# Suivi de paramètres de modèle géométrique à partir de séquences vidéo multi-vues

Yannick Perret

**UFR Informatique** **École doctorale EDIIS**

# Suivi de paramètres de modèle géométriques à partir de séquences vidéo multi-vues

## THÈSE

présentée et soutenue publiquement le 17 Décembre 2001

pour l'obtention du

**Doctorat de l'université Claude Bernard – Lyon 1**

**(spécialité informatique)**

par

Yannick Perret

**Composition du jury**

| | | |
|---|---|---|
| *Rapporteurs :* | M Kadi Bouatouch, | Professeur, IRISA, Université Rennes 1 |
| | M Edoardo Ardizzone, | Professeur, Université de Palerme, Italie |
| *Examinateurs :* | M Radu Horaud, | Directeur de recherche, INRIA Rhône-Alpes |
| | M Liming Chen, | Professeur, École Centrale de Lyon |
| | M Denis Vandorpe, | Professeur, LIGIM, Université Lyon 1 |
| | Mme Saïda Bouakaz, | Maître de conférence, LIGIM, Université Lyon 1 |

**Laboratoire d'Informatique Graphique, Image et modélisation**

# Remerciements

*à **Linux**, à **gcc**, à **LaTeX**, à la **cafetière** du LIGIM...*

# Table des matières

# Première partie

# Introduction générale

L'image prend une place de plus en plus importante dans notre société. Que ce soit à travers la télévision, le cinéma ou l'informatique, l'extension des moyens de stockage, de traitement et de diffusion fait que l'on trouve l'image à tous les niveaux, au travail comme dans les loisirs. Tout d'abord immobile, puis film dont on est simple spectateur, l'image devient interactive grâce à l'avènement du multimédia. Celui qui regarde n'est plus forcément un spectateur, mais peut interagir avec l'image.

Les domaines de la réalité augmentée et de la réalité virtuelle apportent une nouvelle façon d'interagir avec l'image : la participation active de celui qui regarde. Pour parvenir à cela, il est nécessaire de repérer et de suivre de façon précise les mouvements de la personne interagissant avec l'image.

Dans des films récents tels que *Vidocq* ou *Le seigneur des anneaux*, les acteurs, c'est-à-dire les images originales, sont transformés et intégrés dans des lieux artificiels à travers les effets spéciaux. En chirurgie se développent deux aspects complémentaires : d'un coté la simulation d'actes aussi proches de la réalité que possible, et d'un autre coté l'assistance visuelle pour des opérations réelles, en insérant des informations inaccessibles à l'oeil humain dans les images. D'autres applications moins spectaculaires mais tout aussi importantes s'inscrivent dans l'appréhension de l'image et du mouvement, telles que le suivi de cibles, la vidéo-surveillance ou encore la vidéo-conférence.

C'est pourquoi on assiste à l'heure actuelle à un regain d'intérêt pour l'analyse du mouvement dans les images. Cela va de la simple détection de la présence de mouvements au suivi précis d'un objet qui se déplace dans un environnement 3D. Les mouvements possibles de l'objet sont appelés paramètres du mouvement, et ce sont leurs valeurs que l'on va chercher à obtenir au cours du temps.

Partant du constat que dans les applications de réalité augmentée et de réalité virtuelle l'objet à suivre est connu d'avance, cette thèse propose une approche du suivi 3D exploitant la notion de modèle d'objet. Le modèle est ici définie comme étant un ensemble d'informations liés à l'objet : sa géométrie, son apparence, sa cinétique.

La méthode que nous proposons utilise des flux vidéo issus de caméras filmant l'objet à suivre comme données d'entrée. Ces images sont combinées aux informations apportées par le modèle de l'objet afin d'effectuer le suivi (Fig. 1). Le point essentiel de cette méthode réside dans la transformation de vecteurs de paramètres en images, grâce à la synthèse d'images. Ce sont ces vues artificielles qui seront confrontées aux images réelles afin de déterminer le vecteur de paramètres concordant le mieux avec la réalité.

Nous avons articulé ce manuscrit autour de quatre grandes parties.

Dans la première partie nous présentons le problème du suivi de mouvements, et nous situons notre approche par rapport aux approches existantes.

La seconde partie est consacrée à la description de notre approche. Après avoir abordé les problèmes liés à la modélisation de l'objet et de la scène incluant celui-ci, nous pré-

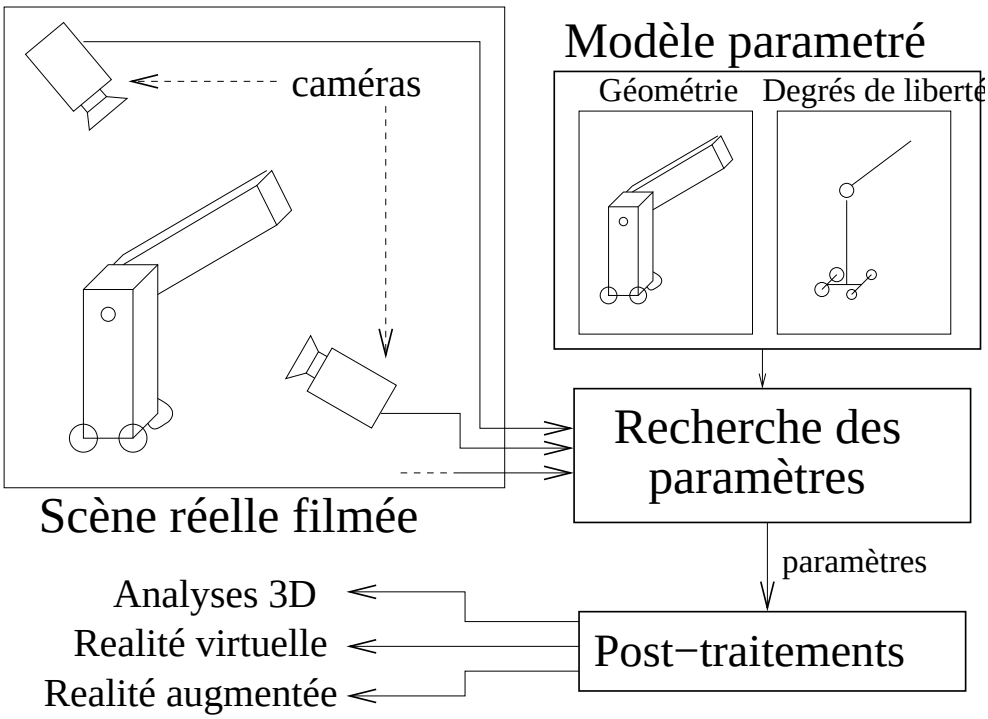



FIG. 1 – *Situation générale d'un suivi d'objet : l'objet est filmé par plusieurs caméras, et ces données couplées au modèle géométrique sont utilisées pour extraire et suivre les mouvements de l'objet*

sentons les différentes composantes mises en oeuvre dans notre approche qui sont :

- la définition d'une mesure d'écart entre des images réelles et des images de synthèse ;
- et la recherche de l'optimum de cette mesure par rapport aux paramètres ;
- et finalement la prise en compte du temps pour réaliser le suivi proprement dit.

Dans cette seconde partie de nombreux exemples simples viendront illustrer les différents points abordés.

Dans la troisième partie nous présentons les méthodes que nous avons mises en œuvre pour permettre l'affinage automatique des modèles d'objets. Deux points importants composent cette partie :

- l'affinage de la géométrie du modèle par rapport aux vues réelles disponibles ;
- l'extraction automatique des textures des objets afin d'augmenter la ressemblance du modèle.

Enfin la quatrième partie présente de nombreux résultats de suivis obtenus grâce à notre méthode. Ces résultats sont présentés selon deux catégories :

- les suivis réalisés à partir de séquences vidéo de synthèse. Le contrôle du contenu de ces séquences nous permet ainsi de valider théoriquement notre approche, et d'effectuer des mesures d'erreurs sur les résultats.
- les suivis réalisés sur des séquences vidéo réelles, afin de montrer la fiabilité de notre

approche dans des situations réelles.

# Deuxième partie

# Suivi d'objets : approches existantes

# 1 Problématique

Comme cela a été indiqué, nous nous intéressons au suivi de mouvements dans des flux d'images. Partant du fait que dans les applications de réalité augmentée et de réalité virtuelle l'objet suivi est connu d'avance, nous supposons disposer d'un modèle de cet objet. Ce modèle va décrire la géométrie de l'objet, ainsi que son apparence et sa cinétique. Nous cherchons ainsi à obtenir les valeurs des paramètres indépendants qui contrôlent la cinétique de l'objet au cours du temps.

La réalité augmentée ainsi que la réalité virtuelle ont toutes deux besoin des valeurs des paramètres du mouvements d'objets réels. Cela peut être pour l'intégration précise d'entités réelles dans des environnements de synthèse, pour l'interaction avec ces environnements ou pour ajouter de l'information à des scènes réelles[BCC00]. D'autre applications sont également concernées par cette problématique, et requièrent une connaissance plus ou moins complète des paramètres liés aux mouvements d'objets. Cela concerne entre autre l'étude des mouvements chez les sportifs, ou bien encore l'analyse de *crash-tests* dans le cadre de la sécurité automobile.

Il existe d'autres familles d'applications où la maîtrise de paramètres est moins exigeante. Seule une partie des paramètres liés au mouvements sont recherchés, ou bien une faible précision est suffisante pour les besoins correspondants. On peut citer entre autre :

- l'analyse de scènes urbaines pour la télé-surveillance ou la gestion de trafic routier ;
- le repérage / suivi de cibles ;
- le suivi de mouvements d'entités complexes (machines ou humains) à des fins d'analyse, pour la prédiction de trajectoires.

Comme nous l'avons vu, l'hypothèse de travail de cette thèse est que l'on possède un modèle géométrique générique de l'objet à suivre. Le thème de l'étude du mouvement est très actuel. La littérature propose de nombreux travaux qui relèvent aussi bien du domaine de la vision que du domaine de la synthèse d'image et/ou de la réalité virtuelle. De nombreux travaux existent, mais peu de gens se sont intéressés à obtenir tous les paramètres caractérisant un mouvement.

# 2 Senseurs actifs et passifs

C'est ainsi que certaines méthodes passent par l'utilisation de senseurs actifs ou passifs, posés sur le sujet à suivre. Un senseur est un appareillage dont le rôle est de fournir une information géométrique. Les senseurs actifs peuvent être des capteurs de position ou de rotation, qui transmettent (souvent par câble) leurs données. Un exemple populaire de senseurs actifs sont les *data-gloves*, sorte de gants permettant d'obtenir tous les paramètres de mouvements d'une main, et fréquemment utilisés dans les systèmes d'immersion dans des environnements virtuels [MATea99]. Les senseurs passifs ont des caractéristiques (le plus fréquemment magnétiques) qui permettent de les localiser à distance dans l'espace. Le site [Sen] présente toute l'actualité de l'industrie des senseurs, avec leurs caractéristiques et leurs domaines d'utilisation. Ces méthodes utilisent ces senseurs pour suivre un ensemble de points de l'objet, afin de caractériser les mouvements au cours du temps (voir figure 2 page suivante).

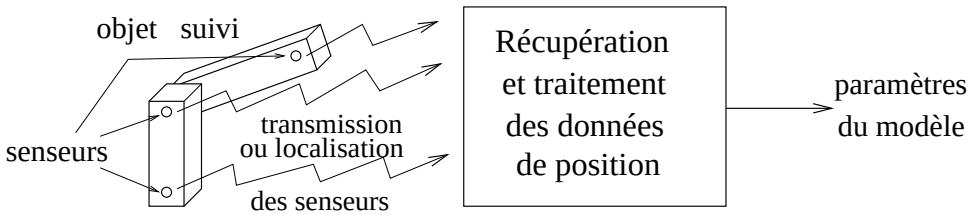



FIG. 2 – *Caractérisation de mouvements par l'utilisation de senseurs actifs ou passifs disposés sur l'objet à suivre. Les informations fournies par les senseurs permettent de retrouver l'état spatial de l'objet.*

Dans [HDN96] et [NT97], les auteurs traitent principalement de l'acquisition des mouvements humains à partir de senseurs, le premier a comme perspective de réduire au maximum le nombre de senseurs, le deuxième plus spécifiquement l'immersion en milieux virtuels. On retrouve les même préoccupations dans [FZMC96], avec plus spécifiquement des problématiques liées au temps réel.

Notons toutefois que l'utilisation de senseurs ne va pas sans contraintes, telles que les coûts, les difficultés de mise en place sur l'objet à suivre ainsi que les limitations liées à la transmission des information.

# 3 Approches utilisant les séquences vidéo

Parmi les approches utilisant des sources vidéo pour traiter le suivi d'objet il est possible de dégager deux grandes catégories. Celles-ci diffèrent par les hypothèses sous-jacentes mises en œuvre :

- les approches supposant la présence de marqueurs sur le sujet à suivre ;
- les approches ne faisant pas cette supposition.

# 4 Approches avec marqueurs

Les marqueurs sont des entités ayant des propriétés visuelles spécifiques, et étant généralement de petite taille. Il existe des marqueurs actifs et passifs. Les marqueurs actifs vont émettre une lumière caractéristique, soit directement soit par réflexion d'une source éclairant la scène dans laquelle se trouve l'objet. Les marqueurs passifs se contentent d'être facilement identifiables et localisables dans les images. La nature et la forme de ces marqueurs peuvent être très diverses : motifs contrastés, disques ou sphères réfléchissant certains types de lumière (telle que la *lumière noire*, dans le proche ultraviolet), gamme de couleur spécifique...

Le point commun que partagent tous ces marqueurs est d'être facile à repérer dans les images vidéo. À partir de leurs coordonnées 2D dans les images il est ensuite possible d'extraire les coordonnées spatiales correspondante à partir des informations de calibrage des caméras, par des techniques de projection inverse (voir figure 3).

Ces points sont associés à des parties connues de l'objet suivi. Ce sont des positions clé pour le mouvement, telles que les articulations pour le corps humain. Il est alors possible

de déduire les caractéristiques spatiales de cet objet, en transposant les positions 3D de ces points sur le modèle géométrique de l'objet.

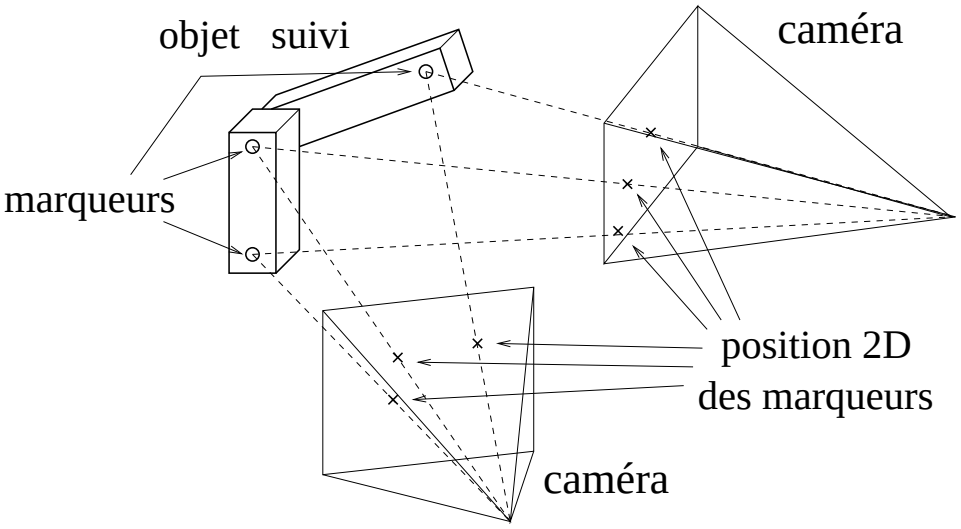



FIG. 3 – *Caractérisation de mouvements par l'utilisation de marqueurs disposés sur l'objet à suivre, ceux-ci étant repérés dans l'espace de chaque caméra. La position 3D est ensuite retrouvée par projection inverse, en utilisant les informations de calibrage des caméras.*

Ce type d'approche est utilisé dans de nombreuses applications. Les systèmes de création d'animations réalistes l'utilisent pour simuler le mouvement de personnages dans les jeux vidéo. Parmi les applications on peut citer les jeux de combat en 3D tels que *Tekken*[Nam].

Ces mêmes techniques sont utilisées pour les effets spéciaux et les trucages de films. Cela concerne toutes les situations où un personnage non humain doit être intégré dans une scène filmée réelle. Dans la plupart des cas un acteur humain est utilisé. Des senseurs sont posés sur toutes les parties importantes de celui-ci, et ses mouvements servent de référence pour animer un acteur virtuel qui sera ensuite intégré dans la scène où il doit apparaître.

Il est à noter que dans les deux cas précédemment cités les traitements sont parfois effectués par des systèmes de senseurs ou par une combinaison des deux techniques – système par marqueurs pour les mouvements généraux et senseurs pour les mouvements précis et/ou difficilement visible comme les mains.

Dans [Wil90], les auteurs utilisent des points irridescents sur un visage pour suivre les attitudes faciales. De la même façon, lors de l'étude de l'impact des accidents de véhicules sur l'organisme humain (*crash-tests*), l'idée est de suivre précisement les mouvements de mannequins simulant des êtres humains. La plupart des senseurs résistent mal à ces chocs, et des techniques de *suivi distant*, via l'utilisation de marqueurs posés sur le mannequin sont utilisées.

Certains auteurs utilisent des marqueurs plus complexes que des points. Dans [KWT88] des contours actifs (*active snakes*) sont utilisés pour suivre les déformations de courbes tracées sur des visages, afin de suivre les mouvements et déformations de celui-ci.

Cette approche est relativement robuste et rapide. Elle est répandue car elle nécessite moins de ressources techniques – les senseurs actifs et passifs coûtent chers et sont plus contraignant à utiliser, et les données d'entrée (images vidéo) peuvent également être dépouillées à la main, même si cela est plus long. Elle présente cependant quelques limitations. En premier lieu elle nécessite une phase de pose des marqueurs, qui peut être relativement longue et s'adresse obligatoirement à des objets accessibles et instrumentables. En particulier cela n'est pas utilisable pour des applications "extérieures", vidéo-surveillance par exemple, où les sujets à suivre échappent au contrôle des opérateurs. La pose de ces marqueurs demande de plus une bonne précision, afin que leur position réelle corresponde à la position théorique, associée au modèle de l'objet.

Une autre limitation est liée aux problèmes d'occultations. Si un marqueur est caché par un objet extérieur ou par l'objet lui-même (auto-occultation) il peut ne pas être possible d'obtenir toutes les informations sur la partie de l'objet concernée. Des techniques de prédiction de trajectoire peuvent limiter ces problèmes dans le cas d'occultations de courtes durées, mais cela ne résout pas le problème lorsque cette situation se prolonge.

# 5 Approches sans marqueurs

Ce type d'approche n'utilise pas d'instrumentation directe sur les sujets à suivre. Il n'y a aucune pré-connaissance nécessaires sur des aspects visuels précis que l'on sait trouver à des positions précises sur l'objet.

Il existe de nombreux travaux pour ce type d'approche, car dans de nombreuses situations il est impossible d'avoir accès physiquement aux objets suivis, et donc de poser des marqueurs sur ceux-ci. Dans tous ces cas une approche n'utilisant que les images vidéo est nécessaire.

Il est possible de classer ces approches selon plusieurs critères, dépendant le point de vue que l'on adopte. Nous avons choisi de les caractériser par la manière dont la notion de modèle d'objet – sous-jacent ou explicite – est utilisée, et par rapport aux hypothèses qui sont faites sur ce modèle.

Selon cette classification, nous allons donc distinguer en premier lieu les approches faisant le moins appel à la notion de modèle d'objet, soit que celui-ci soit implicite, soit que les informations qui y sont liées soient réduites au minimum. Nous présenterons ensuite les approches classées selon un degré d'utilisation croissant du modèle de l'objet, avec en bout de chaîne les approches exploitant un modèle cherchant à reproduire fidèlement l'objet à suivre.

À partir de ce critère, nous allons distinguer les approches comme suit :

1. les approches pour lesquelles le modèle de l'objet est absent.
   Ces approches ne cherchent pas en général à suivre un objet précis mais à déterminer les mouvements perçus dans les images. Dans ces cas là la notion de modèle n'est pas directement exploitée pour effectuer le suivi, mais est éventuellement générée a posteriori à partir des résultats trouvés ;
2. les approches pour lesquelles le modèle sert de support structurel.
   Ce modèle est dans ces cas là utilisé pour représenter une ou plusieurs caractéris-

tiques mesurables de l'objet telles que la définition d'invariants ou la répartition de certaines propriétés géométriques connues ;

3. les approches utilisant le modèle de l'objet pour sa géométrie.
   La géométrie peut être utilisée de deux manières différentes : soit en 2D, ce sont alors des informations géométrique sur les projections de l'objet suivi qui sont exploités ; soit en 3D, le modèle est situé dans un espace tridimensionnel visant à simuler la réalité ;
4. les approches où l'aspect visuel – en plus des autres – est exploité.
   Dans ces cas là le but est généralement d'exploiter une visualisation de ce modèle via la synthèse d'images et de confronter ces vues de synthèse à la *réalité* telle qu'elle apparaît dans les flux vidéo.

## 5.1 Modèle absent

Dans la première catégorie d'approches décrite on trouve toutes les méthodes cherchant à déterminer des points ou zones en mouvement dans des images. Dans ce contexte il n'y a pas de modèle explicitement présent. La seule modélisation de l'objet suivi que l'on puisse mettre en avant est le mouvement perçu lui-même. Cela concerne un grand nombre d'applications de suivi 2D, dans lesquelles on cherche à identifier puis à suivre des zones en mouvement, avec généralement des contraintes de cohérence spatiales et/ou temporelles.

Certaines approches utilisent les changements visuels dans les textures pour déterminer les mouvements apparents[PN92, MNR92]. Dans [PD99] les auteurs utilisent des contours actifs et des ensembles de niveaux (*level set*) pour détecter et suivre des objets en mouvement dans une séquence vidéo.

L'avantage de ce type d'approche est qu'il n'y a aucun besoin de modèle a priori de l'objet, et donc que ces méthodes s'appliquent à de nombreuses situations. L'adaptation du système aux mouvements perçus permet de plus de déduire un modèle a posteriori de l'objet suivi. Cela rend ces méthodes particulièrement adaptées aux problèmes de type "analyse globale" d'un mouvement, par exemple pour déterminer des trajectoires de véhicules ou de piétons. Toutefois ces approches ne s'appliquent généralement qu'au suivi 2D, et l'absence de structure de l'objet ne permet pas d'obtenir d'information sur d'éventuelles articulations. De plus il n'est généralement possible d'obtenir des informations sur l'objet suivi – identification – qu'à travers sa projection dans l'image, avec toutes les limitations que cela implique.

## 5.2 Structure de l'objet

La deuxième catégorie d'approches utilise la notion de modèle d'objet comme support d'une information de structure sur celui-ci. Cette information n'est pas toujours directement liée à la géométrie de l'objet. Dans [SMB88] ce sont des invariants de l'image ainsi que leur organisation spatiale qui sont utilisés comme *signature* de l'objet, c'est-à-dire comme modèle. Dans d'autres approches [BW96] le modèle représente une distribution spatiale de points ayant des propriétés décelable par l'analyse d'image – ici des points de

l'objet présentant une forte variance – et est utilisé pour obtenir la position et l'orientation de cet objet, en calculant les transformations permettant de passer de la projection 2D à l'espace 3D. Dans [ASHP93] un algorithme de recherche de points d'intérêt dans les visages est utilisé, leur position 3D étant retrouvée par l'utilisation d'un filtre de Kalman étendu.

L'hypothèse qui sous-tend ces approches est la constance de l'information recherchée – points, invariants géométriques... – dans les images possibles de l'objet suivi. C'est donc également la limite qui fixe les possibilités offertes, en particulier lorsque des occultations ou de forts changements de points de vues modifient – ou éliminent – les éléments d'information.

## 5.3 Géométrie de l'objet

La troisième catégorie présentée regroupe les approches utilisant la géométrie de l'objet suivi, définie à travers un modèle géométrique.

Cette information de géométrie peut être utilisée de plusieurs manières. Une approche que l'on retrouve dans plusieurs travaux consiste à utiliser les contours de l'objet définies par la géométrie du modèle[HYAT01, DF99]. Il s'agit de comparer les contours détectés dans les images vidéo avec les contours de la projection du modèle géométrique – dans les mêmes conditions que les vues réelles. L'écart entre ces contours est utilisé pour ajuster les degrés de liberté du modèle afin qu'il se rapproche des contours, par exemple par résolution numérique des équations de projection[HYAT01] ou bien encore par des systèmes de forces appliquées au modèle [DF99]. Un certain nombre d'auteurs font appel aux flux optiques. Dans [DM97] le calcul des flux optiques est contraint par la structure d'un modèle de visage. L'interaction entre ces contraintes et les mouvements planaires détectés par les flux optiques permettent le suivi des mouvements de ce visage. Similairement dans [MNP00] les auteurs cherchent à trouver les paramètres d'un modèle de visage de façon à ce que le flux optique généré par le déplacement du modèle corresponde au flux optique constaté dans les images vidéo.

Ces types d'approche permettent d'effectuer des suivis d'objets avec une bonne précision. Il faut toutefois faire attention à la phase d'extraction des contours et/ou au calcul des flux optiques, la précision de ceux-ci déterminant la précision finale. De plus il faut gérer les problèmes d'occultations et d'auto-occultations, car ceux-ci peuvent perturber la recherche des contours ainsi que la précision des flux optiques.

## 5.4 Aspects visuels

Enfin la dernière catégorie d'approches que nous avons dégagée se fonde sur l'exploitation de l'aspect visuel de l'objet étudié – en plus des aspects structurels et géométriques – pour effectuer le suivi. Notons tout de même qu'il existe des approches où l'aspect visuel d'un objet est exploité, sans pour autant que l'on ait explicitement sa géométrie et sa structure. Par exemple dans [DDJ01] les auteurs utilise une collection de vues d'un objet – vues prises sur une sphère englobant l'objet – comme références de l'aspect de cet objet, ceux-ci étant rigides. Mais dans la plupart des cas l'ensemble de ces données sont utilisées.

C'est directement ce type d'approche qui est utilisé dans les problèmes de recalage 2D, via des comparaisons au niveau image (*image registration*), pour caractériser des rotations, translations et éventuellement des changements d'échelle par rapport à une image de référence. On pourra trouver dans [Bro92] un état de l'art des différentes méthodes existantes.

Toujours en 2D, certaines approches utilisent une structure de l'objet à suivre, rigide ou articulée, et extraient – ou utilisent directement si cette information est disponible – l'apparence correspondante dans l'image. Le suivi se fait alors en cherchant les variations du modèle qui correspondent au mieux à la nouvelle apparence de l'objet suivi dans les nouvelles images. Dans [IS98] le modèle est un maillage d'une portion de l'image dont la texture est extraite, ce maillage étant déformable – le but étant justement de suivre les déformations de l'objet dans le temps.

Dans un cadre tridimensionnel on trouve également de nombreuses approches. Plusieurs d'entre elles utilisent comme outil les flux optiques. Dans d'autres travaux le flux optique entre deux vues successives est utilisé pour déplacer le modèle de l'objet, par projection des vecteurs 2D issues de chaque vue dans l'espace du modèle. La synthèse d'image est utilisée pour valider et corriger ce déplacement. On trouve également des méthodes combinant l'utilisation de marqueurs et de synthèse d'image. Le suivi de danseurs est réalisé en recherchant la présence de marqueurs sur les sujets (sous la forme de motifs sur les vêtements) puis en validant les résultats grâce à la synthèse d'image [GD95]. Dans [KM00] les auteurs utilisent la synthèse d'image afin de déterminer les disparités entre ces images générées et une image réelle pour corriger la position du modèle. La même idée est exploité dans [OH99] pour l'analyse des gestes, à partir d'un modèle articulé de main. Dans [GG00] un modèle texturé est utilisé pour le suivi d'objets rigides dans des séquences mono-vues, en comparant des rendus de synthèse de l'objet avec les images vidéo.

# 6 Approche proposée

C'est dans cette dernière catégorie que se placent nos travaux. L'exploitation d'un modèle de l'objet suivi comprenant à la fois la structure, la géométrie et l'apparence permet une simulation visuelle précise de l'objet. La recherche des paramètres du mouvement se pose alors en termes de recherche des simulations les plus fidèles par rapport aux images vidéo.

Nous pensons en effet qu'avec les technologies actuelles en matière de rendu de synthèse, il est possible d'obtenir rapidement des images de qualité, d'autant plus que l'avènement des cartes 3D et leur support par des outils tels que Mesa (OpenGL) permet d'atteindre des vitesses impressionnantes. Ceci réduit l'un des principal défaut de ce type d'approche mis en avant dans le passé : le coût prohibitif des phases de rendu. De plus l'avènement d'outils de modélisation et même d'obtention de modèles automatisé ([Ph3, La3]) rend la création de modèles d'objets beaucoup plus simple, ces modèles étant de toute façon présents dans de nombreuses applications : simulation, animation réaliste (réalité virtuelle, films...), CAO/CFAO...

Nous avons donc tous les outils en main pour pouvoir exploiter pleinement ce type

d'approche, qui offre par ailleurs de nombreux avantages tels que la robustesse aux occultations et une bonne précision des résultats, comme nous le montrerons dans cette thèse.

# Troisième partie

# Recherche de paramètres de modèle géométrique

# 1

# Notre approche du problème

La réalité augmentée et la réalité virtuelle utilisent le suivi de mouvements d'objets. Il s'agit de situations où l'objet à suivre est connu d'avance. Nous partons donc de l'hypothèse que nous disposons d'un modèle paramétré de cet objet. Ce modèle définit la géométrie de l'objet ainsi que son apparence. Il définit également la cinétique associée à l'objet, qui permet de spécifier les paramètres contrôlant les mouvements. Ce sont les valeurs de ces paramètres que nous cherchons à suivre au cours du temps.

Nous avons rappelé dans la partie précédente les divers techniques utilisées pour le suivi d'objets. En particulier un certain nombre d'entre elles associent analyse et synthèse d'image, sous diverses formes, pour réaliser ce suivi.

Notre approche est fondée sur l'idée suivante : rechercher les paramètres qui correspondent à l'objet réel grâce à la synthèse d'image. L'analyse simultanée des images réelles et de vues de synthèse de l'objet permet de déterminer les paramètres qui définissent l'objet réel (Fig. 1.1).

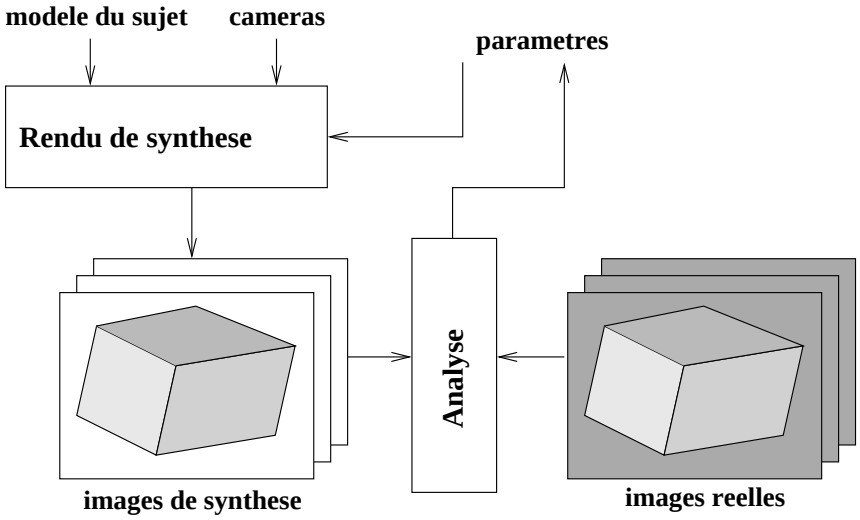



FIG. 1.1 – *notre approche : recherche des paramètres de l'objet en analysant des vues de synthèse et les vues réelles de l'objet.*

La méthode va comparer les images générées aux images de référence issues de sé-

quences vidéo, afin d'en améliorer la pertinence.

## 1.1 Hypothèses de travail

Il est nécessaire de définir précisément les données à partir desquelles nous allons travailler.
Nous avons tout d'abord des séquences vidéo de l'objet étudié. Ces séquences sont issues de plusieurs caméras filmant le sujet au cours de son évolution, pris à partir de plusieurs positions quelconques.
Nous disposons également d'un modèle géométrique paramétré de l'objet suivi. Ce modèle inclut les aspects structurels de l'objet et les paramètres qui y sont liés, ainsi que les aspects géométriques. Les aspects géométriques sont utilisés pour la synthèse d'image. Les aspects structurels regroupent quant à eux les informations liées à la cinétique de l'objet, que nous appelons paramètres. On peut citer le positionnement et l'orientation globale de l'objet, ainsi que l'état des différentes articulations. Nous considérerons par la suite que les paramètres sont tous à effet continu sur l'objet. Cela regroupe la plupart des paramètres imaginables, mais exclut des paramètres de type "tout-ou-rien", comme par exemple une lampe qui s'allume ou s'éteint (un feu de circulation).
Nous disposons de plus d'un moteur de rendu 3D, capable de générer les images de synthèse nécessaires pour notre approche. Ce moteur de rendu est considéré par notre approche comme étant une "boîte noire". Il suffit qu'il soit en mesure de fournir des vues de synthèse de l'objet à partir de son modèle géométrique paramétré et d'un ensemble de valeurs pour les paramètres. Il faut bien entendu au moteur de rendu les informations de calibration liées aux caméras, afin que celui-ci soit capable de générer des vues simulant les vues réelles de l'objet.

## 1.2 Recherche des paramètres à un instant $t$ donné

Dans un premier temps nous faisons abstraction de l'aspect temporel des séquences et nous nous plaçons à un instant $t$ donné. Nous supposons que nous disposons d'un vecteur de paramètres initial, "proche" de la solution pour cet instant $t$ (le choix des valeurs initiales sera discuté dans la section 6). Nous avons donc comme données : un vecteur initial de paramètres proche de la solution théorique et un ensemble de vues de l'objet à cet instant $t$ (et bien évidemment le modèle géométrique paramétré de l'objet).

Nous cherchons à partir de ces vues à obtenir les valeurs des paramètres du modèle telles que la projection de ce modèles dans l'espace image corresponde au mieux aux différentes vues issues des caméras.

La détermination des paramètres de mouvements d'un objet est ainsi la recherche des valeurs des paramètres d'un modèle géométrique de l'objet, de telle façon que des vues de synthèse $\Im_i(p)$ de ce modèle soient les plus proches des images de référence $I_i$, avec $i$ le

numéro de la vue puisque nous travaillons avec des séquences multi-vues (cf Fig. 1.2).

En introduisant $\mathcal{E}$ une fonction d'écart entre paires d'images réelles et d'images de synthèse, nous cherchons $p$ un vecteur de paramètres tel que

$$\mathcal{E}(\{\mathcal{I}_1(p), I_1\}, \{\mathcal{I}_2(p), I_2\}, ..., \{\mathcal{I}_c(p), I_c\}) \tag{1.1}$$

soit optimal Cet optimum est recherché dans l'espace des paramètres qui gouvernent le modèle et donc qui optimisent la fonction. Une approche fondée sur la méthode du simplexe sera présenté à la section 4.
Nous n'abordons dans cette partie que le problème de la recherche des paramètres d'un modèle à un instant $t$ donné. L'extension temporelle des traitements – c'est-à-dire la recherche des paramètres au cours du temps – sera abordée dans la section 5.

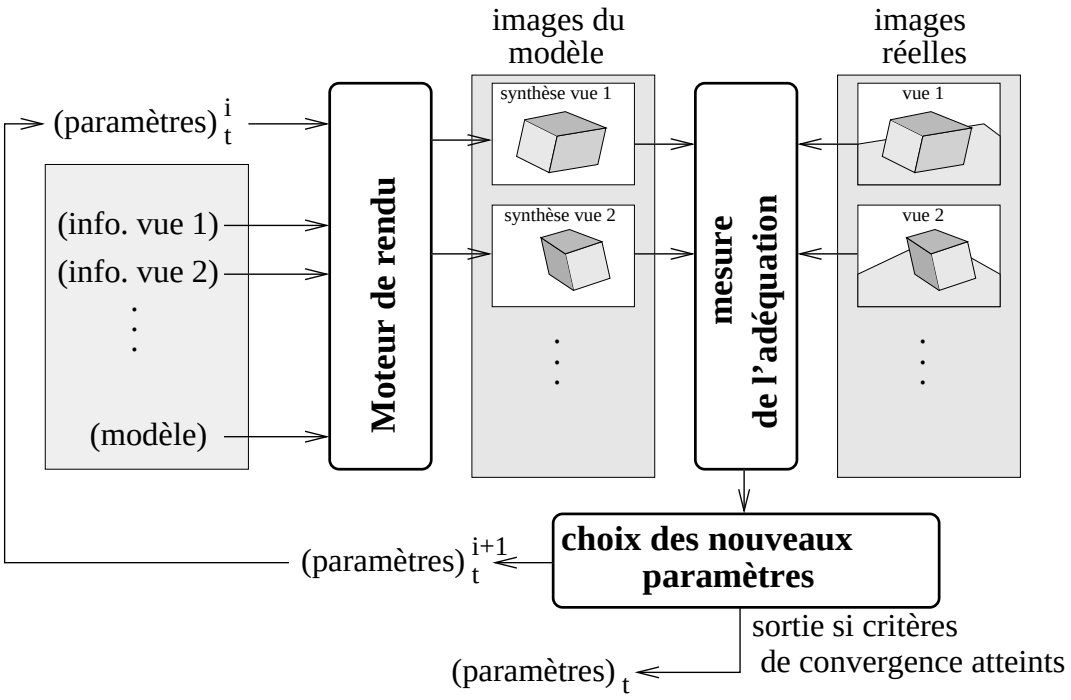



FIG. 1.2 – *Recherche des paramètres par rapport aux images de référence. À partir de paramètres initiaux, les vues de synthèse correspondant à chaque caméra sont générées, puis comparées aux vues réelles afin d'ajuster les valeurs des paramètres. Ce processus est itéré jusqu'à atteinte d'un critère de convergence.*

Nous devons pour cela développer plusieurs points afin d'arriver à ce but :
- nous commençons par préciser le sens de modèle d'objet tel que nous l'utilisons, et la façon de définir celui-ci ainsi que la scène dans laquelle il se trouve ;
- dans la section 3 page 27 nous abordons la notion de mesure d'écart entre images réelles et images de synthèse. Nous avons évité le terme de "distance" entre images pour éviter toute confusion avec le concept mathématique correspondant qui doit vérifier des axiomes bien précis, ce qui n'est pas le cas pour la mesure que nous introduisons ;

- comme nous l'avons déjà souligné, l'idée de base est de retrouver la meilleure projection du modèle, au sens "la plus ressemblante" aux images réelles. Nous définissons donc la meilleure projection comme étant celle qui optimise la mesure d'écart évoquée ci-dessus. Nous présentons l'approche que nous utilisons pour y parvenir dans la section 4 page 43 ;
- nous abordons ensuite la façon dont nous étendons ces traitements au niveau des séquences, afin de réaliser le suivi au cours du temps proprement dit (section 5) ;
- le principe du simplexe et différentes variantes procède par itération, ce qui pose l'inévitable question de l'initialisation. La section 6 est une tentative de réponse à ce problème.

# 2

# Modélisation de la scène et de l'objet suivi

## Sommaire

Comme nous l'avons dit, nous travaillons à partir d'un modèle "synthétisable" de l'objet. Ce modèle décrit à la fois les informations structurelles liées à la cinétique de l'objet et les informations nécessaires à la création des vues artificielles – géométrie et propriétés visuelles. Les paramètres représentent les caractéristiques variables de l'objet, telles que sa position, l'état des articulations où encore la couleur des surfaces.

Nous allons préciser en quoi consiste précisement ce modèle, et quelles sont les informations qu'il apporte.
Nous parlerons également du problème de la modélisation de la scène dans laquelle l'objet évolue, et en particulier de la définition des caméras virtuelles et de l'éclairage de cette scène.
Disposant du modèle de l'objet, un moteur de rendu nous permet de générer les vues de synthèse correspondant à un vecteur de paramètres. Une description succincte de ce moteur ainsi que des possibilités de description associées sera donné.

## 2.1 Modèle de l'objet

Dans notre approche la modélisation de l'objet suivi est une étape importante. En effet c'est sur la base de ce modèle, par l'intermédiaire du moteur de rendu, que nous pourrons tester et améliorer la validité des paramètres courants.

