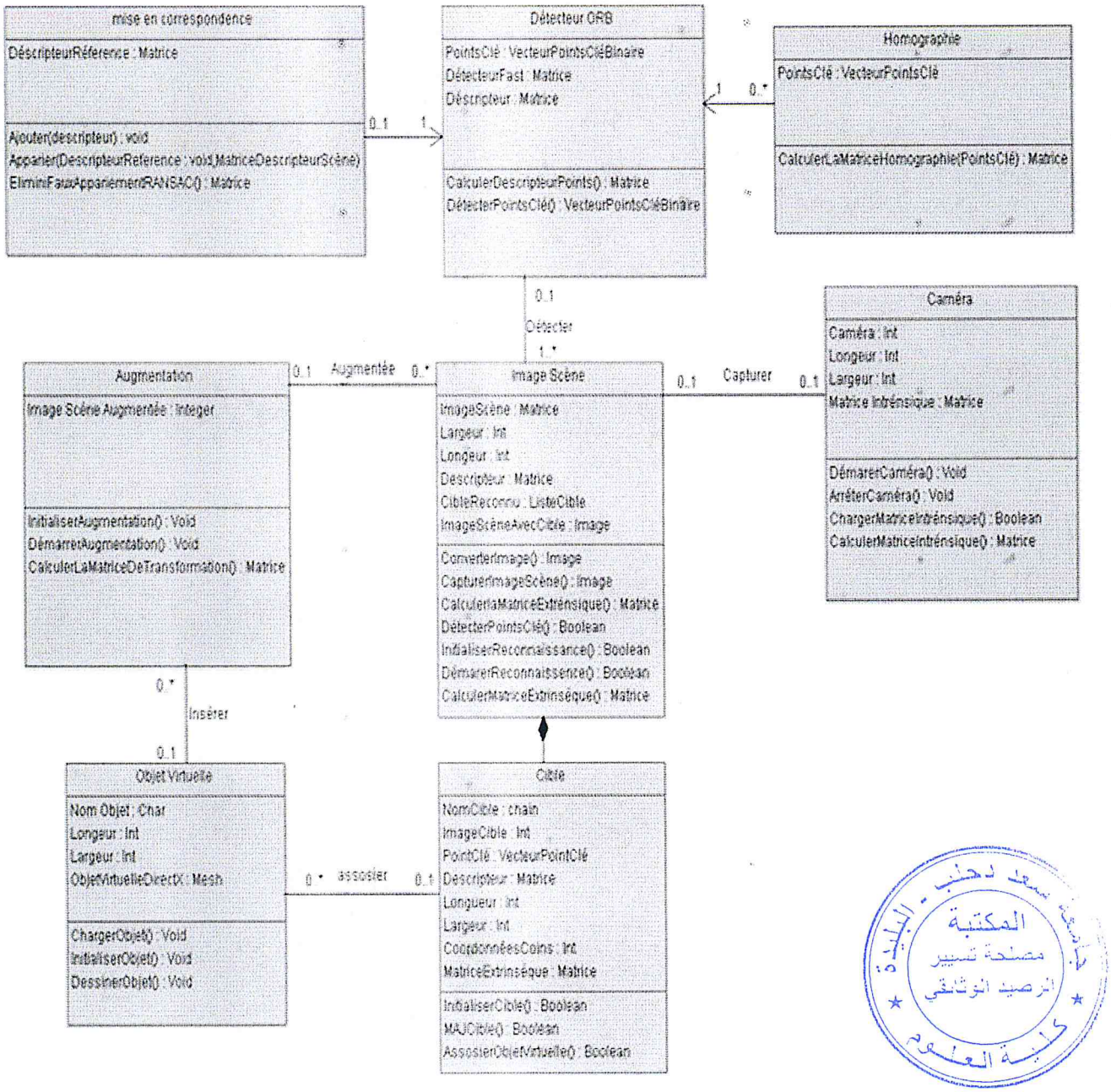


Fig. 3.6. Diagramme de classe globale.

3.5.3. Diagramme de séquence

Le diagramme de séquence représente les interactions entre les objets (instanciation de classes) qui constituent le système selon un ordre chronologique. Les objets communiquent entre eux par envoi de messages et par appels de procédure. Dans un diagramme de séquence le temps est représenté à travers une dimension (La dimension verticale) et s'écoule de haut en bas. En se basant sur le diagramme de classes global, on établit un diagramme de séquence pour chaque diagramme de cas d'utilisation.

3.5.3.1. Diagramme de séquence global

La Figure 3.7 représente le diagramme de séquence global, ce diagramme enveloppe les trois scénarios et leurs diagrammes qui seront présentés juste après.

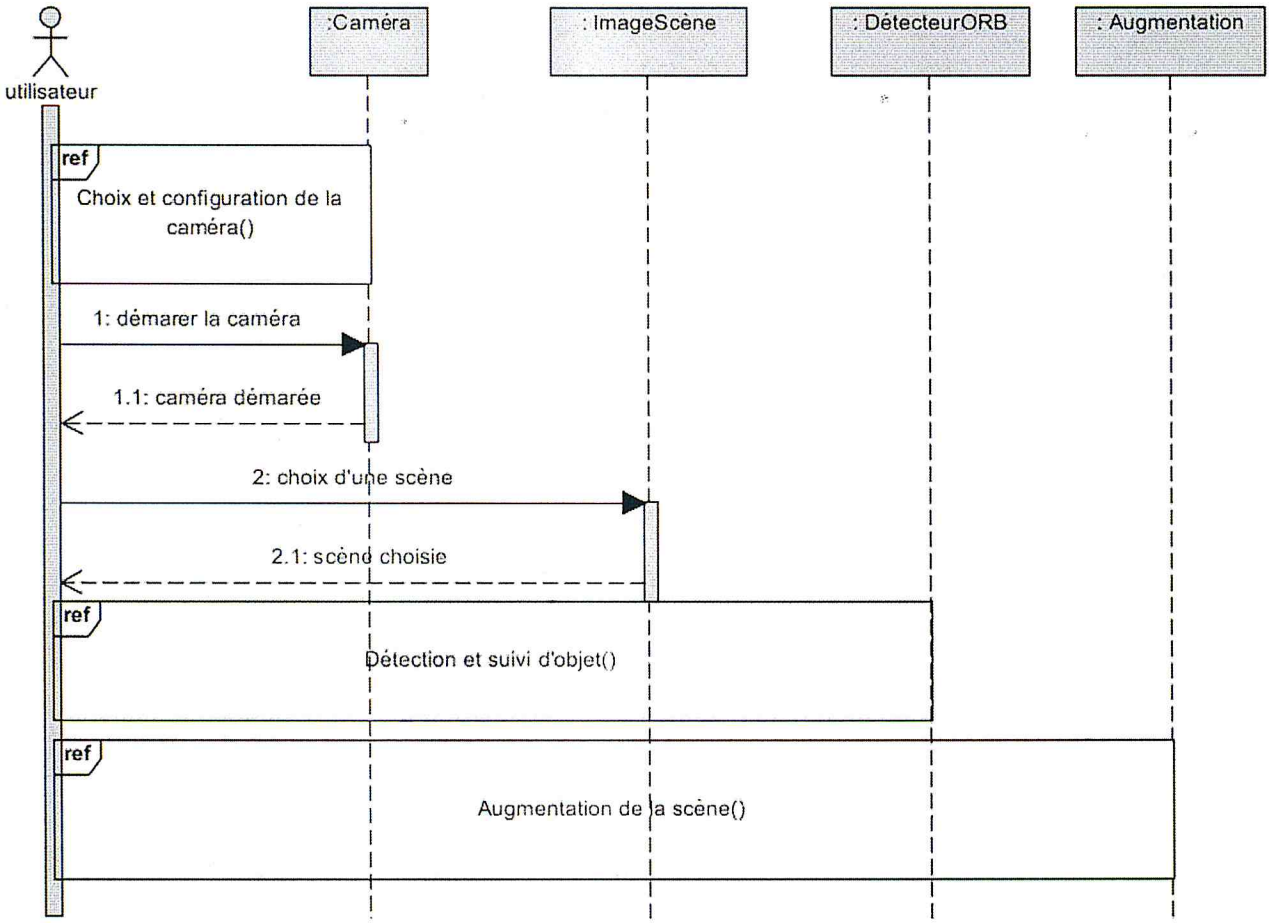


Fig. 3.7. Diagramme de séquence globale.

3.5.3.2. Diagramme de séquence de configuration et calibrage de la caméra

Ce diagramme représente les opérations de configuration et de calibrage de la caméra comme illustre la Figure 3.8. Scénario :

- ✓ Une liste de caméras détectées est proposé à l'utilisateur, ce dernier et choisir une et sélectionne une résolution.
- ✓ Si les paramètres intrinsèques ont été calculés auparavant lors d'une utilisation précédente de la caméra.
- ✓ charger les paramètres intrinsèques à partir d'un fichier. Sinon : Calibrer la camera.

