

Des méthodes efficaces pour l'incrustation d'objets virtuels dans des séquences d'images

Efficient methods for image composition in video sequences

par Gilles SIMON et Marie-Odile BERGER

Loria/ Inria Lorraine
BP 239, 54506 Vandoeuvre les Nancy

résumé et mots clés

Nous présentons dans cet article le système de réalité augmentée que nous avons réalisé. Notre objectif est de concevoir des outils et des méthodes permettant d'incruster des objets virtuels dans des séquences d'images le plus automatiquement possible.

Nous commençons par étudier les différents problèmes qui se posent lors de la composition d'images (calcul du point de vue, gestion des occultations entre les objets virtuels et la scène filmée, interactions photométriques entre monde réel et monde virtuel..). Puis nous décrivons dans le détail les solutions que nous avons apportées au problème du calcul automatique du point de vue et à celui de la résolution des occultations. Nous avons en particulier conçu un système autonome permettant un calcul robuste du point de vue à partir de la connaissance de certains éléments tri-dimensionnels de la scène. Nous montrons de plus comment notre système est capable de gérer d'éventuelles occultations entre les objets virtuels et la scène observée. Des exemples variés d'application grandeur réelle viennent illustrer nos travaux.

Recalage robuste, réalité augmentée, occultations.

abstract and key words

We present an augmented reality system for image composition. We have worked with a view to avoiding strong and tedious interactions with the user. Because accurate registration between real and virtual objects is crucial for augmented reality applications, we especially focus on the temporal model-based registration method we have devised. It integrates tracking, pose determination and updating of the visible features in a very robust way and is able to retain up to date the scene-camera pose relationships even when tracking error occurs.

While current augmented reality systems simply overlay computer-generated objects on real imagery, our system is capable of resolving occlusions between real and virtual objects for scenes where contour information is relevant.

We demonstrate the efficiency of our system on real experiments.

Robust registration, augmented reality, occlusions.

1. introduction

La réalité virtuelle, qui propose d'immerger l'utilisateur dans un environnement complètement calculé par ordinateur, a monopolisé l'attention des média depuis quelques années. Cependant,

l'obtention de mondes virtuels réalistes nécessite de disposer de modèles très précis de l'environnement et se révèle donc très coûteuse, surtout dans le cas d'environnements complexes. A l'opposé, la réalité augmentée cherche à améliorer ou à compléter la vision de l'utilisateur sans chercher à remplacer ou à synthétiser le monde réel. Par exemple, des systèmes dédiés à l'étude de

l'impact d'un nouveau bâtiment permettent de visualiser l'image de synthèse de la nouvelle construction sur une vidéo tournée sur le lieu d'implantation [7, 3]. D'autres systèmes ont été développés notamment dans le domaine médical [2, 16, 14] et dans le domaine de l'apprentissage [10].

Dans notre approche de la réalité augmentée, nous nous intéressons plus particulièrement à l'insertion d'objets, dont le modèle 3D est connu, dans une séquence d'images vidéo. Ce problème est crucial pour la plupart des systèmes de réalité augmentée. En effet, la réalisation de systèmes performants nécessite de mêler de façon convaincante les objets ajoutés (que nous appellerons virtuels) avec la scène. Bien que plusieurs systèmes soient actuellement prometteurs, de nombreux défis restent à relever avant que la conception d'applications de réalité augmentée puisse être considérée comme une tâche facile.

1.1. prérequis pour une composition réaliste

Le premier et peut être le plus important problème à résoudre est celui du recalage entre objets réels et objets virtuels. En effet, pour une visualisation correcte, l'image de l'objet virtuel doit être calculée avec la position de la caméra utilisée pour l'image considérée. Ce recalage doit être réalisé avec soin car l'oeil humain détecte très facilement les erreurs, même minimales, de recalage.

Cependant, il ne suffit pas d'assurer un bon recalage temporel sur la séquence pour obtenir une composition réaliste. D'autres facteurs interviennent également dans la perception réaliste d'une scène : une gestion correcte des occultations pouvant intervenir entre la scène et les objets virtuels est évidemment indispensable, bien que très peu de systèmes de réalité augmentée envisagent actuellement le problème. Enfin il serait souhaitable pour accroître encore le réalisme de la scène que les interactions photométriques entre les objets virtuels et la scène, les ombrages en particulier, soient pris en compte.

1.2. nos objectifs

On peut distinguer schématiquement deux grandes classes de systèmes de réalité augmentée. La première utilise des capteurs ou des solutions très interactives pour résoudre les problèmes qui se posent. Le problème du calcul de point de vue peut par exemple être résolu en utilisant des capteurs de positions (capteurs Polhemus) [1] qui permettent d'obtenir directement la position et l'orientation de la caméra à chaque instant. Cependant, cette approche ne peut être utilisée que dans des environnements de taille restreinte et elle est de plus sensible à la présence

de matériaux perturbant le bon fonctionnement du capteur (par exemple de fer dans le cas de capteurs magnétiques).

Pour des environnements plus vastes qui ne peuvent être instrumentés, une bonne solution est de recourir à un *suivi visuel* : des points sont identifiés puis suivis dans la séquence d'images et permettent le calcul du point de vue à chaque instant à condition de connaître quelques éléments 3D sur la scène. Cependant, un calcul fiable du point de vue nécessite que les points suivis soient très précisément identifiés dans chaque image. Aussi, de nombreux systèmes positionnent-ils des structures ou des balises aisément détectables dans la scène de manière à avoir une extraction facile et précise des points servant au calcul du point de vue [7]. Le système commercial « 3D Equalizer » par exemple, se base sur des points qui peuvent être facilement suivis par corrélation, ces points ayant été au préalable désignés par l'utilisateur dans deux images de la séquence.

L'intérêt des approches de nature vision est de ne nécessiter aucune instrumentation de la scène. Ces approches sont cependant tributaires de la détection et de la mise en correspondance de primitives permettant le calcul du point de vue. La plupart des systèmes existants utilisent donc soit des environnements suffisamment simples dans lesquels des points peuvent être facilement extraits et suivis, soit positionnent des balises dans des environnements plus complexes afin de permettre un calcul facile du point de vue. Si l'on veut être capable de traiter des applications de réalité augmentée dans des environnements complexes, et notamment en extérieur, il faut pouvoir identifier et suivre des structures naturelles de la scène; en effet il est la plupart du temps impossible de positionner des balises en extérieur et par ailleurs, le champ de vision, généralement très large, couvert pendant le mouvement de la caméra, imposerait un trop grand nombre de balises.

Idéalement, un système de réalité augmentée devrait fonctionner sans que l'utilisateur ait à intervenir pour corriger les erreurs éventuelles du système. Cet objectif est évidemment utopique, sauf si la scène est très simple, pour au moins trois raisons :

1. Le recalage du système avec l'image initiale est très complexe et très coûteux si aucune connaissance a priori n'est disponible sur la position de la caméra par rapport à la scène.
2. La qualité du recalage temporel dépend de nombreux facteurs tels que le mouvement de la caméra, le niveau de bruit dans l'image et évidemment la complexité de la scène. Il est donc indispensable de disposer d'algorithmes robustes pour le recalage temporel afin de minimiser l'influence des erreurs de mise en correspondance entre deux images.
3. Comme nous l'avons déjà souligné, une composition réaliste nécessite une gestion correcte des occultations entre objets virtuels et objets réels. Jusqu'à présent ce problème a été peu abordé et reste un défi pour les systèmes de réalité augmentée.

Notre travail vise à rendre le processus de composition d'images plus robuste et moins interactif que les méthodes actuellement

utilisées. Nous souhaitons en particulier pouvoir traiter des environnements complexes comme les scènes d'extérieur. Nous avons utilisé pour cela une approche basée modèle qui utilise des connaissances 3D sur la scène, qui sont le plus souvent disponibles dans les applications de réalité augmentée (dans des études d'impact par exemple, les structures principales du site sont en général connues). Nous présentons dans ce papier les solutions que nous avons apportées aux problèmes 2 et 3 mentionnés ci dessus. Nous présentons en particulier un système de recalage temporel robuste basé sur la mise en correspondance de primitives variées (points, droite et courbes quelconques); ceci nous permet de considérer des environnements très complexes, en particulier d'extérieur. Nous présentons de plus une méthode destinée à gérer les occultations.

1.3. applications

Les méthodes que nous avons développées seront illustrées au fur et à mesure par des applications grandeur réelle que nous avons réalisées. Nous donnons ici brièvement leurs caractéristiques. Le lecteur intéressé par des démonstrations vidéo plus complètes, pourra consulter l'URL <http://www.loria.fr/~gsimon/videos.html>.

Un des thèmes applicatifs de l'équipe étant le génie architectural, nous nous intéressons à l'incrustation d'objets dans des séquences d'images tournées en milieu urbain. Trois applications seront en particulier étudiées :

- Le projet d'illumination des ponts de Paris : dans cette application il s'agit de tester des projets d'illuminations concernant le Pont neuf. Une vidéo du pont a donc été tournée à la tombée de la nuit. Notre travail a ensuite consisté à déterminer le point de vue de la caméra pour chaque image de la séquence puis à remplacer le pont par son image de synthèse illuminée. Outre la difficulté d'extraire et de suivre des indices pertinents dans des images peu contrastées, une autre difficulté de cette application provient de la très mauvaise précision du modèle du pont. En effet, ce modèle a été construit à partir de plans d'architecte et ne correspond pas toujours à la réalité terrain (erreur de 50cm par endroit).