Ce modèle englobe plusieurs aspects nécessaires (voir figure 2.1) :

- l'aspect structurel de l'objet. C'est ce qui va décrire les mobilités associées à l'objet, et la façon dont elles interagissent ;
- l'aspect géométrique. Cela inclue tout ce qui concerne la géométrie des éléments de l'objet, leurs positions et tailles ;
- les caractéristiques visuelles de l'objet. Tout ce qui a trait à l'habillage des surfaces et qui permet un rendu plausible de l'objet se trouve dans cette catégorie. On peut citer pour exemple les couleurs et les propriétés des surfaces, telles que la réflectance.

Le premier point cité est l'aspect structurel de l'objet. C'est à cet aspect que sont liés les paramètres du modèle.
Dans la description structurelle se trouvent regroupé tout ce qui défini l'organisation des parties de l'objet, à travers ses mobilités, ses articulations... Cette description structurelle se rapproche du concept de *squelette* de l'objet souvent utilisé dans le domaine de la modélisation. Les degrés de liberté définissent la dimension de l'espace des paramètres. Les paramètres représentent les axes de cet espace, leurs valeurs permettant d'instancier le modèle dans une posture donnée.
Il est à noter qu'à chaque paramètre correspond un certain nombre de contraintes. Les principales sont les limites de variations des valeurs du paramètre (s'il y en a), et la tolérance accordée, qui donne le seuil à partir duquel deux valeurs sont considérées comme identiques.

La génération d'images de synthèse du modèle correspondant à une vue donnée implique d'avoir des informations de nature géométrique sur l'objet, tels que ses dimensions. Ces informations géométriques sont utilisées pour effectuer la projection du modèle dans l'espace image via la synthèse d'image. De même lors de la synthèse d'image le moteur

de rendu a besoin des caractéristiques graphiques des surfaces de l'objet pour effectuer le rendu lui-même, afin de générer les couleurs, reflets, textures... correspondant.

Ainsi lorsque nous parlerons de modèle géométrique paramétré nous ferons référence à un modèle d'objet munie d'une description géométrique et visuelle, et incluant les paramètres contrôlant son état.

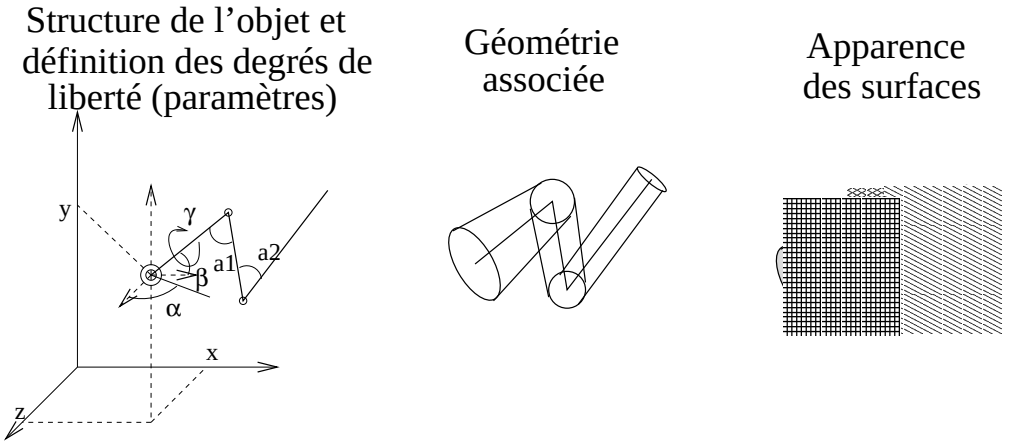



FIG. 2.1 – *Les différentes informations présentent dans le modèle de l'objet : la structure (liée aux paramètres), la géométrie et l'habillage. La structure définie comment les degrés de liberté de l'objet se traduisent au niveau du modèle. La géométrie décrit comment l'objet occupe l'espace autour de l'*ossature *définie par la structure, et enfin l'habillage permet de définir l'apparence de l'objet pour la synthèse d'image.*

## 2.2 Modélisation d'une scène

Nous avons précisé dans la section précédente ce que nous appelons modèle paramétré d'un objet. Pour être utilisable, c'est-à-dire pour pouvoir être projeté afin d'obtenir des vues de synthèse réaliste de celui-ci, il doit appartenir à une scène. Cette scène va englober un ensemble d'éléments nécessaires au rendu (voir figure 2.2) :

- un référentiel : il est arbitraire, mais permet de définir un rapport de mesures dans l'espace de l'objet et de positionner les différents éléments de la scène ;
- un ensemble de vues : définies par un centre et une direction de projection, elles doivent se rapprocher des caméras réelles qui filment l'objet ;
- un éclairage : il définit la source et les caractéristiques des lumières de la scène.

Le référentiel, arbitraire comme nous l'avons dit, sert à définir et placer les différentes entités de la scène les unes par rapport aux autres, ainsi qu'à définir l'unité de mesure des paramètres géométriques.

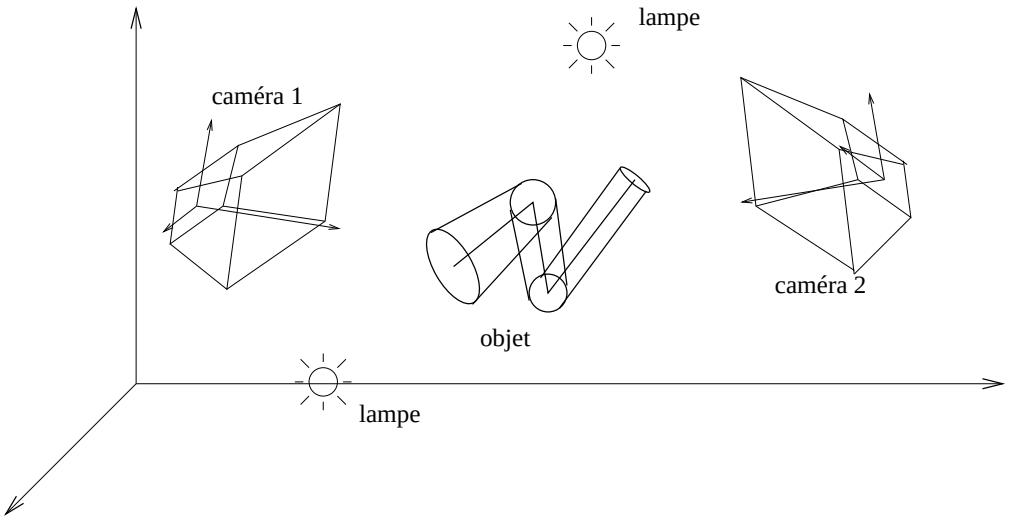



FIG. 2.2 – *Entités présente dans une scène : l'objet suivi, les caméras et les éclairages*

### 2.2.1 Calibrage des caméras

Les caméras constituent un point important de la scène 3D. Celles-ci vont en effet simuler ce que "voient" des caméras réelles.

Le processus de calibrage a été longuement traité par la communauté de la vision et la littérature sur ce sujet est très riche. Une excellente présentation peut être trouvée dans [Fau93]. En général le calibrage est réalisé sur la base de l'acquisition d'un ensemble de points 3D dont la position est connue. L'association de ces points à leur projection dans les images permet de former un ensemble d'équations faisant intervenir les paramètres de calibrage.

La littérature propose plusieurs modèles de caméra [Ama97, Bou94, McR97], que l'on peut classer en trois grandes catégories :

- Sans déformation :
  - avec projection perspective (modèle sténopé). Il s'agit d'un modèle de projection centrale ;
  - avec projection orthographique. Il s'agit du cas limite des projections perspectives pour lesquelles le centre de projection est à l'infini.
- Avec déformation et rétro-projection (modèle sténopé avec déformation). Il s'agit du même principe que le modèle sténopé précédent, auquel on adjoint un déplacement local permettant de modéliser les déformations dues aux lentilles ;
- Avec déformation et sans rétro-projection (modèle des multi-plans). La droite de projection de chaque pixel est calculée à partir d'exemples de projections pour deux plans de calibrage distincts. En pratique ce processus est appliqué à un échantillon de points et une interpolation est utilisée pour les autres.

Les modèles sans et avec déformation se différencient par la possibilité d'avoir ou non directement la droite de rétro-projection d'un point. Rappelons que la droite de rétro-projection d'un point de l'image est la droite passant par ce point et reliant le point 3D correspondant. Dans certains modèles (multi-plans en particulier) cette droite n'est pas

directement accessible.

Le modèle de caméra utilisé dépendra de plusieurs facteurs, tels que :

- le type de caméra physique utilisé ;
- les conditions de prise de vues, qui vont définir les approximations qu'il est possible de faire, en particulier pour négliger les déformations (effets de lentille par exemple) ;
- la précision finale désirée, un modèle plus simple limitant la fiabilité des projections calculées.

Le calibrage consiste à modéliser les caméras et à les positionner dans le repère de la scène. Dans le cadre de nos travaux nous supposons que les informations liées au calibrage sont fournies par l'utilisateur. Ces traitements doivent donc être réalisés en travail préparatoire, avant de filmer les objets. Il est également possible d'utiliser des caméras auto-calibrées, celles-ci étant capable de transmettre directement leur position, orientation, focale... à un ordinateur. Ce type de matériel est fréquemment utilisé dans le domaine des effets spéciaux, où scène réelle et éléments de synthèse sont mélangés. Si l'on n'a pas accès à se type de matériel – la principale limite étant le coût élevé – on doit recourir à des méthodes de calibrage *logicielles*.

De nombreux travaux existent sur le calibrage des caméras, dont une description peut être trouvée dans [Bou94], [Ama97] et [HDLC97]. Cela peut se faire à partir d'une mire dont la forme et la position sont parfaitement connues (par exemple dans [Ama97] et [TLV+95]), le but étant d'effectuer une acquisition de cette mire et de déterminer le système de passage de la mire à l'image.

La recherche des paramètres projectif peut également passer par l'étude des conséquences d'une projection sur des ensembles de primitives – en général des points. On peut ainsi exprimer des relations de contraintes entre un espace 2D et un ensemble de points 3D [Qua98, WR96]. En fonction du nombre de points et du nombre d'images disponibles [Qua98, ST96], différents type de contraintes peuvent être définies, tels que des invariants projectif entre lignes à partir d'invariants projectif entre points [QK97]. Il s'agit ensuite, à partir de ces contraintes, de résoudre les équations générales afin d'obtenir la calibrage des caméras. D'autres approches se basent sur la donnée d'un modèle de surface [FL94] et ajustent les paramètres de la caméra en minimisant une fonction basée sur l'appartenance des projections des points à la surface en question. Dans [Fau98], le calcul variationnel est utilisé pour étalonner un ensemble de vues les unes par rapport aux autres, permettant de définir les paramètres (relatifs) correspondant à ces vues.

Des méthodes plus simples et plus répandues [Bou94] se basent sur la recherche des paramètres de la caméra par minimisation entre les projections de primitives 3D (points, lignes, cercles...) dans les images et leurs projections 2D théoriques – via le modèle de la caméra.

Pour réaliser le calibrage de nos caméras pour nos séquences vidéo de tests, notre choix s'est porté sur un type d'approche minimisant les distances entre des primitives 3D de la scène – des points – et leur projection constatée dans les images réelles.

Ce choix était naturel dans notre cas car, comme nous le verrons dans la partie suivante, notre méthode de recherche des paramètres d'objets se base sur la minimisation d'une mesure d'écart entre images. Ayant déjà à utiliser des techniques d'optimisation, seule la définition de la mesure à minimiser change. Cette mesure est dans le cas du ca-

librage liée aux distances entre la projection de points 3D connus dans la scène réelle et les points correspondants dans les images.

Cette technique sera détaillée plus longuement dans la section 6.3 page 68, après que nous ayons présenté notre approche pour le suivi proprement dit.

Nous avons choisi cette dernière, car elle présente l'avantage d'être relativement aisé à mettre en œuvre, pour des précisions généralement suffisantes aux vues des qualités optiques des caméras utilisées.

### 2.2.2 Éclairage de la scène

Un autre aspect important d'une scène 3D est l'éclairage, c'est-à-dire la définition des sources lumineuses éclairant la scène.
Pour être cohérent vis-à-vis de la scène réelle où l'objet évolue, ces sources de lumières doivent simulées l'éclairage réel. Ce problème est délicat et suscite encore des recherches. La difficulté vient du fait que :

- les positions des sources lumineuses réelles ne sont pas forcement accessible, ou bien seulement de façon approximative ;
- la forme de ces sources lumineuses – néons, ampoules, fenêtres... – peut être complexe et donc difficile à traiter par le moteur de rendu ;
- la quantité de lumière émise par ces lampes est difficilement quantifiable en général, sauf en utilisant du matériel professionnel de mesure ;
- des réflexions indirectes sur des objets non modélisés peuvent contribuer à l'éclairement global.

Tous ces points rendent difficile la simulation précise de l'éclairage d'une scène. Nous avons pris le parti de tolérer des éclairages approximatifs, aussi bien au niveau de la position, de la forme et de la quantité de lumière émise, car cela rendrait le rendu nettement trop long par rapport au gain de précision espéré. Les déviances entre la scène réelle et la scène de synthèse seront directement prises en compte par la mesure d'écart entre images, dans la section 3.

## 2.3 Moteur de rendu

Le modèle de l'objet, et de la scène 3D représentant la scène réelle, est utilisé pour simuler des vues de l'objet, telles que les voient les caméras.

Les approches existantes pour générer des images de synthèse sont nombreuses, et il existe une littérature très fournie sur ce domaine.

Les méthodes de rendu les plus connues (pour le rendu réaliste) sont :

- les méthodes utilisant la technique du *ZBuffer*, dont l'exemple le plus célèbre est *OpenGL*[OpG] ;
- les méthodes *lancé de rayons*[App68, Whi80], simulant les rayons lumineux arrivant à la caméra ;
- les méthodes dites *radiosité*[GTGB84, NN85], recherchant l'illumination globale d'une scène en utilisant les propriétés physiques des surfaces et de la lumière.

Les différences entre ces méthodes sont liées aux techniques mises en oeuvre pour calculer l'apparence et visualiser les éléments d'une scène 3D. Les méthodes telles celles utilisées par OpenGL sont en général rapide et relativement simple à mettre en oeuvre, mais ne permettent pas de prendre en compte certains phénomènes physiques, ou alors seulement en les simulant (ombres douces, caustiques...). Les méthodes de type lancé de rayons permettent de prendre en compte un grand nombre de phénomènes, tels que les réflexions, transparences... Une limite est de ne pouvoir gérer les illuminations de façon globale, bien que certain travaux actuels cherchent à représenter ces effets dans le cadre des lancés de rayons[ZP98, Zan98]. Les méthodes fondées sur la radiosité prennent en compte l'aspect global de l'illumination, permettant de calculer les contributions globales des sources de lumières, aboutissant à un grand réalisme dans le rendu. Elles ont toutefois l'inconvénient d'être très coûteuses en calculs. De nombreux travaux tels que [MBM98] portent sur l'accélération de ces traitements afin de pouvoir obtenir des qualités de rendu élevées en des temps de calcul raisonnables, par la parallélisation ou encore la gestion des niveaux de détails.

Dans notre cas la ressemblance visuelle est plus importante que la validité physique du rendu. De plus le temps de calcul est très important car notre méthode sera appelée à effectuer de nombreux rendus. Ainsi notre moteur de rendu est basé sur OpenGL. La raison de ce choix est la grande portabilité de celui-ci, OpenGL étant supporté par de nombreuses plates-formes. En outre direct est le support d'OpenGL par de nombreuses cartes d'accélération 3D, permettant une optimisation matérielle – et transparente – des rendus. Dans notre processus, la phase de rendu est souvent utilisée, une accélération n'est donc pas à négliger.

### 2.3.1 Description

Nous avons créé un système de description de scènes et de modèles. Celui-ci permet de définir la structure de l'objet de façon hiérarchique, ainsi que les paramètres qui se rattachent au degrés de liberté de l'objet. À cela s'ajoute l'ensemble des caractéristiques géométriques, de la géométrie des surfaces en passant par les caractéristiques visuelles de celles-ci.
Ce système permet également la définition des sources de lumière de la scène et leurs caractéristiques (couleur, forme, quantité de lumière émise).

En dernier lieu vient la définition des caméras. En sus des caractéristiques constantes telles que la résolution et le type (couleur ou noir et blanc par exemple) s'ajoute la définition des paramètres de projection, représenté par une matrice de projection homogène $4 \times 4$ (les déformations liées aux optiques sont négligées). Ces caractéristiques peuvent être constante – dans le cas de caméras fixes – ou bien liées à un fichier décrivant leur évolution au cours du temps.

Ce système, contrôlé par notre programme de suivi, permet d'obtenir la projection de la scène pour une caméra donnée, à partir d'un jeu de valeurs pour les paramètres (Fig. 2.3 page suivante).

Il est à noter que l'on peut inclure dans la scène d'autres objets que celui suivi. Cela peut s'appliquer au sol ou aux murs, ou bien d'autres éléments de "décors" connus. L'intérêt est alors un gain de précision si certains de ces objets peuvent cacher l'objet

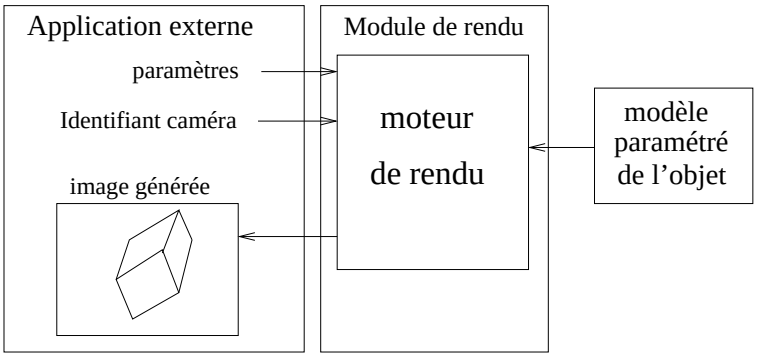



FIG. 2.3 – *Utilisation du moteur de rendu à partir d'un programme extérieur. Le moteur de rendu est considéré par notre système comme une boîte noire : on lui transmet les valeurs des paramètres à utiliser ainsi que la définition de la vue et celui-ci retourne l'image de synthèse correspondante.*

suivi, ou générer des ombres sur celui-ci. La prise en compte de ces effets contribue à rendre les vues de synthèse plus fidèles à la réalité.

Notons toutefois que notre approche a été conçue pour tolérer des images de synthèses présentant de petites divergences par rapport aux images réelles, en particulier au niveau des erreurs d'éclairement, comme nous le verrons plus loin. De même la définition des éléments de décors permet d'augmenter la précision de notre approche, mais n'est pas fondamentale. Seul le modèle de l'objet suivi est requit pour un fonctionnement normal.

# 3

# Mesure d'écart entre images

## Sommaire

La mesure d'écart entre images réelles et images de synthèse est au cœur de notre méthode. C'est en effet elle qui va permettre de mesurer à quel point un ensemble d'images de synthèse est proche de l'ensemble de vues réelles correspondantes. Les images de synthèse étant liées aux valeurs des paramètres utilisés pour le rendu, cette mesure d'écart permet donc de déterminer la qualité des valeurs des paramètres par rapport à l'objet suivi.

Afin d'être adapté à notre problème, la mesure d'écart doit prendre en compte plusieurs points importants. En premier lieu elle doit présenter un minimum pour les valeurs théoriques des paramètres, afin que la phase d'optimisation qui suivra ait un sens. Elle doit également prendre en compte les différences entre les images à comparer, comme par exemple :

- les erreurs de modélisation ;
- les ombres portées dues à des objets présents dans la scène mais non modélisés ;
- la présence de bruit ou d'aliassage ;
- le type d'images vidéo utilisées – couleur ou noir et blanc.

Cette mesure d'écart doit de plus être robuste aux occultations, c'est-à-dire au fait que l'objet réel suivi peut être partiellement caché par des objets non modélisés, et donc présents seulement dans les images réelles.

Nous allons en premier lieu détailler plus précisement les contraintes qu'imposent le fait de comparer des scènes réelles et des images de synthèse.

Suivra ensuite la définition de la mesure d'écart, tout d'abord entre une paire d'image réelle / image de synthèse, puis pour l'ensemble des paires disponibles à un instant $t$.

Nous précisons ensuite la façon dont l'information de couleur est prise en compte dans les images, en particulier vis-à-vis des contraintes de robustesse que nous nous somme fixé.

## 3.1 Introduction de la notion d'écart entre images

Pour quantifier la ressemblance entre des images réelles et des images de synthèse, il nous faut définir les critères sur lesquels est basé cette ressemblance. En effet la notion de concordance entre image, encore appelé similarité, dépend fortement de ce que l'on cherche à mettre en concordance.

### 3.1.1 Similarité d'images

La notion de similarité entre images regroupe en fait un grand nombre de démarche différentes, selon les critères que l'on cherche à appliquer. Il est possible de définir trois grandes catégories :

- les approches basées sur des critères de haut niveau. Généralement utilisés pour la reconnaissance d'objets ou encore la classification automatique. Elles reposent sur la détermination de caractéristiques des objets à partir des images, telles que les couleurs, formes et orientations, ou bien à partir d'éléments de plus haut niveau tels que la courbure d'une entité ;

- les approches cherchant à déterminer la qualité d'un rendu de synthèse, c'est à dire essayant de mesurer les différences entre ce rendu et une situation de référence – photo, mesures photogramétriques... – afin d'en déterminer les différences ;
- les approches basées sur les pixels eux-mêmes. Il s'agit dans ces cas là de déterminer des ressemblances à partir de critères visuels (quel que soit le sens que l'on donne à ce terme), ou encore de calculer les transformations (affines par exemple) entre des images.

La première de ces catégories peut reposer sur des critères très divers. Cela peut être des contours, des régions ou des points d'intérêt... On trouve dans [SJ96] de nombreux exemple de critères de similarités utilisés pour la classification. Ces approche permettent de reconnaître la similarité entre deux objets, mais ne sont guère aptes à quantifier les différences entre ces deux objets.

La deuxième catégorie se retrouve principalement dans les étapes de validation des système de rendu réalistes, visant à simuler des phénomènes réels tels que la contribution énergétiques d'un éclairage dans une scène par exemple. On trouve dans [Ahu93] un état de l'art sur des métriques de qualité pour les images de synthèse. D'autres travaux tels que [RWP+95],[ATS94] et [TLG91] décrivent également des approches possible pour établir la qualité d'un rendu, par rapport à des critères tels que l'illumination. Ces techniques sont très utiles pour valider des systèmes de rendu dont la vocation est le réalisme (jusque dans les phénomènes physiques mis en œuvre), mais n'est pas réellement intéressante pour nous car nous cherchons à obtenir une mesure d'erreur sur les positionnements des objets, et non pas sur la qualité intrinsèque du rendu. De plus le propos du rendu de synthèse dans notre approche n'est pas d'obtenir des images en accord avec les lois de la physique, mais simplement d'obtenir des images suffisement ressemblante pour que les comparaisons aient un sens.

La troisième catégorie décrite est généralement utilisée pour calculer des transformations entre deux objets, comme par exemple les translations, rotations et éventuellement déformations. Souvent appelée dans la littérature anglaise *image registration*, cette technique sert beaucoup dans les problèmes de recalage, et présente l'avantage de fournir une mesure "d'éloignement" entre les deux images comparées.

Cette dernière correspond au type de problème que nous avons, puisqu'il s'agit pour nous de mesurer l'écart entre une image de synthèse d'un objet et une vue réelle de celui-ci.

Ce type d'approche suppose que les images comparées sont visuellement similaire. Nous soulignons plusieurs différences qui peuvent exister entre une image réelle et une simulation de synthèse :

- l'objet peut être partiellement occulté dans les images réelles. Cette occultation ne sera a priori pas présente dans les images de synthèse, car nous ne supposons connu que le modèle de l'objet suivi ;
- les erreurs de modélisation au niveau de l'objet ou de la scène, qui vont engendrer des différences entre les images issues de la vidéo et les vues de synthèse ;
- des ombres portées, dues à des objets inconnus (non modélisés) dans la scène réelle,

et donc absentes des images de synthèse ;
- l'aspect "différent" des vues réelles et des images de synthèse, comme la présence de bruit ou d'aliassage dans les images vidéo par exemple ;

Tout ces points font que les images qui vont être comparées ne sont pas identique. La mesure d'écart que nous introduisons doit en tenir compte. La façon dont ceci est réalisé est décrite dans les sections suivante.

## 3.2 Mesure d'écart entre deux images

La mesure d'écart que nous cherchons à définir a pour but de quantifier l'écart apparent entre des images d'un même objet, les unes étant des vues réelles et les autres des vues artificielles, générées à partir du modèle géométrique paramétré de cet objet.

Nous allons tout d'abord nous concentrer sur la définition d'une mesure d'écart entre deux images, l'une réelle contenant l'objet étudié, et l'autre de synthèse contenant une simulation de ce même objet, dans une "posture"[1] pouvant être (légèrement) différente. La prise en compte de l'ensemble des vues disponibles sera décrite dans la section 3.4.

On cherche à ce que cette mesure soit proportionnelle à la différence apparente, et ainsi qu'elle présente un minimum lorsque la concordance visuelle est maximum. Nous travaillons toujours dans des situations où les images sont relativement proches. Ainsi l'objet dans les images de synthèse est relativement proche de l'objet dans l'image réelle, et les méthodes de type comparaisons denses (*image registration*) sont très bien adaptées, puisqu'elles sont très efficace pour mesurer les transformations entre deux motifs similaires. De plus l'utilisation d'objets articulés n'est pas gênant, puisqu'il est possible de ramener le problème à la comparaison d'un ensemble d'objets rigides, chacun étant ici une des parties de l'objet articulé initial.

### 3.2.1 Formulation

Soit $\Pi$ le plan de projection de l'image vidéo considérée. On défini le domaine $\mathcal{D}$ de $\Re^2$ comme étant le domaine de $\Pi$ correspondant à la projection du modèle de l'objet dans $\Pi$. En pratique $\mathcal{D}$ correspond à la silhouette de l'objet sur le plan de projection de la caméra. Notons que $\mathcal{D}$ est utilisé comme domaine de projection pour les images réelles et les images de synthèse car nous travaillons sur des images de synthèse proche de la réalité et de plus le domaine correspondant à l'objet réel n'est pas accessible. Si $\mathcal{M}$ est le modèle de l'objet, c'est-à-dire un ensemble de points de $\Re^3$ on a donc :

$$\mathcal{D} = Proj_{\Pi}(\mathcal{M}) \tag{3.1}$$

Afin de pouvoir mesurer la différence entre deux images $I_1$ et $I_2$ nous introduisons ce que nous appelons une fonction d'écart $\mathcal{E}$. Cette fonction doit satisfaire plusieurs propriétés :

[1] *posture* est un terme qui s'applique normalement à un être vivant. Par abus de langage nous l'emploierons ici pour désigner l'état spatial d'une entité, que celle-ci soit un être vivant ou un objet éventuellement articulé.

- $\mathcal{E}$ doit être à valeurs positives dans $\Re$;
- $\mathcal{E}$ doit être décroissante au sens de la différence entre les images, c'est-à-dire que $\mathcal{E}$ est une mesure de corrélation entre images;
- $\mathcal{E}$ doit présenter un minimum, et en particulier : $\mathcal{E}(I, I) = 0$.

### 3.2.2 Formulation continue

Soient deux images $I_1$ et $I_2$, et soient $I_{1/\mathcal{D}}$ et $I_{2/\mathcal{D}}$ respectivement la restriction de $I_1$ et $I_2$ au domaine $\mathcal{D}$. On dit que $I_1$ est $\alpha$-adéquat à $I_2$ au sens de la fonction écart $\mathcal{E}$ si et seulement si :

$$\mathcal{E}(I_{1/\mathcal{D}}, I_{2/\mathcal{D}}) < \alpha \tag{3.2}$$

Bien sûr cette fonction peut avoir de multiples expressions. Dans notre cas nous la définissons comme suit :

$$\mathcal{E}'(I_1, I_2) = \int_{\mathcal{D}} d_{coul}(I_1[x, y], I_2[x, y]).dx.dy \tag{3.3}$$

avec $d_{coul} : \mathcal{D} \times \mathcal{D} \rightarrow \Re$ une distance entre couleurs qui sera détaillée dans la section suivante.

Nous normalisons finalement cette fonction par rapport au domaine $\mathcal{D}$ :

$$\mathcal{E}(I_1, I_2) = \frac{\mathcal{E}'(I_1, I_2)}{\int_{\mathcal{D}} dx.dy} \tag{3.4}$$

### 3.2.3 Formulation discrète

Notre domaine d'application étant discret, nous définissons une expression discrète de $\mathcal{E}$. L'intégrale sur le domaine $\mathcal{D}$ correspond à un calcul de sommes sur les points des images. Pour deux images $I_1$ et $I_2$, nous définissons ainsi la mesure discrète d'écart entre deux images comme suit :

$$\mathcal{E}(I_1, I_1) = \frac{1}{Card(\mathcal{D})} \sum_{(x,y)\in\mathcal{D}} d_{coul}(I_1[x, y], I_2[x, y]) \tag{3.5}$$

Nous explicitons dans la section suivante la distance entre pixels $d_{coul}$, puis nous étendons la mesure d'écart pour qu'elle prenne en compte l'ensemble des vues disponibles.

## 3.3 Distance entre pixels

Dans la section précédente nous avons introduit une distance entre pixels pour réaliser notre mesure d'écart entre images.

Dans le cas d'images en niveaux de gris, il s'agit alors de mesurer une distance entre deux valeurs scalaire (généralement ramenées entre 0 et 1). Les solutions envisageables sont alors en nombre restreint, les plus utilisées étant la distance euclidienne $|v_2 - v_1|$ ou la distance quadratique $(v_2 - v_1)^2$, si $v_1$ et $v_2$ sont les valeurs de deux pixels à comparer.

Comme nous nous plaçons dans le cas plus général d'images en couleur, chaque pixel doit être alors considéré comme un triplet de valeurs scalaires. Notons que nous nous plaçons dans le cas de caméras couleur. Il existe des matériels spécifiques permettant d'obtenir des images autres que noir et blanc ou couleur, avec en particulier les infrarouge ou même les rayons à haute énergie (rayons X et rayons *gamma*, en radio-astronomie principalement). La définition d'une distance présente alors beaucoup plus de variations possibles, et va dépendre de ce que l'on cherche à mettre en avant.

Avant de définir notre distance entre pixels, dans la section 3.3.2, nous allons tout d'abord présenter le problème du choix de l'espace colorimétrique et des distances entre couleurs dans la section suivante.

### 3.3.1 Espaces colorimétriques

Un aspect à ne pas négliger est l'utilisation d'images couleur. Nous nous plaçons en effet dans le cas d'images vidéo – et donc d'images de synthèse – en couleurs. Ceci a des impacts à plusieurs niveaux.

Les pixels ne sont plus assimilables à une valeur scalaire comme dans le cas des images en niveaux de gris, mais à un triplet de valeurs : rouge, vert et bleu ou luminosité, teinte et saturation dans la plupart des cas.

Il existe en fait de très nombreux espaces de représentation de la couleur [Pra91]. Ces espaces ont été définis pour obtenir des propriétés particulières que ne possèdent pas les formats "classiques" en informatique, en particulier au niveau de la définition de distances dans ces espaces.

L'espace RGB, très répandu car utilisé par la plupart des systèmes graphiques pour la représentation des couleurs, ne possède pas de propriétés particulières. Des notions utiles telles que la luminosité, la *couleur* ou la saturation ne sont pas directement accessibles. D'autres espaces tels que le HLS (*Hue, Light, Saturation*) ou le HSV (*Hue, Saturation, Value*) séparent ces caractéristiques de façon directe. Le but est d'avoir des axes qui soient indépendants les uns par rapport aux autres (contrairement au RGB par exemple). Le lecteur se reportera à l'annexe A pour plus de précisions sur les espaces colorimétriques et leurs propriétés.

La mesure d'écart utilisant des comparaisons pixel à pixel, il faudra être capable de définir une distance entre des triplets de valeurs. Notons au passage que la couleur et l'information qu'elle porte est de plus en plus étudiée, que ce soit pour la définition d'invariants[GMDP00] ou pour la segmentation[lim99] dans les séquences vidéo.
Un autre point important est la différence de nature des images comparées. D'un coté nous avons des images vidéo, et de l'autre des images de synthèse, dont les caractéristiques de couleur seront différentes. On citera en particulier :

- la fonction *gamma*, qui correspond à la réponse en luminosité d'une image. Celui-ci n'a *a priori* pas la même valeur que la fonction gamma des images vidéo, générant une dynamique des couleurs légèrement différente ;
- les variations d'éclairage. La scène de synthèse ne peut reproduire toutes les subtilités d'un éclairage réel (réflexions, diffusions, positions et formes des lampes...). Ceci

a un impact sur l'aspect exact des surfaces de l'objet suivi ;
- les états de surface de l'objet. Une surface plus ou moins rugueuse, et surtout une distribution irrégulière de ces caractéristiques entraîne des variations de réponse de la surface par rapport aux lumières. Ces aspects sont très difficiles à modéliser dans une scène de synthèse ;
- des ombres portés, dues à des objets non modélisés, peuvent modifier l'apparence de certaines surfaces.

Même dans le cas d'une comparaison entre une image réelle et une image de synthèse montrant le même objet dans la même posture, toutes ces variations ont un effet sur la comparaison entre les pixels – les pixels comparés n'ayant pas exactement la même couleur – et donc sur la mesure d'écart.

La plupart des différences entre ces images ont un impact principalement sur la luminosité des surfaces, et beaucoup moins sur la "couleur" elle-même.
C'est en particulier le cas des ombres, qui assombrissent les surfaces sans changer fondamentalement les couleurs, ou encore les erreurs d'éclairage dans la mesure ou les lampes sont monochromatiques (des lampes changeant de couleurs auraient bien sur des effets différents).
Certaines autres différences entraînent des variations de la luminosité et de la couleur, c'est le cas des états de surfaces qui peuvent générer des reflets. Ce sont par contre des effets généralement localisés, et donc les erreurs qu'ils génèrent sont compensés par l'approche globale de la comparaison entre images (voir la section 3.2).

C'est pour toutes ces raisons que nous avons choisi d'exprimer les couleurs dans un espace colorimétrique séparant les caractéristiques de luminosité (appelé *luminance*) et de couleur (appelé *chrominance*).

Nos préoccupations étant d'avoir accès séparément aux composantes de luminance et de chrominance, nous avons choisi parmi les espaces séparant ces composantes l'espace colorimétrique $YC_rC_b$, dans lequel $Y$ exprime la luminance et $C_r$ et $C_b$ les composantes de la chrominance. La chrominance s'exprime donc sous forme d'un espace à deux dimensions. La matrice de passage de l'espace RGB à l'espace $YC_rC_b$ s'écrit :

$$\begin{pmatrix} Y \\ C_r \\ C_b \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} . \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.6}$$

Chaque fois que nous parlerons de couleur de pixels par la suite, nous ferons référence aux composantes $YC_rC_b$ correspondantes.

### 3.3.2 Distance entre couleurs

Parmi les contraintes que nous nous sommes fixé pour la mesure d'écart se trouvent la robustesse aux erreurs de modélisation de l'éclairage – position et intensité – ainsi qu'aux ombres portées non prévues – venant d'objets non modélisés. Nous avons vu dans la section précédente que ces variations entre les images de synthèse et les images vidéo se traduisent principalement par un impact sur la luminance des pixels correspondants, laissant la chrominance globalement inchangée.

Ainsi pour définir notre distance entre couleurs nous séparons le problème en deux sous-problèmes, exprimés dans respectivement dans l'espace des chrominance et dans l'espace des luminances. Nous définissons deux distances partielles $d_{chrom}$ et $d_{lum}$, chacune dans un des espace considéré. Soit $c_1$ et $c_2$ deux couleurs ayant pour valeurs dans l'espace $YC_rC_b$ $(y_1, c_{r_1}, c_{b_1})$ et $(y_2, c_{r_2}, c_{b_2})$. La distance entre deux couleurs $d_{coul}$ s'écrit :

$$d_{coul}(c_1, c_2) = \alpha.d_{chrom}(c_1, c_2) + (1 - \alpha).d_{lum}(c_1, c_2) \tag{3.7}$$

On définit les deux sous-distances $d_{chrom}$ et $d_{lum}$ à valeurs de $\mathcal{C} \times \mathcal{C}$ dans $\Re_+$comme suit :

$$d_{chrom}(p1, p2) = \sqrt{(c_{r_1} - c_{r_2})^2 - (c_{b_1} - c_{b_2})^2} \tag{3.8}$$

$$d_{lum}(p1, p2) = |y_1 - y_2| \tag{3.9}$$

La figure 3.1 montre le comportement de ces deux distances pour des variations d'éclairage et d'ombres portées non prévues. Sur la première ligne l'image de référence utilisée est l'image de synthèse avec sur-échantillonage et bruit blanc, avec en regard respectivement la carte des distances dans l'espace des chrominances et des luminances. La deuxième ligne montre le même objet, mais avec un éclairage plus fort. Enfin, la troisième ligne illustre ce qui se passe lorsqu'une zone de l'objet est dans l'ombre. Toutes ces images sont comparées avec l'image de la dernière ligne, qui est le modèle généré par le moteur de rendu. On constate que dans tous les cas la distance dans l'espace des luminances est plus élevée que la distance dans l'espace des chrominances. Le "bruit" dans les cartes de distance est dû au bruit blanc présent dans les images de référence.

Il est à noter que la gamme de luminosité des cartes de distance a été augmenté afin d'être plus visible. En réalité sur cet exemple les différences sont relativement faibles et les erreurs restent très petites. Par contre pour chaque couple distance luminance / distance chrominance la même augmentation à été utilisée afin de permettre la comparaison entre les deux.

Afin d'être robuste vis-à-vis des points cités plus haut, nous devons tenir compte principalement de l'information apportée par la distance de chrominance, la distance de luminance apportant bien sur sa part d'information, mais seulement dans les cas où aucun problème d'éclairement ou d'ombres non prévues sont présents dans la scène.

Le terme $\alpha$ est un paramètre de poids variant entre 0 et 1 qui permet de contrôler la contribution relative des deux sous-distances. Dans notre cas, $\alpha$ sera choisi tel que $\alpha > \frac{1}{2}$ afin de respecter les contraintes que nous avons fixé.

## 3.4 Mesure d'écart entre plusieurs paires d'images

Nous avons vu dans la section précédente la définition de la mesure d'écart entre deux images. Il nous faut maintenant étendre cette mesure afin qu'elle prenne en compte les différentes vues disponibles de l'objet.

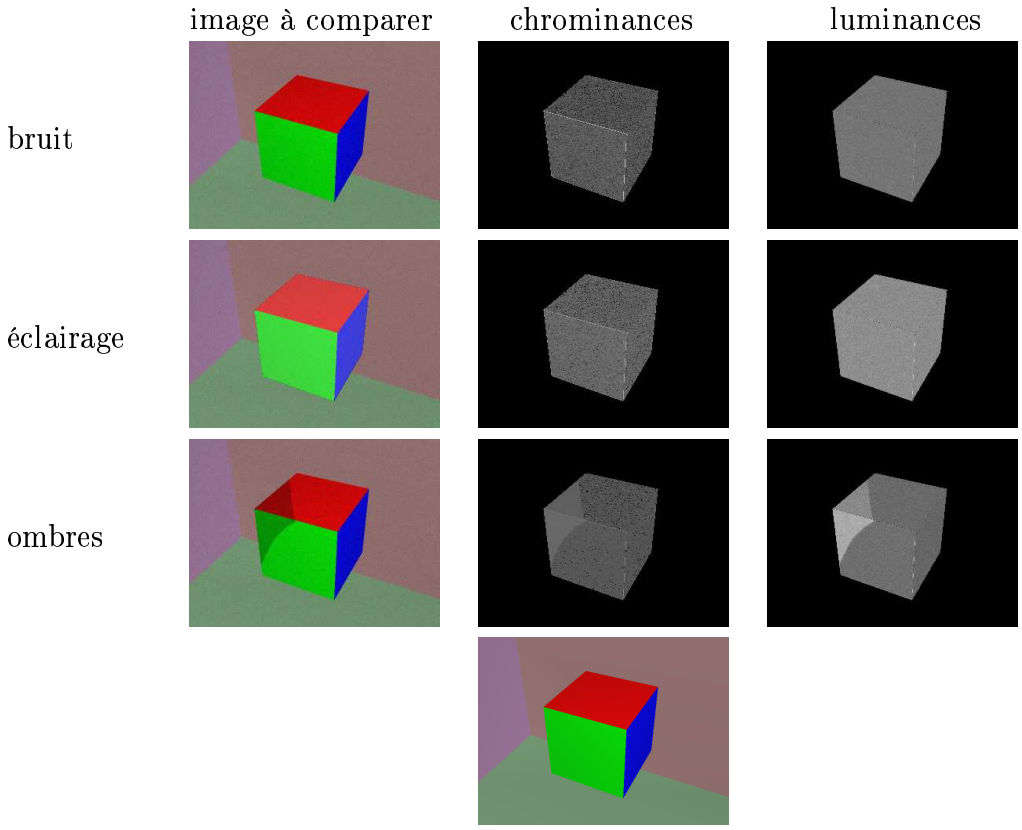



FIG. 3.1 – *De gauche à droite : image de référence, distances pixel des chrominances et distance pixel des luminances. Dans chaque cas, l'image est comparée avec l'image donnée à la dernière ligne, notre image de référence. Notons que pour plus de visibilité, les contrastes des images ont été augmentés, les gammes de valeurs correspondent donc à un facteur près aux valeurs initiales.*

Les images de synthèse $\mathcal{I}_c(p)$ correspondant à chacune des caméras $c$ sont générées, à partir d'un jeu de paramètres.
La mesure d'écart entre deux images $\mathcal{E}$ est appliquée à chacune des paires image réelle / image de synthèse. Ces mesures sont ensuite regroupées grâce à une combinaison linéaire, pour donner la mesure d'écart finale $D$ :

$$D_m(p) = \sum_{i=0}^{i<N_c} \alpha_i \mathcal{E}(I_i, \mathcal{I}_i(p)) \tag{3.10}$$

avec $N_c$ le nombre de caméras, $p$ un vecteur de paramètres et les $\alpha_i$ des poids tels que $\alpha_i >= 0$ et$\sum_{i=0}^{i<N_c} \alpha_i = 1$.

