

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMARI de Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département Informatique

Mémoire de fin d'étude

En vue d'obtention du diplôme Master II
en Informatique Spécialité Systèmes
Informatiques

Thème

Réalité Augmenté :
**Présentation d'objets virtuels
sur une scène réelle**

Proposé par :

Mr. DIB Mohammed

Mr. HOCINI Lotfi

Réalisé par :

Mr. DOUFENE Said

Mr. HADJAB Rachid

Année 2012/2013

Remerciements

Nous exprimons nos remerciements et nos gratitudees à notre promoteur Mr Dib M et Mr Hocini L pour l'aide qu'ils nous ont apporté tout au long de ce travail.

Nos plus vifs remerciements vont aussi aux membres de jury pour nous avoir fait l'honneur de juger ce travail.

Nous réservons ici une place particulière pour remercier vivement nos familles pour leur affection et leur soutien continu. Et à tous ceux qui, d'une manière ou d'une autre, nous ont aidés et encouragés à la réalisation de ce modeste travail.

En bref, merci à tous et à toutes.

Table des matières :

| | |
|---|----|
| Introduction générale : | 1 |
| I. Vision Assistée par Ordinateur (VAO) et Traitement d'images: | 2 |
| I.1 Introduction: | 2 |
| I.2 définitions de base : | 3 |
| I.3 Les étapes du traitement artificiel d'une image en VAO..... | 5 |
| I.3.1 Acquisition | 6 |
| I.3.2 Prétraitement | 6 |
| I.3.2.1 La compression | 6 |
| I.3.2.2 La restauration | 7 |
| I.3.2.3 L'amélioration | 7 |
| I.3.2.3.1 La modification d'histogramme | 7 |
| I.3.2.3.1.1 <i>Le recadrage dynamique</i> | 8 |
| I.3.2.3.1.2 <i>Egalisation de l'histogramme</i> | 10 |
| I.3.2.3.2 Le filtrage | 10 |
| I.3.2.3.2.1 <i>Le filtrage linéaire</i> | 11 |
| I.3.2.3.2.2 <i>Le filtrage non linéaire</i> | 13 |
| I.3.3 L'analyse | 14 |
| I.3.3.1 Segmentation par extraction de contour | 14 |
| I.3.3.2 Segmentation en régions homogènes | 16 |
| I.3.4 L'interprétation..... | 17 |
| I.4 Conclusion..... | 18 |
| II. La réalité augmentée : Illustration et usage..... | 19 |
| II.1 Introduction..... | 19 |
| II.2 Définition de la réalité augmentée | 19 |
| II.3 Outils de la RA..... | 20 |
| II.3.1 HMD optique | 20 |
| II.3.2 HMD vidéo..... | 21 |
| II.3.3 Comparaison entre les HMD vidéo et les HMD optique..... | 21 |
| II.4 Systèmes de RA existants..... | 22 |
| II.4.1 Calibration de la caméra..... | 23 |
| II.4.2 Les systèmes basés modèle..... | 24 |
| II.4.3 Les systèmes basés image..... | 25 |
| II.5 Domaines et applications de la réalité augmentée..... | 25 |
| II.5.1 Culture et éducation..... | 26 |

| | | |
|-----------|--|----|
| II.5.2 | Guidage visuel pour la maintenance | 26 |
| II.5.3 | Projet héritage culturel..... | 27 |
| II.5.4 | Apprentissage du geste médical | 28 |
| II.5.5 | Divertissement | 28 |
| II.6 | Conclusion..... | 29 |
| | | |
| III. | Détection et suivi d'objets dans une séquence d'images..... | 30 |
| III.1 | Introduction : | 30 |
| III.2 | <i>Le pattern (le modèle)</i> | 30 |
| III.3 | <i>Détections des points d'intérêt</i> | 31 |
| III.3.1 | Méthodes existants..... | 32 |
| III.3.1.1 | Détecteurs à échelle fixe..... | 33 |
| III.3.1.2 | Détecteurs multi-échelle..... | 33 |
| III.3.2 | Etude de quelques méthodes | 33 |
| III.3.2.1 | Détecteur SIFT : Invariant Feature Transform..... | 34 |
| III.3.2.2 | Détecteur PCA-SIFT : Principal Component Analysis-Invariant FeatureTransform | 37 |
| III.3.2.3 | Détecteur SURF : Speeded-Up Robustfeatures..... | 37 |
| III.3.3 | Comparaison des méthodes..... | 38 |
| III.3.3.1 | Temps d'exécution | 39 |
| III.3.3.2 | Changement d'échelle | 39 |
| III.3.3.3 | Rotation image | 40 |
| III.3.3.4 | Image taché | 40 |
| III.3.3.5 | Changement d'illumination | 42 |
| III.3.3.6 | Transformation affine..... | 42 |
| III.3.4 | Discussion des résultats | 43 |
| III.4 | Conclusion..... | 44 |
| | | |
| IV. | Conception et réalisation de l'Application..... | 45 |
| IV.1 | Introduction | 45 |
| IV.2. | Outils utilisés..... | 45 |
| IV.2.1 | IDE: MicrosoftVisual Studio..... | 45 |
| IV.2.2 | OpenCV | 46 |
| IV.2.3 | OpenGL | 46 |
| IV.2.4 | 3ds MAX | 47 |
| IV.3 | Architecture de l'application..... | 48 |
| IV.4 | Acquisition de l'image | 48 |
| IV.5 | Prétraitement | 49 |

| | |
|---|----|
| IV.5.1 Transformation de la séquence au niveau de gris | 49 |
| IV.5.2 Seuillage..... | 49 |
| IV.6 Détection des points d'intérêt, et extraction des paramètres utiles à la projection..... | 50 |
| IV.6.1 Extraction des les points d'intérêts | 50 |
| IV.6.2 Détermination des perspectives | 52 |
| IV.6.2.1 Calcule du point de translation..... | 52 |
| IV.6.2.2 Calcule de la distance..... | 53 |
| IV.6.2.3 Calcule des rotations..... | 53 |
| IV.6.2.3.1 Suivant le repère X | 54 |
| IV.6.2.3.2 Suivant le repère Y | 54 |
| IV.6.2.3.3 Suivant le repère Z | 55 |
| IV.7 Projection de l'objet | 56 |
| IV.8 Résultats expérimentaux et critiques | 57 |
| IV.9 Conclusion | 58 |
| Conclusion générale | 59 |

Table des figures :

| | |
|---|----|
| Figure I.1 La chaîne de VAO..... | 2 |
| Figure I.2 Une partie d'image en niveau de gris et sa matrice représentative..... | 4 |
| Figure I.3 Une image et son histogramme représentatif..... | 5 |
| Figure I.4 Exemple de recadrage dynamique..... | 9 |
| Figure I.5 LUT de binarisation..... | 9 |
| Figure I.6 Egalisation d'histogramme..... | 10 |
| Figure I.7 Filtrage linéaire d'une image : à gauche l'image initiale et à droite l'image filtrée en utilisant un filtre moyenne. | 12 |
| Figure I.8 Filtre gaussien. | 13 |
| Figure I.9 Filtrage non linéaire d'une image. A gauche l'image initiale et à droite l'image filtrée en utilisant un filtre médian de TUCKEY. | 14 |
| Figure I.10 Opérateur de Roberts..... | 15 |
| Figure I.11 Opérateur de Sobel sur la précédente image. | 15 |
| Figure I.12 Laplacien sur la même image. | 16 |
| Figure II.1 Continuum réalité-virtualité. | 20 |
| Figure II.2 HMD optique..... | 20 |

| | | |
|---------------|--|----|
| Figure II.3 | HMD optique..... | 20 |
| Figure II.4 | HMD vidéo..... | 21 |
| Figure II.5 | HMDvidéo..... | 21 |
| Figure.II.6 | Ecran d'affichage ordinaire. | 21 |
| Figure II.7 | Correspondance entre le monde réel et le monde image. | 23 |
| Figure II.8 | Décomposition du modèle de Tsai. | 24 |
| Figure II.9 | Mire 3D de calibration. | 25 |
| Figure II.10 | Pattern 2D de calibration. | 25 |
| Figure II.11 | Géométrie épipolaire. | 25 |
| Figure II.12 | Assistance laboratoire. | 26 |
| Figure II.13 | Indication pour assistance moteur. | 27 |
| Figure II.14 | Archéologue : un temple virtuel reconstitué sous forme 3D, est rajouté sur l'image réelle de son état actuel..... | 27 |
| Figure II.15 | limitation de la zone d'excision dans le rein. | 28 |
| Fig. II.16 | Images issues du jeu Eye of Judgment. | 29 |
| | | |
| Figure III.1 | Pattern planaire..... | 31 |
| Figure III.2 | Différence de gaussien..... | 34 |
| Figure III.3 | Détection d'extrémums dans l'espace des échelles..... | 35 |
| Figure III.4 | Construction de l'histogramme ce l'orientation..... | 36 |
| Figure III.5 | Construction d'un descripteur SIFT..... | 37 |
| Figure III.6 | Pour le test des images. A et H ont les transformations affines, B et C ont un changement d'échelle, D la rotation, E et F ont des tâches sur les images et G le changement d'illumination..... | 38 |
| Figure III.7 | Changement d'échelle. | 40 |
| Figure III.8 | Comparaison des taches. Groupe E, les données représente la répétabilité des taches..... | 41 |
| Figure III.9 | Comparaison des rotations. Groupe D Les données représentent la répétabilité..... | 41 |
| Figure III.10 | Comparaison suivant la rotation. Groupe D, degré de rotation est de 45°, la premier comparaison montre celle de SIFT (10 juste/10), la deuxième PCA-SIFT (1/10) et la dernière celui de SURF (2/2). | 41 |
| Figure III.11 | Comparaison avec taches. Groupe E et F, de gauche à droit, les résultats sont : SIFT (10/10), SURF (3/3) et PCA-SIFT (1/10). | 41 |
| Figure III.12 | Comparaison des transformations affine, groupe A et H, quand le changement du point du vue et large, le resultat et presenter de la gauche vers la droite : SIFT (5/0), SURF (0), PCA-SIFT (1), PCA-SIFT(2/10). | 43 |
| | | |
| Figure IV.1 | Schéma illustratif de conception de l'application réalisée..... | 48 |
| figure IV.2 | Template utilisé pour le suivie | 50 |
| Figure IV.3 | Suivie de la template au long de la séquence..... | 52 |
| figure IV.4 | Calcule du point de translataion | 53 |
| figure IV.5 | Calcule de la distance | 53 |
| figure IV.6 | Calcule des rotations..... | 54 |
| figure IV.7 | Calcule des rotations suivant l'axe X. | 54 |
| figure IV.8 | Calcule des rotations suivant l'axe Y. | 55 |
| figure IV.9 | Calcule des rotations suivant l'axe Z. | 55 |
| Figure IV. 10 | Projection de l'objet 3D sur la scène réelle. | 57 |

Introduction générale :

Avec le progrès réalisé dans les technologies matérielles et logicielles, les solutions d'interaction, de compréhension, d'apprentissage... proposées à l'homme se multiplient et se complexifient. La genèse des techniques de visualisation a accentué le phénomène du développement d'applications d'interaction homme-machine (IHM). Jusqu'à présent cette modalité, a été le domaine où le développement a été le plus important. Ainsi, le travail que nous allons réaliser illustrera un domaine en quête de ces technologies, la réalité augmentée, et qui ne cesse d'acquiescer l'attention des sociétés informatiques, et cela, grâce à la contribution primordiale de cette dernière dans divers domaines.

nous proposons d'étudier dans le cadre de ce mémoire la visualisation machine de façon générale, avant d'entrer dans le domaine de la réalité augmentée, ses utilités, et quelques exemples illustratifs, pour ensuite se lancer dans les différents environnements de développement, et les bibliothèques les plus utilisées, pour enfin se trancher dans l'IDE que nous avons utilisé dans la réalisation de l'application, et représenter l'exécutif de cette dernière qui permettra la représentation d'objets virtuels dans une scène réel.

I.1 Introduction

L'homme antique a toujours rêvé de laisser empreinte sans grand succès, se contentant de graver son imagination sur des plaques rocheuses, mais cela s'est rendu possible, après des siècles d'évolution. Depuis le milieu du 19^{ème} siècle, la vision assistée par ordinateur (VAO) n'a cessé d'évoluer pour devenir une science à part dont les services se sont avérés très utiles à un grand nombre de domaines (Médecine, industrie, IA,...).

La VAO est donc un ensemble d'outils qui permet à l'ordinateur d'imiter la perception humaine afin d'extraire des informations d'une image brute pour pouvoir prendre des décisions.

Nous pouvons alors dire que la VAO constitue une chaîne de traitements allant de l'acquisition de l'image brute jusqu'à son interprétation par la machine.

Le schéma suivant (figure 1) montre les étapes essentielles dans un système de VAO.

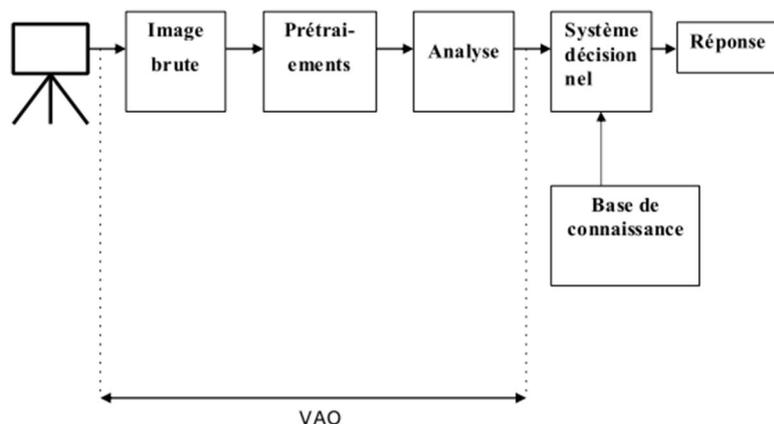


Figure I.1 La chaîne de VAO

Les domaines d'application de la VAO sont vastes, parmi eux on trouve l'imagerie aérienne et spatiale dans laquelle les traitements concernent l'amélioration des images satellites, l'analyse des ressources terrestres, la cartographie, Les technologies biomédicales dont l'exemple le plus connu utilisant le traitement d'image est le scanner mais on trouve des utilisations de cette technique dans l'échographie, la robotique qui connaît actuellement le plus grand développement et dont les tâches usant de l'imagerie

sont principalement l'assemblage (pièces mécaniques, composants électroniques, ...), le contrôle de qualité ainsi que la robotique mobile.

I.2 définitions de base :

- Image : ce terme peut être vu selon plusieurs contextes, théoriquement, c'est une transformation soudaine d'un fait concret en une donnée arrangeable, ou aussi une forme discrète d'un phénomène continu obtenue après discrétisation. Le plus souvent cette forme est bidimensionnelle. L'information dont elle est le support est caractéristique par des propriétés dont l'intensité lumineuse, et une intensité d'informations qu'elle contient.

Techniquement c'est une modélisation mathématique des données acquise d'une projection quelconque de rayons lumineux sur un plan.

$$I : [0, L - 1] * [0, C - 1] \rightarrow [0, M] \quad (1.1)$$

I est une image de L lignes et C colonnes dont l'information portée est définie sur un espace à P dimension.

Si I est une image binaire, alors $(P, M) = (1, 1)$.

Si I est une image en niveaux de gris, alors $P=1$ et le plus souvent $M=255$.

Si I est une image couleur, alors $P=3$ et le plus souvent $M=255$.

Plus simplement, une image est une grille d'éléments représentée internement par une matrice (voir figure I.2).

- Pixel : Un pixel est le nom associé à une unité de base de l'image qui correspond au plus petit élément de la représentation visualisée, caractérisé par sa position et sa valeur (sa couleur), il est utilisé pour représenter différemment ces derniers.
- Maillage : Le maillage est l'arrangement géométrique des pixels dans l'image, on trouve généralement le maillage carré comme la plupart des moniteurs, parfois le maillage hexagonal (maillage de référence pour le domaine de morphologie mathématique), aussi Le maillage triangulaire est plus confidentiel.

| | | | | |
|-----|-----|-----|-----|-----|
| 5 | 9 | 10 | 7 | 11 |
| 7 | 7 | 8 | 10 | 12 |
| 8 | 6 | 11 | 18 | 21 |
| 10 | 13 | 19 | 20 | 20 |
| 13 | 16 | 22 | 26 | 35 |
| 20 | 30 | 46 | 73 | 106 |
| 50 | 78 | 108 | 147 | 180 |
| 119 | 150 | 177 | 203 | 209 |
| 180 | 197 | 208 | 219 | 214 |
| 209 | 212 | 211 | 214 | 209 |
| 213 | 208 | 205 | 210 | 213 |

Figure I.2 Une partie d'image en niveau de gris et sa matrice représentative.