- Le château jouet : cette application nous a permis de tester notre système dans le cas d'un mouvement se situant majoritairement dans la direction de l'axe optique. Ce type de mouvement est en général difficile à traiter car il induit de fortes variations de taille des primitives d'une image à l'autre. Il est nécessaire d'avoir un système robuste permettant de s'affranchir d'éventuelles erreurs de suivi inévitables pour ce type de mouvement.

- La place Stanislas : cette application nous permet de montrer les capacités du système actuel : nous présentons l'incrustation dans la scène d'une voiture de synthèse en présence d'occultations entre la voiture synthétisée et certains éléments non modélisés de la place (la statue).

2. travaux connexes

Avant de décrire l'architecture de notre système, nous discutons des méthodes capables de résoudre les problèmes posés et nous justifions nos choix.

Calibrage des caméra versus calcul du point de vue

Si un nombre suffisant de correspondances 2D/3D sont disponibles, alors le recalage se ramène à un calibrage classique qui permet de calculer les paramètres internes ainsi que le point de vue. Dans notre cas, le nombre de primitives en correspondance peut être faible et de plus les connaissances 3D ne sont pas forcément précises. Ceci justifie que nous ayons découplé les processus de calcul des paramètres internes de celui du calcul de la position. Les paramètres internes sont alors déterminés préalablement à la prise de la séquence en utilisant une mire de calibration. Ceci suppose que la focale reste fixe au cours de séquence.

Calcul du point de vue

Les nombreux travaux sur ce sujet peuvent être apparentés à deux classes. La plus classique utilise des correspondances 2D/3D pour résoudre le problème. L'autre alternative est purement 2D [15] : si un nombre suffisant de points en correspondance sont observés à partir de deux (ou plusieurs) positions différentes, le point de vue peut être théoriquement calculé ainsi que la position 3D des points observés (à un facteur d'échelle près). Cette méthode est en particulier utilisée dans le système « 3D Equalizer » pour un ensemble de points 2D facilement identifiables. Malheureusement, ce type d'approche se révèle sensible aux imprécisions dans l'extraction des correspondances 2D, surtout lorsque la translation entre deux positions consécutives de la caméra est faible. Elle est donc peu envisageable pour des applications de réalité augmentée traitant d'environnements naturels. C'est pourquoi nous utilisons une approche à base de modèle.

Mise en correspondance

Le recalage à base de modèle est un processus de mise en correspondance entre modèle et image. Pour le traitement de séquences vidéo, il est raisonnable de supposer que l'utilisateur peut localiser les indices à suivre dans la première image. Les mises en correspondances ultérieures sont alors assurées par le suivi des primitives dans la séquence. Cependant, une seule mise en correspondance erronée peut avoir une forte influence sur le calcul du point de vue. Pour y remédier, on peut soit essayer d'améliorer le processus de mise en correspondance, soit essayer de rendre plus robuste le calcul du point de vue de manière à lui permettre d'ignorer les mises en correspondance aberrantes. L'amélioration de la mise en correspondance est généralement coûteuse et nécessite des connaissances sur l'environnement considéré. C'est pourquoi nous avons préféré utiliser des méthodes statistiques robustes qui utilisent la mise en correspondance induite par le suivi et permettent en quelque sorte d'ignorer les mises en correspondances

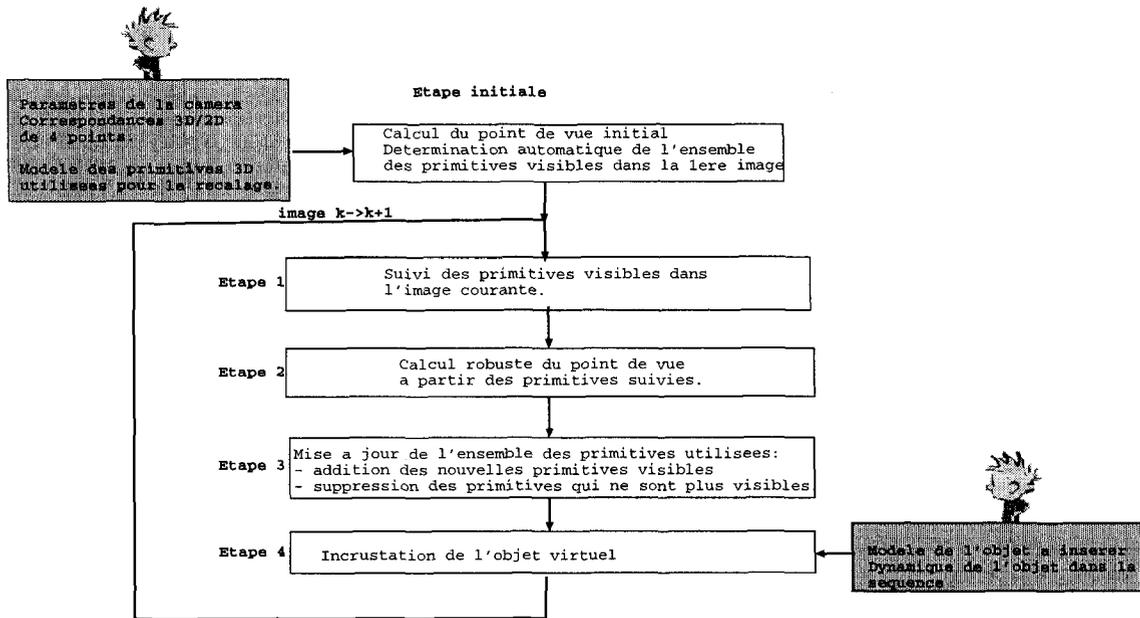


Figure 1. – Schéma du traitement d'une séquence.

aberrantes. Une approche de ce type a déjà été utilisée dans [10]; les auteurs utilisent une méthode robuste pour le calcul du point de vue mais ils ne considèrent que des points en correspondance alors que nous considérons des primitives quelconques (points, droites, courbes).

Gestion des occultations

Le problème de la gestion des occultations dans les systèmes de réalité augmentée a été peu abordé jusqu'à présent. Et la plupart des systèmes se contentent de superposer l'objet virtuel dans les images sans chercher d'éventuelles occultations. Si le modèle complet de la scène est connu, comme dans [breen96a], le problème peut être facilement résolu puisqu'il suffit de comparer les profondeurs de l'objet à insérer avec la profondeur de la scène. Dans les applications grandes réelles que nous considérons, seuls certains éléments 3D de la scène sont disponibles (niveau du sol, structures des principaux bâtiment. . .) Les connaissances que nous possédons sur la scène sont donc très parcellaires et ne peuvent être utilisées pour gérer les occultations entre objets réels et objets virtuels. Il est donc nécessaire de déduire une carte de profondeur de la scène pour résoudre les occultations. Nous présentons dans cet article une méthode de gestion des occultations qui utilise uniquement la notion de contours et qui permet, par régularisation, de déduire le masque des objets venant occulter les objets virtuels.

3. structure du système

Nous décrivons dans cette section notre système de réalité augmentée. Ce système est capable d'effectuer au vol et automatiquement le recalage temporel dans la séquence d'images à partir d'un ensemble de données 3D disponibles sur la scène. L'algorithme de gestion des occultations est, quant à lui, utilisé hors ligne car il nécessite deux images pas trop proches de la séquence pour déterminer, puis suivre, les objets occultants. Nous donnons ici les grandes lignes des algorithmes en insistant sur notre démarche et nous précisons les opérations qui restent à la charge de l'utilisateur. Ces interventions figurent en grisé dans le schéma général (figure 1); elles concernent l'initialisation du système et la donnée du modèle 3D des objets à insérer dans la scène ainsi que leur dynamique.

Les différentes étapes de la composition d'images sont illustrées dans le cadre de l'application des ponts de Paris (voir les figures 2 et 3).