Les $\alpha_i$ contrôlent les poids relatifs accordés aux différentes vues. Par défaut ceux-ci valent tous le même poids, mais l'utilisateur peut décider de modifier ces poids. Les critères pour accorder plus ou moins de poids à une vue donnée sont multiples. L'idée fondamentale est de pouvoir accorder un poids plus important à une vue apportant plus d'information ou de discrimination sur l'objet. On peut citer par exemple :

- la résolution de la vue. Une image de 16 pixels est a priori moins porteuse d'information qu'une image de plus haute résolution ;
- le cadrage. Un objet occupant un espace maximum dans l'image est plus intéressant, le reste de l'image n'apportant rien. La difficulté de ce point est que le cadrage peut changer au cours du temps ;
- la visibilité de détails importants. Un être humain vu de dos, même bien cadré, n'apporte aucune information sur la position des bras, par exemple. Encore une fois cela peut changer au cours du temps quand l'objet se déplace.

Ces différents poids n'étant pas calculable a priori, ils sont laissés à l'appréciation de l'utilisateur.

Notons toutefois que le cardinal de $\mathcal{G}$ (l'ensemble des pixels utiles pour une vue) permet de connaître la surface de l'objet généré dans l'image, et surtout le rapport de cette surface avec la surface de l'image : $S = \frac{Card(\mathcal{G})}{L.H}$. Ce terme $S$ indique le taux d'occupation de l'image par l'objet, et peut servir de base pour déterminer les $\alpha_i$.

## 3.5 Prise en compte du domaine des paramètres

Dans de nombreux cas les paramètres ont un domaine de variation défini. Pour des paramètres de positionnement spatial, cela peut être les dimensions de la pièce. Pour des articulations, cela correspond au domaine de variation de celles-ci. Il peut bien entendu exister des paramètres non bornés, simultanément aux autres.

Nous gérons le domaine de validité des paramètres grâce à une technique dite de *pénalisation*. Il s'agit d'introduire dans la mesure d'écart un terme pénalisant les valeurs de paramètres en dehors de leur domaine.
La fonction de pénalisation $P$ s'applique à un jeu de paramètres, et retourne une valeur qui est nulle si les paramètres sont à l'intérieur de leur domaine de définition, et une valeur fortement croissante dès que la distance à la frontière du domaine est positive.

Si $\mathcal{D}$ est le domaine de validité des paramètres, on a donc pour un jeu de paramètres $p$ :

$$P : \Re^n \rightarrow \Re \tag{3.11}$$

$$P(p) = \begin{cases} 0 & \text{si } p \in \mathcal{D}, \\ g(d(p, \mathcal{D})) & \text{si } p \notin \mathcal{D} \end{cases} \tag{3.12}$$

avec $g$ une fonction strictement croissante et $d(p, \mathcal{D})$ une distance du vecteur de paramètre $p$ au domaine $\mathcal{D}$.

Nous n'utilisons pas une fonction de pénalisation "tout-ou-rien" afin de conserver autour de la frontière du domaine la continuité de la fonction d'écart. En pratique nous avons utilisé la distance quadratique entre le vecteur de paramètre et la frontière de $\mathcal{D}$ la plus proche. La distance entre la frontière et le vecteur considéré est aisée à déterminer dans notre cas puisque nos domaines sont construits par l'union des domaines de chaque paramètre :

$$\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_n \tag{3.13}$$

avec $\mathcal{D}_i$ le domaine du $i^{eme}$ paramètre.

La fonction d'écart finale $D$ s'écrit donc :

$$D(p) = D_m(p) + P(p) \tag{3.14}$$

avec toujours $p$ un vecteur de paramètres.

## 3.6 Validation expérimentale de la fonction d'écart

Nous étudions dans cette section le comportement expérimental de la fonction d'écart dans diverses situations. Le but est de montrer que celle-ci répond bien aux critères que nous nous sommes fixé. En particulier, les points à vérifier sont la présence d'un minimum de la fonction d'écart pour les valeurs théorique des paramètres, et la bonne tenue de cette fonction par rapport aux occultations.

### 3.6.1 Étude des variations de la fonction d'écart

Pour cela nous utilisons des vues de synthèse d'un objet comme images de référence. Dans notre exemple, nous avons utilisé trois vues d'un cube dont chaque face a une couleur différente (voir Fig. 3.2). Ce cube se trouve dans une pièce aux murs colorés. La phase de rendu comporte un sur-échantillonage ainsi qu'une addition de bruit blanc, afin de simuler les caractéristiques des caméras réelles.

Le modèle géométrique paramétré de l'objet décrit uniquement le cube lui-même et les couleurs de ses faces, et le bruit et le sur-échantillonage sont absents des images de synthèse générées. Les paramètres sont les six degrés de liberté du cube : position et orientation dans l'espace. Les trois vues de référence ont été générées avec des valeurs de paramètres que nous appellerons par la suite valeurs réelles ou valeurs théoriques.

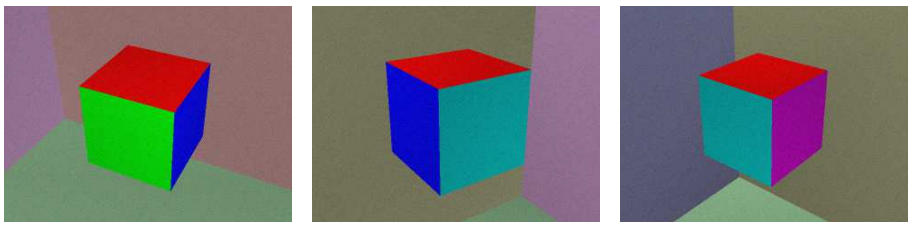


FIG. 3.2 – *Trois vues de synthèse d'un cube, avec sur-échantillonage et bruit blanc. Ces trois vues servent d'images de référence pour le calcul des valeurs de la fonction d'écart.*

Afin de pouvoir représenter graphiquement des variations de la fonction d'écart, nous avons choisi de faire varier deux paramètres à la fois. Ces deux paramètres et la valeur de la fonction d'écart correspondante définissent une surface tridimensionnelle $z = \mathcal{E}(p_1, p_2)$.

Dans l'exemple fig. 3.3 page suivante, deux paramètres de rotation du cube varient, ces variations allant de plus ou moins 45 degrés par rapport aux valeurs réelles. La figure montre en perspective la fonction $\mathcal{E}(\{r_x, r_y\})$, la fonction d'écart correspondante aux valeurs de paramètres $r_x$ et $r_y$, les autres paramètres étant considérés constants et de valeur exacte. Il est à noter que dans cet exemple la fonction d'écart n'utilise qu'une des vues, la première. Sur la partie droite de l'image se trouve la projection de $\mathcal{E}$ sur le plan des paramètres afin d'obtenir une image 2D. Dans chaque cas, et pour bien se représenter les variations des valeurs de la fonction d'écart, nous avons fait apparaître les lignes de niveaux de celle-ci. Chaque image de droite représente la projection sur le plan des paramètres de la fonction d'écart. Les lignes de niveaux permettent de suivre la forme de la surface, le minimum étant au centre de l'image.
On constate que le minimum de la fonction correspond aux valeurs réelles des paramètres (au centre à chaque fois), montrant sur cet exemple la validité de notre fonction d'écart. Les petites "vallées" présentes sur la surface sont dues à des variations des paramètres ayant peu d'impact sur la mesure d'écart, comme une translation le long d'un axe du cube.

On retrouve des résultats similaires dans la figure 3.4, qui représente la même fonction pour des variations de deux paramètres de translation, d'une amplitude de plus ou moins la moitié de la taille du cube.

La figure 3.5 représente quant à elle les variations d'un paramètre de rotation et d'un paramètre de translation, avec les même bornes que précédemment.

Dans les trois exemples précédents, la fonction d'écart n'utilise qu'une seule des vues de référence. La figure 3.6 montre les mêmes variations que pour le premier exemple, mais en utilisant les trois vues simultanément. On remarque que la fonction d'écart présente les mêmes caractéristiques, avec en plus une atténuation des effets de vallée. Ceci est dû à la compensation des effets d'axes de symétrie entre les vues, ces effets étant très dépendant de la vue considérée.

Pour terminer, la figure 3.8 montre les variations de rotation d'une boite de *Scotch*, filmée par deux caméras. Le modèle utilisé est un parallélépipède, auto-affiné et auto-

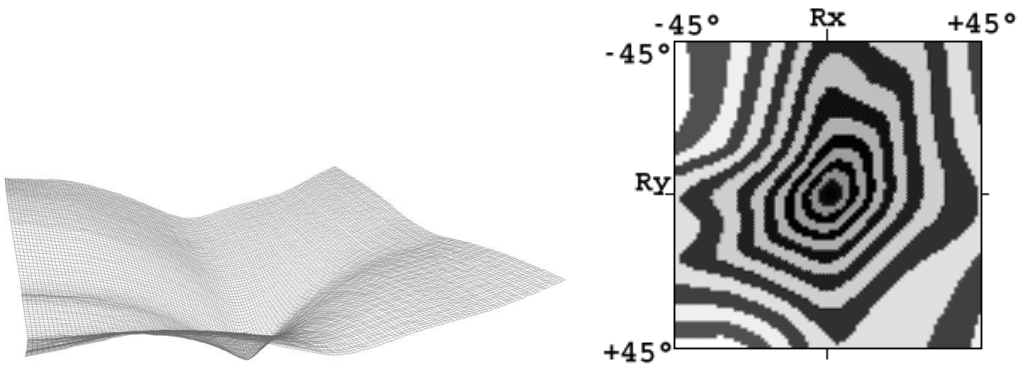



FIG. 3.3 – *Variations de la fonction d'écart en fonction des variations de deux paramètres de rotation. L'image de gauche montre une vue en perspective de la fonction* $z = \mathcal{E}(\{r_x, r_y\})$*, avec* $r_x$ *et* $r_y$ *les deux paramètres de rotation. La partie droite montre la carte des valeurs de la fonction, représentée par lignes de niveaux.*

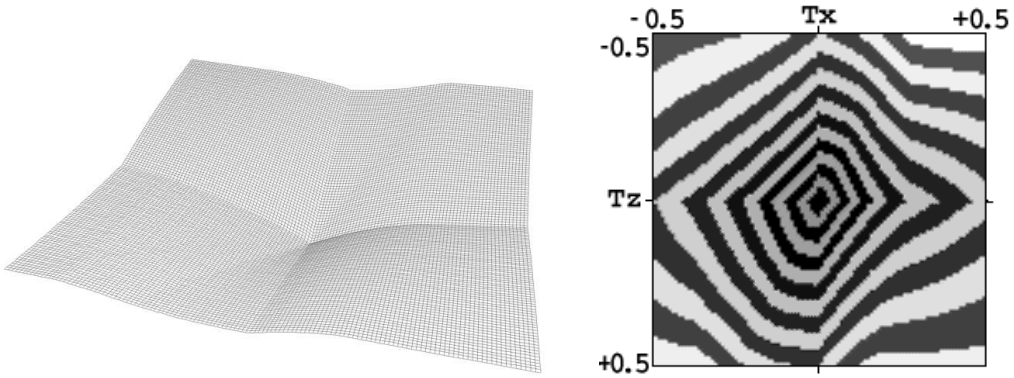



FIG. 3.4 – *Variations de la fonction d'écart en fonction des variations de deux paramètres de translation. L'image de gauche montre une vue en perspective de la fonction* $z = \mathcal{E}(\{t_y, t_z\})$*, avec* $t_y$ *et* $t_z$ *les deux paramètres de translation.*

texturé par la méthode décrite section IV. La figure 3.7 montre les deux vues de référence utilisées et une vue du modèle.
On constate sur cet exemple que la surface est bruité, phénomène relativement absent des exemples de synthèse. Ceci est dû aux erreurs de modélisation, d'éclairage dans la scène de synthèse et d'extraction des textures. Malgré cela, la surface reste descendante en direction du centre, qui correspond aux valeurs réelles des paramètres (mesurées de façon physique).

### 3.6.2 Résistance aux occultations

Nous reprenons dans cette section le même modus operandis que dans la section précédente, cette fois pour montrer que notre fonction d'écart est robuste aux occultations. Le même cube est utilisé, mais cette fois les vues de référence sont partiellement occultées

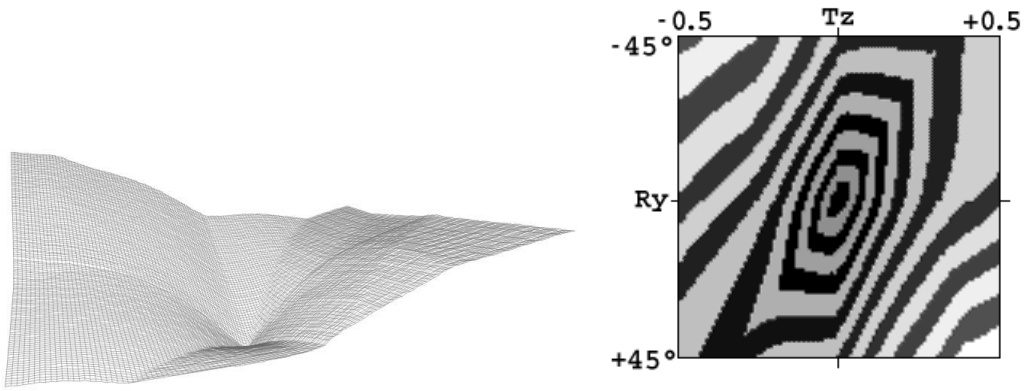



FIG. 3.5 – *Variations de la fonction d'écart en fonction des variations d'un paramètres de rotation et d'un paramètre de translation. L'image de gauche montre une vue en perspective de la fonction $z = \mathcal{E}(\{r_y, t_z\})$, avec $r_y$ et $t_z$ les deux paramètres respectivement de rotation et de translation.*

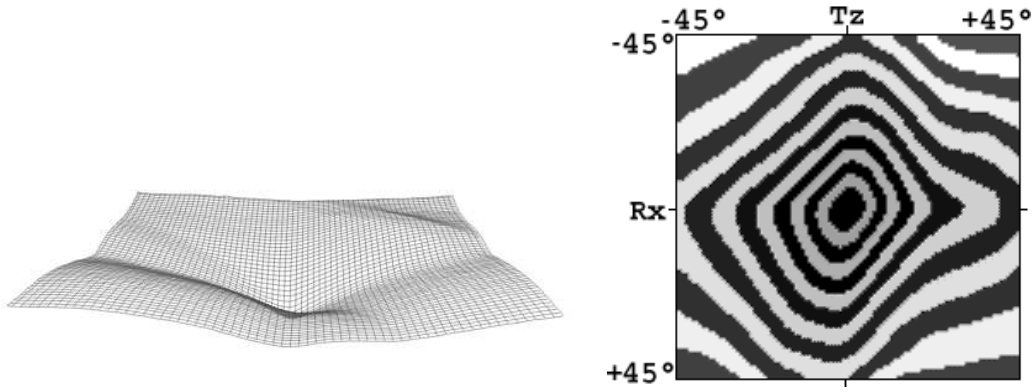



FIG. 3.6 – *Variations de la fonction d'écart en utilisant les trois vues de référence ensemble. L'image de gauche montre une vue en perspective de la fonction $z = \mathcal{E}(\{r_y, r_z\})$, avec $r_y$ et $r_z$ les deux paramètres de rotation. Dans cet exemple la mesure d'écart utilise les trois vues de référence du cube.*


FIG. 3.7 – *Les deux images de gauche montrent les vues de référence (réelles) d'une boite. Les deux images de droite montrent les deux vues de synthèse du modèle de cette boite, pour la même projection que les deux caméras réelles. La façon dont ce modèle a été obtenu est abordée plus loin.*

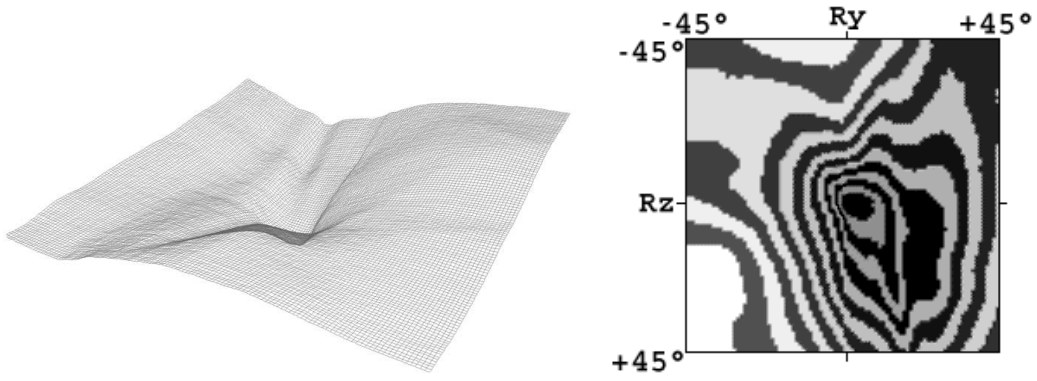



FIG. 3.8 – *Variations de la fonction d'écart sur un exemple réel : la boite de* Scotch. *L'image de gauche montre une vue en perspective de la fonction* $z = \mathcal{E}(\{r_y, r_z\})$, *avec* $r_y$ *et* $r_z$ *les deux paramètres de rotation utilisés. On constate que la fonction d'écart est plus bruité que pour les exemples de synthèse, mais que le minimum correspond aux valeurs théoriques.*

(figure 3.9). Les figure 3.10,3.11,3.12 montrent les variations de la fonction d'écart pour une rotation et une translation du cube, pour plusieurs degrés et types d'occultations. Comme précédemment, notre système n'a connaissance que du modèle du cube, et ignore tout des autres entités de la scène.

On constate la conservation du minimum de la fonction d'écart pour les valeurs théorique des paramètres, avec une diminution de la "netteté" de ce minimum quand les occultations augmentent. En pratique, on constate que les lignes de niveaux se ressèrent et qu'apparaissent sur les bords de l'images de nouveaux minima locaux. Ainsi c'est principalement le domaine dans lequel des paramètres initiaux permettent à coup sûr de trouver le minimum qui se réduit, rendant le suivi plus sensible aux décrochages. Au delà d'un certain niveau d'occultation – qui dépend de l'objet suivi et de l'occulteur, ce minimum n'est plus discernable et est noyé dans le bruit généré par l'occultation.

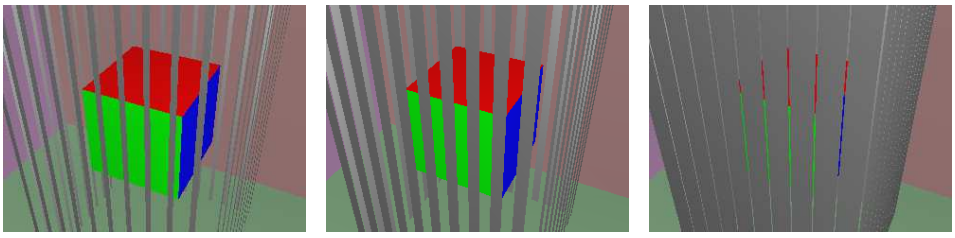


FIG. 3.9 – *Trois images de référence mettant en oeuvre plusieurs niveaux d'occultation. Ces images correspondent à la première des trois vues montrées plus haut.*

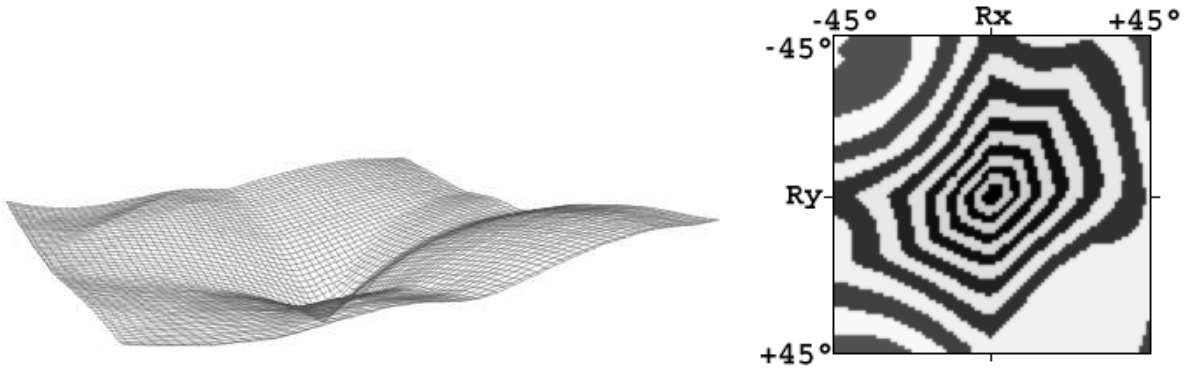



FIG. 3.10 – Variations de la fonction d'écart pour le cube, avec une occultation faible. On constate que le minimum de la fonction correspond aux valeurs théoriques.

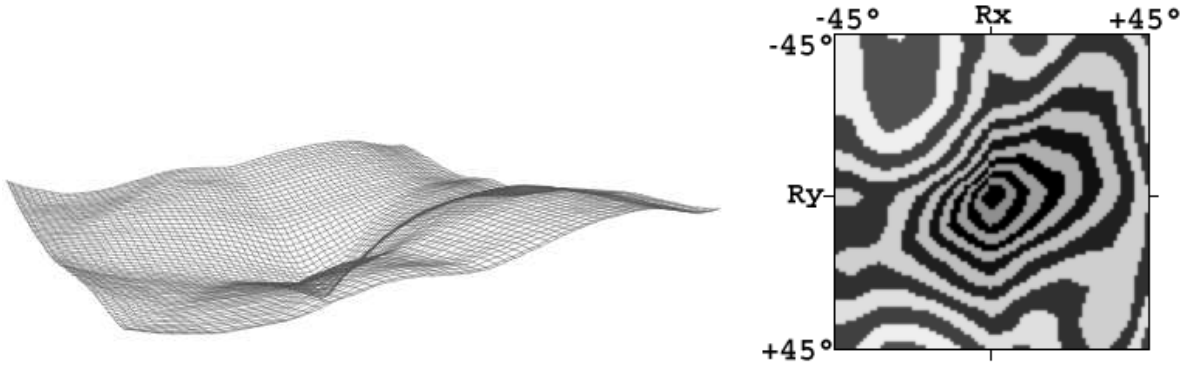



FIG. 3.11 – *Variations de la fonction d'écart pour le cube, avec une occultation forte. On constate que le minimum de la fonction correspond toujours aux valeurs théoriques, mais que la fonction est perturbée.*

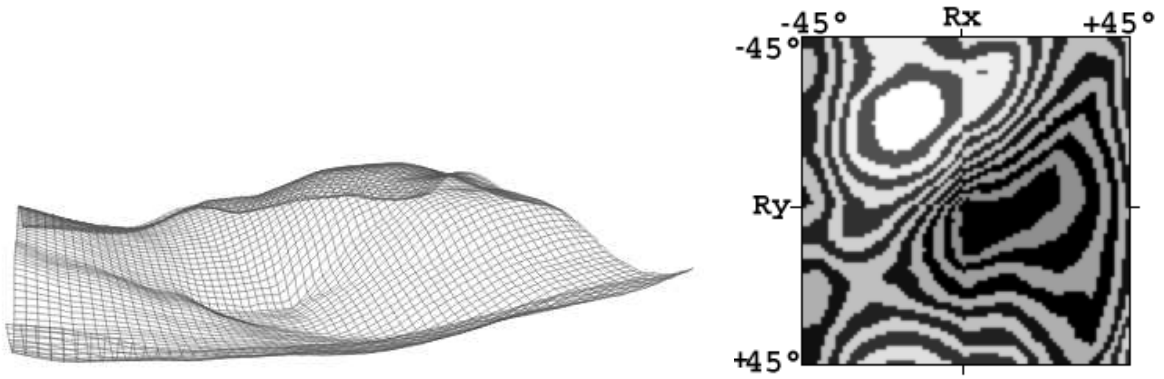



FIG. 3.12 – *Variations de la fonction d'écart pour le cube, avec une occultation extrême. On constate que la fonction devient très perturbée. De plus des minima locaux apparaissent, proches des valeurs théoriques. Cela réduit la zone où une recherche de minimum local mène effectivement à la solution désirée.*

# 4

# Recherche de l'optimum de la fonction d'écart

## Sommaire

Nous voulons positionner correctement le modèle de l'objet par rapport à l'objet réel. Ce sont donc les paramètres de ce modèle que l'on va chercher à trouver, la résolution s'effectuant dans l'espace des paramètres. Ce sont en effet les paramètres qui minimisent l'écart entre images réelles et images de synthèse qui correspondent, pour un jeu d'images vidéo donné, à la solution de notre problème.

Il existe de multiples algorithmes de minimisation dont une grande majorité sont exposés dans [PFTV92, Bis95, Fle87]. Le problème se pose sous la forme

$$\text{minimiser} \quad \mathcal{E}(p) \quad \text{avec} \quad p \in \Re^n \tag{4.1}$$

où $n$ correspond à la dimension de l'espace des paramètres et $\mathcal{E}$ est la fonction d'écart.

Dans le domaine de la minimisation de fonction, deux grandes stratégies existent. La première cherche à obtenir un minimum global de la fonction à optimiser, la seconde effectue une recherche locale, et conduit donc à des minima locaux. Du point de vue de la complexité, les recherches d'optimum global sont plus difficiles à réaliser. D'un autre coté, les recherches locales peuvent s'arrêter sur des minima locaux autres que celui recherché (lorsque la fonction à minimiser admet plusieurs minima locaux, ce qui est généralement le cas).

Dans notre cas, les méthodes de type recherche globale ne sont pas applicable, car bien que notre fonction d'écart soit conçue pour être décroissante à l'approche de la solution, d'autres minima peuvent exister "loin" de cette solution. Ces minima lointains peuvent être dus par exemple à des symétries de l'objet, principalement par rapport aux rotations. De plus l'espace des paramètres pouvant être très étendu, une recherche sur l'ensemble du domaine – à cause de l'optimisation globale – serait très coûteuse. Nous nous restreignons donc aux approches locales.

Deux grands types d'approche existent pour ces méthodes :

- avec dérivées (*derivative based*) : dans ce cas, la fonction est dérivable et sa formule est connue son gradient, et parfois même son Hessien. Ce sont des méthodes telles que NEWTON ou encore LEVENBERG-MARQUARDT ;
- sans dérivées (*derivative free*) : la fonction n'est pas forcement dérivable, mais calculable. On ne possède pas d'expression explicite de la fonction. Une estimation des dérivées est bien sûr possible, mais coûteuse.

Comme indiqué ci-dessus, nous ne possédons pas les dérivées de notre fonction d'écart. Nous nous plaçons donc dans le cadre des méthodes n'utilisant pas les dérivées.

Dans ce cadre, nous devons prendre en compte les spécificités de la fonction que nous cherchons à minimiser :

- pas d'expression explicite des dérivées de la fonction, comme nous l'avons vu ;
- l'évaluation de la fonction est d'un coût non négligeable, et sera (mais c'est toujours le cas) à réduire le plus possible ;
- le nombre de paramètres peut être élevé et donc l'espace de recherche étendu, comme c'est le cas pour des objets articulés complexes. Il faudra donc utiliser une méthode dont la complexité reste raisonnable lorsque le nombre de paramètres augmente ;
- la fonction d'écart est a priori bruité, pour diverses raisons telles que les erreurs de

modélisation ou encore le bruit dans les images vidéo, pouvant générer de petits minima locaux de faible amplitude.

Les méthode basées sur une exploration aléatoire ou semi-aléatoire de l'espace des paramètres telles que Monté-Carlo sont difficilement utilisables dans notre cas. Tout d'abord, pour un nombre élevé de paramètres l'espace à explorer est étendu, et la convergence est lente. De plus il est difficile avec ce type d'approche d'estimer les erreurs et de déterminer des critères d'arrêt objectifs.

Nous avons adapté à nos besoins une méthode robuste présentée par Nedler et Mead[NM65]. Cette méthode utilise le principe du simplexe, étendu au domaine continu, et porte dans la littérature le nom de ses concepteurs : la méthode *Nelder-Mead*.

## 4.1 Méthode d'optimisation : le simplexe généralisé

Il est connu que la méthode du simplexe d'applique à des fonctions convexes et affines par morceaux. La méthode introduite par NELDER et MEAD permet de se passer de l'hypothèse d'affinité par morceaux, et traitent de fonctions à valeurs de paramètres quelconques dans $\Re$.

La méthode d'optimisation Nelder-Mead est fondée sur l'utilisation d'un groupe de points – le simplexe – qui par ses caractéristiques va permettre la recherche du minimum de la fonction à optimiser.

Soit $f$ une fonction à $n$ variables dans $\Re$. Il s'agit de trouver les variables $v_1, v_2, ..., v_n$ telles que $f(v_1, v_2, ..., v_n)$ soit un minimum local. On pose $V_1, V_2, ..., V_n$ les valeurs de départ, utilisées pour initialiser la recherche de l'optimum.

On considère les $n+1$ points de $\Re^n$ $P_0, P_1, ..., P_n$, qui vont former le simplexe, et $\overline{C}$ le centre de gravité de ce groupe de points :

$$\overline{C} = \frac{\sum_{i=0}^{i\leq n} P_i}{n+1} \tag{4.2}$$

On définit également les point $h$ et $l$ tels que :

$$f(h) = \max_{i=0}^{i\leq n} f(P_i) \tag{4.3}$$

$$f(l) = \min_{i=0}^{i\leq n} f(P_i) \tag{4.4}$$

respectivement le point ayant la plus forte et la plus faible valeur par rapport à la fonction à minimiser.

La recherche du minimum se fait au travers d'un ensemble de transformations du groupe de points. À chaque étape le point $h$ est remplacé par un nouveau point, en utilisant divers opérateurs qui vont transformer le groupe de points. Ces transformations sont : une réflexion, une contraction et une dilatation. L'opération de réflexion permet de remplacer le point $h$ le moins intéressant par un nouveau point, choisi dans direction opposée au point le moins intéressant. L'opération de dilatation va permettre d'étendre le groupe de

point dans la direction de déplacement actuelle, accélérant le déplacement global de celui-ci. La dernière opération, la contraction, permet au simplexe de contrecarrer les effets de la dilatation lorsque celle-ci n'est plus adaptée, c'est-à-dire lorsqu'un changement de direction s'impose. La figure 4.1 montre un cas simple (une seule variable) de déplacement du simplexe.

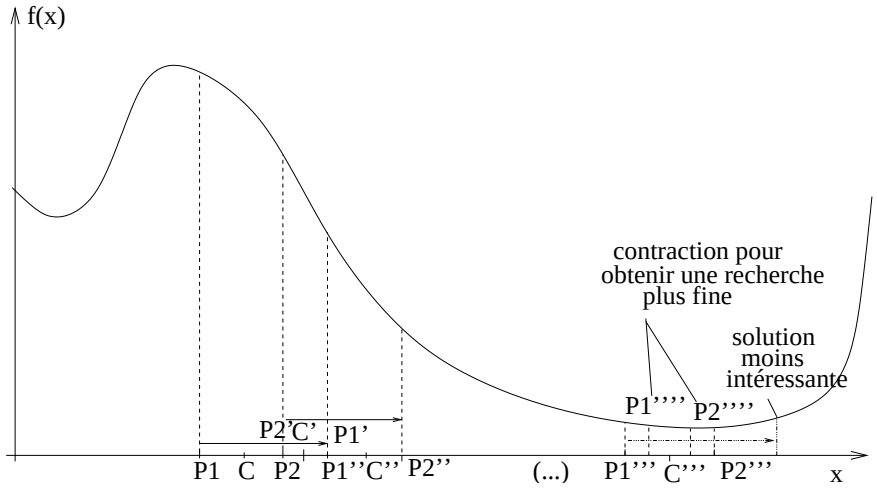



FIG. 4.1 – *Évolutions du groupe de points lors de l'optimisation sur un exemple simple à une variable. À chaque étape l'algorithme remplace le point le moins intéressant par le symétrique de celui-ci par rapport au centre de gravité $C$ puis en appliquant une dilatation dans cette direction. Lorsque le nouveau point n'est pas intéressant on opère une contraction afin de permettre une recherche plus fine de la solution.*

Chacune de ces transformations dépend d'un paramètre qui va contrôler son amplitude. En pratique, les nouveaux points sont calculé à partir de $h$, $l$ et du centre de gravité $\overline{C}$ comme suit :

- pour une réflexion : $h' = (1+\alpha)\overline{C} - \alpha h$. $\alpha$ contrôle la distance de réflexion par rapport au centre de gravité ;
- pour une dilatation : $h'' = \gamma h' + (1-\gamma)\overline{C}$, avec $h'$ le point calculé lors de la phase de réflexion. Ici $\gamma$ contrôle la quantité d'élongation à appliquer, avec $\gamma$ supérieur à 1 (on sort ainsi de l'enveloppe du groupe de point, but de l'élongation) ;
- pour une contraction : tout point $p$ différent de $l$ est remplacé par $p'$ tel que $p' = (p+l)/2$.

La figure 4.2 montre un exemple de chacune de ces transformations, pour une fonction à deux variables.

Il existe de plus une autre transformation de contraction. Celle-ci est appliquée lorsque la phase de réflexion donne un résultat plus mauvais que tous les points du simplexe (y compris $h$). Elle consiste à calculer $p''$ tel que $p'' = \beta h + (1-\beta)\overline{C}$, ce qui revient à faire une contraction selon l'axe de déplacement actuel du simplexe. Si cette phase échoue l'algorithme utilise alors la contraction globale définie plus haut.

Il est à noter qu'à partir des points trouvés par l'algorithme il est possible de calculer certaines informations utiles. Cela s'obtient grâce à une ré-expression des points du

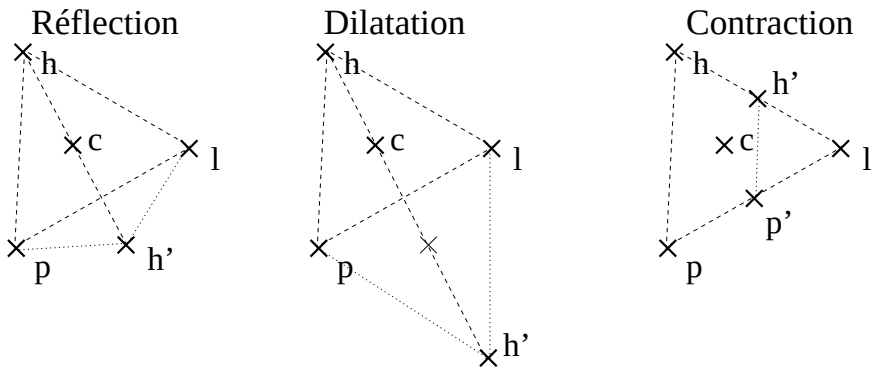



FIG. 4.2 – *Évolutions possibles du groupe de points du simplexe par rapport à la fonction à minimiser. C est le centre de gravité du groupe de points, et h et l respectivement le moins bon et le meilleur point du groupe. Les points suffixés par ′ sont les nouveaux points créés.*

simplexe, permettant d'obtenir une approximation quadratique de la surface autour du centre de gravité. À partir de cette surface quadratique un certain nombre d'opérations sont possibles, en particulier le calcul du minimum de cette surface quadratique et l'estimation de la matrice des variances-covariances de la fonction $f$ autour du point solution. Le lecteur se référera à l'article [SHH62] plus plus d'information sur ces derniers points.

Le lecteur intéressé par plus de détails sur cette méthode pourra également se référer l'article original des auteurs[NM65] ou bien aux articles plus récents proposant des critiques et des améliorations de celle-ci [LRWW98, McK98].

## 4.2 Accélération de la convergence

Nous avons adjoint à cette méthode quelques améliorations, afin de prendre en compte la spécificité de notre problème et d'accélérer la convergence.

En pratique il apparaît des situations où le groupe de points qui constitue le simplexe se contracte de façon excessive. Cela se produit normalement à l'approche de la solution, lorsque le groupe de points réduit sa taille pour affiner les résultats. Ce type de situation survient également lorsqu'une contraction se produit autour d'un point relativement éloigné de la solution. La distance restante doit alors être parcourue avec un simplexe de petite taille. Cela a pour conséquence de ralentir la vitesse de déplacement du simplexe – la vitesse étant proportionnelle à la "taille" du groupe de points et donc soit de diminuer la précision, soit d'augmenter le nombre d'étapes nécessaires pour atteindre une précision donnée.

Pour dynamiser le groupe de points, nous avons utilisé un principe que l'on retrouve dans les algorithmes génétiques[Gol94] ou le recuit simulé : l'introduction de perturbations aléatoires dans un ensemble stable.

### 4.2.1 Méthode des perturbations

L'idée – du point de vue des algorithmes génétiques – est de considérer le groupe de points comme une population d'individus, de considérer la fonction $f$ comme leur fonction objectif (*fitness value*), et de leurs appliquer, dans certaines conditions, des mutations, c'est-à-dire une perturbation aléatoire.
Dans notre cas, ces mutations se déclenchent lorsqu'une certaine stabilité est détectée dans le groupe de points, c'est-à-dire lorsque la meilleur solution $l$ se modifie très peu – en dessous du seuil de précision – pendant plusieurs étapes.
Ces mutations peuvent prendre des formes variables. Celles que nous avons choisi modifient les "mauvais" points, c'est-à-dire ceux ayant les valeurs par rapport à la fonction d'écart les plus grandes (puisque dans notre cas nous cherchons le minimum de cette fonction). Ces mutations prennent la forme de perturbations aléatoire d'une ou plusieurs valeurs de paramètres de ce point. Ces perturbations sont contrôlées par des probabilités d'apparition, le choix de ces taux dépendant de la vigueur des changements que l'on souhaite créer. Un taux faible ne change que très peu le fonctionnement global, alors que des taux très élevés risquent de "brouiller" la structure qu'a pris le simplexe.

Dans tous les cas, les mutations s'arrêtent dès qu'une nouvelle solution intéressante est trouvée, et le système reprend son fonctionnement normal.

Les effets de ces perturbations sont principalement une dynamisation du groupe de point, lui permettant de reprendre une convergence plus rapide, ainsi que la possibilité de s'extraire de certains minima locaux. En effet la perturbation aléatoire peut projeter un point à l'extérieur d'un minimum local, ce point entraînant ensuite les autres à l'extérieur. De même ces perturbations tendent à accroître la taille du simplexe, lui permettant de reprendre une progression plus rapide par la suite.

### 4.2.2 Exemple

La figure 4.3 montre la comparaison entre l'algorithme sans et avec mutations, dans les mêmes conditions initiales. Le début des deux courbes est confondu, tant que l'algorithme classique progresse. Puis, au premier ralentissement de la méthode, les mutations permettent un bond dans une direction intéressante, accélérant sensiblement la vitesse de convergence par rapport à la méthode classique. Bien entendu le gain de vitesse dépend en partie de la "chance" de trouver rapidement une mutation intéressante, mais reste statistiquement intéressante.

