

Mémoire de magistère:

Suivi de modèle et Incrustation d'Objets Virtuels pour la Réalité Augmentée

Présenté par : Mohamed BADECHE

Président : Dr. Allaoua Chaoui, (Université de Cne, Dpt. Informatique)

Rapporteur : Pr. Mohamed Benmohammed, (Université de Cne, Dpt. Informatique)

Examineurs : Dr. Djamel Eddine Saidouni, (Université de Cne, Dpt. Informatique)

Dr. Bilami Azzedine, (Université de Batna, Dpt. Informatique)

Dr. Abdelmadjid Zidani, (Université de Batna, Dpt. Informatique)

« ...*Ainsi la recherche passe par des moments durables et distincts ; intuition, aveuglement, exaltation et fièvre. Elle aboutit un jour à cette joie et connaît cette joie celui qui a passé par ces moments singuliers... »*

Albert Einstein

Remerciements

Je tiens à remercier en premier lieu mon encadreur, pour avoir accepté de diriger ce travail ; je le remercie pour la confiance qu'il a mis en moi, pour la liberté qu'il a bien voulu me laisser, pour sa modestie, sa simplicité et son assistance ; il était là chaque fois que j'en avais besoin.

Je remercie chaleureusement mes parents ainsi que chaque membre de la famille, cette famille que je considère comme étant l'un des plus chers dons qu'on m'a jamais offert.

Je remercie beaucoup et particulièrement mon ancien binôme durant la préparation du projet de fin d'étude de notre ingénieur « Lamine Berrandjia », qui sans lui ce travail n'aurait certainement jamais pu voir le jour, et je remercie également mon ancien encadreur « Mme S. Baatouche ».

Je remercie tous mes amis, toutes mes connaissances et tous ceux qui n'ont pas manqué de me soutenir avec leurs encouragements et souhaits de succès et spécialement mes amis à l'association « El Irtiq'a », chacun par son nom.

Et avant et après tout, louange à Allah, Seigneur de l'univers.

Résumé

La réalité augmentée est une technologie qui permet l'insertion d'objets virtuels 2D ou 3D dans des scènes du monde réel en temps réel. La majorité des avancées dans la réalité augmentée ont eu lieu ces dix dernières années.

Les systèmes de composition d'images sont aujourd'hui utilisés aussi bien pour la production d'effets spéciaux (cinéma ou publicité), que pour la reconstruction visuelle de faits ou de structures historiques. Par contre, les applications à base de HMD (Head Mounted Display) sont plus expérimentales que réellement utilitaires. Nous trouvons en particulier des applications dans les domaines de la médecine, militaire, l'assistance en milieu industriel et le design intérieur ou du jeu.

Les techniques de suivi (Tracking) basées modèles sont très prometteuses pour les applications de réalité augmentée. Dans ce mémoire nous décrivons une méthode basée sur le suivi d'un modèle 2D le long d'une séquence vidéo tout en respectant la contrainte temps sur de simples ordinateurs de bureau.

D'abord, une recherche d'un pattern bien défini dans la première prise de vue de la séquence vidéo est effectuée moyennant une mise en correspondance entre cette dernière et le pattern en question. Ensuite, un suivi des primitives coins du pattern est appliqué en exploitant les capacités de prédiction du filtrage de Kalman discret.

Enfin, l'augmentation proprement dite de la scène par l'objet choisi est effectuée. Une estimation robuste de l'homographie planaire nécessaire pour l'incrustation de l'objet 2D dans la scène est assurée par l'utilisation d'une variante de l'algorithme RANSAC.

Les résultats obtenus par application de la méthode sur plusieurs séquences d'images sont très prometteurs et démontrent son efficacité et son respect à la contrainte temps réel sur de simples ordinateurs de bureau.

Mots clés : Réalité augmentée, Suivi de points, Suivi temps réel, Filtre de Kalman, Détecteur de coins 'Harris', Mise en correspondance 'Pilu', l'algorithme 'RANSAC'.

Table des matières

REMERCIEMENTS.....	3
RESUME.....	4
TABLE DES MATIERES.....	1
LISTE DES FIGURES.....	4
LISTE DES TABLES.....	7
PRESENTATION DE LA THESE.....	9
INTRODUCTION.....	9
CONTRIBUTIONS.....	10
ORGANISATION DU MEMOIRE.....	10
.....	12
PROBLEMATIQUE.....	13
CHAPITRE 1 : INTRODUCTION GENERALE A LA VAO.....	13
1.1 INTRODUCTION.....	13
1.2 DEFINITION DE BASE.....	14
1.3 LES ETAPES DU TRAITEMENT ARTIFICIEL D'UNE IMAGE EN VAO.....	17
1.3.1 Acquisition.....	17
1.3.2 Prétraitement.....	18
1.3.2.1 La compression.....	18
1.3.2.2 La restauration.....	18
1.3.2.3 L'amélioration.....	18
1.3.2.3.1 La modification d'histogramme.....	19
Le recadrage dynamique.....	19
Mise en évidence de zones de l'histogramme.....	20
Egalisation de l'histogramme.....	20
1.3.2.3.2 Le filtrage.....	21
Le filtrage linéaire.....	21
Le filtrage non linéaire:.....	23
1.3.3 L'analyse.....	23
1.3.3.1 Segmentation par extraction de contour.....	24
1.3.3.2 Segmentation en régions homogènes.....	25
1.3.4 L'interprétation.....	25
1.4 PROBLEME POSE PAR LA REALITE AUGMENTEE.....	26
CHAPITRE 2 : INTRODUCTION A LA REALITE AUGMENTEE.....	27
2.1 INTRODUCTION.....	27
2.2 MOTIVATION ET DOMAINES D'APPLICATION.....	28
2.2.1 Médecine.....	30
2.2.2 Assistance en milieu industriel.....	30
2.2.3 Design intérieur.....	31
2.2.4 Jeux.....	31
2.2.5 Effets spéciaux.....	32
2.2.6 Etude d'impact.....	32
2.3 LES SYSTEMES DE REALITE AUGMENTEE.....	33
2.3.1 Caméra et technologies <i>d'affichages</i>	33
2.3.1.1 Configurations basées Moniteur.....	33
2.3.1.2 Configurations basées vidéo vue à travers.....	34
2.3.1.3 Configuration optique vue à travers.....	35
2.3.1.4 Comparaison entre la méthode vidéo et la méthode optique.....	36

2.3.2 Appareils permettant l'interaction à l'utilisateur	37
2.4 LES APPROCHES DE RECALAGE	37
2.4.1 L'approche basée capteurs	38
2.4.2 L'approche basée vision.....	38
2.5 LES ERREURS DE RECALAGE	39
2.5.1 Les erreurs statiques.....	39
2.5.2 Les erreurs dynamiques	39
2.6 PROBLEMATIQUE	39
2.6.1 Cohérence géométrique.....	40
2.6.2 Cohérence photométrique.....	40
CHAPITRE 3 : CONCEPTION D'UN SYSTEME DE REALITE AUGMENTEE	42
3.1 PROBLEMATIQUE	42
3.1.1 Le cas 2D.....	42
3.1.2 Extension pour le cas 3D	45
3.1.2.1 Calibration.....	45
3.1.2.2 La géométrie affine.....	47
3.2 DESCRIPTION DU SYSTEME PROPOSE	47
3.3 PHASE DE RECHERCHE DU MODELE.....	49
3.3.1 Choix du pattern (le modèle)	49
3.3.2 <i>Extraction des points d'intérêt</i>	51
3.3.3 Localisation du modèle	51
3.4 PHASE DE SUIVI DU MODELE LE LONG DE LA SEQUENCE	53
3.4.1 Introduction théorique au filtre de Kalman	54
3.4.1.1 Le processus à estimer	54
3.4.1.2 Méthode.....	55
3.4.2 <i>Suivi du modèle sur la séquence d'images</i>	57
3.4.2.1 Suivi des primitives à base du filtre de Kalman	57
3.4.2.2 Algorithme du suivi des primitives	58
3.4.2.3 Réglage du filtre.....	59
3.4.2.4 Calcul robuste de l'homographie.....	59
3.5 PHASE D'INCRUSTATION DE L'OBJET VIRTUEL.....	61
CHAPITRE 4 : TEMPS REEL	63
4.1 INTRODUCTION.....	63
4.2 DEFINITION D'UN SYSTEME TEMPS REEL	63
4.3 CLASSIFICATION DES SYSTEMES TEMPS REEL.....	64
4.3.1 Temps réel strict	64
4.3.2 Temps réel souple	65
4.3.3 Temps réel ferme	66
4.3.4 Système embarqué et système détaché.....	67
4.4 BESOIN DE TRAITEMENT EN TEMPS REEL DANS LA REALITE AUGMENTEE.....	67
4.4.1 <i>L'approche adoptée</i>	68
4.5 VITESSE DE TRAITEMENT DES ALGORITHMES D'IMAGERIE.....	69
4.5.1 Solution logicielle	70
4.5.2 Solution matérielle	71
4.5.3 Cas de figure particulier.....	72
4.5.3.1 Détection de coin.....	72
4.5.3.2 Filtrage.....	74
4.6 CONCLUSION	76
CHAPITRE 5 : RESULTATS EXPERIMENTAUX ET DISCUSSIONS.....	77
5.1 SUIVI.....	77
5.2 ROBUSTESSE	80
5.2.1 Au niveau de la première phase.....	80
5.2.2 Au cours du suivi.....	81
5.2.3 <i>Effet d'orientation</i>	81
5.2.4 <i>Effet d'échelle</i>	82
5.3 REALISME	83
5.4 TEMPS REEL	84

5.4.1 Conclusion	87
CHAPITRE 6 : CONCLUSION ET PERSPECTIVES	88
BIBLIOGRAPHIES.....	91
ANNEXE : REALISATION DU SYSTEME.....	94
SPECIFICATION DES BESOINS.....	94
SPECIFICATION DES COMPOSANTS.....	94
Module de recherche	95
Module de Suivi.....	95
<i>Module d'incrustation</i>	96
DIAGRAMME DE FLUX DE DONNEES	97
ARCHITECTURE GLOBALE DU SYSTEME	97
COMMUNICATION	98

Liste des figures

Figure 1.	La chaîne de VAO	13
Figure 2.	Une partie d'image en niveau de gris et sa matrice représentative	15
Figure 3.	Une image et son histogramme représentatif	16
Figure 4.	Exemple de recadrage dynamique.....	20
Figure 5.	LUT de binarisation.....	20
Figure 6.	Egalisation d'histogramme	21
Figure 7.	Filtrage linéaire d'une image ; à gauche l'image initiale et à droite l'image filtrée en utilisant un filtre moyenne 5x5.....	22
Figure 8.	Filtre gaussien.....	23
Figure 9.	Filtrage non linéaire d'une image. A gauche l'image initiale et à droite l'image filtrée en utilisant un filtre médian 5x5.....	23
Figure 10.	Opérateur de Roberts	24
Figure 11.	Opérateur de Sobel sur la précédente image	24
Figure 12.	Laplacien sur la même image.....	25
Figure 13.	Exemple de la réalité augmentée pour la construction de pont	28
Figure 14.	Exemple de la réalité augmentée basée suivi de modèle [5]	29
Figure 15.	Visualisation du fœtus à l'intérieur du ventre de la patiente [7].....	30
Figure 16.	Une Interface permettant de positionner des meubles virtuels dans une salle réelle 31	
Figure 17.	La réalité augmentée appliquée au jeu du TIC TAC TOC [11]	32
Figure 18.	Certains effets spéciaux de la trilogie Star Wars utilisent des images de synthèse superposées aux images réelles	32
Figure 19.	Exemple d'illumination artificielle d'un pont. A gauche l'image réelle, à droite l'image augmentée.....	33
Figure 20.	Configuration basée moniteur [14].....	34
Figure 21.	HMD vidéo vue à travers [14]	34
Figure 22.	Système de réalité augmentée basé vidéo vue à travers [14]	35
Figure 23.	HMD optique vue à travers [15].....	35
Figure 24.	Système de réalité augmentée basé optique vue à travers [14]	36
Figure 25.	Modèle de calibration de la caméra	39
Figure 26.	Cohérence géométrique et photométrique pour la séquence du pont.....	41
Figure 27.	Systèmes de coordonnées de la caméra et du modèle	43
Figure 28.	Relation entre le monde réel et le monde image.....	45
Figure 29.	Décomposition du modèle de Tsai	46
Figure 30.	Calibration à base de 4 points (géométrie affine) [14]	47

Figure 31.	(a) exemple d'un objet virtuel à insérer (b) image de référence du modèle	.48
Figure 32.	Calibration à base de pattern planaire	50
Figure 33.	Exemples de modèles [20]	51
Figure 34.	Détection de coins avec HARRIS	51
Figure 35.	Mise en correspondance pour la localisation du modèle	52
Figure 36.	Génération d'une mauvaise correspondance	53
Figure 37.	Traitement de RANSAC de la mauvaise correspondance	53
Figure 38.	Fonctionnement du filtre de Kalman	56
Figure 39.	Utilisation de RANSAC pour la correction des mauvaises correspondances en cas d'occlusion	60
Figure 40.	Projection de l'objet virtuel dans la séquence vidéo	61
Figure 41.	Visualisation réaliste dans le cas d'occlusion du modèle	62
Figure 42.	Qualité d'un Système temps réel dur en cas de dépassement d'une échéance [32]	65
Figure 43.	Qualité d'un Système temps réel souple en cas de dépassement d'échéance [32]	66
Figure 44.	Qualité d'un Système temps réel ferme [32]	66
Figure 45.	Augmentation avec l'image intérieure du cerveau d'un patient issue de donnée Rayon X [15]	67
Figure 46.	Parallélisme de donnée	70
Figure 47.	Détection de contour	71
Figure 48.	Exécution de la détection de coins par module	72
Figure 49.	Impact du choix de la partie d'image sur la détection de coins	73
Figure 50.	Principe du filtrage	75
Figure 51.	Visualisation de la légère différence entre les résultats du filtrage sur l'image entière et après partition de celle-ci sur quatre parties	75
Figure 52.	Visualisation du processus de suivi d'un point à travers le filtrage Kalman	77
Figure 53.	Visualisation du processus de suivi d'un point par le filtre de Kalman	78
Figure 54.	Ambiguïté sur le point à suivre due à une taille de la fenêtre trop grande	79
Figure 55.	Perte de points en raison d'une taille de la fenêtre trop petite	79
Figure 56.	Récupération des points perdus	80
Figure 57.	Détection et correction des mauvaises correspondances issues de la phase de mise en correspondance	80
Figure 58.	correction des mauvaises correspondances issues d'une occlusion partielle	81
Figure 59.	Suivi du modèle tout au long d'une séquence vidéo de 200 prises de vue, en présence de variations d'orientation	82
Figure 60.	Effet d'échelle sur le suivi le long d'une séquence vidéo de 400 prises de vue	83
Figure 61.	Positionnement de l'objet virtuel au dessous de l'objet générant l'occlusion	84

Figure 62.	Visualisation graphique du Tableau 1	85
Figure 63.	Visualisation du Tableau 2	86
Figure 64.	Insertion d'objet 3D dans une séquence vidéo.....	89
Figure 65.	Suivi à base de détection de coins du visage d'un être humain	90

Liste des tables

Table 1.	Temps de calcul du processus de suivi de dix images en fonction de la résolution et des caractéristiques de la machine d'exécution.....	85
Table 2.	Détail du temps de calcul du suivi.....	86

Présentation de la thèse

Introduction

Contrairement à la réalité virtuelle, qui émerge l'utilisateur dans un environnement artificiel généré par ordinateur, la réalité augmentée est une technologie qui tente d'améliorer la vue de l'utilisateur du monde réel par insertion d'objets virtuels, tel qu'un texte, des images 2D ou des modèles 3D.

La réalité augmentée trouve son application dans divers domaines allant de la médecine, du secteur militaire, de l'assistance en milieu industriel jusqu'au divertissement.

Pour que la fusion de l'objet virtuel avec la scène réelle soit faite de manière réaliste, il y a plusieurs défis à relever qu'on peut résumer sous trois axes principaux ; les contraintes géométriques, les contraintes photométriques et les contraintes temporelles.

Les premières concernent la précision de calcul de la position hypothétique de l'objet virtuel et aussi son orientation par rapport à la scène réelle.

Par contre, Les contraintes photométriques traitent des questions relatives à l'éclairage et à l'ombre pour que l'objet virtuel puisse apparaître comme étant partie intégrante de la scène.

Alors que les contraintes temporelles, sont évoquées dès qu'on parle d'application temps réel.

Le terme temps réel est relatif à la vitesse de calcul et d'affichage des images. D'un point de vue intuitif, il signifie que l'image désirée doit être calculée et affichée suffisamment rapide pour que l'utilisateur ait l'impression que tout s'est passé instantanément. On convient généralement qu'il faut environ vingt cinq

images par seconde, ce qui correspond à la fréquence de rafraîchissement d'un écran de télévision.

Les contraintes de temps sont alors très élevées. En effet, une image doit être produite toutes les quarante millisecondes. Sachant que pour construire une image, on calcule plusieurs millions de pixels, il faut donc être en mesure de calculer un pixel environ toutes les dix nanosecondes. Ces contraintes limitent donc fortement la complexité des algorithmes que l'on peut utiliser. De plus elle suppose une mise en oeuvre de moyens matériels importants.

Contributions

Ce travail vise à faire une synthèse sur le domaine de la réalité augmentée, exposant les notions théoriques relatives au domaine, ainsi que l'état de l'art de ce qui est utilisé comme technologies dans cet axe.

Mis à part la théorie, le travail consiste à implémenter un système complet de réalité augmentée, partant de l'identification et du suivi de modèle et arrivant à l'augmentation proprement dite par des objets virtuels 2D.

La méthode proposée ici était le fruit du travail de fin d'étude de notre ingénieur **[1]** et qui a fait l'objet de deux communications ; la première à JSTA2003 à Guelma, et la deuxième dans le domaine de la productique, à la conférence internationale CIP2003 au CDTA (Centre de Développement des Technologies Avancées d'Alger) **[2]**.

Dans le cadre de cette thèse, nous avons continué dans le même axe, et nous nous sommes investis beaucoup plus, dans l'axe 3D et dans l'aspect temps réel.

Une partie importante donc de ce travail sera consacrée à la discussion de la contrainte temps réel dans le domaine de l'imagerie de manière générale, et particulièrement à l'approche du suivi adoptée par notre méthode.

Organisation du Mémoire

Le mémoire organisé en 6 chapitres fait le tour du domaine de la réalité augmentée. Après une présentation des techniques utilisées et des défis à relever, nous décrivons un système de réalité augmentée basé sur le suivi d'un modèle planaire et nous présentons les résultats de son implémentation sur des PCs.

Dans le premier chapitre, une introduction au domaine de la vision assistée par ordinateur partant des définitions les plus simples dans le domaine de l'imagerie.

Dans le deuxième chapitre, une présentation des notions relatives au domaine de la réalité augmentée ainsi que l'état de l'art des techniques utilisées.

Le troisième, quatrième et cinquième chapitres constituent le cœur de notre contribution ; dans le chapitre 3, nous expliquons les fondements mathématiques et présentons le système implémenté qui consiste à suivre un modèle planaire le long d'une séquence vidéo, tout en incrustant l'objet virtuel au niveau de chaque prise de vue. Tandis que le chapitre 4 est consacré entièrement à discuter la contrainte temporelle exigée dans la réalité augmentée, et finalement le chapitre 5 présente les résultats expérimentaux de l'implémentation du système, notamment la robustesse et le suivi temps réel.

Dans la conclusion, nous terminons par donner le bilan de la méthode et présenter les perspectives futures.



Problématique

Chapitre 1 : Introduction générale à la VAO

1.1 Introduction

Depuis 1950 date de l'apparition des premières analyses d'images dans les chambres à bulles, la vision assistée par ordinateur 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.

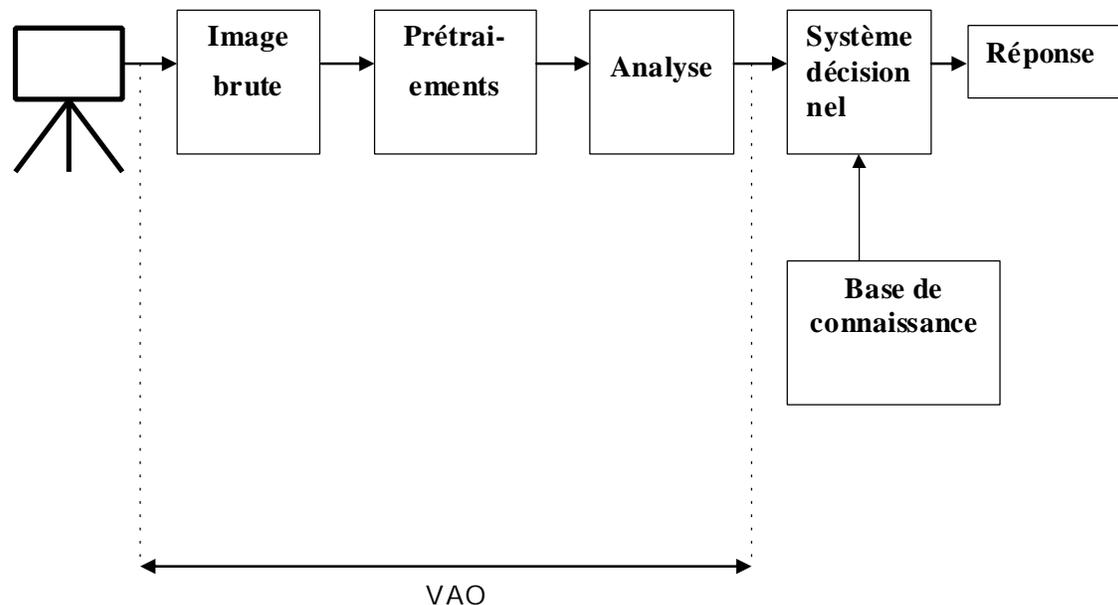


Figure 1. La chaîne de VAO

Les domaines d'application de la VAO se regroupent essentiellement suivant ces trois catégories :

- **L'imagerie aérienne et spatiale :** dans laquelle les traitements concernent l'amélioration des images satellites, l'analyse des ressources terrestres, la cartographie, les analyses météorologiques. Le type d'image utilisé est issu de cameras dans le visible ou l'infrarouge ainsi que le radar.
- **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 résonance magnétique nucléaire ainsi que dans le domaine de la reconnaissance automatique de cellules ou de chromosomes.
- **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.

1.2 Définition de base

- **Image :** Une image I est 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 de l'intensité lumineuse (Couleur ou niveau de gris).

$$I : [0, L-1] \times [0, C-1] \rightarrow [0, M]^P$$

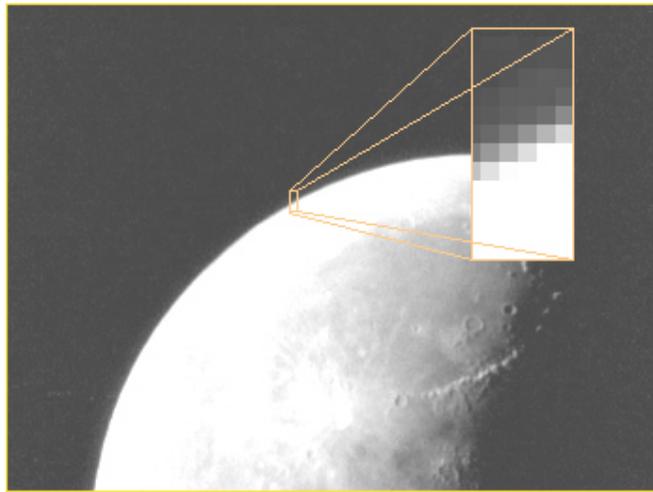
I est une image de L lignes et C colonnes dont l'information portée est définie sur un espace à P dimension.

- 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 intérieurement par une matrice (voir figure 2).

- **Pixel :** Un pixel (contraction de picture element) est le nom associé à une unité de base de l'image qui correspond à un pas de discrétisation. Un pixel est caractérisé par sa position et sa valeur (sa couleur). Le plus souvent ce vocable est utilisé pour représenter indifféremment l'un ou l'autre attribut.

- **Maillage :** Le maillage est l'arrangement géométrique des pixels dans l'image. On utilise généralement un des trois types qui résultent de différentes tessellations du plan par des figures géométriques. Tout d'abord le maillage carré qui correspond à la réalité physique du capteur CCD. On peut également utiliser un maillage hexagonal (maillage de référence pour le domaine de morphologie mathématique). 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
215	210	210	215	214

Figure 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 Manhattan :

$$d_1(P,Q) = |x_p - x_q| + |y_p - y_q|$$

- Distance Euclidienne :

$$d_2(P,Q) = \left[(x_p - x_q)^2 + (y_p - y_q)^2 \right]^{1/2}$$

- Distance de l'échiquier :

$$d_{\text{inf}}(P,Q) = \text{Max}(|x_p - x_q|, |y_p - y_q|)$$

Ces distances sont reliées par la propriété :

$$d_{\text{inf}}(P, Q) \leq d_2(P, Q) \leq d_1(P, Q)$$

- **Voisinage** : On appelle voisinage d'ordre K du pixel P et l'on note $V_k(P)$ l'ensemble des pixels Q défini par :

$$V_k(P) = \{Q : 0 < d(P, Q) \leq k\}$$

- **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 d_1 , il en a 8 avec la distance d_{inf} .

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

$$H(x) = \{P : I(P) = x\}$$

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 3).

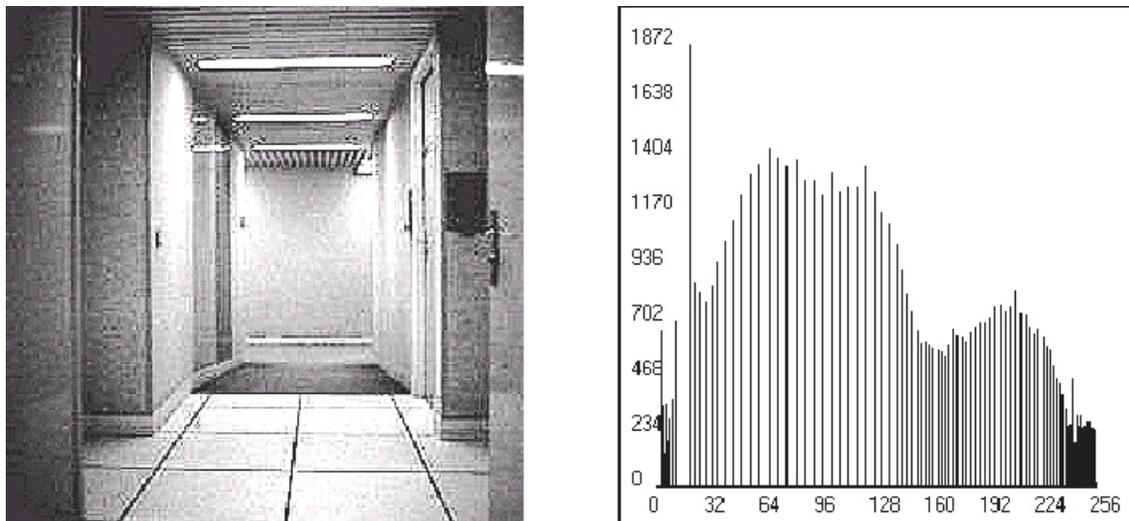


Figure 3. Une image et son histogramme représentatif

- **Bruit** : Dans de nombreux domaines scientifiques, le bruit joue un rôle fondamental. Il est à l'origine d'un grand nombre de difficultés lorsqu'on désire analyser un quelconque signal. L'image ne fait pas exception à cette règle. Il existe de nombreuses sources de dégradations dont on peut citer :

- Bruit lié au contexte de l'acquisition.
- Bruit lié au capteur.