3.1. Initialisation

Les paramètres internes de la caméra (en particulier la taille des pixels) doivent être fournis au système. Le modèle des primitives 3D utilisées pour le recalage est également fourni au système (figure 3a). De plus, on demande à l'utilisateur de désigner dans

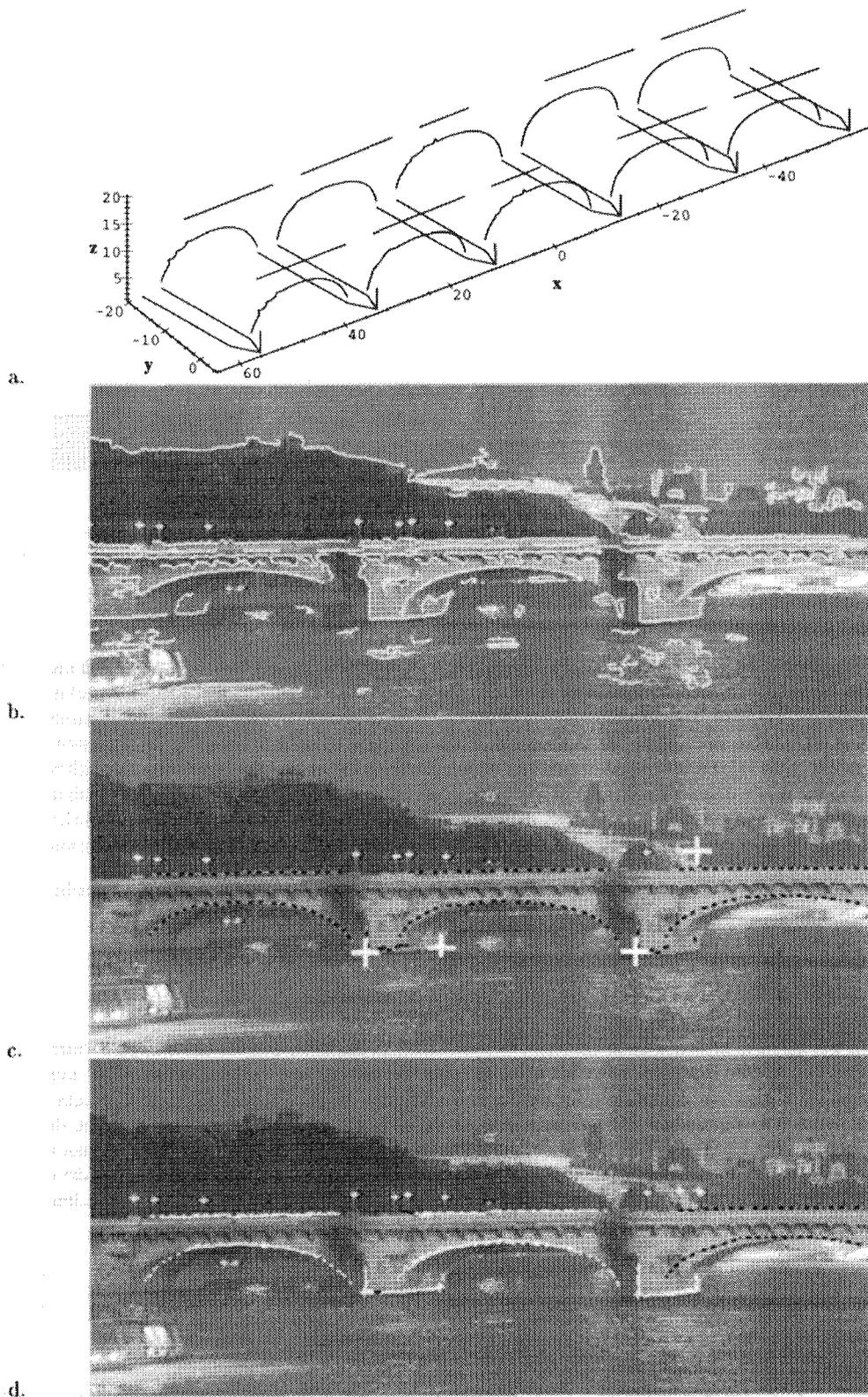


Figure 2. – Initialisation du système. a) le modèle 3D, b) carte de contour de la première image c) les quatre points fournis par l'utilisateur et la projection du modèle en utilisant le point de vue calculé. d) détermination automatique des primitives 2D correspondant au modèle et qui seront suivies dans la séquence.

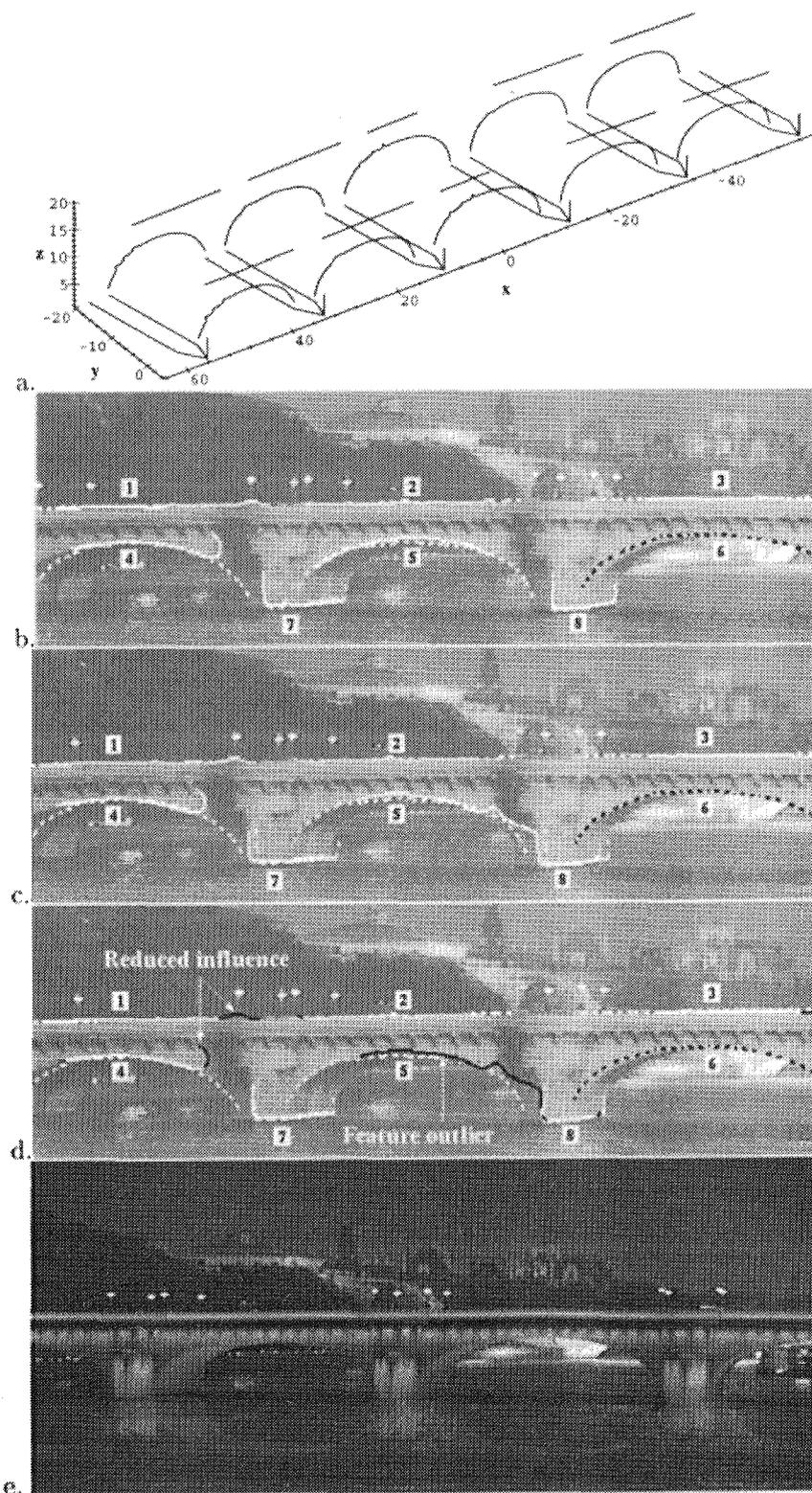


Figure 3. – Un exemple de recalage temporel

(a) Le modèle 3D disponible. (b) L'ensemble des primitives 2D avant le traitement; en pointillés, la projection du modèle sur l'image. (c) Résultat du suivi : la primitive 4 est partiellement bien suivie, la primitive 5 est erronée, les autres sont suivies correctement. (d) Reprojection du modèle en utilisant le calcul robuste du point de vue (en pointillés blancs). Les lignes noires indiquent les parties de contours qui ont été écartées par l'algorithme de calcul du point de vue. (e) Un exemple d'incrustation du pont illuminé dans la scène.

la première image 4 points (figure 2c) et leurs correspondants 3D qui permettront de calculer approximativement le point de vue dans la première image en utilisant la méthode de Dementhon [6]. A partir de cette estimation et en s'appuyant sur la carte de contours 2b, le système est capable, grâce à la procédure du calcul de point de vue, de déterminer automatiquement les contours de l'image qui correspondent aux primitives du modèle (figure 2d).

3.2. recalage temporel

Une fois initialisé, le système parcourt la boucle suivante :

Etape 1 : Suivi des primitives.

L'ensemble des primitives détectées est suivi en utilisant un outil, capable de suivre des courbes, que nous avons développé [3] et dont les grandes lignes sont rappelées dans la section 4. Des erreurs dans le suivi peuvent évidemment apparaître. La figure 3 montre dans le cas de l'application des ponts de Paris que les primitives 1,2,3,7 et 8 ont été bien suivies. La primitive 4 est partiellement erronée car le suivi a dévié sur un autre contour proche du contour à suivre. La primitive 5 est complètement erronée; nous parlerons alors d'*outlier*.

