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

Gilles Simon et Marie-Odile Berger
Loria/ Inria Lorraine
BP 239, 54506 Vandoeuvre les Nancy

Résumé

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, interaction 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 du vue et à celui de la résolution des occultations. Nous avons en particulier conçu un système autonome permettant un calcul très 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.

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 [8, 3]. D'autres systèmes ont été développés notamment dans le domaine médical [2, 19, 17] et dans le domaine de l'apprentissage [13].

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 performant 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 de erreurs, même minimes, 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 est par exemple résolu en utilisant des capteurs de positions (capteurs Polhemus) [1] ou en utilisant éventuellement des balises clairement identifiables dans la scène [8]. Ces solutions permettent donc de composer des images au prix d'une forte interaction avec la scène ou avec l'utilisateur. C'est pourquoi de nombreux systèmes font appel à des techniques de vision par ordinateur qui ne sont pas invasives et ne nécessitent aucune instrumentation. 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 de 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 matching 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 plus robuste et moins interactif que les méthodes actuellement utilisées. 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). Bien que le calcul du point de vue soit théoriquement possible uniquement à partir de données image [18], ces méthodes n'ont pas une précision suffisante pour assurer une composition d'image précise. C'est pourquoi nous avons privilégié des méthodes utilisant les connaissances 3D disponibles dans le processus de composition. 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 très robuste basé sur la mise en correspondance de primitives très 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 automatiquement les occultations.

Afin de ne pas alourdir la présentation, nous décrivons ici les grandes lignes de notre méthode. Le lecteur intéressé par davantage de détails pourra consulter avec profit [16, 4].

2 Travaux connexes

Afin 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 interne de celui du calcul de la pose. Les paramètres internes sont alors déterminés préalablement à la prise de la séquence en utilisant une mire de calibration.

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 alternatives est purement 2D : 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). Malheureusement, ce type d'approche se révèle très sensible aux imprécisions dans l'extraction des correspondances 2D et elle est donc peu envisageable pour des applications de réalité augmentée. 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 objets dans la première image. Les mises en correspondances ultérieures sont alors assurées par le suivi des primitives dans la séquences. Cependant, une seule mise en correspondance erronée peut avoir une forte influence sur le calcul du point de vue. Pour y remédier, on peut essayer d'affiner la mise en correspondance. On peut également essayer de vérifier la cohérence géométrique des appariements réalisés [19]. Ces méthodes sont cependant généralement coûteuses et nécessitent 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 aberrantes. Une approche de ce type a déjà été utilise dans [13]; 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 points et des courbes quelconques. De plus leur stratégie de suivi est basée sur des techniques de corrélation qui ne peuvent être étendues à des primitives ou des mouvements complexes.

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 [6], le problème peut être facilement résolu. Dans le cas contraire, il est la plupart du temps 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. Les détails des points clés de nos algorithme sont décrits dans les sections 4 et 5.

3.1 Initialisation

Les paramètres internes de la caméra (en particulier la taille des pixels) doivent être fournis au système. De plus, on demande à l'utilisateur de désigner dans la première image 4 points et leurs correspondants 3D qui permettront de calculer approximativement le point de vue dans la première image en utilisant la méthode de Dementhon [7]. A partir de cette estimation, le système est capable, grâce à la procédure du calcul de point de vue, de déterminer automatiquement les primitives de l'image qui correspondent aux primitives du modèle et qui sont suffisamment pertinentes.

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].

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 très robuste (section 4) 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. C'est d'ailleurs cette propriété qui permet d'identifier, au niveau de l'initialisation, les primitives du modèle qui sont visibles dans l'image.

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 remettre 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*.

4 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 [11], 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 pour lesquels les primitives courbes sont souvent pertinentes. De plus cette 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 commencons par rappeler quelques éléments de statistique robuste avant de décrire l'algorithme de calcul du point de vue.