- Distance : Tout pixel est caractérisé par un couple de coordonnées (x,y). On peut donc calculer les distances entre pixels. Les distances les plus courantes sont (pour deux pixels P(Xp,Yp) et Q(Xq,Yq) :

Distance de Manathan :

$$d_1(P, Q) = |Xp - Xq| + |Yp - Yq| \quad (1.2)$$

Distance Euclidienne :

$$d_2(P, Q) = [(Xp - Xq)^2 + (Yp - Yq)^2]^{1/2} \quad (1.3)$$

- Ordre de connexité : Il existe principalement deux ordres de connexité : 4 et 8. Ce nombre correspond à la taille du plus petit voisinage non vide d'un pixel.

Un pixel a 4 voisins directs si l'on choisit la distance d1, et 8 voisins connectés.

- Histogramme : On appelle histogramme de l'image I, la fonction H définie sur l'ensemble des entiers naturels par :

$$H(x) = \{P.I(P) = x\} \quad (1.4)$$

C'est à dire que H(x) traduit le nombre d'apparitions des niveaux de gris x dans l'image I. Cette définition se généralise aux images multi bandes. L'histogramme est alors une fonction de P variables où P désigne le nombre de canaux.

L'histogramme est un outil privilégié en analyse d'images car il représente un résumé simple mais souvent suffisant du contenu de l'image (voir figure I.3).

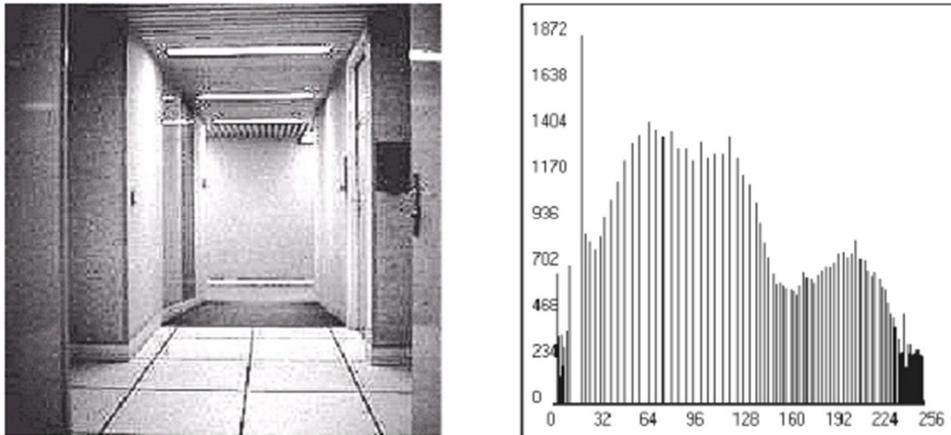


Figure I.3 Une image et son histogramme représentatif.

- **Bruit** : Un grand nombre de difficultés sont à l'origine lorsqu'on désire analyser un quelconque signal, c'est dû au bruit. L'image ne fait pas exception à cette règle.

Il existe de nombreuses sources de dégradations dont on peut citer le Bruit lié au contexte de l'acquisition, Bruit lié à la nature de la scène, ...

I.3 Les étapes du traitement artificiel d'une image en VAO.

Comme on l'a vu précédemment dans le schéma de la chaîne de VAO le traitement d'une image passe par plusieurs étapes avant d'arriver au but final qui consiste à prendre des décisions selon le contexte de l'image.

Le traitement artificiel d'une image passe essentiellement par les étapes suivantes :

- a- Acquisition.
- b- Prétraitement.
- c- Analyse.
- d- Interprétation.

Les trois premières étapes sont appelées : Traitement de bas niveau et la dernière étape : Traitement de haut niveau.

Nous allons maintenant exposer chaque étape en citant les principes brièvement.

I.3.1 Acquisition :

L'acquisition a pour objet le passage de la scène physique à une forme numérique observée. L'acquisition du signal vidéo est effectuée par le digitaliseur (Digitizer). Cet élément est fondamental : c'est en effet lui qui réalise l'échantillonnage spatial et le codage de l'image.

L'échantillonnage correspond au découpage du signal en pixel. Nous pouvons constater que celui-ci est dépendant du type de la camera utilisée car il existe des caméras qui produisent un signal déjà échantillonné.

Le codage correspond à la quantification de l'intensité de chaque pixel en une valeur numérique couramment appelée niveau de gris.

L'étape de l'acquisition nécessite l'utilisation d'un capteur image qui permet d'avoir une image sous sa forme analogique avant sa numérisation.

I.3.2 Prétraitement :

Le prétraitement regroupe toutes les opérations de manipulation de l'image qui permettent d'en améliorer la qualité. Ces manipulations produisent une nouvelle image. Son objectif est d'aider à l'analyse du contenu de l'image. On y retrouve différentes techniques : La compression, La restauration, L'amélioration,

I.3.2.1 La compression :

Le but des algorithmes de compression est de réduire la taille des fichiers pour diminuer l'espace nécessaire à leur stockage sur le disque ou leur transfert par le réseau. Ils encodent de manières différentes les données de l'image afin de les rendre plus compactes.

Deux familles d'algorithmes de compression se distinguent :

- La compression sans perte : Historiquement, les premiers algorithmes de compression furent : Shanon-fano et huffman, mais l'algorithme de compression le plus utilisé dans le domaine de l'image est LZW.
- La compression avec perte : Certains types de données peuvent bien se comprimer si l'on accepte de perdre une partie de l'information. C'est notamment le cas de

sons ou d'images numérisées. Les taux de compression peuvent alors être énormes (par exemple un taux de 80% pour une image).

Les principales méthodes de compression avec perte sont : JPEG, JPG, les ondelettes et les fractales.

I.3.2.2 La restauration :

La restauration consiste en la tentative de restitution d'une image aussi proche que possible de l'image avant dégradation. Elle s'appuie principalement sur une détermination du modèle mathématique du processus qui a engendré la dégradation, puis sur l'application du modèle inverse sur l'image.

La détermination peut se faire de deux manières : à priori et à posteriori. Dans le 1er cas, un modèle est construit à partir d'images tests afin de déterminer la réponse impulsionnelle du système perturbateur. Dans le second cas, la restauration s'effectue à l'aide de mesures réalisées sur l'image à restaurer, le modèle étant à priori inconnu, selon un processus adaptatif.

Dans les deux cas la restauration fait appel à des développements mathématiques lourds et complexes encore inutilisés dans les systèmes de visions temps réel.

1.3.2.3 L'amélioration :

Ensemble de méthodes et techniques destinées à améliorer l'aspect visuel de l'image. L'œil humain est essentiellement sensible aux forts contrastes c'est pourquoi les techniques d'amélioration tentent d'augmenter ceux-ci dans le but d'accroître la séparabilité des régions composant une scène. Différentes techniques sont possibles : la modification d'histogramme, Le filtrage...

1.3.2.3.1 La modification d'histogramme :

L'histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris de l'image, ce qui est plus communément appelé la dynamique de l'image. Il permet notamment d'analyser entre quelles bornes est répartie la

majorité des niveaux de gris dans le cas d'une image trop claire ou trop foncée. Parmi les méthodes qui existent dans le cadre de cette technique :

- Recadrage dynamique.
- Mise en évidence de zones dans l'histogramme.
- Egalisation de l'histogramme.

Avant d'expliquer ces méthodes on va introduire un terme très utilisé dans les techniques de modification de l'histogramme il s'agit de la LUT (look up table) ou table de conversion.

Une LUT est une fonction qui transforme un niveau de gris i en un niveau de gris j et ce, sans modification de la structure spatiale de l'image, c'est à dire que la transformation du niveau de gris d'un pixel est indépendante des pixels voisins.

Plus mathématiquement, une LUT f est défini comme suit :

Soit G_I l'ensemble des niveaux de gris de départ : $G_I = \{0,1,\dots,N_I\}$

Soit G_F l'ensemble de niveaux de gris de l'image résultat : $G_F = \{0,1,\dots,N_F\}$

f est une application de G_I dans G_F telle que :

$$\forall g_I \in G_I \exists g_F \in G_F : g_F = f(g_I) \quad (1.5)$$

1.3.2.3.1.1 *Le recadrage dynamique :*

L'une des applications des LUT est le recadrage dynamique. Certaines images sont initialement trop claires, trop foncées, ou bien peu contrastées. Cela est dû respectivement au fait que les niveaux de gris de l'image sont tassés vers le haut de l'échelle, vers le bas ou bien sont regroupés dans un intervalle étroit. Ce défaut est nettement visible sur l'histogramme.

Le but de la LUT est alors de redistribuer les niveaux de gris de l'image afin de leur faire occuper toute la bande des nuances possibles (figure I.4).

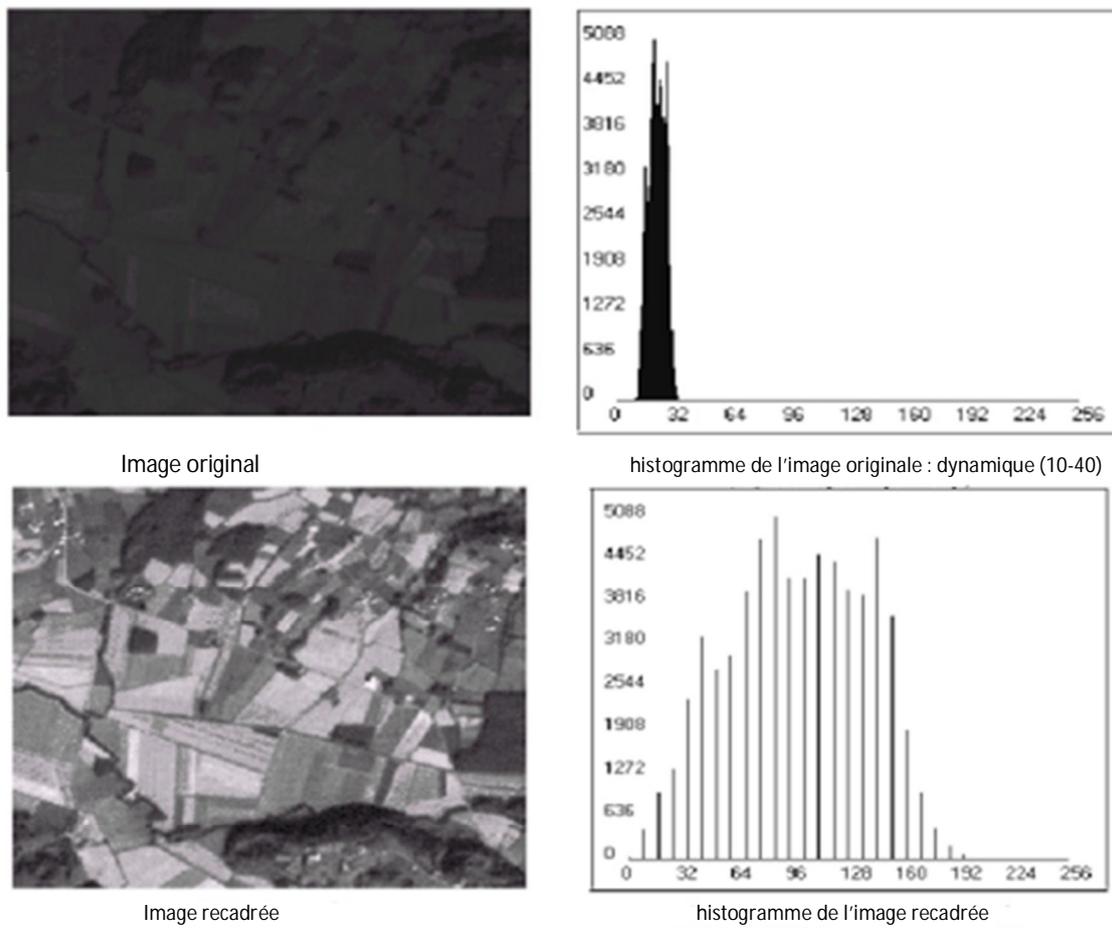


Figure I.4 Exemple de recadrage dynamique.

Mise en évidence de zones de l'histogramme

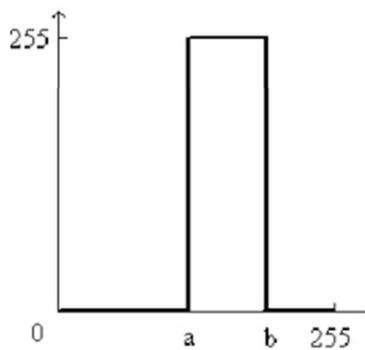


Figure I.5 LUT de binarisation.

Cette méthode utilise aussi la LUT. Elle consiste en la binarisation de l'image pour mettre en évidence certaines zones de l'histogramme. On procède à un seuillage des

niveaux de gris (figure I.5), et on transforme tous les niveaux compris entre le seuil 'a' et 'b' en 255 (soit le blanc), toutes les autres valeurs seront mises à zéro (le noir).

1.3.2.3.1.2 *Egalisation de l'histogramme :*

Elle consiste à redistribuer les valeurs de l'histogramme afin d'équilibrer la distribution des niveaux de gris c'est à dire s'approcher le plus possible de la situation où chaque niveau de gris est représenté par le même nombre de pixels.

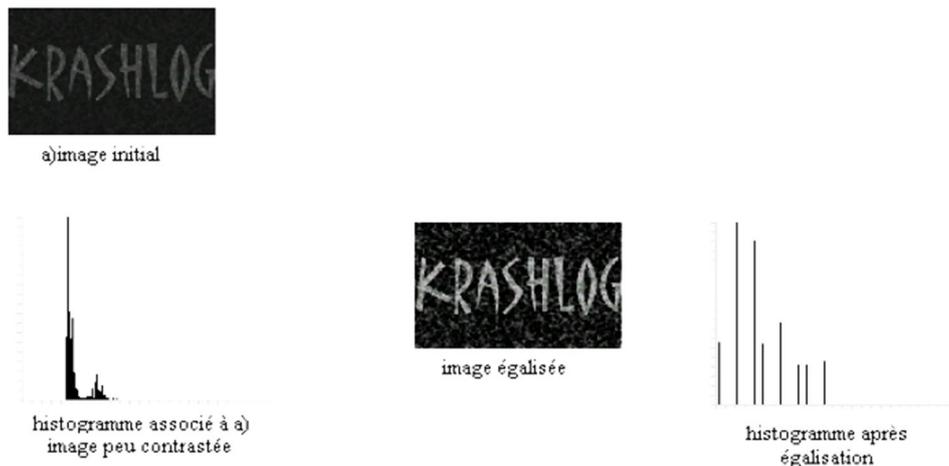


Figure I.6 Egalisation d'histogramme.

I.3.2.3.2 **Le filtrage :**

Nous avons vu jusqu'à présent des techniques qui visaient à améliorer le contraste de l'image. Elles portaient le nom de transformations ponctuelles car chaque pixel était transformé indépendamment de la valeur de ses voisins.

Or nous savons que l'image possède une certaine redondance spatiale, c'est à dire que les pixels voisins ont généralement les mêmes, ou pratiquement les mêmes caractéristiques. Nous savons aussi que le bruit est une brusque variation d'un pixel par rapport à ses voisins.

Tout cela nous amène à déduire que pour lutter contre les effets du bruit, il est nécessaire d'opérer des transformations qui, pour chaque pixel tiennent compte de son voisinage.

Parmi les méthodes les plus utilisées pour réduire l'effet du bruit, celle appelée filtrage de l'image ou filtrage passe bas.

On distingue deux types de filtrage passe-bas :

- Le filtrage linéaire où la transformation d'un pixel est le fruit d'une combinaison linéaire des pixels voisins.
- Le filtrage non linéaire où les pixels voisins interviennent suivant une loi non linéaire.