### 4.2.3 Estimation de la complexité

Un autre point intéressant est la complexité moyenne de cette approche. Dans l'article original[NM65], les auteurs donnent une estimation empirique de la complexité $N$ de l'algorithme (en nombre d'étapes) en fonction du nombre de paramètres $k$ sous la forme :

$$N = 3.16(k+1)^{2.11} \tag{4.5}$$

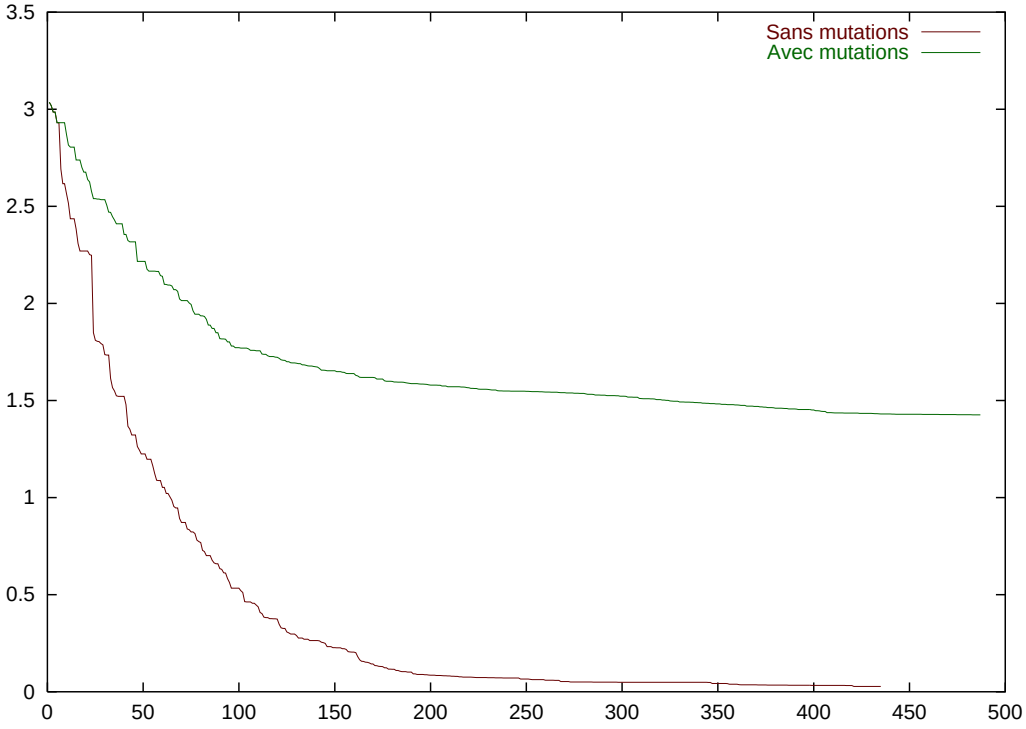



FIG. 4.3 – *Comparaison de la convergence avec et sans mutations. En abscisses se trouve le nombre d'étapes prises par l'algorithme, et en ordonnées la précision de la meilleur solution correspondante. On constate que pour un nombre d'étapes équivalent l'utilisation des mutations donne rapidement des résultats nettements meilleurs.*

Cette estimation a été calculée pour un nombre de paramètres variant entre 2 et 10 (sans doute pour des questions de puissance de calcul à l'époque). Cela revient grossièrement à une complexité de l'ordre de $o(n) = n^2$. De plus cette étude de la complexité a été réalisée sur une seule fonction simple :

$$f(x_1, x_2, ..., x_k) = x_1^2 + x_2^2 + ... + x_k^2 \tag{4.6}$$

Nous avons cherché à étendre cette estimation à la fois au niveau du nombre de paramètres et au niveau des fonctions utilisées. Nous avons donc réalisé un ensemble de tests basés non pas sur des fonctions données par leur formule mais directement sur notre fonction d'écart, en utilisant comme images de test les vues de synthèse du cube données en exemple dans les sections précédentes. Nous avons artificiellement augmenté le nombre de paramètres en multipliant le nombre de cubes dans une même scène, générant ainsi à chaque fois six paramètres supplémentaires.

Nous avons tracé le graphe d'évolution du nombre moyen d'étapes de l'algorithme du simplexe en fonction du nombre de paramètres (voir figure 4.4). On constate sur ce graphe que la complexité effective – sur ce type de scène – est en fait bien inférieure à une courbe de degré deux, comme le laissait supposé l'article original. Nous avons expérimentalement déduit de cette étude que la complexité est de l'ordre de $ln(n)$. Nous pensons que la grande différence entre cette estimation et l'estimation donnée par les auteurs peut être expliquée par plusieurs raisons :

- la faible largeur de l'échantillon (2 à 10) utilisé par les auteurs induit un fort biais sur la validité de l'estimation de la complexité ;
- l'utilisation d'une fonction symétrique de type paraboloïde augmente, pour des méthodes n'utilisant pas les dérivées explicites des fonctions, les risques d'approche "en spirale" de la solution, forcement plus longues. *A contrario* dans notre exemple des vallées prononcées canalisent la descente du simplexe et prévienne ce types d'incidents, tout au moins pour le type de scène utilisé ;
- l'utilisation des mutations pour l'accélération de la recherche réduit grandement la survenue d'itérations inutiles dues à la trop grande contraction du simplexe.

En résumé, la complexité constaté de cet algorithme est très intéressante car bien adaptée aux objets complexes présentant de nombreux paramètres. Il ne faut cependant pas confondre complexité en nombre d'itérations et complexité globale de l'algorithme. La complexité globale doit tenir compte du fait qu'à chaque étape il y a $n + 1$ points à traiter, pour la recherche du minimum et du maximum par exemple. Comme de plus chaque point est lui-même composé de $n$ valeurs, la complexité *globale* de l'algorithme est : $o(n) = n^2.ln(n)$.

## 4.3 Effets de la pénalisation sur la recherche de l'optimum

Comme cela a été décrit dans la section 3.5, nous gérons les contraintes sur le domaine de validité des paramètres en utilisant une technique de pénalisation.

Le principe est d'introduire dans le calcul de la mesure d'écart un terme dont la valeur est nulle si les paramètres sont à l'intérieur du domaine de validité, et strictement

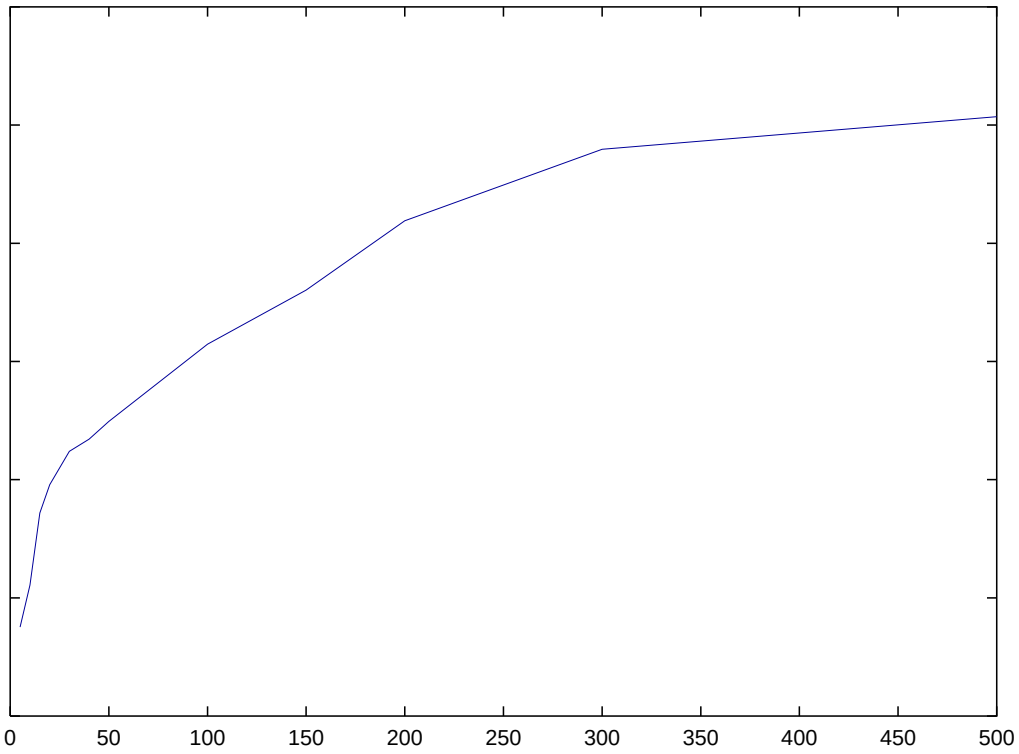



FIG. 4.4 – *Évolution du nombre moyen d'étapes de l'algorithme du simplexe (en ordonnées) en fonction du nombre de paramètres (en abscisses).*

croissante par rapport à la distance à la frontière de ce domaine lorsque les paramètres sont à l'extérieur.

La pénalisation est souvent traitée par une fonction seuil, prenant une valeur très grande dès que les paramètres sortent du domaine.

Dans notre cas nous sommes tenu d'utiliser une fonction croissante avec la distance au domaine, pour le bon fonctionnement de l'algorithme du simplexe.

En effet celui-ci utilise un groupe de points comme interpolant local de la fonction à optimiser. L'utilisation d'une pénalisation à seuil rendrait impossible l'existence de points à l'extérieur du domaine. La taille du groupe de points n'étant pas nul, cela rendrait impossible – ou en tout cas difficile – la recherche efficace de solutions en bordure du domaine.

Avec une fonction de pénalisation croissante, cela confère une certaine souplesse au système, permettant au groupe de points d'empiéter partiellement sur l'extérieur du domaine de validité, et permettant du même coup la recherche de solutions proches de la frontière.

Le risque d'obtenir une solution en dehors du domaine est éliminé en ne classant jamais comme "meilleur point" un point en dehors du domaine (revoir la section 4.1 pour le fonctionnement précis de l'algorithme du simplexe).

## 4.4 Contraintes et dépendances des paramètres

Il existe des facteurs importants qui influent sur la vitesse de convergence dans les problèmes d'optimisation. Parmi ceux-ci, on trouve les problèmes de dépendances et de poids relatifs des paramètres entre eux, par rapport à la fonction à optimiser.

L'optimisation simultanée de paramètres ou de groupes de paramètres indépendants entre eux peut engendrer des effets d'oscillation entre des valeurs de paramètres distincts mais se compensant mutuellement. Par exemple si l'on cherche à optimiser la fonction $f(x, y) = |x| + |y|$, on constate que des iso-lignes de valeurs partagent les différents secteurs en diagonales. Des cheminements en spirale sont alors possibles.

Cet effet est bien évidement limité par le fait que la ligne de plus grande pente est dirigée dans un cas simple comme celui-ci vers le minimum $((0, 0))$, mais le Simplex n'étant qu'une approximation de cette ligne de plus grande pente, des déviations sont tout de même possibles, et ceci d'autant plus facilement que la surface de la fonction à optimiser est bruitée.

Le problème des poids relatif des paramètres a aussi son importance. Imaginons un bras robotisé. Il est intuitivement évident qu'il n'est pas utile de chercher l'angle que forment les pinces par rapport à leur support tant que les positions exacte de l'épaule et du coude ne sont pas trouvées. Il s'agit là de prendre en compte l'impact qu'un paramètre a sur l'objet, et plus généralement sur la fonction à optimiser.

Dans le cadre de notre approche, ces problèmes n'ont que peu ou pas d'impact.

Le premier point – bien qu'important en général – ne concerne que très peu les problèmes sur lesquels nous travaillons. En effet des paramètres réellement indépendants correspondent à des objets distincts, évoluant dans la même scène. Ce genre de situation n'est pas courante dans notre cas, ou nous nous intéressons plutôt au suivi d'un objet unique dans une scène. De plus le suivi de deux objets distincts peut facilement se reformuler comme étant deux problèmes de suivi, chacun portant sur un seul objet.

Le deuxième point n'a également que peu d'impact pour notre approche. Pour tenir compte des poids relatifs des paramètres, il faudrait mettre en place une optimisation "par étape", en commençant les traitements sur les paramètres les plus influents et en n'introduisant les autres que lorsque les premiers approchent de la solution. Or la méthode du simplex utilisée est peu sensible à l'augmentation du nombre de paramètres, et donc le fait de "découper" le problème n'apporte que peu ou pas d'avantages par rapport à une optimisation globale, alors que la multiplication des phases d'optimisation a quand à elle un coup non négligeable.

Toutes ces raisons font que, bien que conscient des techniques existante d'accélération de la convergence, le cadre particulier de notre application nous évite généralement de nous trouver dans des situations où ce genre de problèmes se pose, et lorsqu'ils se posent y remédier n'apporterait que peu ou pas d'accélération des traitements.

# 5

# Extension des traitements dans le temps

## Sommaire

L'approche décrite dans les sections précédentes permet l'obtention des valeurs des paramètres d'un modèle géométrique, ceux-ci correspondants aux caractéristiques 3D de l'objet réel représenté par le modèle. On possède pour cela un vecteur de valeurs initiales pour les paramètres, ainsi qu'un ensemble de vues de l'objet à un instant donné, ces valeurs étant raffinées jusqu'à concordance optimale entre l'objet réel et les projections 2D du modèle paramétré – via la mesure d'écart, bien sûr.

Il nous faut maintenant étendre cela à l'ensemble des séquences vidéo, c'est-à-dire intégrer le facteur temps et obtenir non plus un vecteur de valeurs pour les paramètres mais l'ensemble des variations de ces valeurs au cours du temps.

Le résultat désiré est donc l'obtention des valeurs des paramètres $p_t^i$, pour tout paramètre $p^i$ et pour l'ensemble des instants $t$ présents dans les séquences vidéo.

C'est ce que nous présentons dans la section 5.1, où nous développons l'approche utilisée pour effectuer le suivi proprement dit. La section 5.2 présente une méthode permettant d'améliorer la précision des transitions entre les instants successifs, aussi appelées *étapes* dans ce contexte. Nous abordons ensuite dans la section 5.3 le problème de la détection d'éventuelles erreurs de suivi : le *décrochage*. Enfin quelques exemples simples de suivi sont présentés dans la section 5.4. Des exemples plus complets sont présentés dans la dernière partie Résultats (partie V).

## 5.1 Suivi des paramètres dans le temps

Afin de réaliser le suivi des paramètres, il faut étendre les traitements précédemment décrits dans le temps, pour obtenir les valeurs des paramètres non plus pour un instant donné mais pour l'ensemble des instants composant les séquences vidéo.

Comme nous nous plaçons dans une problématique de suivi de mouvements, cela implique une cohérence temporelle minimum dans les images. Cela veut dire qu'entre deux instants $t$ et $t + dt$ successifs l'objet suivi se déplace relativement peu dans les images correspondantes.
Cette notion de "relativement peu" est difficile à traduire en termes quantifiables – nombre de pixels ou autre. Dans la partie présentant une validation expérimentale de la fonction d'écart entre images, on voit clairement que cette fonction est quasi-convexe pour des rotations de l'objet de plus ou moins 45 degrés et des translations de plus ou moins la moitié de la taille de l'objet (section 3.6). Cette large tolérance est due en partie à la forme simple de l'objet utilisé (un parallélépipède texturé). Des objets présentant des disparités et/ou des symétries plus nombreuses auront de façon assez évidente des domaines de convexité plus restreint. Un cas extrême (et fort heureusement sans grand intérêt pratique) est la sphère non texturé, pour laquelle toute rotation par rapport à son centre laisse ses images inchangées.

D'une façon générale, il est tout de même possible de dégager plusieurs règles empiriques pour décider si un mouvement est "faible" :

- plus l'objet – ou une partie de cet objet – est petit par rapport à la surface de l'image, plus son mouvement apparent doit être faible ;
- les rotations des objets présentant des symétries en rotation doivent être faible par

rapport à l'angle entre les symétries ;
- plus un objet – ou une partie de cet objet – est contrasté par rapport au reste de l'image (c'est-à-dire qu'il ne peut pas être confondu avec d'autres parties de l'image à cause de sa couleur, de ses textures...) plus un déplacement apparent élevé est supporté, l'objet étant un "attracteur" très fort.

D'un façon générale ces critères restent du ressort du jugement humain. Ce sont du reste des critères très répandus, de manière implicite. Lors d'épreuves sportives, par exemple, les journalistes utilisent des caméras à haute fréquence pour pouvoir suivre de façon nette les actions des sportifs. De même lorsqu'un crash-test est filmé les vitesses relatives des objets sont très élevées au moment de l'impact et des caméras allant jusqu'à mille images/seconde peuvent être utilisées.

Ainsi il apparaît que dans ce type de séquences vidéo la position de l'objet suivi à un instant $t$ est relativement proche de la position de ce même objet à l'instant $t+1$. Par extension cela veut dire que les valeurs des paramètres à l'instant $t$ sont proches des valeurs des paramètres à l'instant $t+1$, que nous cherchons.

Ces valeurs trouvées à l'instant précédent vont donc servir de point de départ pour la recherche des valeurs à l'instant courant (voir Fig 5.1).

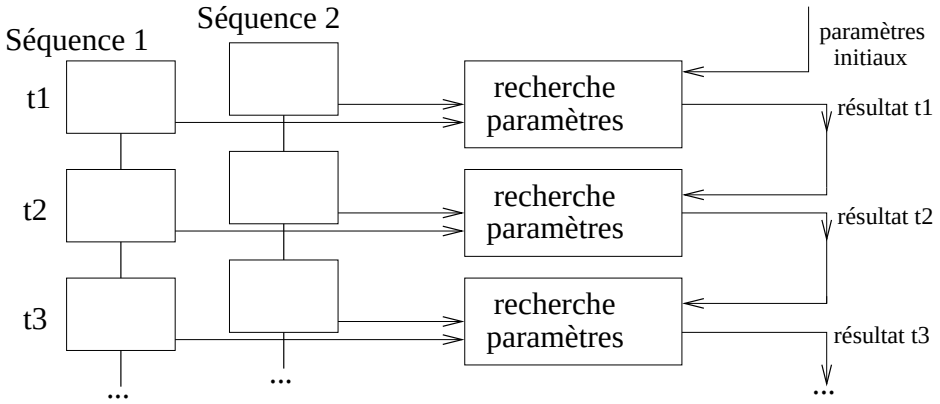



FIG. 5.1 – Réutilisation des valeurs de paramètres trouvés à l'étape précédente comme valeurs initiales pour l'étape courante. De part l'hypothèse de faibles variations entre deux instants, les valeurs correspondant à un instant $t$ sont proches des valeurs correspondant à l'instant $t+1$.

Le suivi temporel des paramètres s'effectue ainsi en propageant d'instant en instant les valeurs trouvées aux étapes précédentes comme valeurs initiales de l'étape courante, en utilisant la cohérence temporelle des mouvements de l'objet suivi, qui permettent de supposer une variation des valeurs faible entre deux instants successifs.

## 5.2 Prédictions sur l'évolution des paramètres

L'utilisation des valeurs des paramètres à l'étape précédente comme valeurs initiales de l'étape courante est bien sûr une approximation. La précision de cette approximation dépend du déplacement de l'objet entre les deux instants considérés. Si l'objet est immobile

entre ces deux instants il ne s'agit plus d'une approximation : les valeurs sont exactes. Par contre plus l'objet s'est déplacé au cours de ce laps de temps et plus l'erreur commise sera grande.

En pratique tant que cette erreur laisse le point de départ de la recherche des paramètres à "l'intérieur" de la zone convexe entourant la solution, la convergence sera assurée. Cette contrainte est assurée à la fois par la tolérance de la mesure d'écart et par les faibles variations entre images successives imposé par le fait que l'on filme du mouvement.

Malgré tout, la recherche des valeurs des paramètres reposant sur un algorithme d'optimisation, il est évident que plus les valeurs initiales sont proches de la solution et plus la convergence sera rapide et/ou précise.

Nous avons introduit entre les étapes de recherche des paramètres une phase de prédiction, afin d'augmenter la précision des valeurs initiales de chaque étape (Fig. 5.2). L'idée clé de cette prédiction est qu'une connaissance – extérieur à notre système – peut influencer la façon dont évoluent les paramètres. Ces connaissances dépendent bien évidement de la nature de l'objet suivi ainsi que des conditions dans lesquelles il évolue.

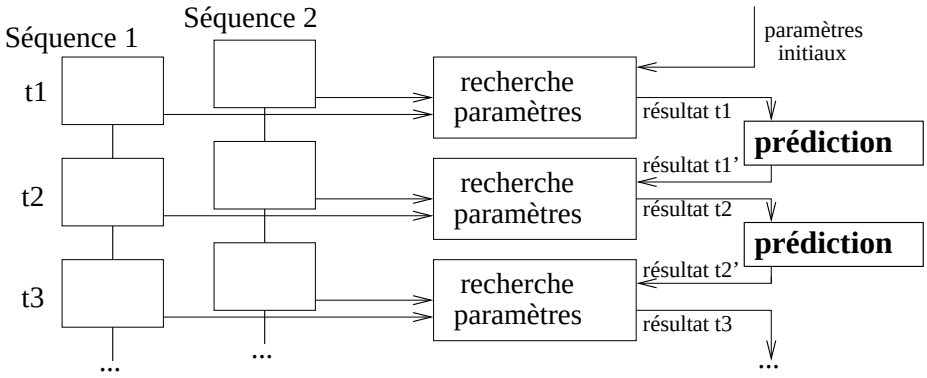



FIG. 5.2 – *Ajout d'une phase de prédiction entre les étapes successives de recherche des paramètres, afin d'augmenter la précision. Grâce à cela, la phase de recherche des paramètres à l'instant $t+1$ commence avec des valeurs initiales plus proches des valeurs recherchées, réduisant les risques d'erreur et le temps nécessaire pour l'optimisation.*

Une méthode de prédiction générique, pour des objets soumis aux lois de la physique, est par exemple la conservation de la quantité de mouvement. Cette méthode est difficilement applicable dans la mesure ou elle requière des informations physique (masse, inertie...) sur l'objet étudié, ces informations dépassant de loin ce que contient le modèle géométrique.

Une autre méthode s'appliquant bien aux objets soumis aux lois de la physique est la conservation du vecteur vitesse ou encore du vecteur accélération. À partir de l'historique des valeurs des paramètres il est possible d'interpoler – au premier ou au deuxième degré – les valeurs des paramètres pour l'instant suivant.

Les autres possibilités de prédiction dépendent fortement du contexte. Dans un contexte industriel (bras articulé, déplacement de pièces...) un grand nombre de connaissances sur les déplacements possibles et/ou probables existent et peuvent être utilisées. Pour le cas d'êtres humains, le type d'activité (sportive, rééducation physique...) conditionne fortement le type de mouvements plausibles.

Il faut cependant tenir compte du fait qu'une prédiction reste une prédiction, et est donc faillible. Il ne faut pas qu'une prédiction erronée amène le point de départ en dehors de la zone de convexité de la mesure d'écart. Ceci peut arriver par exemple dans le cas d'une prédiction basée sur la continuité de la vitesse, alors que l'objet rebondit contre un obstacle. Dans ce type de cas, le point prédit est deux fois plus éloigné de la solution que le point sans prédiction.

Pour cela nous utilisons le fait que l'algorithme d'optimisation utilise un groupe de points pour la recherche de l'optimum. Nous contraignons le premier des points à prendre la valeur des paramètres de l'étape précédente (sans prédiction), et le deuxième à prendre les valeurs prédites. Les autres points sont ensuite choisis selon la méthode habituelle. Ceci nous assure que si le point prédit est moins intéressant que le point sans prédiction, ce dernier sera tout de même présent dans le simplex et permettra une optimisation correcte.

La section 5.4 montre quelques comparaisons de performance et de précision lorsque la prédiction est utilisée.

## 5.3 Détection des erreurs de suivi

Malgré toutes ces précautions, un *décrochage* du suivi peut survenir. Nous appelons décrochage la perte du suivi, c'est-à-dire le fait qu'à partir d'un instant donné les paramètres ne représentent plus le comportement de l'objet suivi.

Ce phénomène peut se produire principalement dans deux situations :

- un déplacement trop grand se produit entre deux instants successifs. Le point de départ de la recherche des paramètres à l'étape suivante (avec ou sans prédiction) peut alors se trouver dans une autre *vallée* que celle menant à la solution ;
- l'objet est occulté de façon importante par un objet, faisant perdre la cohérence de la mesure d'écart. Il est à noter que l'utilisation de plusieurs vues de l'objet rend moins probable ce type de problème, l'objet restant a priori visible dans les autres vues.

Est-il possible de détecter ces décrochages ? Cela est difficile, car la solution trouvée reste toujours un minimum (local). De plus la seule mesure dont nous disposons est justement celle utilisée lors de la phase d'optimisation (voir figure 5.3).

Il existe cependant quelques pistes permettant de déceler ce type d'évenements. En premier lieu la solution trouvée est certes un minimum local, mais sa valeur – la mesure d'écart en ce point – est généralement supérieure à la valeur qu'elle avait aux étapes précédentes. Ainsi lorsque la solution à l'étape précédente est correcte on constante expérimentalement une brusque augmentation de la valeur de la mesure d'écart pour l'étape courante. La valeur de cette augmentation dépend hélas fortement de la scène traitée et est peu aisé à déterminer.

Une solution consiste à détecter ces brusques augmentations, et à effectuer un retour en arrière afin de refaire la phase d'optimisation avec des critères différents. La difficulté restant que l'on ne connaît pas la raison de ce décrochage, et qu'il est donc difficile de corriger a priori les critères de l'optimisation correctement.

Il existe de plus des situations sans décrochage pouvant générer une brusque augmentation de la valeur de la mesure d'écart. La principale est un changement d'illumination

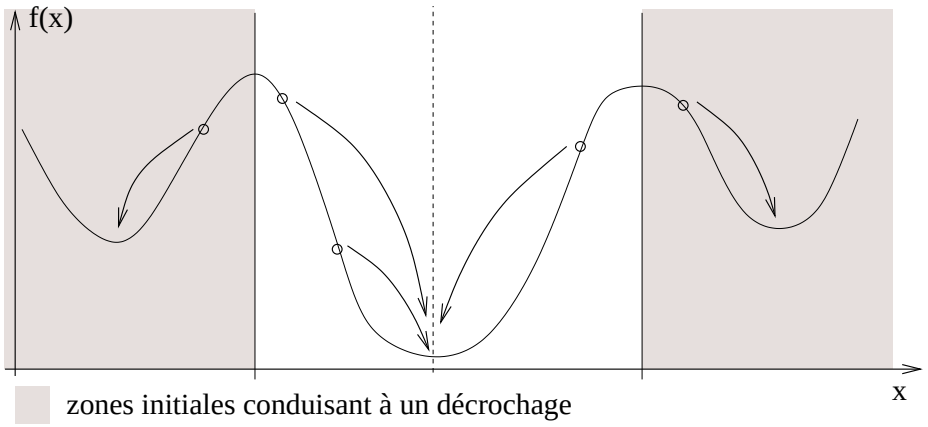



FIG. 5.3 – *Cause possible de décrochage : si le point de départ est à l'extérieur de la zone convexe autour de la solution le système part dans une autre vallée.*

dans la scène réelle, par exemple une lampe s'allumant dans la scène, sans que ce changement soit modélisé. Ce type d'évenement a pour effet d'augmenter globalement l'écart entre les images réelles et les images de synthèse. Concrètement, cela correspond à une translation de la surface définie par la mesure d'écart vers le "haut" (une augmentation en tout point de la valeur de la mesure d'écart) et ne change pas fondamentalement le comportement de l'algorithme d'optimisation. De plus la mesure d'écart étant relativement robuste aux erreurs de luminances, le minimum local de cette mesure reste correct, mais la valeur associée à ce minimum augmente.

D'une manière générale, nous ne disposons à l'heure actuelle que d'heuristiques pour détecter les décrochages. Ces heuristiques fournissent lors du suivi des suggestions sur des instants des séquences ayant pu poser problème, l'opérateur ayant la charge de vérifier si le problème était réel ou non.

## 5.4 Quelques exemples de suivi dans le temps

Nous présentons dans cette section quelques exemples simples de suivi de paramètres. Ces exemples de synthèse montrent principalement l'influence de la prédiction sur la durée des traitements ainsi que la détection des décrochages.

### 5.4.1 Sphère en rebond

Cette animation de synthèse est composée de deux vues d'une sphère rebondissant sur un sol. Les caractéristiques sont :

- images en $320 \times 240$ sans anti-crénelage ;
- 50 images par séquence, avec deux caméras virtuelles filmant la scène ;
- les paramètres sont les trois coordonnées de positionnement spatial.

La figure 5.4 page suivante montre quelques images issues des deux vues à des instants différents.

La figure 5.5 montre le nombre d'étapes nécessaires à l'algorithme d'optimisation pour chaque instant des séquences, et la figure 5.6 montre la précision correspondante. On s'intéressera ici seulement aux courbes en rouge qui correspondent aux traitements sans prédiction. La précision peut ici être calculé pour des séquences de synthèse puisque l'on dispose des valeurs réelles des paramètres utilisés lors de la création des séquences.

On constate une précision et un nombre d'étape nécessaires relativement constant au long des séquences. L'erreur reste très faible et le suivi se déroule donc dans de très bonnes conditions, ce qui était prévisible car les conditions sont optimales (images vidéo de même nature que les images de synthèse, modèle paramétré identique...).

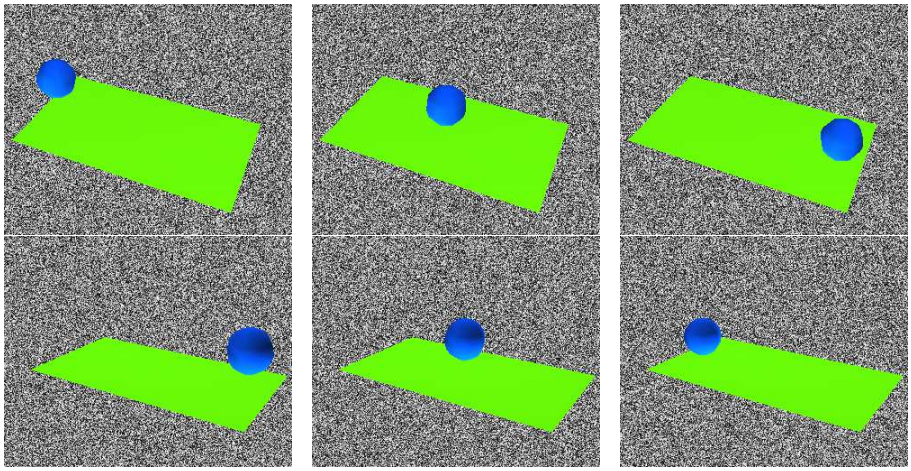


FIG. 5.4 – *La première ligne (resp. la deuxième) montre des images issues de la première caméra virtuelle (resp. la deuxième) au début, au milieu et à la fin de la séquence, de gauche à droite. En pratique, la sphère avance en rebondissant sur le support vert.*

### 5.4.2 Sphère en rebond avec prédiction

La même séquence est reprise ici, mais cette fois en utilisant une prédiction basée sur la conservation de la vitesse (premier degré), adaptée à ce type de mouvement.

On constate sur les figures 5.5 et 5.6 (courbes vertes) décrites précédemment une amélioration de la précision et une diminution du nombre d'étapes d'optimisation nécessaire lorsque la prédiction est utilisée. On constate également une baisse de la précision au moment du rebond – ce qui est logique puisqu'à ce moment là la prédiction est erronée – qui reste malgré tout du même niveau que lorsque la prédiction n'est pas utilisée (ceci grâce à la conservation d'un point sans prédiction lors de l'initialisation du simplex).

### 5.4.3 Exemple de décrochage

Dans cet exemple on retrouve la sphère précédemment vue, avec un décrochage artificiellement généré : à une étape donnée on bloque la transmission des valeurs des paramètres d'étape en étape (cela veut dire qu'à chaque début d'étape on utilise comme valeurs de départ les valeurs initiales de la séquence). Ceci permet d'augmenter la distance entre

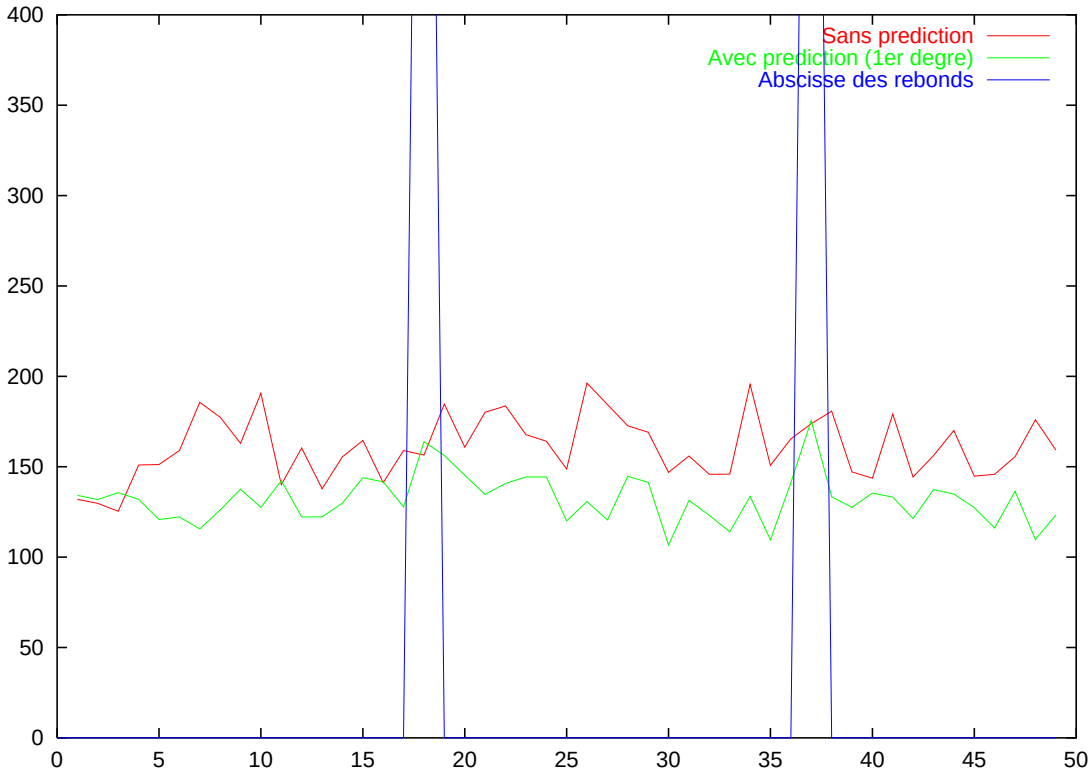



FIG. 5.5 – *Comparaison du nombre d'étapes d'optimisation de l'algorithme de suivi (en ordonné) pour chaque instant de la séquence (en abscisse), avec et sans prédiction (resp. la courbe verte et la courbe rouge). Les traits verticaux indiquent les instants des séquences où la sphère rebondit sur le sol.*

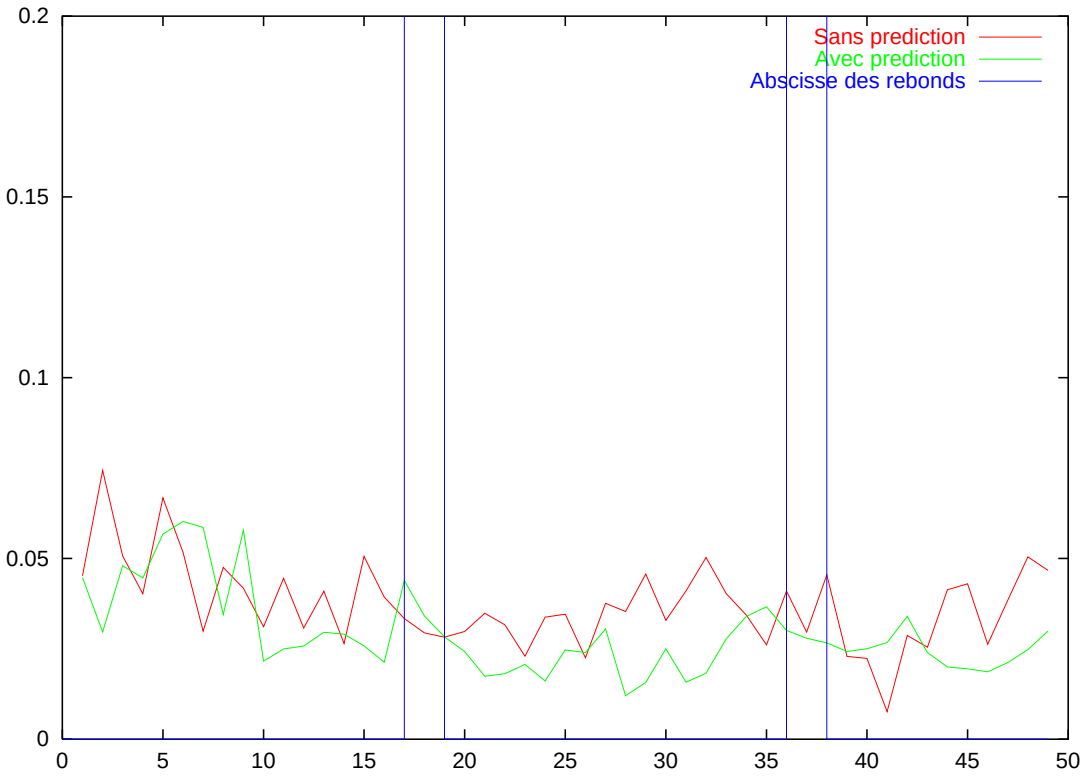



FIG. 5.6 – *Comparaison de la précision de notre système de suivi (en ordonné) pour chaque instant de la séquence (en abscisse), avec et sans prédiction (resp. la courbe verte et la courbe rouge). Les traits verticaux indiquent les instants des séquences où la sphère rebondit sur le sol.*

valeurs théoriques et valeurs initiales utilisées au fur et à mesure que les séquences se déroulent. La figure 5.7 montre comment varie la précision obtenue pour chaque instant des séquences.

On constate que jusqu'à une certaine distance initiale le système est robuste et maintient une précision de même niveau qu'en fonctionnement normal. On reste alors dans le domaine de convexité de la mesure d'écart). Au delà de cette distance critique le système diverge brusquement – on voit une brusque augmentation de l'erreur – et celle-ci reste ensuite à une valeur élevée. Les variations de la mesure d'écart après ce point correspondent à des différences d'optimisation dues à l'initialisation des points du simplex au début de chaque étape, celle-ci étant en partie aléatoire. On est dans ce dernier cas en dehors de la zone de convexité de la mesure d'écart, et la phase d'optimisation termine sur un minimum local à l'extérieur de la zone qui nous intéresse.

Notons également qu'au delà d'un certain point la distance initiale ne varie plus. Ceci est dû au fait qu'à partir de ce point la sphère dans les images de référence est totalement disjointe de celle générée par images de synthèse. En effet les paramètres initiaux ne changeant pas, celle-ci reste toujours à son point de départ alors que la sphère réelle se trouve à l'autre bout de l'image.

