N^o d'ordre : D02-09

THÈSE

présentée

DEVANT L'INSA DE RENNES

pour obtenir

le grade de : ***DOCTEUR DE L'INSA DE RENNES***

Mention : Informatique

par

Nicolas Courty

Équipes d'accueil : SIAMES (IRISA, RENNES)
FRANCE TELECOM R & D RENNES

École doctorale : Mathématiques, Informatique, Signal, Électronique et Télécommunications

Composante Universitaire : INSA de Rennes

Titre de la thèse :
Animation référencée vision : de la tâche au comportement

Soutenance effectuée le 28 Novembre 2002, devant la commission d'examen

COMPOSITION DU JURY :

M.	Kadi	BOUATOUCH	Président
M.	Ronan	BOULIC	Rapporteurs
Mme	Sylvie	GIBET	
MM.	Bruno	ARNALDI	Examinateurs
	Yves	DUTHEN	
	Eric	MARCHAND	
Mme	Danielle	PELÉ	Membre invité

L'étudiant Doko vient voir un maître zen et dit :
"Je cherche la vérité. Quel état d'esprit dois-je rechercher, pour parvenir à la trouver ?"
Le maître dit :
"Il n'y a pas d'esprit, ainsi ne peut-il être dans aucun état. Il n'y a pas de vérité, ainsi ne pouvez-vous pas chercher à l'atteindre"
"Si il n'y a pas d'esprit à former ni aucune vérité à trouver, pourquoi appelez-vous ces moines à se réunir devant vous chaque jour pour étudier le Zen et se former à son étude ?"
"Mais il n'y a absolument pas de place ici, dit le maître, comment les moines pourraient-ils se réunir ? Je n'ai pas de langue, alors comment pourrait-je les appeler à se réunir ou leur dispenser un enseignement ?"
"Oh, comment pouvez-vous mentir ainsi ?" demande Doko
"Mais si je n'ai pas de langue pour parler aux autres comment puis-je vous mentir ?"
Doko dit alors tristement :"Je ne vous suis pas. Je ne vous comprends pas."
"Je ne me comprends pas moi-même" dit le maître.
Köan Zen

Remerciements

Cette thèse s'est déroulée au sein de l'Institut de Recherche en Informatique et Systèmes Aléatoires, au sein du projet Siames, ainsi qu'au laboratoire de recherche de France Télécom R&D à Rennes.

Je tiens à remercier Kadi Bouatouch, Professeur à l'Université de Rennes I, pour m'avoir fait l'honneur de présider mon jury de thèse, ainsi que pour toutes les discussions que nous avons pu avoir pendant ces trois années.

J'adresse également tous mes plus vifs remerciements à Sylvie Gibet (maître de conférence habilitée à l'Université de Bretagne Sud) ainsi qu'à Ronan Boulic (Adjoint scientifique au Virtual Reality Lab de l'EPFL) d'avoir accepté de rapporter cette thèse, et d'avoir réalisé cette tâche avec beaucoup de sérieux et d'intérêt.

Merci à Yves Duthen, Professeur à l'Université de Toulouse, de m'avoir fait le plaisir d'être examinateur dans ce jury de thèse.

Merci à Danielle Pelé, Ingénieur de recherche France Télécom R&D, de m'avoir accueilli dans son laboratoire, et de m'avoir permis de mener cette thèse jusqu'au bout. Je tiens notablement à saluer ses efforts pour conserver un contact avec mes travaux durant toute cette thèse, même si ma présence à France Télécom n'a été parfois que sporadique. Je profite de cette occasion pour saluer tous les gens du laboratoire DIH/HDM, avec qui j'ai eu énormément de plaisir à travailler, et j'espère à ce titre que de futurs collaborations verront le jour.

Je souhaite remercier Bruno Arnaldi, Professeur à l'Insa de Rennes et responsable du projet Siames, pour m'avoir accueilli dans son équipe ainsi qu'encouragé, conseillé et aidé tout au long de cette thèse. Je tiens à lui préciser que je serai toujours disponible pour participer à ses déménagements, ainsi qu'aux festivités qui vont avec.

Une grande partie de ces remerciements vont à Éric Marchand, Chargé de recherche INRIA pour l'encadrement de mes travaux, sa disponibilité de tous les moments, ainsi que pour les multiples relectures de ce document. Son aide m'a été plus que précieuse.

Cette thèse s'est déroulée au sein du bureau du bonheur E114, qui, avec un passif déjà lourd (Guillaume, Gwenola Turbo, Richard), s'est vu peuplé par une nouvelle bande de gentils allumés : . Un grand merci aux anciens occupants, ainsi qu'à Babouze et à Romano, vous savez ce que je vous dois. J'ai été très heureux et chanceux de pouvoir effectuer ma thèse dans le projet Siames, dont je tiens à remercier tous les membres : Tangi glouzaille *Scratchy scratchy dog* (de son vrai nom Michel Palourdiau), Guillermax, Stéphane Ménardais, Guillaume, Stéphane "Darth Vader" Donikian, Georges "Moustacheman", Jiff, Jean-Eudes, Mathilde, Gérald, Thierry, QTS (pour de franches rigolades) et tous les autres. J'apporte ici une mention spéciale aux journées au vert qui nous ont fait découvrir des facettes insoupçonnées des forces vives de Siames, ainsi qu'à Evelyne et Catherine pour nous avoir simplifié la vie.

L'Irisa est sans contestes un environnement de travail très épanouissant où l'on rencontre de multiples personnalités. Parmi celles-ci, Thomas, mon fidèle compagnon de route

depuis les TPs informatique de la première année d'INSA, mérite toute mon amitié pour les cartes sur table, la buleria, les tourtes au thon et les soirées VIP/salsa du TNB, de même que Pierre pour les repas de con, les soirées vin/fromages et nos multiples débats. Un très grand merci à Matt pour les bidouilles (bisous à Isa, Félix et Basile), Samy la langouste pour ses odeurs de pop-corn, Tanguy pour m'avoir initié et perfectionné au GO (et m'avoir par la même occasion désociabilisé le midi à la cafet'), Dou-bidou, Colcombix pour le kibbitz, Benoït pour les imitations de Deniro, PAA pour les poissons cuisinés aux 102 légumes, les dames de la cafet', les dames de la bibliothèque, tous les autres ainsi que tous les anciens (Pink, Jean Nico, ...).

Sorti de l'Irisa, les réjouissances continuaient dans la maison du bonheur. Merci à Filou l'italo-picard, Titine l'accordéoniste, Erwann le compositeur, Stouf "Miss Courte Echelle", Django le chien, Vince le Bacri aveyronnais, Sandrine, Marianne la philosophe. Merci aussi à mes éternels amis : Etienne ragarou (olie-to-croute), Mister Zogz, Cha-cha, Grokuik le magnifique.

Mille tendresses à Margot qui m'a apporté beaucoup pendant ces trois années.

Merci à mes parents de m'avoir amené jusqu'ici, cette thèse leur est dédiée.

Table des matières

Notations **11**

Introduction générale **13**

1 Contexte de l'étude et contributions **15**
1.1 De l'animation guidée à l'animation comportementale : problématique . . . 15
1.2 Animation référencée vision . . . 16
1.3 Animation et simulation comportementales . . . 17
1.3.1 Cinématographie virtuelle . . . 18
1.3.2 Exploitation de la perception visuelle d'un humanoïde de synthèse . 19
1.4 Organisation du document . . . 20

I Animation référencée vision **21**

2 Contrôle d'une caméra virtuelle **23**
2.1 État de l'art sur les techniques d'animation de caméra . . . 24
2.1.1 Calcul de trajectoires . . . 24
2.1.2 Spécifications de l'image à obtenir . . . 25
2.1.3 Discussion . . . 26
2.2 Contrôle de caméra et asservissement visuel . . . 26
2.2.1 L'asservissement visuel . . . 27
2.2.2 Asservissement visuel 2D . . . 28
2.2.3 Définition d'une tâche visuelle . . . 29
2.2.4 Résultats : asservissement visuel et mouvement de caméra . . . 30
2.2.5 Conclusion partielle . . . 33
2.3 Introduction de contraintes dans l'asservissement visuel . . . 34
2.4 Suivi de trajectoires . . . 36
2.4.1 Suivi d'une courbe . . . 36
2.4.2 Résultats . . . 37
2.5 Changement réactif de point de vue . . . 40
2.5.1 Evitement d'obstacles . . . 40
2.5.2 Evitement des occultations . . . 41
2.5.3 Résultats . . . 49
2.6 Problème de la "photographie" . . . 56
2.6.1 Modélisation . . . 56

2.6.2 Modification de la position de la source lumineuse 57
2.6.3 Résultats 58
2.7 Conclusion 60

3 Contrôle d'un humanoïde virtuel 61
3.1 État de l'art sur les techniques d'animation d'humanoïde 62
3.1.1 Techniques d'animation 62
3.1.2 Exigences de l'animation comportementale d'humanoïde 65
3.1.3 Discussion et définition d'un cahier des charges 66
3.2 Modélisation à l'aide de l'asservissement visuel 67
3.2.1 Importance et représentation de la chaîne articulaire 68
3.2.2 Modélisation et formalisme 68
3.2.3 Résultats : possibilités et intérêts 71
3.2.4 Conclusion partielle 72
3.3 Gestion des butées articulaires 75
3.3.1 Approches classiques 75
3.3.2 Calcul d'un mouvement adéquat 76
3.3.3 Résultats 77
3.4 Transitions entre différentes tâches 80
3.4.1 Transition d'un modèle de robot à un autre 80
3.4.2 Transition d'une tâche visuelle à une autre 80
3.4.3 Résultats 85
3.5 Gestion du mouvement de l'humanoïde 87
3.5.1 Modélisation de la tâche visuelle 87
3.5.2 Gestion de la locomotion 87
3.5.3 Résultats 89
3.6 Réalisme visuel des animations : vers une posture de moindre effort 90
3.6.1 Couplages particuliers 90
3.6.2 Gestion des postures de moindre effort 91
3.6.3 Résultats 91
3.7 Conclusion 93

Conclusion 95

II Animation référencée vision et comportements 97

Animation et simulation comportementales 99

4 Techniques d'animation et de simulation comportementales 101
4.1 Etat de l'art sur la modélisation de comportements 101
4.1.1 Modèles comportementaux simples 102
4.1.2 Modèles comportementaux comportant les aspects réactifs et cognitifs 103
4.1.3 Rappel des objectifs et discussion 107
4.2 Formalisme utilisé : HPTS 107
4.2.1 Description 107
4.2.2 Coordination des comportements 108

4.2.3 Lien avec notre système d'animation . . . 109
4.3 Conclusion . . . 110

5 Cinématographie virtuelle 111
5.1 Etat de l'art sur la cinématographie virtuelle . . . 112
5.1.1 Temps-réel ou post-production ? . . . 112
5.1.2 Approche agent . . . 112
5.1.3 Discussion et définition d'une architecture . . . 113
5.2 Animation et contraintes "cinématographiques" . . . 114
5.2.1 Définition de primitives de contrôle . . . 115
5.2.2 Le metteur en scène . . . 117
5.3 Conclusion . . . 123

6 Perception visuelle d'un humanoïde de synthèse 125
6.1 Généralités sur la perception visuelle humaine . . . 125
6.1.1 La perception visuelle chez l'homme . . . 126
6.1.2 Modélisation de la perception visuelle en synthèse d'image . . . 127
6.1.3 Discussion . . . 129
6.2 Modélisation des comportements de vision . . . 129
6.2.1 Gestion de la cinématique associée à la perception visuelle . . . 130
6.2.2 Processus endogène : guidé par la volonté . . . 132
6.2.3 Processus exogène : utilisation de cartes de saillance . . . 134
6.2.4 Dualité du contrôle . . . 137
6.3 Mise en œuvre et résultats expérimentaux . . . 139
6.3.1 Comportement de surveillance . . . 139
6.3.2 Perception passive . . . 139
6.4 Conclusion . . . 145

7 Validation pratique de ces méthodes 149
7.1 Agent conversationnel . . . 149
7.1.1 Principe . . . 149
7.1.2 Résultats . . . 151
7.2 Exemple de fiction interactive . . . 154
7.2.1 Présentation . . . 154
7.2.2 Environnement de développement et implémentation . . . 156
7.2.3 Résultats . . . 157

Conclusion 161

Conclusion générale 163

Annexe 167

A Principe du Point de vue/Point visé ou *Look-At/Look-From* 169

B Quelques matrices d'interaction **171**
B.1 Calcul de la matrice d'interaction associée à un point 171
B.2 Matrice d'interaction associée à une droite 172
B.3 Matrice d'interaction associée à une sphère 172

C Splines de Kochanek **173**

D Définition d'un modèle de robot : les paramètres de Denavit-Hartenberg **177**

Table des figures **180**

Bibliographie **185**

Publications **197**

Notations

Règles générales

- a : scalaire
- $\mathbf{a}$: vecteur
- $\mathbf{A}$: matrice

Mathématiques

- $\mathbf{I_n}$: matrice identité de dimension n
- $\dot{\mathbf{v}}$: variation temporelle du vecteur v
- $\mathbf{M}^+$: pseudo-inverse de la matrice $\mathbf{M}$

Géométrie

- $\mathbf{p}$: point 2D exprimé dans le repère de l'image
- $\mathbf{p}(X,Y)$: coordonnées 2D du point
- ${}^o\mathbf{P}$: point 3D exprimé dans le repère o
- ${}^o\mathbf{P}(x,y,z)$: coordonnées 3D du point
- $\mathbf{t}$: vecteur de translation
- $\mathbf{R}$: matrice de rotation
- ${}^o\mathbf{M_c}$: matrice homogène exprimant le passage du repère o au repère c

Robotique

- θ : angle de rotation
- h_s : fonction de coût
- $\mathbf{e}$: fonction de tâche
- $\mathbf{r}$: vecteur représentant l'état de l'effecteur du robot
- $\mathbf{q}$: vecteur des coordonnées articulaires du robot
- $\mathbf{Jac}(\mathbf{q})$: matrice jacobienne du robot
- $\mathbf{P}$: vecteur d'état courant
- $\mathbf{P_d}$: vecteur d'état désiré
- $\mathbf{L_X}$: matrice d'interaction associée à la primitive $\mathbf{X}$

Introduction générale

La synthèse d'images offre l'opportunité de créer l'image de choses qui n'existent pas. Au delà d'une représentation purement symbolique ou artistique de notre monde est apparu un usage fonctionnel de ce média ; l'univers synthétiquement créé devient vecteur d'informations servant à la fois les applications multimédia, la production de films et de jeux, et la simulation. Cette diversité du champ applicatif impose la création d'outils génériques pouvant s'adapter à différentes contraintes. Celles-ci sont de natures multiples : temps d'exécution, réalisme, esthétisme, déictisme, interactivité, etc. Il apparaît ainsi légitime de penser qu'un "bon" outil dispose d'une bonne flexibilité vis-à-vis de ces contraintes. Les différentes exigences que l'on peut définir posent le problème du **contrôle**, c'est à dire les modalités d'interactions entre l'outil et le programme ou l'utilisateur. En effet on peut penser qu'à une exigence précise correspond un ensemble de paramètres d'entrée précis. Le niveau le plus bas de contrôle est sans doute le contrôle au niveau géométrique (transformation, rotation, mise à l'échelle) et, à l'opposé, se trouve le contrôle au niveau tâche (proche d'une description formulable en langage naturel). Ce travail propose une réflexion sur la notion de niveau de contrôle, au travers de l'utilisation d'une nouvelle technique d'animation dite "référencée vision" reposant sur les techniques d'asservissement visuel. Ces techniques, issues du domaine de la robotique, permettent d'obtenir des mouvements 3D du capteur de prise de vue à partir de la spécification d'une tâche dans un espace à deux dimensions (une image perçue depuis le capteur). L'intérêt de l'utilisation de cette méthode dans le domaine de la synthèse d'image est de pouvoir proposer un niveau de contrôle intuitif pour les animateurs, autorisant ainsi un niveau de spécification parfois impossible à atteindre avec des méthodes classiques. Ce principe est, au cours de ce document de thèse, décliné selon deux classes distinctes de problèmes : l'animation d'une caméra et l'animation d'un humanoïde de synthèse. Ces deux problématiques apparemment distinctes trouvent ici des résonances dans les exigences possibles quant au niveau de contrôle. Au travers de l'étude de cette technique d'animation se pose le problème de son intégration dans des architectures plus complexes, ayant trait aux modèles comportementaux ou à des problèmes applicatifs précis. Cette intégration pose les problèmes de compétition/coopération de cette approche avec des approches plus classiques. Ainsi la deuxième contribution de cette thèse réside dans l'exploitation et la validation de notre technique d'animation référencée vision dans le cadre de la cinématographie virtuelle ou la simulation de la perception visuelle pour un humanoïde de synthèse, ces deux thématiques étant abordées selon une approche commune.

Chapitre 1

Contexte de l'étude et contributions

1.1 De l'animation guidée à l'animation comportementale : problématique

Le processus de création est un processus complexe, il met en jeu de nombreux principes conscients et inconscients. La conception d'un monde synthétique opère à différents niveaux de l'esprit ; on peut distinguer une phase de conceptualisation, une phase d'abstraction et une phase de réalisation. La phase de conceptualisation aboutit à la définition de ce que l'on cherche à représenter, la phase d'abstraction détermine les modalités nécessaires à cette représentation (définition des mécanismes, des modes de représentation, critères esthétiques, pédagogiques, ergonomiques, ludiques, etc.) et enfin la phase de réalisation représente la phase cruciale du passage de ces concepts et de ces modalités à une réalité informatique concrète. La recherche en synthèse d'images s'intéresse à la fois à la phase d'abstraction et à la phase de réalisation. Il existe de nombreux ponts entre ces deux phases, les difficultés associées à la réalisation conditionnant souvent les modalités de la représentation. Dès lors, le concepteur de monde virtuel, par inversion du processus, se base sur les outils dont il dispose pour établir ce qu'il peut réaliser. Cette ingérence de l'outil, même si elle existe dans la plupart des arts ou des domaines de l'ingénierie, constitue un obstacle au processus de création et son atténuation représente l'un des buts actuels des recherches menées.

Le cadre de cette thèse s'inscrit dans cette démarche et, plus précisément, dans le domaine de l'animation et de la simulation, en proposant des outils permettant de modéliser des éléments de base utilisables par des applications intégrant des mondes virtuels et synthétiques. Il est essentiel de percevoir les besoins associés à ces outils. D'une manière générale, ces outils permettent de modéliser des fonctionnalités précises, et d'automatiser certains processus de conception, pour rendre ce travail plus rapide et plus simple. Cette automatisation pose le problème du **contrôle**, c'est à dire des paramètres permettant d'interagir avec le processus automatisé. Prenons un exemple : il est possible de modéliser en animation le mouvement de marche d'un humanoïde. D'une manière élémentaire, ce mouvement peut être vu par la modification cohérente des variables d'état articulaires des deux jambes de l'humanoïde. Un animateur, travaillant sur les principaux logiciels de création numérique comme 3DStudio Max ou Maya, peut redéfinir toutes ces valeurs articulaires à

chaque pas de temps de son animation. Cette démarche, longue et fastidieuse, ne garantit pas de surcroît le réalisme du mouvement. Il faut alors considérer un outil permettant de générer automatiquement ce mouvement. Cet outil, basé sur un modèle théorique ou empirique de marche, peut offrir plusieurs modes de contrôle, en considérant par exemple la vitesse et l'orientation, les empreintes des pas, le style, etc. La contrepartie de cette automatisation est que l'outil doit pouvoir s'adapter à différents modèles géométriques d'humanoïde, et proposer suffisamment de modalités de contrôle pour pouvoir s'adapter à la totalité des cas. Disposer d'un tel outil permet de contrôler l'humanoïde à un niveau **tâche**, c'est à dire avec une abstraction du contrôle suffisamment importante pour être considérée comme proche du langage naturel. Une abstraction "d'ordre supérieur" définit un **comportement**, c'est à dire la réalisation avec une cohérence intrinsèque d'un ensemble de tâches au sens défini précédemment. A chaque niveau d'abstraction peut donc correspondre un ensemble de paramètres de contrôle plus ou moins intuitifs. En règle générale, cette abstraction implique une diminution du nombre de paramètres de contrôle, rendant donc l'outil plus facile à manipuler. Les avantages directs de cette abstraction du contrôle constituent paradoxalement un point négatif: une perte de maîtrise sur les résultats de l'outil qui peuvent ne pas correspondre à ce que l'on recherche idéalement.

Le travail réalisé dans cette thèse correspond à l'étude de cette abstraction du contrôle au travers de l'utilisation d'une nouvelle technique d'animation dite "référencée vision".

1.2 Animation référencée vision

L'animation "référencée vision" se base sur un ensemble de techniques largement étudiées en robotique: l'asservissement visuel. Au travers des informations perçues par un capteur (généralement une caméra embarquée sur un robot), des mouvements 3D du capteur sont générés dans le but de réaliser une consigne exprimée dans un espace 2D, l'image perçue par le capteur. Ces consignes "visuelles" définissent une **abstraction du contrôle**. Si l'utilisation d'une telle commande référencée capteurs est quasi indispensable en robotique pour définir le mouvement du robot, on peut directement se poser la question de l'intérêt de cette technique en synthèse d'images, car la plupart des informations sur l'environnement sont disponibles. L'intérêt se situe du fait des facilités de spécification offertes par une telle technique, et donc du niveau de l'abstraction du contrôle offerte. Ce niveau de contrôle est un contrôle au niveau tâche, et plus précisément des **tâches visuelles**. La figure 1.1 est un exemple d'une tâche visuelle. La tâche est spécifiée ainsi: "je veux voir le tableau de Monet centré dans l'image". Dans ce cas, seules les positions dans l'image des quatre coins du tableau sont spécifiées. Par rapport à une spécification classique de la position et de l'orientation dans l'espace $3D$, la facilité de spécification de la tâche est accrue.

Cette catégorie de contrôle s'adapte tout particulièrement bien à deux classes de problèmes: le contrôle d'une caméra en environnement virtuel et la gestion de la cinématique de l'humanoïde associée à la perception visuelle, la notion de tâche visuelle prenant un sens évident dans ces deux cadres. Dans ce document, nous étudions comment il est possible de piloter ces deux entités au travers de contraintes visuelles, et comment le formalisme que nous avons retenu nous permet de prendre en compte des contraintes inhérentes à

FIG. 1.1 – *Exemple de tâche visuelle : "voir le tableau de Monet centré dans l'image perçue" (a) état initial (b) état final*

l'animation de ces entités.

1.3 Animation et simulation comportementales

Définir un mode de contrôle par des tâches visuelles signifie une prise de contrôle au niveau de ce qui est perçu. Le choix de ces tâches constitue un problème à part entière, relatif aux notion d'**autonomie** et d'**interactivité**. L'autonomie d'une entité se traduit par l'automatisation de tous les processus lui permettant d'évoluer de manière cohérente et réaliste dans un monde virtuel. L'interactivité désigne les capacités d'un utilisateur à interagir avec les éléments du monde virtuel. Si le choix de la tâche visuelle se fait de manière automatique, une forme d'autonomie apparaît. Les modèles permettant ces choix sont appelés modèles comportementaux, le contrôle se situant alors à un niveau plus abstrait. La figure 1.2 montre le positionnement de grandes classes d'application vis-à-vis de ces deux concepts. La création de contenu (films de synthèse et effets spéciaux) ne nécessite généralement ni autonomie, ni interactivité avec les éléments créés. Dans ce domaine, tout est guidé et directement déterminé par les volontés de l'utilisateur. D'une manière complètement opposée, la notion de fiction interactive met en oeuvre des **acteurs autonomes** capables d'interagir avec des utilisateurs, ce cadre nécessitant donc le maximum d'autonomie et de possibilités d'interaction. Les jeux vidéo, la création d'assistants personnalisés et le domaine des simulateurs sont des cadres applicatifs intermédiaires, qui ne disposent à l'heure actuelle que de capacités limitées en terme d'autonomie et d'interactivité. Il est intéressant de noter que la plupart des évolutions dans ce domaine tentent d'accroître leurs potentiels en terme d'interactivité et d'autonomie. La production de contenu (films) cherche aussi à automatiser une partie du processus de création par l'utilisation d'entités autonomes (gestion de foule par exemple).

Nos travaux dans le domaine de l'animation et de la simulation comportementales s'inscrivent dans cette optique. Notre objectif est ainsi de concevoir des systèmes autonomes qui ne pénalisent pas l'interactivité, c'est à dire doués de faculté d'**adaptation** et de **réactivité**. Les deux domaines que nous avons abordés sont la cinématographie virtuelle (relative au problème de contrôle d'une caméra virtuelle) et la simulation de la perception visuelle d'un humanoïde synthétique, ces deux thématiques étant liées directement aux notions de tâches visuelles. Nous allons maintenant présenter ces deux domaines, ainsi que les perspectives applicatives liées à chacun d'eux.

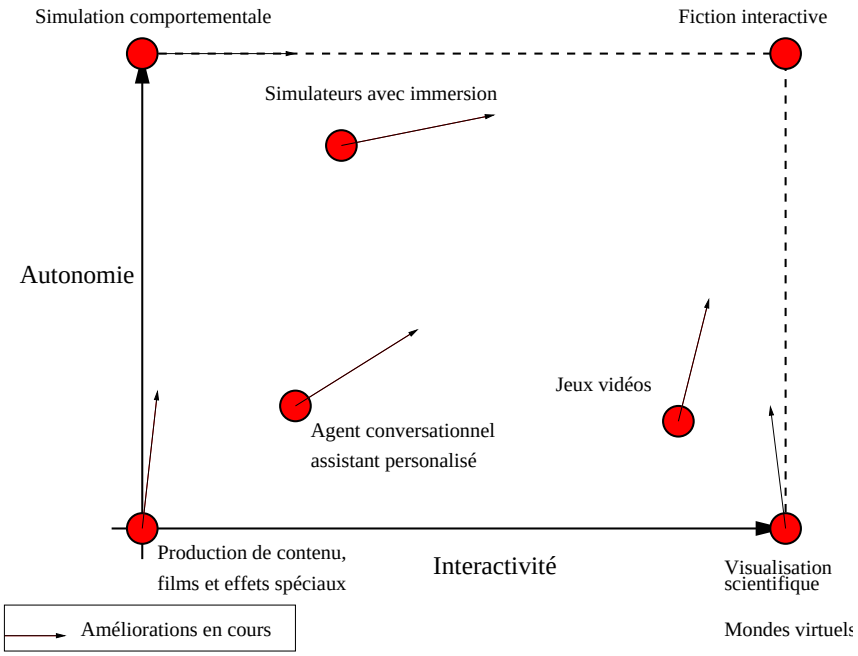


FIG. 1.2 – *Différents cadres applicatifs et leurs besoins relativement à la notion d'autonomie et à la notion d'interactivité*

1.3.1 Cinématographie virtuelle

La cinématographie virtuelle désigne la capacité à pouvoir "filmer" l'environnement virtuel de manière automatique, avec certaines règles privilégiant des aspects particuliers ou relatifs à des contraintes issues de la cinématographie réelle (le septième art). Voici quelques unes des applications possibles d'un tel système :

- *jeux vidéo*. La plupart des jeux vidéo actuels mettent en scène des héros et héroïnes évoluant dans des univers 3D. Le joueur contrôle ces personnages, et lorsque le mode de représentation n'est pas une vision subjective, il est alors nécessaire de considérer une caméra filmant l'action. Cette caméra doit alors s'adapter aux modifications de l'environnement en temps réel, et éventuellement rendre un contenu émotionnel,
- *visualisation scientifique*. La visualisation de gros ensembles de données ainsi que de simulations diverses nécessite une gestion parfois automatique du point de vue. Un système de cinématographie virtuelle autorise dans ce cas de pouvoir choisir automatiquement ces points de vue en respectant des règles établies par les concepteurs. A titre d'exemple, considérons la simulation d'une foule de gens. Il peut être intéressant de disposer d'une caméra se focalisant automatiquement sur les éléments de la simulation pertinents ou porteurs de sens, par exemple des points de passage particuliers ou des comportements erratiques,
- *applications collaboratives* orientées réalité virtuelle. Ce troisième thème fait référence aux applications partagées en réalité virtuelle. Dans des environnements collaboratifs, plusieurs personnes distantes physiquement travaillent sur les mêmes objets virtuels. Ce cadre applicatif soulève le problème de l'image perçue, au sens ergonomique. Un système gérant plusieurs caméras peut aider à établir une certaine cohérence sur ce

qui est montré dans les différents sites,

- *production de films et d'effets spéciaux.* Un système de cinématographie virtuelle peut créer des ébauches de plans relativement à des storyboards (spécifications de haut niveau sous forme illustrée), ces ébauches constituant alors une base de travail pour les animateurs.

1.3.2 Exploitation de la perception visuelle d'un humanoïde de synthèse

Ce domaine désigne la capacité de l'humanoïde à pouvoir acquérir de l'information sur son environnement en utilisant la vision, ainsi qu'à gérer la cinématique associée (c'est à dire les mouvements générés pour percevoir) . En voici quelques applications potentielles :

- *avatars réalistes* et communautés virtuelles. Avec le développement d'internet, de nombreuses communautés virtuelles ont émergé (*Active World* ou *Blaxxun* par exemple), ou les dialogues en mode texte ont été remplacées par des personnages synthétiques symbolisant les différents interlocuteurs (*avatars*). L'utilisation de modèles 3D permet d'établir des modes d'interaction plus riches, et ajoute une personnalisation des échanges qui favorise une communication plus humaine. La plupart des avatars peuvent rejouer une série d'animations pré-enregistrées, et l'utilisateur perçoit la scène du point de vue subjectif de son avatar ou d'un point de vue global. On s'aperçoit très rapidement que dans ce genre d'environnement il est très difficile de savoir qui s'adresse à qui, ou comment capter l'attention de quelqu'un. Il existe donc une nécessité dans le cadre de ce domaine applicatif de modéliser les comportements visuels des différents avatars. Cette information ajoute en effet un sens qui modifie notre compréhension des interactions. De nombreux travaux [Gibet 98,Cassell 00] se focalisent sur la modélisation des comportements non-verbaux dans le cadre des conversations, ainsi que leur importance vis-à-vis des dialogues. Cependant la gestion du regard se limite souvent à regarder ou non l'interlocuteur. Même si l'importance du regard est magnifiée, peu de modèles théoriques abordent ce problème,
- *jeux vidéo.* Les possibilités offertes par la 3D ont été exploitées dans la conception des dernières générations de jeux vidéos. Dans ce genre d'applications la notion d'acteur autonome de synthèse est devenue indispensable, que ce soit pour tirer sur le joueur ou participer avec lui à des quêtes romanesques, historiques ou d'anticipation. La crédibilité associée à ces personnages virtuels dépend bien entendu de la qualité de leurs animations ainsi que de la cohérence comportementale qu'ils témoignent tout au long du jeu. L'ajout de modèles de perception permet de définir des interactions plus complexes entre le joueur et ces personnages virtuels,
- *production de films et d'effets spéciaux.* La synthèse d'images apparaît comme un médium novateur aux possibilités faiblement limitées. Le nombre de productions de films basés sur cette technologie a considérablement augmenté depuis ces dernières années, avec notamment des œuvres "phares" comme *Toy Story, Final Fantasy* ou bien encore *Shrek.* Comme toute production susceptible de ramener une grosse quantité d'argent, le rythme de production de ces films s'est considérablement accru, le travail des artistes étant paradoxalement soumis à des exigences de qualité croissantes. Une solution consiste donc à déporter une partie de leur travail en tâches automatisées qui génèrent des ébauches de courbes d'animation, retouchables et modifiables (cette automatisation permet de se focaliser sur de nouveaux aspects comme

le texturage ou l'éclairage). Cette approche est d'autant plus justifiée dans le cadre de la modélisation de l'attention visuelle que de concevoir des comportements d'attention crédibles se révèle être un problème difficile,

- *validation ergonomique de sites virtuels ou réels.* La création de lieux publiques pose des problèmes ergonomiques, c'est à dire relatifs à la qualité de leur utilisation. Les ergonomes définissent la notion de charge cognitive, désignant nos capacités à recevoir de l'information de l'environnement, et servant de critères de qualité pour ces environnements. A titre d'exemple, la conception d'une gare pose le problème de lisibilité des panneaux et autres indicateurs, et il est difficile d'évaluer l'ergonomie de ces lieux autrement qu'en situation réelle (une fois peuplés et avec des publicités un peu partout). Il est alors utile dans ce cas de passer par une phase de simulation permettant de déterminer si la charge cognitive d'un utilisateur serait adaptée à la navigation dans ce genre de lieux. Pour cela il faut donc pouvoir peupler ces environnements avec des acteurs autonomes disposant de modèles de perception visuelle. La validité de ce genre de simulation fait l'objet d'études ergonomiques (modélisation de musées, de gares ou d'aéroports par exemple), mais les modèles employés sont parfois trop simplistes, rendant les résultats approximatifs.

1.4 Organisation du document

Ce document est décomposé en deux parties principales. La première partie traite du formalisme que nous avons établi pour définir la notion d'**animation "référencée vision"**. Dans cette partie seront étudiées à la fois l'animation d'une caméra dans un environnement virtuel ainsi que l'animation d'un humanoïde relativement à des informations visuelles. La deuxième partie est consacrée à l'application de cette technique dans le cadre de l'animation et de la simulation comportementales. Après avoir défini une base commune de modélisation des comportements, nous étudierons en détail les architectures dédiées à la cinématographie virtuelle ainsi qu'à la simulation de la perception visuelle d'un humanoïde de synthèse. Les travaux issus de cette thèse seront dans un dernier temps illustrés par deux exemples d'application.

Première partie

Animation référencée vision

Chapitre 2

Contrôle d'une caméra virtuelle

Le contrôle d'une caméra dans un environnement virtuel est sans doute un des problèmes les plus fréquemment rencontrés par les créateurs de mondes virtuels. On peut facilement faire l'analogie entre la caméra et les yeux d'un observateur ; et l'on comprend aisément que son positionnement et ses déplacements conditionnent totalement la manière que l'utilisateur aura de percevoir l'environnement et les autres acteurs du monde virtuel. Le contrôle de caméra a donc une importance cruciale, et c'est ce problème que nous considérons dans cette partie.

On peut, à ce stade, distinguer deux types d'exigences : d'une part la volonté du créateur de définir explicitement tous les déplacements de la caméra (auquel cas les différentes techniques de contrôle doivent lui permettre de réaliser des effets que l'on pourrait qualifier de cinématographiques), et d'autre part un désir de fournir à l'utilisateur un contrôle de haut niveau sur les éléments de l'environnement virtuel, par exemple poursuivre un objet tout en évitant son occultation par d'autres parties de la scène, ou se focaliser sur des composantes précises de la scène. Une difficulté est ici de prendre en compte les modifications de l'environnement.

Drucker [Drucker 94a] formule cette problématique en cherchant à déterminer la nature des tâches visuelles qu'un utilisateur peut vouloir effectuer dans un monde virtuel, et par extension les paramètres de la caméra qu'il faut alors changer. Il propose alors la classification suivante :

- *domaine interactif de l'exploration.* Le contrôle d'une caméra virtuelle permet à l'utilisateur de parcourir les différentes facettes d'une base de données contenant des informations spatiales. Dans le domaine de la visualisation scientifique ou médicale, une interprétation visuelle d'une grosse masse d'informations permet d'obtenir des résultats conséquents [Hong 97]. On peut aussi envisager des ballades ludiques ou éducatives, avec par exemple la visite d'un musée virtuel [Drucker 94b],
- *planification.* Elle définit la manière de déplacer la caméra ou manipuler des objets de la scène, dans le but de présenter les résultats à un observateur qui n'interagit pas avec l'environnement. Certains systèmes d'animation permettent ce genre de spécifications à l'aide de contraintes narratives ou stylisées [He 96],
- *observation et interaction.* Cette tâche visuelle se définit par ses objectifs, à savoir pouvoir permettre à un utilisateur d'être au courant de tous les faits importants

ou critiques se déroulant dans la scène. Un tel système permettrait à l'utilisateur de pouvoir prédire le comportement de certains objets dans la scène et d'agir en conséquence. Un exemple est le contrôle de trafic aérien,

- *contrôle.* L'utilisateur doit pouvoir exercer un contrôle précis sur l'image que lui propose la caméra dans le but d'accomplir une tâche précise. Ce type de tâche visuelle trouve un écho dans les systèmes de téléopération, ou dans des outils de modélisation.

Un système complet de contrôle de caméra devrait pouvoir permettre à l'utilisateur d'effectuer ces tâches dans leur diversité. Différentes méthodes de contrôle de caméra ont été étudiées au cours des quinze dernières années mais il semble qu'aucune de celles-ci ne satisfasse complètement toutes les exigences du domaine. La technique d'animation référencée vision développée dans cette partie tente de répondre aux exigences de contrôle mises en évidence par Drucker. Les autres aspects évoqués seront envisagés dans la partie relative à la cinématographie virtuelle (car ne relevant pas du contrôle directement).

La structuration de cette partie est la suivante : dans un premier temps, nous allons étudier les différents travaux réalisés dans le cadre du contrôle de caméra afin de dégager les limites des systèmes existants. Dans un deuxième temps, nous spécifierons la modélisation du problème de contrôle de caméra à l'aide d'asservissement visuel. La suite du chapitre est dédiée à l'étude des différentes grandes classes de problèmes inhérents à l'animation de caméra : évitement des occultations et d'obstacles, suivi et planification de trajectoires, problème de la "photographie". Notons que les aspects ayant trait au comportement de la caméra et donc à un contrôle de plus haut niveau sur les enchaînements des tâches d'observation seront traités dans la deuxième partie de ce document.

2.1 État de l'art sur les techniques d'animation de caméra

Il est intéressant en tout premier lieu de mieux comprendre comment une caméra est représentée dans un environnement 3D virtuel. Une caméra virtuelle possède six degrés de liberté (trois degrés en translation et trois degrés en rotation, nous les noterons ddl par la suite). Il semble difficile pour un animateur de pouvoir calculer les variations des six degrés nécessaires définissant le mouvement de la caméra permettant d'obtenir les images (l'animation) désirées. Plusieurs modèles de caméra ont donc été introduits dans le but de rendre son contrôle plus intuitif, ainsi que des techniques prenant en compte le contenu des scènes à visualiser.

2.1.1 Calcul de trajectoires

Pour permettre à l'utilisateur de pouvoir spécifier plus clairement les mouvements de caméra, plusieurs modèles de caméra ont été utilisés :

- *métaphores au sens de Ware et Osborne.* Partant du principe qu'il est identique de faire bouger la caméra ou de déplacer la scène toute entière autour de la caméra, Ware et Osborne [Ware 90] décrivent trois métaphores pour contrôler une caméra à l'aide d'un périphérique à six degrés de liberté. Ces métaphores sont : *scene in hand, eyeball in hand* et *flying vehicle control* (ce dernier traite le cas d'une caméra *non-holonomique*). Ils ont entre autre établi que la dernière de ces métaphores est

plus pratique pour se déplacer dans des espaces fermés, et que *scene in hand* est plus avantageux que *eyeball in hand* quand il faut traiter des objets uniques (dans ce cas c'est la scène que l'on déplace autour de la caméra). Il est alors possible d'enregistrer les différentes trajectoires de la caméra, puis de les rejouer dans le but de faire un film. Ces métaphores sont la base d'un grand nombre de réflexions sur la gestion des caméras,

– *positions clés (Keyframes).* Cette technique permet d'expliciter facilement un mouvement souple et continu. Elle se base sur l'interpolation de positions clés ou *keyframe* spécifiées par l'utilisateur. (voir Fig. 2.1). La principale difficulté d'une telle technique est de spécifier les bonnes postions dans l'espace 3D, car on ne maîtrise pas forcément les interpolations entre chacune de celles-ci (ces interpolations dépendent en plus du modèle de caméra utilisé),

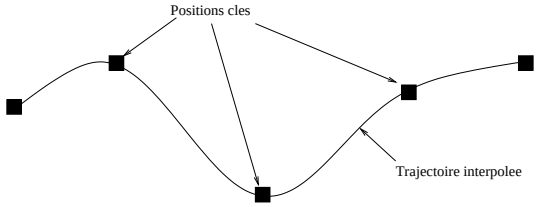


FIG. 2.1 – *Principe du keyframing*

– *modèles physiques.* Pour produire des animations réalistes, Turner [Turner 91] utilise un modèle physique de caméra prenant en compte des paramètres comme la masse, la friction et la viscosité. Il semble que l'apport de ces paramètres n'apportent finalement pas grand chose du point de vue du résultat et nécessitent de surcroît des temps de calcul élevés.

2.1.2 Spécifications de l'image à obtenir

Le contrôle de caméra repose généralement sur la technique du Point de vue/Point visé ou *LookAt/LookFrom* . La difficulté du contrôle revient à préciser ces états à deux variables (position et observation) et à gérer les changements d'un état à un autre (on se référera à l'annexe A pour une description plus détaillée de cette technique). Le processus de gestion de la caméra que nous présentons est singulièrement différent : la génération de la position et de l'orientation de la caméra est obtenue grâce à une spécification de ce que l'on cherche à observer dans l'image. Les premiers travaux basés sur cette approche furent menés par Blinn dans son article *Where am I? What am I looking at?* [Blinn 88]. Différentes approches existent :

– *localisation et orientation de la caméra.* Le travail de Blinn [Blinn 88] est motivé par la volonté de faire une animation d'un vaisseau spatial survolant une planète. La problématique est alors la suivante : comment positionner et orienter la caméra afin d'obtenir une vue d'ensemble intéressante? Blinn suggère de spécifier plutôt la position désirée du vaisseau dans l'image ainsi que celle de la planète, l'orientation de l'axe de la planète dans l'image, etc. Son approche se base sur une réécriture des vecteurs inconnus (position et orientation de la caméra) en terme de vecteurs

connus (définis par le problème). Cependant il apparaît que cette technique ne traite que des cas très spécifiques et ne peut pas se généraliser à des problèmes multi-contraints. C'est cependant un travail intéressant car c'est le premier à considérer une spécification de la tâche *dans l'image* et non plus dans l'espace de la scène,

- *contraintes sur des personnages.* Drucker [Drucker 94b] propose un module d'animation de caméras. Cette plate-forme propose un contrôle de plus haut niveau de la caméra grâce à une série de fonctions sémantiquement beaucoup plus proches du langage cinématographique habituel : suivre un personnage, faire un plan panoramique englobant plusieurs individus, etc. Cependant l'utilisation de ces contraintes se limite à des mouvements simples de la caméra ; définir des trajectoires plus complexes de la caméra fait appel à un processus de planification qui se base sur une fonction d'optimisation fournie par l'utilisateur. Chaque position de la caméra est donc calculée en fonction de cette fonction (les interpolations sont impossibles pour ne pas violer les contraintes),
- *Modélisation déclarative.* Jardillier et Languenou [Jardillier 98] proposent 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, le système 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.

2.1.3 Discussion

Il apparaît au travers de cette étude qu'il existe un fossé entre le niveau déclaratif de ce que l'on cherche à obtenir, et les moyens techniques pour y arriver. Les approches basées image (où la déclaration des contraintes s'effectue dans l'espace de ce que voit la caméra) sont des approches intéressantes dans le sens où la spécification des tâches est plus simple et se rapproche du formalisme utilisé en cinématographie [Arijon 84]. Cependant celles-ci ne disposent la plupart du temps que peu de flexibilité (résolution de problèmes dédiés) ou sont inadaptées pour les applications temps-réel. Dans la partie suivante, nous proposons une approche basée sur l'asservissement visuel, une méthodologie largement utilisée dans le domaine de la robotique. Le but de cette approche est de lever les limitations des systèmes actuels en proposant un cadre supportant des contraintes multiples, pouvant s'exécuter en temps-réel et disposant des mêmes facilités de spécification.

2.2 Contrôle de caméra et asservissement visuel

Une approche particulièrement intéressante a été proposée par Gleicher et Witkin dans l'article *Through-The-Lens Camera Control* [Gleicher 92]. Dans cet article, une technique particulière de contrôle de caméra est proposée : un calcul du torseur cinématique de la caméra est réalisé dans le but de maintenir une configuration voulue dans l'image "perçue" par la caméra. Ce calcul se base sur une technique d'optimisation sous contraintes, et

revient à résoudre un système linéaire aux moindres carrés. Cette approche permet entre autre de contraindre plusieurs caractéristiques dans l'image (exemple: la position ou la trajectoire de deux points). Cet article décrit une tâche d'asservissement visuel classique en robotique; cependant il ne propose pas de généralisation de la méthode, ni de moyen d'introduire des contraintes dans la trajectoire de la caméra, ce qui est utile pour résoudre des problèmes propres au contrôle de caméra virtuelle. Dans un premier temps des généralités sur l'asservissement visuel seront présentées, puis nous étudierons plus en détail le formalisme nous permettant de contrôler une caméra virtuelle.

2.2.1 L'asservissement visuel

L'opération consistant à transformer des consignes en commandes, directement utilisables par un système, est appelée asservissement. Les techniques d'asservissement visuel ou de commande référencée vision [Chaumette 90,Espiau 92,Hashimoto 93,Hutchinson 96] permettent de spécifier la commande d'un robot directement en fonction d'informations perçues par une ou plusieurs caméras. Souvent, la loi de commande qui en découle peut être exprimée comme la régulation d'une fonction de tâche [Samson 91]. Une classification des différents types d'asservissement visuel peut être établie en fonction du type d'informations utilisées pour établir la loi de commande:

- **asservissement 3D** (ou *position-based visual servoing*) utilise en entrée de la boucle de commande des informations tridimensionnelles exprimées dans un repère euclidien. Plus précisément, dans le cadre d'une tâche de positionnement rigide (c'est à dire utilisant les six degrés de liberté de la caméra), la consigne est exprimée sous la forme d'un changement de repère entre deux repères décrivant la position et l'orientation d'un objet d'intérêt et de la caméra, cette transformation étant réévaluée à chaque pas de temps à l'aide de méthodes considérant une connaissance à priori du modèle 3D de l'objet,
- **asservissement 2D** [Chaumette 90,Espiau 92] (ou *image-based visual servoing*) utilise en entrée de la boucle de commande des informations exclusivement extraites de l'image. La consigne est alors exprimée comme la différence entre un motif courant et un motif désiré dans l'espace de l'image. Cet asservissement se base sur la définition des matrices d'interaction ou Jacobien image, liant les mouvements de primitives visuelles à la vitesse de la caméra,
- **asservissement 2D 1/2** [Malis 98]. Cette approche, novatrice et existante depuis peu, se base sur l'estimation de l'homographie qui relie l'image de trois points. Dans le cadre d'un objet plan, un minimum de quatre points appariés sur les images courantes et désirées permet une telle estimation par résolution d'un système linéaire (il en faut huit autrement). L'erreur en rotation est alors estimée par cette homographie. Le découplage dans la loi de commande entre la rotation et la translation engendre des propriétés intéressantes,
- **asservissement d2D/dt** [Crétual 98]. Dans ce cas les mouvements de la caméra sont contrôlés à partir d'informations visuelles dynamiques. La consigne s'exprime alors comme la régulation du mouvement 2D à un champ de vitesse désiré. Ce type d'asservissement pallie aux problèmes liés à l'extraction des primitives visuelles dans l'image.

De nombreux travaux existent sur l'utilisation de l'asservissement visuel dans le cadre où la caméra est embarquée (elle se déplace alors avec le robot) : conduite automatisée d'automobiles [Pissard-Gibollet 93], évitements d'occultations [Marchand 98]. De nombreux travaux existent aussi lorsque la caméra est déportée [Allen 93,Hager 95]. Plusieurs travaux ont enfin considéré l'utilisation de différentes caméras (embarquées et déportées), en règle générale à des fins de reconstruction 3D, pour de l'asservissement visuel basé sur une paire d'images [Hager 97], ou de la coopération entre une caméra globale et une caméra locale [Marchand 98,Flandin 01].

Dans le cadre des travaux réalisés, nous nous sommes surtout intéressés à l'application des techniques d'asservissement visuel 2D. De plus, les problèmes liés à l'utilisation d'une telle technique en synthèse d'images sont extrêmement simplifiés, le calcul de la position des primitives 2D pouvant se faire de manière analytique (et non plus par extraction dans l'image), rendant le calcul de la matrice d'interaction, reliant le mouvement des objets à la vitesse de la caméra, exact (ce n'est plus une estimation). Cet aspect sera complété dans la suite de ce document.

2.2.2 Asservissement visuel 2D

Le principe de l'asservissement visuel 2D repose sur une commande en **boucle fermée** qui a comme paramètres les informations perçues directement dans l'image. Le principe est de spécifier dans cette boucle une configuration à atteindre (voir Fig. 2.2). Cette configuration représente la **tâche visuelle**.

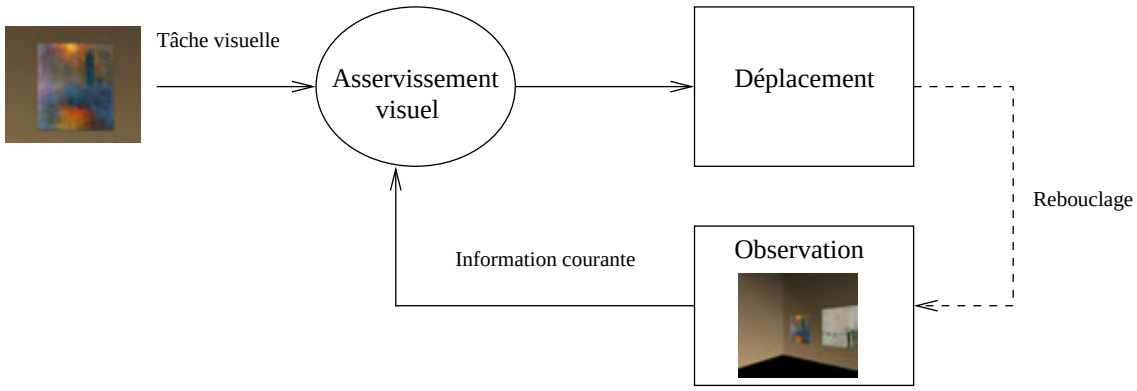


FIG. 2.2 – *Boucle d'asservissement visuel*

Notons $\mathbf{P}$ un vecteur représentant l'ensemble des informations visuelles choisies pour réaliser la tâche, et $\mathbf{P_d}$ la configuration désirée dans l'image. Par exemple, si on considère une droite déterminée par ses deux paramètres ρ et θ (distance au centre de l'image et orientation), on peut définir une configuration finale telle que cette droite apparaisse verticale centrée dans l'image. La configuration désirée est alors : $\mathbf{P_d} = (\rho_d,\theta_d) = (0,\frac{\pi}{2})$.

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) $\mathbf{L_P}$ qui relie le mouvement de l'objet dans l'image au mouvement de la caméra. Cette relation est définie par l'équation

[Espiau 92,Chaumette 90] :

$$\dot{\mathbf{P}} = \mathbf{L_P T_c} \tag{2.1}$$

où $\dot{\mathbf{P}}$ représente le mouvement de l'objet dans l'image et $\mathbf{T_c}$ le torseur cinématique ou vitesse de la caméra.

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

$$\mathbf{e_1} = \mathbf{C}(\mathbf{P} - \mathbf{P_d}) \tag{2.2}$$

où $\mathbf{C}$, appelé matrice de combinaison, est choisie de sorte que $\mathbf{CL_P}$ soit de rang plein le long de la trajectoire $\mathbf{r} \in SE^3$. $\mathbf{C} = \mathbf{L_P^+}$ si la tâche visuelle contraint les six degrés de liberté de la caméra (nous verrons comment définir $\mathbf{C}$ dans le cas où la tâche ne contraint pas les six ddl). $\mathbf{L}^+$ dénote la pseudo inverse de la matrice $\mathbf{L}$. On note que la pseudo inverse d'une matrice $\mathbf{L}$ de dimension (m,n), où inverse généralisée est définie par :

$$\mathbf{L}^+ = \begin{cases} (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T \text{ si } m > n \\ \mathbf{L}^T(\mathbf{L}\mathbf{L}^T)^{-1} \text{ sinon} \end{cases} \tag{2.3}$$

Classiquement, on obtient $\mathbf{L}^+$ par une décomposition en valeurs singulières (méthode SVD). Dans notre cas, m représente le nombre de primitives visuelles, n le nombre de degrés de liberté (six pour une caméra non-contrainte).

Pour assurer une décroissance exponentielle de l'erreur $\mathbf{P} - \mathbf{P_d}$, à savoir :

$$\dot{\mathbf{e_1}} = -\lambda \mathbf{e_1}$$

la vitesse de la caméra est donnée par :

$$\mathbf{T_c} = -\lambda \mathbf{e_1} \tag{2.4}$$

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

2.2.3 Définition d'une tâche visuelle

Une des difficultés de l'asservissement visuel basé image est de dériver la matrice d'interaction $\mathbf{L}$ correspondant à la primitive de contrôle choisie. Une méthode systématique a été proposée dans [Chaumette 90]. Toute information visuelle peut potentiellement être utilisée dans la même tâche de positionnement : coordonnées de points, orientation d'une droite, surface, sphère, distance, etc. Dans l'annexe B quelques exemples de matrices d'interaction sont proposés.

Connaissant ces matrices d'interaction, la modélisation d'une tâche visuelle est directe. Une collection de tâches basiques est ainsi définie. Le système que nous utilisons permet de définir les tâches de type X vers Y, où X et Y sont des éléments de {point, ligne, sphère, cylindre, cercle, etc.}. Définir une tâche de ce type revient à stipuler que l'on cherche à voir l'objet X "sur" l'objet Y dans l'espace de l'image. Notons ici que X et Y ne définissent pas forcément le même type de primitive. A titre d'exemple, une contrainte de type "point vers ligne" stipule que l'on cherche à voir un point 2D sur une ligne 2D.

Il est, de plus, possible d'augmenter cette collection de tâches basiques en "empilant" les

matrices d'interaction ainsi que les vecteurs d'erreur associés. Par exemple pour construire une tâche de positionnement par rapport à un segment défini par deux points $\mathbf{p_1}$ et $\mathbf{p_2}$ avec $\mathbf{p_{d_1}}$ et $\mathbf{p_{d_2}}$ leurs positions désirées dans l'image, la matrice d'interaction est définie par :

$$\mathbf{L_p} = \begin{bmatrix} \mathbf{L_{p_1}} \\ \mathbf{L_{p_2}} \end{bmatrix} \tag{2.5}$$

où $\mathbf{L_{p_i}}$ est définie par (si $\mathbf{p_i} = (X,Y)$ et z est la profondeur du point [Hutchinson 96, Chaumette 90]) :

$$\mathbf{L_{p_i}} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \tag{2.6}$$

Le calcul de cette matrice d'interaction peut être trouvée dans l'annexe B et dans [Chaumette 90]. Le vecteur d'erreur impliqué dans l'équation 2.2 est donné par :

$$(\mathbf{P} - \mathbf{P_d}) = \begin{pmatrix} X_1 - X_{d_1} \\ Y_1 - Y_{d_1} \\ X_2 - X_{d_2} \\ Y_2 - Y_{d_2} \end{pmatrix} \tag{2.7}$$

La loi de contrôle totale est donc donnée par :

$$\mathbf{T_c} = -\lambda \times \begin{pmatrix} -1/z_1 & 0 & X_1/z_1 & X_1Y_1 & -(1+X_1^2) & Y_1 \\ 0 & -1/z_1 & Y_1/z_1 & 1+Y_1^2 & -X_1Y_1 & -X_1 \\ -1/z_2 & 0 & X_2/z_2 & X_2Y_2 & -(1+X_2^2) & Y_2 \\ 0 & -1/z_2 & Y_2/z_2 & 1+Y_2^2 & -X_2Y_2 & -X_2 \end{pmatrix}^{+} \begin{pmatrix} X_1 - X_{d_1} \\ Y_1 - Y_{d_1} \\ X_2 - X_{d_2} \\ Y_2 - Y_{d_2} \end{pmatrix} \tag{2.8}$$

2.2.4 Résultats : asservissement visuel et mouvement de caméra

Dans cette partie, nous allons explorer quelques possibilités offertes directement par l'asservissement visuel : tâche de positionnement, suivi dans l'image d'un objet se déplaçant, et poursuite d'un objet dans une scène.

2.2.4.1 Tâche de positionnement

Nous reprenons dans cette expérience l'exemple de tâche visuelle présenté dans l'introduction. La définition de cette tâche était "voir le tableau de Monet centré dans l'image". Pour réaliser cette tâche nous devons contrôler tous les degrés de liberté de la caméra. Une manière de réaliser cette tâche (il en existe d'autres) est de chercher à voir un rectangle à une position donnée dans l'image. Ce rectangle peut être défini comme quatre points. La matrice d'interaction relatif au rectangle est alors définie par :

$$\mathbf{L_R} = [\mathbf{L_{p_1}}, \mathbf{L_{p_2}}, \mathbf{L_{p_3}}, \mathbf{L_{p_4}}]^T \tag{2.9}$$

où $\mathbf{L_{p_i}}, i = 1,..,4$ est définie dans l'équation 2.6. $\mathbf{L_R}$ est alors une matrice 8×6 et est de rang plein (tous les degrés de liberté sont impliqués dans la tâche).

