Vers un système de réalité augmentée autonome

Gilles Simon

Département de formation doctorale en informatique
UFR STMIA

École doctorale IAE + M

Vers un Système de Réalité Augmentée Autonome

THÈSE

présentée et soutenue publiquement le 14 dédembre 1999

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1
(spécialité informatique)

par

Gilles SIMON

Composition du jury

Président :	Dominique MERY
Rapporteurs :	Michel DHOME Andrew ZISSERMAN
Examinateur :	Luc ROBERT
Directrice :	Marie-Odile BERGER

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Remerciements

Je tiens à remercier particulièrement :

- **Marie-Odile Berger** qui m'a accueilli dans son laboratoire et qui m'a guidé pendant ces années de travail. Sans les nombreux échanges que nous avons eus, ce travail n'aurait pas vu le jour.
- **Michel Dhome et Andrew Zisserman** pour avoir eu la gentillesse d'être rapporteurs de ce mémoire. En particulier, je les remercie d'avoir rédigé les rapports dans un délais très bref.
- **Dominique Mery** qui a bien voulu être mon rapporteur interne et m'a fait l'honneur de présider le jury de soutenance.
- **Luc Robert** pour avoir accepté d'être examinateur de la thèse.
- **Vincent Lepetit**, ami et interlocuteur de tous les jours, qui m'a été d'une grande aide à plusieurs reprises. Il a entre autres étudié et comparé plusieurs algorithmes de détection et d'appariement de points d'intérêt, avant d'écrire le programme que j'utilise dans cette thèse.
- **Brigitte Wrobel** qui m'a beaucoup aidé dans la réalisation de la plateforme logicielle. Je la remercie aussi pour ses nombreux scripts qui m'ont été fort utiles dans bien des cas.
- **Sylvain Petitjean** qui a suivi mon travail de près et dont l'avis m'a toujours été très précieux.
- tous les membres du projet **Isa** pour l'amitié qu'ils m'ont témoigné et les jours que nous avons passés ensemble.
- **mes parents** et les autres membres de **ma famille**, que je ne remercierai jamais assez pour le soutien chaleureux qu'ils m'ont apporté durant toutes mes années d'études.

à Céline

Table des matières

Table des figures

Chapitre 1

La Réalité Augmentée : enjeux et problématique

La Réalité Augmentée (RA) a pour but d'améliorer notre perception du monde réel par ajout d'éléments qui ne sont pas *a priori* perceptibles par l'œil humain. Dans ce chapitre, nous présentons quelques applications de ce concept. Nous décrivons ensuite la problématique sous-jacente et définissons les objectifs de notre travail.

1.1 Définitions

Dans son état de l'art de la Réalité Augmentée, Azuma définit la RA comme un système capable de combiner des images réelles et virtuelles, en 3-D et en temps réel [Azuma97].

La composition doit être effectuée *en 3-D*, c'est-à-dire que nous devons disposer d'objets virtuels modélisés en trois dimensions, et positionnés dans un repère 3-D associé à la scène. La définition exclut donc les compositions 2-D, où des images quelconques (dessins, images de synthèse calculées selon une projection quelconque ...) sont simplement "collées" par-dessus les images réelles.

La composition doit aussi être interactive, *en temps réel* : la définition d'Azuma concerne donc principalement les applications d'immersion où l'utilisateur perçoit des objets virtuels en même temps que l'environnement réel dans lequel il évolue. Cette perception se fait généralement par l'intermédiaire d'un *casque de RA* ou *HMD* (*Head Mounted Display*, littéralement "écran monté sur la tête"). On distingue deux sortes de HMD : les HMD *optiques* et les HMD *vidéos*. Les premiers disposent d'un système optique qui est partiellement transparent, c'est-à-dire que la lumière du monde réel le traverse, et partiellement réfléchissant, ce qui permet de visualiser les images virtuelles projetées sur le système optique, en même temps que le monde réel (figure 1.1.a et b). Les HMD vidéos placent deux écrans opaques devant les yeux de l'utilisateur, qui ne perçoit donc plus *directement* le monde réel. La scène réelle est en fait filmée par deux caméras fixées sur le HMD, et le film est projeté en même temps que les images virtuelles sur les écrans du HMD (figure 1.1.c). Ce système offre donc la possibilité de traiter les images avant de les projeter, ce qui constitue une source d'information extrêmement riche pour la composition, comme nous le verrons tout au long de cette thèse.

Cependant, le terme de Réalité Augmentée est aussi couramment utilisé pour désigner la combinaison d'images réelles et virtuelles en 3-D, mais sans la contrainte temps réel [Thalmann et al.97, Faugeras98, Zisserman et al.99]. On parle aussi de *post-production*, c'est-à-dire que l'insertion des images virtuelles se fait généralement dans une étape postérieure à l'acquisition de la séquence

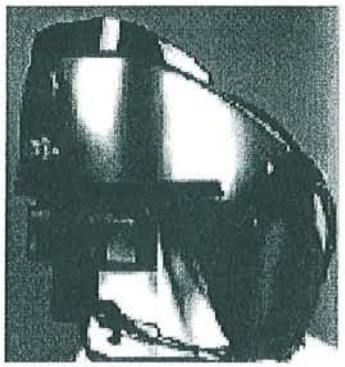
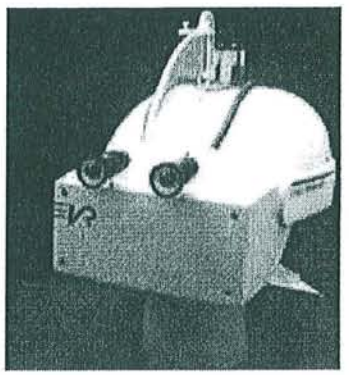
a.  b.  c. 

FIG. 1.1 – *a et b. Deux HMD optiques fabriqués par Hughes Electronics. c. Un HMD vidéo (J. Rolland et F. Biocca - UNC Chapel Hill, Dept. of Computer Science. Photo d'Alex Treml.)*

vidéo. L'opérateur peut donc passer autant de temps qu'il le souhaite pour traiter chaque image de la séquence, ce qui conduit généralement à un résultat plus précis et plus réaliste qu'avec un HMD.

1.2 Applications

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

1.2.1 Médecine

La RA peut-être utilisée par les médecins pour visualiser des données 3-D extraites chez un patient par-dessus le corps du patient (images ultrasonores, tomographie 3-D, images à résonance magnétique etc.). À l'aide d'un HMD, le médecin peut par exemple observer l'image ultrasonore d'un fœtus à l'intérieur du ventre de la mère [State et al.94] (figure 1.2). Peuchot propose un système d'assistance à la chirurgie des scolioses [Peuchot95]. L'objectif est de visualiser les déplacements de la vertèbre sous l'action des forces chirurgicales. Pour cela, une image 3-D de la vertèbre est générée et superposée sur la partie visible de la vertèbre incriminée, qui est ainsi localisée directement dans le champ de vision du chirurgien. Dans le même ordre d'idées, des pointeurs virtuels peuvent désigner certains éléments d'anatomie pour aider les étudiants en chirurgie à les visualiser [Uenohara et al.96].

Plus généralement, de nombreux travaux visent à fusionner automatiquement diverses sources d'images 3-D et 2-D : recalage d'une image tomographique ou à résonance magnétique dans une séquence d'images à rayons X [Feldmar et al.97, Roth et al.99] ou encore recalage de reconstructions 3-D de la vascularisation cérébrale dans des images d'angiographie numérique soustraite [Kerrien et al.99]. Ainsi, en neuroradiologie interventionnelle par exemple, les travaux de Kerrien ont pour but de permettre au radiologue de savoir à tout instant où se trouve son cathéter dans le corps du patient.

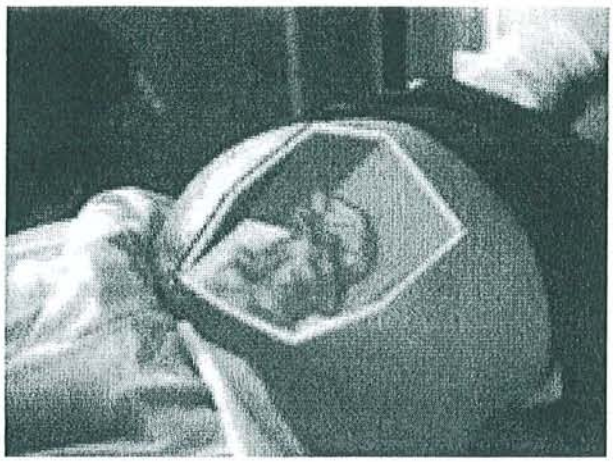


FIG. 1.2 – *Visualisation du fœtus à l'intérieur du ventre de la patiente (UNC Chapel Hill, Dept. of Computer Science).*

1.2.2 Assistance en milieu industriel

En milieu industriel, la RA permet de simplifier considérablement certaines tâches d'assemblage, de maintenance ou de réparation. Ainsi, des chercheurs de Boeing ont développé un système permettant de guider les techniciens dans l'assemblage de réseaux électriques pour les avions [Curtis et al.98]. Avant que ce système ne soit mis en place, les techniciens plaçaient les fils selon des schémas gravés sur des panneaux. Un 747 comportant plus de 1000 réseaux électriques, et les réseaux étant différents d'un avion à l'autre, cela impliquait des coûts considérables pour le stockage, le transport et la construction des panneaux. Avec le nouveau système, tous les réseaux sont stockés en mémoire. Il suffit au technicien de choisir le réseau approprié, qu'il peut alors visualiser par-dessus un panneau vierge en s'équipant d'un HMD. Le même principe est mis en œuvre dans [Reiners et al.98] pour l'assemblage d'un mécanisme de fermeture de porte pour automobile. Des indications virtuelles peuvent aussi être ajoutées pour désigner certaines pièces d'objets manufacturés, comme des photocopieuses [Feiner et al.93] ou des moteurs [Rose et al.94, Ravela et al.96], et faciliter ainsi les opérations de maintenance ou de réparation sur ces objets.

1.2.3 Design intérieur

Un autre domaine d'application de la RA est le design intérieur : le designer dispose d'une base de données de modèles de meubles ou d'éléments décoratifs, qu'il peut positionner et visualiser dans la pièce physique à meubler. Ceci peut se faire par le biais d'une interface graphique [Ahlers et al.95] (figure 1.3) ou d'un HMD [Satoh et al.98] : le designer peut alors se déplacer dans la pièce et visualiser en temps réel les différents éléments ajoutés.

1.2.4 Jeu

La Réalité Augmentée commence à se développer dans le domaine du jeu. Stricker et al. ont par exemple conçu un jeu de Tic Tac Toe interactif en 3-D [Stricker et al.98]. Le joueur, portant un HMD optique muni d'une caméra, place son pion sur la grille et appuie sur un bouton "GO" virtuel. L'ordinateur analyse alors la scène, localise le pion (grâce à sa couleur), place sa propre croix virtuelle et demande au joueur de poursuivre le jeu par l'intermédiaire d'un

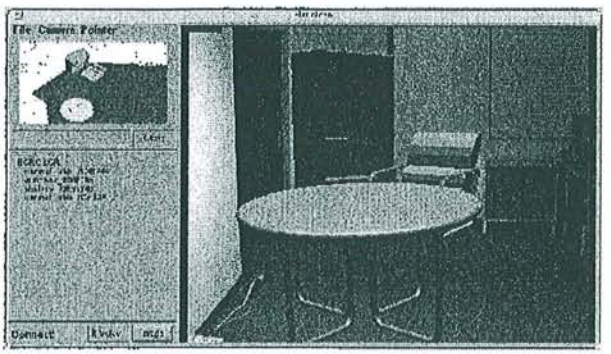


FIG. 1.3 – *Une interface permettant de positionner des meubles virtuels dans une photographie (K. H. Ahlers, A. Kramer, D. E. Breen, P.-Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. T. Whitaker et D. Greer, ECRC).*

écran virtuel (figure 1.4). Un jeu de Hockey sur table "augmenté" a aussi été proposé par Satoh et al. [Satoh et al.98] : les joueurs frappent un palet virtuel avec des maillets physiques sur un terrain de jeu physique.

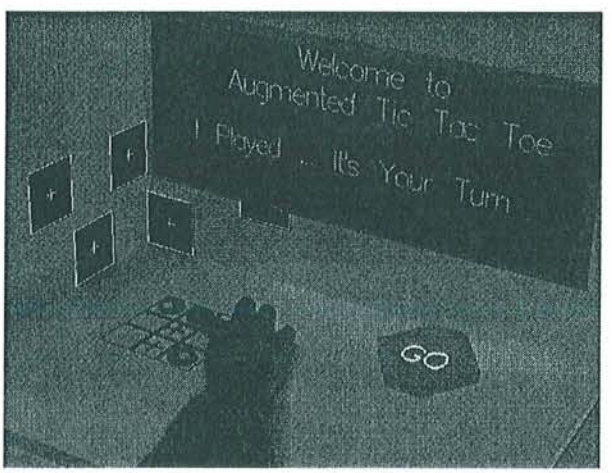


FIG. 1.4 – *La RA appliquée au jeu du Tic Tac Toe (D. Stricker, G. Klinker et D. Reiners, ZGDV, Allemagne).*

1.2.5 Effets spéciaux

Les effets spéciaux pour le cinéma, la publicité ou les clips vidéo intègrent de plus en plus souvent des images de synthèse qui viennent s'intégrer au film : on peut notamment citer quelques réalisations célèbres, comme le film *Titanic* où éléments réels et virtuels se côtoient sans même que le spectateur ne s'en rende compte, ou encore l'édition spéciale de la trilogie *Star Wars*, qui fait apparaître des créatures virtuelles dans un décor réel (figure 1.5). Pour ce film, les compositions ont été réalisées manuellement [Vaz97]. Thalmann et al. décrivent les différentes étapes à prendre en compte pour l'incrustation semi-automatique d'acteurs virtuels animés dans un environnement réel [Thalmann et al.97] : extraction des paramètres de la caméra, création et animation des acteurs et rendu des images finales. En particulier, le rendu tient compte des objets réels cachés par les acteurs virtuels et vice versa, des collisions entre les acteurs virtuels

et l'environnement réel et des ombres des acteurs virtuels sur le monde réel. Plusieurs exemples d'incrustation d'une Marilyn Monroe virtuelle évoluant dans un décor réel sont ainsi proposés (figure 1.6).

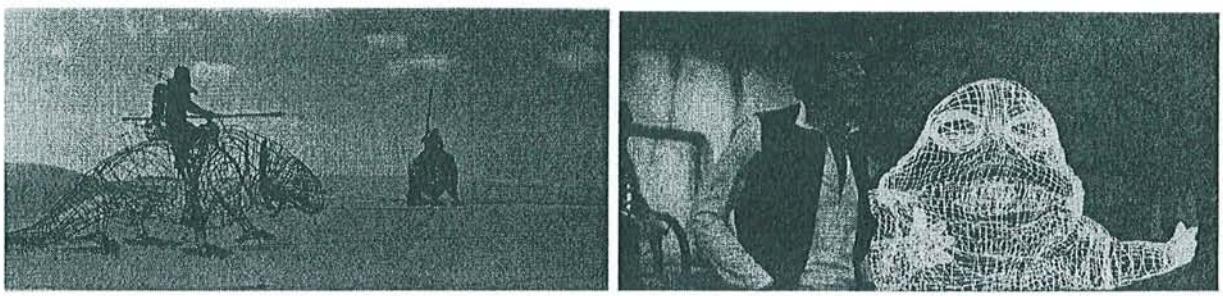


FIG. 1.5 – *Certains effets spéciaux de la trilogie Star Wars utilisent des images de synthèse superposées aux images réelles (images extraites de* Lucasfilm Magazine *n°10).*

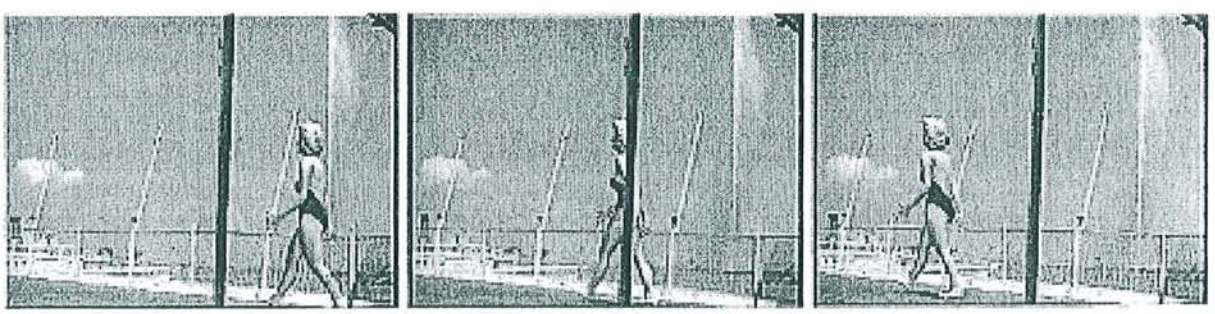


FIG. 1.6 – *Une Marilyn Monroe virtuelle évoluant dans un décor réel (N. Magnenat-Thalmann et D.Thalmann, MIRALab, Université de Genève).*

1.2.6 Étude d'impact

La RA est aussi utilisée en architecture pour les études d'impact [Maver et al.85] ou l'illumination artificielle de monuments [Chevrier et al.95]. En particulier, le projet des ponts de Paris, qui nous a été confié dans le cadre d'une recherche soutenue par le PIR-Ville du CNRS, et en collaboration avec le CRAI à Nancy, vise à illuminer virtuellement un certain nombre de ponts situés autour de l'Île de la Cité. L'objectif est d'évaluer visuellement l'impact sur l'environnement de plusieurs projets d'illumination. Plutôt que d'effectuer des tests *in situ*, il est plus pratique et moins coûteux d'utiliser des images de synthèse. Pour cela, des séquences vidéo nous ont été fournies (comme la séquence du Pont Neuf qui comporte plus de 300 images), dans lesquelles nous devons remplacer le pont réel par un pont virtuel, illuminé selon le système d'éclairage que nous souhaitons évaluer (figure 1.7).

1.3 Problématique

La RA est une discipline relativement récente, dont la majorité des avancées a eu lieu ces cinq dernière années. Les systèmes de composition d'images en phase de post-production sont aujourd'hui utilisés aussi bien par les producteurs d'effets spéciaux que par le grand public, qui

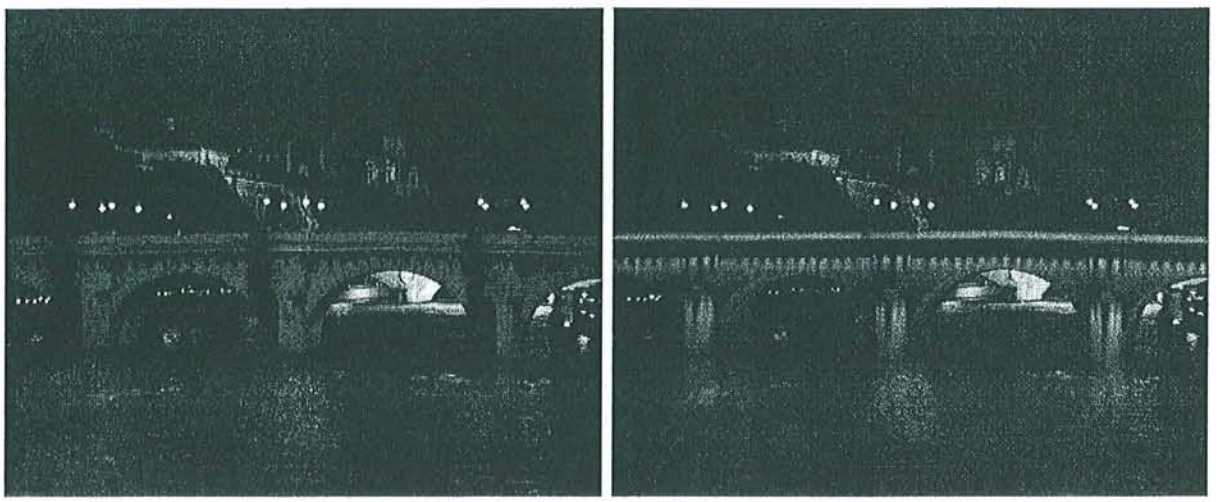


FIG. 1.7 – *Exemple d'illumination artificielle du Pont Neuf. A gauche l'image réelle, à droite l'image augmentée.*

peut se procurer plusieurs logiciels dans le commerce (le degré d'interactivité et la complexité des applications prises en compte étant variables d'un produit à l'autre).

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

En particulier, la visualisation simultanée de données réelles et virtuelles au travers d'un HMD génère des perturbations physiologiques qui doivent être prises en compte pour une utilisation prolongée de cet outil. D'autre part, un délais de l'ordre de 100 ms est inévitable entre l'instant où l'utilisateur perçoit la scène et l'instant où l'image virtuelle est projetée [Azuma97]. Si l'utilisateur bouge la tête entre ces deux instants, l'objet virtuel est décalé par rapport aux objets réels.

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

1.3.1 Cohérence géométrique

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

Le point de vue de la caméra peut être retrouvé grâce à des émetteurs magnétiques (par exemple) placés au niveau de la caméra ou du HMD. Une autre méthode fréquemment employée consiste à localiser dans l'image un certain nombre d'indices de la scène (points, droites ou autre). On recherche alors le point de vue qui minimise la distance entre les indices de l'image et la projection des indices 3-D dans le plan image de la caméra (sur une séquence d'images,

on parle de *recalage temporel*). Les différentes techniques d'estimation du point de vue seront décrites plus précisément au chapitre 2.

En plus de positionner l'objet virtuel au bon endroit de l'image et selon la bonne orientation, il faut aussi prendre en compte les occultations entre objets réels et virtuels. En figure 1.8, nous pouvons voir un exemple d'occultation de l'objet virtuel par un objet réel de la scène (le bateau). Si nous nous contentons de projeter l'objet virtuel par-dessus l'image réelle, cette occultation n'est pas prise en compte et il s'ensuit une incohérence géométrique.

1.3.2 Cohérence photométrique

Afin d'assurer la cohérence photométrique de la composition, il faut par ailleurs synthétiser les inter-réflexions de la lumière entre objets réels et virtuels, ainsi que les ombres portées. Par exemple, pour le projet des ponts de Paris, nous avons dû générer le reflet du pont illuminé dans l'eau de la Seine (figure 1.8).

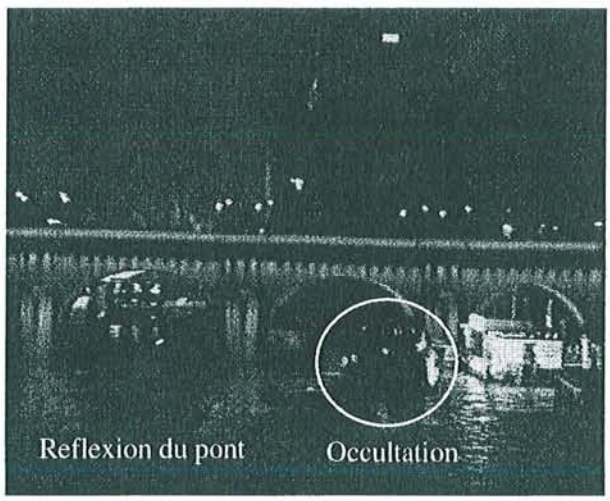


FIG. 1.8 – *Cohérence géométrique et photométrique pour la séquence du Pont Neuf.*

1.4 Objectifs

Le travail présenté dans ce mémoire concerne principalement le calcul des paramètres de la caméra sur une séquence d'images. Nous n'en donnons ici que les orientations générales, que nous affinerons au chapitre 3.

1.4.1 Le projet des ponts de Paris

Notre travail a pour origine le projet des ponts de Paris. En 1995, Christine Chevrier, qui effectuait sa thèse sur la simulation de la propagation de la lumière dans un contexte de Réalité Augmentée [Chevrier96], s'est vue confiée la réalisation des images de synthèse pour le projet des ponts de Paris. Très vite, la détermination du point de vue de la caméra s'est avérée problématique. Les tentatives d'approximation par "tâtonnement" se sont révélées trop fastidieuses. C'est dans ce contexte qu'a démarré mon DEA [Simon95]. Plusieurs méthodes utilisant des appariements 3-D/2-D de points [Dementhon et al.95, Ferri et al.93] ou de droites [Dhome et al.89, Shakunaga93, Kumar et al.94] étaient alors disponibles, mais concernaient des

images isolées, et non des séquences d'images. Certains systèmes prenaient en compte des séquences complètes [Gennery92, Koller et al.92], mais ceux-ci étaient basé sur le recalage d'objets polyédriques, alors que les seuls indices facilement identifiables dans la séquence étaient les taches blanches formées par les lampadaires et les arches du pont. Nous avons donc commencé par suivre des points par corrélation, ainsi que des courbes dont nous extrayions les points anguleux [Berger et al.96a, Berger et al.99]. Cette solution nous a permis de mener à bien le projet des ponts de Paris, mais n'était pas suffisamment autonome et s'est avérée difficilement généralisable à d'autres applications de RA (voir le chapitre 4). Nous avons donc décidé de prendre en compte des courbes quelconques du modèle au lieu de points particuliers de ces courbes, afin d'obtenir un système général et autonome.

Plus généralement, l'objectif de cette thèse est d'obtenir un système de recalage temporel qui soit peu contraignant, autonome et séquentiel.

1.4.2 Un système peu contraignant

Les différents systèmes de RA existants requièrent un degré de connaissance plus ou moins important sur la structure de la scène filmée et les paramètres de la caméra. Nous souhaitons que notre système soit peu contraignant quant à la connaissance requise sur la scène, et qu'il ne présuppose aucune connaissance *a priori* sur le mouvement de la caméra.

Structure de la scène

Plusieurs systèmes de RA (prototypes ou produits commerciaux) sont aujourd'hui disponibles : ceux-ci seront présentés au chapitre 3. Cependant, ces systèmes sont généralement assez contraignants dans la mesure où ils nécessitent soit de positionner des marqueurs facilement identifiables dans la scène [Stricker et al.98], soit de disposer d'une base de données d'images de l'objet recalé, photographié sous plusieurs angles et plusieurs illuminations [Ravela et al.96], soit encore de disposer d'un modèle complet de cet objet sous forme de facettes [Gagalowicz99]. Certains systèmes ne nécessitent aucune connaissance de la scène filmée [Faugeras98, Fitzgibbon et al.98], mais requièrent une forte interaction avec l'utilisateur.

Nous souhaitons que notre système puisse s'appuyer sur la structure *naturelle* de la scène : le positionnement de marqueurs n'est bien souvent pas envisageable pour les scènes d'extérieur. Nous voulons d'autre part pouvoir prendre en compte non seulement des segments de droite, mais aussi des courbes quelconques, ou *courbes de forme libre*, dans le but évident d'obtenir un système général, non restreint au recalage d'objets polyédriques. Enfin, nous voulons nous baser sur une connaissance *minimale* de la scène, c'est-à-dire un petit nombre de primitives visibles dans la séquence, au lieu du modèle complet d'un objet de la scène, qui n'est pas toujours disponible (alors que quelques mesures *in situ* de la scène sont souvent envisageables).

Le mouvement de la caméra

Nous supposons que le mouvement de la caméra est *quelconque*, c'est-à-dire que nous devons déterminer six paramètres extrinsèques (trois pour la translation et trois pour la rotation), et ne voulons imposer aucune restriction sur les mouvements autorisés. De plus, nous partons de l'hypothèse que le mouvement de la caméra est *imprévisible*, c'est-à-dire que nous ne pouvons pas l'approximer par un modèle physique. En effet, certains systèmes utilisent un modèle de mouvement de type *filtre de Kalman* pour prédire la position des primitives à détecter dans les images de la séquence [Gennery92, Gagalowicz99]. Cependant, plusieurs auteurs ont souligné que

l'utilisation d'un modèle de mouvement n'était pas adaptée aux applications de RA, car l'utilisateur déplace généralement la caméra comme il l'entend, et de façon totalement imprévisible [Lowe92, Ravela et al.96, Stricker et al.98].

1.4.3 Un système autonome et séquentiel

Comme l'indique le titre de cette thèse, notre objectif premier est d'obtenir un système qui soit parfaitement *autonome*, c'est-à-dire qui ne requiert aucune interaction avec l'utilisateur en dehors de la phase d'initialisation. En effet, la plupart des produits commerciaux tels que Maya Live® nécessitent que l'utilisateur intervienne à différents niveaux du processus pour corriger les éventuelles erreurs de suivi, prendre en compte de nouvelles primitives etc. (cela est principalement dû à la philosophie de ces logiciels qui est de se passer du modèle de la scène). Nous souhaitons à l'inverse obtenir un système qui soit capable de détecter et corriger lui-même les erreurs de suivi.

Mais l'enjeu n'est pas uniquement de simplifier l'étape de calcul du point de vue en phase de post-production. Nous voulons aussi pouvoir envisager l'utilisation de nos algorithmes dans un contexte temps réel. En effet, une des conclusions de la discussion clôturant IWAR'98, le premier *workshop* international de Réalité Augmentée, était la suivante : la plupart des systèmes de recalage en temps réel utilisent des marqueurs artificiels placés dans la scène. Cependant, si ces approches sont acceptables dans un environnement industriel, elles ne sont pas adaptées aux environnements extérieurs pour lesquels positionner des marqueurs n'est la plupart du temps pas envisageable. Le nouveau défit est donc d'être capable de prendre en compte la structure naturelle de la scène (on pourra se référer à [Behringer et al.99] pour plus de détails sur la discussion).

A notre connaissance, il n'existe en effet aucun système fonctionnant en temps réel adapté aux scènes extérieures. Les algorithmes que nous proposons ne s'exécutent pas en temps réel sur les machines usuelles. Cependant, nous nous sommes appliqués à ce qu'ils soient *théoriquement* conductibles aux applications temps réel, c'est-à-dire à ce qu'ils respectent les deux conditions suivantes :

- le système doit être *autonome* : dans un contexte temps réel, il n'est bien sûr pas possible de faire intervenir l'utilisateur à un niveau quelconque du processus ;
- le système doit être *séquentiel* : certains systèmes opèrent des retours-arrière pour corriger le point de vue *a posteriori*, ou encore utilisent plusieurs images simultanément pour l'affiner. Un système en temps réel doit traiter les images les unes après les autres, au fur et à mesure de leur acquisition.

1.4.4 Qualité du système

En plus de ces deux aspects fondamentaux, nous devons prendre en compte les facteurs qualité suivants, définis ici de manière informelle :

- *stabilité* : une petite variation des données utilisées pour le recalage ne doit pas entraîner de variation arbitrairement grande sur les paramètres calculés ;
- *fiabilité* : la méthode doit fonctionner sans possibilité de défaillance dans des conditions normales d'utilisation ;
- *robustesse* : le système doit retrouver la continuité des opérations après des conditions anormales et être en mesure de supprimer les effets d'une perturbation (occultation, indice mal localisés ...). Nous verrons au chapitre 4 que la robustesse d'un système peut être quantifiée comme la plus petite fraction d'erreurs aberrantes présentes dans les données pouvant provoquer une estimation du point de vue arbitrairement mauvaise ;

- *précision* : les paramètres calculés doivent être aussi précis que possible, compte tenu des erreurs d'imprécision obtenues sur les données de référence. Nous évoquerons au chapitre 2 le degré de précision que nous pouvons espérer obtenir en fonction du contexte dans lequel sont effectuées les mesures.
- *généralité* : le système doit pouvoir être utilisé pour tout type de scène et tout type de mouvement de caméra.

1.5 Plan de la thèse

Le chapitre 2 décrit le modèle de caméra que nous utilisons et différentes techniques d'estimation de ses paramètres (calibration). Nous évoquons aussi les sources d'imprécisions auxquelles la calibration est confrontée, puis nous nous penchons sur le problème de la validation des algorithmes. Le chapitre 3 présente plusieurs systèmes de RA rencontrés dans la littérature, et positionne notre approche par rapport à ces travaux. Les chapitres suivants détaillent les différentes versions de notre système : approche par suivi de points au chapitre 4, par suivi de courbes au chapitre 5, puis utilisation conjointe de courbes 3-D et d'appariements de points 2-D/2-D au chapitre 6. Enfin, la prise en compte de changements de focale est étudiée au chapitre 7, puis nous concluons au chapitre 8.

Chapitre 2

Calibration de la caméra

La cohérence géométrique d'une composition implique le calcul d'une *matrice de projection*, qui transforme les coordonnées 3-D d'un point exprimé dans le repère de la scène en ses coordonnées pixels de l'image. Le modèle de projection le plus couramment utilisé par la communauté vision est le modèle de projection perspective, dit modèle *sténopé*. Nous commençons par décrire ce modèle en montrant que la matrice de projection perspective peut se décomposer en deux matrices, l'une représentant les paramètres intrinsèques de la caméra et l'autre ses paramètres extrinsèques. Ces paramètres peuvent être obtenus de différentes manières suivant le niveau de connaissance disponible. La *calibration forte* utilise la connaissance de mesures tridimensionnelles de la scène, alors que la technique d'*autocalibration* repose uniquement sur des données extraites d'au moins deux images de cette scène. Nous décrivons ces deux techniques en identifiant les sources d'imprécisions auxquelles elles sont confrontées. Nous nous penchons ensuite sur le problème de l'évaluation de la précision des résultats, et définissons un objectif de précision "raisonnable" pour les applications de RA.

2.1 Géométrie d'une caméra : le modèle sténopé

Pour incruster convenablement un objet virtuel dans une image réelle, celui-ci doit être filmé par une caméra virtuelle possédant les mêmes caractéristiques que la caméra réelle. Il faut pour cela adopter un modèle de caméra qui soit le plus réaliste possible : le modèle sténopé a l'avantage d'être simple tout en restant relativement fidèle à la réalité. Il conduit de plus à un modèle géométrique qui se révèle mathématiquement riche et facilement exploitable.

Le modèle sténopé est représenté en figure 2.1 : un point M se projette en un point m sur le plan $\mathcal{P}$, selon une projection perspective de centre C, situé à une distance f non nulle du plan $\mathcal{P}$. Le plan $\mathcal{P}$ est appelé *rétine* ou *plan image*, le point C *centre optique* et la distance f *distance focale*. Cette projection se décompose en deux étapes : les coordonnées du point M sont exprimées dans le repère de la caméra en fonction des paramètres extrinsèques de la caméra, puis ce point est projeté dans le plan image en fonction des paramètres intrinsèques de la caméra.

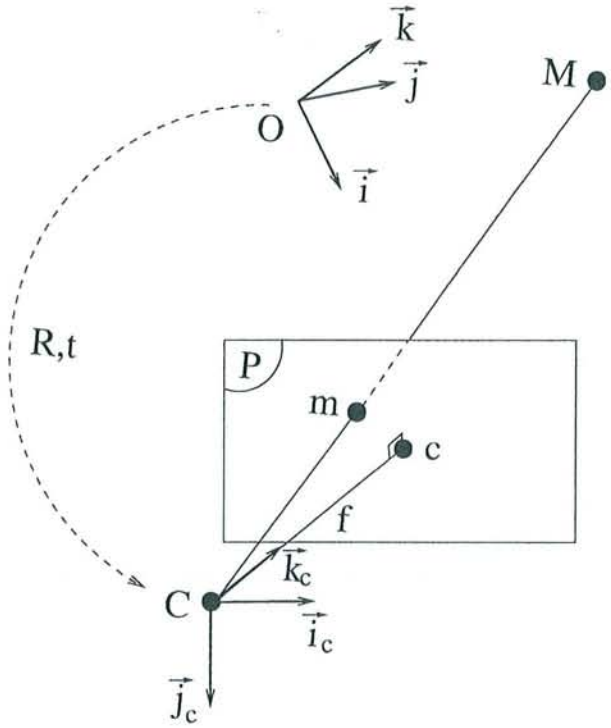


FIG. 2.1 – *Le modèle sténopé de caméra.*

2.1.1 Les paramètres extrinsèques

Si le point M a pour coordonnées (X,Y,Z) dans le repère $(O, \vec{i}, \vec{j}, \vec{k})$ de la scène, ses coordonnées (X_c,Y_c,Z_c) dans le repère $(C, \vec{i_c}, \vec{j_c}, \vec{k_c})$ de la caméra sont données par la relation :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t} = (\mathbf{R} \quad \mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.1}$$

où $(\mathbf{R} \quad \mathbf{t})$ exprime le déplacement rigide entre les deux repère (rotation et translation). La rotation $\mathbf{R}$ est souvent exprimée en fonction des angles de rotation γ, β, α autour respectivement des trois vecteurs de base $\vec{i}, \vec{j}, \vec{k}$:

$$\mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}.$$

Ces angles sont appelés *angles d'Euler*. Les paramètres du changement de repère sont donc au nombre de six : les trois angles d'Euler de $\mathbf{R}$ et les trois composantes du vecteur de translation $\mathbf{t}$. Ces paramètres, définissant l'orientation et la position de la caméra dans le repère de la scène, sont les paramètres *extrinsèques* de la caméra. Nous emploierons aussi le terme de *point de vue* de la caméra pour désigner à la fois sa position et son orientation.

2.1.2 Les paramètres intrinsèques

La projection m du point M dans le plan image a pour coordonnées dans le repère de la caméra :

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = f \begin{pmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{pmatrix}. \tag{2.2}$$

Il faut à présent exprimer le point m dans le repère 2-D dans lequel on mesure effectivement les points images (en coordonnées pixel), que nous appelons *repère de la rétine*. Dans ce repère, les coordonnées pixel (u,v) du point m sont données par l'équation (figure 2.2) :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} k_u & -k_u \cos\theta & u_0 \\ 0 & k_v \sin\theta & v_0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix}, \tag{2.3}$$

où k_u et k_v sont le nombre de pixels par unité de longueur suivant chacun des axes, u_0 et v_0 les coordonnées pixel du *point principal* c, intersection de *l'axe optique* $(C, \overrightarrow{k_c})$ avec le plan image, et θ l'angle entre les deux axes du repère image. Les paramètres k_u,k_v,f,u_0,v_0,θ sont les paramètres *intrinsèques* de la caméra. En pratique, l'angle θ est très bien contrôlé et peut être considéré égal à $\frac{\pi}{2}$. D'autre part, il n'est pas possible de séparer les paramètres k_u et k_v de la distance focale f : seules les valeurs $\alpha_u = k_u f$ et $\alpha_v = k_v f$ peuvent être calculées. Nous considérons donc le modèle simplifié à quatre paramètres α_u, α_v, u_0 et v_0. D'après les équations (2.1), (2.2) et (2.3), nous avons finalement :

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.4}$$

On note $\mathbf{A}$ la matrice des paramètres intrinsèques. La matrice $\mathbf{P} = \mathbf{A}(\mathbf{R} \quad \mathbf{t})$ est appelée *matrice de projection perspective* : elle permet d'exprimer directement la projection d'un point 3-D de la scène en coordonnées pixel de l'image, dites aussi coordonnées *rétiniennes*. Il s'agit d'une matrice 3×4 définie à un facteur d'échelle près, et possédant 11 paramètres indépendants. À partir de la connaissance de cette matrice, nous pouvons remonter aux six paramètres du point de vue et aux cinq paramètres intrinsèques [Faugeras et al.86].

2.1.3 Calibration de la caméra

On dit qu'une caméra est *calibrée* lorsque ses paramètres intrinsèques et extrinsèques sont connus. Suivant le contexte, ces paramètres peuvent être obtenus selon deux techniques différentes : techniques *basées capteurs* ou techniques *basées vision*.

Techniques basées capteurs

L'avantage d'utiliser des capteurs est que le point de vue peut être obtenu très rapidement : ceux-ci sont donc principalement utilisés pour les applications d'immersion en temps

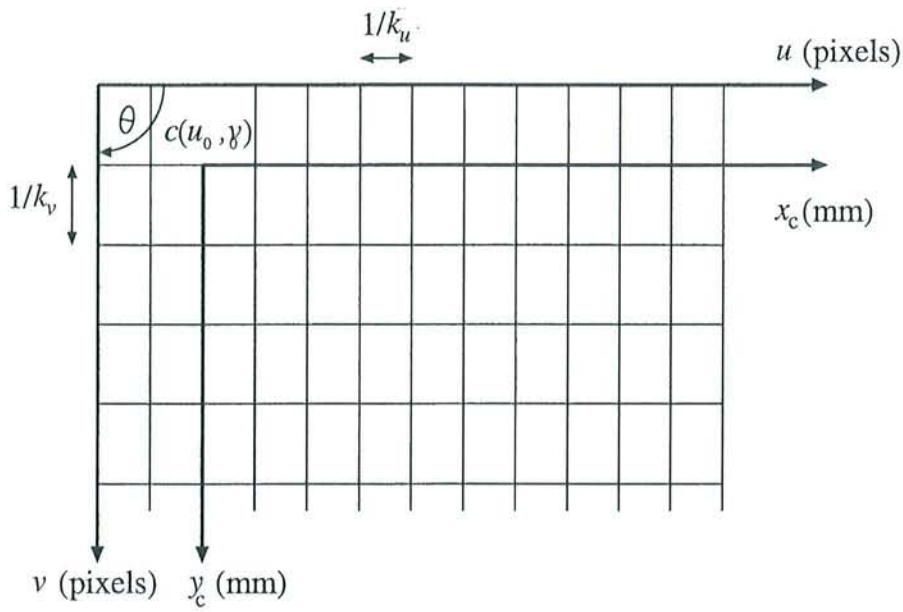


FIG. 2.2 – *Les paramètres intrinsèques*

réel [Azuma et al.94, Rose et al.94, Azuma97]. On trouve différents types de capteurs, les plus couramment employés étant les capteurs mécaniques, magnétiques et optiques.

- **capteurs mécaniques** : un système mécanique permet de détecter les mouvements de l'utilisateur. Cela permet d'obtenir une très bonne précision, mais cette solution est onéreuse, et surtout elle restreint considérablement les déplacements de l'utilisateur qui est physiquement lié au système ;
- **capteurs magnétiques** : des recepteurs miniatures placés au niveau du HMD détectent les champs magnétiques émis par un transmetteur. Cette solution est robuste et impose moins de contraintes sur les déplacements que les capteurs mécaniques (l'utilisateur doit toutefois rester à l'intérieur du champ magnétique). Malheureusement, la précision est relativement faible et le champ magnétique peut être perturbé par des objets métalliques.
- **capteurs optiques** : des marqueurs réfléchissants ou des diodes luminescentes sont filmés par une caméra. Un algorithme reconnaît les marqueurs et calcule le point de vue de la caméra à partir de leur position dans l'image. Cette solution utilise une technique basée vision, mais elle est très rapide puisque les marqueurs sont facilement identifiables. Elle offre une bonne précision et impose peu de contrainte sur les déplacements de l'utilisateur (la seule contrainte est que les marqueurs ne sortent pas du champ de vision de la caméra).

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

Techniques basées vision

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

images) n'utilise par contre que des indices extraits de deux ou plusieurs images.

Certains systèmes peuvent utiliser une approche hybride : par exemple, [State et al.96] intègrent à la fois des capteurs magnétiques et des marqueurs positionnés dans la scène. Ceci leur permet d'obtenir la précision liée à l'utilisation de marqueurs en même temps que la robustesse offerte par les capteurs.

Nous détaillons à présent les techniques de calibration forte et d'autocalibration, avant d'étudier leur précision.

2.2 Calibration forte

La calibration forte repose sur la connaissance des coordonnées 3-D de n points de référence M_i, et de leurs projections m_i dans le plan rétinien, mesurées sous forme de coordonnées pixel $\mathbf{q_i}$ (nous utilisons le style gras pour représenter les vecteurs associés aux points et les matrices associées aux tranformations). À partir de ces correspondances 3-D/2-D, nous voyons qu'il est possible de calculer les N paramètres λ_i de la matrice de projection perspective ($N = 10$ pour le modèle simplifié de caméra) à partir des n équations $\mathbf{q}_i = \mathbf{P}(\lambda_1,...,\lambda_N)\mathbf{M}_i$, pourvu que n soit suffisamment grand.

Nous verrons au chapitre 7 que la détermination simultanée des paramètres intrinsèques et extrinsèques de la caméra est généralement peu précise. Le calcul de ces paramètres est donc le plus souvent découplé : les paramètres intrinsèques peuvent être obtenus en filmant une *mire de calibration* en début de session, formée de motifs répétitifs (cercles, ellipses ou rectangles), choisis pour définir des points d'intérêt qui peuvent être mesurés avec une très grande précision (voir par exemple la mire utilisée au LORIA en figure 2.3). Par la suite, seul le point de vue est recalculé, ce qui suppose que les paramètres intrinsèques ne varient pas au cours de la prise de vue.

Nous décrivons deux méthodes permettant de calibrer les paramètres intrinsèques de la caméra, l'une étant basée sur la résolution d'un système d'équations linéaires, l'autre opérant une minimisation itérative. Nous énumérerons ensuite différentes méthodes de calcul du point de vue à partir d'un certain nombre de points de référence.

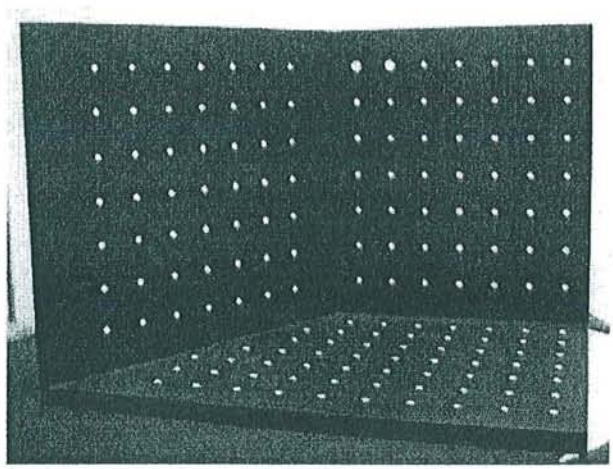


FIG. 2.3 – *La mire de calibration utilisée au LORIA.*

2.2.1 Paramètres intrinsèques

Résolution d'un système d'équations linéaires

Le système d'équations (2.4) peut s'écrire :

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{l}_1^T & l_{14} \\ \mathbf{l}_2^T & l_{24} \\ \mathbf{l}_3^T & l_{34} \end{pmatrix}}_{\mathbf{P}} \begin{pmatrix} \mathbf{M}_i \\ 1 \end{pmatrix} \Leftrightarrow \begin{cases} u_i(\mathbf{l}_3\mathbf{M}_i + l_{34}) - (\mathbf{l}_1\mathbf{M}_i + l_{14}) &= 0, \\ v_i(\mathbf{l}_3\mathbf{M}_i + l_{34}) - (\mathbf{l}_2\mathbf{M}_i + l_{24}) &= 0. \end{cases},$$

où $\mathbf{l}_i = [l_{i1}, l_{i2}, l_{i3}]$. Pour n points, on obtient donc un système linéaire homogène de $2 \times n$ équations :

$$\mathbf{L}\mathbf{l} = \mathbf{0}_{2n},$$

où $\mathbf{L}$ est une matrice $2n \times 12$ dépendant des points de référence 3-D et 2-D, et $\mathbf{l}$ est le vecteur $(\mathbf{l}_1, l_{14}, \mathbf{l}_2, l_{24}, \mathbf{l}_3, l_{34})^T$. En utilisant des contraintes d'invariance de la matrice de projection perspective, Faugeras et Toscani ont mis en évidence l'existence de la contrainte $||\mathbf{l}_3||^2 = 1$, qui permet de résoudre ce système à partir de $n \geq 6$ points 3-D/2-D [Faugeras et al.86].

Méthode itérative

Au lieu de minimiser un critère linéaire, nous pouvons aussi minimiser de façon itérative un critère c non linéaire, qui est la distance entre les coordonnées pixels (u'_i, v'_i) des points 2-D mesurés dans l'image et les coordonnées pixels des projections (u_i, v_i) des points 3-D dans le plan image, données par les équations (2.4) :

$$c = \sum_{i=1}^{n} (u'_i - u_i)^2 + (v'_i - v_i)^2.$$

Chaumette et Rives ont montré qu'en présence de bruit blanc uniforme sur les points image, la minimisation de ce critère conduisait à des résultats moins biaisés qu'avec la méthode linéaire [Chaumette et al.89]. Cette méthode permet en outre de prendre en compte les distorsions radiales qui peuvent apparaître pour certaines caméras (2.4.1).

2.2.2 Calcul du point de vue

La détermination de la position et l'orientation d'une caméra dont les paramètres intrinsèques sont connus, à partir de correspondances entre points de l'espace 3-D et points de l'image est un problème très ancien, bien connu des photogrammètres. De nombreuses méthodes ont donc été proposées à cette fin, qu'il serait difficile d'énumérer de façon exhaustive.

Pour trois points, on est amené à résoudre un système de trois équations quadratiques à trois inconnues. Les inconnues sont les distances des trois points au centre optique de la caméra. En théorie, ce système admet huit solutions, mais il a été établi que pour chaque solution réelle positive, il existe une solution négative. On se ramène donc à quatre solutions plausibles, que l'on obtient directement en résolvant une équation polynomiale de degré 4 (un résumé de cette méthode et de ses variantes peut être trouvé dans [Haralick et al.91]).

A partir de quatre points, nous pouvons obtenir une solution unique si ces points ne se trouvent pas dans une configuration particulière. Toutes les configurations pour lesquelles des solutions multiples ou instables sont inévitables sont connues [Wrobel92] (par exemple lorsque le centre optique est coplanaire avec trois des quatre points de référence). Pour le cas général,

il existe de nombreuses méthodes, itératives ou algébriques, qui permettent d'obtenir la solution unique [Lowe85, Haralick et al.89, Dementhon et al.95, Quan et al.98].

Nous décrirons au chapitre 4 la méthode de Dementhon et Davis [Dementhon et al.95], qui proposent un algorithme itératif en 25 lignes de code rapide, utilisant la projection orthographique à l'échelle comme point de départ.

2.3 Calibration faible et autocalibration

De nombreux travaux ont porté ces dix dernières années sur la calibration d'une caméra à partir de vues quelconques de l'environnement, et donc n'utilisant aucune connaissance *a priori*, ni sur la scène, ni sur les paramètres de la caméra. On peut citer notamment l'ouvrage de Faugeras [Faugeras93], qui fait référence en la matière, et la thèse de Luong [Luong92] qui explore notamment l'applicabilité des théorèmes.

La technique, appelée *autocalibration*, est fondée sur des propriétés algébriques de géométrie projective. Elle repose sur le calcul de la *matrice fondamentale* qui impose des contraintes géométriques entre deux vues, dites *contraintes épipolaires*. La détermination de cette matrice nécessite un minimum de huit correspondances de points. Lorsqu'un minimum de trois vues est disponible, nous pouvons obtenir les paramètres intrinsèques de la caméra au moyen d'un système d'équations polynomiales dites de Kruppa, et en déduire les paramètres extrinsèques.

2.3.1 Géométrie épipolaire

La contrainte épipolaire

Considérons le cas de deux caméras (ou deux vues d'une même caméra), de centres optiques respectifs C et C'. Nous pouvons voir sur la figure 2.4, qu'étant donné un point m dans le plan image $\mathcal{P}$, l'ensemble des points physiques M qui ont pu produire m se trouvent sur la demi-droite Cm. Ainsi, tous les correspondants possibles m' de m dans le plan image $\mathcal{P}'$ sont situés sur l'image donnée par la seconde caméra de cette demi-droite, qui est une demi-droite l'_m passant par le point e', intersection de la droite CC' et du plan $\mathcal{P}'$. Le point e' est appelé *épipole* de la seconde caméra par rapport à la première caméra. La droite l'_m est appelée *droite épipolaire* du point m dans le plan image $\mathcal{P}'$ de la seconde caméra. De manière symétrique, on définit l'épipole e de la première caméra par rapport à la seconde caméra comme intersection de la droite (CC') et du plan $\mathcal{P}$.

La contrainte épipolaire est qu'étant donné un point m dans le plan image $\mathcal{P}$, ses correspondants possibles dans le plan $\mathcal{P}'$ se situent sur la droite épipolaire l'_m.

La matrice fondamentale

Soit $(\Delta\mathbf{R} \quad \Delta\mathbf{t})$ le déplacement relatif (ou *mouvement*) de la seconde caméra par rapport à la première caméra, exprimé dans le repère de la seconde caméra. Les trois vecteurs $\overrightarrow{CC'} = \Delta\mathbf{t}$, $\overrightarrow{CM}$ et $\overrightarrow{C'M}$ étant coplanaires, nous pouvons écrire la relation :

$$(\Delta\mathbf{t} \wedge \Delta\mathbf{R}\mathbf{M}_c).\mathbf{M}_{c'} = 0, \tag{2.5}$$

où $\mathbf{M}_c$ est le vecteur $\overrightarrow{CM}$ exprimé dans le repère de la première caméra, et $\mathbf{M}_{c'}$ le vecteur $\overrightarrow{C'M}$ exprimé dans le repère de la seconde caméra (le vecteur $\overrightarrow{CM}$ ayant pour coordonnées $\Delta\mathbf{R}\mathbf{M}_c$ dans le repère de la seconde caméra).

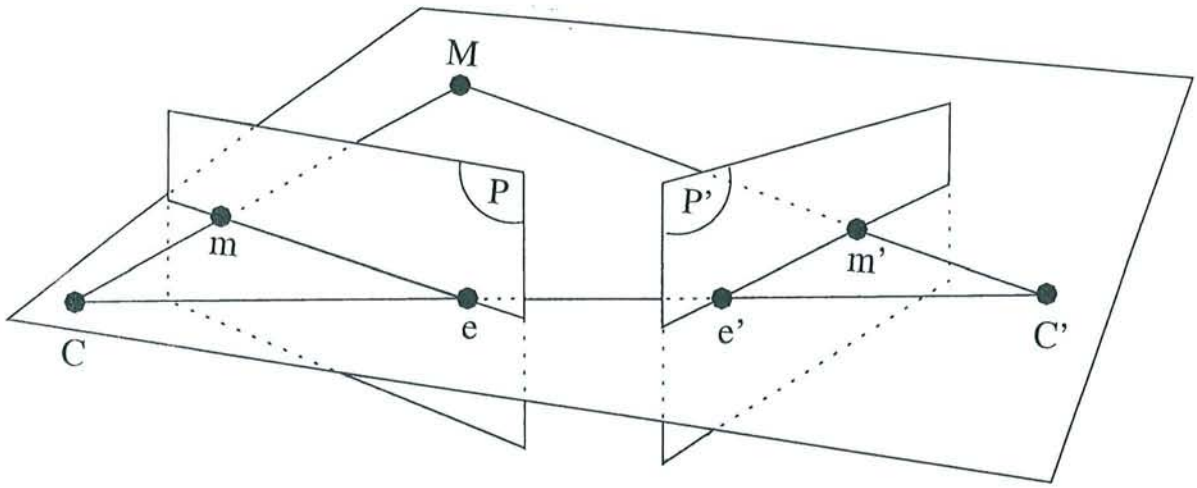


FIG. 2.4 – *La contrainte épipolaire.*

Comme $\Delta\mathbf{t}\wedge\mathbf{x} = \Delta\mathbf{T}\mathbf{x}$ avec $\Delta\mathbf{T} = \begin{pmatrix} 0 & -\Delta t_z & \Delta t_y \\ \Delta t_z & 0 & -\Delta t_x \\ -\Delta t_y & \Delta t_x & 0 \end{pmatrix}$, l'équation (2.5) peut s'écrire :

$$\mathbf{M}_{c'}^T \Delta\mathbf{T}\Delta\mathbf{R}\mathbf{M}_c = 0.$$

Si $\mathbf{q}$ et $\mathbf{q}'$ sont les coordonnées rétiniennes des points m et m' (respectivement), on a d'après l'équation (2.4), $\mathbf{q} = \mathbf{A}\mathbf{M}_c$ et $\mathbf{q}' = \mathbf{A}'\mathbf{M}_{c'}$, où $\mathbf{A}$ et $\mathbf{A}'$ sont les matrices des paramètres intrinsèques des deux caméra. Nous obtenons alors l'équation de Longuet-Higgins :

$$\mathbf{q}'^T\mathbf{F}\mathbf{q} = 0, \tag{2.6}$$

où $\mathbf{F} = \mathbf{A}'^{-T}\Delta\mathbf{T}\Delta\mathbf{R}\mathbf{A}^{-1}$ est la matrice fondamentale ($(\mathbf{X}^{-1})^T$ est noté $\mathbf{X}^{-T}$). Cette équation exprime la contrainte épipolaire, qui est que le point $\mathbf{q}'$ correspondant au point $\mathbf{q}$ appartient à la droite $\mathbf{l}'_\mathbf{q} = \mathbf{F}\mathbf{q}$ ($\mathbf{l}'_\mathbf{q}$ étant le vecteur des coefficients de la droite l'_q).

Détermination

Les propriétés de la matrice fondamentale ont été étudiées en détail par de nombreux auteurs (en particulier [Luong92]). Il s'agit d'une matrice 3×3 de rang 2, et donc de déterminant nul. L'équation (2.6) peut s'écrire :

$$\mathbf{U}^T\mathbf{f} = 0, \tag{2.7}$$

où $\mathbf{U} = [uu',vu',u',uv',vv',v',u,v,1]^T$ et $\mathbf{f} = [F_{11},F_{12},F_{13},F_{21},F_{22},F_{23},F_{31},F_{32},F_{33}]^T$, (u,v) et (u',v') étant les coordonnées pixel des points $\mathbf{q}$ et $\mathbf{q}'$. Si nous disposons de huit appariements de points, nous pouvons donc déterminer une solution unique pour $\mathbf{F}$, définie à un facteur d'échelle près (l'un des neuf coefficients est fixé à 1), en résolvant un système linéaire composé de huit équations (2.7). En général, on dispose d'un nombre d'appariements $(\mathbf{q}_i,\mathbf{q}'_i)$ bien plus grand que huit, et l'équation est résolue aux moindres carrés en minimisant le critère linéaire :

$$\min_{\mathbf{F}} \sum_i (\mathbf{q}'^T_i \mathbf{F}\mathbf{q}_i)^2. \tag{2.8}$$

Malheureusement, le critère linéaire possède deux défauts qui le rendent très sensible au bruit : d'une part, il ne tient pas compte de la contrainte de rang ($\det(\mathbf{F}) = 0$), ce qui entraîne une incohérence de la géométrie épipolaire au voisinage des épipoles et, d'autre part, il est non normalisé, ce qui entraîne un biais dans la localisation des épipoles. Luong propose donc de minimiser un critère non linéaire, qui est la distance euclidienne d'un point à la droite épipolaire de son correspondant. Détaillons ce critère, que nous utiliserons au chapitre 6. La distance euclidienne d'un point $\mathbf{q}'$ de la seconde image à la droite épipolaire $\mathbf{l}'_\mathbf{q} = (l'_1,l'_2,l'_3)^T = \mathbf{F}\mathbf{q}$ est donnée par :

$$d(\mathbf{q}',\mathbf{l}'_\mathbf{q}) = \frac{|\mathbf{q}'^T\mathbf{l}'_\mathbf{q}|}{\sqrt{(l'_1)^2 + (l'_2)^2}} \tag{2.9}$$

On peut envisager dans un premier temps de minimiser le critère suivant :

$$\min_\mathbf{F} \sum_i d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i)$$

Cependant, contrairement au critère linéaire, ce critère n'est pas symétrique puisqu'il ne détermine que les droites épipolaires dans la seconde image : il s'ensuit une incohérence de la géométrie épipolaire obtenue entre les deux images. Pour obtenir une géométrie épipolaire cohérente, nous pouvons inverser le rôle des deux images en transposant la matrice fondamentale. Ceci conduit au critère suivant, qui opère simultanément sur les deux images :

$$\min_\mathbf{F} \sum_i (d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i) + d^2(\mathbf{q}_i,\mathbf{F}^T\mathbf{q}'_i)) \tag{2.10}$$

Ce critère est normalisé, au sens où il ne dépend pas du facteur d'échelle choisi pour $\mathbf{F}$. Nous pouvons aussi prendre en compte le fait que $\mathbf{F}$ est de rang deux en paramétrant cette matrice par le nombre exact de variables indépendantes, qui est de sept, une fois que le facteur d'échelle a été pris en compte (la troisième ligne de la matrice est alors écrite comme une combinaison linéaire des deux premières). Luong montre que le critère non linéaire est beaucoup plus stable que le critère linéaire. Il met toutefois en évidence l'instabilité du calcul de la matrice fondamentale pour certains types de mouvements : les mouvements de faible amplitude, les mouvements dont la translation est parallèle au plan image et les translations pures.

Calibration faible

On dit qu'un couple de caméras pour lesquelles la matrice fondamentale est connue est *faiblement calibré*. On montre que, à partir de la matrice $\mathbf{F}$, on peut accéder à une reconstruction tridimensionnelle de la scène, mais que cette reconstruction est définie à une transformation homographique près [Faugeras92]. L'espace ambiant est alors modélisé comme un espace *projectif* de dimension 3, $\mathbb{P}^3$, obtenu en complétant l'espace affine usuel à trois dimensions X_3 par un plan projectif, dit plan à l'infini et noté Π_∞.

Pour remonter à la structure euclidienne de la scène, il faut déterminer la métrique de la rétine, c'est-à-dire les paramètres intrinsèques de la caméra. On se sert là encore de la matrice fondamentale, qui impose deux contraintes polynomiales sur les paramètres intrinsèques. Ces contraintes sont les équations de Kruppa, que nous voyons à présent.

2.3.2 Les équations de Kruppa

L'introduction d'un invariant projectif important, la *conique absolue*, permet d'obtenir une contrainte géométrique exprimant le fait que le mouvement entre les repères des deux caméras est un déplacement rigide, et non une relation projective linéaire quelconque [Maybank et al.92]. Cette conique, notée Ω, est située dans le plan à l'infini Π_∞ et a pour équations :

$$\mathcal{X}^2 + \mathcal{Y}^2 + \mathcal{Z}^2 = 0, \qquad \mathcal{T} = 0. \tag{2.11}$$

Notons qu'elle ne possède aucun point réel. Soit M un point de Ω : par définition, ses coordonnées projectives sont de la forme $\tilde{\mathbf{M}} = (\mathbf{M}^T \quad 0)^T$ avec $\mathbf{M}^T\mathbf{M} = 0$. On montre que l'image du point M dans la rétine appartient à la conique ω d'équation $\mathbf{q}^T\mathbf{K}\mathbf{q} = 0$, où $\mathbf{K} = \mathbf{A}^{-T}\mathbf{A}^{-1}$ ne dépend que des paramètres intrinsèques de la caméra. En effet, d'après l'équation (2.4), $\mathbf{q}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{q} = \tilde{\mathbf{M}}^T(\mathbf{R} \quad \mathbf{t})^T(\mathbf{R} \quad \mathbf{t})\tilde{\mathbf{M}} = \mathbf{M}^T\mathbf{R}^T\mathbf{R}\mathbf{M}$. Comme $\mathbf{R}$ est une matrice de rotation, $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ et donc $\mathbf{q}^T\mathbf{K}\mathbf{q} = \mathbf{M}^T\mathbf{M} = 0$. L'image ω de la conique absolue ne dépend donc ni de la position, ni de l'orientation de la caméra : cette propriété très intéressante va nous permettre d'établir les équations de Kruppa.

On sait que l'ensemble des tangentes à une conique forme aussi une conique, dite duale. Soit $\mathbf{B}$ la matrice de la conique duale de ω (c'est la matrice des cofacteurs de $\mathbf{K}$). Soit (em) une droite épipolaire de la première image. Cette droite est tangente à ω si et seulement si elle appartient à la conique duale de ω, ce qui s'écrit :

$$(\mathbf{e} \wedge \mathbf{q})^T\mathbf{B}(\mathbf{e} \wedge \mathbf{q}) = 0. \tag{2.12}$$

D'autre part, les tangentes menées de chacun des épipoles e et e' à l'image de Ω se correspondent puisqu'elles sont les traces dans les deux plans images des deux plans tangents menés par la droite (CC') à Ω. Si m appartient à l'une des deux tangentes menées de e à ω dans la première image, sa droite épipolaire représentée par $\mathbf{F}\mathbf{q}$ est donc tangente à la projection de Ω dans la deuxième image, qui est égale à ω si les paramètres intrinsèques de la caméra sont constants entre les deux vues (invariance de ω par déplacement rigide). Ceci s'écrit :

$$\mathbf{q}^T\mathbf{F}^T\mathbf{B}\mathbf{F}\mathbf{q} = 0. \tag{2.13}$$

Les équations (2.12) et (2.13) induisent deux équations quadratiques en les coefficients de $\mathbf{B}$, qui sont les équations de Kruppa. Puisque les paramètres intrinsèques sont au nombre de cinq, trois mouvements sont en théorie nécessaires pour les déterminer. Pour le modèle de caméra simplifié à quatre paramètres intrinsèques, deux mouvements sont suffisants. Cependant, les méthodes de résolution des systèmes d'équations polynomiales sont encore peu développées et nécessitent souvent que le nombre d'équations soit identiques au nombre d'inconnus. D'autre part, les simulations effectuées par Luong dans sa thèse semblent montrer que les matrices fondamentales doivent être déterminées à partir d'appariements dont l'erreur de localisation ne dépasse pas le pixel. Enfin, il existe des mouvements de caméra pour lesquels ces équations dégénèrent (mouvements de translation pure, mouvements plans, mouvements de rotation pure ou de faible amplitude, mouvements orbitaux ...). Ces mouvements sont répertoriés dans [Sturm97]. L'autocalibration est donc une méthode relativement instable dans la pratique.

Lorsqu'on dispose d'un grand nombre de mouvements, l'utilisation d'approches itératives comme le filtre de Kalman étendu ou l'ajustement de faisceau permet d'obtenir de meilleurs résultats (voir chapitre 3) . Cependant, Bougnoux observe dans un cas pratique une instabilité sur l'estimation de la focale [Bougnoux98] : dans son expérience, qui utilise la technique d'ajustement de faisceau à partir de six images, α_u peut prendre des valeurs comprises entre 500 et 800 suivant

l'initialisation, tout en produisant un résidu très faible pour l'ensemble des contraintes. Notons toutefois que nous pouvons très bien envisager de déterminer les paramètres intrinsèques de la caméra par une technique basée modèle, tout en calculant le point de vue de la caméra selon une technique basée images.

2.3.3 Calcul du mouvement

La technique basée images ne permet pas d'obtenir directement la position et l'orientation de la caméra dans le repère de la scène: elle ne calcule que le déplacement relatif ($\Delta\mathbf{R}$,$\Delta\mathbf{t}$), ou *mouvement* de la seconde caméra par rapport à la première caméra. D'autre part, comme les dimensions de la scène ne sont pas connues, la translation $\mathbf{t}$ est définie à un facteur d'échelle près, comme l'illustre la figure 2.5. Une seule mesure dans la scène permet de lever l'indétermination du facteur d'échelle. Le positionnement de la première caméra dans le repère de la scène, nécessaire pour les applications de RA, ne peut par contre être obtenu qu'à partir d'une des techniques basées modèle vues précédemment.

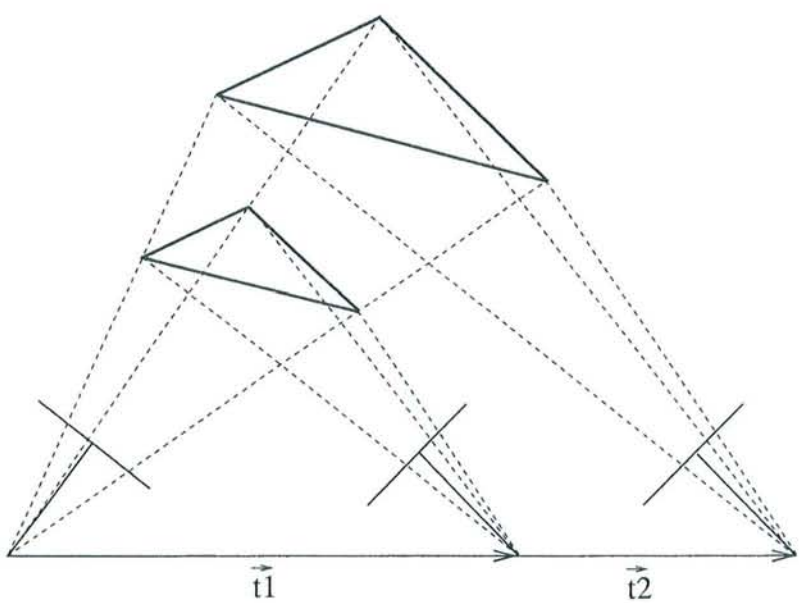


FIG. 2.5 – *Illustration du problème d'échelle lié aux techniques basées images : les translations* $\mathbf{t_1}$ *et* $\mathbf{t_2}$ *donnent lieu à la même image pour des scènes de tailles différentes.*

Décomposition de la matrice essentielle

D'après l'équation (2.6), $\Delta\mathbf{T}\Delta\mathbf{R} = \mathbf{A}^T\mathbf{F}\mathbf{A}$ lorsque $\mathbf{A}' = \mathbf{A}$. La matrice $\mathbf{E} = \Delta\mathbf{T}\Delta\mathbf{R}$, appelée *matrice essentielle*, peut donc être obtenue directement une fois que la matrice fondamentale et les paramètres intrinsèques sont calculés. Les paramètres du mouvement peuvent alors être obtenus en décomposant cette matrice (il s'agit d'un problème classique). Comme $\Delta\mathbf{T}^T\Delta\mathbf{t} = 0$, nous avons la propriété $\mathbf{E}^T\Delta\mathbf{t} = 0$ et pouvons donc déterminer la direction de la translation en résolvant le problème aux valeurs propres $\min_{||\Delta\mathbf{t}||=1} ||\mathbf{E}^T\Delta\mathbf{t}||^2$. Nous pouvons ensuite retrouver la rotation relative $\Delta\mathbf{R}$ en minimisant par une méthode directe l'erreur $||\mathbf{E}-\Delta\mathbf{T}\Delta\mathbf{R}||$ [Toscani87], ou en décomposant $\mathbf{E}$ en valeurs singulières [Tsai et al.84].

Méthode itérative

Le calcul du mouvement par la méthode précédente part de l'hypothèse que nous disposons de la matrice fondamentale, et est donc fondée directement sur son calcul. Nous pouvons aussi exploiter indirectement le calcul de matrice fondamentale pour obtenir le mouvement [Luong92]. La technique consiste à minimiser le critère linéaire (2.8) ou le critère non linéaire (2.10) par rapport aux cinq paramètres du mouvement, en substituant $\mathbf{A}^{-T}\Delta\mathbf{T}\Delta\mathbf{R}\mathbf{A}^{-1}$ à $\mathbf{F}$ dans ces critères. Les avantages sont une minimisation plus efficace et plus précise car seuls cinq paramètres interviennent : trois pour la rotation et deux pour la translation (la troisième composante du vecteur translation est fixée à 1, puisque seule la direction de la translation peut être connue). Les résultats obtenus par cette méthode sont donc potentiellement les plus précis, mais ils sont bien sûr sensibles à l'initialisation. Le mouvement obtenu par décomposition de la matrice essentielle est une initialisation possible.