1.3.2.3.2.1 *Le filtrage linéaire :*

Ce type de filtre utilise la moyenne non pondérée des voisins et peut être sous la forme d'un masque tel que celui-ci :

$$H = 1/9 \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

On va alors déplacer le masque sur toute l'image, le pixel affecté par la transformation étant le pixel central du masque. Le facteur 1/9 est égal à la somme des coefficients du masque et sert à normaliser le filtre, de manière que celui-ci n'influe pas sur l'intensité globale de l'image.

On pose :

I_I : l'image de départ.

I_F : l'image résultat.

$$\text{On a : } I_F(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 H(i + 1, j + 1) \cdot I_I(x + i, y + i) \quad (1.6)$$

De manière plus générale pour un filtre de taille $(n+1) \times (n+1)$ avec n pair et dont la somme des coefficients vaut k :

$$I_F(x, y) = 1/k \sum_{i=-1}^1 \sum_{j=-1}^1 H(i + n/2, j + n/2) \cdot I_I(x + i, y + i) \quad (1.7)$$

Cette formule n'est autre que l'expression en chaque point de l'image d'un produit de convolution discrète, et l'on peut écrire pour l'image entière :

$$I_F = H \otimes I_I \quad (1.8)$$

Une propriété fondamentale du produit de convolution par rapport à sa transformée de Fourier est :

$$\mathcal{F}(H(x, y) \otimes I(x, y)) = \mathcal{F}(H(x, y)) \cdot \mathcal{F}(I(x, y)) \quad (1.9)$$

Donc on peut réaliser le filtrage de la manière suivante :

- On effectue séparément les transformées de Fourier de H et de I
- On multiplie les deux transformées.
- On applique au produit la transformée de Fourier inverse, on obtient alors

L'image filtrée.

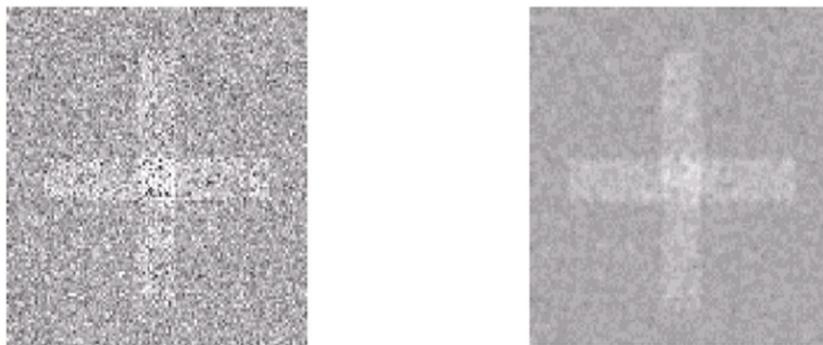


Figure I.7 Filtrage linéaire d'une image : à gauche l'image initiale et à droite l'image filtrée en utilisant un filtre moyenne.

Parmi les filtres linéaires les plus courants, on a le filtrage Gaussien qui se caractérise par sa facilité de mise en œuvre et par la bonne qualité de ses résultats.

Le filtrage Gaussien consiste en la convolution d'une image I avec une Gaussienne $G(x, y, \sigma)$

La fonction $G(x, y, \sigma)$ est séparable, c'est à dire qu'elle est réalisable soit par une convolution soit par un filtre $N \times N$, soit par une succession de deux convolutions par des filtres unidimensionnels de taille N.

Ou :

- σ facteur d'échelle.

$$G(x, y, \sigma) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.10)$$

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

Filtre 3*3

| | | | | | | |
|---|----|----|----|----|----|---|
| 1 | 3 | 7 | 9 | 7 | 3 | 1 |
| 3 | 12 | 26 | 33 | 26 | 12 | 3 |
| 7 | 26 | 55 | 70 | 55 | 26 | 7 |
| 9 | 33 | 70 | 90 | 70 | 33 | 9 |
| 7 | 26 | 55 | 70 | 55 | 26 | 7 |
| 3 | 12 | 26 | 33 | 26 | 12 | 3 |
| 1 | 3 | 7 | 9 | 7 | 3 | 1 |

Filtre 7*7 pour $\sigma=2$.



Image original



filtre gaussien taille 9

Figure I.8 Filtre gaussien.

I.3.2.3.2.2 *Le filtrage non linéaire:*

Ce type de filtrage s'oppose au précédent dans sa dénomination car il n'est pas le résultat d'une combinaison linéaire de pixels.

L'exemple le plus classique de filtre non linéaire est le filtre médian de Tuckey.

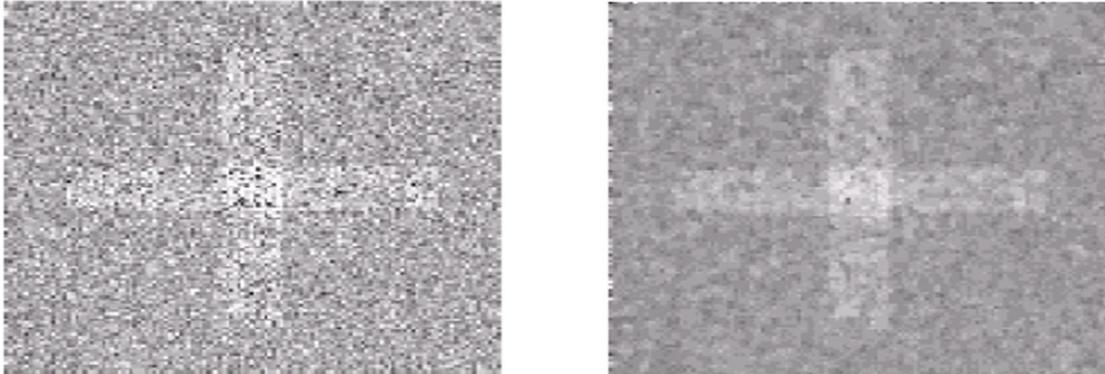


Figure I.9 Filtrage non linéaire d'une image. A gauche l'image initiale et à droite l'image filtrée en utilisant un filtre médian de TUCKEY.

I.3.3 L'analyse :

L'analyse est une suite d'opérations permettant l'extraction de l'information essentielle contenue dans une image, composée par la phase de segmentation qui fournit une partition de l'image.

A partir de cette segmentation, des techniques permettent d'obtenir la description de l'image. Celle-ci peut prendre la forme d'une image ou de toute structure de données permettant une description des entités contenues dans l'image elles-mêmes.

Par opposition avec la phase d'interprétation, l'analyse tente dans la mesure du possible, de ne pas prendre en compte le contexte.

La segmentation d'images ainsi définie est un domaine vaste où l'on trouve essentiellement deux approches :

I.3.3.1 Segmentation par extraction de contour :

Le but d'analyse d'image est d'obtenir une description synthétique des divers éléments qui la constituent à partir de la masse énorme d'information qu'elle contient à l'état brut. L'un des processus fondamentaux dans la chaîne de reconnaissance visuelle, consiste à diminuer cette quantité d'information en ne gardant que les points essentiels de l'image. Les points de contours constituent en ce lieu une approche rationnelle.

Le processus d'extraction de contours peut être décomposé en plusieurs étapes distinctes :

La mise en évidence des contours : elle s'obtient par une différenciation de l'image en utilisant des opérateurs tels que ceux-ci :

Opérateur de Roberts : il est basé sur la notion du gradient et utilise deux masques :

$$H_x = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$



Figure I.10 Opérateur de Roberts

Opérateur de Sobel : utilise comme masque :

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



Figure I.11 Opérateur de Sobel sur la précédente image.

Laplacien : il consiste à calculer la dérivée seconde de l'image.

Il utilise le masque suivant :

$$H = \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$


Figure I.12 Laplacien sur la même image.

Cet opérateur est particulièrement sensible au bruit.

- La réduction des contours : elle agit de manière que les contours mis en évidence ne fassent qu'un seul pixel d'épaisseur.
- La binarisation des contours : Suivant les méthodes, ce processus et le précédent peuvent être exécutés suivant un ordre différent, c'est à dire binarisation-réduction ou réduction-binarisation.
- La description des contours : Elle permet d'organiser les points de contour en structures simples telles que les segments de droites, arcs,...etc, et de détecter les contours fermés afin d'isoler ceux-ci.

I.3.3.2 Segmentation en régions homogènes :

Contrairement à l'extraction de contours qui s'intéresse aux bords d'une région, la segmentation en régions homogènes vise à segmenter l'image en se basant sur les propriétés intrinsèques de la région.

Le choix des propriétés détermine ce qu'on appelle le critère de segmentation.

Les principaux critères utilisés sont le niveau de gris ou la couleur, la texture et le mouvement.

I.3.4 L'interprétation

L'interprétation a pour but le passage de la description structurelle à la description sémantique en regard à certains objectifs. Ces objectifs peuvent être très simples (mesure de certains paramètres sur des formes) ou beaucoup plus complexes (description du contenu de la scène en termes de concepts non mathématiques).

Dans le cadre de la vision industrielle pour un objectif de qualité, les objectifs sont le plus souvent (en ordre croissant de complexité) :

- Contrôle dimensionnel (mesures simples).
- Contrôle d'aspect (mesures de texture).
- Contrôle structurel (analyse des composantes d'un objet).
- Tri (reconnaissance).

1.4 Problème posé par la réalité augmentée

Les outils et les techniques que nous offre la VAO se sont avérés très utiles pour trouver des solutions à des problèmes qui semblaient difficiles à résoudre.

Mais cependant, il y a beaucoup de domaines qui malgré les avantages que nous offre la VAO, ne sont pas totalement maîtrisés.

Parmi ces domaines qui tentent d'utiliser au mieux les outils de la VAO pour dépasser les difficultés : la réalité augmentée.

La réalité augmentée offre un bon terrain d'expérimentation pour mettre en pratique les techniques de la VAO ainsi que d'autres moyens pour arriver enfin à maîtriser d'une manière satisfaisante cette nouvelle technologie.

Conclusion

Nous avons vu au cours de ce premier chapitre l'interprétation de visualisation par ordinateur et les méthodes d'acquisition des flux visuels, ainsi que quelques traitements de niveau bas que peut effectuer la machine dans le but de modification de ces flux.

Dans le prochain chapitre, nous allons faire un pas dans le domaine de la réalité augmentée, en présentant les notions basiques de cette dernière, et présenter quelques exemples illustratifs.

II Introduction

L'homme ne se contente pas du minimum, c'est la loi du développement, ainsi, il s'est pas contenté de préserver la réalité, en se lançant plus loin, et en introduisant des techniques de façon à en concevoir des scènes mélangées entre ce qu'il voit et ce que la machine en aperçoit, la réalité augmenter...

II.1 Définition de la réalité augmentée

Si l'on revient à une caractérisation étymologique, le dictionnaire de l'Académie Française offre les définitions suivantes :

- réalité : Caractère de ce qui existe en fait (et qui n'est pas seulement une invention, une illusion ou une apparence) que l'on résumera comme l'environnement concret et matériel de l'homme.
- augmentée : Rendre plus grand, développer ; étendre, par addition d'une chose de même nature.

Une simple concaténation des deux définitions pourrait être une interprétation directe de la réalité augmentée.

Les systèmes de Réalité Augmentée ont pour but, dans un environnement réel, de combiner des objets réels et virtuels (c'est-à-dire générés par ordinateur) en une même scène, visant à intégrer, par superposition, les objets virtuels dans l'environnement réel de l'utilisateur. Les objets de synthèse enrichissent la perception que les utilisateurs ont de leur environnement en fournissant ou en facilitant l'accès à des informations inaccessibles directement. Ils permettent par exemple de faire apparaître les objets réels cachés, de présenter des annotations graphiques ou textuelles ou encore de superposer des informations issues de capteurs (de température par exemple). On peut en ce sens de définir la finalité des applications de Réalité Augmentée comme l'accompagnement de l'utilisateur dans une tâche réelle afin de la rendre plus facile à exécuter.

les applications de RA doivent satisfaire les objectifs suivants :

- incruster des objets virtuels dans un environnement réel (incruster des objets réels dans un environnement virtuel dans le cas de la virtualité augmentée).
- assurer l'interactivité et le respect des contraintes temps-réel.

- alignement entre les repères réels et virtuels.

Le schéma suivant illustrera la réalité mixte incluant la RA :

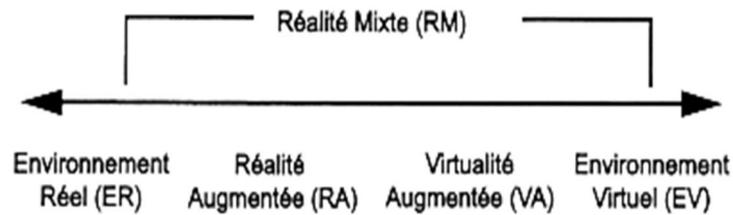


Figure II.1 Continuum réalité-virtualité.

II.2 Outils de la RA

Actuellement, deux moyens sont utilisés pour la réalisation des augmentations en temps réel. Des HMD (Head Mounted Display) avec des verres transparents (Fig.II.2) ou des HMD totalement opaques (Fig.II.4). Les applications optant pour ces derniers peuvent les remplacer par des écrans d'affichage ordinaires (Fig.II.6). Ces moyens contraignent à l'utilisation de techniques spécifiques, en l'occurrence, les systèmes optiques et les systèmes vidéo respectivement.

II.2.1 HMD optique

Pour les systèmes optiques (Fig.II.3), l'occlusion se fait par des verres transparents, partiellement transmissifs et partiellement réfléchissants. De cette façon, ils permettent à la lumière du monde réel de les traverser tout en visualisant des images projetées par un écran disposé au-dessus. Ces images représentent des objets virtuels cumulés à la vue du monde réel.



Figure II.2 HMD optique

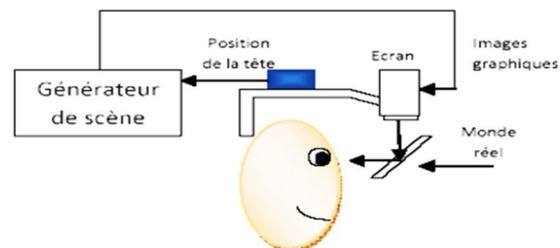


Figure II.3 HMD optique

II.2.2 HMD vidéo

Pour les systèmes vidéo (Fig.II.5), le casque est totalement opaque et ne permet pas de voir directement le monde réel. Il est constitué d'un écran, sur lequel sont projetés la scène réelle filmée par deux caméras et les images virtuelles. Le principe consiste à combiner les vidéos obtenues des caméras avec celles créées par un générateur de scènes. L'ensemble fournit alors des vidéos augmentées.



Figure II.4 HMD vidéo

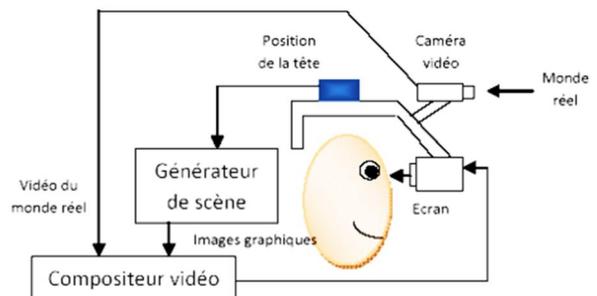


Figure II.5 HMD vidéo

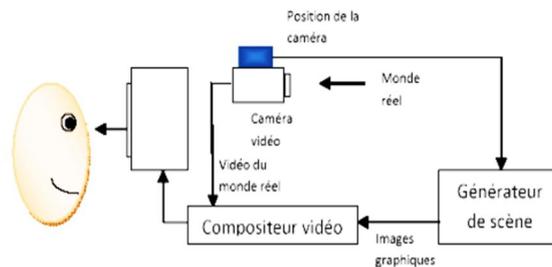


Figure.II.6 écran d'affichage ordinaire.

II.2.3 Comparaison entre les HMD vidéo et les HMD optique

Ici nous allons présenter les avantages et les inconvénients des techniques vidéo et optique l'une par rapport à l'autre. L'approche optique présente les avantages suivants par rapport à l'approche vidéo :

- **Simplicité** : l'approche optique est plus simple et moins coûteuse que l'approche vidéo. En effet l'approche optique n'a qu'un seul flux de vidéo à traiter, à savoir : les images graphiques. Tandis que l'approche vidéo en a deux : les images graphiques et l'image du monde réel.