Etape 2 : Calcul du point de vue

Ceci constitue évidemment le cœur de notre algorithme. Par rapport aux travaux existants, notre algorithme est capable de prendre en compte des correspondances de primitives très variées : points, droites et surtout courbes quelconques. Comme des erreurs de suivi peuvent intervenir, nous avons élaboré un algorithme robuste (section 5) qui permet de ne retenir dans le calcul du point de vue que les primitives (ou les parties de primitives) qui correspondent effectivement au modèle 3D. Dans le cas de l'application des ponts de Paris, la figure 3d montre en noir les primitives rejetées par le système (primitive 5) ainsi que les parties de primitives qui ne correspondent pas au modèle (primitive 4). C'est d'ailleurs cette propriété qui permet d'identifier, au niveau de l'initialisation, les morceaux de contours qui correspondent à des primitives du modèle.

Etape 3 : Mise à jour des primitives

Lorsque le mouvement de la caméra est important, en particulier pour un panoramique, les primitives initialement suivies sortent du champ de vision et il faut alors mettre à jour l'ensemble des primitives suivies en intégrant les primitives du modèle devenues visibles. Cette tâche est réalisée automatiquement en utilisant notre algorithme de calcul du point de vue.

3.3. gestion des occultations

Contrairement au calcul du point de vue qui est effectué au fur et à mesure de l'acquisition des images, la gestion des occultations

est effectuée hors ligne. A partir de deux images de la séquence on identifie les points de contours de la scène qui sont situés devant l'objet virtuel à insérer. Puis une méthode de régularisation permet d'identifier la forme (appelée masque d'occultation) la plus régulière possible s'appuyant sur cet ensemble de points étiqueté *devant* (section 6).

4. le suivi des primitives

Le suivi de primitives courbes est une tâche difficile. Il ne peut être basé sur la seule considération des contours extraits dans l'image car la courbe à suivre peut apparaître comme la réunion de plusieurs chaînes de contours et le suivi passe alors par une phase délicate de découpage/fusion/mise en correspondance des chaînes de contours dans deux images consécutives. Par ailleurs, les courbes ne présentant pas forcément de points caractéristiques, une approche de type *token tracker* est en général inenvisageable. L'utilisation d'une mise en correspondance globale par le biais des contours actifs [8] se révèle alors très intéressante car elle permet de résoudre en même temps le problème de la segmentation et celui du suivi : à partir d'un contour initial détourné par l'utilisateur, la courbe se déforme sous l'influence du champ de gradient créé par l'image suivante et va converger vers le contour le plus proche. Cette méthode est efficace tant que les contours en correspondance sont relativement proches mais elle est inopérante dès qu'ils deviennent éloignés.

La méthode de suivi que nous avons conçue inclut donc une étape de prédiction destinée à fournir une initialisation correcte au processus des contours actifs. Ceci revient à estimer le champ des vitesses 2D entre deux images consécutives. En raison de la flexibilité des contours actifs, une estimation grossière est suffisante et nous avons choisi de modéliser le champ des vitesses \mathcal{M} sur le contour à suivre par un déplacement rigide 2D noté \mathcal{D} .

Il est bien connu que le calcul du flot optique complet en chaque point de l'image est très difficile et ne peut donc en général être utilisé directement pour la prédiction. Le flot normal est par contre plus facile à calculer mais il ne constitue qu'une estimation souvent approximative du déplacement entre les deux images, et ce d'autant plus que le déplacement relatif entre les deux images est important. Nous avons donc opté pour une détermination itérative du champ à partir du flot optique normal; l'idée sous-jacente est de calculer une première estimation du champ des vitesses compatible avec le flot normal sur le contour puis de recalculer les deux images en utilisant le déplacement calculé. Les contours en correspondance sont alors plus proches et le flot optique va donc être plus précis. En itérant ce processus, on parvient petit à petit à amener le contour de la première image sur celui qui lui ressemble le plus dans l'image suivante. Plus formellement :

Soit $(M_i)_{0 \leq i < N}$ la discrétisation de la courbe à suivre C et soit $f_0^\perp(M_i)$ le flot optique normal au point (M_i) (la normale à

la courbe au point (M_i) est notée n_i . On note I_1 et I_2 deux images consécutives de la séquence. Le déplacement rigide 2D \mathcal{D}_0 minimisant

$$\sum_{0 \leq i < N} |(\overrightarrow{M_i \mathcal{D}_0(M_i)} \cdot \overrightarrow{n_i}) \overrightarrow{n_i} - f_0^\perp(M_i)|^2 \quad (1)$$

va fournir une première approximation grossière du déplacement \mathcal{M} . Cette estimation est ensuite affinée itérativement en considérant le flot optique normal f_1^\perp sur $\mathcal{D}_0(\mathcal{C})$ entre les images $I_1(\mathcal{D}_0^{-1}(x, y))$ et I_2 . Des déplacements successifs infinitésimaux $\mathcal{D}_0, \dots, \mathcal{D}_j, \dots$ sont ainsi calculés, amenant la courbe \mathcal{C} sur la courbe de I_2 lui ressemblant le plus d'un point de vue photométrique.

D'un point de vue technique, l'angle de rotation des \mathcal{D}_j est généralement petit et le déplacement peut être approximé par

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

La minimisation de (1) peut donc être calculée aux moindres carrés. Ceci rend le processus de suivi de courbes assez rapide. A titre d'exemple, le suivi d'une primitive comprenant 30 points nécessite 90 ms pour l'étape de prédiction et 50 ms pour l'étape de convergence. Ce temps varie évidemment en fonction du contexte puisqu'il dépend de la distance entre deux primitives homologues.

L'intérêt de choisir un modèle explicite et global du mouvement sur le contour (ici un modèle rigide 2D) permet d'éviter que des instabilités inévitables dans le calcul du flot normal viennent perturber l'estimation du champ des vitesses. L'approche itérative nous permet alors de prendre en compte des déplacements de l'ordre d'une vingtaine de pixels entre les deux images. La prédiction obtenue est alors suffisante pour que le contour actif puisse converger vers le contour correspondant.

Il peut cependant arriver que le suivi soit erroné : soit parce que le flot est très bruité sur une grosse partie de la courbe considérée, soit parce que le contour actif a été attiré par un gradient local très fort. Néanmoins, ceci ne concerne en pratique qu'un petit nombre des primitives suivies.

5. méthodes robustes pour le calcul du point de vue

Alors que de nombreux travaux utilisent des correspondances de primitives simples (points, droites, cercle) ou paramétriques [9], notre algorithme permet de calculer le point de vue à partir de correspondances de courbes quelconques. Ceci s'avère très utile dans des environnements extérieurs dans lesquels les primitives courbes sont très présentes. De plus cet algorithme, qui s'organise

en deux niveaux, est très robuste à la présence d'erreurs de mise en correspondance, ce qui permet l'autonomie du processus de recalage temporel. Nous commençons par rappeler quelques éléments de statistique robuste avant de décrire l'algorithme de calcul du point de vue.

5.1. statistique robuste

Le calcul du point de vue revient à calculer les 6 paramètres $p = [p_1..p_6]$ du déplacement $[R, t]$ faisant passer du repère de la caméra au repère absolu. Pour des points en correspondance, ce déplacement peut être calculé en minimisant

$$\min_p \sum r_i^2 = \min_p \sum d(m_i, \text{proj}(RM_i + t))^2 \quad (2)$$

Malheureusement, une procédure de moindres carrés telle que (2) est très sensible au bruit et une seule correspondance erronée peut avoir une forte influence sur le résultat. Pour remédier à ce problème, les statisticiens [11] ont proposé d'utiliser des *estimateurs robustes*. Parmi eux, les *M-estimateurs* et l'estimation de type *moindres carrés médians (LMS)* sont les plus utilisés dans la communauté vision. L'estimation LMS consiste à minimiser la médiane des résidus à la place de la moyenne :

$$\text{si } r_1 \leq r_2 \leq \dots \leq r_n, \text{ médiane } r_i = r_{[n/2]}$$

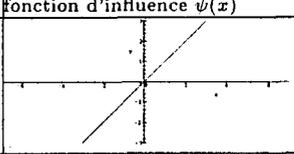
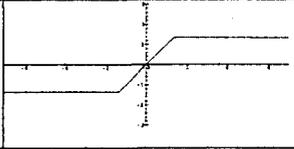
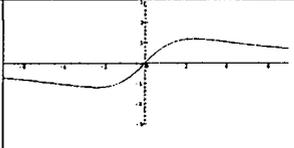
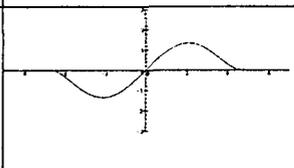
Cette méthode est très robuste, car elle peut tolérer jusqu'à 50% de données aberrantes. Elle est cependant peu précise puisque l'estimation est la valeur correspondant au seul résidu médian. C'est pourquoi on préfère souvent utiliser les M-estimateurs, qui sont plus précis mais ne résistent qu'à un taux de données aberrantes inférieur à environ 30%. La notion de M-estimateurs consiste à minimiser une fonction des résidus

$$\min_p \sum_{i=1}^n \rho(r_i), \quad (3)$$

où ρ est une fonction symétrique, positive, choisie de façon à ce que l'influence de résidus forts (donc de données erronées) soit très réduite, voire même supprimée. L'influence d'une donnée dépend en effet directement de $\psi(x) = \frac{d\rho}{dx}$ (voir [13]), ce qui explique le mauvais comportement de l'estimation aux moindres carrés pour laquelle $\psi(x) = x$. Le tableau 1 fournit quelques exemples de fonctions couramment utilisées et de leur fonction d'influence.