Notons que, par extension, nous pouvons aussi minimiser les critères (2.8) ou (2.10) par rapport aux paramètres intrinsèques de la caméra, en plus des cinq paramètres du mouvement, ce qui revient à imposer un paramétrage qui tient compte directement des contraintes de rigidité. La matrice fondamentale ne comportant que sept degrés de liberté, nous ne pouvons en fait obtenir que deux des paramètres intrinsèques à partir d'un seul mouvement (on peut par exemple fixer le point principal au centre de l'image). Avec deux mouvements, nous pouvons calculer les quatre paramètres du modèle simplifié.

2.3.4 Géométrie de trois caméras

Lorsqu'on considère trois caméras, on peut calculer trois matrices fondamentales qui décrivent la géométrie épipolaire entre deux vues. Malheureusement, ces matrices ne décrivent pas la géométrie *globale* des correspondances. La figure 2.6 représente la géométrie de trois vues : les trois centres optiques C_1, C_2 et C_3 forment un plan, dit plan trifocal, qui dessine sur les trois rétines les trois droites trifocales t_1, t_2 et t_3 sur lesquelles se trouvent les six épipoles, deux par image, $e_{1,2}$,$e_{1,3}$ pour la première, $e_{2,1}$,$e_{2,3}$ pour la deuxième et $e_{3,1}$,$e_{3,2}$ pour la troisième. Dans le cas de deux caméras, la matrice fondamentale permet de réduire la recherche d'un point m' correspondant à un point m en se limitant à la droite épipolaire de m. Dans le cas de trois vues, on a une redondance supplémentaire qui permet de réduire encore plus la recherche : étant donné un couple de points images (m_1,m_2) des deux premières vues satisfaisant la contrainte épipolaire, nous pouvons construire directement le point image m_3 de la troisième vue correspondant au même point 3-D à l'aide d'un tenseur, dit *tenseur trifocal*, qui est donc une extension de la matrice fondamentale à trois vues [Hartley97].

Les travaux portant sur la géométrie de trois caméras et le tenseur trifocal sont très récents et leur état d'avancement est moins important que celui de la matrice fondamentale. Notons que la connaissance du tenseur trifocal, comme celle de la matrice fondamentale, ne permet de reconstruire la scène qu'à une transformation projective arbitraire près. L'utilisation du tenseur trifocal dans un contexte multi-images est par exemple illustrée dans [Fitzgibbon et al.98].

2.4 Précision de la calibration

Les techniques de calibration que nous venons de décrire sont confrontées aux limites du modèle sténopé, ainsi qu'aux imprécisions obtenues dans l'extraction des mesures de référence.

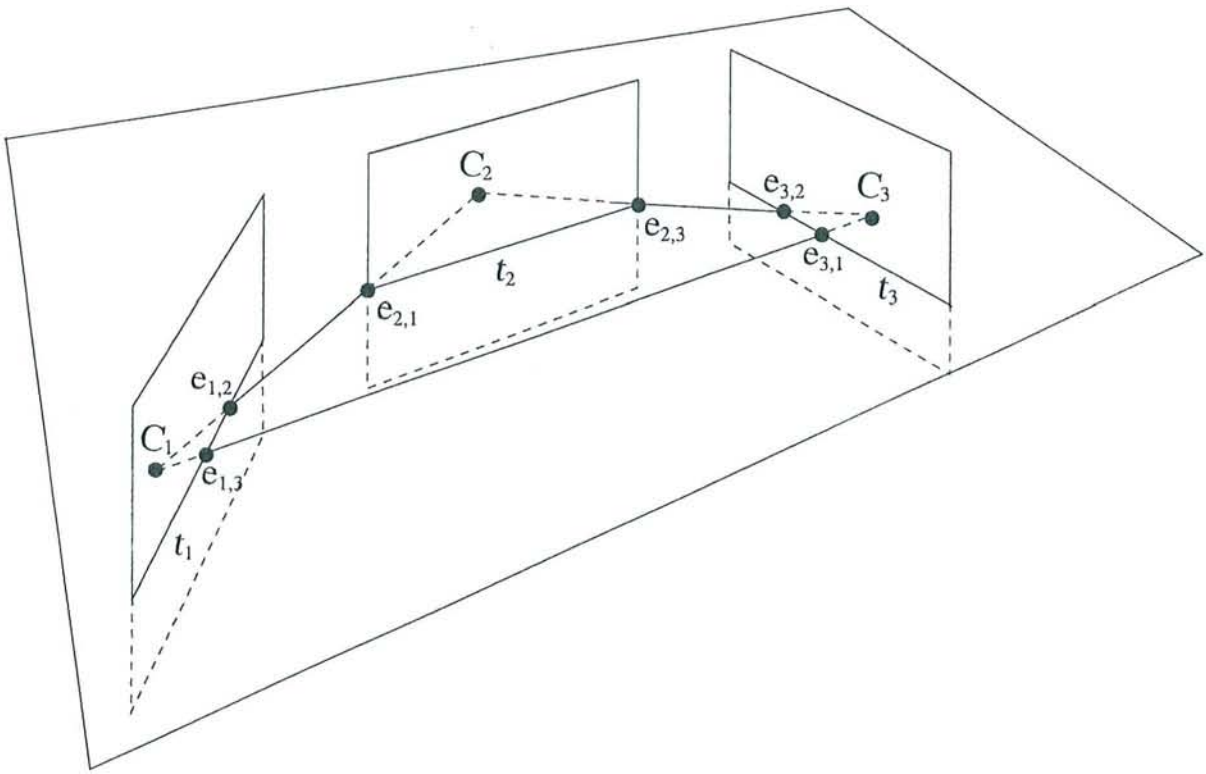


FIG. 2.6 – *La géométrie trifocale.*

2.4.1 Adéquation du modèle sténopé

Le modèle sténopé classique est un modèle simplifié de la caméra, qui ne rend pas compte des distorsions optiques, ni de certains phénomènes liés à l'utilisation d'objectifs à focale variable.

Distorsion optique

Le modèle sténopé suppose que la caméra n'a pas de distortion optique. Pourtant, pour certaines caméras, cette distortion n'est pas négligeable. Celle-ci transforme les coordonnées images idéales (x_c,y_c) en coordonnées :

$$\begin{cases} x'_c = x_c + \delta_x(x_c,y_c), \\ y'_c = y_c + \delta_y(x_c,y_c). \end{cases}$$

En particulier, la distortion *radiale* est le plus souvent prise en compte. Celle-ci augmentant avec l'angle de champ, elle est surtout importante sur les bords de l'image. Elle est représentée par un polynôme :

$$\begin{cases} \delta_x = x_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + ...), \\ \delta_y = y_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + ...), \end{cases}$$

où $r = \sqrt{x^2 + y^2}$. Généralement, une bonne précision peut être obtenue en ne considérant que le premier terme du polynôme.

Toutefois, la prise en compte des distortions pour calibrer la caméra fait perdre la linéarité entre les coordonnées 3-D et 2-D, ce qui complique considérablement la procédure de calibration [Weng et al.92], mais aussi d'autres traitements ultérieurs comme le calcul du point de vue ou

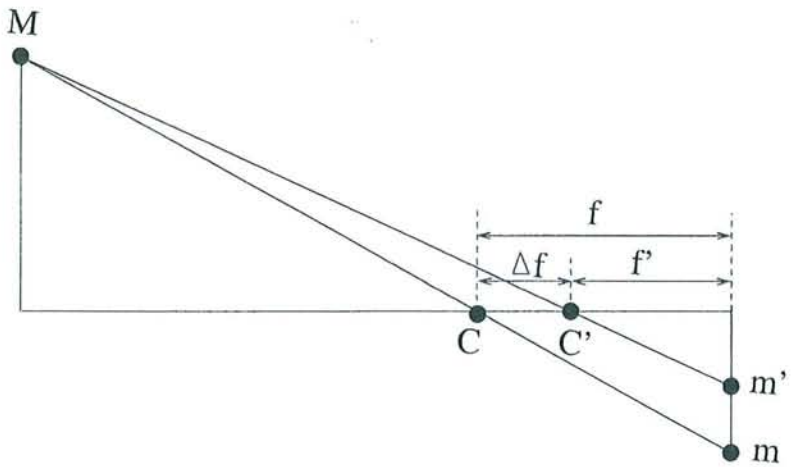


FIG. 2.7 – *Déplacement du centre optique par changement de focale.*

l'appariement de points en stéréo-vision. Par exemple, la droite épipolaire n'est plus une droite si l'on considère les distorsions. D'autre part, lorsque la distortion est faible, si le bruit dans l'extraction des primitives est relativement grand ou que le nombre de points utilisés pour la calibration est faible, les résultats de calibration basés sur le modèle de distortion peuvent être moins bons que ceux basés sur le modèle linéaire [Hung et al.90].

Pour ne pas avoir à prendre en compte les distortions dans la chaîne des traitements, Peuchot propose de *corriger* ces distortions [Brand et al.94, Peuchot94]. Pour cela, il utilise l'image d'une grille qui lui permet d'établir le champ des corrections à appliquer sur l'image distordue. Notons enfin que les caméras les plus récentes, comme celles que nous utilisons, présentent généralement très peu de distortion.

Objectifs à focale variable

[Lavest et al.92] ont montré que le modèle sténopé ne permettait pas de décrire avec une bonne précision les objectifs à focale variable. La représentation de ce modèle utilisée en figure 2.1 est une représentation inversée de celle qu'utilisent les opticiens, permettant d'obtenir directement l'image à l'endroit, alors qu'en fait elle est inversée par le système optique (figure 2.7). Si l'on considère la figure 2.7, on constate qu'un changement de focale doit se traduire par une translation Δf du centre optique de l'image selon l'axe optique : en particulier, une augmentation de la focale devrait correspondre à un rapprochement relatif de l'objet. Or, des résultats expérimentaux obtenus par Lavest et al. montrent qu'en fait l'objet s'éloigne lorsque la focale augmente, et que cet éloignement $\Delta \mathbf{t}_z$ n'est pas proportionnel à Δf (typiquement, une variation Δf de 4.2 cm correspond à un éloignement $\Delta \mathbf{t}_z$ de 10.6 cm!). Ceci démontre la non adéquation du modèle sténopé à la réalité. Les auteurs proposent de le remplacer par un modèle plus complexe de lentille épaisse (*modèle épais*), qui permet d'expliquer les résultats obtenus par le modèle sténopé.

Malheureusement, tout comme la prise en compte des distortions, la prise en compte du modèle épais complique la plupart des tâches de vision. D'autre part, Lavest et al. considèrent que l'utilisation du modèle sténopé reste acceptable pour la plupart des applications si l'interprétation qu'on en fait se réfère au modèle épais. En particulier, dans le cadre qui nous intéresse, nous verrons que le modèle sténopé permet d'obtenir des résultats de composition visuellement corrects et cohérents sur la séquence.

2.4.2 Incertitude sur les paramètres intrinsèques

D'autres techniques que celles que nous avons présentées permettent d'obtenir les paramètres intrinsèques de la caméra : en particulier, pour un objectif à focale variable, Tsai propose de retrouver les coordonnées du point principal par changement de focale [Tsai87]. Dans le cas d'une caméra parfaitement fidèle au modèle sténopé, pour laquelle l'axe optique est perpendiculaire au plan image et le système optique est stable par changement de focale, alors le point principal est fixe dans le plan image durant un zoom. D'après l'équation 2.4,

$$\begin{cases} u = af + u_0 \\ v = bf + v_0 \end{cases},$$

où a et b sont constants par variation de f. Durant un zoom, chaque point de l'image suit donc une trajectoire radiale le long d'une droite passant par le point principal. Si nous capturons plusieurs images d'une même scène pour différentes valeurs de la focale, et que nous parvenons à apparier des points entre les différentes images, les droites déterminées par les points appariés se coupent en un point, qui est le point principal.

Malheureusement, [Willson et al.93] montrent expérimentalement que le point principal obtenu par cette méthode n'est pas détecté au même endroit que le point principal obtenu par la résolution du système d'équations $\mathbf{q}_i = \mathbf{P}\mathbf{M}_i$. Les auteurs proposent d'ailleurs 13 autres définitions du point principal (qui sont en fait 13 façons différentes de l'obtenir), qui conduisent toutes à des positions du point principal différentes (jusqu'à plusieurs dizaines de pixels d'écart!).

La méthode de résolution utilisée ainsi que la non adéquation du modèle de projection perspective classique à la réalité sont bien sûr partiellement responsables de ces écarts. Les mesures utilisées sont généralement fiables du fait de l'utilisation de motifs facilement détectables (à condition d'être bien visibles), et présentant des points d'intérêt mesurables avec une grande précision. Par contre, la position et l'orientation de la mire de calibration a une influence non négligeable sur les valeurs obtenues [Luong92, Li et al.95].

Une autre source d'imprécisions est l'estimation simultanée des paramètres intrinsèques et extrinsèques de la caméra : en effet, même si l'ensemble des paramètres, considérés comme un tout, donne un minimum, chaque paramètre, pris individuellement, n'est pas nécessairement fiable [Luong92]. Si l'on suppose que la fonction minimisée $f(\lambda_1,...,\lambda_N)$ est telle qu'au voisinage du minimum, $|\frac{\partial f}{\partial \lambda_i}| \ll |\frac{\partial f}{\partial \lambda_j}|$, alors la valeur estimée du paramètres λ_i peut être largement perturbée pour compenser une erreur en le paramètre λ_j.

Enfin, les paramètres obtenus lors du processus de calibration peuvent varier dans le temps : sans même parler de changements de zoom intentionnels, des changements incontrôlés peuvent se produire, dus à des variations thermiques et mécaniques. En outre, si la scène filmée n'est pas située à la même profondeur que la mire utilisée pour la calibration, une mise au point de la lentille peut être nécessaire, faisant alors généralement varier la focale et le point principal [Enciso et al.93, Willson et al.93].

2.4.3 Stabilité des paramètres extrinsèques

Supposons que nous connaissions précisément les paramètres intrinsèques de la caméra : quelle est alors la stabilité du calcul du point de vue ou du mouvement vis-à-vis de l'imprécision sur les données 3-D/2-D ou 2-D/2-D utilisées?

Pour les méthodes basées images, nous pouvons citer les conclusions de Luong, basées sur des statistiques portant sur 200 mouvements :

- le calcul du mouvement de la caméra est nettement plus stable que celui de la matrice fondamentale,

- la détermination de la rotation est plus précise que celle de la direction de la translation,
- les résultats obtenus par la méthode itérative qui consiste à minimiser l'un des critères (2.8) ou (2.10) par rapport aux cinq paramètres du mouvement sont potentiellement plus précis que ceux obtenus en décomposant la matrice essentielle. Ils sont cependant plus sensibles à l'initialisation du point de vue.

Pour les méthodes basées modèle, les techniques itératives sont généralement plus précises que les méthodes directes, à condition que le point de vue initial ne soit pas trop éloigné du point de vue attendu. Comme les techniques basées modèle utilisent une connaissance partielle de la scène, on peut s'attendre à ce qu'elles génèrent des résultats plus précis que ceux obtenus par les méthodes basées uniquement sur les données image. Toutefois, cela suppose que les primitives 3-D soient correctement localisées, et bien distribuées. Si les primitives de référence sont incorrectes, cela risque au contraire de fausser les résultats. Si elles sont mal distribuées, nous verrons au chapitre 5 que le point de vue obtenu peut être complètement faux. Néanmoins, en dehors de ces deux configurations, une très bonne précision peut être obtenue sur le point de vue pour les méthodes basées modèle. En particulier, pour certaines applications médicales, la précision du recalage peut atteindre le millimètre [Kerrien et al.91] (il s'agit toutefois d'un cadre bien particulier où les instruments de mesures sont parfaitement contrôlés).

2.5 Objectifs de précision et critères d'évaluation

2.5.1 Évaluation de la calibration

L'évaluation de la précision des paramètres obtenus est un problème délicat. Si la scène est posée sur une table micrométrique (par exemple), et que la caméra est parfaitement contrôlée, nous pouvons comparer les résultats obtenus avec les résultats attendus. Cependant, pour les applications grandeur réelle, nous n'avons bien souvent aucune idée des paramètres intrinsèques de la caméra (si ce n'est que le point principal est à peu près au centre de l'image), et le point de vue n'est la plupart du temps connu qu'approximativement, voire pas du tout! Il n'est donc pas possible de décider immédiatement si une matrice de projection est correcte ou non. Il existe toutefois un certain nombre de critères objectifs qui peuvent nous donner des *indices* sur la qualité de cette matrice.

L'erreur de reprojection

Nous pouvons mesurer les distances en pixels entre les points mesurés dans l'image et les projections de leur correspondant 3-D. Nous appelons ces distances *résidus*, notés r_i, et définissons l'*erreur de reprojection* err_{proj} par la relation :

$$err_{proj} = \sqrt{\frac{1}{n}\sum_i r_i^2},$$

qui évalue globalement la qualité de la reprojection.

Cependant, l'erreur de reprojection peut être faible alors que la matrice de projection est incorrecte : ceci se produit par exemple lorsque les paramètres de la caméra sont calculés à partir d'un objet de référence petit par rapport à la distance objet-caméra. Les paramètres obtenus sont alors localement corrects pour l'objet de référence (et l'erreur de reprojection sur cet objet est petite puisque ce sont justement les résidus de points de l'objet qui ont été minimisés), mais incorrects par rapport à l'ensemble de la scène contenue dans le champ de vision de la caméra.

L'incrustation d'un objet virtuel dans une zone éloignée de l'objet de référence peut alors être complètement fausse (nous illustrerons ce problème par la suite).

Les paramètres du point de vue

Nous pouvons, à partir de la matrice de projection perspective, extraire les six paramètres du point de vue de la caméra : angles d'Euler et translation. Or, bien souvent, il est possible d'avoir une idée approximative de ces paramètres : par exemple, nous connaissons parfois grossièrement la distance entre la caméra et le modèle de référence, ou la hauteur de la caméra par rapport au sol de la scène (qui peut correspondre à la hauteur d'un trépied ou de l'oeil de l'observateur) : cela nous permet au moins de vérifier si la translation obtenue n'est pas aberrante. De même, nous pouvons savoir à peu près en regardant les images si la caméra pointe vers le haut ou vers le bas du modèle, vers sa gauche ou vers sa droite : cela donne une première idée des angles d'Euler.

Dans le cas d'une *séquence* d'images, nous pouvons aussi vérifier la cohérence des courbes d'évolution des paramètres avec ce qui est observé dans le film : trajectoire approximative de la caméra, mouvements réguliers ou saccadés etc. Pour cette raison, nous tracerons systématiquement les courbes des six paramètres du point de vue pour valider nos résultats.

La scène reconstruite

Une autre façon d'évaluer la précision des résultats consiste à *reconstruire* une partie de la scène filmée par triangulation. Le principe est très simple : connaissant un point d'une image, la matrice de projection nous permet de savoir sur quelle droite se trouve le point 3-D dont il est la projection. Si nous parvenons à apparier ce point dans une autre image, nous obtenons une deuxième droite et l'intersection des deux droites est le point 3-D reconstruit. Dans la pratique, les droites ne se coupent généralement pas : on prend alors, par exemple, le milieu du plus petit segment de droite les reliant.

Si la métrique de la scène est connue, nous pouvons alors très précisément comparer les mesures de la scène reconstruite avec celles de la scène réelle : plus la calibration des caméras est correcte, plus les deux scènes doivent se ressembler. Si la métrique de la scène n'est pas connue, nous pouvons aussi vérifier les angles entre les droites (angles droits, droites parallèles ...).

Nous utiliserons ce critère d'évaluation au chapitre 7. Cependant, Bougnoux montre dans un cas pratique, qu'une erreur obtenue sur les paramètres de la caméra peut avoir très peu d'influence sur la qualité de la reconstruction (notamment dans le cas d'une confusion entre le zoom et la translation le long de l'axe optique). Ce critère n'est donc pas toujours très significatif [Bougnoux98].

2.5.2 Évaluation de l'incrustation

Les paramètres de la caméra sont donc en général relativement peu précis, et leur adéquation à la réalité est difficile à évaluer. Cependant, la vocation des applications de RA est de présenter à l'oeil humain des résultats de composition qui lui paraissent géométriquement cohérents. Cet objectif ne rend pas nécessairement le problème plus simple : le système de perception humain est en effet très perspicace pour déceler les plus petites incohérences de l'image. Si l'objet virtuel est sensé être immobile dans les images d'une séquence, une petite erreur sur les paramètres de la caméra peut au contraire donner l'impression à l'observateur que celui-ci glisse ou sautille! En outre, si l'objet virtuel est incrusté à la place de l'objet ayant servi de référence pour la calibration (comme pour l'application des Ponts de Paris), le résultat peut être visuellement correct même si les paramètres de la caméra ne le sont pas, à condition que l'erreur de reprojection soit faible pour

l'objet de référence. Pour ce type d'incrustation, l'"erreur visuelle" est très proche de l'erreur de reprojection qui, comme pour l'erreur de reconstruction, peut être petite alors que la calibration est incorrecte.

Afin d'évaluer visuellement les résultats de nos algorithmes, nous avons donc systématiquement réalisé des séquences vidéo montrant la projection du modèle de référence, ainsi que des incrustations d'objets virtuels, le plus souvent dans des zones éloignées du modèle de référence. Des images extraites sont présentées dans ce manuscrit, les séquences complètes étant disponibles au format MPEG sur notre site internet, à l'adresse :

`http://www.loria.fr/~gsimon/these.html`

Chapitre 3

Intérêts et limites des systèmes de RA existants

Nous décrivons l'architecture générale d'un système de RA, puis différentes implémentations de ce schéma à travers la littérature (quelques produits commerciaux sont aussi évoqués). Lorsque les objets virtuels sont projetés dans une séquence d'images, les techniques présentées précédemment sont utilisées dans un processus bien particulier : on parle de *recalage temporel* pour les méthodes basées modèle, et de calcul de la structure et du mouvement multi-images (*multi-frame structure from motion*) pour les méthodes basées images. Nous présentons ces différentes techniques, puis menons une discussion comparative servant de base à l'élaboration d'une nouvelle stratégie de recalage temporel.

3.1 Architecture générale d'un système de RA

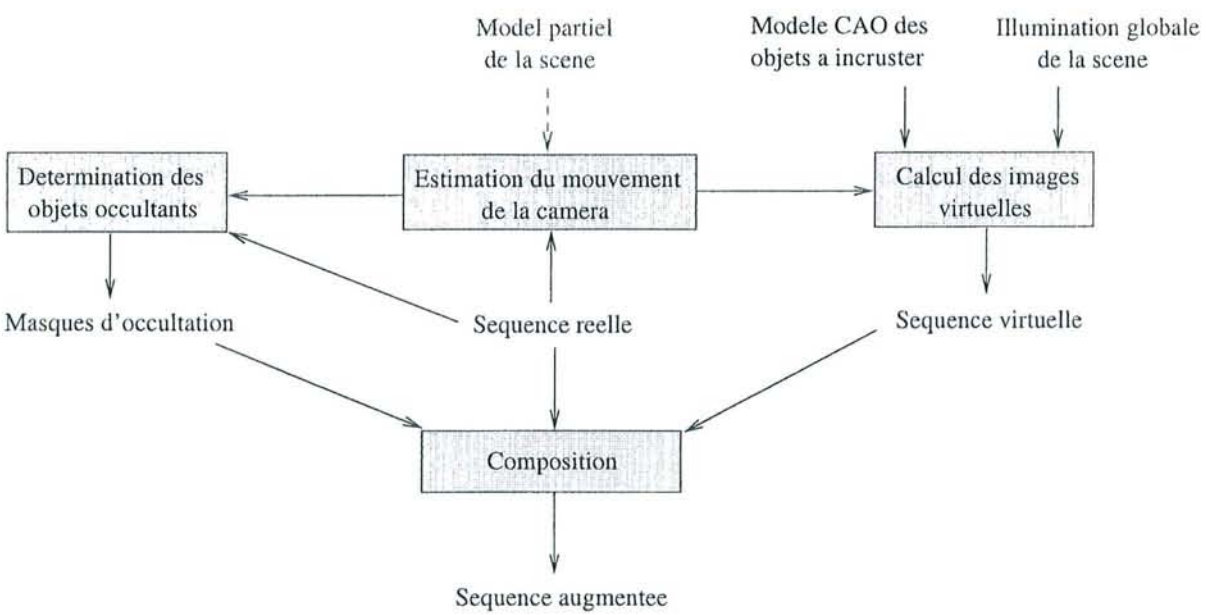


FIG. 3.1 – *Schéma général d'un système de RA.*

La figure 3.1 présente les différents blocs d'un système de RA pour un système à focale fixe

(nous discuterons du problème à focale variable au chapitre 7). Ce système est constitué de trois parties distinctes : l'estimation du mouvement de la caméra (utilisant éventuellement la connaissance d'un modèle partiel de la scène), le calcul des images synthétiques et la détermination des masques d'occultation utilisés pour la composition. Comme on le voit sur le schéma, la détermination du mouvement de la caméra est au cœur du système, puisque les paramètres de la caméra sont utilisés pour calculer les images virtuelles mais aussi pour retrouver les masques d'occultation.

Ce schéma concerne principalement les applications de type post-production : dans une application en temps réel, les occultations ne sont généralement pas traitées et la partie synthèse se résume le plus souvent à la projection du modèle filaire des objets [Peuchot95, Feiner et al.93] ou de simples annotations [Rose et al.94, Ravela et al.96]. D'autre part, les trois composantes sont présentées comme trois processus exécutés à part sur l'ensemble de la séquence, alors que pour certains systèmes (et nécessairement pour les systèmes en temps réel) ceux-ci sont exécutés *en ligne*, image après image.

Nous ne donnerons ici qu'un bref aperçu des aspects synthèse et composition, en décrivant certains aspects abordés au sein de l'équipe ISA.

3.1.1 Synthèse d'image

La partie synthèse consiste à modéliser les objets à incruster dans la séquence et à illuminer ces objets. Modéliser l'objet 3-D signifie :

- modéliser la géométrie (par un ensemble de facettes polygonales par exemple). Cette modélisation peut être effectuée à partir de plans architecturaux, de données laser, etc.
- modéliser les propriétés des matériaux assignés aux surfaces de la scène (ceci est la plupart du temps basé sur des mesures *in situ* [Cazier et al.94]),
- modéliser la position et l'intensité des sources lumineuses en essayant d'imiter les caractéristiques des sources réelles.

Un algorithme de calcul de radiosité peut être utilisé pour calculer les inter-réflexions diffuses entre les surfaces [Fasse et al.94]. Une spécificité intéressante des méthodes de radiosité est que leur calcul est indépendant du point de vue, si bien que les échanges lumineux n'ont pas à être recalculés pour chaque image de la séquence.

Enfin, lorsque la géométrie de la caméra est connue, le calcul des inter-reflexions spéculaires et le rendu des images de synthèse peut être effectué par un algorithme de lancer de rayons.

3.1.2 Composition

Les différents objets virtuels sont incrustés dans l'image réelle en utilisant des opérateurs de composition (recouvrement, addition, multiplication par un coefficient). Pour plus de réalisme, on peut aussi tenir compte de la mise au point de la caméra en ajoutant un effet de flou plus ou moins marqué aux objets incrustés.

Occultations

Le problème majeur de la composition est de tenir compte des occultations (objet virtuel occulté par un objet réel). Afin d'assurer la cohérence géométrique de la scène, il faut en effet déterminer les masques des objets occultants de l'image réelle, c'est-à-dire la silhouette des objets réels venant occulter un ou plusieurs objets virtuels. Si le modèle et la position des objets réels sont connus, cette opération ne pose pas de problème majeur [Breen et al.95] : un simple

algorithme de type lancer de rayon peut être utilisé. Si le modèle seul est connu, et non la position, il existe également quelques solutions, issues des travaux de Reconnaissance des Formes (localisation d'un objet connu dans une image) [Jancene et al.95]. Malheureusement, dans le cas général, les objets présents dans la scène filmée sont inconnus ou difficilement modélisables (arbres, personnes etc.).

La plupart du temps, la détermination des occultations est effectuée à la main par l'opérateur qui détoure les masques dans les images [Nakamae et al.86, Ertl et al.91, Chevrier et al.95].

La détection automatique des occultations en l'absence de modèle des objets de la scène est un problème délicat. En théorie, une reconstruction de type stéréovision de la scène à partir de deux images dont le point de vue est connu devrait permettre de résoudre le problème. Dans la pratique, cette solution s'avère être très sensible aux erreurs d'appariements, à la précision de localisation des caméras, au bruit présent dans les images et à la distance entre les deux caméras [Wloka et al.95].

Dans notre équipe, Marie-Odile Berger propose un algorithme de détection des occultations ne nécessitant pas de reconstruction 3-D [Berger97] : le principe est d'étiqueter les contours et points caractéristiques de l'image comme étant "devant" ou "derrière" l'objet virtuel. Pour cela, le flux optique de chaque point est comparé avec le flux théorique généré par le point 3-D de l'objet virtuel se projetant en ce point. Le détourage de l'objet occultant peut alors être calculé en utilisant l'algorithme de [Garai et al.99] (voir [Simon et al.99b] pour plus de détails).

Récemment, [Ong et al.98] ont proposé un algorithme basé sur la reconstruction approximative de la surface de l'objet occultant à partir du détourage manuel de son masque d'occultation dans plusieurs images-clé de la séquence. Des points d'intérêt sont détectés à l'intérieur de ces masques, puis appariés entre deux images-clé consécutives. Comme les paramètres de la caméra ont été déterminés sur la séquence (par la méthode de [Tomasi et al.92] décrite plus loin), on peut alors reconstruire ces points, et calculer ainsi une boite englobante de la surface d'occultation. Cette boite est ensuite affinée de façon à être correctement projetée dans les deux images-clé. Les volumes obtenus peuvent alors être projetés dans les images intermédiares pour obtenir les masques d'occultation sur toute la séquence.

Inter-réflexions entre objets réels et virtuels

Une composition réaliste doit aussi prendre en compte les inter-réflexions de la lumière entre objets réels et virtuels (surfaces réfléchissantes et ombres portées). Dans ce cas, un modèle partiel des objets réels concernés par les inter-réflexions doit nécessairement être connu (ou calculé), puis les opérateurs de composition sont utilisés pour combiner l'ombre ou l'image réfléchie (provenant d'un objet réel ou virtuel) avec la surface de l'objet cible (réel ou virtuel) : par exemple, pour l'illumination virtuelle du Pont Neuf, le pont virtuel doit se refléter dans l'eau de la Seine. Un plan horizontal ayant une surface réfléchissante est donc introduit dans la scène virtuelle, puis l'image obtenue sur ce plan est déformée avant d'être mixée avec l'image réelle de l'eau [Chevrier et al.95]. Une reconstruction semi-automatique de la scène réelle peut aussi être utilisée pour calculer les inter-reflexions de la lumière entre objets réels et virtuels [Drettakis et al.97].

3.1.3 Paramètres de la caméra

Les paramètres de la caméra sont déterminés dans chaque image de la séquence par l'une des techniques exposées au chapitre 2. Ces techniques étant basées sur la connaissance d'appariements 3-D/2-D ou 2-D/2-D d'un certains nombre de primitives, la difficulté majeure consiste à maintenir un nombre suffisant d'appariements pertinents dans toute la séquence. Généralement,

deux images consécutives d'une séquence étant proches l'une de l'autre, l'étape d'appariement est facilitée par la possibilité de *suivre* les primitives d'une image à l'autre.

Plusieurs techniques de suivi peuvent être mises en oeuvre suivant le type de primitives prises en compte : nous pouvons utiliser la carte des contours pour suivre des segments de droite [Gennery92], ou encore les gradients d'intensité ou le *flux optique* pour suivre des contours courbes (chapitre 5). La technique de *corrélation* 2-D/2-D, basée sur la carte des intensités, est aussi très fréquemment utilisée pour suivre des points [Tomasi et al.92, Ravela et al.96, Uenohara et al.96] : cette technique consiste à définir une fenêtre dite *fenêtre de corrélation* autour du point à suivre, et à rechercher la fenêtre qui lui ressemble le plus dans la deuxième image. Cette recherche se fait généralement dans un voisinage proche de la position du point dans la première image, en minimisant la différence des intensités entre les pixels de la fenêtre de référence et ceux de la fenêtre supposée.

Cependant, ces méthodes peuvent être sensibles au bruit dans l'image, aux variations d'intensité ou aux changements de point de vue. D'autre part, des primitives peuvent être occultées sur certains passages de la séquence. Lorsque les occultations ne sont pas détectées, ces primitives continuent à être suivies alors qu'elles ne sont plus visibles. Il est donc nécessaire d'être en mesure de détecter les appariements incorrects, ou alors d'utiliser des méthodes d'estimation du mouvement qui soient robustes à ces erreurs.

Pour les systèmes utilisant une technique de calibration basée modèle (ou *systèmes basés modèle*), cette vérification est relativement simple puisque nous pouvons projeter *a posteriori* les primitives 3-D correspondant aux primitives suivies, et vérifier si ces projections sont suffisamment proches des primitives suivies. Pour les systèmes utilisant une technique de calibration basée images (ou *systèmes basés images*), nous ne disposons pas directement d'un tel retour d'information (des invariants projectifs peuvent toutefois être utilisés [Uenohara et al.96]), ce qui rend le problème plus compliqué : aussi, une forte interaction avec l'opérateur est-elle le plus souvent requise pour effectuer ou vérifier les appariements.

3.1.4 Implémentations à travers la littérature

L'un des tous premiers travaux concernant l'incrustation d'une image virtuelle dans une photographie date de 1979 [Uno et al.79]. La photographie est alors numérisée à partir d'un scanner à main, et l'image composée est sortie sur un écran 320 × 240 ou une imprimante à aiguille. Malgré les difficultés techniques rencontrées à cette époque (notamment en ce qui concerne la manipulation des objets virtuels et des images), Uno et Matsuka parviennent à plaquer des textures réelles sur des objets virtuels, et même à incruster un immeuble virtuel dans une image réelle. Tous les renseignements nécessaires à la projection de l'objet virtuel dans l'image sont donnés par l'utilisateur.

Six ans plus tard, Maver et al. proposent un logiciel permettant d'évaluer l'impact d'un bâtiment dans un paysage par photomontage [Maver et al.85]. Là encore, les paramètres de la caméra sont entrés par l'utilisateur.

Un premier essai de composition automatique est présenté dans [Nakamae et al.86]. Les auteurs explorent quelques points clé de la Réalité Augmentée : calcul des ombres projetées par les sources lumineuses réelles sur les objets virtuels, simulation de phénomènes atmosphériques, amélioration de la qualité du montage grâce à une nouvelle technique d'*anti-aliasing*. Les paramètres de la caméra sont déterminés grâce à des appariements 3-D/2-D de points, mais ces points sont appariés à la main, et l'aspect séquence n'est pas du tout évoqué.

Il faut attendre le début des années 90 pour voir apparaître les premiers systèmes manipulant des *séquences* d'images au lieu d'images isolées [Ertl et al.91]. On parle alors de *recalage temporel* :

le point de vue est calculé à partir d'un objet de la scène dont le modèle 3-D est connu, qui est suivi dans la séquence. De nombreux systèmes de RA basés modèle se sont développés depuis.

Les systèmes basés images sont apparus plus tard : le problème du calcul du point de vue à partir de données images a commencé à être étudié au début des années 90, mais l'aspect multi-images n'est apparu qu'à partir de 1996 [Zeller et al.96]. L'application de ces travaux à la RA est quant à elle toute récente [Faugeras98, Ong et al.98, Zisserman et al.99].

Nous explorons à présent dans le détail plusieurs sytèmes de RA basés modèle (3.2) ou basés images (3.3). Une approche hybride et un système ne nécessitant pas de calibration sont aussi décrits en 3.4, et quelques produits commerciaux sont évoqués en 3.5. Une discussion comparative de ces différents systèmes est menée en 3.6, nous permettant de justifier nos propres choix.

3.2 Systèmes basés modèle

Lorsque des primitives 3-D de la scène sont connues, le problème se résout de façon séquentielle et est connu sous le nom de recalage temporel. En général, les paramètres intrinsèques de la caméra sont calculés en étape préliminaire (à l'aide d'une mire ou en utilisant les appariements connus dans la première image), et supposés constants sur la séquence. Le problème se résume donc à calculer le point de vue de la caméra image après image, en utilisant l'une des méthodes 3-D/2-D exposées au chapitre 2 (pour les méthodes itératives, l'estimée initiale peut être obtenue à partir des paramètres obtenus dans l'image précédente).

L'autonomie et la précision du système reposent donc sur le maintien au cours de la séquence d'un nombre suffisant de primitives images conformes aux primitives 3-D du modèle. Dans le contexte basé modèle, les erreurs de suivi peuvent en général être détectées par comparaison avec la reprojection du modèle, à condition que l'algorithme de calcul du point de vue soit *robuste* aux erreurs d'appariement, c'est à dire que les points mal suivis obtiennent réellement un résidu élevé. Le maintien d'un nombre suffisant de primitives au cours de la séquence n'est pas un problème simple : certaines primitives peuvent être mal suivies, ou encore disparaître du champ de vision de la caméra. Un système autonome doit donc être capable de *mettre à jour* l'ensemble des primitives suivies, c'est-à-dire de retrouver les primitives mal suivies dans l'image (par la suite nous emploierons le terme de *ré-initialisation* d'une primitive), et d'intégrer les primitives 3-D entrant dans le champ de vision de la caméra au cours de la séquence (pour compenser la perte des primitives sorties), en recherchant leur homologue 2-D dans l'image (*initialisation* d'une primitive).

La boucle de recalage temporel d'un système basé modèle est résumée en figure 3.2. Nous présentons à présent plusieurs systèmes de RA implémentant plus ou moins partiellement cette boucle, et basés sur des primitives de types différents. Nombre d'entre eux utilisent le signal intensité des images. Des marqueurs artificiels peuvent être placés dans la scène et reconnus facilement par des techniques de type corrélation. Certains systèmes se basent directement sur des points naturels de la scène, dont l'apparence (niveaux de gris autour du point) est stockée dans une table. Le système de Gagalowitcz prend en compte les facettes texturées du modèle qui sont suivies d'une image à l'autre de la séquence. Enfin, plutôt que de se baser directement sur le signal intensité, d'autres algorithmes exploitent les *contours* de l'image, c'est-à-dire des chaînes de points présentant un gradient fort du signal.

3.2.1 Introduction de marqueurs artificiels

La plupart des systèmes d'immersion en temps réel utilisent des marqueurs artificiels placés dans la scène par un opérateur [Ertl et al.91, Mellor95, Peuchot95, State et al.96, Stricker et al.98].

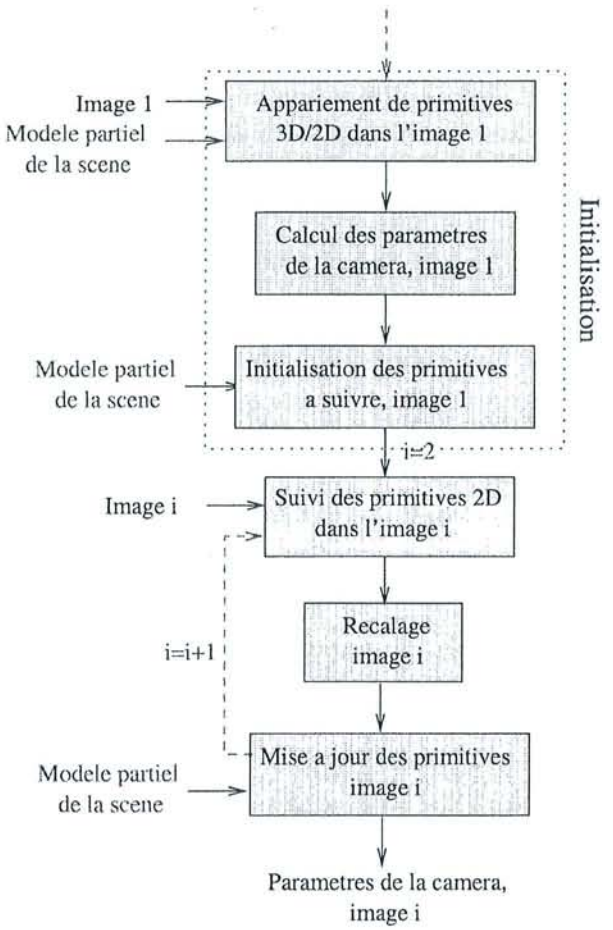


FIG. 3.2 – *La boucle de recalage temporel pour un système de RA basé modèle.*

Ertl et al. ont parmi les premiers présenté un système général appelé *Move-X*, permettant d'incruster des bâtiments virtuels dans une séquence vidéo [Ertl et al.91]. Le recalage est effectué à partir de marqueurs (disques blancs sur fond noir) dont on connaît la position dans la scène. Ces marqueurs sont suivis dans la séquence par une technique de corrélation dans l'espace des couleurs RGB. Le point de vue est estimé par une méthode itérative de type Newton, initialisée à partir du point de vue précédemment calculé dans la séquence. La mise à jour des primitives consiste à rechercher l'aire convexe de couleur blanche la plus proche de la projection de la primitive que l'on cherche à initialiser.

Dans le système de Stricker et al., les marqueurs n'ont plus à être suivis, mais sont identifiés indépendamment de l'image précédemment traitée, grâce à l'utilisation de codes de type codes barres placés sur les marqueurs [Stricker et al.98]. Ce système fonctionne en temps réel, et est adapté à plusieurs applications de RA : maçonnerie (un mur virtuel est incrusté à l'endroit où le mur réel doit être construit), manipulation d'objets virtuels posés sur un support réel et jeu (le tic tac toe présenté au chapitre 1).

3.2.2 Prise en compte de points naturels

D'autres systèmes utilisent directement des points de la scène, identifiés grâce aux niveaux de gris mesurés autour des points.

Par exmple, Uenohara et al. proposent un système de recalage basé sur un suivi de points par corrélation, qui fonctionne en temps réel et intègre toutes les étapes de la boucle présentée en figure 3.2 [Uenohara et al.96]. Au début du processus, le système affiche le modèle filaire de l'objet à recaler dans la première image, et l'opérateur déplace la caméra par l'intermédiaire d'une interface graphique, jusqu'à ce que l'objet virtuel et l'objet réel soient grossièrement alignés (figure 3.3.a). Un certain nombre de points-clé sont alors projetés dans l'image et leur correspondant 2-D est recherché autour de ces projections. Pour cela, des images de référence ont été pré-capturées autour de chaque point-clé, sous diverses conditions d'illumination. Un score de corrélation normalisée est calculé avec toutes ces images de référence pour chaque point situé dans la zone de recherche. Le point obtenant le meilleur score est retenu comme étant le correspondant 2-D recherché, à condition que ce meilleur score soit supérieur à un certain seuil. L'utilisation de plusieurs images de référence permet d'améliorer la robustesse du recalage initial vis-à-vis des changements d'illumination (afin de réduire la complexité des calculs, les images de référence sont approximées par une combinaison linéaire de vecteur propres).

Un certain nombre de points faciles à suivre ont été sélectionnés avant l'exécution. Lorsque la position initiale de la caméra est calculée, ces points sont projetés dans l'image et une fenêtre de corrélation est extraite autour de chaque point comme référence pour le suivi dans l'image suivante (figure 3.3.b). Le suivi est alors effectué par corrélation normalisée. Un invariant géométrique (basé sur cinq points coplanaires) est utilisé pour détecter les éventuels problèmes de suivi. Le point de vue est obtenu itérativement (par l'algorithme de Newton) à partir de la position obtenue dans l'image précédente.

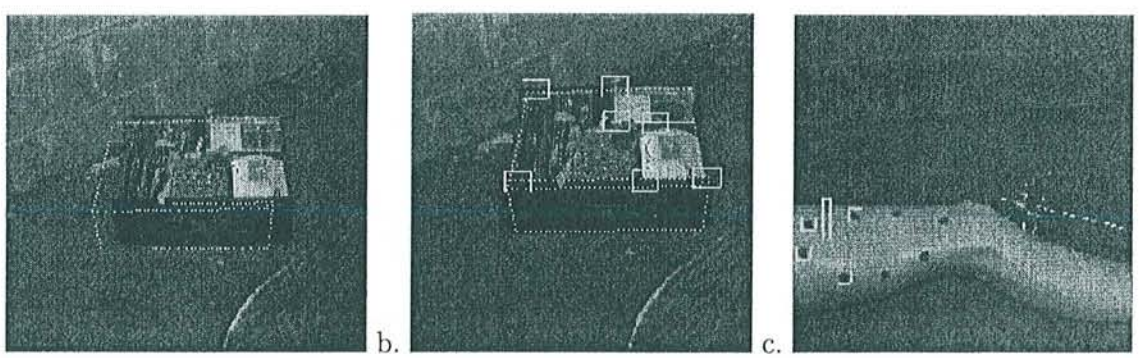


FIG. 3.3 – *Le système de Uenohara et Kanade. a et b: Recalage 3-D/2-D d'un ordinateur : a. initialisation approximative par l'opérateur ; b. fenêtres de corrélation. c. Incrustation d'une épingle virtuelle sur la jambe d'un patient supposée plane. (M. Uenohara et T. Kanade - Vision based object registration for real time image overlay,* Journal of Computers in Biology and Medecine, *1996.)*

L'invariant géométrique peut aussi être utilisé pour effectuer une composition sur un objet plan dont le modèle 3-D n'est pas connu : lorsque cinq points coplanaires sont détectés dans une image, il suffit en effet de suivre quatre de ces points pour déduire la position du cinquième point dans l'image suivante. Cette propriété a été appliquée dans le domaine médical pour maintenir une "épingle" virtuelle (un pointeur) sur un endroit particulier de la jambe d'un patient (assimilée à une surface plane). La pointe de l'épingle est désignée dans une image par le médecin, puis sa position 2-D est calculée par rapport à quatre points d'intérêt de l'image (figure 3.3.c). Lorsque la jambe du patient est en mouvement, les quatre points sont suivis par corrélation et la pointe de l'épingle peut être repositionnée au bon endroit. Un système complet de RA reposant sur un principe similaire (expression de la projection affine d'un point virtuel comme la combinaison

linéaire de points suivis dans la séquence [Kutulakos et al.96]) sera présenté plus loin.

[Ravela et al.96] proposent un système assez semblable à celui de Uenohara et Kanade, appliqué à la maintenance interactive : un technicien regarde l'objet à réparer à travers un viseur sur lequel se superposent des annotations virtuelles. Ces annotations désignent certains éléments de l'objet en fonction des opérations à effectuer (voir la figure 3.4).

La boucle de recalage temporel est initialisée à partir d'un ensemble de correspondances modèle-image de points, des paramètres intrinsèques de la caméra et d'une *table d'aspect* précompilée, qui associe des points de vue discrets avec les primitives visibles depuis ces vues (une primitive est définie par ses coordonnées dans le repère de l'objet et une fenêtre de corrélation traduisant son apparence). L'opérateur spécifie les correspondances initiales, qui sont utilisées pour estimer le point de vue dans la première image. Le système suit alors une boucle à trois étapes :

1. l'information de point de vue est utilisée comme index dans la table d'aspect et une liste des primitives visibles (c'est-à-dire non occultées par l'objet-même) est extraite,
2. les coordonnées 3-D de ces primitives sont projetées dans l'image suivante comme hypothèse de localisation du point image,
3. les fenêtres de corrélation sont localisées dans cette image (par corrélation normalisée avec les fenêtres de corrélation stockées dans la table d'aspect) et un nouveau point de vue est calculé.

Un algorithme itératif robuste est utilisé pour calculer les points de vue à partir de la position dans l'image précédente.

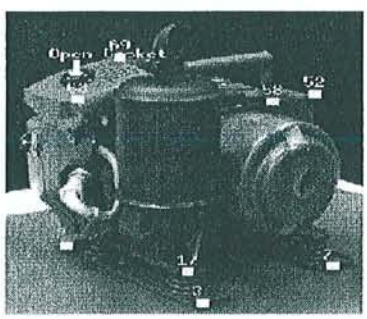


FIG. 3.4 – *Recalage temporel par la méthode de Ravela et al. Les fenêtres suivies par corrélation sont indiquées par des rectangles blanc.*

3.2.3 Utilisation des facettes texturées du modèle

Récemment, Gagalowitcz a présenté un système de recalage temporel basé sur les textures de l'objet 3-D [Gagalowicz99]. Cette méthode requiert la connaissance du modèle complet de l'objet sous forme de facettes. Les paramètres intrinsèques et extrinsèques de la caméra sont calculés dans la première image grâce à des appariements 3-D/2-D de points définis par l'opérateur. Les polygones correspondant aux facettes de l'objet sont alors projetés dans le plan image, et les textures des facettes sont extraites à partir de l'image, à l'intérieur de chaque polygone projeté (*apprentissage des textures*). Le modèle est recalé dans l'image suivante par une minimisation itérative de l'erreur entre les textures projetées et les textures observées (norme de la différence entre les vecteurs RGB pour chaque pixel de chaque facette). Les textures peuvent être réapprises toutes les p images et une approche multi-résolution est utilisée pour réduire les temps de calcul.

Toutefois, le suivi de facettes texturées est sensible aux occultations et à la présence d'ombres portées. L'auteur affirme que l'algorithme est robuste à la présence d'occultations si celles-ci ne couvrent pas une trop grande partie de l'objet, mais pas aux ombres portées. D'autre part, lorsqu'une facette occultée par une partie de l'objet devient visible, celle-ci n'est pas nécessairement prise en compte puisqu'elle est cachée relativement au point de vue de l'image précédente. Cette nouvelle facette ne peut donc pas être apprise. Les solutions proposées par l'auteur sont l'utilisation d'un modèle de mouvement, qui restreint donc les applications possibles [Lowe92, Ravela et al.96, Stricker et al.98], ou le redémarrage du processus dans l'ordre inverse de la séquence, à partir d'une image postérieure à l'apparition de la facette, ce qui rend l'algorithme non linéaire.

3.2.4 Systèmes basés sur les contours de l'image

Plusieurs algorithmes sont basés sur les contours de l'image, qui sont appariés avec des primitives du modèle. La plupart de ces algorithmes utilisent cependant des détecteurs de segments [Lowe90, Gennery92, Harris92, Koller et al.92].

Par exemple, [Gennery92] présente une méthode de recalage d'objets 3-D, dans un contexte d'assemblage de structures en orbite ou de réparation de satellites. Ce système est basé sur les segments extraits de la carte des contours. Un filtre équivalent à celui de Kalman [Jazwinsky70] permet de prédire la position des segments de l'objet 3-D dans une nouvelle image i, à partir de sa position dans l'image $i-1$: pour cela, Gennery considère un modèle de mouvement de caméra à accélération aléatoire (bruit blanc). Comme l'accélération est aléatoire, sa valeur attendue entre les images $i-1$ et i est nulle : la vitesse prédite est donc celle obtenue entre $i-2$ et $i-1$. La position prédite de la caméra en i peut alors être obtenue à partir de sa position en $i-1$ et de la vitesse prédite, puis les segments du modèle 3-D sont projetés en utilisant cette prédiction. Le segment le plus proche de la projection est choisi comme correspondant 2-D (la prise en compte de la direction du segment comme critère plus fin de sélection est aussi envisagée) et un poids lui est attribué en fonction de la précision supposée de sa détection et de sa potentialité à être détecté comme contour fort de l'image (spécifiée dans le modèle de l'objet). Le point de vue ainsi que la vitesse de la caméra sont alors ajustés par des moindres carrés pondérés.

Des résultats précis ont été obtenus sur un prisme hexagonal, tournant autour d'un axe à vitesse constante. Cependant, l'utilisation d'un filtre de Kalman suppose que le mouvement de la caméra vérifie l'hypothèse du filtre. Ceci est approprié dans un cadre très contraint, où il est possible d'imposer des limites précises sur les changements d'accélaration attendus. Par ailleurs, cette méthode n'est envisageable que si nous savons estimer l'incertitude obtenue sur les primitives.

3.3 Systèmes basés images

On trouve très peu de systèmes de RA basés images dans la littérature. En fait, la RA est habituellement présentée comme une application possible des algorithmes plus généraux de structure à partir du mouvement (*structure from motion* - SFM) que nous allons décrire ici. Ces algorithmes permettent de retrouver la *structure* de la scène (plus exactement un ensemble de primitives 3-D reconstruites) en même temps que le *mouvement* de la caméra, à partir uniquement du flot d'images de la séquence et sans aucune connaissance *a priori*, ni sur la scène, ni la plupart du temps sur les paramètres intrinsèques de la caméra.

Il n'est pas toujours possible d'évaluer très précisément la pertinence de ces méthodes pour les applications de RA. En effet, les algorithmes de SFM sont généralement validés par des résultats

de reconstruction tri-dimentionnelle de la scène accompagnés quelquefois de tests comparatifs entre la scène reconstruite et la scène filmée (mesures de distances après avoir résolu le facteur d'échelle [Tomasi et al.92] ou mesures d'angles [Bougnoux97]). On ne trouve que très rarement le tracé de la trajectoire retrouvée de la caméra ou encore une séquence vidéo permettant d'évaluer la qualité de la composition.

Pourtant, reconstruire une scène 3-D en retrouvant les paramètres de la caméra n'est pas exactement équivalent à retrouver une trajectoire correcte de la caméra pour l'incrustation d'objets virtuels dans la séquence. D'autre part, le modèle reconstruit est défini à un facteur d'échelle près, et généralement dans le repère de la première caméra. Pour les applications de RA, il faut nécessairement déterminer le facteur d'échelle, et faire coïncider le repère dans lequel est exprimé l'objet virtuel avec le repère de la scène reconstruite. Ceci ne peut se faire qu'à partir d'une connaissance partielle de la scène (mesure d'une distance pour résoudre le facteur d'échelle) ou par un positionnement par tâtonnement de l'objet virtuel dans la scène reconstruite (comme cela se fait avec les produits commerciaux), ce qui peut s'avérer plus ou moins précis et parfois très laborieux.

Deux aspects sont à prendre en compte lorsqu'on cherche à retrouver la structure de la scène et le mouvement de la caméra à partir du flot d'images. Le premier est l'appariement des primitives 2D-2D dans la séquence, et le deuxième la distribution de l'erreur sur toutes les images de la séquence.

Appariement des primitives

Une approche séquentielle est généralement adoptée pour apparier les primitives entre les images : des primitives sont détectées dans la première image, puis suivies d'une image à l'autre, la plupart du temps par corrélation. Cette approche n'est cependant pas très fiable : la primitive peut changer d'aspect par variation du point de vue ou de l'illumination, et par ailleurs les primitives suivies sont rarement visibles dans toutes les images. Une interaction importante avec l'opérateur est donc souvent nécessaire pour vérifier et réajuster le suivi (une session utilisateur classique est présentée plus loin). Une approche non séquentielle peut aussi être mise en oeuvre, qui consiste à détecter des points d'intérêt dans deux images (pas nécessairement consécutives) puis à rechercher les appariements les plus plausibles entre ces deux ensembles de points selon un *score de corrélation* [Luong92]. Cette approche peut cependant générer de faux appariements, et une approche robuste est alors nécessaire [Zhang et al.95].

Un autre problème causé par l'appariement automatique des points est que leur distribution spatiale est laissée au hasard. Bougnoux propose donc un outil semi-automatique, distribué sous le nom de *TotalCalib*, permettant de faciliter la tâche d'appariement [Bougnoux97]. L'auteur part du constat qu'un faible nombre de points bien distribués et détectés précisément permettent d'obtenir une calibration bien plus stable que de nombreux points détectés automatiquement et éventuellement mal distribués et peu fiables. Le problème est que pour définir des points à une précision sub-pixel, l'opérateur doit fréquemment effectuer des zooms avant et arrière dans l'image, ce qui devient très vite laborieux. Le principe de cet outil est d'affiner la géométrie épipolaire au fur et à mesure que des points sont saisis par l'opérateur : lorsqu'un point est sélectionné dans la première image, le programme peut alors utiliser la contrainte épipolaire pour limiter la zone de recherche du point homologue et contraindre le zoom (un appariement automatique est aussi proposé).

Distribution de l'erreur

Le deuxième aspect à considérer est la distribution de l'erreur obtenue sur le mouvement et la structure, sur toutes les images de la séquence.

Pour cela, des méthodes de factorisation ont été proposées, qui consistent à représenter le flot d'images par une matrice $2m \times n$ (ou $3m \times n$ en coordonnées projectives) mesurant les coordonnées images de n points suivis dans m images. Cette matrice est factorisée par une technique de décomposition en valeurs singulières (SVD) en deux matrices représentant respectivement le mouvement de la caméra et la structure de la scène. La méthode de factorisation de Tomasi et Kanade [Tomasi et al.92] minimise l'erreur de reprojection moyenne sur toute la séquence, mais considère un modèle orthographique de caméra. D'autres méthodes de factorisation ont été proposées pour le modèle à sténopé de caméra [Sparr96, Heyden et al.97, Sturm et al.96], mais ces méthodes minimisent une erreur algébrique au lieu de l'erreur de reprojection. De plus, les méthodes de factorisation sont limitées aux primitives apparaissant dans toutes les images de la séquence.

La technique d'ajustement de faisceau (traduction de *bundle adjustment*) est aussi couramment employée [Zeller et al.96, Fitzgibbon et al.98]. Cette technique consiste à faire varier les paramètres de la structure et des caméras pour minimiser l'erreur de reprojection. Elle n'est pas pénalisée par les correspondances manquantes, mais il n'existe pas de solution directe (comme la solution SVD pour les méthodes de factorisation) : une optimisation itérative doit être mise en oeuvre, ce qui peut conduire à un minimum local dans l'espace des paramètres si l'estimée initiale n'est pas suffisamment bonne (le nombre de paramètres à estimer est $km + n$, où k est le nombre de paramètres non connus de la caméra).

Une alternative à l'ajustement de faisceau est l'utilisation d'un filtre récursif (de type Kalman) dans un processus séquentiel [Beardsley et al.94, Mclauchlan et al.94]. Malheureusement, si l'initialisation à partir de deux vues ou trois vues est mauvaise, cela affecte considérablement la précision des résultats obtenus dans les images suivantes.

Nous présentons trois algorithmes de SFM, qui ont été appliqués à la RA. Le premier (Faugeras et al.) va nous permettre d'illustrer le passage de l'auto-calibration à partir de deux images (présentée au chapitre 2) au cas de n images ("les équations de Kruppa revisitées"). Le deuxième est l'algorithme de factorisation de Tomasi et Kanade, qui permet d'obtenir des résultats satisfaisants dans les limites du modèle orthographique. Enfin, la méthode de Zisserman et al. permet d'obtenir une bonne initialisation pour l'ajustement de faisceau, grâce à l'intégration des résultats les plus récents de statistique robuste et de géométrie projective à partir de trois vues.

3.3.1 Les équations de Kruppa "revisitées"

Zeller et Faugeras [Zeller et al.96] ont proposé la généralisation à un grand nombre d'images de l'algorithme développé dans la thèse de Luong [Luong92] et présenté au chapitre 2. Si nous disposons d'un nombre d'images n supérieur à trois, nous pouvons résoudre les équations de Kruppa en se ramenant à un problème aux moindres carrés non linéaire. La méthode classique de Levenberg-Marquardt peut être utilisée pour la minimisation [Press et al.88].

Le mouvement est ensuite calculé entre chaque image. Cependant, comme seule la direction de la translation peut être calculée, il s'ensuit une incohérence sur les facteurs d'échelle obtenus si l'on se contente de calculer le mouvement entre deux images consécutives. Ce problème est illustré en figure 3.5 : nous supposons que le centre optique de la caméra se déplace suivant les translations $\Delta\mathbf{t}_{12}$,$\Delta\mathbf{t}_{23}$,$\Delta\mathbf{t}_{34}$,$\Delta\mathbf{t}_{45}$. Nous savons que la translation calculée $\Delta\mathbf{t}_{1'2'}$ n'est définie qu'à un facteur d'échelle près. Si nous nous basons uniquement sur les images 2 et 3 pour calculer

la translation $\Delta\mathbf{t}_{2'3''}$ entre ces deux images, nous obtenons un facteur d'échelle différent de celui obtenu pour $\Delta\mathbf{t}_{1'2'}$, et la direction de la translation $\Delta\mathbf{t}_{1'3''}$ n'est pas la même que celle de la translation $\Delta\mathbf{t}_{13}$. En utilisant ce procédé, nous calculons donc une trajectoire 1',2',3'',4'',5'' qui n'est pas identique à la trajectoire 1,2,3,4,5 à un facteur d'échelle près. Pour obtenir un facteur d'échelle cohérent, il faut en plus calculer la translation $\Delta\mathbf{t}_{1'3'''}$ à partir des images 1 et 3 : la position 3' recherchée est alors l'intersection des deux droites portées par $\Delta\mathbf{t}_{1'3'''}$ et $\Delta\mathbf{t}_{2'3''}$. Ainsi, pour déterminer la position de la caméra dans l'image i, les directions des translations $\Delta\mathbf{t}_{1i}$ et $\Delta\mathbf{t}_{2i}$ sont calculées. Les $m-1$ matrices de projection obtenues par cette méthode sont ensuite affinés par un processus itératif à $5+6(n-1)$ variables (Levenberg-Marquardt).

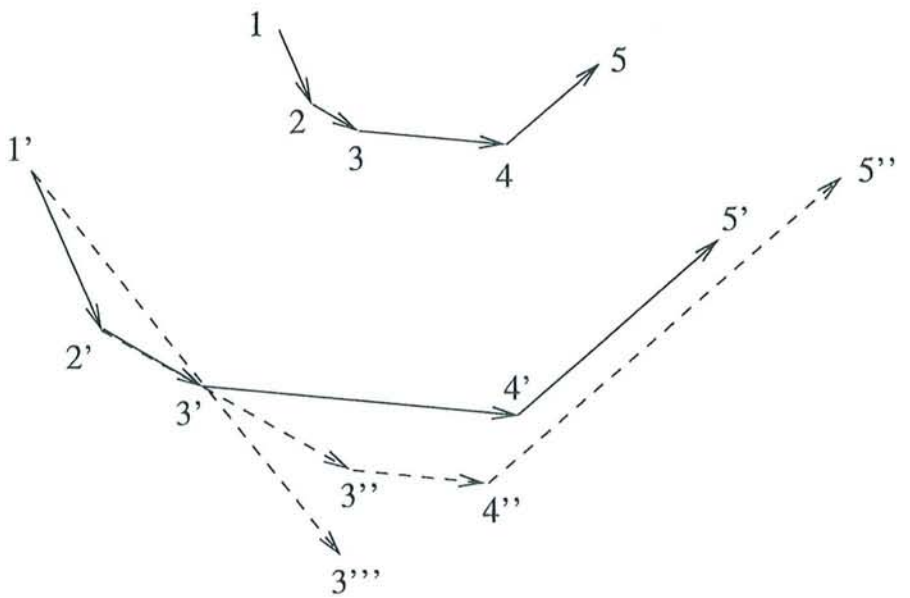


FIG. 3.5 – *Le problème du facteur d'échelle sur n images.*

Un exemple d'application de cette méthode à la RA est proposé dans [Faugeras98], où une lampe virtuelle est posée sur un bureau réel, la scène étant statique et la caméra en mouvement (un autre exemple est proposé pour une caméra statique et la scène en mouvement). Dans un premier temps, des points caractéristiques sont suivis tout au long de la séquence par une technique de corrélation. Un certain nombre de vues de la séquence correspondant à des positions de caméras suffisamment différentes sont alors choisies manuellement. La méthode que nous venons de décrire est utilisée pour calculer les paramètres intrinsèques (supposés constants dans la séquence) et extrinsèques de la caméra aux instants choisis. Les paramètres extrinsèque sont ensuite obtenus sur toute la séquence par simple interpolation à partir des instants choisis. La partie du bureau sur laquelle la lampe doit être incrustée est alors modélisée par stéréoscopie, puis l'objet virtuel est posé sur cette partie de bureau dans un système CAO (une source de lumière virtuelle est aussi positionnée, et les ombres portées sont calculées). Le modèle obtenu est enfin réinséré dans la séquence initiale.

3.3.2 Factorisation pour le modèle orthographique

Tomasi et Kanade proposent une méthode de factorisation permettant de retrouver la structure de la scène et le mouvement de la caméra sous l'hypothèse d'un modèle de projection *orthographique* (chaque point 3-D se projette dans le plan image parallèlement à l'axe optique)

[Tomasi et al.92].

Une séquence d'images peut être représentée par une matrice $2m \times n$, où n est le nombre de points suivis dans les m images de la séquence. Les auteurs montrent que sous l'hypothèse d'une projection orthographique, cette matrice est de rang 3. Basée sur cette observation, la méthode utilise la technique SVD pour factoriser la matrice en deux matrices représentant respectivement la structure de la scène (coordonnées 3-D des points suivis) et la rotation de la caméra. La méthode peut aussi tolérer quelques "trous" dans la matrice de la séquence, obtenus lorsque des primitives suivies sont occultées.

Ce système possède toutefois un certain nombre de limites : tout d'abord, le modèle de projection orthographique est bien adapté pour des objets distants de la caméra par rapport à leur taille, mais des effets de perspective significatifs peuvent affecter les résultats pour des objets plus proches de la caméra. En outre, ce modèle ne permet pas de prendre en compte les déplacements de la caméra vers la scène (dans la direction de l'axe optique). Deuxièmement, la méthode est optimale lorsque tous les points sont visibles dans la séquence : cela signifie d'une part que la caméra est contrainte à filmer une zone limitée de l'espace, et d'autre part que les points doivent être correctement suivis dans toute la séquence.

Cette méthode a été appliquée à la RA par [Ong et al.98], pour visualiser de nouvelles architectures dans leur environnement, en présence d'occultations. Cependant, l'accent est mis sur la détection des occultations, et le calcul du point de vue est peu discuté. Comme la méthode de Tomasi et Kanade ne détermine que la rotation de la caméra et deux composantes de la translation, les paramètres de la translation ainsi que la distance focale supposée constante ($3m+1$ inconnus) sont estimés par minimisation itérative de l'erreur de reprojection des points 3-D reconstruits (algorithme de Levenberg-Marquardt).

Les contraintes de la méthode de Tomasi et Kanade sont respectées : l'objet est incrusté dans une zone éloignée de la caméra, et la caméra filme à peu près la même zone de la scène tout au long de la séquence. Cependant, les quelques images extraites ne nous permettent pas d'évaluer la qualité du point de vue sur toute la séquence (stabilité, précision ...).

3.3.3 Vers une meilleure initialisation pour l'ajustement de faisceau

Le problème majeur de la méthode d'ajustement de faisceau est qu'elle nécessite une bonne initialisation pour pouvoir converger [Oliensis97]. Dans ce but, Fitzgibbon et Zisserman [Fitzgibbon et al.98] ont proposé un système de SFM intégrant les résultats les plus récents de géométrie projective à partir de plusieurs vues (tenseur trifocal) et de statistique robuste [Zhang et al.95, Torr et al.97]. Les améliorations par rapport au système de Tomasi et Kanade sont multiples : premièrement, les auteurs utilisent le modèle général de projection perspective au lieu du modèle orthographique. Deuxièmement, l'appariement des primitives ne se fait plus par le suivi des points dans la séquence, mais par des méthodes robustes utilisant des contraintes géométriques. Enfin, les points n'ont plus à être visibles dans toutes les images de la séquence. Cette méthode comporte les étapes suivantes :

1. **Appariement de points entre deux vues.** Des correspondances de points sont établies dans toutes les paires d'images consécutives de la séquence : le détecteur de [Harris et al.88] est utilisé pour extraire des points d'intérêt dans chaque image, et des correspondances supposées entre les paires d'images sont générées par corrélation croisée dans une fenêtre de taille réduite : les points ne sont donc pas suivis entre deux images, mais ce sont des *scores* de corrélation calculés entre deux points détectés indépendamment dans les deux images qui permettent d'établir les correspondances. Les appariements sont alors établis à

partir de cet ensemble de correspondances supposées, par un algorithme robuste estimant simultanément la géométrie épipolaire et les appariements compatibles avec cette géométrie [Zhang et al.95].

2. **Appariement de points entre trois vues.** Des correspondances sont ensuite établies entre les triplets d'images consécutives, à partir des appariements de l'étape 1. Une partie des appariements obtenus pour deux vues peuvent être erronés. La plupart de ces appariements sont alors éliminés par l'estimation robuste simultanée du tenseur trifocal et des appariements compatibles. Comme nous l'avons vu au chapitre 2, la géométrie trifocale permet de lever les ambiguïtés plus facilement que la géométrie épipolaire puisque, étant donné un point apparié entre deux vues, sa position image est complètement déterminée dans une troisième vue (alors qu'elle n'est que restreinte à une droite pour la géométrie épipolaire). À l'issue de cette étape, on dispose d'un ensemble de triplets d'images se recouvrant, chaque triplet ayant un tenseur trifocal associé et des points appariés sur les trois vues. Les matrices projectives sont générées à partir du tenseur trifocal, et les points 3-D sont reconstruits en minimisant l'erreur de reprojection sur les trois vues.
3. **Appariements de droites entre trois vues.** L'appariement de droites entre deux images est un problème délicat car il n'y a pas de contrainte géométrique équivalente à celle que l'on obtient pour des correspondances de points à partir de la matrice fondamentale. Avec trois vues, une contrainte géométrique est obtenue pour les droites à partir du tenseur trifocal. Des segments de droite sont appariés sur un triplet d'images en deux étapes. Premièrement, étant donné le tenseur trifocal et les correspondances de droites supposées entre deux images, la droite correspondante dans la troisième image est calculée. Un segment de droite devrait donc être détecté à l'emplacement prédit dans la troisième image. Deuxièmement, cet appariement est vérifié par un test photométrique basé sur une corrélation d'intensités au voisinage de la droite.
4. **Appariements sur la séquence.** Les correspondances sont étendues sur plusieurs images en fusionnant les appariements des triplets se recouvrant. Par exemple, une correspondance établie pour le triplet 1-2-3 et aussi pour le triplet 2-3-4 peut être étendu aux images 1-2-3-4. Les matrices projectives et la structure 3-D sont alors calculées pour les images 1-2-3-4. Ce procédé est étendu en fusionnant les groupes d'images voisines jusqu'à ce que les matrices projectives et les correspondances soient établies sur la séquence. L'estimation initiale des points 3-D et des matrices de projection est alors affinée par la technique d'ajustement de faisceau, qui consiste à minimiser itérativement l'erreur de reprojection des points reconstruits dans toutes les images où ils apparaissent. A ce stade, on ne dispose que d'une reconstruction *projective* de la scène, c'est-à-dire définie à une transformation homographique près : le passage à la reconstruction euclidienne (à un facteur d'échelle près) se fait par auto-calibration.