- **Résolution** : l'approche vidéo limite la résolution de ce que voit l'utilisateur comme images graphiques ou monde réel, à celle du dispositif d'affichage tandis que l'approche

optique affiche les images graphiques à la résolution du monde réel, donc la vue de l'utilisateur ne sera pas dégradée.

- Sécurité : dans l'approche vidéo on est obligé d'utiliser des casques fermés, ceci peut présenter un danger dans le cas où le système tomberait en panne ce qui rendrait l'utilisateur totalement aveugle. Cet incident ne peut se produire dans le cas de l'approche optique, car elle utilise des combinateurs optiques qui permettent à l'utilisateur de voir lui-même le monde réel et ceci même s'il venait que le système tombe en panne.

- Pas de décalage des yeux : dans l'approche vidéo, la vue du monde réel est fournie par une caméra. l'usager doit toujours regarder dans le sens de l'orientation de la camera pour éviter le décalage qui pourra se produire. Toutefois, dans la plupart des configurations vidéo, la caméra ne se situe pas à une position appropriée par rapport au sens de regard de l'usager. Cet inconvénient ne figure pas au niveau de l'approche optique. L'approche vidéo présente les avantages suivants par rapport à l'approche optique :

- flexibilité dans les stratégies de l'augmentation : le problème de base avec l'approche optique est que les combinateurs optiques laissent la lumière passer. Tandis que dans la méthode vidéo l'image du monde réel et des objets virtuels sont dans leur forme numérique, ce qui laisse une grande flexibilité pour traiter et manipuler ces images. Un exemple type de cet aspect, est la difficulté, dans l'approche optique, de prendre en compte la profondeur des objets de la scène lors de l'augmentation.

- Méthodes de recalages supplémentaires : dans l'approche optique, la seule information que le système peut avoir, vient du head tracker situé sur la tête de l'utilisateur. Par contre, dans l'approche vidéo le fait d'avoir la forme numérique nous permet d'employer des stratégies supplémentaires sur le recalage.

- Facilité de faire la correspondance entre l'éclairage du monde réel et celui des objets virtuels dans l'approche vidéo.

II.3 Systèmes de RA existants

Après avoir présenté les objectifs qui doivent être satisfaits et les outils nécessaires pour la réaliser, il nous est possible de classer, dans cette section, les systèmes de RA existants par catégorie.

II.3.1 Calibration de la caméra

Afin de projeté des objets virtuelles suivant l'emplacement voulu dans le monde réel, une correspondance géométrique entre les coordonnées 3D du monde réel et le 2D sur l'image doit être faite.

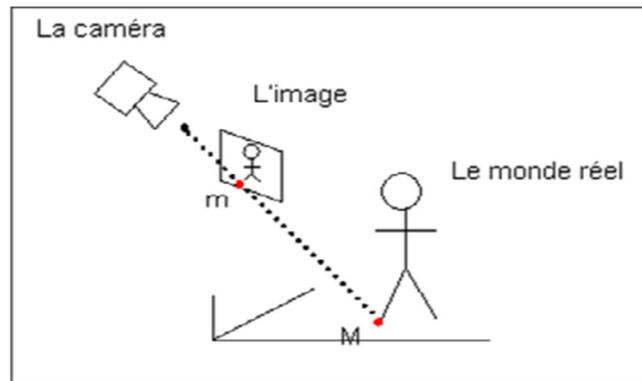


Figure II.7 : correspondance entre le monde réel et le monde image.

Le modèle de caméra de Tsai est basé sur le modèle de projection perspective. De tels modèles permettent de donner, à partir de la position d'un point du monde réel, sa position dans le monde image (voir figure II.8). Il existe un procédé inverse permettant de passer du monde image au monde réel.

Le modèle de Tsai est constitué de 6 paramètres extrinsèques ($R_x, R_y, R_z, T_x, T_y, T_z$) et 5 paramètres intrinsèques (f, C_x, C_y, S_x, k_1). Les paramètres extrinsèques décrivent l'orientation de la caméra dans le monde et doivent être recalculés à chaque fois que la caméra est déplacée. Les paramètres intrinsèques, une fois déterminés ne varient pas pour une caméra donnée, nous aurons donc à les déterminer une fois pour toute.

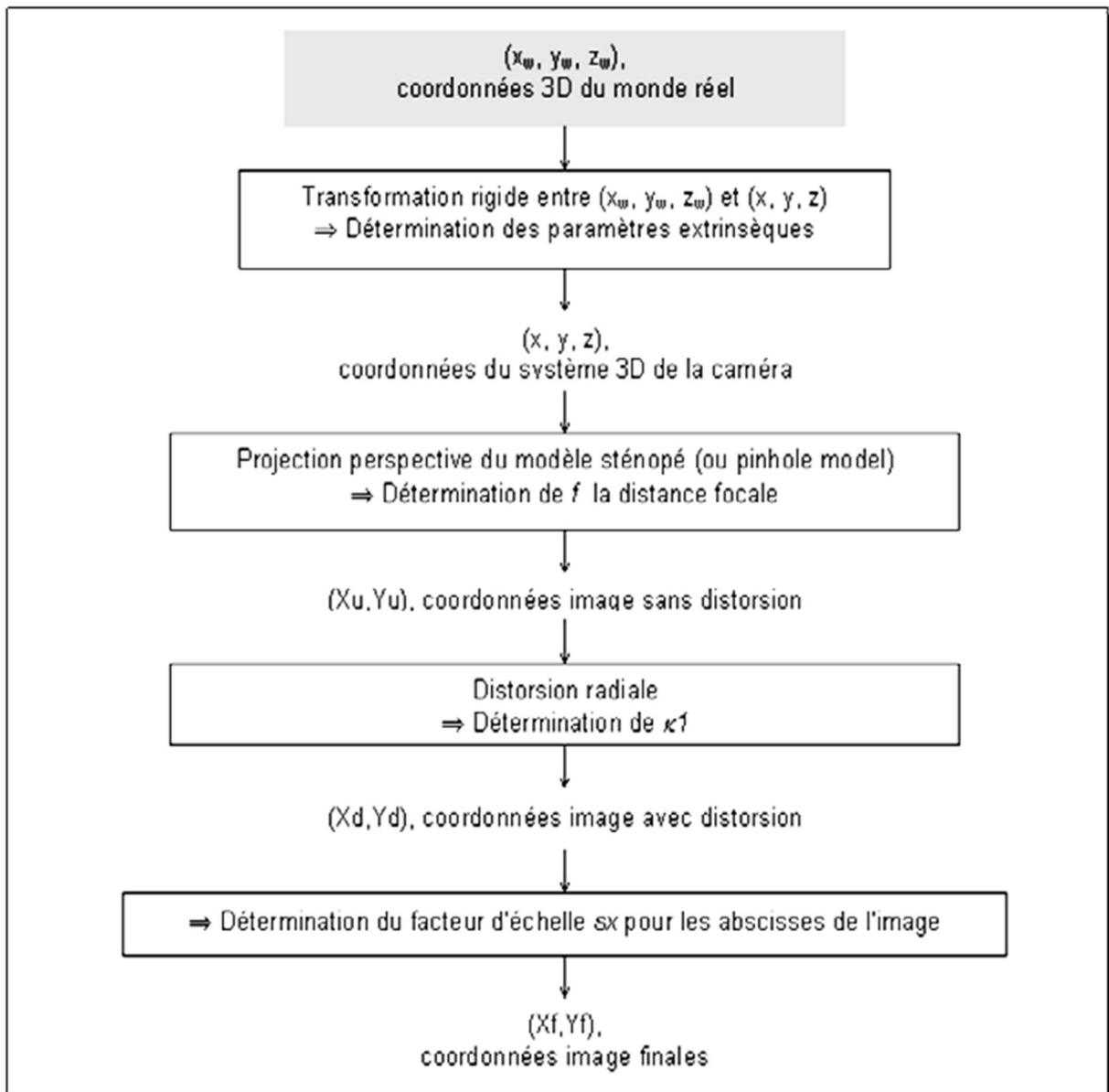


Figure II.8. Décomposition du modèle de Tsai.

II.3.2 Les systèmes basés modèle

Le principe de la calibration dans ces systèmes est d'introduire une mire 3D (Fig.II.9) ou un pattern 2D (Fig.II.10) dans la scène pour calculer les paramètres de la caméra. Pour le cas d'une mire 3D, il suffit de connaître sa géométrie puis de la suivre tout au long de la séquence. Après détection de cette mire, il est nécessaire de déterminer les six points non coplanaires pour résoudre le système d'équations relatif à la matrice de projection perspective. La résolution du système permet d'obtenir les paramètres de la caméra qui seront utilisés par la suite pour l'augmentation.

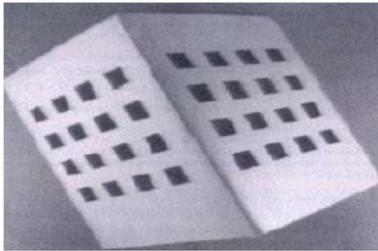


Figure II.9 Mire 3D de calibration.

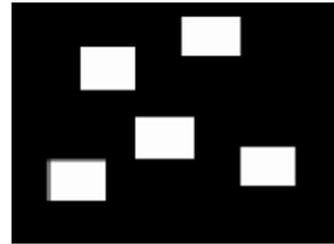


Figure II.10 Pattern 2D de calibration.

II.3.3 Les systèmes basés image

Ces systèmes se basent sur des algorithmes permettant de retrouver la structure de la scène en même temps que le mouvement de la caméra, à partir seulement du flot d'images de la séquence sans aucune connaissance à priori, ni sur la scène, ni la plupart du temps sur les paramètres intrinsèques de la caméra. Toutefois, ces algorithmes sont loin d'être exploitables en temps réel. Ils reposent sur le principe de la géométrie projective (Fig.II.11). Leur principe consiste à calculer une matrice, dite fondamentale, qui impose des contraintes géométriques entre deux vues. Son raisonnement utilise le principe de deux caméras ou deux vues d'une même caméra.

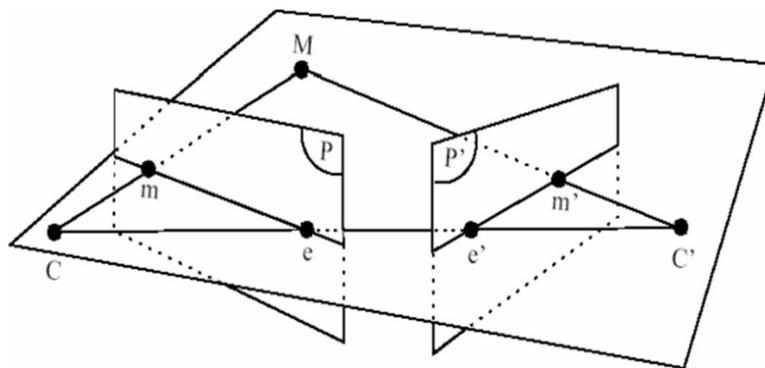


Figure II.11 Géométrie épipolaire.

II.4 Domaines et applications de la réalité augmentée

Depuis le début de la RA classique, l'évolution du niveau des interfaces technologiques et des solutions logicielles permet l'utilisation de ce concept dans de nombreux domaines. Ainsi, de nos jours, il est possible de trouver des solutions dans les domaines médical, militaire, artistique, architectural, éducatif, publicitaire, industriel, ludique, etc.

II.4.1 Culture et éducation

La réalité augmenté joue un rôle important dans la société et ses occupations, ainsi, aujourd'hui, l'avancé de ce concept a conduit à la modernisation du domaine culturel, et commence à dominer les établissements éducatifs, ainsi, les bénéficiaires auront simplicité, flexibilité, et enseignement de meilleur qualité.

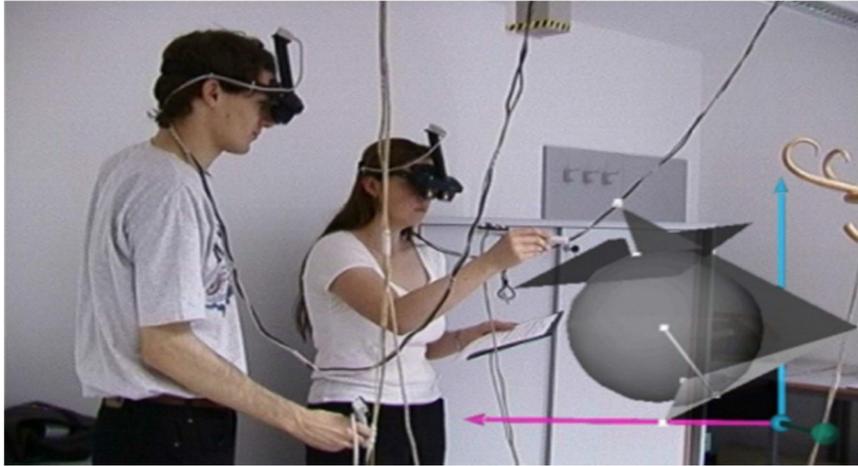


Figure II.12 assistance laboratoire.

II.4.2 Guidage visuel pour la maintenance

En milieu industriel, la réalité augmentée permet de simplifier considérablement certaines tâches d'assemblage, de maintenance ou de réparation. Ainsi, des chercheurs ont développé un système permettant de guider les techniciens dans l'assemblage de pièces, de réseaux électriques, et de dispositifs pour différents appareils. Le même principe est mis en œuvre dans l'assemblage d'un mécanisme de fermeture de porte pour automobile. Des indications virtuelles peuvent aussi être ajoutées pour désigner certaines pièces d'objets, comme des photocopieuses ou des moteurs, et faciliter ainsi les opérations de maintenance ou de réparation sur ces objets.

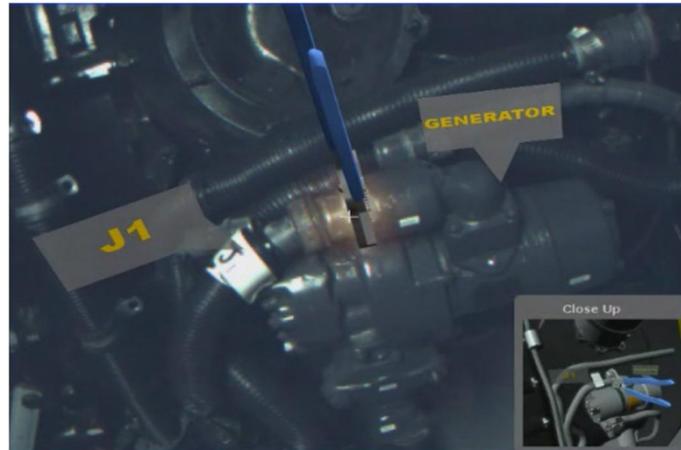


Figure II.13 indication pour assistance moteur.

II.4.3 Projet héritage culturel

La réalité augmenté n'a pas omis le domaine de l'archéologie, elle peut permettre à l'utilisateur de visualiser des sites archéologiques de l'antiquité, en superposant des reconstructions virtuelles 3D à des ruines existantes. Muni de lunettes de RA, d'un ordinateur portable et d'un assistant personnel (PDA), et accédant en temps réel à une base de données, l'utilisateur peut voir des monuments reconstruits ou encore des scènes de la vie d'autrefois. D'autres informations sont ajoutées lors de son déplacement sur le site pour compléter sa connaissance. Cette recherche porte sur l'étude d'une connexion sans-fil permettant l'apport temps réel d'informations en relation avec l'environnement, les mouvements de l'utilisateur sont enregistrés afin de pouvoir connaître les transformations géométriques entre le réel et le virtuel. Un système de reconnaissance vocale ou gestuelle est, de plus, mis en place pour permettre aux visiteurs d'obtenir les informations souhaitées.



Figure II.14: Archéologue : un temple virtuel reconstitué sous forme 3D, est rajouté sur l'image réelle de son état actuel.

II.4.4 Apprentissage du geste médical

Le domaine médical représente l'un des domaines où les enjeux sont importants. En effet, de nombreux instituts investissent dans la RA misant sur un bénéfice non négligeable, de rentabilité pécuniaire et humaine, dans la perspective que dans un avenir prochain, la collaboration à distance et/ou l'assistance par ordinateur applications pouvant prendre avantage de la RA soit monnaie courante. L'utilisation de systèmes de RA permet de faciliter la décision ou la collaboration lors de l'apprentissage ou lors d'une intervention chirurgicale, permettant aussi la simulation qui va permettre aux sages-femmes et aux médecins de pouvoir apprendre, s'entraîner et s'améliorer afin de limiter le nombre de césariennes et/ou de morts infantiles.

