

بيت سترج
17/004.4 17

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

12/ 805

Filière : Informatique

Option : Ingénierie des Medias

Thème :

**INSERTION EN TEMPS RÉEL D'UN OBJET VIRTUEL DANS
UNE SEQUENCE VIDÉO POUR LA RÉALITÉ AUGMENTÉE**

Encadré Par :

Mr. Hallaci Samir

Présenté par :

Chirouf Hichem

Moussaoui Yehya Housseyn

Juin 2012

Remercîments

En premier lieu, je tiens à exprimer ma profonde gratitude à mon encadreur Mr Hallaci Samir de m'avoir proposé ce sujet et diriger constamment de près mon travail. Je le remercie pour sa disponibilité, ses encouragements, son soutien moral, sa sympathie et son amitié sincère dont il a fait preuve le long de ce travail. J'apprécie la confiance qu'il m'a témoigné et les conseils avisés qu'il m'a prodigué.

Un grand merci aux membres de jury qui nous a honorés d'assister à cette soutenance remercie pour l'intérêt qu'il a porté à mon travail. Je ne serais sans doute pas arrivé jusqu'ici sans avoir suivi les cours d'enseignants exceptionnels qui ont participé à ma formation, je pense en particulier aux enseignants du département informatique de Guelma sans oublier ceux qui travaillent au centre de calcul.

Je remercie toute l'équipe des collègues de la promotion d'informatique pour l'amitié et le soutien qu'ils m'ont apporté durant la période d'étude et de préparation de ce travail.

Je ne saurais terminer ces remerciements sans un énorme merci à mes parents qui me soutiennent et m'encouragent depuis toujours et qui m'ont accompagné jusqu'à la soutenance de ce mémoire.

Résumé

Le Projet de fin d'étude proposé se place dans le cadre de la réalité augmentée, entre la vision artificielle et la synthèse d'images. Par système de réalité augmentée, on entend un système qui rend possible l'insertion d'une personne dans un monde virtuel, ou l'insertion d'objets virtuels 2D ou 3D dans des scènes du monde réel en temps réel, et c'est notre cas.

Puisque ce type d'applications demande énormément de réalisme, il est indispensable que l'ajout d'objets dans une scène ne perturbe pas la cohérence du contenu filmé. Le fait de déplacer la caméra implique cependant un mouvement dans l'image de la scène filmée. Pour assurer la cohérence entre les deux flux réels et virtuels, un lien rigide doit être maintenu entre les deux mondes. Afin de donner l'illusion que ces objets fictifs appartiennent au même monde, il est nécessaire de bien les placer, bien les orienter et de respecter des facteurs d'échelle par rapport aux objets réellement filmés. Ceci est possible, grâce à un bon suivi de la position et de l'orientation des objets réels de la scène.

Le travail achevé peut insérer des images de synthèse sur les images du monde réel, en se basant sur le suivi temps réel d'un motif visuel qui est préalable pour l'insertion d'objets virtuels,

Ainsi on a achevé une partie de suivi d'objets puisqu'en plus de son importance, elle représente la partie la plus consommatrice de temps.

Mots Clés :

Réalité augmentée, détection de mouvement, Suivi temps réel, vision par ordinateur, Détecteur de points d'intérêts, Mise en correspondance.

Sommaire

<i>Liste des figures</i>	Page 05
--------------------------------	---------

<i>Introduction Générale</i>	Page 07
------------------------------------	---------

Chapitre I : La vision par ordinateur

1. Introduction.....	Page 09
2. Qu'est ce-que la vision.....	Page 09
2.1. Le système de vision.....	Page 10
3. Modélisation de camera.....	Page 11
3.1. Rappels sur la projection.....	Page 11
a. Projection.....	Page 12
b. Perspective.....	Page 13
3.2. Modélisation géométrique.....	Page 14
3.3. Modélisation algébrique.....	Page 15
a. Étape 1 – Prise en compte de la distance focale	Page 15
b. Étape 2 – Prise en compte des pixels.....	Page 17
c. Étape 3 – Prise en compte des déplacements.....	Page 17
d. Paramètres extrinsèques et intrinsèques	Page 18
4. Calibrage de caméra.....	Page 19
5. Extraction de l'information 3D.....	Page 19
5.1. Caractéristiques	Page 19
5.2. Indices de profondeur	Page 20
a. Classes d'indices de profondeur	Page 21
5.3. Exemples.....	Page 21
b. Textures	Page 21
c. Ombrages.....	Page 21
d. Disparités binoculaires.....	Page 22
6. Conclusion.....	Page 22

Chapitre II : Détection et Suivi du Mouvement

1. Introduction.....	Page 23
2. problèmes de détection	Page 23
3. Les différentes méthodes de détection de mouvement.....	Page 24
3.1. Détection sans modélisation de l'arrière-plan	Page 24
3.2. Modélisation locale de l'arrière-plan.....	Page 25
3.3. Modélisation semi-locale de l'arrière-plan.....	Page 27
3.4. Modélisation globale de l'arrière-plan.....	Page 28

4. Estimation et calcul de mouvement.....	Page 29
4.1. Introduction.....	Page 29
4.2. Mouvement réel, mouvement apparent et mouvement estimé.....	Page 30
4.3. Champ de mouvement réel et champ de mouvement apparent.....	Page 30
4.4. Estimation du mouvement.....	Page 31
4.5. Le calcul du mouvement.....	Page 31
5. Différentes approches de suivi.....	Page 32
5.1 Méthodes de suivi 2D.....	Page 32
a. Méthodes de suivi basées sur l'intensité lumineuse	Page 32
b. Méthodes de suivi basées sur les primitives géométriques.....	Page 33
5.2. Méthodes de suivi 3D.....	Page 36
a. Estimation à partir des transformations 2D.....	Page 37
b. Méthodes basées modèle.....	Page 37
5.3. Méthodes hybrides 2D/3D.....	Page 38
6. Conclusion.....	Page 39

Chapitre III : Réalité Augmenté

1. Introduction.....	Page 40
2. Définition.....	Page 40
3. Historique.....	Page 41
4. Domaines d'application.....	Page 43
a. Médical.....	Page 43
b. Militaire.....	Page 44
c. Industriel	Page 45
d. Navigation.....	Page 46
5. Problèmes de réalité augmentée.....	Page 46
5.1. Alignement spatial : erreurs statiques.....	Page 47
5.2. Alignement temporel . temps de latence.....	Page 47
5.3. Éclairage.....	Page 48
5.4. Occultation.....	Page 49
5.5. Flou de mouvement.....	Page 49
6. Architecture générale d'un système de RA.....	Page 50
7. Classification des systèmes de réalité augmentée.....	Page 51

7.1. Fonctionnelle.....	Page 52
7.2. Technique	Page 52
8. Les architectures modulaires existantes.....	Page 53
8.1. COTERIE	Page 53
8.2. StudierStube	Page 54
8.3. Tinmith	Page 54
8.4. DWARF	Page 55
8.5. AMIRE	Page 56
8.6. DART	Page 57
9. Conclusion.....	Page 57

Chapitre IV: Conception et Implémentation

1. Introduction.....	Page 58
2. Conception.....	Page 58
3. Présentation d'ARtoolKit	Page 59
4. Principe de fonctionnement.....	Page 60
4.1 Algorithme de détection	Page 60
a. Binarisation.....	Page 61
b. Recherche des composants connexes	Page 62
c. Extraction des segments et des coins de la cible	Page 62
d. Normalisation.....	Page 64
e. Identification du marqueur.....	Page 65
5. Création du marqueur	Page 65
6. Création d'un objet 3D	Page 66
7. Implémentation.....	Page 67
7.1. Environnement de développement	Page 67
a. Environnement matériel.....	Page 67
b. Environnement logiciel.....	Page 68
c. La bibliothèque utilisée.....	Page 70
7.2 Interface principale.....	Page 70
a. Les tests de l'application.....	Page 70

7.3 Difficultés rencontrées	Page 73
8. Conclusion.....	Page 74
<i>Conclusion générale</i>	Page 75
<i>Bibliographie</i>	Page 77

LISTES DES FIGURES

<i>Figure 1.1 : système de vision</i>	Page 11
<i>Figure 1.2 : Projection</i>	Page 12
<i>Figure 1.3 : Projection géométrique</i>	Page 12
<i>Figure 1.4 : Perspective d'un carré</i>	Page 13
<i>Figure 1.5 : Ligne d'horizon et point de fuite en perspective</i>	Page 14
<i>Figure 1.6 : Points de fuite d'un objet tridimensionnel</i>	Page 14
<i>Figure 1.7 : Le modèle sténopé</i>	Page 15
<i>Figure 1.8 : Les repères utilisés</i>	Page 16
<i>Figure 1.9 : La mire de calibrage</i>	Page 19
<i>Figure 1.10 : traitement avec des Caractéristiques bas niveau</i>	Page 20
<i>Figure 1.11 : traitement avec des Caractéristiques bas niveau + indices de profondeur + carte de profondeur</i>	Page 20
<i>Figure 1.12 : Perception de la forme par la texture</i>	Page 21
<i>Figure 1.13 : Perception de la forme par l'ombrage</i>	Page 22
<i>Figure 1.14 : Exemple de disparités binoculaires</i>	Page 22
<i>Figure 2.1 : Taxinomie proposée pour les méthodes de détection de mouvement</i> ...	Page 29
<i>Figure 2.2 : Illustration des mouvements réels et apparents dans un système optique de prise de vues</i>	Page 30
<i>Figure 3.1 : le projet de Sutherland invente</i>	Page 41
<i>Figure 3.2 : Steve Feiner (la Touring Machine)</i>	Page 42
<i>Figure 3.3 : jeu Quak</i>	Page 42
<i>Figure 3.4 : Un exemple avec les outils développés par l'IRCAD</i>	Page 43
<i>Figure 3.5: Head-Up Display (HUD)</i>	Page 44
<i>Figure 3.6 : voiture avec affichage haute tête</i>	Page 45
<i>Figure 3.7 : un tram dans un modèle de ville</i>	Page 45
<i>Figure 3.8 : la navigation</i>	Page 46
<i>Figure 3.9 : Incrustation d'une voiture virtuelle dans une scène réelle</i>	Page 50
<i>Figure 3.10 : Architecture générale d'un système de RA</i>	Page 51
<i>Figure 3.11 : réalité mixte</i>	Page 53
<i>Figure 3.12 : Flux de données et disposition de l'architecture en couches du système Tinmith</i>	Page 55

<i>Figure 4.1 : étapes d'incrustation d'objets 3D</i>	Page 58
<i>Figure 4.2 : des exemples d'application ARToolKit</i>	Page 59
<i>Figure 4.3 : Modèles de marqueurs utilisés par ARToolKit</i>	Page 60
<i>Figure 4.4 : les étapes de fonctionnement de l'algorithme</i>	Page 61
<i>Figure 4.5 : Binarisation de l'image</i>	Page 62
<i>Figure 4.6 : Recherche des composants par détection de contours</i>	Page 62
<i>Figure 4.7 : Extraction des contours, lignes et coins</i>	Page 63
<i>Figure 4.8 : processus de reconnaissance de marqueur</i>	Page 63
<i>Figure 4.9 : Principe de normalisation</i>	Page 64
<i>Figure 4.10 : exemple de nos marqueurs</i>	Page 65
<i>Figure 4.11 : Exemple de la fenêtre de création d'un marqueur</i>	Page 66
<i>Figure 4.12 : création texte 3D avec 3D studio max</i>	Page 66
<i>Figure 4.13 : imprime écran du diagnostique direct X (système)</i>	Page 67
<i>Figure 4.14 : version de l'éditeur du code</i>	Page 69
<i>Figure 4.15 : interface principale</i>	Page 70
<i>Figure 4.16 : test 1 marqueur avec motif B et un ballon</i>	Page 70
<i>Figure 4.17 : test marqueur avec motif c et un cube</i>	Page 71
<i>Figure 4.18 : test 3 marqueur avec motif E8 et une texte 3D</i>	Page 71
<i>Figure 4.19 : test 4 marqueur avec motif E8 inversé et un texte 3D</i>	Page 72
<i>Figure 4.20 : test 5 avec deux marqueurs</i>	Page 72
<i>Figure 4.21 : test 6 avec trois marqueurs</i>	Page 73

INTRODUCTION GÉNÉRALE

En 1982, le public des salles de cinéma pu découvrir le long-métrage Tron. Ce film était l'un des premiers à mêler de façon intensive des images générées par ordinateur et des séquences filmées. Depuis cette époque, l'industrie cinématographique n'a eu de cesse d'employer et d'incruster des images de synthèse de plus en plus élaborée dans les séquences filmées, afin d'introduire des effets spéciaux supplémentaires qui sont parfois impossibles à produire dans la réalité. Cette incrustation d'images de synthèse qui n'ont aucune existence physique dans une scène réelle constitue en quelque sorte un apport à la réalité ou une "augmentation".

Les systèmes de réalité augmentée permettent la superposition des images virtuelles sur des scènes réelles. Le concept de la réalité augmentée enrichit notre perception du monde réel, en y ajoutant des informations de manière dynamique et interactive. Ses applications sont multiples et touchent à plusieurs domaines : jeux vidéo, télévision, industrie, médecine, etc.

L'objectif de la réalité augmentée est d'apporter un réalisme et une cohérence visuelle entre le flux réel et virtuel. Afin de maintenir le recalage dynamique des objets de synthèse sur le monde réel, les systèmes d'acquisition doivent en permanence calculer le point de vue de l'opérateur pour donner l'illusion que ces objets virtuels appartiennent au monde réel. Le positionnement de ces objets dans la scène nécessite de connaître la position et l'orientation de la caméra par rapport à un repère lié au monde.

Le problème de la localisation de la caméra est donc important, divers travaux sont menés dans ce domaine pour le résoudre. Plusieurs capteurs peuvent être utilisés pour déterminer le positionnement du point de vue de l'utilisateur dans l'environnement comme: les capteurs magnétiques, les capteurs optiques, les encodeurs ou évidemment les caméras. Toutefois, dans la littérature, la pose est souvent calculée par la caméra en utilisant des méthodes géométriques. Ces méthodes emploient des modèles d'objets et leurs projections dans l'image pour résoudre la transformation entre les repères où les données sont exprimées.

Si l'on considère uniquement les informations acquises par la caméra, le problème de réalité augmentée se ramène à un problème de vision par ordinateur. Les techniques de vision permettent à la fois la reconstruction 3D des points de la scène, la calibration, la localisation 3D, etc. Cependant, le temps de traitement avec les systèmes de vision dépend de la complexité algorithmique des méthodes implémentées.

Le but de ce mémoire est « L'insertion en temps réel d'un objet virtuel dans une séquence vidéo pour la réalité augmentée » ; on utilise le système de marqueur pour la réalité augmentée.

Dans le premier chapitre, nous présentons de manière générale le domaine de Vision par ordinateur en tant que contexte de recherche pour notre problématique, nous rappelons les concepts de base de la vision.

Le deuxième chapitre exposera une étude présentative de différentes méthodes de détection et suivi du mouvement en temps réel.

Le troisième chapitre nous présentons le domaine de la réalité augmentée et leurs domaines d'application et les architectures modulaires existantes.

Le quatrième chapitre sera réservé pour la conception et l'implémentation du système proposé. Enfin ce mémoire se termine par une conclusion récapitulative, qui tracera également les perspectives potentielles que nous envisageons d'entreprendre dans les travaux futures.

CHAPITRE I : VISION PAR ORDINATEUR

1. INTRODUCTION

L'espace qui nous entoure à une structure tridimensionnelle (3D), lorsqu'on demande à une personne de décrire ce qu'elle voit, elle n'éprouve aucune difficulté à nommer les objets qui l'entourent : téléphone, table, livre ...Etc. Pourtant l'information qui est réellement disponible sur la rétine n'est, ni plus ni moins, une collection de points.

En chaque point ou pixel (Picture élément) il y a tout simplement une information qui donne une indication quant à la quantité de lumière et la couleur qui provienne de l'espace environnant et qui ont été projetées à cet endroit de la rétine. Le téléphone, la table ou le livre n'existe pas dans la rétine. Guidé à la fois par l'information codée dans l'image (ou la rétine) et par ses propres connaissances, le processus visuel construit des percepts. Le téléphone ou le livre sont les réponses finales, résultant d'un processus d'interprétation qui fait partie intégrante du système vision. De plus, il n'y a pas de correspondance terme à terme entre l'information sensorielle (la lumière et la couleur) et la réponse finale (des objets 3D). Le système de vision doit fournir les connaissances nécessaires afin de permettre une interprétation non ambiguë.

D'autre part la vision par ordinateur désigne la partie de la perception artificielle concernée par le canal visuel. La perception artificielle a pour objectif de fournir à un système des informations sur le monde qui l'entoure par l'intermédiaire d'un capteur.

2. QU'EST-CE QUE LA VISION

La vision par ordinateur appelée aussi vision artificielle, vision numérique ou récemment vision cognitive est une branche de l'intelligence artificielle, son but est de permettre à une machine de comprendre ce qu'elle « voit » lorsqu'elle est connectée à une ou à plusieurs caméras, en d'autres termes essayer de modéliser et reproduire la vision humaine. En vue d'atteindre cet objectif, plusieurs sciences sont mises à contribution comme les mathématiques, l'informatique, le génie électrique, la psychologie, la biologie, les sciences cognitives et la physique [IMN459., 08].

La vision artificielle se divise en deux catégories :

- La 1ère catégorie concerne le traitement d'images « bas niveau », elle est représentée par les applications suivantes : le suivi des formes, la détection des contours, la segmentation, la super résolution, le rehaussement et la restauration, la transmission et la compression, etc.
- La 2ème catégorie est l'analyse d'image « haut niveau » qui consiste à étudier des techniques pour l'interprétation, la reconstruction et la compréhension d'une scène 3D à partir d'une ou plusieurs images 2D « compréhension de l'image, reconnaissance des formes et stéréovision » [ELE6812].

2.1. Le système de vision

Un système de vision peut être décrit comme un ensemble de processus fortement liés, opérant des transformations progressives des informations sensorielles, en référence à différents modèles (perceptuel, fonctionnel, etc.) intégrant en particulier les connaissances spécifiques aux différents niveaux de traitement. Ces transformations assurent le passage de l'image physique saisie à une description symbolique puis sémantique, en passant par la segmentation et l'identification d'attributs caractéristiques de la scène et des objets qui la composent.

D'une façon plus abrupte nous pouvons dire qu'un système de vision est un ensemble capable de:

- Saisir une image ou une représentation spatiale d'une scène.
- Extraire les informations intéressantes (relativement aux objectifs).
- Interpréter ces informations pour décider d'une action à engager.

Le but fondamental de la vision est de produire à partir d'images une description de la forme, des caractéristiques et de la position des objets de la scène. Avec la naissance de machines de calcul de plus en plus sophistiquées [ELE6812].

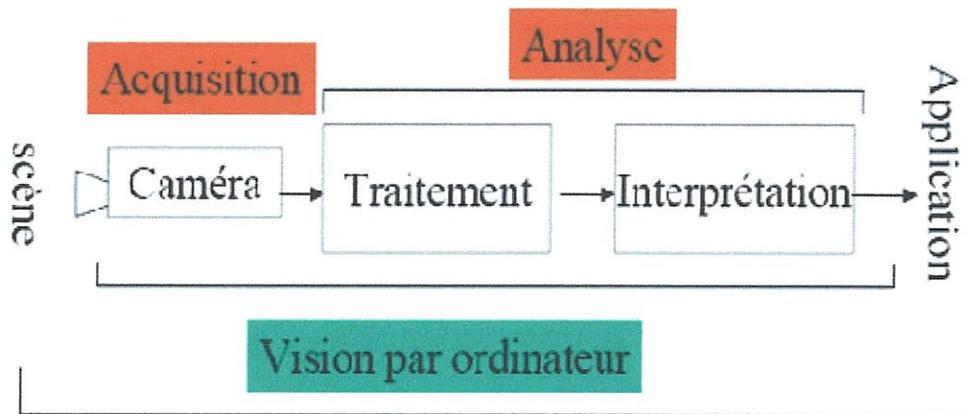


Figure 1.1 : système de vision.

3. Modélisation de la camera

La modélisation d'une caméra est basée sur la définition d'un ensemble de paramètres qui se rapproche le plus possible du comportement physique du capteur à l'aide d'un modèle géométrique, connu comme étant le modèle sténopé (en anglais : pinhole). Le modèle géométrique sténopé est constitué d'un plan, appelé plan rétinien, dans lequel l'image se forme à l'aide d'une projection perspective. Un point O_c est placé à une distance fixe du plan rétinien. Ce point est appelé le centre optique (ou point focal) et il est défini de sorte que les rayons de lumière reflétés par les objets passent par ce point, formant une image perspective de la scène dans le plan rétinien.

A ce moment, deux considérations doivent être prises en compte. La première est que le point focal peut être placé devant ou derrière le plan rétinien. Dans le cas où le centre optique est placé devant le plan rétinien, l'image obtenue est une projection inverse de la scène. Dans l'autre cas, l'image obtenue est une projection de la scène [Mars., 98].

3.1. Rappels sur la projection

L'espace réel est de nature tridimensionnelle, alors que les outils usuels de formation d'une image sont bidimensionnels. La projection est la fonction de passage du 3D au 2D.

L'image 2D est la résultante d'une combinaison des 3 dimensions de l'objet. Cette combinaison donne un aspect géométrique spécifique lorsque la géométrie de l'objet existe (droite, plan, polygone, polyèdre...) [BON].

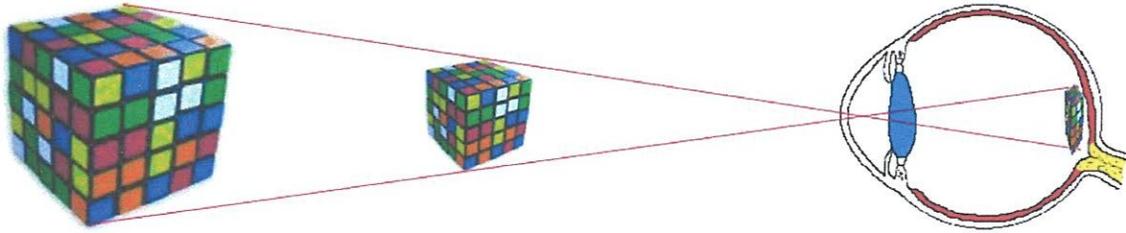


Figure 1.2 : Projection

Le contenu de l'image 2D peut permettre de retrouver certains éléments caractéristiques de la scène initiale, sous réserve d'apporter des informations complémentaires (*connaissance à priori*). Cependant, le seul moyen pour retrouver les caractéristiques 3D est d'augmenter la quantité d'information mesurée; l'une des solutions est la stéréoscopie qui combine deux points de vue différents pour reconstruire la scène 3D [BONNET].

a. Projection

La projection est la loi géométrique selon laquelle les points de l'espace réel 3D se projettent sur la surface du capteur. On suppose que le repère de l'espace des objets (environnement extérieur) correspond avec celui de la caméra.