- Bruit lié à l'échantillonnage.
- Bruit lié à la nature de la scène.

1.3 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.

1.3.1 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 caméra 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. Un capteur image est caractérisé par :

- Son rapport signal sur bruit, sa sensibilité et sa sensibilité spectrale.
- Une résolution maximale.
- Un temps d'intégration.
- Un seuil de saturation.
- Sa rémanence.

Il existe plusieurs types de capteur image tel que :

- Capteur à tube.
- Capteur à CCD (charge coupled device)

Ou encore les caméras infra rouge et les caméras à barrettes CCD.

1.3.2 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.

1.3.2.1 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 : Shannon-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, les ondelettes et les fractales.

1.3.2.2 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 1^{er} 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 définie comme suit :

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

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

f est une application de G_I dans G_F telle que :

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

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 4).

Exemple de recadrage dynamique de l'histogramme

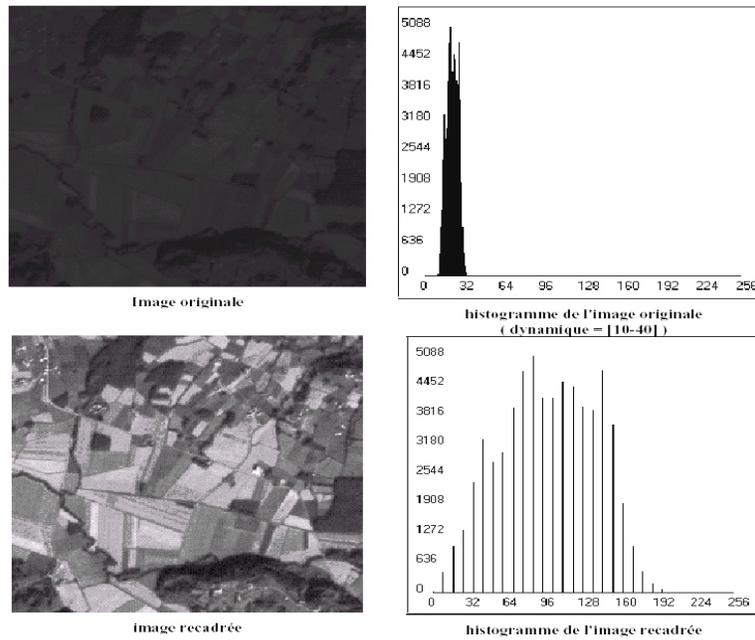


Figure 4. Exemple de recadrage dynamique

Mise en évidence de zones de l'histogramme

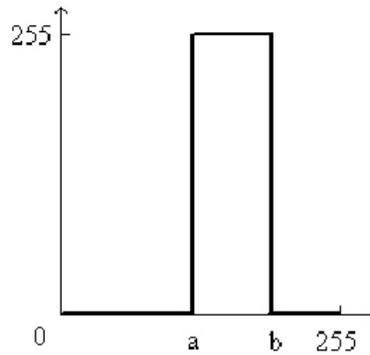


Figure 5. LUT de binarisation

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

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.

La figure 6 montre une image peu contrastée et la même image après égalisation.

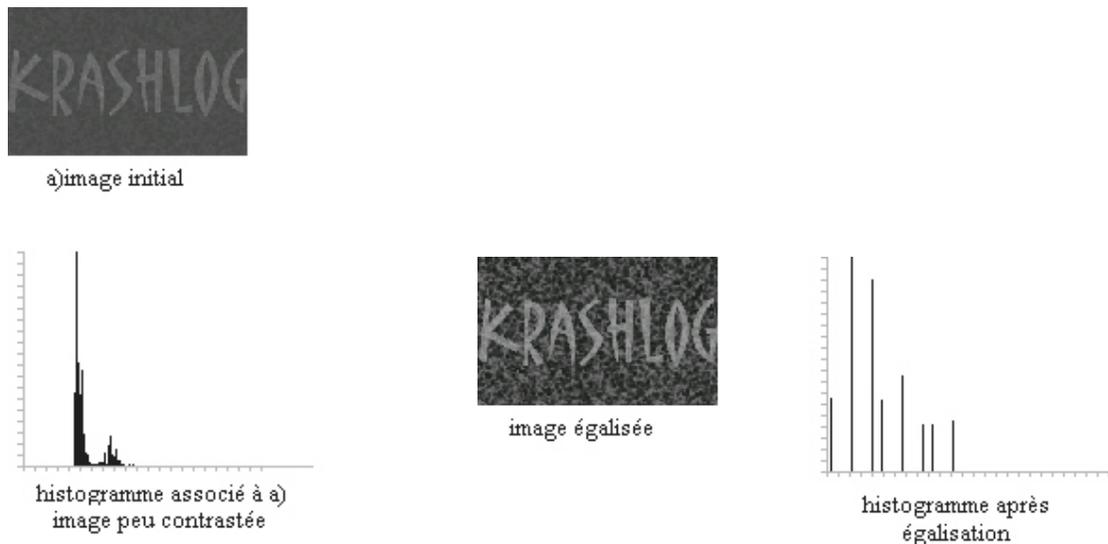


Figure 6. Egalisation d'histogramme

1.3.2.3.2 Le filtrage

Nous avons vu jusqu'à présent des techniques qui visent 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.

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 :

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

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 : l'image de départ.

I_F : l'image résultat.

On a :

$$I_F(x,y) = 1/9 \sum_{i=-1}^1 \sum_{j=-1}^1 H(i,j) \bullet I(x+i,y+j)$$

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 :

$$I_F(x,y) = 1/k \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} H(i,j) \bullet I(x+i,y+j)$$

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 :

$$I_F = H \otimes I$$

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

$$\mathfrak{F}(H \otimes I) = \mathfrak{F}(H) \bullet \mathfrak{F}(I)$$

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.

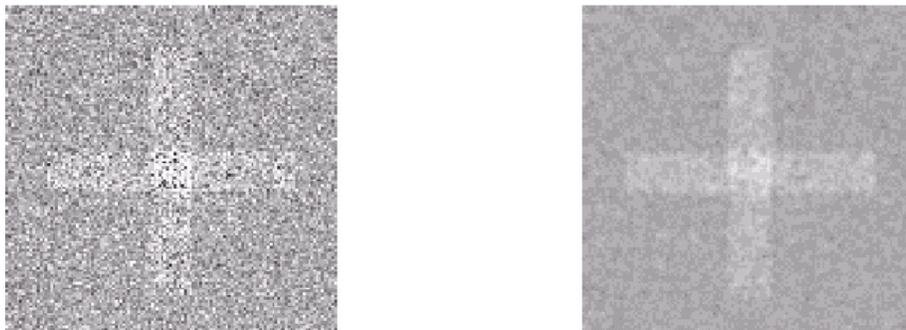


Figure 7. Filtrage linéaire d'une image ; à gauche l'image initiale et à droite l'image filtrée en utilisant un filtre moyenne 5x5

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 par un filtre $N \times N$, soit par une succession de deux convolutions par des filtres unidimensionnels de taille N .



Figure 8. Filtre gaussien

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.

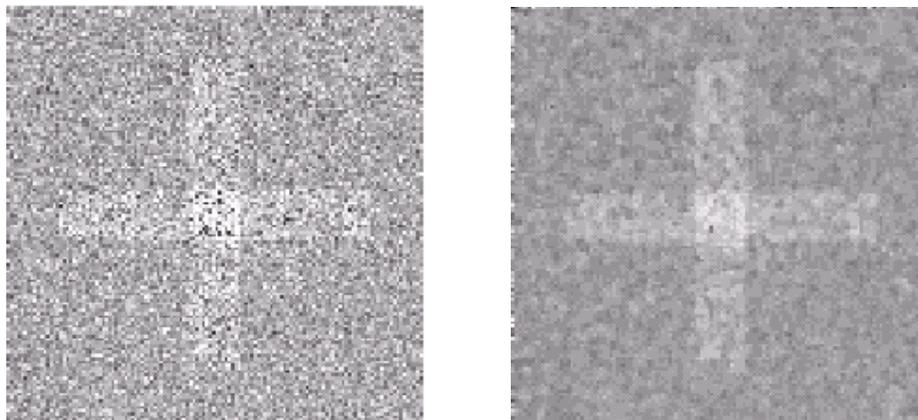


Figure 9. Filtrage non linéaire d'une image. A gauche l'image initiale et à droite l'image filtrée en utilisant un filtre médian 5×5

1.3.3 L'analyse

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

A partir de cette segmentation, des techniques de description / modélisation 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 :

1.3.3.1 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 tel que ceux ci :

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

$$H_x = [-1 \ 1]$$

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

1



a) image initiale



opérateur Roberts

Figure 10. Opérateur de Roberts

Opérateur de Sobel : utilise comme masque :

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

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

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

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

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



opérateur Sobel

Figure 11. Opérateur de Sobel sur la précédente image

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

Il utilise le masque suivant :

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



laplacien

Figure 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.

1.3.3.2 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.

1.3.4 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.

Introduction à la réalité augmentée

1.5 Introduction

Les films hollywoodiens ont toujours combiné des images virtuelles numériques générées par ordinateur avec des scènes réelles, et le résultat est tellement réaliste qu'il est parfois difficile de faire la distinction entre le réel et le virtuel. De l'interaction synchronisée entre l'homme et le dinosaure dans Jurassic Park, en passant par la reconstitution du Titanic et jusqu'au monde imaginaire du maître des anneaux, les artistes des effets spéciaux paraissent maîtriser de façon presque parfaite l'art de combiner des objets virtuels avec des scènes vidéo réelles.

Malgré que les résultats soient fantastiques, le processus par lequel l'artiste intègre un objet 3D dans le monde réel est extrêmement long et délicat. Même pour dix secondes d'un clip, une douzaine d'artistes et d'animateurs peuvent investir des heures par image dans leurs tentatives pour augmenter d'une façon parfaite les objets virtuels dans la séquence vidéo. La question qui se pose est : est ce que cette fusion d'objets 3D avec le monde réel peut être faite d'une manière interactive de telle façon que nous puissions voir et manipuler ces objets en temps réel ? La réalité augmentée est une telle approche pour accomplir ce but.

Contrairement à la réalité virtuelle, qui émerge l'utilisateur dans un environnement artificiel généré par ordinateur, la réalité augmentée est une technologie qui tente d'améliorer la vue de l'utilisateur du monde réel par insertion d'objets virtuels, tel que du texte, des images 2D ou des modèles 3D.

La motivation principale derrière la réalité augmentée est que les compétences visuelles de l'utilisateur peuvent être influencées dans le but d'interagir avec des objets générés par ordinateur ou recevoir des informations supplémentaires sur les objets du monde réel. Par exemple, la figure 13a montre la vue originale d'un ingénieur sur une cité de l'autre côté du fleuve, alors que la figure 13b montre la même vue améliorée en utilisant la technologie de réalité augmentée. Le résultat est tel que l'ingénieur est maintenant capable de prévoir le design du nouveau pont dans son milieu naturel, sans avoir à le construire.

Cependant ils existent différents aspects à prendre en considération dans la création d'applications réalité augmentée. L'un de ces aspects qui est le plus difficile est le calcul du point de vue de l'utilisateur en temps réel afin d'aligner exactement les objets virtuels avec les objets du monde réel [3]. Par exemple, pour modifier automatiquement la scène dans la figure 13a, l'ordinateur doit avoir quelques informations sur l'emplacement et l'orientation de l'environnement, à partir de la perspective de l'utilisateur, avant d'augmenter la scène avec le nouveau pont. En outre, si l'utilisateur veut voir le nouveau pont à partir d'un angle différent ou d'un autre point de vue, le système doit être capable de mettre à jour l'augmentation en temps réel, afin que le pont apparaisse comme une part naturelle de l'environnement. Cet alignement et synchronisation des objets virtuels avec le monde réel sont appelés : processus de recalage.

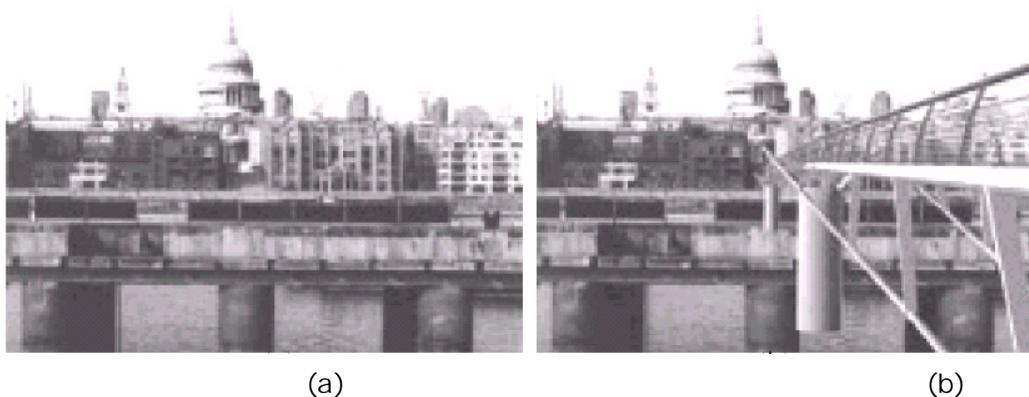


Figure 13. Exemple de la réalité augmentée pour la construction de pont

(a) *La scène originale montrant l'emplacement du pont proposé.*

(b) La scène a été améliorée avec un modèle de la réalité augmentée du pont proposé, donnant ainsi à l'ingénieur un aperçu du pont avant de le construire physiquement

1.6 Motivation et domaines d'application

De nos jours, les ordinateurs deviennent de plus en plus petits et puissants, et la consommation d'appareils électroniques est en augmentation, donc la réalité augmentée a un grand potentiel pour trouver son chemin vers le grand public. Imaginez que vous visitez une ville étrangère pour la première fois, et que vous n'avez aucune idée sur l'endroit dans lequel vous vous trouvez, ou où est ce que vous devez vous rendre. Au lieu de consulter votre dictionnaire pour comprendre ce qui est écrit dans le guide en langue locale, vous mettez votre paire de lunettes et immédiatement votre environnement n'est plus tellement étranger. Avec le système de réalité augmentée, vos lunettes ont converti tous les signes du monde réel en votre langue. Même si vous bouger ou tourner la tête, les signes traduits maintiennent leurs positions et orientations correctes, des flèches et des messages supplémentaires vous guident vers la destination désirée.

Ou encore, considérons qu'un étudiant en médecine s'entraîne pour devenir chirurgien. Au lieu simplement de lire des livres et s'entraîner à travers des vidéos, l'étudiant peut appliquer ses connaissances dans une simulation de chirurgie en utilisant la réalité augmentée. L'opération entière peut être simulée du début à la fin dans une salle d'opération réelle en utilisant des images du patient générées par ordinateur.

Ces deux situations décrites ci dessus paraissent venir directement d'un scénario de film de science fiction, mais elles ne sont pas nécessairement loin d'être faites. La clé pour la création d'une expérience de réalité augmentée effective est d'imiter le monde réel aussi fidèlement que possible. En d'autres mots, à partir d'une interface utilisateur, l'utilisateur n'est pas obligé d'apprendre à utiliser le système de réalité augmentée mais il doit être capable de l'utiliser immédiatement en utilisant son expérience passée dans le monde réel. En clair, l'aspect visuel de la réalité augmentée est un composant critique dans la représentation de l'environnement, donc le processus de recalage joue un rôle central.

Considérons l'exemple de l'application de la réalité augmentée dans l'entraînement pour la chirurgie. Si le processus de recalage est inexact, l'étudiant va développer des compétences qui seront effectives dans l'environnement de la réalité augmentée, mais pas nécessairement dans la salle des opérations du monde réel. Similairement, pour l'exemple de lunette de translation des signes, un recalage inexact va causer le décalage des signes ou bannières augmentées de leurs positions attendues, résultant ainsi un sévère mal de tête chez certains utilisateurs.

Les chercheurs en vision artificielle ont expérimenté pour quelques années des solutions pour le problème de recalage. Une de ces solutions qui est très prometteuse est le suivi de modèles (pattern) bien connu qui comprend des primitives sur une surface plane à travers les images de la séquence vidéo, et utiliser les positions extraites de ces primitives pour accomplir l'augmentation des objets virtuels dans la séquence [4]. La figure 14 montre un exemple sur l'augmentation basée sur le suivi d'un modèle planaire. En traquant ces modèles prédéterminés, les chercheurs espèrent gagner assez d'expérience dans le but ultime de traquer des primitives naturelles dans des environnements arbitraires.



Figure 14. Exemple de la réalité augmentée basée suivi de modèle [5]

- (a) La scène originale, montrant un papier contenant un modèle prédéfini
- (b) La même scène après l'augmentation d'un objet virtuel sur un modèle prédéfini

Pour compliquer encore plus la situation, le processus de recalage doit être assez rapide pour que l'augmentation des objets virtuels puisse être faite en temps réel avec le moindre coût en matériel. En clair, si on veut que les applications de réalité augmentée soient acceptées par les marchés de consommation, on doit premièrement trouver des solutions efficaces pour le problème de recalage en temps réel.

Les applications potentielles de la RA sont multiples : nous trouvons en particulier des applications en temps réel dans les domaines de la médecine, de la maintenance d'objets manufacturés, de l'industrie, du design intérieur ou du jeu. Les applications de post-production sont principalement les effets spéciaux pour le cinéma ou la publicité, mais aussi les études d'impact ou d'éclairage en architecture. Nous développons à présent chacun de ces aspects.

1.6.1 Médecine

La réalité augmentée peut-être utilisée par les médecins pour visualiser des données 3-D extraites chez un patient par-dessus le corps du patient (images ultrasonores, tomographie 3-D, images à résonance magnétique etc.). A l'aide d'un HMD (voir la section 2.3.1.2), le médecin peut par exemple observer l'image ultrasonore d'un **foetus** à l'intérieur du ventre de la maman (figure 15). Dans l'article **[5]**, on expose comment la réalité augmentée peut servir la chirurgie digestive, et dans le même ordre d'idées, l'article **[6]** présente une méthode de guidage de ponctions percutanée à l'aide d'un système de vision.

Plus généralement, de nombreux travaux visent à fusionner automatiquement diverses sources d'images 3-D et 2-D : recalage d'une image tomographique ou à résonance magnétique dans une séquence d'images à rayons X ou encore recalage de reconstructions 3-D de la vascularisation cérébrale dans des images d'angiographie numérique soustraite. Ainsi, en neuroradiologie interventionnelle par exemple, les travaux ont pour but de permettre au radiologue de savoir à tout instant où se trouve son cathéter dans le corps du patient.



Figure 15. *Visualisation du fœtus à l'intérieur du ventre de la patiente* [7]

1.6.2 Assistance en milieu industriel

En milieu industriel, la réalité augmentée permet de simplifier considérablement certaines tâches d'assemblage, de maintenance ou de réparation. Ainsi, des chercheurs de Boeing ont développé un système permettant de guider les techniciens dans l'assemblage de réseaux électriques pour les avions. Avant que ce système ne soit mis en place, les techniciens plaçaient les fils selon des schémas gravés sur des panneaux. Un 747 comportant plus de 1000 réseaux électriques, et les réseaux étant différents d'un avion à l'autre, cela impliquait des coûts considérables pour le stockage, le transport et la construction des panneaux. Avec le nouveau système, tous les réseaux sont stockés en mémoire. Il suffit au technicien de choisir le réseau approprié, qu'il peut alors visualiser par-dessus un panneau vierge en s'équipant d'un HMD (voir la section 2.3.1.2). Le même principe est mis en œuvre dans **[8]** pour l'assemblage d'un mécanisme de fermeture de porte pour automobile. Des indications virtuelles peuvent aussi être ajoutées pour désigner certaines pièces d'objets manufacturés, comme des

photocopieuses ou des moteurs, et faciliter ainsi les opérations de maintenance ou de réparation sur ces objets.

1.6.3 Design intérieur

Un autre domaine d'application de la réalité augmentée est le design intérieur : le designer dispose d'une base de données de modèles de meubles ou d'éléments décoratifs, qu'il peut positionner et visualiser dans la pièce physique à meubler. Ceci peut se faire par le biais d'une interface graphique **[9]** (figure 16) ou d'un HMD **[10]** : le designer peut alors se déplacer dans la pièce et visualiser en temps réel les différents éléments ajoutés.

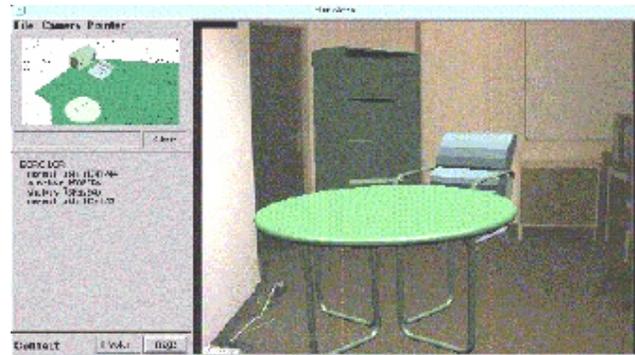


Figure 16. Une Interface permettant de positionner des meubles virtuels dans une salle réelle

1.6.4 Jeux

La Réalité Augmentée commence à se développer dans le domaine du jeu. Stricker et al. ont par exemple conçu un jeu de Tic Tac Toe interactif en 3-D **[11]**. Le joueur, portant un HMD optique (voir la section 2.3.1.2) muni d'une caméra, place son pion sur la grille et appuie sur un bouton "GO" virtuel. L'ordinateur analyse alors la scène, localise le pion (grâce à sa couleur), place sa propre croix virtuelle et demande au joueur de poursuivre le jeu par l'intermédiaire d'un écran virtuel (voir figure 17). Un jeu de Hockey sur table augmenté a aussi été proposé par Satoh et al. **[10]** : les joueurs frappent un palet virtuel avec des maillets physiques sur un terrain de jeu physique.



Figure 17. La réalité augmentée appliquée au jeu du TIC TAC TOC [11]

1.6.5 Effets spéciaux

Les effets spéciaux pour le cinéma, la publicité ou les clips vidéo intègrent de plus en plus souvent des images de synthèse qui viennent s'intégrer au film : on peut notamment citer quelques réalisations célèbres, comme le film Titanic où éléments réels et virtuels se côtoient sans même que le spectateur ne s'en rende compte, ou encore l'édition spéciale de la trilogie Star Wars, qui fait apparaître des créatures virtuelles dans un décor réel (figure 18). Pour ce film, les compositions ont été réalisées manuellement. Il y a plusieurs étapes à prendre en compte pour l'incrustation semi-automatique d'acteurs virtuels animés dans un environnement réel : extraction des paramètres de la caméra, création et animation des acteurs et rendu des images finales. En particulier, le rendu tient compte des objets réels cachés par les acteurs virtuels et vice versa, des collisions entre les acteurs virtuels et l'environnement réel et des ombres des acteurs virtuels sur le monde réel.

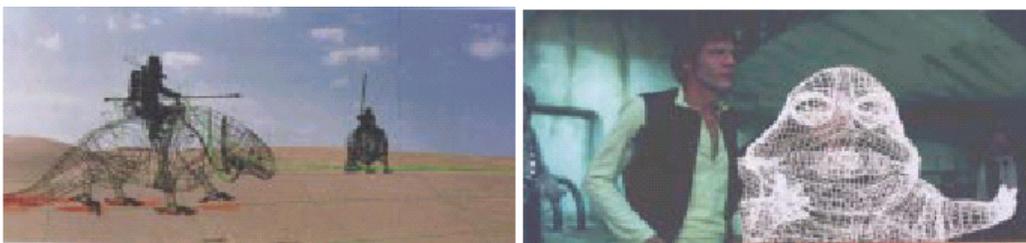


Figure 18. Certains effets spéciaux de la trilogie Star Wars utilisent des images de synthèse superposées aux images réelles

1.6.6 Etude d'impact

La réalité augmentée est aussi utilisée en architecture pour les études d'impact **[12]** ou l'illumination artificielle de monuments **[13]**. En particulier, le projet des ponts de Paris, vise à illuminer virtuellement un certain nombre de ponts situés autour de l'île de la Cité. L'objectif est d'évaluer visuellement l'impact sur l'environnement de plusieurs projets d'illumination. Plutôt que d'effectuer des

tests sur le terrain, il est plus pratique et moins coûteux d'utiliser des images de synthèse. Pour cela, des séquences vidéo ont été utilisées, dans lesquelles un pont réel a été remplacé par un pont virtuel, illuminé selon le système d'éclairage que nous souhaitons évaluer (figure 19).

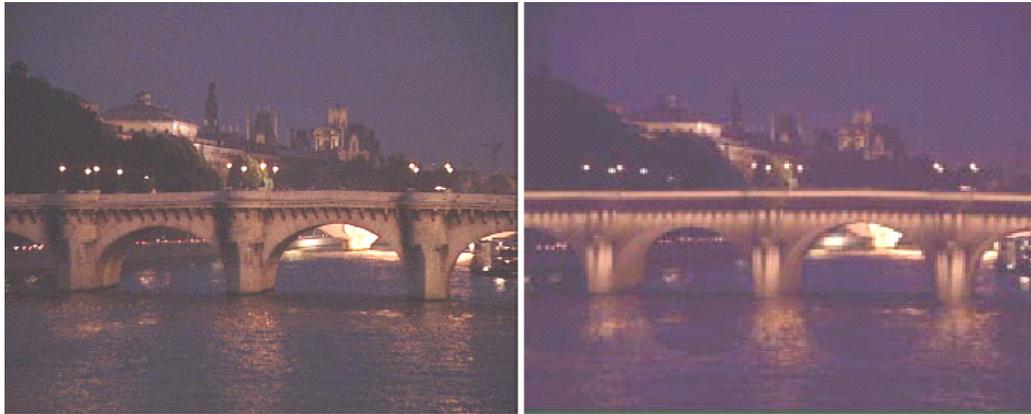


Figure 19. *Exemple d'illumination artificielle d'un pont. A gauche l'image réelle, à droite l'image augmentée*

1.7 Les systèmes de réalité augmentée

Un système de réalité augmentée typique se compose d'une camera, d'un appareil d'affichage et éventuellement d'une interface utilisateur qui permet à l'utilisateur d'interagir avec les objets virtuels.

1.7.1 Caméra et technologies d'affichages

Dans le but de combiner le monde réel avec des objets virtuels en temps réel nous devons utiliser une configuration se composant d'une camera et dispositif d'affichage. Il existe principalement trois configurations utilisées dans la réalité augmentée qui sont les configurations basées Moniteur, les configurations vidéos vue à travers et les configurations optique vue à travers [7][14].

1.7.1.1 Configurations basées Moniteur

C'est la plus simple approche, elle est représentée par la figure 20. La camera vidéo capture en continu des images du monde réel et les achemine au système d'augmentation. Les objets virtuels sont ensuite insérés et alignés dans chaque image les unes après les autres, et finalement ces images augmentées sont affichées sur un moniteur standard. L'avantage de cette méthode est sa simplicité et son accessibilité, car elle requiert un PC et une caméra vidéo USB. En plus, traiter chaque image individuellement, permet au système d'augmentation d'utiliser des approches basées vision pour extraire des informations associées à l'utilisateur (position et orientation), et ce pour accomplir le processus de recalage. Cependant cette simplicité est acquise au détriment du réalisme de la scène du monde. Il est clair que voir le monde réel à travers un petit moniteur, limite le réalisme et la mobilité du monde augmenté. En plus, chaque image capturée avec la camera doit être traitée individuellement par le système d'augmentation, donc il y a un délai potentiel entre le moment ou

l'image a été capturée par la camera et le moment ou cette même image augmentée est affichée sur le moniteur. Finalement, la qualité de l'image est limitée par la résolution de la camera.