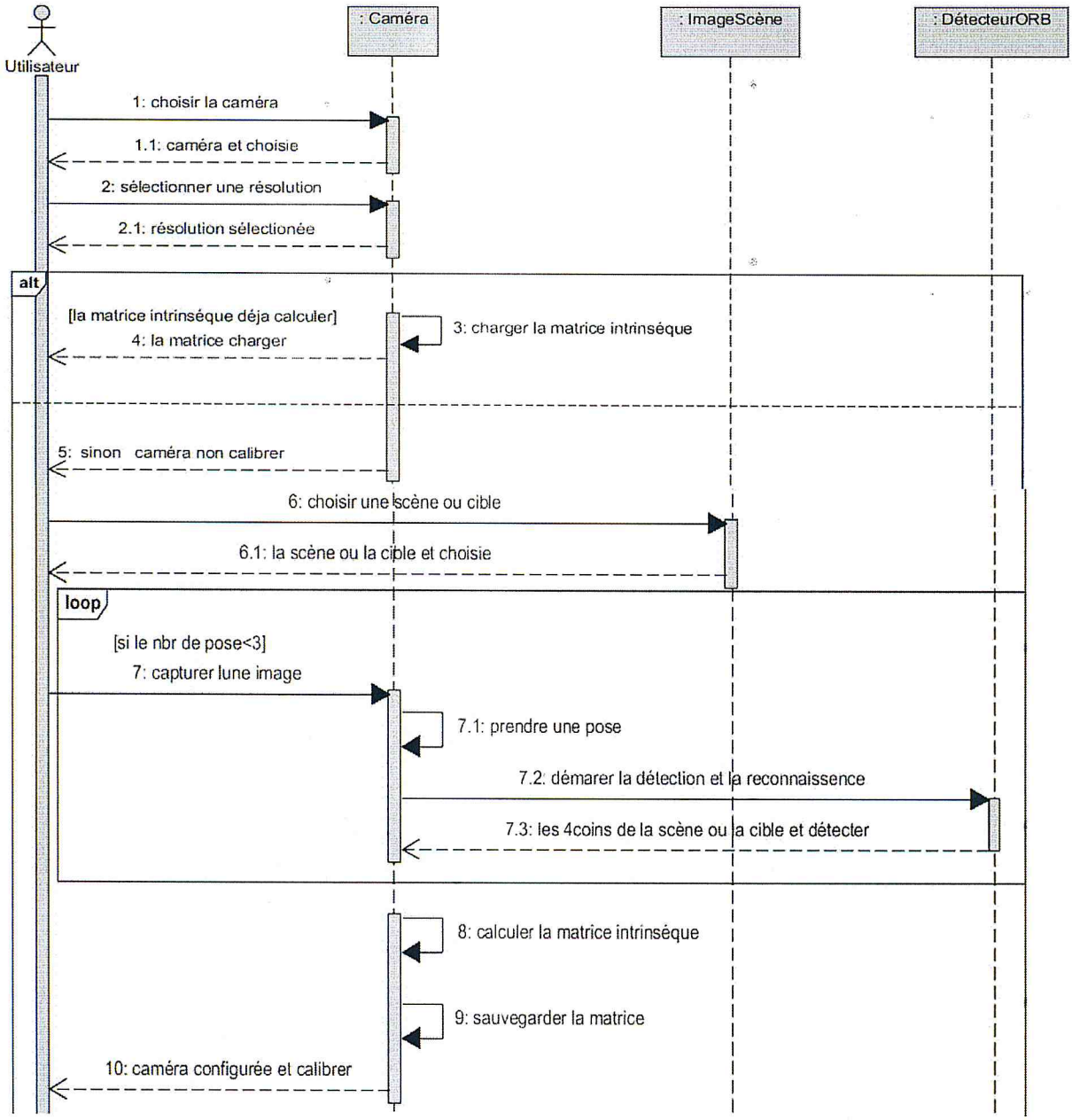


Fig. 3.8. Diagramme de séquence choix et configuration de caméra.

3.5.3.3. Diagramme de séquence détection et suivi d'objet

Ce diagramme décrit les opérations de détection et suivi d'objet comme représente la Figure 3.9. Scénario :

- ✓ Démarrer la reconnaissance (La détection de la cible ou la scène).
- ✓ Capturer à partir de la caméra une image de la scène réelle.
- ✓ Détecter les points d'intérêt dans l'image de la scène réelle.
- ✓ Appariement (la mise en correspondance) des descripteurs des points d'intérêt détectés dans la scène avec ceux détectés (image capture).
- ✓ Afficher pour l'utilisateur la scène avec l'image capturée ou bien cible reconnu.

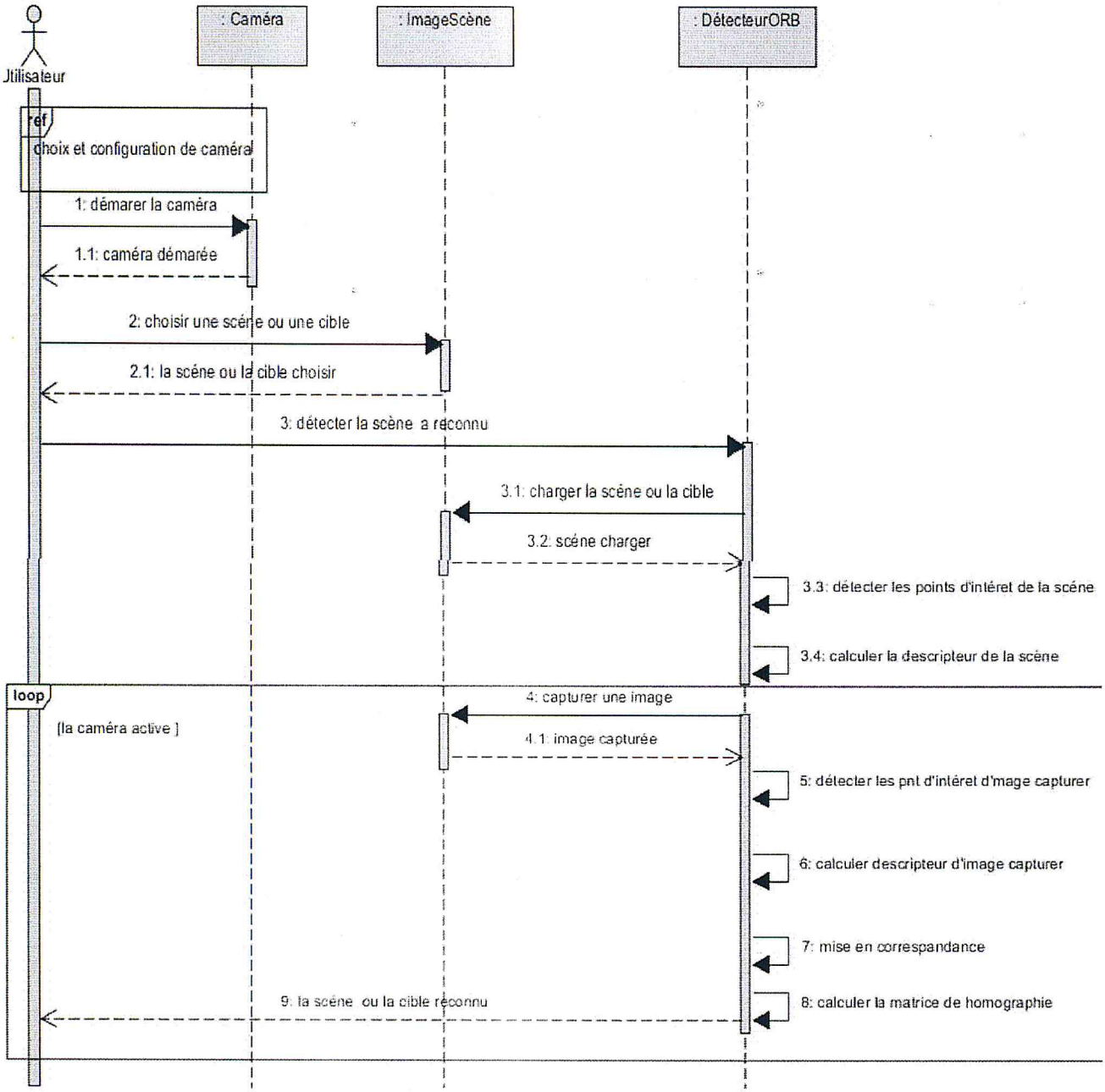


Fig. 3.9. Diagramme de séquence détection et suivi d'objet.

3.5.3.4. Diagramme de séquence Augmentation de la scène

Ce diagramme décrit les opérations de l'augmentation de la scène comme représente la Figure 3.10.

Scénario :

- ✓ L'utilisateur démarre l'augmentation.
- ✓ Une fois les coordonnées 2D de la cible calculée, la matrice de transformation est calculé par un algorithme de calibrage.
- ✓ Insertion de l'entité virtuelle dans l'image de la scène et afficher à l'utilisateur une scène de

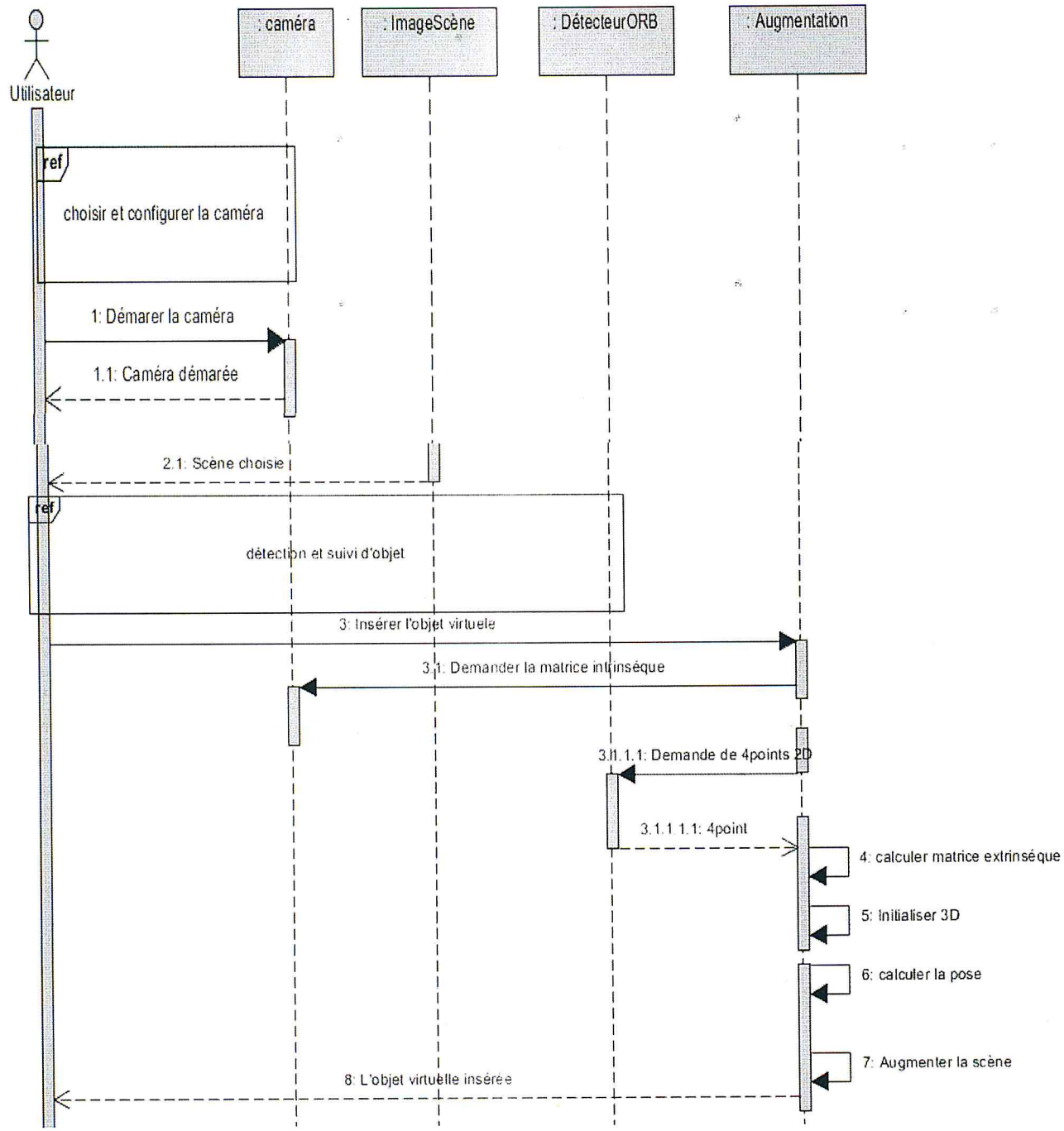


Fig. 3.10. Diagramme de séquence augmentation de la scène.