Un exemple d'application de ce système à la RA est présenté dans [Zisserman et al.99] : des panneaux sont incrustés sur des immeubles filmés à partir d'un hélicoptère.

Les limites de ce système sont les suivantes : premièrement, les images doivent être suffisamment "intéressantes" : si la scène ne possède pas de textures significatives, trop peu de primitives seront appariées. Deuxièmement, le mouvement de la caméra entre deux images consécutives doit être relativement faible (et en particulier les rotations autour de l'axe optique doivent être limitées) sinon la procédure de corrélation utilisée pour apparier les primitives peut échouer. Enfin, l'étape d'ajustement de faisceau rend la méthode non séquentielle.

3.4 Autres approches

3.4.1 Un système hybride

Devebec et al. proposent une approche hybride permettant de modéliser un édifice architectural à partir de quelques photographies (ceci peut ensuite être appliqué à la RA) [Debevec et al.96]. L'opérateur est d'abord sollicité pour construire un modèle grossier de l'objet à modéliser : ce modèle est en fait un ensemble de primitives géométriques paramétrées (appelées *blocs*) telles que des boîtes, des prismes ou des surfaces de révolution. Les informations de type distance, rapports de distance ou angles sont laissées en paramètres puisque la scène est supposée non connue. Par contre, les différents blocs sont organisés en arbre hiérarchique : chaque noeud de l'arbre représente un bloc individuel, tandis que les liens contiennent des relations spatiales entre ces blocs (alignements, rotations ...). Les arêtes correspondant aux différents blocs doivent ensuite être désignées dans l'image par l'opérateur. Un algorithme itératif est alors utilisé pour minimiser l'erreur de reprojection du modèle en fonction à la fois des paramètres du mouvement et des paramètres du modèle.

Un algorithme de plaquage des textures extraites d'une ou plusieurs photographies est aussi proposé. Si plusieurs photographies de la scène sont disponibles, l'algorithme interpole les textures obtenues dans ces différentes photographie en fonction du point de vue. Enfin, un algorithme "stéréo basé modèle" est utilisé pour affiner automatiquement le modèle basique obtenu par la méthode précédente à partir d'une paire d'images (la structure du modèle basique est utilisée pour limiter la zone de recherche des points à apparier). Cette technique peut être utilisée pour déterminer la structure d'ornementations architecturales qui auraient été difficiles à modéliser par l'opérateur.

La méthode proposée permet par extension d'incruster des objets dans une photographie de la scène, mais peut difficilement être appliquée à une séquence complète. Par contre, on peut envisager d'utiliser une technique de recalage temporel 3-D/2-D pour suivre dans une séquence le modèle reconstruit dans la première image. Ce travail est donc aussi particulièrement intéressant, dans la mesure où il permet d'obtenir un modèle partiel de la scène relativement précis, qui peut éventuellement être utilisé pour les systèmes de RA basés modèle lorsque le modèle de la scène n'est pas connu *a priori*.

3.4.2 Un système sans calibration

Kutulakos et Vallino présentent un système de RA *affine* ne nécessitant aucune connaissance métrique, ni sur la scène, ni sur les paramètres de la caméra [Kutulakos et al.96]. Ce système est basé sur l'observation suivante : étant donné un ensemble de plus de quatre points 3-D non coplanaires, la projection des points de cet ensemble peut être calculée par combinaison linéaire de quatre de ces points seulement [Ullman et al.91]. Les objets virtuels sont donc représentés dans un repère affine, ce qui permet de retrouver leur projection par combinaison linéaire de la projection de quatre points formant la base du système de coordonnées affine. Les quatre points de la base affine sont suivis (en temps réel) en tant que marqueurs de couleur spécifique ou sommets de régions polygonales, et l'objet virtuel est repositionné par rapport à ces points. Le système proposé inclut un algorithme de $z-$ *buffer* permettant de comparer dans l'espace affine la profondeur de deux points se projetant sur le même pixel.

La difficulté majeure consiste à positionner l'objet virtuel dans l'espace affine, c'est-à-dire à déterminer les coordonnées de quatre de ses points (non coplanaires) dans la base du repère affine. Ceci est réalisé de façon manuelle par l'utilisateur : celui-ci doit d'abord sélectionner quatre points dans une paire d'images stéréo pour définir la base affine. Il doit ensuite choisir quatre

points non coplanaires de l'objet virtuel et désigner leur projection (supposée) dans l'image 1. Les droites épipolaires de ces points sont alors calculées et tracées dans l'image 2, ce qui aide l'opérateur à désigner la projection des points dans l'image 2. Des solutions sont aussi proposées pour définir la trajectoire d'objets en mouvement, mais cela se fait au prix d'une forte interaction avec l'opérateur.

Le positionnement de l'objet virtuel est donc peu précis puisque (à moins que l'objet virtuel soit réellement présent dans l'image), l'opérateur doit évaluer la position de sa projection dans l'image de gauche et dans celle de droite, aidé de la géométrie épipolaire. D'autre part, la projection affine n'est qu'une approximation de la projection perspective, qui est d'autant moins précise que l'objet projeté est proche de la caméra (jusqu'à 15 pixels d'erreur de reprojection dans les exemples proposés).

3.5 Quelques produits commerciaux

Quelques produits de post-production sont aussi disponibles dans le commerce. Ces logiciels sont brièvement décrits, mais il nous a semblé intéressant de détailler une session utilisateur pour l'un d'entre eux (3D-Equalizer®), ceci afin de rendre compte du niveau d'interaction requis pour obtenir un résultat satisfaisant lorsqu'aucune information n'est disponible sur la scène ou la caméra.

3.5.1 RENOIR®

RENOIR® est un produit de la société INTEGRA, qui permet de retrouver le modèle d'un objet 3-D à partir de sa photographie, ou de composer des images réelles et virtuelles. Ce logiciel implémente les travaux de Debevec présentés précédemment. Un modeleur permet de construire les différents blocs paramétrés de l'objet et d'établir les relations entre ces objets. Les arêtes de ces blocs sont alors désignées dans l'image par l'utilisateur, et les paramètres de la caméra en même temps que les paramètres du modèle sont calculés. Les textures peuvent alors être plaquées sur le modèle obtenu.

Le modèle paramétrique d'un objet de la scène peut aussi être utilisé pour retrouver les paramètres de la caméra et permettre d'ajouter des objets virtuels dans une photographie. Pour la composition, l'éclairage de la scène réelle peut être pris en compte en désignant directement le centre du soleil s'il est visible dans l'image, ou en utilisant la direction des ombres des objets de la scène.

Ce logiciel est donc très pratique pour augmenter une image ou extraire un modèle à partir d'une photographie, mais il n'est pas adapté au traitement de séquences.

3.5.2 3D-Equalizer®

3D-Equalizer® (Science.D.Visions) permet de retrouver la géométrie de la caméra et la structure de la scène (ensemble de points 3-D définis à un facteur d'échelle près) à partir d'une séquence vidéo (ces données peuvent ensuite être exportées pour être utilisées par des logiciels de composition 3-D/2-D). La première version du logiciel requérait la connaissance des coordonnées 3-D de points de la scène, mais la deuxième version permet aussi de n'utiliser que des données images. Quatre méthodes différentes sont ainsi proposées, en fonction du mouvement supposé de la caméra et de l'information disponible :

- une méthode "sans contrainte de distances" qui utilise uniquement des points images. Ces points sont désignés par l'utilisateur dans la première image, puis suivis dans la séquence ;

- une méthode "à position de caméra fixe", adaptée aux rotations pures. Bien sûr, seules les directions 3-D des points suivis par rapport à la caméra peuvent être déterminées ;
- une méthode "à positions définies par l'utilisateur", qui requiert la connaissance des coordonnées d'un certain nombre de points dans le repère de la scène ;
- une méthode "à distances contraintes" basée sur la connaissance des distances entre des points de la scène.

Voici comment se déroule une session utilisateur, pour la méthode sans contrainte de distances :

1. **Initialisation.** L'utilisateur désigne des points dans plusieurs images de la séquence (appelées images-clé) et, le cas échéant, impose des contraintes sur ces points. Pour chaque point, l'utilisateur indique s'il s'agit d'un marqueur (surface de couleur uniforme) ou non. La méthode utilisée pour le suivi des points est la même dans les deux cas (par corrélation), mais dans le cas d'un marqueur, le barycentre de la surface peut être recalculé dans chaque image, ce qui rend la position du point 2-D plus précise.
2. **Suivi des points.** Ces points sont suivis dans la séquence, image après image. Pour cela, l'utilisateur doit désigner autour de chaque point, deux rectangles de tailles différentes : le premier détermine la fenêtre de corrélation et le deuxième (plus grand) la zone de recherche du point homologue dans l'image suivante. La trajectoire 2-D de chaque point suivi peut ensuite être affichée. Comme des erreurs de précision peuvent s'accumuler au cours du suivi, cette trajectoire peut comporter des sauts au niveau des images-clé : l'utilisateur peut alors demander un suivi inverse du point concerné, à partir de l'image où se produit le saut jusqu'à l'image clé précédente. Les trajectoires avant et arrière sont alors fusionnées pour obtenir une courbe lisse.
3. **Calcul de la structure de la scène et des paramètres de la caméra.** Les points suivis sont reconstruits et les paramètres intrinsèques (supposés constants) et extrinsèques de la caméra sont calculés sur toute la séquence (les paramètres intrinsèques qui sont connus peuvent être fixés).
4. **Évaluation du résultat.** La qualité de la reconstruction est évaluée visuellement (position des points reconstruits) et numériquement (erreur de reprojection moyenne des points reconstruits). Si les points reconstruits sont incorrects alors que le suivi des points images est correct, l'utilisateur peut ajuster les paramètres intrinsèques de la caméra en donnant des valeurs plausibles à certains d'entre eux. Il relance alors les calculs, visualise le résultat et éventuellement répète le cycle jusqu'à ce qu'il obtienne un résultat qu'il juge satisfaisant. Un objet virtuel simple peut ensuite être positionné dans la scène et une vidéo montrant sa projection dans chaque image de la séquence est visualisée. Ceci permet d'évaluer grossièrement la qualité des points de vue obtenus. En plus de cette vidéo, la qualité de la perspective est évaluée numériquement pour chaque image. Si cela s'avère nécessaire, des points peuvent alors être corrigés ou de nouveaux points peuvent être désignés et suivis.
5. **Lissage.** Pour terminer, la trajectoire de la caméra peut être lissée dans le domaine fréquentiel. Un filtre de Fourier est utilisé à l'ordre n, où n est donné par l'utilisateur : plus n est petit, plus le lissage est fort (mais plus la courbe obtenue s'éloigne de la courbe initiale). L'utilisateur peut alors comparer la trajectoire lissée avec la trajectoire initiale, et diminuer ou augmenter la valeur de n en essayant de trouver le bon compromis entre le niveau de lissage et l'adéquation à la courbe initiale.

L'utilisateur intervient donc à chaque niveau de la chaîne : désignation des points à suivre dans les images-clé, contrôle et correction du suivi, évaluation visuelle de la reconstruction et

des points de vue, ajustement éventuel des paramètres intrinsèques de la caméra et lissage de la trajectoire.

Deux autres logiciels permettent d'obtenir les paramètres de la caméra sur toute une séquence : il s'agit de Maya Live® (Silicon Graphics Inc.), qui est un *plug-in* du logiciel de modélisation 3-D Maya®, et MatchMover® de la société REALVIZ (transfert technologique de l'INRIA). Le principe de ces deux produits est à peu près identique à celui de la méthode sans contraintes de 3D-Equalizer® : des points sont désignés par l'utilisateur, puis suivis dans la séquence (des contraintes de distances, coplanarités ou coordonnées 3D peuvent être ajoutées). Les paramètres de la caméra sont alors calculés en même temps que la structure de la scène, et un outil intégré permet de réaliser la composition. Concernant ces deux produits, nous ne disposons toutefois que de descriptifs techniques et commerciaux plus ou moins détaillé, qui ne nous donnent qu'un aperçu général des produits.

3.6 Discussion et stratégie adoptée

Nous avons présenté plusieurs systèmes de RA reposant sur des méthodes différentes. Nous résumons à présent les lacunes et les atouts de ces systèmes, et présentons nos propres contributions au problème du recalage.

Rappelons tout d'abord les objectifs : nous souhaitons obtenir un système peu contraignant, autonome et séquentiel (permettant d'envisager son utilisation dans un contexte temps réel), et qui soit aussi stable, robuste, relativement précis et général.

Systèmes basés images

Les systèmes reposant uniquement sur les données images ne permettent pas d'atteindre ces objectifs pour les raisons suivantes :

- ces systèmes ne sont généralement stables qu'au prix d'une forte interaction avec l'utilisateur (supervision du suivi des primitives), ce qui les rend non autonomes, à part pour certains systèmes comme celui de Fitzgibbon et Zisserman qui utilisent explicitement des algorithmes robustes,
- les algorithmes les plus stables sont non-séquentiels afin de pouvoir distribuer l'erreur sur l'ensemble des vues (ajustement de faisceau),
- la précision peut être insuffisante pour les systèmes auto-calibrés en raison du nombre important de paramètres à retrouver [Bougnoux97, Oliensis97],
- les équations de la géométrie projective dégénèrent pour certains mouvements de caméra (notamment lorsque la composante translationnelle est petite). Les vues utilisées doivent être suffisamment espacées pour permettre une reconstruction stable, mais en même temps assez proches pour pouvoir établir suffisamment d'appariements.

D'autre part, une connaissance minimale de la scène réelle est requise pour résoudre le facteur d'échelle et positionner l'objet virtuel dans la scène reconstruite, à moins de réaliser cette tâche par tâtonnement, ce qui ne permet pas d'obtenir des résultats très précis. Nous pensons que si une telle information est disponible, autant l'utiliser pour effectuer un recalage 3-D/2-D, éventuellement affiné par la connaissance 2-D/2-D comme nous le proposons dans cette thèse. La connaissance d'un modèle partiel de la scène n'est d'ailleurs pas une contrainte insurmontable : nous proposons des algorithmes nécessitant une connaissance *minimale* de la scène. Pour les applications architecturales, des plans à l'échelle sont la plupart du temps disponibles. Des mesures *in situ* de quelques primitives (trois ou quatre segments de droites peuvent suffire) sont aussi

la plupart du temps envisageables. L'utilisation de méthodes robustes permet en outre d'exploiter des modèles mesurés à la main, pouvant souffrir de quelques imprécisions. Par ailleurs, si réellement aucune connaissance *a priori* n'est disponible sur la scène, nous pouvons envisager de retrouver un modèle partiel de cette scène en utilisant par exemple la technique de Debevec et al. (logiciel RENOIR®), qui permet d'obtenir des résultats relativement précis grâce à l'intervention de l'utilisateur.

Systèmes basés modèle

Les systèmes basés modèle permettent généralement d'obtenir de façon séquentielle un système robuste et autonome, pouvant prendre en compte des mouvements quelconques de caméra. Ceci provient essentiellement du fait que la reprojection des primitives 3-D dans l'image permet de vérifier et corriger automatiquement les erreurs d'appariement. Cependant, la ré-initialisation des primitives mal suivies, ou la prise en compte de nouvelles primitives lorsque le point de vue varie de façon significative, est un problème qui n'a pratiquement été résolu que dans le cas du suivi de points par corrélation (Uenohara et Kanade, Ravela et al.). Dans ce cas, la construction préalable d'une table d'aspect, basée sur l'observation de l'objet sous plusieurs angles et/ou plusieurs illuminations, est nécessaire. Ceci est particulièrement adapté pour des tâches devant être répétées plusieurs fois, comme les opérations de maintenance, autour d'un objet de taille assez réduite auquel on a facilement accès (nous pouvons par exemple difficilement envisager de filmer un édifice sous diverses conditions d'illumination). Pour d'autres applications, comme le traitement en phase de post-production d'une séquence vidéo, la constitution d'une table d'aspect n'a quasiment aucun sens puisqu'il suffirait, pour un investissement moindre, de définir les fenêtres de corrélation autour des points au fur et à mesure de leur apparition dans la séquence.

Lorsqu'on se base sur la carte des contours de l'image, le problème de la mise à jour des primitives est plus complexe. On peut envisager d'établir une comparaison morphologique entre la primitive projetée et les contours proches, mais on s'expose alors à des problèmes de conflits, d'autant plus importants que la segmentation est dense et/ou bruitée. D'autre part, la matrice utilisée pour projeter la primitive à initialiser ne peut être obtenue qu'à partir des primitives déjà appariées. Or, si ces primitives ne sont pas correctement distribuées dans le repère du monde, la projection de la primitive à initialiser peut être relativement éloignée de sa position 2-D réelle. Ce problème est important, bien que rarement évoqué dans la littérature. Nous montrerons comment l'utilisation de méthodes statistiques robustes permet de retrouver le contour correspondant, même pour certaines distributions non favorables.

Un autre problème majeur lié à la distribution des primitives suivies doit être pris en compte. Les algorithmes 3-D/2-D présentés en 3.2 sont soit des algorithmes de recalage d'objets non appliqués à la RA, soit des systèmes de RA permettant d'incruster les objets virtuels au niveau de l'objet recalé (annotations sur un moteur, épingle sur une jambe ...). L'incrustation d'un objet virtuel dans une zone éloignée de l'objet recalé n'a à notre connaissance pas été étudiée de façon satisfaisante. Cet aspect est pourtant à prendre en considération, car la projection d'un point distant des primitives utilisées pour le calcul de la matrice de projection peut être très éloignée de la projection attendue. L'avantage des méthodes basées images est qu'elles utilisent un nombre important de points distribués de façon aléatoire, et susceptibles de couvrir des zones de l'espace non modélisées. Notre idée est donc d'utiliser cette connaissance 2-D pour affiner le point de vue obtenu et permettre des incrustations dans n'importe quelle partie de la scène.

Stratégie adoptée

La stratégie que nous adoptons est donc la suivante :

- nous définissons un système de recalage 3-D/2-D basé sur la connaissance de courbes de forme libre. Ce système doit être suffisamment robuste pour détecter les primitives mal suivies et capable d'établir automatiquement les appariements 3-D/2-D (chapitre 5) ;
- nous utilisons l'information 2-D/2-D pour affiner le point de vue et permettre l'incrustation d'objets virtuels dans des zones éloignées de l'objet recalé. Pour cela, le point de vue est calculé en utilisant simultanément l'information 3-D/2-D et 2-D/2-D, et le système reste donc séquentiel. L'information 3-D/2-D permet de déterminer le facteur d'échelle et d'assurer l'autonomie et la robustesse du système, tandis que l'information 2-D/2-D permet de résoudre le problème de la distribution des primitives (chapitre 6).

La plupart des systèmes de recalage temporel sont basés sur l'hypothèse de constance des paramètres intrinsèques de la caméra au cours de la séquence. L'introduction de paramètres supplémentaires dans le calcul du point de vue aboutit généralement à des résultats moins précis. Nous proposons donc pour finir un algorithme permettant de prendre en compte des changements de focale, sous l'hypothèse d'une alternance entre plans à focale fixe et plans à caméra fixe (chapitre 7).

Nous présentons à présent notre premier système de RA, basé sur le recalage robuste à partir de points 3-D/2-D.

Chapitre 4

Recalage robuste à partir de correspondances de points

Un premier système de recalage temporel basé modèle a été réalisé pour le projet des ponts de Paris. Comme il s'agissait d'une première maquette, certains choix effectués peuvent paraître naïfs aujourd'hui : le système possède en effet plusieurs lacunes qui le rendent difficilement généralisable à d'autres applications et pas suffisamment autonome. Cependant, l'application du Pont-Neuf présente de nombreuses difficultés, qui se situent aussi bien au niveau du traitement des images que de la précision du modèle utilisé. Cela nous a donc contraint à utiliser des techniques d'estimation robuste, et nous a permis d'identifier clairement les problèmes susceptibles d'apparaître en présence de bruit.

Pour ce premier système, le recalage du modèle est effectué en deux temps : estimation grossière du point de vue par la méthode de Dementhon et Davis, puis affinement par une méthode itérative robuste. Les points servant au recalage sont suivis dans la séquence grâce à deux méthodes différentes, dépendant la nature du point suivi.

Nous commençons par décrire la méthode de Dementhon et Davis, puis introduisons quelques éléments de statistique robuste. La boucle de recalage temporel est alors détaillée et illustrée par l'application du Pont Neuf.

4.1 La méthode de Dementhon et Davis

Lorsque quatre points sont coplanaires, nous pouvons retrouver la perspective en utilisant le birapport comme invariant projectif [Ferri et al.93]. Nous avons étendu cette méthode au cas de n points [Berger et al.96b], en combinant les résultats obtenus pour des configurations optimales de quatre points (les critères de sélection étant basés sur la longueur des diagonales et l'angle entre les diagonales). Cela nous avait paru intéressant pour des applications architecturales, ou bien souvent seules des façades des édifices sont visibles. Malheureusement, comme le montre la figure 4.1 si l'erreur de reprojection des points utilisés pour le calcul du point de vue est généralement faible, celle-ci augmente au fur et à mesure que l'on s'éloigne de ces points (nous reparlerons de ce problème au chapitre suivant).

Nous avons finalement opté pour la méthode de Dementhon et Davis [Dementhon et al.95]. Ces derniers proposent un algorithme itératif en 25 lignes de code, utilisant la projection orthographique à l'échelle comme point de départ. Cette méthode, qui ne nécessite pas d'estimée initiale, nous a paru assez stable dans la plupart des cas, bien que relativement sensible au bruit présent soit dans l'image, soit dans le modèle.

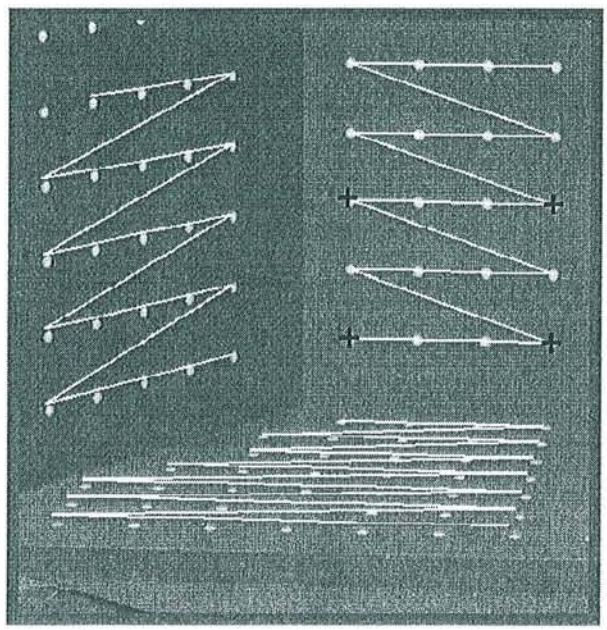


FIG. 4.1 – *Calcul du point de vue par la méthode de Ferri et al., à partir de quatre points mesurés très précisément dans un même plan d'une mire de calibration.*

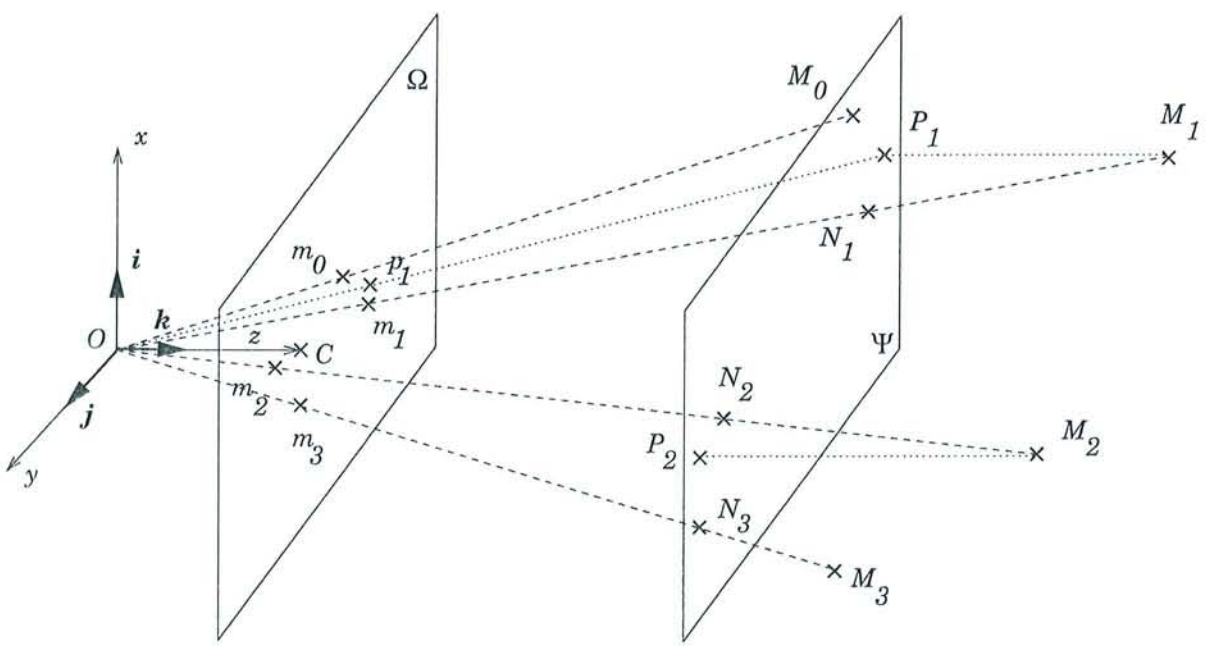


FIG. 4.2 – *Illustration de la méthode de Dementhon et Davis.*

Soit $\{M_i\}_{0\leq i\leq 3}$ les quatre points de référence et $\{m_i\}_{0\leq i\leq 3}$ leur projection dans le plan image (cf. figure 4.2). La matrice de rotation $\mathbf{R}^T$ est la matrice dont les colonnes sont les coordonnées des vecteurs unitaires **i**, **j**, **k** du repère de la caméra exprimés dans le repère de la scène. Pour calculer la rotation, il suffit donc de connaître les vecteurs **i** et **j**, **k** étant le produit vectoriel $\mathbf{i} \wedge \mathbf{j}$. Si l'on considère que le repère du monde est centré en M_0, le vecteur de translation **t** est le vecteur $\overrightarrow{OM_0} = \frac{Z_0}{f}\overrightarrow{Om_0}$ (où Z_0 est la coordonnée en z de M_0 exprimée dans le repère de la

caméra). Globalement, le point de vue est donc défini lorsque $\mathbf{i}$, $\mathbf{j}$ et Z_0 sont connus.

L'idée de base de la méthode décrite par [Dementhon et al.95] est de calculer une première approximation du point de vue en supposant que les points images ont été obtenus par une projection orthographique à l'échelle (*Pose from Orthography and Scaling* - POS pour abréger), c'est-à-dire en considérant que les points de référence ont une profondeur identique Z_0, qui peut alors être calculée. L'objet est ensuite placé à cette profondeur Z_0 en respectant l'appartenance des points de référence à leur ligne de vue. Une nouvelle projection orthographique à l'échelle est alors calculée et le processus est itéré (algorithme nommé POSIT).

Projection perspective

Soit Ψ le plan passant par M_0 et parallèle au plan image. Les lignes de vue passant par M_i intersectent Ψ en N_i, tandis que M_i se projette orthogonalement sur Ψ en P_i. On cherche à déterminer les produits scalaires de $\overrightarrow{M_0M_i}.\mathbf{i}$ et $\overrightarrow{M_0M_i}.\mathbf{j}$. On a

$$\overrightarrow{M_0M_i} = \overrightarrow{M_0N_i} + \overrightarrow{N_iP_i} + \overrightarrow{P_iM_i}.$$

Or il est clair que

$$\overrightarrow{M_0N_i} = \frac{Z_0}{f}\overrightarrow{m_0m_i}.$$

D'autre part, si C est l'intersection de l'axe des z avec le plan image, alors les deux vecteurs $\overrightarrow{N_iP_i}$ et $\overrightarrow{Cm_i}$ sont proportionnels :

$$\overrightarrow{N_iP_i} = \frac{\overrightarrow{M_0M_i}.\mathbf{k}}{f}\overrightarrow{Cm_i}.$$

Le produit scalaire $\overrightarrow{P_iM_i}.\mathbf{i}$ est nul, $\overrightarrow{m_0m_i}.\mathbf{i} = x_i - x_0$ et $\overrightarrow{Cm_i}.\mathbf{i} = x_i$. Ceci nous donne donc la première équation fondamentale de la projection perspective :

$$\overrightarrow{M_0M_i}.\frac{f}{Z_0}\mathbf{i} = x_i(1+\epsilon_i) - x_0,$$

où

$$\epsilon_i = \frac{1}{Z_0}\overrightarrow{M_0M_i}.\mathbf{k}.$$

De même,

$$\overrightarrow{M_0M_i}.\frac{f}{Z_0}\mathbf{j} = y_i(1+\epsilon_i) - y_0.$$

En posant $\mathbf{I} = \frac{f}{Z_0}\mathbf{i}$ et $\mathbf{J} = \frac{f}{Z_0}\mathbf{j}$, le système d'équations devient :

$$\begin{cases} \overrightarrow{M_0M_i}.\mathbf{I} = x_i(1+\epsilon_i) - x_0, \\ \overrightarrow{M_0M_i}.\mathbf{J} = y_i(1+\epsilon_i) - y_0. \end{cases} \tag{4.1}$$

Projection orthographique à l'échelle

Une projection orthographique à l'échelle est une approximation de la projection perspective, où l'on fait l'hypothèse que les profondeurs Z_i des points de la scène ne varient pas beaucoup et peuvent être posées égales à la profondeur d'un point de référence M_0. L'image d'un point M_i est donc un point du plan image Ω ayant pour coordonnées

$$x'_i = f\frac{X_i}{Z_0}, \quad y'_i = f\frac{Y_i}{Z_0}.$$

On peut établir une construction similaire à la précédente. On a :

$$\overrightarrow{M_0M_i} = \overrightarrow{M_0P_i} + \overrightarrow{P_iM_i}.$$

Le vecteur $\overrightarrow{M_0P_i}$ est égal au vecteur $\frac{Z_0}{f}\overrightarrow{m_0p_i}$, où p_i est l'image de P_i. Le produit scalaire de $\overrightarrow{m_0p_i}$ par $\mathbf{i}$ vaut $x'_i - x_0$ et le produit scalaire $\overrightarrow{P_iM_i}.\mathbf{i}$ vaut zéro. On a donc :

$$x'_i = x_i(1 + \epsilon_i),$$

et de même

$$y'_i = y_i(1 + \epsilon_i).$$

Estimation du point de vue

L'idée de base de l'algorithme POS est que si ϵ_i a une valeur fixe quelque soit i, le système (4.1) est un système d'équations linéaire dont les inconnues sont les coordonnées de $\mathbf{I}$ et $\mathbf{J}$. Une fois que l'on connaît $\mathbf{I}$ et $\mathbf{J}$, $\mathbf{i}$ et $\mathbf{j}$ sont obtenus en normalisant $\mathbf{I}$ et $\mathbf{J}$, et Z_0 est obtenu à partir de la norme de l'un des vecteurs : $Z_0 = \frac{f}{||\mathbf{I}||}$.

Comme nous l'avons vu précédemment, calculer le point de vue en fixant ϵ_i revient à déterminer le point de vue pour lequel les points M_i ont comme projections orthographiques à l'échelle les points images de coordonnées $[x_i(1 + \epsilon_i), y_i(1 + \epsilon_i)]$.

Initialement, on peut poser $\epsilon_i = 0$. L'algorithme POS résout alors le système (4.1) pour $\mathbf{i}$, $\mathbf{j}$ et Z_0. À partir de ces valeurs, le vecteur $\mathbf{k} = \mathbf{i} \wedge \mathbf{j}$ est calculé et une nouvelle valeur de ϵ_i est obtenue :

$$\epsilon_i = \frac{1}{Z_0}\overrightarrow{M_0M_i}.\mathbf{k}.$$

Le système (4.1) est alors résolu une nouvelle fois en utilisant cette valeur. La répétition de ce procédé est le cœur de l'algorithme POSIT. Quelques itérations suffisent généralement pour converger vers une estimation correcte du point de vue. Le test de convergence consiste à quantifier (en pixels) les coordonnées des points images obtenus par la projection orthographique à l'échelle et à s'arrêter lorsque deux images consécutives sont identiques.

4.2 Estimation robuste

La méthode de Dementhon et Davis détermine des paramètres du point de vue plus ou moins proches des paramètres attendus. Cette méthode est en effet relativement sensible aux erreurs de localisation des points de référence (même si elle fournit une assez bonne initialisation en général). Dans un contexte de suivi temporel, certains points peuvent être obtenus avec une précision faible, ou même être complètement faux si le suivi s'est mal passé. Nous allons donc utiliser les paramètres obtenus par la méthode de Dementhon et Davis pour converger itérativement vers une solution robuste, c'est-à-dire qui ne soit pas influencée par la présence de points erronés.

Soit l'erreur résiduelle en chaque point $r_i = ||m_i - Proj(\mathbf{R}M_i + \mathbf{t})||$ (où $Proj$ est l'opérateur de projection perspective). L'objectif est de trouver les six paramètres du point de vue $\mathbf{p}$ qui minimisent une fonction des résidus r_i (erreur de reprojection). Une estimation au sens des moindres carrés consisterait à minimiser la fonction :

$$f(\mathbf{p}) = \sum_{i=1}^{n} r_i^2, \tag{4.2}$$

Cette solution n'est cependant pas satisfaisante car très sensible aux données erronée (une seule mesure erronée peut conduire à un résultat aberrant). Les estimateurs robustes ont donc été introduits par les statisticiens [Rousseeuw et al.87] et largement utilisés en vision par ordinateur [Haralick et al.89, Kumar et al.94, Zhang et al.95]. Les estimateurs les plus couramment utilisés sont les M-estimateurs et les moindres carrés médians (*Least Median of Squares* - LMS). Les premiers minimisent la somme d'une fonction ρ des résidus, de manière à ce que les résidus les plus élevés (points erronés) aient une influence moindre dans l'optimisation. Les moindres carrés médians minimisent non plus la somme d'une fonction des résidus, mais la médiane des résidus au carré, de sorte que les résidus les plus élevés n'ont plus aucune influence dans l'optimisation.

Deux mesures sont communément employées pour évaluer la performance d'un estimateur :

- *le point de rupture*, qui est la plus petite fraction d'erreurs aberrantes présentes dans les données pouvant provoquer une estimation arbitrairement mauvaise,
- *l'efficacité relative*, qui est définie dans [Kim et al.89] comme étant la fraction entre la plus petite variance possible pour les paramètres estimés et la variance produite par la méthode donnée (la meilleure valeur possible est donc 1). En d'autres termes, cette mesure traduit la précision de l'estimateur.

4.2.1 Les M-estimateurs

La technique de M-estimation, développée par Huber [Huber81], consiste à minimiser la somme d'une fonction des résidus r_i :

$$f(\mathbf{p}) = \sum_{i=1}^{n} \rho(r_i), \tag{4.3}$$

où ρ une fonction continue et symétrique, ayant un minimum en zéro. En différenciant f par rapport aux six composantes de $\mathbf{p}$ et considérant que les dérivées sont nulles lorsque le minimum est atteint, on obtient les équations suivantes :

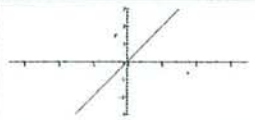
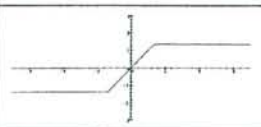
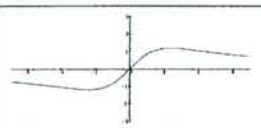
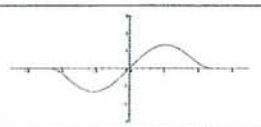
$$\sum_{i=1}^{n} \psi(r_i) \frac{\partial r_i}{\partial p_j} = 0, \text{ pour } j = 1,...,6, \tag{4.4}$$

où la dérivée $\psi(x) = \mathrm{d}\rho(x)/\mathrm{d}x$ est appelée *fonction d'influence*, car elle se comporte comme une fonction pondérante dans les équations (4.4).

La table 4.1 présente quelques fonctions d'influence couramment utilisées, avec leur représentation graphique. Pour une estimation aux moindres carrés classique (LS), l'influence d'un point croît linéairement en fonction de la taille de son résidu, ce qui traduit la non-robustesse de cet estimateur. A l'inverse, cette influence reste constante pour la distribution de Huber, décroît pour distribution de Cauchy ou même s'annule pour distribution de Tukey lorsque le résidu dépasse un certain seuil c.

Le seuil c doit être fixé en fonction du type de données concernées, pour un problème d'échelle évident. Nous prenons donc c égal à kS, où k est une constante et S un facteur d'échelle qui correspond à l'écart type robuste $\hat{\sigma}$ des résidus, dont le calcul est explicité plus loin.

Ces fonctions ont une efficacité élevée (typiquement plus de 0.9), mais un point de rupture de $1/(p+1)$ ou moins [Rousseeuw et al.87], où p est le nombre d'inconnues ($p = 6$ pour le calcul du point de vue). Elles sont donc particulièrement adaptées à des situations où le nombre de données aberrantes est assez faible (typiquement inférieur à 20%).

Nom	$\rho(x)$	$\psi(x)$	Graph. de $\psi(x)$
LS	$x^2/2$	x	
Huber $\begin{cases} \text{si } \lvert x\rvert \leq c \\ \text{si } \lvert x\rvert > c \end{cases}$	$\begin{cases} x^2/2 \\ c(\lvert x\rvert - c/2) \end{cases}$	$\begin{cases} x \\ c * \text{sgn}(x) \end{cases}$	
Cauchy	$\frac{c^2}{2}\log\left(1+\left(\frac{x}{c}\right)^2\right)$	$\frac{x}{1+\left(\frac{x}{c}\right)^2}$	
Tukey $\begin{cases} \text{si } \lvert x\rvert \leq c \\ \text{si } \lvert x\rvert > c \end{cases}$	$\begin{cases} \frac{c^2}{6}\left[1-\left(1-\left(\frac{x}{c}\right)^2\right)^3\right] \\ c^2/6 \end{cases}$	$\begin{cases} x\left(1-\left(\frac{x}{c}\right)^2\right)^2 \\ 0 \end{cases}$	

TAB. 4.1 – *Quelques M-estimateurs couramment utilisés.*

4.2.2 Les moindres carrés médians

La technique LMS, suggérée par Rousseeuw et Leroy [Rousseeuw et al.87], consiste à minimiser la médiane des résidus au carré :

$$f(\mathbf{p}) = \text{med}_i r_i^2. \tag{4.5}$$

On ignore ainsi la deuxième moitié des résidus triés par ordre croissant, ce qui confère à cet estimateur un point de rupture de 0.5. Cependant, l'estimateur LMS a une faible efficacité (voir [Rousseeuw et al.87] pour plus de détails). Pour remédier à cela, Rousseeuw et Leroy proposent d'effectuer une estimation aux moindres carrés triés (*Least Trimmed Squares* - LTS) :

$$f(\mathbf{p}) = \sum_{i=1}^{h} r_{o(i)}^2, \tag{4.6}$$

où $r_{o(1)}^2 \leq \ldots \leq r_{o(n)}^2$ sont les résidus au carré ordonnés, et h est égal à $n - [n/2]$ pour un point de rupture de 0.5. Cette estimation est très semblable à l'estimation aux moindres carrés classique, à ceci près que les résidus les plus grands ne sont pas pris en compte. Elle bénéficie donc d'une meilleure efficacité relative que l'estimation aux moindres carrés médians, mais n'utilise toujours qu'une partie des données disponibles (sauf pour $h = n$ qui nous ramène aux moindres carrés classiques).

Rousseeuw et Leroy préconisent donc d'affiner ces résultats par une étape supplémentaire d'estimation aux moindres carrés réduits (*Reduced Least Squares* - RLS), qui revient prendre en compte toutes les données dont le résidu est inférieur à un certain seuil :

$$f(\mathbf{p}) = \sum_{i=1}^{n} w_i r_i^2, \text{ où } w_i = \begin{cases} 1 \text{ si } \lvert r_i\rvert \leq 2.5\,\hat{\sigma}\,, \\ 0 \text{ si } \lvert r_i\rvert > 2.5\,\hat{\sigma}\,. \end{cases} \tag{4.7}$$

$\hat{\sigma}$ est une approximation de l'écart type des erreurs résiduelles qui doit elle-même être estimée

de façon robuste : on prend

$$\hat{\sigma} = 2.6477 \sqrt{\frac{1}{h} \sum_{i=1}^{h} r^2_{o(i)}}, \tag{4.8}$$

où h est égal à $n - [n/2]$. Le facteur 2.6477 est introduit car $\frac{1}{h} \sum_{i=1}^{h} r^2_{o(i)} = 1/2.6477^2$ lorsque les résidus sont des variables aléatoires distribuées selon la loi de distribution normale $\mathcal{N}(0,1)$.

Cette dernière étape permet dans certains cas d'obtenir des résultats plus précis, mais dans la pratique les estimateurs LMS et LTS aboutissent fréquemment à une configuration telle que les points utilisés dans l'optimisation (4.5) ou (4.6) obtiennent un résidu très petit par rapport à ceux des autres points. Ces derniers ne sont donc pas non plus pris en compte dans l'optimisation (4.7) (des résultats quantitatifs seront donnés au chapitre 5).

Nous avons donc retenu les M-estimateurs pour le recalage, en raison de leur précision et du faible taux de points aberrants observé dans nos expérimentations.

4.2.3 Minimisation de la fonction de coût

On cherche donc à minimiser la fonction (4.3) qui dépend de six paramètres. De nombreux travaux ont été dévolus à la minimisation d'une fonction à plusieurs paramètres partant d'une estimée initiale $\mathbf{p}_0$ [Stoer et al.80, Press et al.88]. Ce problème complexe est généralement décomposé en plusieurs sous-problèmes plus simples : un ensemble de minimisations à une dimension effectuées successivement dans plusieurs directions. La principale différence entre les méthodes proposées réside dans le choix de ces directions. Par exemple :

- Powell propose un algorithme très simple, dit de *convergence quadratique*, qui converge vers un minimum local : la direction utilisée à l'étape $i+1$ est $\mathbf{p}_i - \mathbf{p}_{i-m}$, où m est le nombre de paramètres de la fonction.
- La méthode du gradient conjugué effectue, à l'étape $i+1$, une minimisation dans la direction $-\bigtriangledown f(\mathbf{p}_i)$.

La méthode de Powell est très utile dans le cas où le gradient de f n'est pas connu. Dans notre cas, on connaît la dérivée de ρ, et donc le gradient de f. Chacune des deux méthodes a été implémentée avec la distribution de Cauchy vue précédemment, mais la méthode de Powell s'est avérée être la plus efficace dans notre application : convergence plus rapide et risque moindre de converger vers un minimum local (en fait, pour une initialisation pas trop éloignée de la solution, nous n'avons eu aucun problème de convergence avec Powell, alors que la méthode du gradient conjugué convergeait fréquemment vers un minimum local).

4.3 La boucle de recalage temporel, version 1

Nous détaillons à présent l'implémentation des différentes fonctions de la boucle de recalage présentée au chapitre 3 (figure 3.2), et montrons les résultats obtenus sur la séquence du Pont Neuf.

4.3.1 Le projet d'illumination des ponts de Paris

Le projet d'illumination des ponts de Paris a été présenté au chapitre 1 : il s'agit de déterminer le point de vue de la caméra sur une séquence de 300 images du Pont Neuf (la caméra décrit un

mouvement panoramique). Les difficultés rencontrées pour cette application résident dans :

- le manque de clarté des images, prises à la tombée de la nuit (une image extraite de la séquence est montrée en figure 4.3), ce qui limite sensiblement le nombre de primitives détectables et les stratégies de suivi potentielles,
- la modélisation du pont, réalisée manuellement à partir de plans d'architecte : la figure 4.4 montre que celle-ci est très incertaine à plusieurs endroits (certaines des arches subissent une déformation en forme d'escalier). Une modélisation au laser nous aurait certainement permis d'obtenir des résultats plus précis.
- le manque d'information de profondeur : dans cette séquence, nous voyons principalement la façade du pont, qui est quasiment plane. Seuls les lampadaires et les piles du pont donnent une indication de profondeur, mais en ce qui concerne les piles, il faut pouvoir évaluer le niveau de l'eau que nous ne connaissons pas.

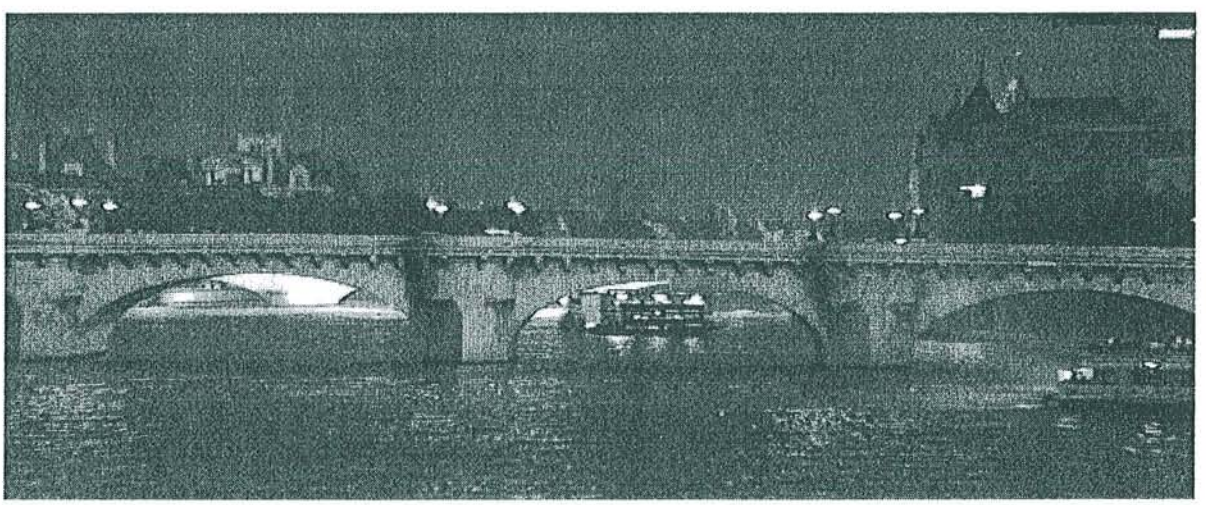


FIG. 4.3 – *La* $160^{ème}$ *image de la séquence.*

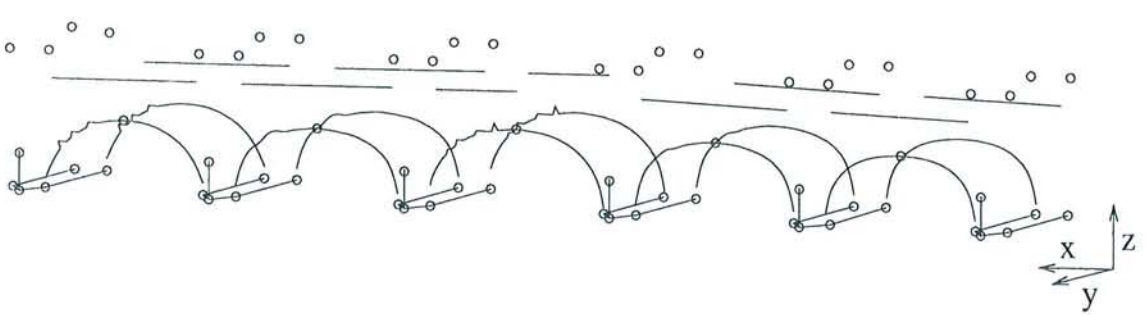


FIG. 4.4 – *Le modèle filaire du pont. La mauvaise qualité de cette modélisation est clairement visible à certains endroits. Les cercles désignent la localisation des points utilisées pour le recalage.*

4.3.2 Initialisation du système

L'initialisation du système comporte trois étapes : la détermination des paramètres intrinsèques de la caméra, la mise en correspondance de points du modèle avec des points de la première image et le calcul du point de vue dans la première image.

Détermination des paramètres intrinsèques de la caméra

Ces paramètres peuvent être obtenus selon deux manières :

- en utilisant une mire de calibration : pour la séquence du Pont Neuf, une mire de calibration a été filmée à la suite de la séquence, ce qui nous a permis de déterminer les paramètres intrinsèques de la caméra [Faugeras et al.86]. Notons que cette mire d'environ 30cm de large a été filmée à une distance de 3 mètres, alors que le pont est situé à environ 300 mètres de la caméra, ce qui renforce l'erreur obtenue sur les paramètres intrinsèques.
- en utilisant le modèle de la scène : si nous disposons de suffisamment de points appariés (au minimum six en théorie), nous pouvons les utiliser pour calculer simultanément les paramètres intrinsèques et extrinsèques de la caméra. Dans la pratique, de nombreux points sont nécessaires pour obtenir un résultat fiable. C'est pourquoi nous préférons découpler la calibration de la caméra du calcul du point de vue, notamment pour cette séquence où le nombre de points appariés est très faible.

Mise en correspondance des points et calcul du point de vue dans la première image

L'appariement de points 3-D/2-D dans la première image est actuellement effectué de façon manuelle. Nous pourrions envisager de détecter automatiquement m points d'intérêt dans l'image (coins, jonctions en T ...) et n points d'intérêt dans le modèle, puis tenter d'apparier automatiquement ces points, en calculant le point de vue pour chaque configuration et en retenant la configuration donnant lieu à la plus petite erreur de reprojection. Malheureusement, cette technique est extrêmement coûteuse ($\frac{m!}{(m-n)!}$ combinaisons possibles), et d'autre part les points d'intérêt détectés dans l'image ne correspondent pas nécessairement à des points du modèle. L'appariement des points est donc effectué à la main par l'opérateur (par le biais d'une interface graphique par exemple). Quatre points suffisent pour calculer le point de vue par la méthode de Dementhon et Davis, puis cette estimation peut être affinée par optimisation robuste. La figure 4.5.a montre le point de vue obtenu par Dementhon et Davis à partir de quatre points désignés à la main (symbolisés par des croix). La figure 4.5.b montre la reprojection du modèle obtenue après 23 itérations de l'algorithme de Powell (l'amélioration du résultat est particulièrement évidente au niveau de la partie inférieure des arches arrières du pont).

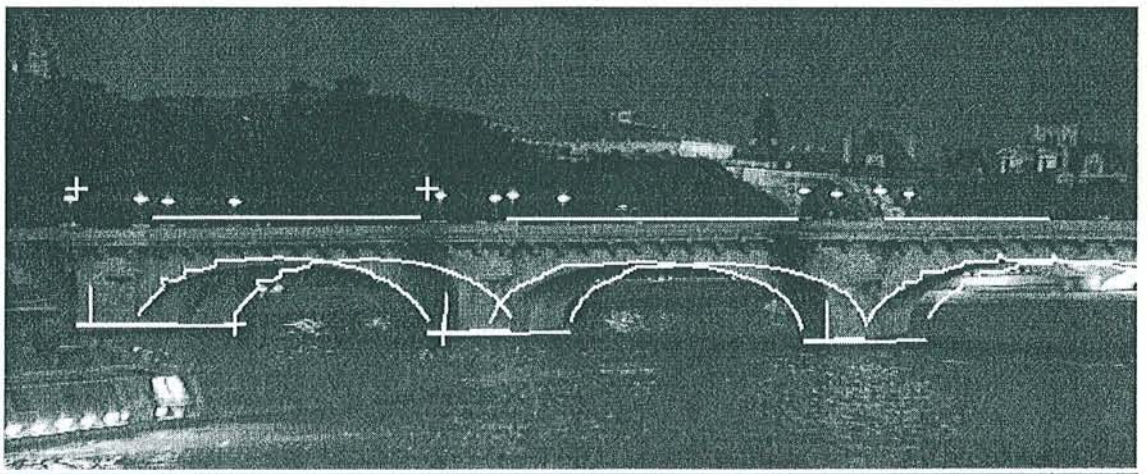
4.3.3 Suivi des primitives

Nous suivons deux sortes de primitives : des points et des courbes (les arches du pont). Dans cette première version du système, les arches ne sont pas utilisées directement pour l'estimation du point de vue, mais uniquement parce qu'elles produisent des points intéressants (le sommet de chaque arche par exemple, comme le point 4 de la figure 4.6.c).

La séquence considérée ayant été filmée à la tombée de la nuit, la segmentation des images est très pauvre, ce qui ne nous permet pas d'utiliser la carte des contours (cf. figure 4.6.a). Nous utilisons donc un algorithme de suivi basé sur le flot optique pour suivre certaines courbes : ainsi, plutôt que de suivre les points situés à la base des piliers du pont, nous suivons les courbes désignées en figure 4.6.b, et utilisons les points anguleux de ces courbes.

Les lampadaires du pont, dont la position 3-D est connue, constituent une seconde source de points pertinents : la lumière émise par ces lampes apparaît en effet comme des taches blanches dans les images, que nous pouvons suivre facilement par une technique de corrélation.

a.



b.

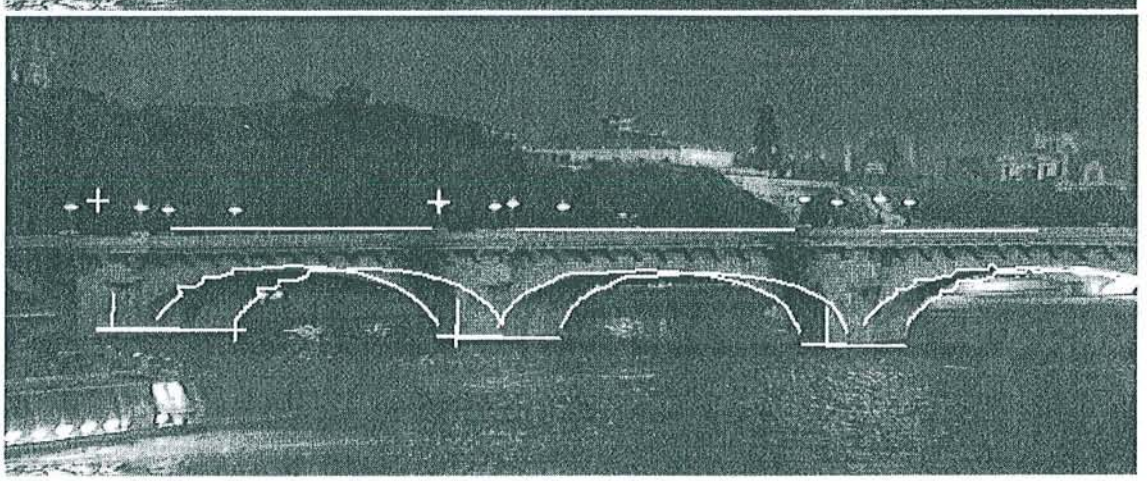


FIG. 4.5 – *Initialisation. a. Estimation initiale par la méthode de Dementhon et Davis appliquée à quatre points non coplanaires. b. Estimation affinée obtenue après 23 itérations de l'algorithme de Powell.*

Suivi par corrélation

Dès que de nouveaux lampadaires apparaissent dans la séquence, le centre de la boule de lumière est détecté de la façon suivante : l'opérateur dessine un petit polygone entourant la boule; comme cette boule apparaît comme une tache blanche dans l'image, elle peut être détourée par un simple seuillage. Le centre de cette tache est considéré comme l'image du centre 3-D de la boule.

La lampe est alors suivie le long de la séquence par corrélation. Partant d'un petit carré entourant la lampe dans l'image courante, l'algorithme de corrélation calcule la translation qui donne le meilleur ajustement de cette fenêtre dans l'image suivante. Le centre de la tache blanche dans cette nouvelle fenêtre donne la position du point dans l'image suivante. Les points détectés par cette méthode apparaissent sur la figure 4.6.c.

Suivi des contours

Les points autres que les centres des lampes sont suivis par un algorithme de suivi de contours. L'algorithme que nous utilisons est décrit dans [Berger94]. Il comporte les deux étapes suivantes :

- une étape de *prédiction* : une approximation du champ des déplacements 2-D est calculée

itérativement sur toute la courbe à partir du flux optique normal;
- une étape de *convergence*: à partir de la courbe prédite, un contour actif [Kass et al.88] converge vers le contour le plus proche qui est généralement le contour homologue (certaines stratégies ont été développées pour mieux assurer cette convergence).

Une fois qu'un point d'intérêt a été choisi, une courbe suffisamment contrastée contenant ce point est détourée en s'aidant de la technique des contours actifs. Cette courbe est suivie le long de la séquence en utilisant l'algorithme qui vient d'être décrit. On recherche ensuite le point anguleux de chacune des courbes: pour chaque point m_i appartenant à une telle courbe, les deux droites approximant au mieux $m_{j_{\{j<i\}}}$ et $m_{j_{\{j\geq i\}}}$ sont calculées. Le point donnant lieu au meilleur score d'approximation est retenu et l'intersection des deux droites est considérée comme étant le point recherché (point 1 sur la figure 4.6.b). Cette méthode est utilisée pour suivre les points anguleux situés entre deux droites ou entre deux courbes. En fait, lorsque la courbe contenant le point est suffisamment petite, l'approximation de la courbe par deux droites est satisfaisante (points 2 et 3 sur la figure 4.6.b). Cette méthode s'avère robuste au bruit, alors que la plupart des points ne seraient pas facilement détectables dans l'image.

Malheureusement, certains de ces points anguleux ne correspondent pas aux points 3-D du modèle parce qu'ils sont à l'intersection du pont avec la surface du fleuve (points 1, 2 et 3 sur la figure 4.6.b). La hauteur de ces points peut cependant être grossièrement estimée car les dimensions du pont sont connues. Le niveau de la surface du fleuve est alors affinée de la façon suivante: à partir de l'estimation initiale h_0, nous calculons, pour chaque hauteur h prise à écarts réguliers dans l'intervalle $[h_0 - 50cm, h_0 + 50cm]$, le point de vue correspondant à l'ensemble des points suivis quand la hauteur des points à la surface de la rivière vaut h. La hauteur correcte est celle pour laquelle l'erreur de reprojection est la plus faible. Cette opération n'est bien sûr effectuée qu'une seule fois dans la première image.

L'ensemble des points suivis dans l'image 60 apparaissent sur la figure 4.6.c.

4.3.4 Recalage

Nous supposons à présent qu'un certain nombre de points 2-D dont les correspondants 3-D sont connus ont été suivis dans l'image courante. Il s'agit maintenant de recaler l'objet par rapport à ces points images pour connaître le nouveau point de vue. Nous pouvons pour cela calculer une première approximation du point de vue par la méthode de Dementhon et Davis puis affiner cette estimation par optimisation robuste. Une autre solution consiste à utiliser le point de vue calculé dans l'image précédente comme point de départ pour l'optimisation de Powell. Ces deux méthodes se sont avérées satisfaisantes dans nos expérimentations, mais seule la première est présentée ici. La figure 4.7 est un tracé de l'erreur de reprojection moyenne incluant toutes les primitives visibles pour chaque image de la séquence. La courbe à motifs carrés indique l'erreur moyenne obtenue après calcul de l'estimée initiale par Dementhon et Davis, tandis que la courbe à motifs ronds montre l'erreur de reprojection obtenue après estimation robuste du point de vue. Malgré les erreurs d'imprécisions liées à la détection des points et à la modélisation du modèle, l'erreur de reprojection reste à un niveau bas et relativement acceptable.

4.3.5 Mise à jour des primitives

Dans cette première version de notre système, la mise à jour des primitives visibles dans la séquence est effectuée à la main. Généralement, une dizaine de points bien répartis dans l'image sont nécessaire pour calculer le point de vue de façon satisfaisante. Malheureusement, le nombre de points suivis diminue au cours de la séquence, soit parce que des primitives disparaissent

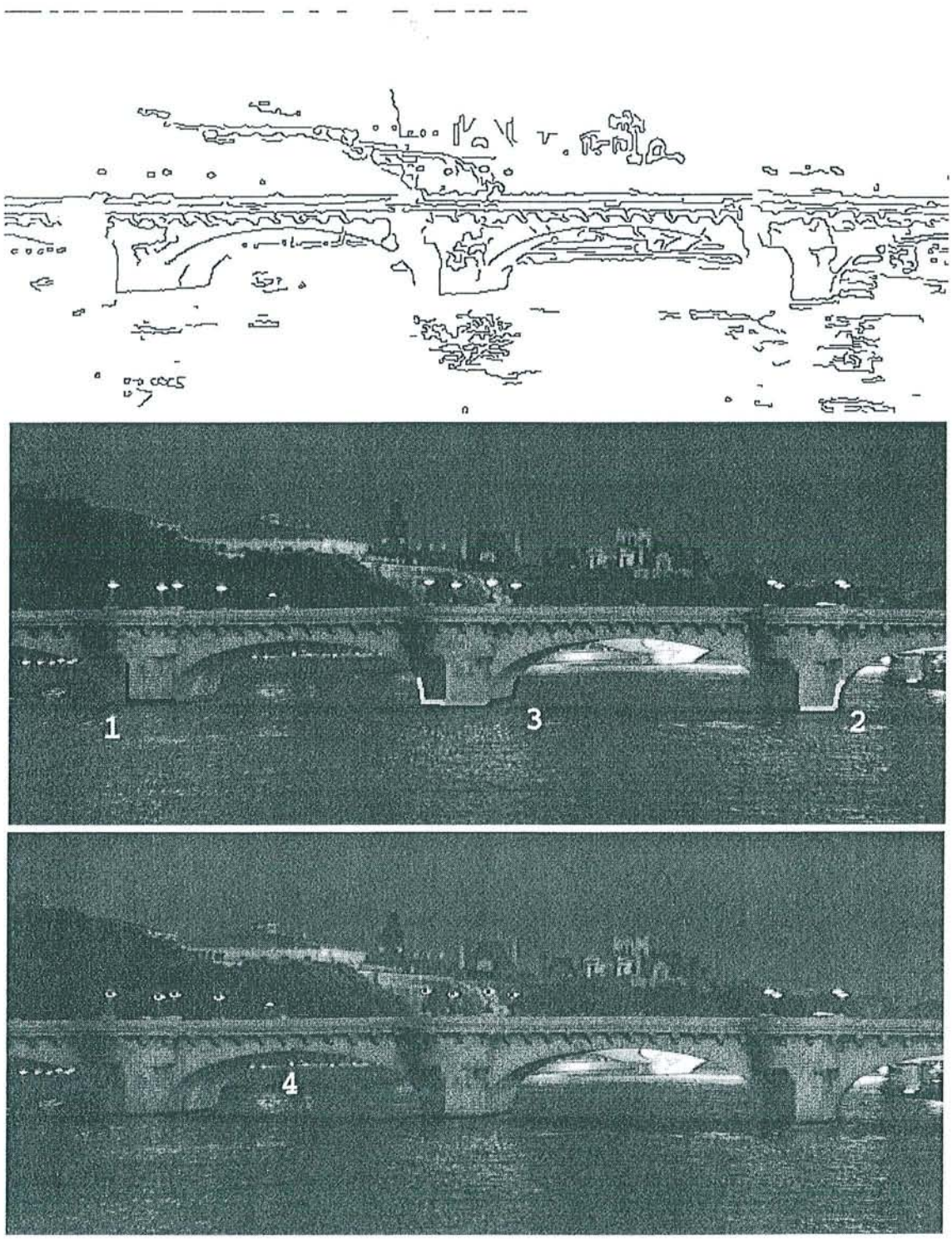


FIG. 4.6 – *Suivi des points dans l'image* 60. *a. Carte des contours. b. Courbes obtenues par la technique des contours actifs. c. Points obtenus par les deux méthodes de suivi.*

du champ de vision, soit parce qu'elles sont mal suivies. Quand ce nombre devient trop petit, il faut désigner de nouveaux points dans l'image, ainsi que la technique qui doit être utilisée

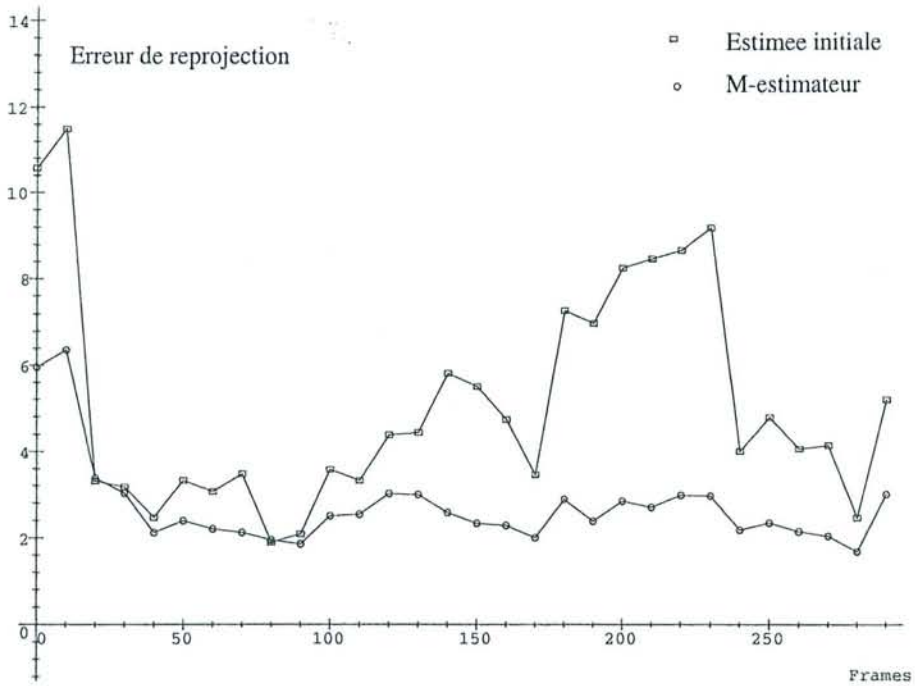


FIG. 4.7 – *Erreur de reprojection en pixels après la première estimation (carrés) et après raffinement par l'estimation robuste (cercles).*

pour les suivre. Typiquement, cette opération de mise à jour a été effectuée environ dix fois sur l'application du Pont Neuf.

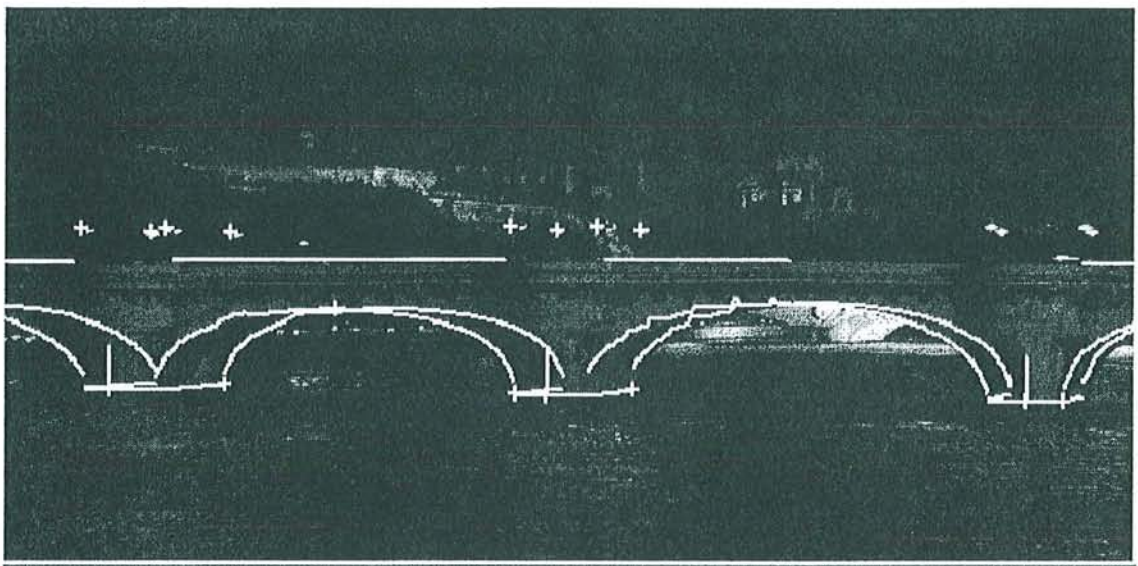
4.3.6 Résultats

La figure 4.8 montre un résultat de composition dans la $60^{ème}$ image de la séquence. En 4.8.a, nous voyons l'image originale sur laquelle est superposée la projection du modèle filaire du pont. La figure 4.8.b montre le résultat final, incluant le rendu du pont et de son reflet dans l'eau. Plusieurs images extraites de la séquence augmentée sont par ailleurs présentées en figure 4.9. Dans cette séquence, les masques d'occultation ont été détourés à la main toutes les 10 images, puis interpolés.

Des vidéos au format MPEG sont disponible sur notre site internet (voir 2.5.2 pour l'adresse). La première vidéo présente les points suivis sur la séquence, la seconde montre la projection du modèle filaire du pont et la dernière est la séquence augmentée.

Ces résultats prouvent que notre système est capable de prendre en compte des images de mauvaise qualité et une modélisation imprécise de l'objet de référence. De plus, il fonctionne sur un large champ de vision : dans l'application du Pont Neuf, le mouvement est purement panoramique et la rotation totale de la caméra est d'environ 20 degrés (le pont mesure 90 mètres de long et le point de vue est situé à 300 mètres du pont). Considérant les conditions que nous avons déjà mentionnées, l'impression visuelle d'ensemble est donc plutôt satisfaisante. Toutefois, un léger effet de sautillement peut être observé sur la vidéo.

a.



b.

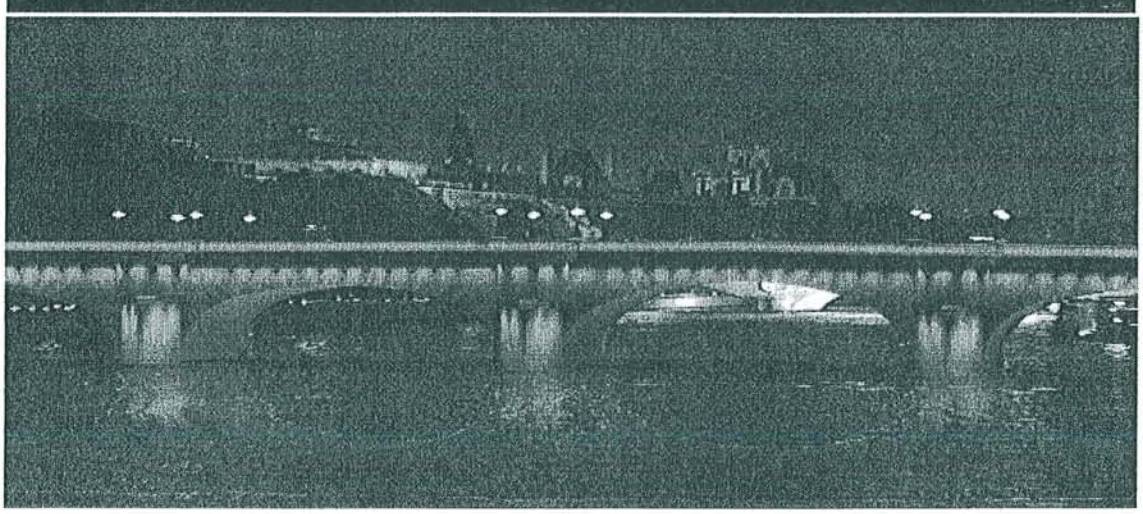


FIG. 4.8 – *Résultats pour la 60ème image de la séquence. a. Projection du modèle filaire et des points utilisés pour le calcul du point de vue. b. Résultat de la composition.*

4.4 Limites du système

Ce premier système nous a permis de résoudre le problème du recalage pour une séquence grandeur réelle de taille relativement importante. Si le résultat est visuellement convaincant pour l'application du Pont Neuf, le système possède toutefois quelques points faibles qui le rendent difficilement généralisable à d'autre applications. Ces points faibles sont principalement la nature des primitive suivies et (par voie de conséquence) la faible autonomie du système.