Les positions désirés des quatre points sont alors définies par (a,b), $(-a,b)$, $(-a,-b)$, $(a,-b)$ où a et b sont relatifs à la taille réelle de l'objet graphique représentant le tableau et de la taille désirée dans l'image. La figure 2.3 montre les résultats associés à cette démonstration. Les images 2.3(a) sont des images relatives à la vue de la caméra et à une vue de dessus montrant cette caméra (verte dans l'image). Les figures 2.3(b,c) montrent l'évolution des vitesses calculées respectivement en translation et rotation. La figure 2.3(d) montre l'évolution de l'erreur associée aux quatre points. Dans cet exemple, les six ddl de la caméra sont contraints, aucun autre mouvement ne peut donc être donné à la caméra.

2.2.4.2 Suivi dans l'image ou tracking

Nous considérons à ce niveau une tâche différente de la tâche de positionnement, dans le sens où la cible est un objet en mouvement. L'erreur $(\mathbf{P} - \mathbf{P_d})$ dépend donc à la fois du mouvement de la caméra et à la fois du mouvement de l'objet d'intérêt (l'objet sur lequel on réalise la tâche de focalisation). Il faut donc rectifier la commande envoyée à la caméra pour tenir compte de ce mouvement. Cette erreur dite de "traînage" dans l'image due au déplacement intrinsèque de l'objet dans la scène peut s'annuler en cherchant à compenser les variations de $\mathbf{P}$ dues aux mouvements de l'objet.
Le mouvement de l'objet dans l'image peut se réécrire par :

$$\dot{\mathbf{P}} = \mathbf{L_P T_c} - \mathbf{L_P T_0} \tag{2.10}$$

où $\mathbf{L_P^T T_c}$ et $\mathbf{L_P^T T_0}$ 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 :

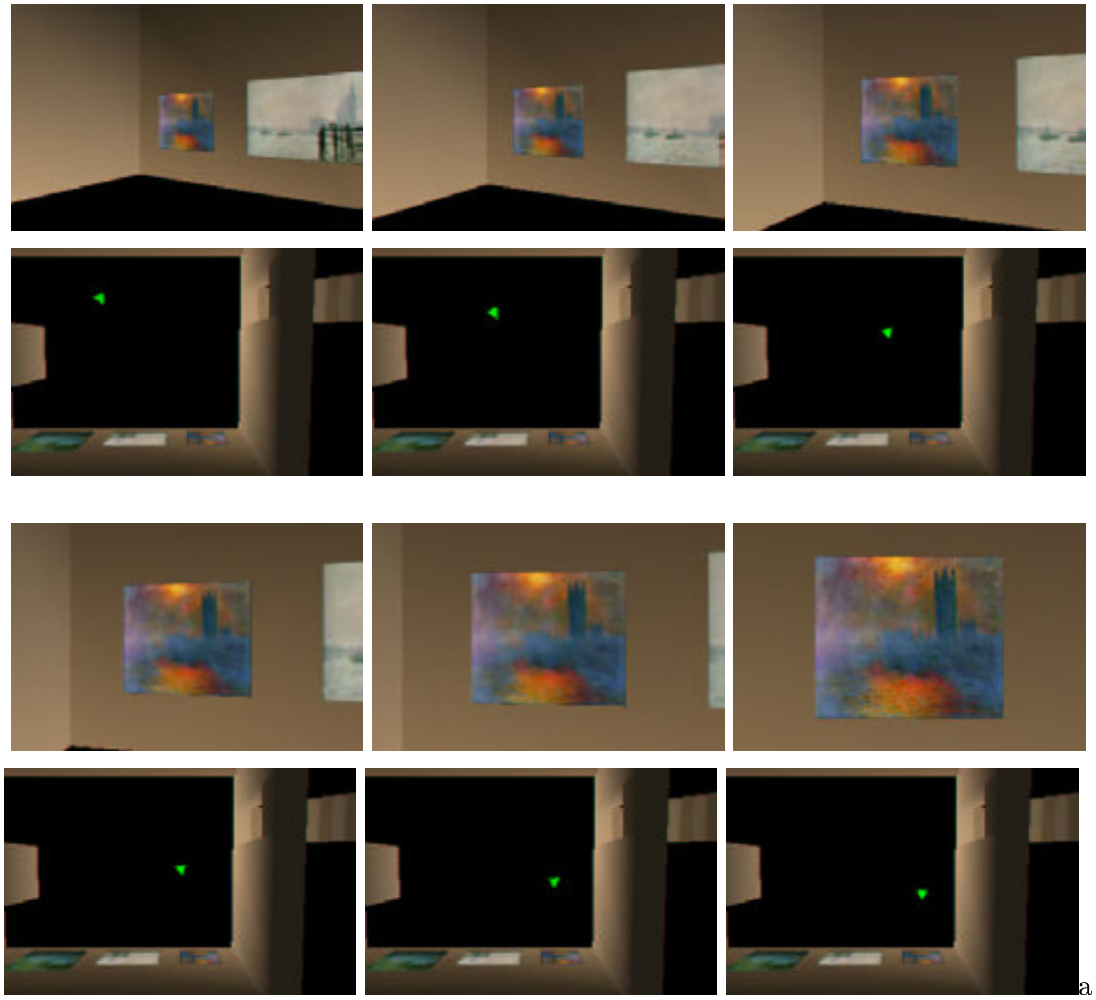
$$\mathbf{T_c} = -\lambda \mathbf{e} - \mathbf{T_0} \tag{2.11}$$

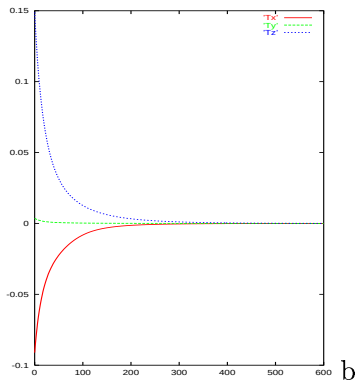
Rajout d'un effet "esthétique" de traînage. Il est possible de ne pas supprimer complètement les erreurs de traîne, pour ne pas avoir un mouvement trop parfait de la caméra et simuler par là une certaine forme d'inertie. Il suffit de pondérer la force du terme de correction en y associant un scalaire :

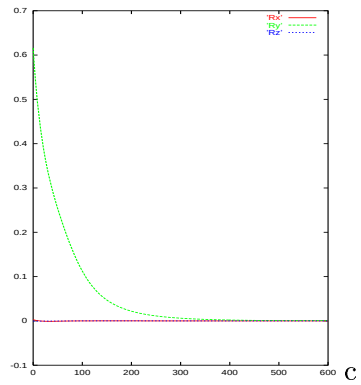
$$\mathbf{T_c} = -\lambda \mathbf{e} - \alpha \mathbf{T_o}$$

où α est un scalaire tel que $\alpha \in [0,1]$.

La figure 2.4 montre les évolutions des erreurs dans l'image dans une expérience de tracking simple. Un objet se déplace dans l'environnement avec une vitesse constante. Un asservissement est réalisé sur un point, le centre de l'objet. Au début de l'animation, on choisit une configuration telle que l'erreur soit nulle (objet face à la caméra). Dans la figure 2.4(a), on s'aperçoit que l'erreur augmente de manière forte, avant de redécroître (cependant elle ne s'annule pas). C'est le mouvement propre de l'objet qui induit cette erreur. La figure 2.4(b) montre une compensation beaucoup plus forte de cette erreur ($\alpha = 0.9$) et enfin la figure 2.4(c) montre une compensation totale. Dans ce cas, l'objet reste toujours centré dans l'image quelque soient ses mouvements.







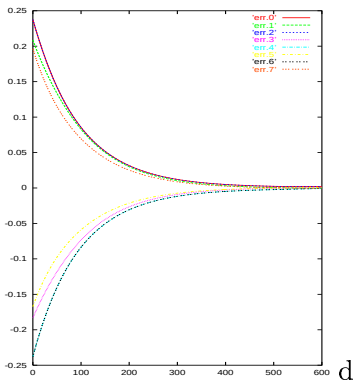


FIG. 2.3 – *Tâche de positionnement relative à un rectangle (a) vue de la caméra et vue du dessus correspondante, la caméra est en vert (b) vitesse de la caméra en translation, (c) en rotation (d) erreurs dans l'image relative aux positions des quatre points*

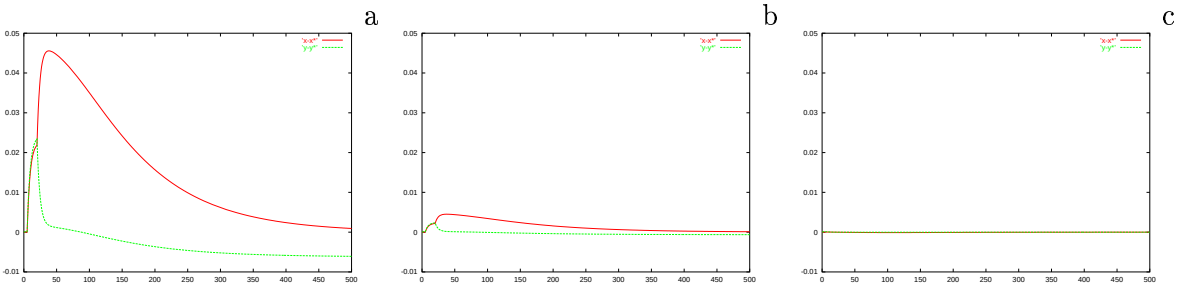


FIG. 2.4 – *Erreurs dans trois expériences de tracking (a) sans correction (b) avec un scalaire* $\alpha = 0.9$ *(c) avec une correction totale de l'erreur*

2.2.4.3 Poursuite d'un objet dans une scène

Plusieurs méthodes existent pour modéliser la poursuite d'un objet dans une scène, c'est à dire la gestion des déplacements de la caméra relatifs à l'objet d'intérêt (au travers des degrés de liberté en translation). Dans un premier temps, nous proposons une méthode basée sur l'utilisation de la sphère comme primitive d'intérêt. Dans la suite de ce document, une deuxième méthode sera proposée, fondée sur le principe de redondance (exploitation des degrés de liberté non utilisés par la tâche visuelle).

Pour pouvoir contraindre la distance entre l'objet et la caméra, il est possible de choisir une sphère comme primitive. La définition de la matrice d'interaction associée à une sphère peut être trouvée dans l'annexe B et dans [Chaumette 90]. Un tel asservissement contraint trois degrés de liberté de la caméra quand on cherche à la voir centrée dans l'image (sa projection étant alors un cercle), dont la translation selon l'axe de la caméra. Quand la sphère se déplace, la caméra utilise cette translation pour maintenir la tâche, et poursuit conséquemment la sphère dans la scène. Les images présentées dans la figure 2.5 sont extraites d'une animation où un tel asservissement est réalisé. Deux types d'images sont proposés : une vue de la caméra et une vue de dessus (cette vue a d'ailleurs été réalisée en utilisant notre technique d'animation). La caméra reste à une distance fixe de l'objet. On note qu'à deux reprises l'objet d'intérêt s'écarte de sa position désirée. Ces deux écarts sont dus aux deux changements d'orientation du mouvement de la sphère au cours de l'animation. Cette erreur dans l'image correspond à l'erreur de tracking que nous avons décrite auparavant, et peut être totalement supprimée en utilisant la technique décrite précédemment.

2.2.5 Conclusion partielle

Dans le début de ce chapitre, nous avons extrait un certain nombre de caractéristiques importantes pour un système d'animation de caméra : simplicité dans la spécification des tâches d'observation, adaptation à des modifications dynamiques de l'environnement, exécution en temps-réel. L'utilisation de l'asservissement visuel nous permet de répondre à la plupart de ces contraintes. L'originalité de ce système réside surtout dans le passage d'une spécification de la tâche *en 2D* à un mouvement *en 3D*. Mais cette approche n'est pas exempte de défauts : on ne dispose que de peu de contrôle sur la réalisation de la tâche, et donc sur les trajectoires 3D résultantes. Cette lacune peut rendre le système im-

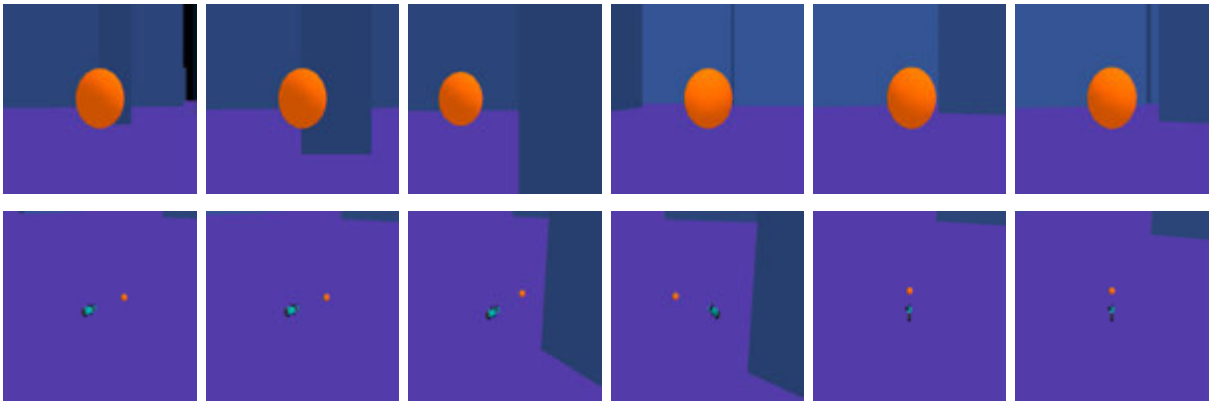


FIG. 2.5 – *Poursuite d'un objet dans une scène, vue de la caméra et vue du dessus*

propre aux besoins d'un animateur qui peut vouloir disposer d'un contrôle total sur les déplacements de la caméra. Nous allons voir dans le chapitre suivant comment utiliser le principe de redondance (le fait que certains degrés de liberté ne soient pas utilisés dans la tâche de positionnement) pour disposer d'un contrôle plus grand sur la caméra et modéliser des problèmes survenant fréquemment dans ce domaine. Les applications actuelles nécessitent de faire se déplacer ces caméras dans des environnements encombrés, parfois avec des contraintes de styles. L'objet de la prochaine section est l'utilisation du principe de redondance, qui permet d'approfondir la spécification du comportement de la caméra. En effet, l'utilisation des degrés de liberté non utilisés par la tâche de positionnement peut servir à réaliser des **tâches secondaires** (cette fois-ci pas forcément exprimées dans l'image), réalisées à condition que les tâches de positionnement soient réalisées.

2.3 Introduction de contraintes dans l'asservissement visuel

Dans cette partie, nous allons voir comment il est possible d'exploiter le principe de redondance au niveau de la loi de commande, et nous étudierons son fonctionnement sur un cas précis : la poursuite d'un objet dans une scène.

Si la tâche de vision spécifiée ne contraint pas les six 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 $\mathbf{C} = \mathbf{W}\mathbf{L}_\mathbf{P}^+$ et nous obtenons alors la fonction de tâche suivante [Samson 91] :

$$\mathbf{e} = \mathbf{W}^+\mathbf{e_1} + (\mathbf{I_6} - \mathbf{W}^+\mathbf{W})\mathbf{e_2} \tag{2.12}$$

où :

- $\mathbf{W}^+$ et $\mathbf{I_6} - \mathbf{W}^+\mathbf{W}$ 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 $\mathbf{L_p}$. En raison du choix de $\mathbf{W}$, $\mathbf{I_6} - \mathbf{W}^+\mathbf{W}$ appartient théoriquement au noyau Ker $\mathbf{L_P}$, ce qui implique que la réalisation de la tâche secondaire n'aura aucun effet sur la tâche primaire ($\mathbf{L_P}(\mathbf{I_6} - \mathbf{W}^+\mathbf{W})\mathbf{e_2} = 0$).

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

La loi de commande est maintenant donnée par [Espiau 92,Chaumette 90] :

$$\mathbf{T_c} = -\lambda \mathbf{e} - (\mathbf{I_6} - \mathbf{W}^+\mathbf{W})\frac{\partial \mathbf{e_2}}{\partial t} \tag{2.13}$$

L'utilisation de cette tâche secondaire permet de résoudre des problèmes non-triviaux dans le domaine du contrôle de la caméra. Le paragraphe suivant montre un exemple d'utilisation directe du principe de redondance sur un exemple déjà évoqué dans le premier chapitre : la poursuite d'un objet dans une scène. L'intérêt associé à l'utilisation de la redondance sera ensuite magnifiée dans le reste de ce chapitre.

Etude de cas : poursuite d'un objet dans une scène. Le problème est identique au problème évoqué dans la partie 2.2.4.3. La tâche consistant à se focaliser sur un objet ne nécessite en général que deux degrés de liberté (*pan* et *tilt* en anglais), désignant respectivement les rotations de la caméra autour de l'axe vertical ainsi que l'axe horizontal du repère de la caméra. Ces degrés de liberté n'ont aucune influence sur la distance à l'objet d'intérêt. Celle-ci doit pouvoir être maintenue à une valeur constante ou bien encore paramétrable (ce cas de figure étant le cas le plus intéressant, en effet faire se déplacer un objet dans une environnement encombré implique que sa distance à la caméra ne soit pas tout le temps fixe, comme nous le verrons par la suite). Selon le principe de définition d'une tâche secondaire, on cherche à créer une fonction de coût h_s nulle lorsque la caméra est à bonne distance de l'objet, et croissante lorsqu'elle s'éloigne de cette distance. Une telle fonction peut se définir simplement par :

$$h_s = \frac{1}{2}(z - z_d)^2 \tag{2.14}$$

où z représente la distance courante entre la caméra et l'objet, et z_d la distance souhaitée. On a alors :

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial h_s}{\partial x} \\ \frac{\partial h_s}{\partial y} \\ \frac{\partial h_s}{\partial z} \\ \frac{\partial h_s}{\partial \theta} \\ \frac{\partial h_s}{\partial \rho} \\ \frac{\partial h_s}{\partial \gamma} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ z - z_d \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.15}$$

Les techniques de suivi implémentées donnent de bons résultats (voir Fig. 2.6), d'autant plus si elles sont couplées avec la technique de suppression de l'erreur de traînage.

Combinaison de plusieurs tâches secondaires. Il est possible de combiner plusieurs tâches secondaires entre elles ; dans ce cas plusieurs possibilités existent. La première consiste à combiner linéairement ces différentes tâches. Il faut alors s'assurer que ces tâches sont compatibles entre elles, ce qui n'est pas dans tous les cas évident. Une deuxième solution consiste à mettre *en cascade* les différentes tâches secondaires. On définit alors un

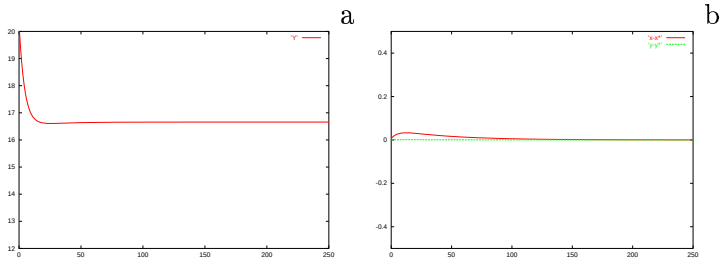


FIG. 2.6 – *Courbes associées à la poursuite d'un objet : (a) distance à l'objet (15 est la valeur souhaitée) (b) erreurs entre la position désirée dans l'image et la position réelle*

ordre de priorité dans l'exécution des tâches : la seconde tâche secondaire ne sera réalisée qu'à condition que la première le soit, etc. Dans ce cas, cette seconde tâche est projetée sur le noyau de la première [Baerlocher 98], et ainsi de suite. L'utilisation de ce type de projection en cascade ne sera pas traité dans ce document.

L'utilisation du principe de redondance est dans la suite de ce chapitre exploitée au travers de trois grandes classes de problèmes : le suivi de trajectoire, le changement réactif de point de vue, qui peut engendrer des collisions avec l'environnement ou des occultations de l'objet d'intérêt, ainsi que le problème dit de la "photographie" qui considère les conditions d'éclairement d'une scène.

2.4 Suivi de trajectoires

Le problème considéré dans cette partie est le suivi d'une trajectoire préétablie dans la scène. Cette trajectoire peut être définie par l'utilisateur ou de manière automatique si l'on considère un pré-travail de planification. L'utilisation d'une tâche secondaire pour assurer la poursuite est de laisser une marge de liberté à la caméra, elle peut par exemple, dans un cas complexe, avoir à s'éloigner de cette trajectoire, ce qui est possible si l'on considère que le suivi n'est réalisé que si la tâche de focalisation l'est.

2.4.1 Suivi d'une courbe

Il faut pour suivre une courbe définir une fonction de coût minimum lorsque la caméra est située sur la courbe et augmentant lorsque celle-ci s'en éloigne. Pour permettre à la caméra de se déplacer sur la courbe, il faut considérer un point se déplaçant sur celle-ci et chercher à l'atteindre à chaque itération. Cette approche est facilitée lorsque la courbe est exprimée sous une forme paramétrée $C(t)$. Pour chaque valeur t telle que $t \in [0,1]$, on obtient un point de l'espace 3D. Faire varier t sur cet intervalle permet de décrire entièrement la courbe. Le choix du temps selon lequel ce paramètre décrit [0,1] spécifie alors le temps que va mettre la caméra par rapport à la courbe. Dans l'annexe C, ce problème est abordé de manière plus détaillée. Le problème se ramène donc à déplacer la caméra vers un point précis de l'espace (voir Fig. 2.7).

Soit $C(t) = (x_s(t),y_s(t),z_s(t))$ un point de la spline. $\|\mathbf{OC(t)}\|$ est alors la distance de ce point à la caméra dans le repère de la caméra. On peut définir une fonction de coût h_s

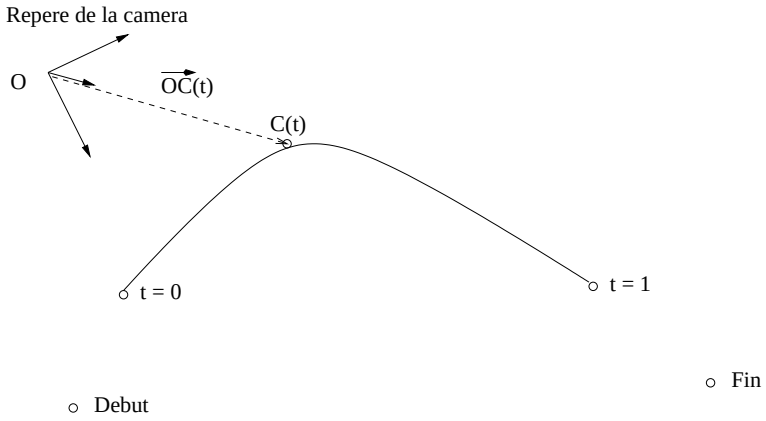


FIG. 2.7 – *Courbe paramétrique : variation du paramètre t*

telle que :

$$h_s = \frac{1}{2}\alpha\|\mathbf{OC(t)}\|^2 \tag{2.16}$$

où α est un scalaire modulant la force de rappel à la courbe. La même tâche secondaire définie à l'aide d'une exponentielle permettrait si on le désire un contrôle plus fort sur les variations de celle-ci relativement à la position par rapport à la courbe (faible quand on est proche, très élevée un peu plus loin).
On a alors (si on ne considère que la variation des paramètres translationnels) :

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial h_s}{\partial x} \\ \frac{\partial h_s}{\partial y} \\ \frac{\partial h_s}{\partial z} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \alpha\|\mathbf{OC(t)}\| \begin{pmatrix} x_s(t) \\ y_s(t) \\ z_s(t) \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

2.4.2 Résultats

Nous avons utilisé dans le cadre des résultats expérimentaux un type de courbe spline particulier, les courbes de Kochanek. L'annexe C détaille l'utilisation de ces courbes. Dans l'exemple présenté figure 2.8(a), la tâche de positionnement s'effectue par rapport à la tour (verticale centrée dans l'image). On peut trouver une tâche similaire dans [Gleicher 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 [Espiau 92]. Seulement deux ddl sont alors contraints, permettant une plus grande liberté de déplacement que dans [Gleicher 92] où l'asservissement était réalisé par rapport au deux points extrémités contraignant ainsi quatre ddl. Les premières images montrent le début de la tâche de positionnement. La suite montre le suivi d'une courbe de Kochanek (voir visualisation de la spline en 3D Fig.2.8(d)) tout en maintenant la tâche de focalisation.

L'analyse des courbes associées à cette expérience (voir Fig. 2.8(b,c)) montre deux choses :

- *variation de l'erreur.* Une fois que la droite sur laquelle on s'asservit est verticale centrée dans l'image, elle ne bouge plus de cette position : erreur quasiment nulle tout au long de l'expérience,
- *distance à la courbe.* La distance au point de la courbe que l'on cherche à rattraper ne varie pas linéairement. Cela vient du fait que dans le cas que l'on considère les points que l'on choisit sur la courbe spline ne sont pas espacés de manière linéaire.

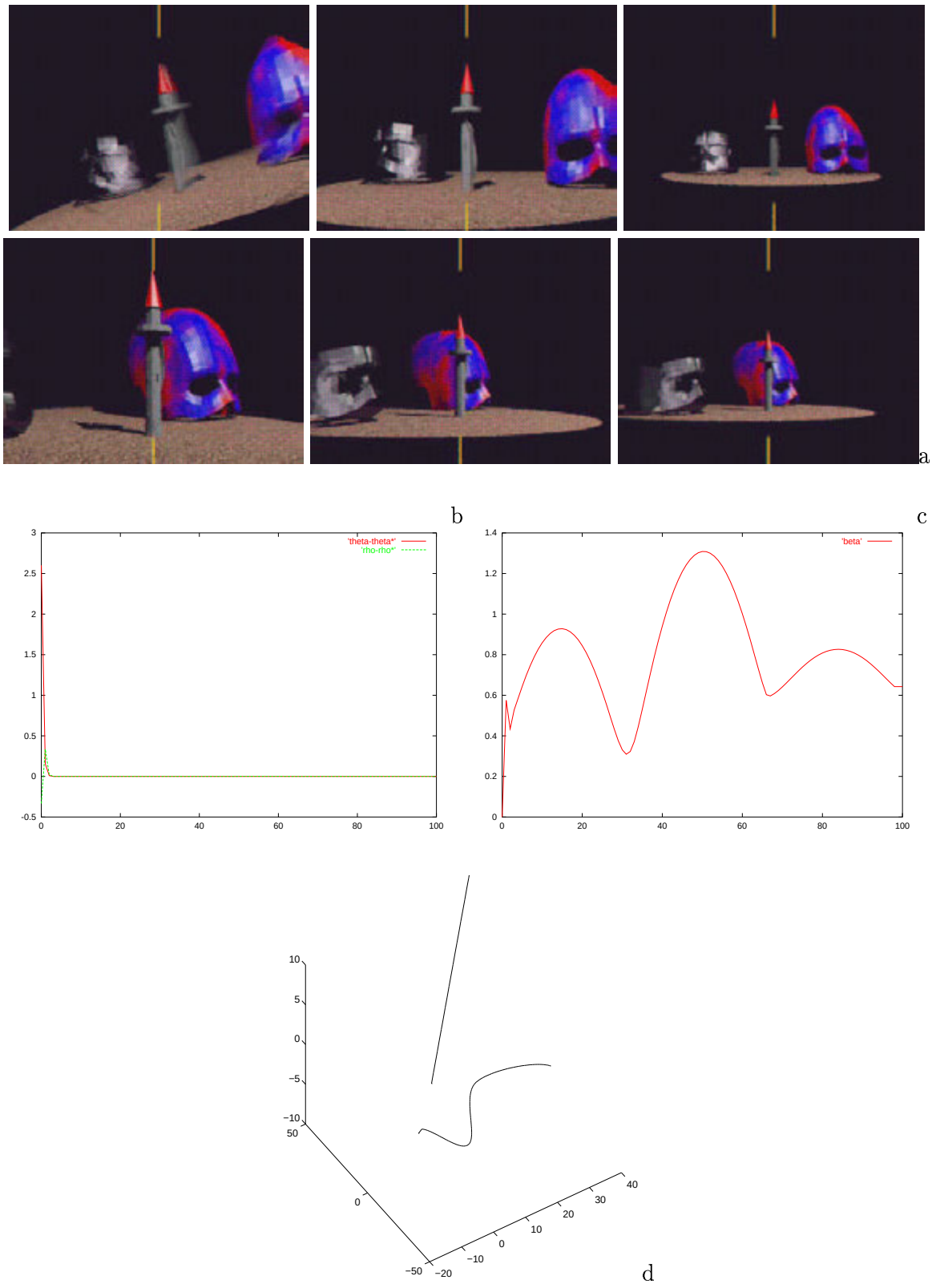


FIG. 2.8 – *Résultats associés au suivi d'une trajectoire (a) images de l'exemple rendues sous Maya (b) erreur associée à la tâche principale (c) variations de la distance au point que l'on cherche à rattraper (d) visualisation de la trajectoire de la caméra décrivant une courbe de Kochanek*

2.5 Changement réactif de point de vue

Les techniques présentées dans le chapitre précédent 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.

2.5.1 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 restants 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 [Khatib 86]. Drucker [Drucker 94a] 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.

Utilisation de la redondance. 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 (si possible infinie) lorsque la distance de la caméra à un obstacle est nulle, par exemple:

$$h_s = \alpha \frac{1}{2\|C -^c O\|^2} \tag{2.17}$$

où ${}^cC(0,0,0)$ est la position de la caméra et ${}^cO(x_c,y_c,z_c)$ est le point de l'obstacle le plus proche de la caméra, tous deux exprimés dans le repère de la caméra.

La tâche secondaire est alors définie par:

$$\mathbf{e_2} = -(x_c,\ y_c,\ x_c,\ 0,\ 0,\ 0)^T \frac{h_s^2}{\alpha} \quad \text{et} \quad \frac{\partial \mathbf{e_2}}{\partial t} = 0 \tag{2.18}$$

Par extension, considérer des obstacles multiples peut s'exprimer par la fonction de coût:

$$h_s = \sum_i \alpha \frac{1}{2\|C -^c O_i\|^2} \tag{2.19}$$

Planification de la trajectoire de la caméra La démarche envisagée maintenant est différente ; un pré-travail sur l'environnement nous permet de déterminer une trajectoire dans l'espace. Une telle technique est appelée *planification de trajectoires* ou *path planning* et se base sur les travaux établis dans [Latombe 91,Khatib 86]. La trajectoire ainsi établie peut être suivie par la méthode décrite dans la partie *Suivi de trajectoires* (2.3). On peut distinguer à ce stade deux classes de méthodes : une méthode pré-calculant le chemin à suivre dans la scène (cette méthode servant quand tous les objets de l'environnement ont des trajectoires prédéterminées ou si ils sont statiques), ou une méthode calculant le chemin dynamiquement au cours de la simulation. Ces deux méthodes présentent des avantages et des inconvénients. Dans la première, un pré-traitement de la scène permet de déterminer les minima locaux (par analyse du résultat de la planification), et donc pouvoir réagir de manière adaptée. La détermination de points de passages permet aussi de construire des courbes de trajectoire plus "esthétiques" ; elle a aussi entre autre l'avantage de ne demander qu'un calcul certes coûteux, mais qui n'est effectué qu'une fois pour toute. Elle est cependant inadaptée dans le cas des environnements dynamiques, où des objets bougent. Dans ce cas on utilise une planification dynamique, permettant de réagir localement. Cette technique ne garantit pas que la caméra ne tombe dans un minimum local. Il est alors commun d'utiliser des méthodes mixtes : les objets de l'environnement étant souvent à des positions déterminées et ne bougeant que très rarement (murs, bâtiments), il est possible de considérer des pré-traitements sur l'environnement rendant la planification possible. Les collisions avec les objets se déplaçant à l'intérieur sont alors traitées à part, à l'aide d'un système de planification locale. Dans le cadre de cette thèse, seule la technique de planification de trajectoires basée sur des champs de potentiel a été testée. L'étude de l'utilisation d'un système mixte (basé sur une triangulation de Delaunay) est en cours.

2.5.2 Evitement des occultations

L'occultation désigne la disparition dans l'image de l'objet d'intérêt. L'évitement des occultations est à priori un problème plus compliqué que l'évitement des obstacles : on considère dans ce cas le système caméra/objet d'intérêt, et non plus seulement les simples déplacements de la caméra. Plusieurs problèmes théoriques sont associés au traitement de ces configurations :

- Détermination géométrique des occultations. Comment définir les critères nous permettant de dire si l'objet est occulté ou non ?
- Prévision/anticipation de l'occultation. Dans le cadre que nous nous sommes fixé, nous ne disposons d'aucune information *a priori* sur le futur. Dans ces conditions, comment anticiper une occultation ?
- Réagir en cas d'occultation. Quelles sont les politiques de contrôle de caméra à employer en cas d'occultation ?

Dans cette partie, nous établissons dans un premier temps une typologie des situations pouvant mener à des occultations, puis nous présenterons un système de prédiction de ces occultations à base de volumes englobants. Enfin, plusieurs possibilités d'éviter ces occultations en utilisant le principe de redondance sont étudiées.

2.5.2.1 *Typologie des situations d'occultation*

On peut distinguer plusieurs configurations amenant l'objet d'intérêt à disparaître de l'image générée par la caméra.

- un objet en mouvement dans la scène passe dans le volume de vision associé à la caméra et à l'objet d'intérêt, créant par là une occultation (voir Fig. 2.9(a)),
- le déplacement de l'objet d'intérêt dans la scène provoque l'apparition dans ce volume de vision d'objets constituant la scène (voir Fig. 2.9(b)),
- le déplacement de la caméra dans la scène provoque l'apparition d'objets constituant la scène (exemples : *murs, reliefs, objets quelconques*) (voir Fig. 2.9(c)).

a b c

FIG. 2.9 – *Différentes situations d'occultation : (a) occultation par un objet en mouvement (b) occultation provoquée par un déplacement de l'objet (c) occultation provoquée par un déplacement de la caméra*

Pour résoudre ces problèmes, une approche à base de volumes englobants est proposée.

2.5.2.2 *Traitement des situations d'occultation*

Une solution pour prévoir ce genre de situations est l'utilisation de volumes englobants. Plusieurs avantages peuvent être mis en avant : l'utilisation de volumes englobants permet de simplifier les calculs de parcours d'une base de données 3D, en s'appuyant sur une homogénéité de la représentation des objets, et en même temps une simplification des calculs inhérente à la représentation des volumes. En effet, leur géométrie simple (le plus souvent sous forme de parallélépipèdes rectangles, sphères ou bien "*slabs*" [1]), permet d'effectuer rapidement des opérations comme les tests d'intersection.
Dans la version que nous avons développée, seul un volume de type parallélépipédique a été utilisé. Cependant on peut distinguer deux natures différentes de volume (voir Fig. 2.10) :

- un type de volume générique pour tous les objets, constitué de plans parallèles aux axes du repère absolu de la scène,
- un volume représentant le volume de vision de la caméra englobant à la fois le plan image de la caméra ainsi que l'objet d'intérêt. C'est principalement à l'aide de ce volume que l'on va travailler pour prévoir d'éventuelles occultations.

C'est à partir de ces volumes englobants que l'on peut essayer de prévoir d'éventuelles occultations.

1. ensembles de couples de plans parallèles délimitant l'objet

a b

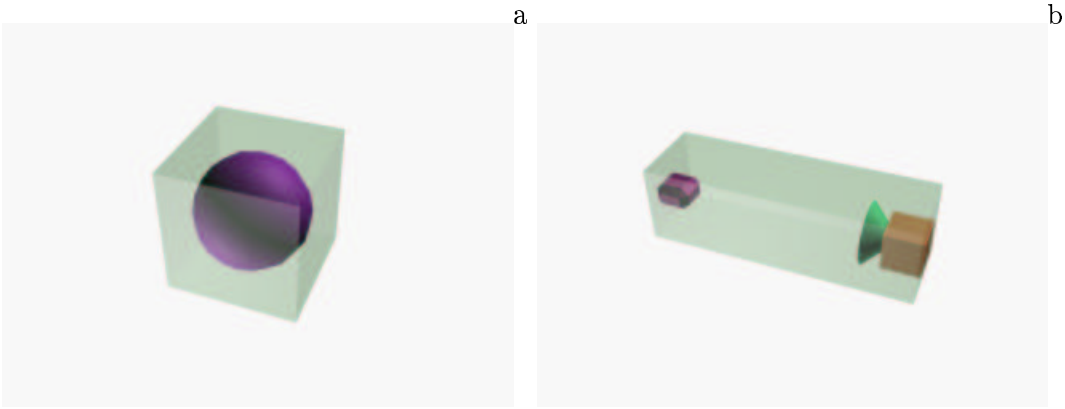


FIG. 2.10 – *Différents types de volumes englobants : (a) associé à un objet (b) associé à la caméra et à l'objet d'intérêt*

2.5.2.3 Prédiction des occultations

Le but est ici de pouvoir déterminer à l'avance (anticipation) si une occultation va survenir. Différentes solutions ont été proposées. Elles correspondent aux trois grandes catégories de risque d'occultation mises en évidence dans le paragraphe précédent.

Occultation par un objet en mouvement (voir Fig 2.9 (a)) Dans ce cas précis, on n'attend pas que l'objet pouvant provoquer une occultation ait pénétré le volume englobant associé à la caméra et l'objet d'intérêt, car il est alors trop tard. Il faut récupérer le vecteur vitesse de cet objet et créer un rayon partant de la position de l'objet et possédant comme vecteur directeur ce vecteur vitesse. L'intersection ou non de ce rayon avec le volume englobant permet alors de dire (dans la mesure où sa direction est constante) s'il va occulter l'objet d'intérêt. Précisons que ce test d'intersection est réalisé à chaque itération, ainsi un objet ayant une trajectoire changeante ne conditionnera pas les mouvements de la caméra sur une longue période.

Occultation liée à un mouvement de l'objet d'intérêt (voir Fig 2.9 (b)) La technique proposée pour résoudre ce problème repose sur l'utilisation du volume englobant associé à la caméra et l'objet d'intérêt. Lorsqu' est détecté un changement de direction de l'objet, on construit un nouveau volume englobant contenant à la fois la caméra ainsi que sa position d'origine et sa position extrapolée (à l'aide de la vitesse de l'objet et d'un paramètre temporel, selon un nombre d'itérations donné).

On regarde alors si ce volume est intersecté par un autre volume associé aux différents objets de la scène. Si tel est le cas, un risque d'occultation est mis en évidence. Cette technique permet aussi de prévoir une éventuelle collision de la caméra avec un obstacle, la caméra étant elle-même contenue dans le volume englobant, toute intersection avec un autre volume amenant une réaction appropriée de celle-ci.

Il est possible de déterminer le temps avant cette hypothétique occultation en procédant par recherche dichotomique : on teste les intersections avec des volumes générés avec un Δt différent. Par exemple, si le volume généré grâce à la position de l'objet à $t+5$ n'intersecte aucun objet de la scène, alors qu'à $t+10$ il intersecte un objet, on peut prévoir qu'il y

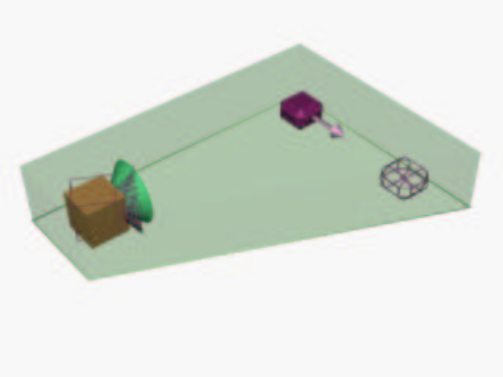


FIG. 2.11 – *Volume augmenté à cause d'un déplacement de l'objet*

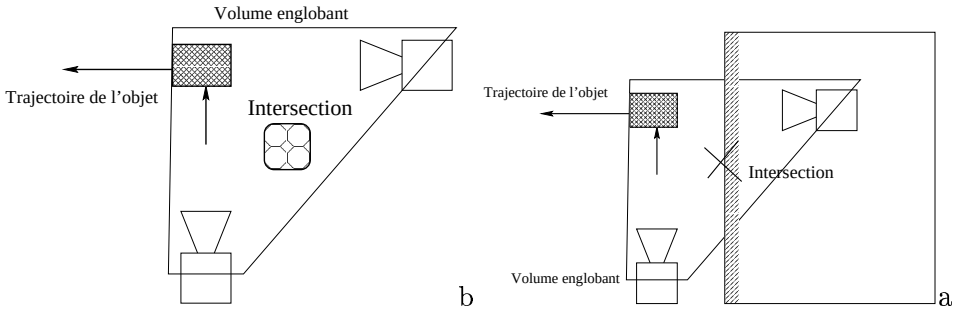


FIG. 2.12 – *Différents types d'intesections avec le volumes englobants : (a) évitement d'occultations (b) évitement d'obstacles*

aura occultation six à dix unités de temps plus tard. Une telle connaissance peut nous permettre de faire un réglage du gain, c'est à dire déterminer avec quelle vitesse doit se déplacer la caméra (ce point sera abordé en détail plus loin). Cette prédiction n'est bien entendu valide que si l'objet ne change pas de direction. L'algorithme 1 est un exemple d'algorithme permettant cette recherche.

Algorithm 1 Calcul du temps avant une occultation

```
1: Entier resultat ← TAILLEMAXIMUM {une puissance de deux}
2: Entier iteration ← 0 {variable de parcours}
3: Entier precision ← PROFONDEURMAX {nombre d'itérations indiquant la profondeur du parcours}
4: updateVolume(resultat)  {Mise à jour du volume englobant pour n iterations}
5: if Intersection() then
6:    updateVolume(resultat/2)
7:    while iteration ≤ precision && resultat est pair do
8:       if Intersection() then
9:          resultat ← resultat − resultat/2
10:      else
11:         resultat ← resultat + resultat/2
12:      end if
13:      updateVolume(resultat)
14:   end while
15:   Return resultat
16: else
17:   Return 0
18: end if
```

On note que cet algorithme est une approche possible, mais il existe d'autres approches envisageables: une fois qu'un objet est détecté dans le volume englobant maximal, un calcul angulaire peut être effectué pour déterminer sa proximité à la caméra. Une autre technique implémentée consiste à remplacer le calcul d'intersection par un lancer de rayon. Dans ce cas, on connaît automatiquement, en fonction du rayon intersectant, la position de l'objet occultant dans le volume englobant.

Occultation liée à un mouvement de la caméra (voir Fig 2.9(c)) Ce dernier cas qui s'apparente au cas précédent est effectivement de même nature, mais cette fois c'est la caméra qui par un de ses mouvements pose problème. Lorsque la caméra change de trajectoire, son propre mouvement peut engendrer une occultation. Il convient alors, de manière identique à celle développée précédemment, de créer un volume à partir de la position de la caméra, d'une position souhaitée de celle-ci, ainsi que l'objet d'intérêt. Le test d'intersection sur ce volume permet de prédire ou non une éventuelle occultation, et par des méthodes similaires à celles décrites précédemment de déterminer le temps avant cette occultation pour fixer le gain associé aux mouvements de la caméra.

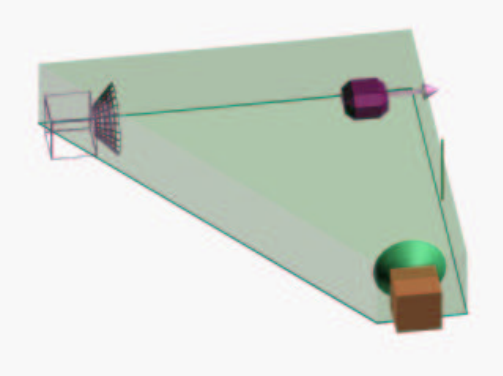


FIG. 2.13 – *Volume augmenté à cause d'un déplacement de la caméra*

2.5.2.4 Evitement des occultations via la tâche secondaire

Dans le cadre où l'on se place, la tâche principale consiste à conserver un objet d'intérêt à une certaine position dans l'image. C'est grâce à la tâche secondaire que l'on peut gérer les réactions de la caméra lorsqu'une éventuelle occultation est détectée. Il est clair que le type de réaction que l'on spécifie à la caméra dépend fortement du type d'occultations que l'on détecte. Il faut être cependant "vigilant" car –sans jeux de mots– une occultation peut en cacher une autre. En effet dans un environnement encombré on peut passer rapidement d'une configuration occultante à une autre. Il faut donc pouvoir basculer d'une politique de mouvements à une autre, et ce en jouant notamment sur la pondération des tâches secondaires qui coexistent (cf. Résultats 2.5.3.4).

Deux politiques distinctes se démarquent en ce qui concerne le mouvement de la caméra en cas d'occultation :

- une première possibilité consiste à utiliser les informations 3D disponibles (mouvements d'un objet occultant, déplacement de l'objet d'intérêt) pour générer un mouvement adéquat,
- la seconde possibilité consiste à spécifier une tâche secondaire en fonction d'informations perçues dans l'image. On cherche alors à maximiser dans l'image la distance entre la projection de l'objet d'intérêt et la projection des objets occultants.

Rester en ligne avec le mouvement de l'objet d'intérêt Dans le cas où c'est un mouvement de l'objet d'intérêt qui provoque une occultation, on peut choisir la politique suivante : toujours avoir un vecteur vitesse aligné avec le vecteur vitesse de l'objet, dans ce cas l'axe optique de la caméra reste colinéaire avec le mouvement de l'objet d'intérêt. Ce genre de technique ne prévient pas toutes les occultations possibles (occultation par un objet en mouvement par exemple), mais est couramment employée dans les jeux vidéos. Lorsqu'une occultation est détectée, on choisit alors d'imprimer à la caméra un mouvement de rotation, jusqu'à ce que la caméra se retrouve dans l'axe de déplacement (voir Fig. 2.14). La caméra est alors soumise à une force de rappel à la direction de déplacement de l'objet. Ce mouvement peut, de même, être paramétré dans son amplitude en fonction du risque d'occultation encouru.

Soit ${}^{\mathbf{c}}\mathbf{P_s}(x_s,y_s,z_s)$ la position souhaitée de la caméra lorsqu'on détecte un changement de direction, c'est à dire la position de la caméra telle que son axe optique soit colinéaire avec le mouvement de l'objet, et qu'elle soit située à la bonne distance d de celui-ci. Il faut alors minimiser la distance entre ces deux points. Une tâche secondaire possible est alors :

$$\mathbf{e_2} = \alpha \begin{pmatrix} x_s & y_s & z_s & 0 & 0 & 0 \end{pmatrix}^T \tag{2.20}$$

où α est un scalaire dépendant de la distance à ${}^{\mathbf{c}}\mathbf{P_s}$, défini tel que plus la caméra est proche de cette position, moins sa vitesse est élevée.

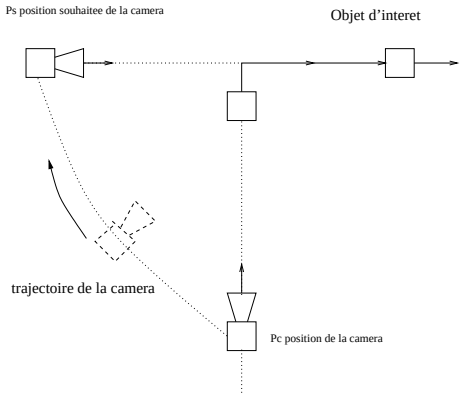


FIG. 2.14 – *Déplacement de la caméra lors d'un changement de direction de l'objet*

Certaines situations critiques nécessitent un traitement spécial. A titre d'exemple nous proposons la situation du couloir (voir Fig. 2.15), où la politique consistant à rester à une distance fixe de l'objet n'admet pas de solution au problème de l'occultation. Cette configuration force à agir de la manière suivante : il faut **diminuer la distance entre l'objet d'intérêt et la caméra**, jusqu'à ce que le risque d'occultation ou de collision disparaisse. Quand tous les risques d'occultation ont diminué de manière significative, on peut revenir à la distance initialement fixée. Cette diminution de la distance se modélise comme une tâche secondaire. On peut là encore paramétrer cette diminution en fonction du risque d'occultation.

Approche référencée vision Cette technique utilise une information référencée vision. 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 [Marchand 98] :

$$h_s = \frac{1}{2}\alpha \sum_{i=1}^{n} e^{-\beta\left(\|pr(T)-pr(O_i)\|^2\right)} \tag{2.21}$$

où $pr(x)$ désigne la projection dans l'image de l'objet x, et α et β deux scalaires. α 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 défini

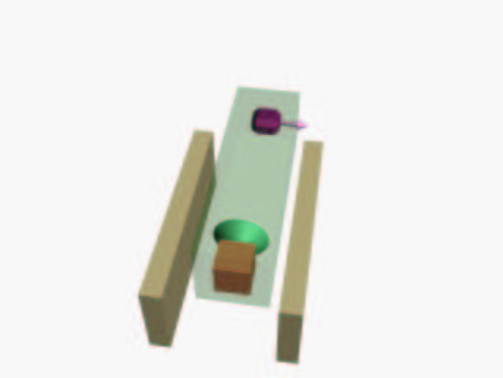


FIG. 2.15 – *Impossibilité de déplacer la caméra latéralement : il faut diminuer la distance entre la caméra et l'objet*

en 2.5.2.3). 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 $L_{\mathbf{p}}$. La difficulté revient donc à calculer $\frac{\partial h_s}{\partial \mathbf{p}}$

Considérons le cas d'un unique objet occultant et d'une cible ponctuelle, la généralisation à plusieurs objets occultant et à d'autres types de cible est immédiate. Soit $\mathbf{p} = (X,Y)$ le centre de la cible, défini comme $\mathbf{p} = pr(T)$ (en règle général, $pr(T)$ est défini comme la projection du centre de gravité). Soit l'objet $\mathcal{O}$ défini par son point $\mathbf{p}_\mathcal{O} = (X_\mathcal{O},Y_\mathcal{O})$ le plus près dans l'image de p (c'est à dire $pr(\mathcal{O}) = \mathbf{p}_\mathcal{O}$), on a :

$$h_s = \frac{1}{2}\alpha e^{-\beta\|\mathbf{p}-\mathbf{p}_\mathcal{O}\|^2} \tag{2.22}$$

et

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{r}} = \frac{\partial h_s}{\partial X}\mathbf{L}_X^T + \frac{\partial h_s}{\partial Y}\mathbf{L}_Y^T \tag{2.23}$$

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 (2.23) est une approximation de $\frac{\partial h_s}{\partial \mathbf{r}}$. En effet $\mathbf{L_p} = \begin{bmatrix}\mathbf{L}_X^T & \mathbf{L}_Y^T\end{bmatrix}$ est la matrice d'interaction associée à un point physique. Dans notre cas, comme le point considéré est défini comme le point 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 (2.23) est cependant une approximation localement correcte.

2.5.3 Résultats

Les techniques d'évitement d'obstacles et d'occultations décrites auparavant ont été implémentées dans le cadre de cette thèse. Plusieurs exemples illustrent notre approche : deux exemples basiques d'évitement d'occultation, puis la visite d'un musée virtuel mêlant des techniques dans un problème complexe plus concret. Le dernier exemple met en avant la possibilité de fusionner plusieurs contraintes.

2.5.3.1 Rester en ligne avec l'objet d'intérêt

Dans le cadre d'évitement des occultations, nous avons dans un premier temps testé la stratégie basique consistant à rester en ligne avec l'objet d'intérêt. Cette technique, inadaptée à régler tous les types d'occultation, montre cependant des résultats efficaces dans des environnements sans objets en mouvement. Dans l'exemple que nous avons traité, la tâche spécifiée pour la caméra est de suivre un personnage se déplaçant dans une ville. La distance entre les deux entités est relativement élevée (dix mètres). On s'asservit sur une sphère et un point situé au dessus de la sphère. La sphère virtuelle est située au centre de l'humanoïde, et le point virtuel est placé automatiquement au dessus de la sphère relativement à la distance souhaitée. Ce type d'asservissement permet entre autres la poursuite d'un objet, et laisse disponibles les degrés de translation latéraux, nécessaires à la réalisation de la tâche secondaire telle que nous l'avons définie. Les images présentées dans la figure 2.16 montrent la vue de la caméra ainsi que la vue du dessus correspondante. On ne considère la tâche secondaire que lorsqu'une occultation est détectée ; ainsi dans cet exemple aucune anticipation d'occultation n'est réalisée.

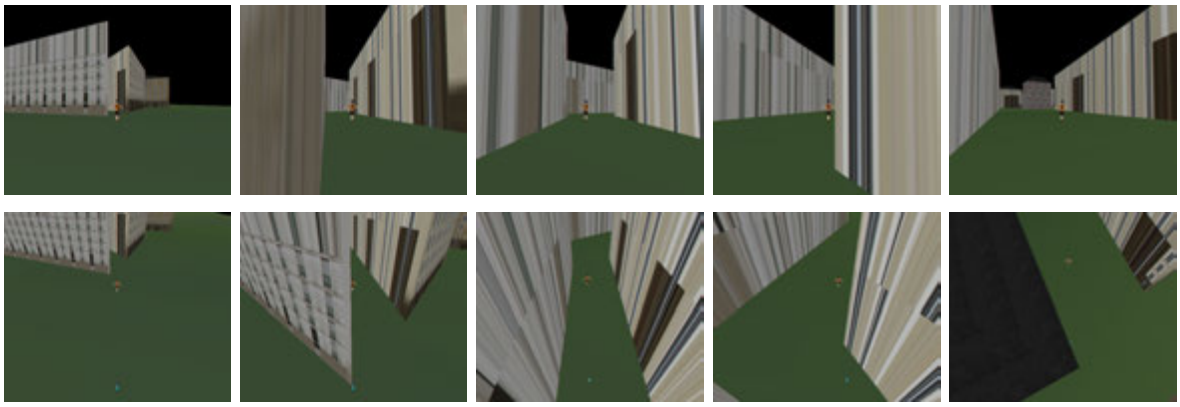


FIG. 2.16 – *Evitement d'occultations en restant en ligne avec l'objet d'intérêt*

2.5.3.2 Exemple du pilier

L'expérience du pilier consiste à suivre un objet qui tourne autour d'un pilier (voir images 2.17(a)). La caméra, qui doit rester à une distance constante de l'objet, doit se déplacer de manière à ce que le pilier ne se mette jamais entre la caméra et l'objet. L'objet d'intérêt dispose d'une trajectoire prédéfinie, alors que les commandes de la caméra sont générées automatiquement. La technique utilisée est celle des volumes englobants pour

prévenir les occultations, et celle de maximisation de la distance dans l'image pour calculer le mouvement de la caméra permettant d'éviter les occultations. On constate lors de cette expérience que la caméra tourne autour du pilier, en conservant bien l'objet à sa position dans l'image. Les courbes (voir Fig. 2.17(b,c,d)) montrent une évolution marquée par plusieurs phases, qui témoignent en fait des changements de direction de l'objet. On voit la trajectoire dans l'espace de la caméra dans la figure 2.17(e).