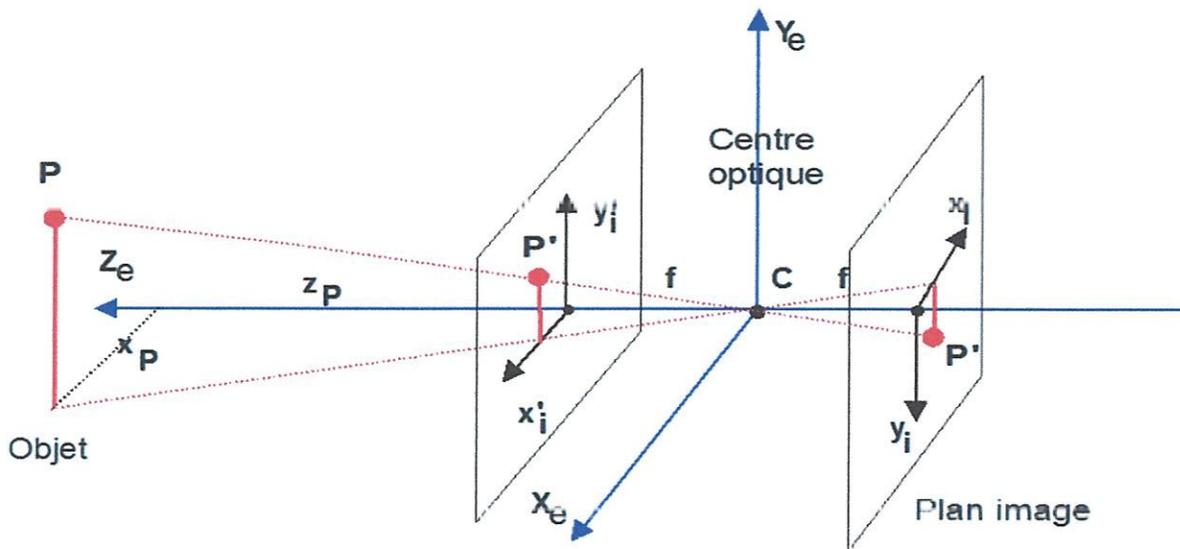


Figure 1.3 : Projection géométrique

Les conditions géométriques sont les suivantes (figure 1.3):

- l'image se forme dans un plan (surface sensible du capteur) perpendiculaire à l'axe optique du système
- l'origine du repère environnement extérieur (x_e, y_e, z_e) est supposée être en C, centre optique de la lentille (point nodal du plan principal objet pour un système optique)

- l'axe Z est confondu avec l'axe principal du système optique.
- les axes x et y du plan image sont parallèles aux axes X et Y du repère objet et leur origine est sur l'axe Z.
- le système optique est du type *sténopé* (pinhole « modèle de projection perceptive »).

Soit un objet P de coordonnées (x_e, y_e, z_e) dans le repère (X_e, Y_e, Z_e) et (x_i, y_i) les coordonnées de son image dans le plan du capteur. L'image se formant avec une *inversion*, on adopte souvent comme repère caméra le repère symétrique par rapport au centre optique (x_i', y_i') .

On remarque la relation par triangles semblables:

$$\frac{x_i}{x_e} = \frac{f}{z_e} \text{ de meme } \frac{y_i}{y_e} = \frac{f}{z_e}$$

On en déduit :

$$x_i = f \frac{x_e}{z_e} \text{ et } y_i = f \frac{y_e}{z_e}$$

On constate que les coordonnées de la projection P' de P dépendent de la profondeur Z_p et que sa localisation (x_p, y_p) ne peut se faire à partir de (x_p, y_p) que si Z_p est connu (cas où les objets sont en parallèle au plan du capteur par exemple).

L'ensemble des points de coordonnées (ax_p, ay_p, az_p) appartenant à la droite $[PC]$ se projette sur la même image; c'est pourquoi la profondeur Z ne peut s'évaluer à partir d'une simple prise de vue [BONNET].

b. Perspective

Le résultat de la projection provoque la notion de perspective. L'exemple le plus visible est celui de l'image d'un rectangle face à la caméra, ayant subi une rotation autour de l'axe vertical d'un angle θ

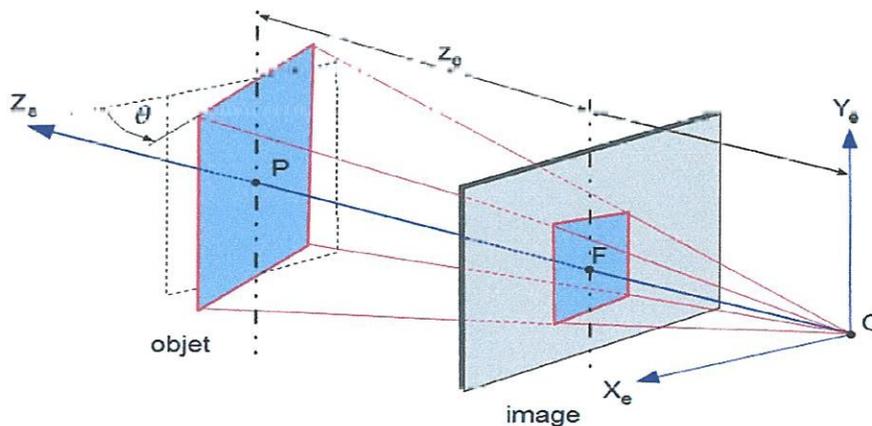


Figure 1.4 : Perspective d'un carré

Les bords verticaux à distance constante sont projetés parallèlement à leur propre direction et restent parallèles entre eux. Les bords supérieur et inférieur sont projetés selon des droites non parallèles. Leur point d'intersection est appelé point de fuite.

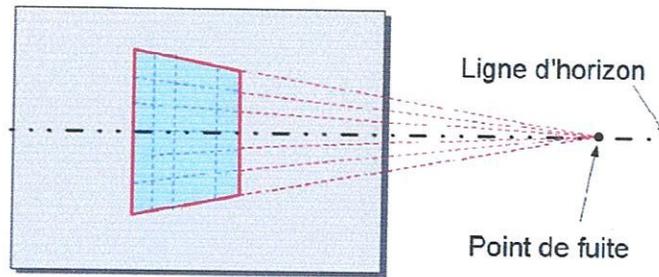


Figure 1.5 : Ligne d'horizon et point de fuite en perspective

Lorsque la caméra n'est pas centrée sur le repère de l'environnement, la ligne d'horizon est modifiée [BONNET].

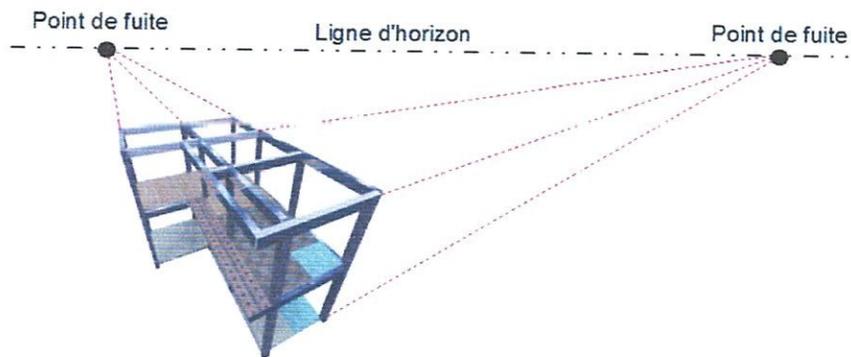


Figure 1.6 : Points de fuite d'un objet tridimensionnel (3ème non tracé)

3.2. Modélisation géométrique

Sur le plan géométrique, le modèle sténopé peut être décrit comme l'ensemble d'un **centre de projection** (ou aussi centre optique) et d'un **plan image** (ou bien d'une **rétine**). Un point 3-D est projeté le long du rayon qui le lie avec le centre de projection a son **point image** étant l'intersection de ce rayon avec le plan image (voir la figure 1.7). Ce rayon est parfois appelé **rayon de projection** ou **ligne de vue**. Cette projection est effectivement une projection perspective.

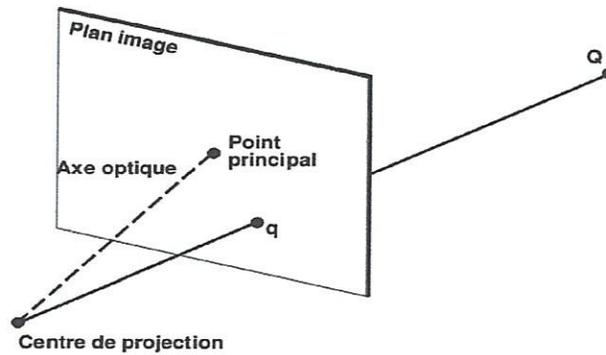


Figure 1.7 - Le modèle sténopé.

La droite qui passe par le centre de projection et qui est perpendiculaire au plan image, est appelé **axe optique**. Le point d'intersection de l'axe optique avec le plan image est le **point principal**.

3.3. Modélisation algébrique

Nous venons de décrire un modèle géométrique pour des caméras. Pour effectuer des calculs, nous devons établir une description algébrique analogue. Dans la suite, nous allons faire ceci, tout en identifiant des paramètres physiques de caméra. Afin de faire des calculs avec des points, nous avons besoin de coordonnées et donc de repères de coordonnées. Nous allons utiliser les coordonnées homogènes [Peter].

a. Étape 1 – Prise en compte de la distance focale

- Nous allons dériver les équations de projection à l'aide de 4 repères (voir la figure 1.8). commençons avec un repère 3D attaché à la caméra – le repère caméra. Comme origine nous choisissons le centre de projection et comme axe des Z l'axe optique. Les axes des X et des Y sont choisis comme étant parallèles au plan image et perpendiculaires entre eux. En plus, les axes sont choisis tels qu'ils sont « alignés » avec les pixels.
- Nous définissons un repère 2D pour le plan image – le repère image. Son origine est le point principal. Ses axes des x et y sont parallèles aux axes des X et Y du repère caméra. En quelque sorte, le repère image peut être vu comme la projection orthogonale du repère caméra.
- la distance focale f comme étant la distance entre le centre de projection et le plan image (donc la distance entre le centre de projection et le point principal).

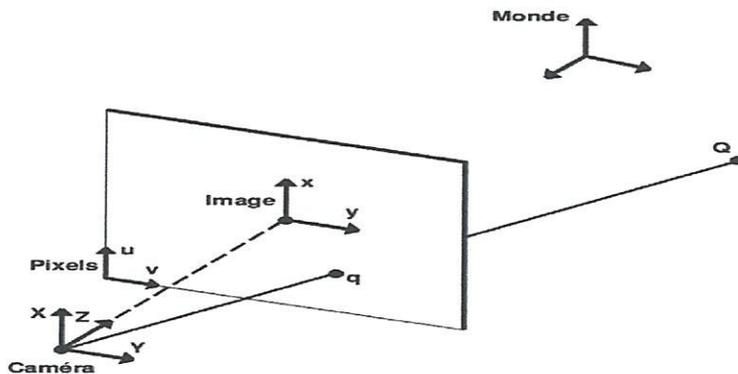


Figure. 1.8 – Les repères utilisés.

Nous pouvons maintenant dériver les équations de projection d'un point 3D 'Q', dont les coordonnées sont donnés par rapport au repère caméra (ce qui s'exprime par l'exposant c):

$$Q^c = \begin{pmatrix} x^c \\ y^c \\ z^c \\ 1 \end{pmatrix}$$

Les coordonnées x et y du point image q, projection de Q, peuvent être calculées à l'aide des relations entre triangles similaires :

$$x = f \frac{x^c}{z^c} \dots\dots\dots(1)$$

$$y = f \frac{y^c}{z^c} \dots\dots\dots(2)$$

En coordonnées homogènes, on peut représenter ces équations par une multiplication

matrice-vecteur : $q = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ z^c \\ 1 \end{pmatrix} \dots\dots\dots(3)$

Nous vérifions maintenant si l'équation (3) est équivalente aux équations (1) et (2). Partons du côté droit de l'équation (3) :

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ z^c \\ 1 \end{pmatrix} = \begin{pmatrix} f x^c \\ f y^c \\ z^c \end{pmatrix}$$

b. Étape 2 – Prise en compte des pixels

Nous considérons surtout des images digitales (donc adaptées pour la vision par ordinateur). Une image digitale est en effet une matrice de valeurs (e.g. Niveaux de gris, couleurs RVB) – les cellules étant les pixels. Pour des traitements d'image par exemple, les pixels sont identifiés par des coordonnées. En effet, on les énumère tout simplement, une fois en direction horizontale, une fois pour la verticale. Ceci implique que nous ne pouvons pas directement utiliser les coordonnées 2D (le repère image), mais que nous devons prendre en compte un changement de repère.

Différents logiciels d'affichage ou éditeurs d'images utilisent parfois différents repères pour identifier les pixels. Soit le coin en haut à gauche, soit celui en bas à gauche de l'image est utilisé comme origine. L'axe de la première coordonnée est soit choisi horizontalement, soit verticalement. Ici, pour simplifier les choses, nous définissons le repère pixels comme c'est montré sur la figure 1.8. Les coordonnées pixelliques sont notées u et v .

Le changement de repère nécessite alors tout d'abord une translation. En plus, il faut effectuer un changement d'unité : le repère image est un repère métrique – on mesure par exemple en mm; dans le repère pixels l'unité est « nombre de pixels » – ce qui n'est pas une unité métrique. Surtout dans de vieilles caméras, les pixels ne sont parfois pas carrés, mais ils ont un côté plus grand que l'autre (c'est dû à des standards de télévision). Ceci implique que chaque axe devra subir un changement d'unité individuel.

c. Étape 3 – Prise en compte des déplacements

Jusqu'ici nous avons représenté les points 3D dans un repère attaché à la caméra. Afin de prendre en compte les déplacements que les caméras vont effectuer, nous devons choisir un repère « statique », attaché à la scène 3D. Nous introduisons donc un repère monde (voir la figure 1.8), qui peut être choisi de manière arbitraire (mais qui restera le même pour toute une application). La position des points 3D et des caméras sont alors décrites par rapport à ce repère.

Nous modélisons dans la suite la position de la caméra. Elle comprend la position proprement dite – la position du centre de projection – et l'orientation de la caméra. La position est représenté par un vecteur t tel que :

$$\begin{pmatrix} t \\ 1 \end{pmatrix}$$

Sont les coordonnées homogènes du centre de projection. L'orientation de la caméra peut être exprimée par une rotation, ce qui sera représenté ici par une matrice de rotation de

taille 3×3 noté \mathbf{R} . À la fin de cette section, nous rappelons brièvement quelques caractéristiques des matrices de rotation. Soient x^m, y^m, z^m les coordonnées du point 3D ' \mathcal{Q} ', exprimées dans le repère monde. Le changement de repère du repère monde vers le repère caméra peut être écrit comme suit :

$$\begin{pmatrix} x^c \\ y^c \\ z^c \end{pmatrix} = R \left(\begin{pmatrix} x^m \\ y^m \\ z^m \end{pmatrix} - t \right) = R \begin{pmatrix} x^m \\ y^m \\ z^m \end{pmatrix} - Rt$$

Et en coordonnées homogènes :

$$\begin{pmatrix} x^c \\ y^c \\ z^c \\ 1 \end{pmatrix} = \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x^m \\ y^m \\ z^m \\ 1 \end{pmatrix}$$

Vérifions que le centre de projection est effectivement l'origine du repère caméra :

$$\begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix} = \begin{pmatrix} Rt - Rt \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Nous avons exprimé la position et l'orientation d'une caméra. Des déplacements peuvent alors être modélisés par exemple par une suite de matrices de rotation \mathbf{R} et de vecteurs de translation \mathbf{t} [Peter].

d. Paramètres extrinsèques et Intrinsèques

Nous pouvons regrouper les paramètres définissant la projection effectuée par une caméra, en deux ensembles :

Les paramètres extrinsèques : la matrice de rotation R et la position t . représentent la position et l'orientation de la caméra par rapport au « monde extérieur ».

Les paramètres intrinsèques : quant à eux, représentent les caractéristiques internes de la caméra, qui sont invariantes à sa position. Il s'agit des paramètres :

f : la distance focale

K_u : la densité de pixels en direction de l'axe des u

K_v : la densité de pixels en direction de l'axe des v

x_0, y_0 : les coordonnées du coin en bas à gauche de l'image, par rapport au repère image [Peter].

4. Calibrage de caméra

Le calibrage d'une caméra a pour but principal la détermination de ses paramètres intrinsèques. Pourtant, il est difficile de dissocier les paramètres intrinsèques de ceux extrinsèques lors de calculs. C'est pourquoi le processus standard de calibrage détermine les deux ensembles de paramètres simultanément.

Le processus standard consiste à prendre une image d'un objet dont la structure est parfaitement connue – **une mire de calibrage**. Pour ce qui est de la mire montrée sur la figure 1.9 par exemple, les positions des « cibles » blanches sont connues avec une très haute précision, dans un repère attaché à la mire. Dans l'image, les cibles sont extraites par un traitement d'image. Il faut ensuite établir une mise en correspondance: pour chaque cible dans l'image, il faut déterminer la cible en 3D dont elle est la projection. Ce processus est typiquement fait de manière semi-automatique [Peter].

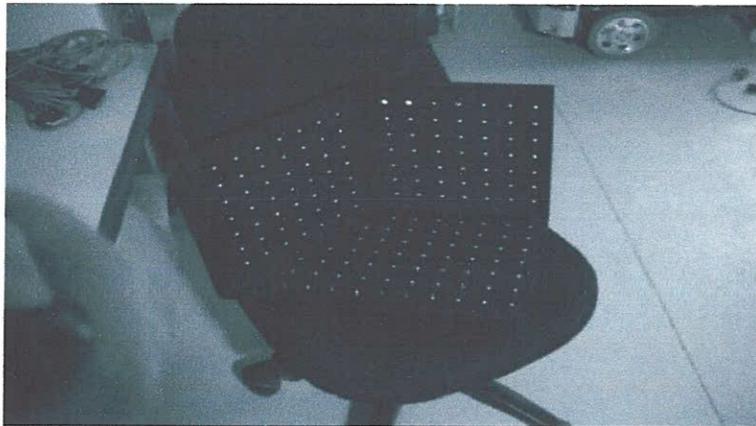


Figure. 1.9 : La mire de calibrage.

La mire (ou cible) doit contenir des formes faciles à reconnaître et dont la position est très bien connue selon le référentiel environnement.

5. Extraction de l'information 3D

5.1. Caractéristiques

L'extraction d'informations 3D s'effectue à partir de certaines caractéristiques (structures) présentes dans les images : (Niveaux de gris, Contours, Couleurs, Jonctions (coins), Textures, Disparités, Mouvement 2D, Ombrages, Quantité de flou...) [IMN459., 08]

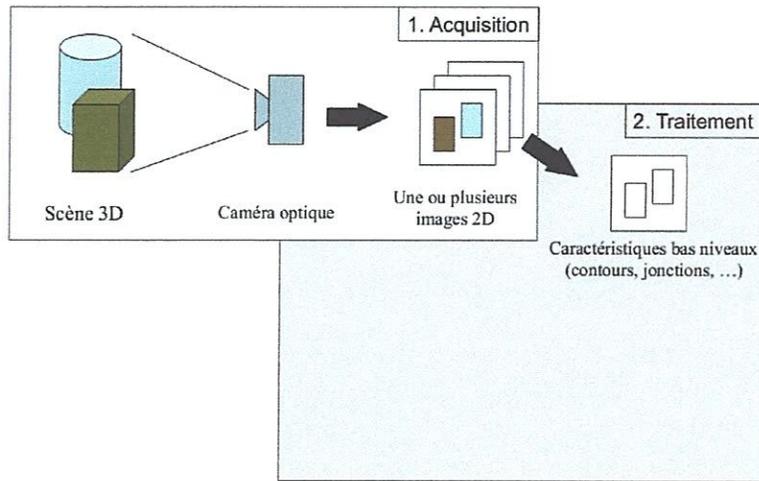


Figure 1.10 : traitement avec des Caractéristiques bas niveau.

5.2.Indices de profondeur

Certaines de ces caractéristiques sont appelées « indices de profondeur »

- ➔ Il faut distinguer la profondeur des indices
- ➔ Les indices de profondeur peuvent être plus complexes à trouver que les simples caractéristiques bas niveau.

Les indices de profondeur dépendent de :

- Système de formation d'images
 - Actif (e.g. laser, sonar, rayon-X)
 - Passif (e.g. caméra optique)
- Nombre d'images (1, 2, ...)
- Domaine (Fourier ou spatial)
- Connaissances a priori (modèles 3D ou non, scène intérieure ou extérieure, ...).

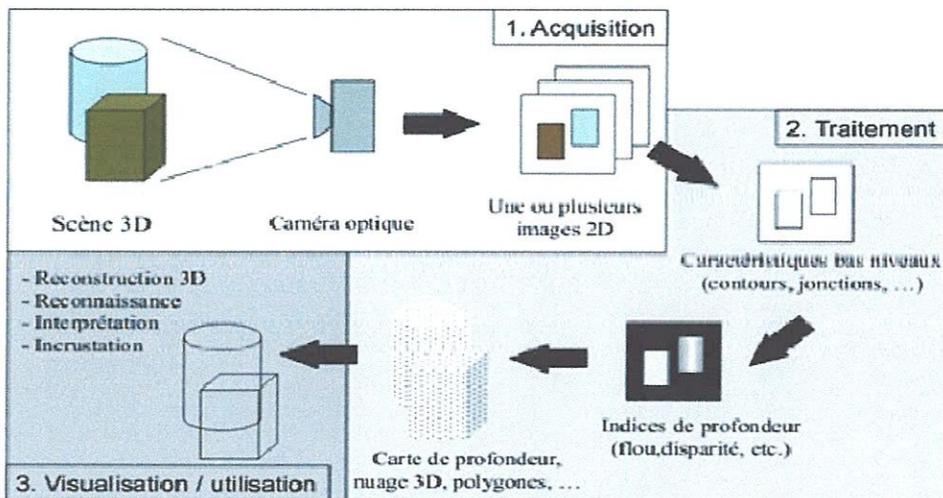


Figure 1.11 : traitement avec des Caractéristiques bas niveau + indices de profondeur + carte de profondeur

a. Classes d'indices de profondeur

- i) Indices multi-oculaires : 2 points de vue différents ou plus
 - Décalages spatiaux
 - Mouvement
- ii) Indices monoculaires : une ou plusieurs images
 - Flou et mise au point
 - Déformation des éléments de texture
 - Diffusion de la lumière dans le milieu (atmosphère, eau, particule)
 - Illumination

5.3. Exemples

a. Textures

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique pouvant se répéter, ou des caractéristiques fréquentielles.

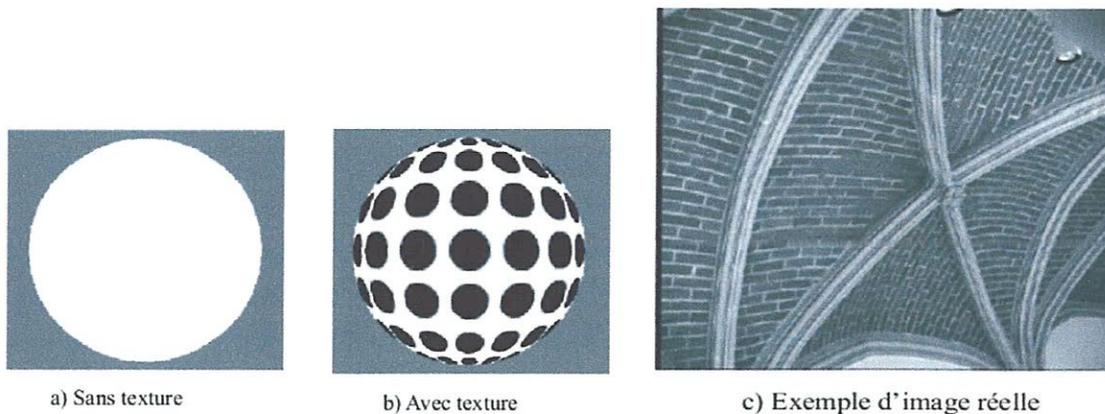


Figure 1.12 : Perception de la forme par la texture.

b. Ombrages

L'ombre est une zone sombre créée par l'interposition d'un objet opaque entre une source de lumière et la surface sur laquelle se réfléchit cette lumière. Elle se matérialise par une silhouette sans épaisseur. Une ombre portée contient la couleur complémentaire de la source de lumière qui la crée.

