Navigation et controle d'une caméra dans un environnement virtuel: une approche rérencée image.

Nicolas Courty, E. Marchand

Navigation et contrôle d'une caméra dans un environnement virtuel

Une approche référencée image

Nicolas Courty — **Éric Marchand**

IRISA - INRIA Rennes
Campus de Beaulieu
35042 Rennes Cedex
Eric.Marchand@irisa.fr

RÉSUMÉ. *Cet article présente une solution originale au problème du contrôle de caméra en environnement virtuel. Notre objectif est de proposer un cadre général permettant de résoudre les problèmes suivants : positionner une caméra par rapport à son environnement et réagir de manière adaptée à des modifications de celui-ci. La méthode proposée repose sur une technique issue de la robotique et de la vision par ordinateur : l'asservissement visuel. Cette approche permet de générer automatiquement des mouvements 3D de la caméra à partir d'une tâche spécifiée dans l'image. La possibilité de prendre en compte de manière dynamique et en temps-réel des modifications de l'environnement découle de l'utilisation de contraintes intégrées dans les lois de commande considérées. Cette approche est ainsi adaptée à des contextes hautement réactifs (réalité virtuelle, jeux vidéo).*

ABSTRACT. *This paper presents an original solution to the camera control problem in a virtual environment. Our objective is to present a general framework that allows the automatic control of a camera in a dynamic environment. The proposed method is based on the image-based control or visual servoing approach. It consists in positioning a camera according to the information perceived in the image. This is thus a very intuitive approach of animation. To be able to react automatically to modifications of the environment, we also considered the introduction of constraints into the control. This approach is thus adapted to highly reactive contexts (virtual reality, video games). Numerous examples dealing with classic problems in animation are considered within this framework and presented in this paper.*

MOTS-CLÉS : *Animation de caméras, asservissement visuel, évitements d'obstacles et d'occultations, cinématographie automatique*

KEYWORDS: *Automatic camera motion, Automatic cinematography, Visual servoing, Animation*

1. Introduction

Problématique. Le contrôle d'une caméra dans un environnement virtuel soulève de nombreuses questions. De façon classique, la caméra doit non seulement pouvoir se positionner par rapport à son environnement, mais elle doit de plus être à même de réagir à des modifications de celui-ci. Concernant le premier point, même en considérant une connaissance complète de l'environnement, ce qui est généralement le cas en animation, réaliser une tâche de positionnement n'est pas un problème trivial (voir les commentaires de Blinn [BLI 98] à ce sujet). Il y a donc besoin d'un contrôle précis des six degrés de liberté de la caméra. Le second problème, qui peut être vu comme l'introduction de contraintes dans la trajectoire, est encore plus complexe. Afin de pouvoir prendre en compte des environnements dynamiques, ces contraintes doivent être adéquatement modélisées et introduites dans la tâche de positionnement. Nous avons choisi pour notre part d'utiliser les techniques d'*asservissement visuel* afin de réaliser le contrôle de la caméra. Il y a deux raisons principales à ce choix. D'une part, les techniques d'asservissement visuel permettent de spécifier les trajectoires en fonction des informations perçues dans l'image, permettant une description plus «intuitive» de la tâche à réaliser. D'autre part il est possible, grâce au formalisme de la redondance, de prendre en compte automatiquement et en temps réel des modifications de l'environnement en introduisant des contraintes sur les trajectoires à réaliser, contraintes pouvant être définies soit dans l'image soit dans l'espace 3D.

État de l'art. L'asservissement visuel ou commande référencée vision consiste à spécifier une tâche (en général une tâche de positionnement) comme la régulation *dans l'image* d'un ensemble de caractéristiques visuelles [ESP 92, HAS 93, HUT 96, CHA 00]. L'asservissement visuel est cependant une approche locale ; il n'est donc pas possible de planifier la trajectoire à réaliser. Pour prendre en compte des configurations indésirables (occultations, obstacles,...), il faut alors considérer des lois de commande spécifiques. Plusieurs travaux ont été menés sur ces problèmes [NEL 94, MAR 98]. Ces approches combinent la réalisation de la tâche visuelle principale avec la minimisation d'une fonction de coût qui reflète la contrainte imposée sur la trajectoire. Il est alors possible de gérer en temps réel des modifications de l'environnement. Des tâches complexes comme la poursuite d'une cible dans un environnement encombré sont donc réalisables [LAV 97].

Dans le domaine infographique, de telles approches référencées images ont aussi été considérées. La principale différence avec le domaine robotique est, même dans un contexte interactif, la connaissance exhaustive de l'état passé et de l'état actuel des objets du système (profondeur, vitesse,...). Ware et Osborne [WAR 90] proposent différentes métaphores pour décrire une caméra à six degrés de liberté («*eyeball in hand*», «*scene in hand*» et «*flying vehicle*»). La plus intéressante de ces métaphores est «*eyeball in hand*», où la main désigne la position et l'œil l'orientation. Contrôler un tel objet n'est pas un problème trivial. Une solution est d'utiliser des périphériques comme une souris 3D ou un joystick à six degrés de liberté (ddl). Obtenir alors un mouvement fluide et réaliste nécessite un opérateur qualifié. La technique classique de paramétrisation de la caméra via trois vecteurs Lookat/Lookup/Vup permet de se

focaliser simplement sur un point précis de l'environnement. Mais spécifier une tâche plus complexe et de plus haut niveau (par exemple «je veux garder cet arbre au centre de mon image et conserver le lapin bondissant autour de ce même arbre dans le quart gauche de l'image») s'avère difficilement modélisable avec cette technique. Des travaux dans ce sens furent menés par Blinn [BLI 98]. Cependant les résultats s'avèrent trop spécifiques et inadaptés aux problèmes multi-contraints.

Le contrôle explicite de la caméra à partir d'informations basées image a été plus profondément étudié dans le domaine infographique par Gleicher et Witkin [GLE 92]. Dans leur article ils proposent de positionner la caméra par rapport à des objets définis par des points virtuels statiques. Cette technique s'appuie sur une inversion locale de la matrice non-linéaire de transformation perspective. Une optimisation sous contraintes est alors utilisée pour calculer la vitesse de la caméra associée aux déplacements désirés des points virtuels dans l'image. Une autre formulation de ce problème a été établie dans [KYU 95]. On retrouve dans les deux cas une problématique et une formulation extrêmement proches de l'asservissement visuel. Cependant, la relation liant les mouvements de la caméra aux primitives visuelles n'est proposée que pour des points, et aucune contrainte ne régit le mouvement de la caméra.

Différentes solutions pour résoudre l'introduction de contraintes ont été proposées tant en robotique [TAR 95, COW 88] qu'en infographie [DRU 94]. Les solutions résultantes sont similaires : chaque contrainte est définie mathématiquement comme une fonction des paramètres de la caméra (position focale, zoom, etc.) devant être minimisée selon des méthodes déterministes (descente de gradient) ou stochastiques (recuit simulé). Cependant elles présentent plusieurs défauts : de complexités souvent grandes, elles empêchent une implémentation temps-réel (recherche dans des espaces de dimension six) et nécessitent une optimisation à chaque itération (*keyframe*). De plus, les problèmes multi-contraints induisent des fonctions de coût souvent très fortement non-linéaires nécessitant une initialisation adéquate et ne présentant pas nécessairement de solutions. Il reste que Drucker et Zeltzer [DRU 94] ont proposé un grand nombre de fonctions de coût modélisant de nombreuses situations classiques dans le domaine cinématographique. Dans [JAR 98], Jardillier et Languenou propose un système calculant les positions de la caméra en fonction des spécifications établies par l'utilisateur et basé sur la notion de *modélisation déclarative*. Le système délivre alors un ensemble de solutions que l'utilisateur peut parcourir. Les contraintes que spécifie l'utilisateur sont là aussi de haut niveau. La technique utilisée repose sur l'arithmétique des intervalles. Le principal reproche que l'on peut émettre vis-à-vis de ce système est l'aspect statique du contrôle de la caméra : connaissant de manière exhaustive l'ensemble des paramètres de la scène, l'ordinateur évalue *toutes les solutions*, puis l'utilisateur choisit. Il n'y a donc pas de contrôle dynamique de la caméra, ce qui restreint les domaines d'application.

Contribution. Notre système considère surtout le problème de la commande de caméra virtuelle dans des *environnements dynamiques*. Les principales applications sont la génération automatique de trajectoires de caméra pour de la cinématographie virtuelle, et un contrôle réactif pour des applications hautement interactives comme les

jeux vidéos. L'hypothèse principale est la connaissance exhaustive de l'ensemble des objets de l'environnement à l'instant courant (aucune connaissance *a priori* sur le futur). Nous proposons une plate-forme permettant de réaliser dans un premier temps des tâches de positionnement par rapport à un ensemble de primitives (points, lignes, cercles, sphères, cylindres, etc.). Dans la mesure où ce positionnement ne contraint pas les 6 degrés de liberté de la caméra, l'adjonction d'une tâche secondaire modélisant des contraintes sur les trajectoires est alors possible. Cette approche peut alors nous permettre de proposer des solutions à des problèmes d'animation non-triviaux (poursuite d'objets, prédiction et évitement d'occultations et d'obstacles) et à des problèmes en cinématographie virtuelle (travelling et panoramique, problème dit de la «photo» : contraintes sur l'éclairage).