En fonction du choix du M-estimateur, l'influence des données erronées peut être constante (Huber), décroissante (Cauchy) ou même nulle (Tukey). Ce dernier choix n'est pas forcément le meilleur, car il peut conduire à éliminer complètement de l'estimation certaines données et mener à un minimum local de (3) privilégiant les données très précises au détriment de celles, un peu moins précises, mais qui peuvent apporter de l'information de profondeur. Pour une discussion plus précise, voir [12]. Notons enfin que le calcul de p se fait à l'aide d'une minimisation

Tableau 1. – Quelques M-estimateurs classiques.

Type	$\rho(x)$	Fonction d'influence $\psi(x)$
Moindrescarrés	$x^2/2$	
Huber	$\begin{cases} \text{if } x \leq c & x^2/2 \\ \text{if } x > c & c(x - c/2) \end{cases}$	
Cauchy	$\frac{c^2}{2} \log \left(1 + \left(\frac{x}{c} \right)^2 \right)$	
Tukey	$\begin{cases} \text{if } x \leq c & \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{x}{c} \right)^2 \right)^3 \right] \\ \text{if } x > c & c^2/6 \end{cases}$	

itérative prenant comme donnée initiale le point de vue calculé dans l'image précédente.

Ce calcul de point de vue utilisant uniquement des points nous a permis d'obtenir des résultats intéressants dans l'application des ponts de Paris [4]. Mais il se révèle trop restrictif dans la mesure où les points dont on connaît les coordonnées tridimensionnelles et qui peuvent être facilement suivis sont très peu nombreux dans la séquence. La base des piles de pont par exemple peut être suivie facilement mais nous n'avons pas de connaissance 3D puisque cela correspond à l'intersection de la pile avec le niveau de la rivière qui est difficilement calculable. Par contre les primitives courbes telles que les arches peuvent être assez facilement suivies dans la séquence.

Nous avons donc développé un algorithme robuste de point de vue pouvant prendre en compte des correspondances de courbes. Par rapport au cas des points, les difficultés sont de deux ordres : la correspondance entre courbes est globale et non ponctuelle; de plus la notion de données aberrantes n'est pas aussi facile à définir que dans le cas des points. En effet, une primitive courbe peut être mise en correspondance de façon partielle (i.e. une seule partie de la courbe suivie est en correspondance avec le modèle comme la primitive 4 dans figure 3d) ou être complètement erronée (cas de la primitive 5 dans 3d).

5.2. calcul du point de vue à partir de correspondances de courbes

Considérons donc le problème de calculer le point de vue à partir de correspondances de courbes. Soient (figure 4)

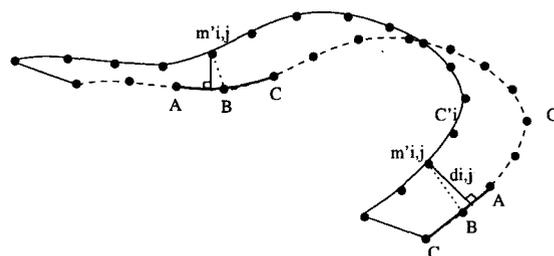


Figure 4. – Correspondance de courbes.

- C_i une courbe 3D décrite par une chaîne de points $\{M_{i,j}\}_{1 \leq j \leq l_i}$
- c_i la projection de C_i dans le plan image, décrite par la chaîne de points 2D $\{m_{i,j}\}_{1 \leq j \leq l_i}$, où $m_{i,j} = \text{proj}(\mathbf{R}M_{i,j} + \mathbf{t})$
- c'_i la courbe détectée (suivie) correspondant à C_i , décrite par la chaîne de points 2D $\{m'_{i,j}\}_{1 \leq j \leq l'_i}$.

$$\min \sum_{i,j} \rho(d_{i,j}) \tag{4}$$

où $d_{i,j}$ est la distance entre $m'_{i,j}$ et la courbe c_i . Une telle solution n'est cependant pas satisfaisante car elle réduit l'ensemble des primitives à un ensemble de points. De plus, elle ne fait aucune distinction entre erreur locale (quand une primitive est partiellement en correspondance) et erreur globale (primitive complètement erronée).

Nous proposons donc d'utiliser une estimation robuste à deux niveaux :

le niveau local évalue pour chaque primitive un résidu robuste.

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

Le niveau global minimise ensuite une fonction robuste de ces résidus :

$$\min_p \sum_1^n \rho(r_i)$$

De cette façon, un primitive très erronée aura un résidu local élevé et ne sera donc pas prise en compte dans le processus global. Une primitive partiellement correcte aura par contre un résidu assez faible et sera prise en compte au niveau global. On arrive ainsi à ne considérer dans le processus d'estimation que les primitives qui sont en correspondance, au moins partiellement, et on rejette les primitives dont un nombre trop important de points est erroné. Le choix des estimateurs pour chacun des niveaux local ou global a bien sûr une influence sur le résultats. Cette influence n'est toutefois vraiment sensible que pour des images de mauvaise qualité.

L'un des avantages de cette méthode est de fournir un moyen explicite d'identifier les primitives aberrantes. En effet une telle primitive se caractérise par un résidu élevé par rapport aux primitives correctes. Une primitive est donc écartée si $r_i > 2.5\hat{\sigma}$

où $\hat{\sigma}$ est l'écart type robuste ($\hat{\sigma} \propto \frac{2}{n} \sqrt{\sum_1^{[n/2]} r_{o(i)}}$ où les $r_{o(i)}$

sont les résidus orientés par ordre croissant). Ainsi, dans le cas de l'application des ponts de Paris (figure 3), la primitive 5 est complètement éliminée. Pour les primitives non écartées au niveau global, nous avons figuré en noir les parties de contours pour les lesquels le résidu est supérieur à c , c étant défini dans le tableau 1.

Cette propriété intéressante est utilisée pour mettre à jour les nouvelles primitives qui apparaissent lorsque la caméra bouge. L'objectif est en fait d'identifier les primitives 3D du modèle, non encore utilisées, qui peuvent être facilement suivies et pourront être utilisées avec profit dans le calcul du point de vue. Soit C une telle primitive. Pour déterminer sa pertinence, nous considérons l'image de contours et nous retenons les contours les plus proches de la projection de C qui sont des candidats pour être le correspondant 2D. Pour chaque candidat c , nous calculons le point de vue correspondant à l'ensemble des primitives utilisées auquel on ajoute c . Si c est rejeté par l'algorithme, cela signifie que c ne peut être considéré comme correspondant au modèle. Si c est accepté, totalement ou en partie, la partie validée de c est considérée comme correspondant au modèle et est ajoutée à l'ensemble des primitives à suivre.

Le correspondant de C , s'il est pertinent, sera celui (ou la partie de celui) qui aura été conservé par l'algorithme de calcul du point de vue.

5.3. évaluation de la robustesse du calcul

Afin d'évaluer la robustesse de notre système, nous montrons ici l'évolution des paramètres de mouvement de la caméra (translation et angles d'Euler) au cours de la séquence. Comme cette séquence a été filmée à partir d'une caméra posée sur un trépied, la translation devrait rester constante (figure 5a,b,c - la translation est exprimée dans le repère du pont, voir la figure 3a pour les axes). On constate que les composantes en x et y sont stables mais que la coordonnée en z est assez instable : ceci est principalement dû aux fortes imprécisions sur le modèle 3D qui affectent essentiellement la hauteur du pont. Les angles d'Euler pour la rotation sont représentés en figure 5d,e,f. L'angle α , qui exprime la rotation autour de l'axe des z , évolue régulièrement quand la caméra tourne. On retrouve bien le fait que pour un panoramique, les deux autres angles sont à peu près constants. Les écarts types des paramètres de la caméra devant être constants sur la séquence panoramique sont montrés au tableau 2.

Tableau 2. - Moyenne et écart type des paramètres de la caméra.

Parametre	Moyenne	Ecart type
t_x (m)	-91.686634	1.678529
t_y (m)	325.359777	2.339981
t_z (m)	13.092605	2.547535
β (rad)	0.009910	0.005982
γ (rad)	-0.000236	0.005982

La figure 6 montre l'évolution sur la séquence du nombre de primitives utilisées (pour les 128 premières images) : le premier niveau des barres (partie claire) indique le nombre de primitives utilisées pour le calcul du point de vue, et le second niveau le nombre de primitives visibles dans l'image (que l'on cherche à mettre à jour). La différence entre les deux niveaux (partie sombre) indique donc le nombre de primitives rejetées par le calcul du point de vue (baisse du niveau 1 avec niveau 2 constant) ou nouvellement présentes dans l'image, et dont la mise à jour a échoué. On voit que le nombre de primitives utilisées varie entre 3 et 7, tandis que le nombre de primitives visibles dans chaque image est compris entre 6 et 8 (5 primitives ont disparu et 6 autres sont apparues dans cet extrait de séquence). On observe deux longs tronçons (images 40 à 61 et 112 à 128) où seules 4 primitives sont utilisées alors que 7 sont visibles dans l'image : ceci est dû au fait que le recalage est très précis sur les 4 primitives considérées et rend difficile l'intégration d'une nouvelle primitive trop bruitée. On constate aussi qu'une primitive nouvelle est très rarement prise en compte immédiatement : ceci est dû à une légère distorsion de l'image dont l'effet est plus important sur les bords (zone où apparaissent les primitives).