La taille de l'ombre portée dépend de la taille de l'objet intercalé et de sa distance relative de la source de lumière. Plus l'objet est près de la source de lumière, plus la zone d'ombre sera grande. Son intensité dépend de la proportion de la lumière apportée par la source masquée. Si la source masquée est la seule source présente, alors la zone à l'ombre est dans l'obscurité totale.

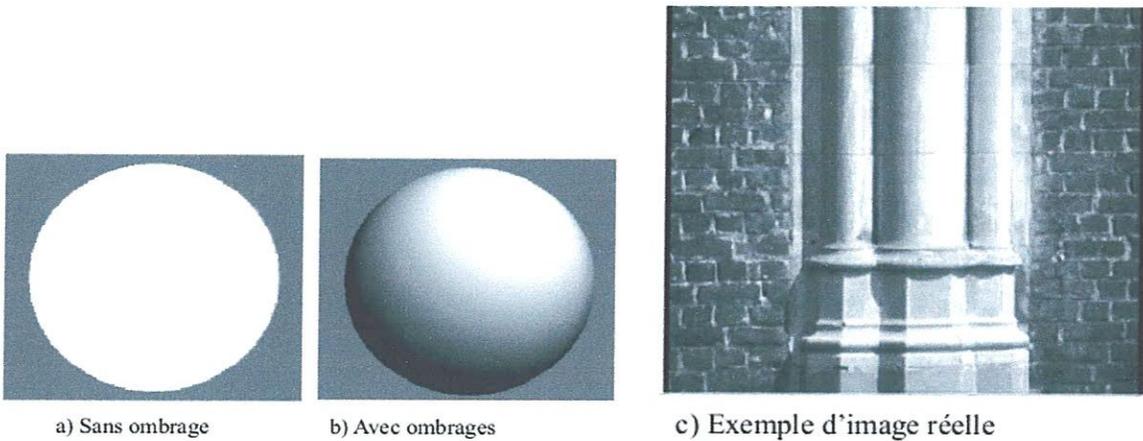


Figure 1.13 : Perception de la forme par l'ombrage.

c. Disparités binoculaires

Les disparités binoculaires, ou parallaxe binoculaire, sont les différences de position relative des projections du même objet sur les rétines des deux yeux [Michel., 04]

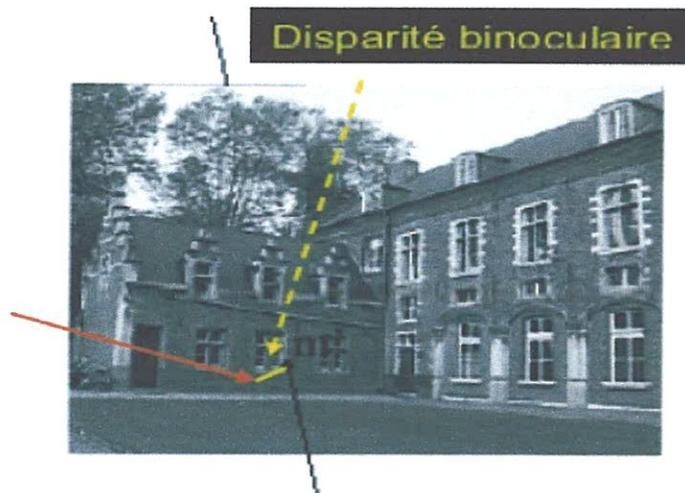


Figure 1.14 Exemple de disparités binoculaires.

6. Conclusion

Dans ce chapitre, Nous avons présenté la notion de la vision par ordinateur, le système de vision, la Modélisation de caméra ou nous avons discuté la modélisation géométrique et algébrique avec leurs étapes et puis les paramètres extrinsèques et intrinsèques, Calibrage de caméra et aussi Extraction de l'information 3D avec des exemples.

CHAPITRE II : DÉTECTION ET SUIVI DU MOUVEMENT

1. Introduction

Pour pouvoir suivre les mouvements dans la vidéo, la première étape est de détecter les mouvements. Cette étape joue un rôle très important dans le système de vision car le résultat de cette étape va influencer toutes les étapes suivantes.

Les caméras numériques présentent sur le marché coûtent de moins en moins cher, et leur encombrement est de plus en plus réduit. Parallèlement, la puissance de calcul des ordinateurs actuels permet d'envisager sérieusement le traitement automatique en temps réel de séquences vidéo. C'est pourquoi les industriels ont aujourd'hui tendance à opter pour des solutions à base de vision artificielle pour résoudre des problèmes qui étaient auparavant traités par d'autres procédés, tels que la surveillance par un opérateur humain ou l'utilisation de capteurs plus mécaniques. Quelle que soit l'application, la première tâche d'un système d'analyse de séquences vidéo est toujours la détection de mouvement, et si possible, la détection (segmentation) des objets mobiles. La difficulté de cette tâche est très variable selon les conditions d'acquisition, la précision et la rapidité du traitement sont comptées. Une liste relativement exhaustive des difficultés liées à l'acquisition et au contenu de la scène.

2. problèmes de détection

Le défi de la détection du mouvement dans le cas de capteurs fixes tient dans la capacité d'effectuer une bonne segmentation des objets en mouvement indépendamment de leurs tailles, ou de leurs contrastes par rapport au fond. Dans un tel contexte, l'élaboration des algorithmes de détection du mouvement se fait en tenant compte des contraintes suivantes :

- Le système doit être en mesure de fonctionner sans intervention humaine pendant un long moment, et être capable de prendre en compte des changements graduels ou soudains tels que les variations d'illumination ou la présence de nouveaux objets statiques dans la scène. Le système doit donc être temporellement adaptatif;

- Le système doit être capable d'éliminer tous mouvements inintéressants tels que le bruit résultant d'un cours d'eau ou de hautes herbes agitées par le vent. Il doit être robuste à de petits mouvements du capteur. Il doit donc y avoir une estimation locale de la confiance dans la valeur estimée du fond

- Le système doit être temps réel, compact et basse consommation. Les algorithmes ne doivent pas consommer trop de ressources, en termes de temps de calcul et de capacité mémoire.

Les deux premières conditions impliquent que les mesures statistiques de l'activité temporelle doivent être effectuées localement en chaque pixel, et constamment remises à jour. Cela exclu toutes les approches utilisant des modèles Simples de différence trame à trame, de seuillage global et les méthodes utilisant une moyenne comme seuil de décision[**Julien**].

3. Les différentes méthodes de détection de mouvement

3.1. Détection sans modélisation de l'arrière-plan.

Ces méthodes consistent à détecter le mouvement par le calcul en tout point de l'image d'une quantité mathématique qui est fonction de l'intensité ou de la couleur de l'ensemble des pixels et qui est censée refléter l'importance du mouvement visible dans la scène. La dérivée temporelle de l'intensité lumineuse, l'entropie spatio-temporelle de l'image et la norme du flot optique. [**Nicolas ,07**]

a. Dérivée temporelle

La manière la plus intuitive de détecter les zones en mouvement dans un champ de vision est de mesurer le changement d'apparence des pixels entre deux trames consécutives, soit la dérivée temporelle en tout point. La première utilisation de cette méthode dans l'analyse de séquence vidéo est généralement attribuée à Jain et Nagel [**Jain et al., 79**]. L'intensité lumineuse d'un pixel étant un signal discret à une dimension, l'estimation de la dérivée temporelle instantanée du signal au temps t est donnée par

$$\forall (x, y) \in \mathbb{E} \forall t > 0 \quad I_t(x, y, t) \approx |I(x, y, t) - I(x, y, t - 1)|$$

[**Tian et al., 05**] observent que cette méthode se montre peu robuste face à des phénomènes tels que les mouvements lents ou saccades, les arrêts brefs d'un objet en mouvement, ou encore la présence de trames redondantes dans certaines séquences vidéo. Il convient donc d'effectuer un lissage temporel de la séquence, c'est-à-dire d'appliquer un

opérateur de moyenne mobile à la mesure obtenue. Ceci peut être fait à l'aide d'une matrice de même taille que les trames d'entrée, appelée accumulateur

b. Entropie spatio-temporelle

L'entropie est une mesure issue de la thermodynamique, associée au degré de désordre d'un système. Il s'agira de mesurer en chaque point, la variabilité de la grandeur mesurée. Concrètement, plus l'intensité lumineuse (ou la couleur, ou le gradient, etc.) aura pris de valeurs dissemblables en un point pendant un certain intervalle de temps, plus l'entropie sera élevée en ce point. Les auteurs proposent, pour calculer l'entropie, une méthode à base d'histogrammes spatio-temporels [Ma et al., 01].

Il s'agit, en chaque pixel désigné par (x, y, t) , de créer un histogramme à partir des points (u, v, t) d'un voisinage spatio-temporel de diamètre W dans le domaine spatial, et de longueur L dans le domaine temporel. Ainsi, pour une image à Q niveaux de gris, on note $H_{x,y,t}(q)$; la fréquence de la q -ème classe de l'histogramme ($0 < q < Q - 1$) associée au point (x, y, t) .

c. Flot optique

Le flot optique est un champ de vecteurs à deux dimensions représentant la projection sur le plan image du mouvement réel observé (tridimensionnel). De nombreuses méthodes ont été proposées depuis l'article précurseur de Horn et Schunck [Horn et al., 81], celles-ci sont détaillées dans plusieurs états de l'art. Dans [Barron et al., 94], neuf algorithmes sont étudiés et comparés selon des critères de précision et de la densité du champ obtenu, mais aucune mention n'est faite de la complexité algorithmique. Les travaux de [Liu et al., 98] permettent de combler cette lacune en mesurant les rapports précision/temps de calcul de ces méthodes. Quelle que soit la méthode choisie, le calcul du flot optique reste une opération très coûteuse en temps de calcul. Le temps-réel peut néanmoins être atteint en sous-échantillonnant les trames et en choisissant un algorithme rapide

3.2. Modélisation locale de l'arrière-plan

Ces méthodes consistent à associer à tout point de l'image une valeur ou une fonction permettant de modéliser l'apparence de l'arrière-plan en ce point. Le modèle d'apparence de l'arrière-plan en un point ne dépend que des observations qui ont eu lieu en ce point. Les autres pixels de l'image n'interviennent pas. La grande majorité des méthodes présentées dans

la littérature bâtissent un modèle statistique (ensemble de paramètres d'une loi, ou ensemble d'échantillons), mais il peut s'agir d'un processus stochastique, d'un filtre prédictif ou simplement d'une valeur d'intensité [Nicolas ,07].

a. Modélisation par une image

Le modèle d'arrière-plan le plus simple serait une image représentant la scène dépourvue d'objets. Cette méthode présente l'avantage de détecter aussi bien les mouvements lents que les mouvements rapides. Par ailleurs, même les objets momentanément immobiles sont détectés. Cependant, en environnement extérieur, les variations d'intensité lumineuse rendent rapidement obsolète un tel modèle, et il est nécessaire de mettre à jour cette image de l'arrière-plan. Par ailleurs, il n'est pas toujours possible d'obtenir d'une image de la scène totalement vide. Dans ces conditions, il est nécessaire de mettre à jour l'image de l'arrière-plan par exemple, dans [Yang et al., 04], ils proposent d'utiliser la différence entre images consécutives pour y parvenir. Ils considèrent la première image de la séquence comme une première approximation du modèle de l'arrière-plan. Ensuite, à chaque nouvelle trame, la différence par rapport à l'image précédente est calculée, et les pixels où aucun mouvement n'est détecté sont utilisés pour mettre à jour le modèle du fond. Pour plus de robustesse, les auteurs préconisent de ne considérer que les points auxquels la dérivée temporelle a été négligeable pendant un certain intervalle de temps.

b. Modélisation statistique

Le problème de la modélisation de l'arrière-plan peut être exprimé d'un point de vue statistique. Il s'agit, pour chaque pixel, d'estimer la probabilité d'y observer telle ou telle couleur (ou niveau de gris) en se basant sur un modèle appris, censé représenter l'arrière-plan de la scène. Le modèle consiste en un ensemble de fonctions de densité de probabilité une par pixel de l'image. Les mesures dont la probabilité d'être observées est élevée correspondent à des pixels qui seront étiquetés comme arrière-plan, tandis que celles dont la probabilité d'être observées est faible correspondent à des pixels qui seront étiquetés comme avant-plan.

Dans [Wren et al., 97], ils proposent de modéliser l'intensité des points de l'arrière-plan par une distribution gaussienne. En tout point la moyenne et l'écart-type sont mis à jour récursivement, et chaque nouvelle observation est déclarée comme appartenant à l'arrière-plan si elle se situe suffisamment près de la moyenne courante, compte tenu de l'estimation courante de l'écart-type. Le même procédé peut être utilisé sur des images en couleur [McKenna et al., 00]. Ce modèle permet d'obtenir de bons résultats pour des scènes

d'intérieur ou l'arrière-plan est parfaitement statique, mais en environnement extérieur, des phénomènes périodiques tels que l'ondulation d'une surface d'eau ou le balancement d'une branche d'arbre peuvent le rendre totalement inopérant car la distribution de l'apparence de l'arrière-plan est alors multimodale.

3.3. Modélisation semi-locale de l'arrière-plan

Ces méthodes sont très semblables à celles de la catégorie précédente, à la différence presque la modélisation de l'arrière-plan en un point dépend des observations qui ont eu lieu dans un certain voisinage de ce point, ou dans la région de l'image à laquelle il appartient [Nicolas, 07].

a. Caractérisation par la texture

La prise en compte du voisinage des points peut également être réalisée en calculant en tout point un vecteur caractérisant la texture à cet endroit, et en utilisant l'ensemble des vecteurs calculent comme espace de représentation des données.

Les auteurs de [Heikkila et al., 06] utilisent le codage LBP (Local Binary Patterns) pour caractériser la texture des pixels de l'image. Le code LBP du point (x, y) est le mot binaire obtenu en concaténant l'intensité souillée de tous les pixels situés dans un voisinage de (x, y) . Le voisinage est constitué de P points $\{(u_i, v_i)\}_{p_i=1}$

Uniformément répartis sur un cercle de rayon R centre en (x, y) . La valeur du seuil utilisé est l'intensité du point (x, y) .

b. Régularisation a posteriori

Plusieurs auteurs Toyama et Elgammal proposent une application dite (multi-couches), c'est-à-dire, a plusieurs niveaux sémantiques. Le niveau le plus bas bâtit un modèle local de l'arrière-plan qui permet de réaliser une première estimation de la segmentation des objets mobiles. Ensuite, un second procède analyse les résultats obtenus et les régularise de manière a augmenté la consistance des régions détectées dans un voisinage spatial et/ou temporel.

Après avoir segmenté l'arrière-plan a l'aide d'une modélisation locale, les auteurs de [Toyama et al., 99] considèrent que parmi les points détectés, ceux ou la dérivée temporelle est importante appartiennent nécessairement des objets mobiles. Dans ce cas, un histogramme normalisé des régions trouvées est construit, et les régions sont utilisées comme germes par un algorithme de segmentation par croissance de région utilisant l'histogramme comme critère d'arrêt.

Dans [Elgammal et al., 00] est également utilisée la notion de voisinage spatial pour supprimer les fausses détections du résultat fourni par la modélisation locale. Les auteurs attribuent les fausses détections à de légers mouvements de l'arrière-plan ou de la camera qui n'auraient pas été modélisés pendant la phase d'apprentissage. Afin de les supprimer, ils proposent de calculer en tout point où un mouvement a été détecté, la probabilité que la valeur observée appartienne à la distribution d'arrière-plan d'un des points avoisinants. Cette probabilité sera élevée si le mouvement est dû à un léger déplacement de l'arrière-plan. Malheureusement, elle sera également élevée pour de vraies détections dont l'apparence est similaire à l'arrière-plan de pixels voisins. Avant d'éviter ce phénomène, les auteurs ne suppriment ces détections que si l'ensemble de la composante connexe qui les contient a subi un tel mouvement. Il s'agit donc d'une régularisation qui utilise deux échelles de voisinage.

3.4. Modélisation globale de l'arrière-plan

Ces méthodes utilisent à chaque instant l'ensemble des observations pour construire un modèle de l'ensemble de l'arrière-plan [Nicolas, 07].

a. Basculement entre plusieurs modèles

Pour prendre en compte la totalité de la scène dans le processus de segmentation entre avant-plan et arrière-plan, les auteurs de [Toyama, 99] décident de garder en mémoire k modèles de l'arrière-plan. La détection de mouvement au niveau local est effectuée avec chacun des modèles, et celui qui détecte le moins de pixels en mouvement est retenu pour la décision finale. Initialement, les k modèles sont acquis en exécutant un algorithme k-means sur les images d'une séquence d'apprentissage. En cours de traitement, les modèles peuvent être mis à jour si une large majorité des pixels de l'image sont détectés comme étant en mouvement.

Dans [Stenger et al., 01] est présente un modèle de Markov caché qui modifie automatiquement sa topologie quand il est confronté à un environnement dynamique. Les auteurs utilisent cet outil pour gérer les difficultés liées aux changements soudains d'éclairage. Typiquement, le MMC (un modèle de Markov caché) aura deux états (jour/nuit, par exemple) et à chaque état sera associé un modèle statistique local de l'arrière-plan. La nature précise du modèle statistique n'est cependant pas précisée.

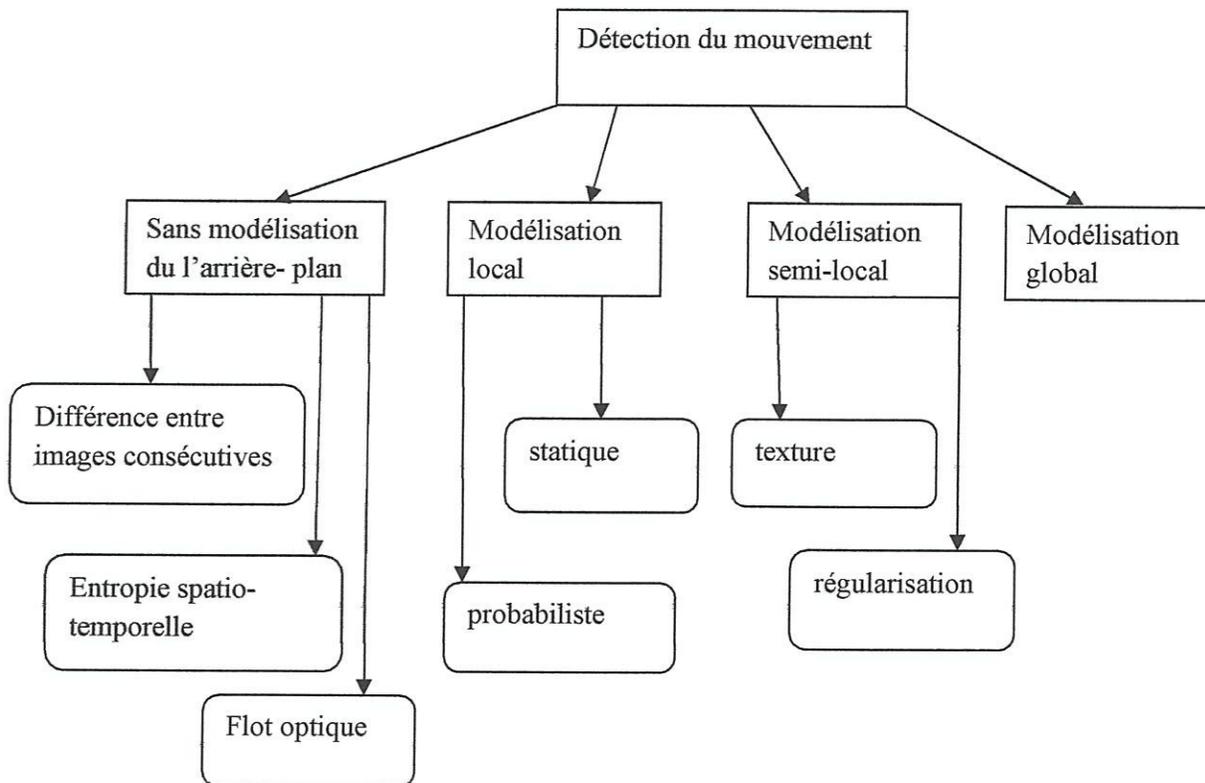


Figure 2.1 : Taxinomie proposée pour les méthodes de détection de mouvement.

4. Estimation et calcul de mouvement

4.1. Introduction

L'estimation du mouvement est une quantification du mouvement simple, par vecteurs de translation, mettant en œuvre des méthodes de calcul de trajectoire. Les applications de l'estimation du mouvement sont surtout la réduction de la redondance temporelle pour la compression et l'analyse de scène. Les méthodes qui reviennent principalement dans l'estimation du mouvement sont [GRAVA, 03]:

- Les méthodes basées sur l'estimation du flot optique (Méthodes différentielles)
- La mise en correspondance de blocs
- Les méthodes fréquentielles
- Les méthodes de multi-résolutions, . . .

La mesure du champ des vitesses est une étape de traitement de l'image dite de bas niveau. On lui trouve de nombreuses applications comme : l'analyse du mouvement des fluides, la compression des séquences d'images vidéo par compensation de mouvement, ou

son utilisation pour des phases de traitement des images de plus haut niveau, comme la reconstruction des scènes tridimensionnelles.

4.2. Mouvement réel, mouvement apparent et mouvement estimé

Les images représentent souvent la projection de scènes réelles 3D. C'est pourquoi le mouvement observé (ou mouvement apparent) dans une séquence temporelle d'images représente généralement la projection du mouvement 3D dans le plan image. On doit différencier [GRAVA, 03]:

- Le mouvement réel
- Le mouvement apparent sur l'image
- Le mouvement estimé

4.3. Champ de mouvement réel et champ de mouvement apparent

Le mouvement (ou déplacement) réel anime la scène réelle, dans un espace tridimensionnel. Ce mouvement est observé par une prise de vue (camera). Quant au mouvement observé ou apparent sous la forme d'une séquence d'images 2D ou 3D, si l'on s'appuie seulement sur l'intensité des pixels. Dans ce cas on observe, en fait des changements de la distribution spatiale de l'intensité lumineuse. Le mouvement ainsi perçu est appelé champ de mouvement apparent ou flot optique qui est en général différent du champ réel de mouvement.

Le champ de déplacement apparent représente en général la projection (orthographe) du mouvement réel, il est appelé aussi mouvement projeté. Ce dernier représente une approximation du mouvement réel.

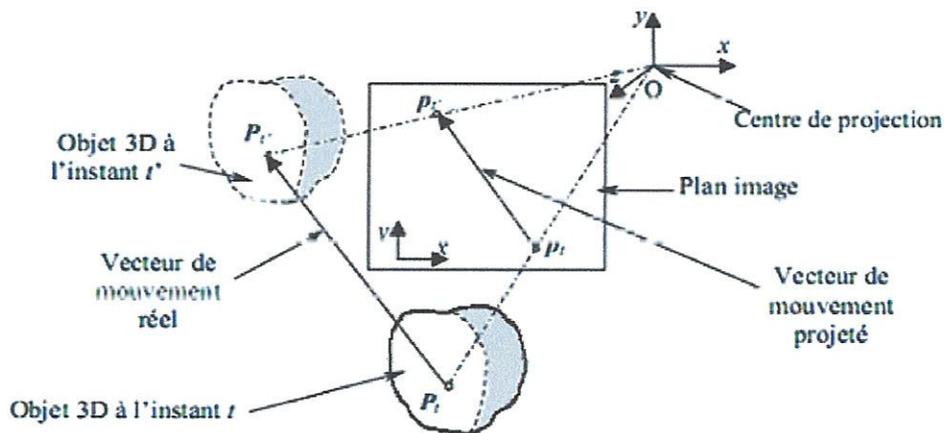


Figure 2.2 : Illustration des mouvements réels et apparents dans un système optique de prise de vues.

5. Différentes approches de suivi

Le suivi d'objets dans une séquence d'images vidéo est un problème qui demande une extraction et un traitement d'informations provenant d'images complexes et incertaines dans beaucoup de cas. Ce problème devient de plus en plus difficile si la contrainte temps réel est exigée.