Cet article est organisé de la manière suivante : la seconde partie est une introduction à l'asservissement visuel et reprend des résultats désormais classiques, la troisième partie propose des solutions au problème de la navigation réactive en environnement complexe. Enfin la quatrième partie traitera l'introduction de contraintes cinématographiques dans notre approche.

2. Mouvements de caméra référencés images

Les techniques d'asservissement visuel ou de commande référencée vision [ESP 92, HAS 93, HUT 96] permettent de spécifier la commande de la caméra directement en fonction des informations perçues dans la séquence d'images. Les mouvements 3D de la caméra sont donc calculés automatiquement uniquement à partir d'un ensemble d'informations visuelles 2D.

Dans notre cas, les tâches s'expriment comme la régulation à zéro d'une fonction de tâche combinant une tâche primaire (telle que les primitives visuelles apparaissent dans l'image à une position spécifiée) et une tâche secondaire permettant la modélisation de contraintes sur la trajectoire de la caméra.

2.1. *Tâches élémentaires de positionnement*

Notons $\mathbf{P}$ un vecteur représentant l'ensemble des informations visuelles choisies pour réaliser la tâche. Pour assurer la convergence de $\mathbf{P}$ vers leur valeurs désirées $\mathbf{P_d}$, il est nécessaire de connaître la matrice d'interaction (ou Jacobien image) $\mathbf{L}_\mathbf{P}^T$ qui relie le mouvement de l'objet dans l'image au mouvement de la caméra. Cette relation est définie par l'équation [ESP 92, CHA 00] :

$$\dot{\mathbf{P}} = \mathbf{L}_\mathbf{P}^T \mathbf{T_c} \qquad [1]$$

où $\dot{\mathbf{P}}$ représente le mouvement de l'objet dans l'image dû au mouvement $\mathbf{T_c}$ de la caméra.

La tâche visuelle $\mathbf{e_1}$ est alors définie par :

$$\mathbf{e_1} = \mathbf{C}(\mathbf{P} - \mathbf{P_d}) \quad [2]$$

où $\mathbf{C}$, appelé matrice de combinaison, est choisie de sorte que $\mathbf{C}\mathbf{L}_\mathbf{P}^T$ soit de rang plein le long de la trajectoire $r \in SE^3$. $\mathbf{C} = \mathbf{L}_\mathbf{P}^{T^+}$ si la tâche visuelle contraint les 6 degrés de liberté de la caméra (nous verrons comment définir $\mathbf{C}$ dans le cas où la tâche ne contraint pas les 6 ddl). $\mathbf{L}^+$ dénote la pseudo inverse de la matrice $\mathbf{L}$. Pour assurer une décroissance exponentielle de l'erreur $\mathbf{P} - \mathbf{P_d}$ (à savoir $\dot{\mathbf{e}}_\mathbf{1} = -\lambda \mathbf{e_1}$) la vitesse de la caméra virtuelle est donnée par :

$$\mathbf{T_c} = -\lambda \mathbf{e_1} \quad [3]$$

où λ est un scalaire positif réglant la vitesse de convergence exponentielle de $\mathbf{e_1}$.

Cette approche permet aisément de spécifier des tâches de positionnement par rapport à n'importe quel objet de la scène. L'avantage principal de cette approche en animation est le calcul automatique d'une commande 3D à partir d'une spécification uniquement 2D (en fonction de ce que l'on désire voir dans l'image).

La tâche de l'animateur se limite donc à choisir les informations visuelles $\mathbf{P}$ entrant en jeu dans la tâche de positionnement et leur positions désirées dans l'image. Toute information visuelle peut potentiellement être utilisée dans la même tâche de positionnement : coordonnées de points, orientation d'une droite, surface, sphère, distance, etc.). Une large collection de tâches de positionnement élémentaires peut être proposée et combinée. Par exemple pour construire une tâche de positionnement par rapport à un segment défini par deux points $\mathbf{P_1}$ et $\mathbf{P_2}$, la matrice d'interaction est définie par :

$$\mathbf{L}_\mathbf{P}^T = \begin{bmatrix} \mathbf{L}_{\mathbf{P_1}}^T \\ \mathbf{L}_{\mathbf{P_2}}^T \end{bmatrix} \quad [4]$$

où $\mathbf{L}_{\mathbf{P_i}}^T$ est définie par (si $\mathbf{P_i} = (X, Y)$ et z est la profondeur du point [HUT 96, CHA 00]) :

$$\mathbf{L}_\mathbf{P}^T = \begin{pmatrix} \mathbf{L}_X^T \\ \mathbf{L}_Y^T \end{pmatrix} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \quad [5]$$

Ces matrices d'interaction élémentaires peuvent être calculées pour un grand nombre de primitives (point, droite, cercle, sphère, cylindre, etc. [CHA 00]).

2.2. *Introduction de contraintes dans la commande*

Si la tâche de vision spécifiée ne contraint pas les n degrés de liberté de la caméra, la redondance disponible peut être exploitée pour accomplir une tâche secondaire. Dans ce cas $\mathbf{C}$ est définie par $\mathbf{C} = \mathbf{W}\mathbf{L}_\mathbf{P}^T$ et nous obtenons alors la fonction de tâche suivante [SAM 91] :

$$\mathbf{e} = \mathbf{W}^+\mathbf{e_1} + (\mathbf{I_n} - \mathbf{W}^+\mathbf{W})\mathbf{e_2} \quad [6]$$

où :

— $\mathbf{W}^+$ et $\mathbf{I_n} - \mathbf{W}^+\mathbf{W}$ sont deux opérateurs de projection qui garantissent que le mouvement de la caméra dû à la tâche secondaire est compatible avec la régulation de $\mathbf{P}$ vers $\mathbf{P_d}$. $\mathbf{W}$ est définie comme une matrice de rang plein telle que Ker $\mathbf{W}$ = Ker $\mathbf{L_p^T}$. En raison du choix de $\mathbf{W}$, $\mathbf{I_n} - \mathbf{W}^+\mathbf{W}$ appartient théoriquement au noyau Ker $\mathbf{L_P^T}$, ce qui implique que la réalisation de la tâche secondaire n'aura aucun effet sur la tâche primaire ($\mathbf{L_P^T}(\mathbf{I_n} - \mathbf{W}^+\mathbf{W})\mathbf{e_2} = 0$).

— $\mathbf{e_2}$ est une tâche secondaire qui peut s'exprimer, par exemple, comme le gradient d'une fonction h_s à minimiser ($\mathbf{e_2} = \frac{\partial h_s}{\partial r}$). Cette fonction de coût est donc minimisée sous la contrainte que $\mathbf{e}_1$ soit réalisée.

La loi de commande est maintenant donnée par [ESP 92, CHA 00] :

$$\mathbf{T_c} = -\lambda\mathbf{e} - (\mathbf{I_n} - \mathbf{W}^+\mathbf{W})\frac{\partial \mathbf{e_2}}{\partial t} \qquad [7]$$

2.3. *Suivi d'objets en mouvement*

Une cible mobile implique des erreurs de suivi qui doivent être supprimées pour permettre un positionnement parfait. Dans ce cas, le mouvement de l'objet dans l'image peut se réécrire par :

$$\dot{\mathbf{P}} = \mathbf{L_P^T}\mathbf{T_c} - \mathbf{L_P^T}\mathbf{T_0} \qquad [8]$$

où $\mathbf{L_P^T}\mathbf{T_c}$ et $\mathbf{L_P^T}\mathbf{T_0}$ sont respectivement la contribution de la vitesse de la caméra et celle du mouvement propre de l'objet au mouvement de celui-ci dans l'image. Dans notre cas la vitesse propre de l'objet $\mathbf{T_0}$ est toujours connue. La nouvelle vitesse de la caméra supprimant les erreurs de traînage est donnée par :

$$\mathbf{T_c} = -\lambda\mathbf{e} - (\mathbf{I_n} - \mathbf{W}^+\mathbf{W})\frac{\partial \mathbf{e_2}}{\partial t} - \mathbf{T_0} \qquad [9]$$

2.4. *Commentaires*

Il faut cependant reconnaître que cette méthode n'est pas exempte de défauts. Le principal défaut est la contrepartie directe de sa principale qualité : le contrôle se faisant dans l'image, on perd le contrôle de la trajectoire 3D de la caméra. Cette trajectoire 3D est calculée *automatiquement* pour assurer la tâche visuelle et les tâches secondaires mais l'animateur ne la maîtrise pas. Un second problème réside dans le faite que l'asservissement est une approche locale : il est donc difficile d'assurer, dans la plupart des cas, une convergence de la tâche spécifiée. Des solutions, comme l'asservissement visuel 2 1/2 D [MAL 99, CHA 00] peuvent cependant apportées un début de réponse à ce défaut. On peut finalement observer d'autres défauts importants liés à l'utilisation de la redondance. Citons, entre autres, l'impossibilité évidente d'utiliser la redondance si tous les degrés de liberté de la caméra sont contraints par la tâche visuelle, la possibilité d'avoir des tâches secondaires antagonistes pouvant menée à des oscillations de la caméra, la présence de minima locaux dans la fonction de coût,...