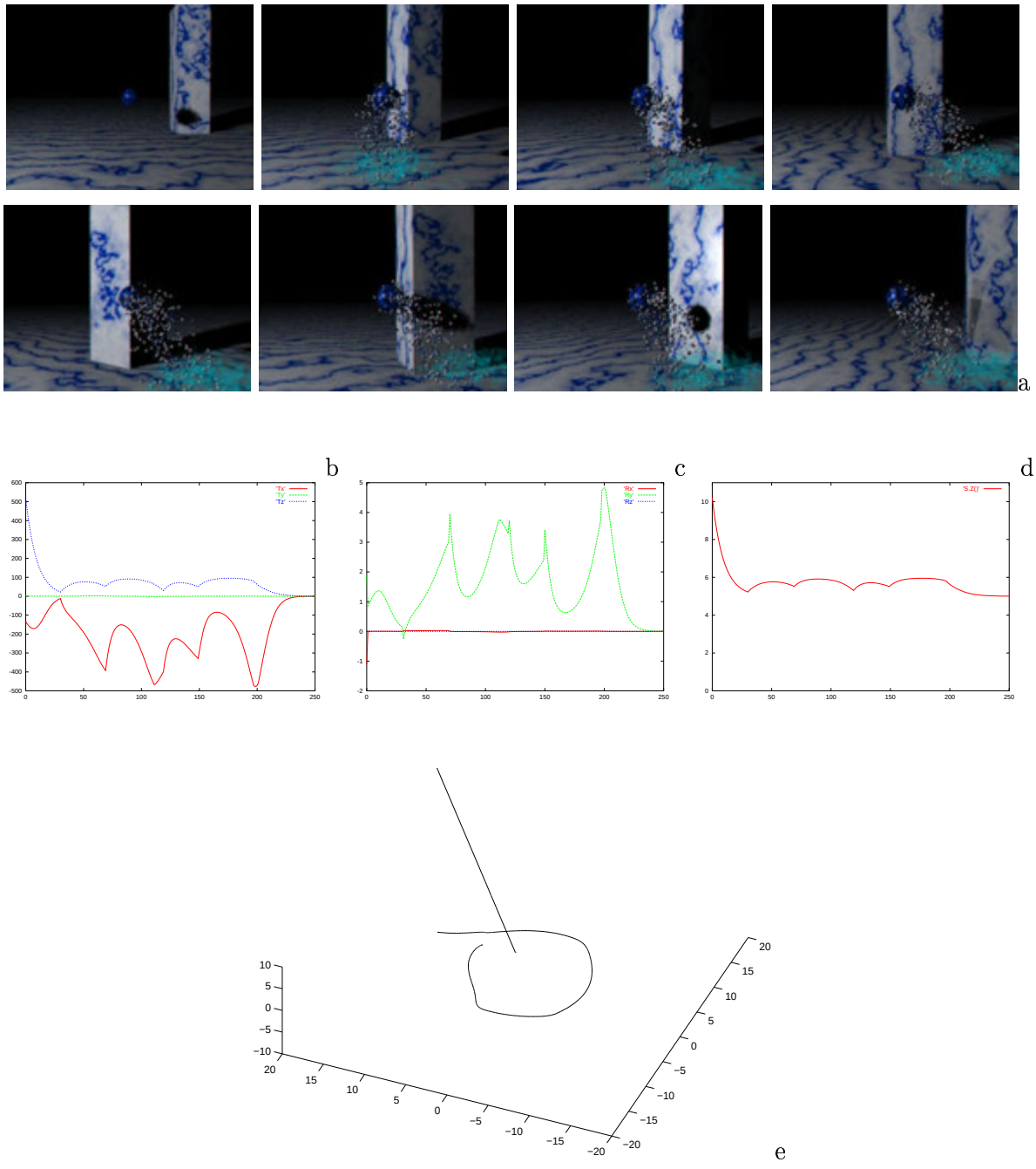


FIG. 2.17 – *(a) extrait de l'animation associée au problème du pilier (b) translations de la caméra (c) rotations de la caméra (d) distance à l'objet d'intérêt (e) trajectoire 3D de la caméra*

2.5.3.3 Evitement des occultations et des obstacles : problème du musée

Dans cet 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 [Espiau 92] et à l'annexe B pour la dérivation complète de la matrice d'interaction associée à la sphère.

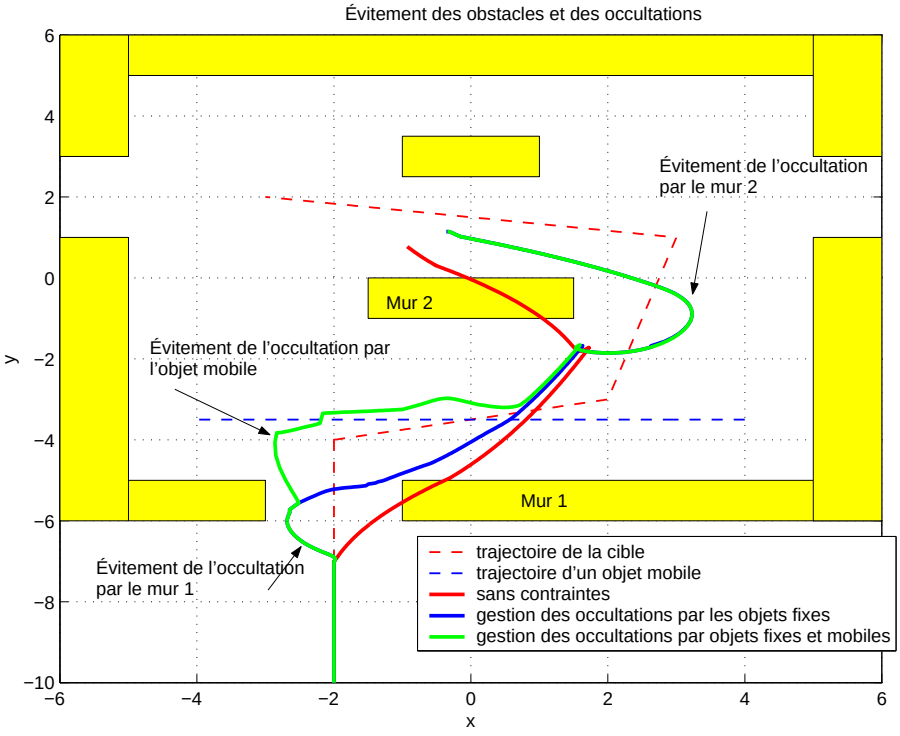


FIG. 2.18 – *Environnement de type musée : trajectoire de la caméra pour différentes stratégies*

La figure 2.18 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 2.20 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 2.21 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 au triplet (caméra, position courante et prédite de la cible)) correspondent aux volumes englobants servant à la prédiction des occultations et à l'évitement d'obstacles.

2.5.3.4 Progression dans un couloir : fusion de contraintes

Dans cette expérience nous considérons une tâche similaire mais la cible se déplace dans un couloir étroit et tourne à droite (voir la figure 2.19). 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.

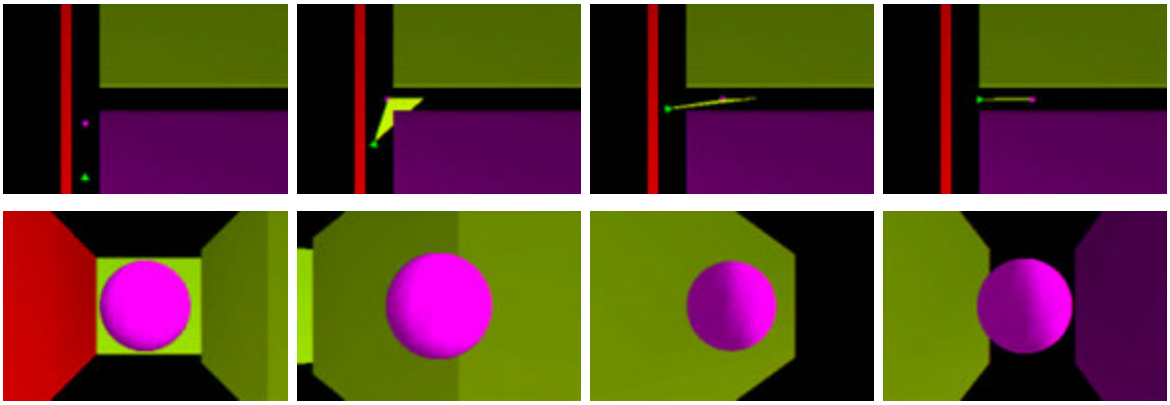


FIG. 2.19 – *Progression dans un couloir : vue de dessus et vue de la caméra*

FIG. 2.20 – *Traversée d'un musée. Le processus d'évitement des occultations et des obstacles n'est pas considéré. La trajectoire résultante comprend de nombreuses pertes de la cible et des collisions avec l'environnement*

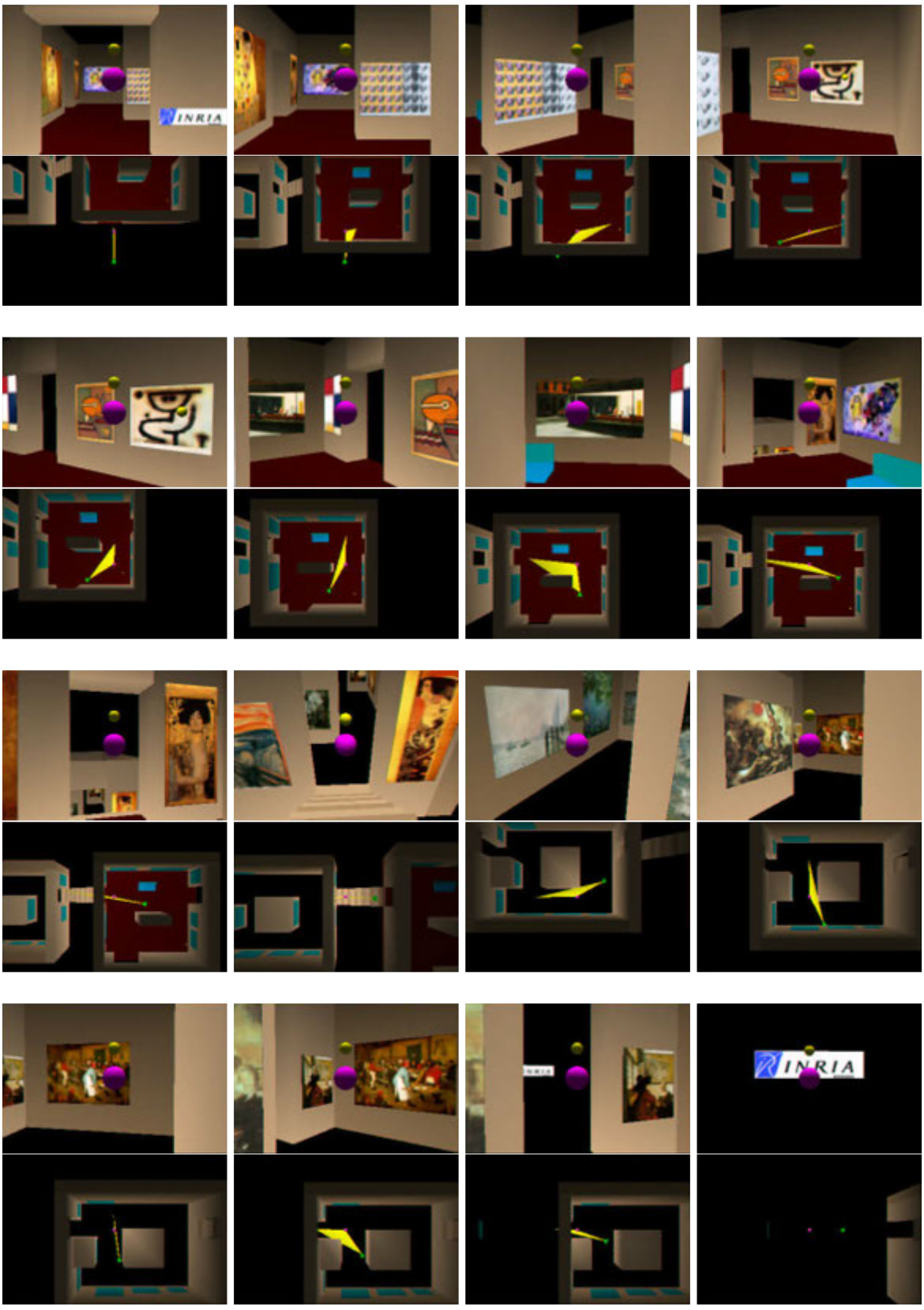


FIG. 2.21 – *Traversée d'un musée : vue de la caméra et vue du dessus. Dans cette animation le processus d'évitement des obstacles et des occultations est considéré. Les volumes englobants apparaissent en jaune*

2.6 Problème de la “photographie”

Optimiser l'éclairage est un problème majeur et difficile pour un cinéaste. Ce problème revient à considérer le positionnement des sources de lumière relativement à la qualité de l'image filmée par une caméra. La nature du problème implique deux approches possibles : soit la caméra est déplacée, soit la lumière. La première difficulté consiste à déterminer les bons critères d'optimalité relatifs à notre problème. Deux critères sont donc proposés pour parvenir à cet objectif. Le premier repose sur une maximisation de la quantité de lumière ré-émise 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 est 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.

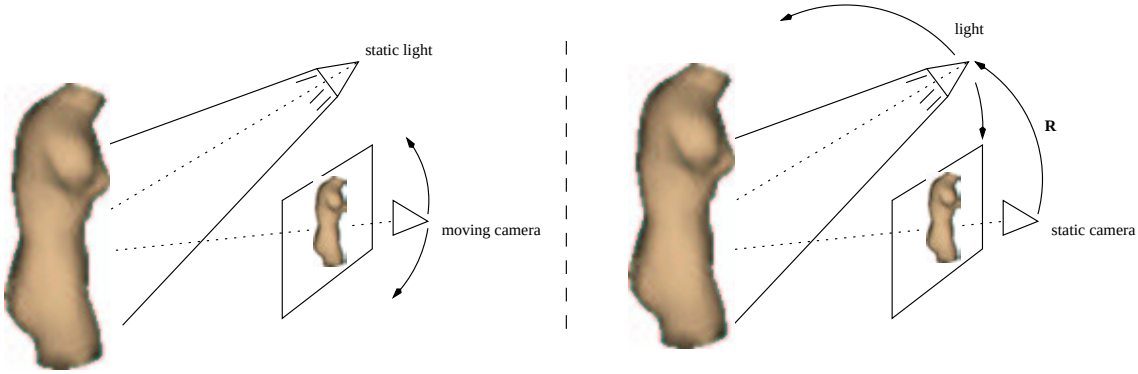


FIG. 2.22 – *Contrôle des conditions d'éclairement. (a) source lumineuse statique, caméra mobile (b) caméra statique, source lumineuse mobile*

2.6.1 Modélisation

Le premier critère considéré consiste à maximiser la quantité de lumière ré-émise 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 :

$$\begin{aligned} \frac{\partial h_s}{\partial \mathbf{r}} &= \frac{1}{n} \sum_X \sum_Y \left(\frac{\partial h_s}{\partial X} \frac{\partial X}{\partial \mathbf{r}} + \frac{\partial h_s}{\partial Y} \frac{\partial Y}{\partial \mathbf{r}} \right) \\ &= \frac{1}{n} \sum_X \sum_Y \left(\nabla_X \mathbf{L}_X^T + \nabla_Y \mathbf{L}_Y^T \right) \end{aligned} \tag{2.24}$$

où $\nabla I_X = \frac{\partial I}{\partial X}$ et $\nabla I_Y = \frac{\partial I}{\partial Y}$ représentent le gradient spatial d'intensité. Là encore (2.24) est une approximation localement correcte de $\frac{\partial h_s}{\partial \mathbf{r}}$ (voir la remarque du paragraphe 2.5.2.4).

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 :

$$h_s = \frac{1}{n} \sum_X \sum_Y \left[\nabla I_X^2 + \nabla I_Y^2 \right] \tag{2.25}$$

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

$$\frac{\partial h_s}{\partial \mathbf{r}} = \frac{1}{n} \sum_X \sum_Y \left(\frac{\partial h_s}{\partial X} \mathbf{L}_X^T + \frac{\partial h_s}{\partial Y} \mathbf{L}_Y^T \right) \tag{2.26}$$

Après quelques réécritures, on obtient finalement :

$$\frac{\partial h_s}{\partial \mathbf{r}} = \frac{2}{n} \sum_X \sum_Y \left[\left(\frac{\partial^2 I}{\partial X^2} \nabla I_X + \frac{\partial^2 I}{\partial Y \partial X} \nabla I_Y \right) \mathbf{L}_X^T + \tag{2.27}$$

$$\left(\frac{\partial^2 I}{\partial X \partial Y} \nabla I_X + \frac{\partial^2 I}{\partial Y^2} \nabla I_Y \right) \mathbf{L}_Y^T \Bigg] \tag{2.28}$$

2.6.2 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 :

$$\mathbf{e} = \mathbf{W}^+ \underbrace{\mathbf{L}^+(\mathbf{P} - \mathbf{P_d})}_{\substack{\text{tâche principale :} \\ \text{focalisation}}} + (\mathbf{I} - \mathbf{W}^+\mathbf{W}) \underbrace{\begin{pmatrix} \mathbf{R} & -\mathbf{R}\mathbf{skew}(-\mathbf{R^T T}) \\ 0 & \mathbf{R} \end{pmatrix} \frac{\partial h_s}{\mathbf{r}}}_{\substack{\text{tâche secondaire définie} \\ \text{par rapport à} \\ \text{la seconde caméra}}} \tag{2.29}$$

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

2.6.3 Résultats

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 images de la figure 2.23(a). L'intensité moyenne évolue très doucement (voir Fig. 2.23(b)). On peut constater (voir Fig. 2.23(d)) que la caméra se place entre la source lumineuse et l'objet d'intérêt, comme cela est prévisible par la théorie.

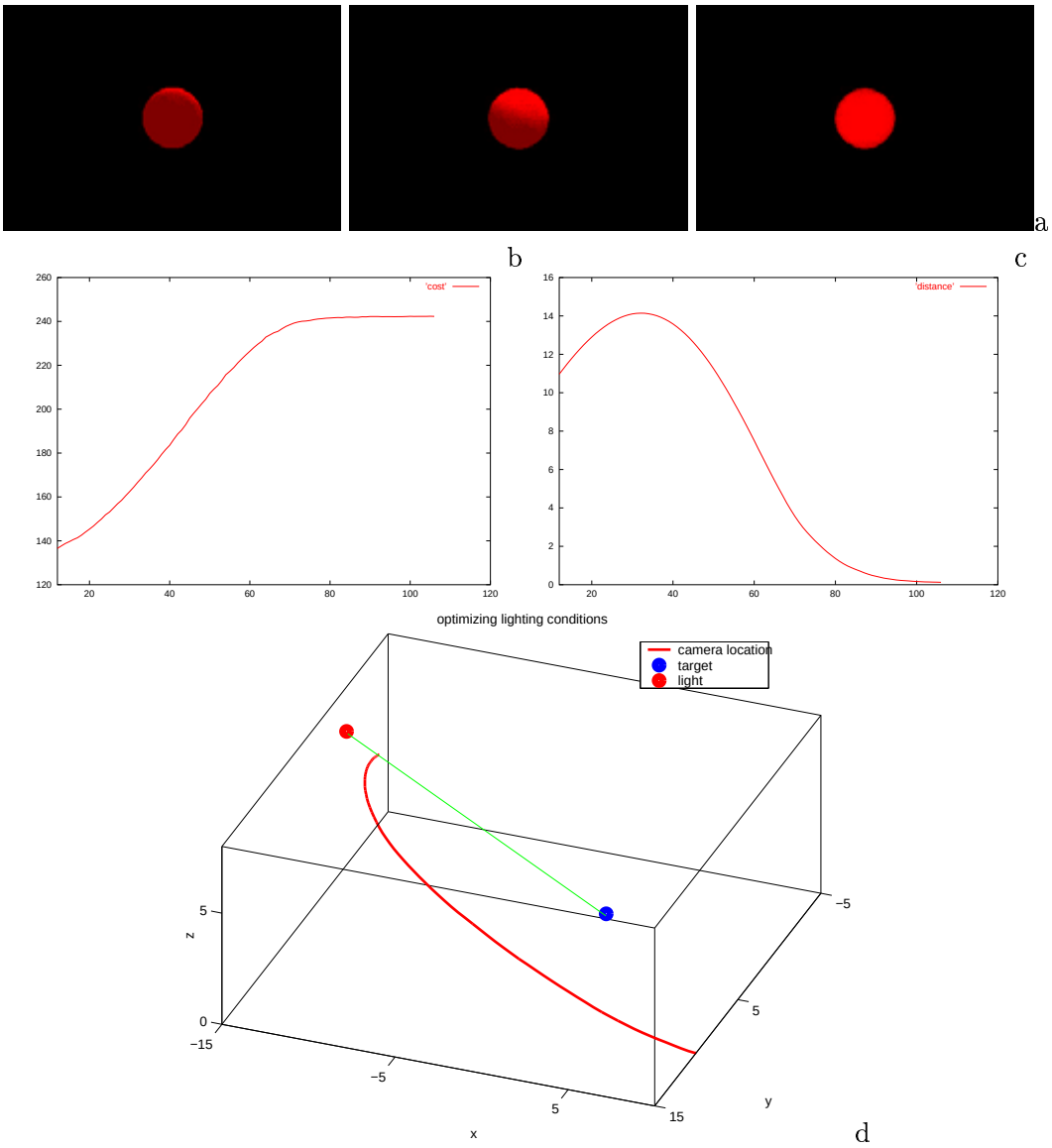


FIG. 2.23 – *Positionnement par rapport à une sphère selon des contraintes d'éclairement (a) vue de la caméra (l'intensité augmente) (b) intensité moyenne dans l'image (c) distance à l'axe objet–source lumineuse (e) positions de la caméra au cours du temps*

D'autres expériences considérant des objets plus complexes ont été réalisées en considé-

rant, 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 2.25). La lumière doit alors se déplacer afin de maintenir un éclairage correct de la statue. Les résultats présentés (voir la figure 2.24) montrent les avantages de notre approche. Les deux fonctions de qualité ont été considérées (la luminance est considérée sur la figure 2.24(a) et le contraste est considéré sur la figure 2.24(b)).

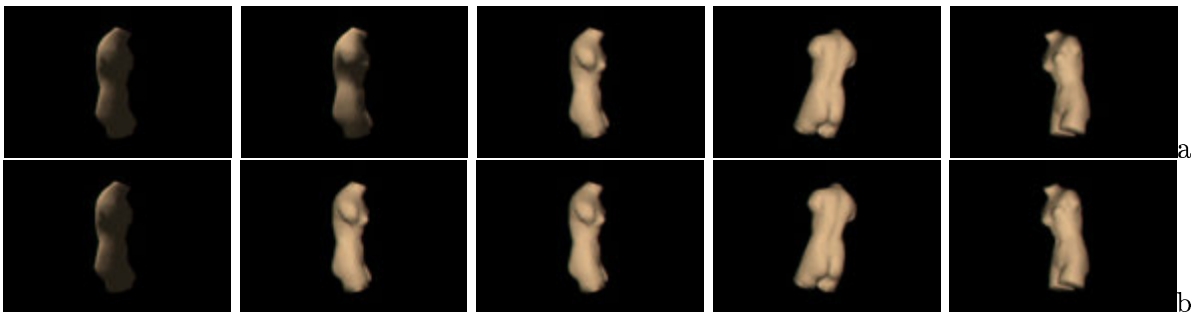


FIG. 2.24 – *Illumination de la Venus de Milo (a) maximisation de la luminance (b) maximisation du contraste. Dans les trois première colonnes, la caméra reste statique puis elle effectue un mouvement de rotation autour de la statue*

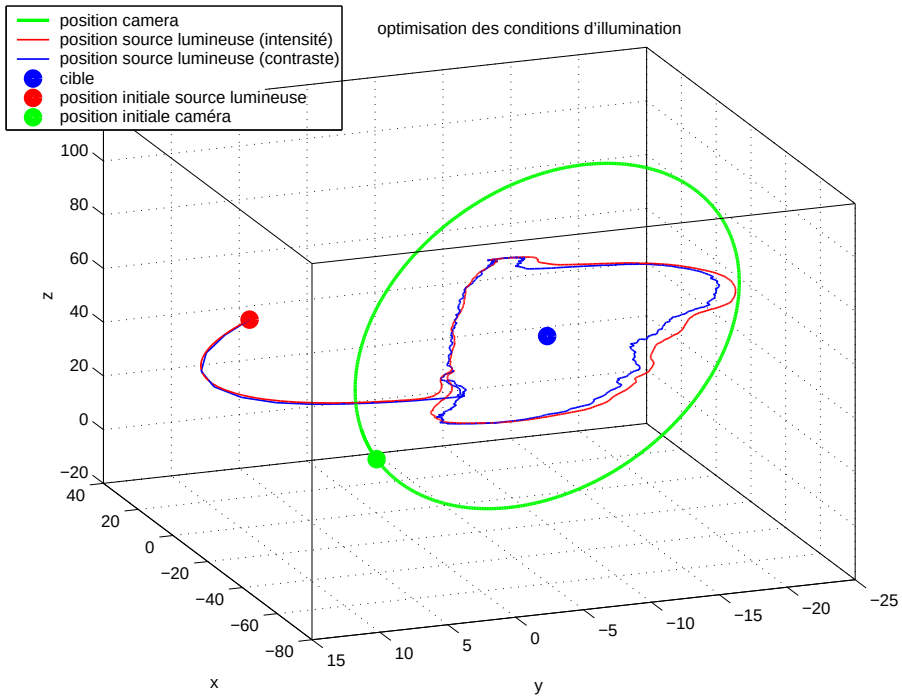


FIG. 2.25 – *Illumination de la Venus de Milo : trajectoires de la source lumineuse et de la caméra*

2.7 Conclusion

Dans le cadre de ce chapitre, nous nous sommes intéressés au contrôle d'une caméra dans un environnement virtuel. L'utilisation de l'asservissement visuel nous a permis de modéliser un système d'animation utilisant comme modalités de contrôle des **tâches visuelles**. Ces tâches permettent de positionner simplement la caméra relativement à des contraintes exprimées dans l'espace de l'image perçue. La réalisation de ces tâches s'effectue en temps réel, autorisant l'utilisation de cette approche dans des environnements dynamiques. De plus, le formalisme étudié permet d'introduire des contraintes dans les mouvements obtenus ; ces contraintes peuvent servir à résoudre des problèmes non triviaux dans le cadre de l'animation d'une caméra virtuelle. Grâce à l'utilisation du principe de redondance, la **tâche secondaire** (qui reflète les contraintes sur le système) n'a aucun effet sur la 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 de notre approche, 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 ou d'obstacles ou encore le positionnement par rapport à la face éclairée d'un objet (afin d'assurer une bonne "photo"). La grande flexibilité des commandes que l'on peut envoyer à la caméra, ainsi que les adaptations possibles vis-à-vis de son environnement, font de cette technique un outil puissant pour des applications où il n'y a pas place pour un animateur. Nous verrons comment dans la deuxième partie de ce document nous pouvons intégrer cette technique d'animation dans un système de cinématographie virtuelle.

Les résultats obtenus valident l'intérêt associé à l'utilisation de cette technique d'animation référencée vision pour une entité simple comme la caméra. Mais il est possible aussi de contrôler des systèmes plus complexes (avec plus de degrés de liberté) au travers du même principe. Dans le chapitre suivant, nous allons étudier comment, par rapport aux techniques déjà existantes, l'utilisation de notre technique d'animation référencée vision permet de contrôler de manière efficace la cinématique associée à l'attention visuelle d'un humanoïde de synthèse.

Chapitre 3

Contrôle d'un humanoïde virtuel

L'animation d'humanoïde se trouve au carrefour de plusieurs domaines applicatifs (mondes virtuels, films de synthèse, jeux vidéos, recherche biomécanique,etc.), elle présente les aspects les plus difficiles de l'animation d'objets graphiques en général. Les exigences que l'on peut avoir vis-à-vis de l'animation d'un humanoïde se situent en effet sur différents niveaux. La complexité des modèles géométriques (nombreux degrés de liberté pour un humanoïde) impose un choix pertinent des techniques d'animation pour assurer la contrainte de vitesse de calcul. Le choix de modèles "simplifiés" de squelette peut aussi se traduire par une perte de réalisme visuel ou physique. Ce réalisme dépend de plusieurs aspects : cohérence des mouvements avec les modèles physiques et biomécaniques, et cohérence avec des modèles comportementaux. Le premier de ces facteurs découle directement de la technique d'animation. Quand celle-ci ne prend pas en compte les différents aspects physiques régissant nos déplacements (ainsi que les contraintes biomécaniques), il faut alors que dans son formalisme ces différents aspects soient pris en compte. Le second facteur place le réalisme au niveau comportemental, c'est à dire sur une cohérence globale des actions de l'humanoïde. L'enchaînement d'actions est motivé par un but, une base de connaissance, une personnalité, des paramètres internes, etc. La cohérence globale associée à un humanoïde correspond à la crédibilité et à la pertinence du choix des actions à entreprendre. Ces différents niveaux de réalisme s'obtiennent en travaillant sur les différents modules qui décomposent l'animation d'humanoïde. Une telle décomposition (voir la figure 3.1) a été donnée dans [Terzopoulos 98].

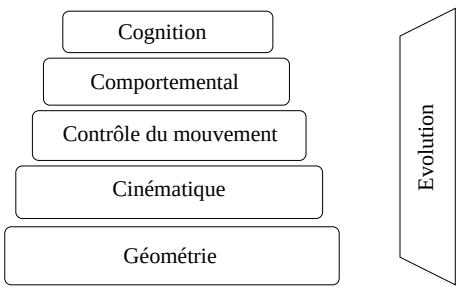


FIG. 3.1 – *Pyramide de l'animation d'humanoïdes de synthèse*

Les travaux les plus récents en animation cherchent cependant à unifier ces différents ni-

veaux dans des approches mêlant à la fois les aspects de bas niveau (théorie du contrôle, animation), et les travaux ayant trait au comportement et à la cognition [Thalmann 90, Amaya 96,Badler 97b,Funge 98,Badler 99a,Kuffner 99], car les liens entre les différents niveaux décrits sur la figure 3.1 sont difficiles à mettre en place, bien que les interactions entre ces niveaux soient importantes.

Les chapitres précédents étaient dédiés à l'animation d'une caméra dans un environnement virtuel. Dans le formalisme retenu, la commande envoyée à la caméra dépend de ce que "perçoit" celle-ci. L'analogie avec un humanoïde virtuel est directe ; la plupart des actions entreprises par celui-ci sur son environnement extérieur dépend de la perception qu'il en a. L'idée de ce chapitre est d'étudier la manière dont on peut appliquer le formalisme précédemment décrit au problème du contrôle d'un humanoïde virtuel, en modélisant la cinématique associée à son attention visuelle dans un contexte de perception active.
Nous présentons dans cette partie les travaux réalisés dans le cadre du contrôle d'un humanoïde de synthèse. La première partie de ce chapitre est consacrée à une étude des exigences de l'animation d'humanoïde. A partir des résultats de cette étude, nous justifions dans un deuxième temps l'emploi de l'asservissement visuel dans le cadre de la simulation de la perception visuelle d'un humanoïde, et nous développons le formalisme sous-jacent ainsi que les résultats obtenus. La suite du chapitre est alors dédiée à la résolution au travers du formalisme retenu de quelques uns des grands problèmes associés à l'animation d'un humanoïde de synthèse : gestion des butées articulaires, transition entre les différentes tâches, couplage avec un autre système d'animation (gestion de la locomotion) et contraintes inhérentes au réalisme des animations.

3.1 État de l'art sur les techniques d'animation d'humanoïde

Dans cette partie, un rapide état de l'art des différentes techniques d'animation d'un humanoïde est proposé. Au travers d'une mise en relation avec la problématique associée aux modèles comportementaux gérant la perception visuelle, un cahier des charges est défini regroupant les aspects nécessaires à un système d'animation.

3.1.1 Techniques d'animation

Le problème principal dans le cadre de l'animation d'un humanoïde revient à générer un mouvement réaliste, au travers de la gestion de trois paramètres : position, orientation et déformation de la géométrie. Les techniques utilisées pour l'animation d'un humanoïde sont donc issues des techniques d'animation générale de solides rigides ou déformables [Arnaldi 89,Arnaldi 91]. Depuis l'utilisation de l'animation image par image, de nombreuses techniques d'animation ont émergé, pouvant se regrouper en trois catégories [Multon 98]:

- les méthodes dites *cinématiques*, qui cherchent à reproduire un certain nombre d'effets sans s'intéresser aux causes, à partir d'une description de ce que doit être le mouvement,
- la génération du mouvement à partir de mouvements, traditionnellement acquis grâce à des systèmes de *motion capture* ; cette approche étant sensiblement semblable à la première, mais adresse de nouveaux problèmes,
- les méthodes dites *dynamiques*, dont les fondements sont les causes du mouvement.

Nous présenterons dans un premier temps ces trois catégories, puis nous ferons une synthèse montrant qu'il est nécessaire d'avoir, dans le cadre de l'animation d'un humanoïde, une architecture pouvant regrouper ces différentes techniques.

3.1.1.1 Méthodes cinématiques

Le problème revient, dans ce cas, à calculer la trajectoire angulaire de chaque articulation du squelette de l'humanoïde. On suppose qu'on peut les représenter par des fonctions continues par morceau $\mathbf{q_i}(t)$, où i représente la i^{me} articulation. Les portions de fonction sont représentées par des valeurs extrêmes, chaque valeur intermédiaire pouvant être calculée par interpolation. La qualité de l'animation dépend alors du choix des positions clé et des fonctions d'interpolation. Le calcul direct des torseurs cinématiques associés aux effecteurs (mains, pieds, tête) $\mathbf{r}$ s'obtient alors comme une fonction des valeurs articulaires $\mathbf{q_i}$:

$$\mathbf{r} = f(\mathbf{q_i}), \;\; i = 1,...,n \tag{3.1}$$

On parle alors de cinématique directe. Dans le but de contrôler la structure, on peut chercher à résoudre le problème inverse [Tolani 96], c'est à dire retrouver les paramètres des valeurs angulaires de chaque articulation à partir des positions finales des effecteurs, en résolvant l'équation :

$$\mathbf{q} = f^{-1}(\mathbf{r}) \tag{3.2}$$

Ce problème est particulièrement étudié en robotique dans le cadre des rigides articulés (bras mécaniques, bipèdes). Considérons un rigide articulé comportant n degrés de liberté $\mathbf{q_i}$, et si m est la dimension de $\mathbf{r}$, alors le Jacobien $\mathbf{Jac(q)}$ est une matrice de dimensions (n,m) définissant la relation entre la variation temporelle de $\mathbf{r}$ et la variation temporelle de $\mathbf{q}$ [Baerlocher 98] :

$$\dot{\mathbf{r}} = \mathbf{Jac(q)}.\dot{\mathbf{q}} \tag{3.3}$$

Le problème revient à pouvoir inverser la matrice $\mathbf{Jac(q)}$. La qualité des mouvements obtenus dépend alors des spécifications des positions finales des effecteurs, et cette méthode s'avère coûteuse en temps de calcul dans le cas où il faut réévaluer $\mathbf{Jac(q)}^+$ à chaque itération. Cependant, des méthodes de calcul rapides existent : inversion symbolique (mais le temps d'évaluation augmente de manière dramatique plus le nombre de degrés de liberté augmente), inversion numérique [Watt 92], inversion itérative [BenIsrael 74] (construction récursive), estimation par la transposée [Welman 93] ou bien encore utilisation de méthodes type descente de gradients.

3.1.1.2 Méthodes dynamiques

Dans cette approche, on considère que les causes du mouvement sont de nature physique. Les lois de la physique (donc de la mécanique) se substituent alors aux connaissances a priori sur le mouvement. Deux formalismes mettent en place les équations du mouvement :

- le formalisme de Newton-Euler, décrivant les relations entre les accélérations en translation/rotation et les forces appliquées au système et leurs masses,

– le formalisme de Lagrange, s'appuyant sur le principe des travaux virtuels.

L'avantage des méthodes dynamiques est de pouvoir générer des mouvements très réalistes. Cependant, les équations différentielles décrivant les lois mécaniques sont généralement fortement non-linéaires et présentent des termes de couplage rendant difficile leur résolution [Zordan 99]. De plus, créer ces systèmes peut s'avérer difficile, voire pratiquement impossible pour des structures compliquées. La gestion des forces à exercer s'avère impossible de manière interactive, et requiert le plus souvent l'utilisation de contrôleurs [Cozot 96,Nougaret 98,Faloutsos 01], qui spécifient de façon indirecte les forces et les couples à appliquer sur chaque articulation pour obtenir le mouvement désiré. Cette méthode est donc difficile à employer pour un contrôle au niveau tâche de l'humanoïde. Notons cependant que ces méthodes ont été utilisées avec succès dans la modélisation de mouvements humains tels que la marche, la course ou des mouvements athlétiques [Brogan 98].

3.1.1.3 Méthodes à base de mouvements capturés

Ces techniques sont apparues dans le courant des années 90 et consistent à enregistrer le mouvement d'une partie ou du corps d'un humain à l'aide d'un système de capture de mouvement (ou *motion capture*) pour l'intégrer à un modèle virtuel.

Si l'on cherche à obtenir des mouvements divers et variés, adaptables à l'environnement, on est limité par le fait qu'on ne peut pas se permettre de créer un panel exhaustif de tous les mouvements possibles. Il faut alors générer de nouveaux mouvements à partir de la base de mouvements existante. Plusieurs méthodologies existent pour pallier à ce problème :

– Composition linéaire des mouvements, ou *motion blending*. Lors de la transition d'un mouvement à un autre, on somme ces deux mouvements avec une pondération variant lors de la transition [Witkin 95,Rose 96]. Différentes techniques émergent, notamment à base de positions clés servant les interpolations entre les mouvements, ces positions étant déterminées par résolution d'un système de contraintes spatio-temporelles.
– Modification du mouvement enregistré ou *motion warping*. Cette technique permet notamment la mise à l'échelle (*retargetting*) d'un modèle à un autre, ce problème étant du au décalage entre le modèle réel utilisé pour la capture et le modèle virtuel sur lequel on applique le mouvement [Lee 99,Popovič 99] (changement de paramètres intrinsèques tels que la longueur des membres). La possibilité de modifier un mouvement est aussi envisageable, par exemple pouvoir introduire les mouvements d'accélération ou de décélération dans un mouvement de course.

L'application de ces techniques de modification à une exécution temps réel a fait l'objet de plusieurs études. On peut citer à titre d'exemples les travaux de Choï [Choi 00] ou Ménardais [Menardais 03]. On note cependant que l'utilisation de ces techniques fait perdre le réalisme naturel issu de l'utilisation des mouvements capturés dans la mesure où les modifications sont trop importantes. Les alternatives consistent alors à contraindre les mouvements déformés à l'aide de critères visuels (par exemple pieds toujours au sol), ou de critères dynamiques. Dans le cadre de la gestion de la cinématique associée à la perception visuelle, Bindiganavale [Bindigavanale 98] propose une méthode pour capturer la gestion du regard d'un acteur, puis l'adapter à un modèle géométrique différent. Dans ce cas, l'adaptation n'est possible que pour un type de tâche visuelle : un suivi dans l'image (*tracking*), relatif à un objet connu de la scène, ce qui est une limite assez forte.

3.1.1.4 Vers un contrôle unifié et multimodal de l'humanoïde

Les techniques d'animation du squelette de l'humanoïde sont nombreuses, et le choix de la technique dépend finalement du cadre applicatif, si l'on cherche à obtenir des mouvements réalistes sur le plan physique, ainsi que cohérents sur le plan de la simulation, mais beaucoup trop coûteux en terme de temps de calcul. La tendance actuelle est à l'utilisation, dans le cadre du contrôle d'un humanoïde, de modèles mixtes, c'est à dire utilisant les différentes techniques présentées. Cette mixité des "opérateurs" d'animation s'exerce soit dans le cadre de modèles de mouvements précis, par exemple, mélange de mouvements capturés et de cinématique inverse [Tak 00] ou association de la cinématique avec la distribution des masses (cinétique inverse) [Boulic 96], soit à l'intérieur d'une même architecture visant à réunir toutes les possibilités de contrôle pour un humanoïde, avec éventuellement des contrôles distincts pour les différentes parties du corps. Ainsi dans le laboratoire de recherche sur les modèles d'humanoïde de l'université de Pennsylvanie, le modèle d'humanoïde *Jack* [Badler 99b] dispose de plusieurs modes de contrôle. Il en est de même pour les humanoïdes développés au VRLab (ancien LIG de l'EPFL) ainsi qu'au MiraLab (Suisse) [Thalmann 95]. Cette nécessité est imposée par la diversité de contrôle requis par la simulation comportementale ; il faut pouvoir disposer d'un humanoïde ayant une pluralité de contrôles, répondant à des exigences présentées dans la prochaine partie.

3.1.2 Exigences de l'animation comportementale d'humanoïde

Comme spécifié dans l'introduction de ce chapitre, les briques d'animation fondamentales doivent pouvoir s'adapter aux exigences des modèles comportementaux. Cette partie s'intéresse précisément aux exigences imposées par l'animation comportementale.

Animation orientée tâche Dans la mesure où l'on souhaite que l'humanoïde de synthèse exécute des comportements complexes dans son environnement, il faut pouvoir disposer d'une simplification des paramètres du système d'animation. L'animateur ou le programme n'a ainsi plus à spécifier des variations angulaires pour obtenir un effet particulier. Dans le cas des acteurs autonomes, ce travail permet d'obtenir des interfaces entre les couches comportementales et les couches basses de l'animation ou du contrôle pure. De plus, les valeurs échangées au niveau de ces interfaces ont la propriété d'être interprétables par un être humain (dû à la simplicité de leurs formes), ce qui rend la phase de conception et de validation plus simple et plus rapide. La nécessité de cette abstraction du contrôle a d'abord été mise en évidence dans un article de Zeltzer [Zeltzer 90]. dans les années 90. L'idée principale de cette abstraction est de pouvoir spécifier à l'humanoïde des directives sous une forme simple, issues par exemple du langage naturel [Badler 90,Badler 99b]. Le système *PAR* (*Parameterized Action Representation*) est un exemple de contrôle au niveau tâche. Ce formalisme permet de décrire une action, ainsi que toutes les informations nécessaires à sa réalisation (chemin, manière, location d'objet, force, etc.). Un analyseur syntaxique ainsi qu'un traducteur se chargent de convertir un sous-ensemble de la langue anglaise en attributs de l'action. La majeur difficulté de cette approche réside dans le passage d'une description qualitative et souvent incomplète en des paramètres nécessaires au système d'animation.

Gestion de la perception dans les modèles comportementaux Un des principaux défauts des techniques classiques d'animation comportementale est qu'elles ne définissent qu'un minimum d'interaction entre la perception (vision synthétique [Tu 94,Blumberg 97, Noser 95,Kuffner 99] ou accès à la base de données [Webber 95,Thomas 00]) et l'animation de l'humanoïde de synthèse : dans la plupart des cas, l'humanoïde "perçoit" puis agit selon une boucle perception-décision-action (ce principe sera détaillé dans la deuxième partie de ce document). L'animation résultante des avatars repose sur des modèles cinématiques simples [Kuffner 99], et ne considère que des tâches de focalisation sur des cibles ponctuelles. L'approche de Terzopoulos dans [Terzopoulos 97] cherche à lier le processus de perception avec les déplacements d'un poisson. Après analyse de l'image rendue du point de vue du poisson, le système de navigation du poisson est activé dans le but de maintenir un poisson cible dans le champ de vision. Les commandes envoyées au poisson sont directement associées à la tâche de perception (fixation sur le poisson cible). Le principal défaut de cette approche est de ne considérer qu'une tâche de fixation simple, et que le problème de contrôle du poisson est nettement simplifié car c'est un animal qui se déplace indifféremment dans les trois dimensions (bien que soumis à des contraintes non-holonomes). Cependant l'approche de Terzopoulos est la première à lier de manière forte le comportement et la cinématique associée à la perception. Ce type de perception est une perception active, ou "autonome". Selon Chopra [ChopraKhullar 99b], il est possible de distinguer trois modes de perception : une basée sur la volonté (perception endogène), une plus réactive (perception exogène, un objet passe dans mon champ de vision et attire l'attention) et enfin une perception passive (le regard se déplace indifféremment sur les entités qui composent la scène). Ces différents modes de perception sont mis en jeu au travers de différents comportements qui reflètent l'état de l'humanoïde. Ils sont donc dirigés par des modules de plus haut niveau.

3.1.3 Discussion et définition d'un cahier des charges

Il est intéressant de mettre en relation les pré-requis associés à l'animation comportementale et les contraintes des différents systèmes d'animation classiques. Le tableau suivant présente ce comparatif. Les critères choisis sont les performances (vitesse d'exécution), le réalisme des animations, les méthodes de spécification des buts à accomplir, la modularité (capacité à changer de tâche et à coexister avec d'autres modules d'animation) et la réactivité (adaptions en temps-réel aux modifications de l'environnement) :

	Critères de choix				
	Performance	**Réalisme**	**Spécification**	**Modularité**	**Réactivité**
Cinématique directe /inverse	temps-réel (dépend de la complexité des chaînes articulaires)	moyen (nécessite l'ajout de fonctions de contrôle)	torseurs cinématiques de l'effecteur	changement de tâche aisé	bonne
Méthodes dynamiques	dépend grandement du nombre de paramètres impliqués	bon (cohérence physique)	difficile (nécessite l'utilisation de contrôleurs complexes)	changement de tâche aisé (si les contrôleurs existent)	bonne
Capture du mouvement	temps-réel (complexité linéaire)	excellent (mimétisme)	très difficile	adaptation coûteuse des mouvements (difficile en temps réel)	aucune

Ce tableau met en avant les capacités des méthodes cinématiques dans le cadre du contrôle temps-réel d'un humanoïde virtuel, notamment grâce à la fléxibilité de leurs modes opératoires. Les points négatifs ou difficiles sont la qualité (au sens du réalisme) des animations résultantes, ainsi que la spécification des tâches. Dans l'idéal, la simulation de la perception d'un humanoïde virtuel requiert l'utilisation d'une méthode permettant :

- une spécification des tâches simples, la simplicité pouvant venir d'une description proche du langage naturel, ou se traduisant par des paramètres de contrôle simples,
- d'être temps-réel et robuste aux modifications de l'environnement, et de même être capable de changer de type de tâche aisément,
- de proposer un formalisme flexible permettant d'introduire des contraintes dans les mouvements générés.

L'utilisation d'une technique référencée vision permet de combler en partie ces lacunes, comme nous allons l'étudier dans la partie suivante.

3.2 Modélisation à l'aide de l'asservissement visuel

Dans cette partie nous nous intéressons à la modélisation de l'animation associée à l'attention visuelle à l'aide de l'asservissement visuel. L'analogie observée entre les problèmes de contrôle de caméra et du contrôle de l'humanoïde amène à envisager l'utilisation du même formalisme. L'idée est de pouvoir lier une chaîne articulaire représentant une partie du corps de l'humanoïde impliquée dans le processus d'attention visuelle à la perception de ce qu'il "voit" au travers de ses "yeux", représentant l'effecteur final de la chaîne. Le choix de cette chaîne est important, et constitue la première partie de cette section. Nous présenterons alors le formalisme utilisé pour animer l'humanoïde, ainsi que des résultats expérimentaux sur des tâches basiques d'observation.

3.2.1 Importance et représentation de la chaîne articulaire

La simulation de l'attention visuelle d'un humanoïde de synthèse met en jeu différentes parties du corps humain. De prime abord, ce sont avant tout les yeux qui jouent le rôle le plus important : ce sont eux qui orientent le regard. Le cou, le torse et le reste du tronc (c'est à dire l'ensemble de la colonne vertébrale) font aussi partie de ce processus, mais dans une part moindre : certaines tâches visuelles ne requièrent pas l'utilisation de l'ensemble de ces degrés de liberté, par exemple regarder son interlocuteur dans une discussion. Si l'on analyse la perception visuelle au niveau tâche, on peut considérer que faire un demi-tour ou se rapprocher d'une cible font partie intégrante des mécanismes physiques mis en jeu par l'attention visuelle. Ainsi, plusieurs chaînes cinématiques sont à considérer. Dans la figure 3.2, deux exemples de ce type de chaîne sont présentés : la figure (a) montre une chaîne composée de neuf degrés de liberté (trois pour l'abdomen, un pour la colonne vertébrale, trois pour le cou et deux pour les yeux). La figure (b) exhibe une version simplifiée de cette même chaîne. Le choix de cette chaîne peut se faire selon différents critères situés à différents niveaux de réflexion :

- au niveau de l'animation : la tâche visuelle spécifiée à l'humanoïde sous tend en elle-même l'utilisation d'un certain nombre de degrés de liberté. Ainsi, une tâche contraignant la distance à l'objet d'intérêt impose l'utilisation d'une chaîne suffisamment longue et éventuellement disposant de degrés de liberté en translation, pour pouvoir générer un mouvement de l'humanoïde. Le choix automatique de la chaîne articulaire en fonction de la tâche est un problème difficile, et ne sera pas traité dans ce document.
- considérations sur les niveaux de détail : la résolution du système d'équations détaillées en 3.2.2 peut être coûteux en temps de calcul. L'augmentation du nombre de degrés de liberté (par exemple une modélisation exhaustive de la colonne vertébrale et de toutes ses articulations) s'opère au détriment du temps de calcul de la loi de commande. Le choix de la complexité de la chaîne articulaire peut s'effectuer en prenant en compte les ressources disponibles. Dans le cadre de foules d'humanoïdes à animer, on peut très bien envisager de dégrader ces chaînes en prenant en compte l'information de point de vue (plus la caméra est loin, et moins on aura besoin d'une chaîne complexe pour exprimer un mouvement visuellement similaire). Ce point fait partie des perspectives concernant les niveaux de détails en animation.
- au niveau de la simulation : le choix de la chaîne cinématique peut dépendre directement du comportement attendu. Le fait que l'action de perception visuelle puisse aider et être impliquée dans des comportements (par exemple un comportement de préhension) va imposer le choix d'une chaîne.

Pour représenter ces différents types de chaînes, nous avons choisi d'utiliser les paramètres de Denavit-Hartenberg, proposant une représentation unifiée, et permettant d'inclure des informations telles que le poids des différentes parties ou la position des centres de masse. Un descriptif de ce paramétrage est donné dans l'annexe D.

3.2.2 Modélisation et formalisme

Dans cette partie notre objectif est de pouvoir contrôler la cinématique de la chaîne articulaire impliquée dans le processus d'attention visuelle. Dans la section 2.2.2, la loi

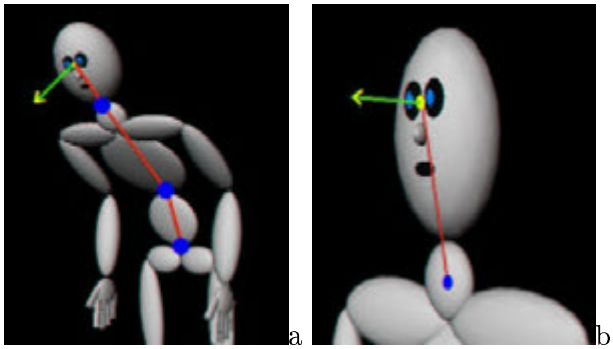

a b

FIG. 3.2 – *Différents types de chaîne articulaires impliquées dans le processus d'attention visuelle (a) neuf degrés de liberté (b) cinq degrés de liberté*

de commande est exprimée dans le repère de la caméra, c'est à dire les yeux dans le cas présent. Pour contrôler le mouvement de la chaîne articulaire, il faut pouvoir exprimer cette commande dans le repère articulaire, via l'utilisation du jacobien du robot. Cette approche est donc extrêmement liée aux techniques de cinématique inverse. Pour cela, il faut établir une nouvelle matrice d'interaction, reliant les variations des primitives géométriques constituant la tâche visuelle aux vitesses articulaires de chaque axe. Cette matrice est définie par :

$$\dot{\mathbf{P}} = \mathbf{H_P}\,\dot{\mathbf{q}} \tag{3.4}$$

Sachant que $\mathbf{T_c} = \mathbf{Jac(q)}\,\dot{\mathbf{q}}$, où $\mathbf{Jac(q)}$ est le jacobien du robot, nous avons :

$$\mathbf{H_P} = \mathbf{L_P}.\mathbf{Jac(q)} \tag{3.5}$$

Si m est le nombre de primitives choisies, et si l'on considère que la chaîne articulaire dispose de n degrés de liberté, la dimension de $\mathbf{H_P}$ est $m \times n$. Si ces primitives sont indépendantes (i.e. pas de caractéristiques communes), le rang l de $\mathbf{H_P}$ vaut m, autrement $m > l$. La fonction de tâche peut alors s'exprimer par :

$$\mathbf{e} = \mathbf{W}^{+}\mathbf{e_1} + (\mathbf{I_n} - \mathbf{W}^{+}\mathbf{W})\mathbf{e_2} \tag{3.6}$$

où $\mathbf{e_1}$ désigne la tâche visuelle et $\mathbf{e_2}$ une tâche secondaire réalisée à la condition que $\mathbf{e_1}$ le soit. Pour que $\mathbf{e}$ ait une décroissance exponentielle et agisse alors comme un système découplé du premier ordre, nous avons :

$$\dot{\mathbf{q}} = -\lambda\mathbf{e} - (\mathbf{I_n} - \mathbf{W}^{+}\mathbf{W})\frac{\partial \mathbf{e_2}}{\partial t} \tag{3.7}$$

avec :

- $\dot{\mathbf{q}}$ la vitesse angulaire donnée aux articulations de l'humanoïde,
- λ est un scalaire réglant la décroissance exponentielle de $\mathbf{e}$.

On rappelle alors que la tâche de vision $\mathbf{e}_1$ est alors donnée par :

$$\mathbf{e_1} = \mathbf{C}(\mathbf{P} - \mathbf{P_d}) \tag{3.8}$$

où $\mathbf{P_d}$ représente les valeurs désirées des primitives visuelles, $\mathbf{P}$ leurs valeurs courantes, et $\mathbf{C}$, appelée matrice de combinaison, doit être choisie telle que $\mathbf{W} = \mathbf{CH_P}$ soit de rang plein le long de la trajectoire $\mathbf{q_r}(t)$.

Mise à jour du modèle réel A chaque pas de temps une nouvelle commande en vitesse est donc envoyée à la chaîne articulaire. Cette commande en vitesse se traduit après intégration en une nouvelle configuration de la chaîne articulaire. Il faut tenir compte du fait que cette même chaîne peut évoluer au cours du temps pour d'autres raisons : un déplacement de l'humanoïde dans la scène indépendant de la tâche de vision, ou bien encore une autre action modifiant des noeuds communs avec la chaîne articulaire. A chaque itération, il faut donc être en mesure de réévaluer l'état réel de la chaîne après modifications. La figure 3.3 montre un diagramme mettant en jeu ce principe. Le processus d'asservissement dispose de sa propre structure interne de chaîne articulaire, lui permettant entre autres de calculer par cinématique directe le torseur cinématique des yeux. Ce modèle interne est remis à jour grâce aux valeurs d'un contrôleur, qui combine le résultat de tous les mouvements de l'humanoïde. Ce contrôleur réalise un mélange de mouvements, basé sur un système de priorités (réalisé par Ménardais [Menardais 03]). On note à ce stade que les mouvements de la chaîne non générés par l'asservissement sont considérés alors comme des mouvements parasites, introduisant des erreurs dans la loi de commande.

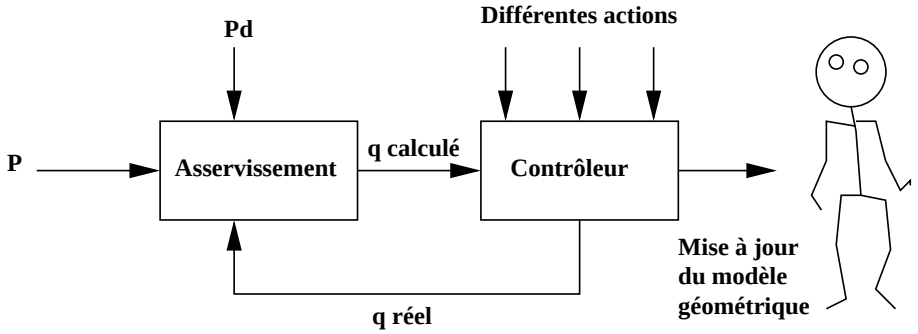


FIG. 3.3 – *Remise à jour de la chaîne articulaire*

Asservissement 3D et 2D1/2 Nous avons considéré dans cette thèse un asservissement visuel 2D classique, car la spécification des tâches est plus intuitive dans un espace 2D. Cependant, notre approche peut facilement se déporter vers un asservissement 3D (en position) classique. Dans ce cas, les primitives $\mathbf{P}$ utilisées dans la loi de commande ne sont plus exprimées relativement à une "tâche visuelle" mais peuvent être décrites, par exemple, comme un déplacement à réaliser par l'effecteur. Ce déplacement peut être calculé relativement à la position de l'humanoïde et de l'objet d'intérêt. Un tel contrôle 3D possède des avantages : les rotations et les translations sont complètement découplées, et la trajectoire de l'effecteur (caméra/yeux) est une pure ligne droite dans l'espace 3D cartésien. L'asservissement 2D 1/2 propose d'utiliser des informations 2D et 3D dans la loi de commande. Une telle approche génère des trajectoires rectilignes dans l'espace 3D, et permet à l'objet d'intérêt de rester dans le champ visuel. Ces deux approches permettent donc un contrôle de meilleure qualité sur les trajectoires 3D produites, mais leurs spécifications sont nettement mois intuitives, c'est pourquoi, dans le cadre des objectifs que nous nous sommes fixés, nous n'avons pas présenté plus en détails ces approches.

3.2.3 Résultats : possibilités et intérêts

Dans les exemples qui suivent nous avons utilisé une chaîne articulaire composée de neuf degrés de liberté. Ces neufs degrés de liberté sont des liaisons de type rotoïde. Ils représentent respectivement l'abdomen (trois rotations), le thorax (un seul degré), le cou (trois rotations) et les yeux (deux rotations).