4.4.1 Nature des primitives suivies

Le fait d'utiliser des appariements de points pour calculer le point de vue présente quelques inconvénients directs :

- premièrement, nous ne disposons pas toujours de suffisamment de points d'intérêts dont nous connaissons le correspondant 3-D dans la scène. Nous avons vu que pour l'application du Pont Neuf, peu de points pertinents étaient finalement disponibles : par chance, les

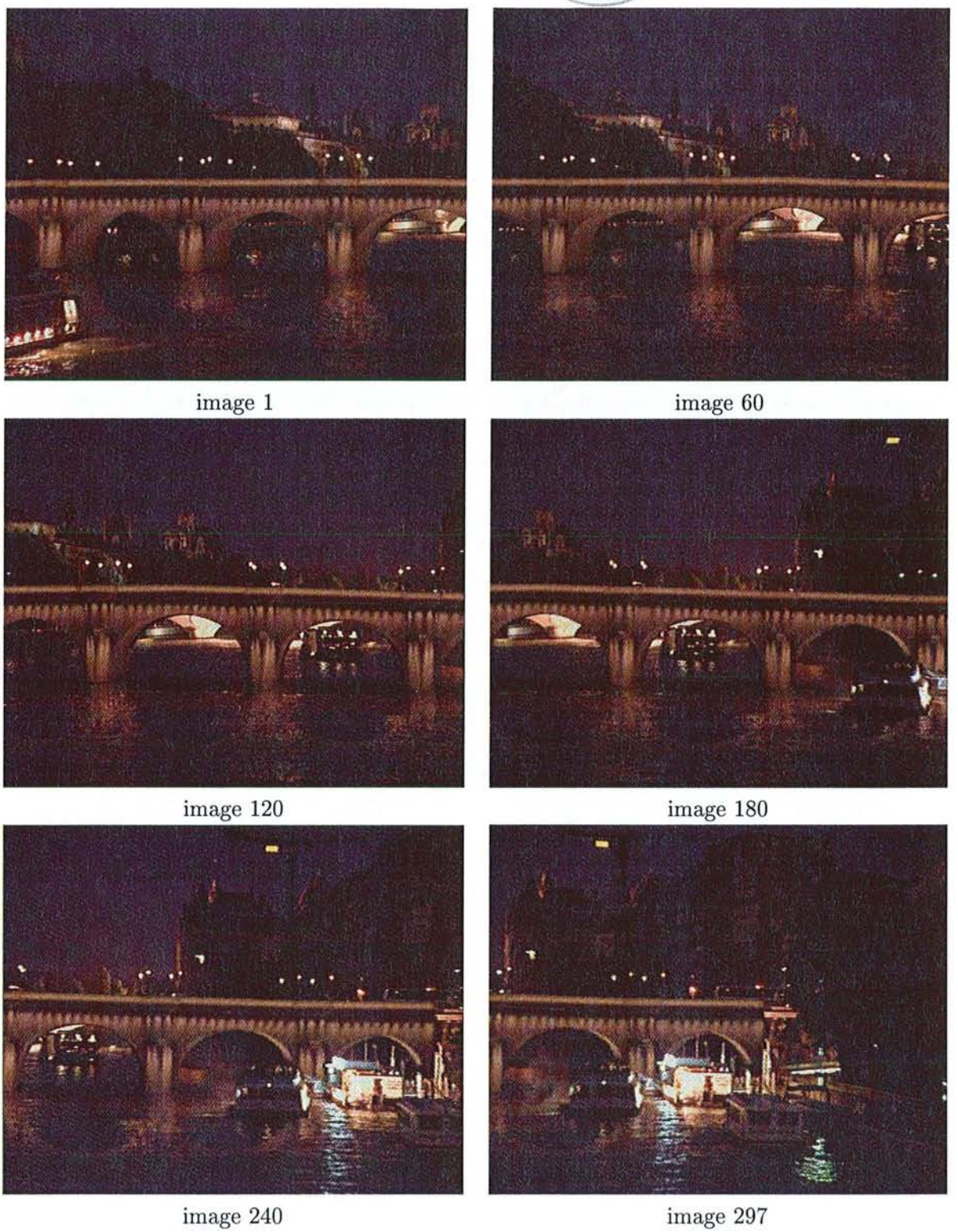


image 1 image 60

image 120 image 180

image 240 image 297

FIG. 4.9 – *Séquence augmentée du Pont-Neuf.*

lampadaires apparaissaient comme des taches blanches très évidentes que nous pouvions suivre facilement par corrélation, mais de telles primitives aussi commodes ne sont mal-

heureusement pas toujours présentes dans une séquence quelconque. Par ailleurs, les autres points utilisés n'ont pas été extraits directement du modèle puisque le niveau de la rivière n'était pas connu, et que le sommet des arches devait être calculé. La détermination de ces points requérait donc une manipulation préalable bien spécifique à l'application.

- les techniques de suivi doivent être adaptées à la nature des points suivis, ce qui signifie que nous devons établir en phase préliminaire les associations entre points d'intérêt et méthodes de suivi adéquates. Pour un modèle de taille importante, cette opération peut s'avérer fastidieuse pour un opérateur humain.

4.4.2 Autonomie du système

L'utilisation de primitives points est aussi indirectement responsable de la faible autonomie du système. Nous avons vu que le nombre de points suivis ne pouvait que décroître au cours de la séquence (points mal suivis ou points sortant du champ de vision) et risquait de devenir trop faible pour calculer le point de vue de façon satisfaisante. La boucle de recalage doit alors être interrompue et de nouveaux points doivent être désignés par l'opérateur, ainsi que la technique de suivi à utiliser.

Ces points ne peuvent être déterminés automatiquement : pour les points anguleux, nous pouvons envisager de prendre le coin le plus proche de la projection du point 3-D recherché, mais on s'expose alors à des problèmes d'ambiguïté lorsque plusieurs coins sont détectés dans une même zone. Pour les points suivis par corrélation, il faudrait pouvoir disposer, comme dans [Ravela et al.96], d'une table d'aspect associant à chaque point que nous souhaitons suivre sa fenêtre d'apparence.

La mise à jour manuelle des points suivis est peu pratique et rend définitivement inenvisageable l'utilisation du système dans un contexte temps réel. Nous allons voir que l'utilisation de primitives courbes de forme libre permet d'obtenir une boucle de recalage généralisable, et rend possible la mise à jour automatique des primitives suivies.

Chapitre 5

Prise en compte de courbes quelconques

Nous montrons dans ce chapitre comment la prise en compte de courbes quelconques permet d'obtenir un système de recalage temporel général et autonome. Nous proposons pour cela une méthode originale de calcul robuste du point de vue, basée sur la connaissance de courbes 3-D de forme libre et d'une estimation initiale du point de vue. Cette méthode utilise les techniques d'estimation robuste à deux niveaux : un niveau local à chaque courbe de référence et un niveau global à l'ensemble de ces courbes. Le choix des estimateurs robustes utilisés pour chacun de ces niveaux est justifié à l'aide de tests synthétiques.

L'aptitude de cette méthode à détecter les appariements non valides permet de contrôler automatiquement la qualité du suivi (et de corriger les éventuelles erreurs pouvant se produire), aussi bien que les apparitions ou disparitions de courbes au cours de la séquence. Elle est donc au cœur de la deuxième version de notre système. Des résultats expérimentaux obtenus sur la séquence du Pont-Neuf et une séquence de laboratoire plus saccadée sont présentés. Enfin, les limites de ce système sont établies.

Nous supposons dans ce chapitre et le chapitre suivant que les paramètres intrinsèques de la caméras sont connus *a priori* **et ne varient pas au cours de la séquence.**

5.1 Une méthode statistique robuste à deux niveaux

La plupart des algorithmes de calcul du point de vue utilisent des primitives simples : points (voir chapitre précédent), droites [Dhome et al.89, Shakunaga93, Kumar et al.94] ou cercles [Ferri et al.93]. Peu de travaux sont consacrés au recalage à partir de courbes 3-D de forme libre. Kriegman et Ponce proposent une méthode algébrique pour calculer le point de vue à partir de l'observation des contours occultant d'un objet courbe [Kriegman et al.90]. Malheureusement, cette méthode ne fonctionne qu'avec des surfaces ou des courbes paramétriques (ou plus exactement exprimables sous forme de fractions de polynômes) et utilise la théorie de l'élimination qui est très lourde à mettre en oeuvre [Simon95] (un logiciel de calcul symbolique doit être utilisé pour déterminer l'équation paramétrique de la courbe projetée, et cette équation doit apparaître dans le code de l'optimisation). Le problème du recalage 3-D/2-D est aussi considéré dans [Feldmar et al.97], pour des applications médicales. Partant d'une estimation grossière du point de vue, un algorithme de calcul itératif du point le plus proche (*Iterative Closest Point* - ICP) est utilisé pour effectuer le recalage. Un filtre de Kalman étendu est utilisé pour éliminer les erreurs grossières en effectuant des tests de χ^2. Cependant, comme le résultat dépend de l'ordre dans lequel on réalise les mesures, l'estimation obtenue risque de n'être que localement minimale, en particulier si l'estimée initiale n'est pas très précise. D'autre part, les travaux de Kriegman et

Ponce et Feldmar et al. se placent dans un contexte bien particulier où l'aspect multi-courbes n'est pas évoqué.

5.1.1 Principe

Le but de notre algorithme est de calculer le point de vue à partir d'une estimée initiale $\mathbf{p_0}$ et des correspondances 3-D/2-D de n primitives quelconques (points, droites et courbes 3-D), décrites par des chaînes de points. Notons :

- C_i une courbe 3-D, décrite par la chaîne de points 3-D $\{M_{i,j}\}_{1\leq j\leq l_i}$ (notons que C_i peut être indifféremment un point, une droite ou une courbe 3-D de forme libre),
- c_i la projection de C_i dans le plan image, décrite par la chaîne de points 2-D $\{m_{i,j}\}_{1\leq j\leq l_i}$, où $m_{i,j} = Proj(\mathbf{R}M_{i,j} + \mathbf{t})$,
- c'_i le correspondant image de C_i, décrit par la chaîne de points 2-D $\{m'_{i,j}\}_{1\leq j\leq l'_i}$.

Une solution simple consisterait à minimiser globalement la fonction

$$f(\mathbf{p}) = \sum_{i,j} \rho(d_{i,j}) \tag{5.1}$$

où $d_{i,j} = Dist(m'_{i,j},c_i)$ et $Dist$ est une fonction qui approxime la distance euclidienne d'un point à un contour (décrite plus loin).

Malheureusement, cette méthode ne nous satisfait pas pour deux raisons majeures, liées au contexte du recalage temporel :

- lorsque des primitives sont suivies dans une séquence, deux types d'erreur bien dissociés peuvent apparaître : certaines primitives peuvent être localement incorrectes lorsqu'elles sont localement mal suivies ou partiellement occultées par un objet de la scène. D'autres primitives peuvent être globalement aberrantes lorsqu'elles sont complètement occultées ou que l'algorithme de suivi converge mal (un exemple obtenu dans la séquence du Pont Neuf où ces deux types d'erreur apparaissent est présenté en figure 5.1). La fonction (5.1) ne tient pas compte de cette distinction, alors que ces deux types d'erreur ne sont pas de même nature et ne pas les dissocier rend l'algorithme moins robuste et moins précis. Nous souhaiterions à l'inverse pouvoir éliminer les primitives aberrantes et utiliser l'information efficace des primitives localement correctes;
- d'autre part, lorsque nous utilisons la fonction (5.1), les primitives sont traitées simultanément comme un ensemble de points et non comme des entités indépendantes. Ainsi, plus la taille d'une primitive est importante, plus elle a de poids dans l'estimation, alors qu'un point isolé (un coin par exemple) peut être aussi pertinent qu'une courbe. En outre, cela ne permet pas de gérer correctement les mises à jour : une courbe doit pouvoir être éliminée ou initialisée globalement, alors que l'utilisation de la fonction (5.1) permet uniquement de traiter des points.

Pour remédier à cela, nous proposons d'utiliser des estimateurs robustes à deux niveaux : à un niveau local, où un résidu robuste est calculé pour chaque primitive, et à un niveau global, où une fonction robuste de ces résidus est minimisée. Le niveau local a pour but de réduire l'influence des erreurs locales tandis que le niveau global doit permettre d'éliminer les primitives aberrantes. Notons qu'un algorithme RANSAC est aussi utilisé à deux niveaux dans [Armstrong et al.95], à partir de correspondances de segments (la méthode pouvant être étendue à des courbes paramétriques).

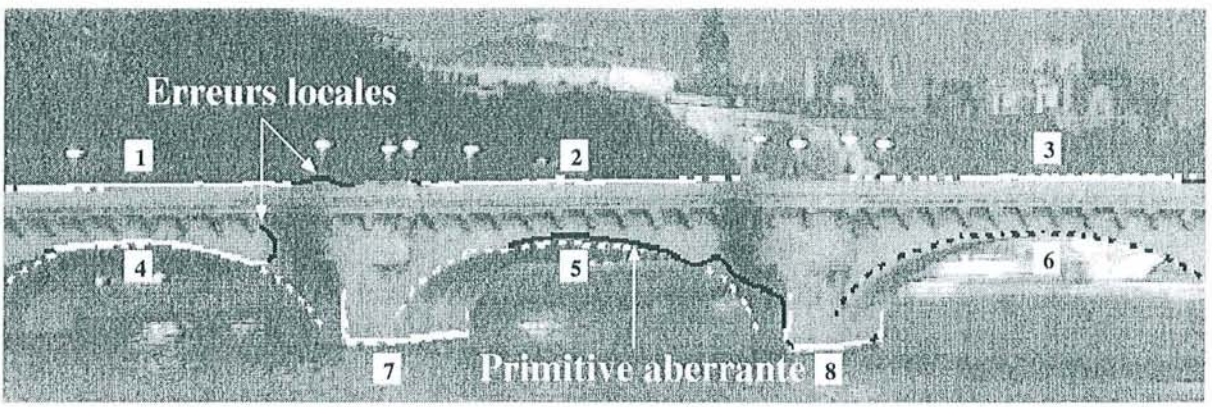


FIG. 5.1 – *Exemple d'erreurs de suivi obtenues sur la séquence du Pont Neuf (une erreur local est obtenue pour les primitives primitives 1 et 4 et une erreur aberrante pour la primitive 5). Les conventions sont les suivantes : lignes blanches continues : primitives images (les sections en noir sont les parties dont l'influence est réduite lors de la minimisation, c'est-à-dire les points pour lesquels le résidu $d_{i,j}$ est supérieur à c, donné en table 4.1) ; lignes blanches discontinues : projection des primitives 3-D correspondantes ; lignes noires continues : primitives aberrantes ; lignes noires discontinues : primitives non initialisées.*

5.1.2 Le niveau local

Nous commençons par calculer pour chaque courbes C_i, une erreur résiduelle r_i qui traduit la distance entre la courbe projetée c_i et la courbe mesurée dans l'image c'_i. Ce résidu est obtenu par une fonction robuste des distances $\{d_{i,j}\}_{1 \leq j \leq l'_i}$, afin de minimiser l'influence des erreurs locales. Différents estimateurs peuvent être utilisés ici :

$$r_i^2 = \frac{1}{l'_i} \sum_{j=1}^{l'_i} \rho_1(d_{i,j}) \tag{5.2}$$

pour un M-estimateur ou

$$r_i^2 = \frac{1}{h} \sum_{j=1}^{h} d_{i,o(j)}^2,$$

où $h = l'_i - [l'_i/2]$, pour l'estimateur LTS.

Le choix de l'estimateur utilisé est primordial. Par définition, un estimateur de type médian favorise les points images retenus (résidus inférieurs à la médiane) qui s'ajustent parfaitement au modèle, au détriment des points non retenus (un exemple obtenu sur la séquence du Pont Neuf est visible en figure 5.13.a). Parmi les M-estimateurs présentés en table 4.1, certains sont plus restrictifs que d'autres : alors que l'influence de la fonction de Tukey est nulle pour les résidus supérieurs à c, celle de Cauchy reste supérieure à zéro tout en décroissant, et celle de Huber reste constante égale à c. Un estimateur peu restrictif comme celui basé sur la fonction de Huber permet d'éviter des problèmes analogues à ceux observés pour les estimateurs médians.

Approximation des distances $d_{i,j}$

La fonction *Dist*, qui approxime la distance euclidienne $d_{i,j}$ du point $m'_{i,j}$ à la courbe c_i, suit l'algorithme suivant (voir la figure 5.2).

Une première approximation de $d_{i,j}$ est obtenue en prenant la distance euclidienne de $m'_{i,j}$ à son point le plus proche sur c_i :

$$d^0_{i,j} = \min_{1 \leq k \leq l_i} ||m'_{i,j} - m_{i,k}||. \tag{5.3}$$

Soit k_0 l'indice pour lequel le minimum de l'équation (5.3) est atteint. Si $k_0 = 1$ ou $k_0 = l_i$, on prend $d_{i,j}$ égal à $d^0_{i,j}$, sinon on affine $d^0_{i,j}$ comme suit : nommons A, B, C les points respectifs m_{i,k_0-1}, m_{i,k_0}, m_{i,k_0+1}. Soit θ l'angle entre les vecteurs $\overrightarrow{AB}$ et $\overrightarrow{AC}$ et ϵ un réel positif arbitrairement petit.

- si $|\sin\theta| \leq \epsilon$ (points A, B, C alignés), on prend $d_{i,j}$ égal à la distance euclidienne de $m'_{i,j}$ à la droite (AC);
- si $|\sin\theta| > \epsilon$, on prend $d_{i,j}$ égal à la distance euclidienne de $m'_{i,j}$ au cercle passant par les points A, B et C.

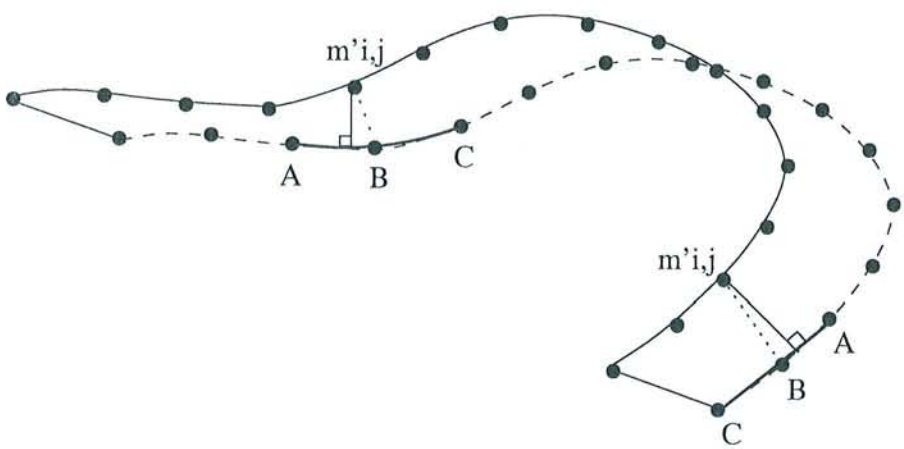


FIG. 5.2 – *Approximation de la distance d'un point à une courbe. La distance est d'abord approximée par la longueur du segment discontinu, puis affiné par celle du segment continu.*

Remarque : cette distance est non symétrique. Nous avons choisi de calculer les distances des points de la courbe c'_i mesurée dans l'image à la courbe projetée c_i, et non l'inverse. Ainsi, les courbes $c'_i \subseteq c_i$ sont considérées comme correctes, alors que les courbes $c'_i \supset c_i$ obtiennent des résidus élevés aux extrémités. Nous verrons que dans le contexte qui nous intéresse, le cas $c'_i \subseteq c_i$ est très fréquent (par exemple lorsque la courbe est partiellement occultée), alors que le cas $c'_i \supset c_i$ est rendu très improbable par la technique mise en œuvre pour l'initialisation des courbes.

5.1.3 Le niveau global

De la même manière qu'au chapitre 4, la minimisation d'une fonction robuste des résidus r_i va nous permettre de réduire l'influence des primitives aberrantes, c'est-à-dire les contours qui sont complètement faux, ou qui contiennent une trop grande proportion de points erronés. Là encore, le choix de l'estimateur est crucial. Utiliser l'estimateur LTS peut s'avérer nécessaire si la segmentation est très mauvaise et donne lieu à plus de 20% de primitives aberrantes. Cependant, comme nous l'avons constaté expérimentalement, ce choix peut facilement conduire à un minimum local (voir l'exemple de la figure 5.13.b où l'information de profondeur n'est

pas prise en compte). Pour cette raison, nous préférons utiliser le M-estimateur associé à la distribution de Tukey, qui est suffisamment restrictif pour supprimer l'influence des primitives aberrantes, mais prend toutes les données en considération.

En conclusion, nous minimisons la fonction

$$f(\mathbf{p}) = \sum_{i=1}^{n} \rho_2(r_i), \tag{5.4}$$

où r_i est donné par l'équation (5.2) (nous utilisons la méthode de convergence quadratique de Powell pour l'optimisation). Les fonctions ρ_1 et ρ_2 qui se sont avérées les plus fiables dans nos expérimentations sont la fonction de Huber pour ρ_1 et celle de Tukey pour ρ_2. Les constantes k pour le seuil $c = kS$ (table 4.1) sont fixées à 2 pour Huber et 4 pour Tukey. Des tests synthétiques plus formels concernant le choix des estimateurs sont présentés plus loin.

Par la suite, nous appellerons *R-pdv* cette procédure à deux niveaux.

5.1.4 Élimination des primitives aberrantes

Lorsqu'une première estimation du point de vue est obtenue par la méthode précédente, nous pouvons très facilement détecter les primitives aberrantes : l'utilisation d'un estimateur robuste au niveau global conduit en effet à une estimation faiblement influencée par ces primitives. Un simple test de comparaison des résidus avec leur écart-type robuste est donc suffisant :

$$\text{si } r_i > 2.5\,\hat{\sigma}\text{ , alors la primitive est éliminée}$$

($\hat{\sigma}$ est donné par l'équation (4.8)). Cette étape est particulièrement intéressante, puisqu'elle nous permet de savoir si une primitive image est conforme au modèle, ce qui sera utilisé par la suite pour la mise à jour automatique des primitives.

Nous pouvons ensuite affiner le point de vue en le recalculant à partir des seules courbes conservées (une estimation au sens des moindres carrés classique est alors utilisée au niveau global).

Nous appellerons *R-pdv+* l'algorithme *R-pdv* suivi de cette dernière étape.

5.2 Justification du choix des estimateurs

Cette section présente les résultats que nous avons obtenus sur des données synthétiques, avec différents estimateurs. Nous verrons que ces tests plus formels sont en accord avec ce que nous avions observé expérimentalement.

5.2.1 Génération des données synthétiques

La génération aléatoire de courbes synthétiques n'est pas un problème simple : celles-ci doivent être à la fois suffisamment génériques pour représenter toutes les primitives susceptibles d'être extraites d'un modèle, et les plus proches possibles de ce que nous pouvons obtenir sur des séquences réelles, notamment en ce qui concerne le bruit. La génération d'une scène complètement aléatoire n'est pas envisageable et ne serait d'ailleurs pas réaliste. Nous avons dû imposer certaines contraintes (dimensions des primitives par rapport à la scène, position de la caméra, type de bruit obtenu ...), qui sont particulièrement proches de ce que nous avons observé dans nos expérimentation (principalement basées sur l'observation relativement éloignée d'édifices de taille importante).

Construction de la scène

Les tests présentés plus loin ont été réalisés de la façon suivante :

1. n courbes 3-D aléatoires $\{C_i\}_{1 \leq i \leq n}$ sont générées à l'intérieur d'un cube (volume de la scène) dont le centre est situé à une distance d du centre optique de la caméra. Les arêtes de ce cube ont pour longueur L et sont parallèles aux axes de la caméra (voir la figure 5.3). Les tests présentés ont été obtenus avec $d = 200$ et $L = 100$.
2. Des erreurs locales ou grossières sont générées au niveau de la projection des courbes dans le plan image.
3. L'estimée initiale $\mathbf{p_0}$ est obtenue à partir du point de vue $(\mathbf{R} \quad \mathbf{t})$ attendu, en ajoutant à $\mathbf{t}$ un vecteur de direction aléatoire et de norme inférieure 7 pour $d = 200$, et en ajoutant aux angles d'Euler de $\mathbf{R}$ un angle de valeur absolue inférieure à $\pi/6$ (cette initialisation est généralement plus défavorable que dans le cas d'une séquence réelle).
4. L'optimisation à deux niveaux utilisant les estimateurs testés est alors effectuée à partir de la position initiale $\mathbf{p_0}$.
5. la différence entre le point de vue obtenu et le point de vue attendu est mesurée : $||\Delta\mathbf{t}||$ (différence entre les translations), $|\Delta\alpha|$, $|\Delta\beta|$ et $|\Delta\gamma|$ (différences entre les angles d'Euler) sont ainsi calculées.

Les étapes 1 à 5 sont répétées 50 fois et les erreurs moyennes sont calculées.

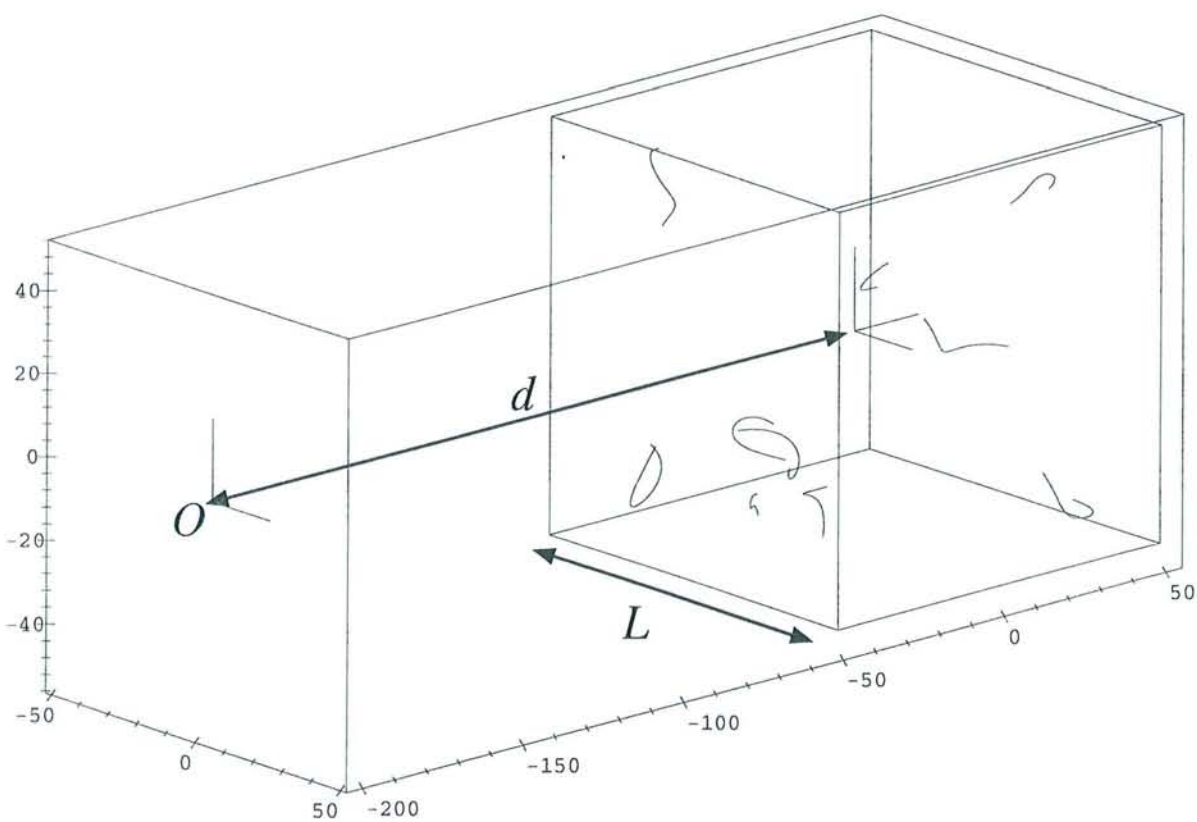


FIG. 5.3 – *n courbes 3-D générées aléatoirement dans un cube de côté L, dont le centre est situé à une distance d du centre optique de la caméra ($n = 10; L = 100; d = 200$).*

Génération d'une courbe 3-D

Une courbe 3-D C_i est synthétisée de la façon suivante : un cube c est positionné aléatoirement à l'intérieur du volume de la scène C. m points 3-D $\{P_i\}_{0 \leq i < m}$ sont alors générés à l'intérieur de c et utilisés comme points de contrôle pour une courbe de Bézier :

$$P(t) = \sum_{i=0}^{m-1} C_{m-1}^{i} t^i (1-t)^{m-1-i} P_i.$$

Pour nos tests, $m = 5$ et la longueur des côtés de c est prise égale à 20.

Génération du bruit

Une fois que les courbes 3-D sont générées, les courbes 2-D $\{c_i'\}_{1 \leq i \leq n}$ sont obtenues en introduisant du bruit sur les projections $\{c_i\}_{1 \leq i \leq n}$ des courbes 3-D (projetées à partir du point de vue attendu). Deux sortes d'erreurs doivent être synthétisées : des erreurs grossières qui génèrent des primitives aberrantes et des erreurs locales déformant la courbe sur une proportion x de leur longueur. Une erreur grossière peut être obtenue très simplement en translatant c_i d'un vecteur aléatoire. Pour les erreurs locales le problème est plus complexe : le résultat doit ressembler à ce qui est obtenu lorsqu'un contour est suivi (et localement attiré par un gradient fort d'intensités), et nous devons de plus être en mesure de contrôler le taux de bruit x présent dans la courbe.

Le bruit local est généré de la façon suivante (voir figure 5.4) : une sous-chaîne s de c_i, ayant une longueur égale à x fois la longueur de c_i, est extraite aléatoirement. Le point situé au milieu de s est translaté d'un vecteur aléatoire (dont la norme est proportionnelle à la taille de c_i), et une courbe de Bézier est générée à partir de ce nouvel ensemble de points s. Nous exploitons ainsi la propriété d'une courbe de Bézier à être modifiée globalement lorsqu'une variation locale se produit : déplacer le point central de s a une influence sur toute la courbe s, alors que l'utilisation d'une B-Spline par exemple n'aurait modifié que localement s. Finalement, s est remplacé par la courbe de Bézier et c_i' est la courbe c_i ainsi modifiée. Les courbes obtenues par cette méthode sont généralement proches de ce que nous obtenons sur des séquence réelles.

5.2.2 Résultats

Ce procédé nous a permis de tester la procédure *R-pdv* pour différents estimateurs, utilisés au niveau global, puis local. L'intérêt de la méthode à deux niveaux par rapport à la méthode à un seul niveau est ensuite illustré par un exemple.

Le niveau global

La figure 5.5 montre l'influence des erreurs aberrantes sur le point de vue estimé pour différents estimateurs (le nombre n de primitives utilisées est égal à 10). Les estimateurs comparés sont LS, LTS, Tukey et Huber, appliqués au niveau global (comme aucun bruit local n'est généré, l'estimateur LS est utilisé au niveau local). On constate que les M-estimateurs obtiennent les meilleurs résultats. La fonction de Tukey, plus restrictive que celle de Huber, obtient aussi des résultats plus précis. Nous vérifions d'autre part qu'à partir de 20% de données aberrantes, les M-estimateurs sont moins performants. L'estimateur LTS obtient les moins bons résultats (cet estimateur s'avère peu précis et plutôt instable), pour la raison déjà évoquée de prise en compte partielle des données, qui conduit fréquemment à un minimum local.

Ces résultats sont illustrés en figure 5.6 : 30% d'erreurs aberrantes sont introduites sur un ensemble de 10 courbes (les courbes 5, 7 et 9 sont aberrantes). Les courbes noires correspondent

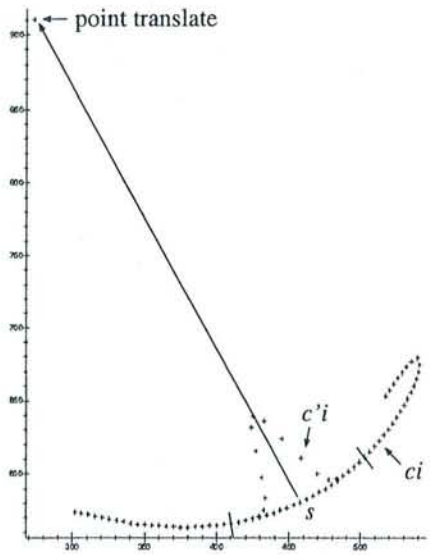


FIG. 5.4 – *Génération aléatoire d'une erreur locale ($x = 30\%$).*

aux primitives 2-D et les grises à la projection des primitives 3-D correspondantes. Comme l'estimateur LS force l'ajustement de toutes les données, l'erreur de reprojection est répartie sur l'ensemble des primitives. A l'opposé, l'estimateur LTS ne minimise les résidus que de la moitié des primitives (courbes 1, 2, 3, 6 et 7), ce qui conduit à un minimum local où les primitives correctes 4, 8 et 10 ne sont pas ajustées. Pour Tukey et Huber, les primitives aberrantes n'ont quasiment aucune influence sur l'optimisation, et les autres primitives sont bien ajustées.

Le niveau local

La figure 5.7 montre l'influence des erreurs locales pour les différents estimateurs. Comme aucune erreur aberrante n'est introduite, l'estimateur LS est utilisé au niveau global. Là encore, les M-estimateurs obtiennent de meilleurs résultats, mais il n'y a pas de réelle différence entre Tukey et Huber. Cette différence apparaît pour un nombre plus faible de primitives (typiquement jusqu'à 5 ou 6 primitives) : la figure 5.8 montre un exemple de résultats obtenus avec 3 courbes. La ligne (a) montre les résultats obtenus sans qu'aucune erreur ne soit introduite, et la ligne (b) les résultats obtenus avec 30% d'erreurs locales sur chaque primitive. Dans les deux cas, le même phénomène est observé : Tukey et LTS conduisent à un minimum local où pour chaque courbe, une partie seulement de la primitive projetée est ajustée à la primitive image correspondante. L'utilisation de la fonction de Huber conduit à de bien meilleurs résultats en raison de sa distribution plus permissive. La table 5.1, qui indique l'erreur moyenne obtenue pour 50 générations aléatoires de 3 courbes, confirme ces résultats.

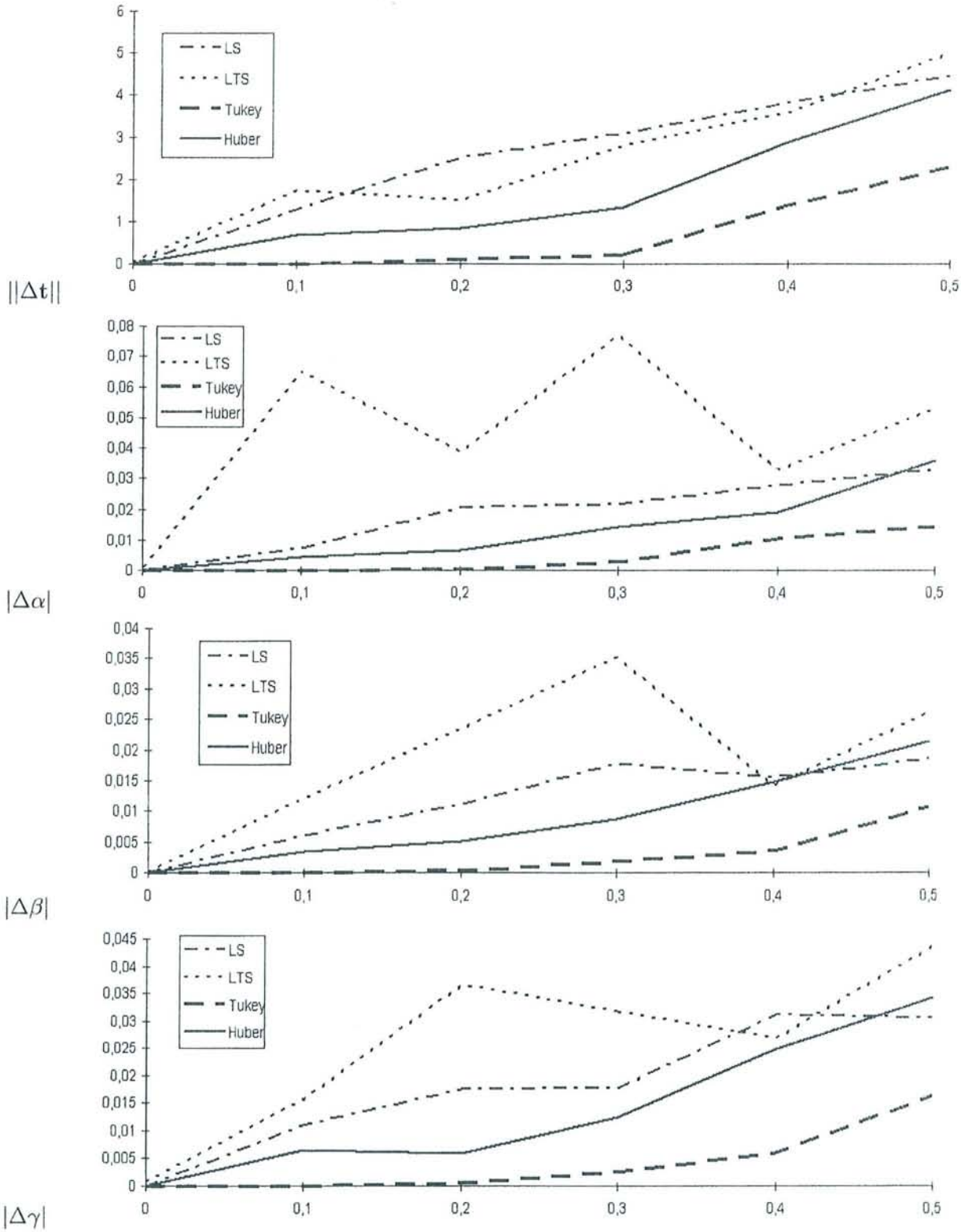
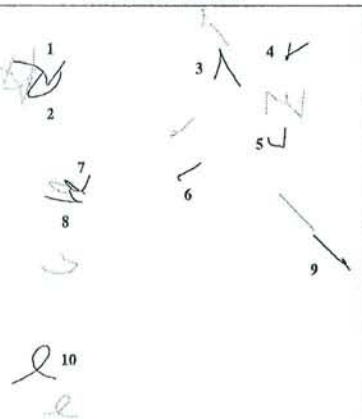
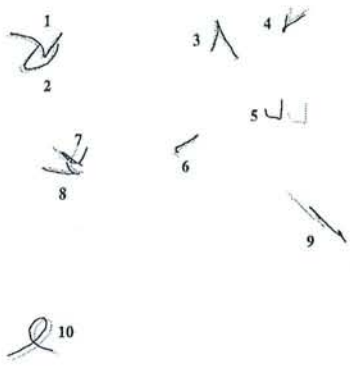
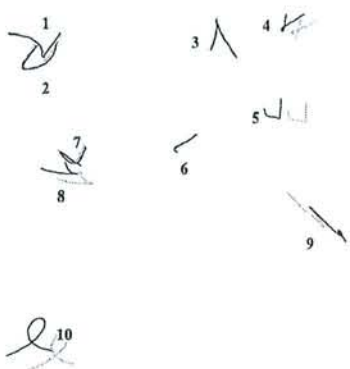


FIG. 5.5 – *Influence des erreurs aberrantes sur le calcul du point de vue, en fonction du taux d'erreurs présentes dans les données et pour différents estimateurs utilisés au niveau global* ($n = 10$).

Initialisation	LS	LTS
		
$\|\|\Delta\mathbf{t}\|\| = 5.745015$ $\|\Delta\alpha\| = 0.075261$ $\|\Delta\beta\| = 0.063373$ $\|\Delta\gamma\| = 0.522704$	$\|\|\Delta\mathbf{t}\|\| = 3.856936$ $\|\Delta\alpha\| = 0.025502$ $\|\Delta\beta\| = 0.014650$ $\|\Delta\gamma\| = 0.032381$	$\|\|\Delta\mathbf{t}\|\| = 2.901643$ $\|\Delta\alpha\| = 0.065329$ $\|\Delta\beta\| = 0.012303$ $\|\Delta\gamma\| = 0.029872$

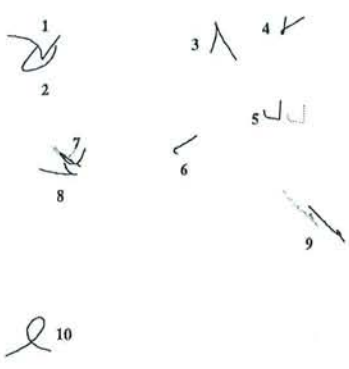
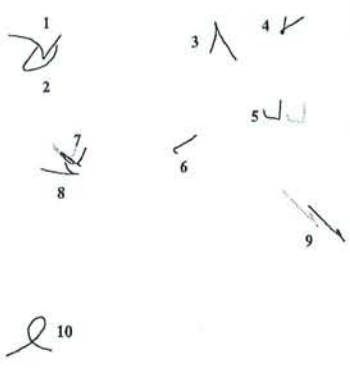
Tukey	Huber
	
$\|\|\Delta\mathbf{t}\|\| = 0.001847$ $\|\Delta\alpha\| = 0.000007$ $\|\Delta\beta\| = 0.000019$ $\|\Delta\gamma\| = 0.000018$	$\|\|\Delta\mathbf{t}\|\| = 0.002263$ $\|\Delta\alpha\| = 0.000012$ $\|\Delta\beta\| = 0.000024$ $\|\Delta\gamma\| = 0.000020$

FIG. 5.6 – *Exemples de résultats obtenus pour 30% d'erreurs aberrantes (courbes 5, 7 et 9) et différents estimateurs utilisés au niveau global. Les courbes noires correspondent aux primitives images et les grises à la projection des primitives 3-D correspondantes.*

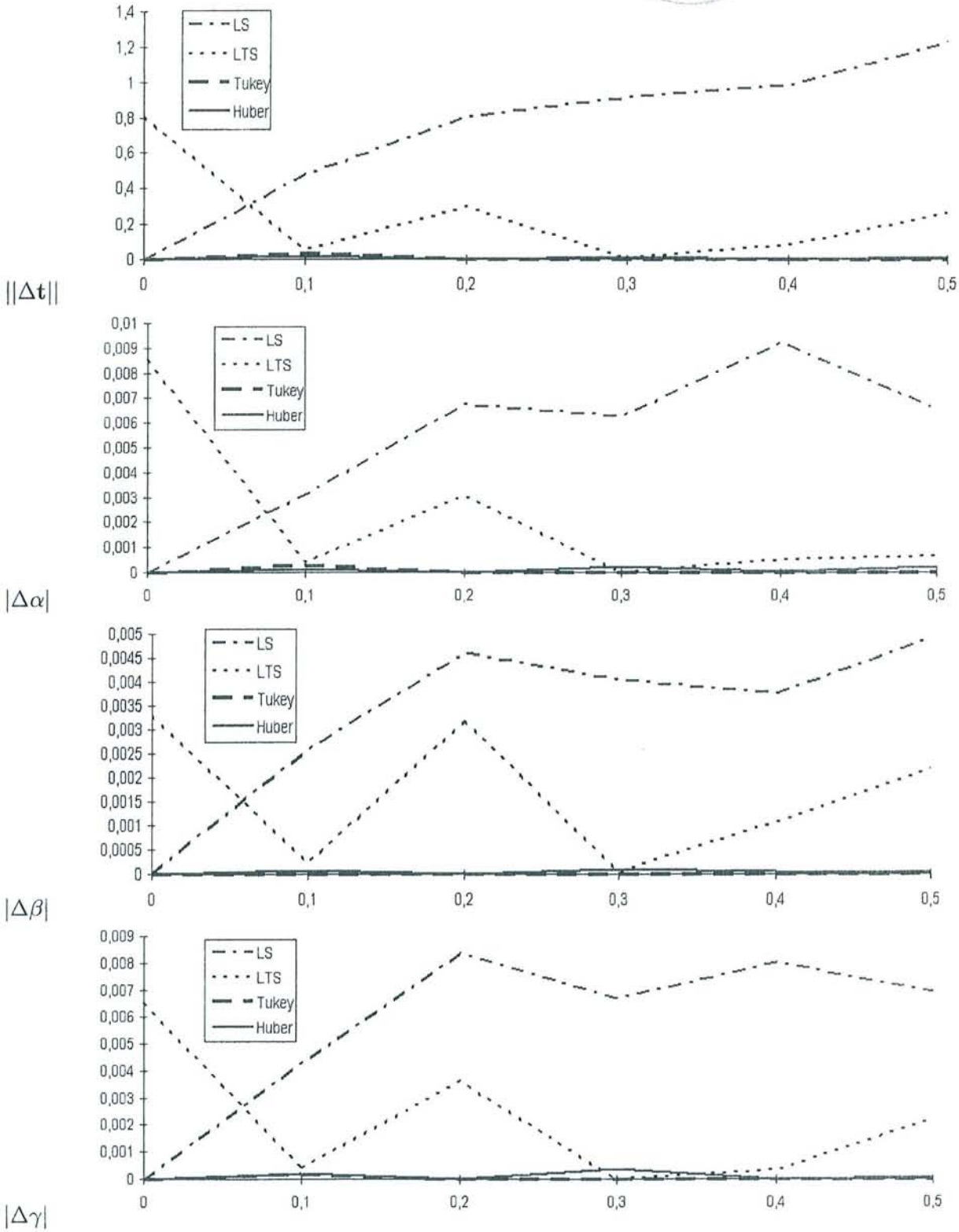


FIG. 5.7 – *Influence des erreurs locales sur l'optimisation, pour différents estimateurs utilisés au niveau local.*

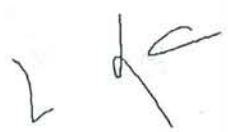
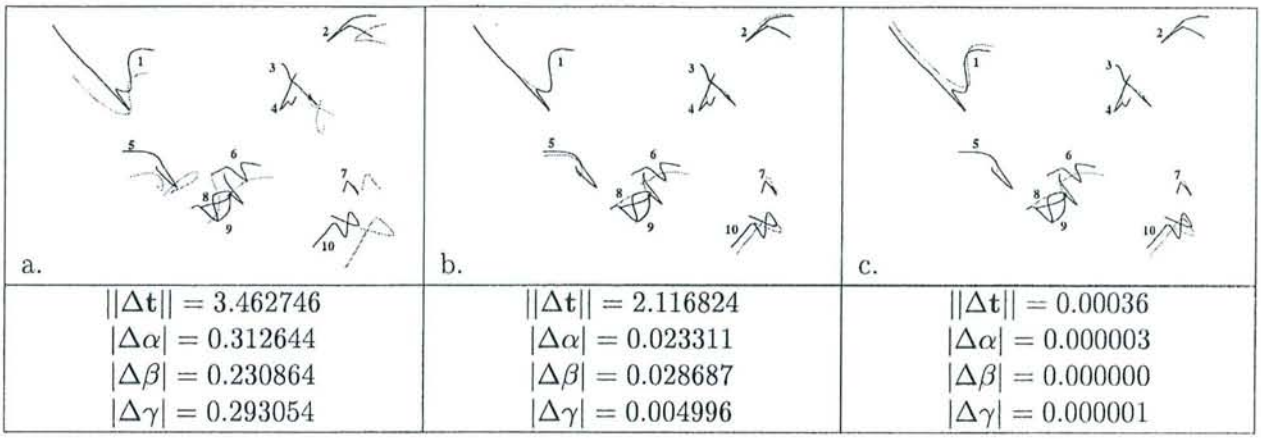
Initialisation	LS	LTS	Tukey	Huber
a. 				
$\|\|\Delta\mathbf{t}\|\| = 6.258518$ $\|\Delta\alpha\| = 0.214981$ $\|\Delta\beta\| = 0.216259$ $\|\Delta\gamma\| = 0.495572$	$\|\|\Delta\mathbf{t}\|\| = 0.008604$ $\|\Delta\alpha\| = 0.000025$ $\|\Delta\beta\| = 0.000017$ $\|\Delta\gamma\| = 0.000015$	$\|\|\Delta\mathbf{t}\|\| = 15.236698$ $\|\Delta\alpha\| = 0.044035$ $\|\Delta\beta\| = 0.003210$ $\|\Delta\gamma\| = 0.019735$	$\|\|\Delta\mathbf{t}\|\| = 15.947834$ $\|\Delta\alpha\| = 0.048148$ $\|\Delta\beta\| = 0.020985$ $\|\Delta\gamma\| = 0.010724$	$\|\|\Delta\mathbf{t}\|\| = 0.000006$ $\|\Delta\alpha\| = 0.000000$ $\|\Delta\beta\| = 0.000000$ $\|\Delta\gamma\| = 0.000000$
b. 				
$\|\|\Delta\mathbf{t}\|\| = 6.258518$ $\|\Delta\alpha\| = 0.214981$ $\|\Delta\beta\| = 0.216259$ $\|\Delta\gamma\| = 0.495572$	$\|\|\Delta\mathbf{t}\|\| = 3.131000$ $\|\Delta\alpha\| = 0.000225$ $\|\Delta\beta\| = 0.002180$ $\|\Delta\gamma\| = 0.004767$	$\|\|\Delta\mathbf{t}\|\| = 8.957683$ $\|\Delta\alpha\| = 0.065539$ $\|\Delta\beta\| = 0.023610$ $\|\Delta\gamma\| = 0.033112$	$\|\|\Delta\mathbf{t}\|\| = 4.925826$ $\|\Delta\alpha\| = 0.048410$ $\|\Delta\beta\| = 0.024341$ $\|\Delta\gamma\| = 0.027564$	$\|\|\Delta\mathbf{t}\|\| = 0.000001$ $\|\Delta\alpha\| = 0.000000$ $\|\Delta\beta\| = 0.000000$ $\|\Delta\gamma\| = 0.000000$

FIG. 5.8 – *Exemple de résultats obtenus pour différents estimateurs utilisés au niveau local. a. aucune erreur n'est introduite. b.* 30% *d'erreurs locales sont introduites sur chaque courbe.*

		LS	LTS	Tukey	Huber
a.	$\|\|\Delta\mathbf{t}\|\|$	2.082570	4.665949	5.507071	1.692501
	$\|\Delta\alpha\|$	0.036419	0.088702	0.110307	0.031864
	$\|\Delta\beta\|$	0.005644	0.038819	0.029227	0.011611
	$\|\Delta\gamma\|$	0.044870	0.067626	0.059164	0.023085
		LS	LTS	Tukey	Huber
b.	$\|\|\Delta\mathbf{t}\|\|$	3.349066	5.512861	3.245572	1.963530
	$\|\Delta\alpha\|$	0.027191	0.091413	0.040580	0.032111
	$\|\Delta\beta\|$	0.014521	0.036188	0.023214	0.011259
	$\|\Delta\gamma\|$	0.021947	0.081896	0.032029	0.012110

TAB. 5.1 – *Erreurs moyennes obtenues sur 50 tests pour différents estimateurs utilisés au niveau local. a. aucune erreur n'est introduite. b.* 30% *d'erreurs locales sont introduites sur chaque courbe.*



a.	b.	c.
$\|\|\Delta\mathbf{t}\|\| = 3.462746$	$\|\|\Delta\mathbf{t}\|\| = 2.116824$	$\|\|\Delta\mathbf{t}\|\| = 0.00036$
$\|\Delta\alpha\| = 0.312644$	$\|\Delta\alpha\| = 0.023311$	$\|\Delta\alpha\| = 0.000003$
$\|\Delta\beta\| = 0.230864$	$\|\Delta\beta\| = 0.028687$	$\|\Delta\beta\| = 0.000000$
$\|\Delta\gamma\| = 0.293054$	$\|\Delta\gamma\| = 0.004996$	$\|\Delta\gamma\| = 0.000001$

FIG. 5.9 – *Exemple de recalage. a. position initiale (les primitives 1, 6 et 10 sont aberrantes). b. Résultat obtenu après l'optimisation à un seul niveau. c. Résultat obtenu après l'optimisation à deux niveaux.*

Minimisation à deux ou un seul niveaux

Les résultats précédents confirment que la distribution Tukey, pour le niveau global, et celle de Huber, pour le niveau local, sont les distributions les plus pertinentes du point de vue de la précision sur des données faiblement bruitées (jusqu'à environ 30% de bruit). La figure 5.9.c montre un résultat obtenu avec 10 courbes par la procédure *R-pdv* utilisant ces estimateurs (la position initiale est montrée en figure 5.9.a). 30% d'erreurs aberrantes (courbes 1, 6 et 10) et 30% d'erreurs locales sur chaque courbe ont été générées. Les primitives aberrantes sont écartées et les sections localement erronées n'ont pas influencé l'optimisation. La figure 5.9.b montre le résultat obtenu par la méthode à un seul niveau (minimisation de la fonction (5.1) avec la fonction de Tukey) : le résultat est moins correct, principalement parce que la primitive aberrante 1, qui est plus longue que les autres, a aussi une plus grande influence dans l'optimisation (dans cet exemple les primitives ont une taille aléatoire comprise en 20 et 50).

5.3 La boucle de recalage temporel, version 2

Nous pouvons maintenant décrire la deuxième version de notre boucle de recalage temporel, en détaillant chacune des étapes du schéma 3.2.

5.3.1 Suivi des primitives

Soit $\mathcal{E}$ l'ensemble des primitives 2-D de l'image courante. Ces primitives sont suivies en utilisant la méthode de [Berger94], déjà évoquée au chapitre 4, que nous détaillons à présent.

Plutôt que de détecter des contours courbes dans deux images, et de les apparier ensuite, nous utilisons une méthode globale basée sur les contours actifs [Kass et al.88], qui permet de résoudre simultanément le problème de la segmentation et le problème du suivi. Une fois que le contour est initialisé dans la première image, il peut être suivi d'image en image sans qu'il n'y ait de nouvelle détection. Cependant, en raison de la flexibilité excessive des contours actifs, cette approche est limitée au cas où le déplacement et la déformation du contour dans l'image sont faibles.

Pour pallier ce problème, Berger considère un processus de suivi en deux étapes. L'étape de prédiction calcule une estimation grossière du champ des vitesses. À partir de cette initialisation, le contour converge généralement vers le contours correspondant. Comme les contours actifs sont déformables, une modélisation approximative du champ des vitesses 2-D est généralement suffisante.

L'algorithme est basé sur le calcul itératif du champ des vitesses 2-D. L'objectif est de déterminer le meilleur déplacement 2-D rigide $\mathcal{D}$ tel que la distribution des intensités sur le contour suivi c'_i et la distribution sur le contour $\mathcal{D}(c'_i)$ soient grossièrement les mêmes. La méthode utilise uniquement sur le flux optique normal, qui peut être obtenu de façon suffisamment précise.

En reprenant les notations indiquées en début de chapitre, $\{m'_{i,j}\}_{1\leq j\leq l'_i}$ est la discrétisation de la courbe c'_i. Soit $\mathbf{f}_0^{\perp}(m'_{i,j})$ le flux optique normal au point $m'_{i,j}$ et $\mathbf{n}_{i,j}$ la normale unitaire à c'_i au point $m'_{i,j}$. Soit I_1 et I_2 deux images consécutives. Le déplacement 2-D rigide $\mathcal{D}_0$ minimisant

$$\sum_{1\leq j\leq l'_i} |(\overrightarrow{m'_{i,j}\mathcal{D}_0(m'_{i,j})}.\mathbf{n}_{i,j})\mathbf{n}_{i,j} - \mathbf{f}_0^{\perp}(m'_{i,j})|^2 \tag{5.5}$$

donne une approximation grossière du déplacement. Cette approximation est ensuite affinée en calculant le flux optique normal $\mathbf{f}_1^{\perp}$ sur $\mathcal{D}_0(c'_i)$ entre l'image recalée $I_1(\mathcal{D}_0^{-1}(x,y))$ et I_2, etc. Des déplacements infinitésimaux $\mathcal{D}_0$,...,$\mathcal{D}_j$,... peuvent ainsi être calculés, qui rapprochent progressivement la courbe $\mathcal{D}_j \circ ... \circ \mathcal{D}_0(c'_i)$ de la courbe correspondante dans l'image I_2.

Techniquement, si $\mathbf{t}_j$ est la composante translationnelle du déplacement $\mathcal{D}_j$, et si l'angle de rotation θ_j est suffisamment faible, $\mathcal{D}_j$ peut être approximé par

$$\mathcal{D}_j \begin{cases} x - \theta_j y + t_j^x \\ \theta_j x + y + t_j^y \end{cases}$$

et une solution à (5.5) peut être obtenue par les moindres carrés. Pour une rotation importante, cette approximation peut servir d'initialisation à une minimisation itérative.

L'utilisation d'un schéma itératif permet de considérer des déplacements importants (jusqu'à 20 pixels environ), à condition que les contours homologues se recouvrent suffisamment. Cependant, le processus de suivi peut échouer dans certains cas. En effet, l'étape de prédiction peut diverger si le calcul du flux optique normal est incorrect sur une grande partie du contour. D'autres problèmes peuvent survenir avec les contours actifs, notamment si le contour est attiré par des gradients forts qui n'appartiennent pas au contour recherché.

5.3.2 Recalage

Le recalage est effectué en utilisant la procédure *R-pdv+* décrite précédemment, à partir de l'ensemble $\mathcal{E}$ des primitives suivies. L'estimée initiale du point de vue $\mathbf{p_0}$ est le point de vue obtenu dans l'image précédente de la séquence. Comme nous l'avons vu, l'utilisation d'estimateurs robustes à deux niveaux de l'optimisation permet de prendre en compte des erreurs de suivi, aussi bien que des occultations partielles ou totales de certaines primitives. D'autre part, les primitives aberrantes de $\mathcal{E}$ sont détectées et éliminées.

5.3.3 Mise à jour des primitives

Deux types d'événements distincts peuvent nécessiter la mise à jour de l'ensemble $\mathcal{E}$ des primitives suivies :

- **entrée-sortie d'une primitive :** au cours de la séquence, des primitives nouvelles peuvent apparaître dans le champ de vision et d'autres peuvent en sortir. A partir de la matrice de

projection perspective qui est obtenue en même temps que le point de vue et des dimensions en pixels de l'image, nous déterminons si une primitive se projette ou non dans l'image. Nous appellerons *primitive visible* une primitive qui se projette dans l'image, même si cette primitive est occultée. Les primitives qui ne sont plus visibles sont alors supprimées de $\mathcal{E}$. Par contre, les primitives qui viennent d'apparaître doivent être *initialisées* dès que possible, c'est-à-dire que les primitives 2-D correspondantes doivent être recherchées dans l'image et insérées dans $\mathcal{E}$.

- **détection d'une primitive aberrante :** lorsqu'un contour aberrant est détecté, celui-ci ne peut plus être suivi. Il faut donc tenter de *réinitialiser* la primitive, c'est-à-dire de retrouver le contour correct qui sera suivi par la suite.

Le fait d'utiliser des courbes de forme libre et une méthode de recalage robuste nous permet de rechercher automatiquement les correspondants 2-D des primitives à initialiser ou à réinitialiser, en se basant sur la carte des contours.

Initialisation d'une primitive

L'initialisation d'une primitive du modèle consiste à rechercher son homologue 2-D dans la carte des contours de l'image. Une première approche consisterait à choisir le contour le plus proche de la projection de la primitive 3-D considérée (en utilisant la distance robuste décrite précédemment). Malheureusement, la complexité de la segmentation est généralement telle que le contour le plus proche n'est pas toujours celui qui convient (voir par exemple la figure 5.10.1). On pourrait envisager une comparaison morphologique entre le contour projeté et les contours détectés autour de la projection (courbure, torsion ...), mais une telle comparaison est sensible au bruit. D'autre part, il arrive fréquemment que la primitive recherchée soit la somme de plusieurs contours détectés, ou au contraire une partie d'un contour.

Nous avons donc opté pour une approche originale, basée sur l'aptitude de la procédure *R-pdv* à détecter les primitives incorrectes. L'algorithme mis en oeuvre est le suivant (voir la figure 5.10) :

1. une détection de contours est effectuée dans l'image par la méthode classique de Canny/Dériche [Deriche87];
2. on considère les contours qui sont suffisamment proches de la projection de la primitive 3-D (par exemple les 6 premiers contours triés selon la taille croissante du résidu robuste) ;
3. pour chacun de ces contours c, on calcule le point de vue en utilisant la procédure *R-pdv+* à partir des primitives $\mathcal{E} \cup \{c\}$ ($\mathcal{E}$ étant toujours l'ensemble des primitives déjà appariées dans l'image). La procédure *R-pdv+* nous indique alors si le contour c est aberrant ou non. Si c est conservé, les sections erronées de ce contour sont éliminées (en comparant les distances $d_{i,j}$ à leur écart-type) ;
4. si aucun contour n'est retenu, l'initialisation échoue, sinon on essaye de concaténer les sections conservées par un algorithme récursif (deux chaînes sont concaténées si la distance entre deux extrêmités est inférieure à un certain seuil). Nous obtenons alors un ensemble de courbes candidates dont certaines peuvent être l'union de plusieurs contours. La courbe la plus longue de cet ensemble est retenue comme étant la primitive recherchée.

Cette méthode permet en général d'obtenir le bon correspondant image, même si celui-ci était divisé en plusieurs contours. Remarquons que chaque primitive est intégrée au fur et à mesure à l'ensemble $\mathcal{E}$, ce qui modifie à chaque fois le résultat du recalage.

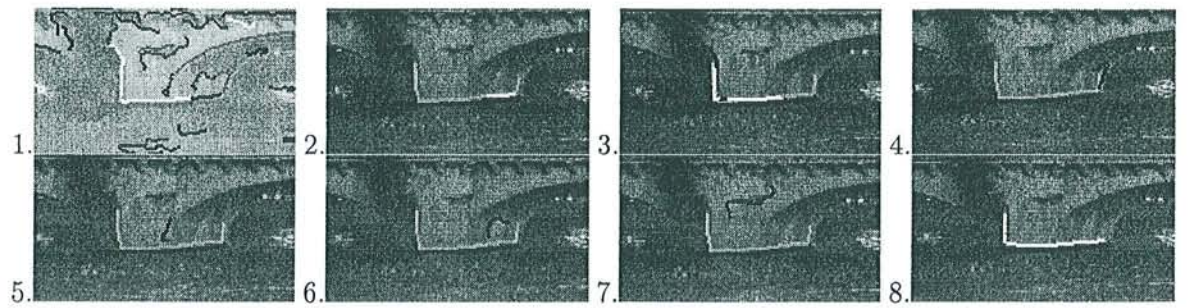


FIG. 5.10 – *Exemple d'initialisation dans la séquence du Pont Neuf. 1 : carte des contours autour de la primitive projetée. 2 à 7 : test de rejet par la procédure R-courbe+ pour les 6 contours les plus proches de la projection. La projection de la primitive 3-D est dessinée en gris, les sections de contours conservées apparaissent en blanc et les sections rejetées ainsi que les contours globalement rejetés sont en noir. 8 : résultat final : la primitive retenue est la réunion des sections conservées.*

5.3.4 Initialisation

L'étape d'initialisation des paramètres intrinsèques et extrinsèques de la caméra est identique à celle de la première version de notre système : un certain nombre d'appariements 3-D/2-D de points (4 au minimum) sont utilisés pour calculer le point de vue dans la première image par la méthode de Dementhon et Davis. Les paramètres intrinsèques de la caméra peuvent être obtenus à l'aide d'une mire de calibration ou directement en utilisant les points appariés s'ils sont en nombre suffisant. L'étape suivante consiste à déterminer les primitives du modèle qui permettront d'assurer le recalage sur toute la séquence, puis les correspondants 2-D des primitives visibles dans la première image, initialisant l'ensemble $\mathcal{E}$ des primitives suivies.

Détermination des primitives de référence

Dans le système actuel, les primitives de référence sont choisies par l'opérateur selon deux critères :

- **un critère d'apparence :** comme les correspondants image de ces primitives sont recherchés dans la carte des contours de l'image, on choisit des primitives susceptibles de donner lieu à des contours forts (arêtes, limites entre zones de couleurs contrastées ...).
- **un critère de distribution :** les primitives doivent être suffisamment nombreuses et présentes à divers endroits de la scène pour permettre de maintenir le recalage sur toute la séquence. D'autre part, une bonne distribution spatiale permet d'obtenir de meilleurs résultats sur le calcul du point de vue (notamment, lorsqu'une direction spatiale est faiblement représentée, la projection des primitives éloignées des autres primitives de référence dans cette direction est très incertaine). Notons que pour améliorer les performances de la procédure *R-pdv* sur des distributions spatiales déséquilibrées, nous pouvons dans un premier temps pondérer les résidus de l'optimisation (5.4) par un coefficient w_i : pour chaque courbe i, on calcule le barycentre G_i de la chaîne de points 3-D, et on prend w_i égal à la distance entre G_i et le barycentre de l'ensemble des points $\{G_i\}_{1 \leq i \leq n}$.

Détermination des correspondants 2-D

Comme le point de vue de la première image est connu, nous pouvons déterminer les primitives visibles dans cette image. L'initialisation de ces primitives se fait alors de façon automatique en utilisant la méthode d'initialisation décrite précédemment, où $\mathcal{E}$ est initialement l'ensemble des points appariés par l'opérateur. Ces points sont ensuite supprimés de l'ensemble $\mathcal{E}$ et seuls les contours sont suivis.

5.4 Résultats expérimentaux

La séquence du Pont-Neuf ayant été obtenue à partir d'un mouvement panoramique de la caméra, elle ne permet pas d'illustrer complètement la capacité de notre système à prendre en compte des mouvements de caméra quelconques. Toutefois, il est intéressant de constater que, partant de l'hypothèse d'un mouvement quelconque, on retrouve bien une translation fixe et un mouvement panoramique de la caméra. D'autre part, il s'agit d'une séquence longue et particulièrement bruitée qui permet donc de mettre en évidence l'autonomie du système grâce à la mise à jour automatique des primitives utilisées, ainsi que sa robustesse.

Afin de valider notre système sur un mouvement quelconque, nous avons aussi filmé un château miniature à partir d'une caméra tenue à la main par une personne en mouvement. La séquence obtenue est donc particulièrement saccadée.

5.4.1 La séquence du Pont Neuf

Initialisation

L'étape d'initialisation est présentée en figure 5.11. Les conventions sont les mêmes que pour la figure 5.1. La figure 5.11.a montre la projection des primitives visibles dans la première image. Le point de vue a été obtenu par la méthode de Dementhon et Davis à partir des quatre points désignés par des croix. La figure 5.11.b montre le résultat de l'initialisation de ces primitives (les deux primitives de droite ne sont pas initialisées car elles n'apparaissent pas entièrement dans l'image).

Itération dans l'image 13

La figure 5.12 montre les résultats des différentes étapes de la boucle de recalage obtenus dans l'image 13 de la séquence du Pont Neuf :

- figure 5.12.a : appariements 3-D/2-D obtenus dans l'image 12. On constate que la primitive 4 n'est pas entièrement correcte (ceci est principalement dû au fait que le contour actif à été attiré par des gradients forts). Cependant, comme une partie de la courbe correspond au modèle 3-D, la primitive est globalement retenue ;
- figure 5.12.b : suivi dans l'image 13. Les primitives sont correctement suivies, exceptée la primitive 5. L'erreur obtenue sur cette primitive est due à un échec de l'étape de prédiction en raison du bruit présent dans l'image.
- figure 5.12.c : recalage. Le point de vue est correctement calculé, et la primitive 5, qui a faiblement influencé l'optimisation, est éliminée. La figure 5.13 montre ce que nous obtenons en utilisant un estimateur médian au niveau local ou global : dans les deux cas, un minimum local est atteint.
- figure 5.12.d : mise à jour des primitives. La primitive 5 est réinitialisée.

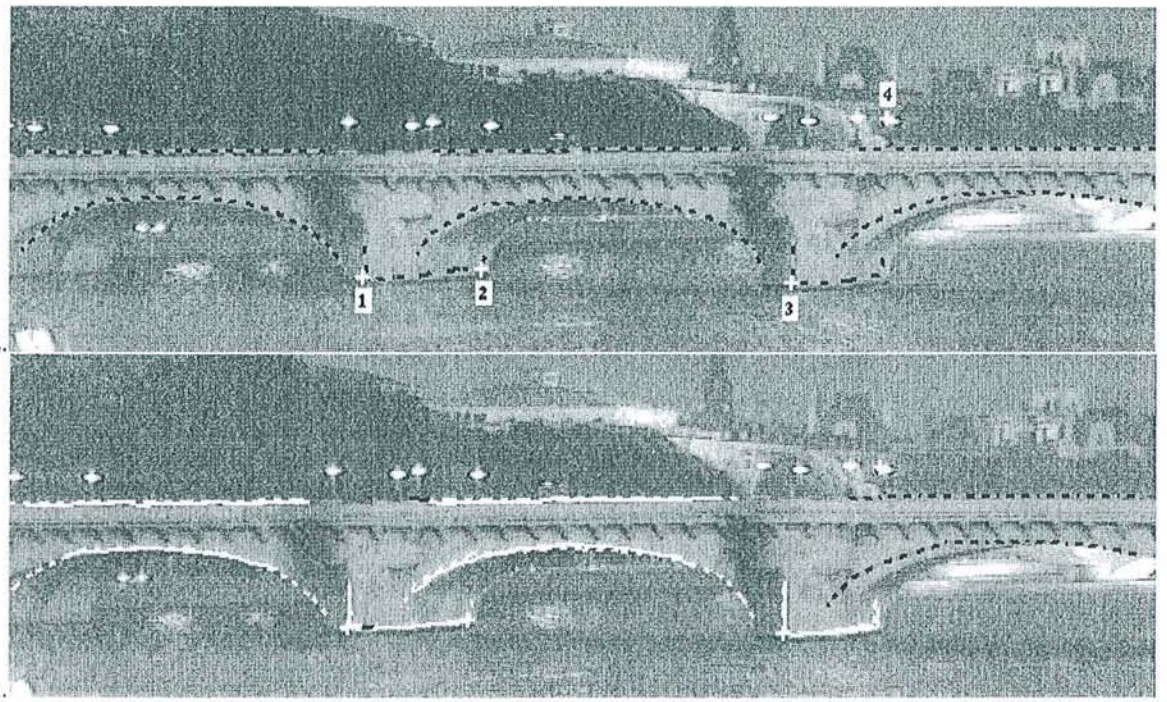


FIG. 5.11 – *Étape d'initialisation pour la séquence du Pont Neuf. a. Initialisation du point de vue par la méthode de Dementhon et Davis à partir des quatre points désignés par des croix. b. Initialisation automatique des primitives à suivre.*

La projection du modèle filaire obtenue à partir du nouvel ensemble de primitives apparaît en figure 5.16. On constate que cette projection est assez proche de l'objet réel.