4.1 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

$$_p \sum _i = \quad _p \sum \quad _i, p \quad R_i \quad t \tag{1}$$

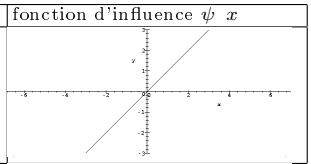
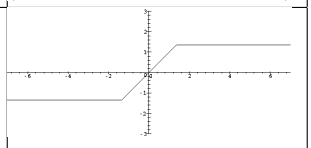
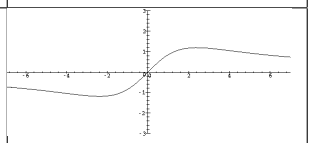
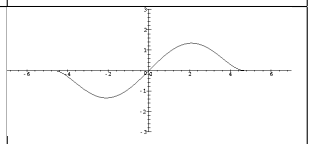
Malheureusement, une procédure de moindres carrés telle que (1) est très sensible au bruit et une seule correspondance erronée peut avoir une forte influence sur le résultat. Pour remédier à ce problème, les statisticiens [14] ont proposé d'utiliser des *estimateurs robustes*. Parmi eux, les *M-estimateurs* 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:

$$_1 \leq \quad \leq \ldots \leq \quad _n, \qquad\qquad _i = \quad _{n/}$$

Cette méthode est très robuste, car elle peut tolérer jusqu'a 50% de données aberrantes. Elle est cependant peu précise puisque l'estimation est la valeur correspondant au seul re'sidu 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

$$_{\mathbf{p}} \sum_{i\ 1}^{n} \quad _i \ , \tag{2}$$

où 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 $= \frac{d\rho}{dx}$ (voir [16]), ce qui explique le mauvais comportement de l'estimation aux moindres carrés pour laquelle $=$. Le tableau ci dessous fournit quelques exemples de fonctions couramment utilisées et de leur fonction d'influence.

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

TAB. 1 – *Quelques M-estimateurs classiques.*

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 (2) 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 [15]. Notons enfin que le calcul de p se fait à l'aide d'une minimisation itérative prenant comme donnée initiale le point de vue calculé dans l'image précédente.

Ce calcul de point de vue utilisant les points nous a permis d'obtenir des résultats intéressants dans l'application des ponts de Paris [5]. Mais il se révèle trop restrictif dans la mesure où les

primitives pertinentes naturellement extraites sont des courbes. 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 (ie une seule partie de la courbe suivie est en correspondance avec le modèle comme la primitive 4 dans Fig. 2.c) ou être complètement erronée (cas de la primitive 5 dans 2.c).

4.2 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

- $_i$ une courbe 3D décrite par une chaîne de points $\{ \ _{i,j}\}_{1\leq j\leq l_i}$
- $_i$ la projection de $_i$ dans le plan image, décrite par la chaîne de points 2D $\{ \ _{i,j}\}_{1\leq j\leq l_i}$, où $_{i,j} = p \quad \mathbf{R} \ _{i,j} \quad \mathbf{t}$
- $'_i$ la courbe détectée (suivie) correspondant à $_i$, décrite par la chaîne de points 2D $\{ \ '_{i,j}\}_{1\leq j\leq l'_i}$.

Une solution consisterait à minimiser en une seule étape la quantité

$$\sum_{i,j} \quad _{i,j} \tag{3}$$

où $_{i,j}$ est la distance entre $'_{i,j}$ et la courbe $_i$. 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.

$$_i = \frac{}{'_i} \sum_{j \ 1}^{l'_i} \quad _{i,j}$$

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

$$_p \sum_{1}^{n} \quad _i$$

De cette façon, un primitive très erronée aura un résidu local élevé est 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 $_i$. où est l'écart type robuste ($\propto \frac{}{n} \sqrt{\sum_1^{n/} \quad _{o\ i}}$ où les $_{o\ i}$ sont les résidus orientés par ordre croissant).

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 sont pertinentes dans les images (i.e qui peuvent être facilement suivies)

et pourront être utilisées avec profit dans le calcul du point de vue. Soit une telle primitive. Pour déterminer sa pertinence, nous considérons l'image de contours et nous retenons les contours les plus proche de la projection de qui sont des candidats pour être le correspondant 2D. Pour chaque candidat , nous calculons le point de vue correspondant à l'ensemble des primitives utilisées auquel on ajoute . Le correspondant de , s'il est pertinent, sera celui (ou la partie de celui) qui aura été conservé par l'algorithme de calcul du point de vue.

La grande robustesse de cet algorithme permet un recalage temporel complètement autonome.