Tâches de positionnement A l'aide de la chaîne décrite précédemment, nous avons réalisé différentes tâches de positionnement. Le comportement associé à ce genre de tâches peut être décrit comme un comportement d'attention visuel, l'humanoïde cherche à regarder un point précis de la scène. Dans l'exemple présenté dans la figure 3.4, l'humanoïde doit observer un écran d'ordinateur. Nous avons choisi comme primitive d'asservissement une sphère, située au centre du moniteur. L'erreur associée à cette tâche (voir Fig. 3.4(d)) décroît exponentiellement au cours de l'animation, jusqu'à être complètement nulle (cette décroissance se retrouve dans l'évolution des vitesses calculées à chaque pas de temps, figure. 3.4(c)). La figure 3.4(b) montre la configuration articulaire de la chaîne au cours de l'animation. On constate que toutes les articulations sont impliquées dans le mouvement.

Tâches de suivi dans l'image Nous avons choisi comme exemple simple de s'asservir sur le centre d'une balle. Une tâche est donnée à l'humanoïde : déplacer cette balle vers une position prédéfinie. La tâche représentant l'attention visuelle consiste à centrer dans l'image "perçue" cette balle. Cette tâche est complètement indépendante du mouvement du bras, et il n'y a aucune connaissance a priori sur le mouvement de la balle intégrée dans l'asservissement. Seul le centre de cette balle est connu à chaque itération. La figure 3.5 montre les résultats associés à cet exemple. La première série d'images est une vue d'ensemble de la scène, et la seconde est une représentation subjective de ce que voit l'humanoïde.

L'évolution de l'erreur selon les deux axes X et Y associée à la tâche est montrée dans la figure 3.6. Lors des premières itérations, l'erreur décroît exponentiellement. Une fois que l'humanoïde commence à déplacer la balle, on note que l'erreur n'est alors plus nulle, dû à un retard introduit dans la loi de commande par le mouvement propre de la balle. La technique présentée dans la section 2.2.4.2 peut permettre de remédier à ce problème.

L'exemple de la peinture Dans cet exemple, on considère une tâche de vision difficile à modéliser avec des techniques d'animation classique. Le but est de regarder une peinture, de telle manière qu'elle soit centrée dans l'image "perçue". La spécification de la tâche est la suivante : quatre points correspondant aux quatre coins du tableau sont choisis. Le motif désiré est de voir apparaître ces quatre coins de sorte que la peinture soit centrée dans l'image. La matrice d'interaction est donc construite en empilant les quatre jacobiens relatifs aux quatre coins. C'est une matrice de dimension 8×6 de rang six, qui définit un lien rigide entre les yeux de l'humanoïde et le tableau. L'animation résultante est présentée dans la figure 3.7 (la peinture n'est pas dans le champ de vision de la figure 3.7(a)). Dans les images représentant la vision subjective de l'humanoïde, nous avons dessiné les quatre positions désirées pour les coins du tableau (en rouge). A la fin de l'animation, ils correspondent aux quatre coins du tableau. La spécification de la tâche s'est donc déroulée en deux étapes : extraire les quatre coins du tableau, et définir leurs positions dans l'image.

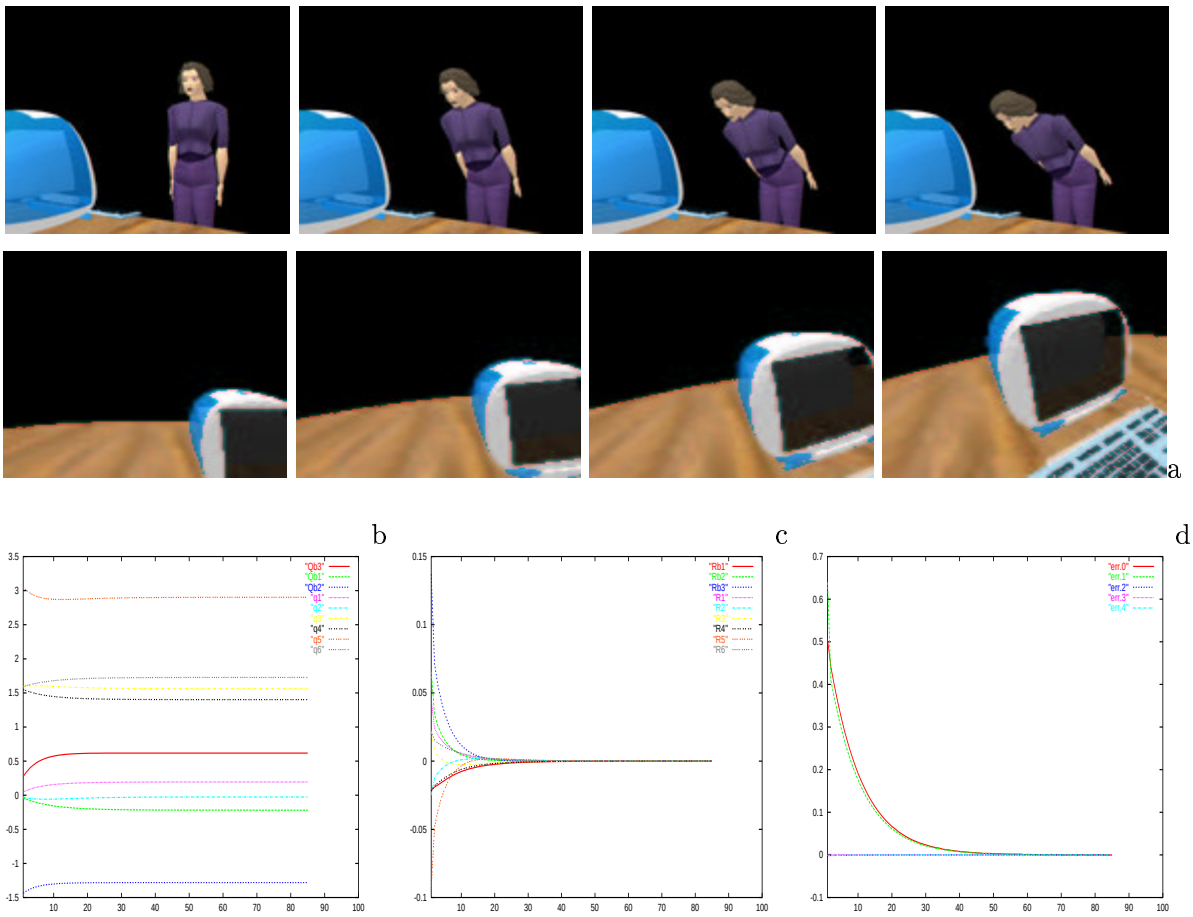


FIG. 3.4 – *Tâche de positionnement relative à une sphère (a) vue d'ensemble et vue du personnage (b) configuration articulaire (c) vitesses calculées (d) erreurs dans l'image relative à la sphère*

Du point de vue de l'animation comportementale, ces informations peuvent être contenues à la base dans l'objet. Ce point sera détaillé dans le chapitre 6.

3.2.4 Conclusion partielle

Dans le début de ce chapitre, nous avons étudié les différents moyens de contrôle d'un humanoïde de synthèse, dans le but de simuler la cinématique associée à la perception visuelle d'un humanoïde de synthèse. Après avoir établi un certain nombres de critères pertinents au regard de l'animation comportementale, nous avons extrait les principales caractéristiques désirées pour un système d'animation simulant la perception visuelle. L'analogie avec la technique de contrôle de caméra présentée dans les chapitres précédents nous a amené à utiliser le même formalisme, l'asservissement visuel. Plusieurs points justifient notre choix : simplification des paramètres d'entrée, mimétisme avec la perception humaine dans la spécification de ces entrées, adaptation dynamique et temps réel du système dans des environnements inconnus (grâce à une boucle de contrôle fermée intégrant des retours du modèle réel). Le principal défaut de cette approche est d'une certaine manière identique

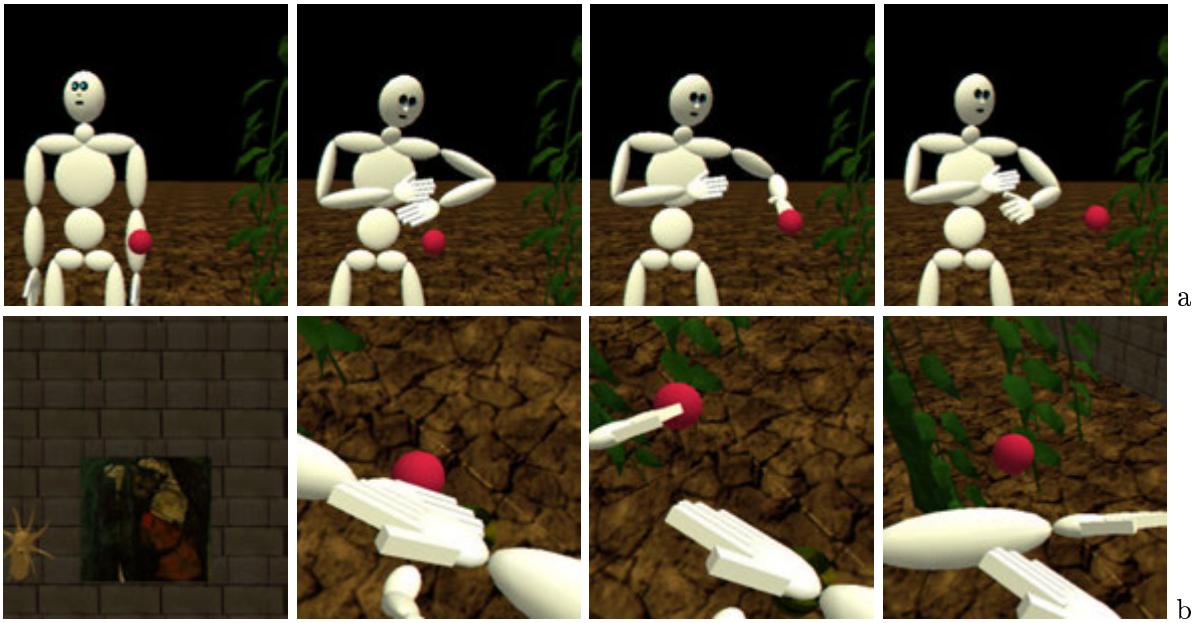


FIG. 3.5 – *Suivre des yeux un objet qui bouge : (a) vue globale de la scène (b) vision subjective de l'humanoïde*

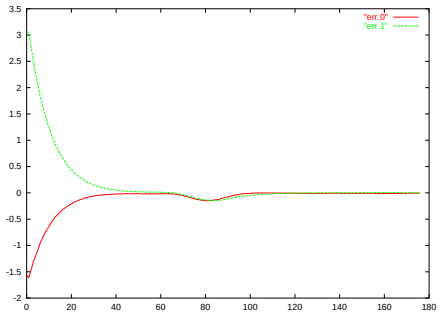


FIG. 3.6 – *Evolution de l'erreur (axe X : itérations, axe Y : valeurs de l'erreur le long des deux axes)*

à celui rencontré dans le cadre du contrôle d'une caméra : on ne dispose que d'un contrôle lâche sur les trajectoires articulaires, pouvant conduire à des mouvements non-réalistes. Dans la suite de ce chapitre, nous étudions la manière dont l'approche retenue permet de s'adapter aux problèmes relatifs à l'animation d'un humanoïde au travers de l'étude de quelques unes de ces grandes classes de problème. Dans un premier temps nous étudierons la manière de régler le problème des butées articulaires. Dans un deuxième temps, une technique d'enchaînement entre les différentes tâches visuelles et les différentes configurations articulaires sera proposée. Nous évoquerons alors la manière de coupler la technique d'animation avec un autre mouvement, en l'occurrence de la locomotion, dans le but de définir des tâches visuelles complexes. Puis nous étudierons le problème du réalisme associé au mouvement obtenu, et nous proposerons des méthodes pour améliorer celui-ci.

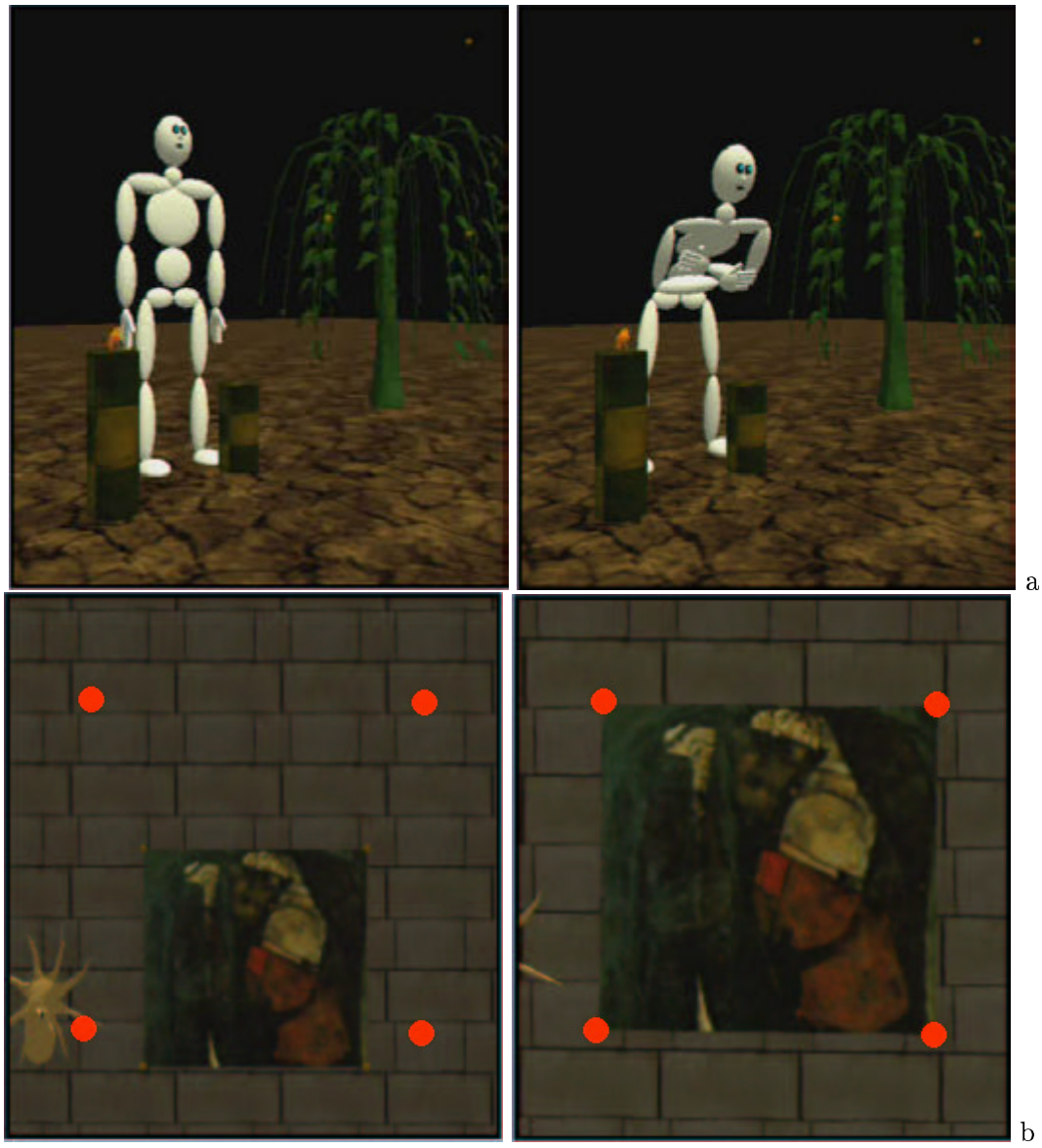


FIG. 3.7 – *Asservissement sur quatre points : (a) vue globale de la scène (b) vision subjective de l'humanoïde*

3.3 Gestion des butées articulaires

Le problème des butées articulaires désigne le problème associé aux limitations physiques de chaque articulation, bornées à un domaine limité de mouvements. C'est un problème fondamental en animation d'humanoïdes, dans la mesure où les animations produites doivent être réalistes. Ainsi, on ne peut pas tourner le cou à plus de plus de 90 degrés autour de son axe (en moyenne, chaque individu disposant de ses propres caractéristiques anatomiques). Un état de l'art rapide concernant les techniques d'évitement des butées sera présenté, puis nous verrons dans un deuxième temps comment aborder ce problème à l'aide d'une méthode qui calcule un gain pour chaque axe et nous présenterons les résultats associés à ces deux méthodes.

3.3.1 Approches classiques

De manière classique, l'évitement des butées articulaires se résout dans les méthodes cinématiques par l'utilisation d'une fonction secondaire. La tâche secondaire est définie comme le gradient d'une fonction de coût h_s ($\mathbf{e_2} = \frac{\partial h_s}{\partial r}$). Cette fonction doit atteindre son maximum lorsque une articulation atteint une de ces deux limites, et son gradient doit être nul lorsque cette fonction atteint son minimum [Samson 91]. Plusieurs fonctions de coût ont été proposées qui reflètent le comportement désiré dans [Chang 95,Marchand 96]. Voici un aperçu de celle proposée dans [Marchand 96] (voir Fig. 3.8).

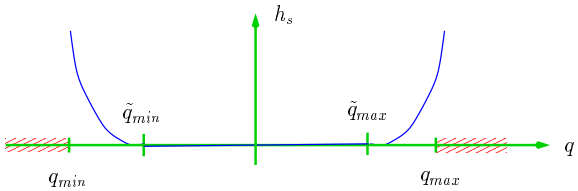


FIG. 3.8 – *Tâche secondaire d'évitement de butées*

Les seuils d'activation $\tilde{\mathbf{q}}_{i_{min}}$ et $\tilde{\mathbf{q}}_{i_{max}}$ de la tâche secondaire sont définis comme suit :

$$\begin{aligned} \tilde{\mathbf{q}}_{i_{min}} &= \mathbf{q}_{i_{min}} + \rho\,(\mathbf{q}_{i_{max}} - \mathbf{q}_{i_{min}}) \\ \tilde{\mathbf{q}}_{i_{max}} &= \mathbf{q}_{i_{max}} - \rho\,(\mathbf{q}_{i_{max}} - \mathbf{q}_{i_{min}}) \end{aligned} \tag{3.9}$$

où $0 < \rho < 1/2$ (en général, $\rho = 0.1$). La fonction de coût est alors donnée par :

$$h_s = \beta \sum_{i=1}^{n} \frac{s_i^2}{\mathbf{q}_{i_{max}} - \mathbf{q}_{i_{min}}} \tag{3.10}$$

$$\text{où} \quad s_i = \begin{cases} \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{max}} & \text{si } \mathbf{q}_i > \tilde{\mathbf{q}}_{i_{max}} \\ \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{min}} & \text{si } \mathbf{q}_i < \tilde{\mathbf{q}}_{i_{min}} \\ 0 & \text{sinon.} \end{cases} \tag{3.11}$$

et où β désigne un scalaire réglant l'amplitude de l'influence de la tâche secondaire sur la

tâche principale. Les valeurs de $\mathbf{e_2}$ et $\frac{\partial \mathbf{e_2}}{\partial t}$ s'expriment comme suit :

$$\mathbf{e_{2}}_i = \begin{cases} \frac{(\mathbf{q}_i - \tilde{\mathbf{q}}_{i_{max}})}{(\mathbf{q}_{i_{max}} - \mathbf{q}_{i_{min}})} & \text{si } \mathbf{q}_i > \tilde{\mathbf{q}}_{i_{max}} \\ \frac{(\mathbf{q}_i - \tilde{\mathbf{q}}_{i_{min}})}{(\mathbf{q}_{i_{max}} - \mathbf{q}_{i_{min}})} & \text{si } \mathbf{q}_i < \tilde{\mathbf{q}}_{i_{min}} \\ 0 & \text{autrement.} \end{cases} , \frac{\partial \mathbf{e_{2}}_i}{\partial t} = 0 \tag{3.12}$$

Dans le cadre de cette méthode, l'importance du paramètre de gain β (cf équation 3.10) est très importante. Si β est trop petit, l'influence de la tâche secondaire sera diminuée par rapport à la tâche principale, résultant en un dépassement des butées. S'il est trop grand, le système peut osciller autour de la butée. Le paramètre β doit alors être configuré de manière manuelle, par une méthode d'essais-erreurs, ce qui n'est pas acceptable. Il existe des manières de régler ce paramètre de manière globale [Marchand 96], en utilisant une phase de prédiction sur l'itération suivante. La solution consiste alors à régler β dans le but de couper la vitesse sur l'axe qui est le plus en difficulté. Cependant le réglage global de ce paramètre ne prend pas en compte la possibilité que d'autres axes puissent rentrer dans des zones critiques. La méthode suivante se propose de régler un gain propre à chaque composant de la tâche secondaire.

3.3.2 Calcul d'un mouvement adéquat

Le problème posé peut se résoudre en coupant tout mouvement sur les axes qui sont en difficulté (près d'une butée), ce qui équivaut à calculer une vitesse nulle pour cet axe (i.e. $\dot{\mathbf{q}}_k = 0$, si l'axe k est en difficulté). L'idée de cette méthode est de calculer un gain pour chaque axe en résolvant un système linéaire.
De manière générale, une fonction de tâche utilisant la redondance peut s'écrire comme[1] :

$$\mathbf{e} = \mathbf{W}^+ \mathbf{e_1} + \sum_{i=1}^{n_0} \mathbf{B}_i \mathbf{E}_{\bullet i} \tag{3.13}$$

avec $n_0 = \dim \mathrm{Ker}\mathbf{W} = n - l$, et $\sum_{i=1}^{n_0} \mathbf{B}_i \mathbf{E}_{\bullet i}$ définissant les mouvements effectués pour ne pas arriver en butée articulaire. Dans ce cadre, $\mathbf{E}$ est une base du noyau de $\mathbf{W}$ de dimension $n \times n_0$ (pas d'influence sur la tâche principale), et $\mathbf{B}$ le vecteur de gains que l'on cherche à calculer automatiquement.
Si l'axe k est en difficulté, nous cherchons un vecteur $\mathbf{B}$ tel que $\dot{\mathbf{q}}_k = 0$. De l'équation (3.13), nous obtenons alors le système suivant :

$$\sum_{i=1}^{n_0} \mathbf{B}_i \mathbf{E}_{ki} = - \left[\mathbf{W}^+ \mathbf{e_1}\right]_k \tag{3.14}$$

Considérons un système ou p axes sont en difficulté, le système (3.14) est linéaire et peut se réécrire sous la forme $\mathbf{FB} = \mathbf{S}$, où $\mathbf{F}$ est de dimension $p \times n_0$ alors que $\mathbf{S}$ et $\mathbf{B}$ sont de dimension n_0. Nous devons alors distinguer 3 cas :

- si $p > n_0$, il y a plus d'axes en difficulté que d'axes redondants. La validité de la méthode n'est alors plus assurée (on cherche à réaliser quelque chose d'impossible).

1. Si $\mathbf{M}$ est une matrice, nous notons $\mathbf{M}_{\bullet i}$ sa $i^{\text{ème}}$ colonne et $\mathbf{M}_{i\bullet}$ sa $i^{\text{ème}}$ ligne

- si $p = n_0$, il existe une unique solution et le problème peut être résolu.
- si $p < n_0$, il y a plusieurs solutions.

Dans tous les cas, une solution peut être obtenue en résolvant $\mathbf{B}^* = \mathbf{F}^+\mathbf{S}$.

Considérons plus précisemment le cas où $(p < n_0)$. Si le vecteur $\mathbf{B}^*$ est calculé tel que $\mathbf{B}^* = \mathbf{F}^+\mathbf{S}$, tout mouvement sur l'axe p en situation critique sera coupé. La loi de contrôle résultante est :

$$\dot{\mathbf{q}} = -\lambda \left(\mathbf{W}^+ \mathbf{e_1} + \sum_{i=1}^{n_0} \mathbf{B}_i^* \mathbf{E}_{\bullet i} \right) \tag{3.15}$$

Des compléments à cette méthode sont donnés dans [Chaumette 00]. On peut noter que le principal défaut de cette méthode est que la loi de contrôle n'a pas de continuité d'ordre deux, pouvant résulter dans des changements brusques dans les vitesses envoyées à l'humanoïde. Cependant le calcul automatique d'un gain pour chaque axe permet de déterminer une réaction appropriée pour tous les axes indépendamment les uns des autres.

3.3.3 Résultats

Pour évaluer le système de butées articulaires, le but de la manipulation est maintenant différent. Un point se déplace devant l'humanoïde. L'humanoïde doit conserver ce point centré dans l'image "perçue" de la scène. Cette tâche est une tâche de *tracking* ou tâche de suivi visuel. Cette tâche constitue un comportement visuel basique chez l'être humain (suivre du regard un objet en mouvement). Un problème survenant lors de la réalisation de cette tâche est que l'objet ayant une vitesse propre dans la scène, une erreur dite de *traînage* intervient Il est possible de régler ce problème en utilisant la technique présentée en 2.2.4.2.

Dans le cas traité, trois axes redondants permettent d'assurer la tâche. Ces axes rentrent progressivement en butée. Dans la figure 3.9, la technique utilisant la tâche secondaire a été utilisée à l'aide de différents paramètres β. Dans les courbes suivantes, les diagrammes sont normalisés, la valeur maximale de la butée étant 1, et la valeur $\tilde{\mathbf{q}}_{i_{max}}$ est spécifiée par la limite de la zone critique. Dans le cas (a), le paramètre β n'est pas assez élevé pour influer sur la réalisation de la tâche principale. Dans le cas (b), sa valeur n'est pas assez élevée pour résoudre le problème de manière efficace. Pour $\beta = 10$ (c), la solution obtenue est assez convaincante, et retarde l'arrivée en butées. Lors de l'utilisation de la méthode permettant de régler un gain pour chaque axe, la coupure se fait de manière très nette sur l'axe en difficulté (voir Fig. 3.9(d)), et le réglage du gain est automatique. Dans les deux méthodes présentées, on peut s'apercevoir qu'une fois que tous les axes pouvant réaliser la tâche sont rentrés en butée la validité des méthodes n'est plus assurée, et il faut alors stopper la réalisation du mouvement.

Le but de ces différentes techniques est de retarder au maximum l'arrivée en butées articulaires. Quand l'arrivée en butées est inévitable, il faut pouvoir réagir de manière correcte et adaptée. Différentes manières d'envisager ce problème sont possibles (relâcher des degrés de liberté, chercher à réaliser une tâche différente, planification). Il est aussi nécessaire d'interpréter ce fait au niveau comportemental, avec adaptation éventuel du comportement. La figure 3.10 montre une comparaison entre deux animations réalisées

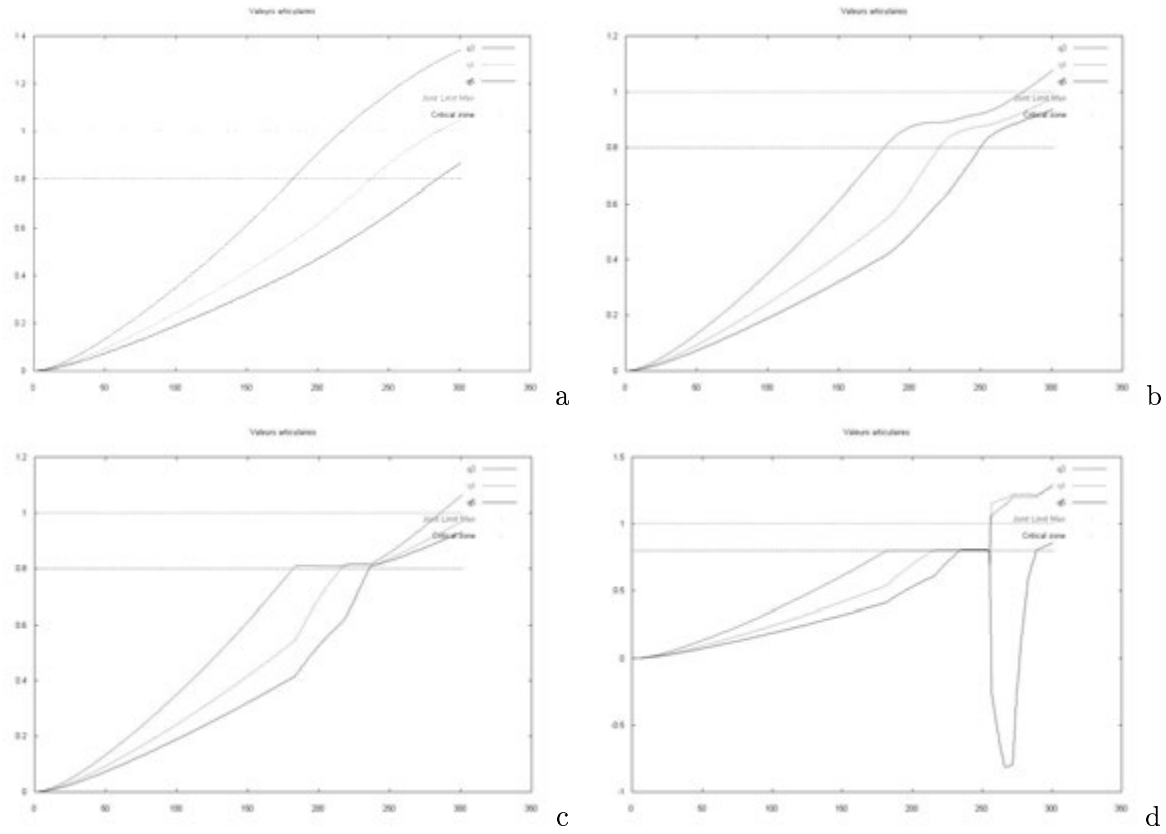


FIG. 3.9 – *Evitement des butées à l'aide de la tâche secondaire : différentes valeurs du paramètre β (a) $\beta = 0.1$, (b) $\beta = 1$, (c) $\beta = 10$ (d) la méthode calculant un gain par axe*

sans et avec la méthode d'évitement des butées articulaires. Dans ce cas, on considère une chaîne articulaire disposant de neuf degrés de liberté (chaîne considérée auparavant). La tâche est une tâche de suivi visuel d'un point se déplaçant sur un axe parallèle à l'axe du corps du personnage. Dans la première série d'images (images 3.10(a)), aucun processus d'évitement des butées n'est considéré. A la fin de l'animation, les yeux sont quasiment révulsés, et le cou a pivoté de 360 degrés. La figure 3.10(c) montre l'évolution angulaire des articulations permettant de réaliser la tâche. Cette évolution est constamment croissante, montrant ainsi que les butées sont dépassées tôt ou tard. A l'inverse la figure 3.10(d) montre l'évolution des mêmes articulations en considérant un processus d'évitement des butées. Les yeux (en vert), stoppent leurs mouvements nettement plus rapidement que le cou et l'abdomen (rouge et bleu), car leurs butées ont été fixées arbitrairement à ± 28 degrés (± 45 degrés pour les deux autres). A la fin de l'animation 3.10(b) où le processus d'évitement des butées est considéré, on constate que l'humanoïde est penché en arrière. Ceci témoigne du fait que d'autres axes sont utilisés pour réaliser la tâche. Lorsque tous les axes redondants sont en butées, on considère alors le mouvement comme impossible et on le stoppe. On note que dans la plupart des animations qui ont été réalisées dans la suite de cette thèse, les paramètres associés aux butées ont été fixés de manière empirique, pouvant conduire à certains problèmes de réalisme. Il est possible de trouver dans la littérature associée à la

biomécanique les valeurs de ces butées, mais ce travail n'a pas été effectué.

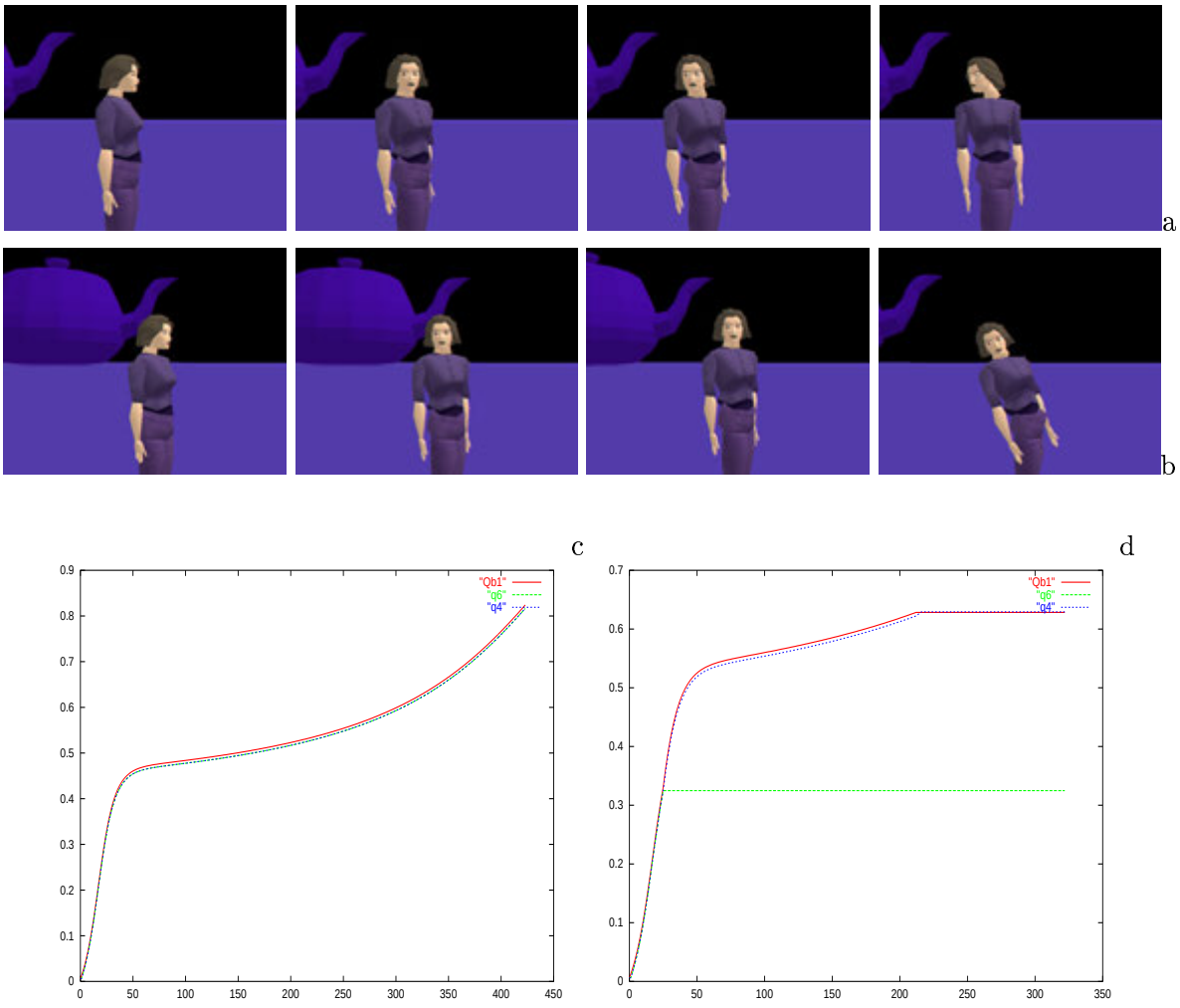


FIG. 3.10 – *Evitement des butées articulaires (a) sans butées (b) avec butées ; évolution des axes permettant d'assurer la tâche (en vert les yeux, en rouge et bleu le cou et le bassin) (c) sans butées (d) avec butées*

3.4 Transitions entre différentes tâches

Dans la mesure où nous avons étudié la manière d'animer la partie supérieure du corps d'un humanoïde en respectant des tâches visuelles, il est intéressant de pouvoir passer d'une tâche à l'autre pour générer des comportements d'attention plus complexes. Dans ce but, il faut séquencer ces tâches d'une manière correcte, tout en considérant des exigences de réalisme (qualité de l'animation), ainsi que les problèmes intrinsèques à la méthode d'animation (convergence de l'algorithme, projection perspective des objets). Nous considérerons trois types d'enchaînement :

- passer d'une chaîne articulaire à une autre en maintenant la même tâche visuelle,
- passer d'une tâche visuelle à une autre tout en conservant la même chaîne articulaire,
- et ces deux transitions en même temps.

Le choix et le sens du passage d'une configuration à une autre relèvent du domaine du comportement. Cette thématique sera abordée dans les chapitres suivant, et plus particulièrement dans la section 6 (en ce qui concerne l'enchaînement de tâches visuelles).

Rappelons quelques notations : $\mathbf{e_i}$ sont les deux tâches considérées (telles que $\mathbf{e_1}$ soit la tâche courante et $\mathbf{e_2}$ la tâche destination). Soit $\mathbf{P_i}$ le vecteur définissant les informations visuelles utilisées dans le processus d'asservissement. Soit d_i la dimension de ce vecteur, c.a.d $d_i = dim(\mathbf{P_i})$. Si $\mathbf{L_i}$ est la matrice d'interaction image associée à la tâche i, on note m_i le rang de cette matrice, c.a.d $m_i = dim(Ker(\mathbf{L_i}))$. Soit $\mathbf{H_i}$ la matrice d'interaction de la tâche définie comme la combinaison du jacobien image et du jacobien de la chaîne articulaire considérée ($\mathbf{Jac_i(q)}$, on a $\mathbf{H_i} = \mathbf{L_i Jac_i(q)}$. On souhaite que la transition dure une période τ autour de l'instant t_0.

3.4.1 Transition d'un modèle de robot à un autre

Il y a plusieurs raisons pour vouloir changer de modèle de chaîne (cf. section 3.2.1). Ce changement doit pouvoir s'opérer dynamiquement, et si possible avec une certaine continuité. En changeant de modèle, on conserve les mêmes primitives visuelles, c.a.d $\mathbf{L_1} = \mathbf{L_2}$. Le paramètre qui change est le jacobien articulaire, soit $\mathbf{Jac_1(q)} \neq \mathbf{Jac_2(q)}$ et donc $\mathbf{H_1} \neq \mathbf{H_2}$. Une solution est de combiner les deux vecteurs solution à l'aide d'une fonction type sigmoïde :

$$\dot{\mathbf{q}} = \alpha(v)\dot{\mathbf{q}}_\mathbf{1} \oplus (1 - \alpha(v))\dot{\mathbf{q}}_\mathbf{2} \tag{3.16}$$

avec α une sigmoïde (cf Fig. 3.11) variant entre 0 et 1 et v un scalaire défini comme une fonction du temps et du temps de transition (par exemple $v = t - t_0 + \frac{\tau}{2}$). Les deux vecteurs $\mathbf{q_1}$ et $\mathbf{q_2}$ sont généralement de dimensions différentes. L'opérateur de composition $\oplus$ représente alors une composition linéaire simple pour les articulations communes, et laisse inchangé les articulations non communes aux deux chaînes articulaires.

3.4.2 Transition d'une tâche visuelle à une autre

Validité de la transition. Avant de changer de tâche visuelle, il est intéressant de se demander si la cible suivante est accessible ou non, du point de vue de l'asservissement. Ce problème est assez complexe, dans la mesure où il faut aussi considérer le cas où les

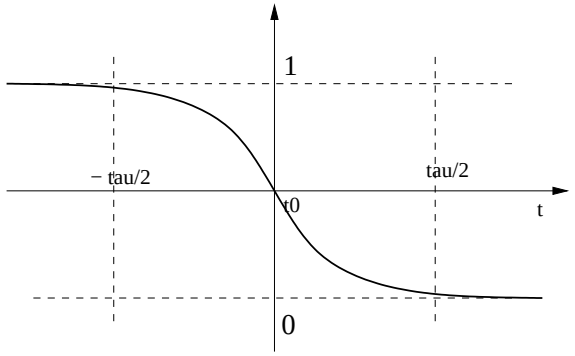


FIG. 3.11 – *Exemple de fonction α*

cibles se déplacent au cours du temps. Cependant, il faut avant toute chose s'assurer que l'objet d'intérêt soit situé *devant le plan image*. En effet, l'information issue de la projection perspective n'est valable que dans le cas d'un objet situé devant les yeux. Dans tous les cas, il apparaît plus judicieux de générer des cibles intermédiaires entre les différentes cibles. Cette approche possède deux avantages :

- il est alors permis de passer d'une primitive à une autre, quelque soient leurs situations respectives (à condition qu'elles soient toutes les deux devant le plan image),
- si l'on décrit un objectif $\mathbf{P_d}(t)$ variant au cours du temps, on supprime la convergence exponentielle vers l'objectif qui peut engendrer des mouvements non réalistes, car on se situe toujours près du but à atteindre. Nous allons envisager cette approche dans deux cas de figures : passage d'une primitive à une autre de même type et transition entre deux primitives de nature différente.

3.4.2.1 Cas simple : transition entre deux primitives de même type

Dans ce cas, on a $dim(\mathbf{P_1}) = dim(\mathbf{P_2})$. On définit alors un nouveau vecteur $\mathbf{P_3}$ comme une fonction du temps t [Pissard-Gibollet 93]:

$$\mathbf{P_3}(t) = \alpha(v)\mathbf{P_1} + (1 - \alpha(v))\mathbf{P_2} \tag{3.17}$$

et un vecteur désiré défini en fonction du temps t:

$$\mathbf{P_{d_3}}(t) = \alpha(v)\mathbf{P_{d_1}} + (1 - \alpha(v))\mathbf{P_{d_2}} \tag{3.18}$$

on calcule alors un nouveau jacobien image $\mathbf{L_3}$:

$$\mathbf{L_3} = \alpha(v)\mathbf{L_1} + (1 - \alpha(v))\mathbf{L_2} \tag{3.19}$$

et la nouvelle vitesse est alors donnée entre $t - \frac{\tau}{2}$ et $t + \frac{\tau}{2}$ par :

$$\dot{\mathbf{q}}_3 = -\lambda \mathbf{H_3}^{+}(\mathbf{P_3}(t) - \mathbf{P_{d_3}}(t)). \tag{3.20}$$

où $\mathbf{H_3}$ désigne la nouvelle matrice d'interaction calculée à partir de $\mathbf{L_3}$. La transformation de $\mathbf{P_1}$ to $\mathbf{P_2}$ peut être considérée comme un *morphing* d'une primitive à une autre (dans ce cas, c'est une transformation barycentrique dépendante de $\alpha(v)$ et de la fonction de composition).

3.4.2.2 Transitions complexes

Une solution adéquate dans le cas où les primitives ne sont pas de même nature (c.a.d généralement $dim\ \mathbf{P_1} \neq dim\ \mathbf{P_2}$) est de générer un vecteur de vitesses intermédiaire combinant les deux vecteurs de vitesse solutions issus des deux tâches (comme dans l'équation 3.16). Cependant nous avons choisi d'effectuer cette transition dans une véritable fonction de tâche, dans le but d'obtenir un contrôle plus propre et une animation plus fluide. Pour cela, on créé une fonction de tâche cherchant à minimiser la différence entre deux primitives de même type. Pour cela, il faut pouvoir exprimer la primitive de dimension la plus faible en une primitive de dimension égale à l'autre. Ce "plongement", au sens algébrique du terme, permet alors de considérer une transition entre deux primitives de même type, et l'on se situe donc dans le cadre étudié avant. Cette transformation n'étant pas canonique (on ne peut pas la modéliser de manière théorique pour toutes les combinaisons de primitives distinctes), nous présentons quelques exemples de celle-ci, ainsi qu'une ébauche de généralisation. Dans la suite de cette section, nous désignons par $\mathbf{P_{aug}}$ la primitive ainsi augmentée.

Un point vers quatre points. Si la primitive de base est un point, et que l'on cherche à passer à quatre points, nous considérons qu'au début de la transition les quatre points sont confondus en un seul, qui est le point de base (voir figure 3.12). $\mathbf{P_{aug}}$ est alors spécifiée comme l'empilement de quatre points identiques.

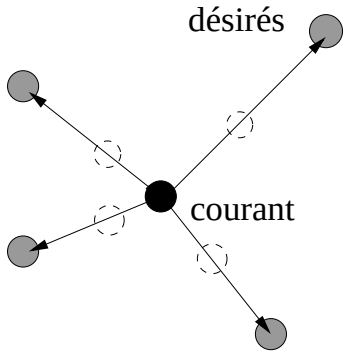


FIG. 3.12 – *Morphing d'un point vers quatre points*

On note que cette technique peut s'appliquer à un nombre indéfini de points. Soient **p1**, **p2**,**p3**, **p4** (avec (X_i,Y_i) leurs coordonnées respectives et z_i leurs profondeurs) les quatre points définissant la tâche visuelle, la nouvelle matrice d'interaction est donnée par :

$$\mathbf{L}_{\mathbf{P}}^{T} = \begin{bmatrix} \mathbf{L}_{\mathbf{p_1}}^{T} \\ \mathbf{L}_{\mathbf{p_2}}^{T} \\ \mathbf{L}_{\mathbf{p_3}}^{T} \\ \mathbf{L}_{\mathbf{p_4}}^{T} \end{bmatrix} \tag{3.21}$$

qui donne :

$$\mathbf{L}^T_{\mathbf{p_{1,2,3,4}}} = \begin{pmatrix} -1/z_1 & 0 & X_1/z_1 & X_1Y_1 & -(1+X_1^2) & Y_1 \\ 0 & -1/z_1 & Y_1/z_1 & 1+Y_1^2 & -X_1Y_1 & -X_1 \\ -1/z_2 & 0 & X_2/z_2 & X_2Y_2 & -(1+X_2^2) & Y_2 \\ 0 & -1/z_2 & Y_2/z_2 & 1+Y_2^2 & -X_2Y_2 & -X_2 \\ -1/z_3 & 0 & X_3/z_3 & X_3Y_3 & -(1+X_3^2) & Y_3 \\ 0 & -1/z_3 & Y_3/z_3 & 1+Y_3^2 & -X_3Y_3 & -X_3 \\ -1/z_4 & 0 & X_4/z_4 & X_4Y_4 & -(1+X_4^2) & Y_4 \\ 0 & -1/z_4 & Y_4/z_4 & 1+Y_4^2 & -X_4Y_4 & -X_4 \end{pmatrix} \tag{3.22}$$

qui est de rang six. Au début de la transformation, on a $X_1 = X_2 = X_3 = X_4 = X_0$, $Y_1 = Y_2 = Y_3 = Y_4 = Y_0$ et $z_1 = z_2 = z_3 = z_4 = z_0$ qui rend $\mathbf{L_{p_{1,2,3,4}}}$ de rang deux et équivalente à la matrice d'interaction d'un point simple (cf. équation 2.6). Dès que $\alpha(v)$ décroît, on calcule une nouvelle matrice d'interaction (cette fois-ci de rang six) donnée par $X_i^d = \alpha(v)X_0 + (1-\alpha(v))X_i$, $Y_i^d = \alpha(v)Y_0 + (1-\alpha(v))Y_i$, et $z_i^d = \alpha(v)z_0 + (1-\alpha(v))z_i$).

Un point vers une sphère. Lorsqu'on part d'un point pour arriver à une sphère, on considère que le point de départ est en fait une sphère de rayon nul. $\mathbf{P_{aug}}$ est dans ce cas un vecteur définissant une sphère de rayon zéro.

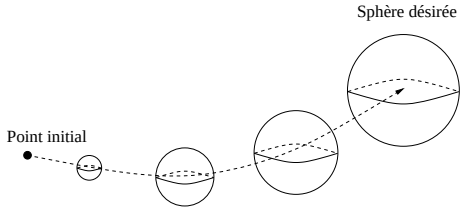


FIG. 3.13 – *Transition d'un point à une sphère*

La figure 3.13 montre une interprétation graphique possible d'une telle transformation. La projection d'une sphère sur un plan est une ellipse [Chaumette 90], définissable par cinq paramètres X_c, Y_c, μ_0, μ_1, μ_2, respectivement le centre de l'ellipse dans l'image, la longueur des deux axes et sa surface. La matrice d'interaction est donnée par :

$$\begin{array}{l} \mathbf{L}^T_{\mathbf{X_c}} = \begin{bmatrix} -1/z_c & 0 & X_c/z_c + a\mu_0 + b\mu_1 & X_cY_c + \mu_1 & -(1+X_c^2) - \mu_0 & Y_c \end{bmatrix} \\ \mathbf{L}^T_{\mathbf{Y_c}} = \begin{bmatrix} 0 & -1/z_c & Y_c/z_c + a\mu_1 + b\mu_2 & 1+Y_c^2 + \mu_2 & -X_cY_c - \mu_1 & -X_c \end{bmatrix} \\ \mathbf{L}^T_{\mu_0} = \begin{bmatrix} -2(a\mu_0 + b\mu_1) & 0 & 2[(1/z_c + aX_c)\mu_0 + bX_c\mu_1] & 2(Y_c\mu_0 + X_c\mu_1) & -4\mu_0X_c & 2\mu_1 \end{bmatrix} \\ \mathbf{L}^T_{\mu_1} = \begin{bmatrix} -(a\mu_1 + b\mu_2) & -(a\mu_0 + b\mu_1) & aY_c\mu_0 + (3/z_c - c)\mu_1 + bX_c\mu_2 & & & \\ & & 3Y_c\mu_1 + X_c\mu_2 & -(Y_c\mu_0 + 3X_c\mu_1) & \mu_0 - \mu_2 & \end{bmatrix} \\ \mathbf{L}^T_{\mu_2} = \begin{bmatrix} 0 & -2(a\mu_1 + b\mu_2) & 2[(1/z_c + bY_c)\mu_2 + aY_c\mu_1] & 4Y_c\mu_2 & -2(Y_c\mu_1 + X_c\mu_2) & -2\mu_1 \end{bmatrix} \end{array} \tag{3.23}$$

avec $z_c = 1/(aX_c + bY_c + c)$.
quand le rayon de la sphère est nulle, on a $\mu_0 = \mu_1 = \mu_2 = 0$ donnant la matrice d'interaction suivante :

$$\mathbf{L}^T_{\mathbf{S_{r=0}}} = \begin{pmatrix} -1/z_c & 0 & X_c/z_c & X_cY_c & -(1+X_c^2) & Y_c \\ 0 & -1/z_c & Y_c/z_c & 1+Y_c^2 & -X_cY_c & -X_c \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.24}$$

de rang deux, et équivalente à la matrice d'interaction associée à un point.

Inconvénients et généralisation. Un des principaux inconvénients de cette méthode est l'absence de transformation canonique permettant de passer d'une ensemble X de primitives à un autre ensemble Y. Une solution pourrait être d'exprimer ces ensembles de primitives dans un représentation commune de dimension finie (par exemple la représentation étoilée utilisée en mathématique algébrique), mais cette approche possède des limites dues à des contraintes topologiques (trous, enchevêtrements, objets en plusieurs parties). On peut alors envisager de créer un graphe groupant l'ensemble des primitives possibles, avec comme arc les transitions entre ces primitives. Trouver une transformation entre n'importe quels noeuds reviendrait à trouver le plus court chemin entre les deux noeuds de ce graphe. Nous n'avons pas exploré cette possibilité dans le cadre de cette thèse. Un deuxième inconvénient est de réaliser une interpolation dans l'image. Si cette approche est intéressante sur le plan du contrôle (souplesse, cohérence), la validité des trajectoires $3D$ générées peut ne pas correspondre à certains critères de réalisme. Il est donc intéressant dans un deuxième temps de s'intéresser à l'interpolation des cibles virtuelles dans l'espace $3D$.

3.4.2.3 Interpolation des cibles virtuelles

L'utilisation de fonctions d'interpolation de cibles permet dans le cadre de la synthèse de mouvements de chaînes articulées d'exercer un contrôle temporel sur la réalisation d'un geste. A titre d'exemple, ce genre de méthodes a été utilisé pour la simulation des gestes co-verbaux [Gibet 94,Lebourque 99], notamment grace à l'utilisation de fonctions d'activation temporelles. Dans notre cas, les interpolations étaient effectuées dans le plan image. Lorsque c'est possible, il peut être intéressant de considérer les interpolations entre les primitives virtuelles qui sont projetées sur le plan image des yeux de l'humanoïde. De cette manière, il est possible de disposer de plus de contrôle sur l'allure générale du mouvement associé à la réalisation de la tâche, et même d'envisager des phases de planification sur les trajectoires articulaires. Dans ce cas, on lève aussi le problème de savoir si la primitive est devant ou derrière le plan image. A titre d'exemple, on peut considérer la transition entre deux sphères. Cette interpolation nous garantit qu'à aucun moment la cible se situera derrière le plan image. Cependant une mauvaise interpolation spatiale peut amener l'humanoïde dans une configuration indésirable (limites cinématiques) ou vers des singularités. Pour pallier à ce problème, le choix des fonctions d'interpolation est primordiale (cf Fig. 3.14). Dans la figure 3.14 (a), l'interpolation linéaire des deux sphères génère un objectif irréaliste (car traversant le corps de l'humanoïde). Il est possible de résoudre ce problème en définissant une fonction d'interpolation respectant une zone interdite autour de l'humanoïde. Pour cela, on conserve une interpolation linéaire mais que l'on effectue sur les coordonnées polaires des centres des sphères (exprimées dans le repère de l'humanoïde). La trajectoire ainsi obtenue est circulaire (voir Fig. 3.14 (b)), et admet comme centre le centre de l'humanoïde. On note que cette méthode ne marche pas systématiquement (exemple : l'humanoïde se déplace en même temps, ou la cible est hors d'atteinte de l'humanoïde). Il faut dans ces cas considérer une phase de planification. Cet aspect n'a pas été considéré dans cette thèse, mais fait partie des perspectives de travail.

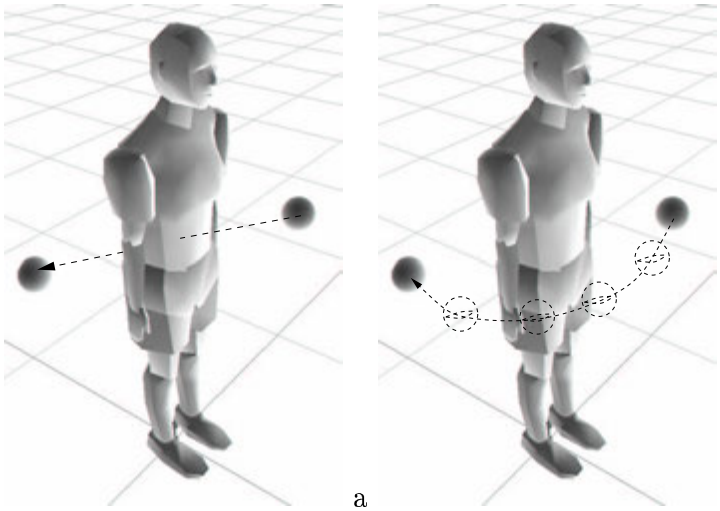


FIG. 3.14 – *Choisir une bonne interpolation spatiale (a) mauvaise (b) bonne*

3.4.3 Résultats

L'étude des techniques de transition s'est effectuée sur un exemple simple (voir Fig. 3.15) : un humanoïde cherche à toucher les trois boules rouges. Avant de toucher ces boules, l'humanoïde les observe. Le système de transition entre les sphères est celui décrit dans le paragraphe précédent. Plus précisément, le type de transition est ici d'une sphère à une autre sphère. L'interpolation réalisée est une interpolation spatiale sur les coordonnées polaires des deux sphères (exprimées dans le repère de l'humanoïde). On constate qu'entre la figure 3.15(c) et 3.15(d), la dernière sphère cible est passée derrière le plan image. Sans le système d'interpolation, l'humanoïde n'aurait pas pu regarder la dernière sphère, car sa projection sur le plan image ne correspondrait à aucune réalité. La figure 3.16 montre l'évo-

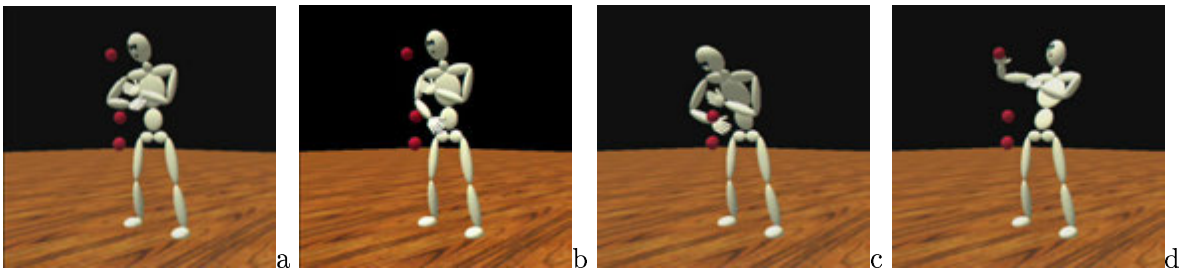


FIG. 3.15 – *Transition entre deux tâches de même nature (asservissement sur une sphère) avec une interpolation spatiale correcte*

lution de l'erreur moyenne dans l'image au cours de l'animation. La première tâche n'étant pas une transition entre deux configurations, son erreur décroît exponentiellement. Lors des deux transitions suivantes, on constate que l'erreur reste faible (on est toujours près de l'objectif) ainsi que constante ; en effet elle représente une erreur de traînage introduite par la fonction d'interpolation.