Résultats obtenus sur la séquence

Une autre façon d'estimer la qualité de l'algorithme est d'estimer sa reproductibilité sur une longue séquence d'images. Les paramètres extrinsèques de la caméra (translation et angles d'Euler exprimés dans le repère du pont - voir la figure 4.4 pour les axes) sont tracés en figure 5.15.b. On remarque tout d'abord que ces paramètres dégénèrent à partir de l'image 164. Pour expliquer ceci, voyons le résultat du suivi dans cette image (figure 5.14) : nous constatons que parmi les primitives apportant de l'information de profondeur, seule la pile du milieu est encore partiellement suivie. Pour poursuivre la séquence, nous aurions donc dû réinitialiser à la main les piles visibles du pont, ce qui va à l'encontre de notre objectif d'autonomie. Nous voyons donc apparaître les limites de cette méthode : lorsque les primitives suivies ne sont pas correctement distribuées spatialement (primitives coplanaires, taille du modèle petite par rapport à la distance modèle-caméra ...), le point de vue obtenu peut être très imprécis. Nous reparlerons de ce problème à la fin de ce chapitre et consacrerons le chapitre suivant à sa résolution.

Commentons à présent les résultats obtenus pour les images 0 à 163, pour lesquelles un minimum d'information de profondeur apparaît.

La courbe continue de la figure 5.15.a montre l'évolution du résidu global obtenu au sens des moindres carrés (c'est-à-dire par la racine carrée de la formule (5.4) où ρ_1 et ρ_2 sont la fonction $\rho(x) = x^2$), après l'étape de mise à jour des primitives. Ce résidu représente la distance globale (en pixels) entre les primitives projetées et les primitives mesurées dans l'image : nous l'appellerons *erreur de reprojection*. Ce résidu tient compte des erreurs locales, les primitives aberrantes ayant

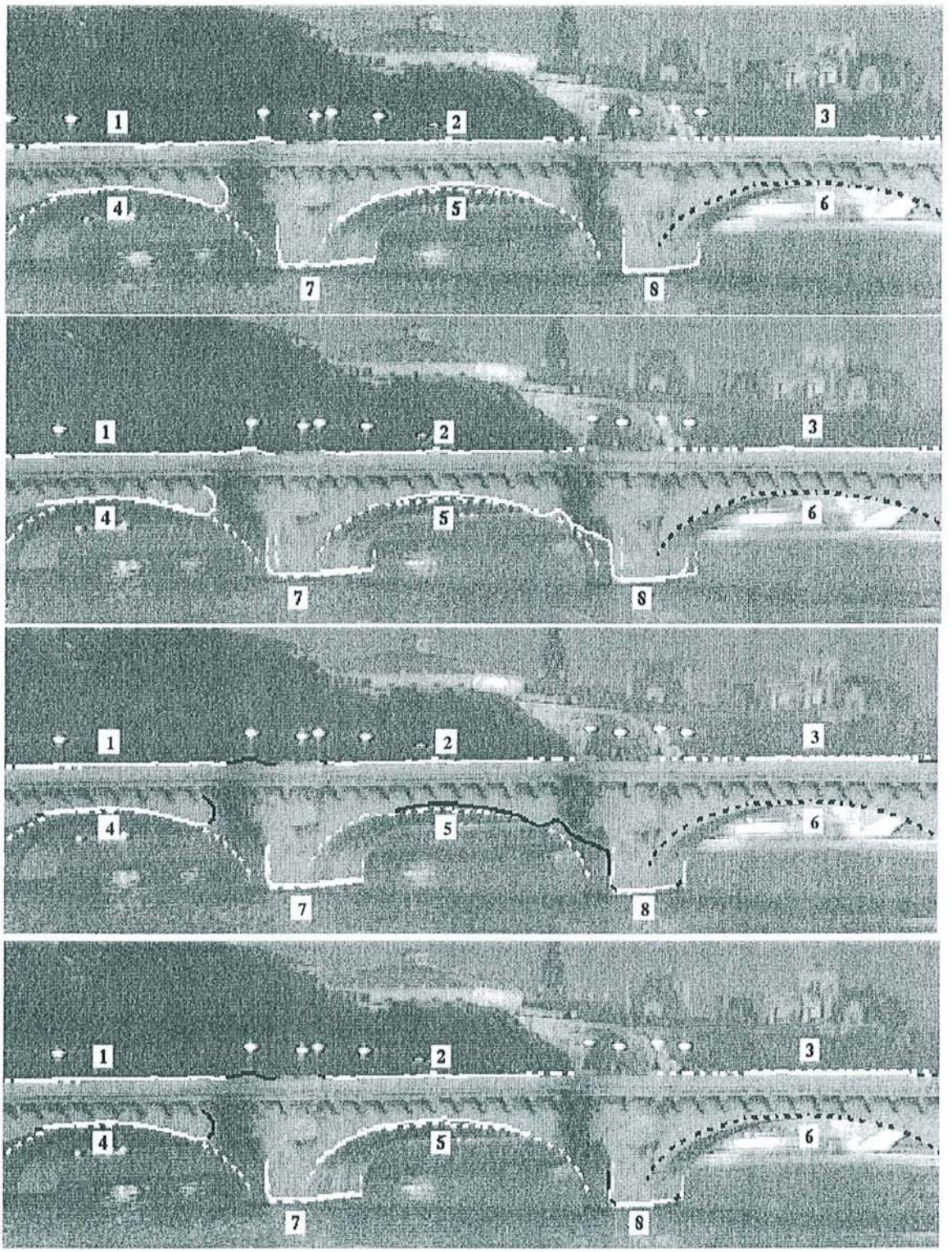


FIG. 5.12 – *Itération dans l'image 13. a. Appariements obtenus dans l'image 12. b. Suivi des courbes dans l'image 13 (les projections des primitives 3-D sont celles de l'image 12). c. Recalage du modèle. d. Mise à jour des primitives.*

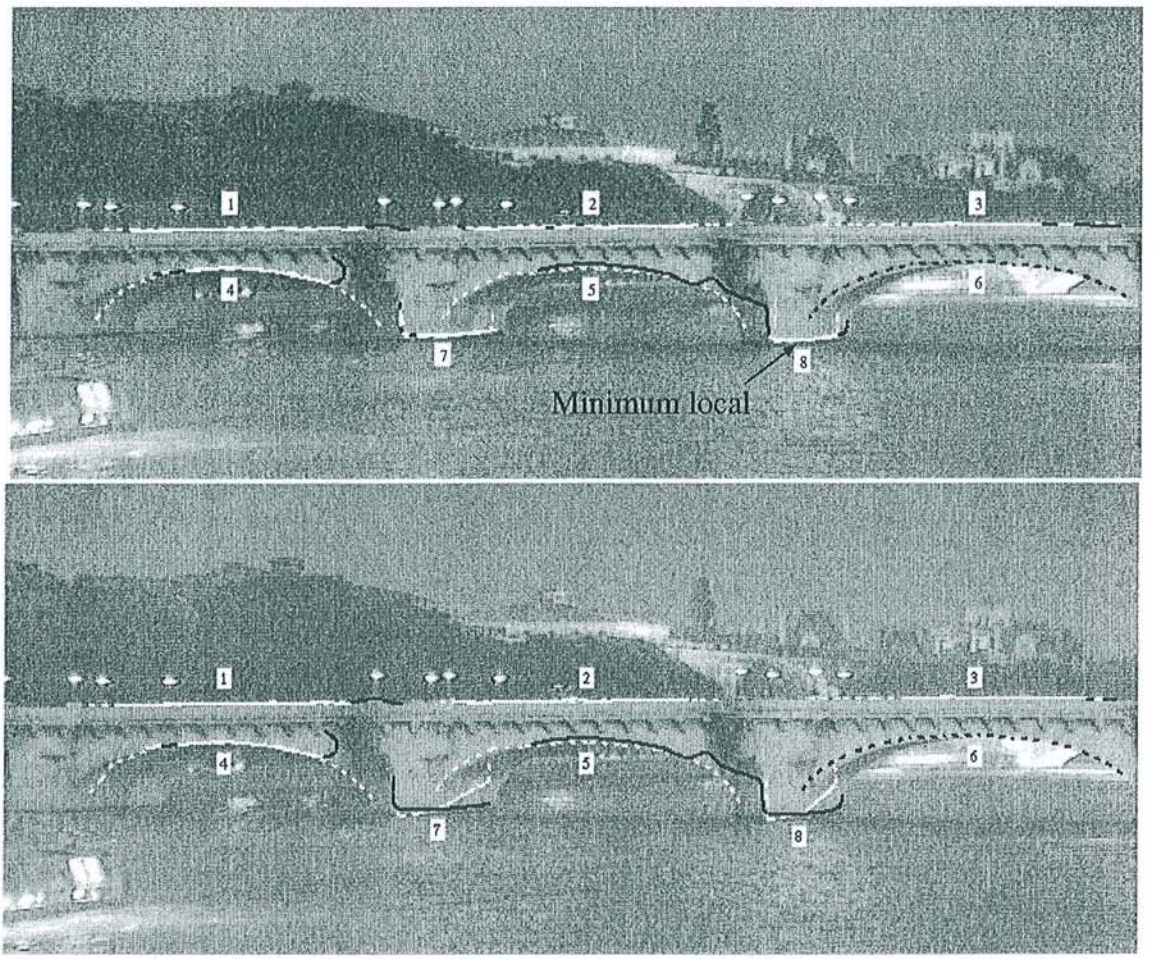


FIG. 5.13 – *Exemples de problèmes inhérents à l'utilisation d'estimateurs médians dans l'image 13 de la séquence du Pont Neuf. a. Utilisation de l'estimateur LTS au niveau local : un minimum local est obtenu pour la primitive 8. b. Utilisation de l'estimateur LTS au niveau global : les primitives 7 et 8 comportant l'information de profondeur ne sont pas prises en compte.*

été éliminées à ce stade. C'est pourquoi nous représentons aussi l'évolution d'un résidu robuste (racine carrée de la formule (5.4) où ρ_1 et ρ_2 sont les estimateurs robustes de Huber et (resp.) Tukey) : un pic de l'erreur de reprojection qui n'apparaît pas sur la courbe du résidu robuste indique donc la présence de données erronées qui n'ont pas perturbé le calcul du point de vue.

Nous voyons que l'erreur de reprojection incluant les erreurs locales est de l'ordre du pixel, ce qui traduit un bon recalage des primitives, confirmé visuellement par la reprojection du modèle filaire en figure 5.16.

Les déplacements de la caméra, représentée par son axe optique et le plan image, sont tracés dans l'espace 3-D en figure 5.17. La caméra étant posée sur un trépied, on retrouve à peu près un mouvement panoramique, mais les axes optiques ne se coupent pas exactement au même point. Toutefois, l'angle α, qui exprime la rotation autour de l'axe des z, évolue régulièrement au fur et à mesure que la caméra tourne (figure 5.15.b). La table 5.2 donne par ailleurs la moyenne et l'écart-type des autres paramètres sensés être constants. On constate que si t_x et t_y sont à peu prêt constants, la translation en z est assez instable (voir aussi la figure 5.17) : ceci est principalement dû au manque d'information de profondeur qui caractérise cette séquence.

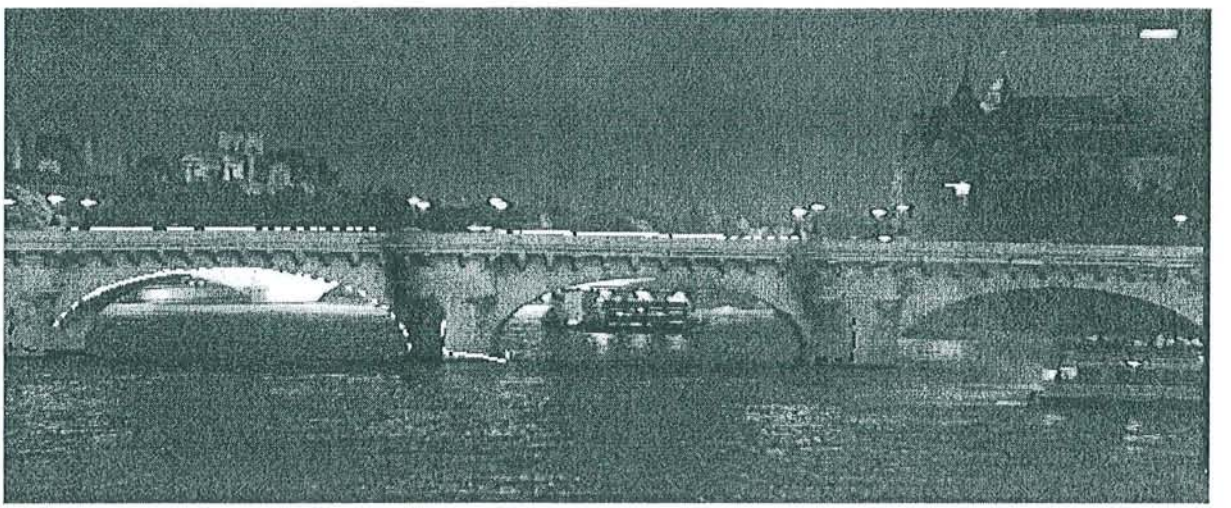


FIG. 5.14 – *L'information de profondeur n'est plus représentée dans l'image 164.*

Paramètre	*Moyenne*	*Écart type*
t_x (m)	-91.686	1.678
t_y (m)	325.359	2.339
t_z (m)	13.092	2.547
β (rad)	0.010	0.006
γ (rad)	0.000	0.006

TAB. 5.2 – *Moyenne et écart-type des paramètres fixes de la caméra sur la séquence du Pont-Neuf.*

La figure 5.15.c montre l'évolution du nombre de primitives appariées et du nombre de primitives visibles selon la définition donnée précédemment. Ceci nous permet d'illustrer l'autonomie du système : la courbe des primitives visibles montre que 9 primitives sont sorties du champ de vision sur cette séquence, et que 7 autres sont apparues. Le nombre de primitives appariées varie entre 3 et 7, tandis que le nombre de primitives visibles varie entre 6 et 10. Les primitives visibles ne sont pas toujours appariées immédiatement : l'utilisation d'estimateurs robustes peut rendre difficile l'intégration d'une nouvelle courbe lorsque les résidus des autres courbes sont très petits. D'autre part, les primitives apparaissent sur les bords de l'image, zone où le phénomène de distorsion est le plus marqué et où la projection de la primitive est donc la plus éloignée de la courbe réelle.

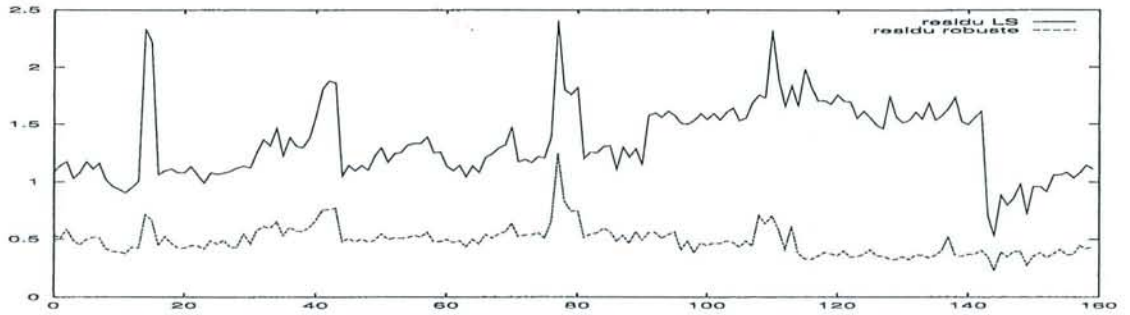
Deux séquences MPEG sont aussi visibles sur le site internet. La première montre les appariement 3-D/2-D obtenus après l'étape de suivi et après l'étape de mise à jour, et la deuxième la reprojection du modèle filaire sur la séquence. L'effet de sautillement observé est la conséquence directe du manque d'information de profondeur.

5.4.2 La séquence du château miniature

Le deuxième exemple est une séquence de château miniature, filmée à la main par un observateur en mouvement. Les principales difficultés pour cette application sont les suivantes :

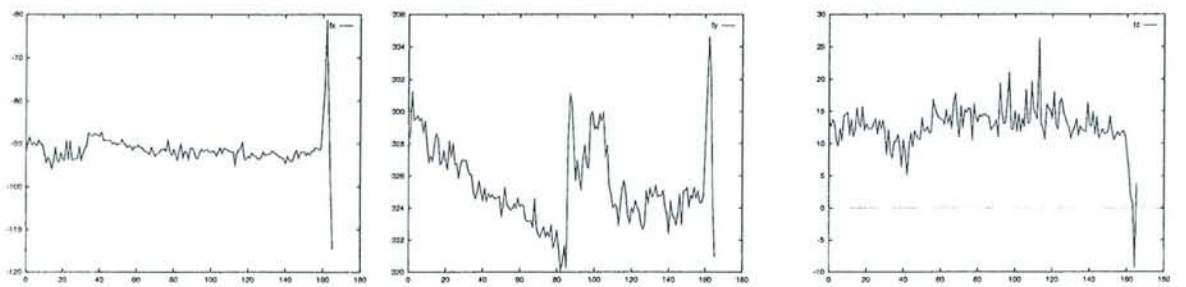
- l'objet de référence a été modélisé à l'aide d'une simple règle souple, ce qui conduit à un modèle peu précis ;

a.

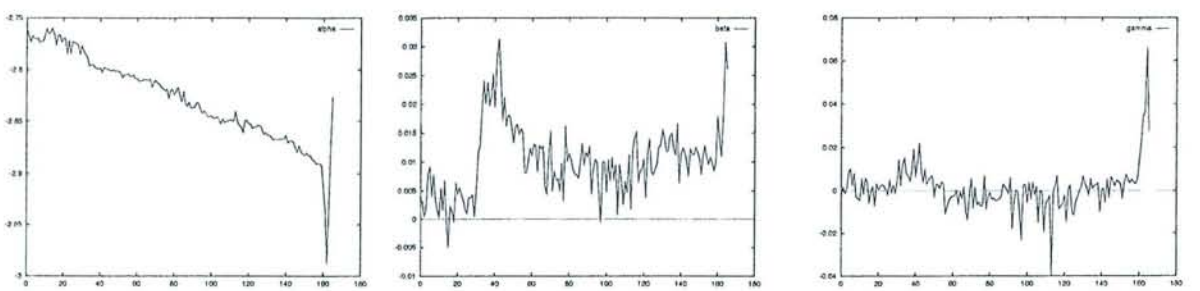


Résidus aux moindres carrés et résidus robustes.

b.

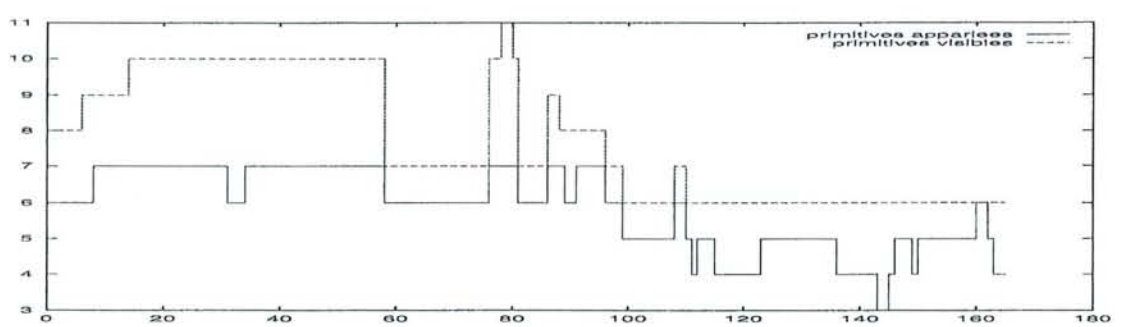


$t_x(m)$ $t_y(m)$ $t_z(m)$



$\alpha(rad)$ $\beta(rad)$ $\gamma(rad)$

c.



Nombre de primitives visibles et appariées.

FIG. 5.15 – *Résultats statistiques obtenus sur la séquence du Pont-Neuf.*

- le mouvement de la caméra est particulièrement saccadé (voir la séquence sur le site internet et la figure 5.22). De plus, le déplacement entre deux images consécutives peut être très important, provoquant un nombre conséquent d'erreurs de suivi ;
- il s'agit principalement d'une translation vers le château, ce qui implique de grands changements d'échelle dans l'image et donc des difficultés de suivi ;

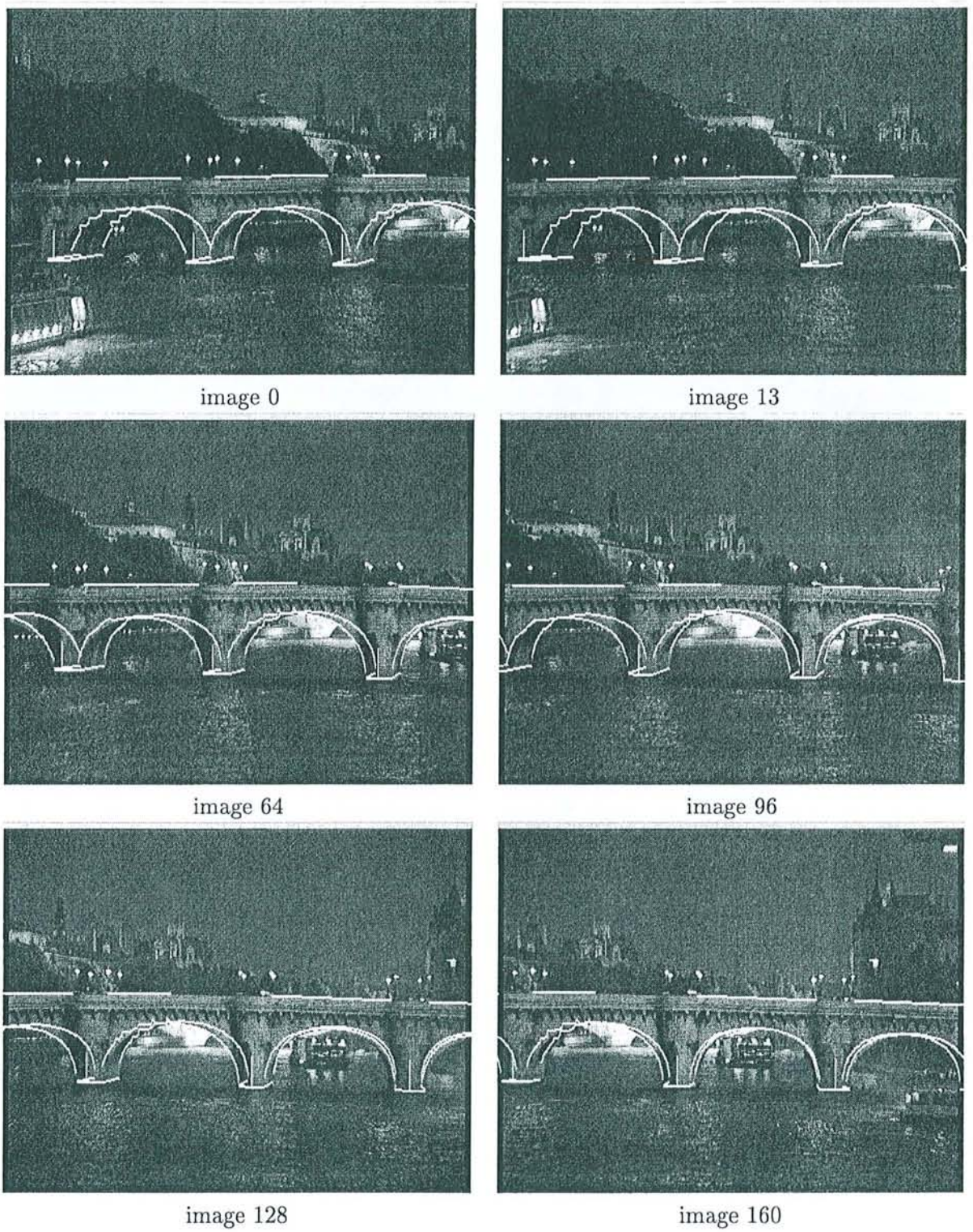


FIG. 5.16 – *Projection du modèle filaire du Pont-Neuf dans les images de la séquence.*

– la carte des contours est dense (figure 5.18), ce qui pénalise l'initialisation des primitives. D'autre part, les nombreuses zones de gradients forts générés au niveau des pierres peuvent perturber l'algorithme de suivi.

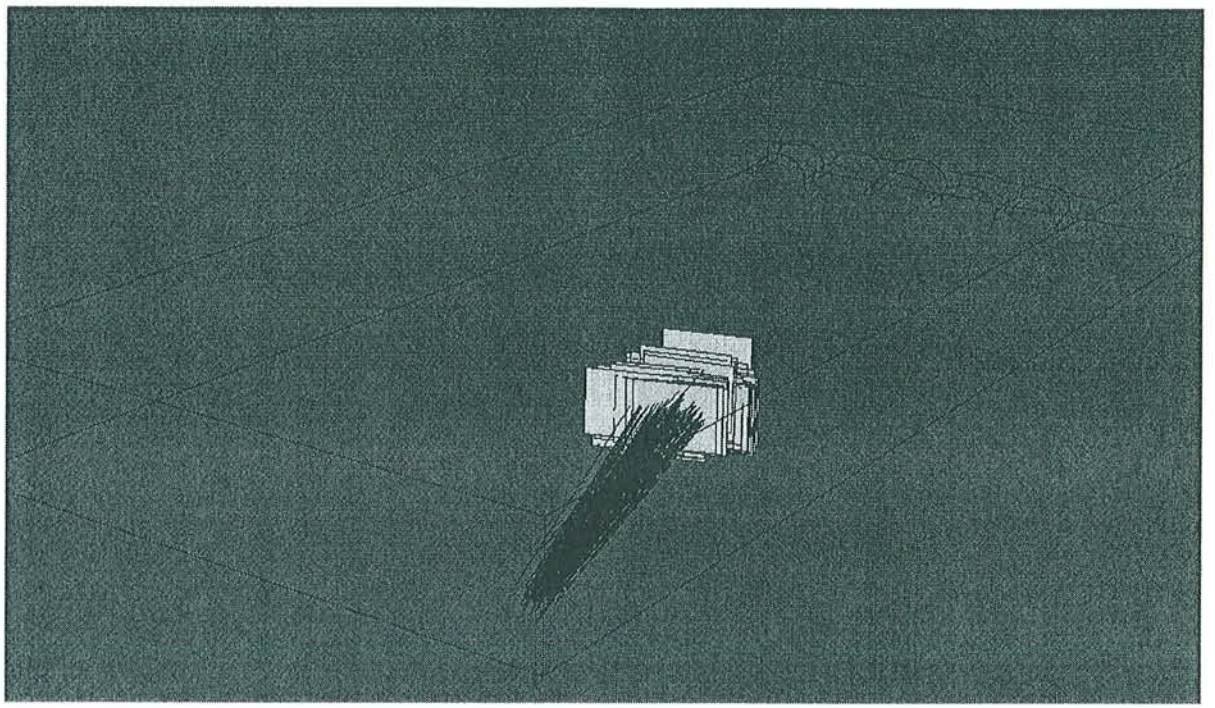


FIG. 5.17 – *Trajectoire 3-D retrouvée pour la séquence du Pont-Neuf.*

Malgré ces difficultés, le système se comporte de façon satisfaisante. La figure 5.19 montre les différentes étapes de la boucle de recalage dans une image de la séquence : en raison principalement de la présence de gradients forts autour des primitives de référence, deux d'entre elles sont mal suivies (primitives 1 et 2), pour seulement six primitives suivies (ce qui représente plus de 30% de données aberrantes). Malgré cela, le point de vue est calculé correctement et les primitives aberrantes sont détectées, et bien réinitialisées.

Les résultats statistiques obtenus sur la séquence sont présentés en figure 5.21 : l'erreur de reprojection robuste reste faible alors que le mouvement de la caméra vers le château est relativement important (voir aussi la figure 5.20). Le nombre de primitives appariées varie entre 6 et 7 (les mêmes primitives sont visibles sur toute la séquence). Les séquences complètes de suivi des primitives et de projection du modèle filaire du château sont visibles sur le site internet.

5.5 Limites du système

Nous avons présenté dans ce chapitre un système de recalage autonome, basé sur une méthode statistique robuste prenant en compte des courbes de forme libre. Toutefois, comme nous l'avons illustré avec la séquence du Pont-Neuf, la précision du point de vue calculé est fortement liée à la distribution des primitives utilisées. Une pondération des primitives en fonction de leur position spatiale permet d'améliorer le résultat, mais devient insuffisante lorsque les directions 3-D ne sont pas toutes représentées (par exemple lorsque l'information de profondeur n'est pas disponible), ou que la taille du modèle est petite par rapport à la distance modèle-caméra. La figure 5.23 montre les erreurs moyennes obtenues sur les paramètres du point de vue, à partir de tests synthétiques utilisant le même procédé qu'en 5.2, en faisant varier la profondeur du volume contenant les primitives de 0 à 100. On constate que pour les profondeurs faibles, le point de vue obtenu est très mauvais.

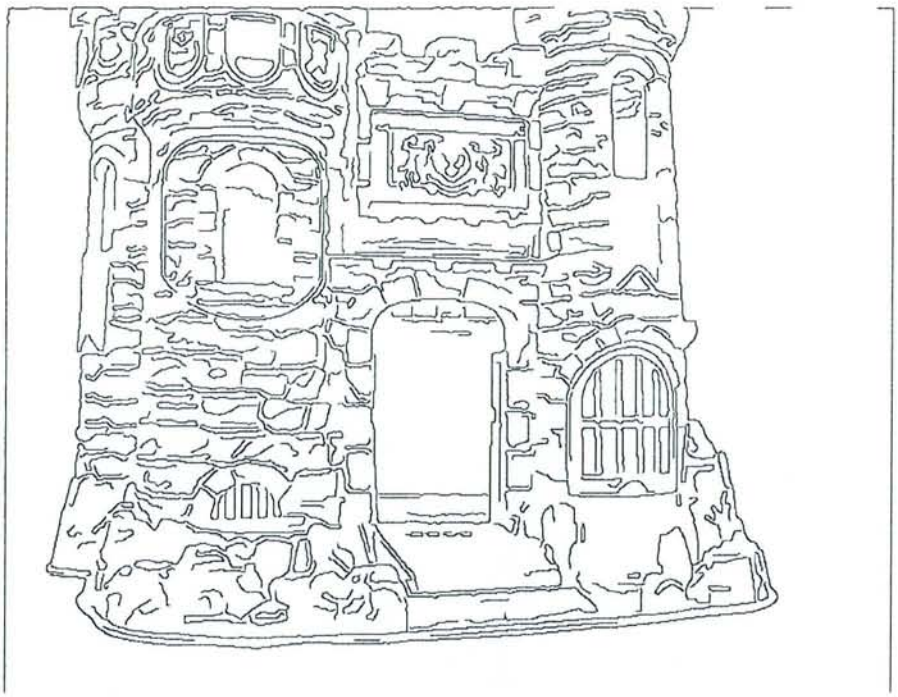


FIG. 5.18 – *La carte des contours sur la séquence du château est particulièrement dense.*

Lorsque l'objet virtuel est incrusté dans une zone proche des primitives suivies, cela peut ne pas avoir de conséquences visuellement : si la reprojection des primitives suivies est correcte, il en va de même pour tous les objets proches de ces primitives. Par contre, le résultat devient visuellement incorrect lorsque l'objet est incrusté dans une zone éloignée des primitives suivies (figure 5.24).

Ce problème est illustré sur la séquence de la place Stanislas : cette séquence montre une partie de la place, sur laquelle apparaissent l'Opéra de Nancy et la statue de Stanislas (figure 5.26). Nous souhaitons incruster un véhicule virtuel roulant sur la place. Pour calculer le point de vue, nous disposons d'une modélisation de la façade de l'opéra (figure 5.25), c'est-à-dire d'une ensemble de primitives quasiment coplanaires (figure 5.26). Nous ne connaissons malheureusement pas le modèle de la statue qui aurait apporté une information de profondeur. La figure 5.27 montre le résultat de l'incrustation du véhicule dans la première image, à distance croissante de l'opéra. Le point de vue a été initialisé par la méthode de Dementhon et Davis à partir de quatre points, dont trois situés sur la façade de l'Opéra et un au pied de la statue dont nous avons grossièrement estimé la position. La procédure *R-pdv+* a ensuite été utilisée pour optimiser le point de vue à partir des courbes visibles en figure 5.26. Nous voyons que plus le véhicule est éloigné de l'Opéra, moins sa projection semble correcte. Pourtant, la projection de l'Opéra (figure 5.28) est tout à fait correcte, ce qui prouve que l'erreur de reprojection n'est pas un critère suffisant pour l'évaluation d'une méthode de recalage.

Pour pallier ce problème, nous allons dans le chapitre suivant utiliser des appariement 2-D/2-D de points, qui sont susceptibles d'être détectés à n'importe quel endroit de la scène, et apporter ainsi l'information tridimensionnelle qui nous fait défaut.

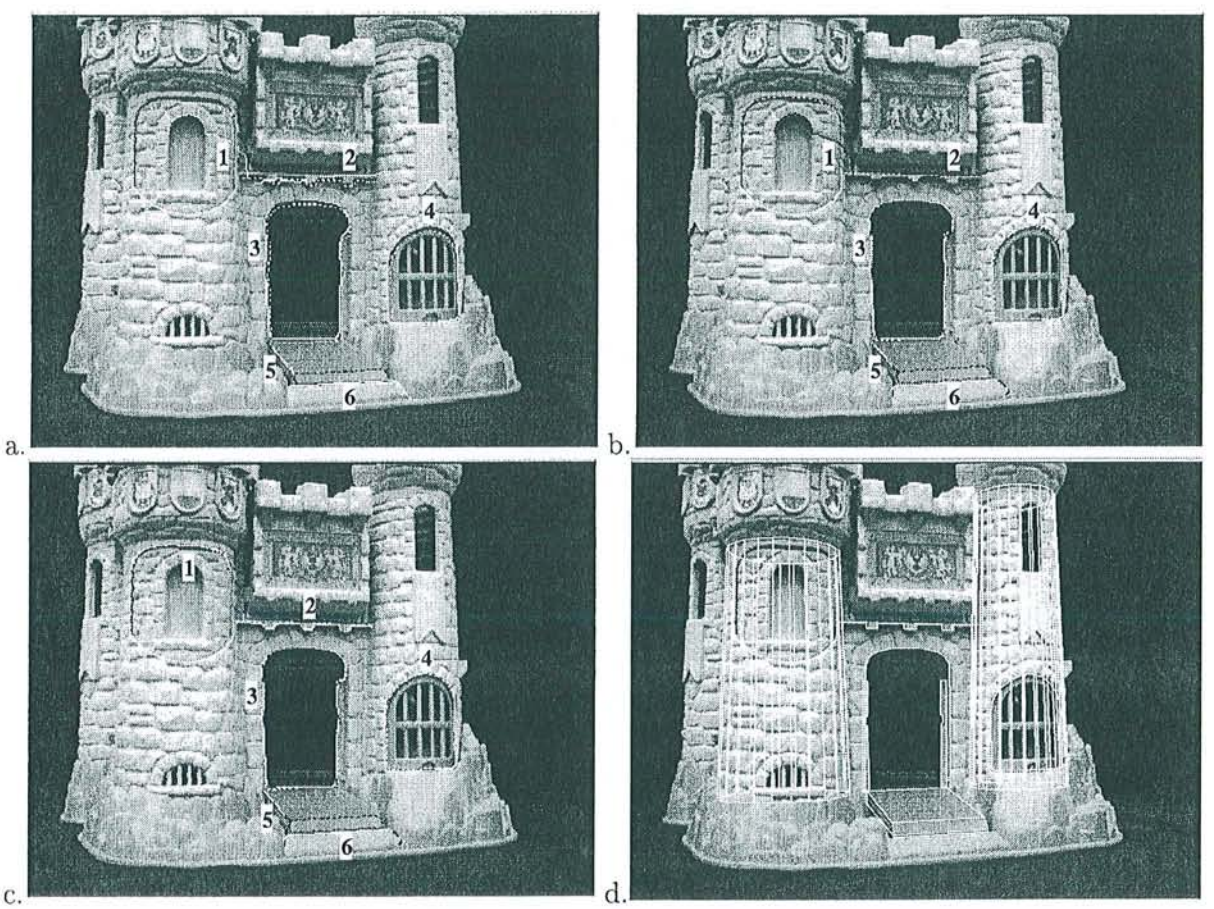


FIG. 5.19 – *Recalage temporel dans une image de la séquence du château miniature. a. Suivi des primitives. b. Recalage. c. Mise à jour des primitives. d. Projection du modèle filaire du château.*

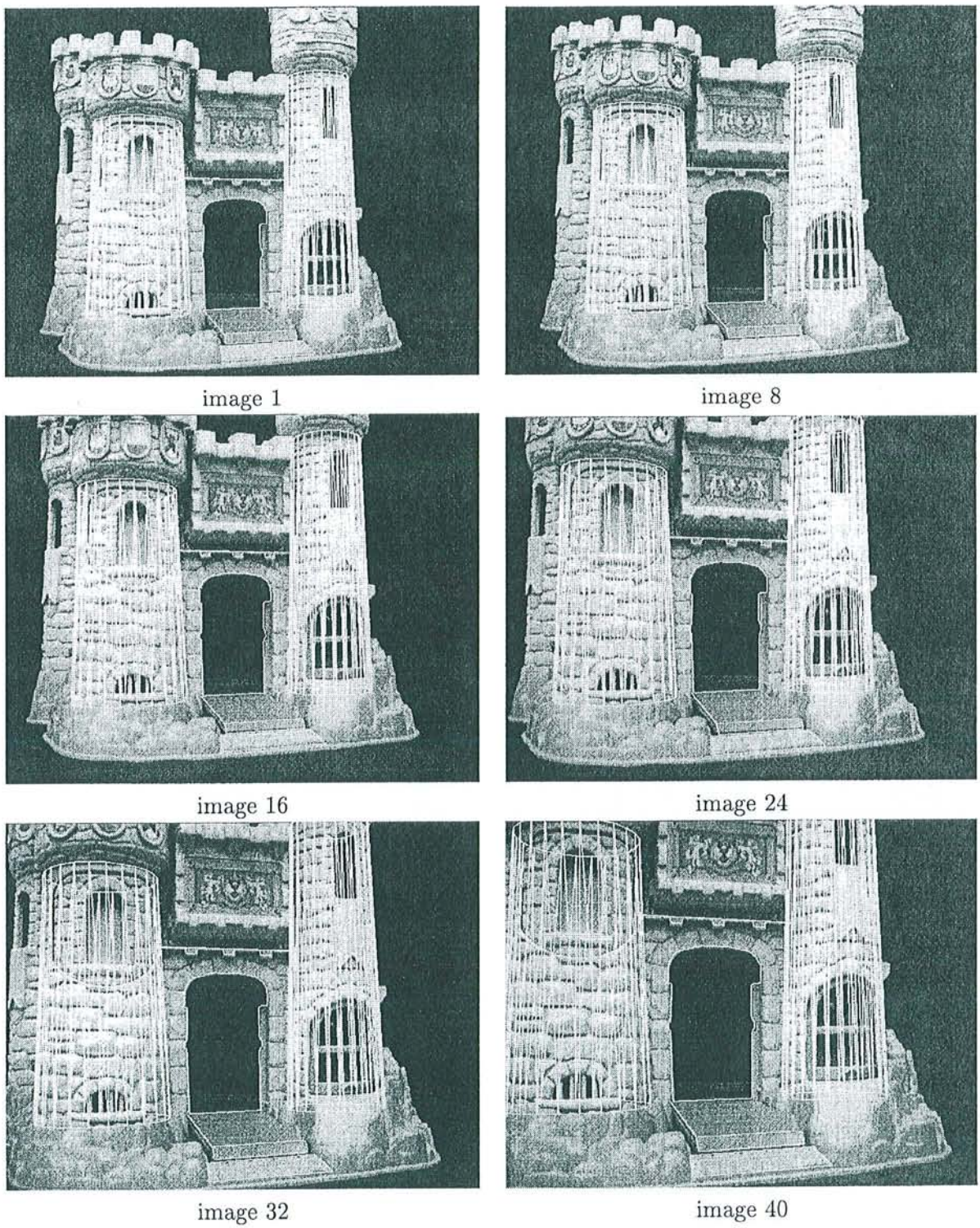
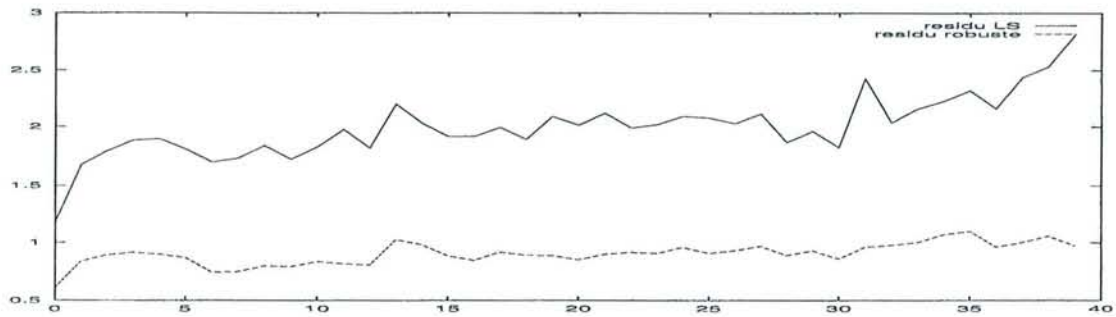
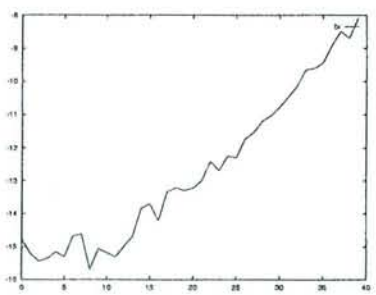


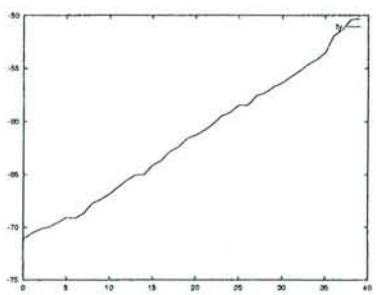
FIG. 5.20 – *Projection du modèle filaire du château miniature dans les images de la séquence.*



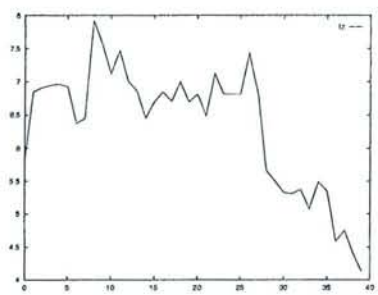
Résidus aux moindres carrés et résidus robustes (pixels).



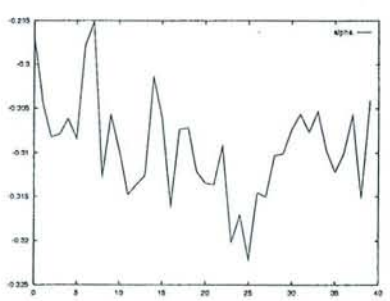
$t_x(cm)$



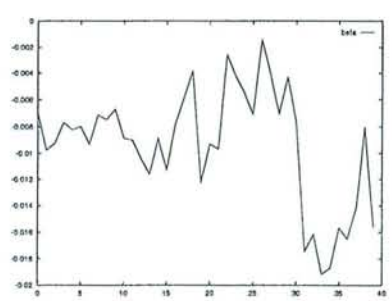
$t_y(cm)$



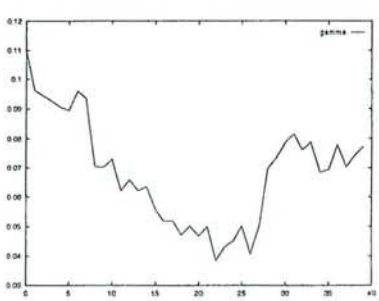
$t_z(cm)$



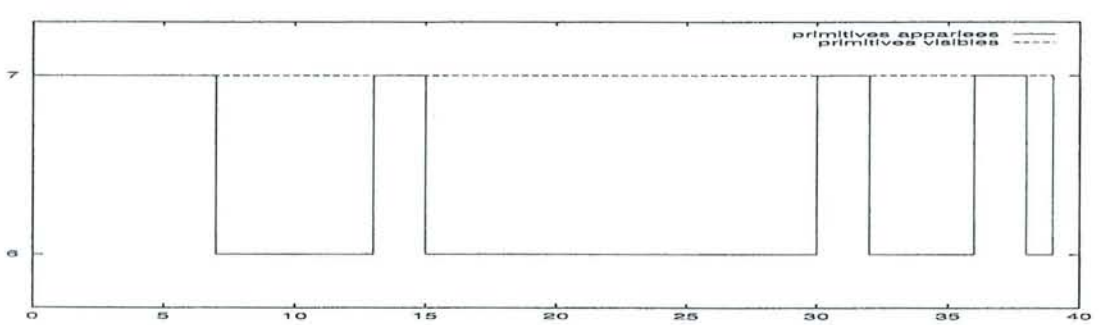
α (rad)



β (rad)



γ (rad)



Nombre de primitives visibles et appariées.

FIG. 5.21 – *Résultats statistiques obtenus sur la séquence du chateau.*

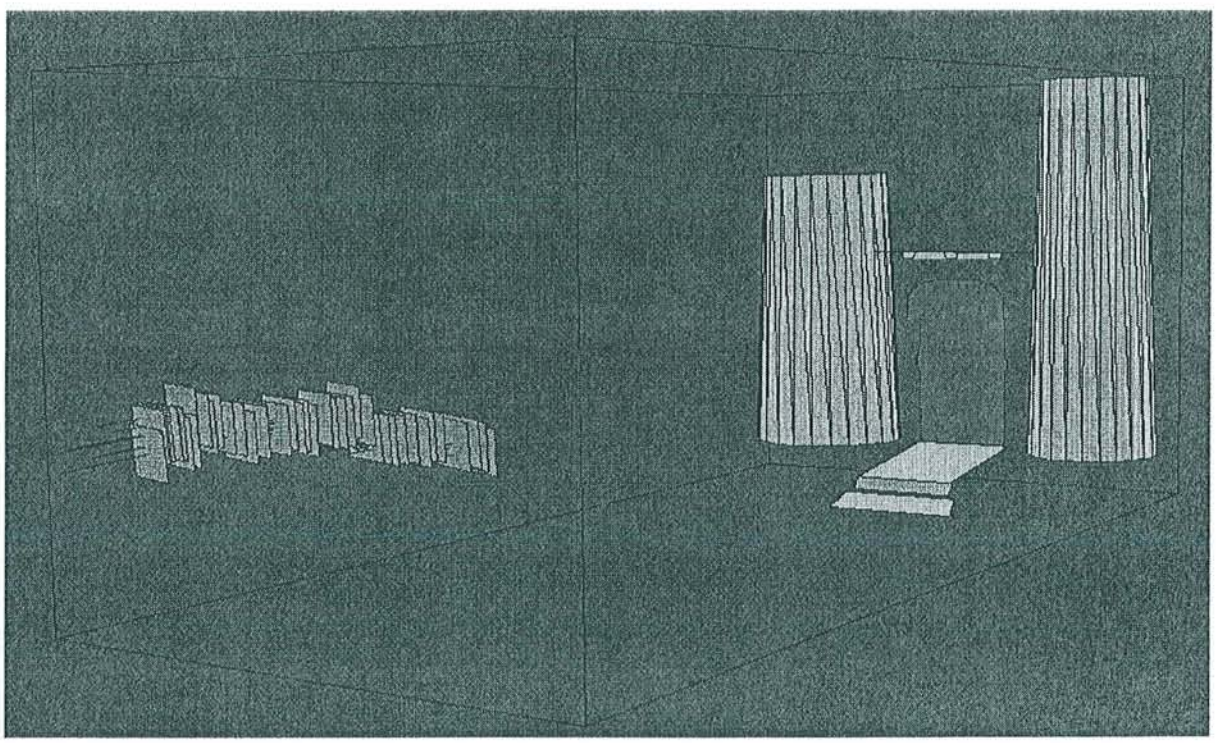


FIG. 5.22 – *Trajectoire 3-D retrouvée pour la séquence du chateau.*

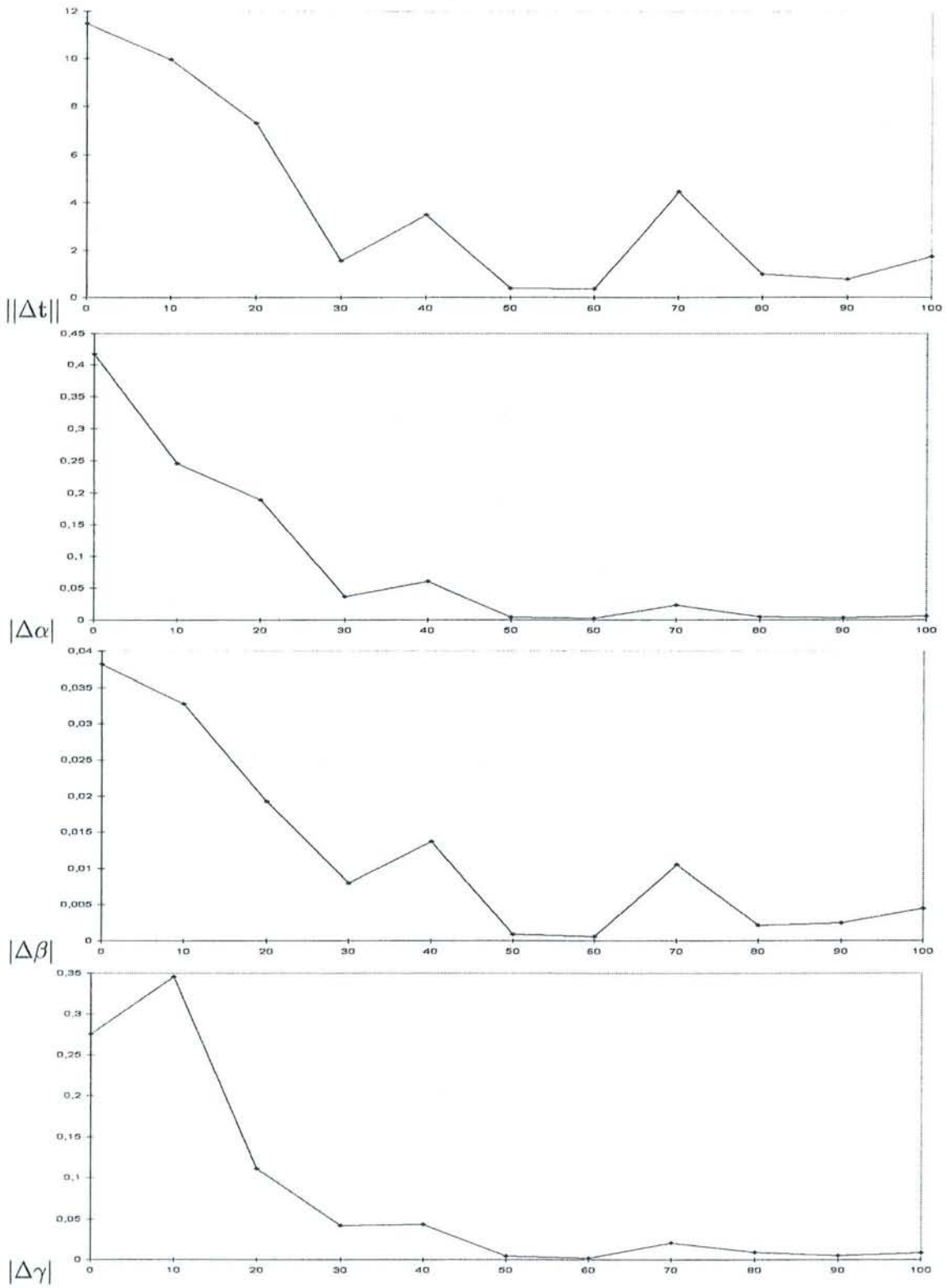


FIG. 5.23 – *Erreur moyenne obtenue sur les paramètres du point de vue en fonction de la profondeur du modèle.*

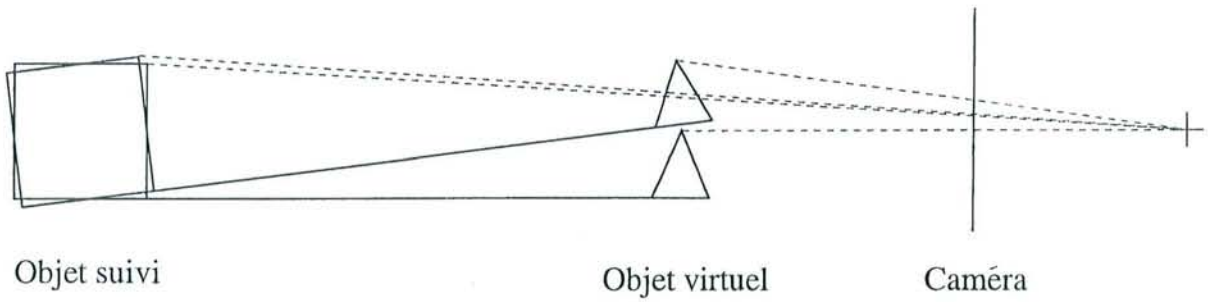


FIG. 5.24 – *Une petite erreur de reprojection au niveau de l'objet suivi peut conduire à une incrustation complètement fausse pour un objet virtuel éloigné de l'objet ayant servi au calcul du point de vue.*

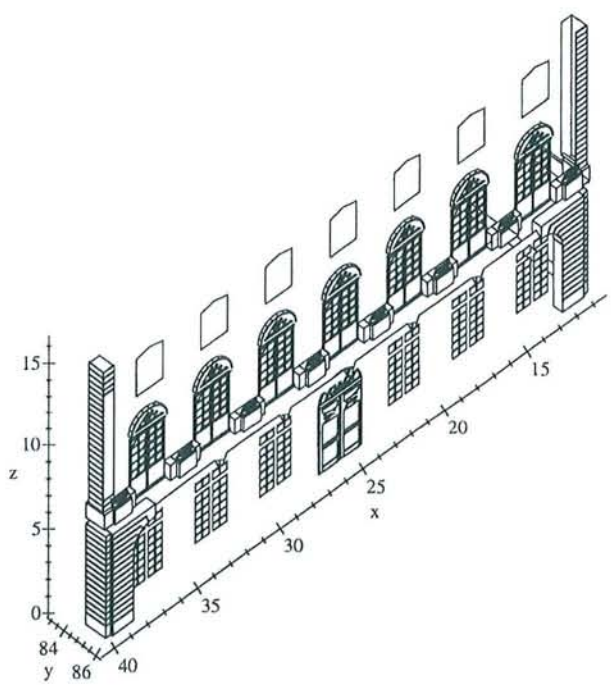


FIG. 5.25 – *Modèle filaire de la façade de l'Opéra de Nancy.*

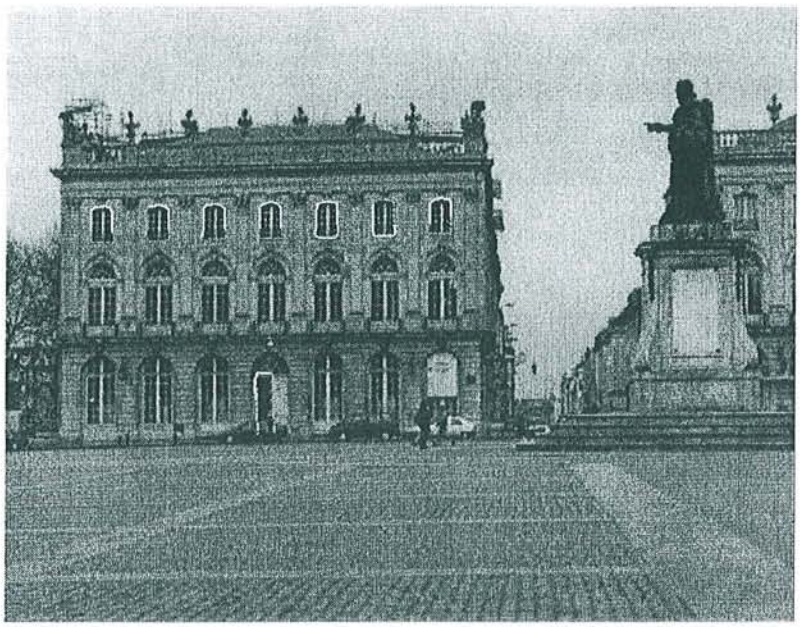


FIG. 5.26 – *Primitives utilisées pour calculer le point de vue dans la première image de la séquence de la place Stanislas.*

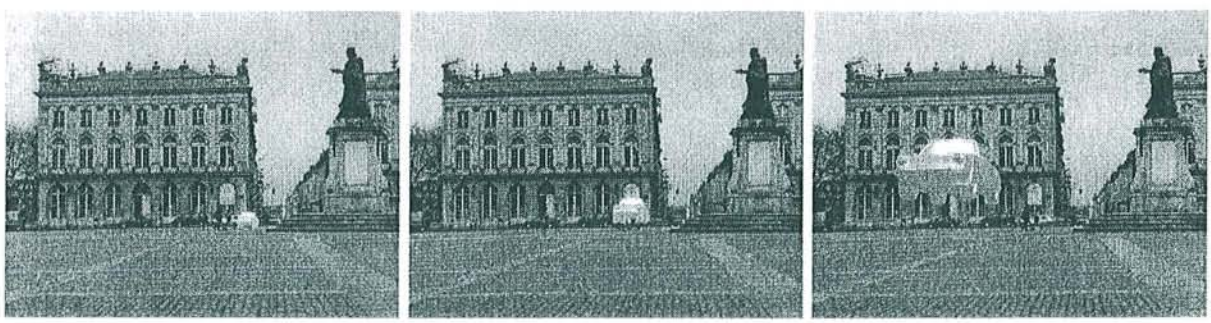


FIG. 5.27 – *Incrustation d'un véhicule dans la première image, à distance croissante de l'Opéra.*

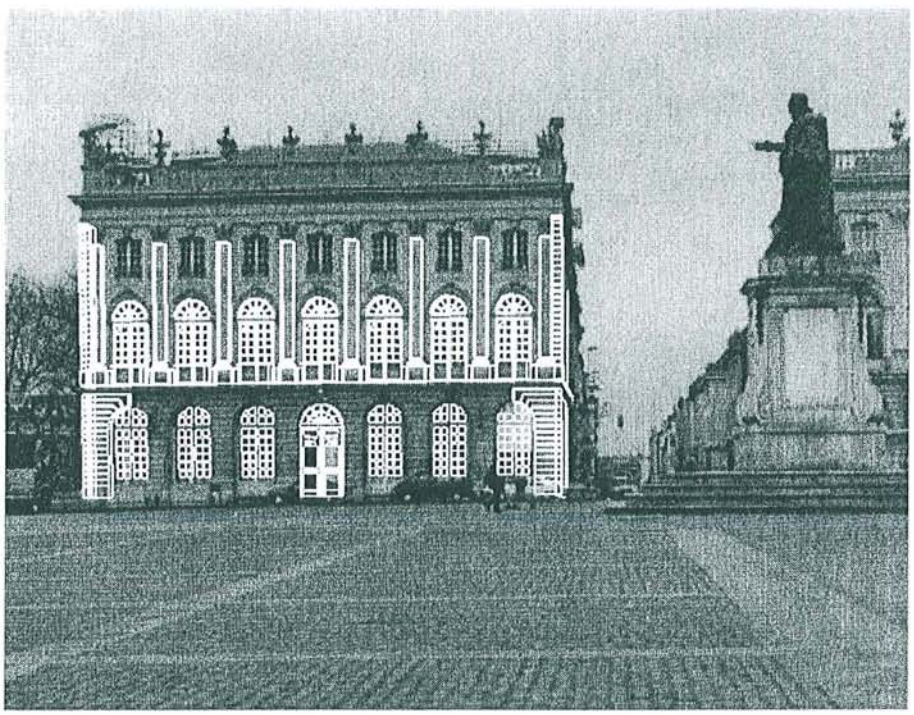


FIG. 5.28 – *La projection de l'Opéra semble correcte alors que le point de vue est faux.*

Chapitre 6

Correction du point de vue basée image

La précision du recalage basé modèle est fortement liée à la distribution spatiale des primitives suivies, alors que bien souvent seule une partie de la scène peut être modélisée. Pour pallier ce problème, nous utilisons des appariements de points 2-D/2-D, qui nous permettent d'obtenir une information supplémentaire sur la structure de la scène, dans des zones non modélisées *a priori*. Nous verrons que, grâce au maintient du suivi 3-D/2-D, cette correction du point de vue basée image peut se faire de façon autonome et séquentielle, conservant ainsi les caractéristiques essentielles de notre système. Des résultats expérimentaux sont présentés sur deux séquences grandeur réelle : la séquence de la place Stanislas et la séquence du Pont Neuf.

6.1 La méthode hybride

Le problème de la distribution spatiale des primitives utilisées pour la calibration a aussi été observé dans [Zhang et al.97]. L'objectif était alors de reconstruire l'environnement à partir d'appariements 2-D/2-D de points d'intérêt, une mire de calibration étant placée dans la scène pour le calcul des matrices de projection. Les auteurs ont alors constaté que si la distance des points d'intérêt aux épipolaires était faible dans le volume de la mire, celle-ci pouvait devenir très importante à d'autres endroits de la scène. Pour pallier ce problème, la caméra était alors calibrée dans l'espace projectif à partir des points d'intérêt, puis la mire était utilisée pour retrouver la structure euclidienne de la scène.

Le principe de la méthode hybride est de minimiser *simultanément* des résidus 3-D/2-D obtenus à partir de courbes suivies dans la séquence, et des résidus 2-D/2-D obtenus à partir de points détectés et appariés entre deux images consécutives de la séquence. Nous commençons par décrire le cas d'un mouvement général de caméra (6.1.1), avant d'évoquer celui des rotations pures, qui est un cas dégénéré pour la géométrie épipolaire (6.1.2). Nous voyons enfin comment il est théoriquement possible de distinguer ces deux cas de façon automatique (6.1.3).

6.1.1 Cas général

Supposons que nous ayons établi n appariements 3-D/2-D de courbes quelconques dans l'image courante I, et m appariements de points $(\mathbf{q}_i,\mathbf{q}'_i)$ entre l'image I et une autre image I' de la séquence, dont on connaît le point de vue $(\mathbf{R}' \quad \mathbf{t}')$. Nous cherchons à calculer les paramètres $\mathbf{p}$ du point de vue $(\mathbf{R} \quad \mathbf{t})$ pour l'image I. Nous avons vu au chapitre 2, qu'à partir d'appariements 2-D/2-D entre les images I et I', la contrainte épipolaire permet de déterminer le mouvement de la caméra $(\Delta\mathbf{R} \quad \Delta\mathbf{t})$ entre ces deux images. En particulier, Luong a montré

que la méthode itérative consistant à minimiser la distance $d(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i)$ des points $\mathbf{q}'_i$ aux droites épipolaires des correspondants $\mathbf{q}_i$ dans l'image I', en même temps que les distances symétriques $d(\mathbf{q}_i,\mathbf{F}^T\mathbf{q}'_i)$, était relativement stable. Nous pouvons donc ajouter une contrainte supplémentaire au calcul du point de vue basé sur le modèle de la scène en minimisant les résidus

$$v_i^2 = \frac{1}{2}(d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i) + d^2(\mathbf{q}_i,\mathbf{F}^T\mathbf{q}'_i)). \tag{6.1}$$

v_i peut être exprimé en fonction de $\mathbf{p}$: d'après 2.6, nous avons en effet $\mathbf{F} = \mathbf{A}^{-T}\Delta\mathbf{T}\Delta\mathbf{R}\mathbf{A}^{-1}$ (en considérant que les paramètres intrinsèques ne varient pas entre les deux images), où $\Delta\mathbf{T}\mathbf{x} = \Delta\mathbf{t} \wedge \mathbf{x}$. D'autre part, on montre facilement que

$$\begin{cases} \Delta\mathbf{R} = \mathbf{R}\mathbf{R}'^T, \\ \Delta\mathbf{t} = \mathbf{t} - \mathbf{R}\mathbf{R}'^T\mathbf{t}', \end{cases}$$

ce qui nous permet d'exprimer $\mathbf{F}$ en fonction de $\mathbf{R}$ et $\mathbf{t}$, et donc $\mathbf{p}$.

Le résidu v_i est particulièrement commode à prendre en compte, puisqu'il s'agit d'une distance mesurable en pixels dans les images, que nous allons donc pouvoir combiner de façon cohérente avec les résidus 3-D/2-D r_i, qui sont des distances entre courbes de l'image, elles aussi mesurables en pixels. Ainsi, l'approche hybride consiste à minimiser la fonction:

$$h(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\rho_2(r_i) + \frac{1}{m}\sum_{i=1}^{m}\rho(v_i). \tag{6.2}$$

La somme de gauche est exactement la fonction robuste à deux niveaux $f(\mathbf{p})$ étudiée en détail au chapitre 5. Un M-estimateur ρ peut aussi être utilisé au niveau 2-D/2-D, bien qu'expérimentalement cela ne se soit pas avéré nécessaire. La figure 6.1 permet de visualiser les deux types de contraintes imposées lors de la minimisation de la fonction $h(\mathbf{p})$: contraintes basées sur la reprojection du modèle connu, et contraintes basées sur la géométrie épipolaire. La contrainte de reprojection nous permet de conserver un système stable et autonome, comme cela a été montré au chapitre 5, tout en levant l'indétermination sur le facteur d'échelle de la translation. D'un autre côté, la contrainte épipolaire nous permet de prendre en compte des éléments de la scène qui ne sont pas modélisés *a priori*, et donc d'obtenir un point de vue plus précis que dans le cas du 3-D/2-D pur.

On peut se demander combien de courbes 3-D/2-D doivent être suivies au minimum, étant donné que les appariements 2-D/2-D permettent d'obtenir le point de vue à un facteur d'échelle près, et qu'en théorie, le suivi d'une seule courbe 3-D suffit pour lever cette indétermination. Cependant, résoudre le facteur d'échelle n'est pas le seul rôle de la fonction $f(\mathbf{p})$: elle a aussi un rôle de stabilisation, grâce à la redondance d'information par rapport au 2-D/2-D pur, due à la connaissance 3-D ; elle est de plus au coeur de la boucle de recalage temporel, qui n'est autonome que grâce au processus robuste de mise à jour des primitives suivies. Si nous nous contentons de suivre une seule courbe, nous perdons tout l'aspect robuste du niveau global de $f(\mathbf{p})$: il suffit que cette courbe soit occultée par exemple, pour que le processus diverge. Nous pouvons par contre nous baser sur le fait qu'un M-estimateur est capable de tolérer jusqu'à environ 20% de données aberrantes. Ainsi, si nous suivons cinq primitives, nous pouvons nous permettre qu'une courbe soit mal suivie ou occultée. Pour dix primitives suivies, nous pouvons tolérer jusqu'à deux erreurs aberrantes, etc. Le nombre de courbes 3-D à suivre dépend donc du "risque" que nous sommes prêts à prendre pour la séquence.

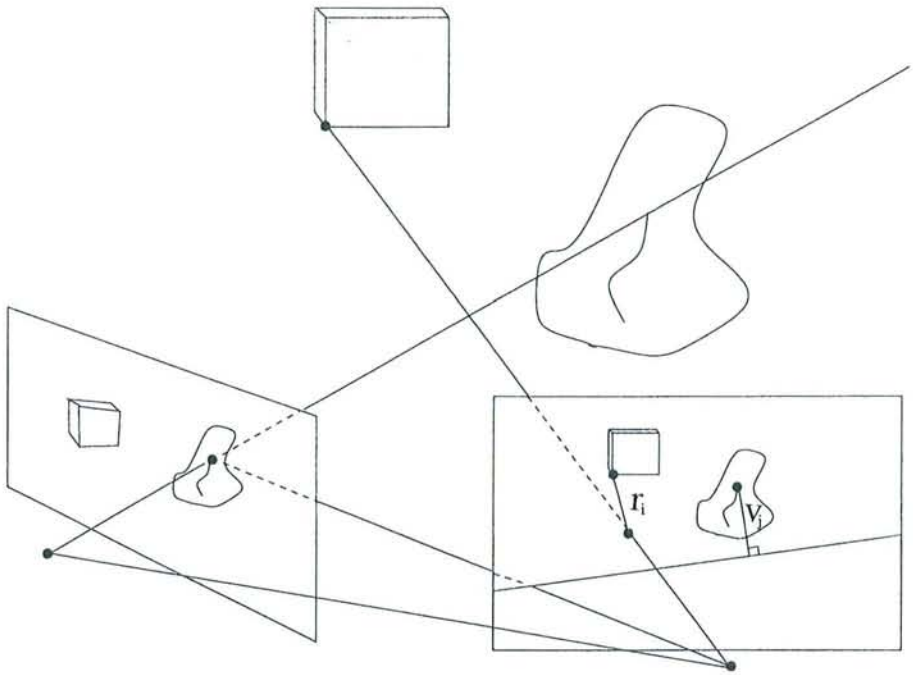


FIG. 6.1 – *la méthode hybride consiste à minimiser simultanément les résidus modèle/image et les distances des points images aux droites épipolaires. Dans cet exemple, le cube est un objet dont le modèle est connu, alors que le second objet n'est pas nécessairement modélisé.*

6.1.2 Cas dégénéré

Dans le cas d'une rotation pure, la géométrie épipolaire dégénère puisque les centres optiques des deux caméras sont confondus. Cependant, on a pour chaque couple de points $(\mathbf{q}_i,\mathbf{q}'_i)$ correspondant au même point 3-D M_i :

$$\mathbf{q}'_i = \mathbf{H}\mathbf{q}_i, \tag{6.3}$$

où $\mathbf{H} = \mathbf{A}\mathbf{R}\mathbf{R}'^T\mathbf{A}^{-1}$. En effet, si $\mathbf{m_i}$ et $\mathbf{m'_i}$ sont les coordonnées du point M_i exprimées dans le repère de chacune des deux caméras, nous avons $\mathbf{m}'_i = \Delta\mathbf{R}\mathbf{m}_i$. Il suffit ensuite d'écrire que $\mathbf{q}_i = \mathbf{A}\mathbf{m}_i$ et $\mathbf{q}'_i = \mathbf{A}\mathbf{m}'_i$ pour obtenir l'équation 6.3. Dans le cas d'une rotation pure, nous pouvons donc minimiser la fonction $h(\mathbf{p})$ en prenant pour v_i:

$$v_i = ||\mathbf{q}'_i - \mathbf{H}\mathbf{q}_i||. \tag{6.4}$$

Cependant, lorsque nous n'avons aucune connaissance *a priori* sur le mouvement de la caméra, nous ne savons pas s'il s'agit d'une rotation pure ou non, et donc quelle expression utiliser pour v_i. Il serait donc intéressant de pouvoir détecter automatiquement les rotations pures.