3. Changement réactif de point de vue

Les techniques présentées dans la partie précédente restent assez simples : elles ne considèrent qu'une cible isolée dans un environnement «vide». Dans cette partie nous allons envisager le cas où la caméra et l'objet d'intérêt évoluent dans un environnement plus complexe où d'autres objets (statiques ou non) peuvent gêner la perception de la scène. Dans cette optique nous proposons un système permettant de résoudre deux classes de problèmes récurrents en informatique graphique : l'évitement d'obstacles et d'occultations. Ce ne sont là que deux exemples mais le même type de méthodologie peut s'adapter à d'autres types de problèmes.

3.1. *Evitement d'obstacles*

Considérons que la caméra se déplace dans un environnement complexe en se focalisant sur un objet, éventuellement mobile, de la scène. Le but est d'assurer la focalisation et le suivi de cette cible tout en assurant que la caméra évite les autres objets présents dans la scène. Il existe plusieurs solutions à ce problème :

— une première solution est d'utiliser les degrés de liberté restant à la caméra pour maximiser la distance caméra/obstacle ;

— une seconde solution consiste à planifier une trajectoire évitant ces configurations indésirables, via par exemple une approche de type champs de potentiel [KHA 86]. Drucker et al. [DRU 94] utilisent cette technique pour générer hors-ligne les mouvements de la caméra. Le système présenté considère donc que les comportements des différents acteurs de l'environnement sont connus *a priori* avant de générer l'animation. L'approche planification peut cependant s'appliquer en temps-réel de manière réactive en ne considérant que des potentiels locaux à la caméra.

A titre d'exemple nous développons la première de ces deux méthodes (la seconde étant décrite dans [COU 99]). Comme nous l'avons vu dans la partie précédente afin d'utiliser les degrés de liberté redondants pour éviter une configuration indésirable, en l'occurrence des obstacles, il faut établir une fonction de coût qui est maximum (infinie) lorsque la distance de la caméra à un obstacle est nulle, par exemple :

$$h_s = \alpha \frac{1}{2\|C - O_c\|^2} \qquad [10]$$

où $C(0,0,0)$ est la position de la caméra et $O_c(x_c, y_c, z_c)$ est le point de l'obstacle le plus proche de la caméra, tous deux exprimés dans le repère de la caméra. Si $O_s(x_s, y_s, z_s)$ est le même point «obstacle» exprimé dans le repère de la scène et $M_c(\mathbf{R} \quad \mathbf{T})$ la matrice homogène décrivant la position de la caméra dans ce même repère, l'équation du point dans le repère de la caméra s'exprime par $X_c = \mathbf{R}^T X_s - \mathbf{R}^T\mathbf{T}$.

La tâche secondaire est alors définie par [SAM 91] :

$$\mathbf{e_2} = -(x_c, \ y_c, \ x_c, \ 0, \ 0, \ 0)^T \frac{h_s^2}{\alpha} \quad \text{et} \quad \frac{\partial \mathbf{e_2}}{\partial t} = 0 \qquad [11]$$

Par extension, considérer des obstacles multiples peut s'exprimer par la fonction de coût : $h_s = \sum_i \alpha \frac{1}{2\|C-O_{c_i}\|^2}$.

3.2. *Evitement d'occultations*

Le problème de l'évitement d'occultations est différent : l'objectif est d'éviter qu'un objet s'interpose entre la caméra et l'objet d'intérêt ou cible. Une typologie assez simple des situations pouvant provoquer une occultation peut aisément être établie. Le premier cas considère simplement un objet mobile passant entre la caméra et la cible (cf figure 1.a). Deux autres possibilités assez similaires peuvent se produire : la cible peut disparaître derrière un objet (cf figure 1.b), ou la caméra peut suivre une trajectoire provoquant l'occultation (cf figure 1.c).

Nous allons maintenant proposer une technique référencée vision qui permet de générer les mouvements de la caméra nécessaires à l'évitement des occultations. Dans un deuxième temps nous verrons comment paramétrer la vitesse de la réponse de la caméra en fonction d'une mesure du risque d'occultation.

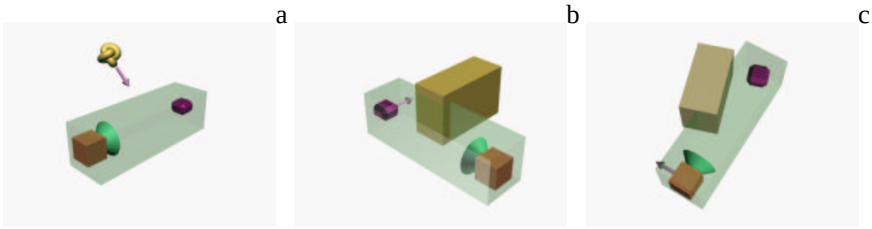

Figure 1. *Différentes causes d'occultations. Occultation (a) par un objet mobile (b) due au mouvement de la cible (c) due au mouvement de la caméra*

3.2.1. *Génération automatique du mouvement adéquat*

Soit $\mathcal{O}$ la projection dans l'image d'un ensemble d'objets pouvant occulter l'objet d'intérêt $T : \mathcal{O} = \{O_1, \ldots O_n\}$. Il faut alors définir une fonction de coût h_s qui atteint son maximum lorsque l'objet est occulté. On cherche alors à maximiser la distance dans l'image entre ces deux objets. Pour cela, on définit h_s par [MAR 98] :

$$h_s = \frac{1}{2}\alpha \sum_{i=1}^{n} e^{-\beta\left(\|T-O_i\|^2\right)} \qquad [12]$$

où α et β sont deux scalaires. $alpha$ règle l'amplitude de la réaction de la caméra (plus α est élevé et plus la vitesse de la caméra sera importante, dans la pratique α dépend directement du risque d'occultation définie en 3.2.2). β permet de définir le moment à partir duquel le mouvement secondaire va se déclencher en fonction de la distance dans l'image entre l'objet d'intérêt et l'objet occultant (plus β est important et plus l'objet peut se rapprocher de l'objet d'intérêt avant que la réaction ne se produise).

La tâche secondaire $\mathbf{e_2}$ est alors donnée par :

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{r}} = \frac{\partial h_s}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial \mathbf{r}}, \qquad \frac{\partial \mathbf{e_2}}{\partial t} = 0$$

où $\frac{\partial \mathbf{P}}{\partial \mathbf{r}}$ n'est autre que la matrice d'interaction $L_{\mathbf{P}}^T$.

Considérons le cas d'un unique objet occultant et d'une cible ponctuelle, la généralisation à plusieurs objets occultant et à d'autre type de cible est immédiate. Soit $\mathbf{P} = (X, Y)$ le centre de la cible. Soit l'objet $\mathcal{O}$ définit par son point $\mathbf{P}_{\mathcal{O}} = (X_{\mathcal{O}}, Y_{\mathcal{O}})$ le plus près dans l'image de P, on a :

$$h_s = \frac{1}{2} \alpha e^{-\beta \|\mathbf{P} - \mathbf{P}_{\mathcal{O}}\|^2} \qquad [13]$$

et

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{r}} = \frac{\partial h_s}{\partial X} \mathbf{L}_X^T + \frac{\partial h_s}{\partial Y} \mathbf{L}_Y^T \qquad [14]$$

avec

$$\frac{\partial h_s}{\partial X} = -\alpha\beta(X - X_{\mathcal{O}}) e^{-\beta \|\mathbf{P} - \mathbf{P}_{\mathcal{O}}\|^2} \quad \text{et} \quad \frac{\partial h_s}{\partial Y} = -\alpha\beta(Y - Y_{\mathcal{O}}) e^{-\beta \|\mathbf{P} - \mathbf{P}_{\mathcal{O}}\|^2}$$

En fait $\mathbf{e_2}$ définie par l'équation (14) est une approximation de $\frac{\partial h_s}{\partial \mathbf{r}}$. En effet $\mathbf{L}_{\mathbf{P}}^T = \begin{bmatrix} \mathbf{L}_X^T & \mathbf{L}_Y^T \end{bmatrix}^T$ est la matrice d'intéraction associée à un point physique. Dans notre cas, comme le point considéré est défini comme le point le de l'objet occultant le plus proche dans l'image de la cible, le point physique correspondant peut être modifié au cours du temps. Considérer $\mathbf{L}_X^T$ et $\mathbf{L}_Y^T$ dans (14) est cependant une approximation localement correcte.

3.2.2. *Calcul du risque d'occultation*

La méthode d'évitement des occultations présentée dans le paragraphe précédent est efficace si l'objet occultant se déplace entre la caméra et l'objet d'intérêt. Cependant Cette méthode reposant uniquement sur des critères images pour générer les mouvements il n'est donc pas nécessairement désirable de bouger systématiquement la caméra. Ainsi aucun test sur la profondeur relative des différent objets (cible et objets «occultants») n'est réalisé pour valider ou invalider l'occultation. De plus, l'intensité de la réaction à l'occultation (représentée par le paramètre α de l'équation (13)) doit dépendre de l'éminence du risque d'occultation.