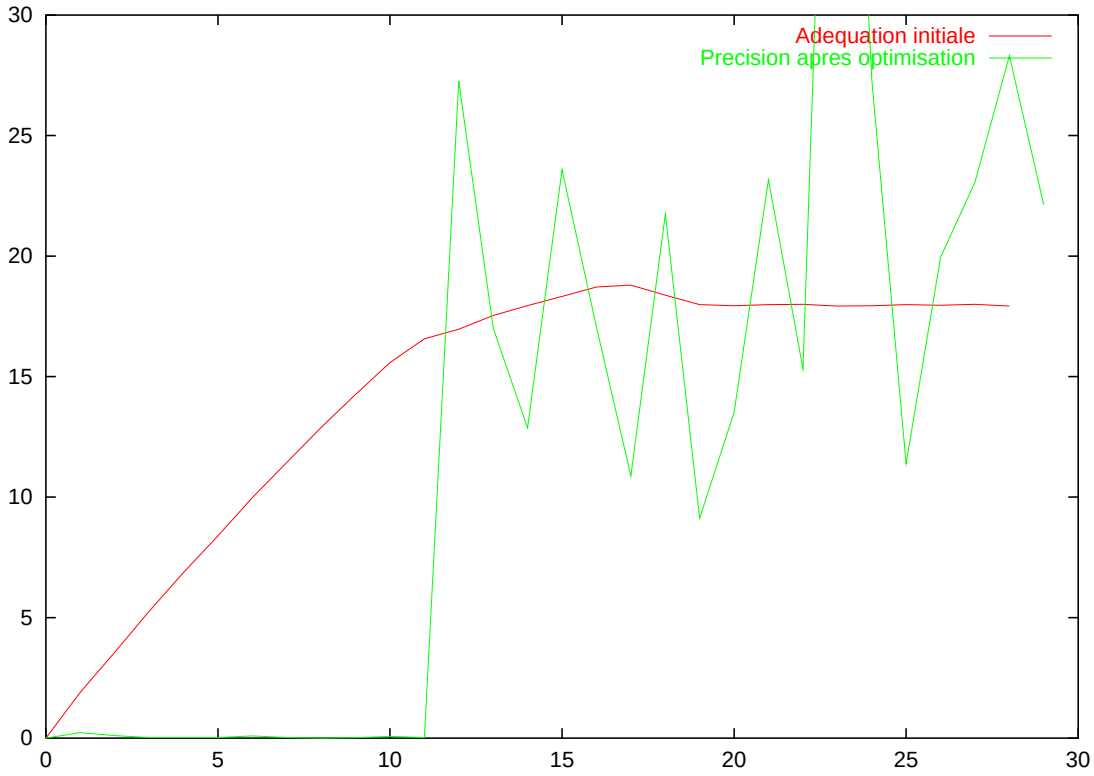



FIG. 5.7 – *Évolution de notre système de suivi lorsque l'on augmente artificiellement la distance entre le jeu de valeurs initiales et les valeurs théoriques correspondantes. En abscisses se trouve l'instant courant dans les séquences vidéo, et en ordonnées la mesure d'écart. La courbe rouge représente la distance entre les valeurs initiales et les valeurs théoriques, et la courbe verte la précision obtenue après la phase d'optimisation.*

# 6

# Comment choisir les valeurs initiales des paramètres

## Sommaire

Comme cela a été détaillé dans les sections précédentes, le suivi des paramètres dans le temps se base sur la propagation des valeurs des paramètres trouvées d'instant en instant, le résultat d'une étape étant utilisé comme valeur initiale pour l'instant suivant. Cela pose le problème de l'initialisation de ce système, c'est-à-dire l'obtention des valeurs initiales des paramètres correspondant au premier instant des séquences vidéo.

Le problème est donc : comment choisir $p_0$ un vecteur de paramètres tel que $p_0$ soit "proche" de $P_0$, le vecteur de valeurs théoriques correspondantes. Il s'agit d'un problème de localisation d'objet dans des images.

Initialement, nous avons utilisé un système manuel pour fixer ces valeurs. Ce système propose tout de même une certaine assistance à l'utilisateur, afin de lui rendre la tâche le plus simple et le plus rapide possible.

Le système se compose d'un ensemble de vues (les vues initiales de chaque séquence), avec en surimposition pour chacune d'elle les différentes vues de synthèse de l'objet étudié. L'opérateur peut alors faire varier les valeurs des paramètres à partir d'une interface graphique, jusqu'à ce que l'écart entre les vues réelles et les vues de synthèse soit nulle, la mesure d'écart étant ici l'oeil et le jugement de l'opérateur.

Ce type de technique est fréquemment utilisée dans des problèmes de cadrage manuel entre des scènes réelles et des objets de synthèse, par exemple dans le domaine des effets spéciaux et des trucages vidéo. Ce n'est qu'une assistance à la recherche de ces paramètres, le travail étant réalisé par l'opérateur.

Nous proposons dans les sections suivantes deux approches permettant d'obtenir des valeurs initiales, ces approches se distinguant par le degré d'automatisation qu'elles offrent.

## 6.1 Choix semi-automatisé

Cette seconde approche que nous proposons élimine le travail de recherche de l'écart pour l'opérateur, avec en contrepartie un phase de repérage d'entités – des points – dans les images réelles.

Notre système de modélisation permet de définir des points dans l'espace du modèle. Ces points ont un nom, et sont "accrochés" à l'objet. L'opérateur va associer des points dans les vues réelles – en 2D – avec des points du modèle. Cela se fait via la même interface graphique que celle précédemment décrite. Dans chaque vue réelle l'opérateur sélectionne des points et leur associe le nom d'un point du modèle. Notre système va alors chercher les paramètres du modèle tels que la projection des points associés au modèle corresponde au mieux avec les coordonnées 2D qui leurs sont associées dans les différentes vues (voir figure 6.1). Si $P_n$ est le $n^{eme}$ point associé au modèle (dans le repère du modèle) et $Proj_{/c}(P_n, p)$ la projection de ce point dans la vue $c$ pour le vecteur de paramètre $p$, alors il s'agit de trouver $p$ minimisant la fonction d'erreur :

$$(p) = \sum_{c=0}^{c<C} (\sum_{n=0}^{n<N} d(Proj_{/c}(P_n, p), q_n^c)) \tag{6.1}$$

soit minimale, avec $q_n^c$ le point 2D correspondant au point du modèle $P_n$ dans la vue $c$.

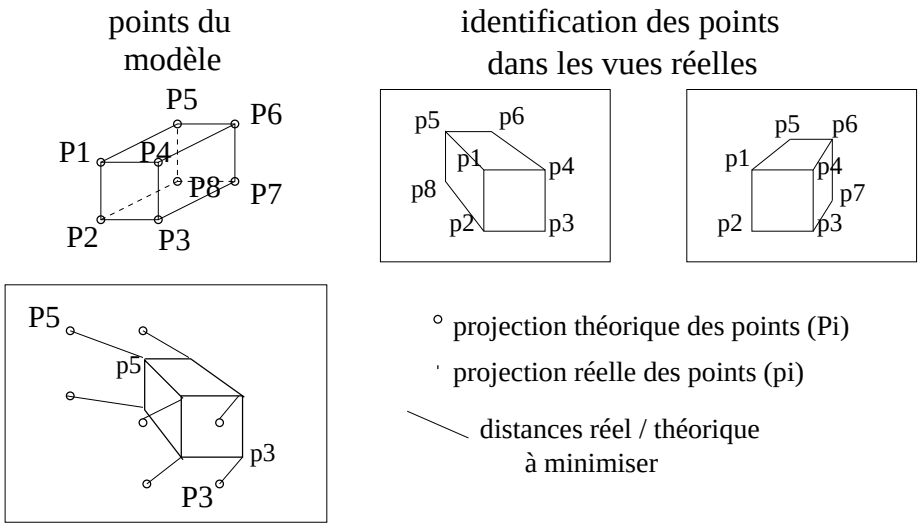



FIG. 6.1 – *Désignation des projections des points du modèle ($P_i$) dans les images ($p_i$), et distances entre la projection théorique des ces points (via les paramètres courant) et les positions données. Ce sont ces distances qui seront optimisées.*

Lorsque le système est suffisement contraint, on obtient ainsi le vecteur de valeurs initiales que l'on cherchait. Il est à noter que pour un objet rigide trois associations "point du modèle / point dans une vue" suffisent théoriquement pour contraindre les six degrés de liberté de l'objet (dans un cas général). Cependant plus on ajoute d'associations et plus la précision finale sera grande, les erreurs de sélection des coordonnées dans les vues réelles se compensant entre elles.

## 6.2 Vers une méthode de choix automatisée

Nous avons cherché à rendre totalement automatique l'obtention des valeurs initiales des paramètres en début de séquence, et donc à définir une autre approche que celle présentée dans la section précédente.

Dans ce type de problème, il n'y a que deux choix possibles : ou bien on exploite l'apparence prévue de l'objet (le modèle rendant cela possible) pour le chercher dans les images, ou bien on effectue une recherche exhaustive dans l'espace des paramètres.

Dans cette approche nous avons couplé une recherche exhaustive peu précise avec notre méthode de recherche des paramètres.

### 6.2.1 Initialisation

L'idée est d'utiliser une méthode de type Monté-Carlo pour effectuer une recherche sur l'ensemble du domaine mais en utilisant des images (réelles et de synthèse) de faible résolution. Les traitements sont alors rapides – le temps de traitement est proportionnel au nombre de pixels dans les images – mais peu précis.

Il s'agit d'utiliser cette exploration rapide et exhaustive pour dégager un certain nombre de solutions candidates. Nous conservons les $k$ vecteurs de paramètres les plus intéressantes, en prenant soin d'éliminer les vecteurs trop proches les uns des autres.

Ces candidats vont être ensuite étudié plus finement, c'est-à-dire à des résolutions plus précises, afin de les départager.

### 6.2.2 Optimisation des solutions candidates

À chaque étape nous effectuons pour chaque vecteur candidat une optimisation de façon à obtenir les valeurs des paramètres minimisant la fonction d'écart. La résolution étant augmentée à chaque étape, on obtient des résultats plus fins permettant de juger plus précisement de la validité des candidats disponibles.

À l'issue de chaque étape les candidats qui s'avèrent les moins intéressant sont éliminés, et l'on recommence en augmentant la résolution.

Le processus se termine lorsque l'on atteint la résolution maximale (la résolution initiale des images) et qu'il ne reste plus qu'un candidat (voir figure 6.2).

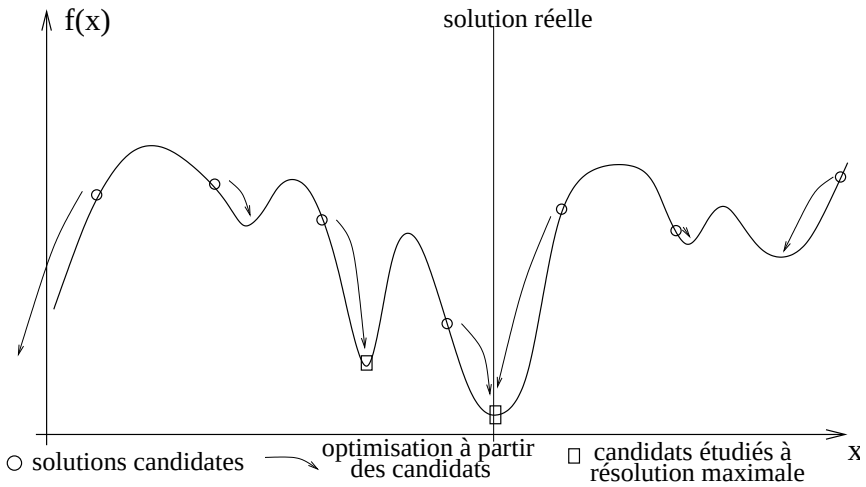



FIG. 6.2 – *Recherche exhaustive de candidats. Les points les plus intéressant servent successivement de vecteur initial pour notre méthode de recherche des paramètres. Nous conservons le candidat aboutissant à la mesure d'écart la plus intéressante.*

Cette technique donne des résultats intéressants, mais "s'égare" parfois. Les cas posant problème sont en général des objets relativement petits, ou fins, et donc fortement dégradés par la diminution de la résolution. Ces objets dégradés peuvent alors correspondre avec des détails de l'arrière-plan – des sortes d'*illusions d'optique* pour la mesure d'écart. Un autre problème pouvant survenir est lié à l'aspect stochastique de la recherche initiale. Si aucun candidat intéressant n'est trouvé dans la zone convexe autour de la solution réelle les résultats seront erronées.

Cette approche ne permet donc pas une automatisation totale de la recherche des paramètres initiaux, car l'opérateur humain doit toujours valider les résultats obtenus. En

cas d'échec il doit alors éliminer manuellement la solution obtenue et relancer le processus. Cela reste néanmoins un premier pas vers une automatisation de cette phase nécessaire qu'est la recherche des paramètres initiaux.

Une amélioration peut consister à passer par une pré-estimation de la position de l'objet, afin de réduire le domaine des paramètres. Nous cherchons en particulier à prendre en compte les zones de visibilité des caméras. En posant l'hypothèse peu réductrice que l'objet est visible dans toutes les vues, il est possible de limiter les paramètres au domaine où ils correspondent à un objet visible.

## 6.3 À propos du calibrage des caméras

Nous avions succinctement présenté dans la section 2.2.1 la technique de calibrage des caméras que nous utilisons. Celle-ci est en fait de même nature que pour la recherche des paramètres initiaux.

Nous utilisons la méthode décrite dans la section 6.1. À partir d'un objet tridimensionnel connu, nous définissons un modèle de celui-ci, en spécifiant un ensemble de points 3D facilement repérables. Nous associons à chacun de ces points 3D les positions 2D leurs correspondants dans les images réelles.

Les paramètres utilisés sont les caractéristiques définissant les caméras à calibrer. La phase d'optimisation va ainsi consister à minimiser les distances entre les projections des points 3D du modèle et leurs positions dans les images. Comme la projection est effectuée en utilisant les paramètres des caméras, atteindre l'optimum revient à trouver la calibrage recherchée.

Dans notre cas, les paramètres de la calibrage sont :
- la position du centre optique ;
- l'axe principal de la caméra, perpendiculaire au plan de projection des images ;
- et un vecteur définissant la verticale de la vue.

Les caméras que nous utilisons ayant une focale connue et fixe, ce paramètre n'est pas recherché. Comme nous négligeons les effets de déformation dus aux optiques, aucune information de calibrage n'est à définir pour cela. Au total, neuf paramètres sont utilisés pour définir la calibrage d'une caméra.

En pratique, nous utilisons une *boite ouverte* comme objet de calibrage. Il s'agit en fait de trois carrés de dimensions connus, tous perpendiculaires les uns par rapport aux autres et accolés par une arête (voir figure 6.3). Dix points de cet objet sont utilisé pour la calibrage, donnant expérimentalement de très bon résultats, à condition :
- que l'objet filmé ne soit pas trop proche de la caméra, car négliger les effets de déformation des optiques n'est pas valable pour de faibles distances ;
- que l'objet filmé soit proche de la position spatiale de l'objet de calibrage.

Ces limitations sont dues au fait que le modèle de caméra trouvé par ce type de méthode repose sur une optimisation à partir de données correspondant à un volume spatial limité. Le modèle n'est donc théoriquement valable que dans le voisinage spatial correspondant, et la précision peut chuter au delà.

Malgré tout, on constate en pratique que pour des objets de dimensions usuelles la précision constatée est suffisement bonne pour obtenir ensuite un suivi correct.

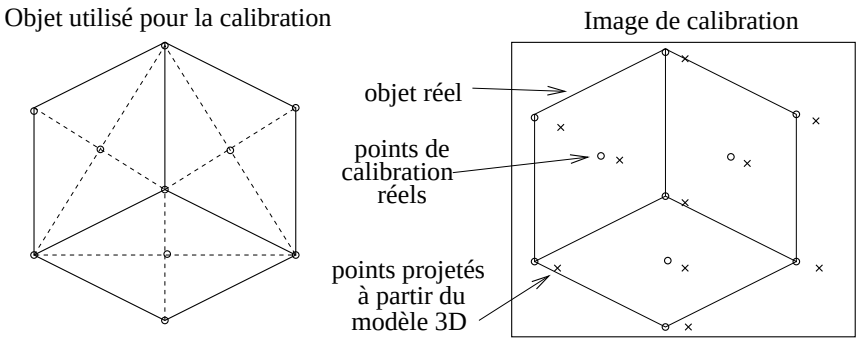



FIG. 6.3 – *Objet utilisé pour la calibrage des caméras. On peut voir sur cette figure les dix points 3D dont les positions sont connues. À chacun de ces points est associé sa projection dans les images réelles. La calibrage consiste à rechercher les paramètres de la caméra minimisant la distance entre les points 2D et la projection des points 3D grâce au modèle de la caméra.*

# Quatrième partie

# Auto-affinage de modèles géométriques

# 1

# Auto-affinage structurel d'un modèle

## Sommaire

Notre approche nécessite pour bien fonctionner de disposer d'un modèle géométrique paramétré *visuellement* fidèle à l'objet suivi. Cette notion de visuellement fidèle veut simplement dire que les vues artificielles de l'objet, créées grâce à ce modèle, doivent être ressemblante à l'original, l'objet réel. Cela implique deux choses : disposer d'une géométrie fidèle de l'objet d'une part, et disposer de l'aspect visuel de ces surfaces – réflectance, réfractance, textures... – d'autre part.

Les effets spéciaux au cinéma et à la télévision font amplement usage d'objets de synthèse réalistes, dans le but d'être intégré dans des scènes réelles (trucages, effets spéciaux...). Il devient possible d'obtenir pour de nombreux objets et êtres vivants un très grand réalisme – on pensera par exemple à des films tels que *Jurrasik Park* ou *Coeur de dragon*. Au delà de la qualité du programme générant les images de synthèse, le point important pour atteindre un réalisme élevé est la finesse de définition du modèle de l'objet lui-même.

Notre système de suivi à été conçu pour être robuste aux petites erreurs et écarts de modélisation. Un modèle *parfaitement* fidèle (quoi que puisse vouloir dire ce terme) n'est donc pas nécessaire. Cependant moins le modèle géométrique sera proche de l'objet réel et moins la précision des résultats sera grande. Il est donc intéressant d'obtenir un modèle le plus fidèle possible si l'on désire obtenir une précision finale élevée.

Il existe de nombreux travaux portant sur la création de modèles d'objet. Au delà des systèmes utilisant des senseurs pour obtenir la géométrie d'un objet – palpeurs 3D, interférométrie, systèmes lasers [La3], photo 3D [Ph3]... – on trouve une grande littérature sur les problèmes de reconstruction d'objets à partir de plusieurs vues d'un même objet. Des logiciels[fac, may] permettent la modélisation d'objets à partir de photos en utilisant les propriétés projectives 3D $\rightarrow$ 2D ainsi que des contraintes sur la géométrie (parallélisme, orthogonalité...). C'est à l'utilisateur de définir dans ce cas les éléments intéressant des images, le logiciel effectuant ensuite la reconstruction. Des systèmes utilisant les réseaux de neurones sont utilisés[ard91] pour obtenir la géométrie d'un objet à partir d'images. Ceci peut également être réalisé par projection inverse des silhouettes, définissant un volume dans lequel se trouve l'objet. L'inconvénient est qu'il faut être sûr que les vues choisies suffisent à contraindre la géométrie de l'objet. D'autres approches utilisent le calcul variationnel [Fau98] pour – entre autre – obtenir un modèle de synthèse d'objets à partir de pairs stéréoscopique. Le principal problème de ces méthodes de création de modèles est qu'il n'est pas possible de reconstruire les articulations et autres degrés de liberté, l'objet n'étant vu que dans une position donnée. Il est toujours possible de générer des modèles statiques pour plusieurs états de l'objet, puis de déterminer les transformations permettant de passer d'un état à l'autre, mais ceci reste compliqué à mettre en œuvre et nécessite un très grand nombre de vues si l'objet a un grand nombre de degrés de liberté. Dans [OFH+98] les auteurs utilisent un modèle 3D générique de l'objet à reconstruire. Ce modèle est déformé pour correspondre à une carte de profondeur obtenue grâce aux flux optiques, à partir des séquences d'images. Des paires stéréoscopiques sont utilisées dans [BSH00] pour reconstruire le modèle d'un objet plan par partie, grâce à la géométrie épipolaire appliquée à des points coplanaires.

Si le modèle a été généré par l'une de ces méthodes, il suffit alors de l'importer dans notre système, à condition que l'on ait bien défini ses paramètres. Si ce n'est pas le cas, la charge revient à l'opérateur de définir ce modèle, via un système de modélisation quelconque, les logiciels de modélisation 3D ne manquant pas.

Afin d'assister l'opérateur durant la phase de définition du modèle de l'objet à étudier, nous avons travaillé sur l'assistance à la définition de ces deux parties importantes du modèle :

- l'aspect géométrique du modèle. Un rendu fidèle passe par la définition d'une géométrie fidèle par rapport à l'objet ;
- les propriétés des surfaces, nécessaires pour que les surfaces visibles du modèle soient fidèles par rapport aux images réelles de l'objet.

Nous présentons à la fin de cette partie quelques exemples d'affinages et d'extraction des textures d'objets à partir de plusieurs objets réels.

## 1.1 Paramètres intrinsèques et extrinsèques

Nous allons tout d'abord définir deux catégories de paramètres intervenant dans un modèle géométrique.

La première catégorie regroupe sous le nom de paramètres extrinsèques l'ensemble des paramètres contraignant les degrés de liberté de l'objet. Il s'agit pour un objet non articulé des six degrés de liberté : trois pour le positionnement spatial et trois pour l'orientation. Dans le cas d'objets articulés se rajoutent les degrés de liberté liés à chaque articulation : la plupart du temps des rotations, mais parfois également des translations dans le cas d'objets industriels par exemple (piston, pompe...). Ces différents paramètres contrôlent la "posture" spatiale de l'objet, mais n'influent pas sur sa géométrie propre.

La deuxième catégorie, appelée paramètres intrinsèques, correspond aux paramètres liés aux caractéristiques géométriques de l'objet. Ce sont par exemple les longueurs, diamètres, épaisseurs... qui définissent la forme de l'objet. Il est à noter que ces paramètres sont constant lors de la phase de suivi, les objets étudiés étant considérés indéformables.

Lors de la création du modèle paramétré d'un objet, l'ensemble des paramètres extrinsèques sont contraints par la structure de l'objet lui-même. La transposition de ces informations dans le modèle s'apparente à la construction d'un *squelette*, où les différentes parties mobiles de l'objet sont reliées par des articulations.

Pour les paramètres intrinsèque, toutefois, un travail de mesure est nécessaire pour obtenir l'ensemble des dimensionnements de l'objet et les reporter dans le modèle. Lorsque l'on a à faire à un objet manufacturer, il est parfois possible d'obtenir des plans de construction précis, aidant à ce travail. Lorsque l'objet à des formes simples, des mesures directes sont également possible. Mais lorsque l'objet présente des formes complexes, cela peut représenter un travail fastidieux.

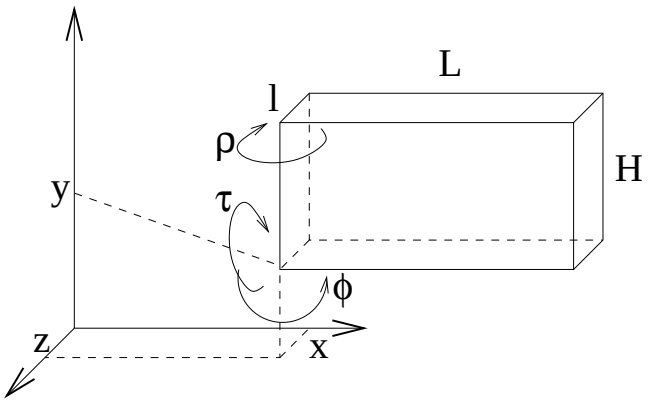



FIG. 1.1 – *Sur cet exemple, $l$, $L$ et $H$ sont des paramètres intrinsèques, car ils définissent la géométrie de l'objet, et $x$, $y$, $z$, $\theta$, $\phi$ et $\rho$ sont des paramètres extrinsèques car ils représentent les degrés de liberté de l'objet et permettent de le contraindre spatialement.*

## 1.2 Ajustement structurel d'un modèle

Nous avons développé une méthode permettant d'ajuster les informations structurelles d'un modèle par rapport à un objet réel afin de faciliter la phase de définition des paramètres intrinsèques.

Il s'agit de partir d'un modèle proche de l'objet étudié, mais dont les dimensions – les paramètres intrinsèques – sont approximatifs. Ce type de modèle est beaucoup plus facile à définir, ne nécessitant pas de mesures physiques précises. L'idée est de considérer les paramètres intrinsèques au même titre que les paramètres extrinsèques : des paramètres dont les valeurs théoriques correspondent à l'objet réel, et dont on cherche les valeurs. Il est ainsi possible d'appliquer notre approche de recherche des valeurs des paramètres, en l'appliquant aux paramètres intrinsèques.

La phase d'affinage géométrique consiste ainsi à appliquer notre méthode de recherche des paramètres d'un modèle aux paramètres intrinsèques, à partir d'un ensemble de vues de l'objet étudié – n'importe quelle vue faisant l'affaire, l'objet étant indéformable. Notre système va rechercher les paramètres intrinsèques tels que les vues de synthèse de ce modèle soient le plus fidèle possible par rapport aux images réelles, et donc correspondent aux valeurs théoriques.

La seule différence avec une phase "classique" de recherche des paramètres est qu'ici la *forme* de l'objet est recherché (en fait affiné) au lieu des paramètres spatiaux lors du suivi.

## 1.3 Prise en compte des paramètres extrinsèques

Pour avoir un sens, cet affinage des paramètres intrinsèques ne peut se faire que si l'on connaît exactement les valeurs des paramètres extrinsèques. En effet si l'objet est mal positionné, ou bien si ses articulations ne sont pas dans l'état correct, tenter d'optimiser la forme de l'objet ne peut fonctionner car la mesure d'écart sera faussée.

Il faut donc obtenir les valeurs de ces paramètres extrinsèques. La méthode utilisée est en fait un aller-retour entre deux phases d'optimisation distinctes :

- une phase d'optimisation des paramètres extrinsèques, avec un modèle approximatif – les paramètres intrinsèques sont alors constants ;
- une phase d'optimisation des paramètres intrinsèques – les paramètres extrinsèques sont alors constants.

En effet, après la première phase de recherche des paramètres extrinsèques, les valeurs trouvées sont forcement approximatives, puisqu'à partir d'un modèle imprécis il n'est pas possible de trouver leur valeurs exactes.

Suit alors la phase de recherche des paramètres intrinsèques, qui elle aussi donnera des résultats imprécis, la position n'étant pas exacte.

Ces deux phases sont itérées jusqu'à ce qu'il y ait stabilisation des différentes valeurs des paramètres – intrinsèques et extrinsèques (voir figure 1.2).

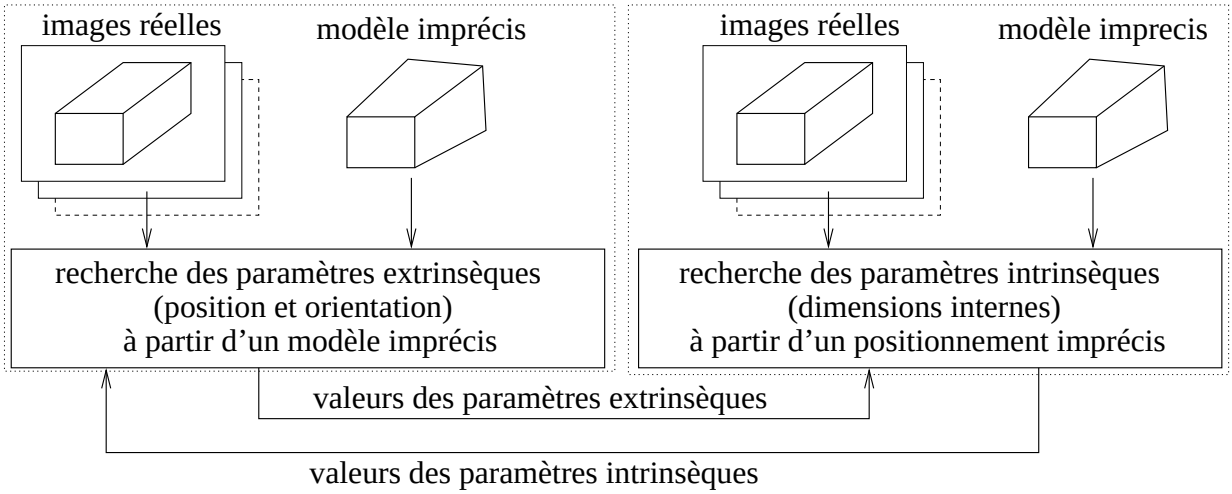



FIG. 1.2 – Cycle de recherche des paramètres intrinsèques / extrinsèques pour l'auto-affinage structurel d'un modèle géométrique.

Cette méthode présente toutefois quelques limites. En particulier si le modèle ne dispose pas de propriétés de surfaces finement définies (couleurs et surtout textures), le biais correspondant peut fausser les résultats. De plus pour des objets complexes – par exemple possédant de nombreux éléments de petite taille comme les doigts d'une main – les imprécisions sur les paramètres extrinsèques peuvent être suffisement grandes pour fausser la recherche des paramètres intrinsèques.

A contrario elle présente plusieurs avantages par rapport à d'autres méthode d'extraction de modèle à partir d'images.

Tout d'abord notre méthode permet de prendre en compte directement les informations en provenance des multiples vues de l'objet disponibles.

Elle s'accomode de plus d'objets articulés. De nombreuses méthodes utilisent en effet le flux de mouvements de l'objet dans les séquences ou encore les corrélations entre points pour déterminer la carte de profondeur liée à la vue, pour ensuite reconstruire l'objet. Cela suppose un objet constant – pas de variations de forme – durant la séquence correspondante.

Elle permet également d'obtenir des informations sur des parties non visibles de l'objet, via les contraintes géométriques implicites dans le modèle. Par exemple les méthodes effectuant une reconstruction de l'objet à partir uniquement de son apparence ne peuvent décider de la géométrie correspondant à des zones non visibles de l'objet. Si l'on cherche à affiner un parallélépipède, par exemple, une face peut être invisible dans les différentes vues mais les informations disponibles sont suffisantes pour trouver la géométrie de l'objet.

## 1.4 Limites de l'approche

Cette méthode présente toutefois une limite importante : elle reste très sensible aux conditions initiales, c'est-à-dire aux valeurs de départ des paramètres fournies.

En effet la phase de recherche des paramètres extrinsèques donne des résultats d'autant moins précis que la géométrie de l'objet s'éloigne de la géométrie réelle.

Similairement, la recherche des paramètres intrinsèques donne des résultats d'autant moins précis que les position / orientation / état des articulations sont imprécis.

Si les imprécisions initiales sont faibles, le système finit par converger. Lorsque ces imprécisions deviennent plus grandes, des effets d'oscillation entre les phases de recherche des paramètres intrinsèques et extrinsèques apparaissent, rendant la convergence de l'ensemble beaucoup plus lente. Finalement, pour des imprécisions trop grandes le système ne se stabilise jamais complètement, la marge de manœuvre étant trop grande.

Nous avons bien évidement envisagé d'effectuer la recherche de tous les paramètres simultanément (intrinsèques et extrinsèques). Cette façon d'aborder le problème permet théoriquement d'éliminer le phénomène d'oscillation, principalement dû à l'alternance des phases de recherche. Cependant les degrés de liberté sont alors très grand pour le système, et dès que les objets deviennent complexe des effets indésirables apparaissent. Ces effets consistent par exemple à compenser une erreur de géométrie par un déplacement de l'objet, ou bien l'inverse. En pratique l'interaction de ces paramètres de types différents génère une fonction d'écart extrêmement bruité, perturbant grandement le fonctionnement de la phase d'optimisation.

Pour limiter cela, nous utilisons en fait cette méthode *après* une phase de pré-affinage semi manuelle. Nous utilisons pour cela la méthode décrite dans la section 6.1 page 65 et consistant pour l'utilisateur à définir un certain nombre de points sur le modèle de l'objet, et à préciser où ceux-ci se projètent dans les images réelles. Le système cherche ensuite les paramètres qui minimisent les distances entre la projection des points du modèle et les positions image indiquées.

Nous utilisons cette méthode à la fois pour le positionnement et pour la géométrie de l'objet. Par exemple pour un cube, un choix de trois points pris aux coins de l'objet suffisent pour le contraindre géométriquement et spatialement (si l'on dispose de plusieurs vues).

Cette méthode ne permet pas d'obtenir une grande précision pour plusieurs raisons, dont la principale est la relativement faible précision de la sélection d'un point d'une image par un utilisateur.

C'est pourquoi, après cette phase qui nous permet relativement facilement d'obtenir un premier modèle quasi-fiable, nous enchaînons avec une phase d'auto-affinage afin

d'améliorer la précision du résultat.

Les exemples que nous présentons plus loin partent tous d'un modèle initial généré par cette méthode.

# 2

# Prise en compte des textures

## Sommaire

Si l'on dispose d'un modèle d'un objet dont la géométrie est fiable, on sait que le rendu de synthèse de ce modèle sera géométriquement correct, c'est-à-dire que la projection d'un point du modèle dans une vue correspondra à la position du point correspondant de l'objet réel dans l'image vidéo.

Pour obtenir une bonne ressemblance avec l'objet réel étudié, il faut que l'aspect des surfaces de ce modèle soit correct. Un modèle de dé à jouer peut avoir exactement les dimensions d'un dé réel, mais si le modèle possède des faces blanches alors que l'objet réel est rouge et présente des numéros sur ses faces, la ressemblance visuelle – et par voie de conséquence la mesure d'écart – seront biaisées.

C'est pour cette raison que nous avons développé une méthode permettant d'extraire automatiquement ces informations à partir du modèle et des valeurs des paramètres extrinsèques, en se basant sur le passage 3D $\rightarrow$ 2D.

## 2.1 Passage de l'espace image à l'espace modèle : la rétro-projection

Si l'on possède un modèle correct de l'objet étudié, et que celui-ci est bien positionné dans l'espace (ses paramètres extrinsèques sont corrects), on peut obtenir les coordonnées images de tout point de la surface du modèle en effectuant la projection des coordonnées 3D de ce point dans le repère de la caméra correspondante à la vue.

Il est ainsi possible de savoir à quoi *ressemble* un point (ou tous les points) de l'objet en réalité – la réalité étant ici les vues issues des séquences vidéo.

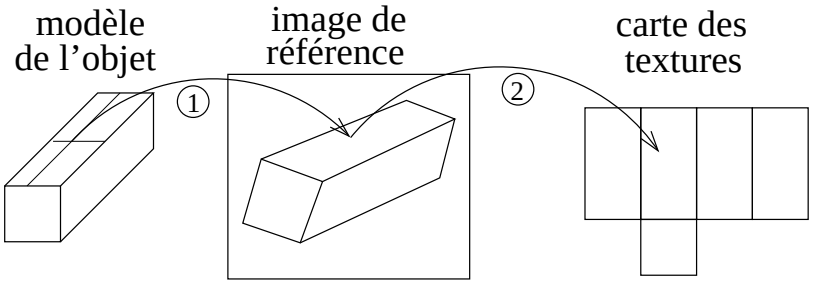
Il existe des méthodes permettant de rechercher l'apparence d'un objet à partir d'un modèle. Dans [LRD98] la rétro-projection est utilisée pour l'apprentissage des textures d'un objet dont le modèle géométrique est connu, mais sans aborder le problème de la fusion des informations issues de plusieurs vues. Ce problème est abordé dans [OSRW97] où les auteurs traitent de la fusion multi-précision de textures. Malgré tout dans ce dernier cas les objets utilisés sont simples, et les problèmes d'auto-occultation et de prise en compte de la qualité des données ne sont pas pris en compte.

Pour parvenir à un résultat de bonne qualité, nous extrayons tout d'abord les textures des surfaces de l'objet en réalisant un parcours exhaustif des points du modèle de l'objet et en leur associant les valeurs des pixels dans les images (voir figure 2.1 page suivante).

Cette méthode n'est bien entendu pas suffisante en elle même. En particulier elle ne prend pas en compte plusieurs points importants :

- la projection 3D $\rightarrow$ 2D indique ou le point du modèle se projète, et pas s'il est visible. Il peut être caché dans deux cas : auto-occultation, le point étant caché par l'objet lui-même dans cette vue et occultation, le point étant caché par un autre objet, inconnu (car non modélisé) ;
- plusieurs vues étant disponible, ce travail est à réaliser dans chacune d'elle, posant le problème de la fusion des informations.

Le premier problème, l'auto-occultation, se règle facilement en utilisant les informations générées par le moteur de rendu : la "profondeur" des pixels générés le long de l'axe de chaque vue. Cette profondeur correspond à la distance du point considéré à la caméra.




① passage d'un point du modèle à un pixel de l'image

② récupération du pixel dans la carte des textures

FIG. 2.1 – Rétro-projection : passage d'un point du modèle à un pixel de l'image, puis utilisation de ce pixel pour construire la texture correspondante. Pour le passage d'un point du modèle à l'image on utilise la matrice de projection de la caméra correspondant à la vue.

Lors de la projection d'un point du modèle dans l'espace image, il suffit alors de vérifier que sa profondeur correspond bien à la profondeur du pixels correspondant dans l'image. Si c'est la cas le point est visible, sinon il est auto-occulté.

Les données issues de plusieurs vues sont gérées comme suit : pour chaque vue l'ensemble des textures est calculée, la couleur par défaut des surfaces étant utilisée lorsque les points ne sont pas visibles. À chaque point est associé un degré de confiance qui dépend de plusieurs critères :

- si le point n'est pas visible son degré de confiance est nul ;
- plus un point de l'image correspond à un point éloigné de la caméra et moins son degré de confiance sera faible ;
- plus un point est vue "en biais" et moins son degré de confiance sera grand. Le "biais" correspond à l'écart entre la normale à la surface au point considéré et l'angle effectif de vue.

## 2.2 Notion de confiance pour les textures extraites

Dans la section précédente nous avons détaillé la façon par laquelle nous obtenons pour tout point de la surface de l'objet sa couleur à partir d'une image vidéo (si ce point est visible).

Disposant de plusieurs vues simultanées de l'objet, nous allons obtenir plusieurs cartes des textures – autant qu'il y a de vues. Ces différentes textures vont devoir être combinées pour créer la texture finale de l'objet, comme cela sera décrit plus loin. Avant cela, il est important de constater que les éléments de texture extraits (les texels) sont d'une qualité variable, information à prendre en compte lors de la fusion.

Nous associons à chaque texel un degré de confiance, qui sera utilisé lors de la fusion. Cette confiance dépend de plusieurs critères :

- la visibilité : un texel non visible (par exemple derrière la vue considérée) aura un degré de confiance nul ;

- la surface en pixels image du texel considéré (voir figure 2.2). Si un texel se projette dans l'espace image sur plusieurs pixels images, sa couleur sera une moyenne des couleurs de ces pixels, et portera donc une information moins précise que lorsqu'un texel correspond à un pixel (ou moins). Cela correspond en fait à un sous-échantillonage des surfaces de l'objet par rapport à la précision avec laquelle celui-ci est visible dans les images ;
- l'angle de vue (voir figure 2.2). Plus un pixel est vu de "biais" et plus sa couleur correspond – dans l'image – à une moyenne des points physiques correspondant. Il est à noter que ce critère est complémentaire du précédent. En effet dans ce type de situation la surface projeté du texel est très inférieure à un, mais le critère précédent ne permet pas de différencier un sur-échantillonage de la surface – inutile mais correspondant à des texels de haut degré de confiance – de ce type de situation où le degré de confiance sera faible.

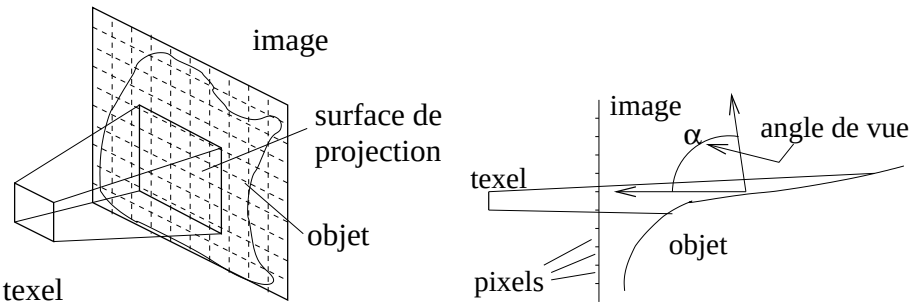



FIG. 2.2 – À gauche : surface de projection d'un texel dans l'espace image. À droite : angle entre la surface effective de l'objet et la caméra.

L'ensemble de ces informations nous permet d'associer à chaque texel un degré de confiance qui représente la fiabilité a priori en chaque point de l'objet, et va permettre d'effectuer la fusion des textures en prenant les informations les plus pertinentes.

Ainsi pour un texel donné, on note $v$ sa visibilité (valant 0 ou 1), $a$ l'angle de vue correspondant (ramené entre 0 et 1 pour des angles allants de $0^o$ à $90^o$, car au delà de $90^o$ le texel est vu "de dessous", et le degré de confiance sera nul), et $s$ sa surface projetée en pixels dans l'image. Alors le degré de confiance $c$ s'exprime comme suit :

$$c = v.(a.\min(1, \frac{1}{s}) \tag{2.1}$$

Ce degré de confiance varie donc entre 0 et 1 de façon décroissante avec l'angle de vue et la surface de projection, sachant qu'il sera nul en cas de non visibilité.

## 2.3 Prise en compte de l'éclairement des textures

Un autre phénomène à prendre en compte est l'éclairement. Les textures telles qu'elles sont vues dans les images sont éclairées par les différentes sources lumineuses de la scène.

En ne tenant compte que de leur aspect visuel, un cube ayant toutes ces faces identiques semblerait avoir une face beaucoup plus claire que les autres, celle orienté vers la lumière.

Dans un exemple comme celui-ci, le résultat est un ensemble de faces identiques. Pour parvenir à cela, il faut "retrancher" l'effet des éclairages de la scène lorsque l'on extrait les couleurs des texels.