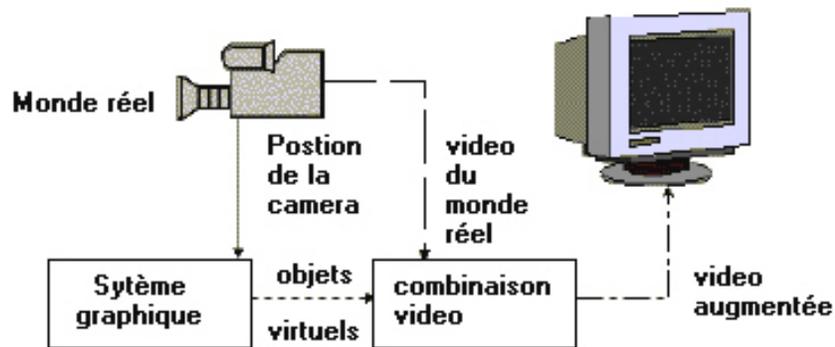


Figure 20. Configuration basée moniteur [14]

1.7.1.2 Configurations basées vidéo vue à travers

Dans le but d'augmenter le sens d'immersion dans les systèmes de réalité virtuelle, les HMDs (Head Mounted Display) qui couvrent complètement la vue de l'utilisateur sont utilisés (Voir figure 21). La figure 22 montre un schéma représentant un système de réalité augmentée basé vidéo vue à travers. Dans cette configuration, l'utilisateur ne voit pas directement le monde réel, mais il voit ce que le système affiche dans des moniteurs minuscules à l'intérieur du HMD.



Figure 21. HMD vidéo vue à travers [14]

La différence entre les HMDs utilisés dans la réalité virtuelle et ceux de la réalité augmentée est l'ajout sur ces derniers de caméra vidéo pour capturer des images du monde réel. Cependant cette configuration est presque identique à celle basée moniteur en termes de fonctionnalités, mais l'utilisation d'une paire de caméra vidéo stéréo (deux caméras) permet de fournir à chaque œil une image différente, ce qui augmente le réalisme et l'immersion que peut fournir la scène augmentée. Comme la configuration basée moniteur, cette configuration est victime de décalages visuels dus à la capture, au traitement et à

l'augmentation de chaque image vidéo. En outre, une grande distance entre les caméras et les yeux de l'utilisateur peut réduire le sens de l'immersion. Des capteurs de position sont placés dans l'HMD pour fournir des informations sur la position des caméras afin d'accomplir le processus de recalage.

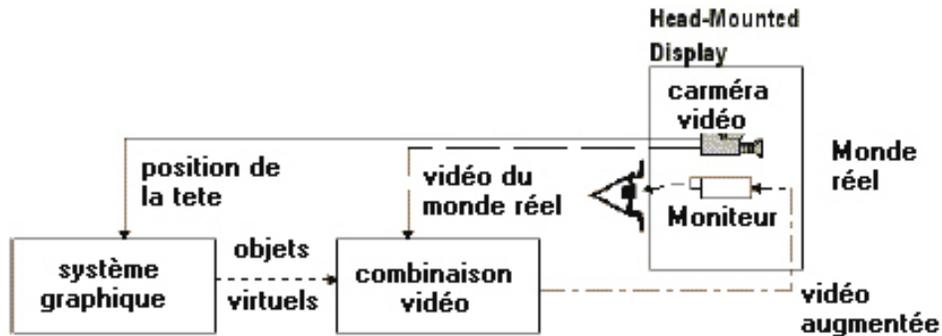


Figure 22. Système de réalité augmentée basé vidéo vue à travers [14]

1.7.1.3 Configuration optique vue à travers

Il existe un autre type d'HMD (Voir figure 23) utilisé pour la réalité augmentée qui est un système d'affichage optique vue à travers représenté par la figure 24. Dans cette configuration l'utilisation de verres semi transparents permet d'une part de voir directement le monde réel et d'autre part de réfléchir les objets virtuels générés par le système d'augmentation.

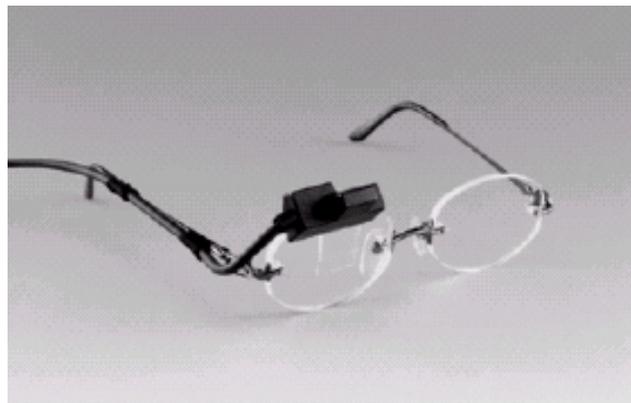


Figure 23. HMD optique vue à travers [15]

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

difficile car le monde réel paraîtra naturel alors que les objets virtuels paraîtront pixélisés. Un autre inconvénient majeur de cette configuration est l'absence de captures d'images du monde réel qui fournissent des informations sur l'environnement, seuls les capteurs de positions placés dans l'HMD fournissent des informations pour accomplir le processus de recalage.

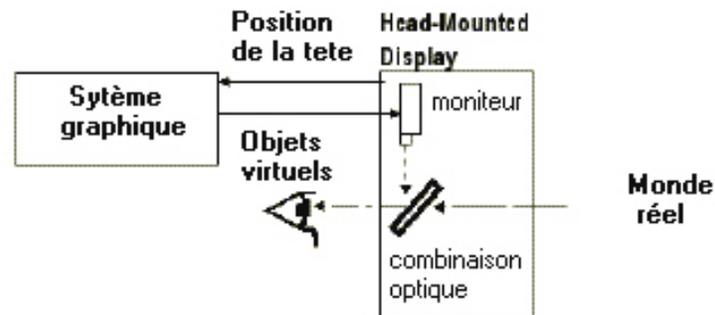


Figure 24. Système de réalité augmentée basé optique vue à travers [14]

1.7.1.4 Comparaison entre la méthode vidéo et la méthode 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 combineurs 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 camera. 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 combineurs 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.

1.7.2 Appareils permettant l'interaction à l'utilisateur

Pareillement à la réalité virtuelle, la réalité augmentée requiert quelques appareils à travers lesquels l'utilisateur peut interagir physiquement avec l'objet virtuel. Ces appareils permettent, non seulement l'interaction avec les objets virtuels mais procurent aussi des sensations comme le toucher.

Considérons une tasse de café qui a été augmentée sur un bureau réel d'un utilisateur. Avec des gants spéciaux, l'utilisateur peut physiquement prendre la tasse de café et la tenir dans sa main. En utilisant des capteurs dans la paume et au bout des doigts, l'utilisateur peut sentir le poids de la tasse aussi bien qu'il sent les textures autour de la surface de la tasse. Plus loin encore, l'environnement augmenté doit simuler la gravité si bien que si l'utilisateur laisse tomber la tasse, celle-ci tombera et se cassera. Par conséquent, si la simulation est correcte, l'utilisateur sera incapable de distinguer la tasse virtuelle de celle réelle.

Cela paraît conceptuellement simple, mais ces appareils sont toujours sujets à d'intensives recherches car la recreation réaliste de sensations physiques reste un défi. De nombreux appareils ont été proposés, comme les gants mentionnés précédemment. Un exposé détaillé sur ces différents appareils qui sont utilisés pour la réalité augmentée est fourni dans [16].

1.8 Les approches de recalage

Il existe principalement trois approches pour accomplir le processus de recalage : les approches basées capteurs et les approches basées vision. Dans ce qui suit nous allons expliquer brièvement chacune des trois approches.

1.8.1 L'approche basée capteurs

Dans le passé, la majorité des systèmes de réalité augmentée utilisaient des capteurs (magnétiques, mécanique ou optiques) pour mesurer la position de la caméra. Notons que la grande précision qu'exige le processus de recalage a fait qu'aucun capteur seul ne suffit pour obtenir des résultats satisfaisants pour la plupart des applications de réalité augmentée [14].

Les types de capteurs les plus utilisés sont :

Capteurs mécaniques

Un système mécanique permet de détecter les mouvements de l'utilisateur. Cela permet d'obtenir une très bonne précision, mais cette solution est onéreuse, surtout qu'elle restreint considérablement les déplacements de l'utilisateur qui est physiquement lié au système.

Capteurs magnétiques

Des récepteurs miniatures placés au niveau du HMD détectent les champs magnétiques émis par un transmetteur. Cette solution est robuste et impose moins de contraintes sur les déplacements que les capteurs mécaniques (l'utilisateur doit toutefois rester à l'intérieur du champ magnétique). Malheureusement, la précision est relativement faible et le champ magnétique peut être perturbé par des objets métalliques.

Capteurs optiques

Des marqueurs réfléchissants ou des diodes luminescentes sont filmés par une caméra. Un algorithme reconnaît les marqueurs et calcule le point de vue de la caméra à partir de leurs positions dans l'image. Cette solution utilise une technique basée vision, elle est très rapide puisque les marqueurs sont facilement identifiables. Elle offre une bonne précision et impose peu de contraintes sur les déplacements de l'utilisateur (la seule contrainte est que les marqueurs ne sortent pas du champ de vision de la caméra).

Ces techniques nécessitent cependant l'emploi de matériel adéquat et ne sont pas adaptées aux scènes d'extérieur.

1.8.2 L'approche basée vision

Les méthodes basées vision utilisent les images capturées par le système d'acquisition pour retrouver les paramètres de la caméra. La calibration forte (que nous appellerons aussi calibration basée modèle) s'aide de la connaissance d'indices 3-D de la scène qui peuvent être des marqueurs bien visibles placés par un opérateur (Voir figure 25) ou bien des éléments naturels de la scène. La technique d'autocalibration (que nous appellerons aussi calibration basée images) n'utilise par contre que des indices extraits de deux ou plusieurs images.

Certains systèmes peuvent utiliser une approche hybride : intégrant par exemple à la fois des capteurs magnétiques et des marqueurs positionnés dans la scène. Ceci leur permet d'obtenir la précision liée à l'utilisation de marqueurs en même temps que la robustesse offerte par les capteurs.

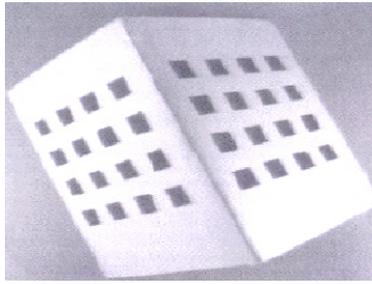


Figure 25. Modèle de calibration de la caméra

1.9 Les erreurs de recalage

Tant que les systèmes de réalité augmentée sont sujets à des erreurs de recalage provenant de plusieurs sources, cela aura pour cause d'affecter la performance et l'immersion de l'environnement augmenté. Il existe principalement deux catégories d'erreurs de recalage qui sont statiques et dynamiques.

1.9.1 Les erreurs statiques

Les erreurs statiques du recalage apparaissent quand aucun objet n'est en mouvement dans la scène. Un exemple d'une erreur statique est un objet qui apparaît dans des emplacements différents quand il est vu à partir de plusieurs points de vue.

1.9.2 Les erreurs dynamiques

Les erreurs dynamiques se produisent quand l'utilisateur ou un objet est en mouvement, ce qui produira un léger changement sur les entités augmentées. En d'autres mots les objets virtuels peuvent éventuellement ne pas être synchronisés avec le monde réel.

La plupart des erreurs dynamiques sont dues au temps de calcul. Le système doit capturer une image de la scène réelle, déterminer la position de la caméra, transformer les coordonnées de l'objet dans l'espace de la caméra, et enfin incruster l'objet dans la scène. Si on suppose que l'image a été capturée en temps t et que la durée du calcul est s , l'utilisateur ne verra pas le résultat de l'augmentation qu'après $t+s$. Si $s > 50\text{ms}$ et dans ce cas l'immersion dans l'environnement augmenté sera compromise.

1.10 Problématique

La réalité augmentée est une discipline relativement récente, dont la majorité des avancées ont eu lieu ces dix dernières années. Les systèmes de composition d'images en phase de post-production sont aujourd'hui utilisés aussi bien par les producteurs d'effets spéciaux que par le grand public, qui peut se procurer plusieurs logiciels commerciaux (le degré d'interactivité et la complexité des applications prises en compte étant variables d'un produit à l'autre).

Les applications à base de HMD sont par contre plus expérimentales que réellement utilitaires. Si des compagnies aériennes telles que Boeing et McDonnell Douglas utilisent ou s'intéressent de très près à cette technologie, celle-ci n'est pas encore suffisamment bon marché, ni suffisamment flexible et confortable visuellement pour être appliquée dans l'industrie. En médecine, la fusion multi modalités est utilisée, mais dans un contexte de diagnostique ou préopératoire. Les applications médicales utilisant un HMD sont encore anecdotiques et de nombreux problèmes restent à résoudre avant qu'un acte chirurgical puisse être réalisé à l'aide d'un casque de réalité augmentée.

En particulier, la visualisation simultanée de données réelles et virtuelles au travers d'un HMD génère des perturbations physiologiques qui doivent être prises en compte pour une utilisation prolongée de cet outil. D'autre part, un délais de l'ordre de 100 ms est inévitable entre l'instant où l'utilisateur perçoit la scène et l'instant où l'image virtuelle est projetée [7]. Si l'utilisateur fait bouger la tête entre ces deux instants, l'objet virtuel est décalé par rapport aux objets réels.

En dehors de ces problèmes techniques et physiologiques, nous considérons à présent les problèmes géométriques et photométriques liés aux incrustations.

1.10.1 Cohérence géométrique

Les objets virtuels doivent être projetés de façon cohérente dans les images réelles. Dans la pratique, les objets sont positionnés dans un repère associé à la scène, et l'objectif est de déterminer le point de vue de la caméra (position et orientation) dans ce repère. En plus de ces paramètres extrinsèques, nous devons connaître les caractéristiques intrinsèques de la caméra (comme la distance focale par exemple), qui définissent la projection de l'objet virtuel dans le plan image de la caméra.

Le point de vue de la caméra peut être retrouvé par exemple, grâce à des émetteurs magnétiques placés au niveau de la caméra ou du HMD. Une autre méthode fréquemment employée consiste à localiser dans l'image un certain nombre d'indices (points, droites ou autre). Il sera question de rechercher le point de vue correspondant à la projection des indices sur le plan de la caméra.

En plus de positionner l'objet virtuel au bon endroit de l'image et selon la bonne orientation, il faut aussi prendre en compte les occultations entre objets réels et virtuels. En figure 26, nous pouvons voir un exemple d'occultation de l'objet virtuel par un objet réel de la scène (le bateau). Si nous nous contentons de projeter l'objet virtuel par-dessus l'image réelle, cette occultation ne sera pas prise en compte et il en résultera une incohérence géométrique.

1.10.2 Cohérence photométrique

Afin d'assurer la cohérence photométrique de la composition, il faut par ailleurs synthétiser les inter-réflexions de la lumière entre objets réels et virtuels, ainsi que les ombres portées. Par exemple, pour l'illumination artificielle du pont précédent, il a fallu générer le reflet du pont illuminé dans l'eau du fleuve (figure 26).

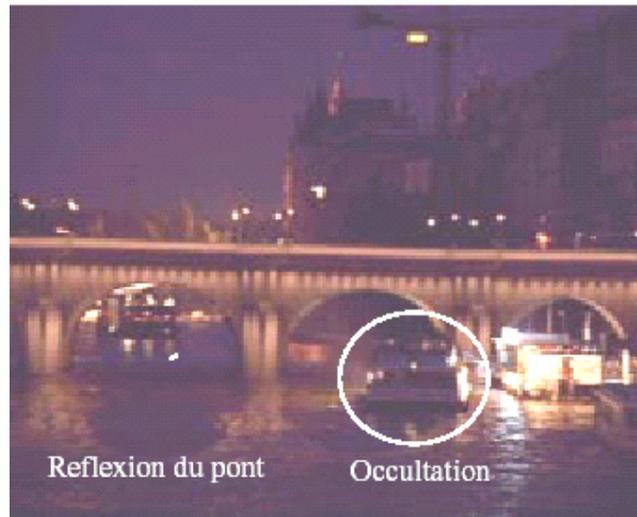


Figure 26. Cohérence géométrique et photométrique pour la séquence du pont

Conception d'un système de réalité augmentée

Dans ce chapitre nous allons décrire la démarche suivie pour la conception d'un système de réalité augmentée permettant d'insérer un objet virtuel 2D dans une séquence vidéo. Nous allons aussi détailler les différents modules qui composent ce système et les outils mathématiques utilisés pour accomplir l'incrustation et l'alignement de l'objet virtuel dans la scène réelle afin qu'il paraisse stable et réaliste aux yeux de l'utilisateur.

1.11 Problématique

Etant donné une séquence vidéo, il est question de trouver la transformation géométrique qui permet la projection de l'objet virtuel à un endroit précis sur une prise de vue donnée de la séquence.

L'outil mathématique qui permet la projection est naturellement une transformation géométrique. Les transformations géométriques diffèrent suivant l'ordre de complexité de la projection à effectuer, ceci aura un impact direct sur le nombre de paramètres nécessaires pour déterminer la transformation géométrique.

1.11.1 Le cas 2D

Puisqu'il s'agit, dans notre cas, de projeter librement des objets planaires (cas 2D) sur une surface plane appartenant à la scène réelle (voir figure 27), la projection projective est la plus adéquate.

La géométrie projective est caractérisée par une relation d'équivalence entre points, cette relation est définie comme suit :

Pour le cas d'un espace de 3 dimensions (espace géométrique), un point $X=(x, y, z, 1)$ est en relation d'équivalence avec un autre point $X'=(x', y', z', 1)$ s'il existe un paramètre λ tel que :

$$X'=\lambda X$$

Dans le cas de notre application qui consiste à insérer un objet virtuel 2D sur un modèle planaire bien défini dans une séquence vidéo, l'utilisation d'une transformation projective 2D-2D est nécessaire. Ce genre de transformations est appelé homographie planaire.

Le problème consiste donc à trouver l'ensemble des homographies H qui permettent de placer correctement l'objet virtuel 2D dans la séquence. Pour positionner l'objet virtuel 2D au niveau de chaque image de la séquence vidéo, il suffit de faire une projection perspective 2D-2D depuis l'objet virtuel 2D vers le modèle se trouvant sur l'image de la séquence.

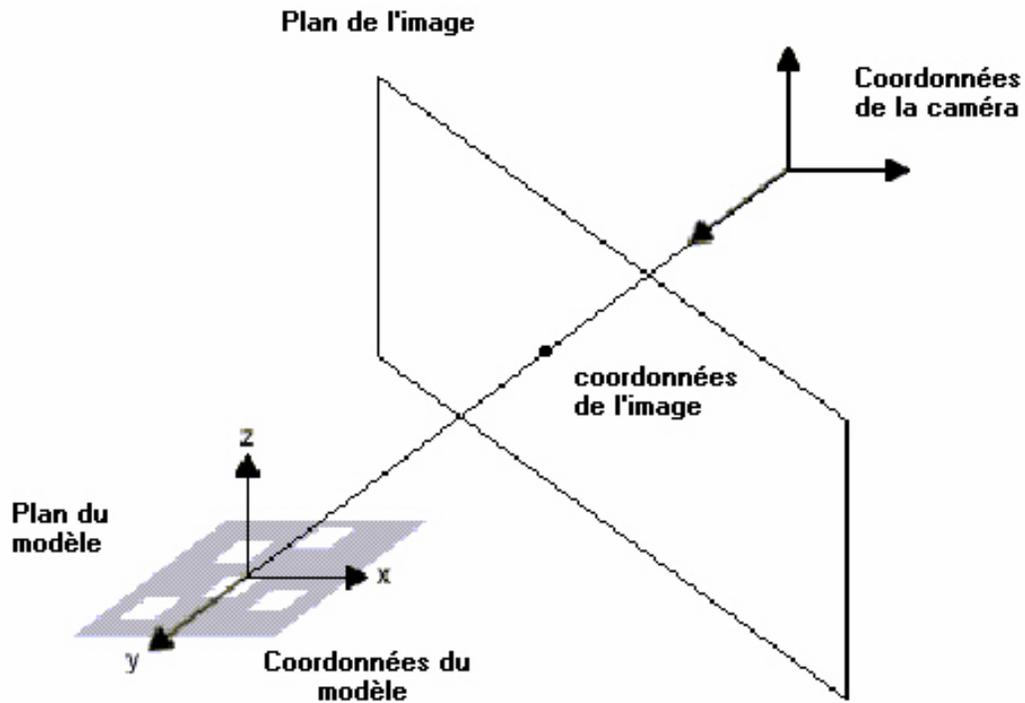


Figure 27. Systèmes de coordonnées de la caméra et du modèle

En d'autres termes, Si $[x \ y \ 1]^t$ représentent les coordonnées homogènes d'un point X de l'objet virtuel par rapport au repère objet virtuel, et $[x' \ y' \ 1]^t$ représentent les coordonnées homogènes de son correspondant X' par rapport au repère image, X est alors lié à X' (noté $X \leftrightarrow X'$) par la relation :

$$X' = H X \quad \text{avec} \quad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Où H est une matrice 3x3 représentant une homographie planaire.

Cela suppose que le plan du modèle est $z=0$ dans le système de coordonnées du monde réel. Chaque correspondance génère deux équations linéaires pour les éléments de H . Donc l'équation précédente nous donne :

$$\begin{aligned} x'(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

Ces deux équations peuvent s'écrire sous la forme suivante :

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{pmatrix} \mathbf{h} = 0$$

Où

$$\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T$$

est la matrice H écrite sous forme de vecteur.

Donc le problème revient à trouver les éléments de la matrice H afin de calculer pour un point X dans le monde réel son correspondant dans l'espace de l'image. La détermination des valeurs de la matrice H revient à résoudre un système d'équations.

Il faut au minimum quatre correspondances générant huit équations linéaires à résoudre pour calculer les éléments de h, afin de trouver la transformation géométrique qui relie le système de coordonnées du monde réel à celui du plan image.

Donc on a pour quatre correspondances :

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1'x_1 & -x_1'y_1 & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1'x_1 & -y_1'y_1 & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2'x_2 & -x_2'y_2 & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y_2'x_2 & -y_2'y_2 & -y_2' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3'x_3 & -x_3'y_3 & -x_3' \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y_3'x_3 & -y_3'y_3 & -y_3' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4'x_4 & -x_4'y_4 & -x_4' \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y_4'x_4 & -y_4'y_4 & -y_4' \end{pmatrix} \mathbf{h} = A\mathbf{h} = \mathbf{0}$$

En utilisant directement l'homographie cela procure un mécanisme pour l'augmentation des informations 2D sur le plan défini par le modèle se trouvant dans la séquence vidéo.

1.11.2 Extension pour le cas 3D

Dans le cas 3D on devra connaître la position précise où sera projeté l'objet virtuel suivant l'emplacement voulu dans le monde réel ; une correspondance entre les coordonnées 3D dans le monde réel et les coordonnées 2D sur l'image doit être faite.

Le traitement d'image issue de vidéo nécessite une phase essentielle de calibration de caméra. Celle ci permet de relier la géométrie du monde réel et celle de l'image.

1.11.2.1 Calibration

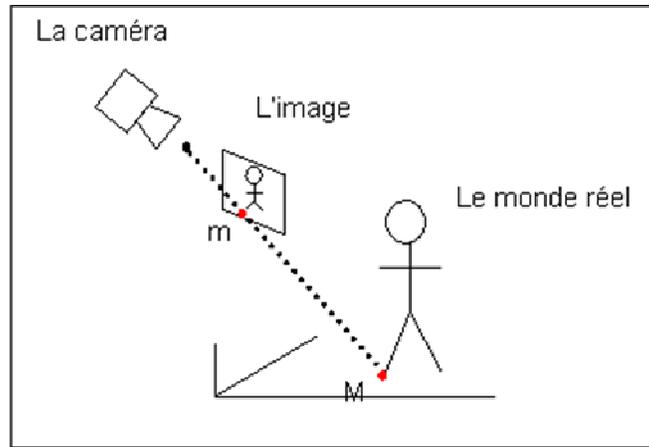


Figure 28. Relation entre le monde réel et le monde image

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 28). 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 ($f, C_x, C_y, s_x, \kappa_1$). 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.

Le passage des coordonnées 3D du monde réel aux coordonnées de la caméra est décomposable de la manière suivante :

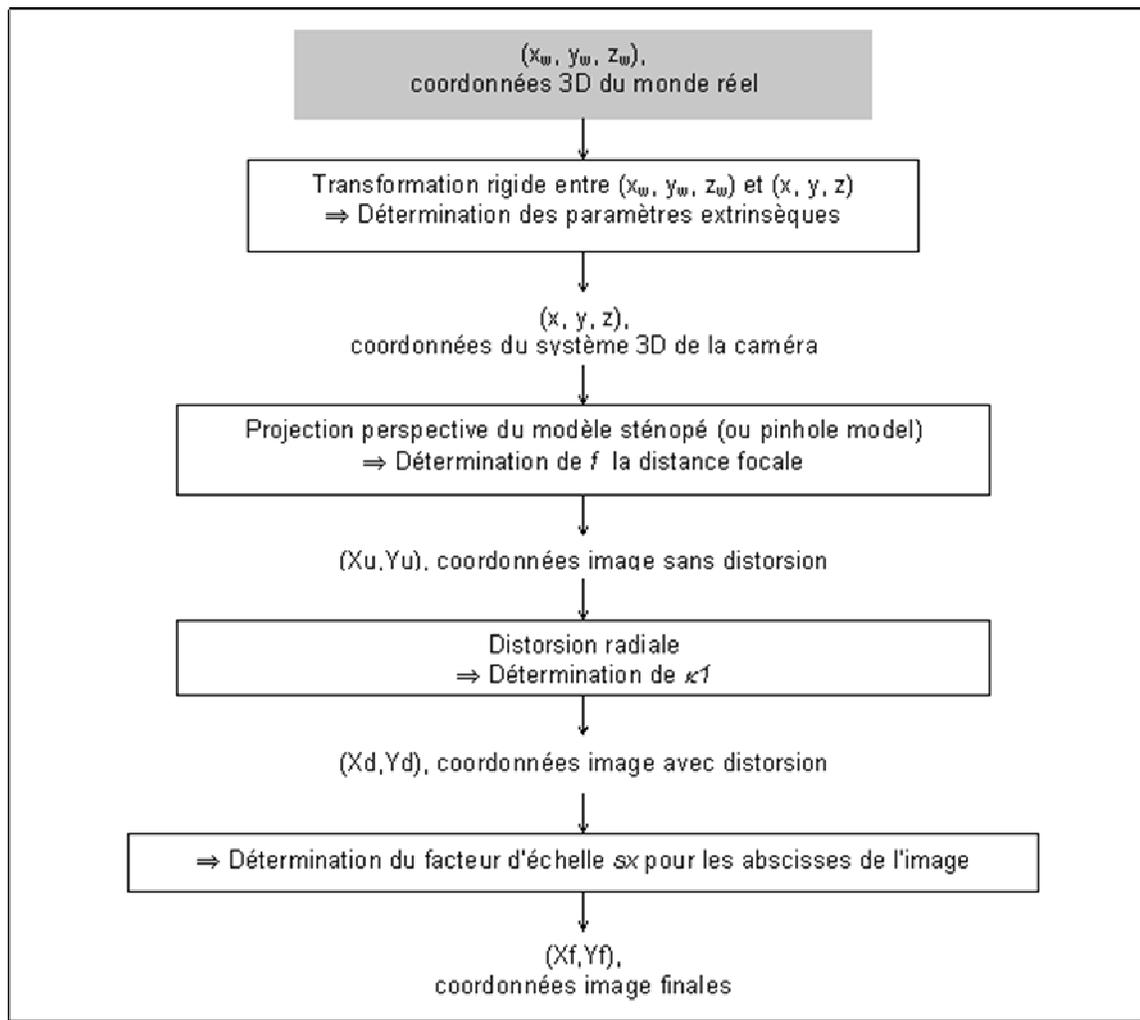


Figure 29. Décomposition du modèle de Tsai

Dans la pratique la plus simple des calibrations est la calibration manuelle.