3.6. Conclusion

Dans ce chapitre, nous avons établi une conception détaillée de notre système. Durant la phase de spécification des besoins, nous avons identifié les utilisateurs, les objectifs et enfin nous avons défini le rayon des fonctionnalités de notre système. Lors de la conception, nous avons établi les diagrammes des cas d'utilisation, les diagrammes des classes ainsi que les diagrammes de séquence qui décrivent le comportement de notre système. Dans le chapitre suivant, nous allons entamer dernière phase de notre travail, c'est la phase de réalisation.

Chapitre 4

Mise en œuvre du système, application et résultats

4.1. Introduction

Ce dernier chapitre consiste la citation des différents outils utilisés pour le développement de système de suivi d'objet pour les applications de la RA. Ainsi que, la présentation des résultats obtenus de système de suivi d'objet par l'utilisation de descripteur ORB, l'évaluation de ces résultats, comparant avec la méthode SIFT et SURF et les résultats d'augmentation.

4.2. Environnement et outils de développement

Au cours de l'implémentation de notre projet, nous avons utilisé comme environnement de développement Microsoft Visual studio 2013. Nous avons implémenté les algorithmes et interfaces sous le langage C# et nous avons utilisé comme librairies externes de développement EmguCV, et DirectX. Dans ce qui suit nous expliquons le choix et l'intérêt de chaque Environnement et outil choisis.

4.2.1. Visual Studio 2013

Visual studio est un ensemble d'outils de développement d'applications de Microsoft. Il permet grâce à son environnement de développement intégré (IDE (Integrated Development Environment)) de proposer des solutions faisant appel à plusieurs langage comme Visual Basic, J#, C#, C++, etc. Par ailleurs le Framework .NET 4.0 qui est Intégré par défaut offre plusieurs fonctionnalités comme l'accès aux données, la prise en charge du protocole web, et la prise en charge de librairies dynamique (DLL (Dynamic Link Library)) de Windows, etc.

4.2.2. Langage C#

Le langage C# est un langage de programmation orienté objet. La syntaxe de ce langage est caractérisée par sa simplicité ce qui rend le processus de développement moins pénible.

A l'aide d'un processus appelé « interopérabilité », C# offre la possibilité d'interagir avec d'autres logiciels et composants Windows tel que des objets COM et DLL Win 32 native. Grace à cette fonctionnalité C# est enrichi par tous les avantages qu'offre le C++.

4.2.3. Librairies de vision par ordinateur et 3D

Nous avons eu recours à des libraires supplémentaires, comme OpenCV, EmguCV utilisées dans le domaine de la vision par ordinateur et DirectX utilisée dans le domaine de la programmation 3D.

4.2.3.1. OpenCV

OpenCV est une bibliothèque open source développée par INTEL. Elle contient plus de 500 algorithmes optimisés, destinés pour le traitement d'images et de vidéos. OpenCV est conçu principalement pour le développement d'application temps réel. Elle prend avantage des performances des nouveaux processeurs multi-cores. Un des objectifs d'openCV est d'offrir une facilité d'utilisation des grandes technologies de traitement d'images à la communauté de vision par ordinateur. Dans sa version 2.3, OpenCV introduit le module OpenCV GPU. Ce module consiste en un ensemble de classes et fonctions permettant d'accélérer les calculs en utilisant le GPU. Ces classes et fonctions ont été développées pour une utilisation sur la plateforme CUDA qui équipe les cartes graphiques NVIDIA récentes.

4.2.3.2. EmguCV

EmguCV est un WRAPPER de la bibliothèque OpenCV en C#. On y retrouve la quasi-totalité des fonctionnalités d'OpenCV permettant d'effectuer toutes les tâches basique d'analyse et de traitement d'image.

4.2.3.3. Direct X

C'est un simulateur virtuel qui fournit des outils permettant de créer des scènes virtuelles en 2D ou en 3D. Pour augmenter une scène réelle avec des objets virtuels, il faut essentiellement adapter la matrice des paramètres extrinsèques de la scène réelle avec la matrice "world" de DirectX, comme il faut aussi initialiser la matrice "projection" de DirectX avec la matrice des paramètres intrinsèques.

4.2.3.4. 3D Studio Max 2015

3D Studio Max (ou 3ds Max) est un logiciel de modélisation et d'animation 3D, développé par la société Autodesk. Il est l'un des logiciels de référence dans le domaine de l'infographie 3D.

3DsMax est ainsi conçu sur une architecture modulaire, compatible avec de multiples *plugins* (extensions) et les scripts écrits dans un langage sous licence appelé (maxscript). Le logiciel 3ds max s'est développé rapidement, en étant utilisé principalement dans le cadre du jeu vidéo. Il a également été utilisé dans d'autres domaines, notamment le film d'animation.

4.3. Implémentation et étapes de réalisation

Nous allons passer en revus dans cette Section les différentes étapes par lesquelles nous somme passer pour le développement de notre application.

4.3.1. La modélisation 3D

En premier lieu, les modèles tridimensionnels sont conçus et modéliser sous 3ds Max (voir la Figure 4.1). Des textures sont appliquées par la suite et enfin viens la définition des différentes animations pour chaque modèle toujours dans le studio 3ds Max.

On peut observer l'objet du Logo de l'Université Saad Dahleb de Blida modélisé sous différente vues dans la Figure 4.1.

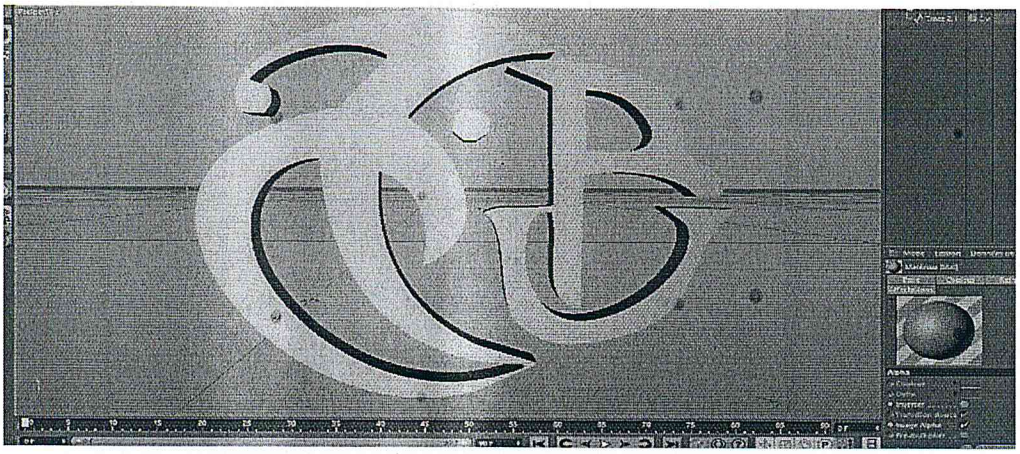


Fig. 4.1. Modèle du logo de l'Université de Blida sous 3DsMax.

4.3.2. Extraction des paramètres (points d'intérêt)

Pour ce qui est des opérations bas niveaux extraction des points d'intérêt avec ORB, nous nous somme servi de la bibliothèque OpenCV avec C# sous Visuel Studio afin d'extraire les points d'intérêt et le calcule de leurs descripteurs. Le résultat est représenté dans la Figure 4.2 qui représente la détection des points d'intérêt (en jaune) sur une image prise par la caméra sur la scène réelle.

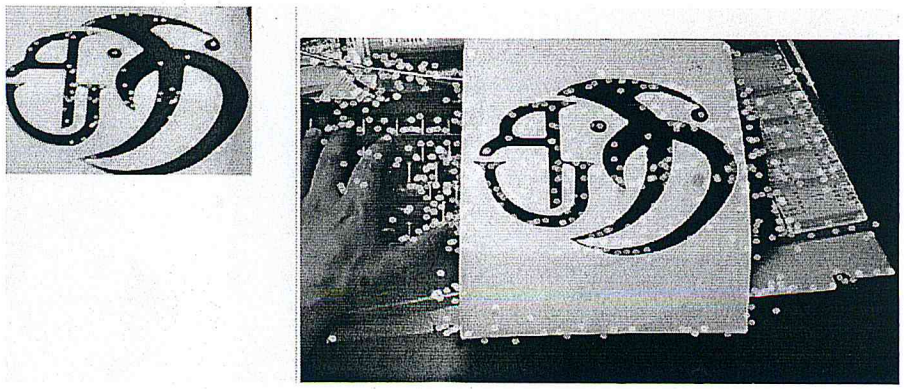


Fig. 4.2. Extraction des points d'intérêt d'une scène réelle utilisant ORB.

4.3.3. Le suivi d'objet basé sur la mise en correspondance

Le résultat de l'extraction des points d'intérêt (descripteurs) à partir de la cible naturelle (détecteur ORB) enregistrée dans un fichier XML. Pour la reconnaissance on sert de ce fichier afin de le comparé aux données de la scène réelle (mise en correspondance par la mesure de similarité). La Figure 4.3 montre comment détecter les points d'intérêt dans la scène réelle et comment fait la mise en correspondance entre l'image scène et la cible, les points d'intérêt en jaune sont ceux au quels ont à put trouver une correspondance dans le fichier, ceux en rouge sont les point sans correspondance. La Figure 4.4 montre la mise en correspondance entre les deux images, la scène réelle et l'image capturée pré-enregistré, l'objet à suivre encadré en rouge.

Une fois les points est détecté et la cible et reconnu il ne reste plus qu'à estimer sa pose tridimensionnelle dans l'environnement réelle et insérer l'objet virtuel.