Nous faisons appel pour cela à la carte des éclairements des images de synthèse. Lors de la rétro-projection des points du modèle, nous calculons (en fait le moteur de rendu effectue cette opération) l'éclairement arrivant sur le texel considéré.

La couleur associée au texel est alors celle qui, dans les conditions d'éclairement calculées, donnerai la couleur effectivement trouvée dans l'image vidéo.

Ceci permet de normaliser la luminosité des textures extraites, en compte les phénomènes d'éclairement et d'ombres portées connues.

## 2.4 Combinaison de textures issues de plusieurs sources

La dernière phase consiste à fusionner ces textures en une seule, regroupant le meilleur de chaque vue pour construire la texture finale de l'objet.

En premier lieu toutes les textures sont ré-exprimées à la résolution de la texture la plus précise.

Ensuite chaque texel ayant une confiance nulle est remplacé par la couleur par défaut associée dans le modèle.

En dernier lieu les texels des différentes vues sont fusionnés en fonction de leur degré de confiance respectif. Si $C_i$ est la couleur d'un texel dans la vue $i$ parmi les $C$ vues disponibles, et $\alpha(C_i)$ son degré de confiance associé, on calcul alors $c$ la couleur finale comme suit :

$$c = \frac{1}{\sum_{i=0}^{i<C} \alpha(C_i)} . \sum_{i=0}^{i<C} C_i.\alpha(C_i) \tag{2.2}$$

avec $n$ le nombre de vues disponibles.

Ce calcul est valide si les texels extraits correspondent réellement à l'objet. Si une occultation non modélisée cache une partie de l'objet, le système n'a aucun moyen de s'en rendre compte et fusionnera des données de nature différente.

C'est pourquoi nous ne fusionnons pas les texels si la variance de l'échantillon dépasse un seuil donné. Au delà de ce seuil, nous considérons qu'au moins un des texels correspond à une occultation non prévue. Le texel le plus éloigné de la moyenne des valeurs est alors retiré et le calcul est recommencé.

Cette technique échoue si l'on ne dispose plus que de deux vues, car il n'est pas possible dans ce cas de décider lequel des deux points est invalide. Nous utilisons alors comme critère de choix la distance à la couleur par défaut associée à ce point. Le point le plus éloigné de cette valeur est éliminé, et la valeur du point restant est utilisée seule.
La section suivante montre un exemple d'extraction de textures, en détaillant les phases intermédiaires des calculs.

## 2.5 Parties non visibles d'un objet

Il est à noter que si une partie de l'objet est invisible dans toutes les vues de référence utilisées pour l'extraction des textures, aucune texture ne sera associée à cette partie et la couleur par défaut de l'objet sera utilisé.

Il se peut cependant qu'au cours des mouvements de l'objet cette partie devienne visible, et que l'absence de texture pénalise la précision.

Si l'on tient a avoir une texture complète de l'objet, on peut alors utiliser plusieurs jeux de vues de l'objet, à des instants différents – et donc avec des visibilités différentes. En pratique, cela ne change rien au principe de l'extraction des textures, sachant que pour chaque jeu de vues de référence il faut disposer d'un positionnement spatial correct (les caractéristiques géométriques étant supposées constante au cours du temps).

À partir de chaque vue, quelque soit l'instant dont elle dépend, on effectue la phase de rétro-projection comme décrit précédemment.

On obtient alors une carte des textures pour chacune d'entre elles, le nombre de cartes étant simplement deux fois plus élevé – ou plus de deux fois plus si l'on utilise des vues issues de plus de deux instants différents.

Ensuite la phase de fusion se déroule identiquement, à ceci près que pour cela on dispose de plus de cartes de texture, et donc a priori d'une précision plus grande – pour les zone visible dans de nombreuses vues – et de zones de non visibilité plus réduite, car la multiplication des points de vues (ou des positions de l'objet, ce qui revient dans ce cas au même) réduit fortement les possibilités d'auto-occultations ou de faces cachées.

# 3

# Exemples d'affinage de modèles

## Sommaire

Nous présentons dans cette partie quelques résultats obtenus avec notre méthode d'auto-affinage structurel de modèles (section 3.1) ainsi qu'avec notre méthode d'extraction des textures (section 3.2 page 87).

Ces deux sections sont suivi d'un exemple complet (section 3.3 page 89) où la structure et les textures sont obtenus à partir d'un échantillon d'images de référence.

## 3.1 Affinages structurels

### 3.1.1 Boîte de synthèse

Dans cet exemple, nous reprenons les trois vues de la boîte de synthèse décrite dans la section 3.6 page 37. Nous cherchons cette fois ci à ajuster la structure du modèle de l'objet par rapport à ces trois vues.

La figure 3.1 présente pour les trois vues de référence du cube une surimposition du modèle de départ en fil-de-fer. Les erreurs sur les dimensionnements de la boîte sont de l'ordre de 20% de la taille réelle, et les erreurs de positionnement et d'orientation sont de l'ordre de 10% de la taille de la boite.

La figure 3.2 page suivante présente quelques résultats intermédiaires (pour une seule des vues) des phases de recherche des paramètres extrinsèques et intrinsèques du modèle. Finalement, la figure 3.3 page suivante montre le résultat définitif.

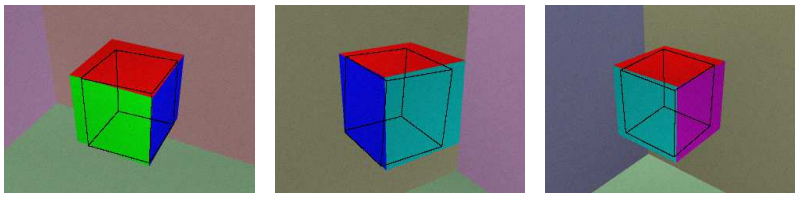


FIG. 3.1 – Les trois vues de références utilisées, avec en surimposition et en fil de fer le modèle dans son état initial

### 3.1.2 La boîte de Scotch

Dans cet exemple, nous reprenons la boîte de *Scotch* déjà vu précédemment, cette fois pour tester l'affinage structurel.

Nous partons donc d'un modèle imprécis de cette boîte – les trois dimensions de largeur, hauteur et profondeur – avec une position de départ elle aussi imprécise. La figure 3.4 page suivante montre pour deux vues de la boîte de Scotch la surimposition en fil de fer du modèle initial, avec ses dimensions et sa position originales.

Nous ne montrons pas, sur cet exemple, les résultats intermédiaires des phases d'optimisation des paramètres, car ceux-ci montrent un comportement identique à celui montré lors de l'exemple précédent. On constate au final (figure 3.5 page 87) une bonne écart visuelle, qui est hélas dans ce cas là la seule qui puisse être utilisée, les dimensions et la position exacte de la boîte n'étant pas connues – en tout cas de façon précise.

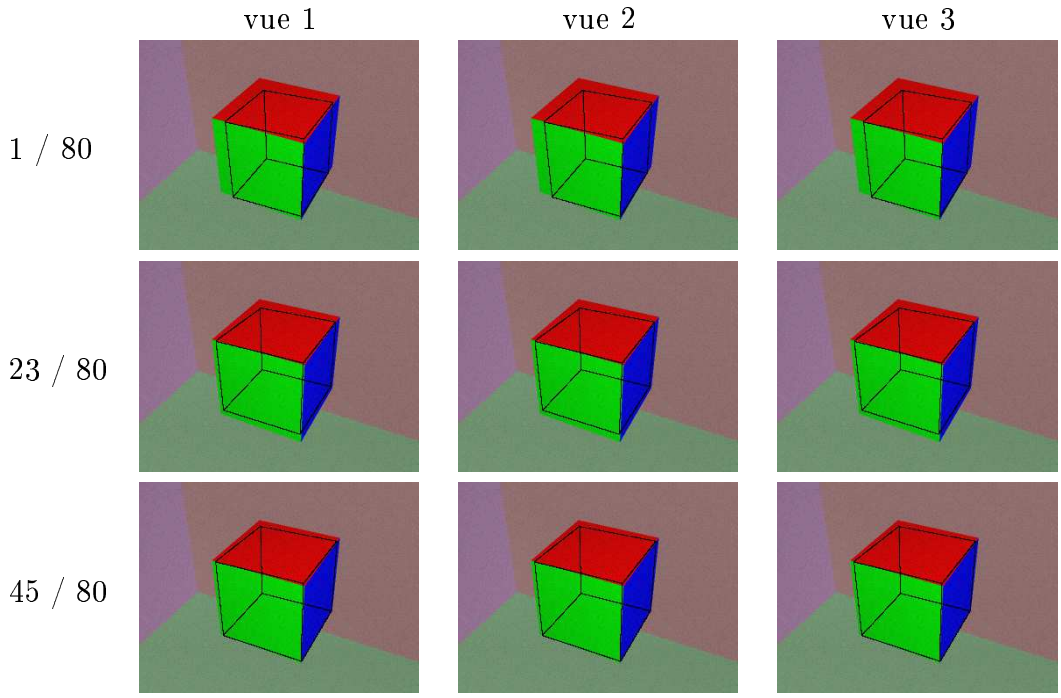



FIG. 3.2 – Pour une des vues, surimposition du modèle trouvé à différentes phases de l'auto-affinage, le numéro à gauche des images indiquant l'étape correspondante

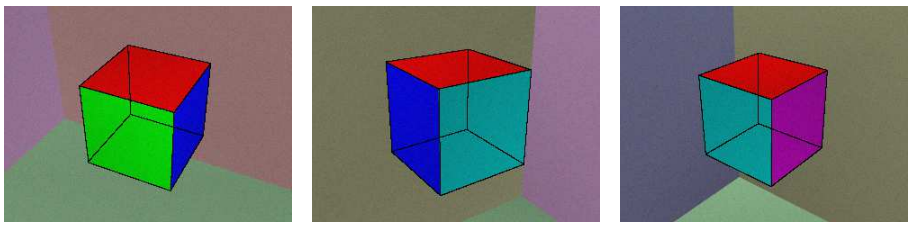


FIG. 3.3 – Surimposition du modèle définitif trouvé pour chacune des vues pour la boîte de synthèse


FIG. 3.4 – Les deux vues de référence de la boîte de *Scotch* avec en surimposition le modèle initial en fil de fer

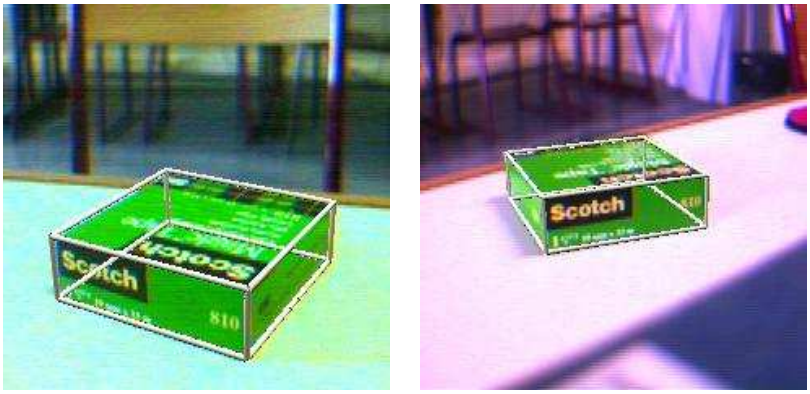



FIG. 3.5 – Surimposition du modèle définitif trouvé pour chacune des vues pour la boîte de *Scotch*

## 3.2 Extraction de textures : exemple simple

Nous présentons dans cette section un exemple simple des traitements effectués pour réaliser l'extraction des textures d'un objet.

Il s'agit de quatre vues d'un rectangle texturé, cette texture se composant d'un damier noir et blanc. La figure 3.6 montre ces quatre vues, parmi lesquelles deux sont des vues où le rectangle n'est pas entièrement visible.

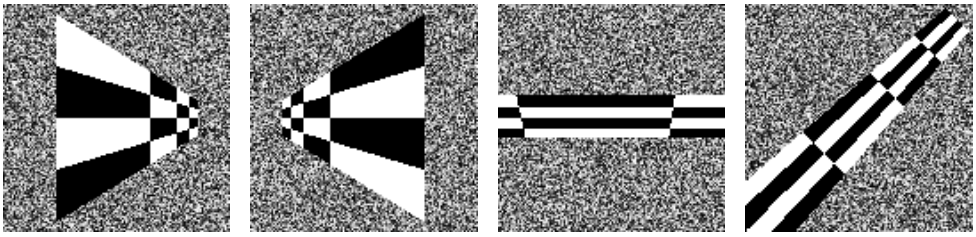


FIG. 3.6 – Quatre vues du rectangle dont on cherche à extraire la texture. Dans deux des vues, le rectangle est partiellement caché.

Le modèle correspondant est le même rectangle, mais sans aucune connaissance à propos de sa texture.

La figure 3.7 page suivante montre les textures obtenues pour chacune des vues de la figure 3.6, dans le même ordre. On remarque sur les deux premières textures une dégradation de la qualité sur le bord droit (respectivement gauche), correspondant à une perte de précision due à la distance des points de l'image. On constate également sur les deux dernières textures que les zones non visible dans les images ont une couleur rouge, correspondant à la couleur par défaut associée à l'objet (cette couleur à été choisie afin de bien se distinguer de la texture, et ne correspond bien sur pas à un choix intéressant dans ce cas).

La figure suivante 3.8 page suivante montre les les niveaux de confiance associés aux textures de la figure 3.7,dans le même ordre (les degrés de confiance varient de noir : confiance nulle à blanc : confiance maximum). On remarque que la confiance diminue

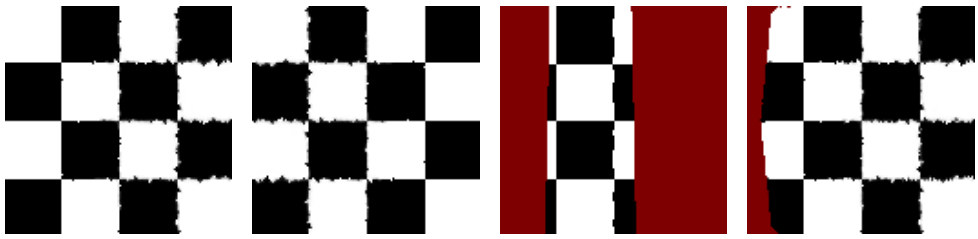


FIG. 3.7 – Les quatre textures extraites pour chacune des vues disponible. Les zones rouge sombre correspondent aux parties non visibles dans les vues correspondantes. On remarque sur les bords droite et gauche des deux premières textures les imprécisions dues à l'éloignement des pixels correspondant dans les images.

dans les zone peu précise, et que cette confiance devient nulle pour les zones non visible des deux dernières textures.

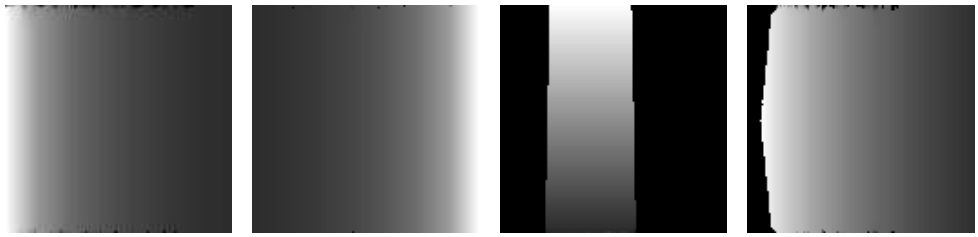


FIG. 3.8 – Les quatre carte des degrés de confiance, associées à chacune des textures extraites. Le contraste a été artificiellement augmenté pour plus de visibilité. La gamme des valeurs ne correspond donc pas exactement à la réalité.

La figure 3.9 montre la texture finale obtenue après la fusion des quatre textures issues de chacune des vues. En regard de ce résultat se trouve la texture originale ayant servie à générer l'objet. Il est à noter que dans cet exemple nous n'avons pas utilisé de source lumineuse, il n'y a donc pas de correction des cartes des textures par rapport à ce phénomène. Un exemple mettant en oeuvre ce point particulier est présenté plus loin.

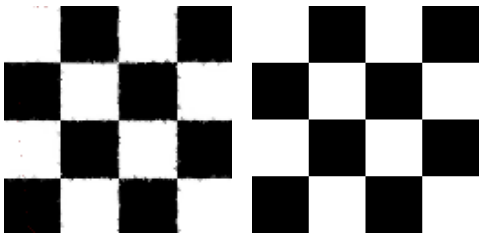


FIG. 3.9 – La texture finale après fusion des quatre textures par rapport aux cartes de confiance, avec en regard la texture de référence utilisée pour générer les vues de référence

## 3.3 Affinages complets : structure et textures

### 3.3.1 La boîte de *Scotch*

Nous avons déjà vu dans une section précédente l'auto-ajustement de la structure de la boîte de *Scotch* à partir de deux vues de celle-ci.

Nous complétons maintenant la définition du modèle géométrique de cette boîte en utilisant l'extraction de textures pour parfaire l'aspect visuel de ce modèle.

Ainsi, à partir de la géométrie – supposée correcte – trouvée lors de la phase d'auto-affinage, nous utilisons la rétro-projection afin d'extraire les textures apparentes dans les différentes vues et de les fusionner pour générer le modèle géométrique final.

Les deux vues de la boîte de *Scotch* utilisées se trouvent à la figure 3.4 page 86. La figure 3.10 montre les textures associées aux faces de la boite, obtenues à partir de notre système d'extraction. On constate qu'il n'y a pas de texture définie pour la face arrière et la face inférieure de la boite, ces deux faces n'étant pas visibles sur les images. La couleur par défaut de l'objet (vert) a donc été utilisé.

La figure 3.11 page suivante montre quand à elle la contribution des éclairages, calculé pour chacune des faces de la boite. On retrouve sur la figure 3.12 page suivante la carte des textures finale, une fois pris en compte la contribution des éclairages. On remarque dans cet exemple que les éclairages étant relativement uniformes cela ne génère pas beaucoup de variations. On remarquera également que lorsqu'une face n'est pas visible elle conserve sa couleur par défaut (face du dessous et face arrière).

La figure 3.13 page 91 montre deux vues de synthèse du modèle de cette boite, tel qu'il a été spécifié après les phases d'auto-affinage et d'extraction des textures. Dans ce cas précis, nous avons générer ces deux vues artificielles dans le même repère que les vues de référence, afin de pouvoir comparer visuellement celles-ci.


FIG. 3.10 – Texture obtenue après fusion des cartes de texture issues de chaque vue pour la boîte de Scotch

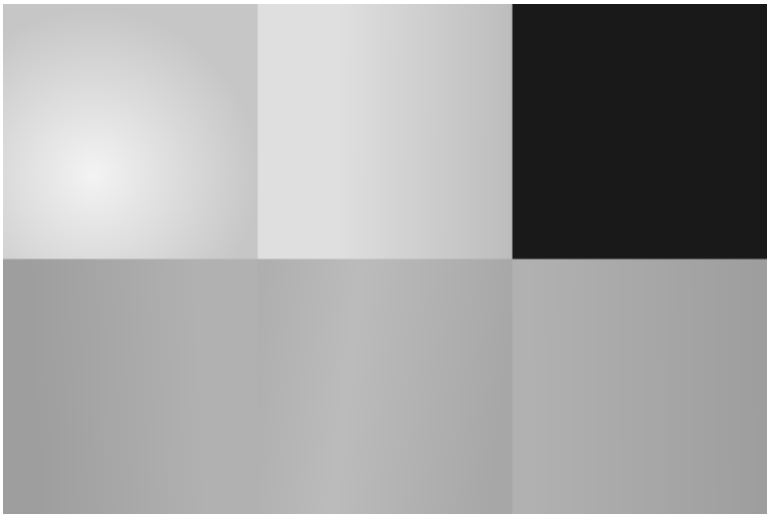


FIG. 3.11 – Carte des éclairages que reçoit la boîte de Scotch, mise en correspondance avec la carte des textures. Note : dans cet exemple le contraste de cette image a été augmenté pour des question de visibilité.


FIG. 3.12 – Texture obtenue après prise en compte de la carte des éclairages. L'éclairage étant relativement uniforme sur cet exemple, on ne remarque que peu de changements


FIG. 3.13 – Deux vues de synthèse du modèle final de la boîte de Scotch, chacun correspondant aux caractéristiques des vues réelles.

# Cinquième partie

# Résultats

# 1

# Séquences vidéo de synthèse

## Sommaire

Cette première partie des résultats se propose de valider le fonctionnement global de notre approche en travaillant à partir de séquences vidéo artificiellement générées. Ces séquences sont produites grâce à la synthèse d'image 3D, et nous permettent de tester spécifiquement des conditions particulières, parfois difficiles à mettre en œuvre dans la réalité.

La section 1.1 détaille les avantages de l'utilisation de séquences de synthèse. Un exemple simple – un objet non articulé – est ensuite présenté à la section 1.2. Nous montrons sur ce même exemple le comportement du système en présence d'occultations (section 1.3). La section 1.4 présente ensuite suivi d'un objet articulé relativement complexe : un bras robotisé. Dans la section 1.5, nous abordons le problème de l'influence du nombre de vues sur la précision obtenue, ainsi que sur le choix du positionnement des caméras. Enfin la dernière section présente quelques tests de performance de l'implémentation actuelle de notre système (section 1.6).

## 1.1 Vérité terrain et contrôle du contenu

Nous utilisons dans cette partie des séquences vidéo générées artificiellement. De nombreuses raisons nous ont motivé pour ce choix :

- les outils existants permettent relativement facilement de réaliser de tels films, alors qu'il est plus complexe et long de réaliser de vrai films ;
- le modèle 3D utilisé pour générer les séquences de synthèse peut directement être réutilisé comme modèle géométrique paramétré pour le suivi, permettant d'éviter une phase de mesures physique pour construire le modèle de l'objet à suivre ;
- les paramètres utilisés pour générer les séquences de synthèse sont connus, et il est donc possible de faire des calculs d'erreur sur les résultats des suivis ;
- une même séquence artificielle peut être reprise et modifiée afin de tester l'impact de tel ou tel élément de scène (éclairage, occultation...) tout en laissant le reste strictement identique.

Comme nous l'avons dit, la connaissance des valeurs des paramètres nous permet de tester la précision de notre approche sur des cas théoriques parfaits : strict identité entre la scène réelle et les images de synthèse générées. De même, l'utilisation de ces séquences de synthèse permet de partir d'une même scène en ne faisant varier qu'un élément à la fois, afin de mesurer son impact sur la précision. On peut ainsi suivre l'évolution de la précision lorsque l'on augmente les différences entre la scène filmée et ce que génère le moteur de rendu : anti-crénelage, bruit, objets inconnus dans la scène, erreurs sur le modèle de l'objet suivi...

## 1.2 Suivi d'un objet simple, non articulé

Dans cet exemple, la scène utilisée représente une chaise se déplaçant et tournant sur elle-même dans une pièce[2]. Cette scène de synthèse est filmée par deux caméras virtuelles, donnant des images de résolution $256 \times 256$, pour un total de 23 images par séquence. Les images sont générées avec un sur-échantillonage et sont retraitées afin d'y ajouter un bruit blanc (stationnaire, additif).

Les paramètres suivis ici sont les six permettant de définir l'état spatial d'un objet rigide, c'est-à-dire les trois paramètres de positionnement spatial et les trois angles définissant l'orientation de la chaise. La figure 1.1 montre quelques extraits de ces séquences pour les deux caméras, à divers instants.

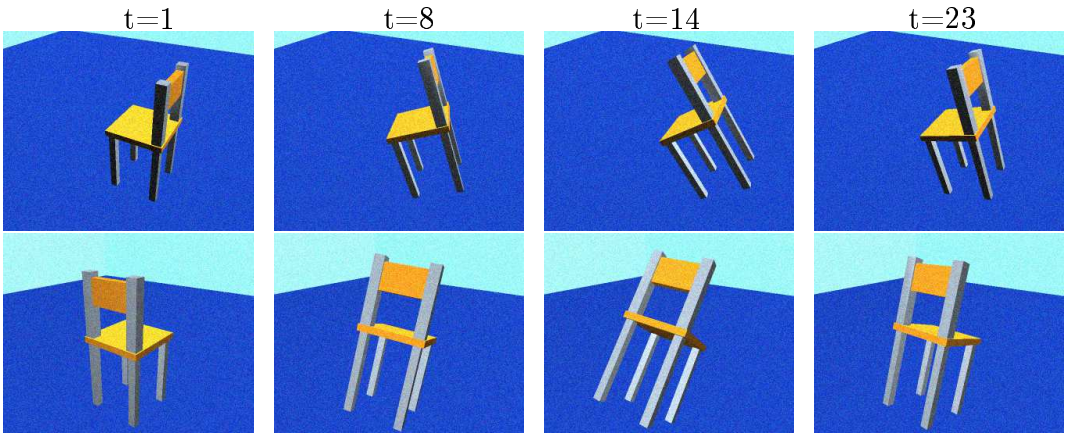



FIG. 1.1 – *Extraits des séquences de la chaise pour les deux caméras. La première ligne correspond à la première caméra et la deuxième ligne à la deuxième caméra. Les images en vis-à-vis vertical correspondent au même instant dans les séquences.*

Le modèle utilisé pour le suivi est celui ayant servi pour la génération des séquences de synthèse. Le système ne possède par contre aucune information relative au décors, et n'a aucune information sur le bruit et le sur-échantillonage utilisé. De plus il n'a qu'une définition imprécise des sources de lumière.

Disposant des valeurs théoriques des paramètres, il est aisé de calculer les erreurs entre les valeurs trouvées et les valeurs réelles. Nous présentons donc le résultat du suivi sous la forme d'un graphique où se trouve résumé pour chaque instant des séquences la précision (ou l'erreur) obtenue. Le graphique 1.2 décrit en ordonnées l'erreur obtenue pour chaque instant dans les séquences, en abscisses. De façon à être homogène, nous avons séparé l'erreur angulaire de l'erreur de position. Chacune d'elle correspond respectivement à l'erreur quadratique sur l'ensemble des paramètres angulaires et de position. Pour avoir un ordre d'idée pour les erreurs de positions, l'unité utilisée est la métrique de l'espace de synthèse, dans laquelle la chaise a pour dimensions $3 \times 3 \times 6$.

On constate que les erreurs restent très faibles au cours du temps, indiquant un suivi de bonne précision.

[2]Il faut considérer ici l'intérêt théorique de l'expérience car, en règle générale, peu de chaises effectuent des loopings dans les bureaux...

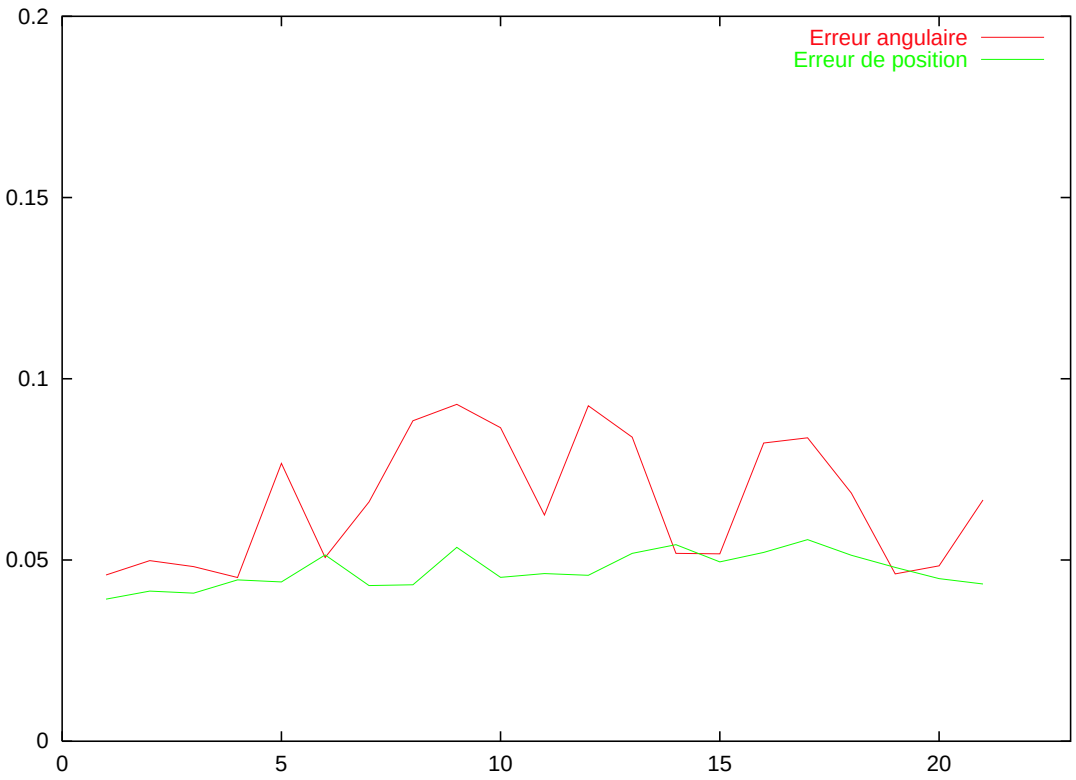



FIG. 1.2 – *Précision obtenue pour le suivi de la chaise. En ordonnées se trouve la précision obtenue aux différents instants des séquences, en abscisses. Les erreurs angulaires et le positions sont chacunes regroupées en des erreurs séparées.*

## 1.3 Résistance aux occultations durant le suivi

Nous reprenons dans cet exemple la séquence de la chaise utilisée à la section 1.2. Afin de mesurer l'impact des occultations sur notre approche, nous mettons la chaise "en cage", c'est-à-dire que nous insérons dans la scène servant à générer les séquences une cage dont l'épaisseur des barreaux peut être contrôlée.

Nous avons ainsi recréé les séquences précédemment décrites, avec des cages d'épaisseurs variables afin de tester des situations d'occultation plus ou moins marquées. La figure 1.3 montre (seulement pour l'une des caméras) des extraits des séquences pour trois niveaux d'occultations correspondant approximativement à 25%, 50% et 75% de la surface de la chaise occulté.

La figure 1.4 présente les résultats des différents suivis, sous la même forme que dans le précédent exemple : pour chaque instant des séquences (en abscisses) se trouve l'erreur angulaire correspondante (en ordonnées). Les trois courbes représentent l'erreur correspondant à chacun des trois niveaux d'occultation (25, 50 et 75%). Nous n'avons reproduit ici que l'erreur angulaire pour des raisons de lisibilité. L'erreur de position présente un comportement identique, et n'apporte donc ici aucune information supplémentaire.

On constate sur le graphique que les erreurs augmentent au fur et à mesure que le degré d'occultation augmente, ce qui est logique. Celle-ci reste dans des limites raisonnables et

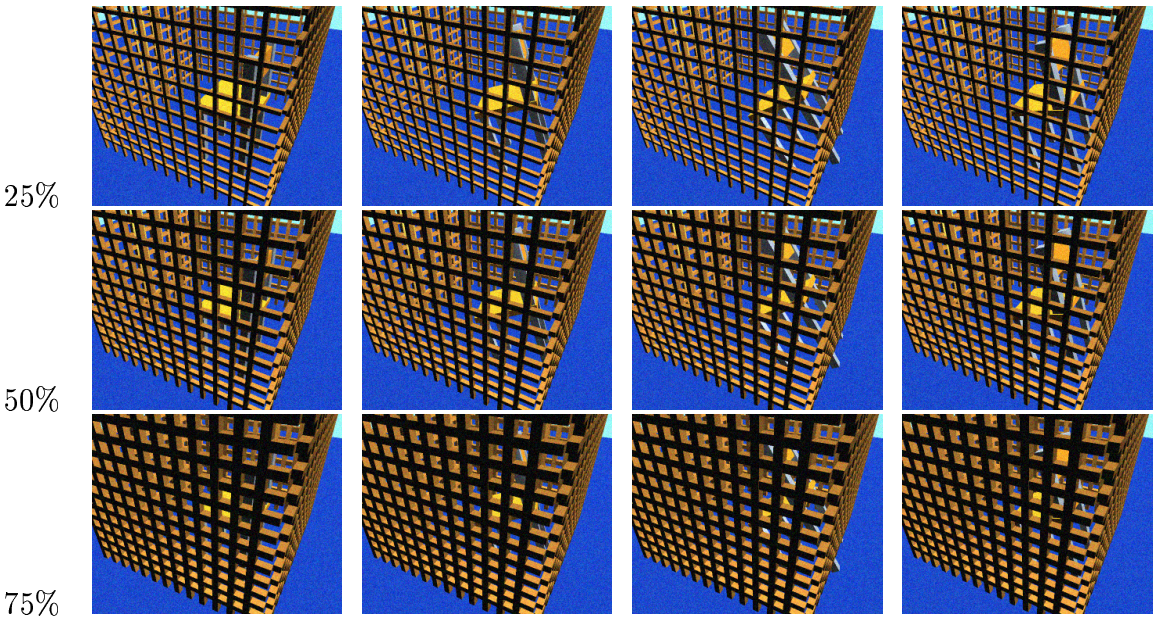



FIG. 1.3 – *Extraits des séquences de la chaise pour la première caméra. Chaque ligne correspond respectivement à un niveau d'occultation de 25%, 50% et 75% de la chaise, toujours pour les instants* $t = 1$, $t = 8$, $t = 14$ *et* $t = 23$.

surtout le suivi continu de s'effectuer correctement. Nous ne l'avons pas représenté ici, mais pour des niveaux d'occultation supérieurs le suivi ne fonctionne plus.

Notons que la "quantité" d'occultation supportable par notre système dépend de nombreux facteurs. L'important est de pouvoir voir des parties discriminantes de l'objet. Par exemple si l'on cache les bords d'un rectangle il est évident qu'il devient difficile de situer sa position avec précision, alors que si seuls les bords sont visibles cela est très facile. Il ne faut donc pas considérer les pourcentages d'occultation donnés ici comme une limite absolue du système, mais plutôt comme une indication du comportement global de notre méthode face aux occultations.

## 1.4 Objet articulé complexe

Dans cet exemple, nous avons créé une séquence de synthèse mettant en action un objet articulé relativement complexe : un bras articulé simplifié. Ce bras se compose d'un *bras*, d'un *avant-bras*, d'une *main* et de trois *doigts*, eux-mêmes composés de deux *phalanges* (voir figure 1.5). Le modèle correspondant dépend de 14 paramètres, se décomposants en :

- trois paramètres de position pour l'origine du bras ;
- trois paramètres de rotation pour l'origine du bras ;
- l'angle du *coude* ;
- l'angle entre la *main* et l'*avant-bras* (cette articulation n'ayant qu'une rotation possible) ;
- pour chaque *doigt*, les deux angles correspondants aux deux *phalanges*.

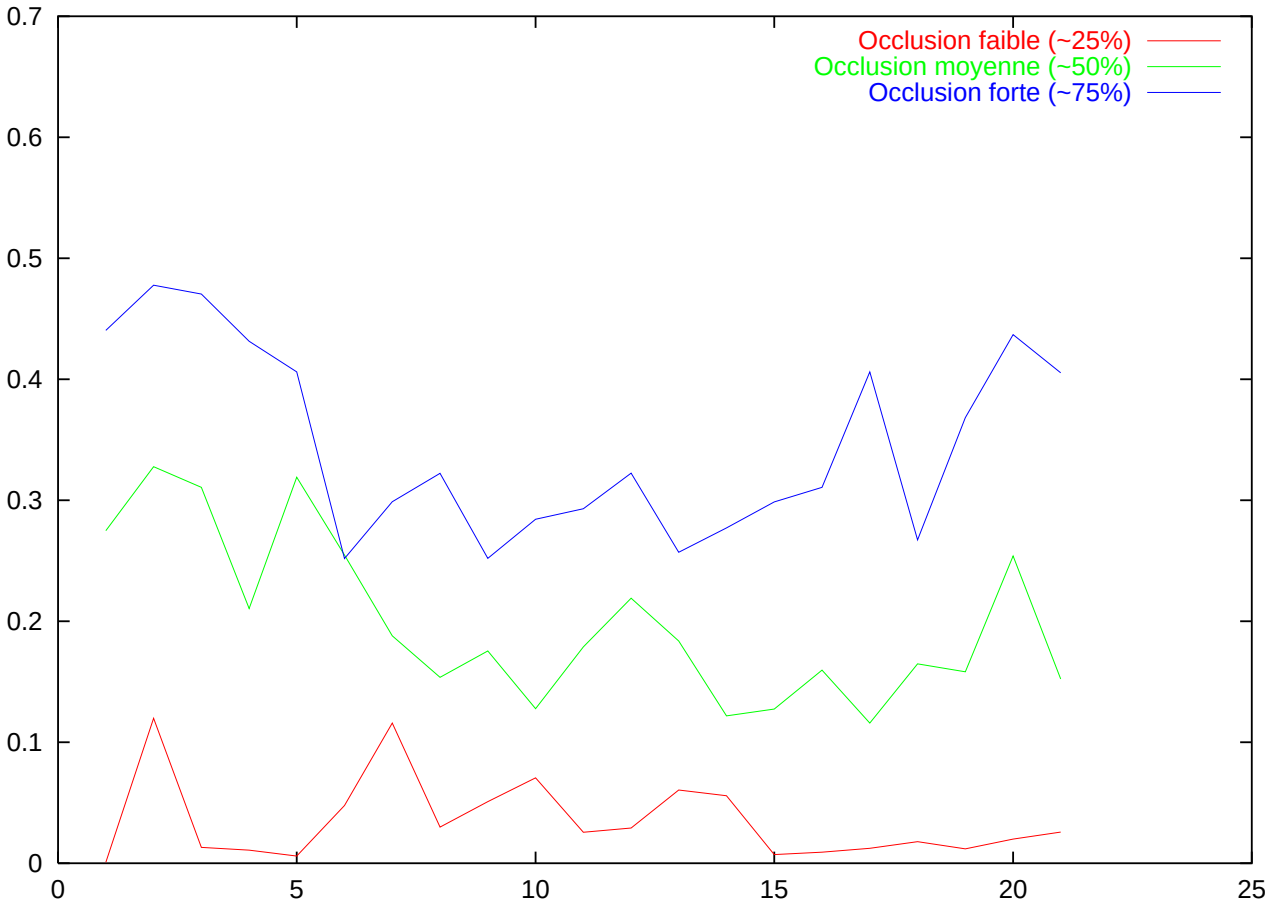



FIG. 1.4 – *Précision obtenue pour le suivi de la chaise, pour plusieurs niveaux d'occultation. En ordonnées se trouve la précision obtenue durant le suivi, correspondant aux différents instants des séquences, en abscisses. Seules les erreurs angulaires sont données ici par souci de lisibilité, mais le comportement des erreurs de positions est similaire.*

Cette scène artificielle est filmée par deux caméras. La résolution de chaque image est de $320 \times 256$, pour un total de 100 images par séquence. Le sur-échantillonage est utilisé, et un bruit blanc à été ajouté aux images. Au cours du temps, la position du bras reste relativement constante, mais tous les angles se modifient selon un mouvement complexe et de grande amplitude. La figure 1.5 page suivante montre des extraits des deux vues à différents instants des séquences.

Par la suite nous travaillons à partir du modèle exacte de l'objet, sans aucune information sur le décors, et avec des positions approximatives pour les sources de lumière de la scène. Bien entendu les vues de synthèse générées lors du suivi n'utilisent pas le sur-échantillonage et aucune information sur le bruit n'est connu.

La figure 1.6 montre les erreurs angulaires obtenues pour les différentes articulations. Les erreurs de position du bras ont été ignorées, car celles-ci sont très faibles. Le point important est ici de constater que la précision diminue lorsque l'on s'approche des arti-

culations des doigts (tout en restant correcte). Ceci est dû au fait que les doigts sont de petite taille et donc un écart de faible amplitude sur un doigt a un très faible impact en terme d'image, alors qu'une petite variation sur un angle du bras a une influence sur l'intégralité du bras.

Si une des caméras filmait la *main* en plus gros plan, des détails plus précis apparaîtraient et permettraient de diminuer les erreurs correspondantes.