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

Une solution serait de reconstruire la scène à l'aide de deux vues de façon à comparer les profondeurs de l'objet virtuel et de la scène. Malgré des progrès récents, une telle reconstruction n'est pas suffisamment précise, notamment en dehors des points de contour, pour permettre de calculer le masque d'occultation de manière fiable. De plus, il est nécessaire d'estimer l'erreur de reconstruction de manière à pouvoir comparer avec fiabilité les profondeurs respectives de la scène et de l'objet virtuel.

Une autre solution consisterait à chercher les contours d'occultation [9, 12] dans les images car ceux-ci traduisent les discontinuités entre objets et donc les frontières des masques d'occultation. Mais la résolution de ce type de problème est très difficile et elle est de plus souvent basée sur des méthodes de vision active qui ne peuvent être utilisées dans le cadre de la réalité augmentée.

Nous avons donc développé une approche utilisant les contours qui permet de détourer 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. Cette étiquetage est fait à partir de la mise en correspondance obtenue par suivi des chaînes de contours dans la séquence. La 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'il 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 souhaitions insérer un véhicule virtuel faisant le tour de place et passant en particulier entre la statue et l'opéra (Fig.3). La figure montre l'insertion d'un objet rectangulaire et un exemple d'incrustation de véhicule dans la séquence.

5.1 Critère d'étiquetage *devant/derrière*

Considérons deux images $_1$ et de la séquence (Fig. 1). Soit la projection de l'objet virtuel sur l'image sans tenir compte d'éventuelles occultations. Pour chaque point de contour $_1$ appartenant à , nous cherchons à savoir s'il est occulté ou non par la scène réelle. Nous commençons par déterminer le correspondant de $_1$ dans : chaque chaîne de contour étant suivie entre les deux images, est obtenu comme intersection de l'épipolaire de $_1$ avec la courbe suivie. Le point 3D peut alors être reconstruit dans l'espace et comparé à la profondeur de l'objet virtuel.

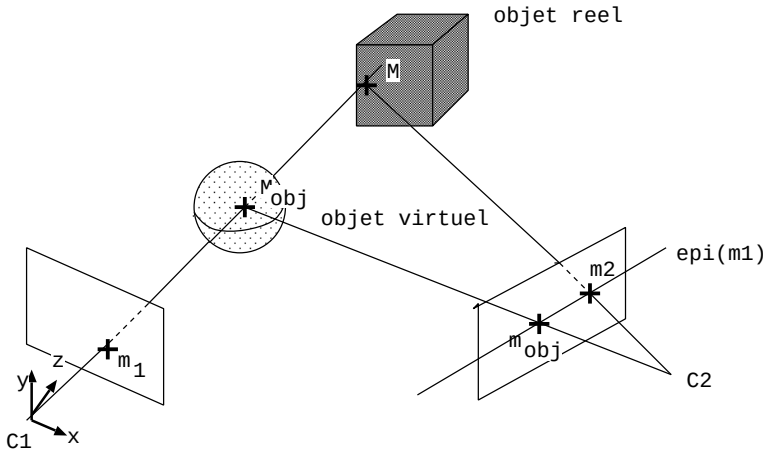


FIG. 1 – *Etiquetage des points de contours.*

Cette comparaison n'a cependant de sens que si l'incertitude sur la reconstruction de est inférieure à la distance entre et $_{o\ j}$. Comme la seule incertitude que nous sachions finalement quantifier est la précision 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 et $_{o\ j}$ soit supérieure à . Dans le cas contraire le point $_1$ est étiqueté *douteux*. Dans le cas du suivi par contour actif, nous avons expérimentalement estimé = . pixels.

Enfin, en considérant les morceaux de chaines constitués uniquement de points étiquetés *devant*, nous disposons donc de contours appartenant au masque d'occultation.

5.2 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 chaines en se basant sur un critère de proximité. De manière classique, nous définissons la distance de deux courbes par $t_{i,\ j} = {}_{x\ i,}$, . Nous considérons que deux chaînes telles que $t_{i,\ j}$ appartiennent au même objet (est en pratique égal à quelques pixels). Nous pouvons ensuite construire un graphe de proximité dont les noeuds sont les chaînes; deux noeuds étant connectés si la distance entre les chaînes correspondantes est inférieure à . La détection des objets occultants revient alors à déterminer les cliques de ce graphe, c'est à dire l'ensemble des courbes tel que: , ′ t t ′ t t .