Fig. 4.3. Reconnaissance dans la scène réelle.

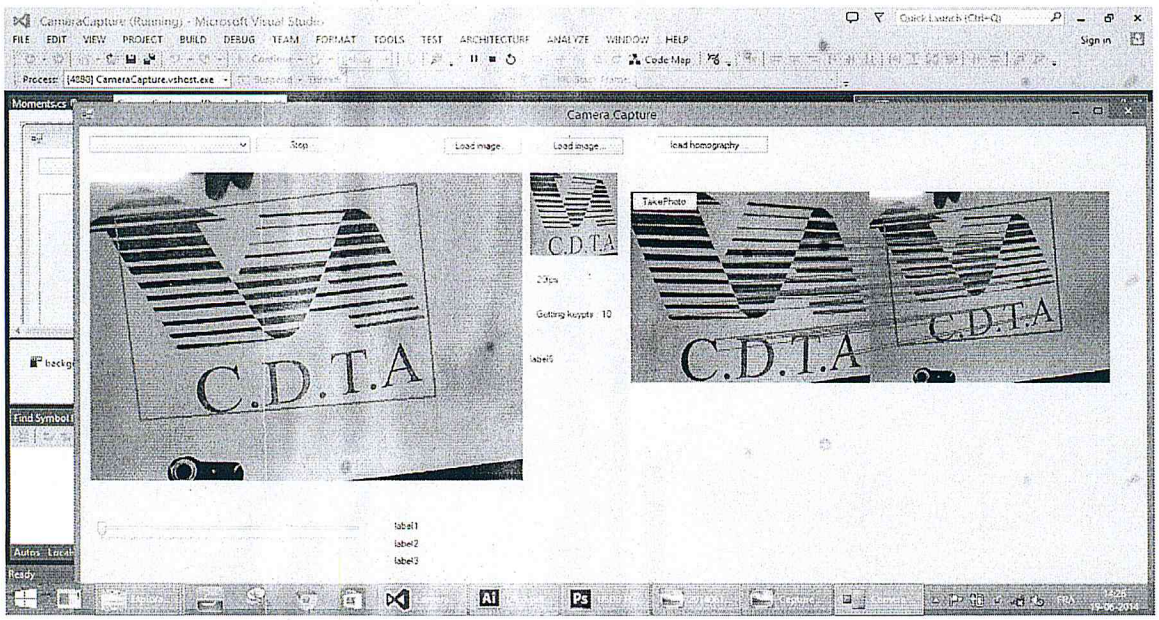


Fig. 4.4. Le suivi d'objet basé sur la mise en correspondance.

4.3.4. L'augmentation de la scène réelle (Insertion des objets 3D)

Pour pouvoir insérer un objet virtuel suivant DirectX, il faut d'abord initialiser la matrice de projection de DirectX avec la matrice des paramètres intrinsèques calculée précédemment (pour plus de détails voir l'Annexe B). Ensuite pour chaque pose de la camera il faut, mettre à jour la matrice World de DirectX avec la matrice des paramètres extrinsèques, c'est-à-dire la matrice de rotation et celle de translation renvoyé par l'algorithme de POSIT pour le calcul de pose.

Une fois la matrice World de DirectX est calculée. Nous pouvons insérer des textures, des objets virtuels en 3D dans la scène réelle. La Figure 4.5 montre l'augmentation de la scène réelle par l'insertion du Logo de l'Université de Blida (en vert).

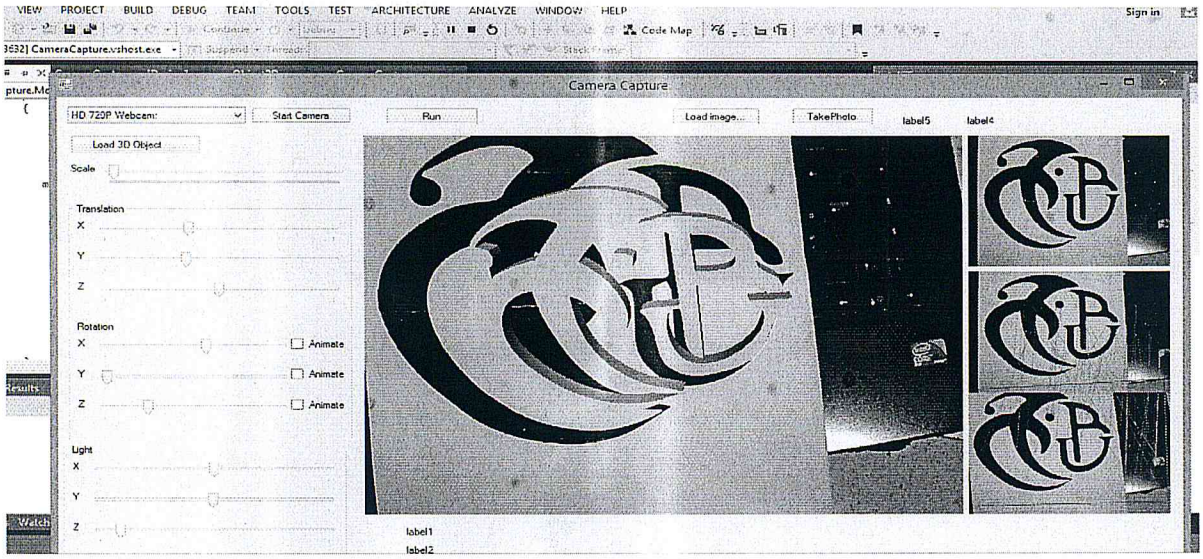


Fig. 4.5. Augmentation de scène réelle par l'insertion d'un objet virtuel.

4.4. Tests et évaluation

Pour l'évaluation des résultats expérimentaux nous avons utilisé une base de données normalisée (Dataset) [48] qui contient plusieurs images avec différents changements. Les résultats obtenus de descripteur utilisé ORB sont comparés avec SIFT et SURF. Nous avons choisi trois images de référence à partir de la base de données qui sont représentées dans la Figure 4.5.

Graffiti (6 images avec changement de point de vue).

New York (6 images avec changement de rotation).

Astérix (6 images avec changement de l'échelle).

Pour évaluer ces différents algorithmes et mettre en avant les avantages et les inconvénients de la méthode choisie, nous avons utilisé les critères suivants :

- Le temps de traitement
- La répétabilité de détecteur

- La précision et l'exactitude.

(a) Graffiti (b) New York (c) Astérix

Fig. 4.6. Images de référence utilisées [48].

4.4.1. Le temps de traitement

On général le temps de traitement dépond de deux facteurs :

- Le nombre *N* de points d'intérêt détecter qui peut être modifié de manière appropriée le seuil de détection de chaque détecteur.
- Taille de l'image entrée.

Pour l'évaluation on prend 6 images de la même taille 1024×768 pixels et on compare le temps de traitement en fonction de nombre de points détecté.

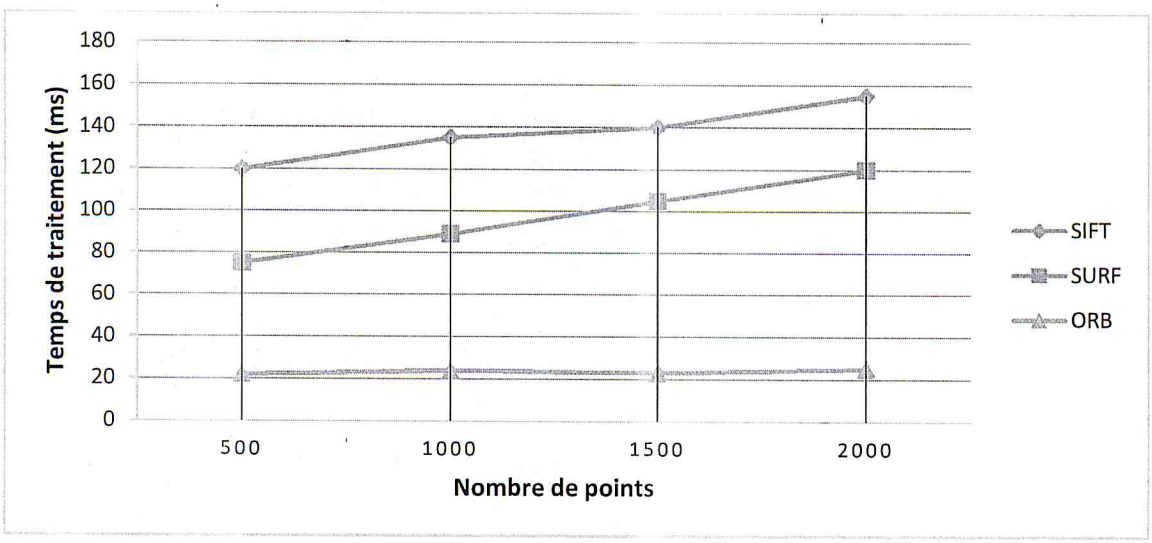


Fig. 4.7. Temps de traitement en fonction de nombre de points.

Les courbes dans la Figure 4.7 représentent le temps moyen de traitement en fonction de nombre de points détectées pour les algorithmes : SIFT, SURF et ORB. Nous remarquons que les courbes de SIFT et SURF sont linéairement croissantes, d'une manière si le nombre de points détectés augmente, le temps de traitement augmente aussi, par contre dans l'algorithme ORB le temps de traitement reste presque constant de l'ordre de 20 ms. Ces résultats montrent la rapidité de traitement de l'algorithme ORB comparant au SURF et SUFT surtout si le nombre de points détectées est important, avec le temps de traitement de SURF un peut plus rapide de SIFT.

Tableau 4.1. Comparaison entre SIFT et SURF et ORB en termes de temps de traitement.