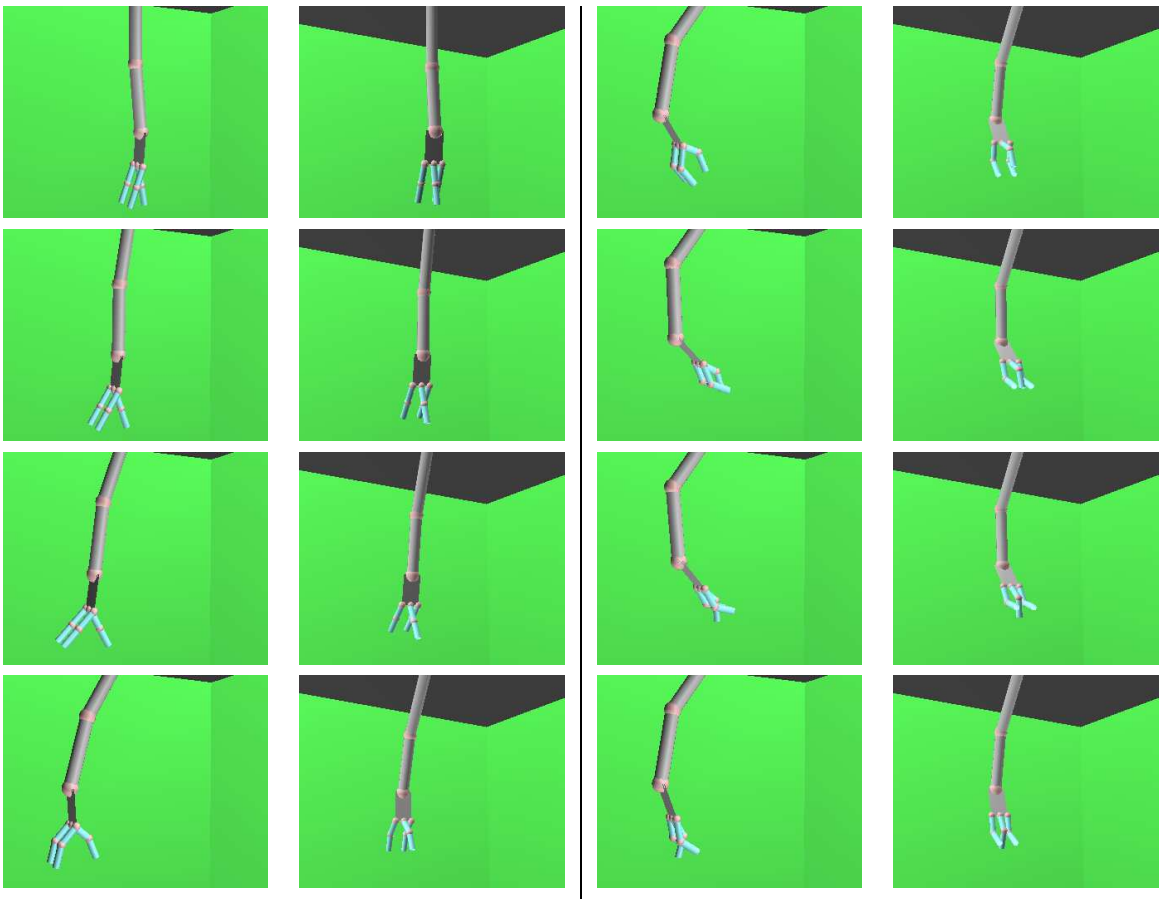


FIG. 1.5 – *Extraits des séquences du bras de synthèse. Première puis troisième colonne (de haut en bas) : extraits pour la première caméra. Deuxième puis quatrième colonne (de haut en bas) : extraits pour la deuxième caméra. Les images en vis-à-vis correspondent aux même instants pour les deux caméras.*

## 1.5 Influence du nombre et de la position des caméras sur la précision

Une question importante à se poser lors de la création des séquences vidéo est la suivante : combien faut-il de caméras filmant l'objet pour obtenir une précision correcte, et où placer ces caméras dans l'espace ?

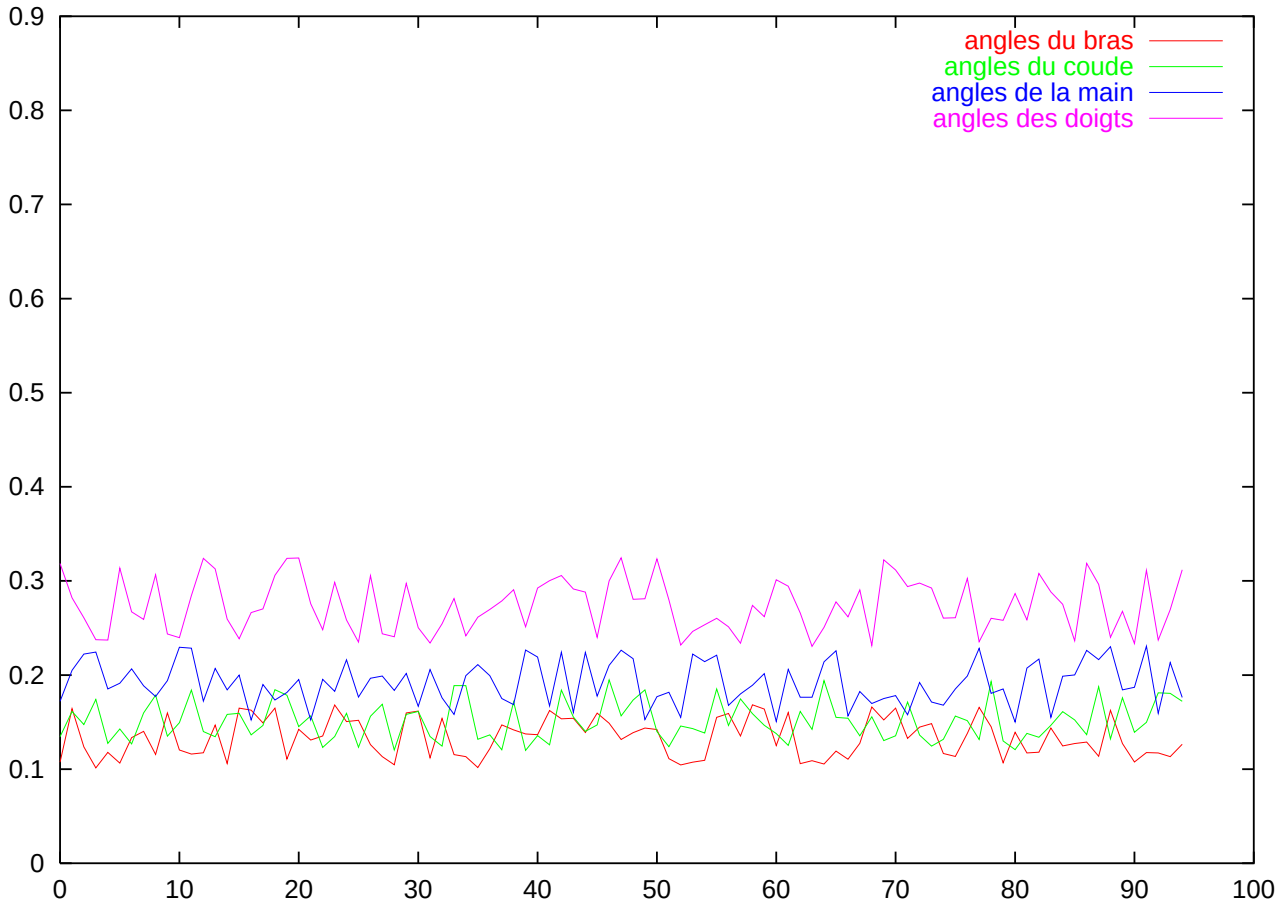



FIG. 1.6 – *Erreurs correspondant à chaque niveau d'articulation sur le bras : bras, coude, main et doigts (les erreurs de position ont été ignorées car très faibles). On remarque une augmentation des erreurs lorsque l'on s'approche des doigts, due à la petitesse des détails correspondants dans les images.*

Théoriquement, plus il y a de vues et plus la précision atteignable est grande. En pratique on ne dispose pas d'une infinité de caméras, ne serait-ce que pour des questions de coût.

En fait il n'est pas possible de répondre de façon formelle à cette question, car la précision dépend de l'objet suivi lui-même et de ses mouvements possibles. Par exemple un objet rigide contraint à se déplacer sur un plan ne nécessite de disposer que d'une caméra, si elle est bien placée.

Il est cependant possible de dégager quelques heuristiques quant au nombre et à la place des caméras :

- sauf restriction particulière sur les mouvements de l'objet – déplacement sur un plan par exemple – un minimum de deux caméras s'impose afin de contraindre spatialement l'objet ;
- trois caméras placées (plus ou moins) suivant trois axes perpendiculaires assurent

un bon suivi d'un objet rigide ;
- si les caméras sont fixe, il faut obtenir le cadrage le plus serré possible sans que l'objet sorte du champs ;
- il est préférable de "voir" l'objet du coté où il est le plus caractéristique, afin d'éviter les ambiguïtés ;
- pour des objets articulés, il faut éviter qu'une partie mobile de l'objet soit cachée dans toutes les vues.

Cette liste n'est bien entendu pas exhaustive. Comme nous l'avons dit, le choix du nombre et des positions des caméras va dépendre de l'objet lui-même, des mouvements qu'il va effectuer ainsi que des contraintes physiques de la scène où il évolue.

En dehors de ces critères, la résolution des images influence également la précision finale. Ceci s'explique facilement par le fait qu'une résolution plus élevées offre plus de détails et permet de réduire certaines imprécisions dues à la petitesse des objets. Toutefois le gain de précision atteint rapidement un plafond lorsque la résolution augmente, et de toute façon celle-ci est limité par les capacités des caméras – limites physiques et de coût. La figure 1.7 montre un exemple de l'évolution de la précision atteignable en fonction de la résolution des images sur l'exemple de synthèse de la chaise (voir section 1.2).

Notons au passage que seule l'allure globale de cette courbe est à retenir, les valeurs exactes qu'elle prend dépendant en réalité de nombreux facteurs tels que l'objet lui-même, la qualité des images, le nombre de vues...

### 1.5.1 Suivi à partir d'une seule caméra

Dans tous les exemples que nous avons montré, nous utilisons un minimum de deux caméras. Au delà des aspects de précision, la question se pose de savoir s'il est possible d'effectuer un suivi à partir d'une seule vue.

Précisons en premier lieu qu'avec une seule caméra l'objet à suivre doit être "caractéristique" dans la vue disponible. En particulier pour des objets articulés l'ensemble des éléments mobiles doivent être visible. Précisons également que la précision des paramètres ayant des effets dans l'axe de la caméra sera *a priori* faible, la différence visuelle générée par exemple par une translation selon cet axe étant faible.

Au delà de ces problèmes, qui vont principalement affecter la précision, se pose un problème plus fondamental : la présence d'un optimum pour la fonction d'écart différent de celui recherché. Cet optimum est très simple à décrire sur un exemple : supposons que nous cherchons à trouver la position et l'orientation d'un rectangle dans l'espace, disons une feuille de papier blanc. La vue de référence va montrer cette feuille dans une position et une orientation donnée, et l'on voudrait trouver les valeurs des paramètres correspondant.

Un optimum consiste à placer la feuille à l'infini (par rapport à l'axe de la caméra). Les vues de synthèse vont présenté un objet réduit à un pixel. Il suffit alors de trouver sur la vue de référence un pixel lui correspondant et l'écart sera minimum.

Toute la question est de savoir s'il est possible de passer des valeurs initiales des paramètres à cette situation dégénérée.

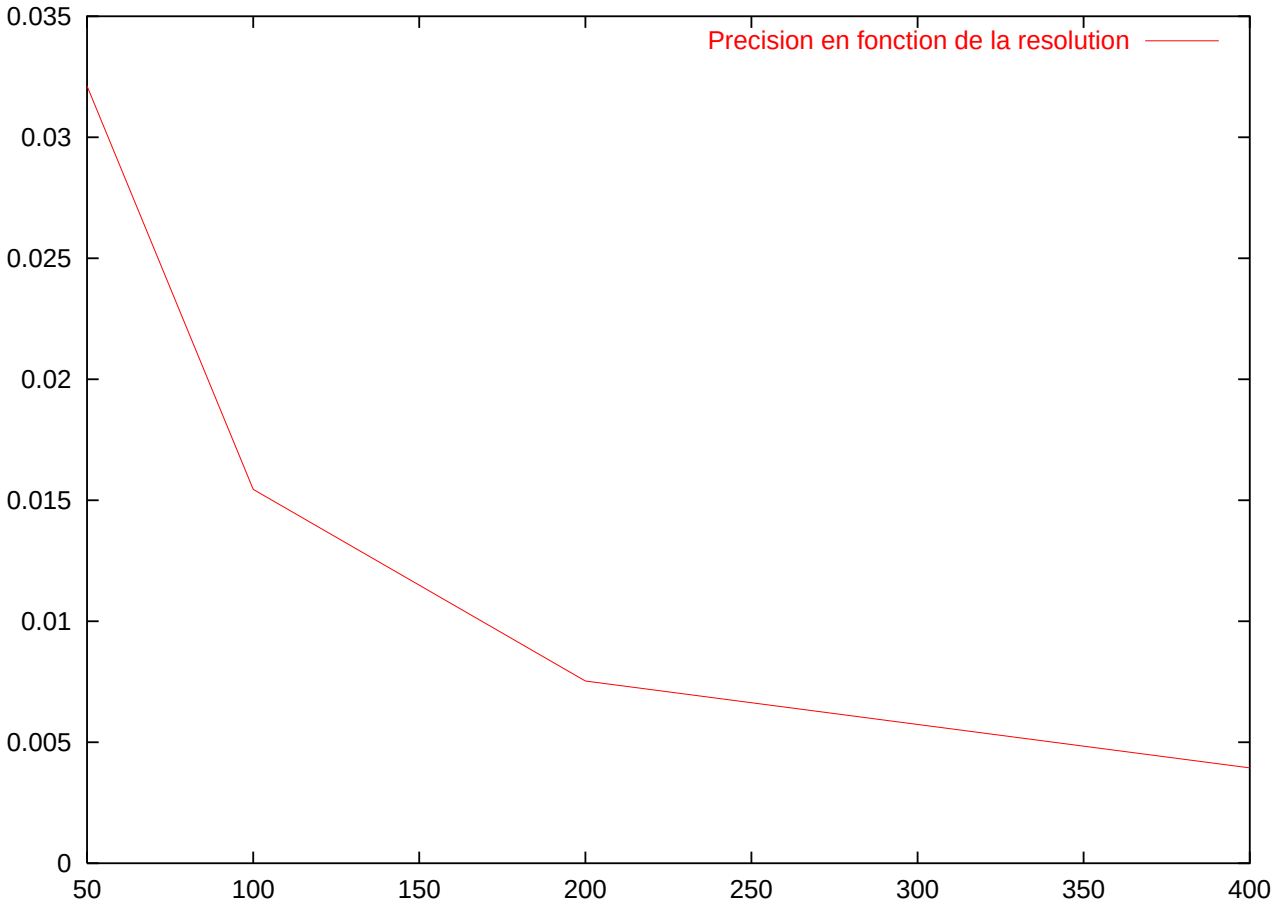



FIG. 1.7 – *Évolution de la précision atteignable (en ordonnée) en fonction de la résolution des images d'entrée (en abscisse), pour l'exemple de la chaise. On constate une très rapide asymptote, montrant que l'augmentation de la résolution doit être limité à des valeurs raisonnables.*

En pratique cela dépend de la forme de l'objet. Dans le cas d'un objet convexe (comme c'est le cas de notre feuille) le recule de la feuille de synthèse ne rencontre aucun obstacle lors de la phase d'optimisation. Pour une feuille de couleur unie on peut en effet trouver une partie de celle-ci qui ressemble également à une feuille. L'optimisation peut ainsi réduire étape par étape la taille apparente de la feuille, conduisant à la situation dégénérée décrite plus haut (voir figure 1.8).

Si l'objet à une topologie plus complexe ce type d'effet peut être impossible. Si la feuille possède un trou en son centre, le recul de celle-ci va générer une zone au niveau du trou où la correspondance visuelle sera mauvaise, interdisant aux paramètres de s'aventurer dans ces zones (voir figure 1.8).

Notons au passage qu'à partir de deux caméras un recule dans une vue devient une translation dans une autre vue et va donc correspondre à une mauvaise correspondance visuelle, empêchant ce type de situation de se produire.

En conclusion, la possibilité de suivre les valeurs de paramètres correspondants à

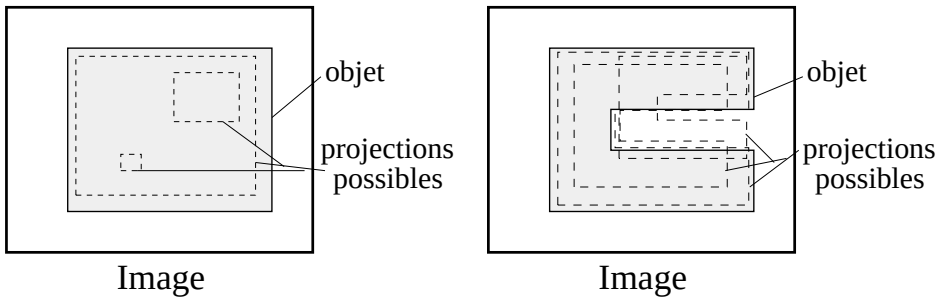



FIG. 1.8 – *À gauche, un exemple d'objet simple pouvant amener à une situation dégénérée : les projections du modèle (en pointillés) ont une mesure d'écart similaire à la projection valide. À droite, un exemple d'objet simple pour lequel la situation ne peut dégénérer. Le "trou" dans l'objet rend les projections du modèle (en pointillés) moins intéressantes que la projection valide.*

des déplacements tridimensionnels avec une seule vue dépend de l'aspect de l'objet à suivre. Pour des objets convexes et dont les textures sont peu discriminantes la possibilité de tomber sur un optimum dégénéré est grande. Notons également que le raisonnement précédent suppose des paramètres correspondent à des déplacements 3D. Si un objet vu de dessus se déplace sur un plan perpendiculaire à l'axe de la caméra, aucun paramètre ne pourra agir dans l'axe de vue, éliminant de fait ces situations dégénérées.

## 1.6 Vitesses de traitement

Bien que nous ne nous soyons pas placé dans une optique de traitements en temps réel, il est intéressant de connaître la vitesse que peut atteindre notre système, ainsi que les possibilités d'accélération possible.

Notre système se décompose en plusieurs modules. Le principal, qui va faire appel à tous les autres, est celui se chargeant de la recherche des paramètres : le module d'optimisation. L'algorithme utilisé est d'une complexité moyenne de $n.ln(n)$, où $n$ est le nombre de paramètres.

À chaque itération de l'algorithme, celui-ci va faire appel à la mesure d'écart pour évaluer les nouveaux paramètres générés. Pour chaque vue, la mesure d'écart effectue une synthèse d'image correspondant aux paramètres à évaluer, via le moteur de rendu. Elle va ensuite effectuer la comparaison avec l'image de référence correspondante. La complexité de la synthèse d'image est difficile à évaluer. Elle dépend de la méthode utilisée (ray-tracing, radiosité, surfacique...). Dans notre cas, nous utilisons OpenGL, ce qui permet l'utilisation de matériel dédié relativement bon marché, les cartes accélératrices 3D. Celles-ci réduisent le temps nécessaire au rendu d'un facteur très élevé.

La comparaison entre l'image de synthèse et l'image réelle est directement proportionnelle au nombre de pixels dans les images.

Si $V$ est le nombre de vues, $N$ le nombre de pixels dans les images et $n$ le nombre

de paramètres, la complexité globale est donc : $n.ln(n).V.N$. Il est à noter que nous ne tenons pas compte ici de la complexité liée au rendu 3D, celle-ci étant relativement difficile à préciser car elle dépend des opérations mises en œuvre : textures, ombres, transparences... celles-ci dépendant de l'objet et de la scène. Pour des systèmes de type OpenGL, on peut considérer que la complexité du rendu est proportionnel au nombre de facettes dans la scène. Ce nombre étant constant pour un objet donné, nous négligerons cet aspect.

Concrètement cela donne sur un Pentium à 600MHz, avec 512Mo de mémoire et sans accélération 3D un temps moyen de traitement de 1,5 minute par étape pour des images de résolution $320 \times 240$ et trois caméras, avec six paramètres à trouver.

De façon évidente, le temps de traitement est linéaire par rapport au nombre de caméras par rapport au nombre de pixels dans les images.

Il est possible de réduire ce temps en utilisant une carte accélératrice pour le rendu, réduisant le temps utilisé par la synthèse d'image d'un très grand facteur (dépendant des caractéristiques de la carte 3D). Il est également possible d'utiliser la parallélisation des traitements pour accélérer un certain nombre de tâches, telles que les comparaisons entre images qui s'y prêtent facilement.

# 2

# Séquences vidéo réelles

## Sommaire

## 2.1 Objets non articulés

### 2.1.1 La boîte de Scotch

Nous reprenons dans cet exemple la boite de *Scotch* précédemment vue. La figure 2.1 montre quelques extraits des séquences vidéo utilisées. Celles-ci proviennent de deux caméras (type *webcam*), avec des résolutions de $320 \times 240$ par image, pour un total de 15 images. Pour faciliter l'animation de la boite, les séquences ont été prises image par image.

FIG. 2.1 – *Extraits des séquences de la boîte de Scotch pour les deux caméras. La première ligne correspond à la première caméra et la deuxième ligne à la deuxième caméra. Les images en vis-à-vis vertical correspondent au même instant dans les séquences.*

Le suivi a été réalisé en utilisant le modèle présenté dans la section 3.3 page 89. Les paramètres à suivre sont les trois paramètres de position et les trois paramètres de rotation définissant l'état spatial de l'objet. L'initialisation des valeurs des paramètres a été réalisée manuellement. Pour effectuer le suivi, notre système dispose uniquement des images issues de la vidéo et du modèle présenté ci-dessus.

La figure 2.2 montre le résultat de ce suivi. Comme dans les exemples précédents nous superposons aux images réelles une vue de synthèse de l'objet en utilisant les valeurs des paramètres trouvés. Un rendu en fil de fer est utilisé pour permettre une visualisation plus claire.

### 2.1.2 Le livre

Dans cet exemple nous déplaçons et tournons un livre. La scène a été filmées par deux caméras dont les caractéristiques sont : INDYCAM, environ 5 images par seconde, résolution de $320 \times 240$. Les deux séquences correspondent à une centaine d'images, le cadrage temporel entre elles étant réalisé à la main. La figure 2.3 montre une vue de synthèse du modèle du livre utilisé pour le suivi. Ce modèle a été réalisé en utilisant les techniques que nous avons décrites dans la partie IV page 71.

Les paramètres suivis dans cet exemple sont les trois paramètres de position $x$, $y$, et $z$ ainsi que les trois paramètres de rotation $r_x$, $r_y$ et $r_z$. Les résultats sont présentés à la figure 2.4, où l'on peut voir sur les colonnes 1 et 3 de haut en bas des extraits d'une des deux séquences vidéo, et sur les colonnes 2 et 4 le résultat correspondant aux images juste

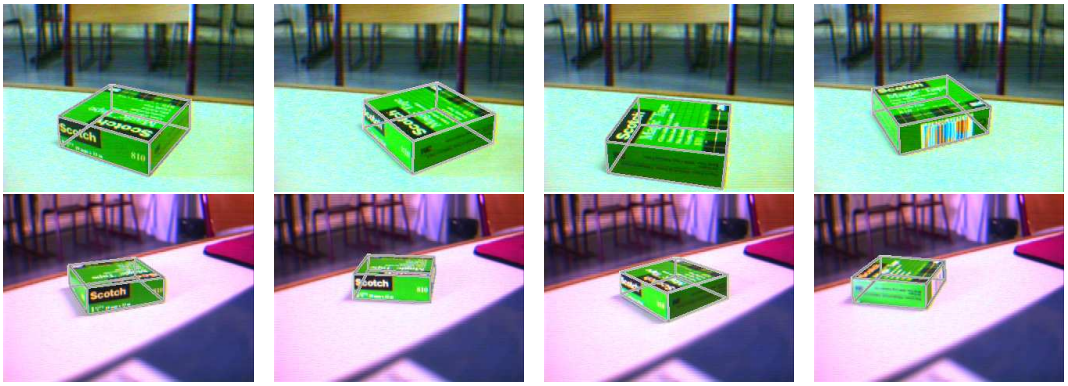


FIG. 2.2 – *Résultat du suivi pour la boîte de Scotch. À chaque image vidéo nous superposons un rendu de synthèse de l'objet en utilisant les paramètres trouvés par notre système.*

à gauche. Les résultats sont présentés sous forme de surimposition du modèle du livre, sous forme fil-de-fer pour des raisons de visibilité, par dessus les images originales.

On remarque un bon suivi au cours du temps, le livre de synthèse restant bien "collé" au livre original. En particulier on peut constater dans les dernières images que le suivi reste stable même en présence d'une occultation relativement forte (la main passant devant le livre).


FIG. 2.3 – *Vue de synthèse du modèle de livre utilisé pour le suivi. Seule la couverture possède une texture, les autres faces du livre étant de couleurs unies.*

## 2.2 Objets articulés

### 2.2.1 Simple articulation

Dans cet exemple nous déplaçons un objet composé de deux pans rectangulaires, articulés entre eux par une rotation selon l'axe perpendiculaire aux rectangles. Au cours des séquences l'angle entre les deux parties se modifie. La scène a été filmées par deux

FIG. 2.4 – *Colonnes 1 puis 3 : vues réelles pour l'une des deux caméras, réparties au long de la séquence vidéo. Colonnes 2 puis 4 : surimposition du modèle du livre en fil-de-fer à partir des paramètres trouvés, correspondant aux images initiales à leur gauche*

caméras dont les caractéristiques sont : INDYCAM, prise image par image, résolution de $320 \times 240$.

Les deux séquences correspondent à 94 images chacune, le cadrage temporel entre elles étant déjà réalisé par le processus de prise image par image. La figure 2.5 montre une vue de synthèse du modèle de l'objet utilisé pour le suivi. Ce modèle a été réalisé par simples mesures de l'objet, celui-ci étant d'une géométrie très simple.

Les paramètres suivis dans cet exemple sont les trois paramètres de position $x$, $y$, et $z$ ainsi que les trois paramètres de rotation $r_x$, $r_y$ et $r_z$ de l'objet, plus l'angle $\alpha$ entre les deux pans. Les résultats sont présentés à la figure 2.6, où l'on peut voir pour chaque caméra des images issues des séquences, avec en regard (à droite dans chaque colonne) la surimposition du modèle de l'objet sur l'image originale, le modèle étant positionné grâce aux valeurs des paramètres trouvés lors du suivi.

On constate que l'objet de synthèse adopte précisement les déplacements de l'objet réel au cours du temps.

FIG. 2.5 – *Vue de synthèse du modèle d'objet articulé utilisé pour le suivi.*

### 2.2.2 Objet complexe

Nous avons appliqué notre approche au suivi d'une main en mouvement. Dans cet exemple deux caméras filment les mouvements d'une main humaine. Les images ont une résolution de $320 \times 240$, et il y a un total de 10 images par séquence (ces séquences ont été prises image par image). La figure 2.7 montre quelques extraits de ces deux séquences.

Le modèle de main utilisé comporte un total de 26 paramètres à suivre :

- trois paramètres de position du poignet ;
- trois paramètres d'orientation du poignet ;
- pour chaque doigt, les angles de chaque phalange (trois au total) plus l'angle d'ouverture du doigt par rapport au plan de la main.

Nous avons réalisé un modèle générique grossier de cette main, où chaque phalange est représentée par un cylindre et la main elle-même par un parallélépipède. On peut voir à la figure 2.8 deux vues de synthèse de ce modèle correspondant aux premières images des séquences.

La figure 2.9 montre le résultat du suivi. Nous avons cette fois utilisé un modèle différent pour le rendu : une représentation sous forme de squelette (anatomiquement peu conforme, mais suffisant pour notre propos). Nous superposons les vues de synthèse de ce squelette aux images réelles de la main. On constate que le suivi global est correct (au niveau de la main elle-même). La précision obtenue est moins bonne pour les phalanges, surtout en bout de doigt. Ceci a plusieurs causes :

- comme dans l'exemple du bras robot, la précision pour les petits éléments de l'objet (les phalanges terminales) est plus faible que pour les gros éléments (la main) ;
- la position des articulations des doigts dans le modèle n'est pas très précise par rapport à la main réelle. Cela génère des imprécisions lorsque les doigts se plient ;
- lorsque les doigts sont pliés ils peuvent être cachés par le reste de la main (comme c'est le cas sur les dernières images), provoquant une baisse de précision ;
- la géométrie "interne" de la main est difficile à définir précisement, en particulier la position des premières articulations des doigts, car celles-ci sont sous la peau ;
- les doigts sont très peu contrastés et la mesure d'écart est très "plate" pour les paramètres concernés.

FIG. 2.6 – *Suivi d'un objet articulé simple. La partie gauche de la figure correspond à la première caméra et la partie droite à la seconde caméra. Dans chaque partie se trouve à gauche l'image vidéo originale et à droite la même image avec le modèle positionné grâce aux paramètres trouvés.*

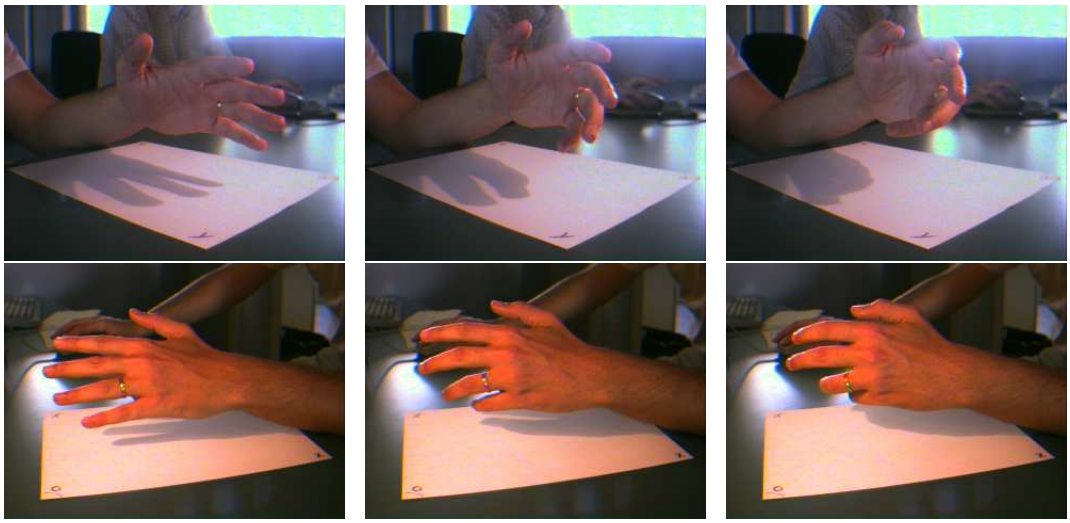


FIG. 2.7 – *Suivi d'un objet articulé complexe. La première et la deuxième ligne correspondent respectivement aux images issues de la première et la deuxième caméra filmant la main. De gauche à droite les images correspondent au début, au milieu et à la fin des séquences.*

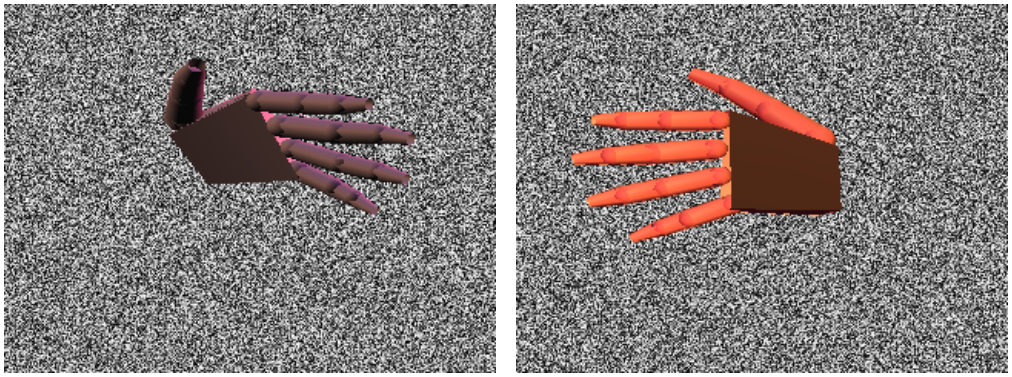


FIG. 2.8 – *Modèle géométrique de la main. L'image de gauche correspond au repère de la première caméra et celle de droite au repère de la deuxième caméra. Ce modèle a été affiné en combinant ajustement manuel et ajustement automatique, pour les paramètres intrinsèques et la couleur de la peau.*

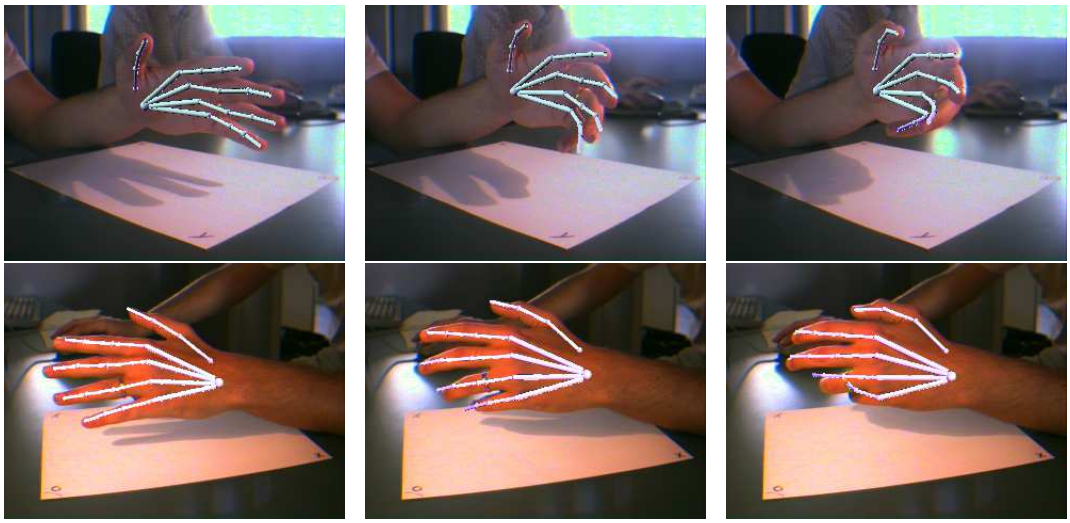


FIG. 2.9 – *Résultat du suivi. Les images sont les mêmes que dans la figure 2.7, avec en surimposition un modèle "squelette" de la main. On remarque que le suivi global est bon, mais que la précision au niveau des phalanges terminales des doigts est moins bonne.*

# 3

# Applications

## Sommaire

Nous présentons dans cette section une application possible du suivi de paramètres de modèle géométrique : la réalité augmenté.

Si l'on dispose des valeurs des paramètres d'un modèle d'objet dans un espace donné, cela nous donne un grand nombre d'informations sur l'objet qu'il représente. En particulier il est possible de connaître la projection de cet objet dans chacune des vues réelles, puisque l'on dispose des paramètres de calibrage des caméras. On peut ainsi "ajouter" de l'information dans ces scènes réelles. En pratique la façon dont nous avons présenté nos résultats dans les sections précédentes est une forme de réalité augmentée. Nous ajoutons aux images brutes une image de synthèse de l'objet suivi, sous forme d'une représentation en fil-de-fer. Cela peut être considéré comme une mise en avant de l'objet, puisque l'on augmente ainsi sa visibilité.

Il est bien évidement possible de transposer un grand nombre d'informations, telles que :

- la géométrie. Une vue fil-de-fer permet de se rendre compte de la géométrie cachée de l'objet (l'arrière de celui-ci, par exemple) ;
- la visibilité. Une augmentation locale du contraste, ou le surlignage des contours permet d'augmenter la visibilité de l'objet ;
- l'historique. Une "trace" de l'objet dans ces précédentes positions peut aider à la compréhension de ses mouvements ;
- la prédiction. Une analyse des variations des paramètres peut permettre d'anticiper certains mouvements, et d'indiquer ceux-ci sous une forme visuelle.

De notre coté nous avons voulu exploiter une autre possibilité : celle de modifier l'objet tel qu'il est visible au départ. Possédant ses caractéristiques 3D, et par projection ses caractéristiques 2D, il est possible d'effectuer une projection du modèle dans les images en utilisant un modèle "altéré".

C'est ce que nous présentons dans cette partie. Il y a toutefois une limite à cela : la connaissance des paramètres du modèle ne nous donne aucune information sur son environnement, et en particulier sur d'éventuels objets cachant celui-ci. Or si l'on ignore une éventuelle occultation, la projection du modèle altéré va se faire par dessus l'occulteur. Pour prendre en compte ce problème nous avons utilisé les informations que fournis note système pour prévenir ce types de situation, ce qui sera développé dans la section 3.2.

## 3.1 Réalité augmentée

Nous montrons dans cette section deux exemples de réalité augmentée, à la limite des problématiques des trucages vidéo, comme l'incrustation / substitution dans des scènes réelles.

Dans le premier exemple nous ajoutons des objets sur un plateau se déplaçant dans l'espace (section 3.1.1). Dans le deuxième exemple nous modifions l'apparence d'un objet dans la scène dont il est issu (section 3.1.2).

### 3.1.1 Insertion d'objets dans une scène

Dans cette séquence un opérateur déplace un plateau (ici représenté par une boîte en carton). Nous effectuons le suivi de ce plateau grâce à notre méthode. Nous obtenons ainsi l'état spatial de ce plateau au cours du temps. Il est alors possible de connaître à tout moment la position de la surface supérieure de ce plateau, et ainsi d'y "poser" des objets virtuels. Ces objets vont alors suivre les déplacements du plateau comme s'ils étaient réellement posés dessus.

La figure 3.1 présente ce résultat. Sur la première ligne nous voyons des vues provenant de l'une des deux caméras filmant l'objet. Les caractéristiques de ces séquences sont : *webcams*, résolution de $320 \times 240$, 100 images par séquence. La deuxième ligne montre ces mêmes images dans lesquelles nous avons "posé" trois objets sur le plateau : deux verres et un cube. Ceux-ci se déplacent solidairement avec le plateau au cours de ses mouvements.

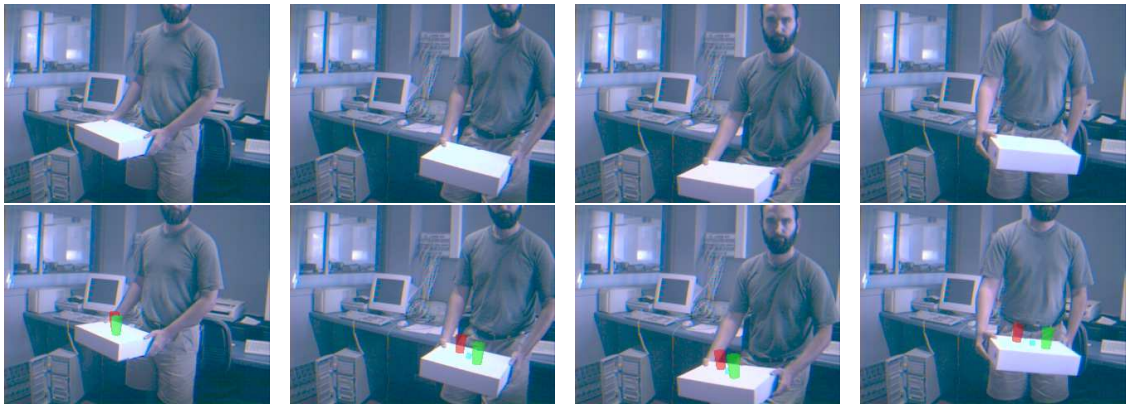


FIG. 3.1 – *La première ligne présente de gauche à droite des extraits de la séquence vidéo utilisée, pour la première des deux caméras. La deuxième ligne reprend les même images en y ajoutant deux verres et un cube, ceux-ci étant "posés" sur la représentation du plateau (le modèle et ses paramètres valués).*

### 3.1.2 Modification de l'objet suivi

La séquence test que nous avons utilisé ici est celle du livre vue précédemment. À partir des résultats obtenus lors du suivi (section 2.1.2) nous utilisons les informations de calibrage des caméras ainsi que le positionnement spatial du livre pour projeter dans les images un autre modèle de synthèse que celui utilisé lors du suivi (voir figure 3.2).

Dans l'exemple que nous avons réalisé, nous ajoutons au livre un chapeau, un nez et deux yeux, et ce "personnage" se trouve lié dans ses déplacements au livre, comme s'il en faisait partie. La figure 3.3 montre quelques images issues de cette séquence modifiée, pour la même caméra que dans l'exemple de suivi initial.