Comme nous l'avons souligné à plusieurs reprises, l'inférence du masque d'occultation à partir de 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* [10] pour inférer le masque d'occultation. Les contours actifs sont des courbes minimisant une énergie de la forme:

′ ″

A partir d'une initialisation, le snake converge vers la courbe la plus régulière possible compatible avec les maxima d'intensité de . 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 , nous laissons le snake évoluer sous l'influence du champ créé par les contours , où

$$, \quad = \quad , \quad pp \quad t \quad t \quad '$$

$$=$$

Ainsi, le contour va progressivement converger vers et on obtiendra le contour le plus régulier s'appuyant sur les points de (fig 3.e,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 résultats

Les méthodes présentées dans dans cet article ont été testées sur plusieurs applications en vraie grandeur de réalité augmentée. Seuls sont présentés ici quelques extraits de ces applications. Les vidéo complètes concernant le recalage temporel et l'incrustation peuvent être consultées à l'URL *http://www.loria.fr/ gsimon/videos.html*.

La première application que nous avons développée concerne les ponts de Paris (Fig. 2). Dans le but de tester des projets d'illumination, il s'agissait de substituer dans une vidéo tournée à la tombée de la nuit, le pont par son image de synthèse illuminée. Le suivi des primitives dans la séquence s'est avéré très compliqué en raison de la mauvaise qualité des images (Fig. 2.b). Malgré cela, notre algorithme de recalage temporel s'est révélé très efficace et a permis le recalage temporel sur la séquence sans intervention de l'utilisateur (Fig. 2.c).

D'autres applications, non illustrées ici, prouvent la robustesse du recalage temporel pour des mouvements divers: mouvement panoramique comme pour les ponts de Paris, mouvement de zoom selon l'axe optique (voir la séquence du château sur notre site Web).

Enfin une application d'incrustation d'objets sur le site de la place Stanislas nous a permis de montrer l'intérêt de l'algorithme de résolution des occultations (Fig. 3). Nous avons en particulier inséré une voiture virtuelle dans l'environnement passant entre la statue et l'opéra.

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. La méthode de recalage temporel que nous avons proposée se révèle en particulier très performante.

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 la texture, doivent être prises en compte.

Références

[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.). – 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 (USA)*, pp. 91–96. – juin 1997.

[5] 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.

[6] Breen (D.), Whitaker (R.), Rose (E.) et Tuceryan (M.). – Interactive Occlusion and Automatic Object Placement for Augmented Reality. *In : EUROGRAPHICS'96, Poitiers, France.* – 1996.

[7] 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.

[8] 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.

[9] Geiger (D.), Ladendorf (B.) et Yuille (A.). – Occlusions and Binocular Stereo. *International Journal of Computer Vision*, vol. 14, 1995, pp. 211–226.

[10] Kass (M.), Witkin (A.) et Terzopoulos (D.). – Snakes: Active Contour Models. *International Journal of Computer Vision*, vol. 1, 1988, pp. 321–331.

[11] Kriegman (D.) et Ponce (J.). – On Recognizing and Positioning Curved 3D Objects from Image Contours. *IEEE Transactions on PAMI*, vol. 12, n12, décembre 1990, pp. 1127–1137.

[12] Little (J. J.) et Gillett (W. E.). – Direct Evidence for Occlusion in Stereo and Motion. *In : Proceedings of First European Conference on Computer Vision, Antibes (France)*, pp. 336–340. – 1990.

[13] 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).* – août 1996.

[14] Rousseeuw (P.) et Leroy (A.). – *Robust Regression and Outlier Detection.* – Wiley, 1987, *Wiley Series in Probability and Mathematical Statistics.*

[15] Simon (G.) et Berger (M.-O.). – *A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type.* – Rapport de recherche n3235, INRIA, août 1997.

[16] 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.

[17] State (A.), Livingstone (M.), Garett (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.

[18] 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.

[19] Uenohara (M.) et Kanade (T.). – Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medecine*, 1996.