Elle se décompose comme suit :

- positionnement dans le monde réel de repères remarquables.
- capture d'une image à partir de la caméra.
- identification de chacun des coins du quadrillage dans l'image, à l'aide d'un mini programme. (ce mini programme affiche l'image capturée et demande de cliquer sur les repères remarquables apparaissant sur l'image).

Toutefois, Dans une vidéo, il ne sera pas possible de donner la main à chaque fois à l'utilisateur pour cliquer sur les points nécessaires afin d'effectuer la calibration manuelle. Il faut donc suivre les points une fois trouvés, et ne plus les perdre.

Pour le cas 3D et dans le cas général la matrice H devient une matrice 4×3 et a donc 12 entrées. Puisque chaque correspondance nous donne deux équations, on aura besoin de six correspondances. Mais on peut réduire les paramètres à 4 même dans le cas 3D en utilisant la géométrie affine.

1.11.2.2 La géométrie affine

Par l'utilisation de la géométrie affine, on réduit le nombre de points nécessaires à la calibration à quatre points et leurs correspondants, au lieu de six. Ceci est dû aux simplifications supposées pour utiliser cette géométrie.

Un exemple des suppositions faites dans la géométrie affine est que deux lignes parallèles doivent rester parallèles après projection, ce qui n'est pas toujours le cas dans une projection camera- image.

Ces suppositions ne sont donc qu'approximatives, et le jitering (le fait que les objets incrustés bougent et ne sont pas stables) est dû dans une mesure à cette approximation.

D'autre part, la géométrie affine présente des avantages :

- Elle permet le calcul de la projection de n'importe quel point sans avoir besoin de connaître la position de la caméra ou de ses paramètres intrinsèques.
- La projection de n'importe quel point est calculée comme une combinaison linéaire des quatre points de base (qui apparaissent en vert sur la figure 30 et qu'on appelle blobs).

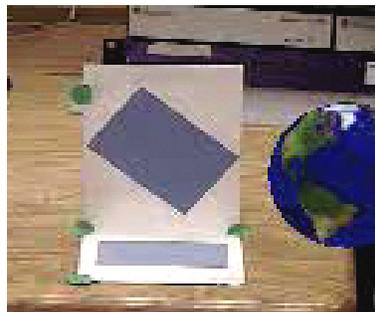


Figure 30. Calibration à base de 4 points (géométrie affine) [14]

Néanmoins, la détection des blobs présente des points faibles:

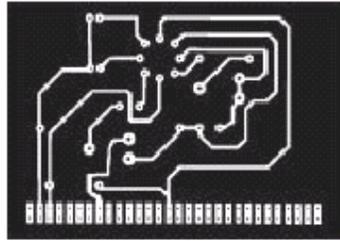
- Sensibles aux changements géométriques et photométriques.
- On doit déterminer le centre du blob et ceci peut ne pas être précis surtout en présence d'occlusion partielle.

1.12 Description du système proposé

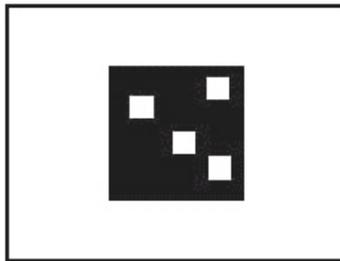
Le système que nous proposons se classe parmi les méthodes basées vision. Pour aligner l'objet virtuel de manière correcte et réaliste dans une séquence vidéo, on utilise un modèle bien défini.

Etant donné une séquence d'images, un modèle bien défini et l'objet à placer dans la séquence (voir figure 31), La méthode développée pour assurer une

insertion réaliste de l'objet virtuel 2D consiste à plaquer correctement l'objet virtuel sur le pattern (modèle) le long de la séquence.



(a)



(b)

Figure 31. (a) exemple d'un objet virtuel à insérer (b) image de référence du modèle

Deux questions majeures se posent donc :

- Comment identifier et suivre le modèle le long de la séquence vidéo ?
- Comment insérer l'objet virtuel dans la séquence vidéo ?

En prenant en compte ces deux questions, le système proposé procède en trois phases pour insérer un objet virtuel d'une manière correcte et réaliste dans une séquence vidéo :

- La Recherche du modèle dans la première image de la séquence vidéo : permet de trouver la position du modèle dans la première image de la séquence vidéo. Cela se fait par l'extraction de points d'intérêt par le détecteur de coins HARRIS [17] et leur mise en correspondance avec la première prise de vue. Pour se faire, on a adopté une méthode initialement inventée pour les problèmes de stéréovision, à savoir la mise en correspondance 'PILU' [18]. L'algorithme de RANSAC (Random sample consensus) [19] a été utilisée ici pour une correction des mauvaises correspondances générées précédemment. (voir la section 3.4.2.4).
- Le Suivi du modèle le long de la séquence vidéo : permet de suivre des points extraits du modèle. Il opère sur deux étapes pour chaque prise de vue : - la prédiction de la position d'un point dans une prise de vue à partir des positions relatives au même point dans les prises précédentes.

La correction de la position prédite par une mesure effectuée dans notre cas par un **détecteur de coins HARRIS**. L'**algorithme de RANSAC** est utilisé pour une estimation robuste des homographies dans le cas par exemple d'une **occlusion partielle du modèle**, ce qui empêche la détection des coins. (Voir la section 3.4.2.4).

- Le **Placement de l'objet virtuel sur le modèle** : permet d'insérer l'objet virtuel dans la séquence vidéo en utilisant les points extraits et suivis du modèle. Cela se fait en calculant l'homographie qui va nous permettre de faire la projection de l'objet virtuel 2D sur l'image de référence du modèle, et ensuite faire une projection sur l'image de la scène en utilisant l'homographie calculée dans la phase de suivi.

Dans ce qui suit nous allons détailler chacune de ces trois phases.

1.13 Phase de recherche du modèle

La phase de recherche du modèle permet de localiser celui ci dans la première image de la séquence vidéo. La localisation du modèle se fait en retrouvant les **points d'intérêt** le caractérisant, afin de les utiliser dans la phase suivante qui consiste à les suivre pour augmenter la scène avec l'objet virtuel. Cette phase se fait en deux étapes :

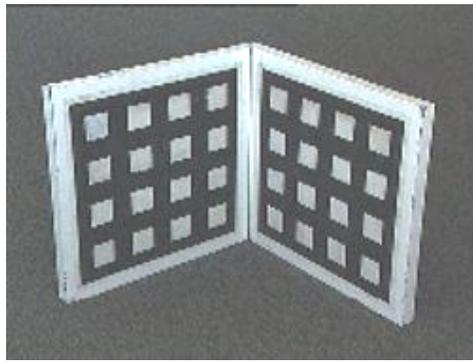
- **Extraire les points d'intérêt du pattern** : cette étape se fait à l'aide d'un détecteur de coins HARRIS.
- **Mise en correspondance des points extraits sur l'image de référence du pattern et ceux de la première image de la séquence vidéo.**

Avant d'expliquer comment s'effectue la localisation du modèle, il convient tout d'abord de décrire la nature du modèle qui est un élément clé dans le système proposé. C'est grâce à ce modèle qu'on peut contourner la calibration classique de la caméra et aligner ainsi l'objet à insérer d'une manière correcte.

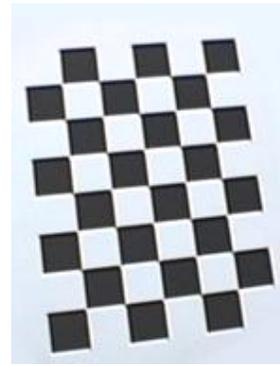
1.13.1 Choix du pattern (le modèle)

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.

La figure 32 illustre deux patterns ainsi qu'un exemple de calibration à base de pattern planaire.



Exemple de pattern non planaire



Exemple de pattern planaire



Figure 32. Calibration à base de pattern planaire

Les modèles illustrés dans la figure 32 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 (comme ce qui est utilisé dans **[20]**) (voir figure 33).

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 de HARRIS.



Figure 33. Exemples de modèles [20]

1.13.2 Extraction 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 HARRIS (voir figure 34). Ces points sont utilisés pour faire une mise en correspondance avec ceux de l'image de référence du 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 la phase de suivi) pour les augmenter à chaque fois avec l'objet virtuel.

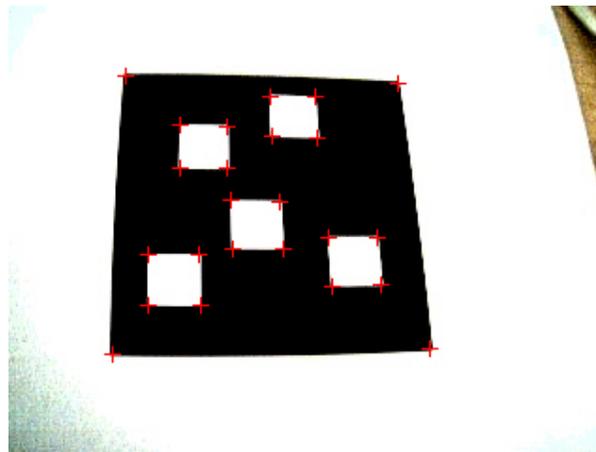


Figure 34. Détection de coins avec HARRIS

1.13.3 Localisation du modèle

Des primitives du type points d'intérêt sont extraites sur le pattern se trouvant dans une image de référence ainsi que la première image de la séquence (voir figure 35) moyennant le détecteur de Harris. La localisation du modèle revient en fait à mettre en correspondance terme à terme les primitives dans l'image de référence du modèle, avec celles se trouvant dans la première image de la séquence vidéo. La figure 36 montre un exemple de localisation du modèle dans la première prise de vue d'une séquence vidéo.

Pour trouver ces correspondances, l'algorithme de Pilu de la mise en correspondance stéréo a été adapté à notre système. Etant donné les deux ensembles de primitives coins extraites de l'image de référence et de la première image et en utilisant le détecteur de coins HARRIS, le principe de la mise en correspondance peut être décrit comme suit :

- Construire une matrice de similarité/proximité G entre les primitives des deux ensembles où chaque élément représente une mesure de corrélation pondérée par la proximité.

$$G_{ij} = e^{-(p_{ij}-1)^2/2\gamma^2} \cdot e^{-d_{ij}^2/2\sigma^2}$$

Où le premier terme de cette mesure est une fonction gaussienne de corrélation, le second terme est une fonction gaussienne de la distance euclidienne entre les primitives. γ et σ sont des paramètres dont les valeurs sont déterminées empiriquement. Le facteur σ est utilisé pour contrôler les interactions entre les primitives. De petites valeurs de σ renforcent les interactions locales alors que de grandes valeurs renforcent les interactions globales.

- Effectuer une décomposition en valeur singulière de G . Soit $G = T D U^T$ où T et U sont deux matrices orthogonales et D est une matrice diagonale contenant les valeurs singulières.
- Calculer la matrice $P = T E U^T$ où E est une matrice diagonale dont les éléments diagonaux sont égaux à 1. La matrice P a la propriété de conserver l'information, contenue dans G , sur la proximité et la similarité en terme de corrélation entre les primitives.
- Générer les correspondances terme à terme en prenant les maximas ligne/colonnes.

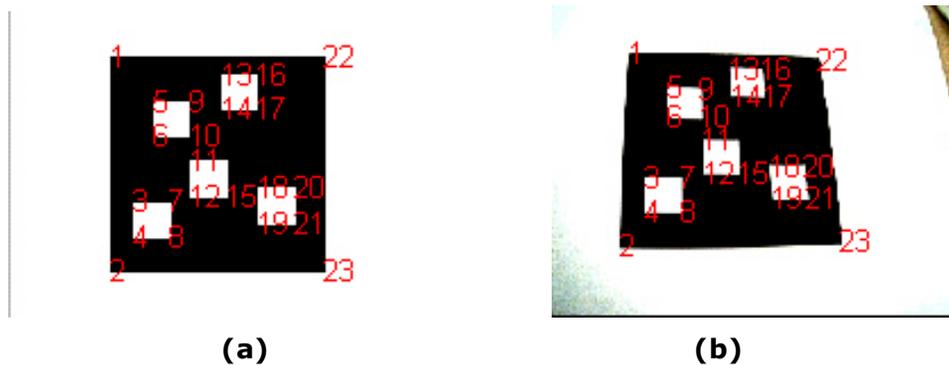


Figure 35. Mise en correspondance pour la localisation du modèle

(a) Image de référence du modèle

(b) Première image de la séquence vidéo

Un problème apparaît tout de même quelque fois lors de la mise en correspondance de la première image de la séquence avec l'image de référence du modèle. Il s'agit de la génération de mauvaises correspondances.

Dans la figure 36, le système n'est pas arrivé à trouver la correspondance du coin numéro 5. Cela est dû généralement au détecteur de coins HARRIS ou bien au processus de mise en correspondance lui-même.

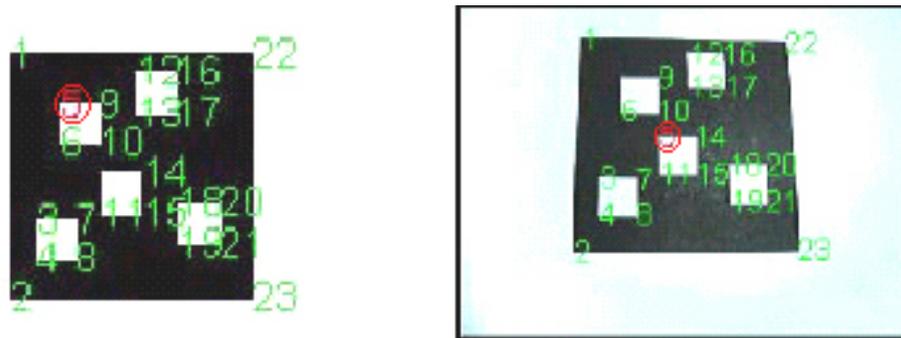


Figure 36. Génération d'une mauvaise correspondance

Afin d'atténuer l'effet des mauvaises correspondances entre les primitives points extraites du modèle et les primitives points extraites de la première image de la séquence vidéo, nous avons opté pour une estimation robuste par RANSAC (voir l'algorithme dans la section 3.4.2.4). La figure 37 montre que le système à travers l'algorithme de RANSAC a réussi à localiser la mauvaise correspondance générée et procède à sa correction (c'est la position du cinquième point qui est mise en rouge).

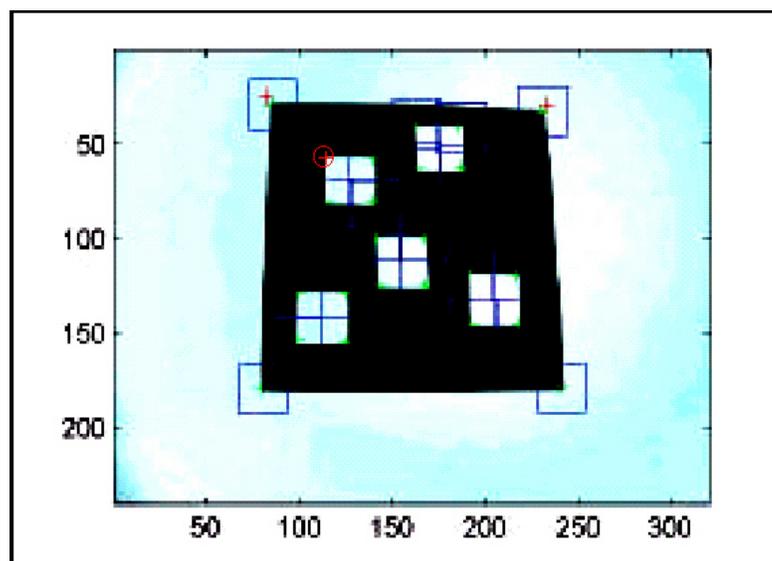


Figure 37. Traitement de RANSAC de la mauvaise correspondance

1.14 Phase de suivi du modèle le long de la séquence

Cette étape est la plus importante dans le processus d'augmentation d'une séquence vidéo car elle permet de savoir dans quelle position se trouve le modèle afin de l'augmenter avec l'objet virtuel.

A cet effet le filtre de Kalman est utilisé pour une estimation de la position de chaque primitive point dans la séquence vidéo. Cette phase se fait en deux étapes : prédiction de la position des coins et correction de la prédiction en prenant en compte un certain nombre de paramètres.

Dans ce qui suit, nous allons donner une introduction théorique au filtre de Kalman, ensuite nous verrons comment l'adapter pour le problème de suivi des primitives points se trouvant sur le modèle, le long d'une séquence vidéo. Nous traiterons aussi comment régler le problème de mauvaises correspondances dues à la perte de points et à l'occlusion.

1.14.1 Introduction théorique au filtre de Kalman

En 1960, R.E.Kalman a publié son fameux article décrivant une solution récursive pour le problème du filtrage linéaire des données discrètes.

En ce temps et du aux grands progrès réalisés dans le domaine du calcul numérique et apparition des premiers ordinateurs, le filtrage de Kalman a été sujet à de nombreuses recherches intensives et applications particulièrement dans le domaine de la navigation assistée. **[21]**

Le filtre de Kalman utilise deux ensembles d'équations mathématiques qui procurent une solution efficace récursive pour estimer un état passé, présent ou futur pour un système à modèle inconnu.

1.14.1.1 Le processus à estimer

La méthode de Kalman est une procédure d'estimation dynamique des paramètres qui sont fonctions du temps.

Généralement le filtre de Kalman s'adresse aux problèmes essayant d'estimer un état $x \in \mathbb{R}^n$ représentant un processus discret temporel qui est gouverné par l'équation linéaire stochastique suivante :

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

Avec la mesure $z \in \mathbb{R}^m$:

$$z_k = Hx_k + v_k$$

w_k et v_k sont respectivement des bruits sur des paramètres à estimer et sur des mesures (observations). w_k et v_k sont des variables aléatoires dont la distribution est normale, donc :

$$P(w) \sim N(0, Q)$$

$$P(v) \sim N(0, R)$$

- En pratique, la covariance du bruit du processus et de la mesure changent à chaque étape ou à chaque mesure, mais nous supposons qu'elles sont constantes.
- $A(N \times N)$ est une matrice qui fait le lien entre les variables d'état du système à deux étapes successives.

- $B(N \times N)$ fait le lien entre les valeurs optionnelles de contrôles et la variable d'état du système.
- $H(M \times N)$ est la matrice qui fait le lien entre la variable d'état du système et les mesures.

1.14.1.2 Méthode

Comme il a été précisé le filtre de Kalman fait usage de deux ensembles d'équations pour estimer la valeur de la variable d'état. L'estimation de la variable d'état à chaque étape passe par deux phases : une phase de prédiction et une autre de correction de la prédiction (la figure 38 illustre le fonctionnement du filtre de Kalman) :

- Phase de la prédiction : Cette phase permet de prédire la valeur courante de la variable d'état en prenant en compte l'état précédent. La variable prédite est appelée *estimation de l'état à priori*. Elle permet aussi de prédire la matrice de covariance d'erreur en prenant en compte la matrice de covariance de l'état précédent. Cette matrice est appelée la matrice de covariance d'erreur à priori. Donc cette phase utilise deux équations pour prédire la valeur de la variable d'état:

$$\begin{aligned} \bar{x}_k &= A x_{k-1} + B u_k \\ P_k^- &= A P_{k-1} A^T + Q \end{aligned}$$

Avec :

\bar{x}_k : estimation de l'état à priori.

P_k^- : estimation de la matrice de covariance d'erreur à priori.

x_{k-1} : estimation de l'état de l'étape précédente.

P_{k-1} : estimation de la matrice de covariance d'erreur de l'étape précédente.

u_k : vecteur de contrôle.

Q : la covariance du bruit lié au processus.

A, B : relie l'état k à l'état $k-1$.

- Phase de la correction : Cette phase permet de corriger les erreurs faites dans la phase de prédiction en prenant en compte la mesure z_k . La variable estimée est appelée *estimation de l'état à posteriori*. Elle permet aussi d'estimer la matrice de covariance d'erreur appelée matrice de covariance d'erreur à posteriori. Cette phase utilise trois équations pour estimer la valeur de la variable d'état et la matrice de covariance d'erreur qui sont les paramètres d'entrée pour l'étape suivante. Ces équations sont:

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ x_k &= \bar{x}_k + K_k (z_k - H \bar{x}_k) \end{aligned}$$

$$P_k = (I - K_k H) P_k^-$$

Avec :

x_k : estimation de l'état à posteriori.

P_k : estimation de la matrice de covariance d'erreur à posteriori.

K_k : le gain de Kalman.

R : la covariance du bruit lié à la mesure.

I : matrice unitaire.

Pour utiliser le filtre de Kalman, la covariance du bruit de mesure (R) et la covariance du bruit du système (Q) doivent être évaluées à l'avance. La détermination de la covariance du bruit du système est généralement plus difficile parce que l'observation directe de l'état du système est impossible.

La figure 38 montre les différentes étapes d'évaluation de la variable d'état du système. Il y a deux étapes principales pour l'évaluation de cette variable d'état; l'évaluation a priori (la prédiction) et l'évaluation a posteriori (la correction). La variable d'état du système est estimée premièrement à partir de valeurs de l'étape précédente et en suite elle est corrigée par des mesures dans l'étape de correction.

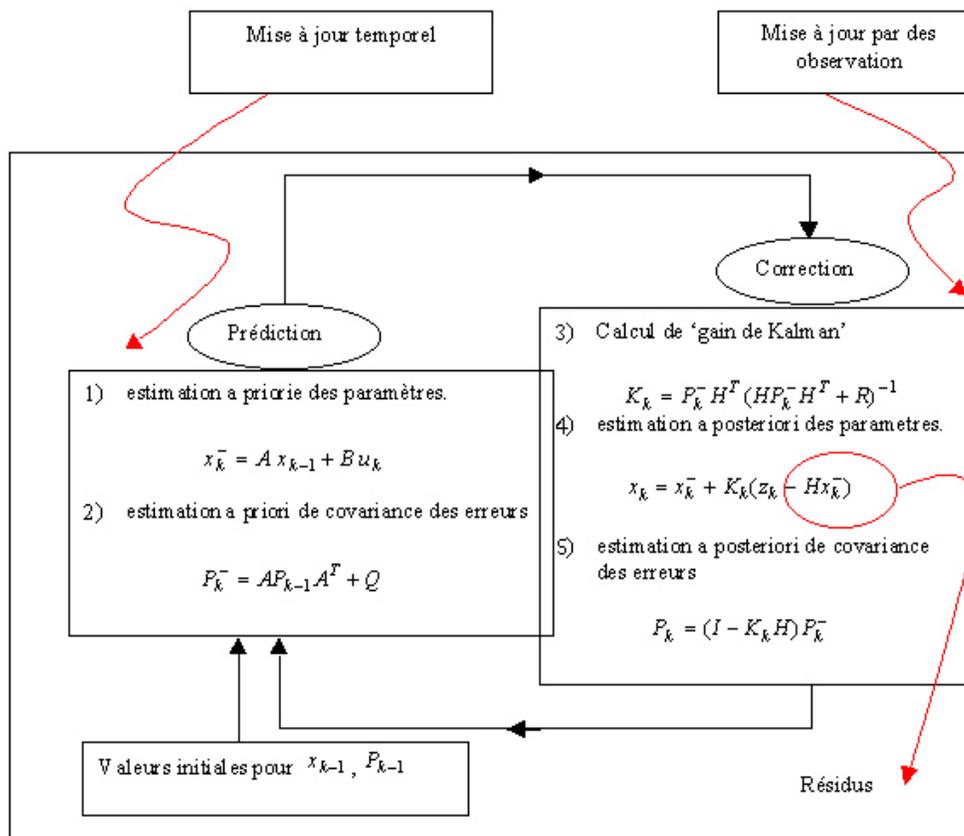


Figure 38. Fonctionnement du filtre de Kalman

1.14.2 Suivi du modèle sur la séquence d'images

Une fois le pattern localisé sur la première image, il est nécessaire de pouvoir prédire sa position sur les autres images de la séquence, ce qui revient de même à effectuer le suivi des primitives coins sur la séquence. A cet effet, le filtre de Kalman discret a été adapté. En dépit de la complexité relative de l'origine probabiliste du filtrage de Kalman, il est simple et souple à utiliser. Simple du fait que l'utilisateur n'a qu'à projeter les paramètres du filtre sur le problème à résoudre, et souple dans le sens où on peut toujours l'adapter pour répondre aux besoins particuliers du problème traité avec le réglage des paramètres covariance du bruit de modèle et celle de la mesure.

1.14.2.1 Suivi des primitives à base du filtre de Kalman

Dans le cas de notre application le filtre de Kalman est utilisé pour suivre des points bien précis localisés sur un modèle bien déterminé tout au long d'une séquence vidéo. Donc il suffit de spécifier ce que représente la variable d'état et comment effectuer la mesure pour estimer la position des points d'intérêt dans la scène. Et à partir de cette spécification on déduira les équations du filtre de Kalman.

Notre système est modélisé par les deux équations suivantes :

$$x_k = Ax_{k-1} + w_{k-1}$$

$$z_k = Hx_k + v_k$$

Où x_k représente le vecteur d'état à l'étape k :

$$x_k = [i_k \ j_k \ v_{xk} \ v_{yk}]^T$$

i_k et j_k représentent les coordonnées du point prédit x_k , v_{xk} et v_{yk} représentent respectivement la vitesse selon l'axe des abscisses et la vitesse selon l'axe des ordonnées.

z_k est le vecteur d'observation ou la mesure effectuée à l'étape k :

$$z_k = [z_x \ z_y]^T$$

z_x , z_y représentent les coordonnées de la mesure z_k du point x_k . La mesure est effectuée dans notre application par le détecteur de coins HARRIS.

A est la matrice de transition ou la matrice qui relie l'état courant à l'état précédent.

$$A = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

H est la matrice qui relie la mesure à la variable d'état.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

w_{k-1} est le vecteur aléatoire d'erreur lié au modèle suivant une loi normale $N(0, Q)$. v_k est le vecteur aléatoire d'erreur lié à la mesure suivant une loi normale $N(0, R)$.

Maintenant qu'on a décrit notre système avec les équations précédentes on déduit les équations du filtre de Kalman afin d'estimer à chaque fois la position des points d'intérêt à suivre sur le modèle le long d'une séquence vidéo. Comme décrit précédemment il y a deux ensembles d'équations : équations de la prédiction et équations de la correction.

Les équations de prédiction sont :

$$\begin{aligned} \bar{x}_k &= A x_{k-1} \\ P_k^- &= A P_{k-1} A^T + Q \end{aligned}$$

Les équations de correction sont :

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ x_k &= \bar{x}_k + K_k (z_k - H \bar{x}_k) \\ P_k &= (I - K_k H) P_k^- \end{aligned}$$

La notation est la même que celle utilisée précédemment.

1.14.2.2 Algorithme du suivi des primitives

L'adaptation du filtre de Kalman à notre application est très facile. Elle se compose de deux étapes principales : réglage du filtre et suivi des primitives coins sur le modèle le long de la séquence vidéo. Le réglage du filtre est le processus d'initialisation des paramètres du filtre de Kalman (Q, R, x_0, P_0), il se fait généralement d'une manière empirique. Le suivi des primitives coins consiste en la prédiction de leurs positions et la correction de cette prédiction en prenant en compte la mesure faite à l'aide du détecteur de coins HARRIS. Elle peut être décrite par l'algorithme suivant :

1- initialiser $Q, R, x_0 = z_0$ et P_0 (réglage du filtre).