Nous allons, dans cette section, classer les méthodes existantes en deux grandes catégories (suivi 2D et suivi 3D) selon le type de transformation. Ensuite, nous allons répartir chaque catégorie en plusieurs groupes selon la primitive visuelle utilisée. Enfin, chaque groupe sera divisé en sous-groupes selon la manière de procéder [Bouzenada ,08].

5.1. Méthodes de suivi 2D

C'est le groupe de méthodes qui se basent sur une transformation qui permet de faire un lien entre un modèle 2D de l'objet avec les observations dans l'image courante. Selon la primitive visuelle utilisée, nous pouvons recenser les deux groupes suivants :

a. Méthodes de suivi basées sur l'intensité lumineuse

Ces méthodes se basent sur la texture. Elles procèdent par l'analyse des niveaux de gris dans l'image. Le modèle le plus simple est celui de Horn qui suppose que le signal lumineux de la projection d'un point 3D reste constant dans le temps et se traduit par :

$$\forall X, I_2(f(X)) = I_1(X)$$

Où X est un point de l'image, I_1 le niveau de gris du point X dans l'image modèle de l'objet, I_2 le niveau de gris du point X dans l'image courante de la vidéo et f la fonction de transfert entre les deux images. Ainsi, le critère de recalage dans ce cas est :

$$\Delta = \sum (I_2(f(X)) - I_1(X))^2$$

i. Flot optique

Le flot optique cherche à estimer pour chacun des pixels de l'image le vecteur déplacement entre deux images successives en se basant sur un critère de corrélation. Le principe consiste à estimer le déplacement en chaque point. L'approximation du signal lumineux est un développement de Taylor du 1^{er} ordre.

Ces approches sont très coûteuses en temps de calcul pour pouvoir être couramment utilisées dans le cadre d'une application temps-réel. De plus, le mouvement est estimé en

chaque point de l'image sans exploiter la contrainte imposée par la structure de la scène. Une information de si bas-niveau est difficilement exploitable pour remonter à la notion d'objets.

ii. Suivi de région ou de motif

Ces méthodes sont dites globales puisqu'elles prennent en considération tous les pixels représentant le motif à suivre. Une analyse des niveaux de gris sera ensuite faite sur ces pixels. En effet, ceci permet d'estimer le mouvement d'une région dans l'image.

L'avantage de ces méthodes est qu'elles ne demandent pas une extraction préalable de primitives visuelles. Par conséquent, beaucoup de travaux dans ce sens ont vu le jour.

Darell [Darell et al., 96], Brunelli. [Brunelli et al., 95] proposent de maximiser un critère de corrélation entre un vecteur caractérisant le modèle de référence et le contenu de l'image. Les temps de calcul, significatifs dans ce cas, peuvent être réduits en travaillant dans des sous-espaces de la représentation initiale de l'image. La limitation principale de ces approches est leur manque de résistance au regard des occultations. Black and Jepson [Black et al., 98] ont surmonté cette limitation en reconstruisant les parties occultées. Ils remplacent la norme quadratique généralement utilisée pour construire l'approximation de l'image dans l'espace propre par une norme d'erreur robuste. Cette reconstruction revient à une minimisation d'une fonction non linéaire, optimisée en utilisant une méthode de descente de gradient simple. Ils utilisent la même stratégie pour trouver la transformation paramétrique alignant le motif sur l'image.

b. Méthodes de suivi basées sur les primitives géométriques

Ce type de suivi repose sur un processus bas-niveau qui permet d'extraire localement des points de contour. Sa recherche est généralement effectuée le long de la normale au contour. Aussi, le cadre probabiliste a été intensivement utilisé pour le suivi de primitives, comme la distance de Hausdorff, le filtre de kalman, etc.

i. suivi par modèle basé silhouette

L'une des méthodes basées sur les techniques de transformation des distances est celle de Huttenlocher [Huttenlocher et al., 92] qui consiste à retrouver la position d'un objet, rigide ou non, dans une séquence d'images à partir d'un modèle très simple. Le modèle utilisé pour effectuer la poursuite est simplement composé d'un ensemble de points d'arêtes. Il s'agit d'un modèle qui peut évoluer avec le temps à condition que la silhouette ne change pas trop rapidement d'une image à l'autre.

Soit M_t le modèle à l'instant t et M_0 le modèle de départ défini, dans la première image de la séquence, par l'utilisateur. Le modèle M_t permet de trouver la position de la cible dans l'image

I_{t+1} à l'instant $t+1$. Pour cela, une détection d'arête de type Canny [Canny, 86] est d'abord appliqué sur l'image pour qu'elle soit comparée au modèle M_1 en utilisant la distance de Hausdorff. La position qui donne la distance la plus faible est considérée comme la position effective de l'objet recherché.

Afin d'améliorer cette méthode pour tolérer les changements brusques d'apparence, une base de modèles est générée. Cette base est enrichie au fur et à mesure par l'ajout d'un nouveau modèle jugé suffisamment différent des autres modèles déjà présents dans la base. Dans ce cas aussi, la distance de Hausdorff est utilisée pour apprécier la différence entre les modèles.

ii. suivi par des contours actifs

Les contours actifs sont des courbes paramétriques qui peuvent être déformées sous l'influence de forces externes et internes [Kass et al., 98].

L'énergie d'un contour actif est définie par : une énergie externe provenant de l'image 2D et pouvant avoir différentes formes selon les caractéristiques qui doivent être capturées par le contour actif et une énergie interne qui a une influence sur l'évolution de la forme du contour. Cette dernière possède deux paramètres : le premier régule l'élasticité de la courbe et le second sa flexibilité. Ces paramètres dépendent de la position de la courbe. L'évolution de la courbe se fait autour de la minimisation de la fonction de son énergie totale. Cette phase de minimisation peut être vue comme une évolution dynamique au cours de laquelle l'énergie dissipée est transformée en énergie cinétique et finalement le contour se trouve dans un état d'énergie plus stable.

Les difficultés rencontrées avec les contours actifs résident dans le choix des nombreux paramètres à utiliser. En plus, le bruit peut attirer le contour vers d'autres objets hors l'objet poursuivi. Pour éviter ce problème, un filtrage des images s'avère nécessaire.

Les contours actifs sont énormément utilisés pour le suivi d'objets déformables. Ils sont bien adaptés au suivi des cellules vivantes. Néanmoins, il faut pouvoir choisir correctement les termes de l'énergie interne et externe, ainsi que les constantes qui gouvernent l'évolution. Les méthodes se basant sur les contours actifs ont un caractère itératif coûteux en temps de calcul. Par conséquent, elles sont mal adaptées pour des applications temps réel.

iii. suivi par des modèles actifs de forme

Baumberg et Hogg [Baumberg et al., 94] utilisent ce modèle pour le suivi d'une personne qui marche. Le modèle est formé d'un ensemble de points de contrôle soumis à une analyse par composantes principales pour réduire le nombre de degrés de liberté formant ainsi un ensemble test. Le suivi est effectué à l'aide d'un filtre de Kalman permettant d'estimer les paramètres du modèle ainsi que ceux du mouvement.

L'inconvénient de cette méthode est lié principalement à l'utilisation d'un modèle qui nécessite un entraînement. Autrement dit, le suivi ne pourra être correct que si certaines configurations figurent dans l'ensemble test.

iv. suivi avec utilisation des informations 3D

Koller [Koller et al, 1994] propose de suivre un véhicule sur une autoroute non pas pour calculer la pose mais plutôt pour analyser le trafic. Cette analyse comprend principalement le recensement du nombre de véhicules, l'estimation de leur vitesse et la détection de certains événements, à savoir : changement de voie, arrêt, etc.

Afin de traiter le problème d'occlusions qui apparaissent dans ce genre d'application, la distance du véhicule par rapport à la caméra est évaluée. La méthode se base principalement sur une combinaison entre un modèle complexe du background et un filtre de kalman. Ce filtre sert pour prédire l'évolution du modèle.

Cette méthode passe par trois étapes :

- ✓ Création d'un masque qui correspond aux objets en mouvement dans la scène. Il s'agit ici de détecter des véhicules qui entrent dans le champ de vision de la caméra.
- ✓ Approximation de chaque objet en mouvement par une spline cubique (fonction définie par des polynômes de troisièmes degrés) avec douze points de contrôle. Pour cela, le déplacement de chaque véhicule est décrit en utilisant un modèle de mouvement affine très simple avec une translation 2D et un facteur d'échelle.
- ✓ Le suivi est alors effectué sous forme d'une boucle de prédiction et de mesures. L'étape de prédiction est faite à l'aide de deux filtres de kalman : l'un pour la prédiction de la position du véhicule, l'autre pour la prédiction de sa forme.

Le suivi simultané de plusieurs véhicules nécessite l'ajout de traitements supplémentaires à cause des chevauchements (occlusions) qui apparaissent fréquemment dans une image 2D. Le raisonnement est simple, le véhicule le plus proche de la caméra peut

cache un autre un peu plus loin de celle-ci. Il est nécessaire, alors, d'estimer la distance qui sépare les véhicules et la caméra. Ceci ne pose aucun problème puisque la caméra est correctement calibrée.

5.2. Méthodes de suivi 3D

Il s'agit, pour ces méthodes, d'estimer pour chaque image de la vidéo, les paramètres de position ou de déplacement de la caméra par rapport à l'objet.

a. Estimation à partir des transformations 2D

Les méthodes de suivi 2D peuvent servir pour déterminer la géométrie de la scène. Ces approches ont souvent plusieurs solutions. L'incertitude peut être levée en utilisant soit une information a priori de la scène, soit au moins trois vues de la même scène. Pour simplifier ces approches, certains exploitent l'existence des structures planes dans la scène [Simon et al., 00].

b. Méthodes basées modèle

Le principe des approches basées modèle, qui estiment la pose 3D à partir d'une séquence d'images, est le suivant:

- ✓ Obtenir une pose initiale de l'objet à suivre (phase initialisation). Cette phase est souvent faite interactivement.
- ✓ Projeter le modèle 3D de l'objet à suivre sur le plan image.
- ✓ Recaler la pose de l'objet avec le modèle projeté en se basant sur une mesure de comparaison.
- ✓ Prédire la pose dans l'image suivante à partir de la pose ajustée. Il est nécessaire, ici, de connaître certains paramètres (vitesse ou accélération).

Ces méthodes utilisent des approches mathématiques pour estimer les paramètres d'une transformation. Ces approches ont une influence importante sur le succès du suivi. Les solutions possibles sont nombreuses : estimation linéaire, non-linéaire (gauss-newton, levenberg-marquardt, filtre de kalman étendu), cadre probabiliste (filtres particuliers). Le système de prédiction entraîne en ce qui concerne les filtres de kalman une erreur de traînage dans l'estimation des paramètres pouvant rompre le suivi en cas de discontinuité temporelle trop importante. De plus, ils reposent sur un bruit gaussien des observations, hypothèse souvent fautive. Les filtres particuliers, par contre, permettent de surmonter ce phénomène mais en gardant plusieurs jeux de paramètres en mémoire et par conséquent demandent plus

de temps de calcul. Ainsi, ils ne s'adaptent pas bien aux applications temps réel. Nous allons, dans la suite, présenter brièvement et à titre d'exemples quelques méthodes:

i. Méthode de Gennery

Elle est parmi les premières méthodes comportant un système complet de suivi de cibles avec estimation de la pose 3D à partir d'images [Gennery, 82]. Elle consiste à suivre des objets polyédriques. La comparaison de la projection du modèle avec les données extraites de l'image est faite selon deux types de primitives : Les points d'intérêt et les segments de droite. Pour les points d'intérêt, la mesure prise est la différence entre les coordonnées de la projection du point d'intérêt et le point image correspondant. Pour les segments de droite, une discrétisation, en un ensemble de points distants de trois pixels environ, est appliquée sur chaque segment projeté du modèle. A chacun de ces points est associé le point d'arête de l'image dont le gradient est le plus proche. La mesure prise dépend ainsi de deux mesures : la distance orthogonale qui est la distance entre le point d'arête et le segment de droite et la distance parallèle qui est la distance entre la projection orthogonale du point d'arête et le point modèle.

ii. Méthode de Baker

Cette méthode a été définie dans le cadre du projet européen ESPRIT. Elle propose de suivre des véhicules. La modélisation a été faite par un modèle en fil de fer. La mesure de comparaison entre les données extraites de l'image et la projection du modèle est faite par une fonction dite « évaluation iconique ». Il s'agit d'abord de projeter le modèle en éliminant les segments de droite qui ne sont pas visibles. Dans le voisinage de chaque segment projeté, le gradient de l'image est calculé dans la direction perpendiculaire. Ces gradients sont ensuite moyennés parallèlement au segment et la valeur maximale est évaluée. La probabilité pour qu'un segment ait une telle valeur est estimée à l'aide de tables pré-calculées avec la technique de monté Carlo. Enfin, Ces probabilités sont utilisées pour tous les segments visibles du modèle dans l'image dans un test du χ^2 . Cette fonction d'évaluation est à la base de trois approches différentes pour l'ajustement de la pose. Baker propose de trouver un maximum local de la fonction « évaluation iconique » pour ajuster la pose. Worrall [Worrall et al., 94] propose un raffinement de la pose en utilisant des forces élémentaires. Worrall propose de remplacer l'évaluation de l'effet d'une force 3D, faite dans l'approche antérieure, par la recherche du minimum d'un terme d'erreur en appliquant un algorithme classique de minimisation par la méthode des moindres carrés.

iii. Méthode de Koller

Elle permet aussi de suivre des véhicules modélisés en fil de fer [koller, 94]. La pose est définie par trois paramètres : deux définissent la position du véhicule dans le plan de la route et le troisième définit l'orientation par rapport à la perpendiculaire de la route. La projection du modèle est faite en éliminant les arêtes non visibles, l'extraction des segments est effectuée par la méthode de Korn [Korn, 98] et la mise en correspondance est assurée par la méthode de Dérêche et Faugeras [Dérêche et al., 90].

iv. Méthode de Lowe

La méthode d'estimation définissant la position de l'objet dans l'espace est proposée par Lowe [lowe, 91] en se basant sur un modèle d'approximation polygonale. Des améliorations [lowe, 92] ont été faites sur cette méthode pour qu'elle soit plus robuste en éliminant les faux appariements.

v. Méthode de Harris

Cette méthode propose de suivre un objet rigide en se basant sur un modèle simple. Ce modèle est représenté par les points d'intérêt se trouvant sur des arêtes bien marquées. Cette méthode présente de bonnes performances en poursuivant des objets ayant des vitesses de rotation allant jusqu'à 10 radian/s et une accélération angulaire qui peut aller jusqu'à 100 radian/s.

5.3. Méthodes hybrides 2D/3D

Ces méthodes reposent, à la fois, sur l'estimation du mouvement 2D de l'objet et sur le calcul de sa pose 3D. Le principal avantage de ces méthodes est d'éviter l'étape de prédiction de type Kalman en estimant le modèle de mouvement qui sera exploité pour fournir une initialisation correcte du calcul de pose. Notons, qu'il existe un certain nombre de travaux en ce sens. Marchand [Andrew et al., 03], par exemple, propose une méthode, dans le cadre d'un projet EDF pour des tâches de maintenance et de surveillance en milieu hostile, permettant un suivi robuste et rapide d'objets complexes pouvant être approximativement modélisés par une forme polyédrique.

Un modèle de mouvement affine 2D est estimé, à partir des déplacements orthogonaux calculés le long des projections des arêtes du modèle dans l'image, grâce à un algorithme robuste. Puisque, le modèle de mouvement affine ne permet pas de représenter totalement le mouvement 3D de l'objet, une seconde étape est nécessaire pour recalibrer la projection du modèle de l'objet dans l'image. Cette étape consiste à calculer la pose de l'objet par rapport à la caméra. Il propose, pour cela, une minimisation itérative d'une fonction d'énergie non linéaire par rapport aux paramètres de la pose.

6. Conclusion

Dans ce chapitre nous avons étudié la détection et le suivi de mouvement dans une séquence d'images en commençant par une introduction, les problèmes de détection et ensuite les méthodes de détection et le calcul du mouvement, enfin on a vu quelques approches de suivi du mouvement.

CHAPITRE III : RÉALITÉ AUGMENTÉE

1. Introduction

Avec la démocratisation des périphériques nomades tels que les téléphones combinant une caméra, une bonne puissance de calcul voire même des outils tels qu'un GPS ou une boussole, le grand public montre actuellement un grand intérêt pour la réalité augmentée.

La réalité augmentée est une approche centrée sur l'interaction Homme- Environnement qui vise l'intégration des objets du monde réel dans l'acte interactionnel. Elle cherche à étendre le spectre des utilisations des objets usuels en fournissant à l'utilisateur des fonctions issues d'un système de traitement de l'information. Ces fonctions n'ont pas pour but de remplacer la réalité; elles visent à l'amplifier par l'observation de l'environnement et notamment par l'analyse des actions de l'utilisateur sur cet environnement. Un système de réalité augmentée doit donc recueillir des observations sur l'environnement réel, les traiter et les restituer dans le mode réel.

2. Définition

L'émergence de la réalité augmentée en tant que domaine de recherche coïncide avec la banalisation des dispositifs de visualisation tête-haute dans les laboratoires de recherche en réalité virtuelle vers le début des années 1990.

Par voie de conséquence, l'une des premières définitions données à la réalité augmentée était la suivante : une forme de réalité virtuelle où le dispositif de visualisation tête-haute de l'opérateur est semi-transparent, ce qui permet une vision nette du monde réel. Par la suite, cette définition a évolué pour sortir de ce cadre très restreint. Dans l'article de Milgram, la réalité augmentée a pour but d'augmenter la rétroaction naturelle de l'opérateur avec le monde réel à l'aide d'indices virtuels [Milgram et al., 94]. Cette définition permet de sortir du cadre de l'utilisation des seuls dispositifs de visualisation tête-haute.

La définition donnée par Azuma demeure plus restrictive car elle introduit des conditions supplémentaires : un système de réalité augmentée complète le monde réel avec des objets virtuels (générés par ordinateur) de telle sorte qu'ils semblent coexister dans le même espace que le monde réel [Azuma et al., 01]. Toujours selon lui, les systèmes de réalité augmentée possèdent les propriétés suivantes :

- Combiner des objets réels et virtuels dans un environnement réel,
- Être temps-réel et interactifs,
- Recaler (aligner) les objets réels et virtuels.

La définition de Fuchs et Moreau [Fuchs et al., 04] semble la plus équilibrée :

La réalité augmentée regroupe l'ensemble des techniques permettant d'associer un monde réel avec un monde virtuel, spécialement en utilisant l'intégration d'Images Réelles (IR) avec des Entités Virtuelles (EV) : images de synthèse, objets virtuels, textes, symboles, schémas, graphiques, etc. d'autres types d'association entre mondes réels et virtuels sont possibles par le son ou par le retour d'effort.

3. Historique

- 1966: Ivan Sutherland invente le premier casque qui combine la vue de la réalité et une image de synthèse . L'affichage était effectué par des tubes cathodiques miniatures et le suivi des mouvements de la tête était assuré par un bras mécanique lié au casque.

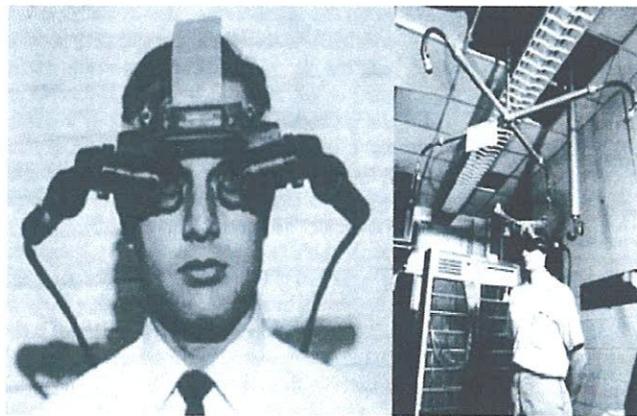


Figure 3.1 : le projet de Sutherland invente

- 1979: McDonnell Douglas développe un casque projetant des informations utiles pour les pilotes de chasse directement sur leur visière.

- 1997: Steve Feiner présente la Touring Machine, le premier système itinérant de réalité augmentée combinant un affichage sur les lunettes à la vue réelle. Le système intègre un GPS et un accès internet par ondes radio.



Figure 3.2: Steve Feiner (la Touring Machine)

- 1999: Hirokazu Kato met à disposition ARToolkit, la première librairie open-source et multi-plateformes dédiée à la création d'interfaces de réalité augmentée.
- 2000: Bruce H. Thomas développe ARQuake, le premier jeu basé sur la réalité augmentée, permettant de jouer à Quake dans un univers réel grâce à la combinaison de caméras, capteurs et GPS.



Figure 3.3 : jeu Quake.

- 2008-2009: Lancement de Wikitude Drive, système de navigation AR pour plateformes Android basé sur NyARToolkit. Il s'agit là de la première application nomade de réalité augmentée disponible pour le grand public. Peu après, des applications similaires furent mises à disposition sur l'iPhone d'Apple.

4. Domaines d'application

La réalité augmentée est un domaine de recherche relativement récent et en plein essor. Les applications sont nombreuses dans des cadres différents tels que la robotique ou la médecine. L'objectif de la réalité augmentée est l'amélioration de la perception et de l'interaction de l'utilisateur avec le monde réel. Les objets virtuels permettent l'affichage des informations que l'utilisateur ne peut pas voir directement ou détecter avec ses propres sens. Ces informations virtuelles aident l'utilisateur à exécuter ses tâches dans le monde réel.

Les applications potentielles de la réalité augmentée sont dans les domaines suivants : médical, visualisation, industrie/maintenance, robotique, militaire, etc.

a. Médical

Dans ce domaine, les utilisations de la réalité augmentée sont de deux types.

Le premier consiste à faciliter les soins et le second à les rendre possibles. En effet, certaines opérations ne sont possibles qu'avec l'aide de la réalité augmentée, certains risques étant fortement diminués grâce à une modélisation 3D. Grâce à des technologies non intrusives, il est possible de soigner les patients avec beaucoup moins de risques (le cancer notamment). Pour le moment, la réalité augmentée permet surtout de valoriser les outils d'imagerie médicale. En revanche, les axes de développement sont multiples. La réalité augmentée est également susceptible de s'appliquer aux sensations tactiles et auditives.



Figure 3.4. : Un exemple avec les outils développés par l'IRCAD

b. Militaire

Dans le domaine militaire, l'utilisation la plus fréquente est l'ajout d'un affichage tête haute ou Head-Up Display (HUD) intégré à la vitre de l'habitacle ou maintenant directement dans les casques des pilotes. L'affichage tête haute consiste à superposer des informations nécessaires au pilotage, à la navigation ou à la réalisation de la mission sur l'environnement extérieur. Il permet au pilote de surveiller son environnement en même temps que les informations fournies par ses instruments de bord.



Figure 3.5 Head-Up Display (HUD)

Cette technologie a par la suite été mise à disposition du grand public dans les voitures haut de gamme avec un affichage tête haute directement sur le pare-brise qui permet de retrouver les informations du tableau de bord sans quitter les yeux de la route. Désormais, combinés à des caméras et autres capteurs, ces systèmes intègrent de nombreuses autres fonctionnalités augmentant la sécurité et le confort ; comme des alertes de franchissement de ligne blanche, de collision, la limitation de vitesse en cours (reconnaissance des panneaux), les informations fournies par un GPS ou même des assistances de parking.



Figure 3.6 : voiture avec affichage haute tête.

c. Industriel

Le but d'utiliser ce principe dans le domaine de l'industrie est évidemment d'améliorer la productivité et de réduire les coûts. De plus, cela permet d'intégrer de nouveaux produits en cours de conception dans leur environnement d'utilisation, comme par exemple insérer une voiture virtuelle dans un trafic réel ou comme ci-dessous un tram dans un modèle de ville.