Il est donc nécessaire de prévoir si et quand une occultation va réellement avoir lieu. Pour ce faire nous considérons le volume englobant $\mathcal{V}$ (voir figure 2ab) qui inclut la caméra et la cible à la date t et $t + ndt$ (la position de la cible à la date $t + ndt$ est calculée en supposant qu'elle est animée d'une accélération constante). Une occlusion se produira si un objet est présent dans ce volume englobant (voir figure 2cd). Le risque d'occultation (ou temps avant occultation) peut être calculé comme étant le plus petit n pour lequel le volume englobant est vide. Si un objet $\mathcal{O}$ de la scène est en

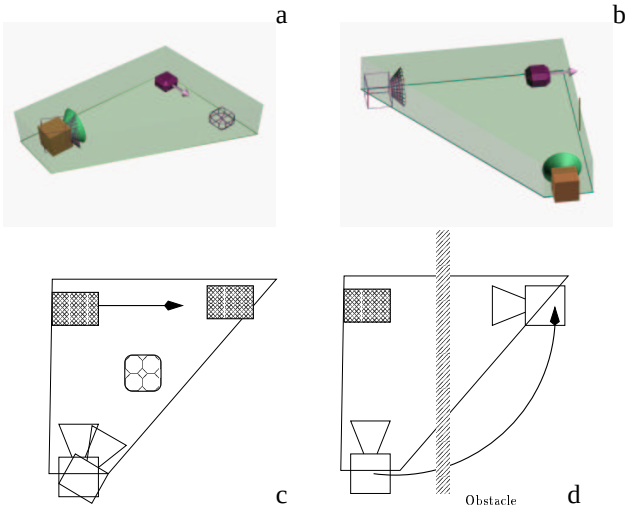

Figure 2. *Deux types de volume englobant (a) mouvement de l'objet d'intérêt (b) mouvement de la caméra (c) détection d'une occultation et (d) d'un obstacle*

mouvement nous considérons alors l'intersection du volume $\mathcal{V}$ définie précédemment avec un volume $\mathcal{V}_{\mathcal{O}}$ contenant l'objet mobile à la date t et $t + ndt$, là encore le temps avant occlusion est le plus petit n pour lequel $\mathcal{V}$ et $\mathcal{V}_{\mathcal{O}}$ ne s'intersecte pas.

Signalons deux points intéressants :

— L'évitement d'obstacles peut être géré en utilisant cette approche. En effet si un obstacle est sur la trajectoire de la caméra, il se retrouvera dans le volume englobant $\mathcal{V}$. Le système interdira donc des mouvements dans cette direction.

— Des cas plus difficiles peuvent cependant se présenter, comme l'exemple du couloir (cf figure 3). Dans ce cas précis, la seule solution est de diminuer la distance entre l'objet d'intérêt et la caméra (possible seulement si la tâche principale ne contraint pas la translation en z).

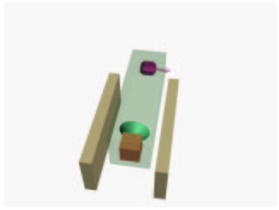

Figure 3. *Causes d'occultations : caméra dans un couloir*

Nous avons donc définie une méthode simple pour détecter et quantifier le risque d'occultation. Les mouvements adéquates de la caméra seront générés en utilisant

l'approche présentée dans le paragraphe précédent et le temps avant occultation servira à calculer le paramètre α qui règle la force de la réaction.

4. Cinématographie virtuelle automatique

Alors que la partie précédente s'attache plus à des applications réactives comme les jeux vidéos, la classe de problèmes abordée dans cette section est relative à la cinématographie virtuelle automatique. La problématique est de savoir comment gérer la caméra durant un plan déterminé en respectant des contraintes purement cinématographiques (voir [ARI 93]) pour plus de détails). Notre but n'est pas de gérer un enchaînement de plans successifs (comme [CHR 96, HE 96]) mais de gérer les mouvements de la caméra pendant la réalisation d'un unique plan. Dans un premier temps, nous verrons comment construire des modules élémentaires de gestion de caméra définis dans [ARI 93] et utilisés dans [HE 96]. D'autres contraintes que celles du positionnement de la caméra, comme par exemple les contraintes d'éclairage, doivent aussi être prises en comptes. Nous verrons alors comment tenter de résoudre ce problème particulier à partir de notre approche.

4.1. *Positionnements élémentaires de la caméra en cinématographie*

Panoramique, poursuite et travelling simple.

Ce sont certainement les mouvements de caméra les plus populaires. Il est possible, de façon évidente, d'utiliser les techniques d'asservissement visuel présentées précédemment pour réaliser ces tâches.

Pour un mouvement de panoramique, il convient de se focaliser successivement sur différents objets de la scène. La caméra effectuera alors un mouvement de «pan». Dans le cas d'une poursuite, un seul objet est à fixer. La principale difficulté revient alors à choisir les primitives sur lesquelles s'asservir (cette primitive n'a pas forcément d'existence «réelle» dans l'environnement). Ce choix est important dans la mesure où il détermine le nombre de degrés de liberté contraints par la tâche de poursuite et donc les mouvements de la caméra. Si l'opérateur spécifie un asservissement par rapport à un point, un segment ou une droite, seul un mouvement de «pan» sera effectué. Par contre, si la caméra doit se déplacer sur une trajectoire parallèle à celle de l'objet d'intérêt, les six degrés de liberté doivent être contraints pour créer une liaison rigide entre la caméra et sa cible (quatre points ou quatre lignes, ou tout autres combinaisons de primitives telles que $\mathbf{L}^T$ soit de rang plein). On a alors un mouvement de travelling relativement simple.

Mouvement complexe de travelling.

Pour des mouvements complexes de travelling la caméra est souvent placée sur un chariot se déplaçant sur des rails montés pour une prise de vue particulière. Suivre une telle trajectoire est relativement aisé : nous souhaitons que la caméra suive une courbe

$\mathcal{V}(t) = (x(t), y(t), z(t))$ définie dans le repère de la caméra. À chaque instant t on souhaite être à la position $\mathcal{V}(t)$. On spécifie alors une tâche secondaire comme une fonction de la distance entre la caméra et le point $\mathcal{V}(t)$. Par exemple :

$$h_s = \frac{1}{2}\|\mathcal{V}(t) - C\|^2 \quad [15]$$

où $C(0,0,0)$ est le centre de la caméra. La tâche secondaire est alors définie comme suit :

$$\mathbf{e_2} = \begin{pmatrix} x(t) & y(t) & z(t) & 0 & 0 & 0 \end{pmatrix}^T h_s \quad [16]$$

4.2. *Contrôler les conditions d'éclairement : «la photo»*

Optimiser l'éclairage est un problème majeur et difficile pour un cinéaste. La première difficulté consiste à déterminer les bons critères d'optimalité. Deux critères sont donc proposés pour parvenir à cet objectif. Le premier repose sur une maximisation de la quantité de lumière réémise par l'objet et le second repose sur les gradient d'intensité de l'image (qui donne une information sur le contraste).

Pour décrire le problème notre objectif initial sera de déterminer la position de la caméra assurant au mieux ces contraintes (la position de la source lumineuse restant fixe). Dans un deuxième temps, afin de maintenir constant l'aspect de l'objet d'intérêt, nous chercherons uniquement à modifier la position de la source lumineuse.

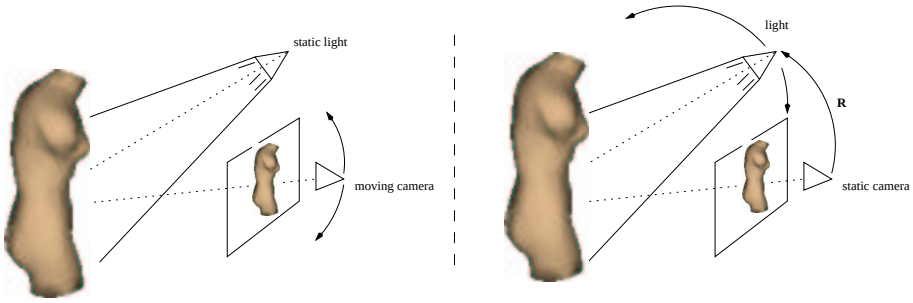

Figure 4. *Contrôle des conditions d'éclairement. (a) source lumineuse statique, caméra mobile (b) caméra statique, source lumineuse mobile*

4.2.1. *Modélisation*

Le premier critère considéré consiste à maximiser la quantité de lumière réémise par l'objet d'intérêt $\mathcal{O}$ afin d'assurer un «bon» éclairage de celui-ci. Ceci revient à maximiser l'intensité moyenne de chaque pixel correspondant à la projection de l'objet dans l'image.

Comme dans les cas précédents, nous souhaitons donc minimiser la fonction de coût suivante :

$$h_s = -\frac{1}{n}\sum_X \sum_Y I(X,Y)$$

où $I(X,Y)$ représente la valeur de l'intensité lumineuse aux points image 2D $(X,Y) \in \mathcal{O}$. La tâche secondaire est alors définie par :

$$\begin{aligned} \frac{\partial h_s}{\partial \mathbf{r}} &= -\frac{1}{n}\sum_X \sum_Y \left(\frac{\partial h_s}{\partial X}\frac{\partial X}{\partial \mathbf{r}} + \frac{\partial h_s}{\partial Y}\frac{\partial Y}{\partial \mathbf{r}} \right) \\ &= -\frac{1}{n}\sum_X \sum_Y \left(\nabla_X \mathbf{L}_X^T + \nabla_Y \mathbf{L}_Y^T \right) \end{aligned} \quad [17]$$

où $\nabla I_X = \frac{\partial I}{\partial X}$ et $\nabla I_Y = \frac{\partial I}{\partial Y}$ représente le gradient spatial d'intensité. Là encore (17) est une approximation localement correcte de $\frac{\partial h_s}{\partial \mathbf{r}}$ (voir la remarque du paragraphe 3.2.1).