2- Pour chaque étape $k=1, \dots$ Faire :

- Prédire P_k^-, \bar{x}_k en utilisant les équations de prédiction.
- Calculer le gain K_k et mettre à jour P_k, x_k en utilisant les équations de correction en prenant en compte la mesure faite dans cette étape.
- Mettre en sortie x_k et P_k pour la prochaine étape.

1.14.2.3 Réglage du filtre

Le réglage du filtre a un impact direct sur la précision du filtre au cours du suivi des primitives points. Par voie de conséquence, un choix judicieux doit être effectué. Dans le cas de notre application, Q doit être relativement petite par rapport à R , c'est à dire que nous ne tolérons pas d'erreurs au niveau de l'estimation théorique comme au niveau de la mesure.

Par ailleurs, pour améliorer la performance et réduire le temps d'exécution, on peut minimiser la recherche des coins dans une partie de l'image notablement plus petite dont la taille correspond à la matrice de covariance P qui représente la marge de confiance avec laquelle le filtre fourni la prédiction. Le choix de la matrice P doit satisfaire les deux conditions suivantes :

- La taille fenêtre doit être suffisamment grande pour suivre correctement le point le long de la séquence vidéo.
- La taille de la fenêtre doit être suffisamment petite pour ne pas suivre un autre coin.

Ces deux conditions doivent être satisfaites pour éviter le problème de perte de point lors du processus du suivi.

1.14.2.4 **Calcul robuste de l'homographie**

Le calcul de l'homographie s'effectue de manière similaire à celle expliquée dans la problématique. Un problème qui se pose dans cette phase est que, si les positions des points pris pour le calcul de l'homographie sont inexactes, la transformation géométrique définie par l'homographie sera fautive, ce qui aura pour effet la génération d'un mauvais recalage de l'objet virtuel dans la scène réelle.

Il y a essentiellement deux raisons qui sont derrière l'apparition des mauvaises correspondances :

- Perte de point au cours du suivi : ceci est dû à l'ajustement de la fenêtre, si elle est trop grande, il y a un risque de suivre un autre coin, et si elle est trop petite il y a risque de perdre ce coin dans la prochaine prise de vue.
- Occlusion partielle du modèle : ce qui aura pour effet de rendre un certain nombre de coins indétectable par HARRIS.

Pour palier à ce problème l'algorithme RANSAC a été utilisé pour corriger les mauvaises correspondances et effectuer une estimation robuste de l'homographie.

Son principe pour le cas qui nous intéresse peut être décrit par les étapes de l'algorithme suivant :

1. Choisir aléatoirement quatre correspondances potentielles entre point pattern dans une image référence et point pattern dans l'image courante de la séquence.

2. Estimer les paramètres de l'homographie à partir des correspondances choisies.
3. Pour l'homographie calculée en 2, calculer pour chaque coin le résidu $X'-H \times X$. considérer ce coin comme étant un 'inlier' si le résidu est inférieur à certain seuil (2 pixels pour notre cas). Sinon le considérer comme étant un 'outlier'.
4. Calculer le nombre d'inliers. Pour qu'une homographie soit acceptée le nombre d'inliers correspondants doit être suffisant (en générale plus de 75%).
5. Retenir l'homographie ayant le plus grand nombre de 'inliers'.

L'estimation de l'homographie proprement dite est effectuée par résolution d'un système linéaire.

La correction des mauvaises correspondances se fait par une reprojction des 'outliers' en utilisant l'homographie retenue (voir figure 38).

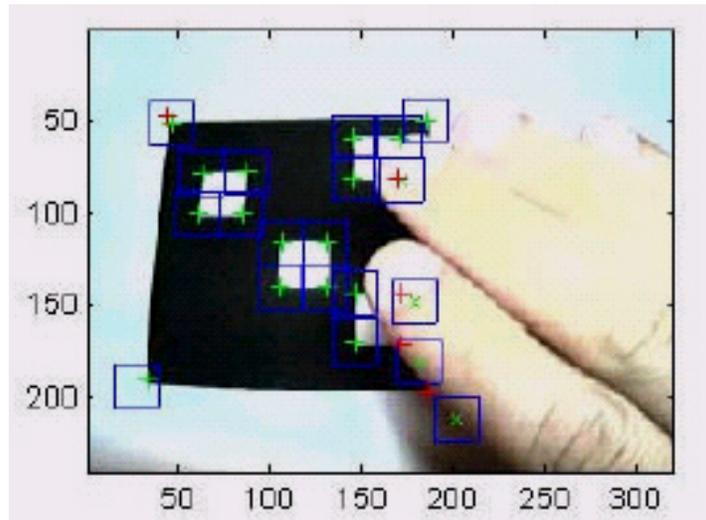


Figure 39. Utilisation de RANSAC pour la correction des mauvaises correspondances en cas d'occlusion

La figure 38 montre la correction de mauvaises correspondances (en \times vert) par les nouvelles (en $+$ rouge) en utilisant l'homographie ayant le plus grand nombre d'inliers.

Les outliers corrigés par RANSAC et les inliers seront les paramètres d'entrée pour la phase de suivi de la prise de vue suivante.

Par l'introduction de la méthode RANSAC on aura bénéficié des avantages suivants :

- On a moins de chance de perdre la cible, puisque au lieu de suivre quatre coins bien déterminés, le processus de suivi peut utiliser quatre points quelconque parmi l'ensemble des coins du modèle.

- Le fait de perdre un coin à travers le processus de suivi ne constitue point un incident fatal.
- Une occlusion partielle du modèle n'affecte pas le bon fonctionnement du système.

1.15 Phase d'incrustation de l'objet virtuel

Cette phase consiste à insérer l'objet virtuel dans la séquence vidéo sur le modèle se trouvant dans la scène. Cette phase est importante et délicate car il faut insérer l'objet d'une manière suffisamment précise afin qu'il apparaisse le plus naturel possible.

L'incrustation de l'objet virtuel se fait d'abord sur l'image de référence du modèle, et ceci se fait par le calcul de l'homographie qui définit la transformation géométrique entre l'image de référence du modèle et l'objet virtuel à insérer dans la séquence vidéo.

Ensuite la projection de l'objet virtuel dans la scène se fait à l'aide de l'homographie calculée précédemment entre l'image de référence du modèle et une prise de vue de la séquence vidéo.

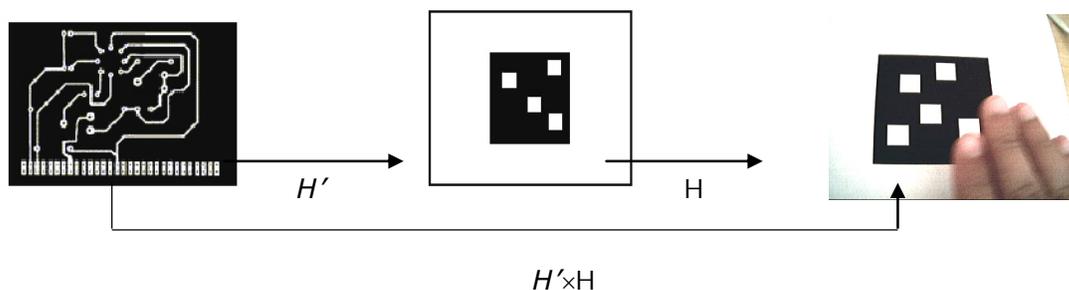


Figure 40. *Projection de l'objet virtuel* dans la séquence vidéo

Donc la projection de l'objet virtuel dans une prise de vue de la séquence vidéo revient en fait à trouver la matrice $H' \times H$ qui représente l'homographie définissant la transformation géométrique à effectuer pour incruster l'objet virtuel.

Pour assurer une visualisation réaliste, l'objet virtuel doit apparaître au-dessous de l'objet qui a généré l'occlusion.

Il est question de détecter tout objet source d'occlusion, et de procéder à l'augmentation sur seulement les régions modèle non occluses.

Nous pouvons tirer profit de la connaissance parfaite de ce que serait le modèle sur la scène réelle puisque nous connaissons sa projection géométrique exprimée mathématiquement sous forme d'homographie. Ceci va permettre de détecter tout objet intrus.

Pour se faire, nous procédons à une différenciation simple entre l'image du modèle sur la scène réelle et celle du modèle de référence après projection.

Tenant compte du bruit sur la scène réelle, nous devons faire un seuillage pour déterminer les régions du pattern dans la scène réelle similaire à leurs correspondants dans la projection de l'image de référence. Nous aurons à la fin

une image binaire pour laquelle le noir indique là où il faut faire l'augmentation et le blanc pour les régions occluses (figure 41.b).

Un problème dû au bruit contenu dans les images capturées est qu'aux alentours des carrés du pattern le système considère que ces régions sont des régions issues d'une occultation.

Pour palier ce problème nous avons procédé à une décomposition de l'image binarisée en régions connexes. Et nous tenant compte que de la région qui a la plus grande taille (figure 41.c).

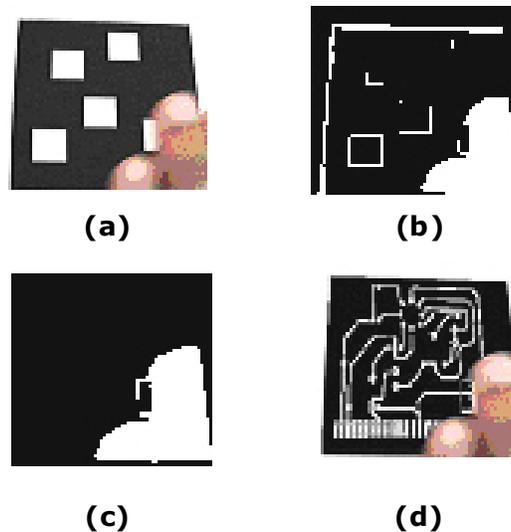


Figure 41. Visualisation réaliste dans le cas d'occlusion du modèle

- (a) Prise de vue de la séquence vidéo
- (b) Différenciation entre l'image du modèle sur la scène réelle et celle du modèle de référence après projection, et binarisation de l'image résultante
- (c) Détection de l'objet générant l'occlusion
- (d) Prise de vue après augmentation

Temps réel

1.16 Introduction

Dans l'année 1999, seulement 2% des microprocesseurs produits, étaient contenus dans des ordinateurs de bureau, alors que la plus grande majorité était intégrée dans des systèmes dit « embarqués » [23][24]. Et dans l'année 2004, environ 260 millions de processeurs ont été vendus pour le marché du PC grand public à comparer aux 14 milliards de processeurs tout type confondu, vendu pour le marché de l'embarqué. [25]

Les systèmes embarqués sont une catégorie des systèmes **temps réels**. Ces derniers sont aujourd'hui présents dans de nombreux secteurs d'activités : dans l'industrie de production par exemple, au travers des systèmes de contrôle de procédés, dans les salles de marché au travers du traitement des données boursières en « temps réel », dans l'industrie du transport au travers des systèmes de pilotage embarqués, ou encore dans le secteur de la nouvelle économie au travers du besoin, toujours croissant, du traitement et de l'acheminement de l'information (vidéo, données, pilotage à distance, réalité virtuelle, etc.).

Ces systèmes informatiques temps réel se différencient des autres systèmes informatiques par la prise en compte de contraintes temporelles dont le respect est aussi important que l'exactitude du résultat. Autrement dit, le système ne doit pas simplement délivrer des résultats exacts, il doit les délivrer dans des délais imposés. [26]

1.17 Définition d'un système temps réel

Un système temps réel est un système réactif devant fournir des sorties logiquement correctes tout en respectant strictement des contraintes temporelles explicites.

Il est considéré comme défaillant s'il ne respecte pas au moins une de ses spécifications logiques ou temporelles. Les défaillances des systèmes temps réel peuvent avoir des conséquences importantes et leur niveau de criticité est souvent élevé. [27]

- Un système réactif est un système qui maintient une interaction permanente avec son environnement.
- L'exactitude logique (logical correctness) veut dire que les sorties doivent être adéquates en fonction des entrées, assurant le comportement désiré pour le système suite à des événements et aux données communiquées.

Donc, on peut définir le caractère temps réel d'une application comme l'aptitude d'une application à répondre à des sollicitations de l'environnement contrôlé, en des temps de réponse garantis. [28]

D'autre part, un système Temps Réel ne veut pas dire un système rapide, comme on pourrait être amené à le croire. Par exemple: le système de guidage

d'un navire peut apparaître comme n'étant pas un système Temps Réel, à cause de sa faible vitesse et qu'habituellement on a "assez" de temps (en minutes) pour prendre une décision avant d'agir. Néanmoins, d'après la définition c'est effectivement un système temps réel.

Par contre, il y a des auteurs qui font la distinction entre "Système Temps Réel" et Système en Temps Réel. Les "Système en Temps Réel" sont généralement des systèmes rapides capables de donner l'illusion de la réalité. Typiquement ce sont les simulations et les jeux interactifs qui ont besoin de donner à l'utilisateur l'apparence de la continuité du temps et un maximum de réalisme. Plus il y a d'images générées par unité de temps, meilleur c'est. [29]

1.18 Classification des systèmes temps réel

Tous les Système Temps Réel ne sont pas identiques, ce n'est pas la même chose de commander un système de freinage ABS d'une voiture, le système d'injection de combustible d'un moteur d'avion ou la décompression et la visualisation d'un fichier vidéo. Dans le premier et le second cas, un petit surcroît de temps d'exécution peut coûter des vies humaines ou de graves pertes de matériel; dans le troisième, il y a simplement dégradation de la qualité du système (l'image pourra rester figée et quelques portions perdues). Le premier type de système est appelé **Système Temps Réel strict**, et le second **Système Temps Réel souple**.

Donc suivant l'importance accordée aux contraintes temporelles, on distingue le temps réel strict ou dur (de l'anglais hard real-time) et le temps réel souple ou mou (soft real-time). Le temps réel strict ne tolère aucun dépassement de ses contraintes, ce qui est souvent le cas lorsque de tels dépassements peuvent conduire à des situations critiques, voire catastrophiques : pilote automatique d'avion, système de surveillance de centrale nucléaire, etc. À l'inverse, le temps réel souple s'accommode de dépassements des contraintes temporelles dans certaines limites au-delà desquelles le système devient inutilisable : visioconférence, jeux en réseau, etc.

On peut ainsi considérer qu'un système temps réel strict doit respecter des limites temporelles données même dans la pire des situations d'exécution possibles. En revanche un système temps réel souple doit respecter ses limites pour une moyenne de ses exécutions. On tolère un dépassement exceptionnel, qui sera peut-être rattrapé à l'exécution suivante. [26]

1.18.1 Temps réel strict

La majorité des systèmes temps réel critiques est exclusivement constituée de traitements qui ont des contraintes temporelles strictes : on parle de systèmes temps réel strict. C'est à dire qu'en condition nominale de fonctionnement du système, tous les traitements du système doivent impérativement respecter toutes leurs contraintes temporelles ; on parle alors de traitements temps réel strict ou dur (hard en anglais). Ceci suppose deux choses : i) qu'on est capable de définir les conditions de fonctionnement nominales en terme d'hypothèses sur l'environnement avec lequel le système interagit ; ii) qu'on est capable de garantir avant exécution que tous les scénarios d'exécution possibles dans ces conditions respecteront leurs contraintes temporelles. Ceci suppose à son tour

qu'on puisse extraire ou disposer de suffisamment d'informations sur le système pour déterminer tous les scénarios possibles. [30]

Le schéma suivant (Figure 42) illustre la qualité d'un système temps réel dur en terme de sa tolérance aux échéances.

On appelle échéance une contrainte temps à laquelle doit au plus tard se produire un événement. [31]

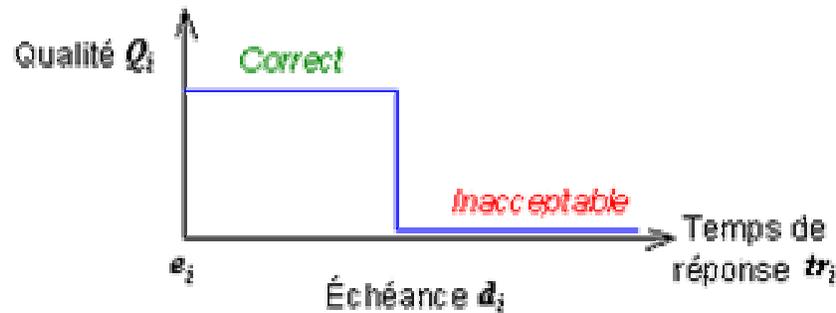


Figure 42. Qualité d'un Système temps réel dur en cas de dépassement d'une échéance [32]

Le non respect d'une échéance d'une fonction ayant une contrainte temps réel dure est considéré comme une défaillance pouvant entraîner des conséquences catastrophiques.

Aussi, un système temps réel dur comporte des fonctions de ce type et éventuellement des fonctions à contraintes souples (section 4.3.2). L'objectif principal dans la réalisation d'un tel système est de satisfaire strictement toutes les contraintes de temps dures.

Les éléments utilisés pour réaliser ces systèmes doivent tous posséder un comportement déterministe. Il est nécessaire de choisir des composants logiciels et matériels et des principes de réalisation souvent spécifiques permettant de garantir un comportement global correct [32].

1.18.2 Temps réel souple

Une autre classe de systèmes est moins exigeante quant au respect absolu de toutes les contraintes temporelles. Les systèmes de cette classe, dits temps réel souple ou mou (soft en anglais), peuvent tolérer un taux "acceptable" de fautes temporelles (non respect des contraintes) de la part d'une partie des traitements (eux-mêmes dits "temps réel souple"), et sans que cela ait des conséquences catastrophiques. C'est le cas par exemple de fonctions de communication ou de traitement multimédia pour lesquelles la qualité générale des documents n'est pas affectée de manière significative par le retard ou le manque d'informations d'une image ou d'un son. Et de manière plus générale, cette classe comprend entre autres les systèmes où la qualité est appréciée par nos sens : on parle dans ce cas de systèmes et d'applications multimédia (téléphonie, vidéo, rendu visuel interactif par exemple).

La mesure du respect des contraintes temporelles prend la forme d'une donnée probabiliste : la qualité de service relative à un service particulier (nombre d'images ou nombre d'échantillons sonores rendus par secondes, par exemple),

ou relative au comportement du système dans son ensemble (nombre de traitements qui ont pu être rendus dans les temps), ou les deux combinés [30].

Donc, dans ce genre de système, le non respect d'une échéance d'une fonction temps réel à contrainte souple n'est pas catastrophique, et la qualité de la réponse, bien que dégradée, reste acceptable comme illustré dans le schéma suivant (figure 43).

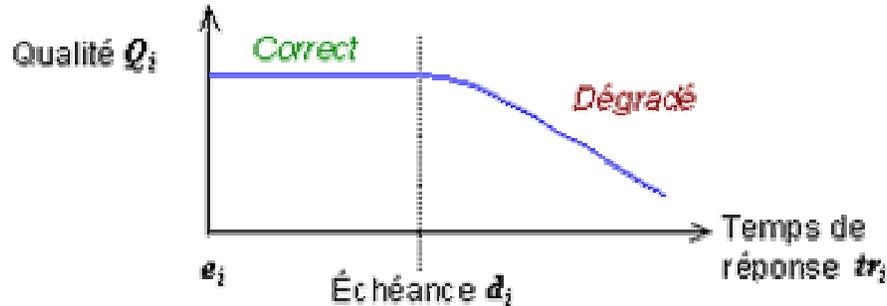


Figure 43. *Qualité d'un Système temps réel souple en cas de dépassement d'échéance* [32]

Aussi, un système temps réel souple ne comporte que des fonctions ayant des contraintes temps réel souples. Il peut être validé en raisonnant en termes statistiques sur le temps de réponse moyen.

Les composants utilisés pour réaliser les systèmes temps réel souples (systèmes d'exploitation, réseaux de communication, etc...) n'ont pas nécessairement des temps de réponse déterministes.

1.18.3 Temps réel ferme

On distingue une sous-famille dans celle des traitements temps réel souple, suivant que le non-respect d'une contrainte temporelle continue d'apporter quelque chose au système, ou non : si le traitement dépasse une contrainte temporelle, et si le fait de continuer dans ces conditions jusqu'à son terme n'apporte rien au système, on parle de traitement temps réel ferme (firm en anglais).

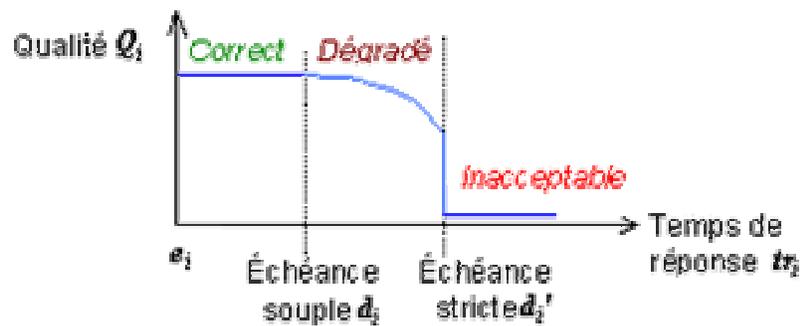


Figure 44. *Qualité d'un Système temps réel ferme* [32]

Dans un tel système, certaines fonctions ayant des contraintes temps réel souples doivent respecter un temps de réponse moyen tout en restant à l'intérieur d'une fenêtre temporelle stricte. De telles fonctions dites "temps réel

ferme" sont prises en compte dans les méthodes d'analyse et de validation temporelle des systèmes temps réel.

C'est le cas par exemple d'une fonction d'échantillonnage périodique devant répondre en moyenne en une période tout en étant capable de conserver au plus les données de quelques périodes. Dans le cas d'une charge importante, celle-ci doit obligatoirement fournir ses résultats avant l'écoulement de ces quelques périodes. **[32]**

1.18.4 Système embarqué et système détaché

D'un autre côté, tout système temps réel est caractérisable par son rapport aux matériels avec lesquels il interagit. On distingue ainsi deux principales catégories de systèmes temps réel : les systèmes embarqués et les systèmes détachés.

Le système embarqué (embedded) est complètement encapsulé dans le matériel qu'il pilote ou contrôle et en est fortement dépendant (ex : contrôle d'un carburateur, stabilisateur d'avion).

Un système embarqué donc, ne possède généralement pas d'entrées/sorties standards comme un clavier ou un écran d'un ordinateur. Le système matériel et l'application sont intimement liés ; le logiciel embarqué étant noyé dans le matériel. Ce dernier et le logiciel ne sont pas facilement discernables comme dans un environnement de travail classique de type ordinateur PC. **[25]**

Par contre, le système détaché (detached) ou organique est peu dépendant du matériel sur lequel il fonctionne et peut en être détaché (ex : contrôle de processus industriel).

Lorsqu'un système se situe entre ces deux catégories, on parle de système semi détaché.

1.19 Besoin de traitement en temps réel dans la réalité augmentée

Imaginons un système de réalité augmentée qui tente d'améliorer la vue d'un chirurgien en pleine opération chirurgicale par l'insertion en temps réel d'autres vues issues de capteurs telles que les images Rayon X. Il est évident qu'un retard dans l'augmentation des images peut engendrer un désastre!



Figure 45. Augmentation avec l'image intérieure du cerveau d'un patient issue de donnée Rayon X [15]

De manière générale, le besoin de traitement en temps réel se fait sentir dès qu'il est question pour une application donnée, de respecter certaines contraintes temporelles.

Dans les applications temps réel de la réalité augmentée, il est question de superposer des objets virtuels sur des scènes réelles en même temps de l'acquisition des séquences vidéo, ce qui implique que le calcul engendré par les algorithmes relatifs au processus de suivi et d'incrustation doit s'effectuer entre la transition de deux prises de vue séquentielles.

De manière plus générale, pour une qualité cinématographique des images, soit de l'ordre de 24 images par seconde, le délai de traitement pour une prise de vue ne doit pas excéder 0.04 seconde.

Cependant, dans les applications de réalité augmentée, cette limite est moins sévère, puisque on fixe à 10 images par seconde la vitesse minimale du flux vidéo pour garantir le réalisme de l'augmentation [14].

1.19.1 L'approche adoptée

La première question qu'on doit se poser est : dans quelle catégorie peut-on classer les applications temps réel de réalité augmentée ? Est-ce du temps réel strict ou souple ?

La réponse dépend, bien évidemment, de l'application elle-même ; s'il s'agit par exemple d'une augmentation de la perception d'un chirurgien au cours d'une intervention chirurgicale, un dépassement de contraintes temporelles ne sera en aucune manière toléré. Et dans ce cas, on va se trouver dans la catégorie des applications temps réel strict. Par contre, on peut très bien tolérer un dépassement de contraintes temporelles, dans une application plus simpliste où il est question, par exemple, d'assister l'utilisateur par des indications, comme dans certaines applications de maintenance. Toutefois ce dépassement est dégradant au niveau de l'application. Là, on est bien évidemment dans la catégorie temps réel souple.

Comme ce qui vient d'être expliqué dans la section 4.3.1, la réalisation d'une application de la première catégorie (temps réel strict), doit impérativement avoir un comportement déterministe, ce qui sous entend l'utilisation de composants logiciels et matériels et des principes de réalisation souvent spécifiques.

Notre objectif étant de réaliser une application qui peut marcher sur de simples ordinateurs de bureau, caractérisés principalement par des caractéristiques physiques ordinaires et ayant surtout, un système d'exploitation non temps réel.

Ces considérations nous amènent à nous intéresser à la deuxième catégorie, à savoir les applications temps réel mou (ou souple) qui peuvent être validées en raisonnant en termes statistiques sur le temps de réponse moyen (section 4.3.2)

Notre approche consiste donc à implémenter notre méthode de suivi, et à mesurer le temps de réponse moyen sur des ordinateurs de bureau, pour conclure sur le respect de la contrainte temps réel exigée, à savoir le suivi d'au moins dix images par seconde.

D'autre part, puisque il s'agit d'une application temps réel souple, dire dix images par seconde n'est pas équivalent à une image par un sixième de seconde.

Ce qu'on va faire, c'est calculer le temps de suivi sur une vidéo de quelques dizaines de prises de vue et déduire le temps moyen de suivi de dix images.

Cette approche ; à savoir le fait de conclure sur le respect de la contrainte temps d'après les résultats statistiques, est tout a fait justifiable, puisque l'objet de traitement dans une application d'imagerie est une grille de données de type identique (les pixels). Et qu'en théorie, le contenu même des prises de vue n'a pas d'impact sur le temps d'exécution. Autrement dit, si on exécute le même programme sur des images différentes, ça prendra en théorie le même temps, en dépit du contenu de celles-ci, mais à condition que les images aient la même résolution (dimension et nombre de canaux), puisqu'il s'agit en fin de compte de faire le même calcul avec le même nombre d'opérations sur le même type d'entités (les valeurs des pixels).

Néanmoins dans la pratique, il y a d'autres facteurs qui font que les délais d'exécution ne sont pas aussi déterministes ; le chargement d'une image par exemple, sa décompression et sa mise en niveau de gris peuvent prendre un temps différent en fonction de l'espace contigu disponible au moment de l'exécution.

Ce non déterminisme est toléré dans notre cas puisqu'il s'agit d'une application temps réel souple.

Nous consacrerons une partie dans le prochain chapitre (chapitre 5) pour discuter et commenter les résultats expérimentaux obtenus. Mais traitons d'abord quelques détails visant à accélérer la vitesse de traitement des applications de l'imagerie.

1.20 Vitesse de traitement des algorithmes d'imagerie

La limite minimale imposée des dix images par seconde a toujours constitué un déficit pour les applications de réalité augmentée, étant donné le volume très grand d'informations à traiter lorsqu'on a affaire aux images, ce qui engendre naturellement un temps d'exécution très grand.