Figure 3.7 : un tram dans un modèle de ville.

Certaines entreprises développent des outils pour la conception de produits comme des lunettes d'aide à la décision, afin de combiner la vision d'un produit en fabrication avec de

nouvelles pièces ou afin de guider l'utilisateur via la superposition d'un manuel de montage à la vue réelle.

d. Navigation

Ce domaine est le plus développé pour le grand public, les solutions proposées profitant de l'essor des technologies nomades. D'abord utilisée sur ordinateur, la technologie de réalité augmentée est aujourd'hui très répandue sur les systèmes embarqués et les téléphones de troisième génération. La réalité augmentée permet d'enrichir l'expérience de l'utilisateur dans le domaine de la navigation. La géo-localisation permet en effet de superposer des informations sur le lieu où l'on se trouve aux images recueillies par une caméra. Les applications sont alors diverses : tourisme, astronomie, position du soleil, reconnaissance de bâtiments.



Figure 3.8 : la navigation.

5. Problèmes de réalité augmentée

L'un des problèmes les plus importants dans les applications de la réalité augmentée est celui de l'alignement virtuel-réel. Les objets dans le monde virtuel doivent être superposés correctement sur les objets réels, de telle manière à ce que l'utilisateur ait l'impression que les deux mondes réel et virtuel coexistent ensemble. Les applications de réalité augmentée nécessitent beaucoup de précision afin de réaliser un bon alignement.

Les erreurs d'alignement sont difficiles à contrôler et elles sont divisées en deux types : statique et dynamique. L'erreur statique apparaît quand le point de vue de l'utilisateur et l'objet dans le monde restent fixes. L'erreur dynamique apparaît quand le point de vue de l'utilisateur ou de l'objet est en mouvement dans l'environnement.

Dans les systèmes actuels, les erreurs dynamiques sont généralement la source principale des erreurs d'alignement. Les problèmes liés à l'éclairage, aux occultations et aux mouvements représentent aussi des verrous dans l'élaboration d'un système de réalité augmentée efficace [Madjid, 07]].

5.1. Alignement spatial erreurs statiques

On définit l'erreur statique par l'erreur d'alignement entre objet réel et objet virtuel, déterminée lorsque l'utilisateur est en position statique. Trois types d'erreurs statiques sont définis :

- distorsion optique,
- erreur du système de suivi,
- mauvaise estimation des paramètres.

La distorsion optique est un problème bien connu et qui est dû aux propriétés des lentilles des caméras. L'erreur du système de suivi est générée par les capteurs électromagnétiques sensibles aux distorsions du champ magnétique. La mauvaise estimation des paramètres est l'erreur la plus étudiée en réalité augmentée. Une estimation fiable nécessite de mettre en œuvre des méthodes robustes basées sur une définition de contraintes dans la procédure de calibration. À partir d'une modélisation du système, on s'intéresse alors à la calibration de ces différentes parties. On peut alors utiliser des méthodes automatiques (par vision, auto-calibrage) ou des méthodes basées utilisateurs (méthodes simples et efficaces) [Grasset et al., 01]

5.2. Alignement temporel : temps de latence

Le recalage dynamique des images de synthèse sur le monde réel est une étape cruciale dans l'élaboration d'un système de réalité augmentée. Les données des différents systèmes d'acquisition, tel que les dispositifs de suivi d'objet et de formation d'images, doivent être enregistrées spatio-temporellement avec le point de vue de l'utilisateur. Chaque dispositif a un retard associé à l'observation du monde et à l'affichage sur le système de visualisation. La différence des retards entre les divers dispositifs d'acquisition ou systèmes de traitement est appelée : latence relative. Elle représente une source du mauvais repérage et doit être réduite. Les sources de retard sont classifiées en six catégories et représentées par : retard d'arrivée des données, retard de traitement, retard du rendu, retard d'affichage, retard de synchronisation et retard de mise à jour.

Beaucoup d'auteurs ont traité la latence et ont essayé de réduire ses effets. Dans [Funkhouser, 93], [Holloway, 95], [Wloka, 93], les auteurs ont développé une technique de calcul de temps critique qui est très commercialisé pour sa vitesse. L'application se rend constamment compte du temps de traitement et tâche de le réduire. La réduction de la latence est basée sur la réduction de la précision des calculs, ce qui ne pourrait pas être toujours recommandée.

La plupart des systèmes graphiques en temps réel sont dotés de hautes performances calculatoires afin de réduire la latence. Dans [Olano et al., 95], les auteurs étudient un système fonctionnant en parallèle, ce qui minimise la latence et permet de réduire les retards dus aux systèmes de visualisation. La parallélisation diminue la latence en augmentant la vitesse des calculs, ainsi, dans [Wloka, 95], l'auteur utilise un processeur très performant pour permettre le traitement rapide des données.

5.3. Éclairage

Une augmentation réaliste demande de tenir en compte de l'éclairage par les sources lumineuses présentes dans la scène réelle. La cohérence, le modèle d'éclairage et la gestion des ombres est en effet, un problème de synthèse d'images et demande une connaissance des caractéristiques des sources lumineuses, leurs positions, couleurs, puissances, etc.

Le degré de luminosité des points de l'image définit le quotient de l'intensité lumineuse d'une surface. Une bonne luminance entraîne une meilleure exploitation des détails dans les zones de l'image et permet de caractériser les objets présents dans l'image et facilite leur détection.

De nombreux travaux de gestion d'éclairage ont été menés pour incruster les objets virtuels dans les images. Récemment plusieurs auteurs se sont intéressés à des solutions temps réel simplifiées: estimation de direction [Hartmann et al., 03] estimation par light probe [Abusant et al., 03] estimation et éclairage basé image [Kanara et al., 02] re-éclairage du modèle initiale avant intégration [Matsuka et al., 02] gestion des ombres [Haller et al., 03] modification par projection [Raskar et al., 01]. Dans ce contexte, Sugano [Sugano et al., 03] ont étudié la relation réel/virtuel par rapport à la notion d'ombres et montrent que la direction de l'ombre est l'information dominante dans un cas de scènes statiques et que dans des scènes dynamiques seule la forme globale est un paramètre important.

5.4. Occultation

La réalisation de tâches en réalité augmentée suppose que les indices visuels nécessaires sont disponibles à tout instant : ils ne doivent donc jamais être occultés partiellement ou totalement. Différentes approches ont été proposées pour faire du suivi en temps réel par des méthodes de vision afin de résoudre le problème d'occultations. Cela consiste en des techniques de suivi de plusieurs marqueurs positionnés dans divers endroits de l'environnement. Toutefois, la non-fiabilité de ces techniques de vision pour la gestion d'occultations, peut alors nécessiter la conservation d'un contrôle utilisateur pour réinitialiser le suivi quand le système n'arrive plus à poursuivre les cibles.

Une approche hybride avec un suivi par vision (lent) et un autre capteur de position (rapide mais peu précis) reste sans doute une des meilleures solutions. Foxlin et Naimark [Foxlin et al., 03] ont réalisé un système de tracking vision-inertiel afin de gérer les occultations dans des conditions de luminosité variées. D'autres travaux s'intéressant à la gestion d'occultations en utilisant les estimateurs robustes ont été réalisés dans la littérature.

5.5. Flou de mouvement

Les méthodes basées vision prennent en compte directement les propriétés de l'image, elles permettent un recalage précis de la scène et des éléments qui y sont ajoutés dans le cas des applications de réalité augmentée. Ces méthodes empêchent, cependant, le suivi lors d'un mouvement brusque du dispositif de tracking à cause du flou de mouvement et de la disparité importante générée dans l'image.

Le flou de mouvement se produit en raison de l'acquisition non-instantanée des images par la caméra. Quand un objet ou la caméra se déplace, les projections se déplacent aussi à travers la formation d'image pendant la durée de l'exposition dans la direction du mouvement. Des déplacements dépassant la fréquence d'acquisition de la caméra provoquent un flou de mouvement et peuvent sévèrement compromettre la visualisation de l'image et toutes les opérations qui y sont appliquées (filtrage, seuillage, détection, segmentation, etc).

Dans ce cas, le mouvement projeté peut représenter la résultante du mouvement de plusieurs objets. On pourrait par exemple obtenir un mouvement nul, du fait de combinaisons inversées de deux ou plusieurs objets en mouvement. Dans le cas continu, le champ de déplacement apparent est défini pour tous les points $P(x, y, t)$ du plan image par un vecteur déplacement $d(x, y, t)$.

4.4. Estimation du mouvement

Pour estimer le mouvement à partir du mouvement apparent, il faut admettre que l'intensité reste inchangée ou elle varie d'une façon prédictible. Cette hypothèse dite de conservation s'exprime par l'équation DFD (Displaced Frame Difference) des différences entre les images déplacées, autrement dit les images prises entre les instants t et $t + dt$.

$$DFD = I(x + dx, y + dy, t + dt) - I(x, y, t) = 0$$

–L'estimation du mouvement réel à partir du mouvement apparent peut être abordée de deux manières différentes

– L'estimation des vecteurs déplacements dans le plan image $d(x, y, t) = (dx, dy)$ estimés entre les images a t et $t + dt$

– L'estimation du vecteur vitesse $v(x, y, t) = (vx, vy)$. Les vecteurs déplacements estimés peuvent varier en espace et en temps.

4.5. Le calcul du mouvement

C'est l'extraction des caractéristiques physiques de l'environnement à partir d'une information fournie par l'image qui ne permet de les recouvrir que partiellement. Par exemple, on déduit la vitesse réelle d'un objet qui se déplace dans un environnement tridimensionnel, or l'information recueillie de l'image ne nous donne que deux paramètres : la vitesse horizontale vx et la vitesse verticale vy , ils sont appelés aussi les vitesses apparentes.

Pour vaincre cette difficulté, il nous faut des informations de plus, la distance de l'objet et la distance focale de la camera pour trouver les vraies coordonnées de l'objet avec seulement les deux coordonnées apparentes sur notre écran. On envisage plusieurs méthodes de calcul de mouvement, chacune de ces méthodes s'appuie sur un algorithme qui lui est particulier.

En général le temps réel est exigé même pour la compression dans le cas d'une transmission indirecte. Le temps de calcul est donc une préoccupation primordiale dans le traitement des séquences d'images.

Exemple :

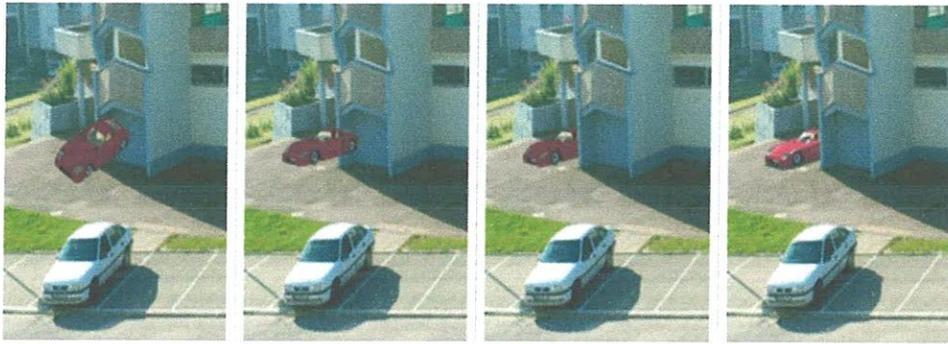


Figure 3.9 : Incrustation d'une voiture virtuelle dans une scène réelle

6. Architecture générale d'un système de RA

En effet, pour un système de RA de type postproduction, trois parties distinctes sont nécessaires avant la composition (*Figure 3.10*) [Bouzenada , 08] :

- ✓ la partie synthèse d'images qui consiste à modéliser les objets virtuels à incruster dans la séquence réelle et à les illuminer,
- ✓ la partie calcul des occultations qui consiste à déterminer les objets réels venant occulter un ou plusieurs objets virtuels afin d'assurer la cohérence géométrique de la scène,
- ✓ enfin, le noyau du système représenté par la partie estimation du mouvement ou calibration de la caméra qui consiste à déterminer les paramètres de la caméra. Ces derniers sont utilisés à la fois pour le calcul des images virtuelles et la détermination des masques d'occultation.

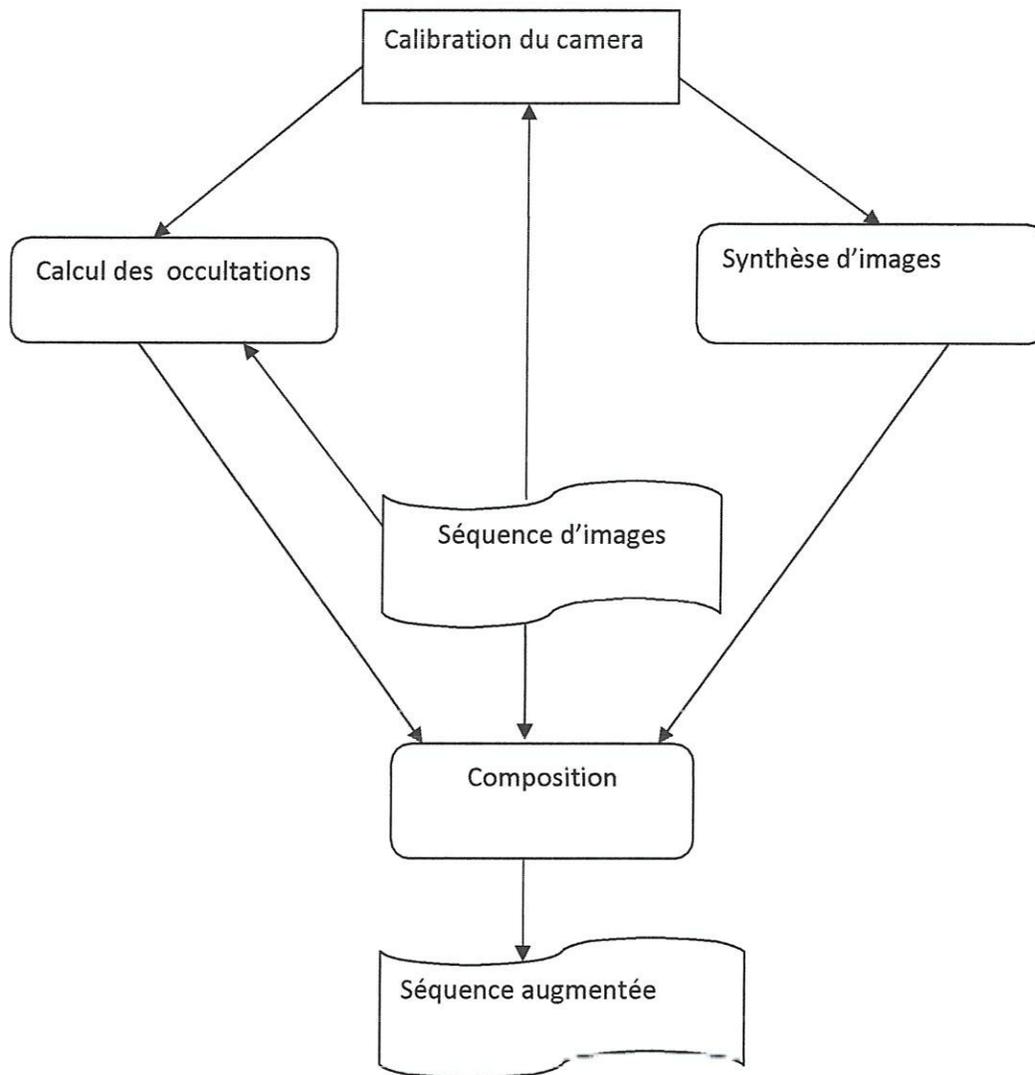


Figure 3.10 : Architecture générale d'un système de RA.

7. Classification des systèmes de réalité augmentée

Au vu des nombreux critères qui constituent un système de réalité augmentée, il convient d'établir une classification des divers systèmes qui existent. Dans la littérature, nous trouvons essentiellement deux systèmes de classification :

- une taxonomie fonctionnelle, qui classe les systèmes suivant la manière dont l'association entre le réel et le virtuel est traitée,
- une taxonomie technique, qui porte sur les différentes techniques d'affichage ou de vision du monde réel.

7.1. Fonctionnelle

Cette classification, empruntée à Fuchs et Moreau dresse une liste de catégories fonctionnelles basée sur la nature des associations entre le réel et le virtuel, indépendamment du dispositif employé pour faire l'association [Fuchs et al., 04]. Cette taxonomie essentiellement concentrée sur les augmentations de type visuel est présentée selon un ordre croissant de la contribution des augmentations visuelles dans la scène réelle.

- Fonctionnalité “réalité documentée” et “virtualité documentée”
- Fonctionnalité “Réalité à compréhension ou à visibilité augmentée”
- Fonctionnalité “Association du réel et du virtuel”
- Fonctionnalité “Association comportementale du réel et du virtuel”
- Fonctionnalité “Substitution du réel par le virtuel” ou “Réalité virtualisée”

7.2. Technique

Cette taxonomie a été introduite par Milgram [Milgram, 94]. Elle permet d'introduire un certain nombre de notions attachées à la réalité augmentée. Elle commence par définir cinq classes de systèmes de réalité augmentée, classifiés selon le type du dispositif d'affichage et la proportion respective du réel par rapport au virtuel.

Il convient tout d'abord d'établir une liste des dispositifs d'affichage possibles :

- les moniteurs vidéo
- les casques de réalité virtuelle (casques de RV)
- La vision sera directe (Réalité augmentée en vision directe - RAVD) si l'opérateur voit directement les objets ou le monde réel. Elle sera indirecte (Réalité augmentée en vision indirecte - RAVI) si ce dernier la perçoit par le biais d'une caméra par exemple.
- Le repère de travail du système est égocentrique s'il s'avère critique de localiser l'opérateur ou la direction de son regard. Dans le cas contraire, le repère sera exocentrique.
- Enfin, la vision des objets réels et virtuels peut s'effectuer à l'échelle 1:1 ou non. En l'occurrence ce sera toujours le cas pour la réalité augmentée en vision directe.

Il est à noter que certains de ces systèmes sont plus proches des systèmes de RV purs que des systèmes de réalité augmentée. Cette taxonomie établit et classe ces systèmes sur une échelle continue entre la réalité augmentée et la réalité virtuelle, échelle représentée la réalité mixée.

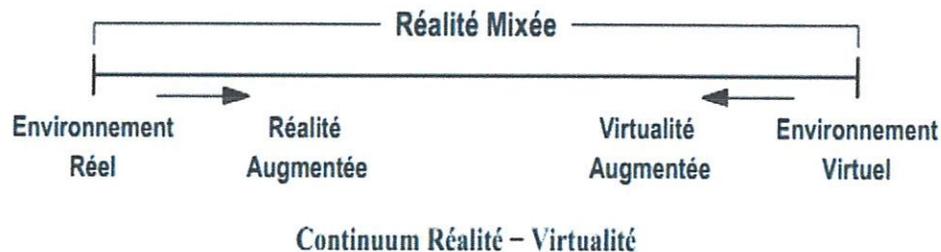


Figure 3.11 : réalité mixte.

Les systèmes sont ensuite classifiés suivant trois critères de discrimination :

- le degré de connaissance du monde,
- le degré de fidélité de représentation du monde,
- le degré d’immersion de l’opérateur par rapport aux informations affichées.

8. Les architectures modulaires existantes

Le domaine de la réalité augmentée est riche en solutions architecturales développées dans le but de résoudre le problème de la modularité des applications développées, et de faciliter la réutilisation du code écrit. Ainsi, ces dernières années, environ une trentaine de solutions de ce type], nous avons sélectionnés quelques unes parmi les plus représentatives[Jean-Yves,05].

8.1. COTERIE :

COTERIE (Columbia Object-oriented Test bed for Exploratory Research in Interactive Environments) était un projet de l’université de Columbia de 1996 à 1999. Ce système supporte la création d’environnements virtuels avec plusieurs utilisateurs simultanés qui peuvent interagir avec une palette hétérogène de dispositifs d’affichage et de périphériques d’entrée.

Cet environnement est développé sur la base d'un modèle de programmation orientée objet, modulaire et multi-processus. Il permet de communiquer de manière transparente via un modèle client-serveur et des objets distribués. Les applications développées sur cette plateforme comprennent EM-MIE un gestionnaire d'environnement fournissant des techniques d'interaction pour les environnements de RA [Butz et al., 99] et MARS (Mobile Augmented Reality Systems) qui est la série des prototypes de RA développée par l'université de Columbia [Feiner et al., 97].

COTERIE se base sur un graphe de scène distribué dont les propriétés sont localement modifiées pour chaque système client. Si cet environnement est particulièrement adapté au prototypage rapide d'applications collaboratives de réalité augmentée, sa faiblesse réside dans le fait que l'essentiel des informations passent par le graphe de scène dont on ne peut se dispenser. Le prototypage de l'application se fait au moyen d'un langage de scripts qui permet de modifier rapidement le comportement des objets du graphe de scène.

8.2. StudierStube :

Développé par l'université de Vienne depuis 1997, la Studier Stube [Schmalstieg et al., 02] est l'architecture spécifique à la RA la plus ancienne encore en activité. Le principe est que chaque utilisateur dispose d'un espace de travail qui peut-être mise en commun avec d'autres, en partageant un graphe de scène commun. La synchronisation des graphes de scène est prise en charge par un protocole réseau robuste. Les espaces de travail peuvent se découvrir les uns les autres grâce à un gestionnaire de sessions centralisé.

Pour ce qui est de la gestion des capteurs et des périphériques d'entrée, ce projet s'appuie sur OpenTracker [Reitmayer et al., 01], un système s'appuyant sur une description XML (eX-tensible Markup Language) des flux de données, ce qui favorise une grande flexibilité et permet de gérer plusieurs capteurs de position et d'orientation simultanément. Le développeur peut programmer entièrement à l'aide de scripts OpenInventor en utilisant des nœuds prédéfinis. Il peut également implémenter ses propres nœuds en C++. Ce système convient pour les projets de réalité augmentée collaboratifs et distribués.

8.3. Tinmith :

L'université d'Australie du Sud démarra en 1998 le projet TINMITH [Piekarski et al., 03] Ce dernier, axé sur la description des flux de données, est une librairie d'objets hiérarchisés calqué sur le modèle des systèmes de fichiers Unix. Cet ensemble de classes

permet de gérer les flux de données issus des capteurs, les différentes opérations de filtrage des données ainsi que le rendu final des applications. Cet ensemble d'objets écrits en C++ s'appuie sur un système de callback et un mécanisme de sérialisation du flux de donnée employant le XML. L'ensemble des communications entre les objets est géré par un graphe de flux de données. Ce système, qualifié de hiérarchisé, est organisé suivant un paradigme de couches que le flux de données doit traverser, de l'acquisition des données jusqu'à l'étape de rendu de la scène qui va mélanger réel et virtuel.

Cette architecture, bien que plus contraignante pour le développeur, est adaptée aux systèmes de réalité augmentée mobile, particulièrement pour les systèmes embarqués grâce à l'effort d'optimisation des briques élémentaires du système.

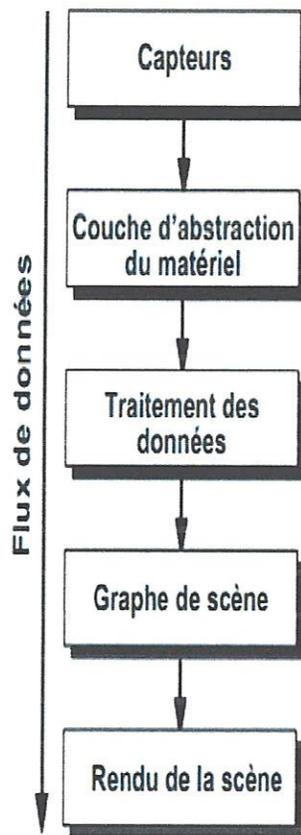


Figure 3.12 : Flux de données et disposition de l'architecture en couches du système Tinnith

8.4. DWARF :

Initié en 2000 par l'université de Munich, le projet DWARF [Bauer et al., 01] (pour Distributed Wearable Augmented Reality Framework) est un système de composants distribués entièrement décentralisé. Le système est constitué d'un ensemble de services basés

sur CORBA (Common Object Request Broker Architecture) orientés vers une architecture générique pour la réalité augmentée.