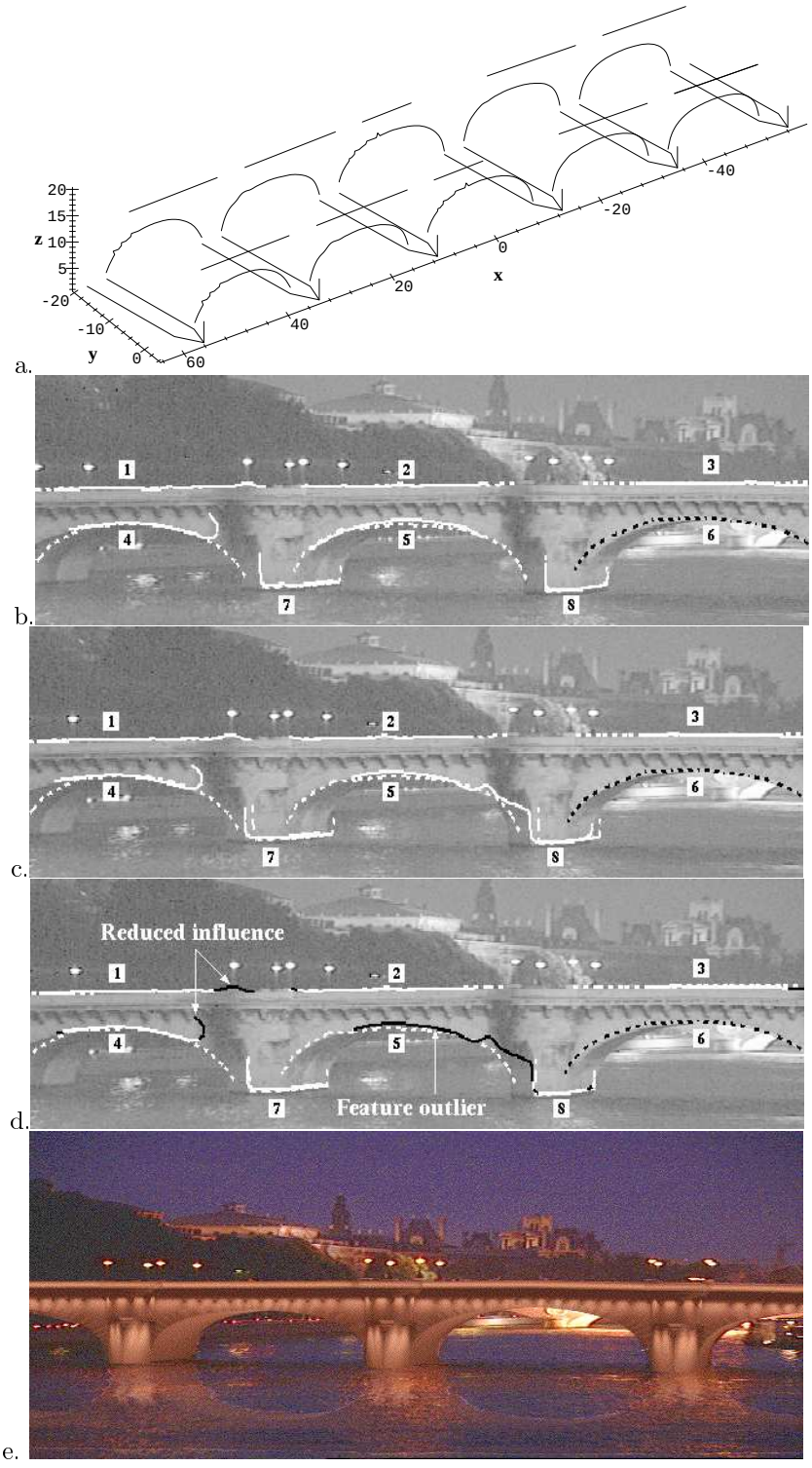


FIG. 2 – *Un exemple de recalage temporel*
(a) Le modèle 3D disponible (b) L'ensemble des primitives 2D avant le traitement; en pointillés, la projection du modèle sur l'image. (c) Résultat du suivi: la primitive 4 est partiellement bien suivie, la primitive 5 est erronée, les autres sont suivies correctement. (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.