Si l'objectif est de maximiser le contraste dans l'image, une condition équivalente est de maximiser la somme des gradients spatiaux d'intensité dans l'image. La fonction de coût correspondante à minimiser est donnée par :

$$h_s = -\frac{1}{n}\sum_X \sum_Y \left[\nabla I_X^2 + \nabla I_Y^2 \right] \quad [18]$$

où $(X,Y) \in \mathcal{O}$. La tâche secondaire est alors donnée par le gradient $\frac{\partial h_s}{\partial \mathbf{r}}$:

$$\frac{\partial h_s}{\partial \mathbf{r}} = -\frac{1}{n}\sum_X \sum_Y \left(\frac{\partial h_s}{\partial X}\mathbf{L}_X^T + \frac{\partial h_s}{\partial Y}\mathbf{L}_Y^T \right) \quad [19]$$

Après quelques réécritures, on obtient finalement :

$$\frac{\partial h_s}{\partial \mathbf{r}} = -\frac{2}{n}\sum_X \sum_Y \quad \left[\left(\frac{\partial^2 I}{\partial X^2}\nabla I_X + \frac{\partial^2 I}{\partial Y \partial X}\nabla I_Y \right) \mathbf{L}_X^T + \right. \quad [20]$$

$$\left. \left(\frac{\partial^2 I}{\partial X \partial Y}\nabla I_X + \frac{\partial^2 I}{\partial Y^2}\nabla I_Y \right) \mathbf{L}_Y^T \right] \quad [21]$$

4.2.2. *Modification de la position de la source lumineuse*

Dans le paragraphe précédent, nous avons considéré une source lumineuse statique avec une caméra mobile. Ce contexte n'est pas le plus intéressant. En effet si la caméra se déplace, l'aspect de l'objet changera au cours du temps. Il serait plus intéressant de contrôler la position et l'orientation de la source lumineuse tandis que la caméra reste statique.

Ici encore nous considérerons le cadre méthodologique de l'asservissement visuel pour diriger la source lumineuse vers l'objet d'intérêt et assurer de bonnes conditions d'éclairage.

Nous couplons d'abord la source lumineuse à une caméra virtuelle (ayant les mêmes position et orientation). La tâche principale est définie comme une simple tâche de focalisation qui contraint les degrés de liberté de rotation du système virtuel caméra/source lumineuse. Nous considérons alors la redondance pour contrôler la translation de ce système et imposer une illumination correcte de l'objet dans l'image saisie par l'autre caméra.

La nouvelle fonction de tâche est donc définie par :

$$\mathbf{e} = \mathbf{W}^{+}\mathbf{W} \quad \underbrace{\mathbf{L}^{+}(\mathbf{P}-\mathbf{P_d})}_{\substack{\text{tâche principale :}\\ \text{focalisation}}} \quad +(\mathbf{I}-\mathbf{W}^{+}\mathbf{W}) \quad \underbrace{\begin{pmatrix} \mathbf{R} & \widetilde{\mathbf{T}}\mathbf{R} \\ 0 & \mathbf{R} \end{pmatrix} \frac{\partial h_s}{\partial r}}_{\substack{\text{tâche secondaire définie}\\ \text{par rapport à}\\ \text{la seconde caméra}}} \qquad [22]$$

où $[\mathbf{RT}]$ est la matrice alignant le repère de la caméra fixe sur le repère associé à la source lumineuse. $\widetilde{\mathbf{T}}$ désigne la matrice antisymétrique associée à $\mathbf{T}$.

5. Résultats

Dans cette section nous présentons quelques résultats obtenus avec les méthodes que nous venons de décrire. Tous les algorithmes présentés ont été implémentés en C++ en utilisant la bibliothèque d'asservissement visuel VISP [MAR 99] à laquelle nous avons rajouté quelques librairies dédiées à la production d'images de synthèse (et reposant sur la bibliothèque Mesa GL émulant Open GL) et à la gestion des volumes englobants. Toutes les séquences ont été produites sur une PC équipé d'un processeur Pentium II à 400 Mhz.

Nous nous attachons ici à commenter le temps de calcul de la commande, le temps de production des images proprement dites n'étant pas significatif (car dépendant trop du matériel utilisé : carte graphique 3D, *pipeline* graphique, etc.). De manière générale la commande doit être calculée le plus rapidement possible afin de pouvoir considérer des applications réactives comme les jeux vidéo. Les temps de calcul de la tâche principale sont totalement négligeables (très inférieurs à la milliseconde). Concernant la tâche secondaire, les temps de calcul associés dépendent de la tâche et de la scène considérée. Si l'on considère le cas de l'évitement d'obstacles ou d'occultations, pour tous les exemples présentés, les temps de calcul restent inférieurs à 20 ms. Le code n'a cependant pas été spécialement optimisé. Tous les objets des scènes considérées sont en effet des polygones et nous calculons, à chaque itération, pour l'évitement d'obstacle le point de ces polygones le plus proche de la caméra et pour l'évitement d'occultations l'intersection entre ces polygones et le volume englobant la caméra et l'objet d'intérêt. Ces calculs sont réalisés pour tous les polygones de la scène (dont le nombre reste cependant modeste). Dans le cas de scènes comprenant un grand nombre de polygones, une hiérarchisation de la scène est sans doute indispensable

afin de considérer les volumes englobants les différents objets et non pas chaque polygone indépendamment. La plupart des méthodes de description de scène utilisées en animation considèrent ce genre de description (Performer par exemple). Les calculs d'intersection entre polygones ou entre volumes englobants peuvent se faire ensuite en considérant les bibliothèques logicielles adéquatement optimisées et généralement disponibles avec les machines dédiées aux applications graphiques (Silicon Graphics ou toute console de jeux). Le portage de nos algorithmes sur de telles machines et l'utilisation de ce type de bibliothèques est en cours de réalisation.

Précisons finalement que les animations de la figure 10 ont cependant été calculées hors-ligne en utilisant le logiciel Maya d'Alias Wavefront. Des résultats plus détaillés sont présentés dans [MAR 00b, MAR 00a].

5.1. *Évitement des occultations et des obstacles :* «suivez le guide»

Dans ce premier exemple, nous avons appliqué la méthodologie proposée à une tâche de navigation dans un environnement complexe. La cible à suivre se déplace, avec un mouvement inconnu, dans un environnement de type musée. Ce «musée» comprend deux pièces sur deux étages reliés par un escalier. L'objectif est de maintenir la cible centrée dans l'image en évitant les occultations par les murs de la pièce tout en considérant *en ligne* les modifications de l'environnement (i.e. la présence d'autres objets mobiles). Dans cet exemple, nous considérons une tâche de positionnement par rapport à une sphère virtuelle. Cette tâche contraint 3 ddl de la caméra virtuelle (i.e. pour réaliser la tâche de fixation et pour maintenir son rayon constant dans l'image). Le lecteur peut se référer à [ESP 92] pour la dérivation complète de la matrice d'interaction associée à la sphère.

La figure 5 montre la trajectoire de la caméra pour les différentes stratégies utilisées lors de la traversée de la première salle du musée. Les obstacles apparaissent en jaune. La trajectoire de la cible est représentée par une ligne pointillée rouge, alors que la trajectoire d'un autre objet mobile est représentée comme une ligne pointillée bleue. La trajectoire rouge représente la stratégie la plus simple : une simple tâche de positionnement par rapport à la cible. Comme rien n'est fait pour considérer l'environnement, des occultations et des collisions avec celui-ci se produisent. La figure 6 montre quelques vues acquises en suivant cette trajectoire. La trajectoire bleue considère en plus une tâche d'évitement des occultations par les objets statiques ; par contre, l'occultation par un objet mobile n'étant pas considérée, elle se produit. Finalement, la trajectoire verte considère l'évitement d'occultations par des objets statiques et mobiles. Les vues des figures 7 et 8 ont été acquises en considérant une tâche d'évitement d'occultations et d'obstacles. La cible reste cette fois toujours dans le champ de vision et n'est jamais occultée. Les collisions avec les murs de la scène sont correctement évitées. Rappelons que l'environnement n'est pas planaire et que ni l'objet d'intérêt ni la caméra ne reste dans le même plan. Les contraintes restent maintenues malgré cette trajectoire 3D. Sur les vues du dessus, les volumes jaunes (associés aux triplet (ca-

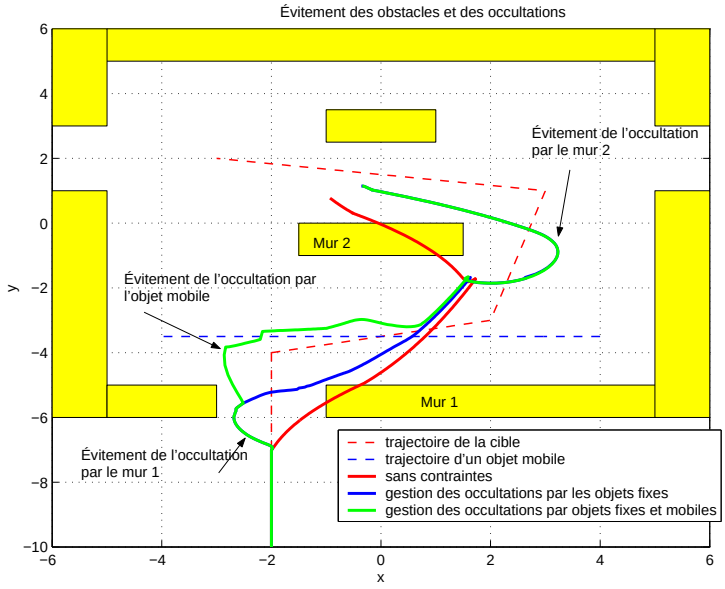

Figure 5. *Environnement de type musée : trajectoire de la caméra pour différentes stratégies*

méra, position courante et prédite de la cible)) correspondent aux volumes englobants servant à la prédiction des occultations et à l'évitement d'obstacles.