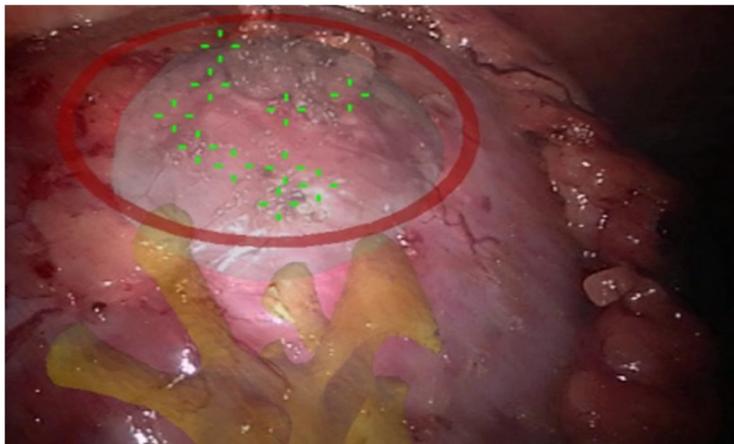


Figure II.15 limitation de la zone d'excision dans le rein.

II.4.5 Divertissement :

D'autres domaines sont conquis par la réalité augmentée, prenant les récents films hollywoodiens intégrant de plus en plus des images de synthèses qui s'y intègre, et le domaine de la production audiovisuelle, les techniques de suivi de caméra permettent d'étendre la technique classique du fond vert en réintégrant en temps réel un acteur dans un environnement virtuel qui remplace le fond. Contrairement aux bulletins météorologiques, le fond n'est pas statique et s'adapte à la position de la caméra. De nombreux événements sportifs diffusent également des images augmentées, permettant par exemple la visualisation des lignes de hors jeu, des distances à l'embut au football ou de la marque du meilleur saut ou du meilleur lancer en athlétisme. Les panneaux publicitaires peuvent également être modifiés virtuellement afin de les adapter au pays de diffusion. Des expérimentations ont été menées pour aider la

localisation des visiteurs dans un musée et proposer des annotations contextuelles attachées aux œuvres présentées. Les jeux 3D sont plus rentables à pratiquer grâce à la RA, comme ceux projeté sur la scène réelle, manipulable avec les mains.

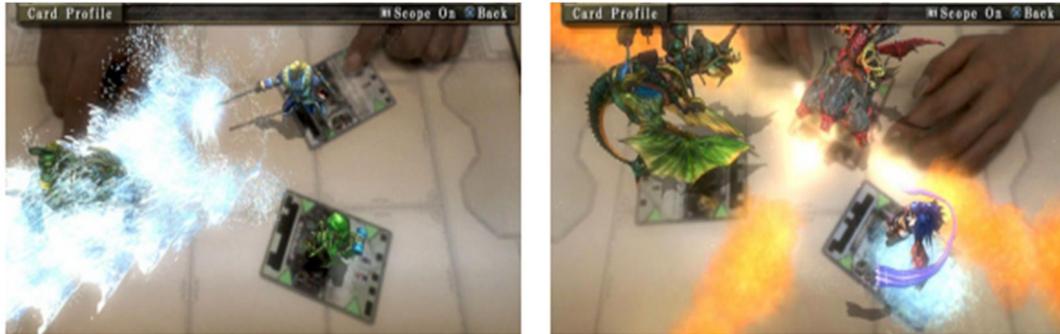


Figure. II.16 Images issues du jeu Eye of Judgment.

Ces différents exemples illustrent la diversité des domaines où l'utilisation de la RA est maintenant présente, permettant de développer des EV plus immersifs et rajoutant une dimension aux simulations virtuelles.

II.5 Conclusion

La réalité augmenté nous a pas laissé de temps pour s'en parer des usages des récentes technologies, comme nous l'avons vu au 2^{ème} chapitre, et cela nous laisse endetté de développer d'avantage ce précieux domaine.

Dans le chapitre suivant, nous nous lancerons droit au but, en allant entamer le domaine, commençant par la présentation de quelque procédures qui nous permettes de projeter des objets virtuels dans une scène réelle l'identification d'un repère d'insertion des objets virtuels, qui est le point de départ à ses vastes possessions.

III.1 Introduction :

Le suivi d'objets dans une séquence d'images vidéo est l'un des systèmes les plus utilisés dans le domaine de la RA, aussi le problème qui demande une extraction et un traitement d'informations provenant d'images complexes et incertaines dans beaucoup de cas. Ce problème devient de plus en plus difficile si la contrainte temps réel est exigée. Dans ce cas, en plus de la complexité et l'incertitude des informations à extraire, le temps de traitement doit être très court, pas plus de 40 ms.

Ce sujet a fait couler beaucoup d'encre et a suscité l'intérêt de beaucoup de chercheurs à travers le monde. Ainsi, un grand nombre de méthodes de suivi existent de nos jours. Ce nombre est dû, d'une part au nombre important de problèmes à traiter et d'autre part à la diversité des types d'applications concernées par le suivi. Ainsi, chacune des méthodes peut être appropriée à quelques domaines d'applications et non pas pour d'autres.

Dans ce chapitre nous allons commencer par la présentation de la notion de pattern et son rôle pour l'extraction des points d'intérêts. Ensuite, on va présenter quelques méthodes de détections des points d'intérêt ainsi que leur fonctionnement, et on termine par une comparaison selon différents critères entre ces méthodes.

III.2 *Le pattern (le modèle)*

En effet, pour être capable de suivre un objet, il est nécessaire de connaître la relation entre un modèle de l'objet (pattern) et son image courante à la quel on cherche sa correspondance dans la séquence . La mise en correspondance spatio-temporelle se fera selon la nature des primitives visuelles retenues.

Afin de reconnaître le pattern dans une image courante, on doit d'abord extraire les points d'intérêts caractérisant ces derniers, ensuite les comparés pour déterminé les extrémités de l'image rechercher.

Un pattern est un quadrillage dont nous connaissons les mesures et qui sera placé dans le monde réel. Les points caractéristiques du modèle seront identifiés dans l'image.

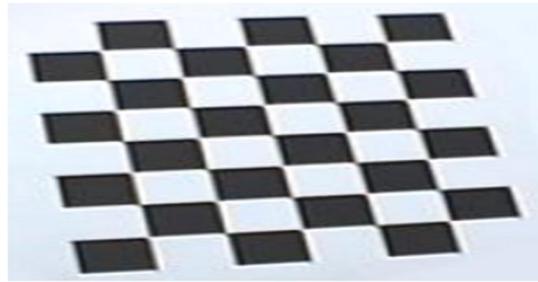


Figure III.1 Pattern planaire

Les modèles illustrés dans la figure III.1 se prêtent bien au problème que nous traitons et qui peuvent être utilisés par le système proposé.

Pour notre système et afin d'identifier aisément le modèle compte tenu de l'information brute énorme apportée par une image, nous avons opté pour un modèle planaire fortement contrasté en noir et blanc (Figure III.1).

Les raisons qui motivent le choix d'un tel modèle peuvent se résumer aux points suivants :

- Il permet de définir un repère convivial en faisant la supposition que le pattern se situe dans le plan $Z=0$.
- Il est fortement contrasté, en n'utilisant que le noir et le blanc, ce qui facilite sa détection dans la scène.
- Il est identifiable de manière unique, ce qui permet de lui associer une orientation bien définie.
- Il se prête bien pour une détection des primitives coins moyennant le détecteur.

III.3 *Détections des points d'intérêt*

L'utilisation d'un modèle fortement contrasté permet de le détecter facilement dans la scène. La détection du modèle dans la séquence vidéo se fait par extraction de points d'intérêt se trouvant sur ce dernier. Ces points d'intérêt sont en fait les coins se trouvant sur la surface du modèle.

L'extraction des points d'intérêt se fait en utilisant un détecteur de coins particulier, un des coins les plus utilisés et probablement le détecteur de coins Harris, néanmoins ce dernier n'est pas invariant au changement d'échelle. C'est-à-dire que cette méthode ne garantit pas que le même objet, représenté à deux échelles différentes, soit détecté de la même manière.

Cependant, plusieurs chercheurs se sont consacré pour développer des méthodes afin de parer à ces désavantages, comme Linderberg, Mikolajczyk et Schmid, Harris-Laplace et SURF.

Une fois ces derniers sont détectés, il faut extraire les caractéristiques de ces points d'intérêts, au final, ces derniers sont utilisés pour faire une mise en correspondance avec ceux de l'image référençant le modèle, afin de localiser le modèle dans la première image de la séquence vidéo et ensuite suivre ces points dans les prochaines prises de vue .

Dans ce qui suit, on va tenter d'expliqué les différentes méthodes existantes, ensuite les comparaisons entre quelques une de ces dernières, en se basant sur une étude qui est faite par des professionnels dans un laboratoire (Computer Graphics Lab) en Corée du sud et qui a été publié dans le journal international de traitement d'image [16].

II.3.1 Méthodes existants

Nous allons présenter brièvement les différents détecteurs utilisés, on distingue deux types de détecteurs :

- « à échelle fixe » – La réponse est calculée avec une taille de fenêtre et une force de lissage gaussien prédéfinis.
- « multi-échelle » – La réponse est calculée avec des tailles de fenêtre (ou des résolutions d'image) et des forces de lissage gaussiennes différentes.

Notons que ces derniers permettent de rendre le détecteur robuste aux changements d'échelle. Les couples stéréoscopiques ne présentent généralement pas de changement d'échelle entre l'image de gauche et l'image de droite. Toutefois, nous pensons que ces méthodes peuvent également permettre de détecter des points d'intérêt sur des structures de différentes tailles.

III.3.1.1 Détecteurs à échelle fixe

- Harris (HA) [1] – La réponse est calculée à partir de la matrice des moments du second ordre de chaque pixel. Cette matrice est calculée à partir des dérivées premières [11, 12, 13, 14].
- Moravec (MO) [2] – La réponse est donnée par une mesure de la variance des niveaux de gris dans huit directions.
- Kitchen-Rosenfeld (KR) [3] – La réponse est déterminée par la courbure du gradient des niveaux de gris.
- Beaudet (BE) [4] – La réponse utilise les dérivées secondes (matrice hessienne).
- SUSAN (SU) [5] – La réponse mesure la surface des niveaux de gris de même intensité au voisinage du pixel considéré.
- FAST (FA) [6] – La réponse étudie la configuration des niveaux de gris des pixels situés sur le périmètre d'un cercle centré sur le point considéré.

III.3.1.2 Détecteurs multi-échelle

- SIFT (SI) [7] – La réponse de ce détecteur repose sur le laplacien des niveaux de gris obtenu par une différence de gaussienne.
- Harris-Laplace, Hessian-Laplace (HL, HEL) [8] – La réponse est basée sur celle du détecteur de Harris ou Beaudet mais les points détectés doivent également être des maxima dans l'espace-échelle du laplacien.
- SURF [9] – La réponse est basée sur le déterminant de la matrice hessienne. Les points détectés doivent également être des maxima dans l'espace-échelle du déterminant de la matrice hessienne.
- Kadir (KA) [10] – La réponse repose sur l'entropie de l'histogramme des niveaux de gris au voisinage du point considéré.

III.3.2 Etude de quelques méthodes :

Etant donné que nous travaillons sur une application qui fait le suivi d'objet dans une scène 3D, donc nous allons utiliser des détecteurs multi-échelle. Nous allons étudier quelques méthodes les plus utilisées comme SIFT, PCA-SIFT et SURF.

III.3.2.1 Détecteur SIFT : Invariant Feature Transform

SIFT se constitue de quatre étapes majeures, détection extrême d'échelle-espace, localisation des points clés, assignement d'orientation et description des points clés.

La première étape utilise la fonction de différence-de-Gaussienne pour identifier les points d'intérêts potentiels, qui sont invariants à l'échelle et l'orientation. DoG a été utilisé au lieu de Gaussien pour améliorer la vitesse de calcul.

En appliquant les filtres gaussiens à différentes échelles, nous obtenons différentes images filtrées, avec différents gradients $L(x, y, \sigma) = G(x, y, \sigma) * (I(x, y))$. (Vu au premier chapitre)

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.1)$$

Où :

- σ est l'écart type.
- K est un coefficient.

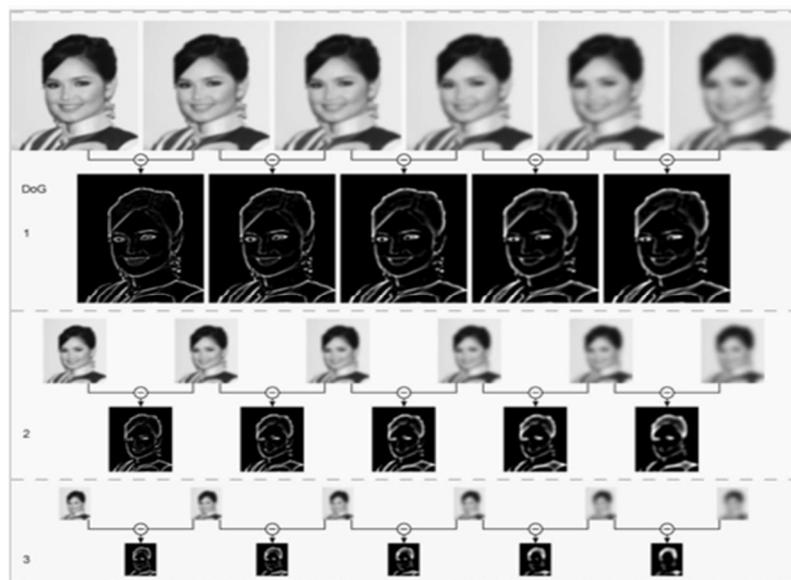


Figure III.2 : différence de gaussien

Une fois l'image ayant le plus grand DoG extraite, les points clé sont extraits suivant les extrémums comme suit :

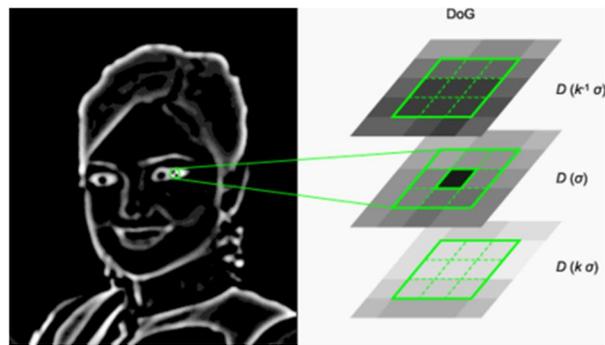


Figure III.3 détection d'extremums dans l'espace des échelles.

Des traitements supplémentaires seront exercés pour amélioration des points d'intérêts détectés, de façon à éliminer les points-clés de faible contraste, à partir du développement de Taylor de cette fonction, et constitue donc un extremum local. Un seuillage absolu sur cette valeur permet d'éliminer les points instables, à faible contraste

De même pour ceux situés sur les arêtes (contours), Les points situés sur les arêtes (ou contours) doivent être éliminés car la fonction DoG y prend des valeurs élevées, ce qui peut donner naissance à des extremums locaux instables, très sensibles au bruit : si l'image devait subir un changement numérique même imperceptible, de tels points-clés peuvent se retrouver déplacés ailleurs sur la même arête, ou même simplement disparaître.

- **Assignment d'orientation**

L'étape d'assignment d'orientation consiste à attribuer à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement à ces orientations, cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation : les mêmes descripteurs doivent pouvoir être obtenus à partir d'une même image, quelle qu'en soit l'orientation

À chaque position dans un voisinage du point-clé, on estime le gradient par différences finies symétriques, puis son amplitude (c.-à-d. sa norme) $m(x, y)$, et son orientation $\theta(x, y)$ [15] où :

$$m(x, y) = \sqrt{((L(x+1, y) - L(x-1, y)))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.2)$$

(15)

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (3.3)$$

$\forall(x, y)$ dans un voisinage de (x_0, y_0)

Un histogramme des orientations sur le voisinage est réalisé avec 36 intervalles, couvrant chacun 10 degrés d'angle. Les pics dans cet histogramme correspondent aux orientations dominantes :

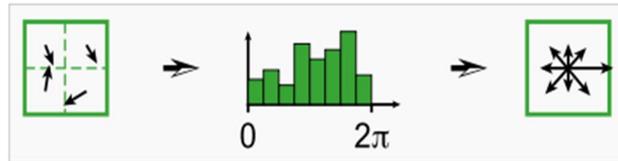


Figure III.4 Construction de l'histogramme des orientations.