En considérant l'addition comme opération de base ; l'objet de base correspond à une addition, une division ou une racine carrée correspond à quatre additions et une élévation à la puissance correspond à huit additions. Si l'on veut afficher des images sur un écran d'un million de pixels à la vitesse de vingt-cinq images par seconde, on a donc un besoin en puissance de calcul de 2,2 Gflops dans le cas général, ce qui est considérable. [1]

C'est exactement pour cette raison que particulièrement dans le domaine de l'imagerie, on ressent le besoin d'accélérer les calculs.

Donc pour réduire le temps d'exécution, deux types de solutions s'imposent :

- Les solutions logicielles ou software.
- les solutions matérielles ou hardware.

1.20.1 Solution logicielle

Il est question ici d'exploiter les spécificités du langage de programmation pour réduire le temps d'exécution, notamment en ce qui concerne la gestion de l'espace mémoire (allocation et libération).

Matlab constitue un très bon exemple pour illustrer cette notion (la solution logicielle). Sachant que nous avons implémenté notre système à travers les deux langages ; Matlab : pour la modélisation du système, et Visual C++ pour l'implémentation du suivi en temps réel.

Matlab est une énorme calculatrice mathématique ; ce n'est pas un langage de programmation classique, la notion de compilation et d'édition de liens n'existe pas, il procède plutôt suivant le mode interpréteur, instruction par instruction, ce qui a comme effet de rendre le traitement des programmes manifestement lent surtout dans notre cas qui est le traitement d'une grande masse de données qui est l'image.

Par contre, l'utilisation des fonctionnalités spécifiques a Matlab réduit le problème et permet une amélioration considérable en terme de vitesse de traitement.

En Matlab, on peut écrire un programme tel qu'il est pris d'un autre langage séquentiel. Mais il est aussi intéressant de transformer le programme en une autre forme plus concise et plus performante à l'aide des fonctionnalités de Matlab, on aura donc besoin de penser d'une manière tout a fait différente en ayant comme souci de bénéficier des possibilités offertes par Matlab.

Pour se faire il faut garder toujours en tête d'écrire une même instruction pour une grande masse de données. Autrement dit, il faut faire comme si on programmait sur une machine vectorielle qui supporte le parallélisme de donnée (dont le principe est illustré sur la figure 46).

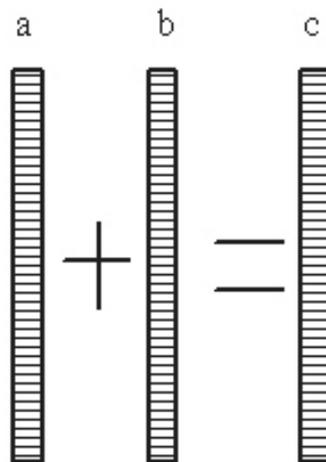


Figure 46. Parallélisme de donnée

Exemple de la détection de contour :

La détection de contour est une opération souvent utilisée dans les algorithmes de traitement d'image ; dans notre cas, la détection de contour est une phase première pour permettre la détection de coins (voir figure 47).

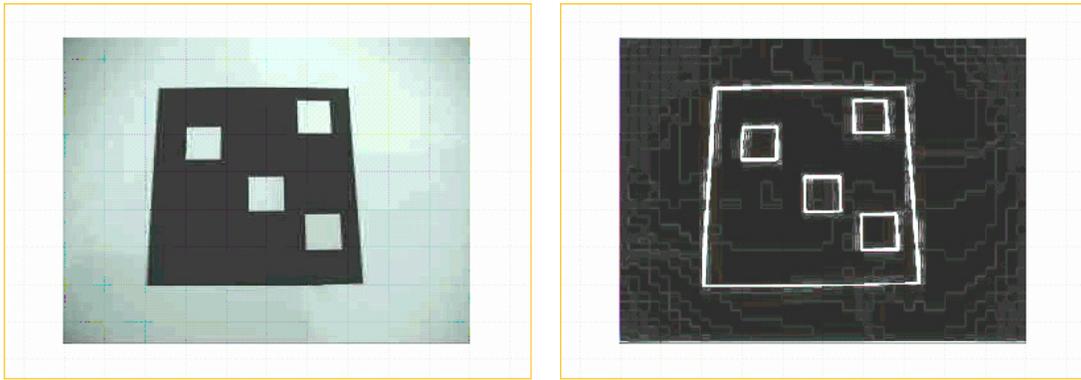


Figure 47. Détection de contour

Voici le programme de détection de contour avant et après prise en compte des fonctionnalités de Matlab :

Programme de détection de contour classique	Programme de détection de contour avec prise en compte des fonctionnalités de Matlab
<pre> for L=2:NBL-1 for C=2:NBC-1 gx(L,C) = (I(L-1,C-1)- I(L1,C+1))+(I(L,C-1)- I(L,C+1))*2.0+(I(L+1,C-1)- I(L+1,C+1)); gy(L,C) = (I(L+1,C-1)-I(L-1,C- 1))+(I(L+1,C)-I(L- 1,C))*2.0+(I(L+1,C+1)-I(L1,C+1)); g(L,C) = sqrt (gx(L,C)^2 + gy(L,C)^2); end end </pre>	<pre> gx=conv2(Id,maskx,'same'); gy=conv2(Id,masky,'same'); g=sqrt(gx.^2+gy.^2); </pre>

Résultat expérimental :

La détection de contour sur une image de résolution 320 * 200 pixels et avec un Pentium 3 ayant un processeur de 500 MHZ :

Dans le premier cas le programme tient sur **44.3740** seconde

Dans le deuxième cas il ne prend que **0.2210** seconde !!!

1.20.2 Solution matérielle

Les limites hardware ont une grande part dans le ralentissement des calculs, donc une solution s'impose, le traitement en parallèle.

Il sera question de décomposer le programme sur plusieurs modules et exécuter chaque module sur un processeur à part.

Nous allons exécuter les modules un par un, pour avoir une idée de ce que peut être le gain en temps d'exécution.

On divise l'image sur quatre parties et on applique la détection de contour sur chaque partie à part et en parallèle puis on aura à fusionner les résultats à la fin (figure 48).

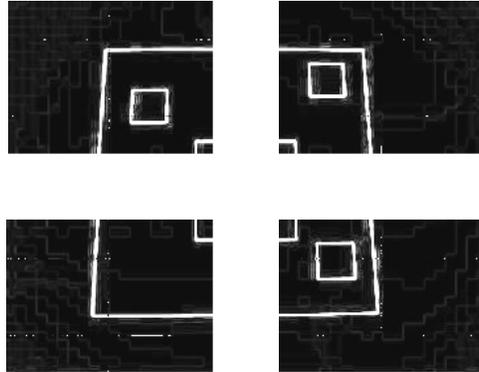


Figure 48. Exécution de la détection de coins par module

Résultat expérimental (pour la même image 320*200 et avec un P3 500 MHZ)

- Le temps d'exécution pour chaque module est de **0.0600** seconde, qui est un temps appréciable pour une vitesse vidéo.
- Comme on peut le noter, dans la pratique le temps d'exécution sur le quart d'une image sera plus grand que le quart du temps d'exécution sur l'image complète ($0.06 > 0.2210/4$).

1.20.3 Cas de figure particulier

Nous allons discuter dans ce qui suit les contraintes à prendre en considération avant de procéder au parallélisme.

Le passage d'un programme séquentiel à un autre parallèle n'est pas direct, il est toujours nécessaire de s'assurer de l'équivalence du programme original avec le programme résultant.

Si on veut par exemple, introduire un parallélisme d'instructions il faut impérativement que ceux-ci soit indépendants pour pouvoir décomposer le programme sur plusieurs modules.

En théorie la vérification formelle demande des outils mathématiques performants et plus ou moins complexes. Ce qui est à lui seul un domaine à part entière.

A notre niveau et dans la pratique, la vérification de l'équivalence des programmes peut se faire de manière plus intuitive.

Voici quelques cas de figures particuliers :

1.20.3.1 Détection de coin

La détection de coin est un processus très important dans le domaine du traitement d'image, il sert entre autre chose à localiser les objets dans leurs scènes.

Dans notre système d'augmentation, nous avons utilisé le détecteur de coins Harris [17] d'une part, pour localiser le modèle et d'autre part pour le suivre par la suite à travers le filtre de Kalman.

A première vue l'algorithme de détection de coins est constitué d'instructions indépendantes qui s'appliquent à chaque fois sur la matrice image en entier, ce qui laisse à penser qu'on peut sans crainte partager le traitement sur plusieurs modules. Mais voyons de près sur la figure 49, le résultat d'exécution de la détection de coins sur l'image originale puis sur des parties séparées de celle-ci.

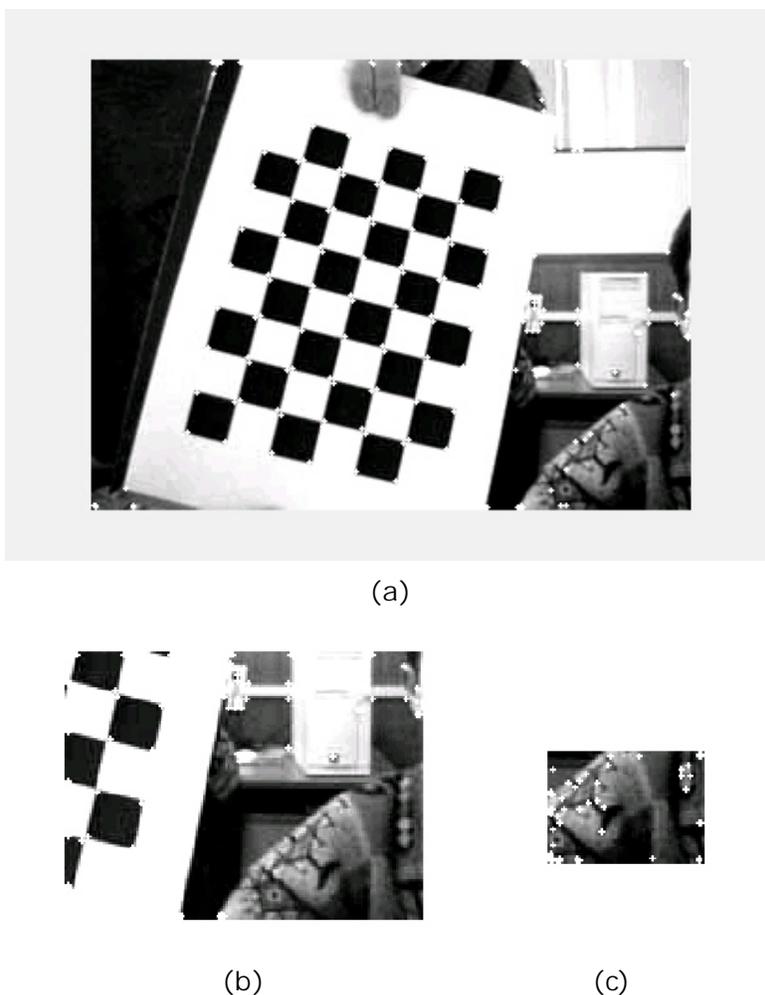


Figure 49. *Impact du choix de la partie d'image sur la détection de coins*

(b) moins de coins détectés que (a)

(c) détection de plus de coins que (a) et (b)

Il est très clair que les coins détectés varient suivant la zone sur laquelle s'applique l'algorithme ; dans la figure 49.b il y a moins de coins détectés que dans la figure 49.a, et dans la figure 49.c il y a plus de coins que dans la figure 46.a et la figure 49.b.

Ceci s'explique par le fait que l'algorithme se base, pour déterminer un coin, sur le calcul d'une mesure, cette mesure à son tour dépend du contexte et du voisinage du pixel en question, ce qui veut dire que l'algorithme de traitement de l'image par partie n'est pas équivalent à celui du traitement de l'image toute entière.

1.20.3.2 Filtrage

Le bruit ennemi du signal image est un processus aléatoire qu'on ne peut pas prévoir. Or nous pouvons faire la constatation suivante :

Une image possède une certaine redondance spatiale ; des pixels voisins ont généralement les mêmes, ou pratiquement les mêmes caractéristiques.

Et nous pouvons définir le bruit dans une image comme un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins.

Ces considérations nous amènent 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.

Les méthodes parmi les plus utilisées, comme il a été expliqué dans le premier chapitre, sont les techniques dites de **filtrage** de l'image.

Une méthode simple consiste à considérer chaque point de l'image et d'en faire la moyenne avec les huit pixels qui lui sont voisins.

Ceci aura pour effet d'adoucir l'image en réduisant les fluctuations des niveaux de gris. **[33]**

Ce filtre utilise un masque H tel que :

$$H = 1/9 \begin{pmatrix} \text{III} \\ \text{III} \\ \text{III} \end{pmatrix}$$

On déplace 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 :

II : l'image de départ.

IF : l'image résultat.

On a :

$$I_F(x,y) = 1/9 \sum_{i=-1}^1 \sum_{j=-1}^1 H(i+1, j+1) \bullet I_I(x+i, y+j)$$

Soit pour chaque pixel de coordonnées x,y :

$$I_F(x,y) = 1/9 \{ I_I(x-1,y-1) + I_I(x-1,y) + I_I(x-1,y+1) + I_I(x,y-1) + I_I(x,y) + I_I(x,y+1) + I_I(x+1,y-1) + I_I(x+1,y) + I_I(x+1,y+1) \}$$

Si on suppose que le bruit a été corrigé par cette méthode, pourquoi ne pas en profiter des pixels déjà corrigés pour poursuivre la procédure ? L'algorithme devient ainsi:

$$I_F = I_I$$

$$I_F(x,y) = 1/9 \{ I_F(x-1,y-1) + I_F(x-1,y) + I_F(x-1,y+1) + I_F(x,y-1) + I_F(x,y) + I_F(x,y+1) + I_F(x+1,y-1) + I_F(x+1,y) + I_F(x+1,y+1) \}$$

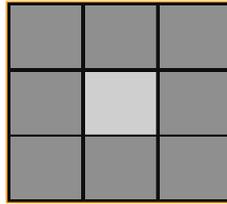


Figure 50. Principe du filtrage

Manifestement, l'esprit du nouvel algorithme est séquentiel, on aura à utiliser les nouvelles valeurs de points voisins pour calculer le point en question, donc à première vue, il ne sera pas possible de paralléliser, mais dans la pratique le résultat des deux algorithmes est presque identiques.

Nous avons procédé à la décomposition de l'image sur quatre parties et comparer le résultat après fusion avec le filtrage de l'image entière.

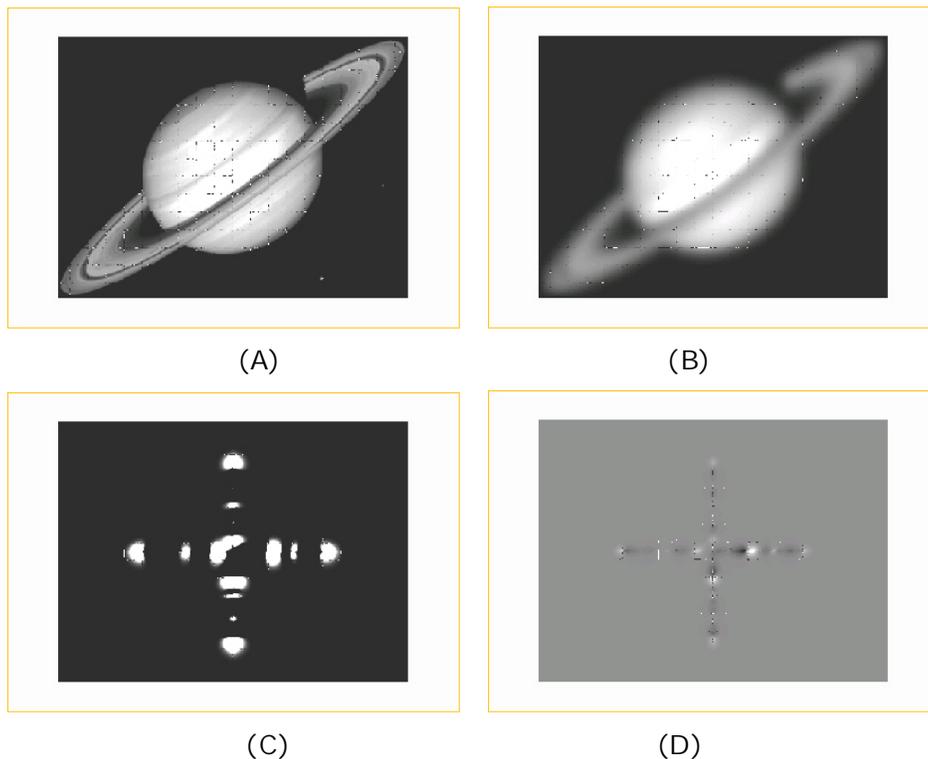


Figure 51. Visualisation de la légère différence entre les résultats du filtrage sur l'image entière et après partition de celle-ci sur quatre parties

(A) image originale

(B) filtrage successif de l'image originale

(C) image différence entre le résultat du filtrage sur l'image entière et celui du filtrage de l'image par partie (le noir pour les pixels similaires, le blanc pour les différents)

(D) image différence suivant les nuance de gris pour visualiser l'écart entre les pixels par l'intensité en niveau de gris

Mathématiquement, les algorithmes ne sont pas équivalents mais dans la pratique les résultats des deux algorithmes sont pratiquement les mêmes.

1.21 Conclusion

D'après tout ce qui a été dit dans ce chapitre, nous tirons les conclusions suivantes:

- Dans le domaine du traitement d'images en général et particulièrement dans notre application, le besoin d'optimiser les programmes est plus qu'incitant.
- Pour réaliser une application de réalité augmentée temps réel sur de simples ordinateurs de bureau, il sera approprié de viser la catégorie temps réel souple.
- La validation de notre application temps réel, se fait comme dans toute application temps réel souple, en raisonnant en termes statistiques sur le temps de réponse moyen.
- Un passage de l'algorithme séquentiel vers un autre parallèle n'est pas automatique et nécessite une réflexion sur la sémantique de l'algorithme lui-même.
- Pour paralléliser le traitement sur une image, il sera plus adéquat de procéder sur les données plus que sur les instructions.
- Puisque nous visons de faire une application de suivi sur de simples ordinateurs de bureaux (PCs), un parallélisme physique ne sera pas envisageable, toutefois l'étude des techniques du parallélisme pourra aider à optimiser le programme surtout si on envisage de faire une application multiposte.

Résultats expérimentaux et discussions

Dans cette section nous allons discuter l'efficacité du suivi et la robustesse de la méthode proposée puis le réalisme de l'augmentation et on terminera par l'exposition du caractère temps réel de l'application du suivi.

1.22 Suivi

L'un des atouts de notre méthode est l'utilisation du filtrage de Kalman pour effectuer le suivi du modèle le long d'une séquence vidéo.

En dépit de la complexité relative de son origine probabiliste, le filtre de Kalman est simple et souple à utiliser. Simple du fait que l'utilisateur n'a qu'à projeter les paramètres du filtre sur le problème à résoudre et souple dans le sens où on peut toujours l'adapter pour répondre aux besoins particuliers du problème traité avec le réglage des paramètres covariance du bruit du modèle (R) et celle de la mesure (Q).

Nous avons pu constater que le suivi par Kalman est très satisfaisant. En effet l'efficacité du filtre est due au fait qu'il est plus qu'un algorithme, c'est un outil mathématique démontré théoriquement.

La figure 52 montre que le système réussit, à travers le filtre de Kalman, à suivre avec précision un point bien défini le long d'une séquence vidéo.

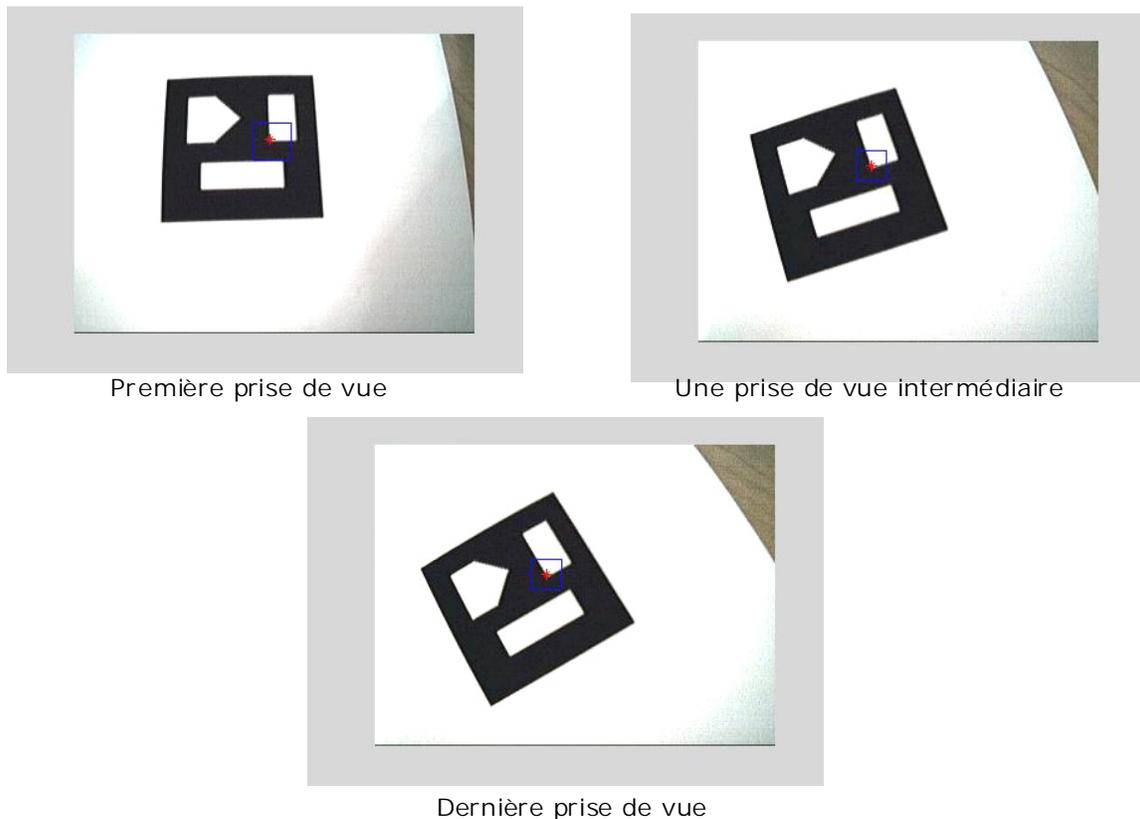


Figure 52. *Visualisation du processus de suivi d'un point à travers le filtrage Kalman*

Un outil qui exprime mieux le processus de suivi est le graphe visualisant la position effective du point à suivre par rapport à celle prédite par le filtre de Kalman. Un tel outil peut représenter une aide pour le réglage du filtre (figure 53).

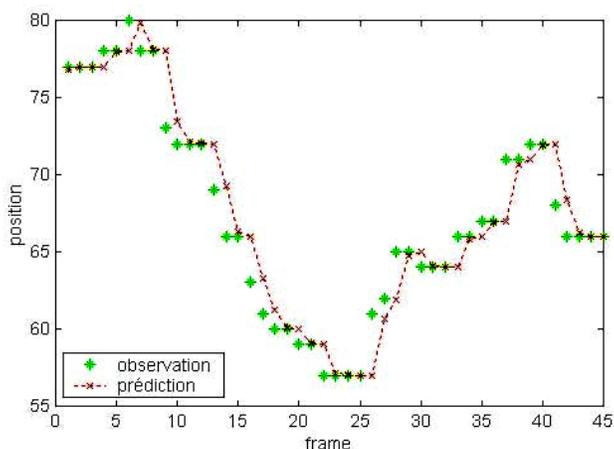


Figure 53. *Visualisation du processus de suivi d'un point par le filtre de Kalman*

On voit bien sur la figure 53 qu'au niveau de chaque prise de vue de la vidéo, la position prédite par le filtre de Kalman d'un point ne diffère guère de sa position effective.

Réglage du filtre

Comme se que prévoit la théorie, nous avons pu expérimenter que la fenêtre locale de recherche doit être suffisamment petite pour ne pas se tromper de point au cours de la recherche, et en même temps relativement grande pour permettre une plus libre dynamique du mouvement du point à suivre.

Le réglage du filtre de Kalman a un impact direct sur la taille de la fenêtre locale de recherche, ce qui rend cette tâche relativement délicate. En effet, en diminuant la taille de la fenêtre locale de recherche, nous limitons l'amplitude autorisée du mouvement, d'autre part si nous tentons de l'augmenter le système risque de prendre le point à suivre pour un autre au fil du processus de suivi.

La figure 54 montre qu'avec une fenêtre trop grande, le système se trompe de point à suivre et finit par 'traquer' (suivre) un autre.

Alors que dans la figure 55 on assiste à une perte de points due à une taille trop petite de la fenêtre de recherche.

Si on ne prévoit pas de mécanisme de récupération, ces anomalies lorsqu'elles se produisent seront tout à fait critiques et le système ne sera plus en mesure de donner des résultats appropriés.

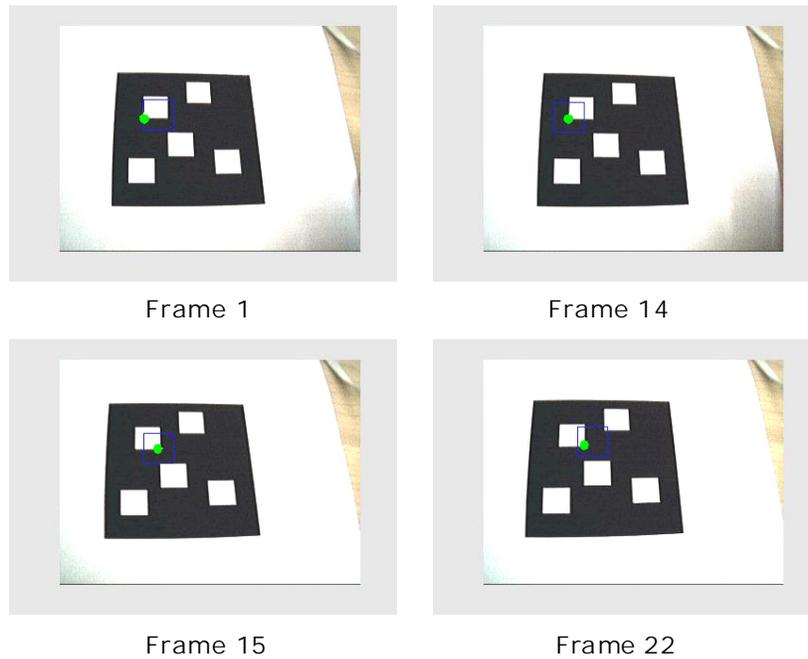


Figure 54. Ambiguïté sur le point à suivre due à une taille de la fenêtre trop grande

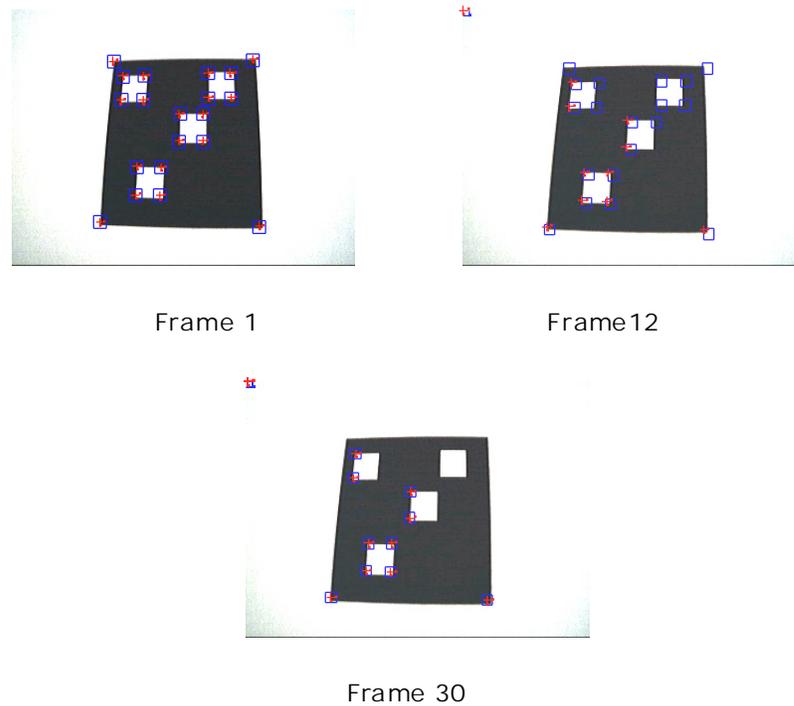


Figure 55. *Perte de points en raison d'une taille de la fenêtre trop petite*

La solution que nous proposons à ce problème est de multiplier les points à suivre. Comme ça nous aurons, pratiquement, toutes les chances d'avoir toujours les quatre coins nécessaires pour calculer l'homographie qui exprime la

projection. Et dès lors nous pouvons réinitialiser le processus de suivi des points perdus par leurs projections depuis l'image de référence du 'pattern'.