Chaque service est décrit en utilisant des fichiers XML et sont indépendants les uns des autres, ce qui leur permet de fonctionner sur différentes machines d'un réseau. Les services sont sollicités dynamiquement à l'exécution, formant un graphe de flux de données distribué. Pour démarrer, le système ne nécessite qu'un composant pour gérer les services. Ce dernier tentera de découvrir si d'autres services sont présents sur le réseau ou non, ce qui permet d'intégrer de nouveaux services à l'exécution.

DWARF est sans conteste l'un des systèmes les plus aboutis en matière de réalité augmentée distribuée. Toutefois, ce système présente certains inconvénients :

- la découverte des services se fait par diffusion sur le réseau complet des méta-données associées à ces services, ce qui entraîne une charge supplémentaire sur le réseau
- DWARF est étendu en fonction des applications de réalité augmentée, et ne donne pas de modèle récurrent aux applications de réalité augmentée
- le système décrit une macro-architecture et ne peut pas traduire les détails de cette dernière. Ceci veut dire que les composants peuvent renfermer plusieurs sous-constituants différents dont l'un d'entre eux, s'il est modifié, entraîne la modification de l'ensemble du composant
- ce système est dépendant, dans une large mesure, d'un environnement distribué, ce qui le rend difficile à employer pour des petites applications de réalité augmentée.

8.5. AMIRE :

AMIRE [Abawi et al., 04] [Amire, url] [Dörner et al., 02] (pour Authoring Mixed Reality) est un projet européen comprenant plusieurs partenaires dont Siemens et le Fraunhofer AGC qui s'est déroulé sur vingt-sept mois, de 2002 à 2004 Le but avoué de ce projet était de définir et de développer une architecture système qui permette de concevoir et d'implémenter rapidement des applications de réalité mixte sans connaître les détails des technologies sous-jacentes qui sont employées.

Le paradigme employé ici est également la programmation orientée composants Il est à noter que cette architecture comporte une granularité à deux niveaux les gemmes (qui sont des

micro-composants) et les composants (qui sont une agrégation de plusieurs gemmes). Les deux classes de composants comportent chacune une interface de configuration, des “slots” d’entrée, des “slots” de sortie qui peuvent être reliés aux slots d’entrée d’autres composants. Par ailleurs, cette architecture possède une ouverture sur les systèmes distribués par le biais de l’utilisation de CORBA.

Enfin, ce système contient également un éditeur graphique d’applications. Ce dernier permet à un utilisateur de connecter les divers composants tout en étant guidé pas à pas pour suivre des règles génériques de conception des applications de réalité mixte. L’éditeur sauvegarde la configuration dans un fichier XML. Ce dernier est ensuite chargé par un gestionnaire de composants qui va lancer l’application.

8.6. DART :

Plus récemment, en 2003, l’université de Columbia a démarré un autre projet appelé DART [MacIntyre, 04] [Dart, url] (Designer’s Augmented Reality Toolkit). Ce projet, orienté vers l’expérimentation et le prototypage rapide d’applications de réalité augmentée, s’appuie sur le moteur de Macromedia Director. Il emploie donc les paradigmes de base de Director qui font de lui un outil d’édition de contenus (il permet de paramétrer le comportement et les propriétés des objets virtuels).

Ce système inclut également un support pour les fonctions de bas niveau qui gèrent les capteurs, caméras et autres capteurs nécessaires. Cette édition présente l’avantage de se faire simplement à l’aide d’une interface graphique, ce qui permet de paramétrer en quelques clics une application de réalité augmentée.

Les développements nécessitent une connaissance du langage C++ et du langage de scripts de Director : LINGO.

9. Conclusion

Dans ce chapitre nous avons présenté la réalité augmentée, suivi d’une petite historique et les domaines d’application, puis les problèmes rencontrés, l’architecture générale d’un système de réalité augmentée et les types de systèmes, et enfin quelques systèmes de réalité augmentée.

CHAPITRE IV : CONCEPTION ET IMPLÉMENTATION

1. Introduction

Après avoir présenté les différentes techniques de détection et de suivi et quelques notions sur la réalité augmentée. On va présenter dans ce chapitre la conception et l'implémentation de notre projet. On utilise une Méthode basée sur la détection spatiale des objets de forme quadrilatérale ayant des motifs particuliers.

2. Conception

Pour incruster un objet 3D dans une séquence vidéo il faut suivre trois étapes :

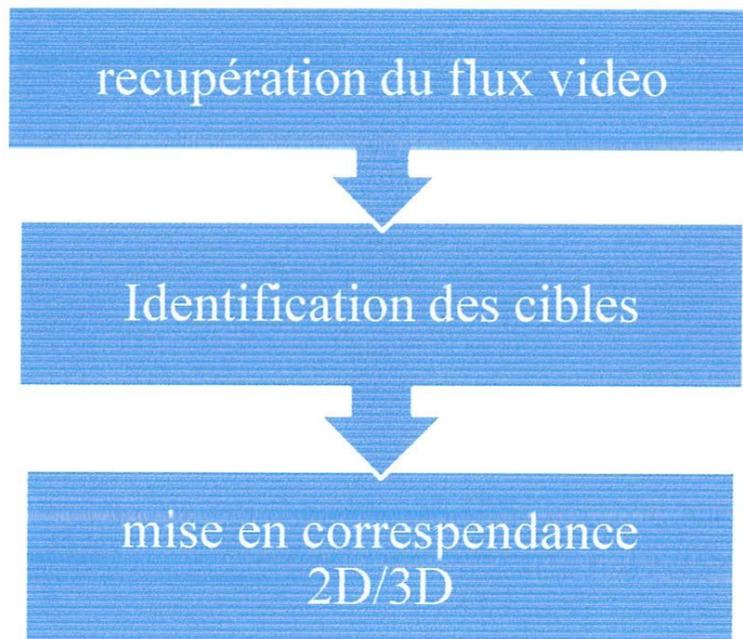


Figure 4.1 : étapes d'incrustation d'objets 3D.

Plusieurs systèmes d'identification de cibles ont été présentés dans la littérature. Toutefois, les trois systèmes les plus connus sont : ARToolKit, CyberCode et le système de cibles circulaires d'Intersense. Dans notre travail on a utilisé ARToolKit.

3. Présentation d'ARtoolKit :

ARToolkit est une librairie de fonctions conçu à la base au langage C/C++, et on trouve d'autres versions pour java, matlab, flash. Cette librairie est open-source et multiplateformes dédiée à la création d'interfaces de réalité augmentée. Elle a été développée initialement par le docteur Hirokazu Kato de l'Université d'Osaka en 1999 et est toujours mise à jour par le Human Interface Technology Laboratory (HITLab) de l'Université de Washington[ARToolkit,url]. L'utilisation est libre dans le cadre d'un usage non-commercial. Des licences commerciales sont disponibles sous le nom ARToolWorks.

Les systèmes basées ARToolKit en générale inclus certaines caractéristiques qui sont:

- ❖ Le suivi de la position et de l'orientation avec une seule caméra,
- ❖ Utilisent des algorithmes de suivi qui utilisent des carrés noirs simples comme marqueurs,
- ❖ La capacité d'utiliser n'importe quel marqueur carré avec motif,
- ❖ Le calibrage de la caméra est simple grâce aux fonctions données,
- ❖ Assez rapide pour des applications AR en temps réel,
- ❖ Coût d'utilisation très bas,
- ❖ Une bibliothèque multiplateforme (Windows, Linux, Mac OS X, SGI),
- ❖ Open Source avec licence GPL pour un usage non commercial.

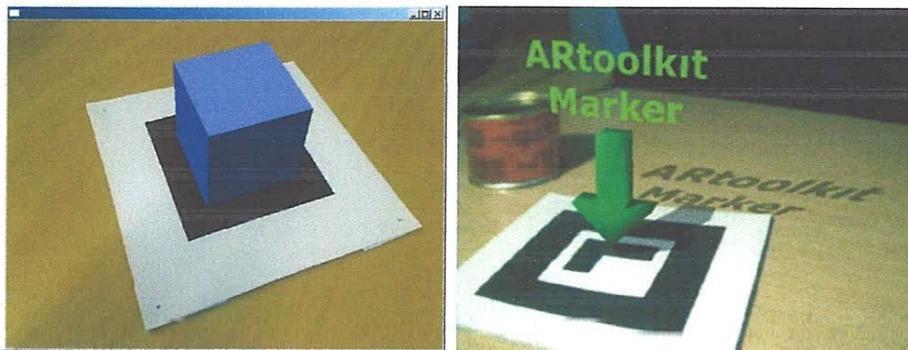


Figure 4.2 : des exemples d'application ARToolKit

Bien que plusieurs méthodes puissent être imaginées pour cela, on utilise un système de marqueurs (Pattern). Un marqueur est un motif simple sur fond blanc entouré d'un cadre noir.

L'algorithme doit tout d'abord détecter puis identifier ce marqueur. Il en déduit en temps réel sa position et son orientation par rapport à la caméra.



Figure 4.3 : Modèles de marqueurs utilisés par ARToolKit.

4. Principe de fonctionnement

4.1. Algorithme de détection :

L'algorithme de détection consiste en une suite d'opérations sur chaque image du flux vidéo afin d'une part de repérer la présence d'un marqueur, puis de l'identifier parmi les différents marqueurs chargés dans l'application. Il fonctionne ainsi :

- i. La caméra capture la vidéo et l'envoie vers l'ordinateur.
- ii. Le programme binarise l'image puis recense tous les cadres noirs dans chaque image de la vidéo (*figure 4.5*).
- iii. Pour chaque cadre trouvé, le programme utilise des formules mathématiques pour déterminer sa position et son orientation vis à vis de la caméra (*figure 4.9*).
- iv. Le motif présent à l'intérieur de chaque cadre est comparé avec les patterns chargés dans le programme afin de lui associer l'objet 3D qui lui correspond.
- v. Cette augmentation, qui prend la forme d'un objet 3D, est alors générée à partir de la position et de l'orientation du marqueur et est superposée à l'image capturée. Sa position, orientation et échelle sont alors ajustés par rapport au marqueur en temps réel.

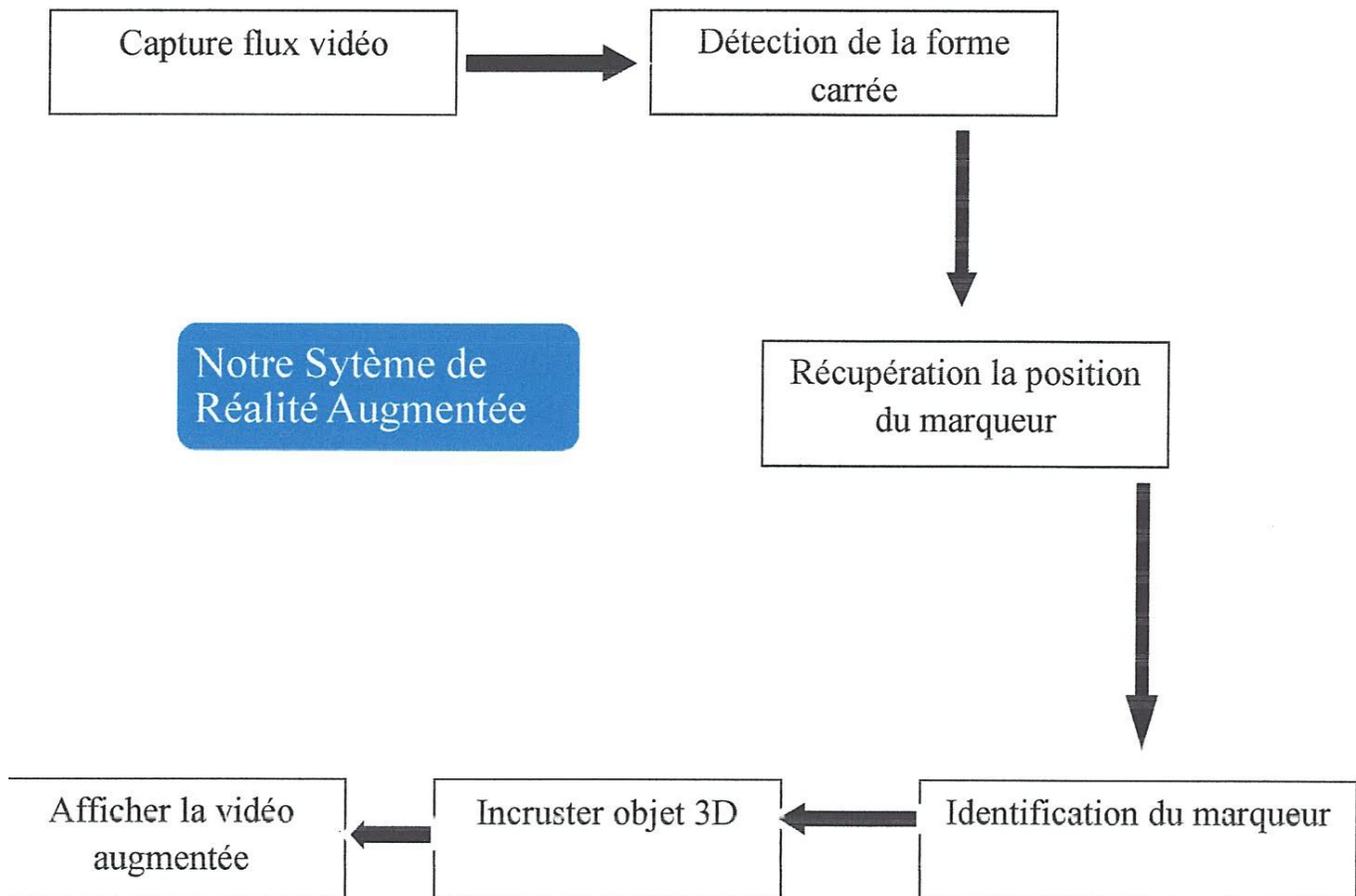


Figure 4.4 : les étapes de fonctionnement de l'algorithme

a. Binarisation

La première étape fondamentale après acquisition d'image, de cette reconnaissance consiste à "binariser" l'image reçue afin de passer celle-ci en noir et blanc pures. Cette binarisation est évaluée selon un paramètre nommé *seuillage* correspondant au taux de luminosité nécessaire pour déterminer si un point de l'image sera noir ou blanc.

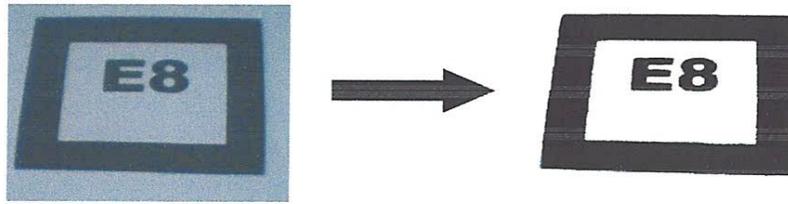


Figure 4.5 : Binarisation de l'image

b. Recherche des composants connexes :

L'étape suivante porte sur la recherche dans l'image binarisée de composants liés afin de dresser une liste de ces derniers. Etant donné la simplicité des motifs utilisés, ces composants seront des éléments du symbole central au marqueur et pourront donc être utile dans les étapes ultérieures, notamment pour la normalisation du marqueur et évidemment la confrontation aux motifs stockés en mémoire.

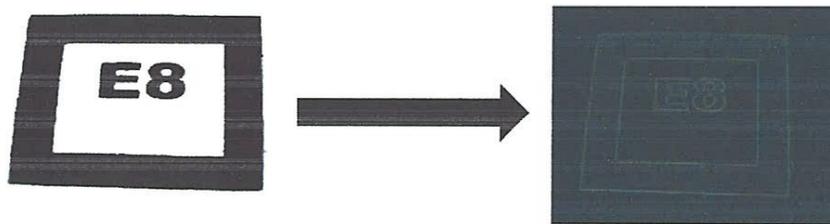
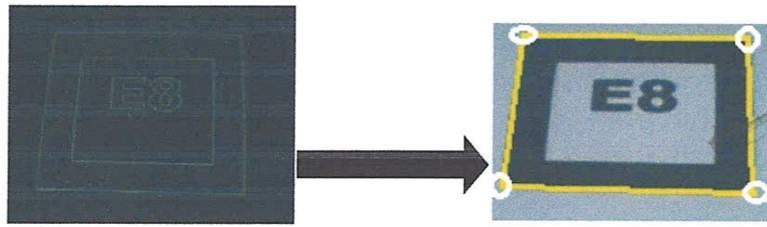


Figure 4.6 : Recherche des composants par détection de contours

c. Extraction des segments et des coins de la cible :

Cette étape consiste à faire l'extraction des contours, lignes et coins du marqueur afin de pouvoir effectuer sa normalisation, puisque le fait d'utiliser un motif carré rend relativement aisé la détection de ses coins et arrêtes, donc on dit que cette étape est primordiale pour l'étape suivante qui est la mise en correspondance entre le marqueur et l'objet 3D correspondant.

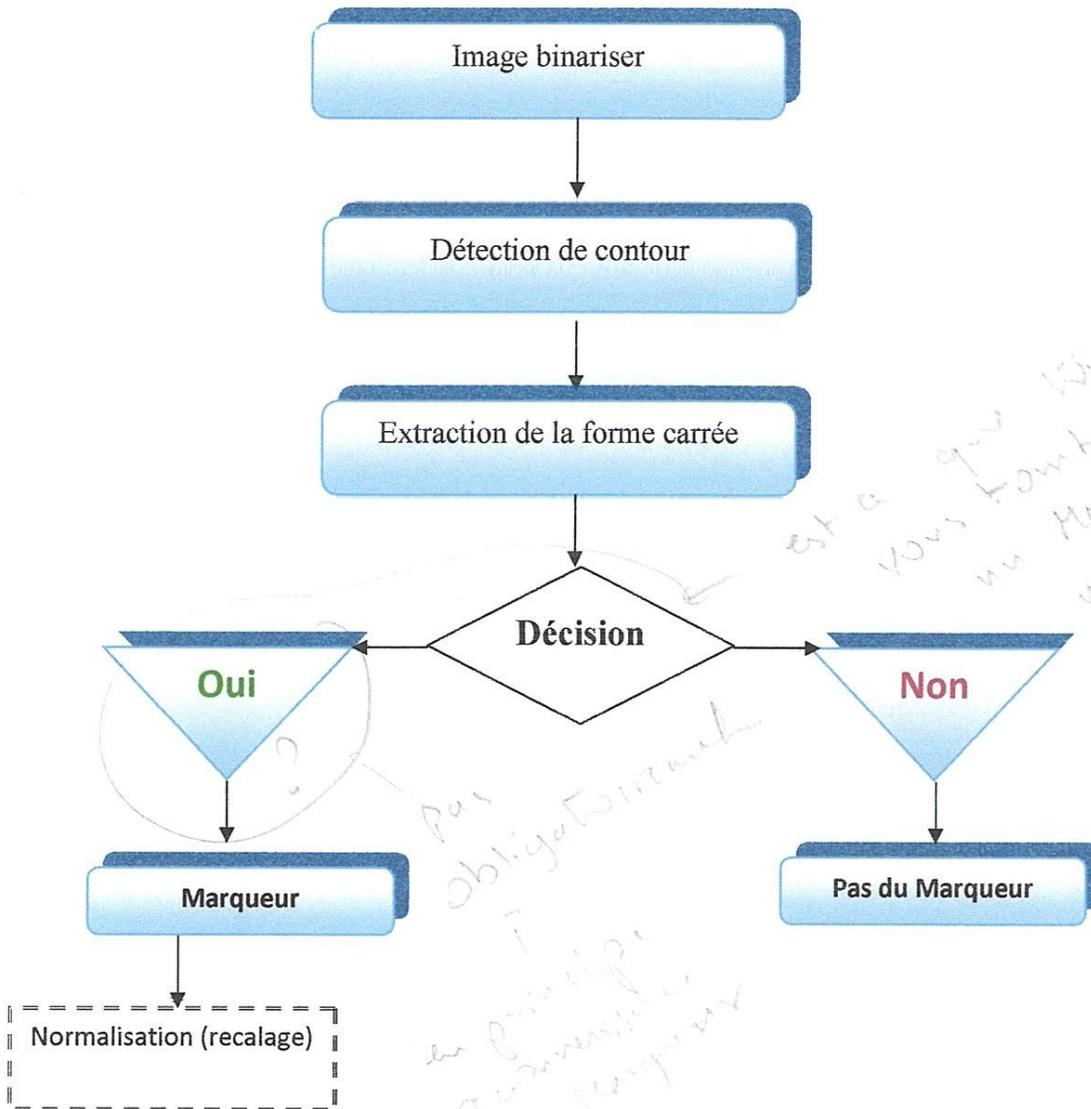
Il ne faut en effet pas oublier que le marqueur n'est pas forcément présenté perpendiculairement à la caméra, il est très probable que celui-ci soit incliné suivant un ou plusieurs degrés de liberté, l'image du marqueur inclinée résultante laissera donc apparaître un parallélogramme.



à pas forcément détecté

Figure 4.7 : Extraction des contours, lignes et coins

On peut récapituler les étapes précédentes dans ce schéma :



*est ce que l'ajout
vous tombe bien
un Marqueur dans
un rectangle
noir.*

*Pas
obligatoirement
en principe
recommandé
de Marqueur*

Figure 4.8 : processus de reconnaissance de marqueur

d. Normalisation

Bien entendu dans la majorité des cas le motif sera déformé par l'orientation du marqueur vis-à-vis de la caméra. C'est là que le carré noir trouve toute son utilité comme on l'a constaté dans l'étape précédente, car à partir de la position de chacun de ses sommets il est possible de normaliser le motif afin de l'identifier. L'image est également redimensionnée.

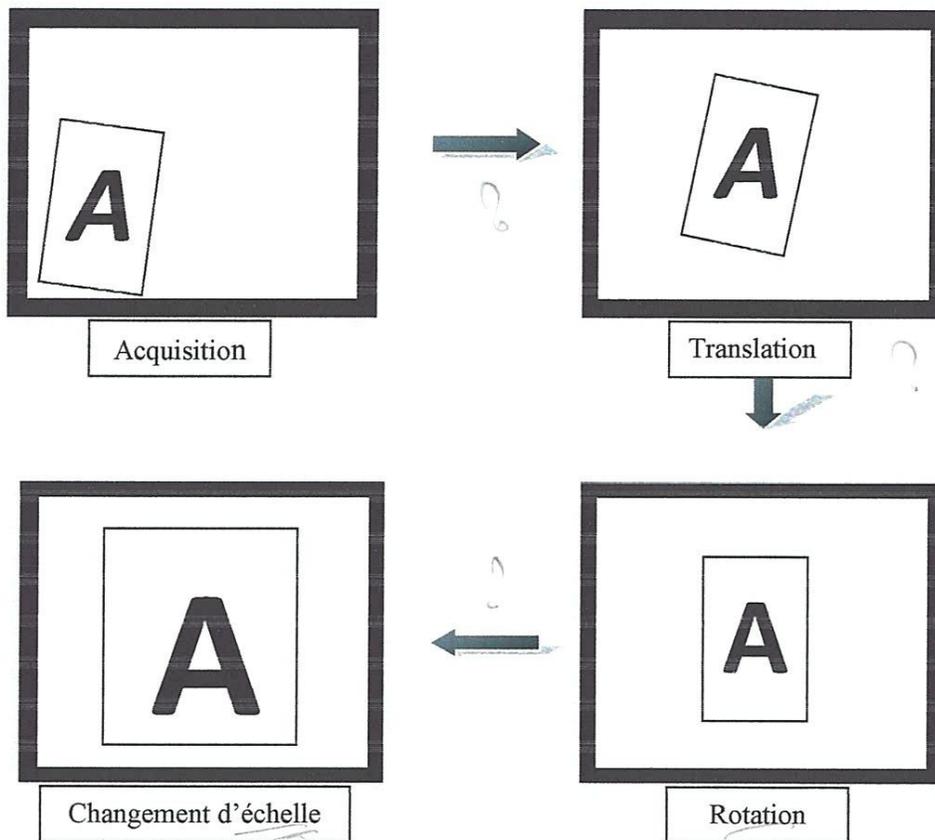


Figure 4.9 - Principe de normalisation

Et voici une matrice 3D homogène qui peut générer toutes les transformations possibles et nécessaires pour la normalisation de notre motif.

$$\begin{array}{l}
 \begin{array}{|l}
 \tau_{absolu}(hom) \\
 y_{absolu}(hom) \\
 z_{absolu}(hom) \\
 w_{absolu}(hom)
 \end{array}
 \end{array}
 =
 \begin{array}{|c|}
 \text{rotation} \\
 \text{scale} \\
 \text{translation} \\
 \hline
 \begin{array}{|c|}
 M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\
 M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\
 M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \\
 M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4}
 \end{array}
 \end{array}
 \times
 \begin{array}{|l}
 x_{local} \\
 y_{local} \\
 z_{local} \\
 1
 \end{array}$$

e. Identification du marqueur

Elle constitue la comparaison du motif détecté avec ceux stockés en mémoire afin de procéder au traitement adéquat par la suite. Les patterns stockés sont des fichiers texte où chaque pixel est représenté par sa valeur en niveaux de gris (entre 0 et 255).