À l'issue de cette étape, un point-clé est donc défini par quatre paramètres (x, y, σ, θ) .

- Descripteur de point-clé

Une fois les points-clés, associés à des facteurs d'échelles et à des orientations, détectés et leur invariance aux changements d'échelles et aux rotations assurée arrive l'étape de calcul des vecteurs descripteurs, traduisant numériquement chacun de ces points-clés. Autour de ce point, on commence par modifier le système de coordonnées local pour garantir l'invariance à la rotation, en utilisant une rotation d'angle égal à l'orientation du point-clé, mais de sens opposé.

Une région de 16×16 pixels, subdivisée en 4×4 zones de 4×4 pixels chacune. Sur chaque zone est calculé un histogramme des orientations comportant 8 intervalles. En chaque point de la zone, l'orientation et l'amplitude du gradient sont calculés comme précédemment. Ensuite, les 16 histogrammes à 8 intervalles chacun sont concaténés et normalisés. Dans le but de diminuer la sensibilité du descripteur aux changements de luminosité, les valeurs sont plafonnées à 0,2 et l'histogramme est de nouveau normalisé, pour finalement fournir le descripteur SIFT du point-clé, de dimension 128.

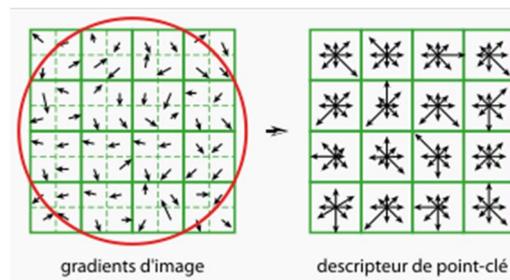


Figure III.5 Construction d'un descripteur SIFT.

III.3.2.2 Détecteur PCA-SIFT : Principal Component Analysis-Invariant Feature Transform

PCA est une technique standard pour réduction du dimensionnement, qui est bien coordonné pour représentation des points clés réparé et nous permet de projeter des échantillons linéairement hauts dimensionnels dans un espace de point bas dimensionnel.

En d'autres termes, le PCA-SIFT utilise PCA au lieu d'histogramme pour normaliser la correction du gradient. Le trait du vecteur est considérablement plus petit que celui de SIFT, et il peut être utilisé avec les mêmes algorithmes assortis, dans PCA-SIFT, comme SIFT, elles utilisent les distances Euclidiennes pour déterminer si les deux vecteurs correspondent au même point-clés dans les deux images différentes. Dans PCA-SIFT, le vecteur de l'entrée est créé par concaténation de gradient horizontale et verticale pour créer une carte 41x41 cases centrées sur le point-clé, auquel on aura $2 \times 39 \times 39 = 3042$ éléments. D'après PCA-SIFT, moins de composants exigent moins de stockage et procéderont à un résultat plus rapide, il choisi la dimension de l'espace du trait, $n = 20$, dans lequel on aura un résultat avantageux et considérable.

III.3.2.3 Détecteur SURF : Speeded-Up Robust features.

Les algorithmes de SURF et SIFT emploient des chemins légèrement différents pour les détections des caractéristiques, SIFT construit une image pyramide, en filtrant chaque couche avec Gaussien et la valeur sigma incrémenté en prennent la différence. De l'autre côté, le SURF crée une "PILE" échantillonnant les niveaux supérieure dans la pyramide et le résultat est mis dans une image de même résolution.

Dû à l'utilisation de l'image intégrale, SURF filtre la pile en utilisant une boîte de filtre d'approximation de dérivé partiel du deuxième ordre Gaussien, et cela par-ce-que les images

intégrales permettent le calcul des boîtes filtres rectangulaires dont le temps est presque constant.

Dans les points-clés qui ne sont pas égaux, le voisin le plus proche est défini comme le points-clés avec le minimum des distances Euclidiennes pour les vecteurs descripteurs invariants. Lowe à utiliser une mesure plus efficace qui obtient la comparaison des distances du voisin le plus proche et le deuxième voisin le plus proche.

III.3.3. Comparaison des méthodes

Pour pouvoir comparer ces trois méthodes, ces dernières doivent être comparées suivant les contraintes spatio-temporelles c'est-à-dire le temps d'exécution, changement de d'échelle, rotation, les tâches, l'illumination et les bruits. Afin de testé ces contraintes nous utiliserons dans la suite la figure III.2 suivante qui se devise en groupe.

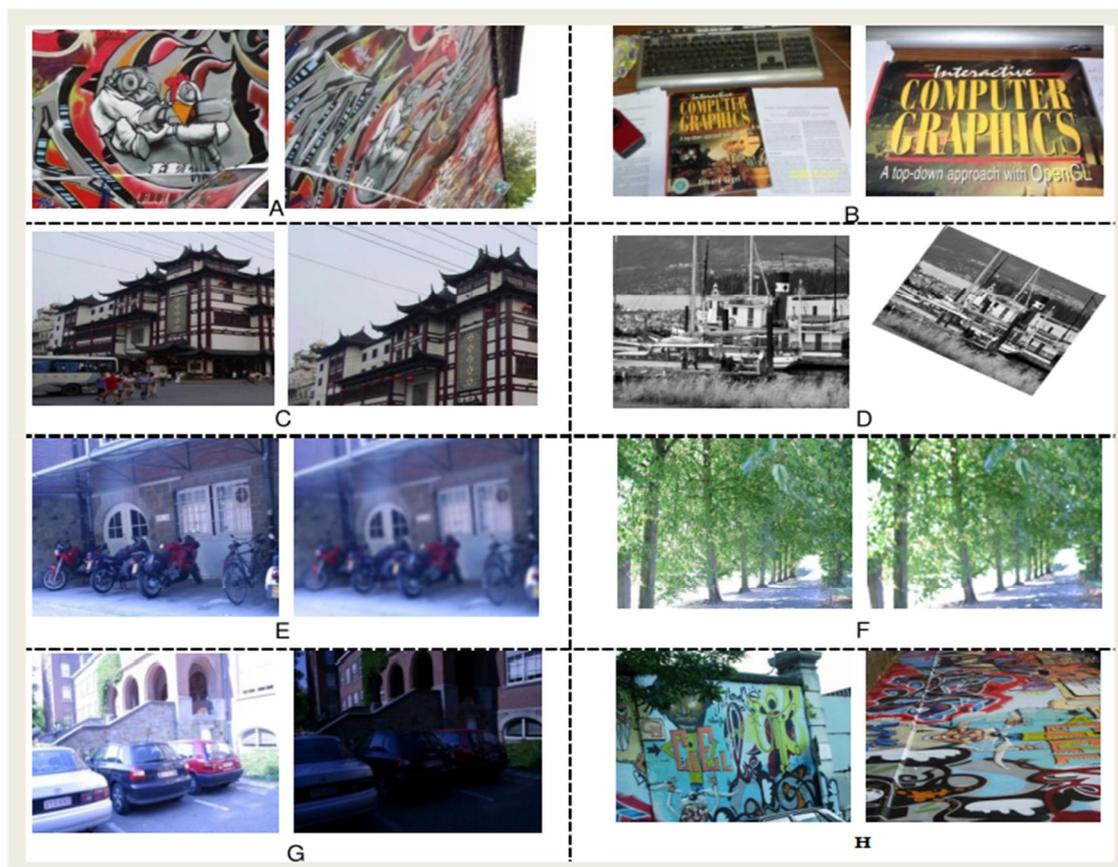


Figure III.6 : pour le test des images. A et H ont les transformations affines, B et C ont un changement d'échelle, D la rotation, E et F ont des tâches sur les images et G le changement d'illumination

III.3.3.1 Temps d'exécution :

Le temps d'évaluation est un résultat relatif, qui a tendance à montrer sur les trois méthodes chronométrées, le coût, il y a des facteurs qui influencent sur le résultat tel que la dimension et la qualité de l'image, l'image écrite à la machine (vue ou texture), et les paramètres de l'algorithme (par exemple la distance relationnelle).

La première partie de l'expérience utilise une image graffiti comme la montre la figure III.2 dans le groupe A et qui a la dimension 300*240.

En utilisant les mêmes paramètres sur les trois algorithmes le temps est compté pour le traitement complet afin de détecter les caractéristiques des points et la comparaison de ces derniers.

La table III.1 montre que le SURF est plus rapide, SIFT le plus lent mais il trouve la plupart des points comparés.

| Items | SIFT | PCA-SIFT | SURF |
|----------------------------|--------------|--------------|---------|
| l des égaux | 271 | 18 | 186 |
| Temps total (ms) | 2.15378e+007 | 2.13969e+007 | 3362.86 |
| 10 temps pour 10 égaux(ms) | 2.14806e+007 | 2.09696e+007 | 3304.97 |

Table III.1: temps d'exécution. Utilisation de groupe A de la figure 1.

III.3.3.2 Changement d'échelle :

La deuxième expérience montre les performances d'échelle invariante, ils utilisent deux groupes d'image B et C de la figure III.2, la figure III.3 quand à elle montre quelque résultat dans la comparaison des images, tandis que la troisième présentation montre quelques résultats de comparaison. L'expérience s'est contentée de montrer les 10 premiers points égaux.

Quand le changement de l'échelle est fait c'est-à-dire plus grand, SIFT et SURF sont de loin mieux que PCA-SIFT, par conséquent, les résultats montrent que les PCA-SIFT n'est pas stable comme SIFT et SURF pour l'échelle invariante et PCA-SIFT détecte moins d'égaux.



Figure III.7 : changement d'échelle.

| Data | SIFT | PCA-SIFT | SURF |
|------|------|----------|------|
| 1-2 | 41 | 1 | 10 |
| 3-4 | 35 | 0 | 36 |
| 5-6 | 495 | 19 | 298 |
| 7-8 | 303 | 85 | 418 |

Table III.2: Comparaison lors du changement d'échelle. Utilisation du groupe B et C de la figure III.2.

III.3.3.3 Rotation image :

La troisième expérience montre l'influence de rotation sur les trois méthodes, comme le Groupe D nous le fait montrer l'image est roté de 5 ou 10 degré, SIFT représente les caractéristiques égaux avec des lignes bleu et comme on le remarque il est stable, par contre SURF régresse, il trouve les égaux les plus petits et obtient les caractéristiques les plus petites, PCA-SIFT trouve seulement un égal sur les dix premiers égaux.

III.3.3.4 Image taché :

La quatrième expérience utilise les tâches Gaussiennes comme les images en groupe E et F de la figure III.2. Le rayon des tâches change de 0.5 à 9.0, comme vue dans la figure III.4, SURF et PCA-SIFT montrent de bonne performance quand le rayon est élargi, les résultats sont montrés sur le résultat de la figure6, SURF détecte peu de comparaison et PCA-SIFT détecte plus que ce dernier mais avec un nombre de correction plus petit, dans cette étape, SIFT montre ces meilleures performances.

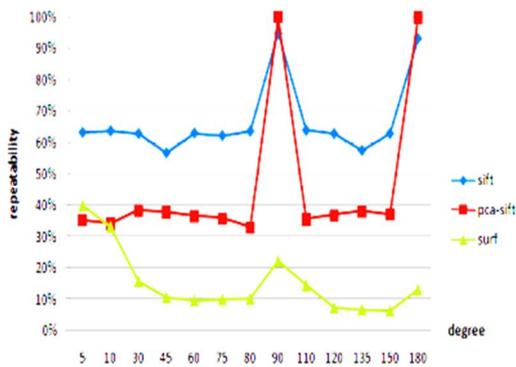


Figure III.9: comparaison des rotations. Groupe D, les données représente la répétabilité de rotation.

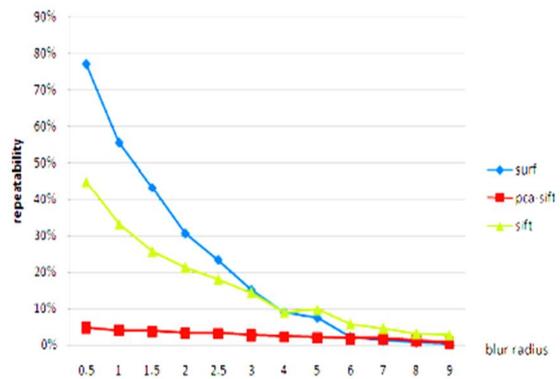


Figure III.8: comparaison des taches. Groupe E, les données représente la répétabilité Des taches.

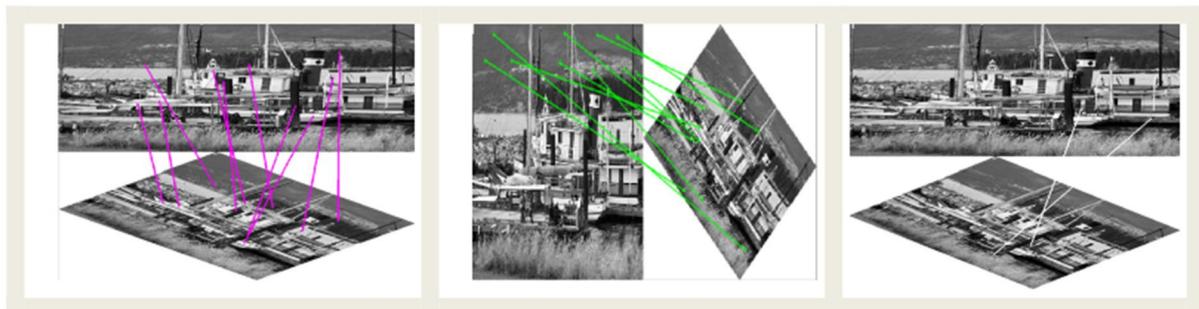


Figure III.10 : comparaison suivant la rotation. Groupe D, degré de rotation est de 45°, la premier comparaison montre celle de SIFT (10 juste/10), la deuxième PCA-SIFT (1/10) et la dernière celui de SURF (2/2).



Figure III.11 : comparaison avec taches. Groupe E et F, de gauche à droit, les résultats sont : SIFT (10/10), SURF (3/3) et PCA-SIFT (1/10).

III.3.3.5 Changement d'illumination :

Dans cette expérience l'équipe a montré l'effet d'illumination sur les algorithmes, comme le montre les images du groupe G de Figure II.2, depuis la donnée 1 à 6, la luminosité de l'image dégradé de plus en plus, SURF a la plus grande respectabilité, 31%, PCA-SIFT montre de très bonnes performances que SURF. Regardant le résultat dans l'expérience de la table III.3, qui montre les dix premiers égaux.

| Data | SIFT | PCA-SIFT | SURF |
|-------|------|----------|------|
| 1-2 | 43% | 39% | 70% |
| 1-3 | 32% | 34% | 49% |
| 1-4 | 18% | 27% | 25% |
| 1-5 | 8% | 18% | 6% |
| 1-6 | 2% | 9% | 5% |
| Moyen | 21% | 25% | 31% |

Table III.3: comparaison lors du changement d'illumination. Groupe G, les données représentent la répétabilité et la moyen de répétabilité.

III.3.3.6 Transformation affine:

Dans la sixième expérience ils ont évalué la stabilité des méthodes aux transformations affines. Ces dernières sont très importantes dans les images touchées (piqué), cette expérience utilise les images de groupe A et H de la première figure, le changement du point de vue est approximativement 50 degrés dans les images 1–6, la répétabilité de transformation affine sont affichés dans la table 4 et le résultat de comparaison et affiché dans la figure 8. Depuis les tables et la figure, SURF et SIFT ont une très bonne répétabilité lorsque le changement de point de vue est petit, mais quand devient grand dans les images 1-4, SURF détecte 0 points égaux et PCA-SIFT a les meilleures performances.

| Données | SIFT | PCA-SIFT | SURF |
|---------|------|----------|------|
| 1-2 | 47% | 15% | 54% |
| 1-3 | 37% | 12% | 33% |
| 1-4 | 22% | 12% | 12% |
| 1-5 | 7% | 10% | 2% |
| 1-6 | 0% | 9% | 0% |

Table 4 : comparaison des transformations affine. Groupe A et H, les données représentent la répétabilité.