5.2. *Progression dans un couloir : fusion de contraintes*

Dans cette seconde expérience nous considérons une tâche similaire mais la cible se déplace dans un couloir étroit et tourne à droite (voir la figure 9). Il n'est *a priori* pas possible de maintenir la cible visible si la distance entre la caméra et la cible demeure constante. Le problème vient du fait que le mouvement calculé pour éviter l'occultation déplace la caméra vers le mur. Un processus supplémentaire d'évitement d'obstacles est alors nécessaire. Nous avons alors trois tâches secondaires : la première associée à la distance caméra-cible, la seconde liée à l'évitement d'obstacles et la dernière liée à l'évitement des occultations. La loi de commande résultante produit automatiquement un mouvement qui éloigne la caméra du mur et réduit la distance caméra-cible. Cette distance, initialement réglée à 3,5 m, diminue pour atteindre moins de 2,5 m au passage du coude puis revient à sa valeur nominale.

Figure 6. *Traversée d'un musée. Le processus d'évitement des occultations et des obstacle n'est pas considéré. La trajectoire résultante comprend de nombreuses pertes de la cible et des collisions avec l'environnement*

5.3. *Suivi de trajectoires*

Dans l'exemple présenté figure 10, la tâche de positionnement s'effectue par rapport à la tour (verticale centrée dans l'image). On peut trouver une tâche similaire dans [GLE 92]. Nous avons choisi d'asservir la caméra par rapport à la droite supportant l'axe de la tour. Dans ce cas la matrice d'interaction est de rang plein et de dimension deux [ESP 92]. Seulement deux ddl sont alors contraints, permettant une plus grande liberté de déplacement que dans [GLE 92] où l'asservissement était réalisé par rapport au deux points extrémités contraignant ainsi quatre ddl. Les figures 10.a et 10.b montrent le début de la tâche de positionnement. Les figures 10(b–f) montrent le suivi d'une courbe tout en maintenant la tâche de focalisation.

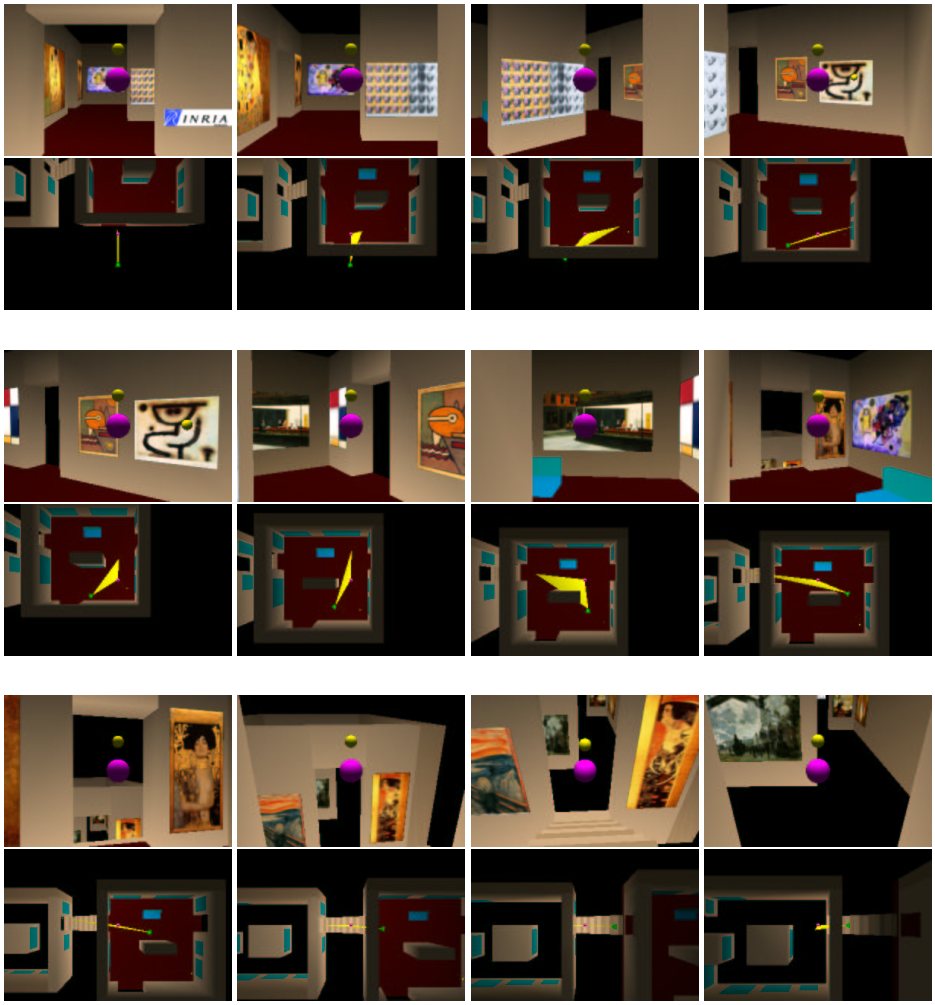

Figure 7. *Traversée d'un musée (1ere partie) : vue de la caméra et vue du dessus*

5.4. *Le problème de la photographie*

Pour valider les résultats concernant cet aspect, nous cherchons dans un premier temps à positionner la caméra par rapport à une sphère éclairée par une source de lumière directionnelle. Les résultats de cette tâche de positionnement sont présentés dans les figures 11.a). L'intensité moyenne évolue très doucement (cf.fig 11.b). On peut constater (cf.fig 11.d) que la caméra se place entre la source lumineuse et l'objet d'intérêt, comme cela est prévisible par la théorie.

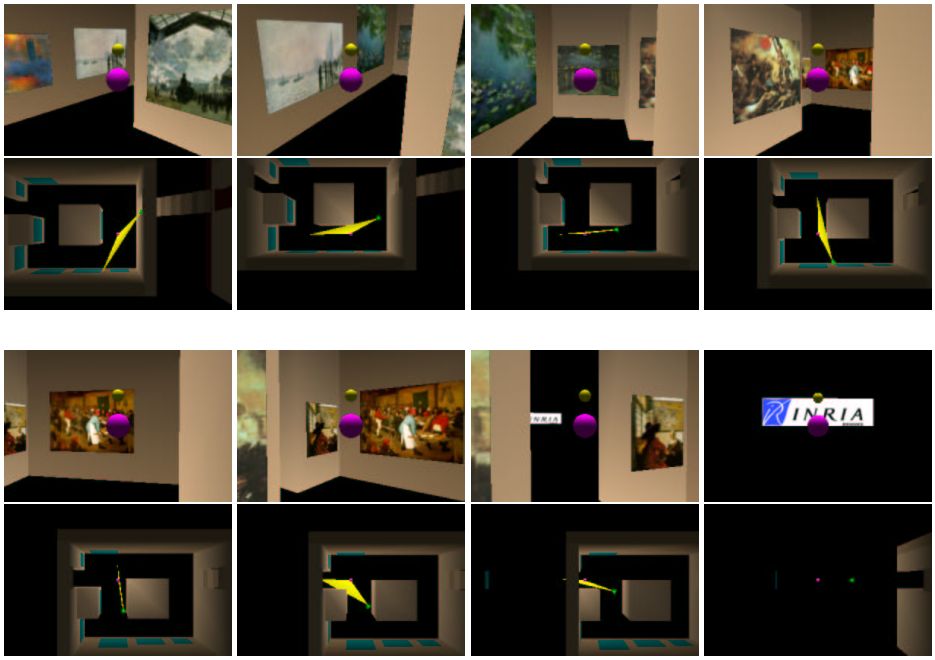

Figure 8. *Traversée d'un musée (2ème partie)*

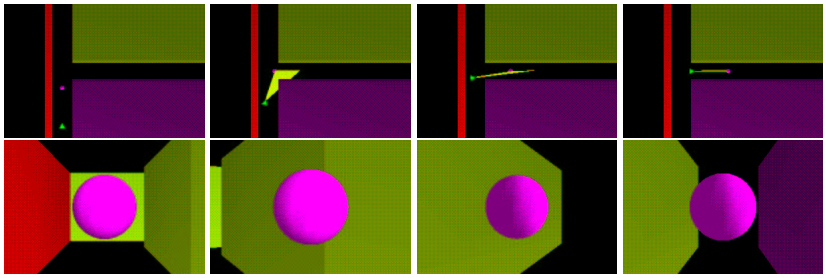

Figure 9. *Progression dans un couloir : vue de dessus et vue de la caméra*

D'autres expériences considérant des objets plus complexes ont été réalisées en considérant, par exemple, un modèle de la Venus de Milo. Dans cette expérience nous considérons dans un premier temps une caméra statique et une source lumineuse mobile. Dans un deuxième temps, quand un minimum de la fonction de coût est atteint, nous imposons un mouvement à la caméra (voir la figure 13). La lumière doit alors se déplacer afin de maintenir un éclairage correct de la statue. Les résultats présentés (voir la figure 12) montrent la validité de notre approche. Les deux fonctions de qualité ont été considérées (la luminance est considéré sur la figure 12.a et le contraste est considéré sur la figure 12.b).