Le deuxième exemple montre le passage d'un point à une cible (voir Fig. 3.17). L'inter-

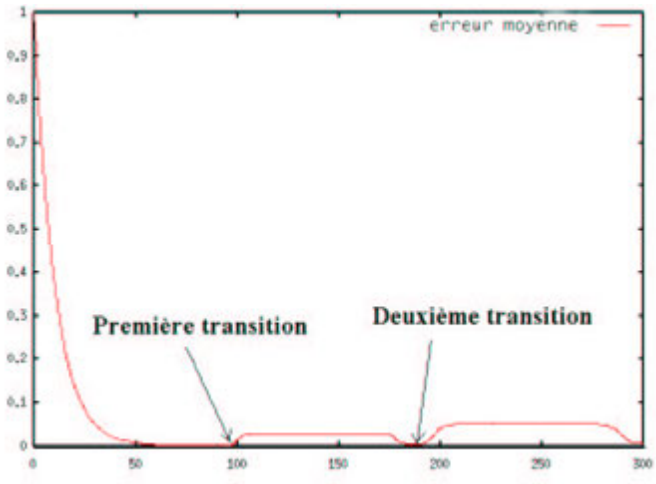


FIG. 3.16 – *Erreur moyenne sur l'ensemble de l'animation*

polation des cibles est réalisée dans l'image. Le temps de transition est fixé à une seconde, rendant la transition très rapide.

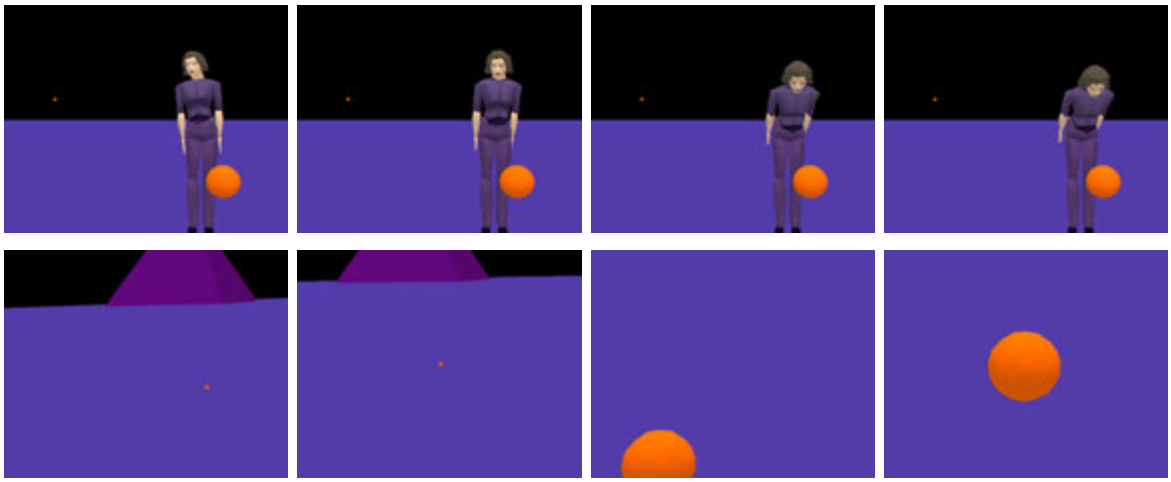


FIG. 3.17 – *Transition d'une primitive de type "point" à une primitive de type "sphère" (a) vue d'ensemble (b) image perçue par l'humanoïde*

3.5 Gestion du mouvement de l'humanoïde

Si l'on considère l'attention visuelle au niveau tâche, il apparaît que la simple cinématique du buste, de la tête et des yeux de l'humanoïde ne suffit pas. Plusieurs mécanismes sont mis en jeu, dont notamment la locomotion, ou d'autres actions comme s'accroupir, etc. Dans cette partie, nous nous intéressons plus particulièrement au problème de la locomotion associée à la perception. Pour pouvoir générer des déplacements, nous rajoutons des degrés de liberté en translation à la chaîne articulaire. Dans cette partie, nous étudions la manière de prendre en compte ces degrés de translation supplémentaires dans la conception de la loi de commande, et nous verrons comment les vitesses calculées sont converties en déplacement de l'humanoïde.

3.5.1 Modélisation de la tâche visuelle

Pour prendre en compte les déplacements de l'humanoïde relatifs à une tâche visuelle, des degrés de liberté en translation sont rajoutés à la racine de la chaîne articulaire considérée. Ces translations correspondent à la marche de côté et la marche normale (avancer et reculer) ; elles peuvent aussi représenter des mouvements plus complexes : se baisser ou grimper dans le cas des translations verticales. A chaque pas de temps, ces vitesses sont recalculées, et interprétées par le module de locomotion (section suivante).

Nous avons choisi de modéliser le déplacement de l'humanoïde comme une tâche secondaire. Grâce aux opérateurs de projection $\mathbf{W}^+$ et $\mathbf{I}-\mathbf{W}^+\mathbf{W}$ de l'équation (3.6), cette tâche secondaire n'a aucun effet sur la tâche de positionnement de base. A titre d'exemple, nous proposons une tâche secondaire possible permettant de maintenir une distance constante d entre l'humanoïde et l'objet d'intérêt. A chaque instant t cet objet se trouve à une position $\mathcal{V}(t) = (x(t),y(t),z(t))$ dans le repère de l'humanoïde. La tâche secondaire est définie comme une fonction de la distance entre l'humanoïde et l'objet d'intérêt $\mathcal{V}(t)$. La fonction peut être donnée par :

$$h_s = \frac{1}{2}\|d - (\mathcal{V}(t) - \mathcal{H})\|^2 \tag{3.25}$$

où $\mathcal{H}$ est la position de l'humanoïde. $\mathbf{e_2}$ est alors définie par :

$$\mathbf{e_2} = \begin{pmatrix} x(t) & y(t) & z(t) & 0 & 0 & 0 \end{pmatrix}^T h_s \tag{3.26}$$

Notons qu'à cause des contraintes inhérentes au mouvement de marche, le mouvement attendu de l'humanoïde ne correspond pas exactement aux vitesses générées par la loi de commande (c.a.d, si on donne une vitesse T_x à l'instant t et que l'humanoïde se trouve à une position X, l'humanoïde ne se trouvera pas exactement à la position $X + T_x$ à l'instant $t + 1$). Le fait de marcher provoque des mouvements dans tout le corps. De ce fait, la marche est considérée comme une perturbation à l'égard de la loi de commande. Disposer d'un modèle de cette perturbation permettrait de supprimer complètement cette perturbation dans la loi de commande. Nous n'avons pas considéré cet aspect dans le cadre de ce document.

3.5.2 Gestion de la locomotion

Nous avons utilisé un module de locomotion développé par Ménardais [Menardais 03]. Cette action peut gérer plusieurs types de marche (avant, arrière et sur le coté) issues de

mouvements capturés. L'utilisateur peut choisir dans ce formalisme la vitesse et la direction (locale à l'humanoïde) du déplacement. Ce vecteur de paramètres découle directement de la loi de commande. D'autres paramètres permettent de choisir et moduler différents types de marche (normale, course, dépitée, joyeuse, etc.), et ce de manière dynamique (fluidité des transitions et cohérence spatiale). Lors de l'exécution du module de locomotion, la direction globale de l'humanoïde est conservée. Les jambes et la partie supérieure du corps de l'humanoïde sont découplées, ce qui permet de contrôler la partie supérieure de l'humanoïde sans changer la direction du déplacement, en respectant cependant ses contraintes articulaires.
Pour générer un déplacement correspondant aux vitesses calculées dans la loi de commande, il faut pouvoir mélanger différents types de marche ou de course. Chaque mouvement élémentaire contient une séquence de pas de la forme p_1,p_1p_2 ou $\emptyset$, p_2 , p_1p_2 ou $\emptyset$, où p_1 (resp. p_2) correspond à un contact du pied droit (resp. gauche) avec le sol (p_1p_2 signifie que les deux pieds sont en contact avec le sol et $\emptyset$ l'inverse). Soient les quatre directions cardinales $\mathbf{v_1}$,$\mathbf{v_2}$,$\mathbf{v_3}$ et $\mathbf{v_4}$. Pour un vecteur $\mathbf{v_u}$ définit par l'utilisateur on a :

$$\exists i,j \quad \mathbf{v_u} = K(k\mathbf{v_i} + (1-k)\mathbf{v_j}) \tag{3.27}$$

Soient les mouvements de locomotion M_i et M_j issus de la capture et respectant respectivement les directions v_i et v_j. Dans un premier temps ces deux mouvements sont mélangés pour respecter les paramètres de l'utilisateur. Ce mélange pondéré s'opère par interpolation des quaternions associés aux mouvements. Il considère aussi la durée de chaque mouvement Δt_i^s (resp. Δt_j^s) et le déplacement de la base de l'humanoïde Δp_i^s (resp. Δp_j^s) conformément aux séquences de pas de chacun des deux mouvements. Le mouvement final est obtenu par un mélange final :

$$M = kM_i + (1-k)M_j$$

La vitesse moyenne v de ce mouvement est calculée à partir des durées des mouvements M_i et M_j de la manière suivante :

$$v = \frac{\Delta p}{\Delta t} = \frac{\sum_{s=1}^{4} \Delta p^s}{\sum_{s=1}^{4} \Delta t^s} = \frac{\sum_{s=1}^{4} k\Delta p_i^s + (1-k)\Delta p_j^s}{\sum_{s=1}^{4} k\Delta t_i^s + (1-k)\Delta t_j^s} \tag{3.28}$$

Si v et $|\mathbf{v_u}|$ sont différents, il faut réadapter le mouvement obtenu pour assurer que le mouvement final ait bien la vitesse du mouvement désiré. Cette phase est réalisée à l'aide de déformation temporelle (ou *time warping*). Les séquences de pas sont mises à l'échelle grâce au scalaire K présent dans l'équation 3.27.

$$K = \frac{v}{|\mathbf{v_u}|} \qquad \text{alors } \forall\ s = 1..4 \qquad \Delta t_{final}^s = \Delta t^s Kc$$

Le mouvement final respecte la vitesse générée par la loi de commande, ce qui permet d'assurer la tâche visuelle. On note que cette adaptation se fait de manière dynamique, et respecte donc les vitesses envoyées à chaque pas de temps.

3.5.3 Résultats

Nous avons étudié la validité cette approche au travers de l'exemple suivant : la tâche de vision donnée à l'humanoïde est de maintenir une certaine distance entre ses yeux est une balle qui se déplace. Le mouvement de cette balle est perpendiculaire à la direction de l'humanoïde. La primitive sur laquelle on s'asservit est un point situé au centre de la balle. Lors de la première phase de l'animation (figure 3.18(a,b,c)), l'humanoïde doit avancer vers la balle pour assurer la contrainte de distance, ainsi que jouer sur les rotations associées au cou et aux yeux pour la tâche d'observation. Comme l'objet d'intérêt bouge constamment, le module de locomotion doit adapter le mouvement : dans les figures 3.18 (d,e,f) l'humanoïde adopte une marche en pas chassé. La transition entre la marche avant et les pas de coté s'est effectuée de manière fluide au moment où l'humanoïde arrive près de la balle. Cet exemple montre comment le module de locomotion est asservi au contrôle du regard de l'humanoïde.

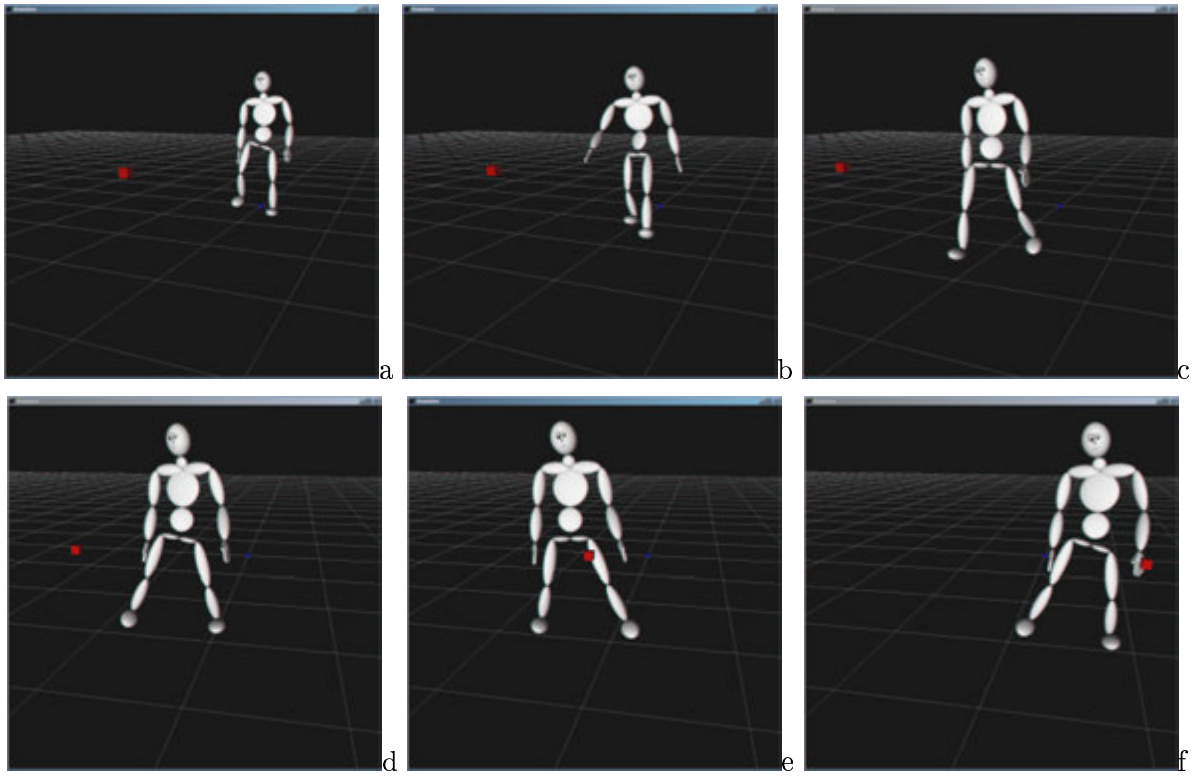


FIG. 3.18 – *Regarder de près un objet qui bouge: (a,b,c) l'humanoïde se rapproche de la balle (d,e,f) pas de coté pour maintenir la distance souhaitée*

La figure 3.19 est un autre exemple étayant notre approche. Dans ce cas la chaîne articulaire considérée comporte douze degrés de liberté. Dans cet exemple, seul un asservissement sur une sphère est considéré. Pour éviter que l'humanoïde ne se penche en début d'animation (pour se rapprocher de la sphère), on ne considère que les commandes en translation au début du mouvement. Lorsque l'humanoïde est suffisamment proche de la sphère, tous les vitesses calculées sont utilisées pour réaliser la tâche.

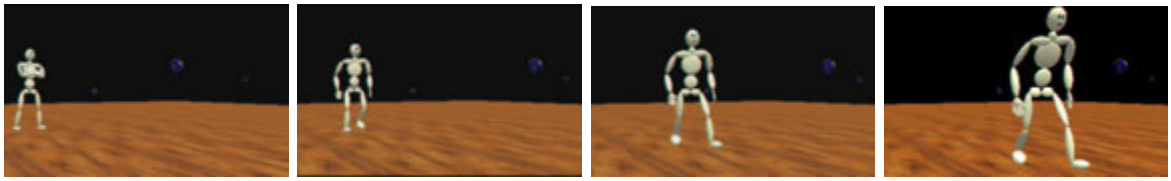


FIG. 3.19 – *Exemple de gestion de la locomotion avec une chaîne articulaire comportant douze degrés de liberté (dont trois en translation)*

3.6 Réalisme visuel des animations : vers une posture de moindre effort

L'utilisation du modèle proposé pour animer un humanoïde virtuel pose forcément le problème du réalisme : le fait de respecter une contrainte visuelle ne garantit pas le réalisme du mouvement. Ainsi la loi de commande utilise indistinctement tel ou tel degré de liberté pour assurer la contrainte imposée, alors qu'un opérateur humain opérera différemment et choisira (consciemment ou inconsciemment) le geste de moindre effort (dans la limite de ce que sa tâche visuelle lui impose). Il faut remarquer que le choix de la contrainte visuelle est déjà une manière de contraindre le mouvement dans un but de réalisme : préférer une droite verticale à un point comme primitive d'intérêt permet d'éviter que la tête ou le corps ne se penchent trop latéralement, ce qui est un mouvement (même si il est mécaniquement possible) pour le moins peu courant. Dans cette partie nous allons voir comment il est possible d'exprimer ce type de contrainte dans la loi de commande, dans le but d'obtenir des mouvements plus réalistes. Dans un premier temps, nous étudierons quels sont les couplages particuliers que peuvent contenir les chaînes cinématiques relatives à la gestion de l'attention visuelle d'un humanoïde, puis nous proposerons une méthode basée sur la redondance permettant de se rapprocher au cours de l'animation de la posture dite de "moindre effort".

3.6.1 Couplages particuliers

Dans la loi de commande considérée jusqu'alors, tous les degrés de liberté sont considérés de la même manière, et évoluent donc de manière équivalente au cours de la réalisation de la tâche. Cependant, des chercheurs en neurophysiologie ont prouvé l'existence de couplages particuliers dans les processus contrôlant les mouvements de la tête et les mouvements des yeux [Morasso 86]. Ainsi, dans le cadre du suivi du regard d'une cible, il existerait une correspondance géométrique linéaire entre les mouvements de la cible et la variation des valeurs articulaires de la chaîne cou-tête-yeux. De plus, l'être humain est pourvu d'un réflexe spécial appelé réflexe vestibulo-oculaire (RVO), qui lui permet de maintenir l'orientation de son regard en dépit de mouvements volontaires ou involontaires de la tête. Lors d'une tâche de focalisation, les yeux bougent rapidement pour se centrer sur l'objet désiré, alors que la tête, plus lourde (donc possédant une inertie plus grande), bouge pour maintenir la ligne de vue perpendiculaire à la ligne d'horizon (ce mouvement a pour effet d'augmenter la précision de l'évaluation des distances par notre système stéréoscopique). Les petits mouvements rapides des yeux à l'origine du mouvement sont appelés *saccades*. D'une manière générale, il faut donc pouvoir privilégier les rotations des yeux en

début de tâche (en leur associant un gain plus fort), puis les faire revenir à une position médiane, selon le principe du RVO.

3.6.2 Gestion des postures de moindre effort

L'approche considérée dans [Boulic 96] présente la notion de posture de moindre effort comme une contrainte forte pour la notion de réalisme. La posture debout représente une posture où l'humanoïde fait le moins d'effort (quand il ne peut ni s'asseoir ni s'allonger), à l'égard du travail musculaire et de contraintes biomécaniques et physiologiques [Kuo 93]. L'idée est alors d'avoir une fonction secondaire permettant de converger vers cette posture de repos (si cela est possible). De plus, cette tâche est, si l'on ne considère qu'une tâche secondaire définie relativement par rapport aux yeux, cohérente avec le principe du RVO définit auparavant. Si on définit la posture de repos par $\mathbf{q^{repos}}$, une telle fonction de coût peut être modélisée par :

$$h_s = \alpha \frac{1}{2} \sum_i (q_i - q_i^{repos})^2 \tag{3.29}$$

La tâche secondaire se définit alors par :

$$\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{q}} = \alpha \begin{pmatrix} q_1 - q_1^{repos} \\ q_2 - q_2^{repos} \\ \vdots \\ q_n - q_n^{repos} \end{pmatrix} \tag{3.30}$$

où α est un scalaire permettant de réguler la force du rappel vers cette position de repos. Cette fonction de coût se rapproche de la méthode utilisée dans le cadre d'évitement des butées articulaires (section 3.3). On note que la qualité du résultat dépend de la définition de $\mathbf{q^{repos}}$. Dans la suite de ce document, la posture de repos considérée a été définie comme la posture où toutes les articulations sont en position médiane relativement à leurs butées articulaires.

3.6.3 Résultats

Nous avons défini dans un premier temps une tâche secondaire ne prenant en compte que la position des yeux (*i.e.* $\mathbf{q^{repos}}$ n'était défini que relativement aux yeux) dans le but de simuler le réflexe vestibulo-occulaire. Robinson [Robinson 64] a établi, relativement à des observations sur des patients, les liens entre les rotations des yeux et les rotations du cou relativement à l'observation d'une cible fixe (cf. figure 3.20 (a)). Nous avons comparé l'allure générale de ces courbes avec les résultats obtenus par simulation du réflexe vestibulo-occulaire (cf. figure 3.20 (b)). Les deux diagrammes montrent qu'il y a une forte ressemblance dans cette allure générale, ce qui nous permet de justifier l'emploi d'une telle technique pour augmenter le réalisme du mouvement.

Il était difficile de quantifier l'apport en terme de réalisme de l'ajout d'une fonction de coût relative à la posture de moindre effort sur une animation simple. Cependant, les résultats associés à une telle technique sont beaucoup plus explicites sur animation prolongée. Nous avons choisi comme tâche visuelle de faire suivre une cible mobile à un

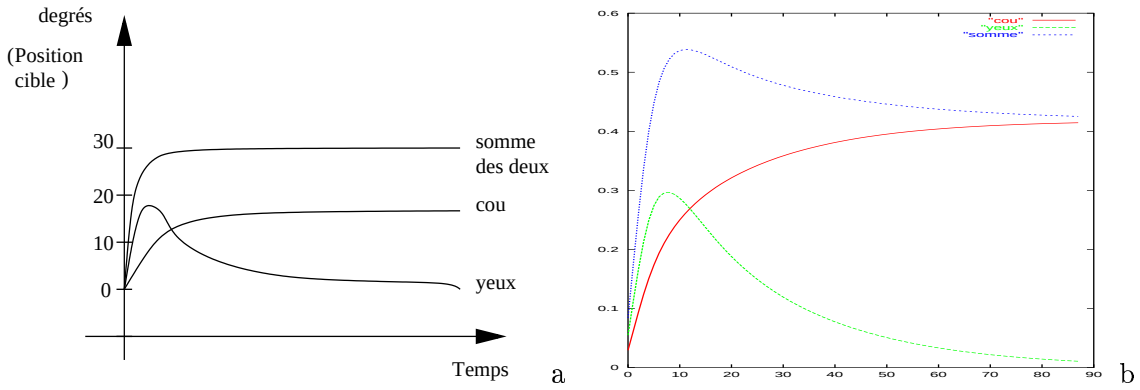


FIG. 3.20 – *Coopération cou/yeux pour le suivi visuel d'une cible. Comparaison entre l'étude de Robinson [Robinson 64] (a) et les résultats obtenus avec notre système d'animation (b)*

humanoïde. La chaîne articulaire considérée ne compte que cinq degrés de liberté (trois pour le cou, deux pour les yeux, cette configuration n'étant pas limitée). Au bout d'un certain temps, il existe une certaine divergence par rapport à la position de repos qui rend le mouvement non réaliste (cf. figure 3.21 (a)). Même si la tâche visuelle est toujours réalisée, le mouvement est dénaturé. L'ajout d'une fonction de coût relative à toute la chaîne et visant à se maintenir dans une posture de moindre effort permet de conserver un mouvement réaliste tout le long de l'animation, comme le montre l'image 3.21 (b) réalisée à la même itération que la précédente dans les mêmes conditions expérimentales mais avec utilisation de la tâche secondaire.

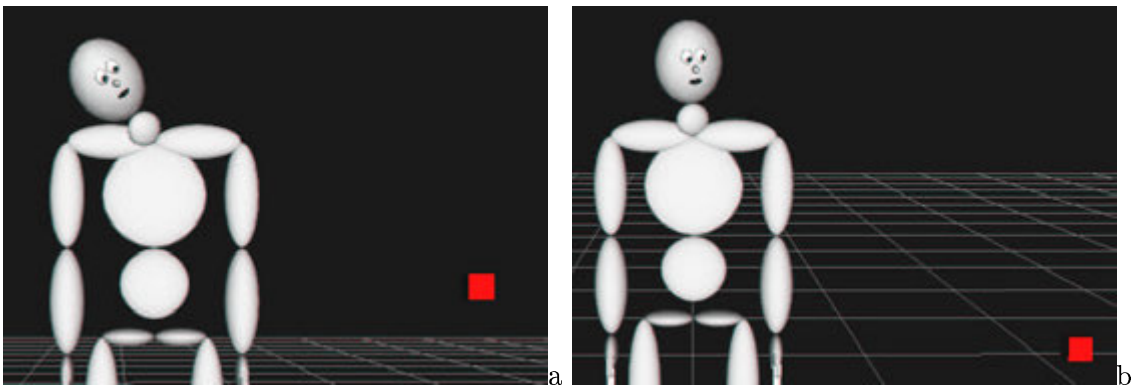


FIG. 3.21 – *Divergence au cours du temps, les deux captures d'écran sont prises à l'itération 3000 de la simulation (a) sans contrainte sur le réalisme (b) avec*

Perspectives. L'approche précédente ne se base sur aucun modèle dynamique pour établir une contrainte sur le mouvement. Une méthode d'optimisation globale de la posture via le contrôle du centre de masse a été proposée par Boulic [Boulic 96] et plus récemmement via les moments d'inertie dans [Baerlocher 00]. Il est envisageable que, lors de la réalisation d'une tâche, l'humanoïde adopte une posture, qui, même si elle est cohérente vis-à-vis de la tâche spécifiée, ne satisfasse plus la contrainte d'équilibre. Dans le but d'éviter de telles

configurations, il est envisageable de déterminer une fonction de coût visant à minimiser la distance entre la projection du centre de gravité de la partie supérieure de l'humanoïde (contrôlée par la loi de commande) et le polygone de sustentation. Le problème majeur étant de pouvoir dériver cette fonction selon les variables articulaires, il faut alors considérer les matrices d'inertie relatives aux différents parties animées de l'humanoïde. En effet à partir de ces matrices, il est possible de quantifier le déplacement du centre de gravité de la chaîne considérée relativement aux variations de la chaîne articulaire. Ce point fait partie des perspectives pouvant améliorer le réalisme des animations générées. D'autres approches sont aussi envisageables comme l'utilisation de mouvements capturés pour contraindre le mouvement de la chaîne [Tak 00,Monzani 00]. Dans ce cas la fonction de coût est alors définie comme la distance de la configuration de la chaîne articulaire à la configuration correspondante du mouvement capturé.

3.7 Conclusion

Le thème de ce chapitre était la gestion de l'attention visuelle d'un humanoïde de synthèse. Grâce à notre technique d'animation référencée image, nous avons pu contrôler la chaîne cinématique impliquée dans le processus de perception visuelle. La modalité de contrôle est ici, de même que pour les caméras, une tâche visuelle, exprimée dans l'espace 2D de ce que "voit" l'humanoïde. L'utilisation des degrés de liberté non impliqués dans la tâche visuelle nous a permis de nous intéresser à des problèmes relatifs à l'animation d'un humanoïde de synthèse : évitement des butées articulaires, enchaînement de tâches, gestion de la locomotion comme partie intégrante des tâches de vision et augmentation du réalisme de l'animation. La plupart des résultats ont été obtenus par utilisation du principe de redondance et sont compatibles entre eux, il est ainsi possible de les combiner linéairement entre eux (comme cela a déjà été présenté dans la partie relative au contrôle de caméra) ou par un système de projection en cascade en fonction de la priorité associée à chacun d'eux. L'ensemble de ces fonctionnalités reste transparent vis-à-vis de l'utilisateur (une fois le réglages des différents paramètres effectué), et permet donc de contrôler l'humanoïde relativement à des contraintes visuelles sans avoir à se soucier des contraintes intrinsèques aux mouvements produits. Cette "abstraction" du contrôle est nécessaire si l'on considère des comportements plus complexes, comme c'est le cas dans la deuxième partie de cette thèse.

Conclusion partielle

Dans la première partie de ce document, nous nous sommes intéréssés à la mise en place d'une nouvelle technique d'animation, une animation "référencée vision". Nos contributions ont porté sur l'utilisation de cette approche sur deux domaines différents : le contrôle d'une caméra en environnement virtuel, et le contrôle de l'attention visuelle d'un humanoïde de synthèse. Les atouts de cette approche sont d'une part la **facilité de spécification** des tâches, et les capacités d'adaptation offertes par le principe de redondance. En effet, il a été possible de déterminer plusieurs niveaux de tâches : une **tâche de positionnement** relative à un objectif visuel (pouvant évoluer au cours du temps), et un **ensemble de tâches secondaires** permettant de résoudre des problèmes liés aux deux domaines étudiés. La résolution de ces différentes tâches pouvant s'effectuer sans contraintes sur le temps d'exécution, il existe un intérêt important de cette approche pour les applications temps-réel disposant d'environnements dynamiques. Cependant son utilisation est aussi envisageable dans des applications non temps-réel, car elle peut résoudre des problèmes non-triviaux, générant ainsi des ébauches et des résultats réutilisables hors-ligne. La deuxième partie de ce document est consacrée à l'étude de l'utilisation de cette technique dans des contextes de plus haut niveau où une abstraction du contrôle est obligatoire : l'animation et la simulation comportementales.

Deuxième partie

Animation référencée vision et comportements

Animation et simulation comportementales

L'animation et la simulation comportementales désignent l'automatisation des processus qui régissent les entités virtuelles, dans un but de reproduction ou de simulation des comportements d'opérateurs réels. Un comportement se définit, d'une manière générale, comme étant une manière d'être et d'agir habituelle ou occasionnelle relative à une entité dans un milieu donné. D'une manière plus pragmatique, et dans le domaine de l'informatique, le comportement est défini comme l'enchaînement cohérent d'un ensemble d'actions unitaires, ayant des effets sur le monde et modifiant éventuellement l'état interne de l'entité. Définir des comportements est une **abstraction du contrôle** telle que nous l'avons définie dans l'introduction de ce document. Cette abstraction devient obligatoire quand l'entité ne peut plus être guidée par un opérateur humain ; c'est le cas des applications temps-réel où, pour que l'entité réalise des actions complexes dans son environnement, il est nécessaire d'avoir intégré des modèles de son comportement. Lors de la définition du comportement, une des difficultés majeures est de pouvoir intégrer les différents mécanismes de bas-niveau traduisant les actions ou les variations des états internes des entités. Dans la partie précédente, nous nous sommes intéressés à la définition d'une technique d'animation référencée vision. L'intérêt associé à l'utilisation d'une telle technique réside en deux points : la réactivité (capacité d'adaptation à des changements de l'environnement) et la facilité de spécification de tâches. Le but de cette partie est d'étudier comment cette technique peut d'une part être interfacée avec des modules de plus haut niveau gérant la notion de comportement, et quelles sont les capacités offertes par celle-ci. Ainsi la notion de tâche visuelle sera étudiée au travers de deux grandes classes de problème, directement liés aux thématiques abordées dans la première partie de ce document : le contrôle d'une caméra et de la cinématique associée à la perception visuelle d'un humanoïde de synthèse. Les extensions au niveau comportemental de ces deux domaines sont la cinématographie virtuelle et la simulation de la perception visuelle d'un humanoïde. Ces deux domaines seront donc étudiés au travers d'une base théorique développée dans le premier chapitre de cette partie.

Chapitre 4

Techniques d'animation et de simulation comportementales

Ce chapitre présente un état de l'art sur les techniques d'animation et de simulation comportementales. Dans le but d'établir un système de cinématographie automatique, ainsi qu'une architecture simulant la perception visuelle d'un humanoïde de synthèse, il faut pouvoir être en mesure de modéliser des comportements. Ce domaine, très large, comporte de nombreuses approches qui se focalisent sur des points précis de la modélisation des comportements. La première partie de ce document est un rapide aperçu de ces différentes méthodes. La deuxième partie est une présentation du formalisme retenu pour modéliser les comportements : HPTS (*Hierarchical Parallel Transition Systems*).

4.1 Etat de l'art sur la modélisation de comportements

La modélisation des comportements représente une étape vers la définition d'entités autonomes. On qualifie une entité d'autonome lorsqu'elle est munie de processus de perception, de décision et d'action, ce découpage étant à l'origine de la plupart des théories sur le comportement [Mallot 97,Thomas 99] :

- *perception* L'acteur autonome perçoit son environnent volontairement ou non, et en retire des informations utiles au processus de décision. Deux types de perception sont possibles : une perception *proprioceptive*, qui le renseigne sur son état physique interne ainsi que sur sa position dans l'espace, et une perception *extéroceptive*, qui lui fournit des informations extérieures (c'est en règle général l'aspect le plus étudié),
- *décision* C'est ce processus qui détermine les actions à entreprendre. Il est constitué de deux niveaux en perpétuelle interaction : un niveau réactif, qui détermine un ensemble de réactions directement liées à la perception (réflexes, etc.), et un niveau cognitif (ou rationnel), où la notion de mémoire, de raisonnements ou de planification rentrent en compte,
- *action* Ces mécanismes désignent toutes les manières dont l'entité dispose de manière à modifier son environnement. De manière identique à la perception deux types d'actions sont envisageables : des actions *proprioeffectives* qui modifient l'état interne de l'humanoïde, et des actions *extéroeffectives* qui ont une conséquence directe sur l'environnement.

La figure 4.1 indique les différentes relations entre ces modules. La plupart des modèles

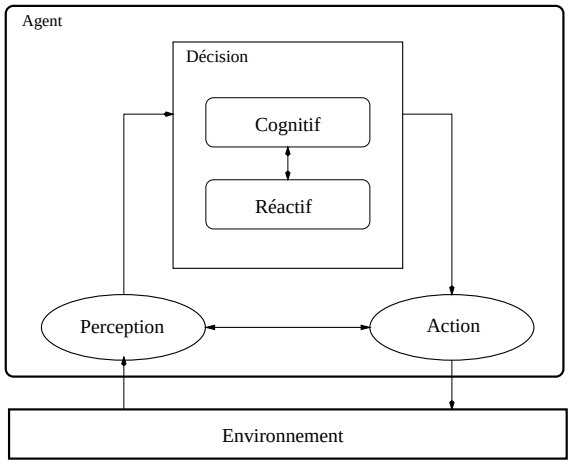


FIG. 4.1 – *Boucle Perception–Décision–Action pour un acteur synthétique*

comportementaux existant intègrent cette notion de boucles perception–décision–action, mais dans différentes proportions. La gestion de la brique action est réalisée par les techniques d'animation déjà évoquées dans la première partie de ce document. Les modèles de perception font l'objet d'une étude plus approfondie dans les parties suivantes. Nous nous intéressons conséquemment plus en détail dans cette partie aux différents modèles décisionnels existants. Deux grandes classes d'approches sont étudiées ; l'une décrivant des modèles simples, très liés à l'aspect réactif du modèle décisionnel, et une autre basée sur les automates, mêlant des aspects cognitifs et réactifs.

4.1.1 Modèles comportementaux simples

Cette classe d'approches regroupe les systèmes purement réactifs, c'est à dire ne réagissant qu'à partir de l'état dans lequel le système se trouve (pas d'utilisation de connaissances).

Approches stimuli–réponses L'approche stimuli–réponses provient des travaux développés en éthologie (comportements des animaux) [Brooks 91,Beer 91]. On définit le comportement comme étant le résultat direct d'une série d'interactions avec l'environnement. Des graphes de noeuds sont définis liant les capteurs aux effecteurs. L'information est modifiée au travers du passage dans ce réseau. La modification des poids des liaisons ou des fonctions associées à chaque noeud peut se faire par apprentissage (dans ce cas on utilise souvent des fonctions de renforcement). On peut citer à titre d'exemple les boucles SCA (*Sense–Control–Action*) de Badler et Webber [Badler 95], qui utilisent entre autre ce type de graphe pour superviser la locomotion. Dans le cadre de l'animation d'humanoïde, ce type de méthode est à la croisée du contrôle de mouvement et de l'animation comportemental, car le niveau d'abstraction est très faible. Un des principaux défauts de ce genre d'approches et de n'offrir qu'une souplesse très faible, car la modification d'une réaction à un stimulus ne s'opère que par reconfiguration totale du réseau, et pour un grand nombre de stimuli en entrée la convergence vers la réaction attendue n'est plus garantie.

Plusieurs autres approches ont été développées pour la simulation de comportements portant sur l'apprentissage de comportement, grâce à des algorithmes génétiques par exemple [Sanza 99]. Ces approches sont très souvent assez spécifiques à un problème donné, et l'utilisation des méthodes d'apprentissage (réseaux de neurones, classification) ne garantissent pas toujours la convergence vers le comportement souhaité (ceci est d'autant plus vrai si les comportements désirés sont complexes). L'intérêt de ce genre d'approches est de modéliser la connaissance sous une forme non formelle (boite noire), cette abstraction étant une boite noire définie par les paramètres intrinsèques du réseau.

Approche à base de règles Le comportement est ici modélisé sous la forme d'un ensemble de règles, représentant un arbre de décision qui peut être plus ou moins complexe. Chaque comportement élémentaire représente un noeud de cet arbre. Le choix du comportement s'effectue en fonction de pré-conditions sur l'environnement (exemple : "si il pleut, prends un parapluie"). La majeure limitation à cette modélisation est le fait qu'il faut établir par avance toutes les règles correspondant à toutes les situations possibles, et que ces définitions ne changent pas en cours d'exécution (pas robuste à un changement d'environnement). Le travail de Reynolds [Reynolds 87] sur les nuées d'oiseaux ou les bancs de poissons traduit bien cette approche. Ainsi les oiseaux sont régis par trois règles : *pas de collisions avec les voisins, cohérence entre sa propre vitesse et celles des voisins, rester proche de ses voisins.* La nuée est alors animée en changeant le point cible vers lequel se dirigent les oiseaux. On note à ce niveau que le comportement émergent est alors un comportement de groupe, en effet un individu seul n'aura aucun comportement autonome.

Blumberg [Blumberg 95] utilise aussi une approche à base d'arbres de décision pour modéliser le comportement d'un chien nommé Sylas. Dans cet arbre les noeuds sont des experts et les feuilles des actions. L'expert sert à déterminer vers quel fils descendre dans l'arbre. L'expression de la décision en utilisant des règles est intuitive et d'un niveau d'abstraction satisfaisant. Cependant, il semble difficile de modéliser des comportements complexes réalistes comme un ensemble de règles ; de plus l'ajout d'une nouvelle règle peut générer des conflits (inhibition ou incomplétude) nécessitant l'ajout de mécanismes surveillant la cohérence de l'ensemble des règles. Notons aussi que certaines fonctionnalités sont absentes, notamment le parallélisme entre plusieurs actions.

4.1.2 Modèles comportementaux comportant les aspects réactifs et cognitifs

Nous présentons ici des approches de modélisation des comportements alliant dans une mesure plus ou moins grande l'aspect réactif et l'aspect cognitif. L'aspect cognitif[1] désigne l'utilisation et l'acquisition de connaissances utiles au modèle décisionnel. Ainsi l'action effectuée dépend à la fois du contexte de l'entité et d'un ensemble de connaissances acquises avant ou pendant la simulation. Il est intéressant à ce stade d'étudier les modèles de comportement humain établis par les psychologues et les biologistes ; Newell, dans une tentative d'unification de tous ces travaux [Newell 90], s'attache à déterminer les critères essentiels à la modélisation de comportements. A la lumière de ces pré-requis nous décrivons

1. **Cognition** *: utilisation ou manipulation de la connaissance. [Gregory 93]*

l'utilisation d'automates pour la modélisation de comportements puis deux langages dédiés à la modélisation du raisonnement.

Modélisations de la cognition Les processus cognitifs sont sans doute les processus les plus complexes observables chez un être humain. Dans l'optique de la simulation comportementale, Newell [Newell 90] s'attache à définir des contraintes à respecter pour obtenir une modélisation de la cognition proche de celle de l'humain :

- flexibilité et robustesse du comportement. L'acteur autonome doit pouvoir s'adapter à des changements d'environnements,
- interactions temps-réel avec son environnement,
- richesse et complexité dans les actions et les connaissances,
- disposer de mécanismes d'abstraction (usage de symbolisme),
- apprentissage, réitérer des solutions déjà éprouvées,
- autonomie partielle ou vie en société, l'acteur doit aussi être soumis à des lois de groupe,
- conscience de soi et perception, l'acteur doit pouvoir remettre en question ses propres buts, et s'extérioriser (grâce à la perception) vis-à-vis de son environnement,
- évolution, à la fois de sa propre architecture mais aussi de son espèce.

Les connaissances actuelles en informatique et l'avancement des réflexions éthiques et philosophiques ne nous permettent pas encore d'obtenir un modèle respectant chacune de ces contraintes (et ne le permettront peut-être jamais). Cependant, la modélisation du module décisionnel doit disposer du maximum de ces fonctionnalités. Les automates parallèles hiérarchisés respectent une partie de ces contraintes : spécification modulaire du comportement, prise en compte de différents niveaux d'abstraction au sein d'une même entité, et exécution de comportements concurrents. La partie suivante expose les approches à base d'automates.

4.1.2.1 Approches à base d'automates

Un automate à états finis est une manière de décrire les comportements. Il définit un ensemble d'états et les transitions qui permettent le passage d'un état à l'autre [Koga 98]. Dans un automate à états finis chaque état est exclusif, c'est à dire qu'on ne peut pas se trouver dans deux états à la fois. Les transitions entre ces états se font en réponse à des stimuli externes ou internes. Si l'on souhaite gérer les conditions de transition entre chacun de ces états et si l'on dispose de N états et M transitions, alors il faut évaluer $N \times M$ cas. Quand N et M deviennent grands, la complexité devient rapidement ingérable. Une approche pouvant résoudre ce problème est l'utilisation d'automates parallèles hiérarchisés (voir Fig. 4.2).

Plusieurs formalismes utilisent ces modèles pour représenter des comportements :

- *PaT-nets* ou *Parallel Transition Networks* développés par Badler [Badler 95]. Ce formalisme a été établi dans le cadre de la simulation d'un humanoïde autonome :"Jack". Les *PaT-nets* modélisent les capacités décisionnelles (complétées par les *SCA* présentées auparavant) de l'humanoïde sous forme d'automates d'états finis. Ceux-ci peuvent s'exécuter en parallèle et ont la capacité d'échanger des messages (via des systèmes de sémaphores). Chaque noeud représente des comportements directement

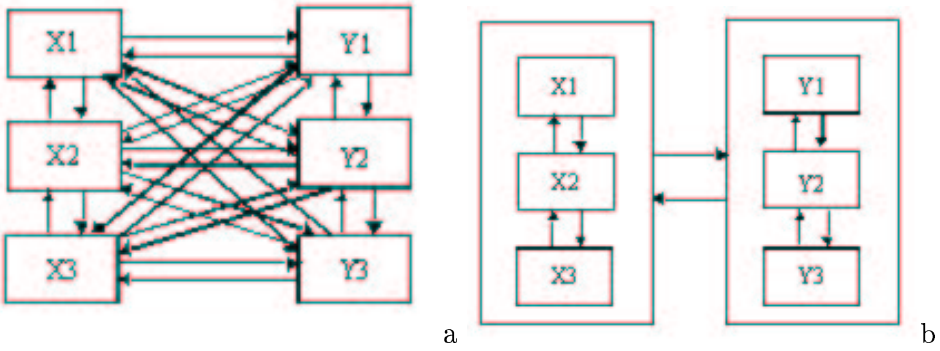


FIG. 4.2 – *Relation automate à états finis et automates parallèles hiérarchisés (a) simple (b) hiérarchisé (extrait de [Koga 98])*

exécutables (*SCA*), d'autre *PaT-nets*, ou des modules spécialisés. La complémentarité des *PaT-nets* et des *SCA* a été validé dans un exemple d'un exemple de jeu de cache-cache : cette activité nécessite des prises de décision impossibles à modéliser au travers des *SCA*. Les *PaT-nets* organisent alors et contrôlent les *SCA* qui a leur tour activent les capacités motrices. La figure 4.3 montre le *PaT-net* associé à ce comportement,

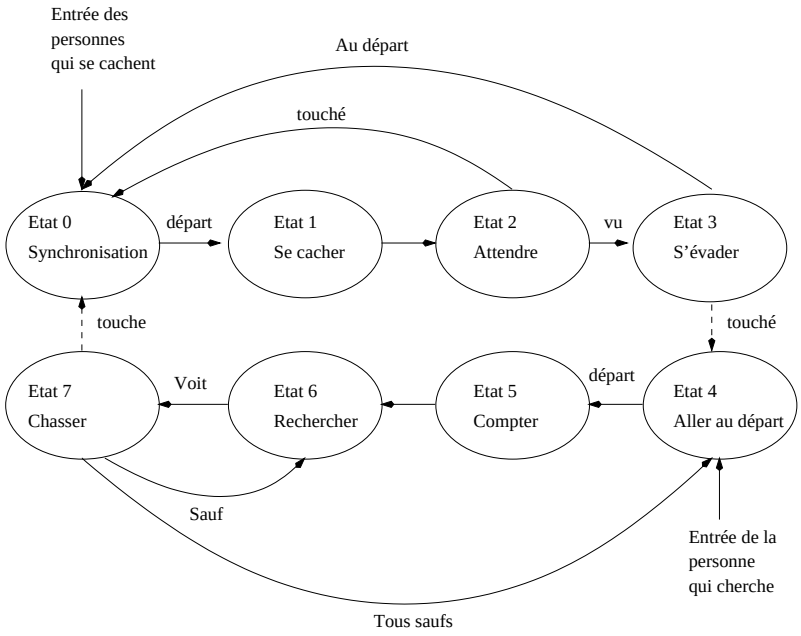


FIG. 4.3 – PaT-net *décrivant le comportement d'un jeu de cache-cache*

– *Pile d'automates.* Cette approche a été mise en œuvre par Noser [Noser 97]. Les actions, représentés par des automates, sont empilées par l'utilisateur en début de simulation. Le processus est hiérarchique, un automate pouvant avoir besoin de sous-automates pour s'exécuter. Lors de certains événements (extérieurs ou relatifs au scénario), les automates sont dépilés pour que s'exécute de nouvelles actions. Cette

approche a été validée dans le cadre de la simulation d'un jeu de tennis.

- *HPTS (*Hierarchical Parallel Transition Systems*)* C'est un langage développé à l'Irisa [Moreau 98a]. La particularité de ce modèle est d'inclure une gestion du temps (délais, durées minimales et maximales, fréquences). Une présentation plus détaillée est fournie dans la partie suivante.

Les approches à base d'automates représentent un pont entre l'aspect réactif et l'aspect cognitif ; en effet les transitions d'un état à un autre se font de manière purement réactive, mais la structure des automates représente une connaissance en soi, et leurs modes de conception et d'exécution font écho avec certains pré-requis établis par Newell. La partie suivante montre deux approches modélisant le raisonnement par le biais de langages.

4.1.2.2 Approches à base de langage

Les différents travaux portant sur la cognition pure s'organisent, dans le domaine de l'informatique, autour de langages de programmation permettant d'intégrer la notion de connaissance. Deux approches sont présentées dans ce document, l'une se nommant *Situation Calculus* et la deuxième *SOAR*.

Situation Calculus La théorie du Situation Calculus [Funge 98] permet de raisonner sur l'effet des actions dans un environnement donné pour effectuer des tâches de planification. En partant d'un état donné du monde, on génère l'ensemble des "mondes possibles" associés aux actions potentiellement réalisables. Pour extraire l'information d'une situation donnée (pour pouvoir en dire quelque chose), Funge définit la notion de *fluent*. C'est une fonction dont l'image est soit un booléen, soit une nouvelle situation. Le *fluent* peut être valué sur un intervalle, donnant par là plus de flexibilité à la manipulation des valeurs. Il définit aussi la notion d'action. Celle-ci n'est réalisable qu'à condition que sa pré-condition soit réalisée (portant sur les *fluents*) et spécifie ses effets sur la situation actuelle (résultats de l'action). Ainsi le Situation Calculus permet de raisonner sur des actions (présentées de manière abstraite) et sur leurs effets sur le monde. Il est alors possible de spécifier des buts à atteindre (comme faits sur les situations). Funge a développé un langage, CML, qui implémente cette technique. Ce langage a notamment la propriété intéressante d'être non-déterministe. Le principal défaut de cette approche est de perdre totalement la notion de temps-réel, dû au fait d'avoir à explorer tous les mondes possibles.

SOAR Le projet SOAR est une tentative de développement et d'application d'une théorie sur la cognition humaine et l'intelligence artificielle, établie par Newell [Newell 90]. La principale difficulté repose sur l'architecture du système, qui met en oeuvre des mécanismes sous-jacents compliqués qui cherchent à résoudre des problèmes de base (résolution, planification, apprentissage, représentation des connaissances, systèmes experts et modélisation psychologiques, etc.). Si en informatique, l'architecture d'un système correspond aux frontières entre le *hardware* et le *software*, ou d'une manière plus générique entre différentes couches d'un même système, en sciences cognitives ce terme existe par référence à "l'architecture" de l'esprit, c'est à dire les structures et relations entre les différents processus cognitifs. Le but de SOAR est donc double : travailler sur une architecture adaptée à une représentation artificielle de l'intelligence, et étudier l'esprit humain sous une approche cognitive. Depuis 1982, plusieurs versions de SOAR ont été développées, les cinq premières

étant en Lisp et les versions suivantes en C. SOAR, qui peut être considéré comme un langage de programmation pour intelligence artificielle, incorpore :

- gestion des espaces de problèmes dans une architecture unifiée, pour tout type de tâches et de sous-tâches qui doivent être résolues,
- production de règles comme unique représentation de la connaissance permanente,
- Manipulation d'objets valués et à attributs comme représentation de la connaissance temporaire ou *Working memory*,
- Génération automatique de sous-buts comme seul système de création de buts,
- *Data chunking* comme mécanisme d'apprentissage, c'est à dire création de connaissances (règles) par analyse de données dans la connaissance temporaire (mécanisme d'abstraction).

4.1.3 Rappel des objectifs et discussion

Cette étude sur les différents modèles de comportement vise à déterminer quelle approche est la plus adpatée pour modéliser les comportements relatifs à la cinématographie virtuelle et à la simulation de la perception visuelle d'un humanoïde. L'idée est de pouvoir offrir **un niveau d'abstraction supplémentaire à la notion de tâche visuelle**. Cette abstraction va se traduire par l'enchaînement d'un ensemble de tâches visuelles selon des contraintes relatives au domaine considéré. Les modèles à base d'automates possèdent des propriétés adaptées à ces besoins : concurrence, hiérarchie et gestion du temps. L'utilisation de modèle de raisonnement peut s'avérer nécessaire à plus long terme, mais l'utilisation d'automates pour modéliser des comportements semble être le choix le plus adapté en terme de conception, modularité et exécution. Ainsi, dans le cadre de cette thèse, nous avons utilisé HPTS, dont voici une présentation plus formelle.

4.2 Formalisme utilisé : HPTS

Dans cette partie nous présentons plus en détail le langage (HPTS) qui nous a permis d'instancier des comportements pour la caméra ou l'humanoïde. Après une description formelle du langage, nous parlerons des travaux plus récents sur les capacités d'adaptation et comment nous nous somme servis de ce langage.

4.2.1 Description

L'objectif ayant motivé la définition HPTS [Moreau 98a,Donikian 01] était d'établir un modèle permettant la définition de comportements adaptables et flexibles, relatifs à une entité évoluant dans un monde de synthèse pouvant être à la fois complexe et hautement dynamique (multiples modifications de l'environnement, interactions possibles nombreuses). Un tel système requiert l'utilisation de paradigmes de programmation : réactivité (ce qui comprend la gestion d'événements asynchrones et la gestion des exceptions), modularité des comportements (incluant la prise en charge du parallélisme entre les comportements, ainsi que la concurrence), flot de données (permettant la transmission de données d'un comportement à un autre), structuration hiérarchique des comportements (préemption) et gestion du temps et des fréquences d'exécution des différents comportements. HPTS correspond dans notre approche au modèle décisionnel. Il offre les outils nécessaires à la

création de comportements concurrents hiérarchisés. Ceux-ci représentent des modèles réactifs ou cognitifs, et peuvent être vus comme un système multi-agents, dans lequel chaque agent est une hiérarchie d'automates à états finis. Chaque agent peut être perçu comme une boite noire possédant des entrées-sorties permettant l'échange de données, un ensemble de paramètres de contrôle ainsi que des états internes. La synchronisation des agents est rendue possible grâce à l'utilisation d'automates. La représentation hiérarchique implique que chaque état du comportement est soit soit un état atomique, soit un état composé de sous-automates.

Un langage de description est utilisé pour décrire ces automates (voir [Moreau 98a, Donikian 01] pour plus de détails). A l'issu de la compilation des automates, du code en C++ est généré, pouvant s'exécuter sur notre plate-forme de simulation. Une version interprétée du langage a aussi été développée dans le but de pouvoir modifier et exécuter les comportements en temps réel (contre une légère augmentation du temps de calcul).

4.2.2 Coordination des comportements

L'être humain est capable de faire plusieurs choses à la fois, tout en maintenant la cohérence globale de ses comportements. La modélisation de comportements complexes nécessite une phase d'ordonnancement sur l'utilisation des ressources (qui peuvent être des parties du corps de l'humanoïde par exemple), ainsi que sur les priorités (qui découlent de la modification des paramètres internes de l'humanoïde). Une approche simpliste consiste à dire que deux comportements qui utilisent les mêmes ressources sont mutuellement exclusifs. Cette hypothèse est beaucoup trop restrictive pour la modélisation de comportements complexes, impliquant des adaptations du comportement. Une autre approche consiste alors à créer des comportements complexes incluant ces notions d'adaptation dans leur formalisme intrinsèque, mais cette approche force l'utilisateur à modéliser des comportements très compliqués, car il lui faut prévoir la totalité des cas possibles, ainsi que toutes les possibilités d'adaptation. De plus, il s'avère qu'une fois créés, ces comportements sont très dédiés, et rarement réutilisables. Ces limitations ont motivé l'ajout des priorités ainsi que des ressources dans les automates, utilisés par un algorithme d'ordonnancement qui s'exécute en temps-réel. Cette approche a été développée par Fabrice Lamarche [Lamarche 01,Lamarche 02]. L'attachement de ressources à chaque état (ou ensembles d'états) des automates permet une gestion implicite des ressources, qui sont alors considérées comme des sémaphores. L'existence de fonctions de priorité et de degrés de préférence permet à l'algorithme d'ordonnancement de choisir à un pas de temps donné les transitions qui satisfont au mieux l'ensemble des comportements. Ces fonctions de priorité peuvent être corrélées avec les variations de paramètres internes (physiologiques, psychologiques) ou externes. A l'aide d'un tel système chaque comportement peut être décrit de manière indépendante, avec ses propres capacités d'adaptation. Ainsi à l'exécution, les comportements s'ordonnent de manière dynamique, et donc de manière réactive. Nous allons voir dans les parties suivantes comment cette notion de ressources nous a servi pour l'établissement de comportements, que ce soit aussi bien en cinématographie virtuelle que pour la simulation de la perception visuelle d'un humanoïde de synthèse.

4.2.3 Lien avec notre système d'animation

L'utilisation d'automates à états finis dans le cadre du contrôle de robot autonome en robotique a fait l'objet de plusieurs études [Kosecká 94], spécialement dans le domaine des robots marcheurs [France 99]. Dans les systèmes que nous allons modéliser, chaque état correspond à l'exécution d'une tâche visuelle. Dans le but d'obtenir une synchronisation efficace, il est nécessaire de disposer d'un retour sur l'état de la tâche en cours (chez l'être humain on parle de *proprioception*). Ainsi les transitions pourront s'effectuer soit sur un événement extérieur, soit à condition que la tâche est réalisée. Pour définir l'état de la tâche, on mesure la "distance" au but final (souvent l'erreur dans l'image). En dessous d'un certain seuil, on considère que la tâche est "réussie". Le problème se complique lorsqu'il faut déterminer si la tâche est réalisable ou non. Il n'existe pas de méthodes à ce jour permettant à l'avance de déterminer cet état. Cependant il est possible de munir les systèmes d'animation d'un certain nombres de contrôles. Par exemple dans le cadre de l'humanoïde, si tous les axes permettant de réaliser la tâche se retrouvent en butées, on considère que celle-ci est impossible. Le concepteur des comportements a donc en charge de gérer ces différents cas de figures, et de gérer les adaptations associées. Nous proposons d'étudier la notion de comportement visuel sur un exemple concret réalisable à l'aide d'HPTS.

Etude de cas : observer tous les tableaux d'une pièce Cet exemple peut s'adapter à une caméra ou un humanoïde. Le but est de définir un comportement visant à observer tous les tableaux d'une pièce. La figure 4.4 montre l'automate réalisant ce comportement.