FIGURE III.12: comparaison des transformations affine, groupe A et H, quand le changement du point de vue et large, le résultat est présenté de la gauche vers la droite : SIFT (5/10), SURF (0), PCA-SIFT (1), PCA-SIFT(2/10).

III.3.4 Discussion des résultats :

La table 5 montre les résultats des expériences, elle montre aussi qu'il n'y a aucune méthode qui satisfait toutes les déformations, donc quand on choisit une méthode d'extraction des points, on doit choisir celle qui satisfait notre objectif, le résultat de ces expériences n'est pas constant pour tous les cas, le changement d'algorithme peut donner d'autres résultats.

Utiliser une petite dimension nous donne des meilleurs résultats pour l'arrangement des points extraits, et l'affichage du résultat. SURF montre sa stabilité et sa rapidité dans ces expériences. On a su depuis que Fast-Hessien détecteur, qui est utilisé dans SURF est 3 fois plus rapide que celui utilisé dans DOG (celui qui est utilisé dans SIFT) et 5 fois plus que celui utilisé dans Laplace-Hessien.

Par la table qui suit, on constate que PCA-SIFT a besoin d'améliorer les performances du changement d'échelle, par contre SIFT montre sa stabilité dans toutes les expériences sauf celle du temps, par-ce-qu'il détecte tous les points et toutes les comparaisons.

Table: conclusion de toutes les expériences.

| Méthode | Temps | Changement d'échelle | Rotation | Taches | illumination | affinement |
|----------|----------|----------------------|----------|----------|--------------|------------|
| SIFT | Moyen | Meilleur | Meilleur | Meilleur | Moyen | Bon |
| PCA-SIFT | Bon | Moyen | Bon | Moyen | Bon | Bon |
| SURF | meilleur | Bon | moyen | Bon | Meilleur | Bon |

III.4 Conclusion:

Dans ce chapitre, nous avons étudié les trois méthodes pour la détection des points d'intérêts, l'extraction et leurs comparaisons dans des images déformées, la méthode SIFT n'est pas vraiment rapide par rapport aux autres, mais cette dernière est invariante aux changements d'échelle, rotations et les transformations affines.

SURF a de bonnes performances et plus rapide que SIFT, mais peu stable par rapport aux rotations et aux changements d'illumination. Choisir une méthode dépend fortement du type d'application.

Notre travail est la projection d'objet sur un pattern en temps réel, donc nécessitant un algorithme qui est à la hauteur des contraintes spatio-temporelles ce qui nous conduit à déduire que la meilleure méthode qui sera implémentée dans notre travail est SURF.

IV.1 Introduction :

Après les bases requises dans les phases théoriques précédentes, nous allons nous consacrer à la réalisation de l'application, et à maitre en œuvre un fruit de ce domaine, qui aura comme but, comme l'intitulé du mémoire, projection d'objets virtuels dans une scène réelle.

Au cours de ce chapitre, nous présenterons les outils et l'environnement de développement de l'application, sa conception et quelques fragments de codes.

IV.2 Outils utilisés:

Au cours de la réalisation l'application, nous avons utilisé les outils suivant :

IV.2.1 IDE: **Microsoft Visual Studio:**

 **Visual Studio** VS est une suite de logiciels de développement pour Windows conçue par Microsoft. Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des Services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer.

Dans l'application, nous avons opté pour le Langage C++, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation.

C + + est l'un des langages de programmation les plus populaires avec une grande variété de plateformes matérielles et systèmes d'exploitation.

L'IDE utilisé pour notre réalisation est Visual Studio Pro 2012, qui est une version complète, comportant tous les outils nécessaire pour le domaine étudié.

IV.2.2 OpenCV :



Open Computer Vision est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2008.

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

Pour l'application, nous avons opté pour la version OpenCV 2.1, qui est une version complète, facile à maitre en usage.

IV.2.3 OpenGL :



OpenGL (Open Graphics Library) est un ensemble normalisé de fonctions de calcul d'images 2D ou 3D lancé par Silicon Graphics en 1992. Cette interface de programmation est disponible sur de nombreuses plateformes où elle est utilisée pour des applications qui vont du jeu vidéo jusqu'à la CAO en passant par la modélisation.

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

L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plates-formes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D et certaines applications 2D vectorielles. Cette bibliothèque est également utilisée dans l'industrie du jeu vidéo où elle est souvent en rivalité avec la bibliothèque de Microsoft : Direct3D. Une version nommée OpenGL ES a été conçue spécifiquement pour les applications embarquées (téléphones portables, agenda de poche, consoles de jeux...).

Nous avons utilisé la version Glut, qui est parmi les versions les plus utilisées.

IV.2.4 3ds MAX :



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.

Le logiciel est issu du programme 3D Studio qui a tourné sous DOS jusqu'à sa version 4. Les programmeurs de Kinetix (une division d'Autodesk rebaptisée maintenant Autodesk Media and Entertainment) ont mis au point un logiciel entièrement nouveau et repensé.

3dsmax est ainsi conçu sur une architecture modulaire, compatible avec de multiples *plug-ins* (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 les films d'animation.

On a utilisé 3ds MAX pour la réalisation de l'objet virtuel à projeté dans la scène.

IV.3 Architecture de l'application:

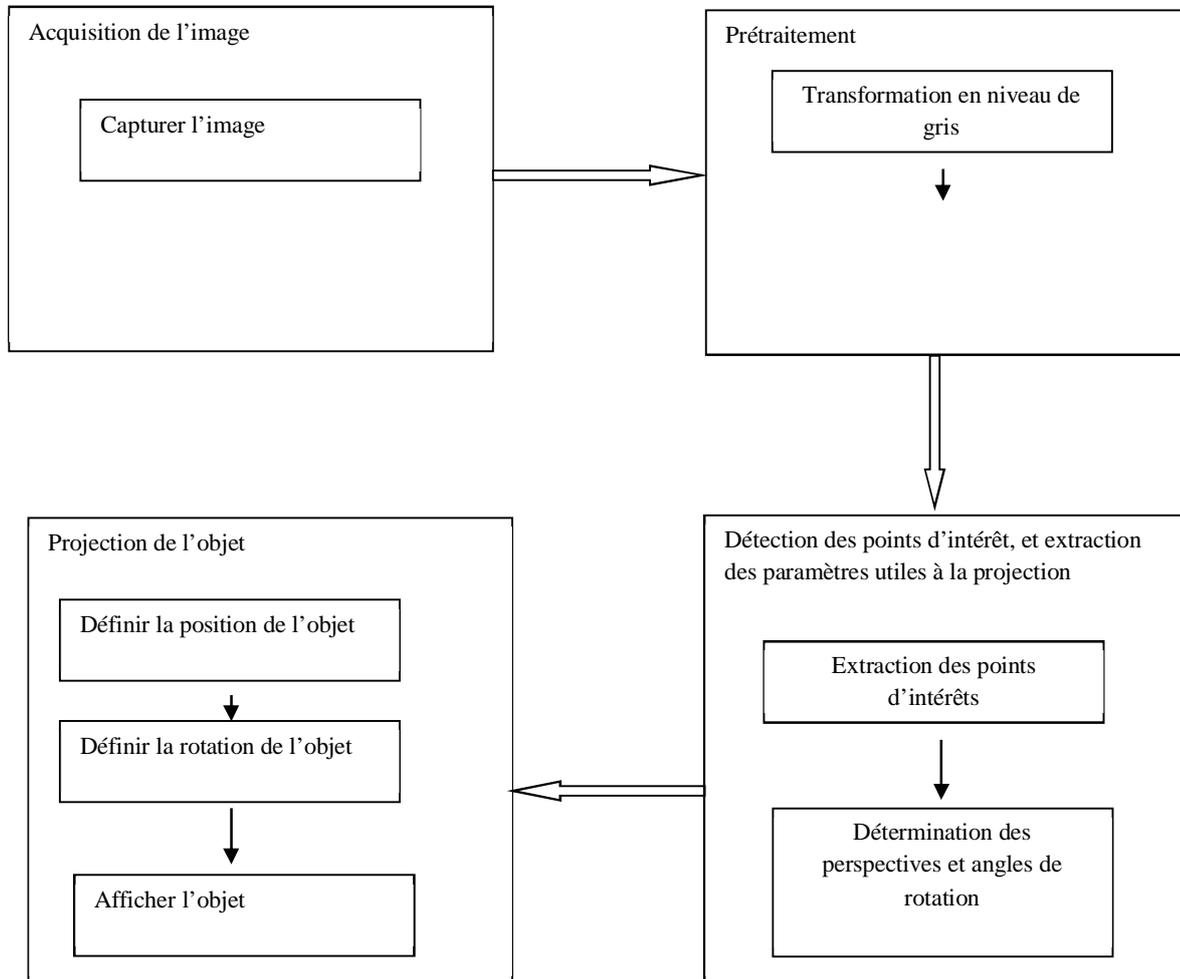


Figure IV.1 Schéma illustratif de conception de l'application réalisée.

IV.4 Acquisition de l'image :

Avant de débiter la conception, nous allons parler de quelques bases d'OpenCV et nous éclaircissant les méthodes utilisées pour la gestion des ressources matérielles (Webcam) et des données (images, attributs...).

OpenCV a mis en notre possession les outils permettant de prendre et prélever des séquences d'images via le périphérique Webcam.

- Les images sous OpenCV sont de type `IplImage` comme suit :

`IpImage *img`; avec `*img` : pointeur vers l'image.

- Pour acquérir une image depuis la webcam, on procède par le type `CvCapture` définissant une séquence :

`CvCapture *capture` ;

- La fonction : `Capture = cvCaptureFromCAM (entrée)`; permet l'acquisition de données depuis la webcam, avec entrée : le code du périphérique vidéo(0 par défaut).

IV.5 Prétraitement :

Afin d'améliorer la qualité du suivi d'objet, on procédera par quelques traitements qui nous permettront d'éliminer les bruits et de ne garder que les données utiles.

IV.5.1 Transformation de la séquence au niveau de gris :

Lors de l'acquisition de données depuis la webcam, la séquence vidéo en cours n'est autre qu'une suite d'images couleur synchronisées (**R**ouge, **V**ert, **B**leu), ces dernières sont enregistrées sous forme matriciel (vu dans le premier chapitre). Pour simplifier et augmenter la vitesse du traitement, nous transformons chaque image au niveau de gris suivant la formule: $Y=0,299.R+0,587.G+0,114.B$.

La fonction : `cvCvtColor (frame, image, CV_BGR2GRAY)` ; permet d'accomplir cette transformation au niveau de gris, avec en entrée dans l'ordre l'image source, la destination, et le type de la transformation à faire.

IV.5.2 Seuillage:

Toujours dans le but de simplifier la scène à étudier, nous allons en faire un seuillage (binarisation) :

```
double cvThreshold( const CvArr* src, CvArr* dst, double threshold, double maxValue, int thresholdType );
```

- `Src` : c'est l'image source
- `Dst` : c'est l'image destination.
- `Threshold` : la valeur du seuil.
- `Type` du seuillage.

Après simplification de la scène, nous allons entamer les phases sensibles de l'application, le but est de suivre un marqueur tout au long de notre séquence, et de prélever par la suite la position, et l'inclinaison de ce dernier.

On prend une image repère (template) que nous allons manipuler sur la scène réelle. Le principe est de prélever les points d'intérêts communs entre cette dernière et les images captées à partir de la webcam, pour en détecter l'orientation et l'emplacement de la template, et cela, tout le long de la séquence. Pour se faire, plusieurs techniques existent dans la littérature [7,8,9]. Dans notre application, nous avons opté à utiliser la méthode SURF pour le prélèvement des points d'intérêts.

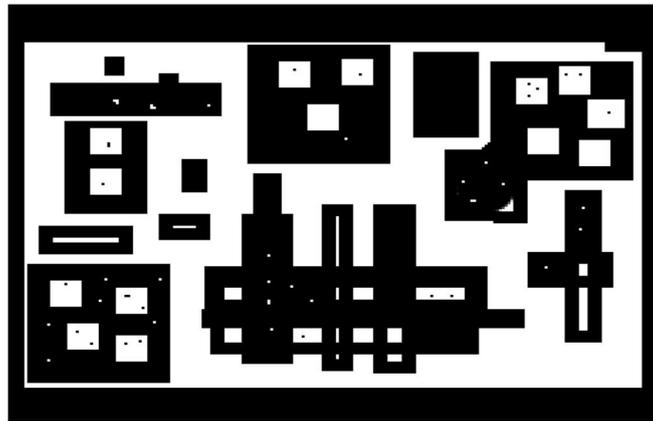


Figure IV.2 : Template utilisé pour le suivi

IV.6 Détection des points d'intérêt, et extraction des paramètres utiles à la projection

IV.6.1 Extraction des points d'intérêts :

Les points sont caractérisés par la configuration de leur voisinage. Le descripteur associé au point doit être le plus robuste possible afin d'être reconnu même si la position et l'orientation du document diffèrent entre les deux images à comparer (voir chapitre III)..

Nous avons choisi la méthode qui utilise le descripteur SURF car elle est principalement connue pour sa rapidité. Des études comparatives ont montré la supériorité du descripteur SURF par rapport à SIFT et PCA-SIFT d'un point de vue de ces performances en temps d'exécution et de sa robustesse aux changements d'illumination [16].

La méthode utilisée sous OpenCV est :

```
Void cvExtractSURF (const CvArr* image, const CvArr* mask, CvSeq** keypoints, CvSeq**  
descriptors, CvMemStorage* storage, CvSURFParams params );
```

Qui extrait les paramètres d'une image avec :

- Image : l'image au niveau de gris.
- Mask : un champ facultatif contenant une surface à ne pas traiter.
- Keypoints : Paramètre de sortie des points à traiter.
- Descriptors : méthode utilisé pour l'extraction des points.
- Storage : La mémoire ou enregistrer les points et les descripteurs.
- Params : Paramètre de l'algorithme

Après extraction des points d'intérêts, on procède à la sélection des extrémités de notre template grâce à une méthode qui détecte les points polaires :

```
locatePlanarObject ( objectKeypoints, objectDescriptors, imageKeypoints, imageDescriptors,  
src_corners, dst_corners );
```

Cette méthode Prend en entrée les descriptions et les paramètres des points extraits.

Les points extrêmes sont de type cvPoint définissant les points avec les coordonnées X et Y sous forme : pt(x,y), avec x : projection du point suivant l'axe X, de même pour Y, ce qui nous a permis de récupérer les coordonnées des quatre coins de la template (pt1, pt2, pt3 et pt4).

En reliant ces différents points, on aura la template encadré dans un rectangle tout au long de la séquence vidéo (figure IV.3) grâce à la méthode :

```
cvLine ( core2, cvPoint (r1.x, r1.y ), cvPoint (r2.x, r2.y ), colors[8] );
```

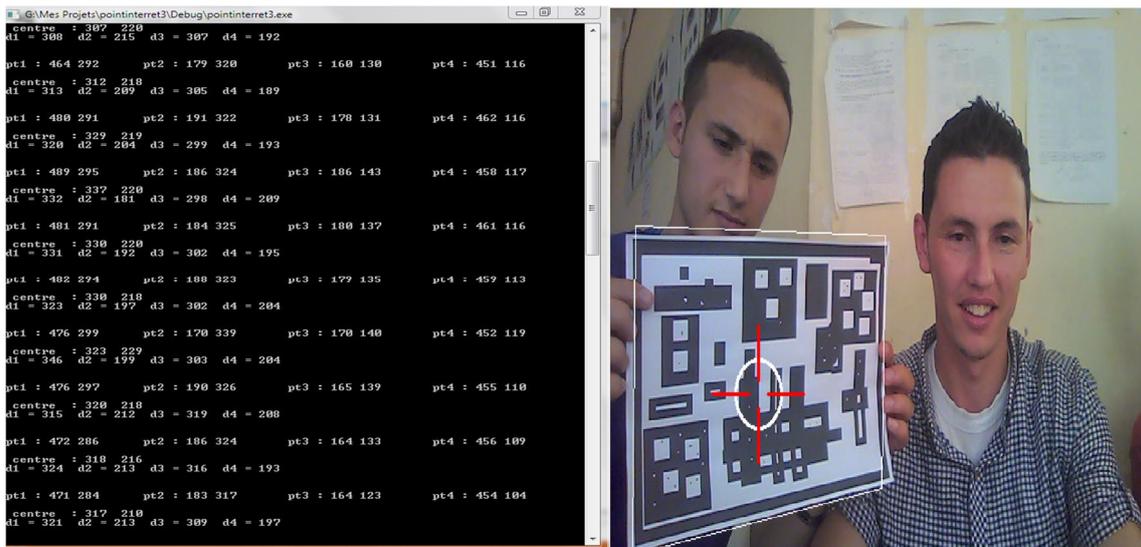


Figure IV.3 Détection et suivi de la template lors de la capture vidéo.