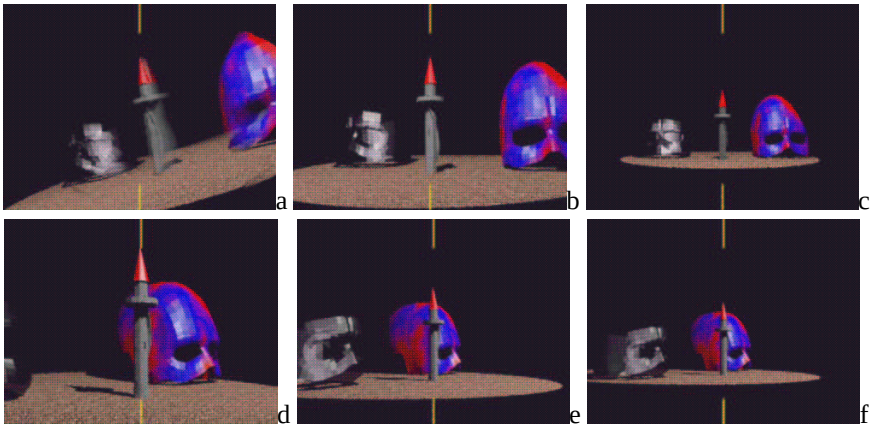

Figure 10. *Positionnement par rapport à une tour et suivi d'une trajectoire prédéfinie*

6. Conclusion et perspectives

Les problèmes associés à la gestion d'une caméra dans un environnement 3D virtuel sont nombreux ; il faut non seulement pouvoir réaliser une tâche visuelle (souvent une tâche de focalisation ou de manière générale une tâche de positionnement) de manière efficace, mais il faut en plus être en mesure de réagir de manière appropriée à des modifications de l'environnement.

Nous avons choisi d'utiliser des techniques issues du monde de la robotique pour résoudre ce problème. L'outil de base que nous avons considéré est l'asservissement visuel qui consiste à positionner une caméra en fonction des informations perçues dans l'image. Ce contrôle dans l'image constitue la première originalité de notre approche. On est en effet ramener à spécifier des tâches à accomplir *dans un espace 2D*, et obtenir des trajectoires de caméra *dans un espace 3D*. C'est donc une approche très intuitive de l'animation que nous proposons puisque celle-ci est réalisée en fonction de ce que l'on souhaite observer dans la séquence d'image produite.

Ce n'est cependant pas le seul avantage de la méthode. En effet, contrairement aux travaux «préliminaires» déjà réalisés sur ce thème [GLE 92], nous ne nous sommes pas limités à des tâches de positionnement dans des environnement statiques. Dans un grand nombre d'applications (prenons le cas des jeux vidéo) il est en effet nécessaire de pouvoir réagir à des modifications de l'environnement, des trajectoires des objets mobiles de la scène, etc. Nous avons donc considéré l'introduction de contraintes dans la commande de la caméra. Grâce au formalisme de redondance, la tâche secondaire (qui reflète les contraintes sur le système) n'a aucun effet sur tâche visuelle (tâche qui doit *impérativement* être assurée) ; il n'y a donc pas de compromis entre les tâches visuelles et secondaires. Pour montrer la validité, nous avons proposé et implémenté un grand nombre de problèmes différents allant de tâches simples comme la poursuite d'un objet à des tâches plus complexes comme l'évitement d'occultations

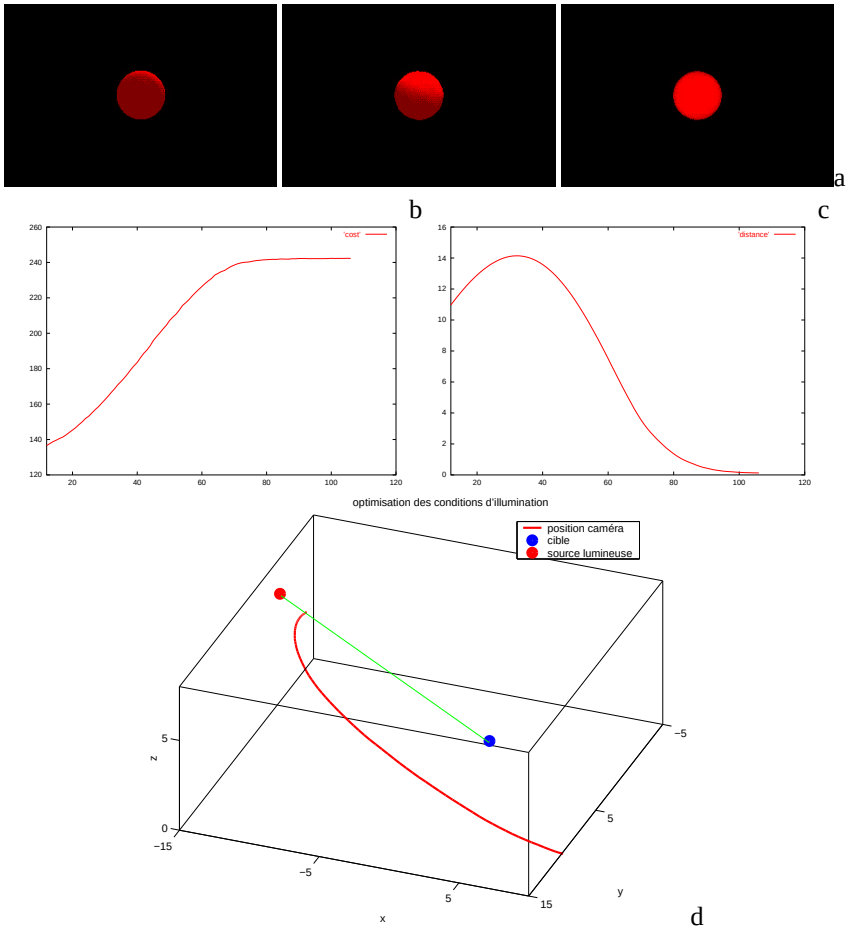

Figure 11. *Positionnement par rapport à une sphère selon des contraintes d'éclairement (a) vue de la caméra (l'intensité augmente) (b) intensité moyenne dans l'image (c) distance à l'axe objet–source lumineuse (e) positions de la caméra au cours du temps*

ou d'obstacles ou encore le positionnement par rapport à la face éclairé d'un objet (afin d'assurer une bonne «photo»). L'approche de contrôle de caméra que nous avons proposée possède donc de réelles qualités, et les résultats très encourageants obtenus peuvent laisser croire que l'utilisation de l'asservissement visuel pour l'animation en environnement virtuel est une technique prometteuse.

On peut cependant voir un intérêt plus grand à utiliser ces techniques dans le cadre d'applications temps-réel, où l'utilisateur se déplace dans un environnement où l'on ne connaît pas *a priori* les mouvements des objets. La grande flexibilité des commandes que l'on peut envoyer à la caméra, ainsi que les adaptations possibles vis-à-vis de son

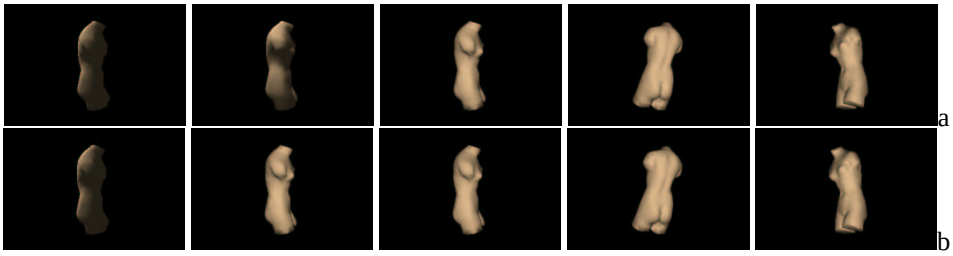

Figure 12. *Illumination de la Venus de Milo (a) maximisation de la luminance (b) maximisation du contraste. Dans les trois première colonnes, la caméra reste statique puis elle effectue un mouvement de rotation autour de la statue*

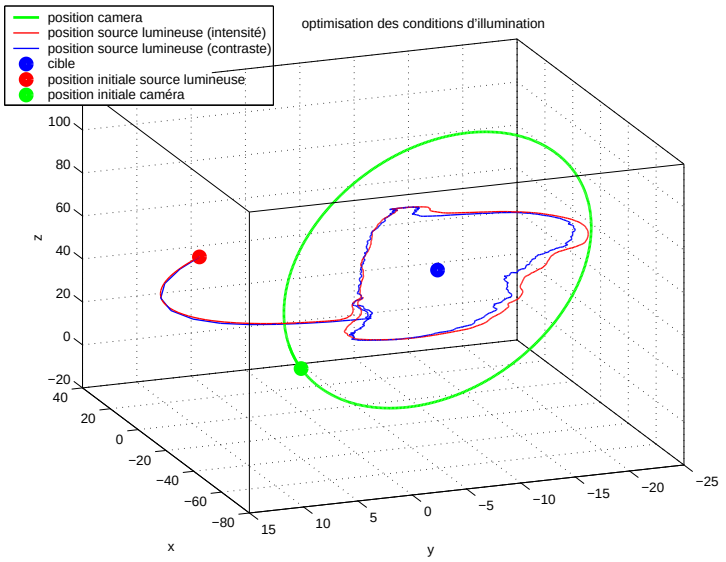

Figure 13. *Illumination de la Venus de Milo : trajectoires de la source lumineuse et de la caméra*

environnement, font de cette technique un outil intéressant pour ces applications où il n'y a pas place pour un animateur.