On constate dans les dernières images le problème que nous avions soulevé précédemment, qui est que les objets cachant le livre ne peuvent être pris en compte car le système

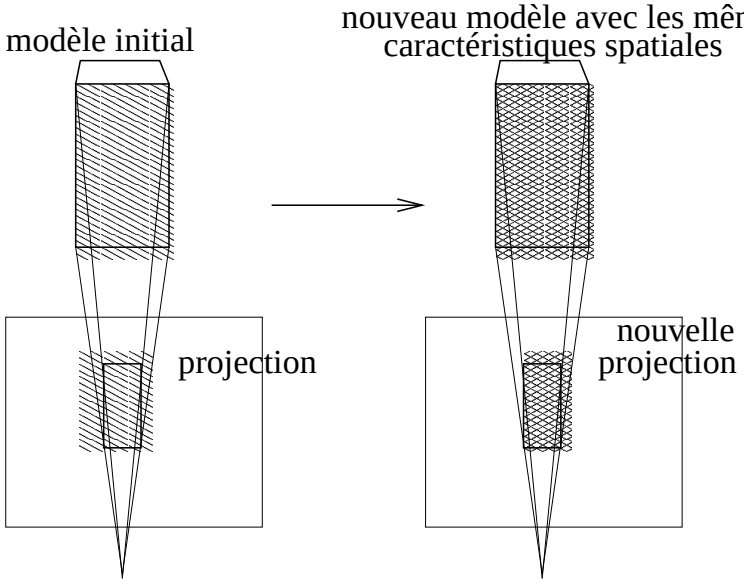



FIG. 3.2 – *Utilisation des informations de projection pour modifier l'apparence de l'objet suivi. En projetant un nouvel objet grâce à la même caméra virtuelle, celui-ci apparaît "par dessus" l'ancien.*

ne possède pas d'information sur eux (ici la main qui passe devant le livre mais apparaît derrière le nez).

## 3.2 Gestion des occultations

Nous cherchons à détecter les occultations de l'objet suivi afin d'en tenir compte lors de l'incrustation d'éléments extérieurs dans la scène originale.

Le problème est le suivant : comment détecter pour chaque pixel de l'objet initial si l'on voit l'objet ou bien si un autre objet se trouve devant et le cache.

Pour réaliser cette détection, nous utilisons encore une fois le modèle de l'objet et sa projection dans l'image. Si l'objet réel est entièrement visible la vue de synthèse de celui-ci doit correspondre à l'image réelle. Chaque pixel de l'image originale ne correspondant pas à son homologue de synthèse – via la distance entre couleurs définie à la section 3.3.2 – est donc considéré comme correspondant à une occultation.

La figure 3.4 montre un objet passant devant un livre. On dispose du modèle de ce livre (un simple rectangle texturé) et l'on utilise notre système de détection des occultations pour remplacer la texture du livre par une autre, dans cet exemple un échiquier noir et blanc.

La figure 3.5 montre la même séquence, mais la texture à été remplacée, tout en préservant les occultations. On remarque malgré tout qu'à certains endroits cette gestion n'est pas parfaite. Ceci est dû au fait qu'à ces endroits l'objet occultant est indifférenciable de l'objet d'origine, c'est-à-dire que les pixels correspondants sont identiques (ou tout au moins très proches).

FIG. 3.3 – *De gauche à droite et de haut en bas : exemple d'incrustation d'images de synthèse dans une scène réelle. On remarque dans les dernières images l'absence de gestion des problèmes d'occultation*

On peut remarquer sur cette même figure la présence d'ombres sur la nouvelle texture utilisée. Nous récupérons cette information en calculant pour chaque point considéré comme appartenant à l'objet le ratio entre la luminosité théorique – issue du modèle – et la luminosité réelle – issue des images. Ce ratio indique ainsi les ombres portées par des objets non modélisés et permet d'adapter la nouvelle texture aux conditions réelles que subit le livre d'origine.

Nous avons pour finir appliqué cette méthode à la séquence du livre présentée dans la section précédente. La figure 3.6 montre la même surimposition d'éléments de synthèse, mais en gérant cette fois les occultations. On constate bien la prise en compte des mains, sauf à la fin de la séquence ou celle-ci passe devant le chapeau. Ceci vient du fait que le chapeau ne correspond pas à une zone de l'objet suivi, et il est donc impossible à cet endroit ci de déterminer les éventuelles occultations.

## 3.3 Réalité virtuelle

Dans cet exemple notre propos est d'intégrer un objet dans une scène virtuelle. L'objet est suivi grâce à notre approche, et les paramètres définissant ses mouvements sont transposés dans un espace virtuel. Il s'agit ici d'une forme simple d'intégration d'un objet réel dans une scène virtuelle. Notre système ne visant pas le temps interactif, c'est le seul mode d'intégration que nous testons ici. Des modes plus *interactifs*, incluant par exemple un système à retour d'efforts, nécessitent forcement une intégration rapide, pour que les retours du monde virtuel aient un sens.

FIG. 3.4 – *Images d'un livre avec un objet passant devant.*

Nous reprenons l'exemple du plateau vu précédemment (section 3.1). Ce plateau va être vu comme un curseur destiné à interagir avec cet espace virtuel.

(résultats en cours de calcul)

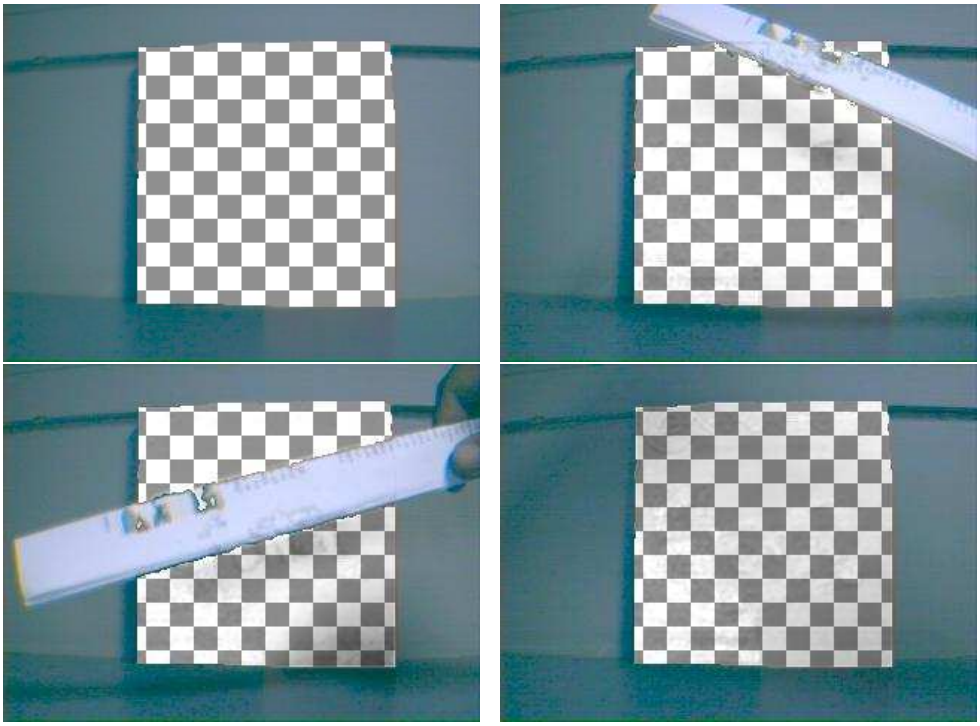


FIG. 3.5 – *Même série d'images que la précédente, avec remplacement de la texture initiale du livre, en prenant en compte les occultations ainsi que les variations d'éclairement.*

FIG. 3.6 – *De gauche à droite et de haut en bas : exemple d'incrustation d'images de synthèse dans une scène réelle, avec gestion des problèmes d'occultation. La restitution des ombres initiales (en bas du livre) est également gérée.*

# Sixième partie

# Conclusions sur nos travaux

Nous avons présenté dans cette thèse nos travaux sur le suivi de paramètres 3D, à partir d'un modèle géométrique paramétré de l'objet suivi et d'un ensemble de séquences vidéo multi-vues.

L'idée que nous avons exploité durant nos travaux est que le modèle de l'objet peut être utilisé dans ses multiples aspects :

- la structure de l'objet, ou squelette, qui va définir les paramètres des mouvements associés à l'objet. Se sont leurs valeurs que nous cherchons à suivre ;
- la géométrie de l'objet, qui spécifie comment l'objet occupe l'espace ;
- les aspects visuels qui fixent les propriétés photogramétriques des surfaces de l'objet.

Ce sont ainsi ces paramètres, dont les valeurs définissent les postures possibles de l'objet, que nous cherchons à suivre, afin que le modèle reproduise les mouvements de l'objet réel.

Dans notre approche nous utilisons les différentes informations apportées par le modèle pour exprimer un ensemble de valeurs de paramètres sous forme d'image. Il s'agit de vues de synthèse du modèle, correspondant aux différentes vues réelles disponibles.

Le problème du suivi de paramètres est donc posé en terme de recherche des paramètres générant les vues de synthèse les plus fidèles par rapport aux images réelles.

La définition d'une mesure d'écart entre images fondée sur les comparaisons denses et prenant en compte les problèmes d'erreurs de modélisation ainsi que les problèmes d'occultations permet de mesurer la pertinence d'un vecteur de paramètres.

Une méthode dérivée du simplexe et étendu au domaine continu permet, à partir de cette mesure, la recherche des paramètres correspondant le mieux aux vues de l'objet.

Ce processus est étendu pour prendre en compte l'aspect temporel et réaliser le suivi proprement dit. Des techniques tels que la prédiction des variations des valeurs des paramètres sont utilisées pour augmenter la précision du suivi.

Un ensemble de séquences vidéo de synthèse permet de montrer la validité théorique de notre approche et de tester des situations précises, telles que la présence occultations.

Un ensemble de séquences réelles permet enfin de montrer la stabilité et la fiabilité de nos travaux dans des conditions réelles.

Au delà des résultats présentés dans ce manuscrit il reste un vaste champs d'investigation ouvert, et de nombreuses applications potentielles.

Les applications cible de ce type de traitements sont multiples. En premier lieu cela s'adresse aux méthodes d'intégration d'entités réelles dans des environnements virtuels – la réalité virtuelle. Cela peut être sous la forme d'un moyen de contrôle et de dialogue avec un système virtuel, ou même pour une intégration – une immersion – d'un objet ou d'un personnage dans ces environnements, sans passer par l'utilisation de senseurs qui restent lourds à mettre en œuvre.

Cela concerne également tout ce qui a trait au contrôle de position et orientation d'objets, par exemple en milieu industriel. Dans le cas de pièces manufacturées à manipuler de façon automatique – par des pinces robotisées par exemple – le système contrôlant la

manipulation a besoin de connaître précisement "l'état" spatial de la pièce pour travailler correctement. Même en l'absence de manipulation, cela peut concerner la vérification du bon positionnement d'objets, par exemple sur une ligne de production, afin d'alerter un opérateur humain en cas de mauvaise position (ce qui peut être lourd de conséquences pour la suite des traitements).

Dans des situations telles que l'étude des *crash-tests*, l'utilisation de senseurs se heurte à des problèmes de résistance aux chocs, et les techniques de marqueurs sont délicate à mettre en œuvre à cause des nombreuses occultations dues à la structure des véhicules. Ainsi des méthodes strictement externe – sans instrumentation du mannequin de test – sont souhaitables pour ce type d'études.

Enfin dans le domaine de la réalité augmenté ce type de méthode peut apporter beaucoup. En effet pour *ajouter* de l'information à une scène réelle, issue de prises vidéo, il faut analyser cette scène afin d'en extraire des informations utiles, destinées à être ajouter aux images originales. Dans des situations telles que l'assistance à la conduite, la prise en compte des objets mobiles et le suivi de leurs trajectoires est important pour la sécurité et la prévention, et correspond au type de problème que nous nous sommes posé.

Au niveau de la réalisation concrète de notre approche, un certain nombre de possibilités existent pour accélérer les traitements, afin de d'obtenir un système en temps interactif, à défaut de temps réel.

Une voie partiellement validée est l'utilisation des accélérations matérielles pour la synthèse d'images. On trouve à des prix plus que raisonnables des cartes accélératrices pour les PC dont les performances sont impressionnantes. Comme désormais ces cartes sont directement gérées par les systèmes d'exploitation des machines, des librairies de rendu telles que OpenGL peuvent faire directement appel à ces technologies et offrir des rendus en des temps records.

Un autre aspect consommateur de temps est la comparaison dense entre images. Hormis l'utilisation, encore une fois, de composants spécialisés l'utilisation de la multi-résolution peut permettre un gain de temps intéressant. L'exploitation d'images de faible résolution en début de traitement permettant ainsi une forte accélération, les résolutions maximales n'étant utilisées qu'en fin de recherche, lorsque toute la précision disponible est nécessaire.

À plus long terme, quelques aspects restent à étudier.

En premier lieu des tests devront être réalisés afin de vérifier la validité de notre approche sur des images autre que vidéo. Nous pensons en effet que des images issues de caméras de types variés peuvent servir de base pour le suivi de paramètres. En particulier une idée a été émise en relation avec l'INRETS pour la recherche de paramètres intrinsèques de parties de corps humains, à partir d'images à rayons X.

La détection des erreurs de suivi est une étape obligatoire pour réaliser un système fiable. Les aspects liés à la détection et éventuellement la correction des décrochages durant le suivi sont donc également importants et mériterons d'être développés.

Un autre point méritant d'être développé est le traitement des occultations. La détection d'occultations durant le suivi lui-même peut permettre une prise en compte directe et une amélioration du suivi. De même, une meilleure caractérisation de ces occultations

permettrait une gestion plus efficace des post-traitements pour la réalité augmenté par exemple, en tenant compte plus précisement de l'environnement de l'objet suivi.

# Table des figures

# A

# Espaces colorimétriques

La représentation des couleurs est un vaste problème. Cette information est tridimensionnelle, mais le choix de l'espace de représentation peut varier selon les opérations que l'on souhaite mener sur les couleurs. Il existe en fait de très nombreux espaces de représentation de la couleur. Ces espaces ont été définis pour obtenir des propriétés particulières que ne possèdent pas les formats "classiques" en informatique, en particulier au niveau de la définition de distances dans ces espaces. La figure A.1 page suivante montre quelques exemples d'espaces colorimétrique. On remarque en particulier que les formes de la zone définissant les couleurs visibles sont très diverses.

L'espace RGB, très répandu car utilisé par la plupart des systèmes graphiques pour la représentation des couleurs, ne possède pas de propriétés particulières. Des notions utiles telles que la luminosité, la *teinte* ou la saturation ne sont pas directement accessibles. D'autres espaces tels que le HLS (teinte, luminosité, saturation ou en anglais *Hue, Light, Saturation*) ou le HSV (teinte, saturation, valeur, en anglais *Hue, Saturation, Value*) séparent ces caractéristiques de façon directe. Le but est d'avoir des axes qui soient indépendants les uns des autres (contrairement au RGB par exemple) par rapport aux caractéristiques de la couleur. D'autres espaces ont été mis au point afin de pouvoir définir des distances entre couleurs qui soient liées à la perception humaine[Loz98], c'est-à-dire qu'une distance donnée dans cet espace sera perçue par un humain comme identique quelque soient les couleurs auxquelles elle s'applique (perceptuellement uniforme). C'est le cas par exemple des espaces Lab et Luv.

La figure A.2 page 129 regroupe une classification des principaux espaces colorimétriques en fonction de leurs propriétés [VAN00].

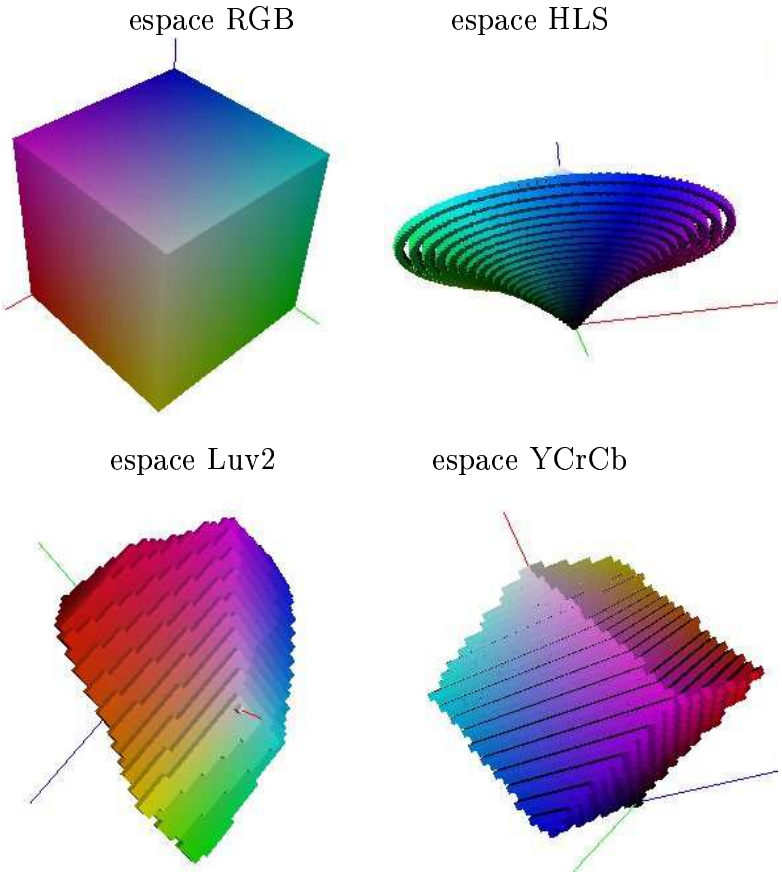



FIG. A.1 – *Exemples d'espaces colorimétriques.*

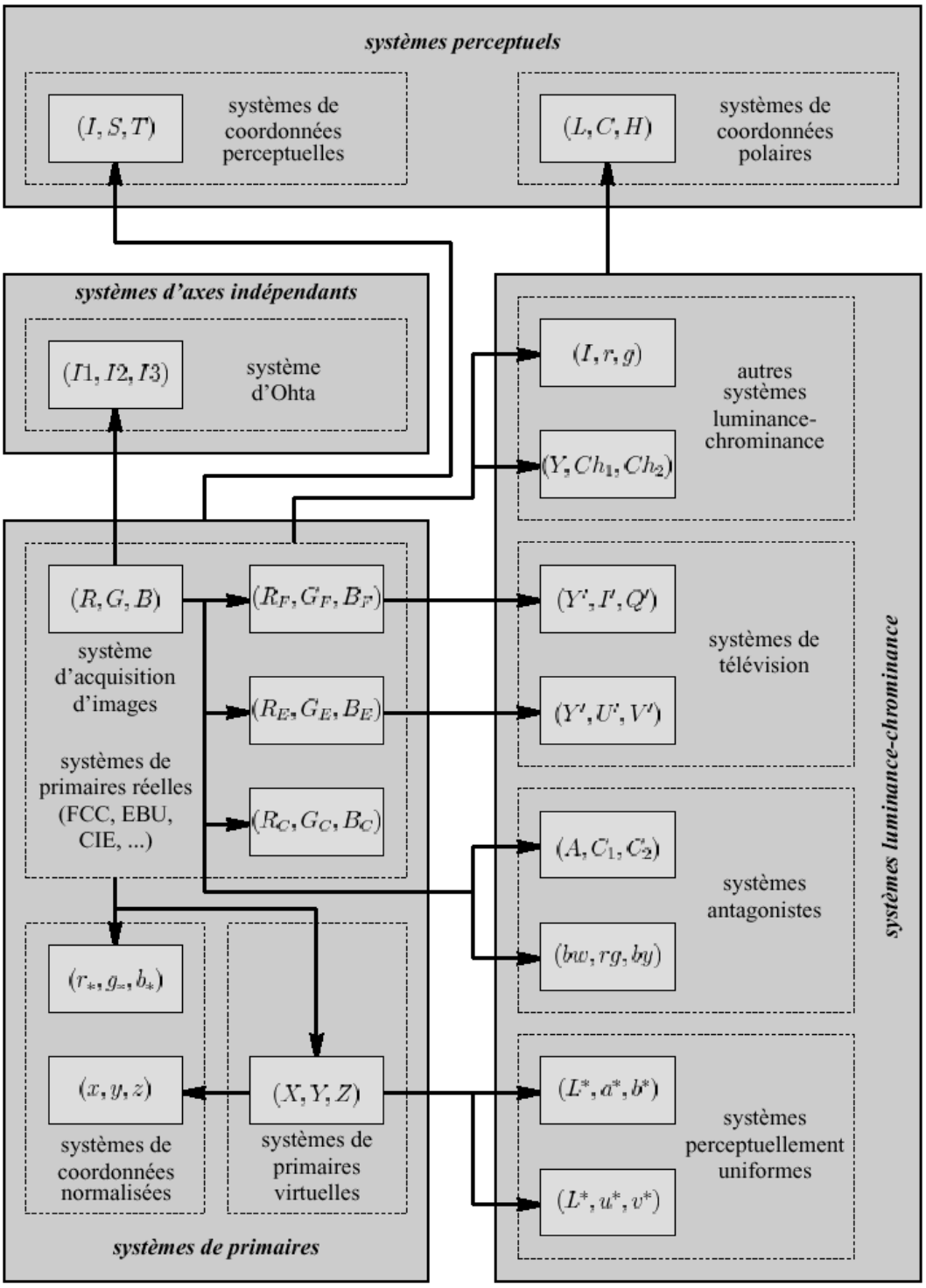



FIG. A.2 – *Classification des principaux espaces colorimétriques en fonction de leurs propriétés.*

# B

# Termes utilisés et langue française

Dans de nombreux domaines scientifique, mais tout particulièrement en informatique, les mots, verbes et noms utilisés sont en "avance" sur le dictionnaire de la langue française.

Ceci à pour conséquences l'utilisation de nombreux néologismes dans les publications en français, et souvent il existe plusieurs néologismes différents pour le même concept. À cela se rajoute l'utilisation fréquente d'anglicismes, soit que le terme original à été apporté par des travaux en anglais et que le terme correspondant en français n'a pas encore été créé, soit que le terme anglais permet d'exprimer le même concept de façon plus concise ou plus souple (transformation d'un nom en verbe ou participe...).

Pour ces différentes raison, nous avons tenu à détailler dans cette annexe la liste des termes que nous utilisons et qui sont soit des néologismes, soit des anglicismes, en expliquant pourquoi les alternatives "légales" de la langue française ne donnaient pas satisfaction.

Dans la suite de cette annexe, nous suivrons toujours la structure suivante : terme posant problème, courte définition du terme, termes reconnus par la langue française, et justification de notre choix.

**texturage** :
- acte de définir, d'appliquer une texture à un objet.
- *texture* existe, ainsi que le verbe *texturer*, mais aucune forme alternative.
- à part l'anglicisme *texturing*, les alternatives sont *texturage* et *texturation*. Le deuxième existe et a un sens différent (lié aux traitements des textiles).

**paramétré** :
- fait de contenir des paramètres, dans notre cas pour un modèle géométrique.
- le verbe *paramétrer* existe, mais est uniquement présent sous forme transitive.
- nous avons choisi d'utiliser la forme intransitive de ce verbe, même si elle n'est pas reconnue.

**texel** :
- élément de texture. Une texture est une image composée de pixels, mais chacun de ces pixels peut se projeter dans l'image de synthèse finale sous de multiples formes

(groupe de pixels, moyenne avec d'autres points...). C'est pourquoi on distingue ceux-ci des pixels en utilisant le terme de texel.
- texel n'existe pas dans les dictionnaires.
- nous utilisons tout de même ce terme car il est très fréquemment utilisé dans la littérature informatique française (dans le domaine de la synthèse d'images).

**occulteur** :
- entité occultant l'objet suivi, dans le contexte de ce manuscrit.
- *occulter* et *occultation* existent, mais pas la forme nominale *occulteur*. Il s'agit donc d'un néologisme.
- nous utilisons tout de même ce terme car l'alternative serait une périphrase. Ce terme étant souvent employé dans certaines sections, cela alourdirait rapidement le texte.

**crash-test** :
- essais d'accident sur des véhicules réels.
- il s'agit d'un anglicisme désormais très répandu.
- en l'absence d'alternative pratique en langue française, nous utilisons directement l'anglicisme.

# Bibliographie


[Ahu93] A.J. Ahumada. Computational image quality metrics : a review. 1993.

[Ama97] M.A. Amadeh. *Une approche unifiée pour la segmentation et la mise en correspondance 2D/3D d'images multi-nodales.* PhD thesis, Institut National Polytechnique de Grenoble, 1997.

[App68] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.

[ard91] *A System Based on Neural Architectures for the Reconstruction of 3-D Shapes from Images*, Berlin, 1991. Springer-Verlag.

[ASHP93] A. Azarbayejani, T. Starner, B. Horowitz, and A.P. Pentland. Visually controled graphics. *PAMI*, 15(6) :602–605, Juin 1993.

[ATS94] J. Arvo, K. Torrance, and B. Smits. A framework for the analysis of error in global illumination algorithms. pages 75–84, July 1994.

[BCC00] F. Bérard, J. Coutaz, and J.L. Crowley. Le tableau magique : un outil pour l'activité de réflexion. In *ErgoIHM'2000, Biarritz, France*, pages 33–40, 2000.

[Bis95] C.M. Bishop. *Neural Network for Pattern Recognition.* Oxford University Press, 1995.

[Bou94] B. Boufama. *Reconstruction tridimensionnelle en visoin par ordinateur : cas des caméras non étalonnées.* PhD thesis, Institut National Polytechnique de Grenoble, 1994.

[Bro92] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4) :325–376, 1992.

[BSH00] A. Bartoli, P. Sturm, and R. Horaud. Research report 4070 : A projective framework for structure and motion recovery from two views of a piecewise planar scene. Technical report, INRIA, Grenoble, 2000.

[BW96] R. Basri and D. Weinshall. Distance metric between 3d models and 2d images for recognition and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4)*, pages 465–470, 1996.

[DDJ01] F. Duculty, M. Dhome, and F. Jurie. Tracking of 3d objects from appearance. *Proceedings of the 12th Scandinavian Conference on Image Analysis, Bergen, NorvËge*, 24(4) :515–522, Juin 2001.

[DF99] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. *7th IEEE International Conference on Computer Vision, Kerkyra, Grèce*, Septembre 1999. (poster).

[DM97] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *University of Pennsylvania technical report MS-CIS-97-23 (to appear in IJCV 2000)*, 1997.

[fac] Modeling and rendering from photographs. `http://graphics3.isi.edu/~debevec/Research/` (dernière visite : 1 Octobre 2001).

[Fau93] O. D. Faugeras. *Three-Dimensional Computer Vision : A Geometric Viewpoint*. Mit press, cambridge (ma) edition, 1993.

[Fau98] O. Faugeras. De la géométrie au calcul variationnel : théorie et applications de la vision tridimensionnelle. *11ème Congrès RFIA'98*, Janvier 1998.

[FL94] P. Fua and Y.G. Leclerc. Registration without correspondences. *CVPR, Seattle, WA*, 1994.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, 1987.

[FZMC96] W. Frey, M. Zyda, R. McGhee, and W. Cockayne. Off-the-shelf, real-time, human body motion capture for synthetic environments. *Tech. Report NPSCS-96-003, Computer Science Dept., Naval Post-Graduate School, Monterey, Californie*, 1996.

[GD95] D.M. Gavrila and L.S. Davis. 3d model-based tracking of humans in action : A multiview approach. *IEEE Intl. Symp. on Computer Vision, Coral Gables, FL*, pages 253–258, 1995.

[GG00] P. Gérard and A. Gagalowicz. Three dimensional model-based tracking using texture learning and matching. *Pattern Recognition Letters*, (21) :1095–1103, 2000.

[GMDP00] V. Gouet, P. Montesinos, R. Deriche, and D. Pelé. évaluation de détecteurs de points d'intérêt pour la couleur. volume II, pages 257–266, February 2000.

[Gol94] D.E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addisson-wesley books edition, 1994.

[GTGB84] C.M. Goral, K.E. Torrance, D.P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. In *SIGGRAPH 84*, pages 213–222, 1984.

[HDLC97] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose : The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 1997.

[HDN96] M. Hirose, G. Deffaux, and Y. Nakagaki. Development of an effective motion capture system based on data fusion and minimal use of sensors. pages 117–123, 1996.

[HYAT01] R. Hoshino, S. Yonenmoto, D. Arita, and R. Taniguchi. Real-time analysis of human motion using multi-view silhouette contours. *Proceedings of the 12th Scandinavian Conference on Image Analysis, Bergen, NorvËge*, pages 537–544, Juin 2001.

[IS98] J. Isodoro and S. Sclaroff. Active voodoo dolls : A vision based input device for nonrigid control. *Proc. Computer Animation*, June 1998.

[KM00] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1453–1459, DÈcembre 2000.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1988.

[La3] Vivid - a 3d digitizer. `http://www.minolta-rio.com/vivid/index-e.html` (dernière visite : 10 Septembre 2001).

[lim99] *Efficient Content-Based ImageRetrieval based on color homogeneous objetcs segmentation and their spatial relationship characterization*, 1999.

[Loz98] V. Lozano. *Contribution de l'analyse d'image couleur au traitement des images textile.* PhD thesis, LIGIV, IUP Ingénierie de la vision, Saint-Étienne, 1998.

[LRD98] F. Lerasle, G. Rives, and M. Dhome. Suivi des membres corporels par vision multi-oculaire. *RFIA'98*, 2 :193–201, Janvier 1998.

[LRWW98] J.C. Lagarias, J.A. Reed, M.H. Wright, and P.E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM J. Optim.*, 1998.

[MATea99] T. Molet, A. Aubel, D. Thalmann, and et al. Anyone for tennis ? *Presence 8(2)*, pages 140–156, April 1999.

[may] Modeling and rendering from photographs. `http://www.aliaswavefront.com/en/WhatWeDo/maya/index.shtml` (dernière visite : 1 Octobre 2001).

[MBM98] D. Meneveaux, K. Bouatouch, and E. Maisel. Memory management schemes for radiosity computation in complex environments. In *Computer Graphics International '98, Hannover, Germany*, 1998.

[McK98] K.I.M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM J. Optim.*, 1998.

[McR97] D.P.R. McReynolds. *Rigidity Checking for Matching 3D Point correspondences under Perspective Projection.* PhD thesis, University of British Columbia, 1997.

[MNP00] M. Malciu, L.T. Nessi, and F. PrÍteux. Pose 3d du visage dans des séquences vidéos : estimation robuste par modèle d'objet. *RFIA'00*, 1 :27–36, 2000.

[MNR92] R. Mehrotra, K.R. Namuduri, and N. Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12) :1479–1494, 1992.

[Nam] Namco - tekken 4. `http://www.namcoarcade.com/tekken4/` (dernière visite : 18 Septembre 2001).

[NM65] J.A. Nedler and R. Mead. A simplex method for function minimisation. *The Computer Journal*, 7 :308–313, Jully 1965.

[NN85] T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. In *SIGGRAPH 85*, pages 23–30, 1985.

[NT97] H. Noser and D. Thalmann. Sensor based synthetic actors in a tennis game simulation. pages 189–198, June 1997.

[OFH+98] L. Oisel, F. Fleuret, P. Horain, L. Morin, J-M. Vezion, F. Prêteux, A. Gagalowicz, and C. Labit. Analyse de séquences non calibrées pour la reconstruction 3d de scène. *RFIA'98*, 1 :189–198, Janvier 1998.

[OH99] H. Ouhaddi and P. Horain. 3d hand gesture tracking by model registration. *Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging, Santorini, GrËce*, Septembre 1999.

[OpG] Opengl - high performance 2d/3d graphics. `http://www.opengl.org/` (dernière visite : 17 Septembre 2001).

[OSRW97] E. Ofek, E. Shilat, A. Rappoport, and M. Werman. Multiresolution textures from image sequences. *IEEE Computer Graphics and Applications 17(2)*, pages 18–29, March 1997.

[PBE01a] Y. Perret, S. Bouakaz, and T. Excoffier. Real Object Parameters Tracking : a Geometric Model Based Approach. In *12th Scandinavian Conference on Image Analysis, Bergen, Norway*, pages 501–507, June 2001.

[PBE01b] Y. Perret, S. Bouakaz, and T. Excoffier. Tracking of 3D Objects Using Multi-View Sequences. In *International Conference on Image and Signal Processing, Agadir, Maroc*, May 2001.

[PD99] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[PEB00] Y. Perret, T. Excoffier, and S. Bouakaz. Parameters matching of objects in video sequences. *In Proceedings of SPIE, Three-Dimensional Image Capture and Applications*, 3958, January 2000.

[PEB01] Y. Perret, T. Excoffier, and S. Bouakaz. Multi-View Parameters Tracking in Video Sequences. In *The Engineering Reality of Virtual Reality 2001, San Jose, California*, January 2001.

[PFTV92] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipies in C : The Art of Scientific Computing*. Cambridge University Press, 1992.

[Ph3] 3d1500 - digital camera for 3-dimensional photography. `http://www2.minolta.com/dp/3d1500/overview/index.html` (dernière visite : 15 Septembre 2001).

[PN92] R. Polana and R.C. Nelson. Recognition of motion from temporal texture. *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Illinois*, pages 129–134, June 1992.

[Pra91] W.K. Pratt. *Digital Image Processing*, chapter 2 and 3, page 21 to 89. Wiley Interscience, 2nd edition, 1991.

[QK97] L. Quan and T. Kanade. Affine structure from line correspondances with uncalibrated affine cameras. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997.

[Qua98] L. Quan. Algebraic relations among matching constraints of multiples images. Technical report, INRIA Rocquencourt, 1998.

[RWP+95] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust. Comparing real and synthetic images : Some ideas about metrics. In *6th Eurographics Rendering Workshop, Dublin*, pages 213–222, June 1995.

[Sen] Sensors online - your resource for sensing, communication, and control. `http://www.sensorsmag.com/` (dernière visite : 11 Septembre 2001).

[SHH62] W. Spendley, G.R. Hext, and F.R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4 :441, 1962.

[SJ96] S. Santini and R. Jain. Similarity matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

[SMB88] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. *International Conference on Computer Vision, Bombay, India*, 1988.

[ST96] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. *ECCV'96, Cambridge, UK*, 1996.

[TLG91] B. Trumbore, W. Lytle, and D.P. Greenberg. A testbed for image synthesis. pages 467–480, September 1991.

[TLV+95] J. Troccaz, N. Laieb, P. Vassal, Y. Menguy, P. Cinquin, M. Bolla, and J.Y. Giraud. Patient setup optimization for external conformal radiotherapy. *Journal of Image Guided Surgery*, 1995.

[VAN00] Nicolas VANDENBROUCKE. *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football.* PhD thesis, Laboratoire d Automatique I3D, Université des Sciences et Technologies de LILLE 1, 2000.

[Whi80] Turner Whitted. An improved illumination model for shaded display. *CACM, 1980*, 23(6) :343–349, 1980.

[Wil90] L. Williams. Performance-driven facial animation. *Proc. SIGGRAPH*, 24(4) :235–242, 1990.

[WR96] I. Weiss and A. Rosenfeld. 3d object recognition from multiple and single view. Technical report, CFAR Laboratory, University of Maryland, 1996.

[Zan98] Jacques Zaninetti. *Modélisation vectorielle de l'éclairement global en lancer de rayons.* PhD thesis, Ecole des Mines de Saint-Etienne, October 1998.

[ZP98] J. Zaninetti and B. Péroche. A vector model for global illumination in ray tracing. In Vaclav Skala, editor, *WSCG '98 (Sixth European Conference in Central Europe on Computer Graphics and Visualization)*, pages 448–455, Plzen, Czech Republic, 1998. University of West Bohemia.

# Résumé

**Mots-clés:** Suivi de paramètres, multi-vues, comparaisons denses, synthèse d'image

Nous présentons dans cette thèse nos travaux sur le suivi de paramètres d'objets réels, articulés ou non, à partir de séquences vidéo multi-vues.

Étudier le mouvement spatial de sujets en évolution à partir de sources vidéo est nécessaire pour de nombreuses applications. On retrouve ce besoin dans la vidéo-surveillance, le suivi de cibles ou encore l'analyse de mouvements en milieu industriel, par exemple. Il s'agit dans tous ces cas de mettre en correspondance les mouvements de l'objet étudié avec un modèle paramétré de celui-ci, afin que ce modèle se "comporte" identiquement au cours du temps.

De nombreuses approches existent pour parvenir à ces résultats. Après une analyse des approches existantes nous présentons la méthode que nous avons développé.

Cette approche est basée sur une collaboration entre synthèse et analyse d'images. Nous disposons en entrée de plusieurs flux vidéo, correspondant chacun à une caméra calibrée, ainsi que d'un modèle paramétré de l'objet étudié, incluant son aspect géométrique et sa structure – qui définie les paramètres.

Le suivi du sujet consiste à rechercher les paramètres générant les images de synthèse les plus fidèles par rapport aux images réelles. Ceci nécessite la définition d'une fonction d'écart entre images, le résultat désiré correspondant aux images qui optimisent cette mesure. Cette mesure est basée sur une comparaison dense entre images, prenant en compte l'information de couleur contenue dans les images. Elle tient également compte de défauts inhérents aux images réelles comme les occultations et le bruit, ainsi que les caractéristiques des images de synthèse telles que les erreurs de modélisation et les biais du rendu.

Pour rechercher l'optimum de cette mesure, nous utilisons une méthode basée sur le principe du Simplex étendue au domaine continu, appelé Nelder-Mead.

À ces traitements qui permettent d'obtenir les valeurs des paramètres à un instant donné, nous ajoutons la prise en compte de l'aspect temporel, qui va permettre le suivi proprement dit. Des techniques telles que la prédiction sur l'évolution des paramètres permettent de plus d'améliorer les performances du système.

Nous avons également développé autour de cette méthode plusieurs points importants.
Le premier permet, à partir d'un modèle dont la géométrie est imprécise, de raffiner ce modèle pour le faire correspondre aux vues réelles de l'objet correspondant.
Le deuxième point présenté utilise les informations de projection 3D $\rightarrow$ 2D pour extraire automatiquement les textures visibles de l'objet étudié dans chacune des vues, puis de les mixer afin d'obtenir une texture globale de l'objet, texture qui sera utilisée pour améliorer l'aspect visuel du modèle.
Enfin, nous introduisons en dernier lieu plusieurs techniques pour obtenir les valeurs initiales du système de suivi, qui sont les valeurs des paramètres au début des séquences vidéo.

De nombreuses expériences ont été réalisées, en se basant sur des films de synthèse ainsi que sur des films issus de scènes réelles. Nous avons également réalisé plusieurs expériences d'affinage de modèles, montrant ainsi les capacités de notre approche à utiliser un modèle d'objet approximatif comme donnée de départ pour le suivi. Les séquences de synthèse nous ont permis de montrer la validité théorique de l'approche et d'effectuer de nombreuses mesures d'erreur, les valeurs théorique des paramètres étant connues. Les séquences vidéo réelles ont montré quant à elles la fiabilité du suivi ainsi réalisé, même en utilisant des sources vidéo de faible qualité.

# Abstract

**Keywords:** parameters tracking, dense matching, image synthesis, occultation robustness

This thesis present our work on 3D parameters tracking of real objects, using multi-vues video sequences.

Studying spatial movements of objects from video streams is a important issue for several areas. This needs are presents in video-survey, target tracking or object analyse in industrial environements. In all these cases the goal is to get movements of a real entity and to make a parametered model of this object to match these movements during time.

Many approaches exist to perform 3D tracking. After a review of the existing one, we describe the method that we have developed to realize this goal. This method is based on image analysis / synthesis collaboration. We use as input data several video stream – each one corresponding to a calibrated camera – and a parametered model of the tracked object which includes its geometrical aspect and its structure – which defines the parameters.

The parameters tracking consist in finding the parameters that generate the closest synthetic images regards to the real ones. In that goal we define an adequation measure between images, and the searched results are the parameters that optimize this measure.

This measure uses dense matching, taking in count several characteristics of our problem :

- we use colored images ;
- noise and occultations can happend ;
- models are not perfect ;
- synthetic rendering is not perfect.

The optimum of the adequation measure is performed using a method called Nelder-Mead, a Simplex based approach extended in coutinuous domain.

In addition to these treatments that allow to obtain parameters values for a given instant, we have take in count the temporal aspect of the sequences in order to realize the tracking along the time, with some features such as parameters prediction that allow to obtain better performances.

We have also developped some other points around our method :

- using a unprecise model, we use our approach to refine this model using views of the real object, giving a model that match better the real images ;
- using the 3D $\rightarrow$ 2D projection we extract textures from each view, and we mix them to create a global texture which is applied to the object model, to increase realism ;
- we also present some technics to retrieve parameters values corresponding to the begin of the video sequences, in order to initialize our system.

Several experiments have been realized using our method, using both synthetic video sequences and films from real scenes.

Synthetic video sequences allow us to show the theoritical validity of our approach, and to compute parameters errors, as the values used to generate the sequences are known.

At last, real video sequences shown that our appoach is reliable, even in cases of occultation and with low quality cameras.