6.1.3 Classification automatique

Ce problème a été étudié dans [Torr97]. Il s'agit en fait d'un problème classique (mais néanmoins complexe) de sélection du modèle correspondant le mieux aux données observées, parmi un ensemble de modèles possibles. Pour cela, plusieurs *critères d'évaluation* ont été proposés. Le critère SSE (*sum of square of errors*) considère les distances e_i entre les données mesurées et les

données obtenues pour le modèle évalué :

$$\text{SSE} = \sum \frac{e_i^2}{\sigma^2},$$

où σ est l'écart-type supposé des mesures. Dans notre application, cela consisterait par exemple à comparer les valeurs de (6.1) et (6.4) après avoir optimisé (6.2) dans les deux cas. Malheureusement, Torr constate que ce critère conduit le plus souvent à la sélection du modèle le plus général. [Akaike74] propose de pénaliser la complexité du modèle en minimisant le critère AIC (*Akaike Information Criterion*):

$$\text{AIC} = \sum \frac{e_i^2}{\sigma^2} + 2p,$$

où p est le nombre de paramètres du modèle choisi. Ce critère ne tient néanmoins pas compte de la dimension du modèle testé : ce problème peut être illustré en considérant le cas d'un nuage de points que l'on souhaite approximer soit par une droite, soit par un point. Les deux modèles ont pour dimension 2, mais il est facile de montrer que le premier terme de AIC (égal à SSE) est toujours plus petit pour un modèle de droite que pour un modèle de point. Le modèle de droite est donc systématiquement choisi. Pour pallier ce problème, [Kanatani96] prend en compte la dimension du modèle en minimisant le critère GIC (*Geometric Information Criterion*):

$$\text{GIC} = \sum \frac{e_i^2}{\sigma^2} + 2p + 2dn,$$

où d est la dimension du modèle et n le nombre de données dont on mesure l'erreur e_i. Ainsi, si nous reprenons l'exemple précédent, nous avons $d = 1$ dans le cas de la droite et $d = 0$ dans le cas du point, c'est-à-dire:

$$\begin{aligned} &\text{GIC(droite)=SSE(droite)} + 4 + 2n, \\ &\text{GIC(point)=SSE(point)} + 4. \end{aligned}$$

Comme SSE(point) $\geq$ SSE(droite), nous pouvons poser SSE(point) = SSE(droite)+r : le modèle de point est donc favorisé si $r \leq 2n$. Enfin, constatant qu'une seule donnée aberrante peut conduire à un mauvais choix, Torr introduit un estimateur robuste en minimisant le critère GRIC (*Geometric Robust Information Criterion*):

$$\text{GRIC} = \sum \rho(e_i^2) + 2p + 2dn.$$

En prenant ce critère, Torr parvient à distinguer les mouvements généraux des mouvements de rotation pure sur des données réelles et synthétiques. Nous pourrions donc envisager de l'intégrer à notre système afin de pouvoir alterner des mouvements quelconques de caméra avec des mouvements de rotation pure. Toutefois, dans la version actuelle de notre boucle de recalage temporel, nous ne cherchons pas à distinguer automatiquement les deux cas. Généralement, soit la caméra est posée sur un trépied et nous utilisons le modèle de rotation, soit elle est en mouvement et le modèle général est pris en compte. Nous mentionnons donc explicitement en début de traitement, de quel type de mouvement il s'agit.

6.2 Détection et appariement des points image/image

Pour l'appariement des points 2-D/2-D, au moins deux stratégies sont possibles : la première consisterait à détecter manuellement ou automatiquement des points d'intérêt dans la première

image, et à suivre ces points par corrélation dans les images suivantes de la séquence. Cette solution est couramment employée lorsque les primitives images seules sont utilisées, afin de pouvoir mettre en œuvre les techniques de factorisation ou d'ajustement de faisceau qui nécessitent que les même primitives soient détectées dans plusieurs images. Elle est cependant peu sûre puisque des erreurs de suivi peuvent se produire, et certaines primitives peuvent se trouver occultées, ou sortir du champ de vision de la caméra au cours de la séquence.

Dans notre cas, nous n'avons aucunement besoin que les même points soient suivis dans plusieurs images, puisqu'il nous suffit de disposer d'un certain nombre d'appariements entre l'image courante et une autre image de la séquence, dont on connaît le point de vue. Nous adoptons donc la stratégie suivante : des points d'intérêt sont détectés automatiquement par la méthode de Harris et Stephens dans l'image courante et l'image précédente (par exemple) de la séquence, puis ces points sont appariés par la méthode de Zhang et al. Ainsi, les points détectés sont nécessairement visibles dans les deux images, et nous ne sommes pas exposés aux problèmes d'occultations ou de sortie des primitives du champ de vision.

6.2.1 Détection des points d'intérêt

Nous utilisons le détecteur de [Harris et al.88] pour identifier des points d'intérêt dans les images de la séquence. Cette méthode prend en compte les dérivées premières de l'intensité (ou niveau de gris), en recherchant pour chaque pixel de l'image les valeurs propres de la matrice

$$e^{-\frac{x^2+y^2}{2\sigma^2}} \otimes \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix},$$

où $\otimes$ est le produit de convolution et $I_x = \partial I / \partial x$. Le terme de gauche a pour fonction de lisser l'image autour du pixel avant de calculer les dérivées. Si les valeurs propres de cette matrice sont grandes, cela indique une forte variation des niveaux de gris autour du pixel, et donc la présence d'un point d'intérêt.

6.2.2 Appariement

Les points ainsi détectés sont appariés en utilisant l'algorithme de [Zhang et al.94], qui utilise une technique de relaxation particulièrement efficace pour choisir les couples de correspondants parmi les candidats. Une deuxième phase consiste à estimer de façon robuste la géométrie épipolaire à partir des points appariés par relaxation, afin d'améliorer encore les résultats en éliminant les appariements qui ne respectent pas la contrainte épipolaire. Toutefois, pour notre système, nous utilisons directement les appariements obtenus à l'issue de la première phase, et nous nous passons de la seconde phase qui est assez coûteuse en temps de calculs, et n'apporte pas d'amélioration tangible au regard de notre application. En effet, l'introduction d'un M-estimateur ρ dans l'optimisation de la fonction (6.2) réduit l'influence des faux appariements, ce qui nous dispense de les éliminer au préalable. Par ailleurs, l'utilisation de ce M-estimateur ne s'est pas avérée nécessaire pour les séquences que nous avons traitées, puisque le taux de faux appariements obtenus par la première phase de la méthode de Zhang et al. était de l'ordre de 1%. Nous décrivons à présent la phase d'appariement par relaxation dans ses grandes lignes.

Identification des couples plausibles

Autour de chaque point d'intérêt m_{1i} de l'image 1, une zone de recherche est définie (un rectangle), dont sont extraits les points d'intérêt m_{2j} de l'image 2. La taille de cette zone de

recherche est choisie en fonction de la disparité maximale estimée entre les images. Pour chaque couple de points (m_{1i},m_{2j}), un *score de corrélation* normalisé c_{ij} est calculé, qui évalue la similarité de fenêtres de corrélation autourant les deux points. Ce score est compris entre -1 pour des fenêtres de corrélation qui ne sont pas similaires du tout et 1 pour des fenêtres identiques. Un seuil est alors utilisé (typiquement 0.8) pour sélectionner les couples plausibles, ou *appariements candidats*.

Définition d'un score d'appariement

À l'issue de l'étape précédente, un point de la première image est susceptible d'être apparié avec plusieurs points de la seconde image, et vice versa. Considérons un appariement candidat (m_{1i},m_{2j}). Soit $\mathcal{N}(m)$ les points voisins de m dans un disque de rayon R. Si (m_{1i},m_{2j}) est un appariement correct, nous nous attendons à obtenir plusieurs appariements (n_{1k},n_{2l}), où $n_{1k} \in \mathcal{N}(m_{1i})$ et $n_{2k} \in \mathcal{N}(m_{2i})$, tels que la position de n_{1k} relative à m_{1i} soit à peu près la même que celle de n_{2l} relative à m_{2j}. À l'inverse, si (m_{1i},m_{2j}) est un mauvais appariement, nous nous attendons à trouver peu d'appariements, voire pas du tout, dans leur voisinage. Le *score d'appariement* utilisé pour la relaxation, $SM(m_{1i},m_{2j})$ (SM pour *Strenght of the Match*), repose sur ce constat. Son expression exacte, reportée dans [Zhang et al.94], est relativement complexe. Pour simplifier, $SM(m_{1i},m_{2j})$ est égal au score de corrélation c_{ij}, multiplié par le nombre d'appariements voisins dont les positions relatives sont identiques, c'est-à-dire dont le rapport

$$r = \frac{|d(m_{1i},n_{1k}) - d(m_{2j},n_{2l})|}{[d(m_{1i},n_{1k}) + d(m_{2j},n_{2l})]/2}$$

est petit. SM est symétrique, c'est-à-dire que $SM(m_{1i},m_{2j}) = SM(m_{2i},m_{1j})$.

Élimination des ambiguités par relaxation

La technique de relaxation utilisée rompt avec les techniques classiques du "*vainqueur prend tout*", qui aboutit fréquemment à un minimum local, ou du "*perdant ne prend rien*", qui n'est pas symétrique. La technique du "*vainqueur prend tout*" consiste à considérer immédiatement comme corrects les appariements les plus plausibles, c'est-à-dire les couples (m_{1i},m_{2j}) dont les point m_{1i} ou m_{2j} n'obtiennent pas de score d'appariement plus élevé avec n'importe quel autre couple plausible qu'ils peuvent former. Tous les appariements associés aux points m_{1i} et m_{2j} sont alors éliminés, et le processus est recommencé avec les appariements non éliminés. La technique du "*perdant ne prend rien*" revient à éliminer à chaque étape le point de l'image 1 qui obtient le score d'appariement le plus faible, jusqu'à ce qu'il ne reste plus que un et un seul candidat pour chaque point.

La technique mise en œuvre peut être appelée "*certains vainqueurs prennent tout*". Tout comme pour la technique du "*vainqueur prend tout*", les p appariements $\{\mathcal{P}_i\}$ obtenant les scores SM les plus élevés sont qualifiés d'*appariements potentiels*, et rangés par ordre décroissant dans une table appelée $\mathcal{T}_{SM}$. Cependant, certains appariements peuvent obtenir un score d'appariement SM élevé, tout en étant ambigus. Une deuxième table $\mathcal{T}_{NA}$ est donc crée, contenant des valeurs NA elles aussi classées par ordre décroissant, et indiquant dans quelle mesure chaque appariement $\mathcal{P}_i$ est non ambigu : $NA = 1 - SM^{(2)}/SM^{(1)}$, où $SM^{(1)}$ est le score SM de $\mathcal{P}_i$ et $SM^{(2)}$ est le score SM du deuxième meilleur appariement. Pour finir, les appariements potentiels $\mathcal{P}_i$ appartenant à la fois aux qp premiers appariements de $\mathcal{T}_{SM}$, où $q \in]0;1]$, et aux qp premiers appariements de $\mathcal{T}_{NA}$ sont considérés comme corrects. Ainsi, les appariements potentiels ambigus ne sont pas sélectionnés même s'ils ont obtenu un score SM élevé, et inversement

les appariements ayant obtenu un score SM petit ne sont pas sélectionnés, même s'ils sont non ambigus.

6.2.3 La boucle de recalage temporel, version 3

La nouvelle boucle de recalage temporel est peu différente de la version 2 : il suffit d'ajouter le calcul des appariements 2-D/2-D entre l'image courante et l'image précédent de la séquence, et de remplacer l'optimisation 3-D/2-D par l'optimisation hybride, y compris pour la mise à jour des primitives (figure 6.2).

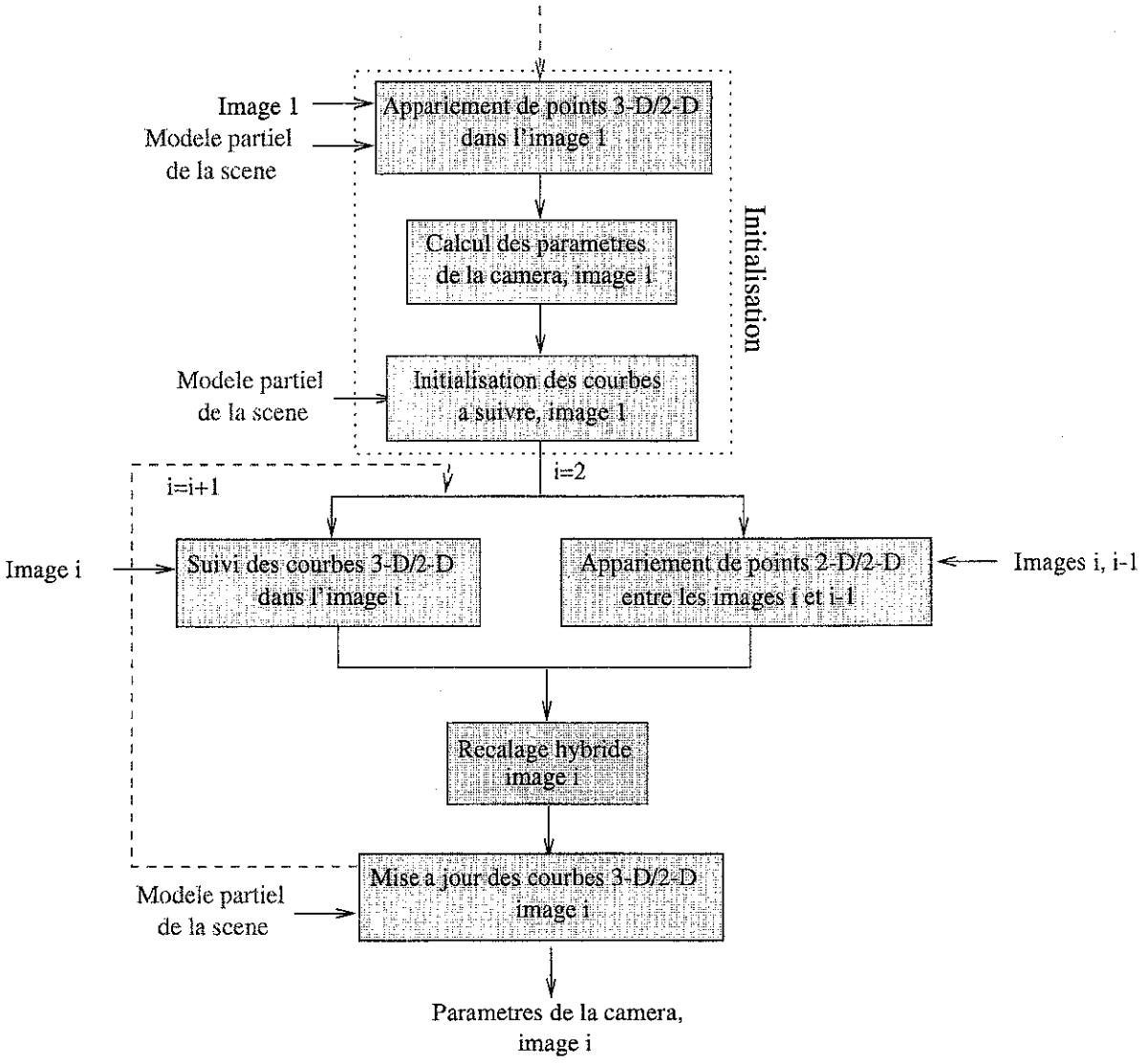


FIG. 6.2 – *Boucle de recalage temporel pour la méthode hybride.*

6.3 Résultats expérimentaux

6.3.1 La séquence de la place Stanislas

La séquence de la place Stanislas a déjà été présentée au chapitre 5. Cette séquence, qui comporte 80 images, a été filmée par la fenêtre d'une automobile tournant autour de la place. L'Opéra, qui est le seul édifice dont le modèle est connu (figure 5.25), est toujours visible dans la séquence. Les paramètres intrinsèques de la caméra ont été obtenus à partir d'une mire de calibration. Comme nous l'avons vu, le recalage basé uniquement sur les primitives de l'Opéra (figure 5.26) donne lieu à des points de vue incorrects, dont les effets sur la projection dans le plan image sont particulièrement dramatiques lorsque l'objet virtuel est incrusté dans une zone éloignée de l'opéra.

Les courbes suivies pour la méthode hybride sont toujours celles de la figure 5.26. La figure 6.3 montre les points d'intérêts que nous avons obtenus par la méthode de Harris et Stephens dans les deux premières images de la séquence, ainsi que les appariements obtenus par la méthode de Zhang et al. Nous voyons que la statue de Stanislas ainsi que les édifices du fond apportent une information de profondeur tout à fait significative. Les séquences complètes de suivi des primitives du modèle et d'appariement des points entre images consécutives sont visibles sur le site internet.

Influence du pas entre les images

Pour cette séquence, nous avons bien sûr pris en compte le cas général de mouvement. Cependant, lorsque deux positions consécutives de la caméra sont trop rapprochées, les résultats deviennent instables puisque nous tendons vers le cas dégénéré d'une rotation pure. Le problème se pose pour cette séquence lorsque 24 images par seconde sont prises en compte, c'est-à-dire lorsque le pas de la boucle de recalage temporel est de un. La figure 6.4 présente la valeur de t_z obtenue sur toute la séquence, pour un pas de un, de deux et de trois. L'évolution du paramètre t_z est un bon critère de comparaison : en effet, nous savons que la hauteur de caméra doit être quasiment constante sur la séquence, puisque la route depuis laquelle la scène a été filmée est plate. Nous voyons que pour un pas de un, t_z est très instable et varie entre -30cm et 1m80. Pour un pas de deux, t_z est à peu près constant, de moyenne 65cm (à peu près la hauteur de la fenêtre de la voiture), et d'écart-type 5cm. Par contre, nous voyons qu'un pas de trois n'améliore pas les résultats (la moyenne de t_z est alors de 67cm pour un écart type de 7cm). Les résultats présentés plus bas ont été obtenus avec un pas de deux, les points de vue intermédiaires pour la composition ayant été obtenus par une simple interpolation linéaire sur les angles d'Euler et le vecteur de translation. Dans un contexte temps réel, une telle interpolation n'est évidemment pas possible. Un pas de deux revient dans ce cas à opérer un rafraichissement de 12 images par secondes au lieu de 24.

Résultats obtenus pour un pas de deux

La figure 6.5 présente les résultats statistiques obtenus sur la séquence pour un pas de deux. L'erreur de reprojection est en moyenne plus élevée que celle que nous obtenions dans les séquences précédentes : ceci s'explique naturellement par le fait que le résidu robuste n'est plus le seul critère minimisé, puisque la distance des points 2-D aux épipolaires est aussi considérée. Les paramètres t_x et t_y évoluent de façon cohérente avec le mouvement de l'automobile supportant la caméra, qui tourne légèrement autour de la place. t_z est à peu près constant comme nous l'avons constaté plus haut. L'évolution de l'angle α traduit le fait que la caméra garde l'opéra dans sa

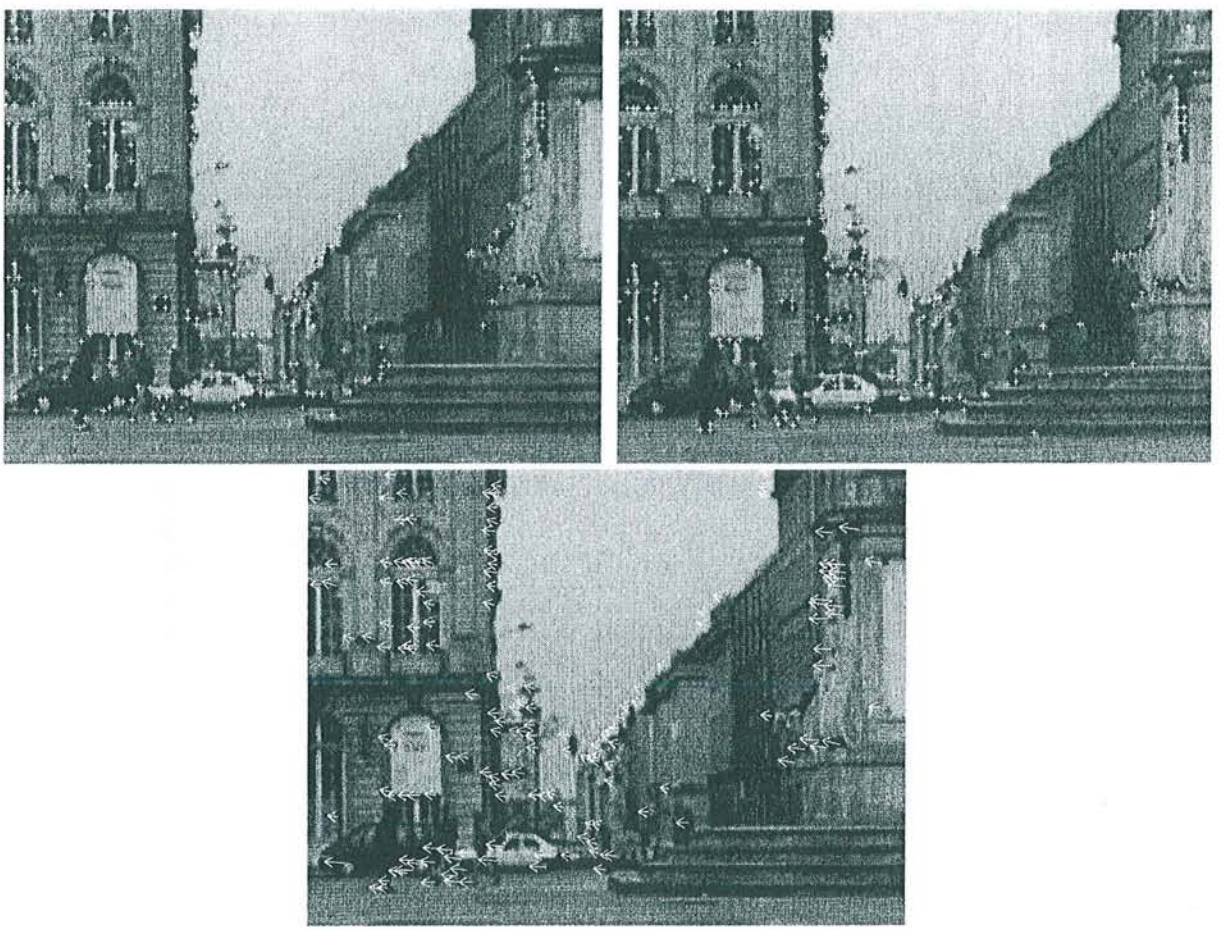


FIG. 6.3 – *Exemple de points d'intérêt et d'appariements obtenus pour les deux premières images de la séquence (pour plus de visibilité, seule une partie de l'image est représentée). Les flèches relient les point d'intérêt de l'image 1, qui est affichée, aux points d'intérêt correspondant dans l'image 2.*

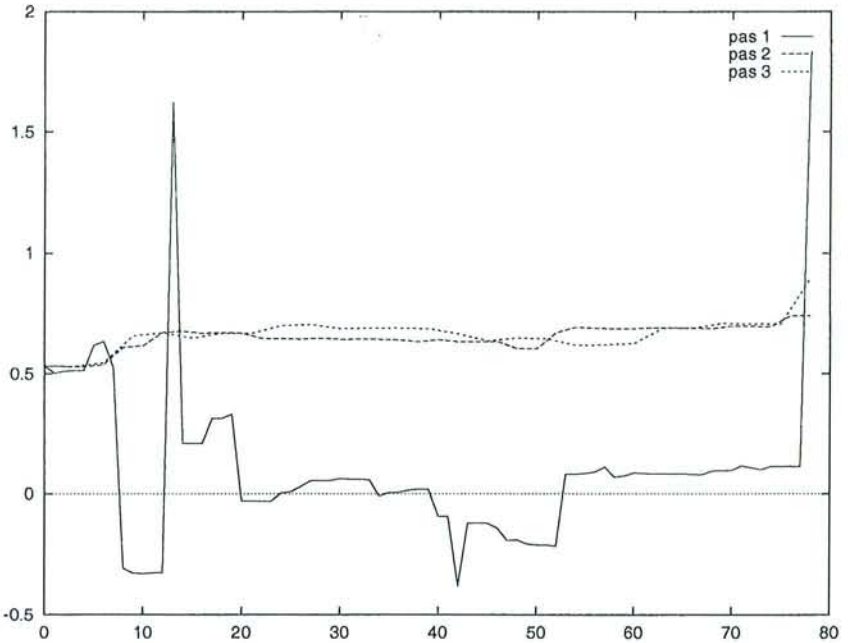


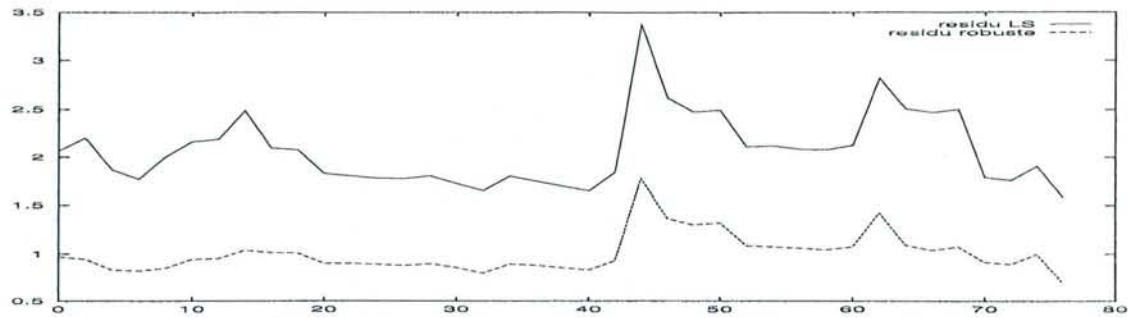
FIG. 6.4 – *Influence du pas entre les images sur t_z pour la séquence de la place Stanislas.*

ligne de visée alors que la voiture avance. Les angles β et γ sont à peu près constants puisque la différence entre leur valeur maximale et minimale sur toute la séquence est d'environ 1 degré. La cohérence des points de vue peut être aussi évaluée visuellement avec la figure 7.27. Le nombre de primitives suivies est à peu près constant (entre 14 et 15), puisque l'Opéra n'est quasiment pas occulté et que les primitives mal suivies sont correctement réinitialisées.

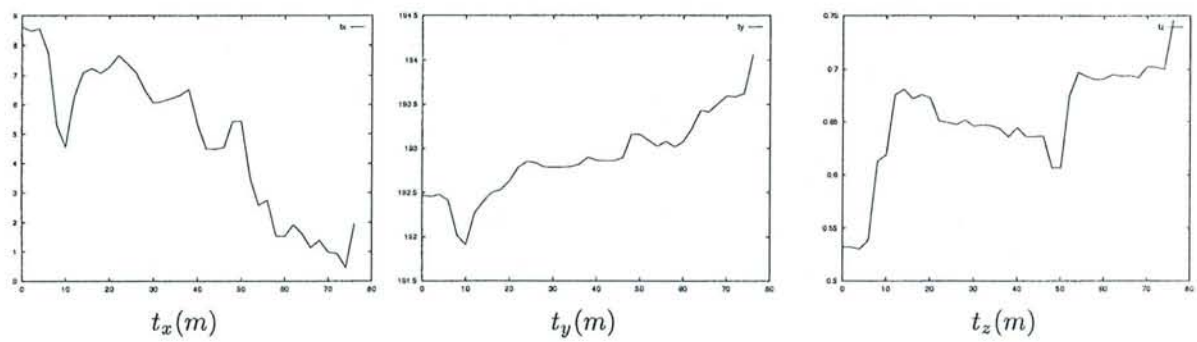
Enfin, la figure 6.7 présente les droites épipolaires obtenues dans l'image 2 pour quelques points de profondeurs différentes de l'image 1. Nous voyons que ces droites passent bien par les points homologues dans l'image 2, ce qui atteste de la bonne qualité du point de vue pour ces deux images.

Composition

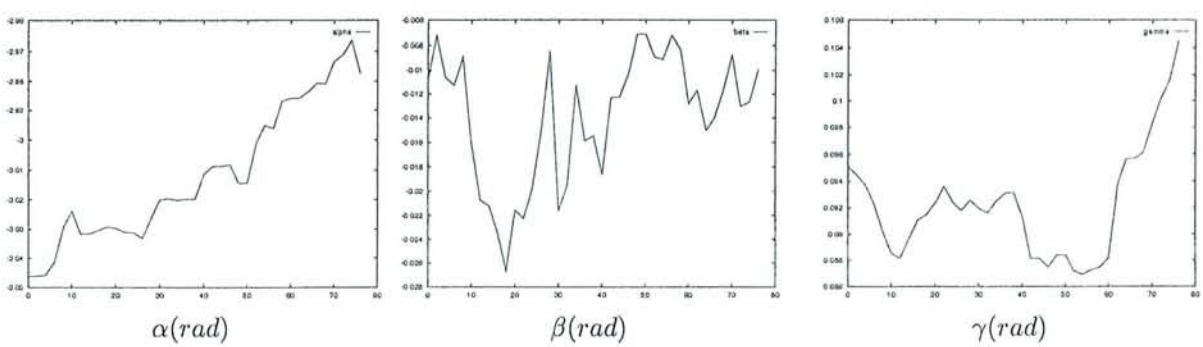
La figure 6.8 montre le résultat de la projection, dans plusieurs images de la séquence, d'une voiture virtuelle circulant autour de la satue de Stanislas, et donc à une distance éloignée de l'Opera. La séquence complète est visible sur le internet. On constate que la voiture semble bien reposer sur le sol, et qu'aucun effet de tremblement n'est perceptible. Les occultations de l'automobile par la statue ont été détectées de façon semi-automatique par un algorithme nouveau, développé dans notre équipe par Vincent Lepetit : le principe de cet algorithme est de détourer à la main le masque d'occultation dans deux images distantes de la séquence. Comme les paramètres de la caméra sont connus pour chaque image, le contour occultant peut alors être approximé en 3-D, puis reprojeté dans les autres images de la séquence. Ce contour approximatif est ensuite affiné par une technique de type *snake*. L'ombre de l'automobile sur le sol a été obtenue en ajoutant un plan virtuel en $z = 0$ pour la composition, et en combinant l'ombre calculée avec la texture de l'image réelle.



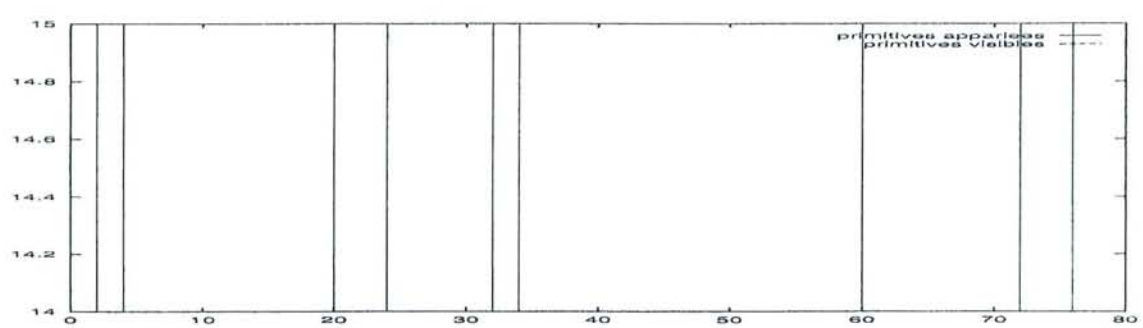
Résidus aux moindres carrés et résidus robustes.



$t_x(m)$ $t_y(m)$ $t_z(m)$



$\alpha(rad)$ $\beta(rad)$ $\gamma(rad)$



Nombre de primitives visibles et appariées.

FIG. 6.5 – *Résultats statistiques obtenus sur la séquence de la place Stanislas.*

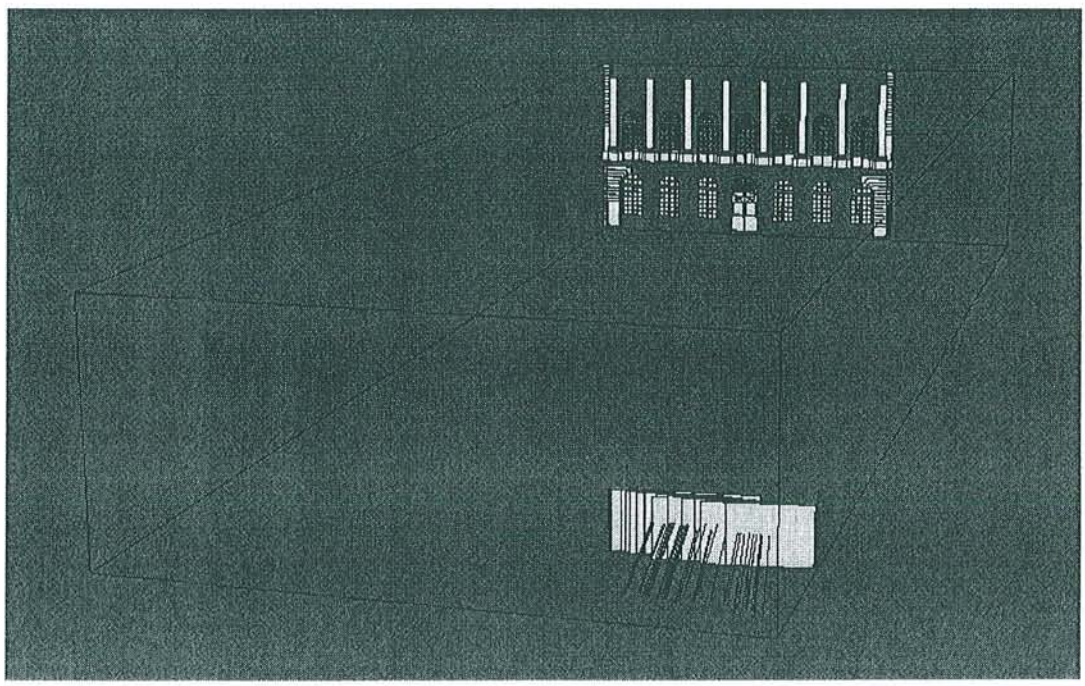


FIG. 6.6 – *Trajectoire 3-D retrouvée pour la séquence de la place Stanislas.*

FIG. 6.7 – *Droites épipolaires correspondant à quelques points de profondeurs différentes.*

Temps de calcul

Les temps de calcul dépendent du nombre de primitives suivies, de leur taille et du nombre de points appariés. La table 6.1 donne un exemple de résultats obtenus pour les différentes étapes de l'algorithme. Ces valeurs ont été obtenues à partir du logiciel *Quantify* pour un processeur de 500Mhz, 15 primitives suivies et 600 points appariés. Plus généralement, les temps de calcul sont de l'ordre de quelques secondes par image. Bien sûr, en cas d'initialisation ou réinitialisation de primitives, il faut ajouter le temps d'optimisation hybride multiplié par le nombre de primitives mises à jour et le nombre de chaînes de contour testées par mise à jour (voir section 5.3.3).

Suivi des courbes 3-D/2-D	.88 s
Appariements des points 2-D/2-D	1.13s
Optimisation hybride	1.38 s

TAB. 6.1 – *Exemple de temps de calcul obtenus pour les différentes étapes de l'algorithme.*

6.3.2 La séquence du Pont Neuf

Le cas des rotations pures a pu être expérimenté sur la séquence du Pont-Neuf, puisque celle-ci a été filmée à partir une caméra posée sur un trépied. Les résultats se sont avérés convainquants : nous avons enfin pu venir à bout des 300 images de la séquence par une méthode autonome. Les points de vue obtenus s'avèrent particulièrement cohérents : la figure 6.9 montre l'évolution de l'angle α, qui traduit la rotation de la caméra autour de l'axe du trépied. Les écarts-types des autres paramètres, supposés constants pour le mouvement panoramique, sont indiqués en table 6.2 : on constate que ces écart-types sont plus faibles que pour la méthode précédente (table 5.2). Enfin, la représentation 3-D du mouvement présentée en figure 6.10, montre un faisceau d'axes optiques bien plus précis qu'avec l'ancienne méthode (figure 5.17).

Paramètre	Moyenne	Écart type
t_x (m)	-90.506	0.706
t_y (m)	326.336	2.377
t_z (m)	13.658	0.101
β (rad)	0.008	0.003
γ (rad)	-0.002	0.000

TAB. 6.2 – *Moyenne et écart-type des paramètres fixes de la caméra sur la séquence du Pont-Neuf.*

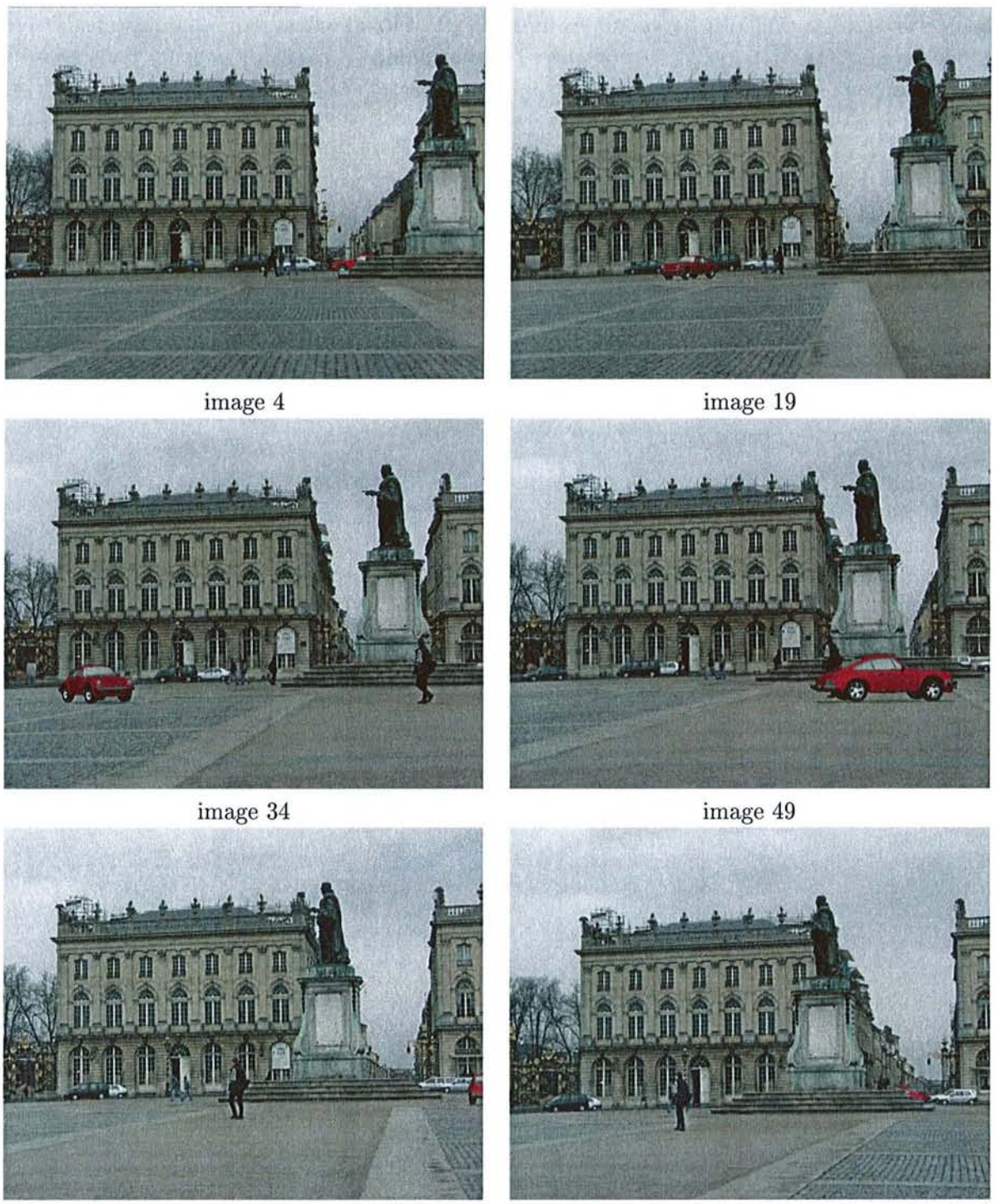


image 4 image 19

image 34 image 49

image 64 image 78

FIG. 6.8 – *Incrustation d'une automobile sur la place Stanislas.*

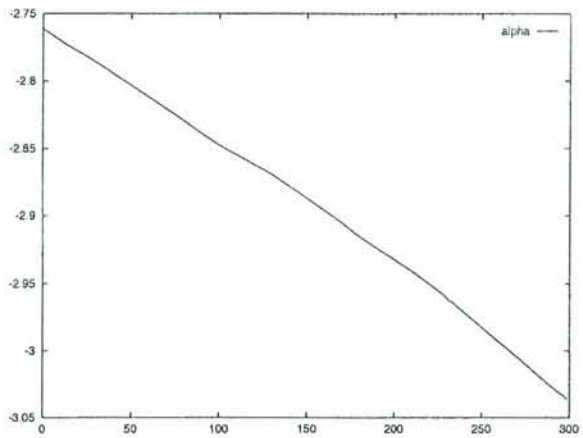


FIG. 6.9 – *Évolution de l'angle α sur la séquence du Pont-Neuf.*

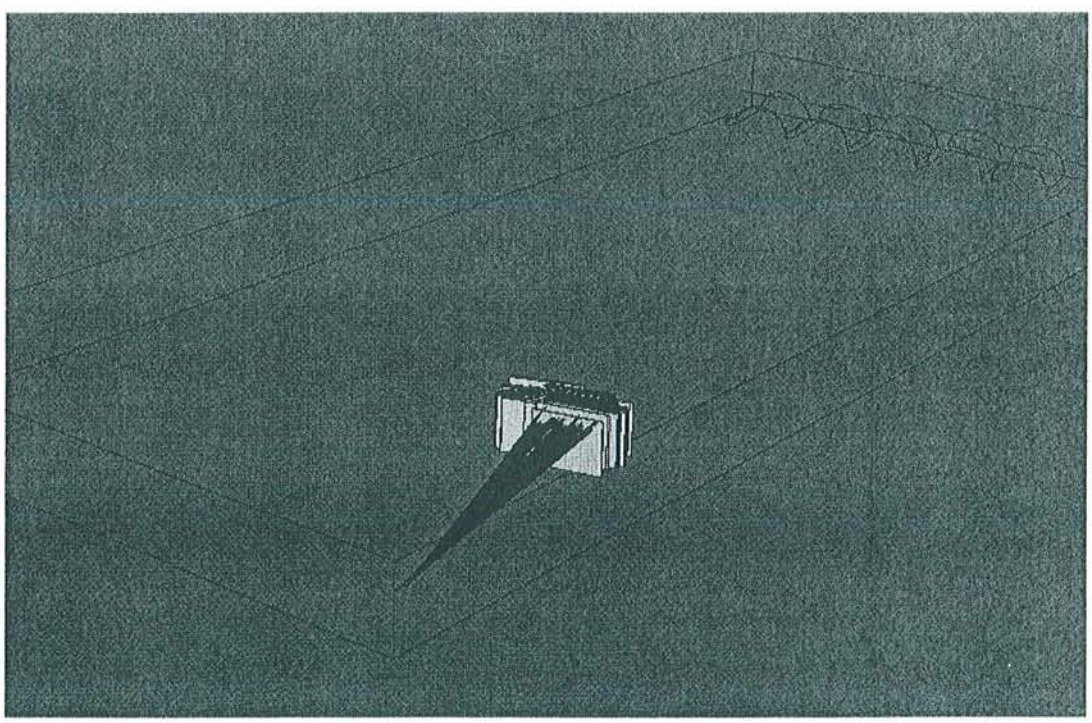


FIG. 6.10 – *Trajectoire 3-D retrouvée pour la séquence du Pont-Neuf.*

Chapitre 7

Prise en compte de changements de focale

Nous nous penchons à présent sur le cas des séquences où la focale de la caméra est susceptible de varier en cours de prise de vue. Ce chapitre concerne donc uniquement les applications de post-production puisque les HMD ne sont pas équipés de zoom. Il s'agit d'un problème délicat : plusieurs auteurs ont observé que la détermination simultanée des paramètres intrinsèques et extrinsèques de la caméra pouvait conduire à des résultats biaisés. Nous constatons ce problème à travers nos expérimentations, puis nous plaçons dans l'hypothèse où zoom et point de vue de la caméra ne varient pas en même temps. Le problème se résume alors à discriminer un zoom d'un déplacement de caméra. Nous supposons dans ce chapitre que la scène est rigide, sans objet mobile.

7.1 Optimisation à neuf paramètres

La solution la plus immédiate lorsqu'on cherche à prendre en compte des variations de focale consiste à minimiser l'erreur de reprojection, non plus uniquement en fonction des six paramètres du mouvement, mais aussi en fonction des quatre paramètres intrinsèques de la caméra α_u, α_v, u_0 et v_0. En pratique, il s'avère que le rapport α_u/α_v ne varie pas lors d'un changement de focale [Enciso et al.93] : une minimisation à neuf paramètres (les six paramètres du mouvement, α_u, u_0 et v_0) est donc suffisante.

Cette solution a cependant été critiquée dans le cadre de l'autocalibration [Bougnoux97, Oliensis97]. En particulier, Bougnoux constate que dans la pratique, les paramètres intrinsèques et la translation sont hautement corrélés : une erreur sur la position du point principal peut être compensée par une translation de la caméra, et surtout il existe une forte ambiguïté entre un zoom et une translation le long de l'axe optique.

Cette ambiguïté s'explique de la façon suivante : soit n points 3-D $M_i = (X_i,Y_i,Z_i)$ exprimés dans le repère de la caméra. D'après les équations 2.2 et 2.3, la projection d'un point M_i dans le plan image a pour coordonnées pixel :

$$\begin{cases} u_i = k_u f \frac{X_i}{Z_i} + u_0, \\ v_i = k_v f \frac{Y_i}{Z_i} + v_0. \end{cases}$$

Si nous appliquons un changement de focale Δf, nous obtenons les nouvelles coordonnées pixel

$$\begin{cases} u'_i = u_i + (u_i - u_0)\frac{\Delta f}{f}, \\ v'_i = v_i + (v_i - v_0)\frac{\Delta f}{f}. \end{cases} \tag{7.1}$$

D'un autre côté, si nous appliquons une translation le long de l'axe optique $\mathbf{t} = (0,0,\Delta t_z)$ à la caméra, nous obtenons les coordonnées pixel

$$\begin{cases} u'_i = k_u f \frac{X_i}{Z_i - \Delta t_z} + u_0, \\ v'_i = k_v f \frac{Y_i}{Z_i - \Delta t_z} + v_0. \end{cases}$$

Si la translation opérée est petite par rapport à la profondeur du point, c'est-à-dire $\Delta t_z \ll Z_i$, cette dernière équation devient :

$$\begin{cases} u'_i \approx u_i + (u_i - u_0)\frac{\Delta t_z}{Z_i}, \\ v'_i \approx v_i + (v_i - v_0)\frac{\Delta t_z}{Z_i}. \end{cases} \tag{7.2}$$

Les équations 7.1 et 7.2 sont alors toutes les deux de la forme :

$$\begin{pmatrix} u'_i \\ v'_i \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix} + k \begin{pmatrix} u_i - u_0 \\ v_i - v_0 \end{pmatrix},$$

où $k = \Delta f / f$ dans le cas du zoom et $k = k_t = \Delta t_z / Z_i$ dans le cas de la translation. k_t dépend de la coordonnée en z du point M_i, mais si l'épaisseur de l'objet projeté est petite par rapport à la distance objet-caméra, c'est-à-dire s'il existe un réel Z_0 tel que $Z_i = Z_0 + \Delta Z_i$ et $\Delta Z_i \ll Z_0$ pour tout i, alors la translation peut être interprétée comme un changement de focale et vice versa.

7.1.1 Expérimentation

Nous expérimentons une optimisation à neuf paramètres basée modèle sur une scène de laboratoire : il s'agit d'une cabane miniature et d'une mire de calibration posés sur une table micrométrique, dont les mouvements sont parfaitement contrôlés (figure 7.1). La caméra est munie d'un zoom commandé à partir d'un ordinateur. Ce dispositif nous permet de valider très précisément nos résultats et de comparer les paramètres calculés aux paramètres attendus.

Les commandes envoyées à la table micrométrique et au moteur du zoom sont données en table 7.1. Pour cette séquence, c'est donc la scène qui est en mouvement, et non la caméra. Ce problème dual est traité de manière habituelle, c'est-à-dire que la scène est considérée comme fixe et la caméra comme étant en mouvement.

Les primitives suivies, au nombre de sept, sont visibles en figure 7.1. Pour le recalage, nous utilisons l'algorithme robuste à deux niveaux présenté au chapitre 5, en ajoutant α_u, u_0 et v_0 aux paramètres estimés.

Les projections dans le plan horizontal de la trajectoire calculée de la caméra et de la trajectoire attendue sont présentées en figure 7.2.a. L'évolution de α_u est donnée en figure 7.2.b, et comparée avec celle obtenue par calibration à partir des points de la mire. Notons que nous utilisons la mire comme référence, et non pas les commandes envoyées au moteur du zoom : il s'est avéré en effet que les valeurs effectives du zoom ne répondaient pas de façon linéaire aux commandes envoyées.

Nous voyons en figure 7.2 qu'en plus du fait que certains zooms sont interprétés comme des translations le long de l'axe optique, une translation inattendue est obtenue entre les images 13

Image	Mouvement/zoom
$0 \rightarrow 20$	rotation 40°
$21 \rightarrow 35$	zoom avant
$36 \rightarrow 40$	translation 10cm
$41 \rightarrow 55$	zoom arrière
$56 \rightarrow 65$	rotation $-20°$

TAB. 7.1 – *Les commandes envoyées à la table micrométrique et au moteur du zoom pour la séquence de la cabane.*

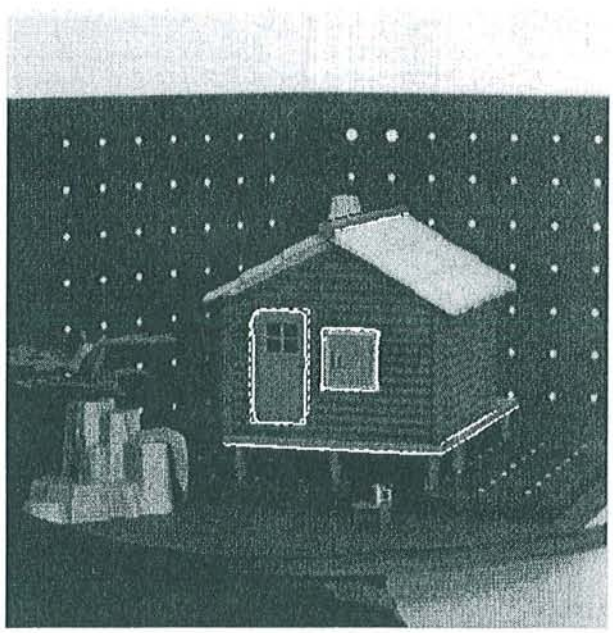


FIG. 7.1 – *Les primitives suivies sur la séquence de la cabane.*

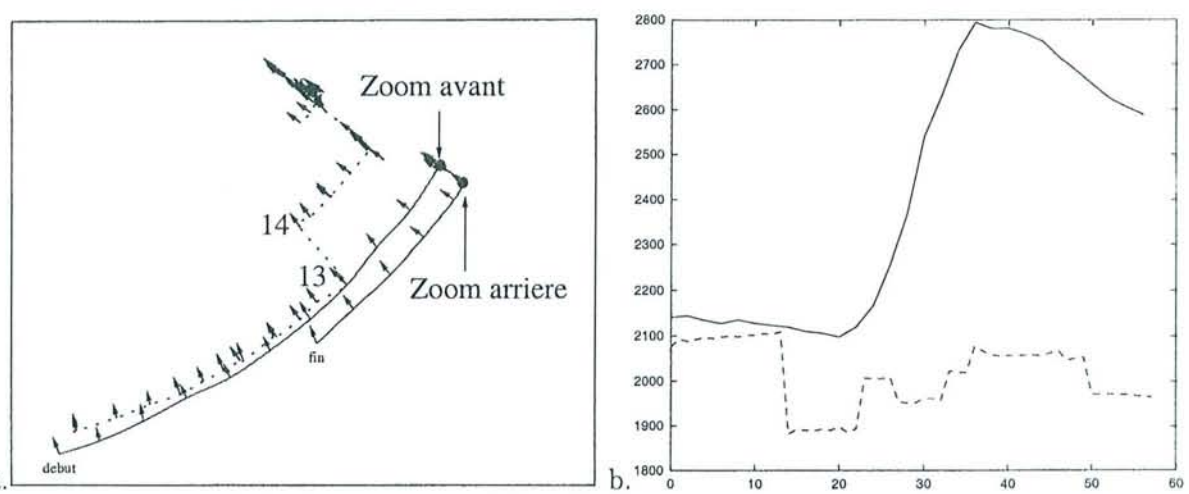


FIG. 7.2 – *Paramètres de la caméra attendus (traits continus) et calculés (traits discontinus) pour une optimisation à neuf paramètres sur la séquence de la cabane. a. Projection de la trajectoire de la caméra dans le plan horizontal. b. Paramètre* α_u.

et 14, celle-ci étant compensée par un zoom arrière tout aussi inattendu. Ces résultats confirment donc l'instabilité d'une optimisation à neuf paramètres.

Dans [Bougnoux97], Bougnoux considère que ces ambiguïtés n'affectent que légèrement les résultats de la reconstruction, et que l'impression visuelle reste satisfaisante. Nous avons reconstruit les points de la mire de calibration visibles dans la séquence, à partir de leur détection sub-pixel et leur appariement entre les images 0 et 20. Les points reconstruits sont présentés en figure 7.3. Leurs projections dans les plans (xy) et (yz) comparées aux projections attendues sont montrées en figure 7.4 (voir la figure 7.3 pour les axes). Le résultat présenté en figure 7.3 peut en effet faire illusion. Par contre, en examinant les résultats de plus près en figure 7.4, on s'aperçoit que les angles et les alignements ne sont pas respectés.

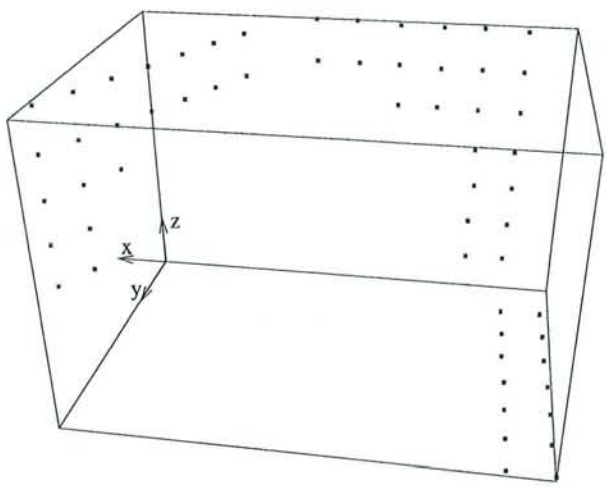


FIG. 7.3 – *Les points de la mire reconstruits à partir des images 0 et 20 pour l'optimisation à neuf paramètres.*

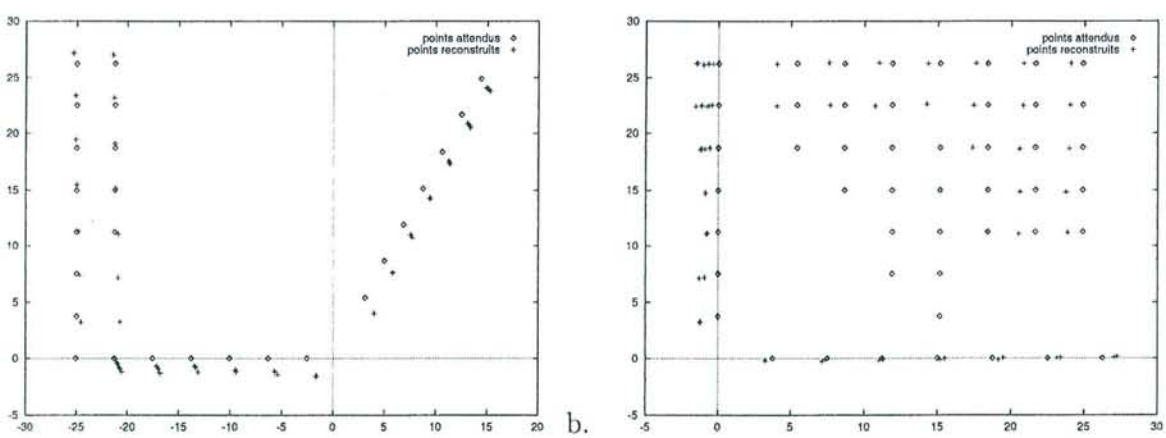


FIG. 7.4 – *Projection des points reconstruits dans le plan (xy) (a) et (yz) (b). Les croix sont les points reconstruits et les losanges les points attendus.*

Tout comme les paramètres obtenus peuvent être satisfaisants pour certaines applications de reconstruction, ceux-ci nous permettent-ils d'obtenir un résultat de composition convaincant sur la séquence? Comme nous l'avons mentionné au chapitre 2, notre but n'est effectivement pas nécessairement d'obtenir les paramètres exacts de la caméra, mais un résultat de composition qui

soit visuellement satisfaisant. Pour le vérifier, nous avons incrusté des objets virtuel à différents endroits de la scène. Pour chaque image isolée, ces objets semblent être à l'endroit où nous souhaitons les voir apparaître. Par contre, en visionnant la séquence (disponible sur le site internet), nous constatons sur la fin des effets de glissement et de saut des objets virtuels d'un endroit à l'autre de la scène. Cependant, la variation de t_z compensée par un zoom arrière entre les images 13 et 14 n'est quasiment pas perceptible (figure 7.5). Cette méthode peut donc convenir dans certains cas, mais elle n'est pas fiable.

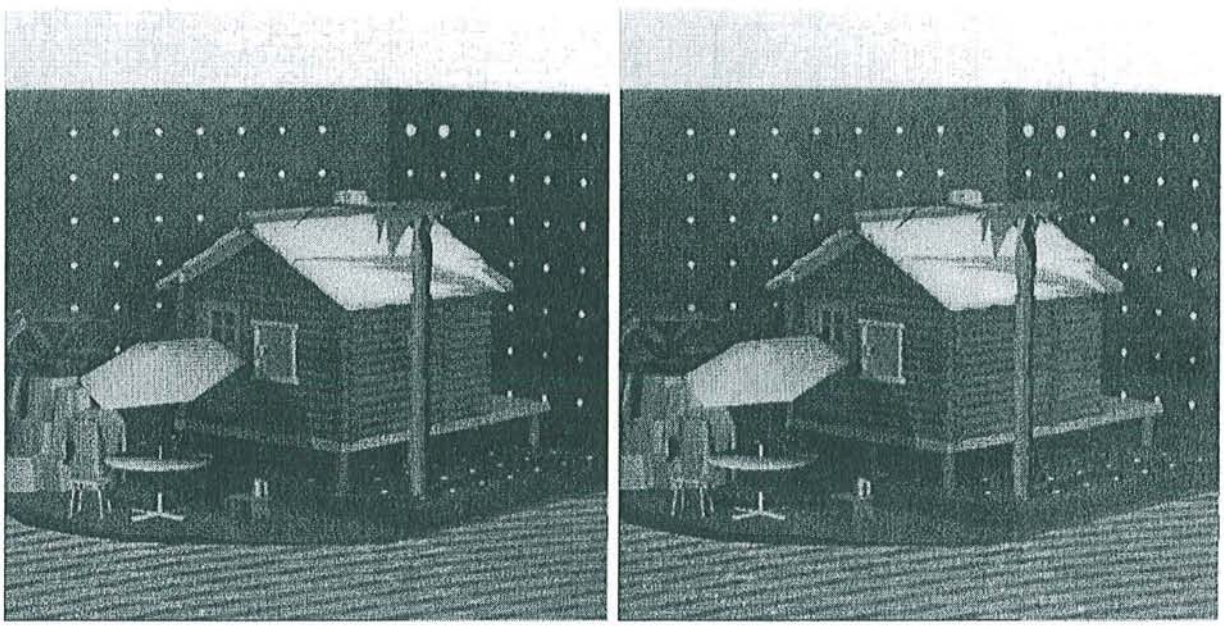


FIG. 7.5 – *Projection d'objets virtuels (la table, la chaise, l'eau et le palmier) dans les images 13 et 14 de la séquence de la cabane pour l'optimisation à 9 paramètres. La confusion entre la translation le long de l'axe optique et le zoom n'est quasiment pas perceptible.*

Il paraît donc difficile d'obtenir une méthode fiable en partant d'une optimisation à neuf paramètres. Sturm réduit le nombre de paramètres de la caméra en exprimant u_0 et v_0 comme une fonction polynomiale de α_u [Sturm96]. Une étape de pré-calibration de la caméra est cependant nécessaire pour déterminer les coefficients des fonctions polynomiales. D'autre part, cela ne résout pas le problème de l'ambiguïté entre la focale et la translation en z.

Par contre, nous pouvons obtenir de meilleurs résultats en considérant que les paramètres intrinsèques et extrinsèques de la caméra ne varient pas simultanément. Cette hypothèse est tout à fait raisonnable : la plupart des réalisateurs de films travaillent en effet par plans de séquence, alternant par exemple zooms et mouvements panoramiques. Les travaux portant sur l'annotation vidéo ne considèrent d'ailleurs que des séquence pouvant être découpées en plans à mouvements dominants. Sous l'hypothèse de rigidité de la scène, une segmentation au sens du mouvement est opérée à partir du flux optique [Sudhir et al.97, Xiong et al.98].

7.1.2 Les techniques d'annotation vidéo

En théorie, les algorithmes d'annotation vidéo sont capables d'établir une classification entre lacets (rotations autour de l'axe vertical), roulis (rotations autour de l'axe de visée z), tangages (rotations autour de l'axe horizontal), translations horizontales, translations verticales, translations en z et zooms.

Par exemple dans [Xiong et al.98], l'image est découpée en sept régions distinctes (figure 7.6). Le flux optique (u,v) est calculé pour chaque pixel de l'image, et le type de mouvement est déterminé en fonction de la moyenne et de l'écart type de u et v dans les différentes régions (la région centrale n'est pas utilisée) : par exemple, si la moyenne de v est nulle en I, II, III et IV, il peut s'agir d'un mouvement panoramique ou d'une translation horizontale. Les valeurs de l'écart-type de u en I, II, III et IV sont alors utilisées pour différencier les deux cas : si les écarts types sont grands, il s'agit d'un panoramique, sinon il s'agit d'une translation horizontale.

La translation en z et le zoom se caractérisent tous deux par $u(X_0) = 0$ (la moyenne de u est nulle en X_0), $v(Y_0) = 0$, $u(Y_0) \neq 0$ et $v(X_0) \neq 0$. Pour les distinguer, les auteurs utilisent le fait que t_z dépend de la profondeur des points : ainsi, l'écart-type de u et celui de v calculés sur l'union des zones I, II, III, IV doit être plus grand pour une translation en z que pour un zoom. En théorie, cet écart-type devrait effectivement être nul dans le cas du zoom, à condition que le foyer d'expansion (le point principal) soit exactement au centre de l'image. Le problème est que cela est rarement le cas dans la pratique [Willson et al.93].

Plus généralement, les auteurs n'indiquent pas ce qu'ils entendent par moyenne "nulle" et écart-type "grand". Nous ne savons pas quels seuils sont utilisés ni comment ils ont été choisis, alors qu'il s'agit d'un problème majeur. Dans la pratique, les erreurs d'interprétation sont d'ailleurs relativement fréquentes (voir par exemple les taux d'échecs dans [Xiong et al.98]).

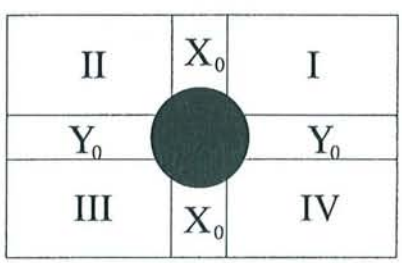


FIG. 7.6 – *Découpage de l'image en sept zones pour la méthode de Xiong et Lee.*

7.2 Détection des changements de focale

Les algorithmes de segmentation vidéo sont basés sur le flux optique, qui est dense mais peu précis. Nous allons au contraire utiliser des appariements de points d'intérêt, moins denses mais plus précis.

D'autre part, nous n'avons pas à discriminer entre un zoom et différents types de mouvements, mais uniquement entre un zoom et un mouvement quelconque de caméra. Nous allons donc exploiter le fait qu'un zoom opère une transformation de l'image qui, comme nous le rappelons en 7.2.1, peut être modélisée par une transformation affine à trois paramètres (*TATP*). Nous montrons qu'un mouvement quelconque de caméra ne peut généralement pas être approximé par une TATP, sauf sous certaines conditions que nous identifions en 7.2.2. Il nous suffit donc de vérifier si l'image se transforme effectivement selon une TATP pour distinguer les deux cas de figure (7.2.3).

7.2.1 Approximation du zoom par un modèle affine à trois paramètres

Considérons un changement de focale $f' = f + \Delta f$. D'après l'équation 7.1, un point (u_i,v_i) se transforme en un point (u'_i,v'_i) par la relation

$$\begin{cases} u'_i = u_i \frac{f'}{f} - u_0 \frac{f'-f}{f}, \\ v'_i = v_i \frac{f'}{f} - v_0 \frac{f'-f}{f}. \end{cases}$$

En notant $C_0 = \frac{f'}{f}$, nous avons donc :

$$\begin{pmatrix} u'_i - u_0 \\ v'_i - v_0 \end{pmatrix} = C_0 \begin{pmatrix} u_i - u_0 \\ v_i - v_0 \end{pmatrix}.$$

En théorie, un zoom opère donc une homothétie de l'image, de centre (u_0,v_0) et de rapport C_0. Cependant, en écrivant l'équation 7.1, nous supposons que k_u, k_v, u_0 et v_0 sont invariants par changement de focale. Enciso et Viéville ont montré expérimentalement que k_u et k_v pouvaient effectivement être considérés comme constants au cours d'un zoom [Enciso et al.93]. Par contre, il s'avère que le point principal ne reste pas fixe par changement de focale, mais varie linéairement dans l'image. Ce phénomène, non prévu par le modèle initial, a été étudié très précisément par Willson et Shafer [Willson et al.93], et vérifié expérimentalement par Enciso et Viéville. Le modèle de l'homothétie n'est donc pas suffisant pour décrire la transformation de l'image par changement de focale : nous devons utiliser un modèle affine à trois paramètres C_0, a_0 et b_0 :

$$\begin{cases} u'_i = C_0 u_i + a_0, \\ v'_i = C_0 v_i + b_0, \end{cases}$$

La matrice des paramètres intrinsèques A' obtenue lors d'une variation de la focale se déduit donc de la matrice précédente A par l'équation :

$$A' = \begin{pmatrix} C_0 & 0 & a_0 \\ 0 & C_0 & b_0 \\ 0 & 0 & 1 \end{pmatrix} A. \tag{7.3}$$

Ce modèle a été validé expérimentalement par Enciso et Viéville. Il s'est avéré très pertinent dans nos expérimentations, décrivant avec précision les transformations de l'image obtenues sur nos plans de zoom (cf. 7.4). Ses paramètres peuvent être déterminés à partir de deux appariements de points (u_i,v_i) et (u'_i,v'_i), mais, bien sûr, la prise en compte de n points par une minimisation aux moindres carrés permet généralement d'obtenir un résultat plus précis.

7.2.2 Le cas d'un mouvement quelconque de caméra

Si nous voulons utiliser une mesure d'adéquation des données image au modèle affine à trois paramètres comme critère de discrimination entre un zoom et un mouvement de caméra, nous devons nous assurer qu'aucun mouvement de caméra ne peut donner lieu à une TATP de l'image. Pour cela, nous nous basons sur les équations du flux optique, qui décrivent la vitesse apparente de la projection des points d'une scène rigide sur le plan image lorsque la caméra est en mouvement.

Soit $\mathbf{T} = (U \quad V \quad W)^T$ la composante translationelle du mouvement de la caméra, et $\Omega = (A \quad B \quad C)^T$ son accélération instantanée. Soit $\mathbf{M} = (X \quad Y \quad Z)^T$ un point de la scène exprimé dans le repère de la caméra, et $\mathbf{m} = (x \quad y \quad z)^T$ sa projection dans le plan image. La vitesse de $\mathbf{M}$ par rapport au repère de la caméra est donnée par l'équation :

$$\dot{\mathbf{M}} = \mathbf{T} + \Omega \wedge \mathbf{M}$$

[Horn et al.81], c'est-à-dire :

$$\begin{cases} \dot{X} &= -U - BZ + CY, \\ \dot{Y} &= -V - CX + AZ, \\ \dot{Z} &= -W - AY + BX. \end{cases} \tag{7.4}$$

D'après l'équation 2.2, nous avons $x = fX/Z$ et $y = fY/Z$. Pour simplifier les équations, nous pouvons poser $f = 1$ sans que cela n'ait de conséquence sur l'interprétation des équations, f étant supposé constant lorsque la caméra se déplace. Le flux optique, ou vitesse apparente de $\mathbf{m}$ dans le plan image, est alors donné par les équations :

$$\begin{cases} \dot{x} = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2}, \\ \dot{y} = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2}, \end{cases}$$

soit, en remplaçant $\dot{X}$, $\dot{Y}$ et $\dot{Z}$ par leurs expressions (7.4) et X/Z et Y/Z par (respectivement) x et y :

$$\begin{cases} \dot{x} = -\frac{U}{Z} + x\frac{W}{Z} + Axy - B(x^2+1) + Cy, \\ \dot{y} = -\frac{V}{Z} + y\frac{W}{Z} + A(y^2+1) - Bxy - Cx. \end{cases}$$

Ce sont les équations du flux optique, largement utilisées en vision par ordinateur.