L'algorithme	Temps de détection (ms)	Temps de description (ms)	Temps de mise en correspondance (ms)
SIFT	33	43 ,45	59
SURF	74	13,43	54
ORB	29	1,36	53

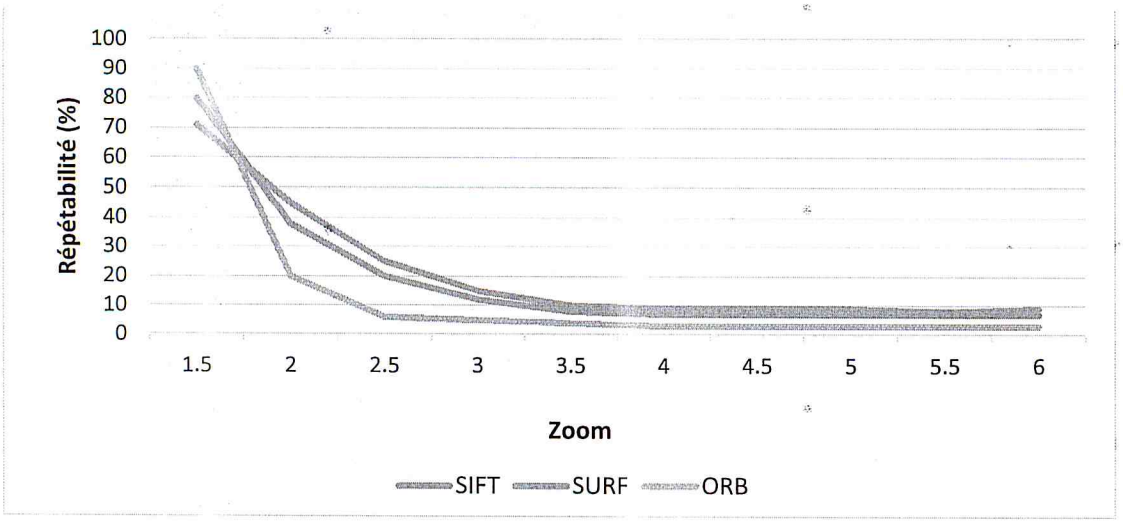
Le Tableau 4.1 représente une comparaison en termes de temps de traitement qui consiste : le temps de détection, le temps de description et le temps de mise en correspondance pour les algorithmes SIFT, SURF et ORB. Les résultats du tableau montrent la rapidité de l'algorithme ORB dans toutes les phases de traitement, surtout dans la phase de description, par ce que se basé sur la description binaire, qui est simple et rapide. SURF est rapide que SIFT dans la phase de description parce que il utilise l'image intégrale. Toutes ces remarques montrent la supériorité de l'algorithme ORB en termes de temps de traitement.

4.4.2. Répétabilité

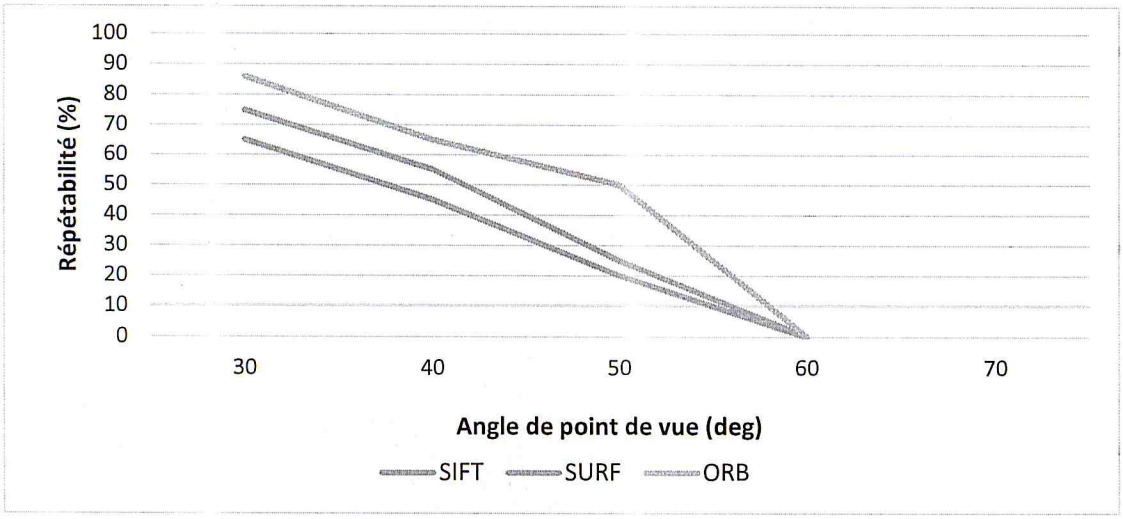
Le détecteur ORB est également évalué à la répétabilité, on prendre des séquences d'image pour faire la comparaison entre eux. Chaque séquence inclut une image de référence et un ensemble d'image qui sont progressivement modifié par un ou plusieurs transformations géométriques et photométriques.

L'analyse de répétabilityé nous permet de mettre en avant la pertinence des points extraits et de choisir en conséquence le procédé le plus adapté à nos besoins. D'un point de vue théorique la répétabilité est définie par [29] :

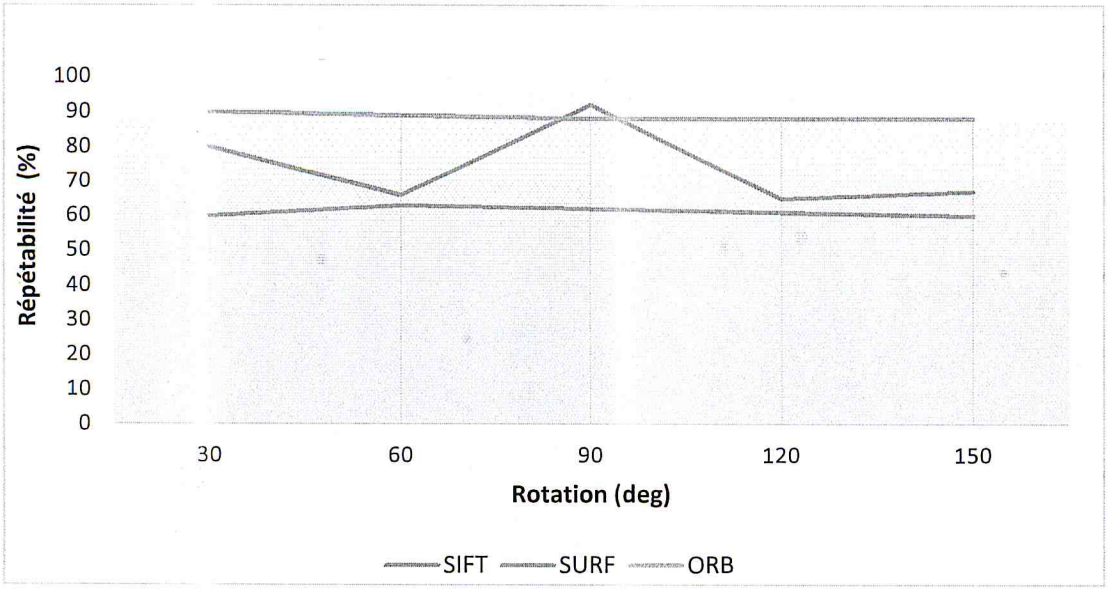
$$\text{Répétabilité} = \frac{\text{Nombre de rétro} - \text{projections correctes}}{\text{Nombre de points détéctés}} \qquad (4.1)$$



(a) Image Astérix (changement d'échelle)



(b) Image Graffiti (changement de point de vue)



(c) Image New York (rotation)

Fig. 4.8. Score de répétabilité pour différentes séquences d'images.

La Figure 4.8 représente le score de répétabilité moyen pour différentes séquences d'image avec différentes changement. Ces résultats montrent un meilleur score de répétabilité de détecteur ORB pour le changement de point de vue et la rotation comparant aux détecteurs SURF et SIFT, tel que ORB attient le maximum score de 90%, d'un autre coté ORB est sensible au changement d'échelle, pour un changement de factor supérieur au 2.5 le score de répétablilité est proche de zéros. L'avantage major de détecteur ORB c'est invariance à la rotation, tel que le score de répétabilité est de l'ordre de 90% pour un changement de rotation entre 30 et 150 deg.

4.4.3. La précision et l'exactitude

La précision dans la mise en correspondance entre deux images I_{m1} et I_{m2} est définie par :

$$\text{Précision} = \frac{\text{Nombre d'appariément correct } (I_{m1}, I_{m2})}{\text{Nombre de correspondance trouvé } (I_{m1}, I_{m2})} \tag{4.2}$$

L'exactitude est défini par :

$$\text{Exactitude} = \frac{\text{Nombre d'appariément correct } (I_{m1}, I_{m2})}{\text{Nombre de correspondance possible } (I_{m1}, I_{m2})} \tag{4.3}$$

Nous avons utilisé ces deux critères pour évaluer les trois descripteurs et les résultats sont représentés dans la Figure 4.9.

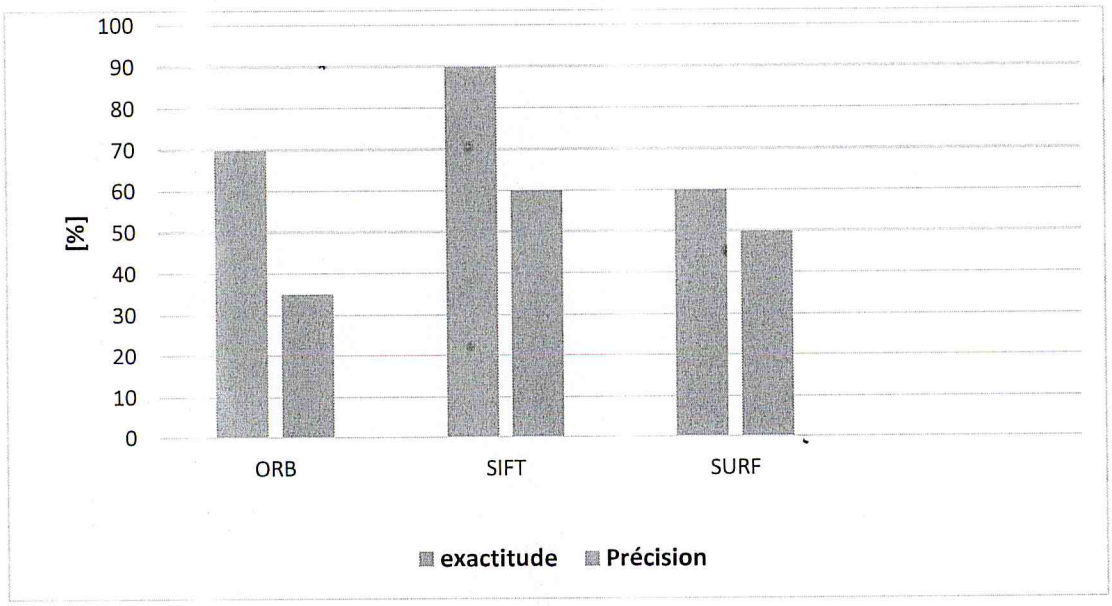


Fig. 4.9. Evaluation d'exactitude et de précision.