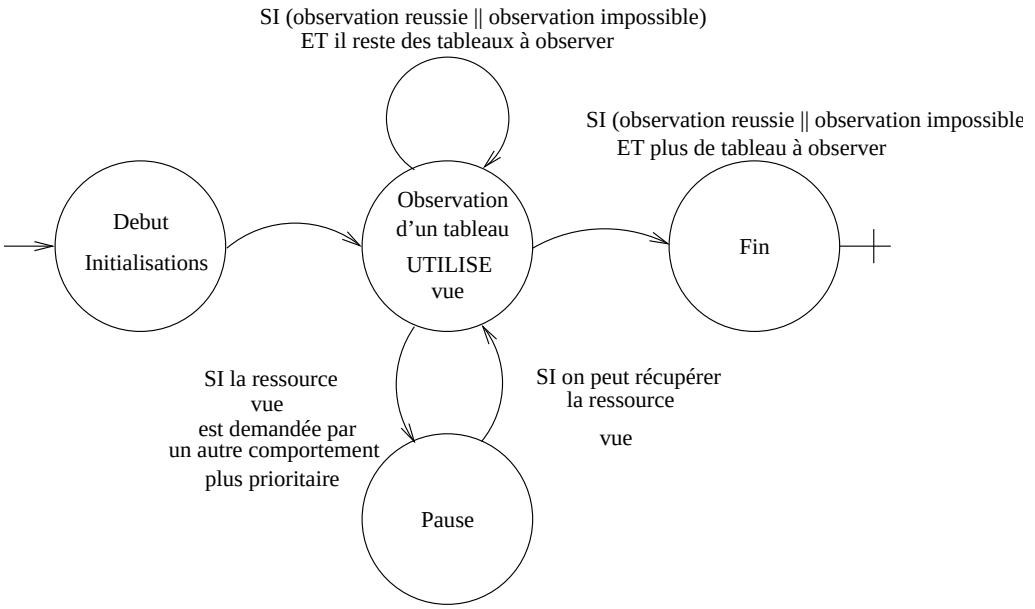


FIG. 4.4 – *Automate modélisant un comportement simple : observer tous les tableaux d'une pièce*

Un seul état réalise la tâche visuelle ; le comportement encapsule d'une certaine manière la réalisation de toutes les tâches visuelles conduisant à l'observation de tous les tableaux. L'abstraction du contrôle réalisée par le système d'animation permet de lier de manière

directe la signification de l'état (à l'intérieur du comportement) avec la tâche réalisée (au niveau animation) ; ce fait se traduit par une modélisation simple et intuitive du comportement. On constate que le retour sur l'état de l'action conditionne deux transitions sur cinq ; cette fonctionnalité est donc primordiale. Le système de coordination des comportements est présent dans l'automate ; l'état relatif à l'observation d'un tableau nécessite l'utilisation d'une ressource, la ressource *vue*. Son utilisation stipule que pour exécuter cet état, la caméra ou la chaîne articulaire de l'humanoïde est utilisée. Si un comportement plus prioritaire a besoin de prendre cette ressource, l'automate transite dans un état *pause*, relâchant ainsi la ressource. L'automate plus prioritaire peut alors s'exécuter, et prendre ainsi le contrôle de la caméra ou la chaîne articulaire de l'humanoïde. On note que cette représentation graphique de l'automate se traduit directement dans le langage de description associé à HPTS.

4.3 Conclusion

Dans le but de pouvoir modéliser des comportements, et ainsi réaliser la partie décisionnelle de nos agents, nous avons décidé d'utiliser le modèle HPTS. Cette approche respecte le formalisme décrit par Newell sur une architecture unifiée de la cognition. De plus, ce mode d'agencement des différentes tâches est particulièrement adapté à notre système d'animation : le formalisme automate nous permet de pouvoir enchaîner les différentes tâches selon des critères propres à la tâche, ou selon des critères extérieurs. L'utilisation du système de gestion de ressources et de priorités va nous permettre d'effectuer plusieurs comportements en parallèle tout en maintenant une cohérence globale du système. Les parties suivantes sont dédiés à la création des systèmes comportementaux relatifs à la cinématographie virtuelle ainsi qu'à la perception visuelle d'un humanoïde de synthèse, via l'utilisation d'HPTS. Nous allons ainsi voir comment le système d'animation "conditionne" grandement la conception de systèmes comportementaux, et quelles sont les facilités et les possibilités amenées par notre système d'animation.

Chapitre 5

Cinématographie virtuelle

La cinématographie virtuelle désigne la capacité à pouvoir choisir les successions de points de vue depuis lesquels les environnements virtuels en 3D sont rendus. En cela, de nombreuses analogies avec le cinéma sont réalisables ; la manière de montrer les choses contient en elle une intention : montrer des événements précis. De plus, cette manière sert souvent la narration, les mouvements de la caméra participent à créer une ambiance, à transmettre un contenu émotionnel. Depuis la création du septième art, les réalisateurs de films ont établi (de façon plus ou moins implicite) un ensemble de règles et de conventions permettant de transmettre les informations d'une manière compréhensible et efficace (voir le livre d'Arijon [Arijon 84] par exemple). Cependant, le niveau de description de ces règles et conventions n'est pas assez détaillé pour définir une grammaire formelle du langage cinématographique. De plus, il existe une latitude nécessaire employée par les auteurs de film, sans quoi la dimension artistique ne serait pas présente. Une autre différence majeure entre la cinématographie réelle et virtuelle est la contrainte de temps réel. Un réalisateur de film connaît à l'avance les différents éléments des scènes, peut disposer les acteurs à sa manière, et réaliser une phase de montage post tournage. A l'inverse, dans un système temps réel, il faut pouvoir au mieux rendre le contenu d'une scène à chaque instant sans en connaître les modifications futures. Une telle contrainte peut nous amener à percevoir la caméra comme un agent (représentant à la fois le cameraman et le metteur en scène). Cet agent doit ainsi être doté de moyen de perception (lui permettant de quantifier la qualité de ce qu'elle voit, d'avoir des objectifs visuels), d'action (se déplacer et s'orienter dans l'environnement), et enfin de décision (que filmer ? dans quelles conditions ?). Si l'on considère simplement le système d'animation décrit dans la première partie de ce document, seul le lien action-perception a été traité. L'objet de cette partie est de pouvoir rajouter un modèle décisionnel (HPTS) dans une optique de cinématographie virtuelle. Ainsi, nous allons voir, après un état de l'art sur la cinématographie virtuelle, comment réaliser des tâches élémentaires de positionnements cinématographiques, ainsi que les mouvements de caméra associés et connecter le module décisionnel au système d'animation, dans le but de définir un système réalisant automatiquement le tournage et le montage de **scènes dynamiques**.

5.1 Etat de l'art sur la cinématographie virtuelle

Plusieurs approches se distinguent dans la réalisation de systèmes de cinématographie virtuelle. Dans un premier temps, nous distinguons deux approches : les approches de post-production et les approches temps-réel, les contraintes établies dans ces deux domaines étant évidemment différentes. Dans un deuxième temps nous étudierons les approches traitant ce problème dans une optique d'animation comportementale, où le cameraman, et éventuellement le réalisateur, sont perçus comme des agents.

5.1.1 Temps-réel ou post-production ?

Il faut, à ce stade, distinguer deux approches : les approches temps-réel avec possibilité d'interactions, et les approches orientées post-production, où des algorithmes de planification hors-ligne sont utilisés. Dans cette dernière approche, Karp et Feiner [Karp 93] décrivent un système de planification permettant de montrer des présentations 3D générés automatiquement par l'ordinateur. Christianson [Christianson 96] a développé dans le cadre de la fiction interactive un planificateur des mouvements de la caméra basé sur des scripts d'animation. Ces approches ne sont pas adaptées au temps réel. A l'inverse, He [He 96] a développé un paradigme permettant le contrôle d'une caméra en temps réel appelé *the Virtual Cinematographer*, ce contrôle étant beaucoup plus proche du montage (au sens cinématographique du terme), que de la réalisation d'un plan. Le système créé repose sur l'existence d'un certain nombre de plans typiques ou idiomes organisés de manière hiérarchique, et génère une succession de ces plans avec un *timing* particulier dans le but de traduire au mieux l'ensemble des événements se produisant. Les règles cinématographiques sont encodées dans des automates à états finis. He prouve la validité de son approche dans une application de fête distribuée sur le réseau ; plusieurs acteurs se retrouvent dans une pièce et discutent entre eux, la caméra doit alors filmer ces conversations, selon le principe champ contre-champ. Les transitions peuvent aussi s'effectuer selon des critères de temps (durée minimale et maximale d'un plan). Chaque état de leurs automates font appel à un "module", qui gère les positionnements de la caméra pour différents plans. He est le premier à avoir transcrit des règles cinématographiques dans un formalisme informatique, les automates. Cette approche est donc liée très nettement avec la notion de comportement, car ces automates représentent une forme de modèle décisionnel. Un des défauts de cette approche est de ne considérer qu'un unique comportement. Pour des systèmes beaucoup plus complexes, il faut alors prévoir tous les cas de figure possibles, et donc modéliser des automates très compliqués. De plus, peu d'informations sont fournies quant au module de placement des caméras. Dans une optique agent affirmée, deux approches ont retenu notre attention.

5.1.2 Approche agent

Funge [Funge 99] est un des premiers à avoir comparé la caméra à un agent. Ainsi, Funge désire encapsuler les connaissances du réalisateur et du cameraman dans un modèle cognitif à l'aide de la notion de *fluent* (voir la section 4.1.2.2). Un ensemble de règles sont établies spécifiant le comportement de la caméra au cours de l'animation. Funge met l'accent sur la notion du temps. Celui-ci est géré dans le même formalisme par un ensemble de *fluents* mesurant si le plan est trop court ou trop long. L'exemple pris par Funge pour

valider son approche est une discussion entre deux personnages et est identique à celui développé par He. Des images extraites de l'animation sont présentées dans la figure 5.1. On note que Funge présuppose l'existence de primitives d'animation lui permettant de réaliser les différentes tâches de positionnement, ainsi que l'existence d'une entité "supérieure" transmettant des informations d'ordre sémantique sur la scène (c'est tel ou tel acteur qui parle).

FIG. 5.1 – Cinemasaurus *par John Funge (extraits de [Funge 99])*

Bill Tomlinson [Tomlinson 00] a plus récemment considéré la caméra comme une créature dans le cadre d'une approche éthologique, c'est à dire par l'utilisation de modèles de comportements d'animaux. L'agent caméra peut aussi animer des lumières dans le monde virtuel. Tomlinson met l'accent sur la capacité d'un système de cinématographie virtuelle à rendre le contenu "émotionnel" de la scène 3D. Pour ce faire, le mécanisme de sélection d'action de l'équipe de Blumberg est utilisé [Blumberg 97]. Plusieurs comportements mutuellement exclusifs sont en compétition pour déterminer quel sera le comportement actif. Cette compétition est orchestrée selon des paramètres relatifs à la nécessité de réaliser le comportement ainsi que selon les "émotions" de la créature (un certain nombre de paramètres internes).

Ces deux approches au problème de cinématographie virtuelle restent cependant assez simples, dans la mesure où elles ne proposent pas des systèmes complets, pouvant intégrer plusieurs modes de réalisation (plusieurs comportements), et sont de surcroît à un niveau très abstrait, ne prenant pas en compte certaines limitations ou possibilités liées au système d'animation.

5.1.3 Discussion et définition d'une architecture

Un des challenges majeurs associés à la modélisation d'un système de cinématographie virtuelle est de pouvoir conserver une cohérence dans la manière de filmer les choses tout en s'adaptant à des modifications dynamiques de l'environnement. Comme établi dans les différentes approches étudiées, il est nécessaire d'avoir à la fois un système d'animation pouvant positionner de manière adéquate la caméra (actions de base) et un module permettant une planification (souvent réactive) et donc un choix des sujets filmés, selon des critères de natures diverses : langage cinématographique, temps (longueur des plans), qualité des plans (éclairage), etc. Les approches orientées agent permettent d'encapsuler ces différents niveaux de contrôle, et de déporter ainsi la gestion à proprement parler de la caméra de la gestion globale des différents éléments de la scène. La "perception" associée à

l'agent caméra correspond alors à un traitement des éléments du scénario de l'application (ce qu'il est intéressant de savoir du point de vue de la caméra).

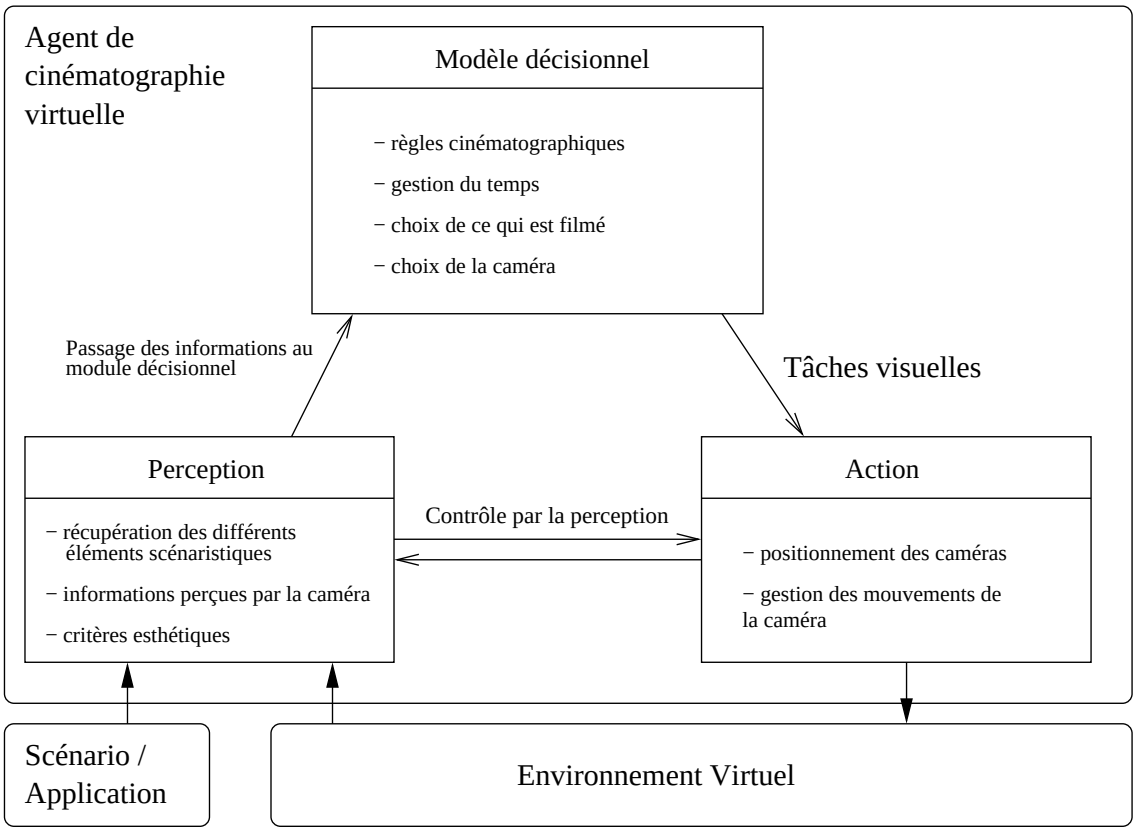


FIG. 5.2 – *Boucle perception-décision-action pour un agent de cinématographie virtuelle*

La figure 5.2 montre l'architecture d'un tel agent, selon le principe de la boucle perception-décision-action, définissant les fonctionnalités présentes dans notre système. La partie suivante fait référence à la brique *action* de l'agent, et la troisième partie (notion de metteur en scène) fait référence au module décisionnel de notre agent de cinématographie virtuelle.

5.2 Animation et contraintes “cinématographiques”

Les principaux travaux en cinématographie virtuelle établissent qu'il faut disposer d'un ensemble de primitives permettant de réaliser les différents plans par la caméra. Dans cette partie nous allons voir comment réaliser ces différentes primitives cinématographiques à partir de notre système d'animation de caméra. Cet ensemble de primitives définissent un **niveau de contrôle** nécessaire à la création des comportements de la caméra. Dans la deuxième partie de cette section nous établirons la manière de connecter ces primitives au module décisionnel en charge du montage.

5.2.1 Définition de primitives de contrôle

L'idée est d'abstraire une partie du contrôle pour pouvoir rendre invisible le formalisme de l'asservissement visuel. Ainsi la bibliothèque de mouvements de caméra doit pouvoir être utilisée par n'importe quel utilisateur. De plus, rajouter un niveau d'abstraction permet une simplification dans la conception des différents comportements de la caméra. Plusieurs types de mouvements cinématographiques de base sont ainsi proposés.

Positionnement. Pour pouvoir positionner simplement et efficacement les caméras dans l'environnement, nous avons utilisé la technique dite du "*look at, look from*". Des détails sur cette technique sont donnés dans l'annexe A. Celle-ci nous permet de spécifier simplement la position et l'orientation de la caméra. Cette spécification sert à positionner initialement les caméras, avant de réaliser des tâches de positionnement plus complexes (éventuellement sur des objets dynamiques) grâce à notre technique d'animation référencée vision.

Panoramique, poursuite et travelling simple. Ce sont certainement les mouvements de caméra les plus populaires. 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* (rotation de la caméra autour de l'axe vertical). 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 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 $\mathbf{L}$ soit de rang plein). On a alors un mouvement de travelling relativement simple. On note que ce mouvement peut aussi se faire en déplaçant manuellement la caméra avec un simple *look at, look from*. La plupart de ces mouvements sont donc encapsulés dans des primitives auxquelles sont spécifiés les objets d'intérêt. Des primitives virtuelles sont alors générées automatiquement. Par exemple, pour l'observation d'un tableau, seules la position et la distance au tableau sont spécifiées. A partir de ces deux paramètres sont calculés automatiquement les quatre points utilisés par l'asservissement.

Suivi de trajectoires. Il est prévu de pouvoir donner une vitesse à la caméra, produisant un mouvement qui assure cependant que la tâche de focalisation est assurée (principe de redondance). Le paramètre de contrôle est alors un vecteur vitesse. Le mouvement résultant dépend cependant complètement du type de focalisation réalisé, et donc de la nature des degrés de liberté restants. La plupart des cinéastes utilisent des rails sur lesquels la caméra se déplace. Une fois la tâche de positionnement spécifiée, nous utilisons la technique décrite dans la partie 2.4 pour faire suivre une trajectoire à la caméra. Les deux paramètres de contrôle sont alors la trajectoire de la caméra (sous forme de spline) et le temps mis pour parcourir cette courbe (on note que le calcul de ce temps peut se faire de manière dynamique, et donc être modifié au cours de l'animation).

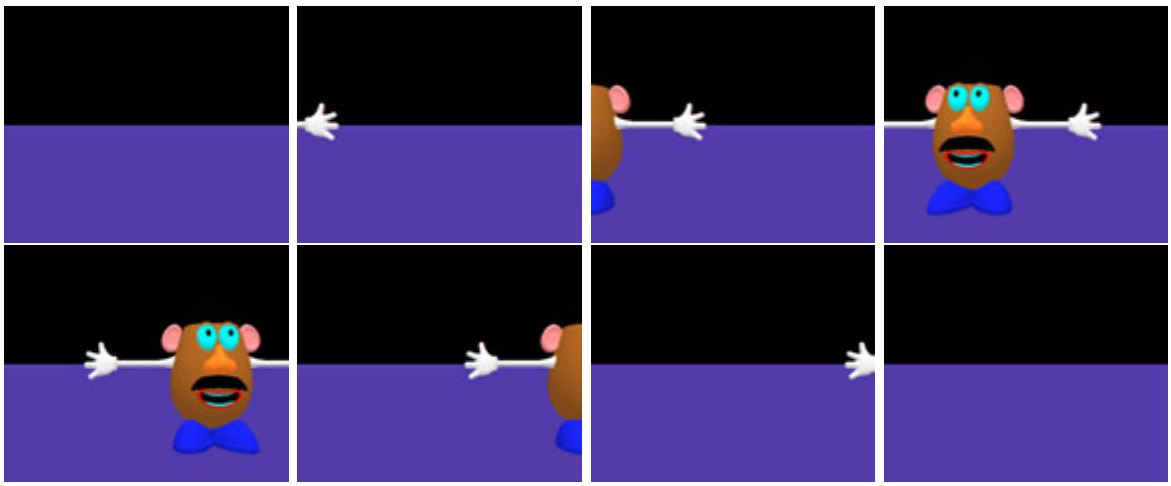


FIG. 5.3 – *Exemple de travelling sur un personnage*

Plans de coupe Une notion importante en cinématographie est la notion de distance. La graduation des distances entre l'objet filmé et la caméra peut être infinie. Cependant les pratiques actuelles distinguent cinq distances de base [Arijon 84] : gros plan, plan rapproché, plan moyen, plein cadre et plan d'ensemble. La figure 5.4(a) montre une représentation possible associée à ces différents plans de coupe. Pour un personnage, il apparaît que ces différentes distances se traduisent par des lignes de découpage magnifiant des qualités esthétiques.

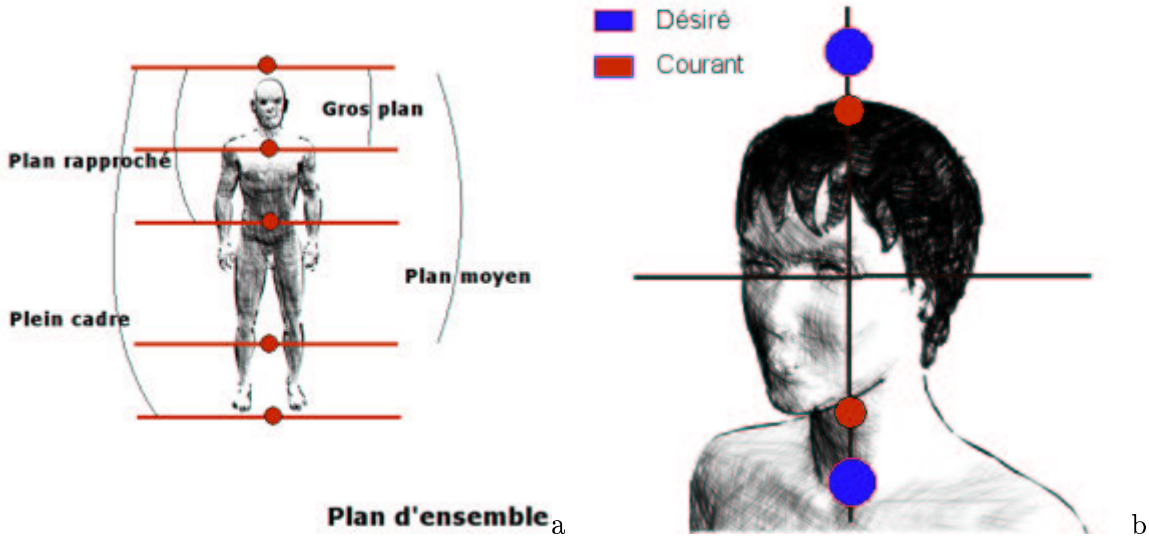


FIG. 5.4 – *Notions de distance en cinématographie : (a) différents plans de coupe. Les lignes de découpage sont en rouge (b) un exemple de primitives utilisées pour réaliser un gros plan*

Respecter ces différents plans de coupe apparaît être un problème non trivial en ciné-

matographie virtuelle. Du point de vue de notre système d'animation, obtenir ces plans peut se faire en choisissant de manière correcte les primitives pour l'asservissement par accès aux données anthropomorphiques des humanoïdes. Ainsi, filmer un gros plan revient à choisir comme primitives deux points correspondant au cou et au sommet de la tête, et à les positionner respectivement en bas et en haut de l'image désirée. Un exemple est montré dans la figure 5.4(b). Les points en rouge correspondent aux points liés au cou et au sommet du crâne. Les points en bleu sont les cibles. Lorsque la tâche est réalisée, la tête est effectivement en gros plan dans l'image. Des exemples de ces différents plans sont donnés dans les figures 5.5(a,b,c,d). Un tel asservissement laisse une possibilité de manoeuvre importante pour la caméra (celle-ci peut tourner autour du sujet ou avoir un mouvement vertical), exploitable au travers du principe de redondance. Les images de la figure 5.6 illustrent ce principe : la contrainte de plan est un plan d'ensemble en *plongée*. On donne alors un mouvement vertical à la caméra en tâche secondaire. La contrainte de plan d'ensemble est maintenue pendant l'animation (tâche principale).

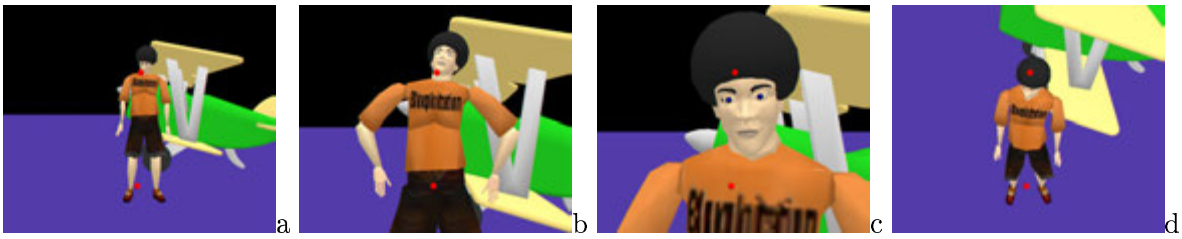


FIG. 5.5 – *Enchaînement entre différents plans. Les points rouges montrent les objectifs visuels dans l'image (a) plan d'ensemble (b) plein cadre (c) gros plan (d) un autre plan d'ensemble avec contre-plongée*

FIG. 5.6 – *Plan d'ensemble avec contre-plongée. On donne à la caméra un mouvement vertical avec maintient de la contrainte plan d'ensemble*

5.2.2 Le metteur en scène

Le metteur en scène est l'entité qui va décider des différents plans ainsi que du montage. L'idée est ici de considérer que l'agent en charge de la cinématographie virtuelle incarne ce metteur en scène. Plusieurs caméras sont à sa disposition, chacune de ces caméras réalisant une tâche de focalisation précise ou un comportement donné. Ainsi, une ou plusieurs caméras peuvent être impliquées dans un même comportement, et selon le principe de concurrence et de hiérarchie, plusieurs comportements peuvent s'exécuter en parallèle (et

utiliser une même caméra). La figure 5.7 illustre ce principe. Une telle architecture nécessite une gestion cohérente de ces différentes caméras.

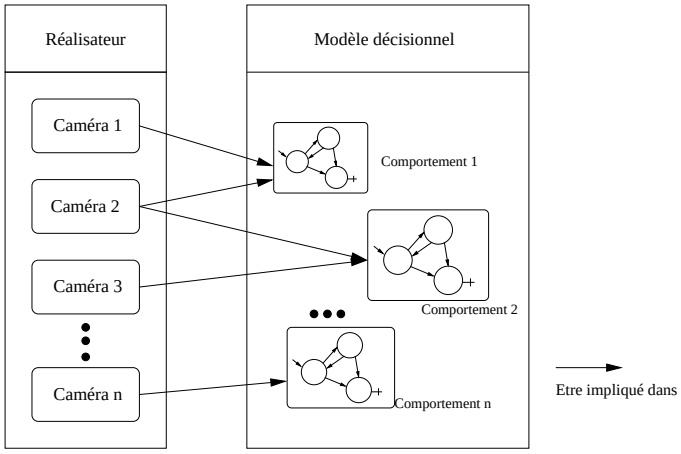


FIG. 5.7 – *Caméras et modèle décisionnel*

Dans un premier temps nous allons voir en quoi consiste un comportement pour une caméra, comment encoder les règles cinématographiques dans des automates, et dans un deuxième temps nous allons voir comment le montage est réalisé, c'est à dire comment on enchaîne les plans.

5.2.2.1 Comportement du réalisateur

Ce que nous appelons à ce stade comportement définit en fait une manière de filmer les choses. Selon le formalisme que nous avons choisi, chaque état des automates que nous définissons va correspondre à une prise de vue donnée. Cette prise de vue peut être statique (plan fixe) ou définir un mouvement de la caméra (réalisé par les primitives que nous avons définies dans le chapitre précédent, ou des primitives correspondant spécifiquement au problème). A ces différentes prises de vue correspondent différentes transitions. Celles-ci peuvent être effectuées pour différentes raisons, internes ou externes à la caméra ou au réalisateur :

- **internes**. Certains mouvements de caméra peuvent être impossibles à réaliser (occultations, vitesses trop élevées de la caméra, etc.). Dans ce cas le concepteur du comportement doit prévoir des alternatives à ce genre de cas. A l'inverse, un mouvement réussi peut entraîner une transition. Il faut aussi parfois tenir compte de la durée des plans. Un plan trop long (ou respectivement trop court) peut provoquer l'ennui (resp. la confusion) chez le spectateur,
- **externes**. Ces transitions font référence à des critères scénaristiques. Ils peuvent être de nature simple, par exemple un des acteurs passe d'une pièce à une autre (on utilise alors des alarmes ou *trigger*, considérés comme des détecteurs, qui font office de canaux de perception pour l'agent). Les éléments de nature plus complexe peuvent être défini : dialogue entre deux acteurs, trois acteurs, nécessité d'un panoramique ou d'une vue d'ensemble, introduction d'un nouveau personnage, volonté de montrer une action précise à un autre endroit, évolution de l'intensité dramatique, etc. C'est

alors au concepteur de prévoir les modalités de transfert de ces informations à l'agent de cinématographie virtuelle.

5.2.2.2 Encodage de règles cinématographiques dans des automates

Notre approche est ici similaire à celle de He [He 96] dans le sens où l'on cherche à encoder des règles cinématographiques dans des automates à états finis. L'idée est ici de dire qu'un état de l'automate correspond à une prise de vue (ou mouvement) particulière, et que les transitions correspondent à des changements de prise de vue. Ces changements peuvent être des coupures franches (on passe d'une caméra à une autre), ou des transitions animées (c'est la même caméra qui filme mais on change de tâche visuelle). Nous étudions maintenant la manière d'encoder des règles cinématographiques sur un exemple récurrent : le dialogue entre deux acteurs.

Dialogue entre deux acteurs Il est écrit dans le livre de Arijon que : *"L'approche la plus simple pour un dialogue face à face est l'utilisation d'un système d'angles opposés externes. Quand l'acteur apparaît en premier plan (de dos vers nous), en contre-champ externe, le bout de son nez ne devrait pas dépasser la ligne de sa joue [...] La répartition de l'espace scénique en un-tiers deux-tiers est fondamentale, bien que des variantes puissent être utilisées si on le désire."* L'analyse de cette règle renseigne sur la disposition des acteurs dans l'image filmée. La figure 5.8 montre les placements de la caméra possible dans l'espace 3D pouvant filmer un dialogue de type face-à-face.

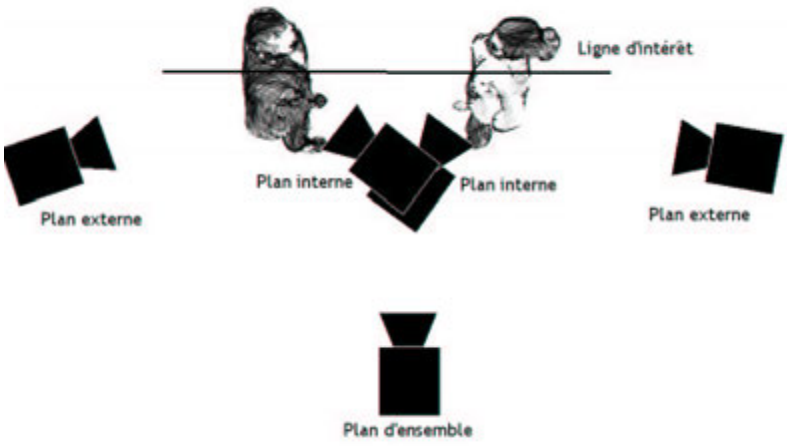


FIG. 5.8 – *Plans possibles pour filmer un dialogue entre deux acteurs, sur le principe du champ contre-champ interne ou externe*

La figure 5.9(a) montre l'automate associé, ainsi que deux vues correspondant aux deux états importants (figure 5.9(c,d)). Pour réaliser les contre-champs, deux caméras sont utilisées. L'idée est de contraindre la position des deux têtes dans l'image perçue par la caméra. On commence par effectuer un placement grossier à l'aide des méthodes de type *look-at, look-from* pour chacune des caméras (dans l'état d'initialisation). Il suffit alors de

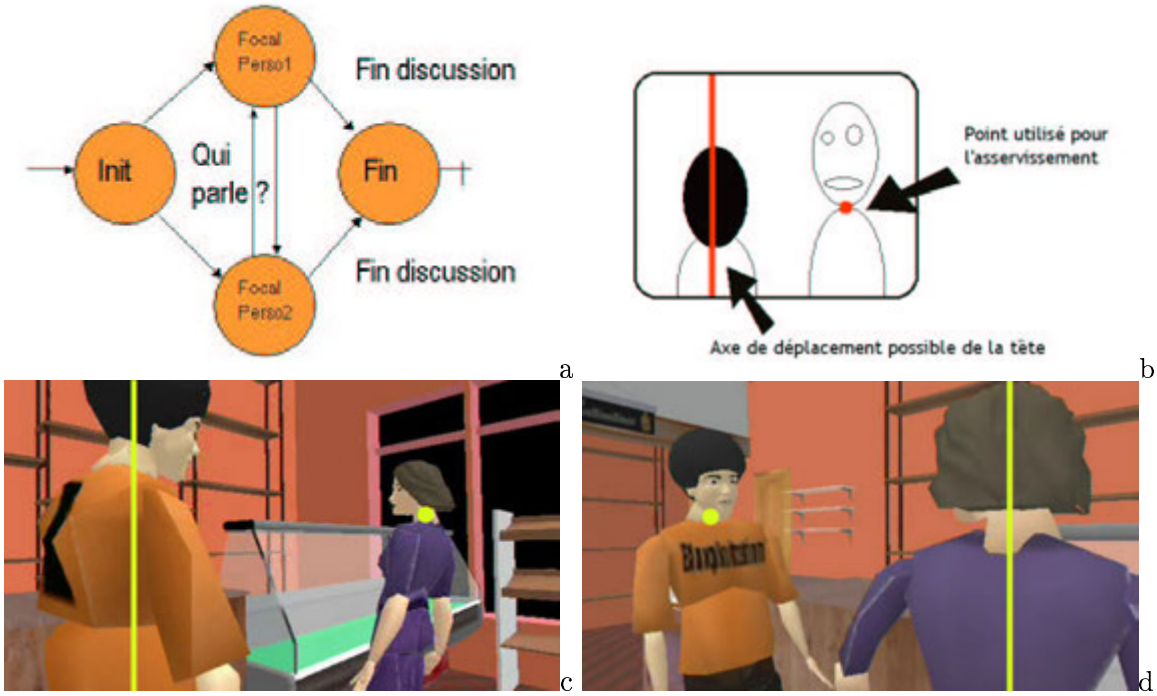


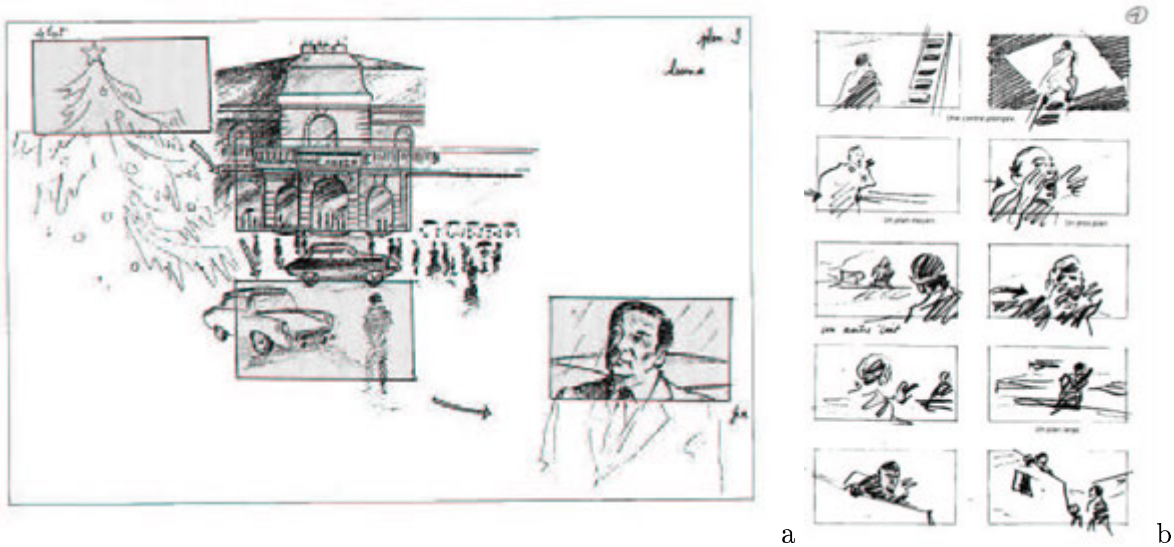
FIG. 5.9 – *(a) Automate simple de gestion d'un dialogue entre deux personnages (b) Primitives d'asservissement pour la réalisation d'un champ contre-champ (c,d) exemples de plans associés aux deux états* Focal perso *; les objectis visuels sont en jaune dans l'image*

s'asservir sur un point situé au niveau de la tête de la personne qui parle, et sur une ligne symbolisant l'axe du corps du deuxième personnage. Cette deuxième primitive garantit que la caméra ne peut tourner sur son axe Z, ce mouvement étant en règle générale très peu usité. La figure 5.9(b) est une représentation graphique associée à cet asservissement. On note que quelque soit l'état dans lequel se trouve l'automate, les deux caméras réalisent cette tâche à chaque pas de temps. Une transition signifie que c'est telle ou telle caméra qui est utilisée pour montrer la scène. L'asservissement n'est donc pas réalisé juste lorsque la caméra est active, mais bien tout le long de l'exécution de l'automate, ce qui évite des mouvements de la caméra dûs à son repositionnement (si les acteurs ont bougé entre deux transitions).

On remarque que ce comportement peut incorporer d'autres états, comme par exemple un état réalisant un plan d'ensemble pendant le dialogue. Le concepteur du comportement est bien entendu libre de définir des manières de filmer une conversation de manière beaucoup plus complexe.

5.2.2.3 Enchaînement de séquences

On dispose à ce niveau d'automates réalisant différentes séquences codant des règles cinématographiques. Le problème est ici de savoir comment passer d'une séquence à une autre. Dans le cas concret où l'on dispose de la structure du scénario avant son déroule-



a b

FIG. 5.10 – *Extraits de storyboards (a)* Garde à vue *de Claude Miller (b)* Himalaya *de Claude Valli*

ment, le concepteur des comportements peut prévoir un **méta-comportement** réalisant le séquencement des différents plans. Ainsi si l'on trouve dans un mode où l'on filme un panoramique de la scène puis un gros plan sur un personnage, le séquencement est automatique et implicite : on réalise les différents plans de manière linéaire, par exemple une fois le panoramique fini on enchaîne sur le gros plan. Cette structure linéaire est commune à celle employée lors de la définition des *storyboards*. Le storyboard désigne un découpage plan par plan du scénario, et est généralement accompagné d'illustrations. Deux exemples de storyboards sont donnés dans la figure 5.10. Pour chaque plan des annotations sont spécifiées, utiles au tournage (taille et durée du plan, angles de prise de vue, hauteur de la caméra, lumière, etc.), le contenu du cadre, les dialogues, des indications sur la musique et les bruitages, des informations sur les raccords. Toutes ces informations peuvent être directement encodées soit dans les automates, soit dans des primitives de contrôle de la caméra (basées sur notre technique d'animation).

Cependant, le problème majeur réside dans le fait qu'on ne dispose pas toujours de la structure du scénario, notamment dans le cas des jeux vidéos ou des applications de fiction interactive. La structure du scénario n'est alors plus du tout linéaire ; il faut pouvoir passer d'un comportement à un autre rapidement, tout en maintenant une cohérence globale dans la manière de filmer. Dans cette optique, deux grandes classes de méthodes existent. La première consiste à prévoir tous les cas de figure possibles, et de concevoir des automates pouvant s'adapter à toutes les situations. Deux problèmes majeurs surviennent alors : plus les cas de figure sont nombreux, plus les automates deviennent grands, et donc difficile à concevoir et coûteux à exécuter (problème de combinatoire). De plus, ces automates deviennent rapidement très dédiés à des contextes précis et ne sont pas conséquemment réutilisables. On perd alors l'aspect modulaire de l'encodage des règles cinématographiques : on n'a plus un automate générique filmant une conversation, mais un automate filmant la conversation dans un cadre précis avec des niveaux d'adaptation dédiés.

Principe et modélisation du problème Dans cette optique, nous avons choisi d'utiliser le système d'ordonnancement de comportements et de gestions de ressources présenté dans la section 4.2.2. L'idée est de pouvoir conserver **la généricité des différents comportements unitaires** possibles pour la caméra, et de les faire rentrer en compétition sur une ressource qui correspond à la prise de vue (c'est à dire ce qui est réellement montré). Pour cela, il faut pouvoir disposer pour chaque comportement défini d'une possibilité d'adaptation (relâcher la ressource prise de vue). L'intérêt majeur associé à cette méthode est le fait de pouvoir concevoir indépendamment les différents comportements unitaires que peuvent prendre les caméras. Dans notre cas, c'est le metteur en scène qui lance les différents automates. Le metteur en scène est lui-même codé sous la forme d'un automate. Ceci est possible du fait que le langage HPTS intègre la notion de hiérarchie et de parallélisme. Les automates représentant les différents comportements s'exécutent alors en parallèle, et se synchronisent par l'utilisation de la ressource de point de vue. Nous proposons d'étudier ce principe sur un exemple précis.

Etude de cas Dans cet exemple, nous nous focalisons sur un personnage (l'acteur 1) qui entre dans une pièce où se trouve un autre personnage (acteur 2). Deux manières de filmer sont prévues, l'une se référant au déplacement du personnage principal dans une pièce (voir Fig. 5.11a), l'autre à un dialogue entre deux personnages (voir Fig. 5.11b). La manière de réaliser les plans est conforme aux méthodes présentées dans le chapitre 5.2 sur la définition de primitives de contrôle. Au début de l'animation, l'automate metteur en scène lance les deux automates correspondant aux deux comportements prévus. Ces deux comportements sont en **compétition** sur la même ressource *prise de vue*. Savoir lequel des deux dispose effectivement de cette ressource dépend de la question "quel est le comportement le plus prioritaire?". En l'absence de dialogue entre deux acteurs (déterminé par le metteur en scène qui "communique" avec le scénario ou l'application), c'est le comportement consistant à filmer l'acteur 1 qui est prioritaire. Il s'exécute, alors que l'automate relatif à un dialogue reste bloqué dans son état initial. Lorsqu'une conversation survient entre les deux personnages, le metteur en scène fait monter la priorité de l'automate relatif au dialogue, qui lorsqu'elle dépasse celle de l'autre automate, provoque :

- la transition vers un des états réalisant un champ contre-champ en fonction de qui parle,
- le passage de l'automate relatif à la pièce dans un de ses états d'adaptation.

Lorsque le dialogue se termine, ou si l'automate filmant l'acteur dans la pièce se retrouve plus prioritaire (sa priorité pouvant évoluer en fonction du metteur en scène), la ressource *prise de vue* est relâchée. Dans ce cas, l'automate 5.11b transite vers l'état qu'il avait quitté avant le dialogue. C'est en cela qu'on garantit la cohérence dans l'enchaînement des différents plans ; **l'automate ne s'exécute pas depuis son état initial**, mais revient dans l'état qu'il avait quitté. On remarque que l'état réalisant un travelling au début de l'automate 5.11b ne dispose pas de transition vers un état d'adaptation. Si un automate devient plus prioritaire, il est obligé d'attendre la fin de cet état pour pouvoir s'exécuter ; c'est un état ininterruptible. Des images associées à l'exécution de ces comportements sont proposées dans la partie 7.2, relative à l'utilisation de ce système dans une application de type fiction interactive

Plusieurs comportements similaires à ces deux comportements peuvent s'exécuter en parallèle, utiliser un nombre illimité de caméras, et se synchroniser sur le mode de fonctionnement que nous venons de décrire. Dans ce cas, la difficulté revient à définir l'évolution des différentes fonctions de priorité. Ces fonctions définissent un "état interne" du metteur en scène (par référence aux paramètres physiologiques ou cognitifs qui régissent un être humain). Définir ces fonctions de priorité de manière adaptée revient alors à "personnaliser" le metteur en scène. Cet aspect fait partie des perspectives de recherche associées à ce travail.

5.3 Conclusion

Nous avons vu dans cette partie comment réaliser un système de cinématographie virtuelle fondé sur la notion d'agent. Au cours de notre étude, il nous est apparu important de pouvoir disposer d'un système d'animation disposant d'une grande flexibilité sur la nature du contrôle qu'il doit offrir. Notre système d'animation basé image apparaît être très intéressant dans le cadre de la cinématographie virtuelle, car il y a une adéquation directe entre les spécifications "cinématographiques" et le choix des primitives géométriques réalisant les tâches. La spécification du module décisionnel de l'agent de cinématographie virtuelle s'est faite au travers d'automates. Les possibilités offertes par l'ordonnancement des comportements nous ont permis de réaliser un système permettant de filmer de manière dynamique des scènes que l'on ne connaît pas à priori, tout en maintenant une cohérence globale. Des exemples et des résultats visuels sont présentés dans la partie relative à la démonstration Imagina.

En terme de perspectives, il est envisageable d'intégrer dans notre système des paramètres relatifs à l'éclairage des scènes (problème de la photo évoqué précédemment). Une autre perspective serait de pouvoir traduire au travers de la modélisation des comportements des contenus émotionnels différents, c'est à dire permettre de changer de manière de filmer en fonction du contenu "dramatique" de la scène qu'on filme.

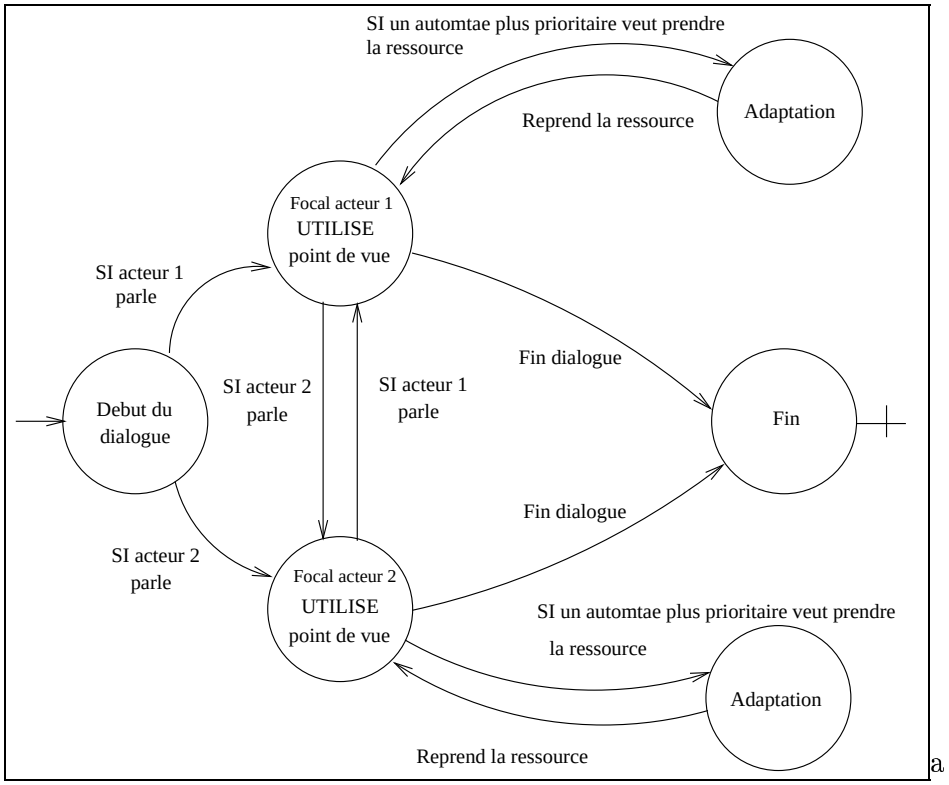


FIG. 5.11 – *Compétition de deux comportements de caméra (a) automate de dialogue entre deux personnages (b) automate filmant un acteur évoluant dans une pièce*

Chapitre 6

Perception visuelle d'un humanoïde de synthèse

La modélisation du comportement lié à la perception visuelle d'un humanoïde de synthèse est un problème difficile ; ce comportement produit des indications subtiles et émergentes sur les intentions et les processus cognitifs régissant l'humanoïde. L'utilisation d'animations capturées ou basées sur des scripts se révèle inadaptée, car la variété des schémas possibles est trop importante pour produire des animations crédibles, et la nécessité d'adaptation à l'environnement implique un contrôle élevé ainsi qu'une adaptabilité forte. Notre objectif est donc de pouvoir simuler la perception visuelle de l'humanoïde pour :

- augmenter le réalisme visuel des animations,
- fournir une brique de base essentielle à la modélisation d'un humanoïde autonome.

Dans la première partie de ce document nous avons présenté un système d'animation répondant à une partie de ces exigences (réactivité, modalité de contrôle). L'objectif de ce chapitre est d'établir le lien entre ce modèle d'animation et un module d'animation comportementale établissant les différents comportements perceptifs de l'humanoïde. La difficulté revient ici à prendre en compte les différents niveaux de contrôle impliqués dans le processus de perception visuelle. En effet, plusieurs mécanismes sont mis en jeu : une attention consciente ou contrôlée, où l'humanoïde regarde des choses précises dans l'environnement, et une attention passive où l'humanoïde récupère des informations par des biais dont il n'a pas nécessairement "conscience". Ces deux processus sont en compétition permanente et en forte interaction. Il faut donc trouver un moyen de faire coexister et coopérer ces deux modes opératoires, tout en garantissant une cohérence globale, c'est à dire un enchaînement logique, motivé par un but, des différents actes de perception de l'humanoïde. Après une étude de l'état de l'art sur la perception visuelle (théorie et modèles), nous présenterons la manière dont nous avons lié le système d'animation à un module simulant les différentes modalités de la perception visuelle humaine.

6.1 Généralités sur la perception visuelle humaine

Dans un premier temps nous rappelons quelques principes physiologiques et psychologiques sur les structures humaines associées à la perception, avant d'établir un état de

l'art sur les différentes modélisations informatiques du problème, que ce soit en synthèse d'images ou en vision par ordinateur.

6.1.1 La perception visuelle chez l'homme

La perception visuelle s'opère au travers des yeux. L'œil est un système optique convergent, comparable à une caméra munie d'un zoom. Les rayons lumineux convergent vers la lentille, car le globe oculaire est bombé au niveau de la cornée. Cette convergence se règle grâce à l'action des muscles ciliaires qui agissent sur le cristallin (voir Fig 6.1). Ce processus se nomme *accommodation*. De la structure du globe oculaire dérive un grand nombre de problèmes, comme la myopie, provoquée par une trop forte convergence. La rétine analyse

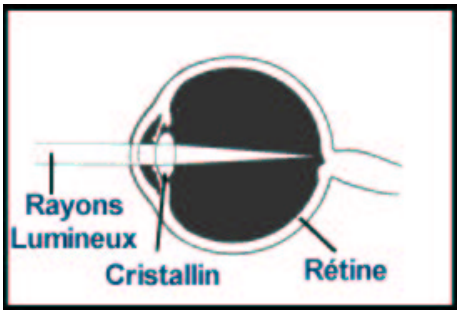


FIG. 6.1 – *Schéma d'un œil humain*

et intègre de nombreux signaux lumineux qui sont finalement interprétés par le cerveau. Elle est composée d'une structure horizontale (opérant des associations dont on connaît certaines fonctions) et d'une structure verticale (cônes et bâtonnets) où les impacts des photons qui composent la lumière provoquent des réactions chimiques qui sont la base des signaux bioélectriques envoyés au cerveau. Les signaux nerveux sont interprétés au niveau cérébral par deux systèmes complémentaires : le système rétino-cortical et le système rétino-tectal. Le premier est le siège de **l'identification des formes** et le second de **la localisation** (voir Fig 6.2). On peut noter la présence à la base du cortex de deux noyaux occulo-moteurs qui servent à commander le mouvement des yeux.

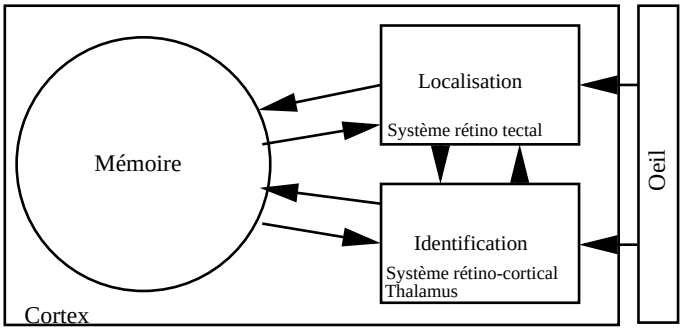


FIG. 6.2 – *Les deux systèmes de vision*

Il est courant de décomposer le processus de vision en trois grandes étapes [Romeny 02] : une vision primaire, qui produit une analyse géométrique de la scène observée. A ce stade, aucune corrélation avec la mémoire ou la notion de concept n'est établie. La forme, la

couleur, le mouvement ou la fréquence des objets sont recueillis dans des canaux plus ou moins séparés. Ces informations sont ensuite relayées à un niveau intermédiaire qui détermine des groupes, analyse des formes plus complexes, traite les informations de profondeur et de mouvement et réalise les premières associations avec des éléments connus. Le dernier niveau fait référence aux processus associés à la cognition, à la reconnaissance de forme et à la perception consciente des objets. Cette dernière étape est d'un point de vue psychophysiologique la moins bien connue. D'un point de vue informatique et applicatif, chacune de ces trois étapes correspond à des problèmes théoriques fondamentaux : le premier niveau est en relation directe avec le traitement d'images, le deuxième avec l'analyse d'image (segmentation, champs de mouvement, reconstruction 3D) et le troisième avec la reconnaissance de formes, ainsi que la plupart des modèles de raisonnement et de modélisation de la mémoire. En marge de cette analyse, il est établi que le mouvement des yeux et le choix des points de fixation contribuent à notre compréhension d'une scène ou d'une image [Yarbus 67]. Le processus de focalisation sur un point précis se nomme *fixation*. Ainsi l'interrogation associée à une image (ou environnement) donnée se traduit par une série de focalisations sur les différents points d'intérêt associés. Il en est de même pour le "rafraîchissement" des représentations mentales que l'on construit. Le problème du choix de l'endroit où notre regard se fixe est connu sous le nom de *fixation sélective* [Hacisalihzade 92,Didday 75]. Dans le cas d'objets en mouvement, cette faculté requiert la possibilité de faire du *tracking*. Le choix des objets ou des zones à fixer peut s'effectuer de différentes manières : modélisation du choix de l'endroit à l'aide de systèmes probabilistes tels les réseaux bayesiens [Rimey 92] intégrant ou non l'existence d'états cachés (modélisation à base de chaînes de Markov cachées [McCallum 97]) , ou plus récemment avec la notion de carte de saillance [Itti 98].

La perception visuelle comporte donc de multiples aspects, et l'on voit clairement que ces différents aspects possèdent de nombreuses interactions : la compréhension des scènes dépend des différentes fixations réalisées, mais ces fixations dépendent du processus d'analyse des images perçues. Cette dualité du contrôle doit être intégrée dans les modèles de perception visuelle synthétiques.

6.1.2 Modélisation de la perception visuelle en synthèse d'image

Plusieurs méthodes ont été proposées pour simuler la perception dans un environnement virtuel. Ces méthodes peuvent se scinder en deux grandes classes : approches basées sur le traitement d'une image (rendue du point de vue de l'humanoïde), et modélisation de la perception comme un accès particulier à un ensemble d'objets de la base de données. Dans les deux cas, la notion de visibilité des objets est importante ; l'humanoïde dispose d'un champ de vision restreint, et le problème revient à déterminer les facettes visibles d'un ensemble d'objets 3D, en considérant les éventuelles occultations. Il est intéressant d'étudier par la suite quelques travaux concernant la mémoire, les interactions entre la mémoire et les processus de perception se révélant être nombreuses.

6.1.2.1 Traitement de l'image

Ce type d'implémentation repose sur le rendu de la scène à partir du point de vue de l'acteur autonome. Le problème du temps de calcul d'une nouvelle image depuis un nou-

veau point de vue est aujourd'hui clairement résolu dans la mesure où les cartes graphiques actuelles permettent de générer ces images très rapidement. Tu et Terzopoulos [Tu 94] ont implémenté un modèle de vision synthétique pour leurs poissons artificiels basé sur du lancer de rayons. Blumberg utilise une technique basée image d'évaluation d'énergie pour le déplacement et la navigation de son chien artificiel [Blumberg 97]. Rabie et Terzopoulos [Rabie 96] se servent d'images pré-calculées des objets de la scène pour effectuer une reconnaissance (basée sur des histogrammes de couleur) de motifs (*pattern matching*) associés aux objets observés. Renault [Thalmann 90], dont les travaux ont été repris par Noser [Noser 95], ainsi que Kuffner utilisent des algorithmes de rendu en fausse couleur sans modèle d'illumination pour simplifier la phase de traitement de l'image. Un des avantages majeurs de ces techniques est d'inclure directement dans leur formalisme la notion de visibilité des objets (déterminée à partir du *Z-buffer*), ce qui évite des calculs géométriques coûteux.