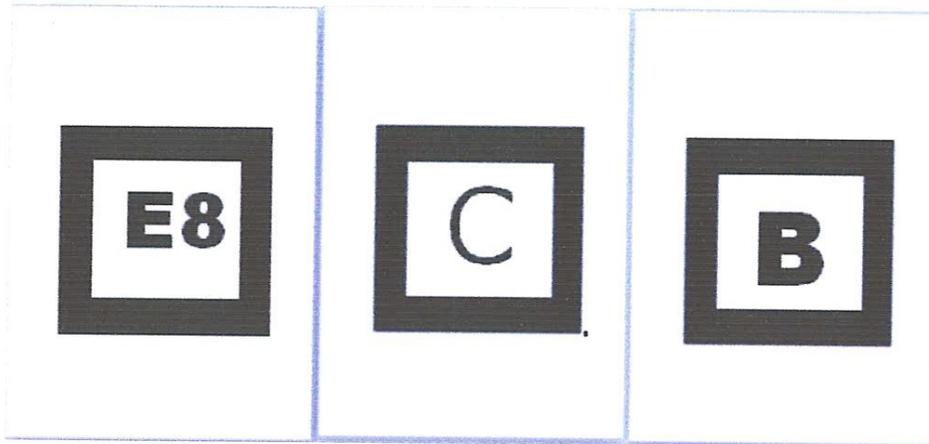


Figure 4.10 : exemple de nos marqueurs

5. Création du marqueur

La création du marqueur se fait par les étapes suivantes :

- Création des marqueurs selon la *figure 4.10*.
- Création d'un exemple normalisé et enregistré dans la base des marqueurs dans le programme, pour être utilisé après dans l'étape de mise en correspondance ; et la création de cette base se fait avec le **Marker Generator** : il faut placer notre pattern devant la caméra et Marker Generator va détecter les côtés du carré. Ce travail se présente dans la (*Figure 4.11*).

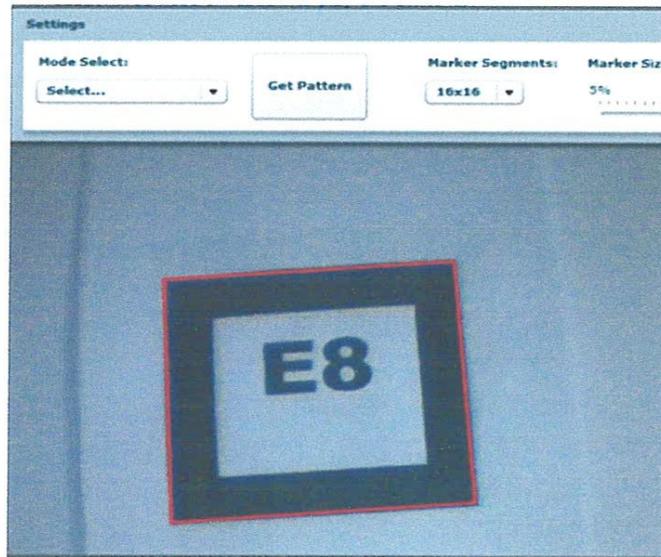


Figure 4.11 : Exemple de la fenêtre de création d'un marqueur

6. Création d'un objet 3D

Pour construire un objet 3D il y a plusieurs logiciels comme Blender, Maya, Autocad, 3D Studio Max...etc, et dans notre application on a utilisé 3D Studio Max. et la figure « *Figure 4.12* » présente un objet 3D qu'on a conçu pour notre système de RA.

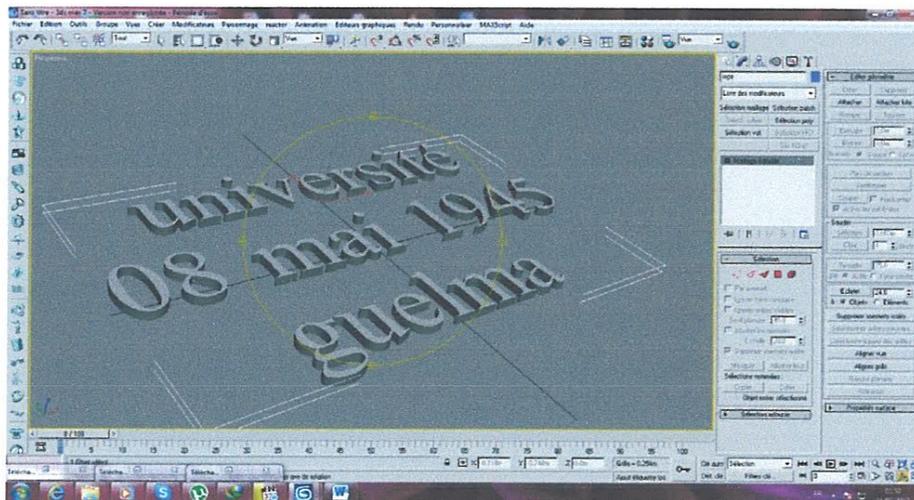


Figure 4.12 : création texte 3D avec 3D studio max

7. Implémentation

Cette partie est réservée pour le côté technique et applicatif du travail, elle abordera les différentes étapes de l'implémentation du modèle conçu précédemment.

7.2. Environnement de développement

a. Environnement matériel.

- **Micro-ordinateur :**

Le micro-ordinateur utilisé tout au long de la réalisation du travail est un « HP G6031 EA », équipé de la version 6.1 de Windows 7 Édition Familiale Premium 32bit, et doté des capacités suivantes :

- Processeur : AMD Athlon (tm) 64*2 Dual-Core processor TK-53 1.70 GHz,
- Mémoire Vive : 2Gb,
- Carte graphique : NVIDIA GeForce Go 6100, 825 Mb.

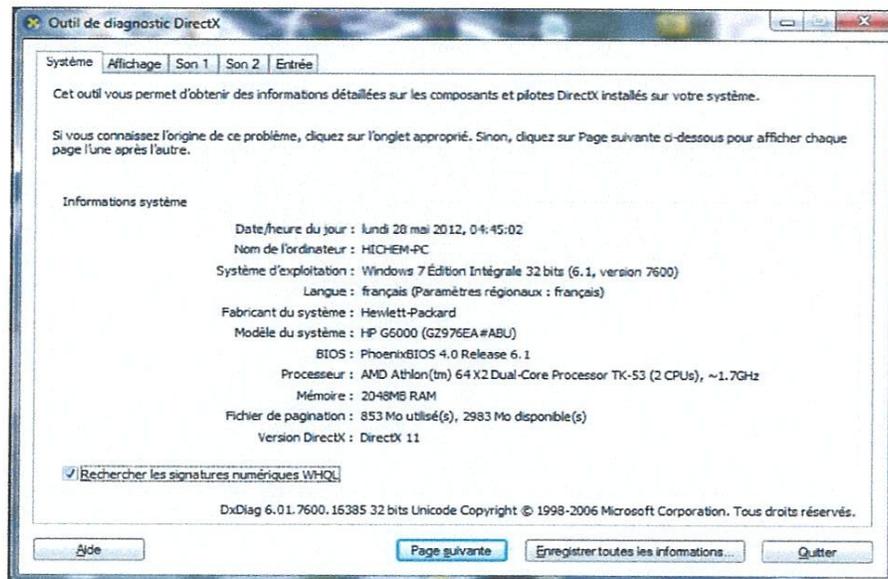


Figure 4.13 : imprime écran du diagnostique direct X (système).

- **Une webcam :**

Le matériel d'acquisition utilisé est une webcam qui a les caractéristiques suivantes:

- ✓ Marque : Havit
- ✓ Type : PC Camera-168 #2
- ✓ Interface : USB

b. Environnement logiciel

b.1 Le langage de programmation

La plate-forme Java est une plate-forme produite par Sun Microsystems créé par James *Gosling et Patrick Naughton*, permettant de développer et d'exécuter des programmes écrits en langage Java indépendante de tout processeur et de tout système d'exploitation.

La plate-forme se compose principalement d'un moteur d'exécution (appelé une machine virtuelle) et d'un compilateur fourni avec un ensemble de bibliothèques standards dont il existe plusieurs implémentations pour divers matériel et système d'exploitation, de façon à ce que les programmes Java puissent s'exécuter de façon identique sur chacun d'entre eux. La version installée et utilisée est la version `jdk 1.7.0_03`.

b.2 Éditeur du code

Eclipse est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent. Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciel, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, de reporting, etc. Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n'importe quel langage de programmation.

Le projet Eclipse est pour cela organisé en un ensemble cohérent de projets logiciels distincts, sa spécificité tenant à son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de l'atelier logiciel doivent être développées en tant que *plug-in* bâti autour de l'**IDE Eclipse Platform**.

Eclipse recouvre donc notamment également à cet effet tout un Framework de développement logiciel fournissant des briques logicielles à partir desquelles développer tous ces outils. C'est la raison pour laquelle Eclipse est présenté dans la littérature tout autant comme un EDI ou comme un Framework.

La version utilisée dans notre travail est la version 3.6, appelé « Eclipse Helios Service Release 1 » sortie en juin 2010.

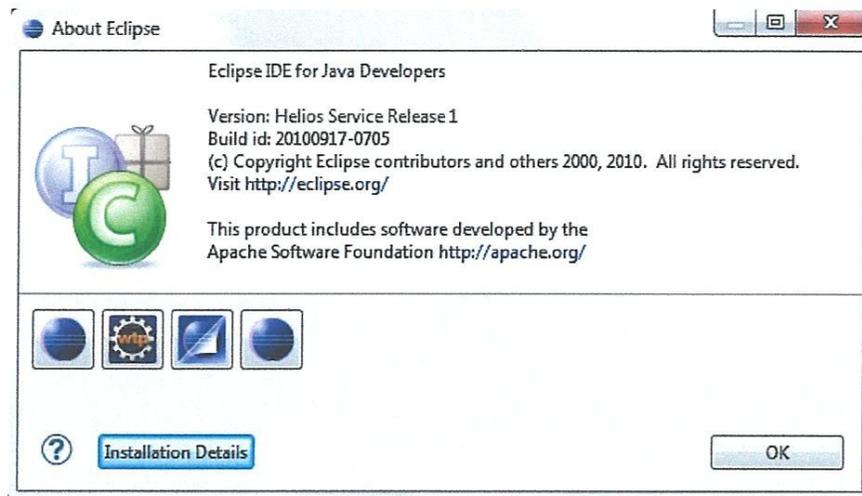


Figure 4.14 : version de l'éditeur du code

c. La bibliothèque utilisée

c.1 Java Media FrameWork

Java Media Framework (JMF) est une architecture unifiée pour la synchronisation, le traitement, l'affichage de données temporelles comme les données audio, la vidéo, le format MIDI, etc. à l'intérieur d'applications indépendantes ou d'applets. Elle a été utilisée pour la récupération du flux vidéo.

c.2 Java 3d

Est une interface de programmation (*Application Programming Interface* ou *API*) pour la plateforme Java visant la synthèse d'image 3D basée sur les graphes de scène. Cet API fait appel soit aux fonctions d'OpenGL ou Direct3D. Depuis sa version 1.2, Java 3D est développé sous l'organisation du *Java Community Process*. Elle a été utilisée pour la manipulation des objets 3D.

c.3 Nyartoolkit

Développée par un Japonais connu sous le nom de Nyatla (le 'Ny' de l'appellation). Toute la documentation est écrite en japonais, et est donc plus difficilement abordable malgré les traductions à la volée de Google Translate. NyARToolKit n'est pas un binding d'ARToolKit. La bibliothèque a été complètement repensée en programmation orientée objet et réécrite en Java, C#, C++, et est disponible pour Android et Processing.

7.2 Interface principale

- ✓ Affichage du web Cam.
- ✓ Le lancement de la réalité augmenté.

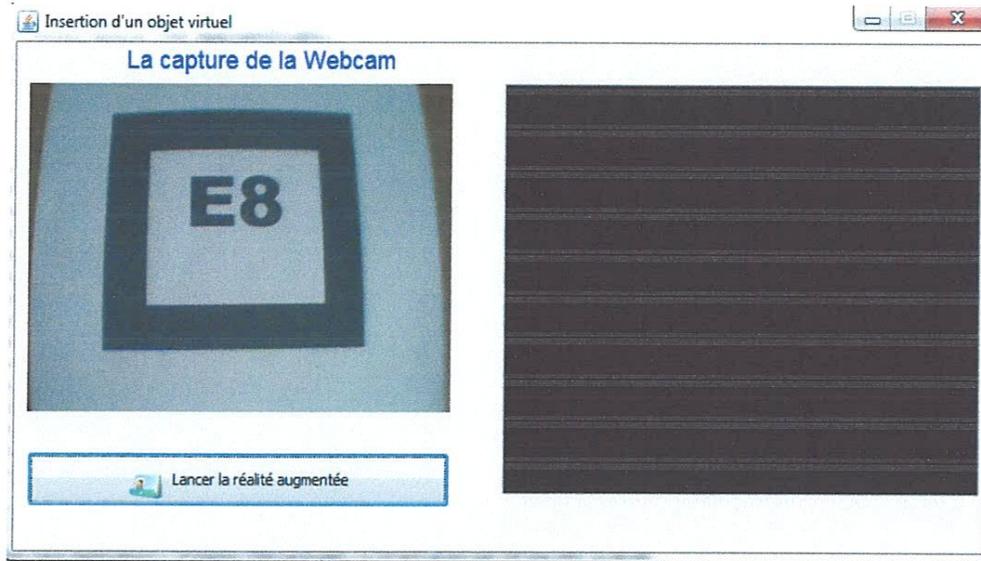


Figure 4.15 : interface principal

a. Les teste du l'application

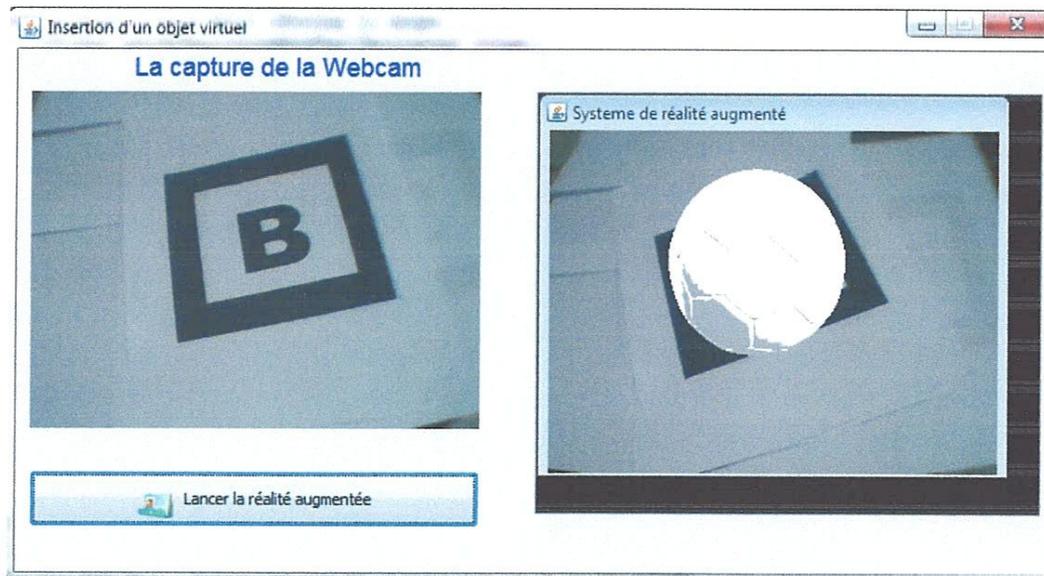


Figure 4.16 : test 1 marqueur avec motif B et un ballon

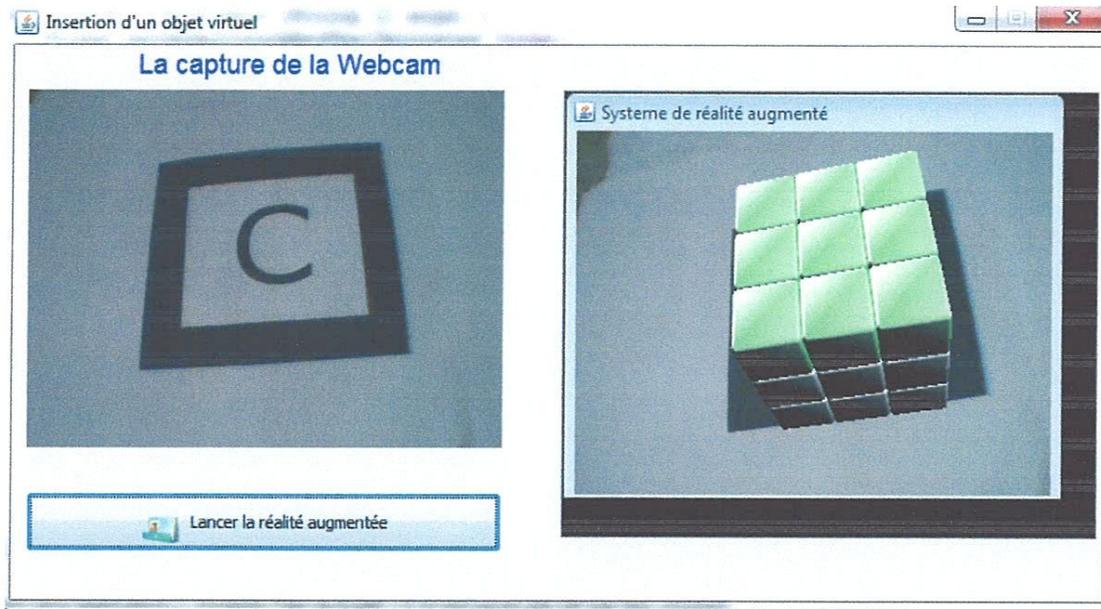


Figure 4.17 : test marqueur avec motif c et un cube

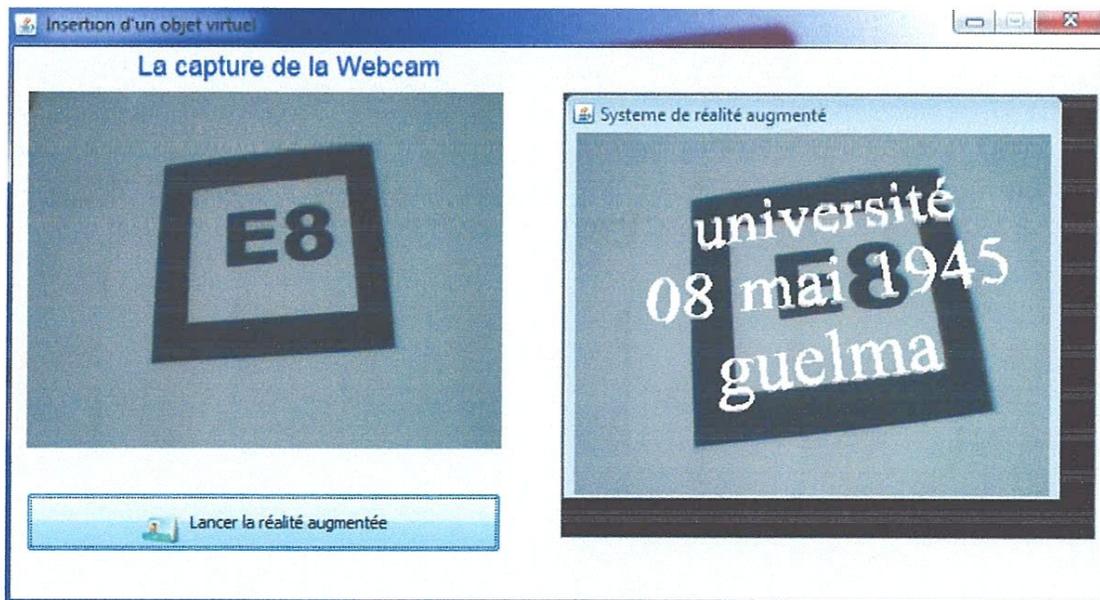


Figure 4.18 : test 3 marqueur avec motif E8 et un texte 3D

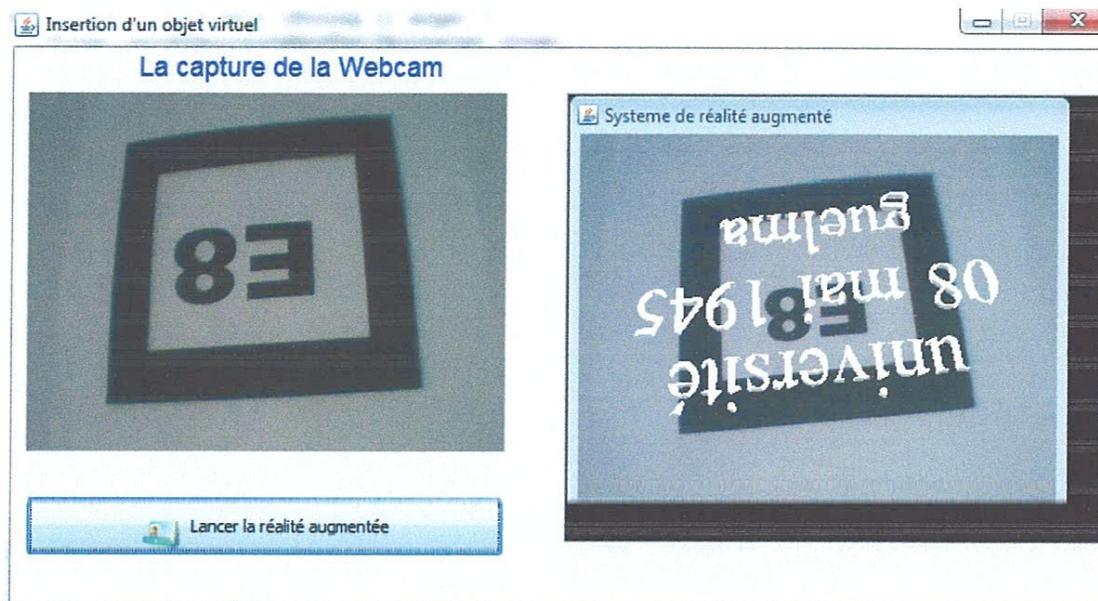


Figure 4.19 : test 4 marqueur avec motif E8 inversé et un texte 3D

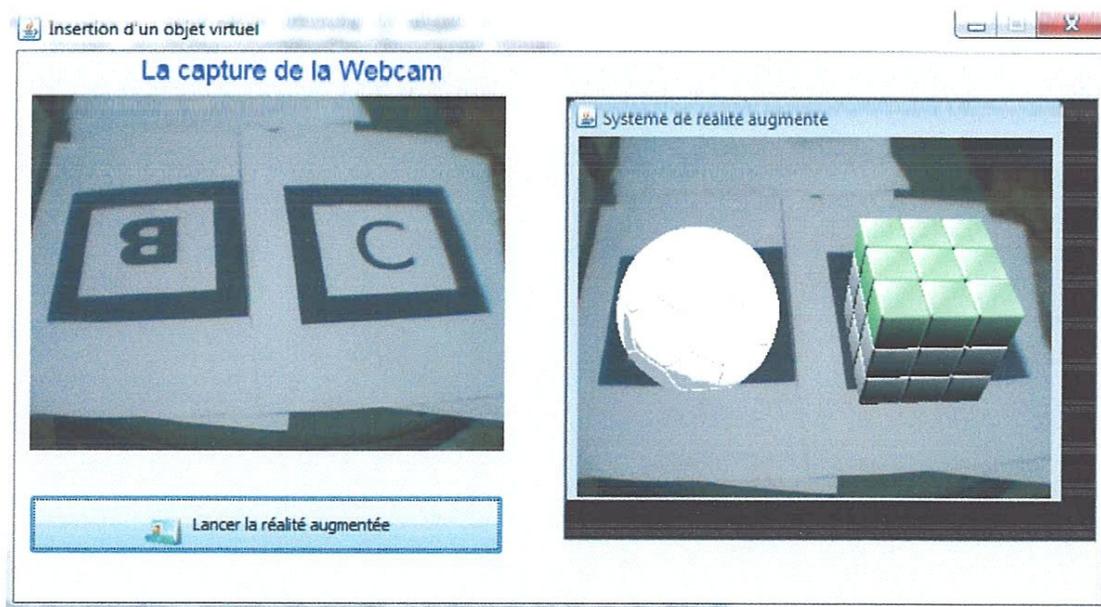


Figure 4.20 : test 5 avec deux marqueurs

8. Conclusion

Dans ce chapitre nous avons vu la conception et l'implémentation de notre travail , pour la première partie on a présenté les étapes de l'algorithme utilisé, la deuxième partie est conservé pour l'implémentation du projet , on a présenté l'environnement de développement et les outils utilisés puis les tests de validation de l'application et à la fin on a mentionné les difficultés rencontrées dans la réalisation de notre travail.