Les résultats de la Figure 4.9 montrent des meilleures précisions et exactitude pour le descripteur SIFT comparant avec SURF et ORB. Ainsi que meilleur exactitude de descripteur ORB attient 70% par rapport au SURF, et le contraire en termes de Précision. Ces résultats démontrent l'avantage des

descripteurs qui sont basés sur l'histogramme du gradient (SURF et SIFT) en termes de précision par rapport aux descripteurs binaires.

4.5. Conclusion

Dans ce chapitre, nous avons réalisé une application qui permet de reconnaitre et de suivre un objet dans une scène réelle, avec l'insertion des objets virtuels.

Nous avons présenté aussi une évaluation de détecteurs et descripteurs de points d'intérêt (SIFT, SURF et ORB) avec différentes changement de l'image. Les résultats obtenus montrent la supériorité de l'algorithme ORB comparant aux descripteurs SURF et SIFT en termes de rapidité et la répétabilité de son détecteur avec quelques inconvénients aux changements d'échelle et la précision.

Le compromis est toujours existe entre la complexité de calcul et la précision. Donc sa dépend de l'application, le choix de descripteur reste une étape primordiale dans un système de suivi basé sur les points d'intérêt.

Conclusion générale

La réalité augmentée est un type d'application du domaine de vision par ordinateur. Elle consiste à ajouter des objets virtuels dans le monde réel pour augmenter la perception visuelle de l'observateur. Les applications de cette technologie sont nombreuses tel que : la maintenance, la médecine, la robotique, l'architecture, etc. L'évolution de la RA dans ces dernières années est assez rapide avec émergence de la technologie : les tablets, les smartphones, les HMDs, etc., Ainsi que le développent des applications de RA sous Androïde et iOS pour faciliter l'utilisation et assurer la mobilité de l'utilisateur.

Assurer une cohérence entre les objets réels et virtuels est un challenge dans le domaine de la réalité augmentée. Une cohérence spatiale, temporelle, et photométrique pour rendre l'objet virtuel réaliste le plus possible. On distingue deux type de réalité augmentée : avec marqueur et sans marqueur (marqueur naturel), selon le critère de détermination de point de vue.

Actuellement, la plupart des applications de réalité augmentée utilisent des marqueurs, ou des tags, qui sont prédéfinis et dont la forme est connue et simple à reconnaitre. Cela permet d'avoir un traitement rapide. Mais le problème de ces méthodes de reconnaissance qui sont basées sur les marqueurs souffrent certaines limitations, comme par exemple : la sensibilité aux occlusions partielles et les changements d'illumination, tous ça influent sur la stabilité de l'objet virtuel dans la scène réelle augmentée.

Un système de suivi d'objet généralement réalisé pour le but d'assurer un recalage correct entre l'objet réel et virtuel en temps réel (l'objet virtuel est bien positionné par rapport au camera). Le système a réalisé basé sur des primitives visuelles (la mise en correspondance des points d'intérêts) qui sont extrait directement à partir de la scène réelle (sans marqueurs). Le choix de détecteur et descripteur des points d'intérêt est primordial pour l'extraction des paramètres significatifs.

Dans ce travail, nous avons réalisé un système de suivi d'objet rigide basé sur la mise en correspondance des points d'intérêt utilisant l'algorithme ORB. Ce système est réalisé pour les applications de réalité augmentée, une fois, on la possibilité de suivre un objet réel dans la séquence vidéo, en peut ajouter l'objet virtuel d'une manière correcte sur la scène réelle.

Une étude bibliographique à été fait sur les définitions de réalité augmentée, les différents domaines de l'application, le concept d'un système de réalité augmentée, les dispositifs utilisés dans ce domaine, avec leurs différentes problématiques.

Ensuite, une étude théorique à été exposé sur la détection et la description des points d'intérêt : les différents détecteurs et descripteurs de points d'intérêt existent dans la littérature. Ainsi que, les différentes étapes utilisées pour le suivi d'objet à base de la mise en correspondance des points d'intérêt.

Après l'étude théorique La conception de notre système à été fait pour répondre à la problématique posée en utilisant les notations UML. Les différents diagrammes de conception sont exposés en détail pour la réalisation d'un système de suivi d'objet pour les applications de réalité augmentée.

Finalement, la réalisation à été fait d'un système de suivi d'objet baser sur ORB, ainsi que l'augmentation de la scène réelle par l'insertion des objets virtuels. Les résultats de l'évaluation entre ORB, SURF et SIFT ont montré de meilleures performances de l'algorithme ORB comparant au SURF et SIFT en termes de rapidité (complexité de calcul réduite), l'invariance à la rotation et la répétabilité de détecteur. Ces performances sont importantes pour le recalage réel-virtuel dans la réalité augmentée. D'un autre coté, quelques points faibles en termes de précision et exactitude, un compromis entre la complexité de calcul et la précision.

En perspective de ce travail, et pour améliorer les performances de ce système, nous avons proposé d'utiliser une région d'intérêt autour de l'objet à suivre pour détecter les points d'intérêt seulement à l'intérieur de cette région d'une manière à optimiser le temps de calcul. Ainsi que la proposition des approches de l'algorithme ORB pour assurer l'invariance au changement d'échelle et améliorer le taux de reconnaissance.

Bibliographie

[1] R.T. Azuma, "A survey of augmented reality," Presence: Teleoperators and Virtual environments, Vol. 6, no. 4, pp. 355 - 385, 1997.

[2] M. Maidi, "Suivi hybride en présence d'occultations pour la réalité augmentée," Thèse de Doctorat en Robotique, Université d'Evry Val d'Essonne, 2007.

[3] W. E. Mackay, "Réalité Augmentée : le meilleur des deux mondes," La Recherche, no. 285, pp. 32-37, 1996.

[4] W. E. Mackay, "Augmented Reality: Linking real and virtual worlds A new paradigm for interacting with computers," In Proceedings of the working ACM conference on Advanced Visual Interfaces (AVI'98), pp. 13-21, Italy, 1998.

[5] J. Bowskill, J. Morphett, J. Downie, "A taxonomy for enhanced reality systems," In Proceedings of International Symposium on Wearable Computers (ISWC'97), pp. 175-176, 1997.

[6] P. Milgram, F. Kishino, "A taxonomy of mixed reality visual displays," In IEICE Trans on Information Systems, Vol. 77, no. 12, pp. 1321-1329, 1994.

[7] R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, "Recent advances in augment reality," Computer Graphics and Applications, IEEE, Vol. 21, no. 6, pp 34-47, 2001.

[8] J. Landrieu, "Apports des réalités virtuelles et augmentées dans la planification et le suivi in situ de travaux de rénovation," Thèse de Doctorat en Informatique-Traitement du signal, L'École Nationale Supérieure d'Arts et Métiers,Paris Institut des sciences et technologies, 2013.

[9] M. Chouiten, "Architecture distribuée dédiée aux applications de Réalité Augmentée mobile," Thèse de Doctorat en Informatique, Université d'Evry Val d'Essonne, 2013.

[10] P. Milgram, H. Takemura, A. Utsumi, F Kishino, "Augmented Reality: a class of displays on the reality-virtuality continuum," In Photonics for Industrial, pp. 282-292, International Society for Optics and Photonics, 1994.

[11] M. Bouzenada, "Incrustation d'objets virtuels dans des séquences vidéo pour la réalité augmentée en temps réel," Thèse de doctorat en Informatique, Université Mentouri, Constantine, 2008.

[12] L. Masson, "Suivi temps-réel d'objets 3D pour la réalité augmentée," Thèse de doctorat en Informatique, Université Blaise Pascal, Clermont-Ferrand II, 2005.

[13] http://www-igm.univ-mlv.fr/~dr/XPOSE2009/Xpose-Kevin-Le-Jannic Promo2007/rendu_image.html. Consulter le 02/06/2014.

[14] B. Furht, "Handbook of augmented reality," Ed. Springer, 2011.

[15] A. B. Craig, "Understanding Augmented Reality: Concepts and Applications," Ed. Elsevier, Newnes, 2013.

[16] K. Ben abderrahim, M. Kallel, M. S. Bouhlel, "Towards an interactive medical system by augmented reality," International Journal of Computer Applications & Information Technology, Vol. 2, pp. 26-30, 2013. Disponible sur : http://www.ijcait.com/IJCAIT/index.php/www-ijcs/article/viewFile/258/136.

[17] C. Bichlmeier, N. Navab, "Virtual window for improved depth perception in medical AR," In International Workshop on Augmented Reality environments for Medical Imaging and Computer-aided Surgery (AMI-ARCS).

[18] N. Zenati-Henda, "Contribution à la conception et à la réalisation d'un système de réalité augmentée pour la maintenance," Doctoral dissertation in automatic, Ecole Centrale de Lille, 2008.

[19] S. Feiner, B. Macintyre, D. Seligmann, "Knowledge-based augmented reality," Communications of the ACM, Vol. 36, no. 7, pp. 53-62, 1993.

[20] P. Milgram, S. Zhai, D. Drascic, J. J. Grodski, "Applications of augmented reality for human-robot communication," In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems' 93, IROS'93, Vol. 3, pp. 1467-1472, 1993.

[21] W. Broll, I. Lindt, J. Ohlenburg, M. Wittkämper, C. Yuan, T. Novotny, A. Strothman, "Arthur: A collaborative augmented environment for architectural design and urban planning," Journal of Virtual Reality and Broadcasting, Vol. 1, no. 1, pp. 1-10, 2004.

[22] I. E. Sutherland, "The ultimate display," Multimedia: From Wagner to virtual reality," pp. 506-508, 1965.

[23] http://www.vrealities.com/head-mounted-displays. Consulté le 02/06/2014.

[24] http://www.fnac.com/Google-Glass-des-lunettes-connectees-en-realite-augmentee/cp23488/w-4. Consulté le 02/06/2014.

[25] H. Kato, M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," In Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality, (IWAR'99), pp. 85-94, 1999. ARtoolKit est disponible dans : http://www.hitl.washington.edu/artoolkit

[26] http://www.arlab.com/blog/markerless-augmented-reality/. Consulté le 03/06/2014.

[27] C. Schmid, R. Mohr, "Local gray value invariants for image retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, no. 5,pp. 530-534, 1997.

[28] H. P. Moravec, "Towards automatic visual obstacle avoidance," In Proceeding of Fifth International Joint Conference on Artificial Intelligence, Cambridge Massachusetts USA, pp. 584-587, 1977.