Ces résultats prouvent la robustesse de notre algorithme. A partir d'un minimum d'interaction avec l'utilisateur au niveau

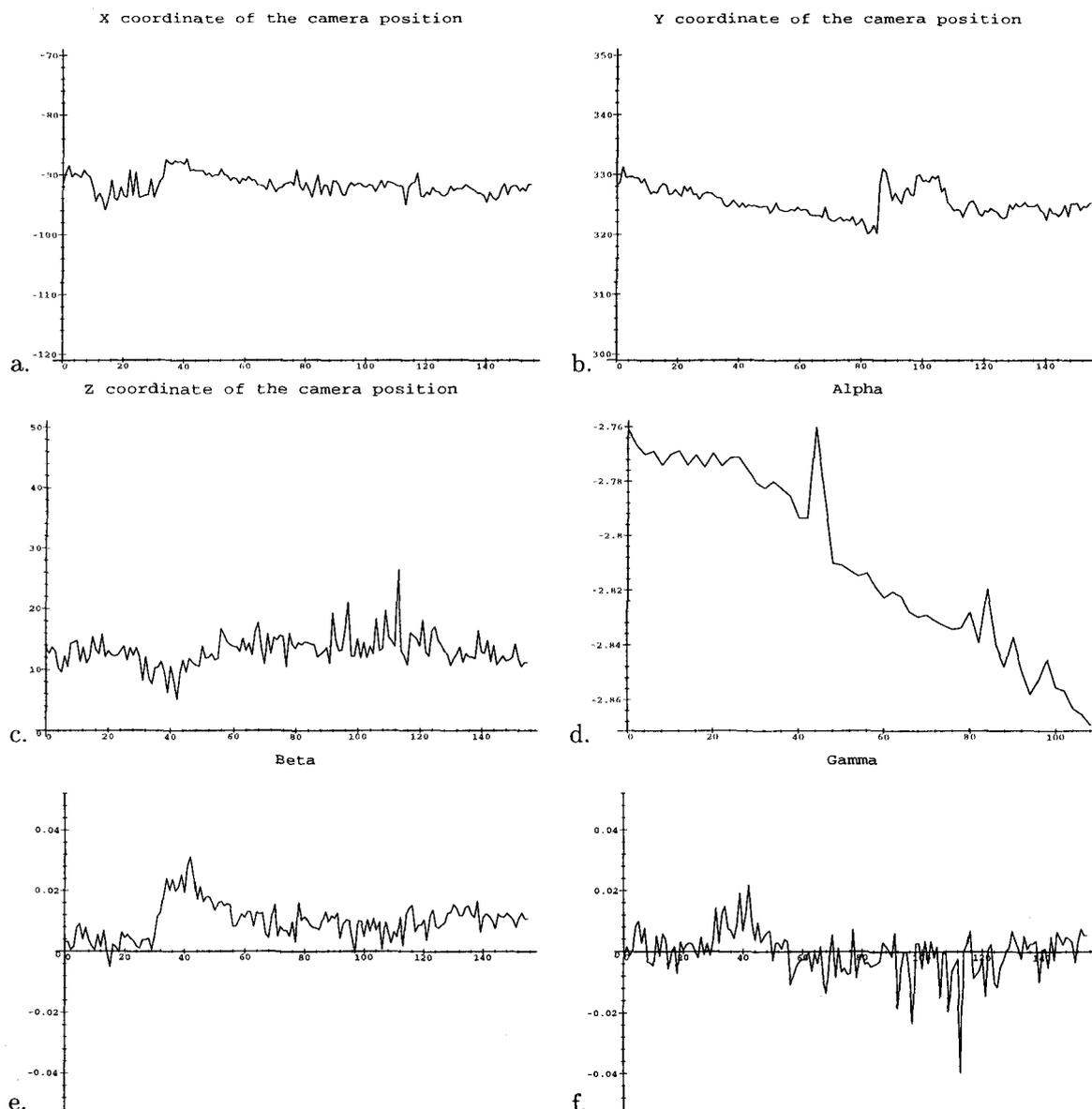


Figure 5. – Evolution du point de vue (translation et angles d'Euler) sur la séquence. Les distances sont données en mètres et les angles en radians.

de la première image de la séquence, le calcul du point de vue s'effectue de manière complètement automatique. Une façon complémentaire de montrer l'efficacité de l'algorithme consiste à laisser l'oeil humain juger de la composition (voir notre site Web); l'oeil est en effet très sensible aux erreurs d'alignement entre scène réelle et objets introduits!

Nous montrons également la reprojction du modèle obtenue dans la *séquence du chateau* pour un mouvement de caméra situé majoritairement le long de l'axe optique. La reprojction est là encore de bonne qualité bien que le suivi des primitives soit beaucoup plus délicat dans ces conditions (figure 7).

6. gestion des occultations

Le calcul du point de vue pour chaque image de la séquence permet de projeter l'objet virtuel au bon endroit dans la séquence. Il reste ensuite à déterminer le masque d'occultation de l'objet, c'est-à-dire sa partie visible, car des objets de la scène peuvent se trouver devant l'objet virtuel. Alors que dans certains environnements manufacturés il est envisageable de disposer des modèles

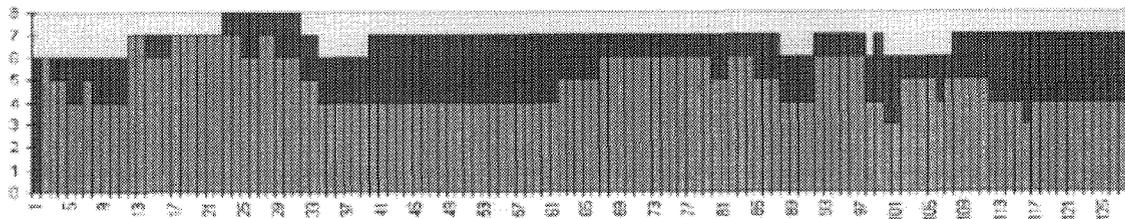


Figure 6. – Evolution sur la séquence du nombre de primitives utilisées pour le calcul du point de vue (premier niveau) et du nombre de primitives visibles dans l'image (deuxième niveau).

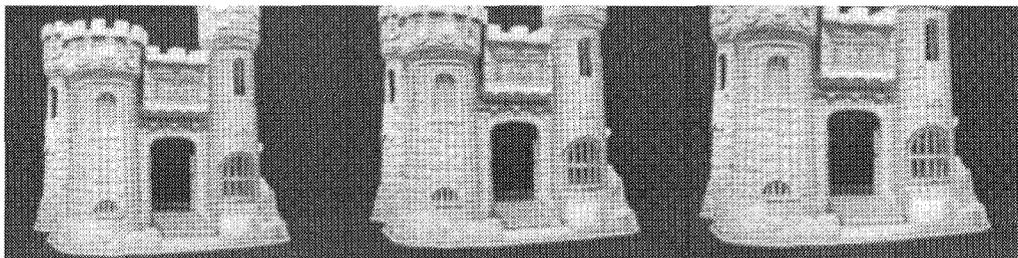


Figure 7. – Le point de vue avec un mouvement le long de l'axe optique; reprojection du modèle 3D pour 3 images extraites de la séquence.

d'éventuels objets occultants [5], cette possibilité est exclue dans les applications en extérieur que nous considérons, car seuls certains éléments 3D de la scène sont connus.

La solution la plus naturelle est de chercher à reconstruire localement la scène au voisinage de l'objet à insérer de manière à comparer les profondeurs de la scène et de l'objet virtuel. Comme nous avons calculé le déplacement de la caméra entre chaque vue, une reconstruction de type stéréovision est donc envisageable. Cependant cette information est lacunaire car la reconstruction 3D n'est réellement fiable qu'au niveau des contours. De plus la précision de la reconstruction obtenue dépend de différents facteurs dont la profondeur du point dans la scène et la distance entre les deux positions de la caméra. De surcroit, la position entre les deux caméras est estimée par notre algorithme et non bien connue comme dans le contexte de la stéréovision, ce qui est une autre source d'imprécision pour la reconstruction. Pour ces raisons, l'utilisation directe de la carte de profondeur produite par la stéréovision mène à un masque comprenant de très nombreuses irrégularités [17]. Pour éviter ces problèmes, nous avons conservé le principe d'une reconstruction stéréoscopique en lui adjoignant un procédé de régularisation de façon à s'affranchir des reconstructions parfois erronées fournies par la reconstruction.