Un mouvement de caméra est qualifié d'*ambigu* vis-à-vis du zoom si sa projection dans le plan image peut être interprétée comme une TATP. Nous allons rechercher les ambiguïtés possibles sur chacun des déplacements élémentaires : translation horizontale T_x, translation verticale T_y, translation en z T_z, tangage R_x, lacet R_y et roulis R_z. Les flux optiques obtenus pour ces déplacements élémentaires sont donnés en table 7.2. Un mouvement quelconque pouvant être décomposé en mouvements élémentaires, nous pouvons nous référer à la table 7.2 pour déterminer s'il s'agit d'un mouvement ambigu ou non, sachant que la composée de deux TATP est une TATP. Nous voyons que théoriquement, aucun de ces mouvements ne correspond à une TATP. Sous certaines conditions, ils peuvent néanmoins être approximés par une TATP, comme nous le voyons à présent.

Translation horizontale et translation verticale

S'il existe un réel Z_0 tel que les coordonnées en z de chaque point de la scène peuvent s'écrire $Z = Z_0 + \Delta Z$ avec $\Delta Z \ll Z_0$ (condition 1), alors pour une translation horizontale, $\dot{x} = \frac{-U}{Z_0}\left(1 - \frac{\Delta Z}{Z_0} + o(\frac{\Delta Z}{Z_0})\right)$. Le mouvement de l'image peut alors être approximé par une transformation affine à trois paramètres $C_0 \approx 1$, $a_0 \approx -k_u\frac{U\Delta t}{Z_0}$, $b_0 \approx 0$ (nous utilisons l'approximation $\dot{x} = \frac{\dot{u}}{k_u} = \frac{u'-u}{k_u\Delta t}$ and $\dot{y} = \frac{v'-v}{k_v\Delta t}$). La condition 1 signifie que l'épaisseur de la scène observée est petite par rapport à la distance scène-caméra.

De même, si la condition 1 est vérifiée pour une translation verticale, nous pouvons approximer le mouvement 2-D par une TATP de paramètres $C_0 \approx 1$, $a_0 \approx 0$, $b_0 \approx -k_v\frac{V\Delta t}{Z_0}$.

Translation en z

Si la condition 1 est vérifiée, nous avons $\dot{x} = x\frac{W}{Z_0}\left(1 - \frac{\Delta Z}{Z_0} + o(\frac{\Delta Z}{Z_0})\right)$ et $\dot{y} = y\frac{W}{Z_0}\left(1 - \frac{\Delta Z}{Z_0} + o(\frac{\Delta Z}{Z_0})\right)$. Le mouvement 2-D peut donc être interprété comme une TATP de paramètres $C_0 \approx 1 + \frac{W\Delta t}{Z_0}$, $a_0 \approx -u_0\frac{W\Delta t}{Z_0}$ et $b_0 \approx -v_0\frac{W\Delta t}{Z_0}$.

Tangage et lacet

Généralement, si $f = 1$, nous avons $x < 1$ et $y < 1$, car l'angle de vue est rarement supérieur à 90°. Plus le rapport taille de l'image sur distance focale est petit (c'est-à-dire plus la focale augmente), plus x et y sont petits par rapport à la focale. Pour un zoom important, nous pouvons donc avoir $x \ll 1$ et $y \ll 1$ (condition 2). Sous cette conditon, nous avons $\dot{x} = o^2(A)$ et $\dot{y} = A + o^2(A)$ et le mouvement peut être interprété comme une TATP, avec $C_0 \approx 1$, $a_0 \approx 0$ et $b_0 \approx A$.

De même, si la condition 2 est vérifiée, le mouvement 2-D induit par un lacet peut être interprété comme une TATP de paramètres $C_0 \approx 1$, $a_0 \approx -B$ et $b_0 \approx 0$.

Roulis

Comme $\dot{x} = Cy$ et $\dot{y} = -Cx$, un roulis ne peut pas être approximé par une TATP.

Ces différents cas sont résumés en table 7.2. Parmi les cinq mouvements élémentaires pouvant être ambigus sous certaines conditions, seul le cas de la translation en z est réellement difficile à discriminer : en effet, pour les quatre autres cas (translations horizontale et verticale, tangage et lacet), C_0 est très proche de 1, avec a_0 ou b_0 non nul. Le centre de la transformation affine, de coordonnées $\left(\frac{a_0}{1-C_0}, \frac{b_0}{1-C_0}\right)$ se trouve donc loin du centre de l'image. À l'inverse, le centre de la transformation affine pour un zoom ou une translation en z est par nature très proche du point principal. Il suffit donc de vérifier si le centre du déplacement affine calculé se trouve ou non à l'intérieur d'un cercle de centre le centre de l'image et de rayon arbitraire pour lever l'ambiguïté (dans la pratique, nous vérifions simplement si le centre de la TATP est à l'intérieur de l'image ou non).

Mouvement	$\dot{x}$	$\dot{y}$	Condition	TATP
T_x	$-\frac{U}{Z}$	0	1	$(1,-k_u\frac{U\Delta t}{Z_0},0)$
T_y	0	$-\frac{V}{Z}$	1	$(1,0,-k_v\frac{V\Delta t}{Z_0})$
T_z	$x\frac{W}{Z}$	$y\frac{W}{Z}$	1	$(1+\frac{W\Delta t}{Z_0},-u_0\frac{W\Delta t}{Z_0},-v_0\frac{W\Delta t}{Z_0})$
R_x	Axy	$A(y^2+1)$	2	$(1,0,A)$
R_y	$-B(x^2+1)$	$-Bxy$	2	$(1,-B,0)$
R_z	Cy	$-Cx$	-	-

TAB. 7.2 – *Flux optique obtenu pour les mouvements élémentaires et TATP possible sous certaines conditions. Les trois paramètres de la TATP sont donnés sous la forme* (C_0,a_0,b_0).

7.2.3 Discrimination entre zoom et mouvement de caméra

Pour différencier les zooms des mouvements de caméra, nous pouvons donc nous baser sur le mouvement 2-D en vérifiant s'il correspond à une TATP ou non. Pour cela, nous procédons en deux étapes :

1. calcul de la TATP qui approxime au mieux le déplacement de points d'intérêt de l'image,

2. validation (ou non) du modèle affine à trois paramètres à l'aide de tests photométriques sur des contours de l'image.

Nous détaillons à présent chacune de ces deux étapes.

Calcul des paramètres de la TATP

Nous cherchons à déterminer quels paramètres de la caméra (intrinsèques ou extrinsèques) ont varié entre deux images I_{i-1} et I_i de la séquence. Pour cela, nous apparions n couples de points d'intérêt $((u_i,v_i);(u'_i,v'_i))$ entre les deux images, en utilisant la méthode de Zhang et al. décrite au chapitre 6. Nous effectuons alors une minimisation aux moindres carrés pour déterminer les trois paramètres C_0, a_0 et b_0 de la TATP qui approxime au mieux la transformation entre ces appariements. Plus exactement, nous minimisons le critère

$$c_1 = \frac{1}{n}\sum_{i=1}^{n}\left(u'_i - (C_0 u_i + a_0)\right)^2 + \left(v'_i - (C_0 v_i + b_0)\right)^2 .$$

En dérivant par rapport aux trois paramètres a_0, b_0 et C_0, nous obtenons une solution directe :

$$\begin{cases} C_0 &= \frac{n\sum u'_i u_u + n\sum v'_i v_i - \sum u'_i \sum u_i - \sum v'_i \sum v_i}{n\sum u_i^2 + n\sum v_i^2 - (\sum u_i)^2 - (\sum v_i)^2}, \\ a_0 &= \frac{\sum u'_i - C_0 \sum u_i}{n}, \\ b_0 &= \frac{\sum v'_i - C_0 \sum v_i}{n}. \end{cases} \tag{7.5}$$

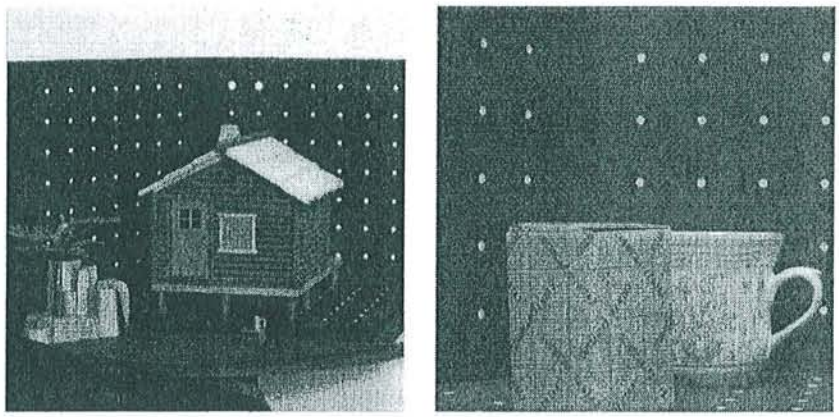
Notons qu'une estimation robuste de ces trois paramètres peut aussi être envisagée si nous soupçonnons des erreurs d'appariements.

Le critère de discrimination

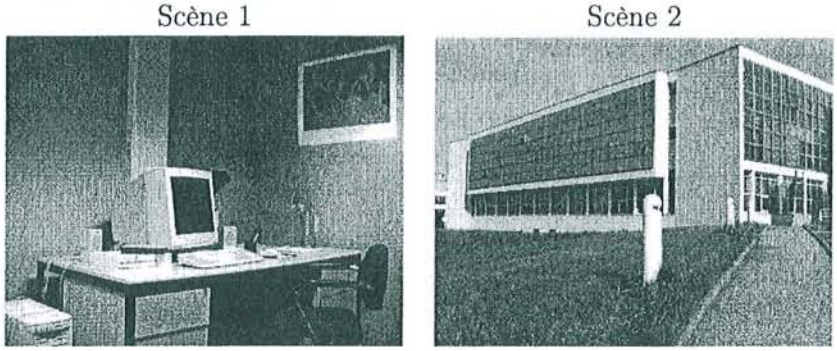
Nous devons à présent valider ou invalider l'hypothèse du zoom. Des tests statistiques tel que le test du χ^2 sont souvent utilisés pour estimer la compatibilité des mesures avec le modèle à un degré de confiance donné. Cependant, l'écart-type σ des valeurs prises par les données de référence doit être connu. Dans notre cas, il est difficile d'estimer l'erreur de localisation des points d'intérêt. D'autre part, le test du χ^2 est basé sur le critère c_1, à un facteur d'échelle près qui est l'inverse de l'écart-type σ. Or, la table 7.3 montre que ce critère n'est pas discriminant : nous avons calculé la moyenne et l'écart-type de c_1 sur plusieurs images correspondant à un certain type de mouvement ou à un zoom, pour quatre scènes différentes (figure 7.7) : la scène de la cabane miniature (scène 1), une scène de laboratoire (scène 2), une scène d'intérieur (scène 3) et une scène d'extérieur (scène 4). Deux caméras différentes sont utilisées pour filmer les scènes 1 et 2 et les scènes 3 et 4. La table 7.3 indique les paramètres de la caméra qui varient, la scène et le nombre d'images considérés, puis la moyenne $E(c_1)$ et l'écart-type $\sigma(c_1)$. Nous voyons que c_1 peut être plus petit pour certains types de mouvements que pour certains zooms. Nous ne pouvons donc pas l'utiliser comme critère de discrimination.

La difficulté réside dans le fait que le modèle est validé sur les données ayant servi à en calculer les paramètres. Le critère c_1 étant le critère minimisé, il n'est pas étonnant que celui-ci soit faible, y compris lorsque la transformation de l'image ne correspond pas exactement à une TATP.

Pour cette raison, il est donc préférable d'utiliser des primitives qui ne sont pas utilisées pour minimiser c_1. Par exemple, nous pouvons utiliser d'autres points $(u_i,v_i)_{i>n}$ de l'image I_{i-1} (points d'intérêts non appariés ou points quelconques uniformément répartis dans l'image). Dans



Scène 1 Scène 2



Scène 3 Scène 4

FIG. 7.7 – *Extraits des scènes utilisées pour estimer la pertinence des critères c_1 et c_2.*

ce cas, nous ne pouvons plus comparer la position des points $(C_0u_i + a_0, C_0v_i + b_0)$ prédite dans l'image I_i avec la position mesurée (u'_i, v'_i), puisque ces points ne sont pas appariés. Par contre, nous pouvons comparer l'intensité des points (u_i, v_i) dans l'image I_{i-1} avec l'intensité des points de coordonnées $(C_0u_i + a_0, C_0v_i + b_0)$ dans l'image I_i. Si elles sont similaires, le modèle est vérifié, sinon l'hypothèse de zoom est rejetée.

La solution que nous avons adoptée est semblable à celle-ci, sauf que nous avons trouvé plus judicieux d'utiliser des contours plutôt que des points pour valider l'hypothèse. En effet, en plus des déformations du contour qui ne correspondent pas à une TATP dans le cas d'un mouvement général de caméra, nous pouvons aussi tirer profit des contours d'occultation qui sont particulièrement discriminants : dans le cas d'un mouvement quelconque de caméra (y compris une translation le long de l'axe optique), des parties de la scène peuvent apparaître ou disparaître autour du contour d'occultation, ce qui modifie l'apparence photométrique de ce contour (figure 7.8). Dans le cas d'un zoom, l'image ne subit par contre qu'une TATP et les mêmes éléments sont visibles avant et après la transformation.

Nous utilisons donc l'ensemble des contours détectés dans l'image I_{i-1} et dont la TATP ne sort pas de l'image I_i (si $C_0 < 1$, c'est-à-dire s'il s'agit d'un zoom arrière, le rôle des deux images est inversé). Pour chaque point $m_i = (u_i, v_i)$ d'un contour donné, nous calculons un score de corrélation entre les intensités $I_{i-1}(u_i, v_i)$ et $I_i(C_0u_i + a_0, C_0v_i + b_0)$, qui tient compte du voisinage des points (ce score est couramment utilisé en vision par ordinateur [Luong92, Zhang et al.94]) :

$$s_{point}(m_i) = \frac{\sum_{l,h=-N}^{l,h=N} (I_{i-1}(u_i + l, v_i + h) - \overline{I_{i-1}(u_i, v_i)}).(I_i(C_0(u_i + l) + a_0, C_0(v_i + h) + b_0) - \overline{I_i(C_0u_i + a_0, C_0v_i + b_0)})}{\sqrt{\sum_{l,h}(I_{i-1}(u_i + l, v_i + h) - \overline{I_{i-1}(u_i, v_i)})^2 \sum_{l,h}(I_i(C_0(u_i + l) + a_0, C_0(v_i + h) + b_0) - \overline{I_i(C_0u_i + a_0, C_0v_i + b_0)})^2}},$$

Variation des paramètres de la caméra	Scène	Nombre d'images	$E(c_1)$	$\sigma(c_1)$	$E(c_2)$	$\sigma(c_2)$
Zoom	1	6	0.617	0.030	0.747	0.055
	2	4	0.460	0.266	0.860	0.055
	3	32	0.860	0.057	0.677	0.133
	4	29	0.515	0.014	0.561	0.064
Translation	1	2	0.651	0.020	0.393	0.066
le long de	2	4	0.841	0.018	0.274	0.035
l'axe optique	3	16	1.380	0.190	0.047	0.277
Rotation + translation	1	10	3.593	1.439	-0.591	0.171
Mouvement panoramique	4	15	0.630	0.066	-0.209	0.315

TAB. 7.3 – *Moyenne et écart-type des critères c_1 et c_2 pour différentes variations des paramètres de la caméra.*

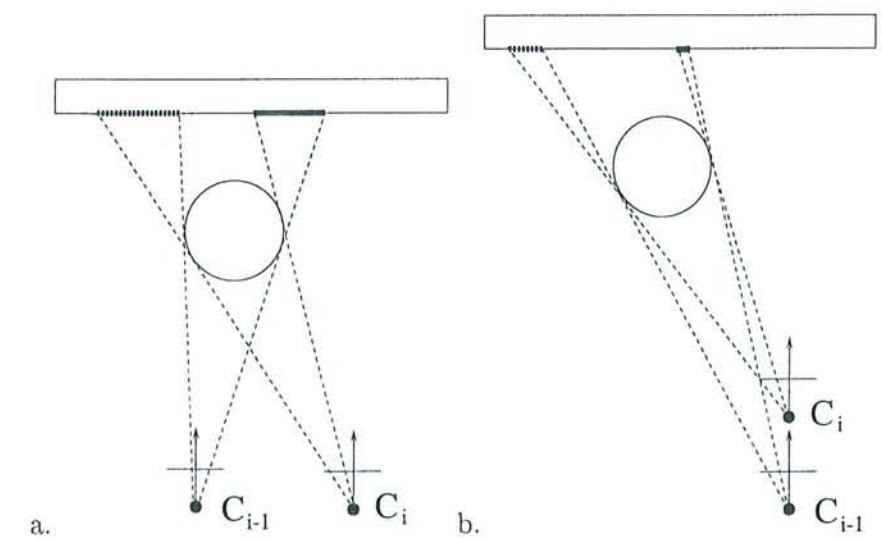


FIG. 7.8 – *Dans le cas d'un mouvement général de la caméra (a), des éléments de la scène apparaissent (trait épais) ou disparaissent (trait épais discontinu) autour des contours d'occultation. Ce constat est aussi valable pour les translations le long de l'axe optique (b).*

où $\overline{I(u,v)}$ est la moyenne des intensités sur la fenêtre de corrélation $[u - N,u + N] \times [v - N,v + N]$. Ce score est compris entre -1 et 1 et est d'autant plus proche de 1 que les points comparés se ressemblent. Il s'agit d'un score normalisé, qui est donc peu sensible aux changements d'illumination entre les deux vues.

Le score d'un contour C_i est défini par la moyenne des scores de ses p points :

$$s_{contour}(C_i) = \frac{1}{p} \sum_{j=1}^{p} s_{point}(m_j).$$

$s_{contour}(C_i)$ est donc compris entre -1 et 1. La figure 7.9 montre un exemple de scores obtenus pour la scène 3, dans le cas d'une translation le long de l'axe optique : plus le score est petit, plus le trait représentant le contour est foncé. Certains contours de type arête obtiennent un score faible, mais nous voyons que l'un des scores les plus petits est obtenu pour un contour d'occultation (au niveau de l'accoudoir cylindrique de la chaise).

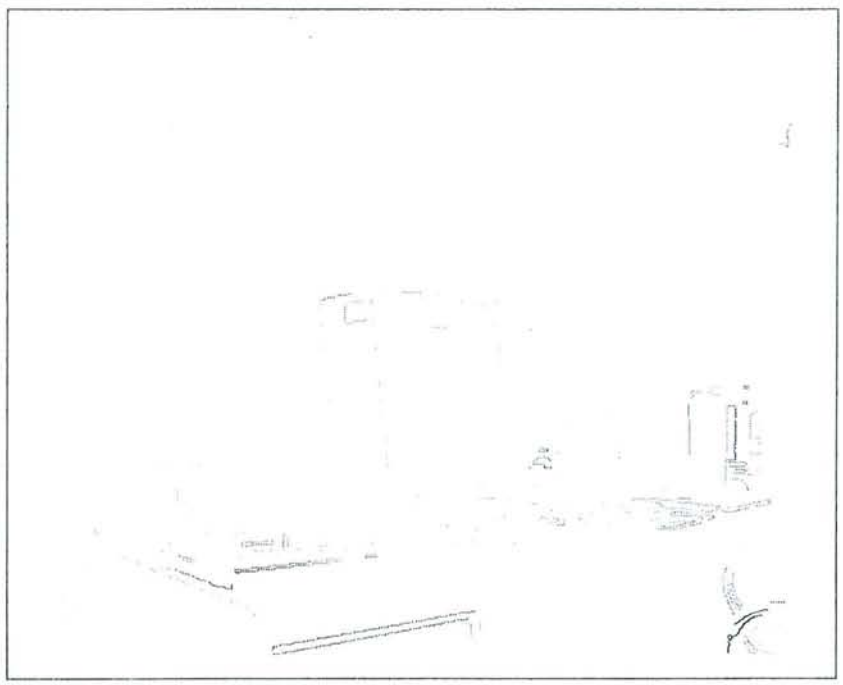


FIG. 7.9 – *Scores obtenus pour la scène 3 dans le cas d'une translation le long de l'axe optique. Les contours ont un niveau de gris allant du blanc pour un score de 1 au noir pour un score de -1.*

Nous pouvons maintenant définir un nouveau critère de discrimination en fonction des scores $s_{contour}(\mathcal{C}_i)$. La moyenne de ces scores est un critère envisageable. Cependant, dans certains cas, le déplacement de la caméra est tellement proche d'un zoom du point de vue de la transformation de l'image, que seuls quelques contours obtiennent un score réellement faible : la moyenne des scores n'exploite alors que faiblement ces contours et le critère devient peu discriminant. Comme la TATP permet de décrire très précisément les zooms, aucun contour ne doit obtenir de score faible dans le cas d'un zoom. Nous pouvons donc utiliser le minimum des scores $s_{contour}(\mathcal{C}_i)$ comme critère de sélection :

$$c_2 = \min_i s_{contour}(\mathcal{C}_i).$$

Comme nous pouvons le voir en table 7.3, le critère c_2 est nettement plus discriminant que le critère c_1. Nous l'utilisons donc pour distinguer les zooms des mouvements de caméra : si $c_2 > s$, alors l'hypothèse de zoom est retenue, sinon il s'agit d'un mouvement de caméra. Pour le choix du seuil s, nous avons calculé c_2 sur un certain nombre d'images et de séquences, en plus des séquences présentées en 7.3. La valeur $s = 0.5$ s'est avérée discriminante dans la plupart des cas, y compris pour les cas délicats de translations le long de l'axe optique.

Notons cependant que ce critère n'est pas robuste à la présence d'objets mobiles dans la scène : un objet qui se déplace pendant un zoom génère de mauvais scores au niveau des contours de l'objet. Comme c_2 est le minimum des scores $s_{contour}(\mathcal{C}_i)$, sa valeur est alors faible alors qu'il s'agit d'un zoom. L'utilisation de la médiane des scores n'est pas satisfaisante car, comme nous l'avons dit plus haut, pour certains mouvements de caméra seuls quelques contours obtiennent un score faible, qui ne seraient donc pas pris en compte par la médiane. Une détection automatique des objets mobiles de la scène permettrait sans doute de résoudre ce problème (voir les perspectives au chapitre 8).

7.3 La boucle de recalage temporel, version 4

Nous voyons maintenant comment le recalage est effectué lorsqu'un zoom est détecté, avant de présenter la dernière version de notre boucle de recalage temporel, tenant compte des changements de focale éventuels.

7.3.1 Recalage dans le cas d'un zoom

Lorsqu'un zoom est détecté, nous devons réaliser deux opérations :

1. calculer les nouveaux paramètres intrinsèques de la caméra tenant compte du zoom,
2. calculer la nouvelle position des courbes 2-D dans l'image : un plan de zoom doit en effet pouvoir être suivi d'un mouvement de caméra, et donc d'un recalage à six paramètres des courbes 3-D/2-D qui doivent alors être à jour.

Pour cela, deux solutions s'offrent à nous :

- à partir des paramètres C_0, a_0 et b_0 donnés par les équations (7.5), nous pouvons calculer directement les nouveaux paramètres intrinsèques de la caméra en utilisant les équations (7.3), et mettre à jour les primitives 2-D en calculant leur TATP (solution 1) ;
- comme au chapitre 5, nous pouvons aussi suivre les courbes 2-D appariées en utilisant la méthode de Berger, puis effectuer un recalage 3-D/2-D à trois paramètres α_u, u_0 et v_0 à partir des nouveaux appariements (solution 2). L'étape de prédiction pour le suivi des courbes 2-D peut être remplacée par un calcul de la TATP de ces courbes ;

La solution 1 est bien-sûr plus rapide que la solution 2, mais elle possède l'inconvénient d'accumuler les erreurs obtenues sur les valeurs de C_0, a_0 et b_0 entre les images consécutives : une imprécision non négligeable peut alors être obtenue sur les paramètres intrinsèques et la position des courbes 2-D à la fin d'un long plan de zoom. Si les courbes ne correspondent plus exactement aux contours réels de l'image, leur suivi pour le recalage à six paramètres risque d'échouer.

La solution 2 est plus lente, mais elle maintient des primitives 2-D cohérentes avec les gradients forts de l'image. Elle nous permet de plus de mettre à jour l'ensemble des primitives visibles de la même façon qu'au chapitre 5, en remplaçant les six paramètres du mouvement par les trois paramètres intrinsèques dans les optimisations. Ceci peut s'avérer très utile pour les zooms importants qui peuvent faire apparaître (zoom arrière) ou disparaître (zoom avant) de nombreuses primitives. Les primitives sortantes sont alors supprimées, tandis que les nouvelles primitives peuvent être initialisées dès leur apparition, à condition que le contour 2-D correspondant soit effectivement détecté.

La solution 2 a été adoptée pour la séquence de la cabane présentée en 7.4. Pour gagner du temps, nous pouvons aussi utiliser la solution 1 à chaque fois qu'un zoom est détecté entre deux images, mais ajuster les paramètres intrinsèques de la caméra et les courbes 2-D appariées en utilisant la solution 2 toutes les k images (voir la séquence du Loria en 7.4).

7.3.2 Optimisation alternée

L'algorithme de recalage temporel alternant optimisations à trois et à six paramètres est donc le suivant (figure 7.10) :

1. détection et appariement de points d'intérêt entre les images I_{i-1} et I_i,
2. calcul des paramètres C_0, a_0 et b_0 à partir de ces appariements,
3. détection des contours dans l'image I_{i-1} (ou I_i si $C_0 < 1$), puis validation de l'hypothèse *zoom* si $c_2 > s$. Si $c_2 < s$, l'hypothèse *mouvement* est retenue ;

4. dans le cas d'un zoom, utilisation de la solution 1 ou 2 pour calculer les nouveaux paramètres intrinsèques de la caméra. Dans le cas d'un mouvement de caméra, utilisation de la méthode hybride ou de la méthode 3-D/2-D pour calculer les nouveaux paramètres extrinsèques ;
5. mise à jour des courbes 3-D/2-D,
6. $i = i + 1$.

La figure 7.10 représente le système obtenu avec la méthode hybride dans le cas d'un mouvement de caméra et la solution 1 dans le cas d'un zoom, avec un ajustement (solution 2) toutes les p images.

7.3.3 Influence du pas entre les images

Lorsque le mouvement de la caméra est lent sur la séquence, il peut être plus discriminant d'utiliser non plus deux images consécutives I_i et I_{i+1} pour la segmentation, mais deux images I_i et I_{i+pas}. Cependant, les paramètres de la caméra doivent être calculés sur toutes les images de la séquence. La solution simple que nous adoptons pour pour les images intermédiaires $I_{i+1},...,I_{i+pas-1}$ consiste à utiliser le même type d'optimisation (à trois ou à six paramètres) que celui utilisé pour l'image I_i.

Pour un pas important, une erreur peut donc être générée sur les paramètres de la caméra si le changement de plan réel de la séquence a lieu entre les images I_i et I_{i+pas}. Nous ne considérons pas ce problème dans le système actuel : bien souvent, une période transitoire de quelques images est observée entre un mouvement et un zoom, pendant laquelle la caméra ne bouge pratiquement pas. Cette période correspond à une stabilisation de la caméra avant d'opérer un zoom après un mouvement de la caméra, ou un mouvement de la caméra après un zoom (nous observons par exemple ceci sur la séquence du Loria présentée plus loin). Si le nombre d'images de cette période transitoire est plus grand que pas, aucune erreur n'est générée.

Pour prendre un compte des changements abruptes de plans pour un pas important, nous pourrions comme dans [Xiong et al.98] utiliser une méthode de détection de la coupure de type dichotomique (rappelons que dans ce chapitre nous nous situons dans un contexte de post-production).

Nous pouvons aussi effectuer la segmentation de la séquence en commençant par l'image 0 ($\mathcal{S}_0$), puis une nouvelle segmentation en commençant par l'image 1 ($\mathcal{S}_1$), etc. jusqu'à l'image $pas-1$ ($\mathcal{S}_{pas-1}$). Supposons que la coupure pour la segmentation $\mathcal{S}_0$ soit détectée dans l'image im_0, la coupure pour $\mathcal{S}_1$ dans l'image im_1 etc. Comme un mouvement de caméra suivi ou précédé d'un zoom est en principe interprété comme un mouvement de caméra (sauf ambiguïtés), la coupure réelle a lieu dans l'image im telle que $im = \min_{i=0}^{pas-1} im_i$ pour une transition zoom/mouvement ou $im = \min_{i=0}^{pas-1} im_i - pas$ pour une transition mouvement/zoom. Considérons par exemple un pas de 4 (figure 7.11). Supposons qu'une coupure a lieu dans l'image 17 de la séquence : jusqu'à l'image 17 la caméra est en mouvement, après l'image 17 il s'agit d'un zoom. Les quatre segmentations obtenues sont données en figure 7.11, un mouvement de caméra étant indiqué par un M et un zoom par un Z. Nous voyons que le zoom est détecté en image $im_0 = 24$, $im_1 = 21$, $im_2 = 22$ et $im_3 = 23$. Nous avons bien min(24,21,22,23)-4=17.

7.4 Résultats expérimentaux

Nous présentons des résultats de segmentation et de recalage obtenus pour une séquence de laboratoire (la séquence de la cabane) et une séquence grandeur réelle (la séquence du Loria).

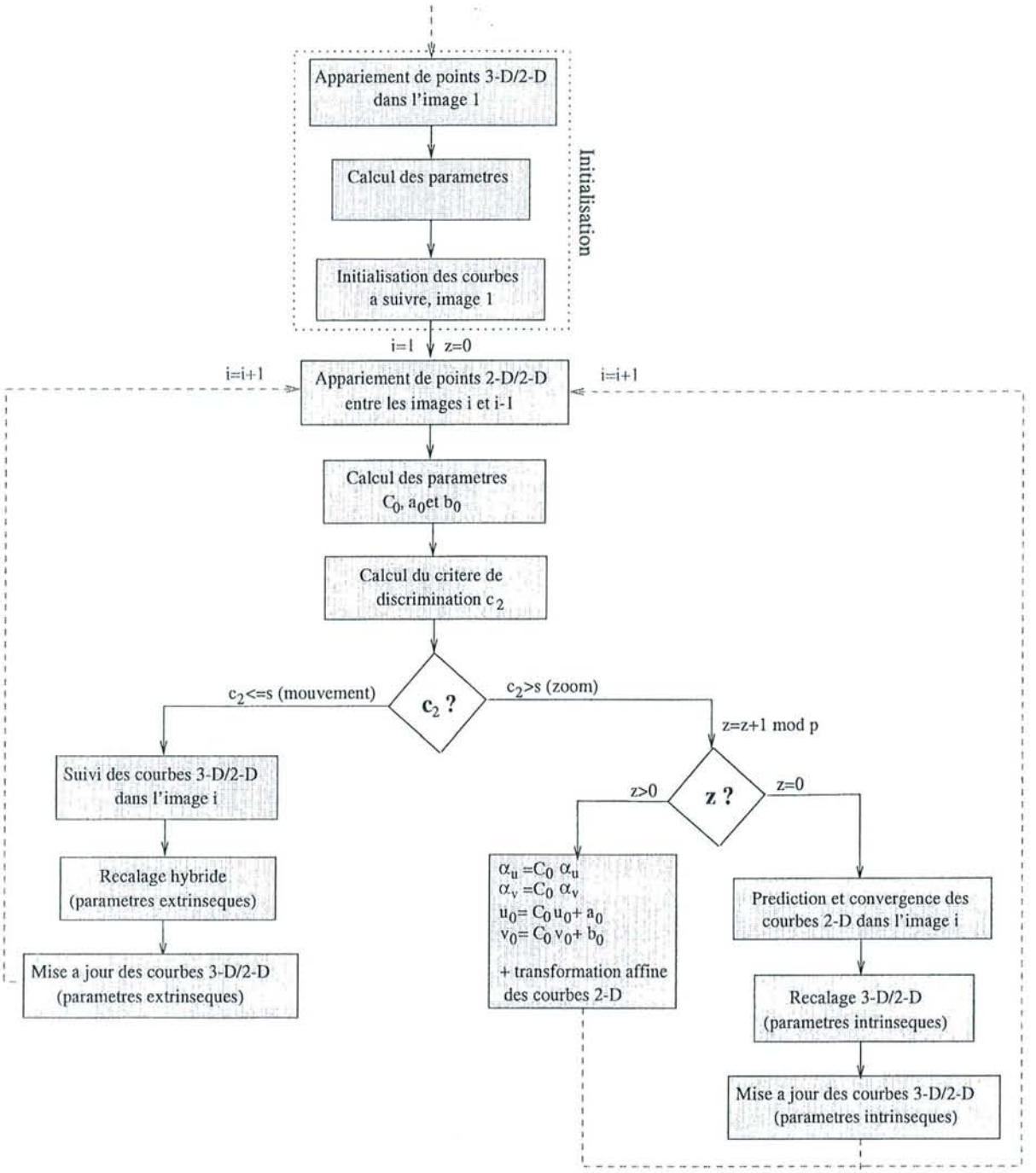


FIG. 7.10 – *Boucle de recalage temporel pour une séquence à focale variable.*

Comme pour les chapitres précédents, toutes les séquences de recalage, appariements de points et compositions d'images sont visibles sur le site internet.

7.4.1 La séquence de la cabane miniature

La séquence de la cabane miniature a été présentée en 7.1.1. L'utilisation d'une mire de calibration va nous permettre de valider très précisément les résultats.

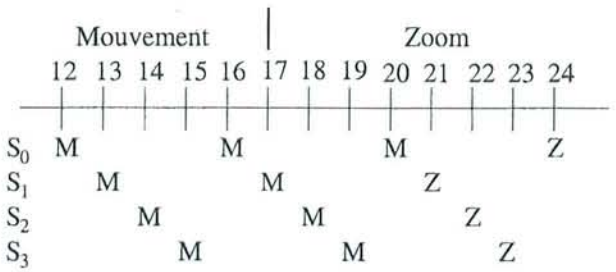


FIG. 7.11 – *Illustration d'une détection possible des coupures entre plans de mouvement et plans de zoom pour un pas de 4.*

Segmentation zoom/mouvement

La figure 7.12 montre les valeurs du critère c_2 obtenues sur la séquence, ainsi que le résultat de la segmentation (0 pour un mouvement et 1 pour un zoom), pour différents pas d'incrémentation entre les images. Pour un pas de 1, nous voyons que la translation le long de l'axe optique est interprétée comme un zoom : ceci n'est pas étonnant car entre deux images consécutives de la translation, le déplacement de la table micrométrique est très faible ($2cm$ à une distance de plus de $2m$), et donc l'image varie très peu (typiquement, la distance entre deux points appariés est en moyenne inférieure à 2 pixels). Par contre, pour un pas de 2 ou de 4, nous voyons que la segmentation calculée est quasiment identique à la segmentation attendue. Une seule erreur est obtenue entre les images 34 et 36 pour le pas de 2 et entre les images 32 et 36 pour le pas de 4, ce qui est tout à fait compréhensible puisque nous avons un zoom jusqu'à l'image 35, puis une translation entre les images 35 et 36 qui est, comme nous l'avons dit, quasiment insignifiante.

Nous utilisons la segmentation obtenue pour un pas de deux.

Recalage

Le recalage est effectué en utilisant la méthode hybride pour les mouvements de caméra et la solution 2 (recalage dans chaque image) pour les zooms.

La projection de la trajectoire de la caméra dans le plan horizontal et l'évolution du paramètre α_u ainsi obtenues sont représentées en figure 7.13. Nous pouvons constater que les résultats calculés sont très proches des résultats attendus.

Les courbes statistiques habituelles sont données en figure 7.14. L'erreur de reprojection est petite, ce qui est confirmé visuellement par la figure 7.16. Notons que là encore, l'estimation robuste se justifie pleinement puisque l'erreur de reprojection aux moindres carrés affiche un pic qui n'apparaît pas avec l'erreur robuste. Le nombre de primitives utilisées pour le recalage est compris entre 5 et 7. La trajectoire 3-D de la caméra dans le repère du monde, ainsi que les axes de ce repère sont représentés dans la figure 7.27.

Nous avons par ailleurs reconstruit les points de la mire à partir des mêmes images que celles que nous avions utilisées pour la reconstruction en 7.1.1 (optimisation à neuf paramètres). Nous constatons que les points reconstruits sont à présent beaucoup plus proches des points attendus qu'ils ne l'étaient pour l'optimisation à neuf paramètres.

Composition

Nous avons incrusté plusieurs objets virtuels dans la séquence de la cabane : une table, un parasol, une chaise, un palmier et de l'eau en mouvement. Pour plus de réalisme, nous calculons

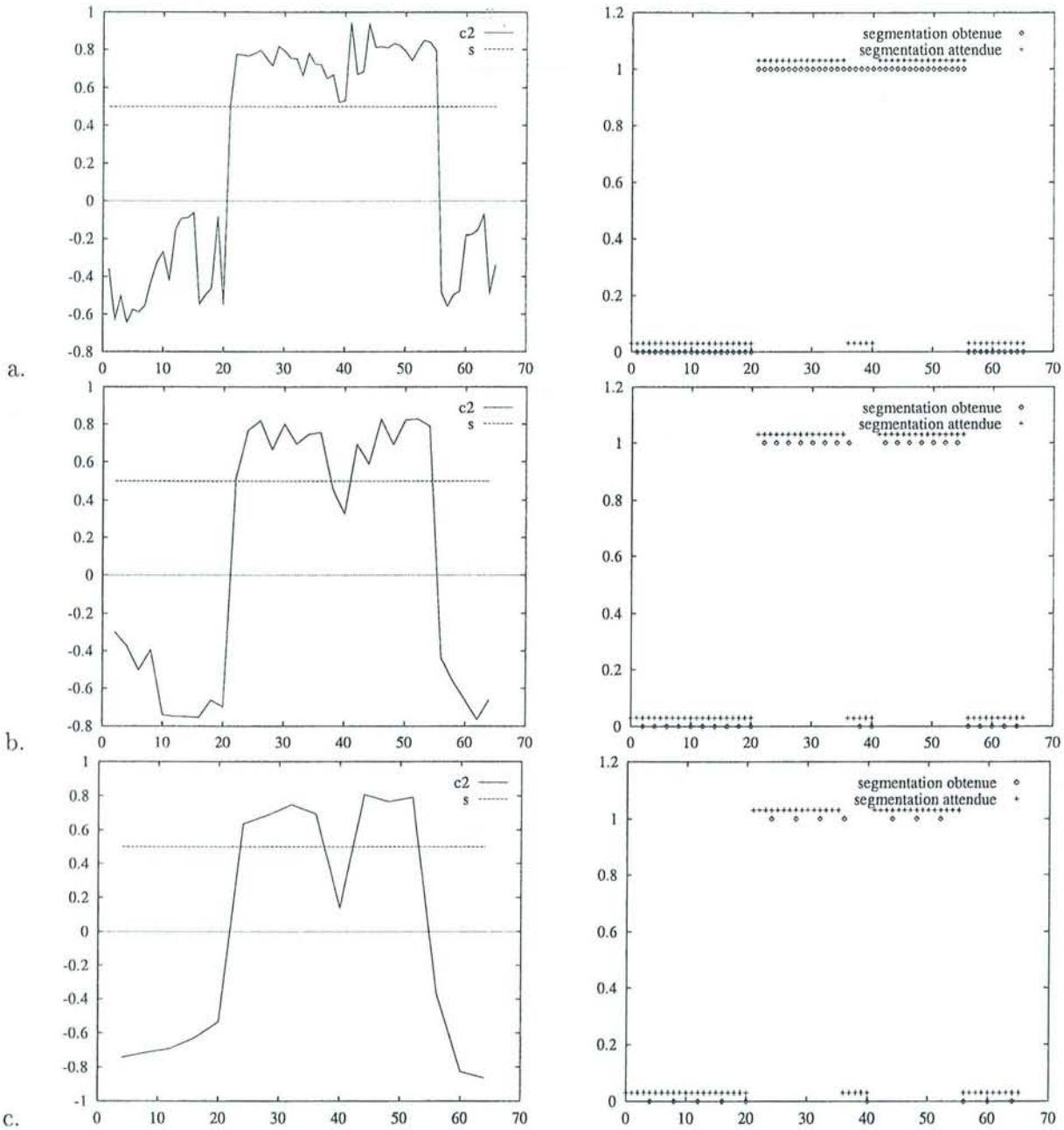


FIG. 7.12 – *Valeurs du critère c_2 (à gauche) et résultat de la segmentation (à droite) sur la séquence de la cabane. a. Pas de 1 entre les images. b. Pas de 2. c. Pas de 4.*

les ombrages à partir d'une reconstruction grossière de la scène, basée sur des points appariés automatiquement entre les images 0 et 20 et une triangulation de Delaunay sur les points reconstruits. La figure 7.18 montre une image de synthèse obtenue à partir de la scène reconstruite et les objets virtuels. Les ombres portées par la scène reconstruite n'ont pas été conservées car la reconstruction n'est pas suffisamment fine pour obtenir des ombres réalistes. Par contre, les ombres portées par les objets virtuels sur la scène réelle sont mixées avec l'image réelle, ce qui ajoute au réalisme de la composition.

La figure 7.19 montre les résultats de la composition obtenus pour diverses images de la

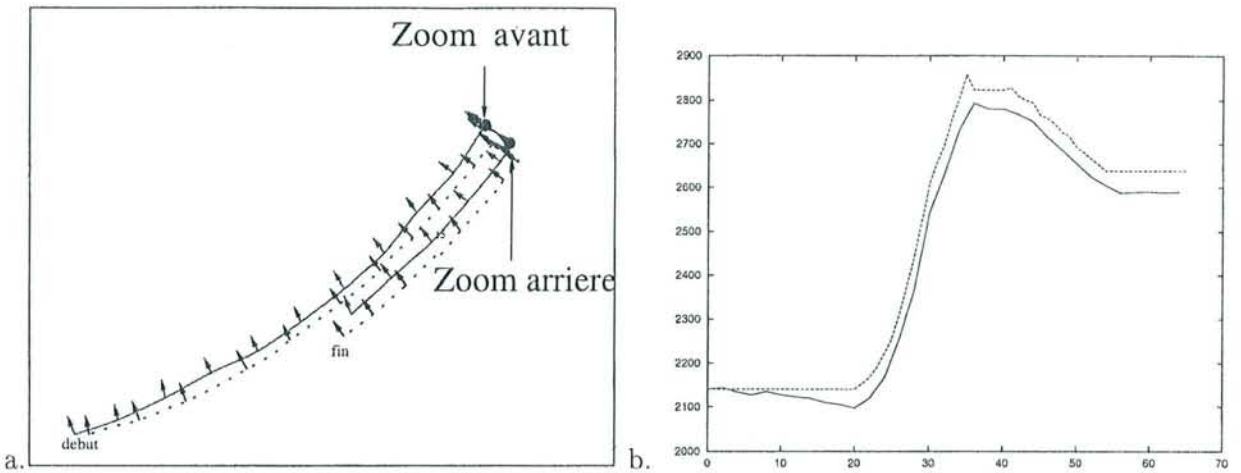


FIG. 7.13 – *Paramètres de la caméra attendus (traits continus) et calculés (traits discontinus) pour l'optimisation alternée sur la séquence de la cabane. a. Projection de la trajectoire de la caméra dans le plan horizontal. b. Paramètre* α_u.

séquence. Le résultat est visuellement cohérent et sur la séquence complète, les objets semblent fixes.

7.4.2 La séquence du Loria

Nous validons aussi cette méthode sur une longue séquence de notre laboratoire Loria, filmée à partir d'une caméra posée sur un trépied. Cette séquence comporte 630 images et est constituée d'un mouvement panoramique suivie d'un zoom avant très important, d'un zoom arrière puis à nouveau d'un panoramique. La segmentation est présentée en table 7.4. Le découpage ayant été réalisé manuellement, les bornes indiquées sont approximatives.

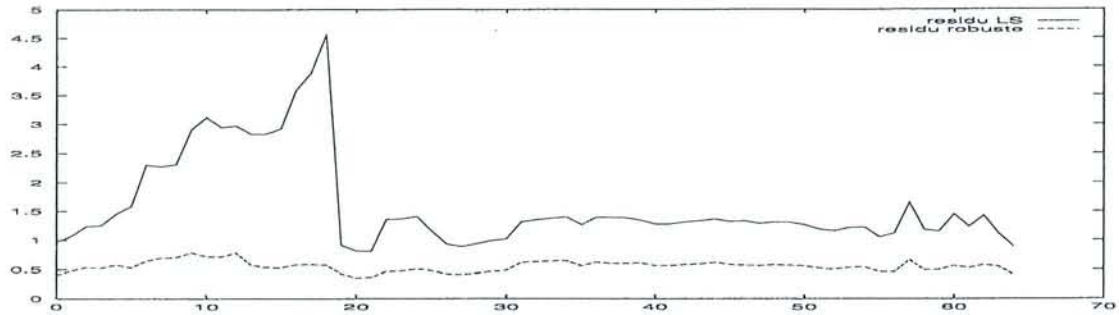
Image	Mouvement/zoom
$0 \rightarrow 120$	panoramique
$121 \rightarrow 346$	Zoom avant
$347 \rightarrow 399$	caméra et zoom fixes
$400 \rightarrow 594$	Zoom arrière
$595 \rightarrow 630$	panoramique

TAB. 7.4 – *La séquence du Loria.*

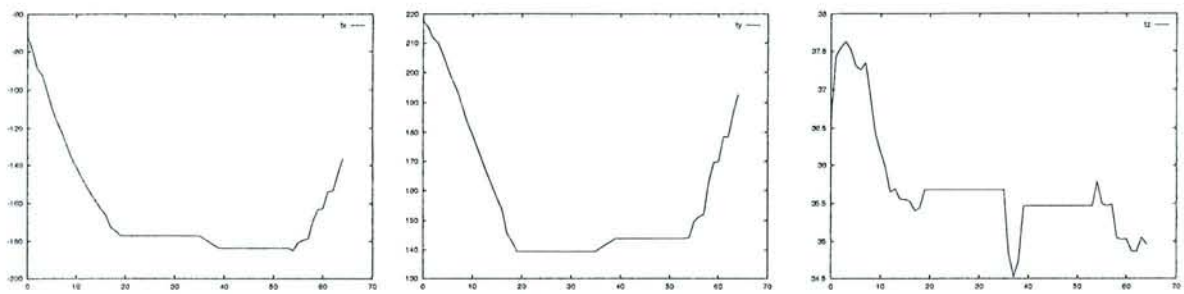
Difficultés

La séquence du Loria est l'une des plus délicates que nous ayons eues à traiter, pour les raisons suivantes :

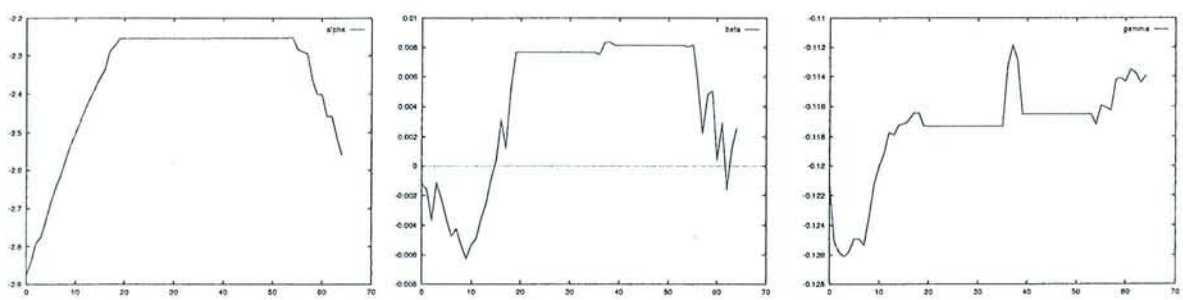
- le modèle 3-D du Loria est très approximatif, comme nous l'a confirmé son créateur. En particulier, plusieurs mesures ont été effectuées à la main pour compléter les plans papier initiaux,
- il s'agit d'une séquence très longue, qui fait apparaître différentes parties du laboratoire (figure 7.20),



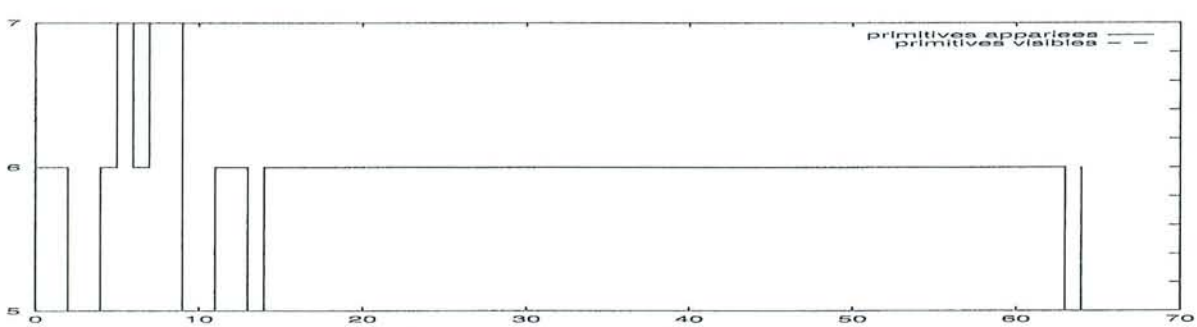
Résidus aux moindres carrés et résidus robustes.



$t_x(cm)$ $t_y(cm)$ $t_z(cm)$



$\alpha(rad)$ $\beta(rad)$ $\gamma(rad)$



Nombre de primitives visibles et appariées.

FIG. 7.14 – *Résultats statistiques obtenus sur la séquence de la cabane.*

– la variation de la focale est elle-aussi très importante : les dernières images du zoom avant ne font notamment apparaître que quelques fenêtres du laboratoire (figure 7.20.c).

En particulier, la détermination des paramètres intrinsèques de la caméra dans la première image s'est avérée problématique pour cette séquence. Nous avons tout d'abord utilisé 15 ap-

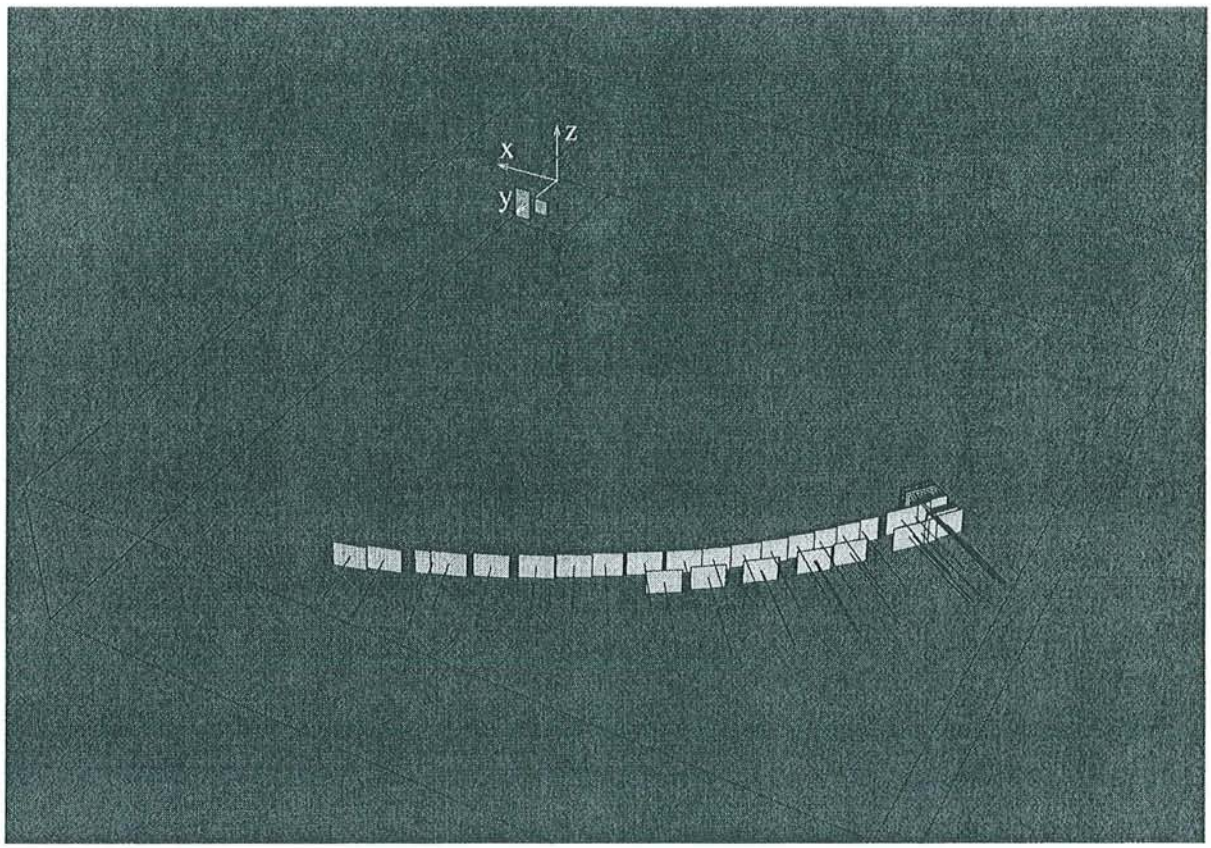


FIG. 7.15 – *Trajectoire 3-D retrouvée pour la séquence de la cabane.*

pariements de points 3-D/2-D extraits du bâtiment B du Loria, puis utilisé l'algorithme de calibration itératif présenté au chapitre 2 (figure 7.21.a). Nous avons alors constaté que l'erreur de reprojection moyenne ainsi que l'erreur de reprojection maximale des points appariés étaient assez importantes (respectivement 2.24 et 4.80 pixels), ce qui révélait effectivement la mauvaise qualité du modèle.

Par la suite, lorsque nous avons cherché à recaler des primitives du modèle dans les autres images, nous nous sommes aperçu que les paramètres intrinsèques que nous avions conservés ne convenaient plus pour la fin du mouvement panoramique. Pour vérifier ceci, nous avons alors apparié 19 points 3-D/2-D extraits des bâtiments A et B dans l'image 120 (dernière image du panoramique), et estimé le point de vue de la caméra minimisant l'erreur de reprojection de ces points aux moindres carrés (figure 7.21.b). L'erreur moyenne obtenue était alors de 3.94 pixels, et l'erreur maximale de 8.41 pixels! Une calibration itérative sur ces 19 points donnaient d'ailleurs des paramètres intrinsèques différents des paramètres obtenus dans la première image (ces paramètres sont donnés en figure 7.21) : ceci illustre le problème que nous avons mentionné au chapitre 2 : les paramètres obtenus sont localement corrects pour l'ensemble des points de référence, mais pas nécessairement pour les autres points de la scène.

Pour le recalage, nous avons finalement choisi d'utiliser la moyenne des paramètres intrinsèques obtenus dans les deux images calibrées.

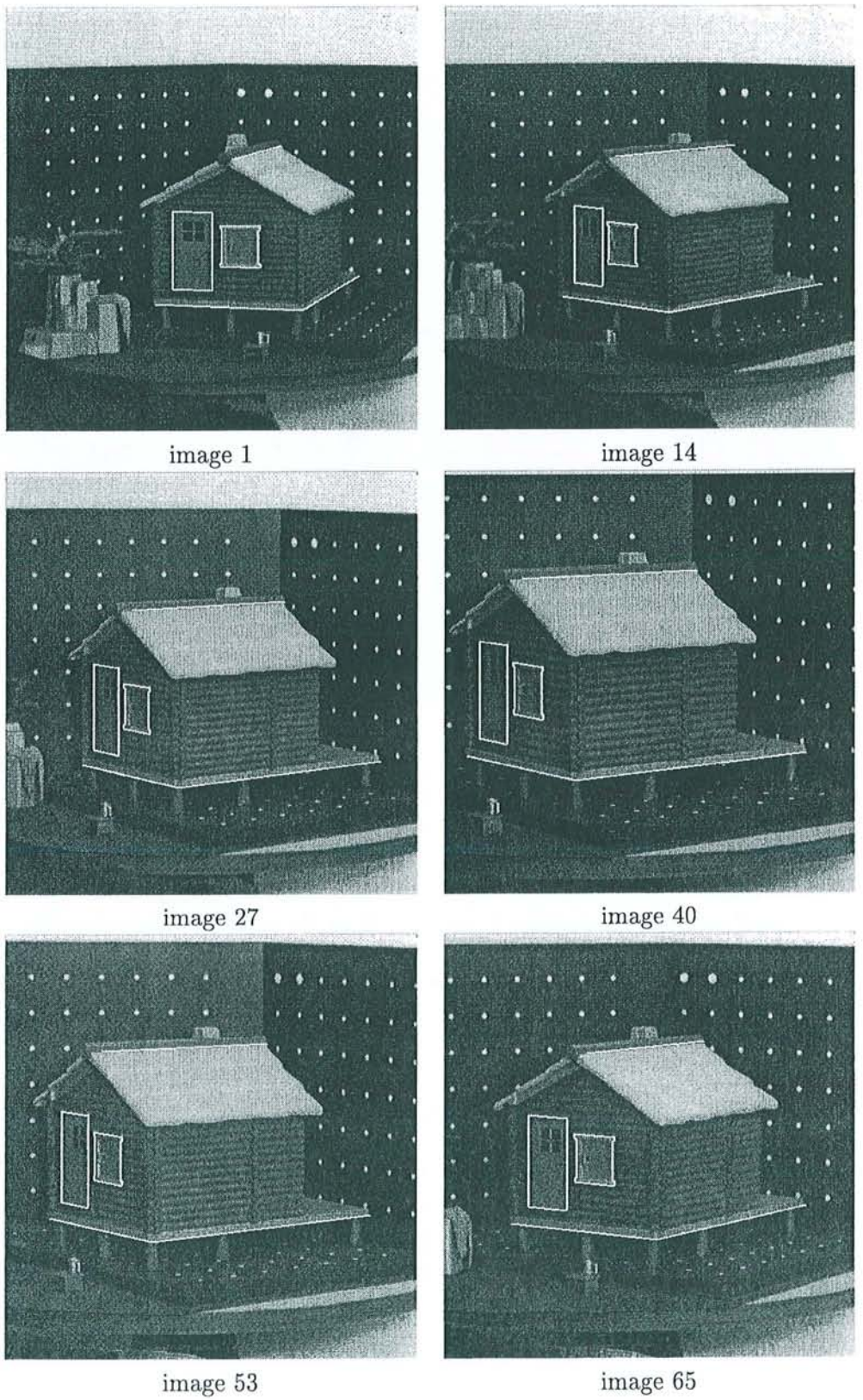


FIG. 7.16 – *Projection du modèle filaire de la cabane dans les images de la séquence.*

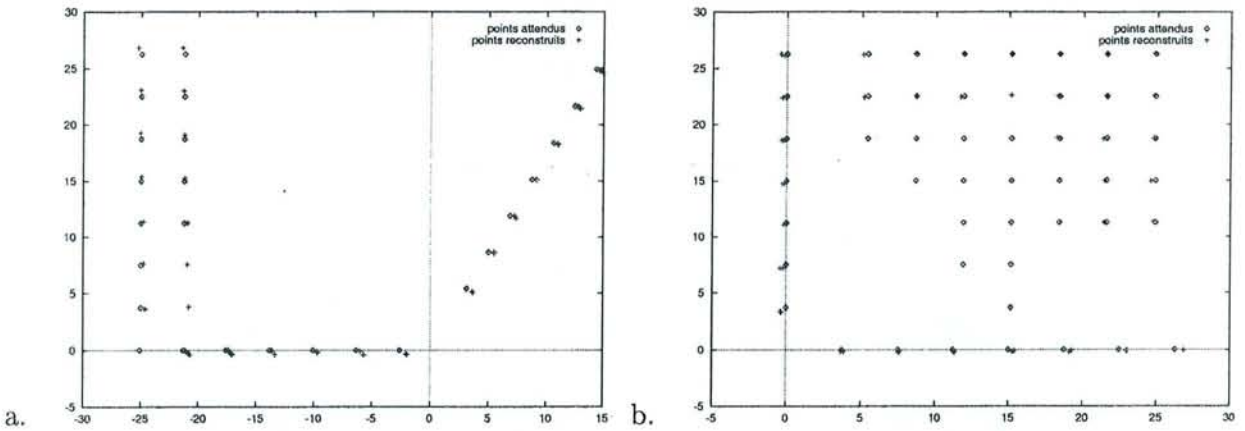


FIG. 7.17 – *Projection des points reconstruits dans les plan (xy) et (yz) pour l'optimisation alternée. Les croix sont les points reconstruits et les cercles les points attendus.*

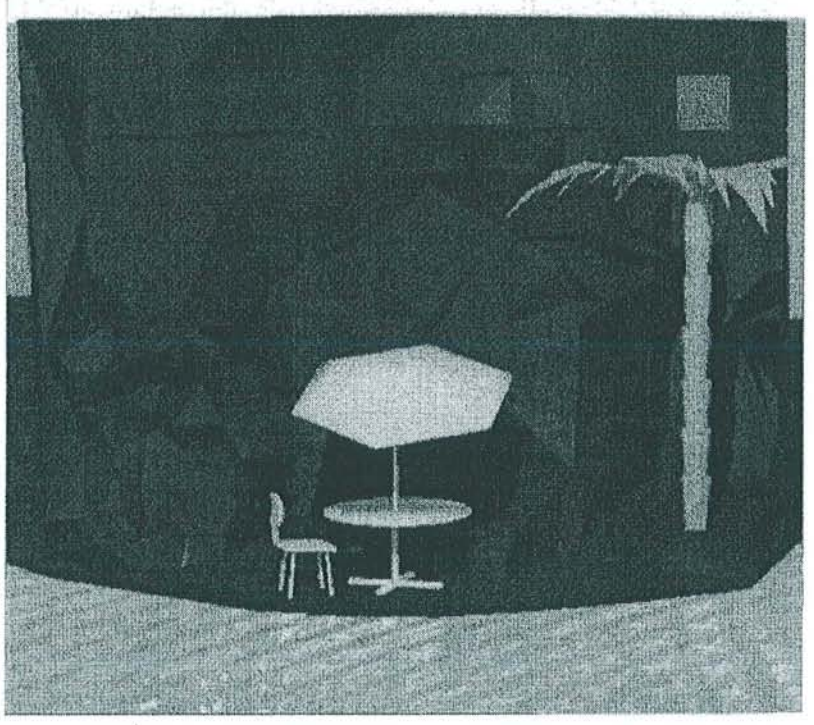


FIG. 7.18 – *Calcul des ombres entre objets réels et virtuels à partir d'une reconstruction grossière de la scène.*

Segmentation zoom/mouvement

Pour la segmentation, nous utilisons un pas de 8 entre les images. La figure 7.22 montre l'évolution du critère c_2 sur la séquence, accompagnée du test sur la position du centre de la TATP permettant de lever certaines ambiguïtés (1 si le centre est à l'intérieur de l'image, 0 sinon). La segmentation donnée en figure 7.22 détecte un zoom si c_2 est supérieur au seuil s ET

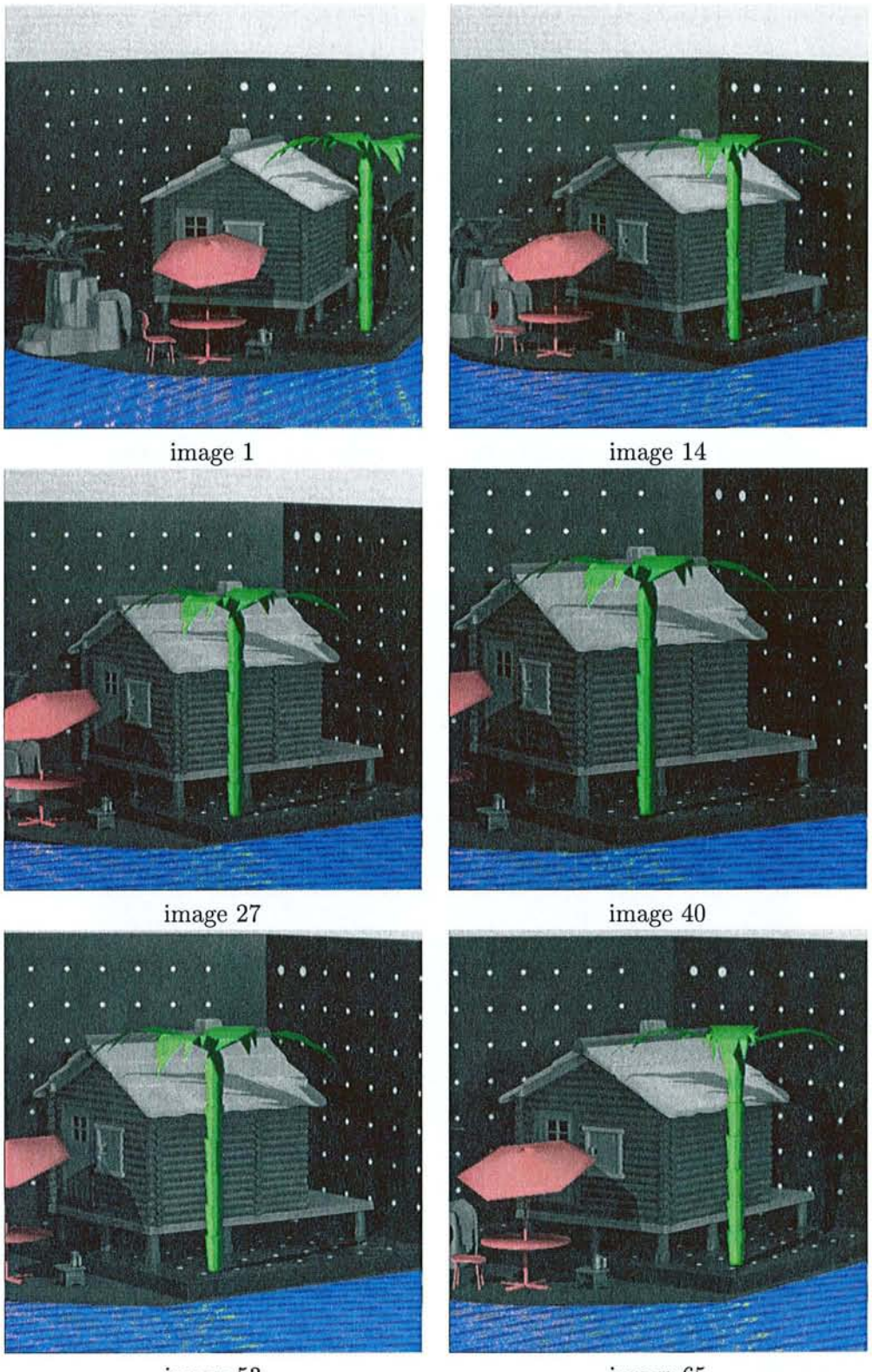


FIG. 7.19 – *Incrustation d'objets virtuels dans la séquence de la cabane.*

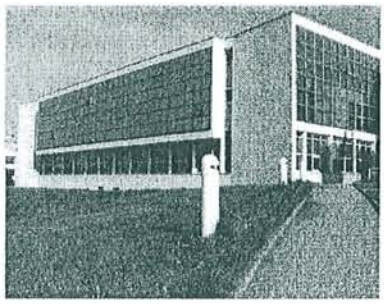
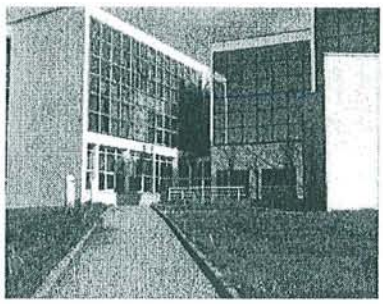
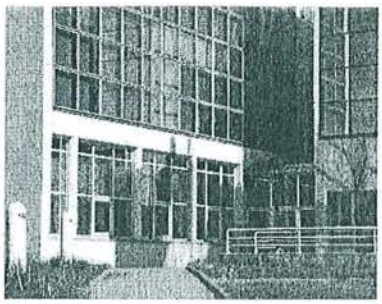
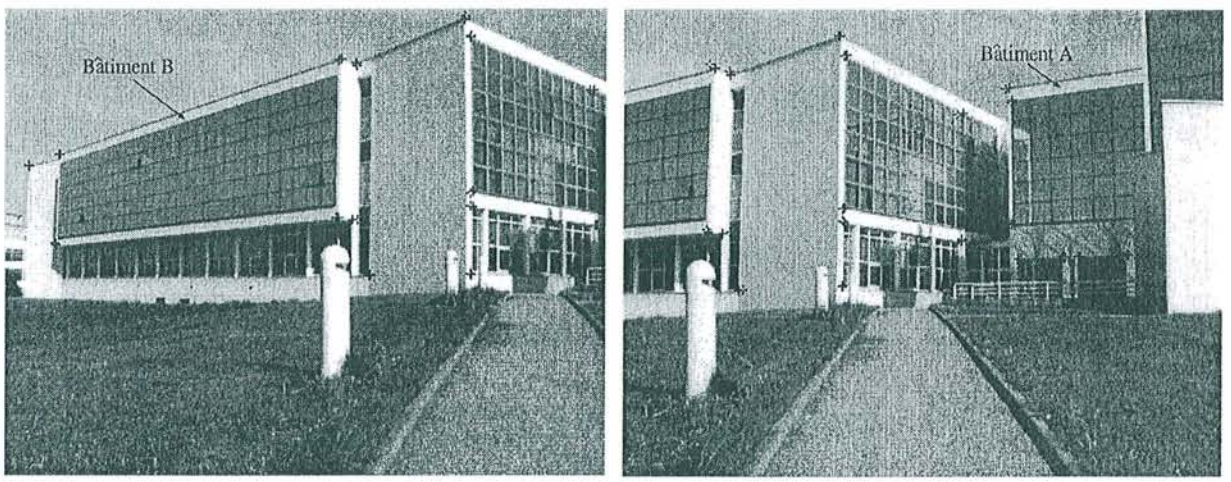
a.  b.  c. 

FIG. 7.20 – *Une des difficultés majeures pour la séquence du Loria réside dans la variation importante des paramètres intrinsèques et extrinsèques de la caméra au cours de la prise de vue. a. Image 0. b. Image 630. c. Image 344.*



a. image 0 : $u_0 = 337.88$, $v_0 = 270.79$
$\alpha_u = 802.61$, $\alpha_v = 840.96$

b. image 120 : $u_0 = 430.90$, $v_0 = 305.72$
$\alpha_u = 860.41$, $\alpha_v = 888.24$

FIG. 7.21 – *Points appariés et paramètres intrinsèques obtenus pour les images 0 et 120 de la séquence. Les croix continues indiques les points 2-D extraits dans l'image et les croix discontinues la projection de leurs correspondants 3-D.*

le centre de la TATP est à l'intérieur de l'image.