Autrement dit, au lieu de bâtir le processus en entier sur le suivi de quatre coins bien définis nous parions sur l'esprit collectif de tous les coins du pattern.

La figure 56 montre la correction en rouge des points perdus.

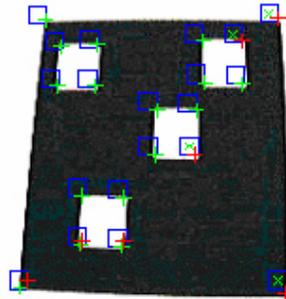


Figure 56. Récupération des points perdus

1.23 Robustesse

La robustesse de notre méthode est due dans une large mesure à l'introduction de l'algorithme de Ransac comme expliqué dans le passage dernier.

En effet nous avons pu apprécier une robustesse intéressante qui se manifeste tout au long du processus d'augmentation que ce soit pour la phase de mise en correspondance ou pour le reste du processus.

1.23.1 Au niveau de la première phase

Le système parvient à détecter dès la première prise de vue de la séquence les mauvaises correspondances et procède à leur correction (point rouge de la figure 57).

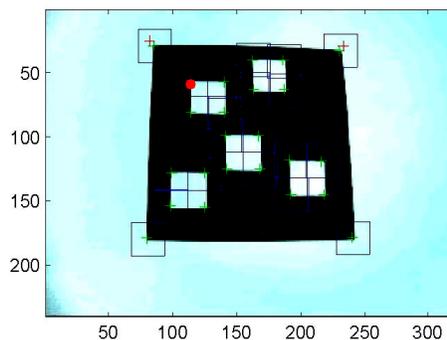


Figure 57. Détection et correction des mauvaises correspondances issues de la phase de mise en correspondance

1.23.2 Au cours du suivi

La perte d'un point au niveau de la phase de suivi n'affecte pas le bon déroulement du processus en entier. Il suffit qu'il y ait 4 points bien matchés (bien mis en correspondance), pour que le système puisse trouver l'homographie et peut ainsi corriger les mauvaises correspondances.

C'est le cas dans la figure 58 des points en forme de 'x' en vert et qui sont des mauvaises correspondances issues de l'occlusion de la main.

Une fois l'homographie bien calculée, il sera possible de corriger les mauvaises correspondances par de simples projections. Les croix en rouge représentent les positions corrigées ; on voit bien qu'ils correspondent effectivement aux coins du modèle mal matchés.

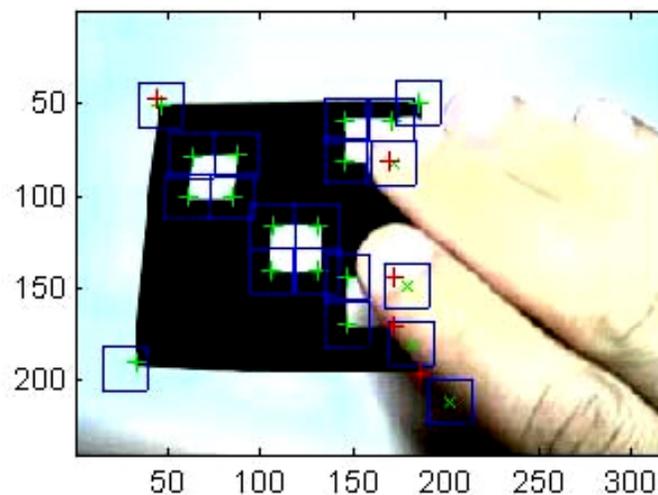


Figure 58. *correction des mauvaises correspondances issues d'une occlusion partielle*

Les croix en vert sont de bonnes *correspondances*, les 'x' en vert sont de mauvaises correspondances et les croix en rouge sont les positions corrigées des mauvaises correspondances

Nous allons dans ce qui suit étudier l'effet d'orientation et celui de l'échelle afin de souligner encore une fois la robustesse de la méthode due à l'introduction de l'algorithme RANSAC.

1.23.3 Effet d'orientation

Nous constatons, expérimentalement, que le système réussit à maintenir son suivi du modèle jusqu'à des angles relativement sévères de celui-ci, soit à peu près 20° et par fois moins.

La figure 59 illustre le suivi d'une vidéo dans laquelle le modèle subit des changements d'orientation dans tous les sens. Le système parvient à suivre le modèle le long de la séquence, en présence de variations d'orientation.

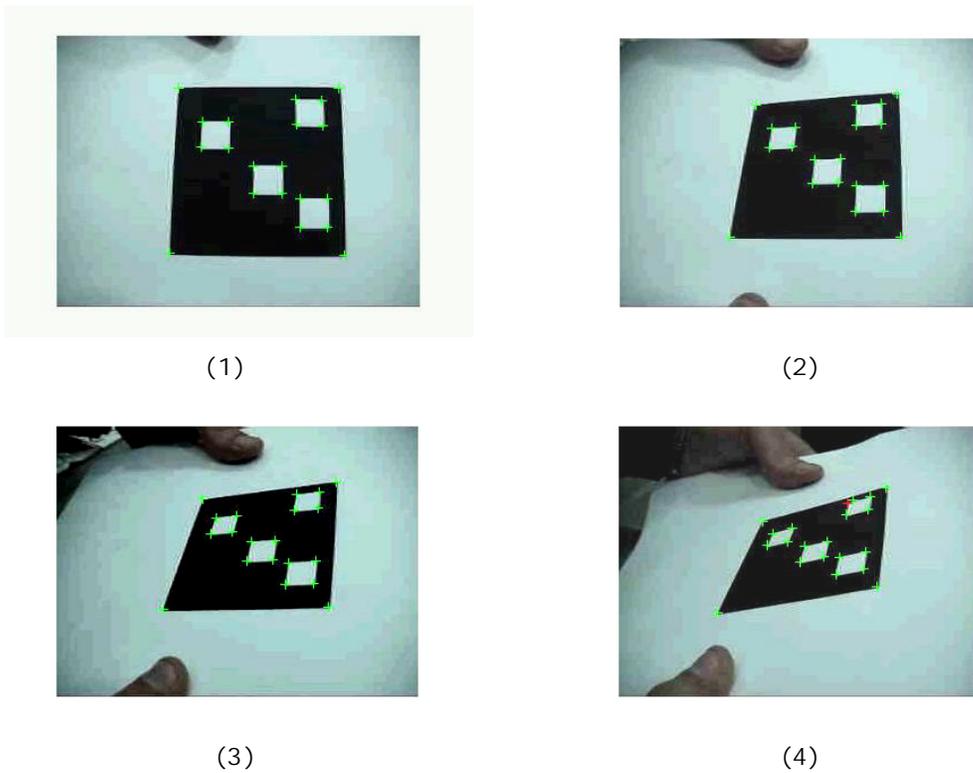


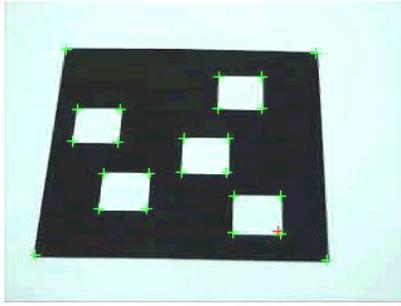
Figure 59. *Suivi du modèle tout au long d'une séquence vidéo de 200 prises de vue, en présence de variations d'orientation*

1.23.4 Effet d'échelle

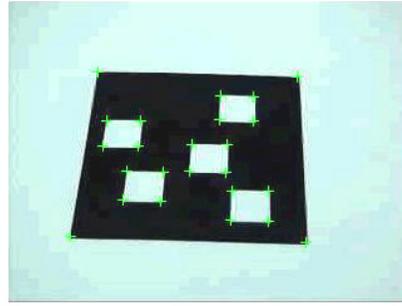
Un changement au niveau d'échelle n'entrave pas le suivi de celui ci. Ceci est dû d'une part à la contribution mutuelle de tous les points du modèle et d'autre part à l'avantage essentiel de l'utilisation des primitives points, qui est l'insensibilité aux variations d'échelle.

Dans La figure 60 on expérimente l'impact du changement d'échelle sur une vidéo de 400 prises de vue ; on fait approcher puis éloigner le modèle de la caméra pour voir si le système réussit à maintenir le modèle ou pas.

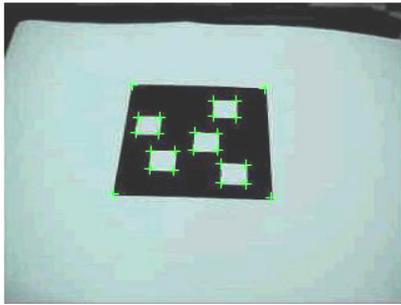
Les résultats expérimentaux montrent qu'en présence d'importants changements d'échelle, le suivi reste robuste et ne lâche plus le modèle une fois trouvé. Remarquons à quel point le modèle s'est éloigné de la caméra dans la dernière prise de vue de la figure 60, il est devenu si petit au point que la fenêtre de recherche locale d'un point peut le contenir en entier, et malgré ça le système reste capable de l'identifier.



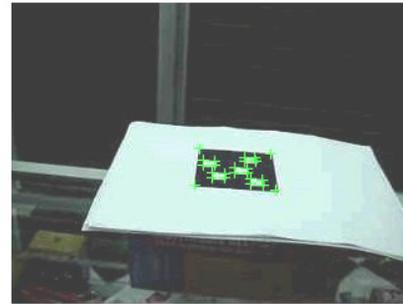
(1)



(2)



(3)



(4)

Figure 60. Effet d'échelle sur le suivi le long d'une séquence vidéo de 400 prises de vue

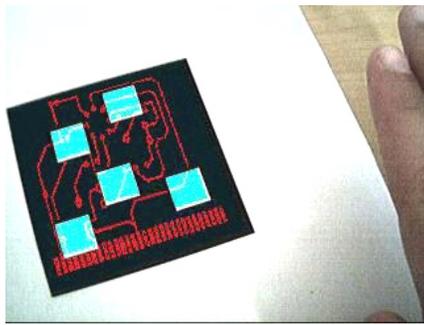
1.24 Réalisme

Le volé qui traite le réalisme de la scène dans notre méthode se répartit sur deux axes :

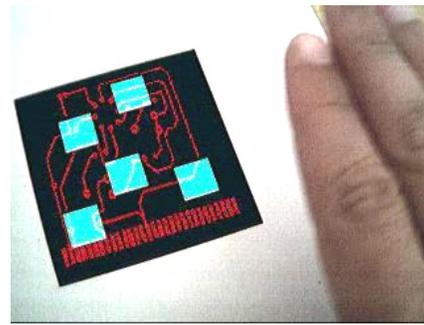
- Tenir compte de la position de l'objet virtuel vis à vis des objets source d'occlusion.
- Faire en sorte que l'objet virtuel apparaisse le plus possible stable sur le pattern.

En ce qui concerne le premier axe nous sommes parvenus globalement à avoir des résultats satisfaisants.

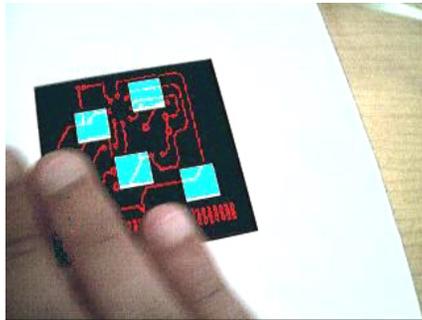
La figure 61 montre une vidéo augmentée avec l'image transparente d'un circuit électronique, et ce, en présence d'occlusion partielle. On note que le système parvient à faire l'identification de l'objet qui est à l'origine de l'occlusion et réussit par conséquent à augmenter la scène de façon à avoir l'impression que le circuit se trouve réellement sous la main.



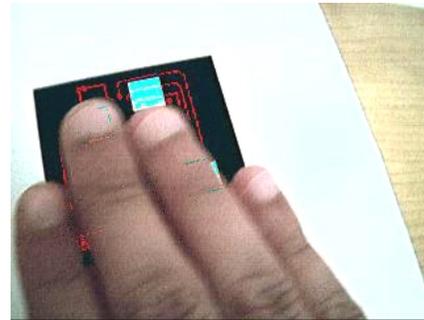
(1)



(2)



(3)



(4)

Figure 61. *Positionnement de l'objet virtuel au dessous de l'objet générant l'occlusion*

Cependant, il est à noter que la méthode présente une sensibilité due au fait qu'il est délicat de trouver un seuillage adéquat pour assurer une parfaite distinction entre les régions du modèle et les régions intruses à celui-ci.

Pour ce qui est du deuxième axe, nous avons pu réduire l'instabilité de l'objet augmenté en adoptant une solution utilisée en intelligence artificielle.

En effet, le calcul d'une homographie dépend des quatre paires choisies. Plus les points sont dispersés, plus l'homographie reflète bien la projection de l'objet virtuel sur le pattern.

Un choix tout à fait aléatoire des quatre points utilisés, comme ce qui se fait dans Ransac se traduit par une instabilité de l'objet virtuel sur la scène réelle, ce qui diminuera de manière significative le réalisme de la scène.

Pour réduire ce problème, on associe une probabilité plus grande pour chacune des quatre paires qui ont induit l'homographie meilleure au sens de Ransac.

Ceci a fait qu'au fil du déroulement de la séquence, le système apprenait et l'incrustation devenait plus en plus réaliste.

1.25 Temps réel

Dans ce qui suit, nous allons discuter la méthode en terme de temps d'exécution sur des ordinateurs de bureaux. On commence premièrement par le suivi d'une séquence d'images sur plusieurs machines à caractéristiques physiques différentes pour voir l'impact de celles-ci sur la vitesse de calcul de notre

application et on termine par donner des mesures détaillées de chaque phase du suivi. Ce qui va nous permettre de conclure sur l'efficacité et le réalisme de notre méthode.

Le tableau suivant (Tableau 1) ainsi que son graphe (figure 62) affiche le temps de suivi de 10 images à différentes résolutions en fonction de la machine sur laquelle la méthode a été appliquée.

	0,48/32	1/128	2,1/96	2,1/128	2,8/256	3,2/512
320/240	4238	2430	1427	1335	961	739
160/120	1162	775	330	325	239	187
80/60	525	328	125	122	88	75

Table 1. Temps de calcul du processus de suivi de dix images en fonction de la résolution et des caractéristiques de la machine *d'exécution*

- Les lignes représentent la résolution en pixel des images à utiliser.
- Les colonnes représentent les caractéristiques de la machine d'exécution ; vitesse du processeur en Giga hertz (Ghz) et capacité de la mémoire en Méga octet (Mo).
- Les intersections des lignes et colonnes représentent le temps d'exécution d'une séquence d'images à une résolution donnée et avec un PC de caractéristiques données. Le temps est mesuré en Milli seconde (ms).

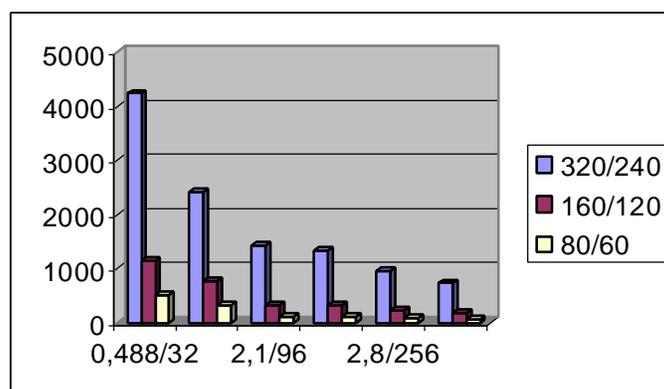


Figure 62. Visualisation graphique du Tableau 1

D'après le graphe ci haut (figure 62) on tire les conclusions suivantes :

1. La limite des 10 images par seconde (nécessaire pour garantir la fluidité de la vidéo) est facilement atteinte avec de simples ordinateurs de bureau ; un ordinateur ayant 2.8 GHz comme vitesse du processeur et 256 Mo de RAM suffit pour atteindre la vitesse requise de suivi d'images avec une résolution de 320/240 pixels.

2. La réduction de la résolution, induit l'augmentation de la vitesse du suivi mais en même temps, une diminution de la précision. Toutefois avec des images 320/240 pixels (et même moins) la qualité est préservée et la précision de calcul est satisfaisante.
3. par une interpolation on peut déduire approximativement le temps d'exécution pour une résolution donnée.
4. les deux caractéristiques qui sont vitesse du processeur et capacité de la mémoire influent sur le temps de calcul du suivi.

Le tableau suivant (Tableau 2) ainsi que son graphe affiche le temps de chaque phase dans le suivi de dix prises de vue pour un Pc (3.2 Ghz/ 512 Mo)

	suivi d'un coin	suivi de 24 coins	Détection de coins
320/240	692	739	470
160/120	156	187	160
80/60	60	75	50

Table 2. Détail du temps de calcul du suivi

- Les lignes représentent la résolution de l'image en pixels.
- Les colonnes représentent les détails du processus de suivi.
- Les intersections représentent le temps d'exécution en milli seconde.

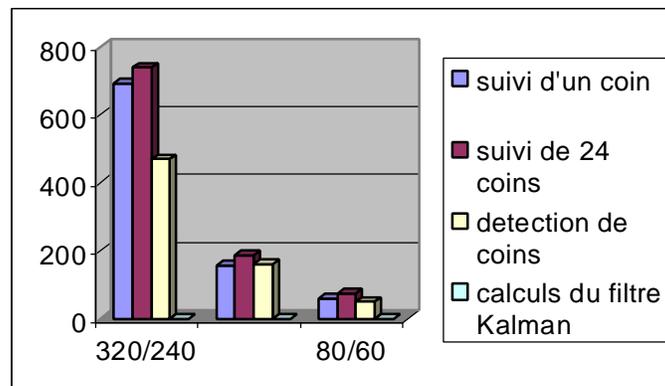


Figure 63. Visualisation du Tableau 2

D'après les résultats obtenus on tire les conclusions suivantes:

1. le temps engendré par les calculs relatifs au filtre de Kalman n'est pas significatif par rapport au coût total du suivi.

2. Dans la pratique, le nombre de coins du modèle n'a pas d'impact sur le temps d'exécution ; par conséquent, suivre un modèle à 24, 20 ou 16 coins n'affecte guère le temps du suivi.
3. à part le temps perdu dans les purs détails de la programmation tel que l'allocation/la libération des images et des matrices, c'est la phase détection de coins qui est la plus gourmande en terme de temps d'exécution.

1.25.1 Conclusion

La méthode du suivi présentée ici manifeste une précision intéressante et surtout un temps de calcul satisfaisant, dans le sens où elle réussit à franchir la limite des 10 images par seconde, et ce sur de simples machines de bureau.

On constate essentiellement que :

- La contrainte temps nécessaire à l'exécution temps réel du suivi est garantie.
- Malgré l'origine probabiliste relativement compliquée du filtre de Kalman, son coût de calcul est négligeable. On gagne donc de la précision due à l'utilisation d'un tel outil démontré mathématiquement et en revanche les calculs engendrés ne sont point pénalisants.
- Il y a une liberté relative dans le choix du modèle à suivre puisque le nombre de coins n'influence guère sur le temps de calcul.

Conclusion et perspectives

Dans ce mémoire nous avons décrit un système de réalité augmentée pour l'insertion d'objets virtuels 2D dans des séquences vidéo.

Nous avons vu que ce système utilise des modèles planaires pour l'augmentation d'objets virtuels, ce qui implique le suivi de ces modèles le long des séquences vidéo. Pour cela la localisation du modèle était nécessaire et elle est effectuée par une mise en correspondance, entre la première image de la séquence vidéo et l'image de référence du modèle.

Pour ce besoin, nous avons utilisé le détecteur de coins HARRIS pour extraire les points d'intérêt des deux images, et l'algorithme de PĪLU pour effectuer la mise en correspondance. Les mauvaises correspondances sont corrigées à l'aide de l'algorithme RANSAC.

Ensuite, pour effectuer le suivi du modèle, nous avons utilisé le filtre de KALMAN qui procède pour chaque prise de vue en deux phases : Prédiction de la position des points d'intérêt et correction de l'estimation faite dans la prédiction, en utilisant des mesures effectuées par le détecteur de coins HARRIS. Ici également, nous avons utilisé l'algorithme RANSAC pour corriger les mauvaises correspondances, ce qui rend le système plus robuste même en cas d'occlusion du modèle.

Nous avons aussi vu comment régler le problème d'apparence de l'objet virtuel en cas d'occlusion.

Toutes ces techniques ont été combinées pour la réalisation d'un système de réalité augmentée assez robuste et avec une augmentation assez réaliste : robuste, notamment envers les changements d'orientation, d'échelle et envers le phénomène d'occlusion, et réaliste, en ce sens où les objets virtuels présentent une stabilité intéressante et sont augmentés en respectant l'ordre de profondeur des objets de la scène.

Une partie de ce mémoire était consacrée à la discussion de la contrainte temporelle ; nous avons pu discuter, avec des exemples, les aspects à prendre en compte pour réduire le temps d'exécution d'un programme de traitement d'image, puis présenter les résultats de l'implémentation du module de suivi dans notre méthode, en matière de temps d'exécution, et ce, en fonction de la taille des images utilisées et des caractéristiques des machines d'exécution, pour conclure finalement sur l'efficacité de l'approche utilisée sur de simple ordinateurs de bureau.

Au court de la préparation de ce projet nous avons pu acquérir beaucoup de connaissances ; premièrement : faire le tour des fondements et techniques de base dans le domaine du traitement d'image, partant des notions les plus élémentaires tel que le filtrage et la segmentation, et arrivant à des algorithmes plus évolués comme la détection de coins et la mise en correspondance de deux images.

Et dans tout ça, nous n'avons pas manqué de renforcer les notions théoriques avec des implémentations concrètes dans des langages comme C, Matlab et C++.

D'autre part, ce projet nous a permis de cerner quelques notions mathématiques évoluées, comme l'homographie, la géométrie projective, la géométrie affine et le filtre de KALMAN.

Les perspectives futures pour notre travail seraient d'améliorer le système pour l'insertion d'objets 3D. pour le cas 3D, le problème est plus compliqué, il s'agit de faire une calibration de la caméra, ou autrement dit le processus qui a pour fin de déterminer la matrice de projection 3D-2D depuis l'espace 3D vers l'image. Ce qui nécessite la connaissance des paramètres intrinsèques et extrinsèques de la caméra.

Mais techniquement, le problème peut se ramener à un problème similaire au cas 2D avec cependant le calcul de la matrice de projection avec six paires au lieu de quatre. Nous sommes déjà parvenus à avoir des résultats dans ce sens.

La figure 64 montre une vidéo d'un modèle 3D qui est augmenté avec un cube 3D.

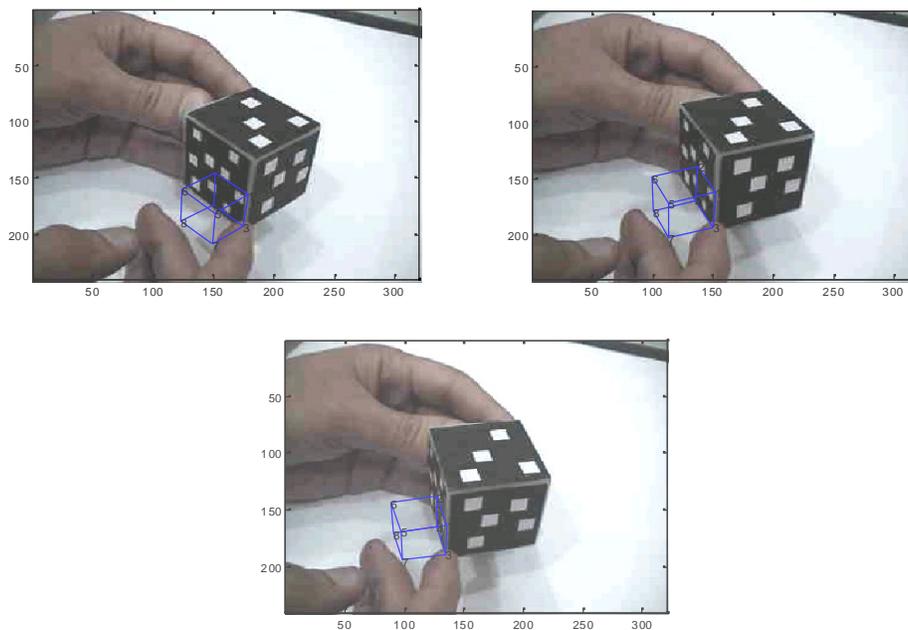


Figure 64. Insertion d'objet 3D dans une séquence vidéo

D'autre part, le module de suivi de notre approche peut très bien servir dans d'autres domaines, tel que la surveillance par exemple. Nous envisageons de l'adapter pour suivre le visage d'un homme comme ce qui est illustré sur la figure 65.

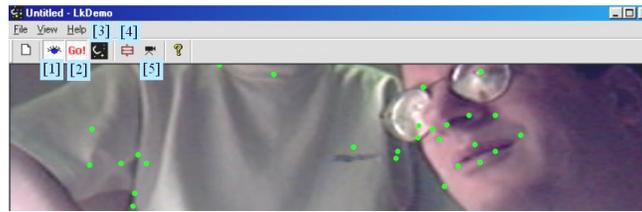


Figure 65. *Suivi à base de détection de coins du visage d'un être humain*

Une partie du travail présenté dans cette thèse a fait l'objet d'une communication dans la « IEEE - 2nd International Conférence on Information & Communication Technologies » qui aura lieu en Syrie le 24-28 Avril 2006, et il a été accepté.