Les perspectives de ce travail sont multiples. Comment nous l'avons déjà précisé la trajectoire 3D de la caméra n'est pas maîtrisée. D'autre techniques d'asservissement visuel récemment introduites pourrait permettre de résoudre en partie ce problème. L'asservissement visuel 2 1/2 D [MAL 99, CHA 00] ou 2D 3/4 permettent en effet d'assurer des trajectoires rectilignes de la caméra. D'autre part, nous avons considéré dans ce rapport un nombre limité de tâches secondaires. On peut cependant penser à un grand nombre d'extensions possibles, ces contraintes doivent s'exprimer sous la

forme d'une fonction de coût, on peut souvent trouver des critères modélisant tel ou tel comportements (on peut ainsi penser, par exemple, aux multiples règles régissant de façon très stricte le cinéma [ARI 93]). Ce ne sont cependant pas les seules perspectives à ce travail, on peut aussi imaginer intégrer ces tâches élémentaires dans des dispositifs plus ambitieux où les éléments (par exemple des acteurs virtuels) réagiraient à la fois à des scénarii prédéfinis mais aussi à leur environnement au travers de ce principe de vision active.

Remerciements

Les auteurs tiennent à remercier François Chaumette pour tous ses commentaires.

Animations

La plupart des animations présentées dans cette article peuvent être visualisées sous la forme de fichier Mpeg sur le site WWW du projet VISTA de l'IRISA - INRIA Rennes : `http://www.irisa.fr/vista` (suivre le lien `demo`).

7. Bibliographie

[ARI 93] ARIJON D., *Grammaire du langage filmé*, Édition Dujarric, Paris, 1993.

[BLI 98] BLINN J., «Where Am I ? What Am I Looking At ?», *IEEE Computer Graphics and Application*, 1998, p. 76-81.

[CHA 00] CHAUMETTE F., «Asservissement visuel.», DOMBRE E., Ed., *Commande des robots rigides et souples*, Traité IC2, Hermès, á paraître, 2001.

[CHR 96] CHRISTIANSON D., ANDERSON S., HE L.-W., SALESIN D., WELD D., COHEN M., «Declarative Camera Control for Automatic Cinematography», *Proc of AAAI'96 conference*, Portland, Oregon, 1996, p. 148-155.

[COU 99] COURTY N., «Navigation et contrôle d'une caméra dans un environnement virtuel: une approche référencée image», Rapport de DEA, IRISA Université de Rennes 1, Juin 1999.

[COW 88] COWAN C., KOVESI P., «Automatic Sensor Placement from Vision task Requirements», *IEEE trans. on Pattern Analysis and Machine intelligence*, vol. 10, n° 3, 1988, p. 407-416.

[DRU 94] DRUCKER S., ZELTZER D., «Intelligent Camera Control in a Virtual Environment», *Graphics Interface'94*, Banff, Canada, 1994, p. 190-199.

[ESP 92] ESPIAU B., CHAUMETTE F., RIVES P., «A new approach to visual servoing in robotics», *IEEE Trans. on Robotics and Automation*, vol. 8, n° 3, 1992, p. 313-326.

[GLE 92] GLEICHER M., WITKIN A., «Through-the-lens camera control», *ACM Computer Graphics, SIGGRAPH'92*, Chicago, Juillet 1992, p. 331–340.

[HAS 93] HASHIMOTO K., *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*, World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapoure, 1993.

[HE 96] HE L.-W., COHEN M., SALESIN D., «The virtual cinematographer: a paradigm for automatic real-time camera control and directing», *Proc. of ACM SIGGRAPH'96, in Computer Graphics Proceedings*, New Orleans, Août 1996, p. 217-224.

[HUT 96] HUTCHINSON S., HAGER G., CORKE P., «A tutorial on Visual Servo Control», *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, 1996, p. 651–670.

[JAR 98] JARDILLIER F., LANGUÉNOU E., «Screen-Space Constraints for Camera Movements: the Virtual Cameraman», *Proc. of EUROGRAPHICS'98*, 1998, p. 176–186.

[KHA 86] KHATIB O., «Real-Time Obstacle Avoidance for Manipulators and Mobile Robots», *Int. Journal of Robotics Research*, vol. 5, n° 1, 1986, p. 90–98.

[KYU 95] KYUNG M., KIM M.-S., HONG S., «Through-the-Lens Camera Control with a Simple Jacobian Matrix», *Proc. of Graphics Interface '95*, Quebec, Canada, Mai 1995, p. 171-178.

[LAV 97] LAVALLE M., GONZÁLEZ-BAÑOS H.-H., BECKER C., LATOMBE J.-C., «Motion strategies for maintaining visibility of a moving target», *Proc. IEEE International Conference on Robotics and Automation, ICRA'97*, vol. 1, Albuquerque, Avril 1997, p. 731-736.

[MAL 99] MALIS E., CHAUMETTE F., BOUDET S., «2 1/2 D Visual servoing», *IEEE Transactions on Robotics and Automation*, vol. 15, n° 2, 1999, p. 238-250.

[MAR 98] MARCHAND E., HAGER G.-D., «Dynamic Sensor Planning in Visual Servoing», *IEEE Int. Conf. on Robotics and Automation*, vol. 3, Louven, Belgique, Mai 1998, p. 1988–1993.

[MAR 99] MARCHAND E., «ViSP: A Software Environment for Eye-in-Hand Visual Servoing», *IEEE Int. Conf. on Robotics and Automation, ICRA'99*, vol. 4, Detroit, Michigan, Mai 1999, p. 3224-3229.

[MAR 00a] MARCHAND E., COURTY N., «Image-Based Virtual Camera Motion Strategies», FELS S., POULIN P., Eds., *Graphics Interface Conference, GI2000*, Montreal, Quebec, Mai 2000, Morgan Kaufmann, p. 69-76.

[MAR 00b] MARCHAND E., COURTY N., «Visual Servoing in computer Animation. What you want to see is really what you get !», rapport n° 1310, Mars 2000, IRISA.

[NEL 94] NELSON B., KHOSLA P., «Integrating sensor placement and visual tracking strategies.», *IEEE Int. Conf. Robotics and Automation*, vol. 2, San Diego, May 1994, p. 1351–1356.

[SAM 91] SAMSON C., LE BORGNE M., ESPIAU B., *Robot Control: the Task Function Approach*, Clarendon Press, Oxford, Royaume-Uni, 1991.

[TAR 95] TARABANIS K., ALLEN P., TSAI R., «A Survey of Sensor Planning in Computer Vision», *IEEE trans. on Robotics and Automation*, vol. 11, n° 1, 1995, p. 86-104.

[WAR 90] WARE C., OSBORN S., «Exploration and virtual camera control in virtual three dimensional environments», *Proc. 90 Symposium on Interactive 3D Graphics*, Mars 1990, p. 175–183.

Article reçu le 27 mars 2000
Version révisée le 24 août 2000
Rédacteur responsable : Rachid Deriche

Nicolas Courty *est né en Décembre 1976 à Limoges. Ingénieur de l'Institut national de sciences appliquées de Rennes, option informatique, il a soutenu en 1999 un DEA en informatique à l'université de Rennes 1. Il prépare actuellement un doctorat dans le projet* SIAMES *de l'*IRISA-INRIA *Rennes en collaboration avec France Telecom R&D. Ses travaux de recherche concernent la commande référencée capteur, principalement la vision, adaptée à l'animation comportementale d'acteurs autonomes de synthèse.*

Éric Marchand *est né en janvier 1970 à Rennes. Il a soutenu en 1996 un doctorat en informatique à l'Université de Rennes 1. Il a effectué en 1997 un séjour Post-Doctoral dans le laboratoire de robotique et vision artificielle de l'Université de Yale. Il est maintenant chargé de recherche INRIA dans le projet* VISTA *à l'*IRISA-INRIA *Rennes. Ses travaux de recherche concernent les stratégies de perception par vision active et principalement la coopération entre la perception et l'action.*