Partant du point de vue, assez raisonnable, que les objets occultants ne comportent généralement pas de trous et sont assez réguliers, nous cherchons à déterminer le masque d'occultation le plus régulier possible, compatible avec l'information de profondeur fournie par la stéréovision. Plus précisément, nous avons donc développé une approche utilisant les contours qui permet de détourner le masque d'occultation rapidement. Le principe est d'étiqueter les contours *devant* ou *derrière* en fonction de leur position par rapport à l'objet à incruster, grâce à un critère que

nous décrivons ci dessous. Cet étiquetage est fait à partir de la mise en correspondance obtenue par suivi des chaînes de contours dans la séquence. Le calcul du masque d'occultation à partir de l'ensemble des points étiquetés *devant*, est ensuite réalisé par régularisation : certains points peuvent être en effet mal étiquetés et surtout certains contours appartenant au masque peuvent manquer s'ils définissent des contours peu marqués. La détermination de la forme la plus régulière s'appuyant sur cet ensemble de points permet alors d'obtenir un résultat très satisfaisant pour le masque.

Les étapes essentielles de l'algorithme sont détaillées dans les deux sections qui suivent et sont illustrées par la séquence de la place Stanislas. Dans cette application, nous souhaitons insérer un véhicule virtuel faisant le tour de place et passant en particulier entre la statue et l'opéra (figure 9). Dans cette application, les données 3D disponibles concernent l'opéra. En particulier, la statue sur la place n'est pas modélisée.

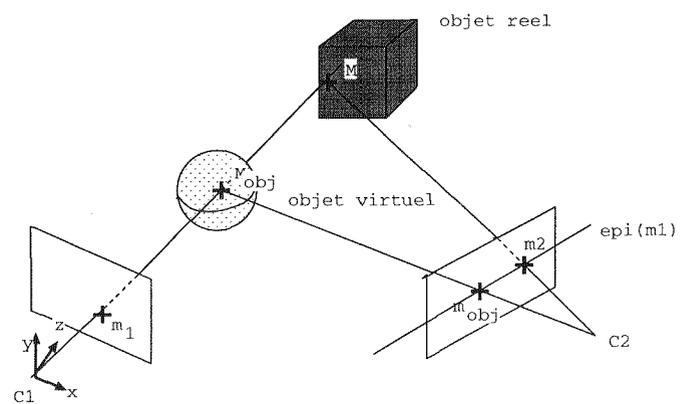


Figure 8. – Etiquetage des points de contours.

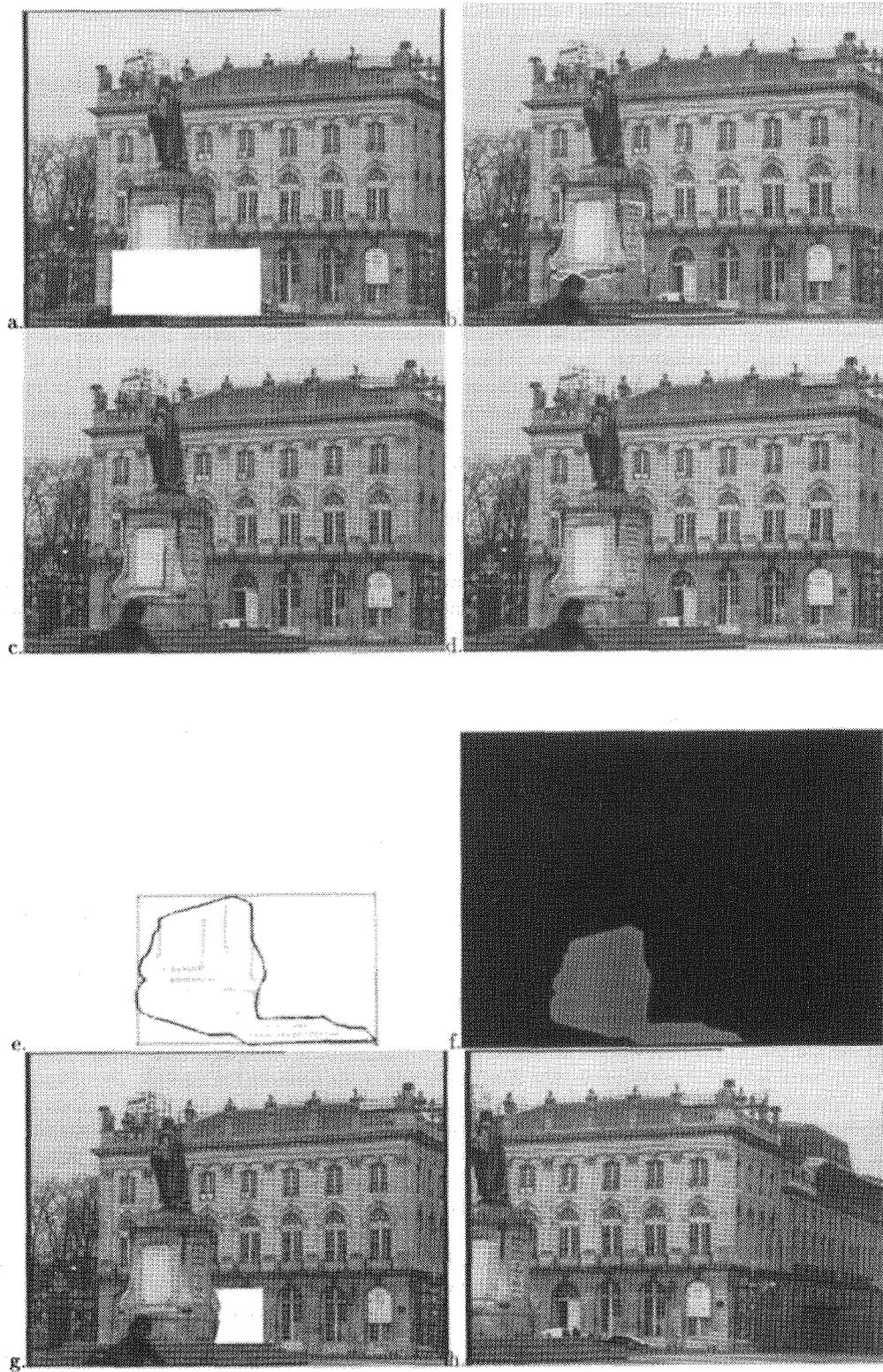


Figure 9. – Un exemple de résolution des occultations

(a) Le rectangle à incorporer dans la scène entre la statue et l'opéra. (b) Les contours suivis. (c) Les points étiquetés devant. (d) Les points étiquetés derrière. (e) Génération du masque à l'aide des contours actifs. (f) Le masque d'occultation. (g) Insertion du rectangle en tenant compte des occultations. (h) Un exemple d'adjonction de voiture dans l'environnement.

6.1. critère d'étiquetage devant/derrière

Considérons deux images I_1 et I_2 de la séquence (figure 8). Soit m_v la projection de l'objet virtuel sur l'image sans tenir compte d'éventuelles occultations. Pour chaque point de contour m_1 appartenant à m_v , nous cherchons à savoir s'il est occulté ou non par la scène réelle. Nous commençons par déterminer le correspondant m_2 de m_1 dans I_2 : chaque chaîne de contour étant suivie entre les deux images, m_2 est obtenu comme intersection de l'épipolaire de m_1 avec la courbe suivie. Le point 3D peut alors être reconstruit dans l'espace et comparé à la profondeur de l'objet virtuel. Il faut noter que si la direction du contour est trop proche de l'épipolaire, le correspondant ne peut être calculé manière fiable. Dans ce cas aucun correspondant n'est attribué au point m_1 .

Cette comparaison n'a cependant de sens que si l'incertitude sur la reconstruction de M est inférieure à la distance entre M et M_{obj} . Comme la seule incertitude que nous sachions finalement quantifier est la précision ϵ sur l'extraction du point dans l'image, une condition nécessaire pour que le résultat de la comparaison des profondeurs soit fiable est donc que la distance entre m_2 et m_{obj} soit supérieure à ϵ . Dans le cas contraire le point m_1 est étiqueté *douteux*. Dans le cas du suivi par contour actif, nous avons expérimentalement estimé $\epsilon = .3$ pixels.

A l'issue de cette étape d'étiquetage, en considérant les morceaux de chaînes constitués uniquement de points étiquetés *devant*, nous disposons donc de contours appartenant au masque d'occultation (figure 9c).

6.2. détermination du masque d'occultation

Il s'agit maintenant de déduire les éventuels objets occultants à partir de l'ensemble des morceaux de contours étiquetés *devant*. Comme plusieurs objets occultants peuvent être présents, nous regroupons d'abord les chaînes en se basant sur un critère de proximité. De manière classique, nous définissons la distance de deux courbes par distance $(C_i, C_j) = \inf_{\{x \in C_i, y \in C_j\}} d(x, y)$. Nous considérons que deux chaînes telles que distance $(C_i, C_j) < s$ appartiennent au même objet (s est en pratique égal à quelques pixels). Nous pouvons ensuite construire un graphe de proximité \mathcal{G} dont les noeuds sont les chaînes; deux noeuds étant connectés si la distance entre les chaînes correspondantes est inférieure à s . La détection des objets occultants revient alors à déterminer les cliques de ce graphe, c'est-à-dire l'ensemble des courbes $\mathcal{H} \subset \mathcal{G}$ tel que : $\forall C \in \mathcal{H}, \exists C' \in \mathcal{H}$ tel que C et C' sont connectés.