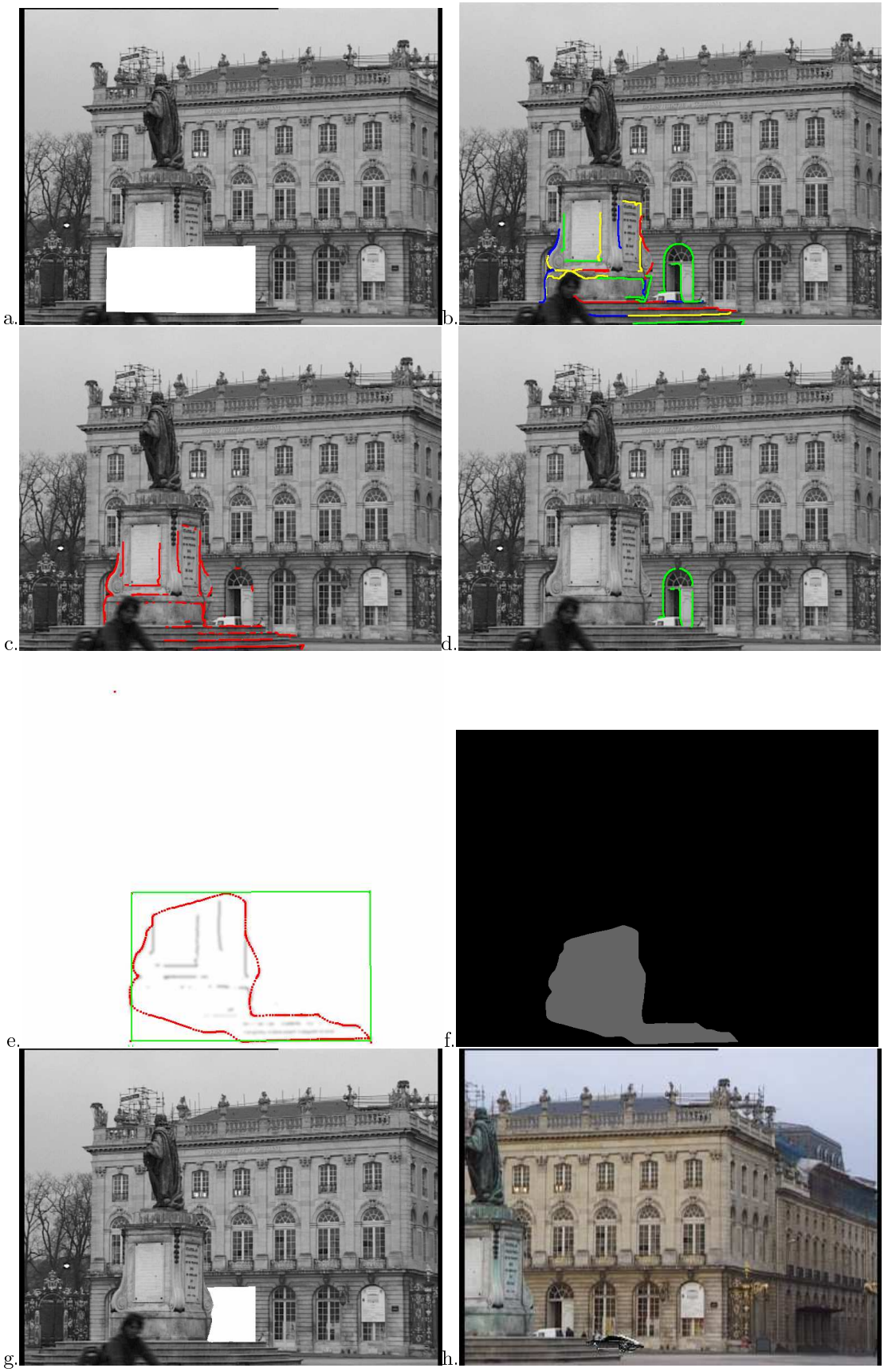


FIG. 3 – *Un exemple de résolution des occultations*
(a) Le rectangle à incorporer dans la scène entre la statue et l'opéra. (b) Les contours suivis. (c) Les points étiquetés *devant*. (d) Les points étiquetés *derrière*. (e) Génération du masque à l'aide des contours actifs. (f) Le masque d'occultation. (g) Insertion du rectangle en tenant compte des occultations. (h) Un exemple d'adjonction de voiture dans l'environnement.