6.1.2.2 Interprétation et accès aux informations de la base de données

Une autre approche consiste à envisager la perception comme étant un accès direct (omniscient) aux objets de la base de données. Les objets peuvent alors porter en eux différents types d'informations (positionnelles, sémantiques, etc.). Pour des raisons de réalisme et d'accès limités à la base de données, l'acteur autonome ne peut accéder qu'aux objets se trouvant dans une certaine zone de proximité (qui peut être le cône visuel). C'est le cas dans les nuées de Reynolds [Reynolds 87], où chaque entité a la connaissance de la position de ses voisins immédiats. Dans l'optique d'utiliser des modèles de perception visuelle temps-réel, la nécessité du calcul de visibilité par des raisonnements sur la géométrie est pénalisante (modulo l'utilisation de bases de données géométriques dédiées dans le sens où elles sont structurées ou disposent de modes de représentation faisant chuter la complexité du calcul, par exemple des *octrees* [Funge 98]). Badler [Badler 97a] pourvoit ses acteurs synthétiques de différents capteurs (de proximité par exemple) permettant d'instancier des comportements de haut niveau. La structuration de la base de données est dans cette perspective très importante. L'accès aux informations pertinentes peut être grandement facilité si les données sont triées selon des propriétés sémantiques ou topologiques [Thomas 99]. La relation établie entre les objets de l'environnement et le comportement de perception montre une cohérence avec le modèle des "affordances" établi par Gibson [Gibson 79]. Cette théorie unifiée de la perception met en valeur le fait que la perception des objets offre à l'humanoïde des informations sur leur fonctionnalités, plutôt que sur leurs formes ou leurs couleurs, plaçant ainsi la perception visuelle à un niveau beaucoup plus abstrait. Cette théorie a été utilisée dans quelques travaux en synthèse d'images, notamment par Widyanto [Widyanto 91], puis plus récemment dans les *Smart Objects* (objets contenant des informations sémantiques) par Kallmann [Kallmann 98].

6.1.2.3 Mémoire et modèles décisionnels

De nombreux travaux issus des sciences cognitives ont porté sur la mémoire, avec l'émergence d'un débat sur la validité de l'approche représentationnelle, c'est à dire qui considère que notre esprit réfléchit avant tout sur des représentations de notre environnement. Il est préférable en un sens de chercher à exploiter ce formalisme, dans la mesure où il simplifie grandement le travail de "raisonnement" dont on peut doter un acteur virtuel. Les deux

travaux les plus marquants sont ceux de Noser [Noser 95], qui construit une représentation de l'occupation de l'environnement à l'aide d'*octrees*, et Kuffner [Kuffner 99] qui maintient une liste de pointeurs sur des objets de la base qui ont été observés. Toute la difficulté associée à ces méthodes est de prendre en compte la dynamique associée à un environnement virtuel. Comment considérer un objet qui bouge vis-à-vis de la représentation de l'environnement ? Comment modéliser le fait que l'on oublie ? Pour l'instant, ces modèles de mémoire ont servi à effectuer une planification sur les trajectoires empruntées dans l'environnement. L'humanoïde raisonne et planifie selon la représentation mentale qu'il construit de l'environnement. Encore peu de travaux utilisent ces représentations mentales et modèles de mémoire à un niveau plus abstrait ou conceptuel ; ainsi il n'existe pas encore de mécanismes d'abstraction entre ce qui est perçu au niveau géométrique et le "concept" de l'objet (abstraction de sa fonctionnalité).

6.1.3 Discussion

L'étude de ces différents travaux amène à définir plusieurs fonctionnalités nécessaires à une architecture simulant la perception visuelle :

- être capable de déterminer ce qui est vu par l'humanoïde. Il faut alors être en mesure de prendre en compte les occultations totales ou partielles des objets par d'autres objets de l'environnement. On note que cette notion de visibilité est directement contenue dans les images rendues du point de vue de l'humanoïde. Le résultat obtenu peut servir directement à un modèle de mémoire ou de planification,
- être capable d'associer les objets visibles avec un contenu sémantique ou un ensemble de propriétés. Dans le cadre des humanoïdes de synthèse, les approches basées sur des environnement informés semblent plus adaptées, car elles reposent à la fois sur des bases psychologiques, et évitent des phases d'interprétation des images (reconnaissance de formes), semblant trop complexes à mettre en oeuvre,
- intégrer une dualité de contrôle. Le processus de perception visuelle intègre à la fois un contrôle de haut niveau : choix des tâches visuelles, stratégies de perception (on qualifie alors ce processus descendant ou *top-down*), ainsi qu'un contrôle passif, i.e. guidé par des informations visuelles de bas-niveau (processus ascendant ou *bottom-up*). Chopra [Badler 99a,ChopraKhullar 99a] réalise cette distinction en décrivant trois types de perception : une perception *endogène* – basée sur la volonté –, une perception *exogène* – un objet rentre dans le champ visuel et attire notre attention – et une perception *oisive*, où notre regard se promène indifféremment sur les différents objets de la scène.

6.2 Modélisation des comportements de vision

Dans cette section nous présentons la modélisation de notre système de perception visuelle. Comme établi dans la partie précédente nous distinguons deux processus attentionnels régissant la perception : un processus endogène et un processus exogène. Nous nous attachons dans cette partie à voir comment la technique d'animation référencée vision introduite dans la première partie de ce document est utilisée, comment sont mis en œuvre ces deux aspects de la perception, et enfin comment une cohabitation entre ces deux processus est modélisable.

6.2.1 Gestion de la cinématique associée à la perception visuelle

La simulation de la perception visuelle de l'humanoïde nécessite de gérer la cinématique de celui-ci relativement à l'attention visuelle. Pour gérer cette cinématique, nous avons défini dans la première partie de ce document une méthode de contrôle reposant sur la simplification de la modélisation des tâches visuelles, et proposant une gestion automatique de problèmes spécifiques à l'animation d'humanoïdes. Dans cette partie nous rappelons les différents niveaux de contrôle proposés, en réalisant une distinction entre ce qui est géré de manière automatique, et ce qui constitue les bases du contrôle (primitives de contrôle) utilisables lors de la modélisation des comportements.

Fonctionnalités intrinsèques au système d'animation . Ces fonctionnalités sont cachées, dans le sens où l'utilisateur n'a pas à spécifier de réglages précis, ou se soucier du fonctionnement interne. Voici un descriptif de ces fonctionnalités :

- **gestion interne des butées articulaires**. Le but est ici d'éviter que lors de la réalisation d'un mouvement (voir chapitre 3.3), l'humanoïde ne dépasse ses butées articulaires. Les valeurs de ses butées sont inclues dans des fichiers de description de l'humanoïde. A l'initialisation, on récupère directement ces valeurs. Lorsque tous les axes sont en butée, on spécifie que la tâche est *impossible*. Plusieurs politiques sont alors possibles : soit l'humanoïde revient en position initiale, soit il reste en butées articulaires. On considère que c'est au concepteur du comportement de prévoir ce cas, et d'imaginer une adaptation convenable,
- **gestion implicite des interpolations**. Un ensemble d'interpolateurs par défaut gère les interpolations entre les différentes tâches visuelles et les différents modèles de chaînes articulaires. Ainsi, lorsqu'une transition est détectée, son type détermine le type d'interpolateur utilisé. Pour un meilleur contrôle, plusieurs paramètres peuvent modifier ces fonctions d'interpolation (voir chapitre 3.4),
- **gestion implicite de la locomotion**. Pour toutes les tâches contraignant la distance, on ajoute des degrés de liberté en translation aux chaînes considérées. Grâce au contrôleur (qui joue le rôle de filtre, voir partie 3.5), on ne considère que des déplacements qui ont une amplitude significative.

Interface de contrôle . Ces différentes méthodes permettent de spécifier les tâches visuelles à réaliser, mais aussi à contrôler (dans une certaine mesure) la réalisation de celles-ci. En voici une énumération :

- **réglage de la vitesse de réalisation de la tâche**. L'utilisateur peut paramétrer la vitesse à laquelle l'humanoïde réalise la tâche visuelle, en modifiant directement le gain de la tâche. On note qu'on bride ce gain à des vitesses ne pouvant pas dépasser les limites humaines (ce qui a comme bonne propriété par ailleurs d'éviter des oscillations autour du but),
- **choix du type de chaîne considérée**. L'utilisateur peut choisir et changer à n'importe quel moment le type de chaîne articulaire utilisée dans l'acte de perception visuelle, plusieurs "modèles" de chaînes étant proposés. On note que les informations sur ces chaînes (longueur des membres) sont contenues dans des fichiers de description disponibles avec les humanoïdes (et donc utilisées automatiquement et dynamiquement),

- **retour sur l'état de la tâche**. Un système de drapeaux est mis en œuvre pour spécifier à l'utilisateur si la tâche visuelle est *en cours*, *réussie* ou *impossible*. Un réglage de précision est offert, correspondant à la limite à partir de laquelle on considère que la tâche est réussie (basée sur la norme de l'erreur dans l'image),
- **définition de la tâche visuelle**. Il est possible de définir la tâche visuelle en spécifiant directement les primitives géométriques utilisées dans la loi de commande. De plus, certaines tâches visuelles sont directement disponibles sans avoir à spécifier manuellement la totalité des primitives. Ainsi, regarder un point précis de la scène peut se faire sans que l'utilisateur n'ait à spécifier qu'il désire voir ce point centré dans l'image. De la même manière, il est possible de régler la "distance d'observation" en ne spécifiant qu'un point et une distance. Dans ce cas, les primitives géométriques utilisées sont construites de manière automatique (une sphère par exemple). Il est aussi possible de rajouter des contraintes d'axes (par asservissement sur une ligne), empêchant par exemple de regarder des objets avec la tête penchée. La figure 6.3 fait référence à cette contrainte. L'utilisateur est alors en charge de spécifier un objet à regarder, ainsi que l'axe d'observation (vertical par défaut).

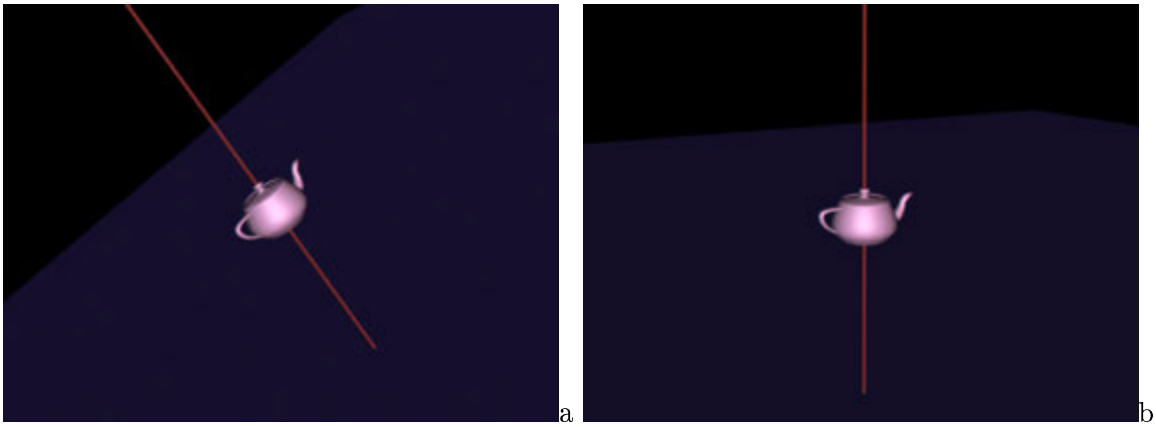

a b

FIG. 6.3 – *Contrôle de l'axe. La tâche d'observation se fait selon la théière. La ligne droite doit être maintenue verticale dans l'image (a) mauvaise configuration (b) bonne configuration*

Effets visuels Au cours des premières expérimentations, il est apparu de manière très nette qu'il manquait quelque chose au réalisme des animations : la gestion du mouvement des paupières. Ces battements ont un rôle tout d'abord physiologique (humidification du cristallin), mais présentent aussi des indices sur l'activité cognitive de l'humanoïde. Ainsi des battements répétés peuvent traduire une nervosité accrue. Nous avons donc réalisé un modèle de paupière contrôlable. Basé sur un système d'animation très simple (paramétrique), il simule le battement des deux paupières. L'objet graphique paupière tourne autour du centre de l'oeil (voir Fig 6.4(a)), ce qui permet d'éviter l'interpénétration des deux objets graphiques. Le paramètre de contrôle est le niveau d'ouverture de la paupière (normalisé entre 0 pour fermée et 1 pour ouverte). Deux modes de contrôle sont proposés : un mode *aléatoire*, où les intervalles entre chaque battement sont irréguliers (on note qu'on peut tout de même modifier la fréquence générale pour simuler différents états), et un mode

guidé où l'on contrôle directement le niveau d'ouverture. Ce dernier mode est notamment utile pour donner une expression au visage ou plus précisément au regard (voir Fig 6.4(b)).

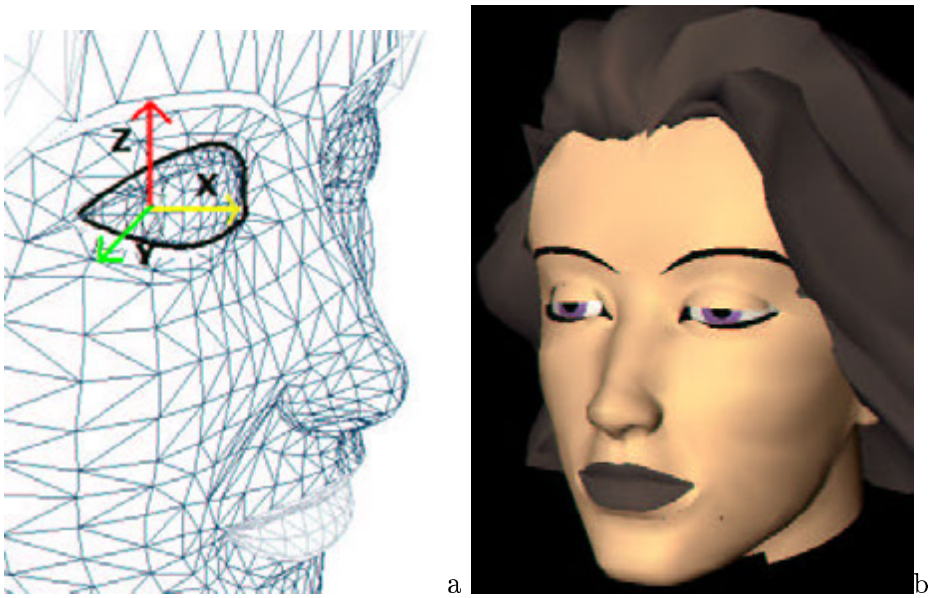


FIG. 6.4 – *Battement des paupières (a) l'objet graphique paupière (délimité par un trait noir) possède un centre de rotation confondu avec celui de l'oeil (b) tête de sara avec les paupières à moitié fermées, lui donnant un air maussade*

6.2.2 Processus endogène : guidé par la volonté

La modélisation des comportements de base est effectuée au travers d'automates grâce au langage HPTS (voir la section 4.2). La plupart du temps différents états correspondent à différentes tâches visuelles, et la transition entre ces différents états s'opère selon des conditions intrinsèques au comportement (focalisation réussie, temps mis pour la focalisation, etc.) ou extrinsèques (un autre comportement requiert l'utilisation de la perception). Il apparaît important de pouvoir contrôler ces automates à l'aide de paramètres ayant trait à la psychologie du comportement. Ainsi Chopra [ChopraKhullar 99a] introduit la notion d'AVA (*Automated Visual Attending*), qui intègre à la fois la notion d'activité cognitive et la notion d'activité motrice. L'activité cognitive impose un certain nombre de contraintes et induit la notion d'incertitude et de surcharges visuelles et cognitives. A ces différents paramètres sont ajoutés des paramètres comportementaux, tels que la distractibilité (par opposition à la concentration), la capacité à anticiper, la confiance en soi (libérer l'attention visuelle sur un objet que l'on prend par exemple, alors que le mouvement n'est pas fini) et la confiance envers ses propres représentations mentales (qualité de la mémoire). Nous allons montrer comment dans notre système nous pouvons prendre en compte ce genre de paramètres, cependant nous ne présenterons pas une description exhaustive des comportements d'attention possibles. Ainsi le prochain paragraphe est dédié à l'étude d'un comportement intégrant des paramètres cognitifs.

Gestion de l'attention lors d'un comportement de marche Les tâches de surveillance visuelle s'opèrent à différentes fréquences pouvant être déterminées par l'échantillonement de Nyquist [Moray 93] ; par exemple un pilote d'avion surveille ses compteurs à des intervalles de temps donné. Cette fréquence est directement corrélée au niveau d'incertitude associé à la mesure. Ce type de tâches de surveillance est le genre de tâches typiques s'effectuant à des intervalles de temps donné, s'effectuant en parallèle d'autres tâches. Lorsque l'humanoïde marche, il regarde à la fois devant lui et à la fois l'endroit où il pose ses pieds [Firby 91]. La fréquence à laquelle il regarde le sol dépend directement de sa connaissance du lieu (niveau de confiance). Un changement de lieu, de nature du sol ou de sa configuration peut ainsi provoquer une diminution de son niveau de confiance. Nous allons voir comment ce fait est modélisable au travers d'un automate.

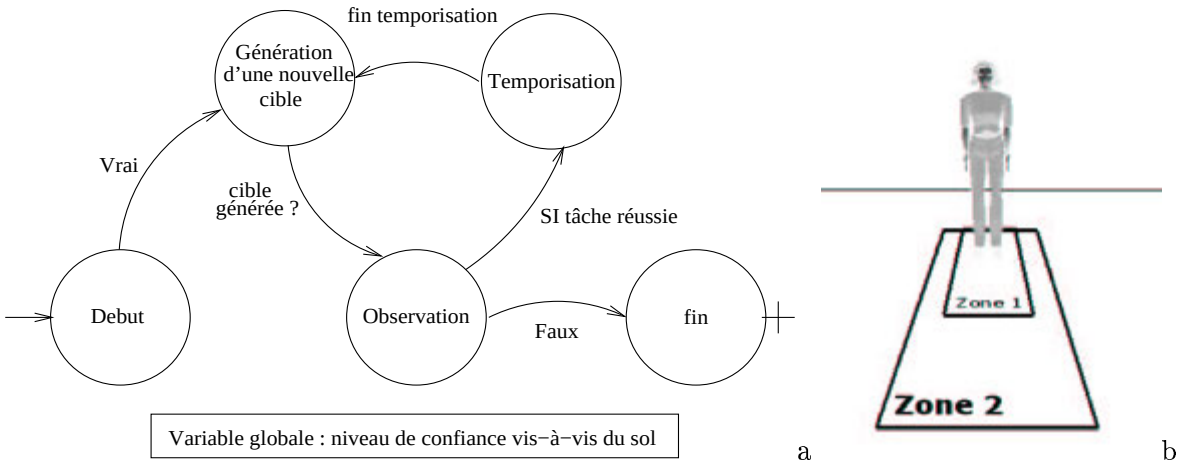


FIG. 6.5 – *Tâche de surveillance pendant un comportement de marche (a) automate associé au comportement visuel (b) zones où l'on effectue le tirage aléatoire des cibles*

La figure 6.5(a) représente un automate modélisant ce comportement. Cet automate simple dispose de cinq états. L'état *Observation* correspond au moment où l'humanoïde regarde la cible générée au niveau du sol. Les possibilités de transition sont soit la terminaison de l'automate (cette transition étant spécifiée infranchissable la plupart du temps) et une transition vers un état où l'humanoïde regarde l'horizon. Cette transition est effectuée une fois que l'humanoïde a atteint sa cible. Le niveau de précision sur la focalisation de la cible est directement lié au niveau de confiance de l'humanoïde. Lorsque l'humanoïde a regardé sa cible, il entre dans l'état *Temporisation*, réalisant une pause entre deux observations. Ce temps de temporisation est directement lié au niveau de confiance lié à la connaissance du lieu. Ce paramètre peut évoluer dynamiquement au cours de la simulation. L'état *Génération* correspond à la génération d'une nouvelle cible. Ce processus est détaillé dans le chapitre suivant.

Du point de vue de l'animation, deux types de primitives sont utilisées : une ligne pour modéliser l'horizon, et un point pour les cibles sur le sol. Le choix de cette cible est aussi directement lié au niveau de confiance. Ainsi une zone de proximité est définie (plus la certitude est élevée, plus l'humanoïde peut regarder loin devant lui). La figure 6.5(b) montre

les zones dans lesquelles on effectue un tirage aléatoire des cibles. La zone 2 correspond à un niveau de confiance élevé vis-à-vis du sol : l'humanoïde peut regarder des cibles à une distance relativement élevée, avec une fréquence faible (à l'inverse de la zone 1). On note qu'entre chaque transition une interpolation de cibles permet de réaliser des transitions fluides entre le fait de regarder le sol et l'horizon. Les résultats associés à l'exécution de cet automate sont présentés dans la partie relative aux résultats.

Cet exemple simple montre la capacité de notre système à modéliser des comportements visuels intégrant des paramètres cognitifs élémentaires en ce basant sur notre système d'animation. Ce type de comportement est en fait un des multiples comportement de perception s'exécutant en parallèle au cours de la simulation de l'humanoïde. Plusieurs de ces comportements ont été modélisés : coordination entre la préhension d'un objet et les yeux, attention visuelle lors d'une discussion, observation méticuleuse d'un environnement ; cependant leurs modélisation en sera pas détaillé dans ce document. Après avoir étudié les mécanismes nous permettant de simuler la perception exogène, nous reviendrons sur la manière de conserver une cohérence globale (enchaînement logique) entre ces différents comportements exécutés en simultané.

6.2.3 Processus exogène : utilisation de cartes de saillance

Nous venons d'étudier à travers un exemple simple la manière de modéliser des comportements d'attention endogènes, c'est à dire étant motivé par la volonté. Une autre composante importante de la simulation de la vision humaine est le fait que notre regard soit attiré par des éléments de l'environnement, selon des critères qu'il faut pouvoir définir. Cette information est primordiale si l'on souhaite considérer les propriétés physiologiques de la perception, et augmenter la finesse (et donc le réalisme) de l'acquisition des informations (plutôt que de traiter directement à tous les objets présents dans le cône de vision). Chopra-Khullar [ChopraKhullar 99a] utilise dans son approche un traitement simplifié reposant sur la récupération de l'image couleur rendue du point de vue de l'humanoïde. Celui-ci détermine les zones de haut gradient dans l'image. Par la suite, l'agent regarde cette zone. Cette approche très locale ne prends pas en compte les spécificités de l'appareil visuel humain. Nous proposons d'utiliser la notion de cartes de saillance, formalisme dédié à la modélisation du parcours du regard sur une image, pour modéliser cette modalité perceptive. Dans un deuxième temps nous verrons comment ce processus s'intègre dans une architecture plus globale, et comment il est possible de récupérer des informations de nature sémantique par ce biais.

6.2.3.1 Cartes de saillance

Une carte de saillance (appelée aussi carte rétinoptique) est une représentation topographique du champ visuel. Celle-ci code la "conspicuité" locale de la scène, c'est à dire le potentiel attractif des éléments de la scène. Le but d'une telle représentation est de pouvoir déterminer les différents points de localisation du focus attentionnel permettant la compréhension globale de la scène. Les cartes de saillance sont généralement utilisées dans le cadre de modèle prédictif du parcours des yeux sur une image ou une suite d'images. Les informations visuelles utilisées pour modéliser cette conspicuité sont les fréquences spatiales, les orientations, la luminance, la couleur [Itti 98]. Généralement, ces informations

sont extraites en parallèle dans différentes **cartes de détection de traits** [Niebur 96], qui sont mises en correspondance dans une carte de saillance. Les derniers travaux du domaine portent sur l'intégration comme base pour coder la conspicuité d'informations sur le mouvement, la profondeur et certaines lois de la Gestalt. La figure 6.6 montre des exemples de carte de saillance obtenue avec la méthode de Itti [Itti 98].

FIG. 6.6 – *Exemples de carte de saillance sur des images réelles (extraites de [Itti 98]). Les cercles jaunes montrent les zones saillantes et les traits rouges l'ordre de détermination de celles-ci*

6.2.3.2 Utilisation de cartes de saillance pour modéliser la perception exogène

Principe de fonctionnement L'utilisation d'un système évaluant la conspicuité des éléments perçus par l'humanoïde doit pouvoir nous fournir *des éléments sur les endroits où celui-ci va poser son regard* au cours de la simulation. Le principe, mis en image dans la figure 6.7 est le suivant : à intervalles de temps non réguliers, une image de ce que voit l'humanoïde est calculée. Sur cette image, un traitement d'image particulier nous permet de déterminer une carte de saillance selon différents critères (exposés dans la partie relative aux résultats). Une série de points d'intérêts est ainsi extraite, et est utilisée comme cible pour l'humanoïde. On note l'intérêt de disposer d'une méthode d'animation référencée vision, nous permettant de contrôler directement l'humanoïde à partir des résultats du traitement.

Implémentation Le calcul des cartes de saillance est un processus souvent lourd et coûteux en terme de temps de calcul. C'est un calcul qui à l'heure actuelle ne peut s'effectuer en temps-réel (analyse multi-résolution par bancs de filtres de Gabor multi-orientation). Cependant, notre démarche est motivée par des inspirations biologiques et physiologiques. Dans la littérature, le temps moyen associé à la première fixation est d'environ 320 ms. Le calcul de la carte de saillance peut s'effectuer dans un processus parallèle synchronisé avec le comportement. L'idée est de pouvoir avoir un temps de calcul avoisinant les temps de fixation, ce qui nous permet de maintenir une liste des cibles à jour après avoir épuisé une liste établie quelques secondes auparavant. Cependant le décalage entre le moment où la carte de saillance est calculée et le moment où l'humanoïde regarde nous impose de conserver la configuration dans laquelle l'humanoïde se trouve au moment de l'acquisition de l'image, pour pouvoir effectuer une phase de recalage de la cible. Des détails sur l'exécution sont donnés dans la partie *Résultats* associée à ce chapitre.

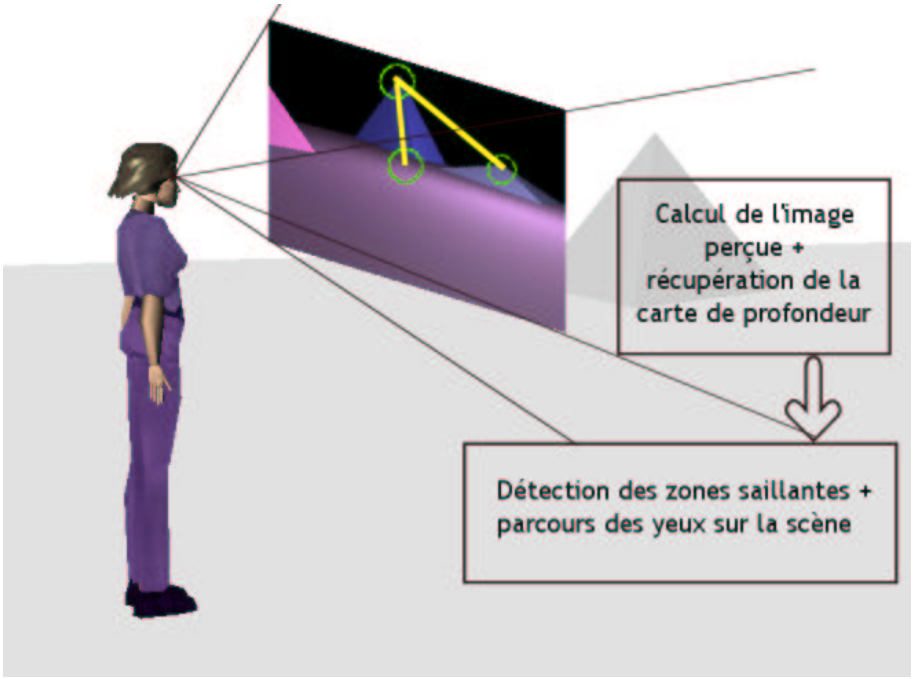


FIG. 6.7 – *Utilisation du principe de saillance*

Validation théorique ? Un des problèmes associé à l'emploi de cette méthode est la validation associée au processus. Disposer d'un système d'analyse plus proche des capacités humaines permet en théorie de simuler d'une manière plus appropriée le comportement humain. Cependant, à notre niveau, l'analyse dépend grandement du moteur de rendu calculant l'image, ainsi que de la modélisation du monde virtuel. A titre d'exemple, l'utilisation d'un rendu utilisant les notions de spécularité ou de transparence générera des images possédant des artefacts de lumière qui pourront être traités comme des points saillants dans l'image. De ce fait, la validation par comparaison avec des modèles réels est difficile, le niveau de détail présent dans les images que nous observons étant nettement plus important que dans un monde synthétique.

Importance relative du mouvement : limitation L'utilisation de cartes de saillance permet de déterminer quels sont les zones d'intérêt uniquement sur une image fixe. Cette information reste insuffisante, dans la mesure où le mouvement joue un rôle important dans la notion de conspicuité. Ainsi un objet en mouvement dans un environnement statique va en priorité attirer notre attention. Il faut par conséquent dans une architecture simulant la perception visuelle d'un humanoïde disposer d'un module scrutant l'environnement à la recherche des objets en mouvement visibles par l'humanoïde. Ce travail peut se faire par questionnement de la base de données, ou par analyse d'images successives (détection du mouvement dans une séquence d'images). L'intégration de cette fonctionnalité n'est pas présentée dans ce document.

6.2.4 Dualité du contrôle

Nous venons de décrire la manière dont nous traitons distinctement deux des aspects fondamentaux de la perception visuelle humaine : une perception active (processus endogène) et un perception passive (processus exogène). Cette partie est consacrée à la manière de mettre en relation/compétition ces deux processus.

6.2.4.1 Concurrence avec la notion de ressources

Nous étudions ici la manière de faire coexister les deux modalités perceptives présentées. Du point de vue de l'animation, l'exécution de ces deux comportements requiert de **prendre le contrôle sur la chaîne cinématique** (ressource de vision) impliquée dans le processus de perception. Cette ressource est donc partagée entre ces deux modalités. La cohérence est alors définie comme le fait de pouvoir prendre à tour de rôle le contrôle sur la chaîne articulaire de manière logique et relativement à un certain nombre de critères cognitifs que nous allons définir. Pour gérer la coexistence de ces deux comportements, nous utilisons une méthode similaire à celle décrite dans la partie 5.2.2.3 sur l'enchaînement de plans et se basant sur une théorie développée dans la partie 4.2.2 (ordonnancement des comportements). Les comportements de perception endogène et exogène s'exécutent en parallèle, et sont en compétition sur une ressource unique : la vision. Une telle approche permet de concevoir séparément les multiples comportements utilisant cette ressource. Il faut alors prévoir pour chacun de ces comportements une adaptation possible, c'est à dire un biais par lequel la ressource de vision est relâchée. Une fois cette adaptation modélisée, il faut pouvoir définir selon quel critère on peut passer d'un comportement à un autre, c'est à dire faire évoluer la priorité associée à chaque comportement de perception. On peut distinguer deux classes de paramètres faisant évoluer ces priorités :

- **paramètres conscients**. Entreprendre une action peut imposer l'utilisation du regard. Par exemple, prendre un objet nécessite l'utilisation de la vue. La priorité est alors directement lié à la nécessité de réaliser l'action,
- **paramètres inconscients**, ou de l'ordre du réflexe. Ces paramètres font référence aux paramètres cognitifs évoqués précédemment. La distractibilité, la capacité de concentration ou les notions de surcharge cognitive en font partie. De la même manière, une valeur de saillance élevée peut amener l'humanoïde à prendre la ressource de vue pour observer cet objet. Un objet en mouvement peut aussi attirer l'attention de l'humanoïde.

On note que le comportement de perception par défaut est un comportement de type endogène : le regard de l'humanoïde est attiré par les zones saillantes de l'environnement.

Il est nécessaire de préciser que la conception de tels systèmes de priorité n'est pas forcément aisée, dans le sens où les différentes pondérations associées à ces priorités se déterminent de façon plus ou moins empirique. En effet, même si l'on peut souligner par ce biais des traits de caractère ou de personnalité (par l'utilisation de paramètres issus des sciences cognitives), il est difficile de quantifier respectivement ces caractéristiques en terme de valeurs numériques.

6.2.4.2 Environnements et objets informés : accès à des informations sémantiques

Les différents mécanismes que nous venons de mettre en œuvre ne permettent pas encore à proprement parler de récupérer des informations sur l'environnement. Réalisée au niveau cortical selon des mécanismes encore mal définis, l'étape de l'identification et de la reconnaissance des objets présente de nombreuses difficultés du point de vue de la modélisation. Notre approche est singulièrement différente, dans le sens où l'on considère que l'information sémantique associée à un objet est contenu dans celui-ci. Cette optique est en ce sens proche du modèle des "affordances" établi par Gibson [Gibson 79] et exploité en synthèse d'images par Kallmann [Kallmann 99]. Au travers du processus de perception passive, l'humanoïde déplace son regard sur différents éléments de la scène. Une fois son regard positionné sur ces objets, on peut se permettre de lancer **un seul rayon** au travers de la base de données (éventuellement partitionnée) pour récupérer un identifiant de l'objet regardé (le premier objet intersecté). Ce lancer de rayon préserve les contraintes de visibilité (par construction). Il possède la bonne propriété d'être très peu coûteux en terme de temps calcul (ce qui n'est plus vrai lorsqu'on balaye l'environnement avec un faisceau de rayons). L'objet intersecté, s'il présente un intérêt, contient alors des informations sémantiques qui peuvent être récupérées par l'humanoïde aux travers d'un ensemble de filtres pouvant modéliser sa "subjectivité". Ces informations peuvent entre autre modifier le processus attentionel de l'humanoïde. Par exemple, dans un comportement de recherche d'un objet, si l'on sait qu'on vient de voir l'objet recherché, ce comportement arrive alors à sa terminaison. Cet aspect n'a pas été abordé au cours de cette thèse (mais fait partie des perspectives).

Un intérêt majeur à cette information de l'environnement est qu'on peut renseigner l'humanoïde sur les manières d'observer l'objet. En effet, un exemple d'information pouvant être contenue dans ces objets peut être un ensemble de primitives visuelles (au sens de notre système d'animation), éventuellement liées par un système d'interpolation précis. L'humanoïde, percevant ainsi cet objet, va alors générer une animation correspondante qui sera propre à l'objet (manière de regarder), et à la configuration de l'humanoïde. Cette information sur *comment regarder l'objet* est ainsi modélisable grâce aux facilités de spécifications associées à notre système d'animation. A titre d'exemple, considérons le cas où l'humanoïde perçoit un panneau de signalisation. Ce panneau ne peut être humainement lu que si l'on se trouve à une certaine distance de lui. Cette information de distance peut se traduire par l'utilisation d'un ensemble de points comme primitives visuelles de base (les quatre coins du panneau par exemple). Accéder à cet objet informé signifie ainsi "*réalise l'asservissement selon les primitives contenues dans l'objet*".

Cette utilisation possible magnifie la nature entrecroisée des interactions entre les différents processus de perception. La figure 6.8 montre un schéma décrivant la chaîne fonctionnelle réalisant cette interaction.

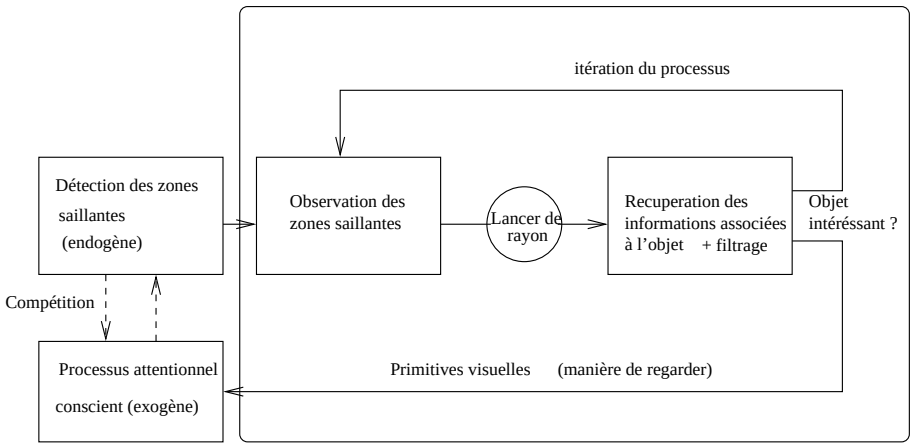


FIG. 6.8 – *Interaction entre les processus attentionnels endogène et exogène*

6.3 Mise en œuvre et résultats expérimentaux

Des résultats associés à cette partie sont présentés dans ce paragraphe. Ceux-ci recoupent les différents thèmes abordés et sont disposés selon une progression identique à la partie théorique que nous venons de décrire.

6.3.1 Comportement de surveillance

Le comportement étudié ici est un exemple typique de comportement de surveillance. Notre personnage, *nonos*, marche dans un environnement virtuel. Le comportement mis en œuvre est celui décrit dans l'automate de la figure 6.5 (a). Un système de tirage aléatoire de cibles a été modélisé. La figure 6.9 montre une séquence d'images provenant de l'exécution d'un tel comportement. Le seuil de confiance, défini entre 0 et 1, est ici fixé à 0,2, ce qui correspond à un niveau de confiance faible vis-à-vis du sol. La fréquence des coups d'oeil jetés au sol est d'environ un toutes les trois secondes (ce qui est relativement élevé). Le résultat visuel est assez significatif. Les coups d'œil jetés par nonos étant assez rapides (gain élevé), une impression de "malaise" ressort du personnage, ce qui d'une certaine manière valide l'utilisation d'un paramètre cognitif lié à la connaissance du lieu. Pour un paramètre plus élevé, cette impression disparaît. L'utilisation d'un paramètre cognitif pour la définition des comportements permet d'envisager une évolution dynamique de ce paramètre au cours de l'animation. Ce paramètre devient alors une des modalités de contrôle du comportement.

6.3.2 Perception passive

Cette partie décrit l'utilisation de modèles de comportements de type exogène à base de cartes de saillance. Dans un premier temps nous décrivons le type de cartes de saillance utilisées, puis nous présentons les parcours visuels obtenus sur des scènes virtuelles, avant de montrer les types d'animations résultants de l'utilisation d'un tel système.

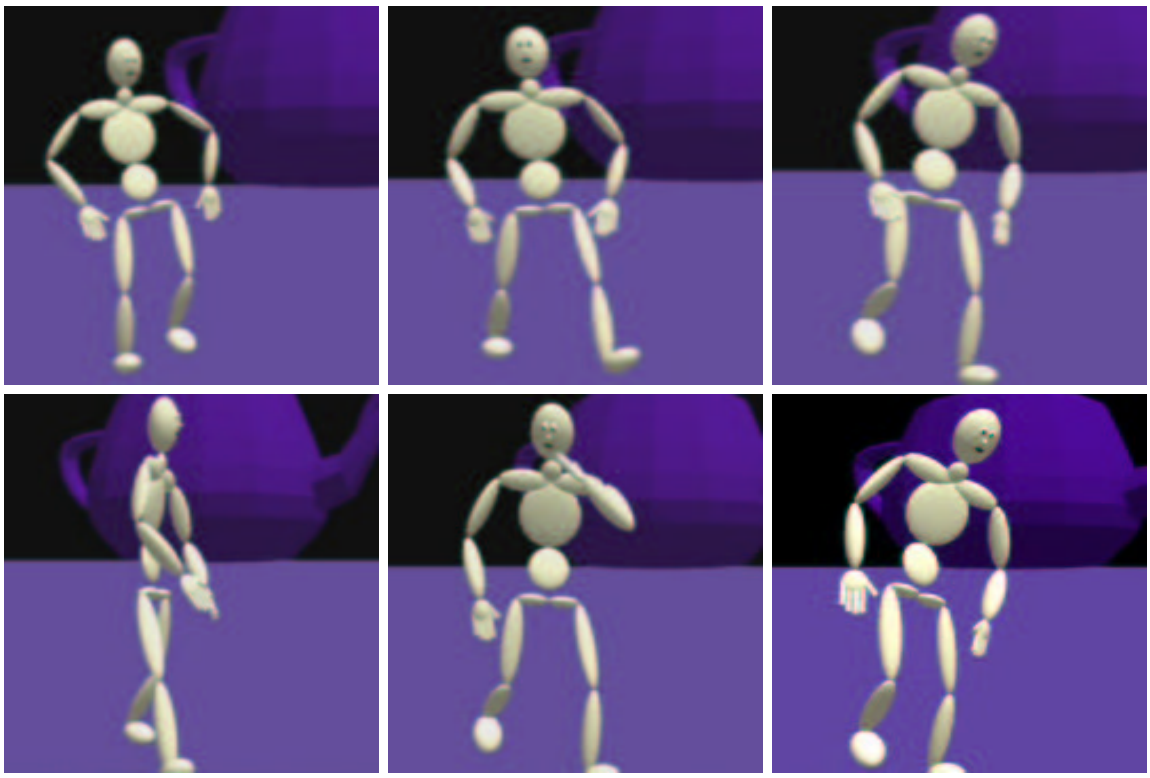


FIG. 6.9 – *Extrait de l'animation où nonos avance dans un environnement inconnu*

6.3.2.1 Définition d'un modèle de cartes de saillance

Nous avons mis en œuvre un modèle de cartes de saillance simple pour pouvoir tester l'architecture de notre système. Ce modèle repose sur deux informations précises : informations spatiales fréquentielles et profondeur. Il est envisageable de compléter ce panel avec d'autres types d'informations visuelles (la couleur par exemple). Ces deux informations sont extraites en parallèle dans deux cartes de traits différentes, puis combinées entre elles. Nous allons voir comment ces informations sont extraites puis combinées.

Acquisition de l'image et de la carte des profondeurs Pour obtenir l'image perçue par l'humanoïde, on utilise directement le moteur 3D utilisé par l'application ; il suffit alors de rendre une image de la scène depuis le point de vue de l'humanoïde. La phase de rendu nous permet d'obtenir directement la carte des profondeurs ou *Z buffer*, car c'est une information nécessaire au processus de rendu. A l'heure actuelle, la plupart de ces opérations sont directement câblées sur les cartes graphiques, ce qui rend l'accès à ces informations pratiquement instantané (cadence d'affichage). On note qu'il est possible de paramétrer dans la plupart des moteurs 3D existants le point de vue depuis lequel est rendu l'image, et notamment les dimensions de la pyramide de vue. Dans la plupart des tests que nous avons effectués, l'angle d'ouverture de cette pyramide a été choisi de manière à respecter la taille du champ de vision humain, c'est à dire en moyenne une ouverture d'environ cent vingt degrés (ouverture du cône de vision).

Analyse en orientation et en fréquence Nous avons utilisé un banc de filtres de Gabor pour filtrer l'image perçue par l'humanoïde. Ces filtres permettent d'obtenir une partie du spectre de l'image selon différentes orientations, en phase et en amplitude. Ces bancs de filtres possèdent de très bonnes propriétés, en ce sens qu'ils modélisent le comportement de la rétine [Marcelja 80]. Ce sont des filtres complexes orientés passe-bande, composés d'une gaussienne modulée en fréquence. La fonction que nous avons utilisée est définie par :

$$Gabor_{f_{x_0},f_{y_0}}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}) exp(j2\pi(f_{x_0}.x + f_{y_0}.y)) \qquad (6.1)$$

où σ_x et σ_y correspondent respectivement aux largeurs de bande en x et y, et f_{x_0} et f_{y_0} les orientations selon x et y. Les gaussiennes que nous avons utilisées sont circulaires ($\sigma_x = \sigma_y$), et forment des couronnes dans l'espace fréquentiel : $f_{x_0} = f.cos\theta_k$ et $f_{y_0} = f.sin\theta_k$, avec f le rayon de la couronne et θ_k l'orientation des filtres.

Les images sont donc traitées à l'aide de ces filtres selon différentes échelles et différentes orientations. Une analyse multi-échelle nous permet de prendre en compte des informations importantes à un niveau local ainsi qu'à un niveau global. Toutes les images résultantes du filtrage sont alors combinées dans une seule et même image contenant le résultat du filtrage à chaque orientation et à chaque échelle. La figure 6.10 montre le résultat d'un tel filtrage sur des images synthétiques. L'exemple (a) montre un champ de fleurs. Les fleurs apparaissent clairement comme des objets saillants dans l'environnement. On note que l'absence d'anti-aliasing dans le rendu de l'image fait apparaître un crénelage sur les lignes droites (l'horizon par exemple). Ce crénelage induit des artefacts dans le résultat du filtrage de Gabor, faisant apparaître des zones saillantes sur des endroits où il n'y en a pas théoriquement. On s'aperçoit dans cet exemple que la qualité des résultats dépend donc nettement de la qualité du rendu. Une manière simple de supprimer ces artefacts consiste à passer un filtre gaussien sur l'image (ayant pour effet d'adoucir les contours). Dans l'exemple (b), un humanoïde apparaît sur fond neutre. Le résultat de l'analyse magnifie bien ce personnage, et plus particulièrement les zones disposant d'un haut niveau de détail comme le visage. Les exemples (c) et (d) montrent le traitement d'images plus complexes, disposant notamment de textures et de nombreux objets. Le résultat du traitement fait apparaître de multiples zones saillantes sur les images, disposées un peu partout sur celles-ci.

Combinaison des cartes de traits Une fois le filtrage de l'image perçue réalisé, l'image filtrée est normalisée puis combinée à l'image correspondant à la carte de profondeurs. Les objets plus près sont donc magnifiés par rapport aux objets plus distants. Cette notion se justifie par l'aspect stéréoscopique de la vision humaine. Nous partons du principe que les objets plus prêts de nous attirent plus facilement notre regard [Wolfe 96] (bien que ce principe ne soit pas vérifiable dans tous les cas).

Dans un deuxième temps, il faut tenir compte d'une deuxième spécificité neuro-physiologique : le centre de la rétine, appelé fovéa, est l'endroit le plus précis de l'image[1]. L'image est donc

1. La fovéa n'est composée presque uniquement que de cônes, et chacun de ceux-ci est connecté à un seul neurone, connecté lui-même au cerveau. A la périphérie de l'oeil, la vision semble beaucoup plus floue, le

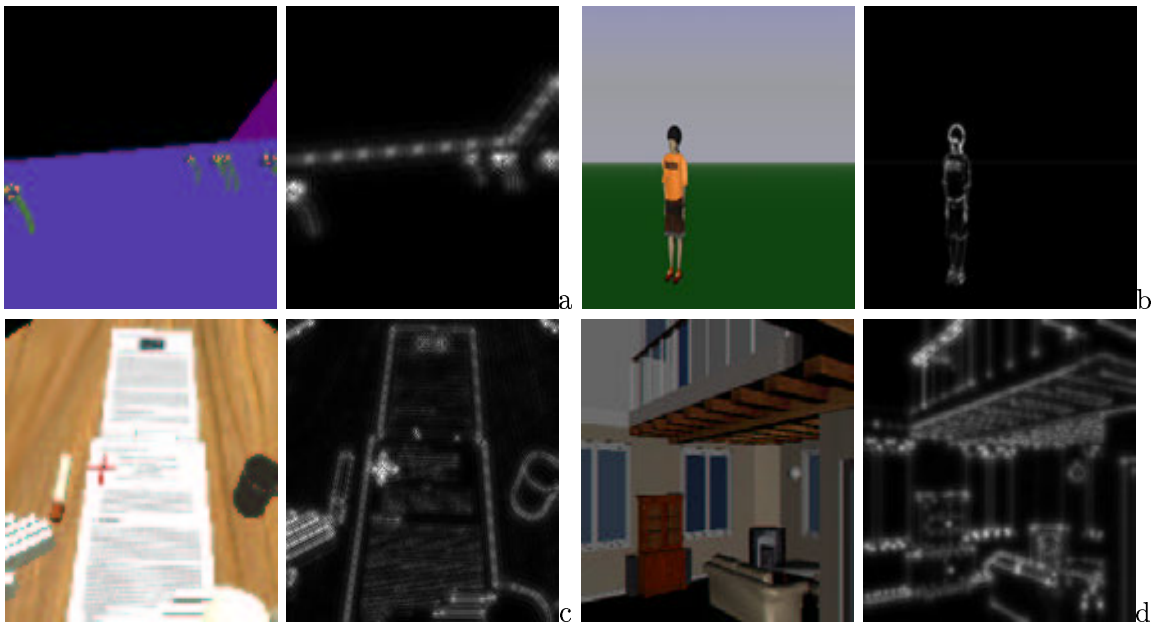


FIG. 6.10 – *Différents résultats de l'utilisation d'un banc de filtres de gabor sur différentes images synthétiques. Pour chacune des images, quatre niveaux d'échelle sont utilisés avec huit orientations de filtres différentes*

plus précise au centre de la rétine qu'en périphérie ; c'est pour cela que nous avons toujours l'habitude de focaliser l'élément que nous regardons au centre de notre rétine. Ce phénomène est appelé *center-surround* en anglais. Pour modéliser cette spécificité, nous combinons notre image avec l'image d'une gaussienne dont la largeur est paramétrable. Le but de cette combinaison est de pénaliser les éléments se trouvant au bord de l'image par rapport aux éléments au centre de celle-ci. La taille de cette gaussienne peut ainsi être modifiée au cours du temps pour traduire différents états physiologiques de la personne. A titre d'exemple, une consommation éhontée d'alcool réalisée par notre humanoïde ferait à la fois diminuer l'ouverture du champ de vision ainsi que le rayon de cette gaussienne.
La figure 6.11 montre la chaîne de traitement permettant d'aboutir à la carte de saillance finale. L'image originale est un champ de fleurs, avec une pyramide en fond. cette image est délibérément "simple" dans un but didactique.

6.3.2.2 Parcours des yeux sur la scène

Une fois la carte de saillance définitive obtenue, il convient de déterminer une succession de zones d'intérêts appelés zones de fixation. Ces zones d'intérêt sont en fait les maxima d'énergie dans la carte de saillance. Un parcours simple de l'image permet de déterminer rapidement le premier maximum (première zone saillante). Une fois sa position récupérée, on découpe alors l'image selon une zone de taille déterminée empiriquement et fonction de l'ouverture du cône de vision [Itti 98], pour éviter qu'une seule zone de l'image n'attire tous les points de fixation. Le processus est alors réitéré un nombre de fois équivalent au

ratio de photo-récepteurs par neurone étant plus grand qu'au centre de l'oeil. C'est cependant en périphérie que l'on perçoit le mieux le mouvement.

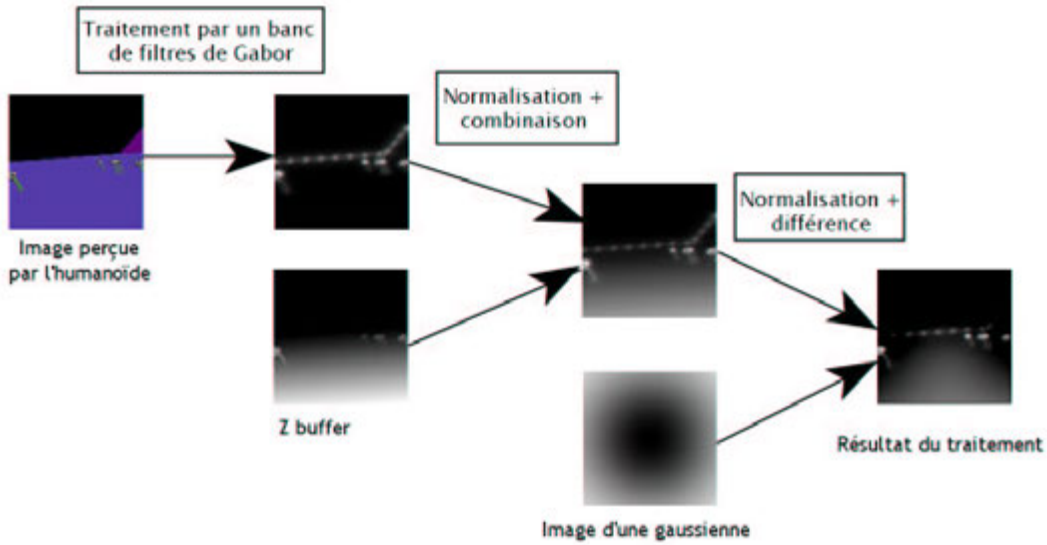


FIG. 6.11 – *Chaîne de traitement aboutissant à la création de la carte de saillance*

nombre de zones de fixation souhaité. Sur la figure 6.12 est montré la succession d'étapes conduisant à l'obtention d'une série de quatre zones de fixation. L'image originale est l'image traitée dans la figure 6.11. A l'issu de ce traitement, on constate que les quatre zones fixation ont été déterminées sur les fleurs présentes dans l'environnement. Ces éléments sont effectivement les seuls éléments "originaux" ou "importants" de la scène.

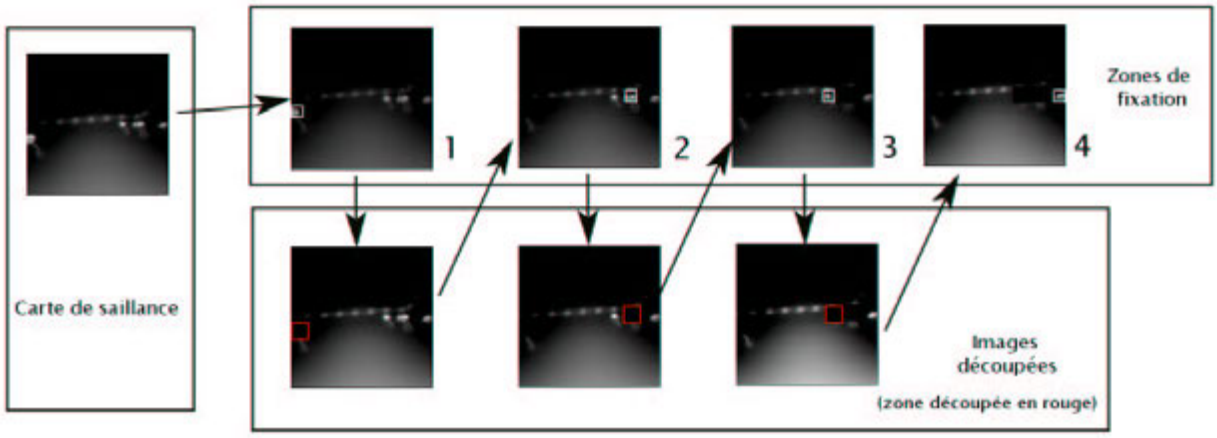


FIG. 6.12 – *Calcul d'une suite de quatre zones de fixation, les carrés blanc montre les zones possédant un maximum d'énergie, et les carrés rouges montrent les zones découpées de l'image*

6.3.2.3 Lien avec le système d'animation

Le temps de calcul de la carte de saillance et des zones de fixation est actuellement d'environ 1 seconde sur une station SGI 330 disposant de processeurs Athlon cadencés à 1 Ghz. Ces temps sont obtenus sur des images de dimension 128x128, avec un filtrage de Gabor comprenant 4 niveaux et 8 orientations de filtres par niveau. Le temps de calcul est directement lié à la complexité de la phase de filtrage, et évolue exponentiellement avec la taille des images traitées ainsi que le nombre d'échelles et d'orientations utilisées. Le fait que ce calcul ne se fasse pas à la cadence image impose d'une part de faire s'exécuter ce calcul dans un processus à part (ce qui implique une synchronisation entre les différents processus, réalisée par l'utilisation d'HPTS), et de tenir compte de cette latence au niveau du système d'animation. Ainsi, la configuration de la chaîne articulaire utilisée est mémorisée lorsqu'on commence un nouveau calcul (cette configuration pouvant avoir été modifiée à la fin du calcul). A l'issu du calcul, les points 2D résultats servent à évaluer un point 3D ${}^{c}\mathbf{P}(\mathbf{x},\mathbf{y},\mathbf{z})$ dans le repère des yeux. Si X et Y sont les résultats en pixel du calcul de saillance, I_X et I_Y respectivement la largeur et la longueur de l'image traitée, et θ_X et θ_Y respectivement les angles d'ouverture en largeur et en hauteur de la pyramide de vision, on a :

$$\begin{cases} x = Z.tan(\frac{X}{I_X}.\theta_X) \\ y = Z.tan(\frac{Y}{I_y}.\theta_Y) \\ z = Z \end{cases}$$

où Z est la profondeur du point, récupérée à partir de la carte de profondeur (*Z-buffer*). Un changement de repère est en suite effectué pour déterminer la position du point ${}^{c}\mathbf{P}$ dans le repère absolu, puis introduit dans la loi de commande. Une fois l'asservissement réussi sur une fixation, on passe à la fixation suivante. L'automate utilisé pour réaliser cette synchronisation est montré dans la figure 6.13. On note ici que l'utilisation de notre technique d'animation référencée vision n'est pas exclusive. Regarder un point peut aussi se faire par cinématique inverse simple.