Bibliographies

- [1] M. Badeche, M.L. Berrandjia. "Une méthode robuste pour *l'incrustation d'objets virtuels 2D dans des séquences vidéo*". Mémoire de fin de cycle, Université de Constantine. 2003.
- [2] M.L. Berrandjia, M. Badeche, M. Bouzenada, S.Meshoul. "Une *Méthode Robuste pour l'insertion réaliste d'objets Virtuels dans une séquence vidéo : Application à la fabrication et la maintenance industrielle*". CIP2003. 2003.
- [3] H. Kato, M. Billinghurst, I.Poupyrev. "ARToolKit User Manual, Version 2.33". Human Interface Technology Lab, University of Washington. November 2000.
- [4] G. Klinker. "Augmented Reality: A Problem in Need of Many Computer Vision-Based Solutions". Confluence of Computer Vision and Computer Graphics. Kluwer Academic Publishers, pp 267-284. 2000.
- [5] L. Soler, S. Nicolau, J. Schmid, C. Koehl, J. Marescaux, X. Pennec, N. Ayache. "Virtual Reality and Augmented Reality in Digestive Surgery". Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04), pages 278-279, November 2004.
- [6] S. Nicolau, A. Garcia, X. Pennec, L. Soler, N. Ayache. "Guidage de ponctions percutanées à l'aide d'un système de vision basé sur une méthode de recalage 3D/2D". Actes du colloque Imagerie pour les sciences du vivant et de l'ingénieur (IMVIE03), Strasbourg, 2003.
- [7] R.T. Azuma. "A Survey of Augmented Reality". Teleoperators and Virtual Environments pp. 355-385. August 1997.
- [8] D. Reiners, S. Stricker, G. Klinker, S. Müller. "Augmented Reality for Construction Tasks: Doorlock Assembly". First International Workshop on Augmented Reality, San Francisco. 1998
- [9] K.H. Ahlers, A. Kramer, D.E. Breen, P.Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R.T. Whitaker, D. Greer. "Distributed Augmented Reality for Collaborative Design Applications". Proceedings of Eurographics'95, pp. C_03_C_14. 1995
- [10] K. Satoh, T. Ohshima, H. Yamamoto. "Case Studies of See-Through Augmentation in Mixed Reality Project". First International Workshop on Augmented Reality, San Francisco. 1998
- [11] D. Stricker, G. Klinker, D. Reiners. "A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications".

- First International Workshop on Augmented Reality, San Francisco. 1998
- [12] T.W. Maver, C. Purdie, D. Stearn. "Visual Impact Analysis *Modelling and Viewing the Natural and Built Environment*". Comput. & Graphics, vol. 9 (2), pp. 117_124. 1985.
- [13] C. Chevrier, S. Belblidia, J.C Paul. "Compositing Computer Generated Images and Video Films: An application for Visual Assessment in Urban Environments". Computer Graphics: Development in Virtual Environments, éd. par R.A. Earnshaw et J.A. Vince, pp. 115-125. Academic Press, June 1995.
- [14] J.R. Vallino. "Interactive Augmented Reality". PhD Thesis, University of Rochester, Rochester, NY. November 1998.
- [15] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, B. MacIntyre. "Recent Advances in Augmented Reality". IEEE Computer Graphics and Applications. November/December 2001.
- [16] M.A. Srinivasan, C Basdogan. "*Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges*". Computers & Graphics, Issue 21. pp 393-404. 1997
- [17] C. Harris, M. Stephens. "A combined corner and edge detector", Proceedings of the Fourth Alvey Vision Conference, pp. 147-151, 1988.
- [18] M. Pilu. "Uncalibrated Stereo Correspondence by Singular Value Decomposition". Technical Report HPL-97-96, Digital Media Department, HPLaboratories Bristol, August 1997.
- [19] M.A. Fischler, R.C. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography", Communication Association and Computing Machine, 24(6), pp.381-395, 1981.
- [20] S. Malik, G. Roth, C. McDonald. "*Robust 2D Tracking for Real-Time Augmented Reality*". Vision Interface 2002. Calgary, Alberta, Canada. pp. 399-406. May 2002.
- [21] G. Welch, G. Bishop. "*An introduction to Kalman filter*". SIGGRAPH'2001 course 8, pp.19-33, 2001,
- [22] S. Gilles, "Réalité augmentée temps réel", [Enligne]. http://cch.loria.fr/presentation/CCH_2/pdf/prop_30_SIMON.pdf 24 Janvier 2001.
- [23] T. HALFHILL, "*Embedded market breaks new ground*", Microprocessor Report. 2000.
- [24] G. Logothetis, "*Specification, Modelling, Verification and Runtime Analysis of Real Time Systems*", Dissertation Thesis, University of Kalsruhe. 2004.

- [25] P. Kadionik, "*les systèmes embarqués* : une introduction", [Enligne]. http://www.enseirb.fr/cosynux/grand_public/embedded_intro_2006.pdf. Décembre 2005
- [26] "Système temps réel, Wikipédia" [Enligne]. http://fr.wikipedia.org/wiki/Syst%C3%A8me_temps_r%C3%A9el. (Dernière modification de la page 24 mai 2006).
- [27] "Une définition des systèmes temps réel" [Enligne]. <http://www.rennes.supelec.fr/ren/perso/pchlique/tpsreel/defstr.htm>. (Page consulté le 20 mai 2006.)
- [28] N. Debaci, "Les systèmes multi-agents et le temps réel" [Enligne]. <http://www.limsi.fr/~jps/enseignement/examsma/2005/2.applications/parties/ApplicationAgentsTempsReel.htm#intro>. (Page consulté le 20 mai 2006.)
- [29] I. Ripoll, "Linux Temps Réel (RT-Linux)" [Enligne]. <http://www.tldp.org/linuxfocus/Francais/May1998/article44.html>. (Page consulté le 20 mai 2006.)
- [30] "Définitions et problématique du temps-réel" [Enligne]. <http://david.decotigny.free.fr/rt/intro-ordo/intro-ordo002.html#sec:def-TR>. (Page consulté le 20 mai 2006.)
- [31] Introduction aux systèmes temps réel. Ch Bonnet & I. Demeure, Ed Hermes. 1999.
- [32] P. Chlique, "les contraintes de temps" [Enligne]. <http://www.rennes.supelec.fr/ren/perso/pchlique/tpsreel/acctr.htm>. (Page consulté le 20 mai 2006.)
- [33] J.J. Toumazet, "*Traitement de l'image sur micro-ordinateur*", Cybex. p. 75-95. 1987.

Annexe : Réalisation du système

L'étape de la réalisation du système est importante car elle permet de définir les besoins et de décrire le système de façon générale.

L'approche de description qu'on va utiliser, est une approche GL (Génie Logiciel). On commence par la spécification de chaque partie du système avec ses sous composants, en précisant les entrées et les sorties de leurs modules, puis on décrit le fonctionnement du système et le passage des données d'un module à un autre via le DFD (diagramme de flux de données) et enfin on donne la conception globale du système en utilisant le SD (structure diagram).

Spécification des besoins

Notre travail consiste en la réalisation d'un système de réalité augmentée qui permet l'augmentation d'une séquence vidéo avec un objet virtuel 2D.

De là on déduit les entrées de notre système perçu, à ce niveau, comme étant une boîte noire :

- La séquence vidéo du monde réel contenant un modèle bien défini pour accomplir l'augmentation.
- L'objet virtuel à insérer dans la séquence vidéo.

La sortie du système représente le résultat final du processus d'augmentation, à savoir, la séquence originale augmentée par l'objet virtuel.



Spécification des composants

Pour faciliter le travail et lui donner un aspect plus organisé, on décompose le système en plusieurs modules, qui effectuent chacun un travail bien défini. Chaque module utilise des entrées qui peuvent être le résultat d'un autre module ou des données spécifiées par l'utilisateur.

Comme on l'a vu précédemment, notre système se compose principalement de trois modules :

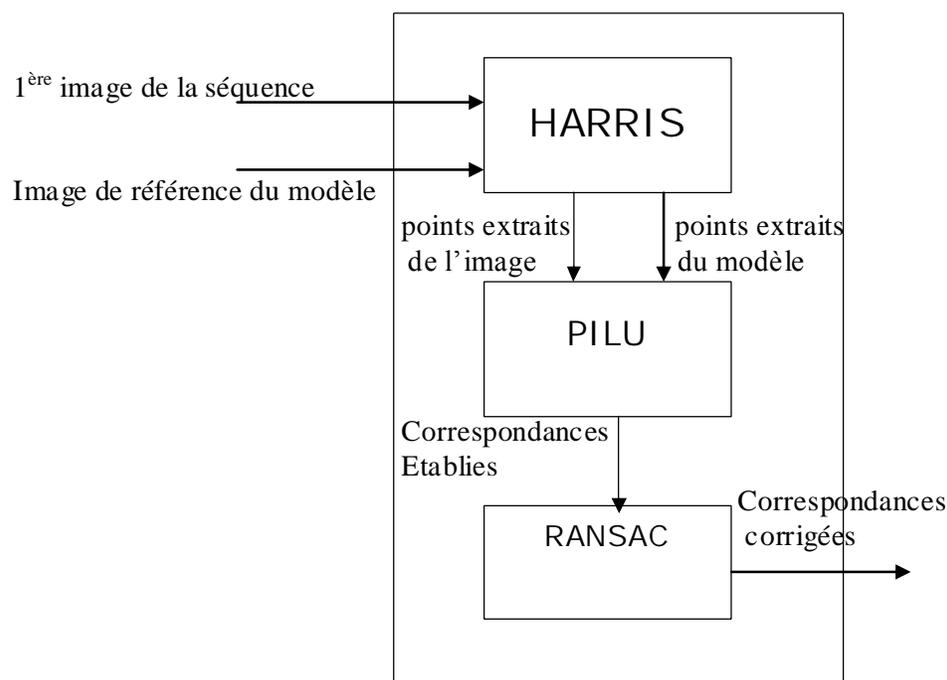
- Module de recherche.
- Module de suivi.
- Module d'incrustation de l'objet virtuel.

Dans ce qui suit nous allons décrire chaque partie en précisant les entrées et les sorties et les sous modules qui les composent.

Module de recherche

Le travail de ce module consiste en la localisation du modèle dans la première image de la séquence. Cela se fait d'abord par extraction des points d'intérêt de la première image de la séquence et de l'image de référence du pattern en utilisant le détecteur de coins HARRIS, ensuite par une mise en correspondance entre ces points en utilisant l'algorithme de PILU. Les mauvaises correspondances seront corrigées par l'algorithme RANSAC.

Cette partie peut être décrite par le schéma suivant :



Module de Suivi

Ce module est le plus important car il permet de suivre le modèle le long de la séquence vidéo afin de l'augmenter avec l'objet virtuel.

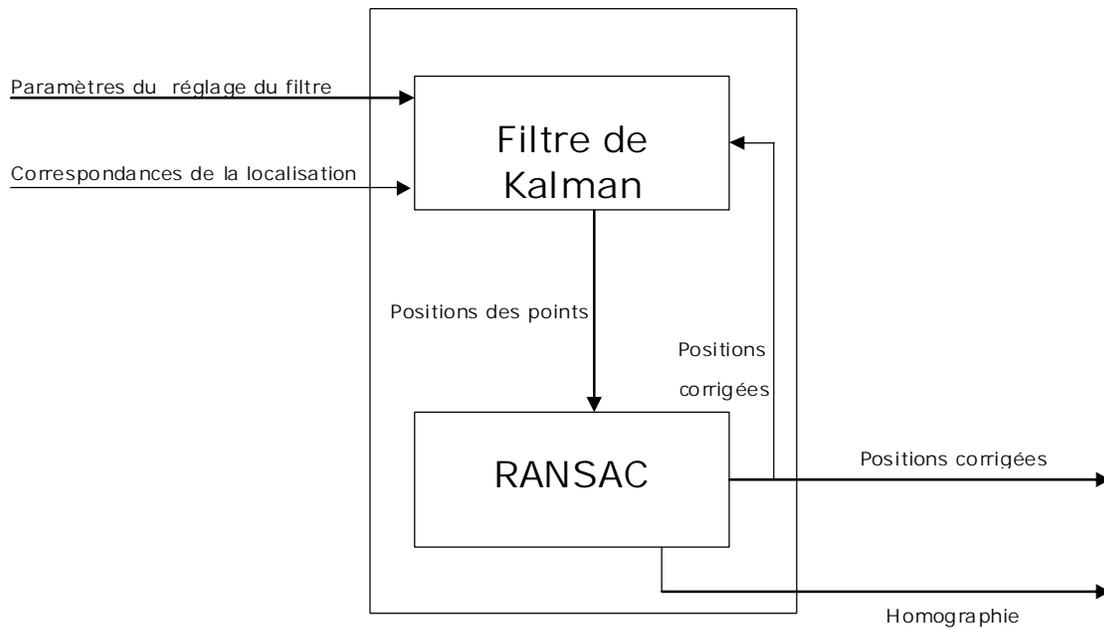
Comme nous l'avons vu précédemment, le suivi s'effectue à base du filtre de Kalman qui procède en deux étapes :

- Prédiction de la position des points d'intérêt.
- Correction de la prédiction avec la mesure effectuée par le détecteur de coins HARRIS.

L'utilisation de l'algorithme RANSAC permet de palier le problème des mauvaises correspondances en cas d'occlusion du modèle. La correction des mauvaises correspondances se fait par une re-projection des outliers en utilisant la meilleure homographie calculée à partir de 4 inliers.

Donc les paramètres d'entrée de ce module sont les correspondances établies dans la phase précédente. Le suivi s'effectue ensuite pour chaque prise de vue en utilisant les positions corrigées par RANSAC. En plus des paramètres précédents, il y a aussi les paramètres du réglage du filtre de Kalman qui doivent être initialisés par l'utilisateur.

A chaque prise de vue, la sortie de ce module est l'homographie permettant la projection de l'objet virtuel sur la scène réelle.



Module d'incrustation

Ce module permet l'insertion de l'objet virtuel dans la séquence vidéo en utilisant l'homographie calculée dans le module précédent.

La méthode d'incrustation consiste à plaquer l'objet virtuel 2D sur la scène réelle par projection de chaque pixel le constituant à travers l'homographie calculée.

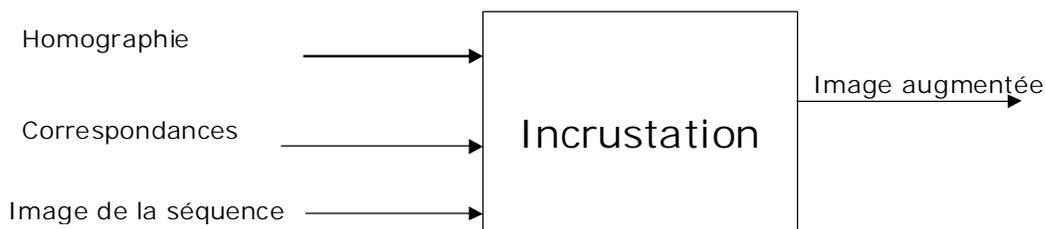
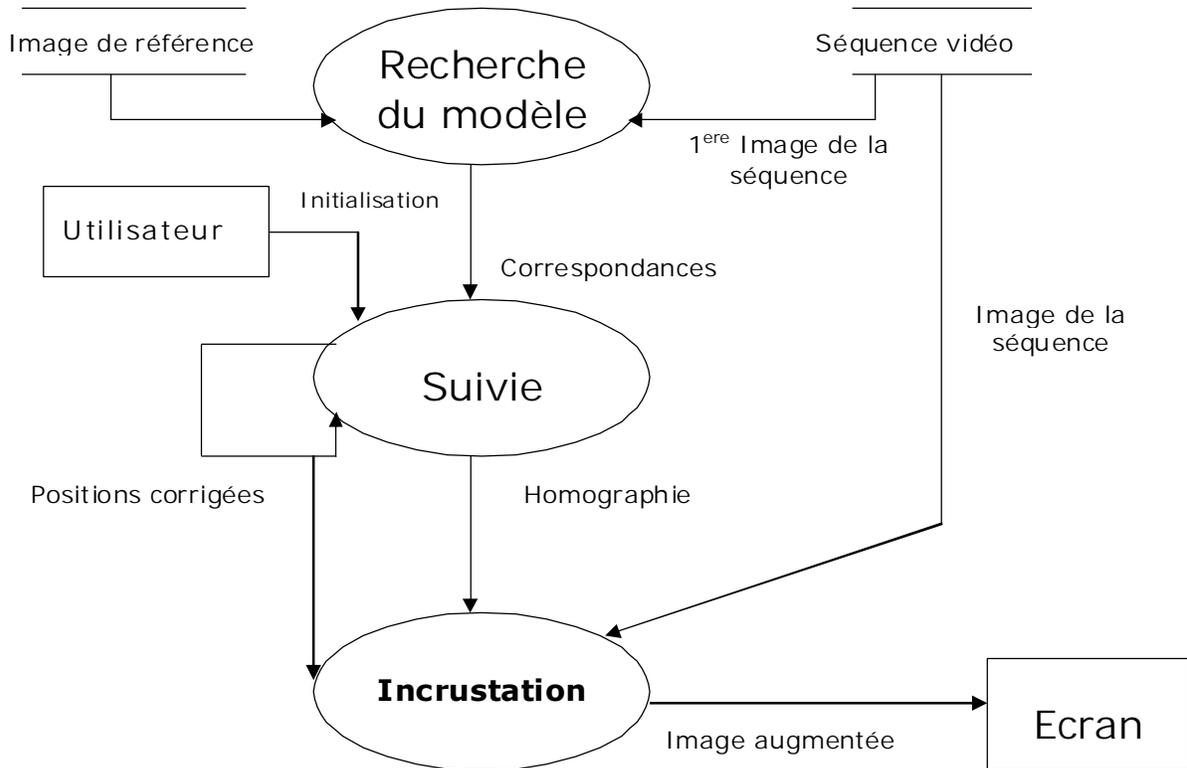
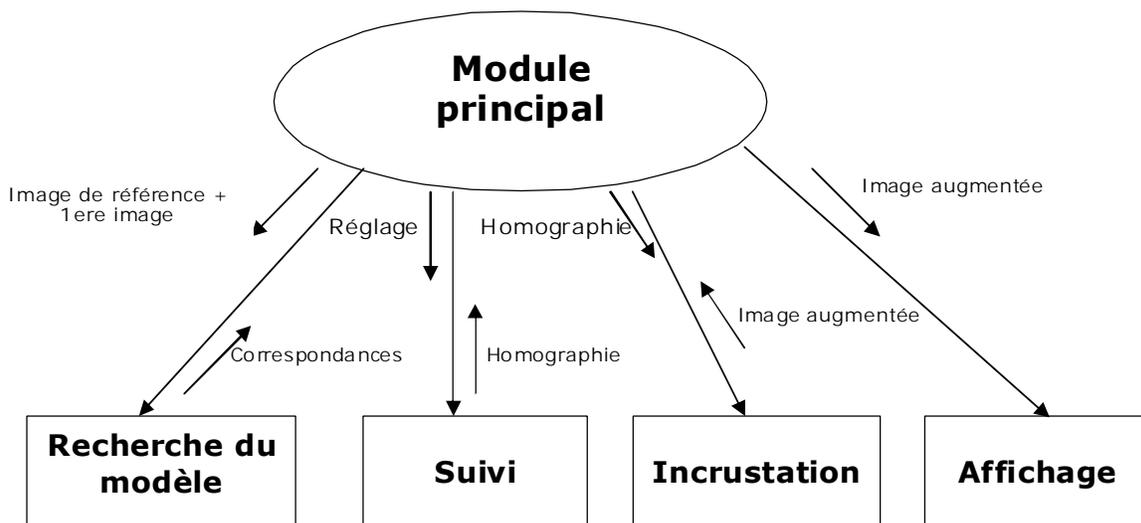


Diagramme de flux de données



Architecture globale du système



Communication

Une partie du travail présenté dans cette thèse a fait l'objet d'une communication dans la « IEEE - 2nd International Conference on Information & Communication Technologies » qui aura lieu en Syrie le 24-28 Avril 2006, et il a été accepté.

L'appel à communication, la lettre d'acceptation ainsi que l'article objet de communication sont à la suite de cette thèse.



CALL FOR PAPERS 2nd International Conference on Information & Communication Technologies: from Theory to Applications - ICCTA'06

(<http://iccta.enst-bretagne.fr/>)

April 24 - 28, 2006, Omayyad Palace, Damascus, Syria



General Chair

B. Solaiman, ENST-Br, France

Honorary Chairs

O. Alansari, Nossitia, Syria
A. Amanazi, GD, SSR, Syria

Technical Committee Co-Chairs

G. Martineau, ENST-Br, France

R. Razouk, Univ. Damascus, Syria

Technical Committee

R. Arda, ESIS, Spain
A. Albas, PUV, Valencia, Spain
F. Algosa, Bogazici Univ., Turkey
M. Aghous, Univ. OF Virginia, USA
I. Azar, HAST, Syria
Z. Belhadj, SupCom, Tunisia
M. Ben Ahmed, ENSI, Tunisia
A. Benjelloun Toumi, FT-R&D, France
M. S. Boudjah, ESIS, Sfax University - Tunisia
D. Bull, Univ. Bristol, UK
N. Canagajah, Univ. Bristol, UK
H.T. Chuah, MMU, Malaysia
P. Coulton, Lancaster Univ., UK
G. M. Dimirovski, Dogus Univ., Turkey
M. Driou, INSA, Rennes, France
G. El Zain, IETR, INSA, Rennes, France
F. Ghaila, UTC, France
P.Y. Glesener, INSA, Rennes, France
X. Gu, TUBS, Braunschweig, Germany
M. Habti, S. Qaboos Univ., Oman
H. Hammanen, TKK, Finland
W. Hamoula, Concordia Univ., Canada
A. Hamou, McGill, Montreal, Canada
JP. Haton, LORIA, France
A. Heston, CTI, Denmark
H. Ishibuchi, Osaka Univ., Japan
J. Jaem, Univ. Qatar
A. Jamalipour, Univ. Sydney, Australia
K. Kayali, Aleppo Univ., Syria
A. Khamchaf, ENSIETA, France
F. Kwon, AEU, Syria
X. Lagrange, ENST-Br, France
R. Lepage, ETS, Montreal, Canada
D. Mamanas, Ibn Zohr Univ., Agadez, Morocco
R. Moller, Univ. Wuppertal, Germany
P. Monasse, ESIA Amey, France
D. Ngo, MMU, Malaysia
V.F. Nicola, EEMCS, Netherlands
M. Papageorgiou, Techn. Univ. Crete, Greece
F. N. Pavlidou, AUT, Greece
C. Phillips, QMUL, UK
A. Pitsillides, Univ. Cyprus
R. Pyskahl, ENST-Br, France
S. Ramos, ETSI Univ. Spain
M. Rocetti, UNIBO, Italy
J. Rodrigues, Univ. Beira Interior, Portugal
J. Roth, Univ. Dusseldorf, Germany
S. Safadi, Univ. Damascus, Syria
H. Sari, Supelec, France
S. Shaban, Cairo University, Egypt
A. Shalabi, ENIB, France
S. Tatroh, Larent Technologies, UK
R. Tezcanli, MRC, Connecticut, USA
R. Trauttmiller, J. K. U, Linz, Austria
F. Ullah, Univ. Michigan, USA
S. N. Venka, Space Technology, USA
H. Wang, Qualcomm, USA
E. Wacziarg, Univ. Paris 11, France
W. W. Wu, AT&T, USA
M. Zaefer, Lebanese Univ. Lebanon
K. Zouk, Caen Univ., France

Organization Committee Chair

S. Rifai, ASST, asst@net.sy

Organization Committee

S. Shalabi, ASST, Syria
C. Chahadeh, CYC Tech, USA
L. Lecormu, ENST-Br, France
F. Ghorbel, ENSI, Tunisia
B. Alachkar, HAST, Syria
M. Sarfraz, KFUPM, Dhahran, SA

Industrial Liaisons

A. Fakhri, Global Com., amir@globalcom.ae

Tutorials

A. Midani, SI Paris, France

Publications

S. Rifai, ASST, Syria, asst@net.sy

Local Arrangements

R. Rifai, ASST, Syria, ramrifi@hotmail.com

Registration - Lodging

N. Nasser, Nossitia, nossitia@scs-net.org
Tel: 963-11-6127482, Fax: 963-11-6127483

The International Conference on Information & Communication Technologies: from Theory to Applications - ICCTA'06 - will be held from April 24 - 28, 2006 in Damascus, Syria. Through its technical program, the conference aims to provide an outstanding opportunity for *both academic and industrial communities alike* to address new trends and challenges, emerging technologies and progress in standards on topics relevant to today's fast moving areas of Information and Communication Technologies. ICCTA'06 will offer oral, poster sessions, tutorials and professional meetings. Submitted papers are expected to cover state-of-the-art technologies, theoretical concepts, standards, products implementation, ongoing research projects and innovative applications of the IC Technologies use.

TOPICS

Topics of interest include, but are not limited to:

Networking theory & technologies
Network Management and services
Broadband & Intelligent networks
3G&4G Mobile Communication Services
Mobile networks & services
Next generation network
QoS management
Information systems & Applications
Data Base Management
Knowledge Management & Decision Making
Internet applications & performances
Web Engineering
Emerging technologies & Applications
E-Learning & E-Business
E-Commerce & E-government
E-Health & Biomedical applications
Agents and Multi-Agents systems for ICT
Integrated Circuits for communications
Antennas & Propagation

Wireless & Wireline Communications
Communication Systems
Satellite & Space Communications
Optical Communications
Information & data security
Knowledge Based Systems
Knowledge Management
Information Processing
Data Mining-Data fusion
Systems & Software Engineering
Signal & Image Processing
Information indexing & retrieval
Machine Vision & Remote sensing
Image & Multimedia applications
Fuzzy, ANN & Expert Approaches
ICT & Education
ICT in Environmental Sciences
ICT & Banking
ICT & Intelligent Transportation

SATELLITE WORKSHOPS & TUTORIALS

ICCTA'06 organizers encourage scientists & industry leaders to organize and participate in special satellite workshops, tutorials and panel sessions. A session proposal should indicate the topic and identify invite moderators and session technical papers.

PAPERS SUBMISSION

Papers will be reviewed on the basis of two pages extended abstracts of sufficient detail to permit reasonable evaluation. The final version of accepted papers should be six pages long. Author kit and submission details are available on the conference's web page <http://iccta.enst-bretagne.fr/>.

IMPORTANT DATES

Abstract submission due: December 31, 2005

Notification of Acceptance: January 31, 2006

Final Camera Ready Papers due: March 5, 2006

Venue: The ICCTA'06 conference will take place in Omayyad Palace/Damascus. A social program will be organized to allow visiting the oldest city in the world (Damascus) and some imprints of the world's oldest civilizations (the city of Bosra, mentioned in the lists of Tutmosé III and Akhenaten in the fourteenth century B.C., is also the first Nabatean city in the 2nd century B.C., will be visited).

FURTHER INFORMATION

Pr. B. Solaiman,
ENST-Br, Brest- France,
Tél. 33 (0)2.29.00.13.08 / 33 (0)6.30.51.38.34,
<http://iccta.enst-bretagne.fr/>
iccta@enst-bretagne.fr

S. Rifai, ASST, Damascus,
asst@net.sy, Tel: 963-11-5118904, Fax: 963-11-5111083
N. Nassr, NOSSITIA, Damascus
nossitia@scs-net.org
Tel: 963-11-6127482, Fax: 963-11-6127483



Brest, 27 February 2006

Object: ICTTA'06 Acceptance Certificate & Invitation Letter

Dear colleague,
It is my great pleasure to inform you that the following paper:

Paper Title: **Real Time 2D model tracking in a PC augmented reality application**

Paper Assigned Ref.: **ICTTA097**

Authors: **M. BADACHE and M. BENMOHAMMED**

Affiliation: **Département d'informatique, Université El Hadj Lekhdar, Batna, Algérie**

has been **accepted** in the IEEE - 2nd International Conference on Information & Communication Technologies: From Theory to Applications (ICTTA'06) to be held in Damascus-Syria (April 24-28, 2006).

I would like to congratulate you for this acceptance, certainly due to the high quality of your contribution, and would like to inform you that the papers acceptance rate in ICTTA'06 is 59 % only. Hence, **we kindly invite you to take part in the Conference ICTTA'06 and look forward to seeing you in Damascus and, to attend the presentation of your excellent contribution.**

Best regards,
Pr. B. Solaiman
ICTTA'06 General Chairman

Le Chef du Département
Image et Traitement de l'Information



Basel SOLAIMAN