Globalement, le résultat obtenu correspond au résultat attendu. Nous observons effectivement quelques ambiguïtés au niveau des panoramiques, levées par le test sur la position du centre de la TATP. Dans de rares cas le zoom n'est pas reconnu, mais nous voyons que dans ces cas c_2 est tout de même très proche du seuil. Comme ces erreurs d'interprétation sont isolées, nous pourrions les éliminer facilement à l'aide d'un filtre médian par exemple.

Entre les images 345 et 408, la caméra est fixe et il n'y a pas non plus de zoom (nous utilisons la valeur 0.5 pour la segmentation attendue). Cette immobilité correspond à un zoom de facteur C_0 proche de 1 et de déplacement $(a_0 \quad b_0)^T$ quasi nul (figure 7.23). Elle est cependant interprétée comme un mouvement de caméra car le centre de la TATP est à l'intérieur de l'image. En fait, la position $\left(\frac{a_0}{1-C_0}, \frac{b_0}{1-C_0}\right)$ de ce centre est théoriquement instable puisque a_0 et b_0 sont proches de 0 et C_0 est proche de 1. Lorsque la caméra et le zoom sont fixes, le résultat de la segmentation n'a cependant pas d'importance puisque que les paramètres calculés doivent être identiques aux paramètres initiaux quelque soit le modèle utilisé. Remarquons aussi les phases transitoires entre

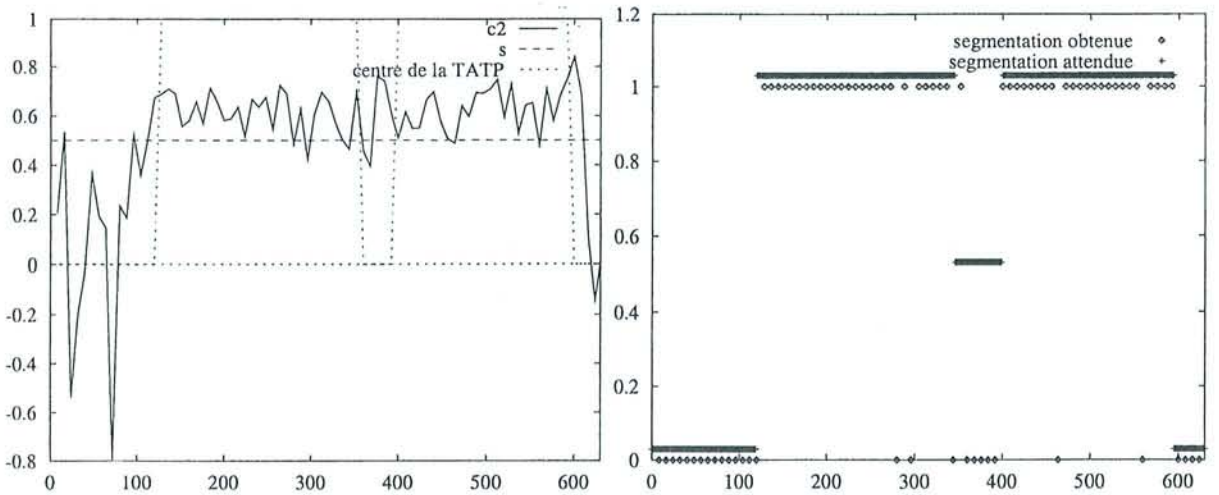


FIG. 7.22 – *Valeurs du critère c_2 (à gauche) et résultat de la segmentation (à droite) sur la séquence du loria.*

les panoramiques et les zooms, où les paramètres de la caméra sont quasiment fixes puisque C_0 est proche de 1 et a_0 et b_0 sont proches de 0 (figure 7.23).

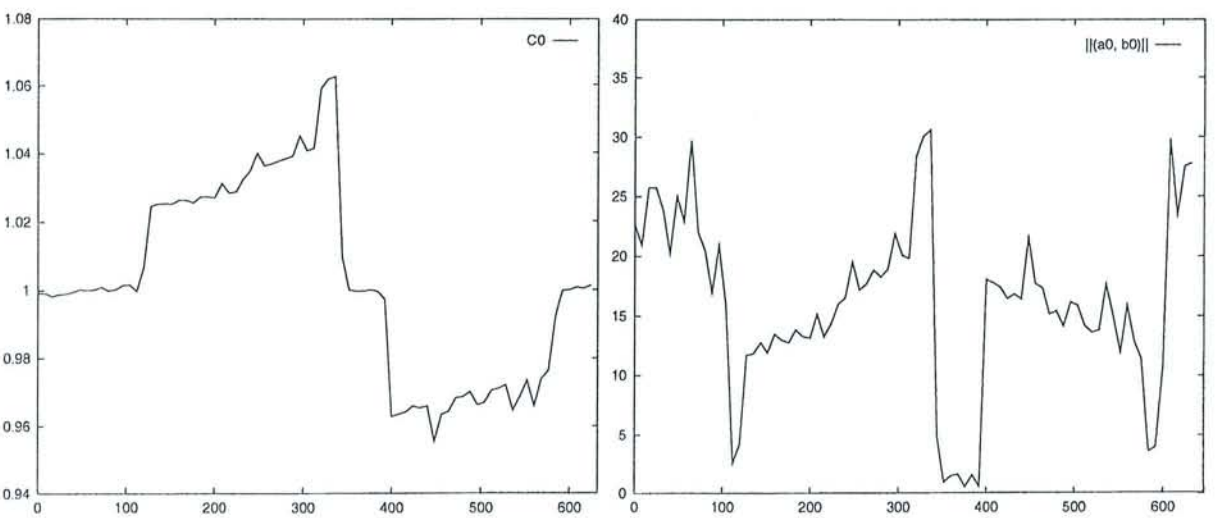


FIG. 7.23 – *Évolution de C_0 et de la norme du vecteur $(a_0 \quad b_0)^T$ sur la séquence du loria.*

Recalage

Le recalage est effectué à partir de la méthode hybride pour les mouvements panoramiques et en utilisant la solution 1 ainsi que la solution 2 toutes les 8 images pour les zooms. Les primitives suivies sont montrées pour quelques images de la séquence en figure 7.24. Leur nombre est compris entre 4 et 10, ce nombre étant bien sûr d'autant plus faible que la focale est grande (figure 7.25).

Nous voyons numériquement en figure 7.26 et visuellement dans les images de la figure 7.24 que l'erreur de reprojection de ces primitives est globalement élevée. Ceci est la conséquence directe de la mauvaise modélisation du bâtiment qui ne permet pas d'obtenir un recalage correct sur l'ensemble des primitives suivies.

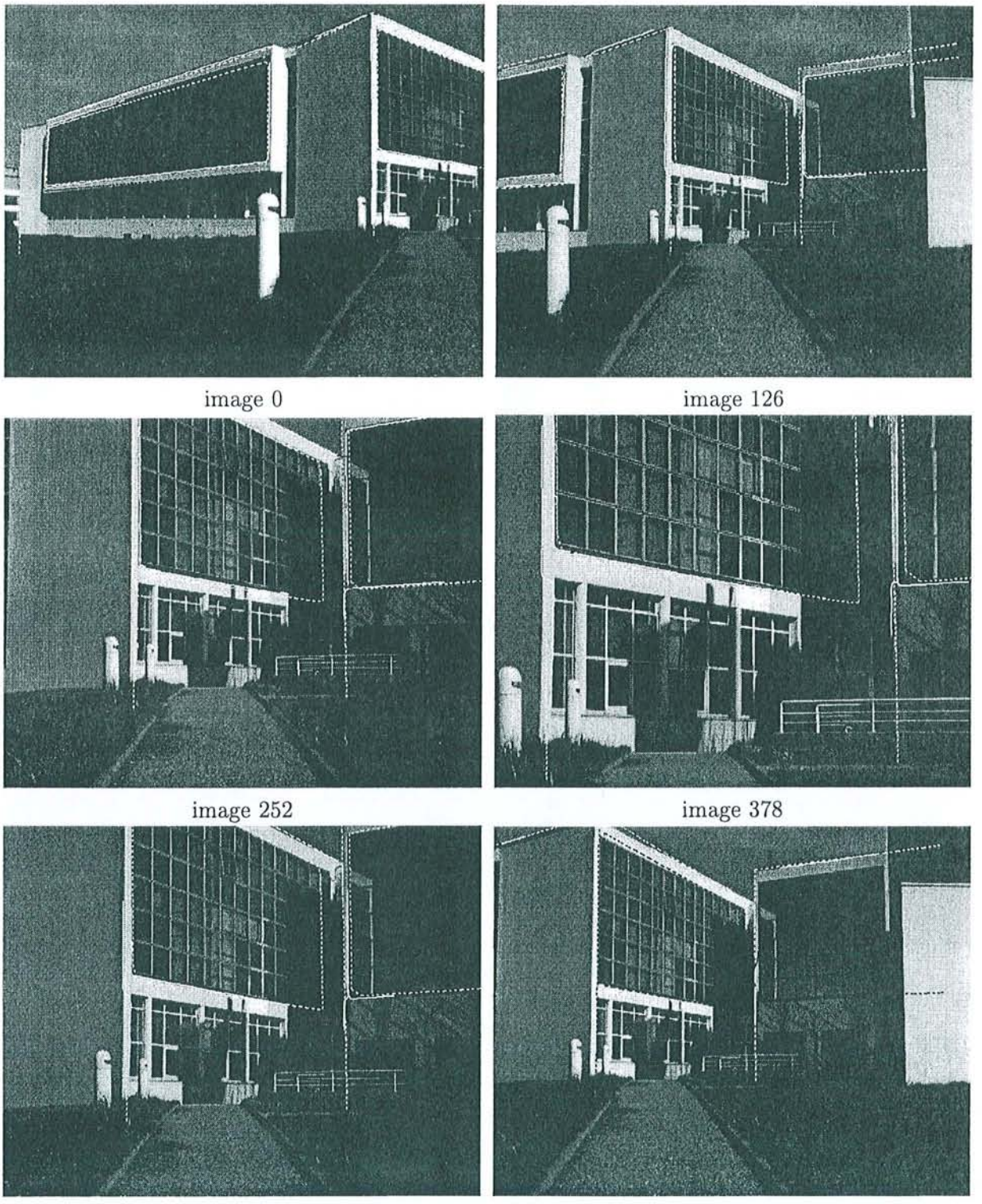


FIG. 7.24 – *Primitives 3-D/2-D suivies sur la séquence du Loria.*

L'évolution du plan image de la caméra dans le repère de la scène est donnée en figure 7.27. Nous retrouvons à peu près le mouvement panoramique attendu, suivi des zooms et d'un dernier

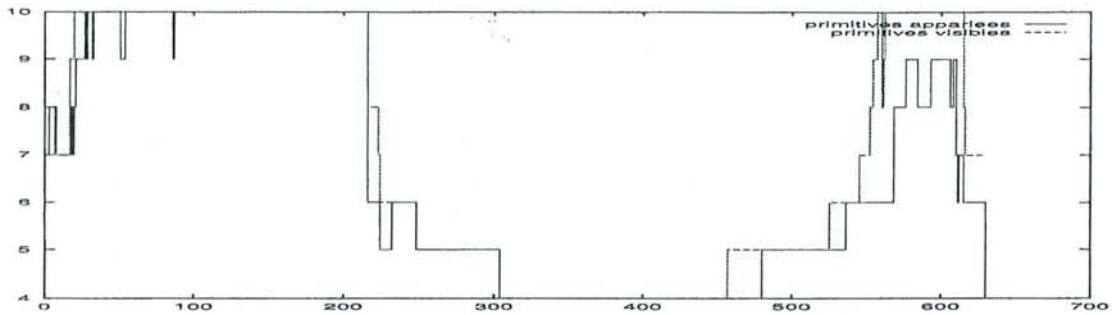


FIG. 7.25 – *Nombre de primitives visibles et appariées sur la séquence du loria.*

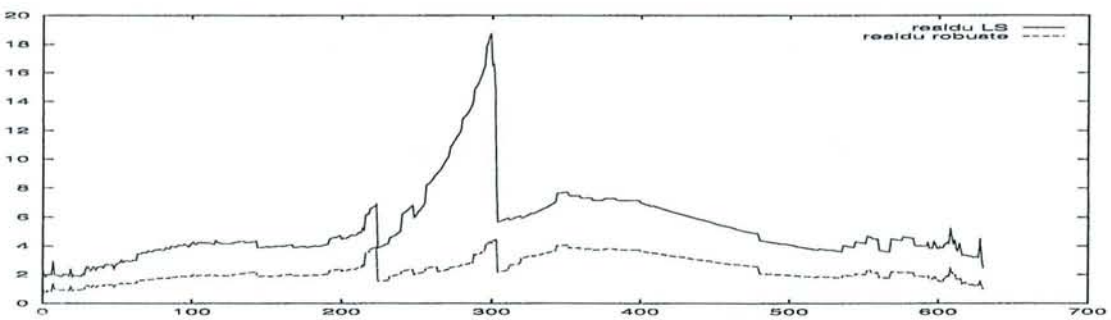


FIG. 7.26 – *Résidus aux moindres carrés et résidus robustes sur la séquence du loria.*

panoramique.

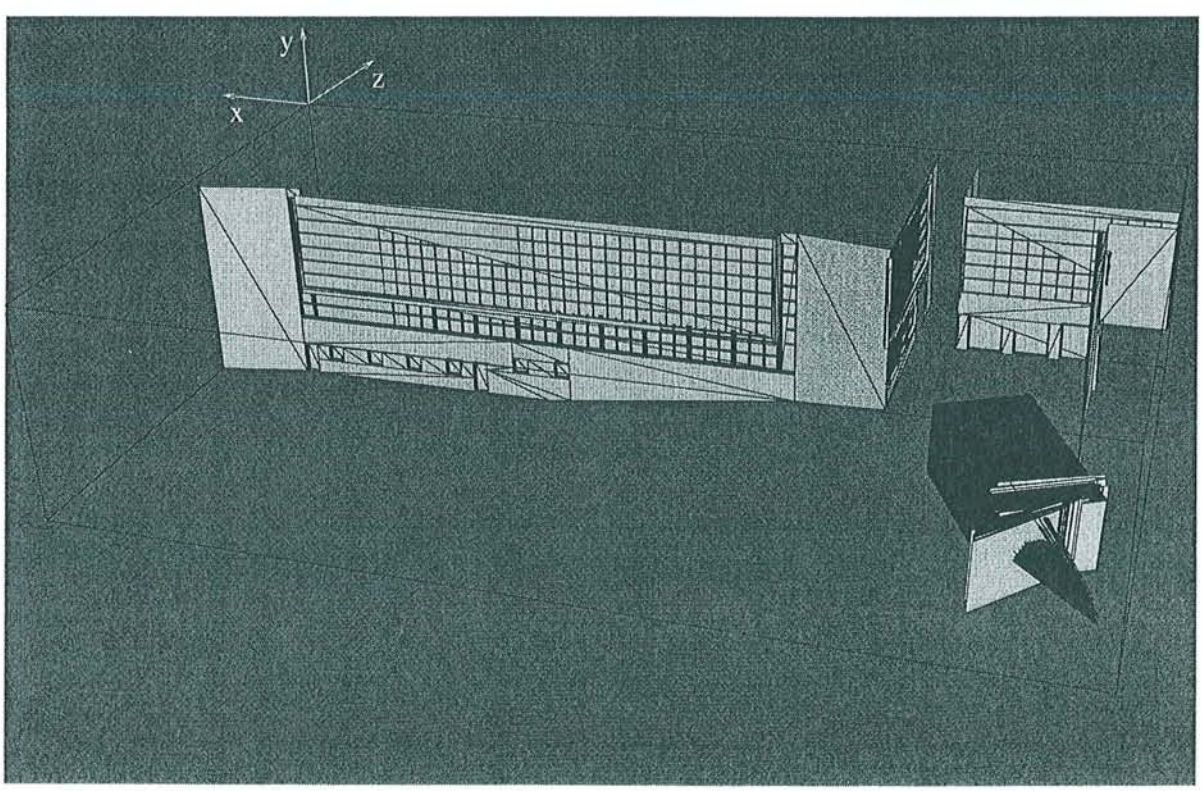


FIG. 7.27 – *Trajectoire 3-D retrouvée pour la séquence du loria.*

Étudions toutefois plus précisément l'évolution des paramètres extrinsèques (figure 7.28) et intrinsèques (figure 7.29) de la caméra. Nous voyons que l'angle β, qui est l'angle de rotation autour de l'axe vertical (les axes sont représentés en figure 7.27), décrit bien les deux mouvements panoramiques attendus et dessine une courbe particulièrement lisse. Par contre, les paramètres de la translation sensés rester constants sur les mouvements panoramiques suivent une courbe non constante et relativement saccadée. Nous avons choisi comme paramètres intrinsèques la moyenne des paramètres optimisant l'erreur de reprojection dans les images 0 et 120. Pour ces deux extrémités, il y a donc une compensation de la différence imposée sur les paramètres intrinsèques par la translation, le critère minimisé est toujours l'erreur de reprojection. Entre les deux images, une compensation pour la reprojection est aussi observée : comme nous avons utilisé la moyenne des paramètres intrinsèques calculés dans les images 0 et 120, il n'est pas étonnant d'observer des courbes de la translation quasiment symétriques par rapport à l'image 60.

Le paramètre α_u évolue selon la courbe attendue (figure 7.29). u_0 et v_0 répondent globalement de façon linéaire à α_u, mais les deux courbes sont assez saccadées. En fait, un saut plus ou moins important se produit toutes les 8 images, à chaque fois qu'un recalage 3-D/2-D à trois paramètres est effectué. Ces sauts expriment les corrections apportées toutes les p images afin que les courbes suivies correspondent effectivement aux gradient forts de l'image. Ces ajustements sont plus ou moins corrects, puisque liés à la pertinence du modèle 3-D.

La figure 7.30 montre la projection du modèle filaire du Loria dans plusieurs images de la séquence. Nous pouvons constater que, malgré les imprécisions du modèle, le recalage du Loria est acceptable. Par contre, nous allons voir que les résultats obtenus sont moins satisfaisants pour l'incrustation d'un objet virtuel fixe dans une zone relativement éloignée du modèle.

Composition

L'objectif est d'étudier l'impact d'une sculpture d'Art Moderne devant le bâtiment Loria. Pour cela, nous avons incrusté une reproduction virtuelle de la *Femme à la chevelure défaite* du peintre et sculpteur Mirõ à une vingtaine de mètres du bâtiment (figure 7.31). Si chaque image isolée de la composition peut paraître correcte, nous observons sur la séquence complète des effets de sautillements de l'objet virtuel, qui sont la conséquence directe du caractère saccadé des paramètres que nous avons calculés.

Comme nous nous plaçons dans un contexte de post-production, nous pouvons toutefois lisser ces courbes avant la composition. Nous effectuons donc un lissage dans le domaine fréquentiel, en utilisant la transformée de Fourier. Cette transformée approxime une courbe quelconque par une fonction :

$$x(t) = a_0 + \sum_{k=1}^{n} a_k \cos(kt) + b_k \sin(kt),$$

où n est un paramètre fixé par l'utilisateur. Plus n est grand, plus la courbe obtenue ressemble à la courbe initiale. À l'opposé, plus n est petit, plus la courbe obtenue est lisse, mais plus elle s'écarte de la courbe initiale. Il s'agit donc de trouver le bon compromis entre le lissage et l'adéquation à la courbe initiale : la figure 7.32 montre les transformées de Fourier de la courbe u_0 obtenues sur le plan de zoom pour différentes valeurs de n.

Nous avons donc utilisé cette méthode pour lisser par morceaux les 10 paramètres de la caméra. Les courbes obtenues sont représentées en figures 7.28 et 7.29. La séquence augmentée générée à partir des paramètres lisses est plus satisfaisante que la séquence brute, mais les effets de sautillements ont été remplacés à certains moments par des effets de glissements. Le lissage n'est donc pas la solution "miracle", et il est bien-sûr préférable d'obtenir des paramètres qui soient initialement lisses.

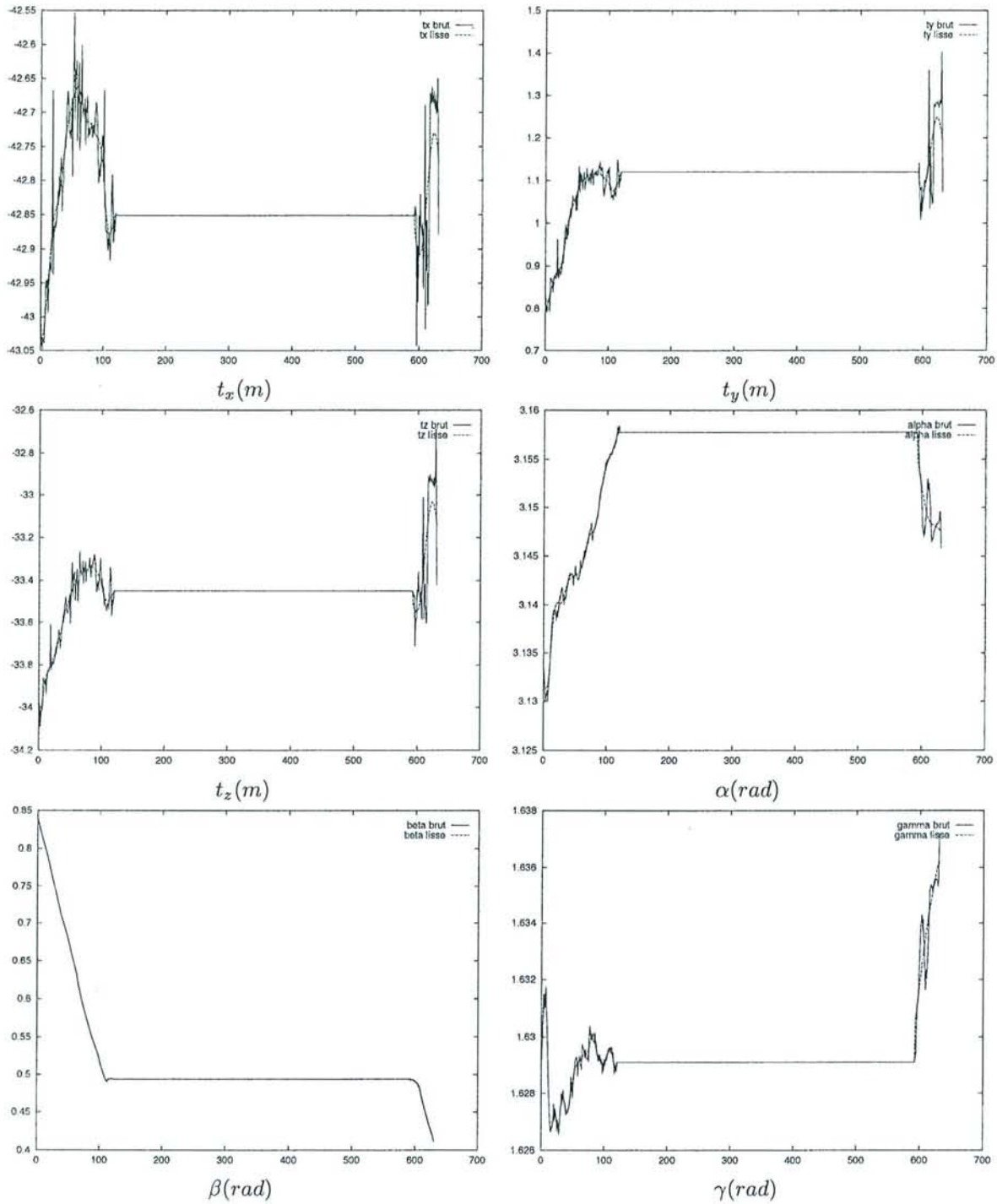
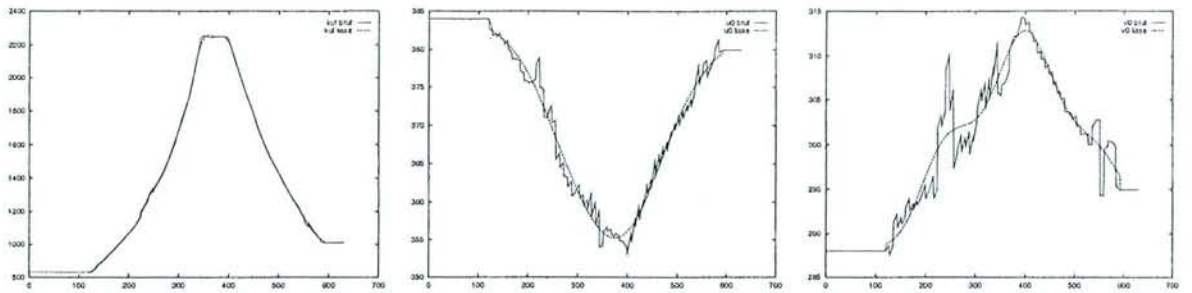


FIG. 7.28 – *Paramètres du point de vue obtenus sur la séquence du loria.*

Pour la séquence du Loria, nous avons rencontré un certain nombre problèmes dus aux conditions défavorables d'expérimentation (modèle incertain, séquence longue, grandes variations de la focale). De meilleurs résultats auraient certainement pu être obtenus si nous avions disposé d'un modèle plus juste du bâtiment. Cependant, il paraît difficile d'imputer entièrement la pré-



α_u u_0 v_0

FIG. 7.29 – *Paramètres intrinsèques de la caméra sur la séquence de la loria.*

cision insuffisante des résultats à la mauvaise qualité du modèle. En particulier, la différence des valeurs obtenues sur les paramètres intrinsèques de la caméra aux extrémités du premier mouvement panoramique peut aussi être due à une instabilité réelle de ces paramètres dans le temps.

En dehors des problèmes mentionnés, nous avons pu valider notre algorithme de segmentation sur une longue séquence en milieu extérieur, et obtenir un recalage du modèle somme toute relativement correct.

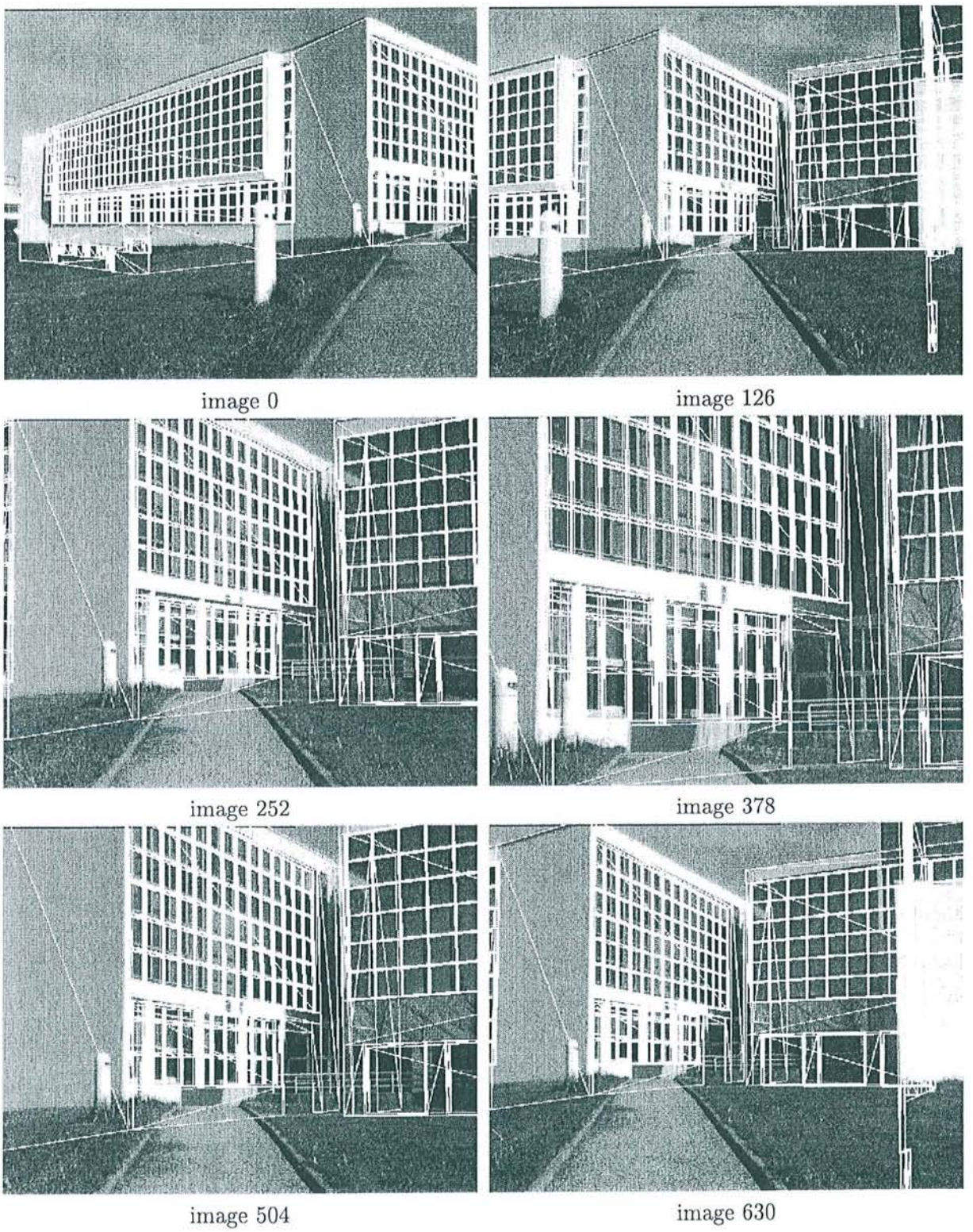


FIG. 7.30 – *Projection du modèle filaire du loria dans les images de la séquence.*

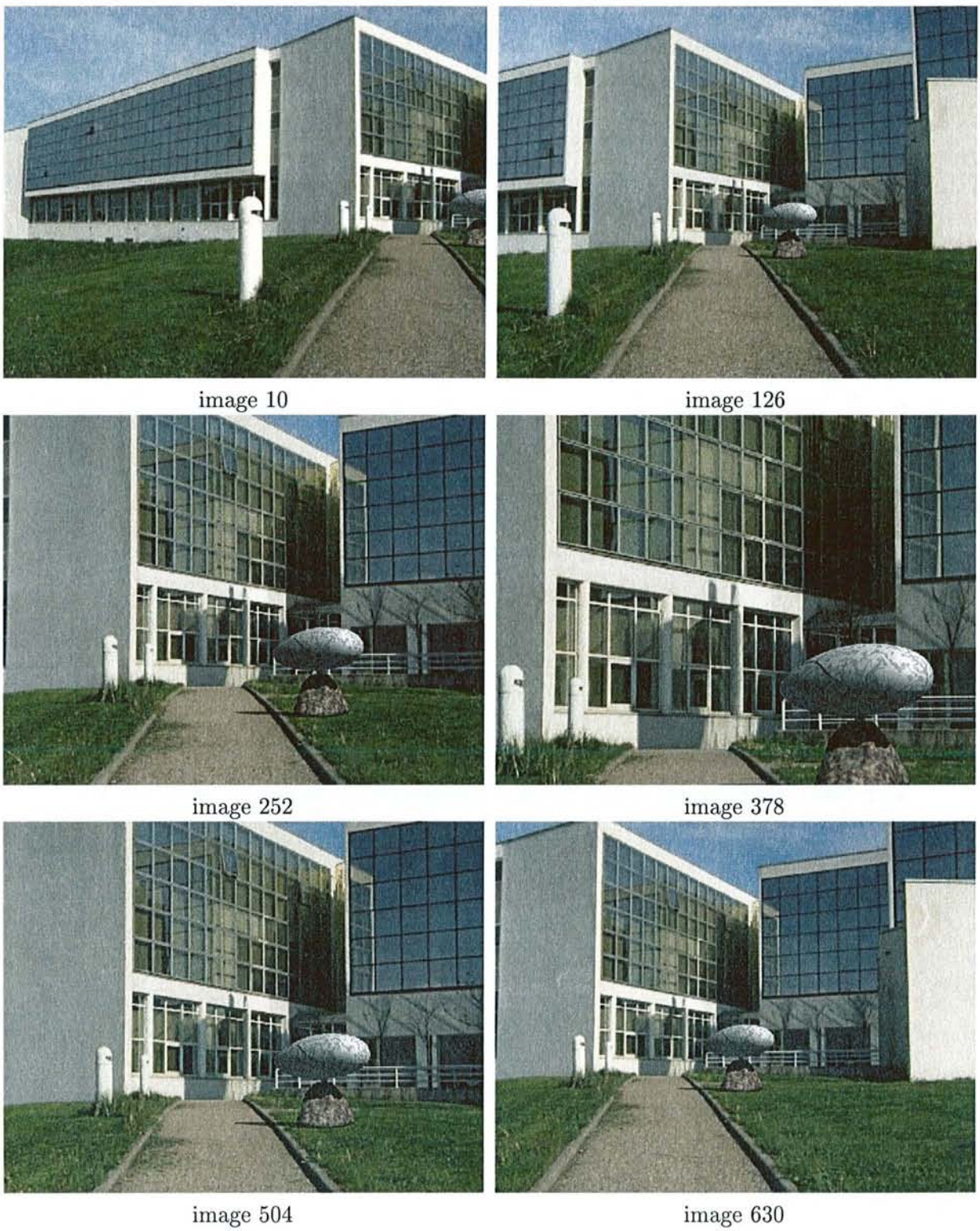
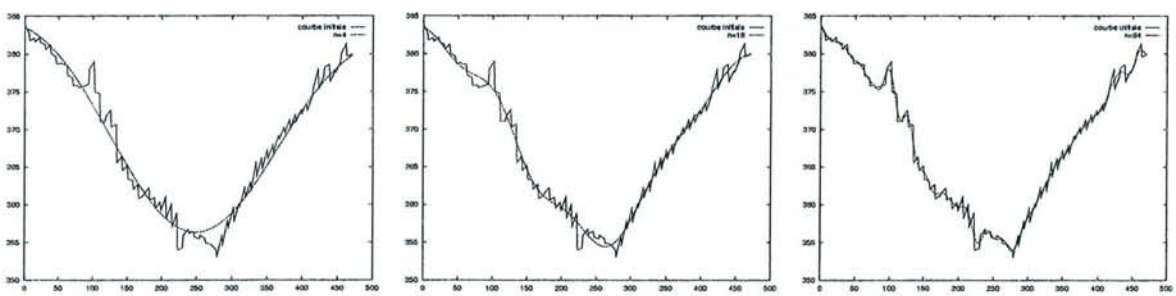


FIG. 7.31 – *Incrustation d'une sculpture d'Art Moderne dans la séquence du loria.*



$n = 4$ $n = 15$ $n = 60$

FIG. 7.32 – *Transformée de Fourier de la courbe u_0 pour différentes valeurs de n.*

Chapitre 8

Conclusion

8.1 Apports de la thèse

Dans cette thèse, nous avons essayé d'apporter notre contribution au problème du recalage temporel dans un contexte de Réalité Augmentée. Pour cela, nous avons proposé un certain nombre d'algorithmes, qui ont par ailleurs fait l'objet de publications dont nous donnons ici les références :

- une méthode statistique robuste à deux niveaux, calculant le point de vue de la caméra à partir d'appariements 3-D/2-D de courbes de forme libre [Simon et al.98a]. Cette méthode s'intègre dans une boucle de recalage temporel autonome et séquentielle ;
- une méthode hybride prenant en compte à la fois des appariements 3-D/2-D de courbes de forme libre et des appariements 2-D/2-D de points d'intérêt [Simon et al.98b, Simon et al.99b]. Cette méthode exploite la distribution aléatoire de points de l'image, afin d'améliorer le calcul du point de vue lorsque les primitives 3-D/2-D sont mal distribuées. Elle permet ainsi l'incrustation d'objets virtuels dans des zones éloignées des primitives suivies ;
- un algorithme de segmentation de la séquence en plans à focale fixe et plans à caméra fixe, puis une méthode de recalage alterné suivant le type de plan considéré [Simon et al.99a]. Cette méthode suppose que la séquence peut effectivement être décomposée en plans de zoom et plans de mouvement, et qu'aucun objet mobile n'est présent dans la scène.

Au travers de nos expérimentations, nous avons par ailleurs identifié un certains nombre de points problématiques qui devraient être pris en compte dans tout processus de recalage pour la Réalité Augmentée :

- plus les objets virtuels sont éloignés des primitives du modèle, plus leur insertion dans l'image est sensible à l'imprécision obtenue sur le recalage ;
- l'évaluation visuelle ou numérique de la reprojection du modèle utilisé pour le recalage est insuffisante pour juger de la pertinence des points de vues pour la composition. Les courbes d'évolution des paramètres de la caméra au cours de la séquence donnent une indication de la qualité du recalage, mais la visualisation de la séquence augmentée reste le meilleur moyen de valider l'algorithme ;
- les paramètres intrinsèques obtenus à partir de points d'une image de la séquence ne sont pas nécessairement valides pour les autres images de cette séquence, même lorsqu'aucun zoom n'est opéré. Ceci peut s'avérer problématique pour le recalage, notamment sur de longues séquences ;
- le lissage *a posteriori* des paramètres de la caméra ne résout pas nécessairement le problème des effets de "sautillements" qui peuvent apparaître au niveau de l'objet virtuel lorsque les

paramètres calculés sont saccadés. Pour un mouvement de caméra continu par morceaux, nous pouvons utiliser des filtres comme la transformée de Fourier, mais les "sautillements" peuvent alors être remplacés par un effet de "glissement".

Les deux premiers points ont été pris en compte dans la thèse, mais les deux autres problèmes restent ouverts.

8.2 Futurs axes de recherche

Nous souhaitons axer nos recherches futures sur la détection des objets mobiles dans la scène, le problème du lissage des paramètres de la caméra et les aspects temps réel.

8.2.1 Détection des objets mobiles

La présence d'objets mobiles dans la scène perturbe la détection des plans de zoom, ce qui limite le champ des applications potentielles pour les séquences à focale variable. Il serait donc intéressant de pouvoir détecter les objets mobiles présents dans la scène, afin de ne pas les prendre en compte dans l'algorithme de segmentation.

Il s'agit d'un problème délicat, qui n'a pour le moment été résolu que dans des cadres bien contraints : Yokoya et al. utilisent deux caméras pour détecter les mouvements du visage ou de la main d'une personne par triangulation, mais se basent sur la couleur de la peau pour détecter ces éléments [Yokoya et al.99]. Dans [Kanade et al.99], Kanade et al. construisent un modèle dynamique de personnes jouant avec une balle, mais utilisent 51 caméras pour la reconstruction. De Murcia et al. détectent en temps réel des objets en mouvement, mais utilisent une caméra posée sur un trépied [Murcia et al.97].

Une détection des incohérences du flux optique nous permettra peut-être de récupérer les objets en mouvement dans un cadre plus général [Carlsson et al.90, Thompson et al.90].

8.2.2 Lissage des paramètres de la caméra

Le lissage des paramètres de la caméra est un problème intéressant. Pour un mouvement de caméra saccadé, il est difficile d'appliquer un lissage : par exemple lorsque la caméra est tenue par un piéton, chaque pas correspond à une variation abrupte des paramètres du point de vue. Pour un mouvement de caméra lisse (au moins par morceaux) nous pouvons par contre utiliser des filtres hors ligne, mais le degré de lissage est un paramètre qui ne doit pas être choisi indépendamment de l'image. Un lissage plus intelligent tenant compte de la nature du problème devrait être envisagé.

8.2.3 Application en temps réel

L'un des nouveaux défis de la RA est aujourd'hui de prendre en compte la structure naturelle de la scène pour le recalage en temps réel [Behringer et al.99]. Notre algorithme de recalage temporel utilise la structure naturelle de la scène et est à la fois autonome et séquentiel. Il se positionne donc parmi les algorithmes les plus avancés vers cet objectif. Il nous reste maintenant à le faire fonctionner effectivement en temps réel.

Pour cela, nous pouvons dans un premier temps concentrer nos efforts sur l'optimisation des calculs et la parallélisation des tâches. Le processus de parallélisation est facilité par le fait que chaque primitive utilisée pour le recalage est considérée comme une entité indépendante. Ainsi, son suivi et le calcul de son résidu peuvent être effectués parallèlement aux autres primitives.

Si cet effort ne suffisait pas à faire fonctionner notre algorithme en temps réel sur les machines usuelles, nous envisagerions alors de court-circuiter l'étape de suivi des primitives : pour cela, nous minimiserions la distance des primitives aux gradients forts de l'image, directement en fonction des paramètres du point de vue. C'est ce qu'a fait Luc Robert dans le cadre de la calibration de la caméra à partir d'une mire, obtenant d'ailleurs des résultats très précis puisque non liés à l'incertitude sur l'extraction des primitives [Robert94]. Il nous faudra tout de même étudier la convergence du modèle, et en particulier vérifier que les primitives apparaissant dans l'image ne convergent pas vers un minimum local.

De nouvelles applications s'offriraient alors à nous, comme l'étude d'impact au travers d'un HMD (d'autant que des techniques de rendu réaliste en temps réel sont aujourd'hui opérationnelles [Wojdala99]). Inversement, nous pouvons imaginer avec Azuma que des visiteurs de sites archéologiques munis d'un HMD puissent visualiser les sites tels qu'ils étaient dans le passé [Azuma97]. Plus généralement, nous pourrions apporter de la sémantique à l'espace environnant, pour le tourisme ou l'aide à la navigation.

Bibliographie

[Ahlers et al.95] Ahlers (K. H.), Kramer (A.), Breen (D. E.), Chevalier (P.-Y.), Crampton (C.), Rose (E.), Tuceryan (M.), Whitaker (R. T.) et Greer (D.). – Distributed Augmented Reality for Collaborative Design Applications. *In : Proceedings of Eurographics'95*, pp. C–03–C–14.

[Akaike74] Akaike (H.). – A new look at the statistical model identification. *IEEE Trans Aut Ctrl*, vol. 19 (6), 1974, pp. 716–723.

[Armstrong et al.95] Armstrong (M.) et Zisserman (A.). – Robust Object Tracking. *In : In Proc. Asian Conference on Computer Vision, vol. I*, pp. 58–61.

[Azuma et al.94] Azuma (R. T.) et Bishop (G.). – Improving static and dynamic registration in an optical see through display. *Computer Graphics*, July 1994, pp. 194–204.

[Azuma97] Azuma (R. T.). – A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, vol. 6 (4), August 1997, pp. 355–385.

[Beardsley et al.94] Beardsley (P.), Zisserman (A.) et Murray (D. W.). – Navigation using affine structure and motion. *In : Proceedings of 3rd European Conference on Computer Vision, Stockholm (Sweden)*, pp. 85–96.

[Behringer et al.99] Behringer (R.), Klinker (G.) et Mizell (D.). – International Workshop on Augmented Reality. *VR News*, vol. 8 (1), Jan./Feb. 1999, pp. 18–20.

[Berger et al.96a] Berger (M.-O.), Chevrier (C.) et Simon (G.). – Compositing Computer and Video Image Sequences: Robust Algorithms for the Reconstruction of the Camera Parameters. *In : Computer Graphics Forum, Conference Issue Eurographics'96, Poitiers, France*, pp. 23–32.

[Berger et al.96b] Berger (M.-O.), Simon (G.), Petitjean (S.) et Wrobel-Dautcourt (B.). – Mixing Synthesis and Video Images of Outdoor Environments: Application to the Bridges of Paris. *In : Proceedings of the 13th International Conference on Pattern Recognition, Vienna (Austria)*, pp. 90–94.

[Berger et al.99] Berger (M.-O.), Wrobel-Dautcourt (B.), Petitjean (S.) et Simon (G.). – Mixing Synthetic and Video Images of an Outdoor Urban Environment. *Machine Vision and Applications*, vol. 11 (3), 1999.

[Berger94] Berger (M.-O.). – How to Track Efficiently Piecewise Curved Contours with a View to Reconstructing 3D Objects. *In : Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem (Israel)*, pp. 32–36.

[Berger97] Berger (M.-O.). – Resolving occlusion in augmented reality: a contour-based approach without 3d reconstruction. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 91–96.

[Bougnoux97] Bougnoux (S.). – TotalCalib: a fast and reliable system for off-line calibration of image sequences. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA).*

[Bougnoux98] Bougnoux (S.). – From Projective to Euclidiean Space under any Practical Situation, a Criticism of Self-calibration. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pp. 790–796.

[Brand et al.94] Brand (P.), Mohr (R.) et Bobet (P.). – Distorsions optiques : correction dans un modèle projectif. *In : Actes du 9e Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Paris*, pp. 87–98.

[Breen et al.95] Breen (D.), Rose (E.) et Whitaker (R.). – *Interactive Occlusion and Collision of Real and Virtual Objects in Augmented Reality.* – Technical report, ECRC, Munich, 1995.

[Carlsson et al.90] Carlsson (S.) et Eklundh (J. O.). – Object detection uding based prediction and motion parallax. *In : Proceedings of 1st European Conference on Computer Vision, Antibes (France)*, pp. 297–306.

[Cazier et al.94] Cazier (D.), Chamont (D.), Deville (P.M.) et Paul (J.-C.). – Modeling characteristics of light: a method based on measured data. *In : Proceedings of the Second Pacific Conference on Computer Graphics and Applications*, pp. 113–128.

[Chaumette et al.89] Chaumette (F.) et Rives (P.). – *Realisation et calibration d'un système experimental de vision composé d'une caméra embarquée sur un robot-manipulateur.* – Technical report, IRISA/INRIA - Rennes, Campusde Beaulieu 35042 Rennes Cedex, INRIA, March 1989.

[Chevrier et al.95] Chevrier (C.), Belblidia (S.) et Paul (J.-C.). – Compositing Computer Generated Images and Video Films: An application for Visual Assessment in Urban Environments. *In : Computer Graphics: Development in Virtual Environments*, éd. par Earnshaw (R. A.) et Vince (J. A.), pp. 115–125. – Academic Press, June 1995.

[Chevrier96] Chevrier (C.). – *Génération de séquences composées d'images de synthèse et d'images vidéo.* – Vandœuvre-lès-Nancy, Thèse de doctorat, Université Henri Poincaré Nancy I, June 1996.

[Curtis et al.98] Curtis (D.), Mizell (D.), Gruenbaum (P.) et Janin (A.). – Several Devils in the Details: Making an AR App Work in the Airplanee Factory. *In : First International Workshop on Augmented Reality, San Francisco.*

[Debevec et al.96] Debevec (P. E.), Taylor (C. J.) et Malik (J.). – Modeling and Rendering Architecture from Photographs. *In : Proc. SIGGRAPH.*

[Dementhon et al.95] Dementhon (D.) et Davis (L.). – Model Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, vol. 15, 1995, pp. 123–141.

[Deriche87] Deriche (R.). – Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, vol. 1 (2), 1987, pp. 167–187.

[Dhome et al.89] Dhome (M.), Richetin (M.), Lapresté (J.T.) et Rives (G.). – Determination of the Attitude of 3-D Objects from a Single Perspective View. *IEEE Transactions on PAMI*, vol. 11 (12), 1989, pp. 1265–1278.

[Drettakis et al.97] Drettakis (G.), Robert (L.) et Bougnoux (S.). – Interactive Common Illumination for Computer Augmented Reality. *In : 8th Eurographics workshop on Rendering, St. Etienne, France.*

[Enciso et al.93] Enciso (R.), Viéville (T.) et Faugeras (O.). – *Approximation du Changement de Focale et de Mise au Point par une Transformation Affine à Trois Paramètres.* – Rapport de recherche 2071, INRIA, 1993.

[Ertl et al.91] Ertl (G.), Müller-Seelich (H.) et Tabatabai (B.). – MOVE-X: A System for Combining Video Films and Computer Animation. *In : Eurographics*, pp. 305–313.

[Fasse et al.94] Fasse (I.), Paulo (S. Santo), Perrin (J.-P.) et Paul (J.-C.). – Accurate synthesis images for architectural design. *In : Proceedings of the Conference on Multimedia for Architecture and Urban Design*, pp. 229–243.

[Faugeras et al.86] Faugeras (O. D.) et Toscani (G.). – The Calibration Problem for Stereo. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL (USA)*, pp. 15–20.

[Faugeras92] Faugeras (O. D.). – What Can Be Seen in Three Dimension with an Uncalibrated Stereo Rig? *In : Proceedings of 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 563–578.

[Faugeras93] Faugeras (O.). – *Three-Dimensional Computer Vision: A Geometric Viewpoint.* – MIT Press, 1993, *Artificial Intelligence.*

[Faugeras98] Faugeras (O.). – De la géométrie au calcul variationnel : théorie et applications de la vision tridimentionnelle. *In : Actes du 11e Congrès de Reconnaissance des Formes et Intelligence Artificielle (RFIA'98), Clermont-Ferrand*, pp. 15–34.

[Feiner et al.93] Feiner (S.), MacIntyre (B.) et Seligmann (D.). – Knowledge-based Augmented Reality. *Communications of the ACM*, vol. 36 (7), July 1993, pp. 52–62.

[Feldmar et al.97] Feldmar (J.), Ayache (N.) et Betting (F.). – 3D-2D Projective Registration of Free Form Curves and Surfaces. *Computer Vision and Image Understanding*, vol. 65 (3), 1997, pp. 403–424.

[Ferri et al.93] Ferri (M.), Mangili (F.) et Viano (G.). – Projective Pose Estimation of Linear and Quadratic Primitives in Monocular Computer Vision. *CVGIP: Image Understanding*, vol. 58 (1), July 1993, pp. 66–84.

[Fitzgibbon et al.98] Fitzgibbon (A.W.) et Zisserman (A.). – Automatic Camera Recovery for Closed or Open Images Sequences. *In : Proceedings of 5th European Conference on Computer Vision, University of Freiburg (Germany)*, pp. 311–326.

[Gagalowicz99] Gagalowicz (A.). – Use of Image Analysis/Synthesis Techniques for 3D Object Tracking. *In : Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia.*

[Garai et al.99] Garai (G.) et Chaudhuri (B.). – A Split and Merge Procedure for Polygonal Detection of Dot Pattern. *Image and Vision Computing*, vol. 17, 1999, pp. 75–82.

[Gennery92] Gennery (D.). – Visual Tracking of Known Three Dimensional Objects. *International Journal of Computer Vision*, vol. 7 (3), 1992, pp. 243–270.

[Haralick et al.89] Haralick (R. M.), Joo (H.), Lee (C. N.), Zhuang (X.), Vaidya (V.G.) et

Kim (M. B.). – Pose Estimation from Corresponding Point Data. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19 (6), 1989.

[Haralick et al.91] Haralick (R. M.), Lee (C.), Ottenberg (K.) et Nölle (M.). – Analysis and Solutions of The Three Point Perspective Pose Estimation Problem. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Maui, HI (USA)*, pp. 592–598.

[Harris et al.88] Harris (C.) et Stephens (M.). – A Combined Corner and Edge Detector. *In : Proceedings of 4th Alvey Conference.* – Cambridge, August 1988.

[Harris92] Harris (C.J.). – Tracking with Rigid Models. *In : Active Vision*, chap. 4. – Blake and Yuille, MIT Press, 1992.

[Hartley97] Hartley (Richard I.). – Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, vol. 22 (2), March 1997, pp. 125–140.

[Heyden et al.97] Heyden (A.) et Aström (K.). – Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 438–443.

[Horn et al.81] Horn (B.) et Schunck (B.). – Determining Optical Flow. *Artificial Intelligence*, vol. 17, 1981, pp. 185–203.

[Huber81] Huber (P. J.). – *Robust Statistics.* – Wiley, New York, 1981.

[Hung et al.90] Hung (Y.-P.) et Shieh (S.-W.). – When Should we Consider Lens Distorsion in Camera Calibration. *In : Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo (Japan).* IAPR, pp. 367–370.

[Jancene et al.95] Jancène (P.), Neyret (F.), Provot (X.), Tarel (J. P.), vezien (J. J.), Meilhac (C.) et Verroust (A.). – *RES: computing the interactions between real and virtual objects in video sequences.* – Technical report, INRIA, 1995.

[Jazwinsky70] Jazwinsky (A. M.). – *Stochastic Process and Filtering Theory.* – Academic Press, 1970.

[Kanade et al.99] Kanade (T.), Rander (P.), Vedula (S.) et Saito (H.). – Virtualized reality: Digitizing a 3d time-varying event as is and in real time. *In : Proc. ISMR'99 (International Symposium on Mixed Reality)*, pp. 41–57. – Yokohama, Japan, March 1999.

[Kanatani96] Kanatani (K.). – *Statistical Optimization for Geometric Computation.* – Amsterdam, Elsevier Science, 1996.

[Kass et al.88] Kass (M.), Witkin (A.) et Terzopoulos (D.). – Snakes: Active Contour Models. *International Journal of Computer Vision*, vol. 1, 1988, pp. 321–331.

[Kerrien et al.91] Kerrien (E.), Berger (M.O.) et Vaillant (R.). – *Etude de la précision de la machine pour le recalage 2D/3D d'images d'angiographie soustraite .* – Technical report, Inria Sophia Antipolis, 1991. Séminaire Orasis, La Colle sur Loup, 6-10 Octobre 1997.

[Kerrien et al.99] Kerrien (E.), Berger (M.-O.), Mauricomme (E.), Launay (L.), Vaillant (R.) et Picard (L.). – Fully Automatic 3D/2D Substracted Angiography Registration. *In : Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, pp. 664–671.

[Kim et al.89] Kim (D. Y.), Kim (J. J.), Meer (P.), Mintz (D.) et feld (A. Rosen). – Robust computer vision: A least median of squares based approach. *In: The DARPA Image Understanding Workshop.* – Palo Alto, CA, May 1989.

[Koller et al.92] Koller (D.), Daniilidis (K.) et Nagel (H. H.). – Model-Based Object Tracking in Traffic Scenes. *In: Proceedings of 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 437–452.

[Kriegman et al.90] Kriegman (D.) et Ponce (J.). – On Recognizing and Positioning Curved 3D Objects from Image Contours. *IEEE Transactions on PAMI*, vol. 12 (12), December 1990, pp. 1127–1137.

[Kumar et al.94] Kumar (R.) et Hanson (A.). – Robust Methods for Estimating Pose and a Sensitivity Analysis. *CVGIP: Image Understanding*, vol. 60 (3), 1994, pp. 313–342.

[Kutulakos et al.96] Kutulakos (K. N.) et Vallino (J. R.). – Affine Object Representations for Calibration-Free Augmented Reality. *In: Proceedings of 1996 IEEE Virtual Reality Annual International Symposium*, pp. 25–36.

[Lavest et al.92] Lavest (J. M.), Rives (G.) et Dhome (M.). – Utilisation d'un objectif à focale variable en vision monoculaire en vue de la reconstruction 3D. *Traitement du Signal*, vol. 9 (6), 1992, pp. 491–506.

[Li et al.95] Li (Mengxiang) et Lavest (Jean-Marc). – *Some Aspects of Zoom-Lens Camera Calibration.* – Technical Report TRITA-NA-9503, Stockholm, Sweden, CVAP, Royal Institute of Technology, 1995.

[Lowe85] Lowe (D. G.). – *Perceptual Organization and Visual Recognition.* – Norwell, Massachusetts, Kluwer Academic Publishers, 1985.

[Lowe90] Lowe (D. G.). – Integrated treatment of matching and measurement errors for robust model-based motion tracking. *In: Proceedings of 3rd International Conference on Computer Vision, Osaka (Japan)*, pp. 436–440.

[Lowe92] Lowe (D.). – Robust Model based Motion Tracking Through the Integration of Search and Estimation. *International Journal of Computer Vision*, vol. 8 (2), 1992, pp. 113–122.

[Luong92] Luong (Q. T.). – *Matrice fondamentale et calibration visuelle sur l'environnement, vers une plus grande autonomie des systèmes robotiques.* – Thèse de doctorat, Université de Paris Sud, centre d'Orsay, December 1992.

[Maver et al.85] Maver (T. W.), Purdie (C.) et Stearn (D.). – Visual Impact Analysis — Modelling and Viewing the Natural and Built Environment. *Comput. & Graphics*, vol. 9 (2), 1985, pp. 117–124.

[Maybank et al.92] Maybank (S. J.) et Faugeras (O. D.). – A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, vol. 8 (2), 1992, pp. 123–152.

[Mclauchlan et al.94] McLauchlan (P. F.), Reid (I. D.) et Murray (D. W.). – Recursive affine structure and motion from image sequences. *In: Proceedings of 3rd European Conference on Computer Vision, Stockholm (Sweden)*, pp. 217–224.

[Mellor95] Mellor (J. P.). – Realtime Camera Calibration for Enhanced Reality Visualization. *In: Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine'95 (CVRMed'95)*, pp. 471–475.

[Murcia et al.97] Murcia (C. De), Niemaz (M.) et Viéville (T.). – Détection et suivi de cibles sur une durée indéterminée. *In: Journées Orasis'97.*

[Nakamae et al.86] Nakamae (E.), Harada (K.), Ishizaki (T.) et Nishita (T.). – A Montage Method: the Overlaying of the Computer Generated Images onto a Background Photograph. *ACM Computer Graphics*, vol. 20 (4), August 1986, pp. 207–211. – Proc. SIGGRAPH, Dallas.

[Oliensis97] Oliensis (J.). – *A Critique of Structure from Motion Algorithms.* – Technical report, NECI, April 1997.

[Ong et al.98] Ong (K. C.), Teh (H. C.) et Tan (T. S.). – Resolving occlusion in image sequence made easy. *The Visual Computer*, vol. 14, 1998, pp. 153–165.

[Peuchot94] Peuchot (B.). – Utilisation de détecteurs subpixels dans la modélisation d'une caméra. *In : Actes du 9e Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Paris*, pp. 691–695.

[Peuchot95] Peuchot (B.). – Virtual Reality As An Operative Tool During Scoliosis Surgery. *In : Proceedings of Imagina, Monte-Carlo, France*, p. 262.

[Press et al.88] Press (W. H.), Flannery (B. P.), Teukolsky (S. A.) et Vetterling (W. T.). – *Numerical Recipes in C, The Art of Scientific Computing.* – Cambridge University Press, 1988.

[Quan et al.98] Quan (L.) et Lan (Z.). – Linear $N \geq 4$-Point Pose Determination. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India).*

[Ravela et al.96] Ravela (S.), Draper (B.), Lim (J.) et Weiss (R.). – Tracking Object Motion Across Aspect Changes for Augmented Reality. *In : ARPA Image Undertanding Worshop, Palm Spring (USA).*

[Reiners et al.98] Reiners (D.), Stricker (S.), Klinker (G.) et Müller (S.). – Augmented Reality for Construction Tasks: Doorlock Assembly. *In : First International Workshop on Augmented Reality, San Francisco.*

[Robert94] Robert (L.). – *Camera Calibration without Feature Extraction.* – Rapport de recherche 2204, INRIA, 1994.

[Rose et al.94] Rose (E.), Breen (D.), Ahlers (K.), Crampton (C.), Tuceyran (M.), Whitaker (R.) et Greer (D.). – *Annotating Real-World Objects Using Augmented Reality.* – Technical report, ECRC, Munich, 1994.

[Roth et al.99] Roth (M.), Brack (C.), Burgkart (R.), Czopf (A.), Götte (H.) et Schweikard (A.). – Multi-view contourless registration of bone structures using a single calibrated X-ray fluoroscope. *In : Computer Assisted Radiology and Surgery (CARS'99)*, pp. 756–761.

[Rousseeuw et al.87] Rousseeuw (P.) et Leroy (A.). – *Robust Regression and Outlier Detection.* – Wiley, 1987, *Wiley Series in Probability and Mathematical Statistics.*

[Satoh et al.98] Satoh (K.), Ohshima (T.) et Yamamoto (H.). – Case Studies of See-Through Augmentation in Mixed Reality Project. *In : First International Workshop on Augmented Reality, San Francisco.*

[Shakunaga93] Shakunaga (T.). – Robust Line Based Pose Enumeration From a Single Image. *In : Proceedings of 4th International Conference on Computer Vision, Berlin (Germany)*, pp. 545–550.

[Simon et al.98a] Simon (G.) et Berger (M.-O.). – A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pp. 261–266.

[Simon et al.98b] Simon (G.), Lepetit (V.) et Berger (M.-O.). – Computer Vision Methods for Registration: Mixing 3D Knowledge and 2D Correspondences for Accurate Image Composition. *In : First International Workshop on Augmented Reality, San Francisco.*

[Simon et al.99a] Simon (G.) et Berger (M.-O.). – Registration with a Zoom Lens Camera for Augmented Reality Applications. *In : Second International Workshop on Augmented Reality, San Francisco.*

[Simon et al.99b] Simon (G.), Lepetit (V.) et Berger (M.-O.). – Registration Methods for Harmonious Integration of Real Worlds and Computer Generated Objects. *In : Computer Graphics Forum, Conference Issue Eurographics'99, Milano, Italy.*

[Simon95] Simon (G.). – *Détermination du point de vue à partir d'une observation d'un objet 3D dont le modèle est connu.* – Rapport de DEA, Université Henri Poincaré Nancy I, September 1995.

[Sparr96] Sparr (G.). – Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. *In : Proceedings of the 13th International Conference on Pattern Recognition, Vienna (Austria)*, pp. 328–333.

[State et al.94] State (A.), Chen (D. T.), Tector (C.), Brandt (A.), Chen (H.), Ohbuchi (R.), Bajura (M.) et Fuchs (H.). – Case Study: Observing a Volume Rendered Fetus within a Pregnant. *In : Proceedings of IEEE Visualization*, pp. 364–368.

[State et al.96] State (A.), Hirota (G.), Chen (D.), garett (W.) et Livingston (M.). – Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. *In : Computer Graphics (Proceedings Siggraph New Orleans)*, pp. 429–438.

[Stoer et al.80] Stoer (J.) et Bulirsch (R.). – *Introduction to Numerical Analysis.* – New York, Springer-Verlag, 1980.

[Stricker et al.98] Stricker (D.), Klinker (G.) et Reiners (D.). – A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications. *In : First International Workshop on Augmented Reality, San Francisco.*

[Sturm et al.96] Sturm (P.) et Triggs (W.). – A factorization based algorithm for multi-image projective structure and motion. *In : Proceedings of 4th European Conference on Computer Vision, Cambridge (United Kingdom)*, pp. 709–720.

[Sturm96] Sturm (Peter). – Self Calibration of a moving Zoom Lens Camera by Pre-Calibration. *In : British Machine Vision Conference, Edinburgh, Scotland*, pp. 675–684.

[Sturm97] Sturm (P.). – Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 1100–1105.

[Sudhir et al.97] Sudhir (G.) et Lee (J. C. M.). – *Video Annotation by Motion Interpretation using Optical Flow Streams.* – Technical Report HKUST-CS96-16, Kowloon, Hong Kong, The Hong Kong University of Science and Technology, department of Computer Science, July 1997.

[Thalmann et al.97] Thalmann (N. M.) et Thalmann (D.). – Animating Virtual Actors in Real Environments. *ACMMS'97*, vol. 5 (2), 1997, pp. 113–125.

[Thompson et al.90] Thompson (W. B.) et Pong (T-C.). – Detecting Moving Objects. *International Journal of Computer Vision*, vol. 4, 1990, pp. 39–57.

[Tomasi et al.92] Tomasi (C.) et Kanade (T.). – Shape and Motion from Image Streams under Orthography: A Factorization Method. *International Journal of Computer Vision*, vol. 9 (2), 1992, pp. 137–154.

[Torr et al.97] Torr (P.H.S.) et Zisserman (A.). – Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, vol. 15, 1997, pp. 591–605.

[Torr97] Torr (P.H.S.). – An Assessment of Information Criteria for Motion Model Selection. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 47–52.

[Toscani87] Toscani (G.). – *Systèmes de Calibration et Perception du Mouvement en Vision Artificielle.* – Thèse de Doctorat, Université de Paris-Sud, Orsay, 1987.

[Tsai et al.84] Tsai (R. Y.) et Huang (T. S.). – Uniqueness and estimation of 3D motion parameters of rigid bodies with curved surfaces. *IEEE Transactions on PAMI*, vol. 6, 1984, pp. 13–27.

[Tsai87] Tsai (R. Y.). – A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, vol. 3 (4), August 1987, pp. 323–331.

[Uenohara et al.96] Uenohara (M.) et Kanade (T.). – Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medecine*, 1996.

[Ullman et al.91] Ullman (S.) et Basri (R.). – Recognition by Linear Combinations of Models. *IEEE Transactions on PAMI*, vol. 13 (10), 1991, pp. 992–1006.

[Uno et al.79] Uno (S.) et Matsuka (H.). – A general purpose graphic system for computer-aided design. *Computer Graphics Proceedings, Annual Conference Series*, vol. 13 (2), 1979, pp. 25–32.

[Vaz97] Vaz (M. C.). – Star Wars Trilogy: Everything Old Is New Again. *Cinefex magazine*, no69, March 1997, pp. 15–30.

[Weng et al.92] Weng (J.), Cohen (P.) et Herniou (M.). – Camera Calibration with Distorsion Models and Accuracy Evaluation. *IEEE Transactions on PAMI*, vol. 14 (10), 1992, pp. 965–980.

[Willson et al.93] Willson (R. G.) et Shafer (S. A.). – What is the Center of the Image? *In : cvpr93*, pp. 670–671.

[Wloka et al.95] Wloka (M.) et Anderson (B.). – Resolving Occlusions in Augmented Reality. *In : Symposium on Interactive 3D Graphics Proceedings, (New York)*, pp. 5–12.

[Wojdala99] Wojdala (A.). – Can virtual look real? a review of virtual studio techniques. *In : Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia.*

[Wrobel92] Wrobel (B.P.). – *Calibration and Orientation of Cameras in Computer Vision.* – Washington DC, Springer-Verlag, August 1992.

[Xiong et al.98] Xiong (W.) et Lee (J.C.M.). – Efficient scene change detection and camera motion annotation for video classification. *Computer Vision and Image Understanding*, vol. 71 (2), 1998, pp. 166–181.

[Yokoya et al.99] Yokoya (N.), Takemura (H.), Okuma (T.) et Kanbara (M.). – Stereo vision based video see-through mixed reality. *In : Proc. ISMR'99 (International Symposium on Mixed Reality)*, pp. 131–145. – Yokohama, Japan, March 1999.

[Zeller et al.96] Zeller (Cyril) et Faugeras (Olivier). – *Camera Self-Calibration from Video Sequences: the Kruppa Equations Revisited.* – Rapport de recherche 2793, INRIA, February 1996.

[Zhang et al.94] Zhang (Z.), Deriche (R.), Faugeras (O.) et Luong (Q.). – *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry.* – Rapport de recherche 2273, INRIA, 1994.

[Zhang et al.95] Zhang (Z.), Deriche (R.), Faugeras (O.) et Luong (Q.). – A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, vol. 78, October 1995, pp. 87–119.

[Zhang et al.97] Zhang (Z.), Faugeras (O.) et Deriche (R.). – An Effective Technique for Calibrating a Binocular Stereo Through Projective Reconstruction Using Both a Calibration Object and the Environment. *Videre: A Journal of Computer Vision Research (MIT Press)*, vol. 1 (1), 1997, pp. 58–68.

[Zisserman et al.99] Zisserman (A.), Fitzgibbon (A.) et Cross (G.). – VHS to VRML: 3D graphical models from video sequences. *In : Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia.*

Monsieur **SIMON Gilles**

DOCTORAT de l'UNIVERSITE HENRI POINCARE, NANCY-I

en INFORMATIQUE

VU, APPROUVÉ ET PERMIS D'IMPRIMER

Nancy, le 14 DÉC 1999 n° 339

Le Président de l'Université

Cl. BURLET

3C-F9

UNIVERSITÉ HENRI POINCARÉ · NANCY 1
24-30, rue Lionnois B.P. 3069 54013 Nancy cedex - Tél. 83 85 48 00 - Fax 83 85 48 48

Adresse électronique : _____ @presn1.u-nancy.fr

Résumé

Le concept de Réalité Augmentée (RA) vise à accroître la perception du monde réel en y ajoutant des éléments non perceptibles *a priori* par l'œil humain. Un des problèmes majeurs qui se posent dans ce type d'études est de pouvoir déterminer le point de vue adopté pour chaque prise de vue (alignement des caméras réelle et virtuelle) afin d'incruster l'objet de synthèse au bon endroit. Ce calcul peut être exécuté en temps réel ou en phase de post-production sur une séquence pré-acquise. Plusieurs méthodes reposent sur la connaissance du modèle d'un certain nombre de primitives présentes dans la scène observée (marqueurs artificiels ou primitives naturelles de type points d'intérêt ou segments de droites). Ces méthodes sont autonomes et séquentielles, ce qui permet leur utilisation dans un processus en temps réel, mais leur précision est liée à la distribution des primitives utilisées. D'autres méthodes reposent uniquement sur des appariements de primitives entre images de la séquence. Celles-ci ne sont précises qu'au prix d'une prise en compte non séquentielle des images, et sont donc adaptées à la post-production.

Nous proposons dans cette thèse une méthode autonome et séquentielle essentiellement basée modèle, mais indépendante de la distribution des primitives de référence. Nous commençons par introduire une méthode à deux niveaux prenant en compte des appariements modèle/image de courbes quelconques. Cette méthode est robuste à la présence d'occultations dans l'image, ainsi qu'aux erreurs pouvant se produire lors du suivi des primitives image. Elle est validée aussi bien sur des données synthétiques que sur des applications en vraie grandeur, comme le projet d'illumination artificielle des ponts de Paris.

La méthode à deux niveaux fonctionne bien dans la plupart des cas, mais lorsque l'objet de référence est petit par rapport à la distance objet-caméra, le calcul du point de vue devient moins précis. Pour pallier ce problème, nous utilisons des appariements de points d'intérêt entre images consécutives de la séquence. Le point de vue de la caméra est alors calculé en minimisant une fonction de coût qui tient compte à la fois des appariements modèle/image de courbes et des appariements image/image de points. Nous exploitons ainsi la distribution aléatoire des points d'intérêt tout en conservant un système autonome et séquentiel grâce aux appariements modèle/image. L'algorithme est validé sur une séquence en vraie grandeur, la séquence de la place Stanislas.

Enfin, nous prenons en compte les changements de paramètres intrinsèques de la caméra qui peuvent se produire au cours de la séquence. Les méthodes calculant à la fois les paramètres internes et le déplacement de la caméra étant instables, nous avons opté pour un partitionnement automatique de la séquence en zooms et mouvements de caméra. Les paramètres de l'optimisation dépendent alors du type de plan considéré. Cette méthode suppose bien sûr que les paramètres intrinsèques et extrinsèques de la caméra ne varient effectivement pas en même temps au cours de la séquence. Elle est validée très précisément à partir d'une scène de laboratoire, et permet l'incrustation d'une œuvre d'art dans une séquence du bâtiment Loria.

Mots clés : Réalité Augmentée, vision par ordinateur, calcul du point de vue, recalage temporel, estimation robuste, géométrie épipolaire, zoom.