Comme nous l'avons souligné à plusieurs reprises, l'inférence du masque d'occultation à partir de \mathcal{H} est assez délicate puisque des erreurs peuvent intervenir dans l'étiquetage. Le recours à une méthode de régularisation permet, en introduisant des contraintes

de lissage sur le masque, d'éliminer l'influence de points isolés mal étiquetés et de combler les vides produits par des contours manquants. Nous avons donc utilisé une méthode de type *contours actifs* [8] pour inférer le masque d'occultation. Les contours actifs sont des courbes v minimisant une énergie de la forme :

$$\int \alpha |v'(s)|^2 + \beta |v''(s)|^2 ds - \int I(v(s)) ds$$

A partir d'une initialisation, le *snake* converge vers la courbe la plus régulière possible compatible avec les maxima d'intensité de I . Comme il est de plus bien connu que le *snake* se rétracte en l'absence d'intensité, le masque d'occultation est ainsi déterminé : partant d'une initialisation contenant \mathcal{H} , nous laissons le *snake* évoluer sous l'influence du champ créé par les contours $I_0 * Gauss(\sigma)$, où

$$\begin{aligned} I_0(x, y) &= 255 \text{ si } (x, y) \text{ appartient à l'une} \\ &\quad \text{des chaînes } \in \mathcal{H} \\ &= 0 \text{ sinon} \end{aligned}$$

Ainsi, le contour va progressivement converger vers \mathcal{H} et on obtiendra le contour le plus régulier s'appuyant sur les points de \mathcal{H} (figure 9e,f). La composition d'images est ensuite facilement réalisée en ne conservant que les points de l'objet virtuel n'appartenant pas au masque d'occultation.

6.3. discussion

La fiabilité de la gestion des occlusions dépend évidemment de la possibilité de détecter des contours dans la zone où on veut insérer l'objet. Une zone faiblement texturée fournira peu d'information de profondeur sur la scène et ne permettra pas une gestion fiable des occultations. Dans le cadre de l'application de la place Stanislas par exemple, le masque de la statue est bien délimité; par contre le masque est nettement moins précis au niveau des marches d'escalier car les contours des contre marches ne peuvent être détectés. La régularisation effectuée permet néanmoins d'obtenir un masque qui respecte bien l'essentiel des contours délimitant le masque. Le masque obtenu n'est pas toujours très précis, mais l'observation de la vidéo obtenue avec ces masques montre que l'impression visuelle est assez satisfaisante (figure 10).

Un second facteur influant sur la qualité du masque concerne le choix des deux images utilisées pour l'étiquetage des contours. En effet, plus le déplacement relatif des contours entre les deux images est important, plus l'étiquetage est simple. Nous n'utilisons donc pas deux images consécutives pour la gestion du masque mais deux images plus distantes.

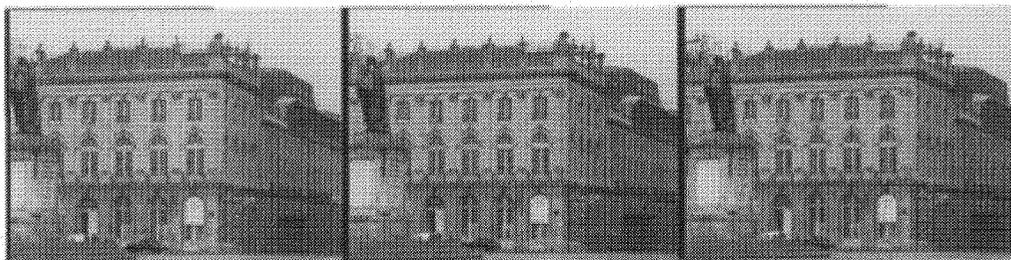


Figure 10. – Quelques images extraites d'une séquence d'incrustation.

7. conclusion

Nous avons proposé dans cet article des méthodes qui facilitent et automatisent la composition d'images dans le cadre d'applications de réalité augmentée lorsqu'un certain nombre de connaissances 3D sont disponibles sur la scène. Alors que de nombreux systèmes se restreignent à des environnements contenant des points facilement identifiables permettant le recalage, nous avons proposé des méthodes permettant de considérer des environnements complexes se recalant sur des structures naturelles de la scène. La méthode de recalage temporel que nous avons proposée se révèle en particulier très performante; en effet elle nécessite une interaction minimale avec l'utilisateur, les primitives qui apparaissent ou disparaissent sont gérées automatiquement. Enfin, le calcul du point de vue est robuste et permet un fonctionnement automatique.

Nous poursuivons actuellement nos recherches pour améliorer encore l'automatisation du processus. Nous souhaitons en effet réduire les contraintes de calibration préalable de la caméra et être capable de prendre en compte des variations des paramètres internes pendant la séquence. Nous poursuivons de plus nos recherches sur l'algorithme de gestion des occultations. La méthode actuelle s'avère en effet efficace lorsque l'information de contours est pertinente dans l'image. Dans le cas contraire, d'autres informations, comme certains points spécifiques liés à la texture, doivent être prises en compte.

BIBLIOGRAPHIE

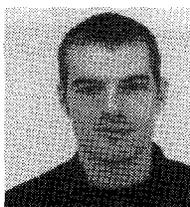
- [1] Azuma (R.). – Tracking Requirements for Augmented Reality. *Communications of the ACM*, juillet 1993, pp. 50–51.
- [2] Bajura (M.), Fuchs (H.) et Ohbuchi (R.). – Merging Virtual Objects with the Real World : Seeing Ultrasound Imagery within the Patient. *Computer Graphics*, vol. 26, n° 2, septembre 1992, pp. 203–210.
- [3] 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. – 1994.
- [4] 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. – août 1996.
- [5] Breen (D.), Whitaker (R.), Rose (E.) et Tuceryan (M.). – Interactive Occlusion and Automatic Object Placement for Augmented Reality. In : *EUROGRAPHICS'96, Poitiers, France*. – 1996.
- [6] 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.
- [7] Ertl (G.), Müller-Seelich (H.) et Tabatabai (B.). – MOVE-X : A System for Combining Video Films and Computer Animation. In : *Eurographics*, pp. 305–313. – 1991.
- [8] Kass (M.), Witkin (A.) et Terzopoulos (D.). – Snakes : Active Contour Models. *International Journal of Computer Vision*, vol. 1, 1988, pp. 321–331.
- [9] Kriegman (D.) et Ponce (J.). – On Recognizing and Positioning Curved 3D Objects from Image Contours. *IEEE Transactions on PAMI*, vol. 12, n° 12, décembre 1990, pp. 1127–1137.
- [10] Ravela (S.), Draper (B.), Lim (J.) et Weiss (R.). – Tracking Object Motion Across Aspect Changes for Augmented Reality. In : *ARPA Image Understanding Workshop, Palm Spring (USA)*. – août 1996.
- [11] Rousseeuw (P.) et Leroy (A.). – *Robust Regression and Outlier Detection*. – Wiley, 1987, *Wiley Series in Probability and Mathematical Statistics*.
- [12] Simon (G.) et Berger (M.-O.). – A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. – Rapport de recherche n° 3235, INRIA, août 1997.
- [13] 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. – janvier 1998.
- [14] State (A.), Livingstone (M.), Garrett (W.), Hirota (G.), Whitton (M.) et Pisan (E.). – Technologies for Augmented Reality Systems : Realizing Ultrasound Guided Needle Biopsies. In : *Computer Graphics (Proceedings Siggraph New Orleans)*, pp. 439–446. – 1996.
- [15] Tomasi (C.) et Kanade (T.). – Shape and Motion from Image Streams under Orthography : A Factorization Method. *International Journal of Computer Vision*, vol. 9, n° 2, 1992, pp. 137–154.
- [16] Uenohara (M.) et Kanade (T.). – Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medicine*, 1996.
- [17] Wloka (M.) et Anderson (B.). – Resolving Occlusions in Augmented Reality. In : *Symposium on Interactive 3D Graphics Proceedings, (New York)*, pp. 5–12. – août 1995.

Manuscrit reçu le 2 avril 1998.

Incrustation d'objets virtuels dans des séquences d'images

LES AUTEURS

Gilles SIMON



Gilles Simon est doctorant au sein du projet Synthèse et Analyse d'Images (ISA), au Laboratoire lorrain de recherche en informatique et ses applications (Loria), Nancy, France. Il s'intéresse aux techniques robustes pour le recalage d'objets et aux solutions menant à un système de réalité augmentée autonome. Gilles Simon est diplômé de l'Ecole Supérieure d'Informatique et Applications de Lorraine (ESIAL) et a reçu son Diplôme d'études approfondies en informatique en 1995. Il devrait obtenir son diplôme de Doctorat en 1999.

Marie-Odile BERGER



Marie-Odile Berger est chercheur à l'Institut national de recherche en informatique et en automatique (Inria) pour le projet ISA. Ses recherches portent sur la réalité augmentée, la vision par ordinateur et l'imagerie médicale. Dr Berger a obtenu une Agégation de Mathématiques en 1986 et un Doctorat d'informatique de l'Institut National Polytechnique de Lorraine en 1991.