IV.6.2 Détermination des perspectives :

Le but des traitements et suivi fait précédemment est de prélever les coordonnées du repère dans la scène réelle, ainsi que les rotations suivant les différents axes, afin de les passer comme paramètres à la prochaine fonction qui projetera l'objet virtuel sur le marqueur.

La méthode SURF avec laquelle on a prélevé les points d'intérêts nous a permis de prendre les extrémités de notre repère, prélevant les coordonnées des quatre points A, B, C et D suivant les axes X et Y de l'image. A partir de ces points, nous allons pouvoir étudier les changements subits au repère: sa position, translation et sa rotation suivant les axes X, Y, et Z.

Dans ce qui suit, nous allons illustrer la technique utilisée pour déterminer les propriétés décrites ci-dessus.

IV.6.2.1 Calcul du point de translation:

Ce point exprimé selon les valeurs X et Y est primordial pour la suite de l'application, prenant un point (le centre du repère détecté) suivant le mouvement de la template dans la scène réelle. :

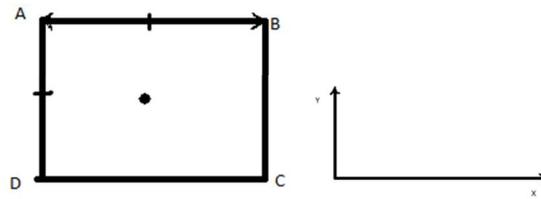


Figure IV.4 : calcul du point de translation.

La template étant localisée dans la scène sera encadré dans le rectangle ABCD, dont on connaît les valeurs des nœuds par rapport au repère X et Y.

Le centre du rectangle centre(x, y) est alors calculé comme suit :

$$\text{Centre} = ((B.x+A.x)/2), (D.y+A.y)/2). \quad (4.1)$$

V.6.2.2 Calcul de la distance:

Indiquant la distance de la template par rapport à la webcam, cette méthode est importante pour l'utiliser par la suite dans la projection de l'objet, en ajustant la taille de l'objet suivant cette distance pour maitre en scène un système homogène, proche de la réalité.

La fonction : float dist(CvPoint pt1, CvPoint pt2, CvPoint pt3, CvPoint pt4){} ;

Afin de calibrer la caméra, cette fonction fonctionne sous le principe de ressemblance des triangles, en prenant une feuille de largeur connu **D** et on la place sur une distance connu **Z**, on calcule la largeur **d** de la feuille a cette distance sur l'écran, donc on aura la distance focale **f = d*Z/D**. On prend maintenant une autre frame de la vidéo et on calcule sa largeur on aura la valeur **d'**, et toujours par la règle de ressemblance des triangles on aura **f/d' = Z'/D**. ce qui nous donne Alor **Z' =D*f/d'** et c'est la distance de la feuille par rapport à la caméra.

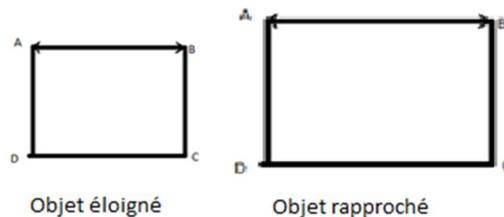


Figure IV.5 : calcul de la distance

IV.6.2.3 Calcul des rotations:

Cette étape est aussi faite grâce aux sommets de la template détecté.

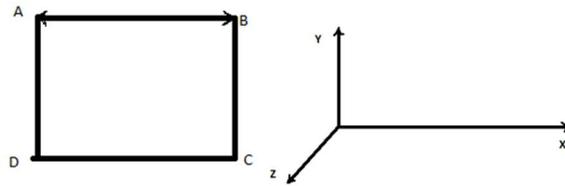


Figure IV.6 : calcul des rotations.

IV.6.2.3.1. Suivant le repère X :

La fonction : `float calculrotX(CvPoint pt1,CvPoint pt2){ } ;`

Sachant d'après les tests qu'on a effectués sur OpenGL, le rapport des deux distances opposées (haut et bas) nous donne une valeur 0,7 (la petite sur la grande distance) après avoir fait une rotation de 45° degré sur un carré. Donc la fonction nous retourne l'angle de rotation suivant l'axe X selon le changement des distances d'extrémités.

En prenant X comme axe de rotation, on aura une rotation d'angle α par rapport à la position initial ($\alpha=0$) comme suit:



Figure IV.7 : calcul des rotations suivant l'axe X.

IV.6.2.3.2 Suivant le repère Y :

La fonction : `float calculrotY (CvPoint pt1, CvPoint pt2, CvPoint pt3, CvPoint pt4){ } ;`

Cette fonction a le même principe que la précédente.

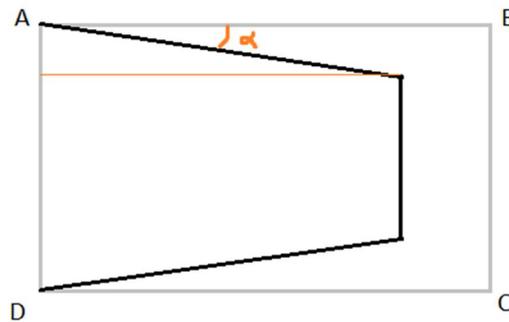


Figure IV.8 : calcul des rotations suivant l'axe Y.

IV.6.2.3.3 Suivant le repère Z :

La fonction : float calculrotZ (CvPoint pt1, CvPoint pt2)

En prenant le point D comme le point de rotation, on obtient :

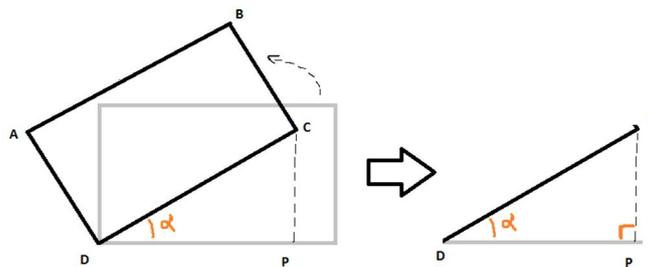


Figure IV.9: calcul des rotations suivant l'axe Z.

Le marqueur tourne sur l'axe Z, sur le point D, Formant un angle α par rapport à sa position initial ($\alpha=0$), en appliquant la loi de Pythagore sur les triangles rectangulaires, on trouve :

$$\text{Sinus } \alpha = \frac{\text{Côté Opposé}}{\text{Hypoténuse}}$$

Avec : Côté Opposé=CP (P c'est la projection du point c(x,y) sur l'axe x)

Hypoténus = DC.

Les calculs que nous avons fait précédemment nous ont permis de prélever la translation et la rotation de l'objet suivie, c'est ce que dont nous avons besoin en réalité, parce que c'est à partir de ces données que nous allons ajuster notre objet à projeter.

IV.7 Projection de l'objet :

Nous sommes arrivés à une autre étape de l'application, qui est la projection d'objet, et comme nous allons le voir, OpenGL a mis à notre portée tous les outils et méthodes nécessaires à cette projection.

La fonction qui permet de manipuler un programme sous OpenGL est :

```
Void display (void){ } ;
```

Au début, dans cette fonction on doit charger l'objet conçu avec 3DSmax à OpenGL :

```
char lect_objet_3d ( ptr_objet, char *p_filename) ;
```

avec ptr_objet : pointeur vers l'objet 3ds.

p_filename : chaîne de caractère indiquant le nom de l'objet.

Cette fonction nous permet de charger un objet 3DsMax (sous l'extension .3DS), autrement dit : elle permet d'interpréter le code en programme compréhensible par OpenGL.

Une fois l'objet récupéré, on doit le projeter de façon à respecter les coordonnées de la template détectée précédemment, le régler de façon à le projeter sur la template, et de suivre les rotations de cette dernière suivant les différents axes.

La fonction permettant de définir la position de l'objet dans la scène est :

```
glTranslatef (centre.x, centre.y, translatZ);
```

Cette fonction prend en argument les coordonnées du point centre calculé précédemment et la distance de la template par rapport à la camera.

L'application représentant l'objet projeté sur la scène réelle, suivant la position et la distance par rapport à la template, il lui manque que les rotations pour être achevé.

La fonction : glRotatef (rotationX, rotationY, rotationZ); permet d'accomplir les rotations, en prenant en paramètre les angles de rotation calculés précédemment.

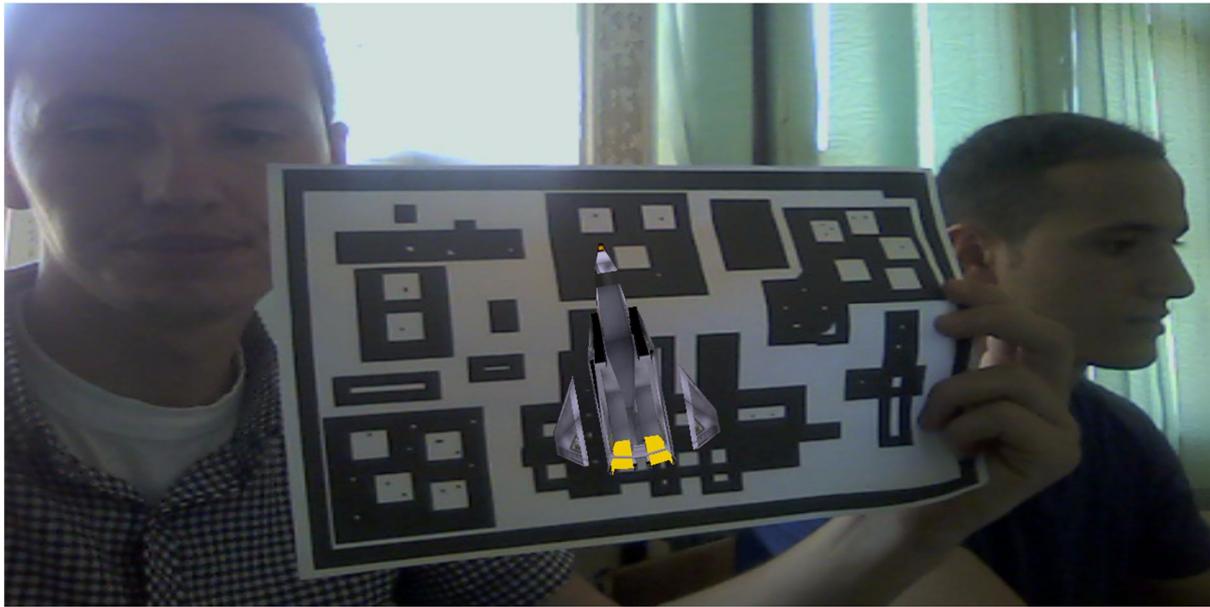


Figure IV. 10. Projection de l'objet 3D sur la scène réelle.

IV.8 Résultats expérimentaux et critiques :

L'application réalisée, certes nous a permis de mettre en œuvre un des usages clés de la réalité augmentée, qui est la projection d'objets virtuels sur une scène réelle, mais, présentant aussi quelques imperfections, parfois inévitables, vu la sensibilité du domaine, et les exigences de performances côté matériel. Parmi les lacunes rencontrées en mettant en œuvre l'application, on cite :

- Limitation de la vitesse de traitement, ressentie lors de la détection de la template, problème essentiellement dû à la qualité de la webcam (qualité d'image) lors de l'acquisition de données.
- Limitation de la détection de la template, en l'inclinant d'un degré extrême, dû aux lacunes de l'algorithme mis en œuvre (SURF), et de la qualité d'image citée ci-dessus.
- Limitation de la procédure de projection due au chargement continu de texture mis en œuvre, problème principalement dû aux capacités de la machine.

IV.9 Conclusion :

Au cours de ce chapitre, nous avons réalisé notre modeste application, qui nous a permis d'exploiter le domaine de la réalité augmentée, et d'acquérir des connaissances avancées que

ça soit dans le domaine, concevant un programme qui illustre ce dernier, et dans l'un des langages de programmation les plus utilisés qui est le C++.

Dans la conception, nous nous sommes épaulé seulement sur quelques méthodes et fonctions que nous avons utilisées au cours de la programmation, qui nous ont permis de mieux expliquer nos procédés suivies.

Conclusion générale :

Dans ce mémoire, nous avons pris part au domaine de la réalité augmentée, nous nous sommes intéressés à la vision assistée par ordinateur, où nous avons présenté quelques concepts de bases et les étapes de traitement artificiel d'une image accompagnées de quelques exemples concrets.

Cette première démarche nous a permis d'entamer la réalité augmentée en définissant le domaine, ses outils et son usage avec des exemples illustratifs. Et ensuite nous familiariser en présentant et comparant les méthodes les plus utilisées pour le suivi d'objets dans une séquence vidéo.

En dénouement de l'étude, nous nous sommes consacrés à la conception et réalisation d'une application. Cela en mettant en œuvre un système projetant un objet virtuel sur une scène réelle.

En résumé, notre mémoire permet de découvrir le domaine la réalité augmentée, et son usage prometteur dans la scène technologique.

Bibliographie

- [1] C. HARRIS et M. STEPHENS.
« A Combined Corner and Edge Detector ». Dans Alvey Vision Conference, p. 147–151, 1988.
- [2] H. P. MORAVEC.
« Toward automatic visual obstacle avoidance ». Dans IJCAI, volume 2, p. 584–584, 1977.
- [3] L. KITCHEN et A. ROSENFELD.
« Gray level corner detection ». PRL, 1(2) :95–102, 1982.
- [4] P. R. BEAUDET.
« Rotationally invariant image operators ». Dans ICPR, p. 579–583, 1978.
- [5] S. M. SMITH et J. M. BRADY.
« SUSAN – A New Approach to Low Level Image Processing ». IJCV, 23(1) :45–78, 1997.
- [6] E. ROSTEN et T. DRUMMOND.
« Machine learning for high-speed corner detection ». Dans ECCV, volume 1, p. 430–443, 2006.
- [7] D. G. LOWE.
« Object Recognition from Local Scale-Invariant Features ». Dans ICCV, volume 2, p. 1150–1157, 1999.
- [8] K. MIKOLAJCZYK et C. SCHMID.
« Scale & Affine Invariant Interest Point Detectors ». IJCV, 60(1) :63–86, 2004.
- [9] H. BAY, A. ESS, T. TUYTELAARS et L. V. GOOL.
« SURF : Speeded Up Robust Features ». Dans CVIU, p. 346–359, 2006.
- [10] T. KADIR, A. ZISSERMAN et M. BRADY.
« An affine invariant salient region detector ». Dans ECCV, p. 404–416, 2004.
- [11] W. FÖRSTNER.
« A framework for low level feature extraction ». Dans ECCV, p. 383–394, 1994.
- [12] V. GOUET, P. MONTESINOS, R. DERICHE et D. PELÉ.
« Évaluation des détecteurs de points d'intérêt pour la couleur ». Dans RFIA, p. 257–266, 2000.
- [13] C. HARRIS et M. STEPHENS.
« A Combined Corner and Edge Detector ». Dans Alvey Vision Conference, p. 147–151, 1988.
- [14] J. A. NOBLE.
« Finding corners ». IVC, 6(2) :121–128, 1988.
- [15] David G. Lowe,
« Distinctive image features from scale-invariant keypoints », International Journal of computer Vision, vol. 60, n° 2, p. 91–110, 2004
- [16] Luo G. , Oubong G.

« International Journal of Image Processing (IJIP) » Vol 3, Issue 4,p.143-152.