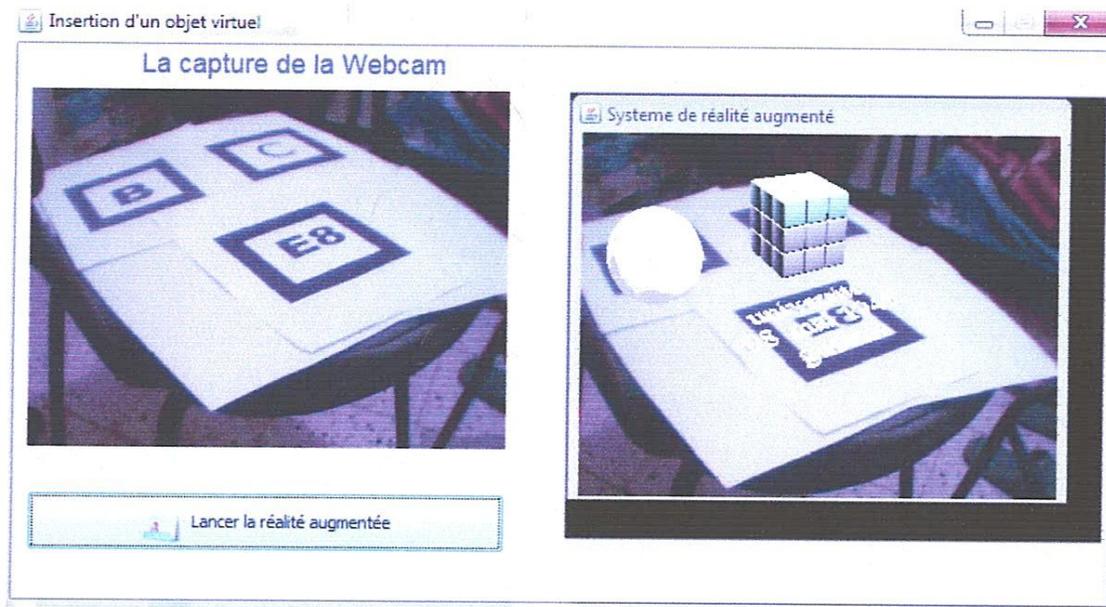


Figure 4.21 : test 6 avec trois marqueurs

b. Difficultés rencontrées :

- ⊗ Le plus gros problème été la familiarisation avec la 3D. Nous n'avions pas de connaissances réelle-dessus, et cela nous avons couté beaucoup de temps pour se former sur le sujet. De plus nous avons vu qu'il y avait un problème avec java3D. En effet, on a été confronté à une multitude de petits bugs assez inexplicables.
- ⊗ Ensuite nous avons remarqué que parmi les nombreux marqueurs censés être reconnus, seuls certains fonctionnaient réellement. On a constaté qu'il y avait un conflit entre les marqueurs lorsque les motifs étaient similaires ou proches, et ça est dû à la résolution d'acquisition des marqueurs, puisque si elle augmente on a de bons résultats mais l'application devient lourde, sinon si on applique une résolution trop faible on gagne le temps mais on perd de précision, nous pensons qu'il faut une caméra de meilleure qualité pour qu'il analyse les petites différences présentes entre les marqueurs.
- ⊗ Encore une fois, étant donné qu'on se sert de caméra, l'environnement lumineux et l'arrière plan derrière l'utilisateur jouent énormément. Il faudra donc un bon éclairage.
- ⊗ Et aussi le problème d'intégrer la bibliothèque Artoolkit, qui a pris beaucoup de temps, puisqu'il y a un manque de documentations, et la documentation qui existe est souvent en japonais.

CONCLUSION GÉNÉRALE

La réalité augmentée (RA) est une discipline récente. Elle est située au carrefour de plusieurs disciplines. Elle a pour but l'intégration simultanée du réel et du virtuel. Le réel correspond aux scènes concrètes relatives à un environnement donné qu'on désire augmenter avec d'autres. Le virtuel correspond aux scènes synthétiques construites par ordinateur ou d'autres scènes réelles préenregistrées. L'augmentation consiste alors à les mixer de manière intelligente et en temps réel.

Le domaine de la réalité augmentée constitue le croisement de plusieurs domaines, dont la réalité virtuelle, la vision artificielle, l'infographie, les mathématiques et l'optique sont les piliers. C'est ce qui fait que plusieurs de ses résultats dépendent de ceux obtenus dans d'autres domaines.

En effet, un bon système de RA est un système qui permet de garder à tout moment un alignement correct entre les objets virtuels et les objets réels de la scène. Pour chaque mouvement accompli par un utilisateur du système les objets virtuels doivent suivre la position et l'orientation des objets réels de la scène. Cela est possible, grâce à un suivi efficace des objets réels dans chaque point de vue. La détection et le suivi d'objet constitue, alors une phase importante dans l'insertion d'un objet virtuelle pour la réalité augmentée.

Notre travail repose sur un système sur l'utilisation de marqueurs spécifiques, qui se composent d'un cadre carré de couleur noire entourant une image binaire représentant un identifiant unique. Ces marqueurs sont détectés, reconnus et leurs positions ainsi que leurs orientations sont mesurées en temps réel. Cette méthode consiste à réduire le temps du calcul avec un bon suivi de l'objet pour la réalité augmentée.

Dans le premier chapitre on a vu une introduction sur la vision par ordinateur, puis la modélisation et le calibrage d'une caméra, et l'extraction d'information 3D

Dans le deuxième chapitre on a vu une introduction sur la détection de mouvement quelque problème de détection et les méthodes utilisées, après introduction sur l'estimation et le calcul du mouvement et puis on a vu les différentes approches de suivi et la fin une conclusion

pas nouvelle

Dans le troisième chapitre introduction a été réservé pour la présentation du domaine réalité augmentée sa définition et historique e, puis les problèmes rencontraient pendant incrustation d'objet virtuelle. L'architecture globale d'un système de RA et quelque Les architectures modulaires existantes.

Le dernier chapitre de ce mémoire a exposé comment insertion d'objet a été implémenté, pour ce faire ce chapitre on a réservé une partie pour la conception du travail et l'algorithme de détection et d'insertion pour la deuxièmes partie de ce chapitre on a été présenté les outils utilisé pour ce travail et le résultat qu'on a obtenus et les difficultés rencontrées pendant la réalisation du projet.

Malgré tous les problèmes et les contraintes qu'on a affrontées pendant la réalisation du projet qui demande beaucoup d'effort et de performance, nous avons bien atteint notre objectif. Ce travail peu être intégré dans plusieurs domaine comme les Smartphone, les jeux vidéo, l'éducation...etc. Les agences de communication se verront obligées de se diriger vers cette technologie car elle est de plus en plus demandée. On pense que l'on pourra déjà voir un bousculement économique autour de la réalité augmentée, car ce ne sont plus seulement des grosses sociétés qui en font la demande.

BIBLIOGRAPHIE

[Abawi et al., 04] D.F. Abawi, R. Dörner, M. Haller, et J. Zauner (2004). Efficient mixed reality application development. Dans 1st European Conference on Visual Media Production (CVMP), pages 289–294, Londres. IEEE.

[Agusanto et al., 03] K. Agusanto, L. Li, Z. Chuangui, and N. W. Sing. Photorealistic rendering for augmented reality using environment illumination. In ISMAR'03, pages 208–216, Tokyo, Japan, 2003.

[Andrew et al, 03] Comport Andrew I., Éric Marchand, François Chaumette. A real-time tracker for markerless augmented reality. Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03), 2003.

[Azuma et al., 01] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, et B. MacIntyre (2001). Recent advances in augmented reality. IEEE Computer Graphics and Applications, 21(1) :34–47.

[Barron et al., 94] Barron, J. L., Fleet, D. J. et Beauchemin, S. S. (1994). Performance of optical flow techniques. Int. Journal of Computer Vision, 12(1):43–77.

[Bauer et al., 01] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, et M. Wagner (2001). Design of a component-based augmented reality framework. Dans Proceedings of the International Symposium on Augmented Reality (ISAR).

[Baumberg et al., 94] Baumberg A.M., Hogg D.C., "An efficient method for contour tracking using active shape models", 1994

[Black et al., 98] Black M.J. and Jepson A.D., 1998, "Eigen tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation", Int'l J. Computer Vision, vol. 26, no. 1, pp. 63-84.

[BONNET] P. BONNET - Cours de Traitement d'Images – USTL (PROJECTION DE L'ESPACE TRIDIMENSIONNEL).

[Bouzenada ,08] BOUZENADA Mourad Incrustation d'objets virtuels dans des séquences vidéo pour la réalité augmentée temps réel Université MENTOURI Constantine 2008

[Brunelli et al., 95] Brunelli R. and Poggio T., 1995, Template Matching: Matched Spatial Filter and Beyond, A.I. Memo 1549, Massachusetts Inst. of Technology.

[Butz et al., 99] A. Butz, T. Höllerer, S. Feiner, B. MacIntyre, et C. Beshers (1999). Enveloping users and computers in a collaborative 3d augmented reality. Dans IWAR '99 : Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, page 35, Washington, DC, USA. IEEE Computer Society.

[Canny, 86] Canny J.F., "A Computational approach to edge detection", IEEE trans. Pat. Anal. and Mach. Intel., Vol 8(6), pp.34-43, 1986.

[Darell et al., 96] Darell T., Essa I.A. and Pentland A.P., 1996, "Task-Specific Gesture Analysis in Real-Time Using Interpolated Views" IEEE Trans. PAMI, vol. 18, no. 12, pp. 1236-1242.

[Dérêche et al., 90] Dérêche R. and Faugeras O., "Tracking Line Segments", Image Vision Computation, Vol. 8, 1990, pp 261-270.

[Dörner et al., 02] R. Dörner, C. Geiger, M. Haller, et V. Paelke (2002). Authoring mixed reality. A component and framework-based approach. Dans First International Workshop on Entertainment Computing (IWEC 2002), Makuhari, Chiba, Japon.

[ELE6812] ELE6812. Traitement d'image & analyse (vision artificielle). *est ce un livre ?*

[Elgammal et al., 00] Elgammal, A. M., Harwood, D. et Davis, L. S. (2000). Non-parametric model for background subtraction. In Proc. 6th European Conf. on Computer Vision (ECCV 2000), volume II, pages 751-767, Dublin, Irlande.

[Endres et al., 05] C. Endres, A. Butz, et A. MacWilliams (2005). A survey of software infrastructures and frameworks for ubiquitous computing. Mobile Information Systems Journal, 1(1).

[Feiner et al., 97] S. Feiner, B. MacIntyre, T. Hollerer, et A. Webster (1997). A touring machine : Prototyping 3d mobile augmented reality systems for exploring the urban environment. Dans ISWC '97 : Proceedings of the 1st IEEE International Symposium on Wearable Computers, page 74, Washington, DC, USA. IEEE Computer Society.

[Foxlin et al., 03] E. Foxlin and L. Naimark. Vis-tracker : A wearable vision-inertial self-tracker. In VR'03 : Proceedings of the IEEE Virtual Reality 2003, pages 199-206, Los Angeles, California, USA, March 2003.

[Fuchs et al., 04] P. Fuchs et G. Moreau (2004). Le traité de la réalité virtuelle, volume 1 : Fondements et interfaces comportementales. Presses de l'École des Mines de Paris, deuxième édition.

[Funkhouser et al., 93] T. A. Funkhouser and C. H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In Computer Graphics (SIGGRAPH'93 Proceedings), pages 247-254, Anaheim, California, USA, August 1993.

[**Gennery, 82**]Gennery D.B., "Tracking Known Three-Dimensional Objects", 2nd National Conference on Artificial Intelligence, Pittsburg, PA, August, 1982, pp 13-17.

[**Grasset et al., 01**]R. Grasset and J. D. Gascuel. Environnement de réalité augmentée collaboratif : Manipulation d'objets réels et virtuels. In AFIG'01 (Actes des 14èmes journées de l'AFIG) , pages 101–112,November 2001.

[**GRAVA, 03**]Cristian GRAVA. Compensation de mouvement par réseaux neuronaux cellulaires. Application en imagerie médicale. Thèse Institut National des Sciences Appliquées de Lyon, 2003.

[**Haller et al., 03**] M. Haller, S. Drab, and W. Hartmann. A real-time shadow approach for an augmented reality application using shadow volumes. In VRST'03 : Proceedings of the ACM symposium on Virtual reality software and technology, pages 56–65, Osaka, Japan, 2003.

[**Hartmann et al., 03**]W. Hartmann, J. Zauner, M. Haller, T. Luckeneder, and W. Woess. Shadow catcher : A vision based illumination condition sensor using artoolkit. In IEEE International Augmented Reality Toolkit Workshop , pages 44–45, Tokyo, Japan, October 2003.

[**Heikkila et al., 06**]Heikkila, M. et Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. IEEE Trans. on Pattern Analysis and Machine Intelligence, 28(4):657{662.

[**Holloway, 95**]R. L. Holloway. Registration Errors in Augmented Reality Systems . PhD thesis, UNC Chapel Hill, Department of Computer Science, USA, 1995.

[**Horn et al., 81**] Horn, B. K. P. et Schunck, B. G. (1981). Determining optical flow. Artificial Intelligence, 17:185{203.

[**Huttenlocher et al., 92**]Huttenlocher D.P., Noh J.J., Rucklidge W.J., "Tacking non-rigid objects in complex scenes", TR92-1320, Computer science departement, cornell university, dec. 1992.

[**IMN459 ., 08**] IMN459. Fondements de la vision par ordinateur, chapitre I. B. Cyganek and P. Siebert. An introduction to 3D Computer Vision Techniques and Algorithms, Wiley, 2008.

[**Jain et al., 79**]Jain, R. et Nagel, H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence.

[**Jean-Yves,05**] Jean-Yves DIDIERContributions à la dextérité d'un système de réalité augmentée mobile appliqué à la maintenance industrielle 2005

[**Julien**] , M. Julien Richefeu & M. Antoine Manzanera Ecole Nationale Supérieure de Techniques Avancées Détection de mouvement par capteur intelligent

[**Kanbara et al., 02**] M. Kanbara and N. Yokoya. Geometric and photometric registration for real-time augmented reality. In ISMAR'02 : Proceedings of the International Symposium on Mixed and Augmented Reality, pages 279–280, Darmstadt, Germany, 2002.

[**Kass et al., 98**] Kass M., Witkin A.P. and Terzopoulos D, "Snakes: Active Contour Model", International Journal of Computer Vision, Vol. 1, No. 4, jan 1998, pp. 321-331

[**Koller et al., 94**] Koller D., Weber J. and Malik J., "Towards realtime visual based tracking in cluttered traffic scenes", Proc. Of the Intelligent Vehicles Symposium, october 1994.

[**Korn, 98**] Korn A.F., "Towards a symbolic representation of intensity changes in images", IEEE TPAMI, Vol. 10, 1998, pp 610-625.

[**Liu et al., 98**] Liu, H., Hong, T.-H., Herman, M. et Chellappa, R. (1998). Accuracy vs. efficiency trade-offs in optical flow algorithms. Computer Vision and Image Understanding, 7(3):271{286}.

[**lowe, 91**] Lowe D.G., "Fitting Parameterized Three-Dimensional Models to Images", IEEE TPAMI, Vol. 13, No. 5, May 1991, pp 441-450.

[**lowe, 92**] Lowe D.G., "Robust Model-Based Motion Tracking Through the Integration of Search and Estimation", International Journal of Computer Vision (8), No. 2, August 1992, pp 113-122

[**Ma et al., 01**] Ma, Y. F. et Zhang, H. J. (2001). Detecting motion object by spatio-temporal entropy. In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2001), pages 265{268, Tokyo, Japon.}

[**MacIntyre et al., 04**] B. MacIntyre, M. Gandy, S. Dow, et J.D. Bolter (2004). Darl : a toolkit for rapid design exploration of augmented reality experiences. Dans Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST 04), pages 197–206. ACM Press.

? X [**Madjid, 07**] Madjid MAIDISuivi Hybride en présence d'Occultations pour la Réalité Augmentée 2007. ?? these a quoi??

[**Mars., 98**] Mars - Août 1998 Groupe de Recherche sur l'Analyse et la Commande des SYstèmes (GRACSY) Université de Picardie Jules Verne – AMIENS.

[**Matsuka et al., 02**] H. Matsuka, A. Onosawa, and E. Osaya. Environment mapping for objects in the real world. In International ARToolkit Workshop (ART) , October 2002.

[**McKenna et al., 00**] McKenna, S. J., Jabri, S., Duric, Z., Wechsler, H. et Rosenfeld, A. (2000). Tracking groups of people. Computer Vision and Image Understanding, 80(1):42{56}.

[**Michel., 04**] M. le Professeur Michel DE GLAS ,Président Directeur de Recherche Mme Annie MONOT Optimisation du traitement des indices de profondeur pour application à la vision artificielle Soutenu en Juillet 2004

[**Milgram et al., 94**] P. Milgram, H. Takemura, A. Utsumi, et F. Kishino (1994). Augmented reality : A class of displays on the reality-virtuality continuum. SPIE : Telemanipulator and Telepresence Technologies, 2351 :282–292.

[**Nicolas ,07**]Suivi d'objets en mouvement dans une séquence vidéo, Université PARIS DESCARTES Centre universitaire des Saints-Pères 2007.

[**Olano et al., 95**]M. Olano, J. Cohen, M. Mine, and G. Bishop. Combatting rendering latency. In SI3D'95 : Proceedings of the 1995 symposium on Interactive 3D graphics, pages 19–24, Monterey, California, USA, 1995.

[**Peter**] Peter Sturm INRIA Rhône-Alpes, Equipe-projet PERCEPTION
Peter.Sturm@inrialpes.fr

[**Piekarski et al., 03**] W. Piekarski et B.H. Thomas (2003). An object-oriented software architecture for 3d mixed reality applications. Dans The Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03), Tokyo, Japan.

[**Raskar et al, 01**]R. Raskar, G. Welch, K. L. Low, and D. Bandyopadhyay. Shader lamps : Animating real objects with image-based illumination. In Proceedings of the 12th Eurographics Workshop on Rendering Techniques, pages 89–102, London, UK, 2001.

[**Reitmayr et al., 01**]G. Reitmayr et D. Schmalstieg (2001). An open software architecture for virtual reality interaction. Dans Proceedings of the ACM symposium on Virtual reality software and technology, pages 47–54. ACM Press.

[**Schmalstieg et al., 02**]D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavari, L. M. Encarnação, M. Gervautz, et W. Purgathofer (2002). The studierstube augmented reality project. Presence : Teleoperators and Virtual Environments, 11(1) :33–54.

[**Simon et al., 00**]Simon Gilles, Andrew Fitzgibbon, Andrew Zisserman. “Markerless Tracking using Planar Structures in the Scene”. Proceedings. IEEE and ACM International Symposium on Augmented Reality ISAR 2000, 2000. pp 120-128.

[**Stenger et al., 01**]Stenger, B., Ramesh, V., Paragios, N., Coetzee, F. et Buhmann, J. M. (2001). Topology free hidden Markov models: Application to background modeling. In Proc. 8th IEEE Int. Conf. on Computer Vision (ICCV 2001), volume I, pages 294{301, Vancouver, Canada.

[Sugano et al, 03]N. Sugano, H. Kato, and K. Tachibana. The effects of shadow representation of virtual objects in augmented reality. In ISMAR'03 : Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality , page 76, Tokyo, Japan, 2003.

[Tian et al., 05] Tian, Y. L. et Hampapur, A. (2005). Robust salient motion detection with complex ackground for real-time video surveillance. In Proc. IEEE Work-shop on Motion and Video Computing

[Toyama et al., 99]Toyama, K., Krumm, J., Brummit, B. et Meyers, B. (1999). Wallflower:Principles and practice of background maintenance. In Proc. 7th IEEE Int.Conf. on Computer Vision (ICCV'99), volume I, pages 255{261, Kerkyra(Corfou), Grece.

[Wloka., 93]M.Wloka. Dissertation proposal : Time-critical graphics . Tech. Rep. TR-CS-93-50, Computer Science Department, Brown University, USA, 1993.

[Wloka., 95] M. Wloka. Lag in multiprocessor virtual reality. In Presence : Teleoperators and Virtual Environments, 4(1) :50–63, 1995.

[Worral et al., 94]Worral A.D., Sullivan G.D. and Baker K.D., "Pose refinement of active models using forces in 3D", Third European Conference on Computer Vision, Stockholm, Sweden, 1994.

[Wren et al., 97]Wren, C., Azarbayejani, A., Darrell, T. et Pentland, A. (1997).Pfinder: Real-time tracking of the human body. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7):780{785.

[Yang et al.,04]Yang, T., Li, S. Z., Pan, Q. et Li, J. (2004). Real-time and accurate segmentation of moving objects in dynamic scene. In Proc. ACM 2nd Int. Workshop on Video Surveillance & Sensor Networks (VSSN 2004), pages 136{143, New York (NY), USA.

[ARToolkit,url] <http://www.hitl.washington.edu/artoolkit/> derniere condultatoin 18-05-2012

[Amire, url.] Amire web site. <http://www.amire.net>. derniere condultatoin 28-03-2012

[Dart, url] . Dart web site. <http://www.gvu.gatech.edu/dart/> . Dernière consultation 28-03-2012