[29] M. Grand-brochier, "Descripteurs 2D et 2D+t de points d'intérêt pour des appariements robustes," Thèse de doctorat en Visio pour la Robotique, Université Blaise Pascal - Clermont II, 2011.

[30] C. Harris, M. Stephens, "A Combined corner and edge detector," In Proceeding of Forth Alvey Vision Conference, Vol. 15, pp. 147-151, 1988.

[31] http://mmlab.ie.cuhk.edu.hk/archive/gbq/csc5280_project_2.htm. Consulté le 04/06/2014.

[32] E. Rosten, T. Drummond, "Fusing points and lines for high performance tracking," In IEEE International Conference on Computer Vision, Vol. 2, pp. 1508–1511, 2005.

[33] B. Lefaudeux, "Détection, localisation et suivi des obstacles et objets mobiles à partir d'une plate forme de stéréo-vision," Thèse de doctorat en Informatique temps réel, robotique et automatique, Ecole nationale supérieure des mines de Paris, 2013.

[34] T. Chesnais, "Contextualisation d'un détecteur de piétons : Application à la surveillance d'espaces publics," Thèse de doctorat en Vision pour la robotique, robotique et automatique, Université Blaise Pascal - Clermont II, 2013.

[35] C. Maaoui, H. Laurent, B. Emile, "Reconnaissance et détection robuste d'objets couleur. In 20ème Colloque sur le traitement du signal et des images, FRA, GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2005

[36] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," In Proceedings of the International Conference on Computer Vision (ICCV), pp.1-8, 1999.

[37] S. Zhao, "Apprentissage et recherche par le contenu visuel de catégories sémantiques d'objets vidéo," Mémoire de Master en Informatique, université Paris Descartes, 2007.

[38] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding (CVIU), Vol. 110, no. 3, pp. 346-359, 2008.

[39] M. Calonder, V. Lepetit, C. Strecha, P. Fua, "Brief: Binary robust independent elementary features," In Computer Vision–ECCV, Springer Berlin Heidelberg, pp. 778-792, 2010.

[40] M. Kottman, "Improving binary feature descriptors using spatial structure," Information Sciences & Technologies: Bulletin of the ACM Slovakia, Vol. 5, no. 2, 2013.

[41] http://gilscvblog.wordpress.com/2013/09/19/a-tutorial-on-binary-descriptors-part-2-the-brief-descriptor/. Consulté le 05/06/2014.

[42] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: an efficient alternative to SIFT or SURF," In IEEE International Conference on Computer Vision (ICCV), pp. 2564-2571, 2011.

[43] S. Leutenegger, M. Chli, R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," pp. 1-8, 2011.

[44] A. Alahi, R. Ortiz, P. Vandergheynst, "Freak: Fast retina keypoint," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510-517, 2012.

[45] http://gilscvblog.wordpress.com/2013/10/04/a-tutorial-on-binary-descriptors-part-3-the-orb descriptor/. Consulté le 05/06/2014.

[46] M. A. Fischler, R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," ACM, Vol. 24, no. 6, pp. 381–395, 1981.

[47] L. Audibert, "UML 2: De l'apprentissage à la pratique," Ellipses, Vol. 298, 2009.

[48] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, no. 10, pp. 1615–1630, 2005. Disponible dans : http://lear.inrialpes.fr/people/mikolajczyk/Database/index.html. Consulté le 10/06/2014.

[49] Y. I. Abdel-Aziz, H. M. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," In Proceedings of the ASP/UI Symposium on Close-Range photogrammetry, 1971.

Annexe A

ARToolKit

A.1. Introduction

ARToolKit (Augmented Reality ToolKit) est une librairie software open-source utilisée pour concevoir des applications de réalité augmentée. Elle a été développée en 1999, par le docteur Hirokazu Kato de l'université d'Osaka au Japon. Puis les recherches ont été poursuivies par "the Human Interface Technology Laboratory" (HITLab) à l'université de Washington, et HITLab NZ à l'université de Canterbury en Nouvelle Zélande.

ARToolKit utilise les techniques de recalage basées vision pour calculer la position et l'orientation réelles de la caméra relative à des marqueurs. Le programmeur peut utiliser cette information pour dessiner l'objet 3D et l'aligné correctement à l'objet réel. La librairie ARToolKit possède plusieurs types de marqueurs. Ceci dépend de la forme de l'objet virtuel incrusté.

En plus, ARToolKit garantit le suivi de l'objet virtuel ajouté lorsque la caméra (ou l'utilisateur) change de position. Les marqueurs sont sous forme d'un carré noir avec plusieurs motifs comme montre la Figure A.1. Ce type de marqueur est une forme simple qui peut être facilement identifié pour l'incrustation de l'objet virtuel. La position et l'orientation de la caméra peuvent être calculées par l'identification de marqueurs dans un flux vidéo.

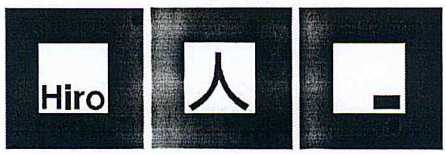


Fig A.1. Exemple de marqueurs utilisés par ARToolKit.

A.2. Fonctionnement d'ARToolKit

L'algorithme de détection d'ARToolKit effectue plusieurs étapes avant d'arriver au résultat désiré. Les étapes fondamentales sont : Acquisition, Détection, Localisation spatiale, et Traitement. Les étapes d'exécution sont illustrées dans la figure suivante :

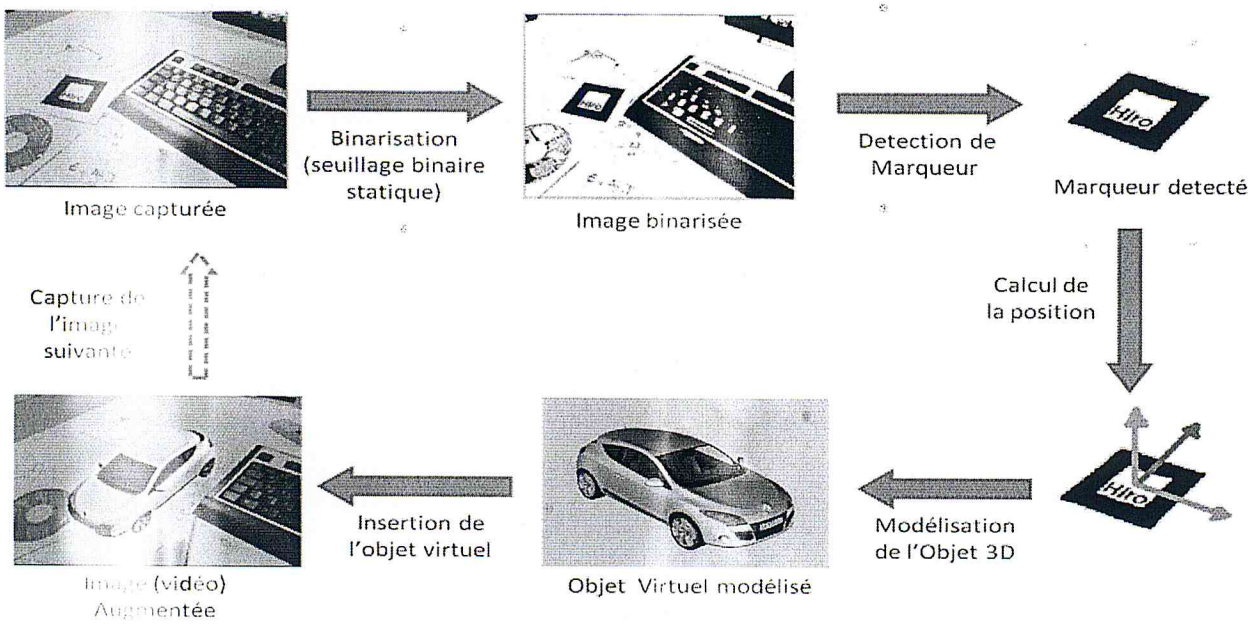


Fig. A. 2. Fonctionnement global d'ARToolKit.

- Etape 1 : Acquisition numérique d'un flux vidéo à l'aide d'une caméra

  Une image du monde réel est capturée grâce à un dispositif vidéo en entrée (webcam, caméra, …etc.).

- Etape 2 : Binarisation et détection des carrés noirs (marqueurs)

  Cette étape consiste à "binariser" l'image reçue. Il s'agit d'obtenir une image en noir et blanc pures (pas de nuances de gris intermédiaires).

  Une fois la binarisation est effectuée par seuillage de la luminosité, l'algorithme effectue une opération de détection de contours pour extraire des carrés noirs qui représentent les contours des marqueurs existant dans la scène.

- Etape 3 : Localisation spatiale

  Une fois le carré noir détecté, la librairie calcule mathématiquement la position et l'orientation de la caméra par rapport à ce carré (matrice de 3x4 de passage de la caméra au pattern).

- Etape 4 : Identification du motif à l'intérieur du marqueur

  Le système cherche une correspondance entre le motif à l'intérieur du carré et ceux déjà présents en mémoire afin de connaitre l'augmentation qui lui est associée.

- Etape 5 : Traitement de l'objet 3D et affichage du rendu graphique

  Pendant cette étape, un objet 3D est généré. Cette augmentation est ensuite superposée à l'image capturée, positionnée, orientée et mise à l'échelle suivant les données fournies par le marqueur.

Malgré les avantages que présente ARToolKit dans l'incrustation des objets 3D dans une scène réelle, certaines performances restent à améliorer. Ainsi, ARToolKit présente des limites en termes de reconnaissance de la scène (problèmes de changement de luminosité) et de stabilité de l'objet virtuel incrusté.

Divers outils permettent de créer un pattern. Nous avons pour notre part utilisé Marker Generator une application Air mise à disposition par Saqoosha.net permettant de générer un fichier .patt à la dimension souhaitée.

Annexe B

Calibrage de caméra

B.1. Modèle de projection perspective: (Le modèle sténopé)

Le modèle de caméra le plus fréquemment utilisé dans les systèmes de vision par ordinateur [46] est le modèle de projection perspective; L'hypothèse géométrique fondamentale de ce modèle consiste à supposer que tous les rayons qui lient un point dans l'espace avec sa projection correspondante sur le plan image concours vers un point nommé : "*le centre de projection perspective*".

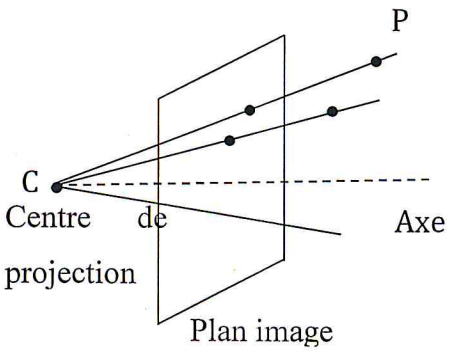


Fig B.1. Le centre de projection perspective "C".

De manière générale, les coordonnées d'un point dans l'espace P(X, Y, Z) et sa projection sur le plan image Q(u, v) sont liées par la matrice de projection M selon la relation [48]:

$$Q = M P \tag{4.1}$$

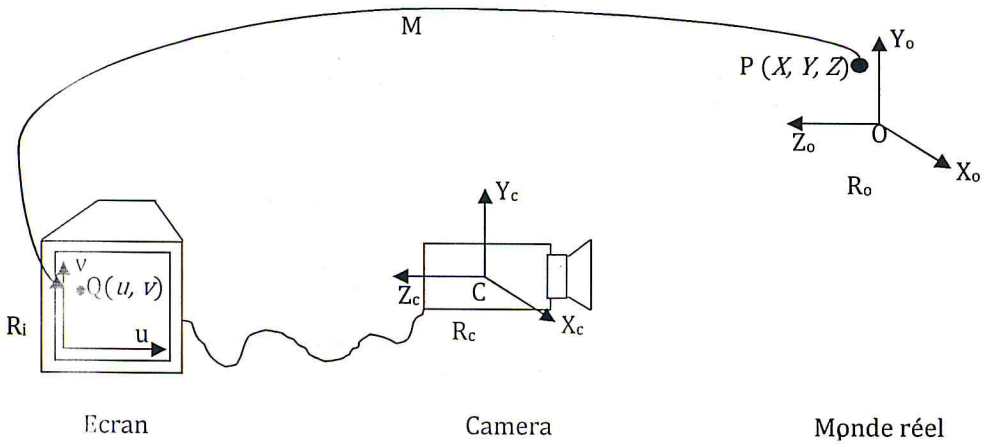


Fig. 4.2 . Les différents repères employés pour la calibration de la camera.

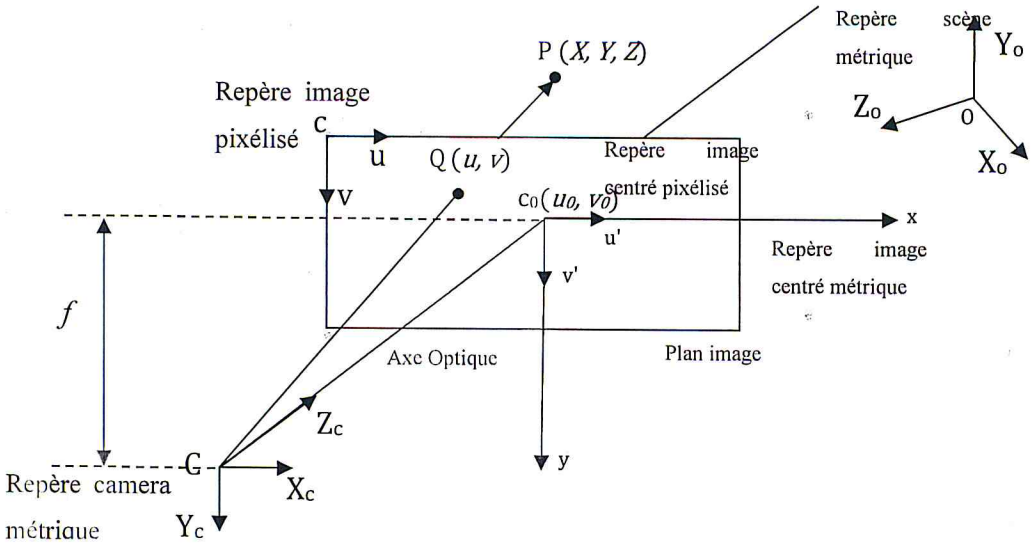


Fig. 4.3. Le modèle sténopé simple.

La Fig. 4.3 décrit le modèle sténopé; un point P de l'espace 3D forme une image Q dans le plan rétinien (plan image). C représente le centre optique de la camera. L'axe "CZ_c" perce le plan rétinien au point c_0 de coordonnées (u_0, v_0). La distance "Cc_0" est la distance focale de la camera.

B.1.1. Les paramètres externes

Si le point P a pour coordonnées (X,Y,Z) dans le repère (O,X_o,Y_o,Z_o) de la scène, ses coordonnées (X_c,Y_c,Z_c) dans le repère (O,X_c,Y_c,Z_c) de la camera sont données par la relation:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathrm{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathrm{t} = (\mathrm{R}\ \mathrm{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.2}$$

Ou (R t) exprime le déplacement rigide entre les deux repères (rotation et translation). La rotation R est souvent exprimée en fonction des angles γ, β, α autour respectivement des trois axes X_o,Y_o,Z_o :

$$\mathrm{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

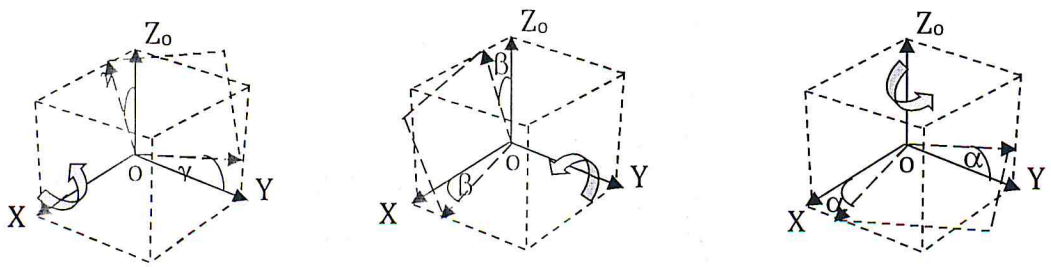


Fig. 4.4. Représentation des rotations d'angles γ,β,α sur les axes respectif X_o,Y_o et Z_o.

Ces angles sont appelés angles d'Euler. Les paramètres du changement de repère sont donc au nombre de six : les trois angles d'Euler de R et les trois composantes du vecteur de translation t. Ces paramètres, définissants l'orientation et la position de la camera dans le repère de la scène, sont les paramètres externes (appelés également paramètres extrinsèque) de la camera.

B.1.2. Les paramètres internes

Soit (C, X_c, Y_c, Z_c) le repère 3D lié à la caméra et soit (o, x, y) le repère 2D du plan image (voir Fig. 4.3). On a les relations suivantes :

$$\frac{f}{Z_c} = \frac{x}{X_c} = \frac{y}{Y_c} \tag{4.4}$$

Si on change les unités de mesure de l'axe des x et des y sur le plan image (ce qui correspond à l'échantillonnage : (Fig. 4.5) :

$$x \Rightarrow \frac{u}{k_u}$$

$$y \Rightarrow \frac{v}{k_v}$$

et on translate l'origine :

$u \longrightarrow u - u_0$

$v \longrightarrow v - v_0$

on a les relations suivantes :

$$x = \frac{u - u_0}{k_u} \tag{4.5}$$

$$y = \frac{v - v_0}{k_v} \tag{4.6}$$

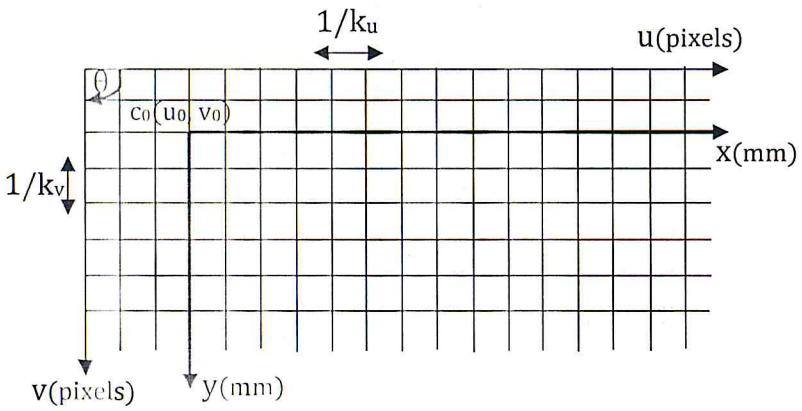


Fig. 4.5. Les paramètres internes.

Avec :

- k_u et k_v sont le nombre de pixels par unité de longueur suivant chacun des axes (>0).

- u_0 et v_0 les coordonnées pixel du point principal c (intersection de l'axe optique $(C, \overrightarrow{k_c})$ avec le plan image).
- θ l'angle entre les deux axes du repère image.

Les paramètres k_u, k_v, f, u_0, v_0, θ sont les paramètres internes (ou intrinsèque) de la camera. En pratique, l'angle θ est très bien contrôlé et peut être considéré égal à $\frac{\Pi}{2}$. D'autre part, il n'est pas possible de séparer les paramètres k_u et k_v de la distance focale f, on pose alors :

$$\alpha_u = k_u.f \tag{4.7}$$

$$\alpha_v = k_v.f \tag{B.8}$$

Nous considérons donc le modèle simplifié à quatre paramètres α_u, α_v, u_0 et v_0. D'après les équations (B.2), (B.4), (B.5), (B.6), (B,7) et (B.8), nous avons finalement

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = A \underbrace{(R\ t)}_{T} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{B.9}$$

On note "A" la matrice des paramètres internes, et "T" la matrice des paramètres externes. La matrice M = A (R t) est appelée matrice de projection perspective : elle permet d'exprimer directement la projection d'un point 3D de la scène en coordonnées pixel de l'image. Il s'agit d'une matrice 3×4 définie à un facteur d'échelle près, et possédant 11 paramètres indépendants.

$$M = A\,T = \begin{pmatrix} \alpha_v & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix}$$

$$= \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} = \begin{pmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{pmatrix} \tag{4.10}$$

Avec : « r_i » le $i^{ème}$ vecteur ligne de la matrice de rotation « R » et « m_i » le $i^{ème}$ vecteur ligne de la matrice « M » privé de la dernière coordonnée.

D'après (B.9) et (B.10) : Tout point $P_i(X_i, Y_i, Z_i)$ dans l'espace et sa projection sur le plan image $Q_i(u_i, v_i)$ sont liés par la relation:

$$s\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \tag{4.11}$$