6.3.2.4 Animation résultante

Nous avons testé cette technique sur un environnement de type urbain. Un mouvement de marche a été donné de manière arbitraire à notre personnage (Sara). La figure 6.14 montrent des images issues de cette animation. On peut voir sur ces images que Sara jette de nombreux coup d'oeil à gauche et à droite.

La figure 6.15 montre des cartes de saillance calculées lors de son parcours. On constate que la plupart des objets importants comme le panneau d'affichage, la maison, les parcmètres sont observés au cours du processus. On remarque aussi que Sara regarde systématiquement la bordure du trottoir à coté d'elle ; ceci s'explique par deux raisons : c'est tout d'abord un élément saillant au sens fréquentiel, et en plus c'est un des éléments les plus proches d'elle. Cet état de fait montre clairement les limitations d'un tel système, à chaque nouvelle carte calculée les nouveaux points de fixation sont générés d'une manière totalement indépendante des résultats antérieurs. Il faudrait pouvoir introduire une notion de mémoire visuelle, qui éviterait une redondance non naturelle dans l'observation des objets de la scène. De plus, certains éléments perçus comme saillant peuvent ne présenter aucun intérêt (le sol par exemple, ou bien des textures répétitives riches en détail, comme

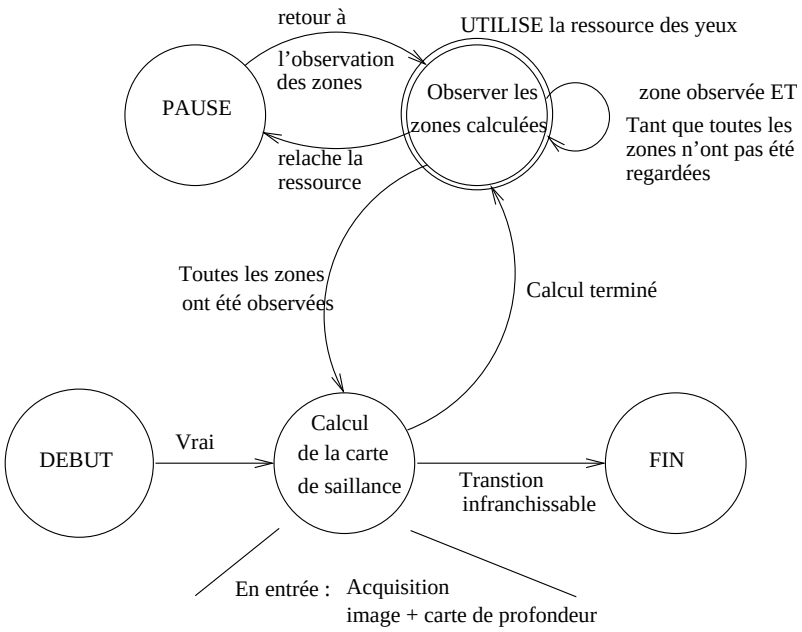


FIG. 6.13 – *Automate servant à la synchronisation du calcul des zones saillantes et de leurs observations. L'état* **Pause** *sert à relâcher la ressource des yeux si un comportement plus prioritaire intervient.*

une façade d'immeuble ou un sol carrelé). Une deuxième perspective serait de pouvoir sur demande supprimer des éléments de la scène rendue du point de vue de l'humanoïde (un rendu sélectif en quelque sorte). Cette possibilité pourrait alors être une solution aussi au problème de redondance de fixations d'une carte à une autre. Cependant, cette technique pose un problème fondamental qui est le problème de visibilité : si on élimine un objet du rendu, les objets derrière deviennent alors visibles, et donc peuvent participer au processus de sélection.

6.4 Conclusion

Au cours de ce chapitre, nous avons développé un formalisme permettant de mettre en place une architecture pouvant simuler la perception visuelle d'un humanoïde de synthèse. Après avoir singularisé deux modalités perceptives distinctes : une perception endogène et une perception exogène, nous avons établi deux méthodes permettant de traiter ces deux aspects. Dans les deux cas, la cinématique associée à l'acte de perception visuelle a été réalisée grâce au système d'animation référencée vision décrit dans la première partie de cette thèse. L'utilisation d'HPTS (protocole de définition des comportements) nous a permis de définir des comportements pouvant intégrer des paramètres de type cognitif, ainsi que des comportements de perception passive basés sur un modèle de cartes de saillance. L'utilisation de cette approche, nouvelle dans le domaine, constitue une solution intéressante au problème. A la différence des travaux réalisés dans le domaine de la perception visuelle d'un humanoïde de synthèse, nous combinons les deux aspects *environnements informés* et *acquisition d'informations à partir de l'image perçue par l'humanoïde*. La coexistence

FIG. 6.14 – *Extraits de l'animation relative à l'utilisation d'une carte de saillance dans un environnement urbain*

de ces deux aspects permet de se rapprocher de manière significative de la manière dont l'être humain perçoit les objets.

Perspectives Il serait intéressant d'étudier, dans le cadre de la poursuite des travaux menés dans cette thèse, des comportements complexes mêlant à la fois des modalités perceptives endogène et exogène. Un exemple typique de comportement réunissant ces deux aspects est le comportement de recherche d'un objet dans un environnement connu ou partiellement connu. Plusieurs éléments sont à envisager. Tout d'abord le module de perception visuelle doit pouvoir être connecté à un module de raisonnement (module décisionnel de haut niveau) pouvant guider la recherche visuelle. Un module de mémoire semble aussi important, dans la mesure où la définition d'une représentation interne (servant de base au module de raisonnement) est obligatoire pour éviter toute redondance dans la recherche et par souci d'efficacité. En observant de nouveaux objets et lieux, l'humanoïde perçoit passivement de nouvelles entités. Si il décide de les observer, alors de nouvelles choses entrent dans son champ visuel. C'est à ce niveau que joue donc la coopération/compétition des deux modalités de perception. Si ces interactions sont orchestrées par le module décisionnel (selon des informations relatives à la mémoire), le comportement émergent peut s'avérer très intéressant à étudier.

FIG. 6.15 – *Différentes cartes de saillance établies lors de la promenade de Sara. La première colonne montre la vue subjective associée au personnage. La deuxième montre les cartes de profondeur associées, et la troisième le résultat du calcul des zones saillantes, avec l'ordre d'obtention des fixations*

Chapitre 7

Validation pratique de ces méthodes

Dans le cadre de ce chapitre, deux applications distinctes sont présentées, utilisant dans une plus ou moins grande mesure les travaux réalisés dans le cadre de notre thèse. La première de ces applications est l'ajout d'une fonctionnalité à un agent communicant réalisé par France Telecom R&D, où un avatar suit du regard un utilisateur (qui regarde l'écran). La deuxième est une fiction interactive, démonstration réalisée à l'occasion du festival Imagina 2002 de Monte-Carlo.

7.1 Agent conversationnel

Cette thèse s'est déroulée dans le cadre d'une collaboration avec le laboratoire de recherche de France Telecom R&D. Un des axes de recherche concernant la synthèse d'images est la création d'un agent communiquant, humanisant l'interface entre les services proposés par France Telecom et les utilisateurs potentiels. L'agent communiquant est soit un avatar, soit un clone d'une personne existante. Deux formes de communication existent entre l'entité synthétique et l'utilisateur : une communication verbale et non-verbale. La communication verbale se traduit par l'utilisation d'un système de synthèse vocale, permettant de synthétiser des phrases dans différentes langues et selon différents schémas (voix d'homme, de femme, d'enfant). Cette synthèse est accompagnée d'une animation faciale cohérente. Il est établi que cette communication verbale ne représente qu'une partie des modalités de communication [Cassell 00,Gibet 01]. La plupart des gestes accompagnant tout discours contribue directement à notre compréhension. Dans ce domaine, l'importance du regard est primordiale [Cassell 00], et donne de nombreux indices sur l'état émotionnel de l'interlocuteur. Pour renforcer la crédibilité de l'agent communicant, il a donc été imaginé de lui adjoindre une vision "réelle" de l'utilisateur par ajout d'une *webcam*, le but étant de faire regarder l'avatar dans la direction des yeux de l'utilisateur. Cette partie traite de la manière dont a été résolu ce problème.

7.1.1 Principe

Le principe de l'application est assez simple. Un algorithme de détection de visage est utilisé pour récupérer à chaque pas de temps la position du visage de l'utilisateur dans l'image perçue de la webcam. A partir de cette information, une animation est générée par

notre système d'animation, et le rendu est exécuté par le moteur d'animation faciale. La figure 7.1 montre une représentation de cette architecture.

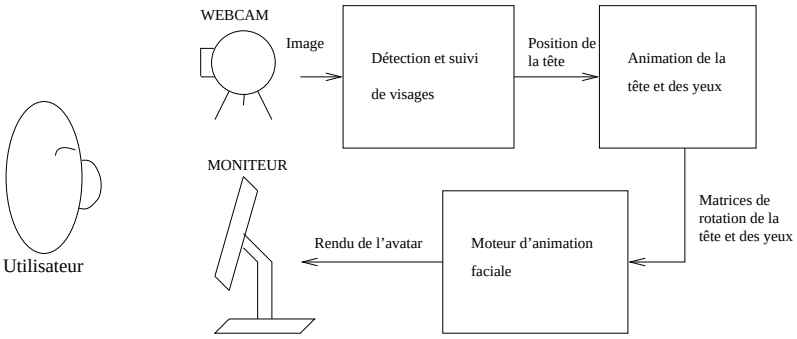


FIG. 7.1 – *Principe de fonctionnement*

Chacun de ces composants va maintenant être étudié dans le détail, le premier paragraphe traitant des outils utilisés, et le second de la manière dont nous avons utilisé le système d'animation.

7.1.1.1 Architecture

Algorithme de détection et de suivi de visages Nous avons utilisé un algorithme de détection et de suivi de visages réalisé par Olivier Bernier [Feraud 01] pour France Telecom, pouvant tourner en temps réel. La méthode utilisée est basée sur l'utilisation d'un modèle statistique de suivi basé sur deux informations : couleur des pixels correspondants au visage et position de certaines parties du visage telles que les yeux, la bouche, le nez et les sourcils, donnée par un filtre.

Visages parlants Le module FaceEngine, développé par Gaspard Breton [Breton 00] pour France Telecom, est un moteur d'animation qui permet d'afficher en temps réel un visage animé. Le modèle du visage est un maillage disposant d'un squelette minimal (cou et visage), composé d'un ensemble de maillages 3D détenant leurs propres caractéristiques de textures et matériaux. L'animation faciale est réalisée en mixant deux techniques différentes de déformation de maillage. Le modèle paramétrique permet d'animer une partie entière du visage par rotations (correspondant aux articulations réelles du crâne : cou, mâchoire, ainsi que les paupières, les yeux et le cou). Le maillage ainsi obtenu est par la suite déformé de façon locale (autour du muscle) par un système d'animation musculaire. Chaque expression faciale peut ainsi s'exprimer par un ensemble de contractions de muscles, de façon simple et peu coûteuse. La génération des visèmes (expression du visage et des lèvres relatives à la prononciation d'un son ou phonème) est réalisée en collaboration avec le module de synthèse vocale. Le réalisme de l'animation repose sur la synchronisation fluide des mouvements des lèvres avec la parole ; il peut être enrichi par l'insertion dans le texte d'entrée de marqueurs conversationnels contrôlant les expressions du visage, ce qui permet à l'humanoïde d'afficher des émotions ou d'insister sur un point précis (emphase).

7.1.1.2 Définition de la loi de commande

Pour animer l'avatar, seul l'information issue du tracking est disponible (coordonnées en pixel du centre du visage), ainsi que les paramètres intrinsèques de la webcam. A partir de la taille du cadre englobant le visage, on estime la distance du visage à la webcam. Cette estimation peut être grossière car elle influe peu sur les résultats. A partir de cette distance la position du visage ${}^{o}\mathbf{P}$ de l'utilisateur est donc estimé dans le repère de la webcam. Il faut alors évaluer la matrice ${}^{a}\mathbf{M_o}$ définissant le changement de repère entre les yeux de l'avatar et le repère absolu (webcam). Cette estimation est empirique car elle ne repose sur aucune réalité tangible (mise en commun d'un monde virtuel et réel...). Pour les besoins de l'application, une simple translation a été considéré entre le repère absolu du monde dans lequel évolue l'humanoïde et le repère absolu de la webcam. Quand l'humanoïde est centré dans son propre repère, il doit être au centre de l'écran. Connaissant la matrice reliant les yeux de l'avatar au repère absolu du monde virtuel ${}^{yeux}\mathbf{M_a}$, il est alors possible d'estimer la position ${}^{yeux}\mathbf{P}$ de l'utilisateur dans le repère des yeux de l'avatar par :

$$ {}^{yeux}\mathbf{P} = ({}^{yeux}\mathbf{M_a}\,{}^{a}\mathbf{M_o}).{}^{o}\mathbf{P} $$

Cette information est alors directement introduite dans la loi de commande. On note que cette information est remise à jour à chaque pas de temps. Le schéma de la figure 7.2 montre l'agencement des différents éléments utilisés.

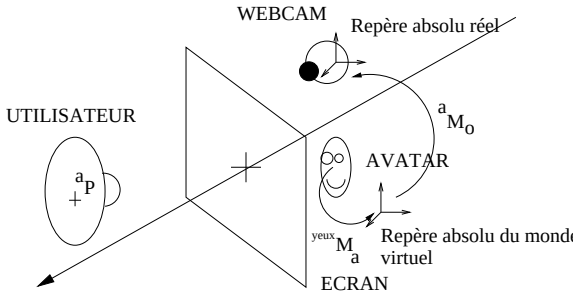


FIG. 7.2 – *Schéma représentant les différents éléments utilisés par le système d'animation*

7.1.2 Résultats

L'exécution de cette application s'est faite sur des Pentium 4 cadencés à 1 Ghz, avec une carte graphique de type GeForce2. C'est la détection des visages qui est la partie la plus gourmande en temps de calcul (tournant à 10 images par seconde), alors que l'animation et le rendu s'est effectué en temps réel (plus de 70 images par seconde sur cette configuration). Cette légère latence dans la détection peut amener certains artefacts dans l'animation du visage, comme une certaine lenteur dans l'exécution de mouvements amples (par exemple si l'on se déplace rapidement dans la fenêtre d'acquisition). Cependant les résultats sont très encourageants, car le simple ajout du suivi visuel de l'utilisateur par l'avatar rajoute grandement en cohérence et en réalisme au personnage synthétique, et ce d'autant plus que l'acquisition des images par la webcam est très discrète (un utilisateur non informé du procédé est rapidement très surpris). Les images présentées dans la figure 7.3 montrent le résultat de la mise en place de ce système.

Perspectives Comme établi dans l'introduction, le regard de l'avatar fait partie des modalités de communication non-verbale. Pour l'instant, le suivi mis en œuvre n'est pas du tout relié au texte prononcé par l'avatar. Il serait intéressant d'ajouter une notion de comportement visuel directement lié à une "personnalité" de l'avatar ainsi qu'au texte dit par celui-ci. Un système tel que *BEAT* développé au M.I.T dans l'équipe de Justine Cassell permet de "décorer" une phrase par des attributs directement liés à la communication non verbale [Cassell 01]. L'idée serait ici de lier un tel système avec la notion de comportements définis dans le chapitre précédent. Ainsi, pour une phrase précise, plusieurs comportements visuels basiques pourraient être enchaînés (toute la difficulté revenant ici à réaliser la synchronisation entre le flot du texte et les comportements). De plus, l'information perçue par la webcam pourrait être utilisée directement pour paramétrer ces comportements. Ces pistes font partie des perspectives de recherche associées à cette thèse.

a

b

FIG. 7.3 – *Suivi du visage et animation d'un agent communicant pour France Telecom R&D. (a) Vue d'ensemble du système (b) Captures d'écran présentant l'application. L'incrustation vidéo représente l'image perçue par la webcam, et la croix en rouge représente le résultat du suivi*

7.2 Exemple de fiction interactive

Dans ce chapitre nous présentons une réalisation ayant pour base le thème de la fiction interactive. Cette démonstration, présentée au festival Imagina2002, présente une unification logicielle et conceptuelle de la majeure partie des travaux présentés dans le cadre de cette thèse. Après avoir présenté le principe, nous étudierons de manière plus formelle la base applicative utilisée (cette base étant commune à la majorité des démonstrations présentées dans cette thèse) et nous présenterons quelques résultats associés.

7.2.1 Présentation

Dans cette partie, nous présentons successivement la notion de fiction interactive ainsi que le thème de la démonstration que nous avons réalisée.

Fiction interactive La fiction interactive désigne toutes les formes de narration dans lesquelles l'auditeur est amené à participer, directement ou indirectement, au contenu de l'histoire. D'un point de vue informatique, la fiction interactive trouve des applications dans le domaine des jeux vidéos, de la production de contenu interactif et artistique, et dans certaine formes de simulation immersive. C'est un domaine relativement récent [Mateas 97]. Plusieurs problèmes majeurs se posent dès lors que l'on étudie ce domaine. A un niveau global, la narration, de part son aspect changeant, impose une structure du récit non linéaire (voir figure 7.4). A différentes étapes du récit, plusieurs possibilités d'évolution sont accessibles, dépendant des choix de l'utilisateur et de l'évolution des personnages.

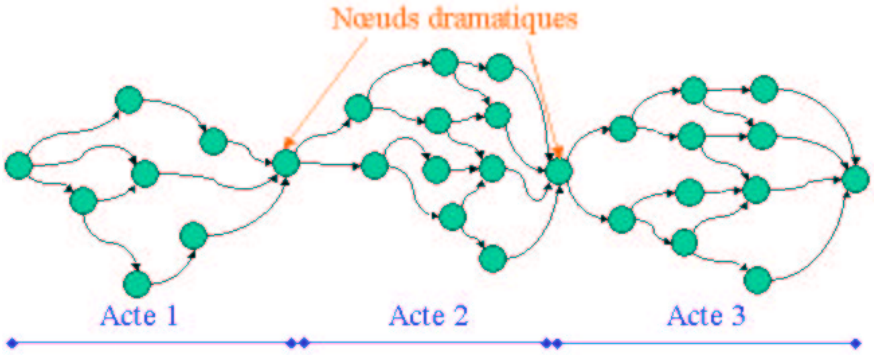


FIG. 7.4 – *Graphe d'états d'une fiction non linéaire*

A un niveau plus local, la mise en œuvre d'une telle application nécessite la prise en compte :

- *d'acteurs autonomes.* Les différents personnages impliqués dans le scénario doivent disposer d'une forte capacité d'adaptation, impliquée notamment par les différentes modalités d'interaction de l'utilisateur. Dans ce cas l'autonomie est un prérequis, car elle formalise ces capacités d'adaptation,

– *d'une gestion de scénario*, qui détermine les grands ressorts dramatiques de l'histoire et influe sur le comportement des acteurs. Il existe alors un caractère paradoxal, car l'autonomie des entités présentes est atténuée par l'aspect dirigiste du scénario. Etablir une frontière entre ces deux aspects constitue un problème fondamental.

Tranche de vie à l'épicerie de nuit Ce paragraphe est dédié à une présentation du scénario de la démonstration. Au travers d'un scénario paramétrable par le spectateur avant et pendant la simulation, deux acteurs, Jean et Sara (présentés dans la figure 7.5), évoluent dans une maison et dans une épicerie. Les paramètres pouvant être modifiés avant le début de la simulation étaient relatifs à l'état psychologique ou physique des deux personnages ; à titre d'exemple Sara peut être enceinte ou non. Au cours de l'exécution de la fiction interactive, l'utilisateur dispose de la possibilité de faire sonner un téléphone. Cet élément, survenant à n'importe quel moment, peut (selon la nature de la conversation téléphonique) modifier le cours du scénario (exemple : une maîtresse éconduite de Jean téléphone, et Sara doute alors de l'intégrité et des motivations de Jean). Par manque de temps, seule cette modalité d'interaction a été prévue pour la démonstration.

FIG. 7.5 – *Présentation de la démonstration de fiction interactive (a) générique de début (b) Jean (c) Sara*

Contributions Quatre personnes ont travaillé sur les problèmes théoriques liés à cette démonstration (sans compter les personnes ayant travaillé sur les bases applicatives qui nous ont servies) : acquisition, traitement et restitution de mouvements capturés (Stéphane Ménardais), planification réactive de chemins dans l'environnement, modélisation des comportements (Fabrice Lamarche), gestion d'un scénario (Stéphane Donikian). La majeure partie de mon travail a consisté à intégrer un système de cinématographie virtuelle et à modéliser des comportements liés à la gestion du regard des deux acteurs. Dans le cadre de la gestion automatique des caméras, différents modes de fonctionnement ont été établis :

– *génériques, ellipses.* Les panneaux relatifs aux génériques, aux ellipses ou aux remerciements ont été traités comme des objets graphiques à part entière. De ce fait, une caméra disposait d'un comportement lui permettant d'enchaîner des prises de vue relatives à ces éléments en fonction du déroulement du scénario,
– *prises de vue associées à un lieu.* Les deux acteurs évoluent dans plusieurs pièces distinctes (épicerie, salon, cuisine). Plusieurs caméras ont été disposées dans ces pièces, le passage d'une caméra à une autre s'effectuant selon des contraintes géométriques relatives à la position des acteurs,

- *prises de vue relatives à une situation.* Certains situations nécessitent une manière de filmer précise. Ainsi les phases de dialogue ou les phases ou Jean répond au téléphone ont été considérées de manière distincte.

La difficulté majeure revient alors à pouvoir enchaîner ces différents comportements. Exécutés en parallèle, la réalisation effective de ces comportements dépend du scénario qui régit les montées de priorité associées à ces différents comportements. A titre d'exemple, lorsqu'une conversation survient entre Jean et Sara, le comportement associé à une discussion devient actif et prend la ressource de point de vue.
Du point de vue de la gestion de l'attention visuellle des acteurs eux-mêmes, plusieurs comportements ont été mis en œuvre :

- *comportements visuels* lors de la marche, de la recherche d'un objet dans une armoire. Ces deux comportements étaient basés sur la génération de cibles aléatoires respectivement sur le sol ou sur l'armoire,
- *couplage préhension-attention.* Lorsqu'un des deux acteurs prend un objet, son regard se porte sur l'objet le temps de la saisie. Inversement, lorsque l'acteur pose l'objet, l'acteur regarde l'endroit où il le pose ,
- *gestion de l'attention visuelle lors d'une conversation.* Lorsque les deux acteurs discutent, le regard de des deux interlocuteurs se posent alternativement sur les yeux de la personne lui faisant face et vers le sol. Ce comportement est paramétré par une variable reflétant la timidité. Ainsi, plus l'acteur est timide, plus la fréquence de ses coups d'oeil vers le sol est augmentée. Cette variable est réglée par le scénario, qui supervise l'état psychologique et interne des deux acteurs.

De manière identique à la gestion des caméras, le scénario active ces différents comportements au cours de l'exécution de l'application. En ce sens, l'autonomie des deux acteurs est relativement restreinte.

7.2.2 Environnement de développement et implémentation

Ce chapitre traite de considérations relatives à l'implémentation de cette démonstration. Les éléments présentés sont ceux qui nous ont servis pour mettre en œuvre la plupart des résultats présentés dans le cadre de cette thèse. La figure 7.6 est un schéma structurel regroupant les différents niveaux de bibliothèques implémentées. Nous avons basé tous les mécanismes de base relatifs à l'animation référencée vision sur une bibliothèque d'asservissement visuel développée au sein du projet Vista de l'Irisa : ViSP (*Visual Servoing Platform*) [Marchand 99]. Ces deux modules de contrôle d'une caméra et de pilotage d'un humanoïde virtuel sont génériques, codés en C++ et intégrables dans n'importe quelle application (et portables sur les environnements *Windows* et *Linux/Unix*). A partir de ces modules génériques, deux instances ont été développées pour la plate-forme d'exécution OpenMask, anciennement Gasp [Chauffaut 97,Donikian 98], qui a servi de base à la réalisation de cette démonstration. Les différents comportements établis se sont à la fois basés sur ces modules et sur HPTS [Moreau 98b,Lamarche 02], notre langage de modélisation de comportements. L'application repose sur ces différents comportements. Une telle approche hiérarchique permet donc la capitalisation des différents travaux. Ainsi les différents comportements (modulo quelques modifications spécifiques au scénario) peuvent être réutilisés dans d'autres scénarios, voire d'autres applications.
On note que l'exécution de cette démonstration est temps réel (en moyenne 25 images par

seconde, cette moyenne dépendant de la complexité des scènes au niveau géométrique) sur une machine de type SGI330 munie de deux processeurs Pentium 4 cadencés à 1Ghz, et d'une carte graphique de type *Quadro* équivalente à une GeForce 2.

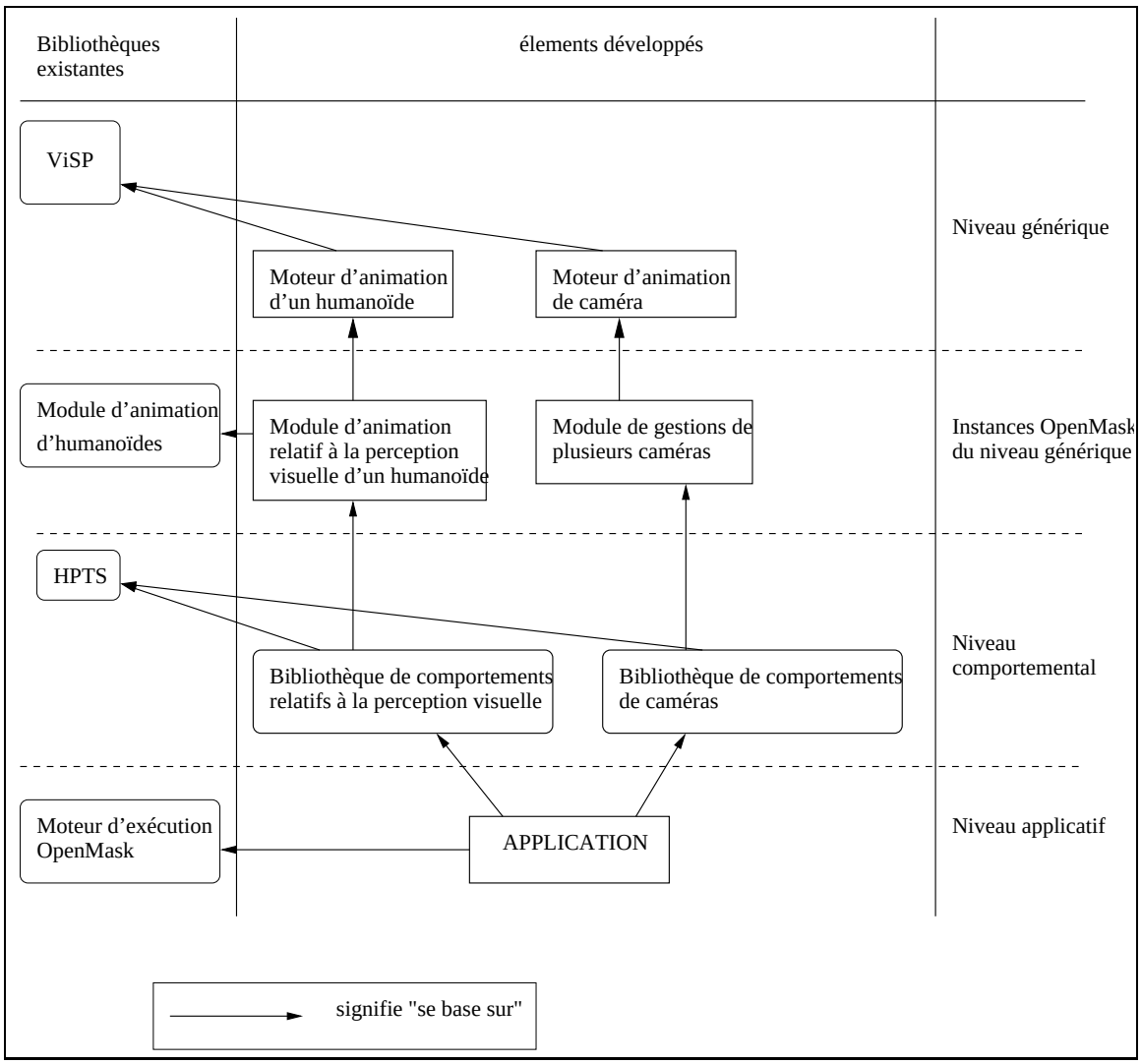


FIG. 7.6 – *Schéma structurel des différents modules utilisés par l'application*

7.2.3 Résultats

Les figures 7.7 et 7.8 montrent des extraits de cette fiction interactive.

Les premières images de la figure 7.7 (a,b) montrent Jean se dirigeant vers la caisse pour payer son paquet de gâteaux. Il rencontre alors Sara, et le dialogue s'engage (c,d,e,f,g). A la fin du dialogue, une ellipse indique la transition vers une autre scène. Il est intéressant de noter la transition entre les plans relatifs au déplacement de Jean dans l'épicerie, et les plans spécifiques au dialogue (réalisés par trois caméras différentes). L'animation sur l'ellipse est la conséquence de la tâche de positionnement de la caméra sur le panneau (effet

FIG. 7.7 – *Extrait de la démonstration Imagina. (a,b) arrivée de Jean (c,d) début du dialogue, Sara parle (e,f) Jean parle (g) Sara parle (h,i) ellipse*

de travelling vertical). Dans cet extrait, on note aussi que Sara regarde alternativement le comptoir et Jean, alors que Jean la fixe des yeux (le paramètre de timidité étant dans cette configuration élevé pour Sara et très faible pour Jean).

Dans la figure 7.8, Sara et Jean discutent autour d'un verre. Les images (a,b,c,d) sont relatives à ce dialogue. L'utilisateur décide alors de faire sonner le téléphone ; Jean se lève et s'écarte pour répondre (e,f,g). Cette interruption met en évidence le principe des comportements réactifs : le mode de discussion est, tout comme la prise en charge des comportements relatifs à la conversation téléphonique, une interruption des comportements de base (filmer une personne dans son appartement). Dans ce cas, celle-ci étant plus prioritaire, on bascule d'un niveau d'interruption (le dialogue) à un autre (conversation téléphonique).

Discussion et perspectives La mise en œuvre d'une fiction interactive a mis en avant plusieurs problèmes théoriques fondamentaux. La coexistence d'entités autonomes et d'un scénario dirigeant l'histoire est paradoxale ; dans l'implémentation réalisée on s'aperçoit que le scénario reste assez dirigiste, forçant la réalisation des différents comportements. On peut imaginer par la suite réaliser une abstraction de ce contrôle (rendant le scénario moins dirigiste) en influant seulement sur l'état interne des personnages. Par exemple, pour que Jean décide d'aller acheter un paquet de gâteaux, il faudrait plutôt faire en sorte que ce

FIG. 7.8 – *Extrait de la démonstration Imagina. (a,b,c,d) dialogue entre Jean et Sara (e,f,g) interruption du dialogue par la sonnerie d'un téléphone ; l'image (g) est en cinémascope*

soit son "envie" de gâteaux qui augmente, rendant l'action incontournable. Cette mécanique présuppose l'existence d'un système de sélection d'actions [Maes 95,Blumberg 95]. L'intérêt d'une telle approche est aussi importante au niveau du système de cinématographie virtuelle, ainsi que pour les comportements liés à la perception visuelle, dans le but de rendre le système moins dépendant des consignes du scénario, améliorant ainsi la réutilisabilité (indépendance vis-à-vis du scénario), et facilitant le processus de modélisation. Une telle approche offrirait sans doute des surprises lors de l'exécution, par apparition de comportements émergents non prévus...

Conclusion

Dans le cadre de cette partie, nous nous sommes intéressés à un système de cinématographie virtuelle ainsi qu'à un système de simulation de la perception visuelle d'un humanoïde de synthèse. Ces deux systèmes ont été traités dans une approche commune, en partie fondée sur l'utilisation de notre système de modélisation de comportements HPTS. L'utilisation de la technique d'animation référencée vision développée dans la première partie de ce document s'est révélée utile comme définition d'une base de contrôle pour les différentes entités considérées. Plusieurs problèmes théoriques ont été abordés, notamment sur la compétition de comportements. Dans ce cadre, nous avons étudié la manière de pouvoir enchaîner de manière adaptée différents comportements modélisés indépendamment, comme différents automates codant des règles de cinématographie, ou des comportements simulant la perception visuelle. Nous avons utilisé un système de gestion des ressources [Lamarche 02] pour modéliser cette compétition. Dans les deux domaines abordés, la création d'une ressource "point de vue" ou "contrôle des yeux" a rendu possible une telle compétition. Les résultats obtenus dans ces deux domaines sont assez encourageants et montrent une certaine innovation par rapport aux travaux déjà établis sur ce domaine. Plusieurs perspectives ont été dégagées de ces travaux, notamment en ce qui concerne une exploitation plus poussée des deux modalités de perception visuelle que nous avons formalisées.

Conclusion générale

Nous avons abordé dans cette thèse des problèmes de contrôle relatifs à l'animation et à la simulation en synthèse d'images. Plus spécifiquement, nous nous sommes intéressés aux liens entre la perception et l'action au travers de deux grandes classes de problème : l'animation d'une caméra virtuelle dans un environnement synthétique et la simulation d'un humanoïde de synthèse.

Synthèse des travaux effectués

Nos travaux ont porté sur deux axes de recherche principaux : la définition d'un nouveau mode de contrôle basé image en animation et son application à des systèmes comportementaux. Cette progression a été motivée par la connexité forte entre ces deux domaines. L'étude des liens entre ces deux parties a d'ailleurs montré que conceptuellement ces deux domaines réalisent une **abstraction du contrôle**. Dans chacune des deux parties de ce document relatives à ces deux domaines nous avons abordé les thématiques du contrôle d'une caméra et d'un humanoïde virtuel selon les mêmes bases théoriques.

Animation référencée vision

Nous avons défini un cadre original en animation : l'**animation référencée vision**. Cette technique permet d'obtenir des mouvements 3D d'une entité géométrique à partir d'une consigne spécifiée dans un espace 2D : l'image "perçue". Basée sur l'asservissement visuel, technique ayant trouvé son essor en robotique, notre approche s'est révélée être très intéressante dans le cadre des applications temps-réel, car elle permet une forte adaptation à des environnements dynamiques. Un des intérêts majeurs de cette technique est de permettre une simplification des paramètres de contrôle (économie de spécifications), ainsi qu'une manipulation plus aisée de ces paramètres car plus proches des spécifications issues du langage naturelle (porteuses de sens). De plus, l'utilisation du principe de redondance nous a permis de traiter des problèmes difficiles relatifs aux deux domaines étudiés : pour le contrôle de caméra, suivi de trajectoires, évitement d'obstacles et d'occultations, problème de la "photo" ; pour le contrôle d'un humanoïde ce formalisme permet d'intégrer un évitement des butées articulaires, une gestion de la locomotion (comme partie intégrante des tâches visuelles), une gestion du réalisme des animations. Les différents résultats ont prouvé que cette approche est flexible, et peut s'adapter à un grand nombre de problèmes.

Animation et simulation comportementales

Nous avons utilisé HPTS (langage de description de comportements développé à l'IRISA) pour mener à bien nos études sur les comportements. Dans un cheminement identique à la partie précédente nous nous sommes intéressés successivement aux problèmes relatifs aux "comportements" d'une caméra et d'un humanoïde virtuel : la cinématographie automatique et la simulation de la perception visuelle. Dans les deux cas l'utilisation de notre technique d'animation référencée vision nous a permis de définir des bases de contrôle nécessaires à la réalisation de ces comportements. Nous nous sommes intéressés à la notion d'autonomie, capacité d'un système à évoluer indépendamment. Cette autonomie s'est traduite dans le cadre de la cinématographie virtuelle par la définition d'un agent autonome ayant le même rôle qu'un metteur en scène. Dans le cadre de la simulation de la perception visuelle, nous avons dégagé deux modalités distinctes de perception, une endogène (motivée par la volonté) et une exogène (passive). Dans le cadre de la perception endogène, nous avons proposé un formalisme permettant d'intégrer des paramètres cognitifs. La perception exogène a été simulé par un système basé sur la notion de cartes de saillance. Une méthodologie a été établie pour modéliser les interactions/compétitions entre ces deux processus, cette dernière partie restant à compléter par la jonction avec un modèle de mémoire et un système de raisonnement.

Validation

Plusieurs expériences ont validé point par point les différents travaux présentés dans ce document. Sur la base de ces différentes techniques des modules génériques d'animation ont été conçus. Ces modules ont, entre autre, servi à la réalisation de deux applications, la première étant la réalisation d'un agent conversationnel pour France Telecom R&D, et la deuxième un exemple de fiction interactive présentée au festival Imagina 2002. Cependant, plusieurs étapes de validation sont encore nécessaires, concernant notamment la simulation de la perception visuelle exogène, où la confrontation de nos résultats expérimentaux avec des résultats issus d'observations réelles serait nécessaire.

Perspectives

Plusieurs perspectives de travail et de recherche ont émergé des différents travaux que nous avons menés. Nous les avons classé par ordre d'apparition dans le document.

Applications de la technique d'animation référencée vision

Concernant l'animation, il reste envisageable de compléter le panel de techniques proposées par de nouvelles tâches secondaires adressant de nouveaux problèmes. A titre d'exemple, il pourrait être intéressant d'ajouter une notion de dynamique dans la loi de commande, dans le but soit d'améliorer le réalisme des mouvements de la caméra (considérer une inertie), ainsi que la gestion de l'équilibre chez l'humanoïde. De plus, cette technique peut être utilisée dans d'autres contextes que ceux d'une caméra ou d'un humanoïde virtuel, par exemple la simulation éthologique (contrôle de poissons [Terzopoulos 97] ou bien encore d'un chien [Blumberg 95]).

Système de cinématographie autonome

Plusieurs perspectives émergent de notre travail sur la cinématographie virtuelle. Tout d'abord, il faut pouvoir augmenter la réutilisabilité des différents comportements "atomiques" relatifs à la cinématographie virtuelle. Ensuite, il apparaît nécessaire de mener une réflexion plus approfondie sur les conditions de transition entre les différents automates, en augmentant l'autonomie du système (en lui déportant une partie du contrôle). L'agent de cinématographie virtuelle pourrait alors choisir lui même le type de plans en temps-réel, selon des critères qui lui sont propres, éventuellement grâce à un mécanisme de sélection d'actions. Une telle perspective rend possible, à plus long terme, la modélisation de grands "styles" de réalisation, comme celui d'Hitchcock, Renoir, Orwell ou Lynch, dans la mesure où l'on est capable de traduire ces styles en règles formelles. On note à ce niveau qu'il est aussi nécessaire de considérer les problèmes relatifs à l'éclairage, ainsi qu'à la direction des acteurs (positionnement des acteurs dans la scène par exemple [Tomlinson 00]).

Perception visuelle

Dans ce domaine, les résultats prometteurs obtenus laissent à penser qu'il serait intéressant de connecter notre système de perception visuelle avec un système gérant la mémoire [Kuffner 99], ainsi qu'un système relatif au raisonnement [Funge 99]. Il serait alors possible de traiter des classes de problèmes rarement abordés avec un niveau de réalisme semblable : découverte de lieux inconnus, recherche d'objets dans un environnement, raisonnement sur l'espace, augmentant ainsi le niveau d'autonomie de l'humanoïde.

Agent conversationnel

Simuler le regard de l'avatar s'est révélé apporter une grande part de réalisme à l'application, ainsi qu'une certaine forme d'humanisation à l'interface. Il faut alors maintenant considérer le regard comme un élément essentiel intervenant dans les processus de communication non verbale [Cassell 00]. L'ajout de comportements relatifs à l'attention visuelle permettrait ainsi de rendre encore plus crédible l'avatar. Les moyens de perception de l'utilisateur pourraient eux aussi être améliorés, en rajoutant notamment des systèmes faisant de la reconnaissance gestuelle.

Fiction interactive

Il reste encore énormément de travail sur la notion de fiction interactive. Les premières mises en œuvre ont montré que beaucoup de progrès doivent être encore faits au niveau du contrôle des entités, ainsi qu'au niveau de la définition des scénarios et de la structure narrative du récit. La plupart des perspectives mentionnées jusqu'à présent peuvent aussi servir dans ce contexte, la fiction interactive étant un domaine fédérateur et nécessitant l'utilisation de nombreuses technologies.

Annexes

Annexe A

Principe du Point de vue/Point visé ou *Look-At/Look-From*

Dans cette annexe nous présentons une technique couramment employée dans le contrôle de caméra virtuelle. Cette technique sert à spécifier simplement la position et l'orientation de la caméra dans un monde 3D. Les paramètres d'entrée sont la position de la caméra ainsi que la position de l'objet que l'on souhaite observer (voir la figure A.1). La technique présentée ici permet de calculer la matrice homogène passant du repère global de la scène au repère de la caméra, cette matrice caractérisant la position et l'orientation de la caméra.

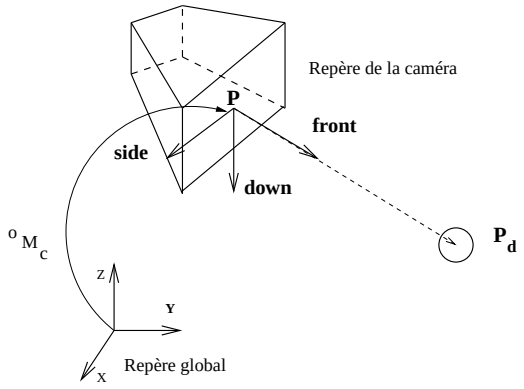


FIG. A.1 – *LookAt/LookFrom :* $\mathbf{P}$ *représente la position désirée de la caméra et* $\mathbf{P_d}$ *la position du point d'intérêt*

Soient $\mathbf{P}(x_p,y_p,z_p)$ et $\mathbf{P_d}(x_{p_d},y_{p_d},z_{p_d})$ ces deux points. Déterminer la position et l'orientation de la caméra revient à déterminer les coefficients de la matrice homogène $^{o}\mathbf{M_c}$ (eq. A.1) associée à la caméra :

$$^{o}\mathbf{M_c} = \begin{pmatrix} a_1 & b_1 & c_1 & t_1 \\ a_2 & b_2 & c_2 & t_2 \\ a_3 & b_3 & c_3 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(A.1)}$$

Les 3 coefficients de translation t_1, t_2, t_3 ne sont autre que les coordonnées x_p, y_p, z_p du point $\mathbf{P}$. Les coefficients a_1, a_2, a_3 ainsi que b_1, b_2, b_3 et c_1, c_2, c_3 sont respectivement

les coordonnées des vecteurs **side**, **down** et **front**, représentants les vecteurs de la base du repère de la caméra, conformément à la figure A.1. Trouver la matrice $^{o}\mathbf{M_c}$ revient donc à trouver les coordonnées de ces trois vecteurs.
Les coordonnées du vecteur **front** sont définies par le vecteur normalisé $\mathbf{PP_d}$.
Il existe une infinité de solutions répondant au problème. Plusieurs solutions réduisant la sous-détermination existent. Une solution consiste soit à spécifier explicitement le vecteur **down**[1]. La solution retenue dans le cadre des implémentations réalisées au cours de cette thèse est différente : on contraint le vecteur **side** à être parallèle au "sol" du monde virtuel (dans notre exemple A.1 il s'agit du plan tel que $z = 0$). Cette contrainte s'exprime sous la forme $\mathbf{side.z} = 0$ et permet de lever l'indétermination. Le vecteur **down** est alors donné par : $\mathbf{side} \wedge \mathbf{front} = \mathbf{down}$. Le choix de cette méthode se justifie par le fait qu'il est visuellement plus acceptable d'avoir une caméra "parallèle" au sol.

1. on appele aussi $\mathbf{v_{up}}$ ce vecteur (mais inversé) dans la littérature

Annexe B

Quelques matrices d'interaction

Cette annexe rappele la manière de calculer les matrices d'interaction ou jacobien image associées à un point, une ligne droite et une sphère, ces primitives étant largement utilisées dans cette thèse.

B.1 Calcul de la matrice d'interaction associée à un point

La méthode présentée ici permet de calculer la matrice d'interaction $\mathbf{L}$ associée à un point. Le lecteur pourra se référer à [Espiau 92] pour plus de détails, ainsi que pour l'extension de cette méthode à d'autres types de primitives.

Soit ${}^{\mathbf{c}}\mathbf{M}(x,y,z)$ la position du point dans le repère de la caméra et $\mathbf{m}(X,Y)$ sa projection sur le plan image. Soit $\mathbf{T} = (\mathbf{V},\Omega) = (V_x,V_y,V_z,\Omega_x,\Omega_y,\Omega_z)$ la vitesse de la caméra (ou torseur cinématique). La relation liant le mouvement du point $\dot{\mathbf{m}}$ à la vitesse de la caméra T est donnée par :

$$\dot{\mathbf{M}} = -V - \Omega \times \mathbf{M} \Leftrightarrow \left\{ \begin{array}{l} \dot{x} = -V_x - \Omega_y z + \Omega_z y \\ \dot{y} = -V_y - \Omega_z x + \Omega_x z \\ \dot{z} = -V_z - \Omega_x y + \Omega_y x \end{array} \right. \tag{B.1}$$

Dériver les équations de perspective $X = x/z$ et $Y = y/z$ par rapport au temps donne :

$$\left\{ \begin{array}{l} \frac{\partial X}{\partial t} = \frac{\partial X}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial X}{\partial z}\frac{\partial z}{\partial t} \\ \\ \frac{\partial Y}{\partial t} = \frac{\partial Y}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial Y}{\partial z}\frac{\partial z}{\partial t} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \dot{X} = \frac{\dot{x}}{z} - \frac{x}{z^2}\dot{z} \\ \\ \dot{Y} = \frac{\dot{y}}{z} - \frac{y}{z^2}\dot{z} \end{array} \right. \tag{B.2}$$

En remplaçant dans (B.2) les valeurs $(\dot{x},\dot{y},\dot{z})$ calculées dans (B.1) on obtient :

$$\left\{ \begin{array}{rcl} \dot{X} & = & -\frac{1}{z}V_x + \frac{X}{z}V_z + XY\Omega_x - (1+X^2)\Omega_y + Y\Omega_z \\ \dot{Y} & = & -\frac{1}{z}V_y + \frac{Y}{z}V_z + (1+Y^2)\Omega_x - XY\Omega_y - X\Omega_z \end{array} \right. \tag{B.3}$$

qui n'est autre que la relation $\dot{\mathbf{m}} = \mathbf{L_P}\mathbf{T_c}$ avec $\mathbf{L_P}$ défini par l'équation (2.6).

B.2 Matrice d'interaction associée à une droite

Dans cette partie on ne détaillera pas les calculs de la matrice d'interaction. Une ligne droite 3D peut être représentée comme l'intersection de deux plans :

$$h(\mathbf{x},\mathbf{p}) = \begin{cases} A_1 x + B_1 y + C_1 z = 0 \\ A_2 x + B_2 y + C_2 z + D_2 = 0 \end{cases} \tag{B.4}$$

avec

$$\begin{cases} A_1^2 + B_1^2 + C_1^2 = 1 \\ A_2^2 + B_2^2 + C_2^2 = 1 \\ A_1 A_2 + B_1 B_2 + C_1 C_2 = 0. \end{cases} \tag{B.5}$$

une représentation minimale de la ligne 2D correspondante est :

$$g(\mathbf{X},\mathbf{P}) = X\cos\theta + Y\sin\theta - \rho = 0 \tag{B.6}$$

avec

$$\begin{cases} \cos\theta &= A_1/\sqrt{A_1^2 + B_1^2} \\ \sin\theta &= B_1/\sqrt{A_1^2 + B_1^2} \\ \rho &= -C_1/\sqrt{A_1^2 + B_1^2}. \end{cases} \tag{B.7}$$

La matrice d'inetraction est alors donnée par [Espiau 92]:

$$\begin{pmatrix} \dot\rho \\ \dot\theta \end{pmatrix} = \begin{pmatrix} \lambda_\rho\cos\theta & \lambda_\rho\sin\theta & -\lambda_\rho\rho & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \\ \lambda_\theta\cos\theta & \lambda_\theta\sin\theta & -\lambda_\theta\rho & -\rho\cos\theta & -\rho\sin\theta & -1 \end{pmatrix} \tag{B.8}$$

avec $\lambda_\rho = -A\rho\cos\theta - B\rho\sin\theta - C$, $\lambda_\theta = B\cos\theta - A\sin\theta$ et $A = -A_2/D_2, B = -B_2/D_2, C = -C_2/D_2$.

B.3 Matrice d'interaction associée à une sphère

Dans cette partie on ne détaillera pas les calculs de la matrice d'interaction. La projection d'une sphère sur un plan est une ellipse. La matrice d'interaction associée à une sphère est donnée par [Espiau 92] :

$$\begin{aligned}
\mathbf{L}_{\mathbf{X_c}}^T &= \begin{bmatrix} -1/z_c & 0 & X_c/z_c + a\mu_0 + b\mu_1 & X_cY_c + \mu_1 & -(1+X_c^2) - \mu_0 & Y_c \end{bmatrix} \\
\mathbf{L}_{\mathbf{Y_c}}^T &= \begin{bmatrix} 0 & -1/z_c & Y_c/z_c + a\mu_1 + b\mu_2 & 1 + Y_c^2 + \mu_2 & -X_cY_c - \mu_1 & -X_c \end{bmatrix} \\
\mathbf{L}_{\mu_0}^T &= \begin{bmatrix} -2(a\mu_0 + b\mu_1) & 0 & 2[(1/z_c + aX_c)\mu_0 + bX_c\mu_1] & 2(Y_c\mu_0 + X_c\mu_1) & -4\mu_0 X_c & 2\mu_1 \end{bmatrix} \\
\mathbf{L}_{\mu_1}^T &= \begin{bmatrix} -(a\mu_1 + b\mu_2) & -(a\mu_0 + b\mu_1) & aY_c\mu_0 + (3/z_c - c)\mu_1 + bX_c\mu_2 & \dots \\ \dots\ 3Y_c\mu_1 + X_c\mu_2 & -(Y_c\mu_0 + 3X_c\mu_1) & \mu_0 - \mu_2 & \end{bmatrix} \\
\mathbf{L}_{\mu_2}^T &= \begin{bmatrix} 0 & -2(a\mu_1 + b\mu_2) & 2[(1/z_c + bY_c)\mu_2 + aY_c\mu_1] & 4Y_c\mu_2 & -2(Y_c\mu_1 + X_c\mu_2) & -2\mu_1 \end{bmatrix}
\end{aligned} \tag{B.9}$$

où X_c, Y_c, μ_0, μ_1, μ_2 sont respectivement le centre de l'ellipse dans l'image, la longueur des deux axes et sa surface.

Annexe C

Splines de Kochanek

Dans cette annexe, nous nous sommes particulièrement intéressés à la définition d'un type de courbe spline précis nous permettant de modéliser des trajectoires que peut suivre la caméra. En animation classique, ces trajectoires sont établies grâce à l'interpolation entre des points particuliers de l'espace appelés aussi points de contrôle. La principale difficulté revient à trouver de bonnes interpolations entre ces points particuliers. Ces interpolations doivent permettre de paramétrer les continuités de la courbe (d'ordre 0 continuité spatiale, d'ordre 1 continuité des tangentes , etc.), les conditions de terminaison ainsi que l'allure de la courbe au voisinage des points de contrôle.
Il existe de nombreux types d'interpolations polynomiales basées sur les formes polynomiales d'Hermite, de Bézier ou de type spline. Parmi ces nombreuses approches, les splines de Kochanek [Kochanek 84] ont des propriétés intéressantes : chaque élément de la courbe $C_i(t)$ entre deux points de contrôle successifs P_i et P_{i-1} est de la forme Hermite, c.a.d basé sur une courbe polynomiale cubique déterminée par 2 points terminaux et deux tangentes D_i et D_{i+1} : $C_i(0) = P_i$, $C_i'(0) = D_i$, $C_i(1) = P_{i+1}$, $C_i'(1) = D_{i+1}$.
Chaque segment de la courbe $C_i(t)$ peut être écrit sous la forme matricielle suivante :

$$C_i(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} . \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} . \begin{pmatrix} P_i \\ P_{i-1} \\ D_i \\ D_{i-1} \end{pmatrix} \qquad \text{(C.1)}$$

avec $t \in [0,1]$.

On peut alors constater que la courbe passe par les $n+1$ points de contrôle P_0,...,P_n. Pour chacun de ces points de contrôle P_i, Kochanek définit 2 vecteurs tangents, une tangente "entrante" DS_i pour le morceau de courbe C_{i-1} et une tangente "sortante" DD_i pour le morceau C_i telles que :

$$DS_i = \frac{(1-t)(1-c)(1+b)}{2}.(P_i - P_{i-1}) \; + \; \frac{(1-t)(1+c)(1-b)}{2}.(P_{i+1} - P_{i-1}) \qquad \text{(C.2)}$$

et

$$DD_i = \frac{(1-t)(1+c)(1+b)}{2}.(P_i - P_{i-1}) \; + \; \frac{(1-t)(1-c)(1-b)}{2}.(P_{i+1} - P_{i-1}) \qquad \text{(C.3)}$$

avec t, c et b des paramètres influant sur la *tension*, la *continuité* ainsi que le *biais* de la courbe. Ces paramètres utilisés de manière courante en animation permettent de modifier localement aux points de contrôle ou de manière globale l'aspect de la courbe. Il est intéressant de remarquer que lorsque le paramètre c est fixé à 1, une continuité d'ordre 1, c.a.d. continuité des tangentes, est assurée. Différents exemples de modification de ces paramètres intrinsèques sont donnés dans [Hégron 93] (voir Fig. C.1).

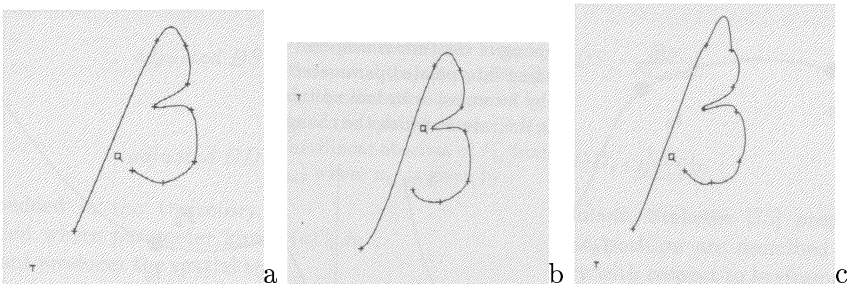


FIG. C.1 – *Différentes formes de courbe après réglages des paramètres intrinsèques : (a) t = 0, c = 0, b = 0 (b) Déplacement des points terminaux T et Q (c) t = 0, c = 1, b = 0*

Dans le cadre de l'animation de caméra, un des principaux intérêts de ces courbes est de pouvoir spécifier une trajectoire à la caméra via des points par lesquels la caméra est sensée passer. La construction de cette courbe est ainsi facilitée pour l'animateur, dans le sens où les points de contrôle qu'il place reflètent directement la courbe induite par ceux-ci, à la différence des splines de Bézier par exemple. Ces points peuvent être obtenus par des techniques non-interactives, au sens où ces points sont produits directement par une pré-étape de calculs, comme c'est le cas lorsqu'un travail de planification de trajectoires est entrepris sur la scène.

Déplacement de la caméra le long de la courbe Faire suivre la courbe par la caméra revient à déplacer le point que l'on cherche à atteindre le long de la courbe. Pour ce faire, il suffit de laisser varier le paramètre t sur l'intervalle [0,1]. A ce stade, différentes politiques peuvent être adoptées :

- *Variation linéaire du paramètre t.* L'animateur décide alors le nombre d'itérations pendant lesquelles la caméra va suivre la spline. Le paramètre t est ensuite simplement calculé de la manière suivante :
$$t = \frac{n}{N}$$
où n est le numéro d'itération courante et N le nombre total d'itérations.
- *Variations non-linéaires du paramètre t.* Au lieu de faire varier linéairement le paramètre t, on réajuste sa valeur en fonction d'une formule mathématique, de tables de correspondances (*Look-up tables*) ou bien de courbes type spline. On peut ainsi moduler indirectement la vitesse de la caméra en lui faisant chercher à atteindre des points qui ne sont pas linéairement espacés(voir Fig. C.2).
Cependant on peut remarquer qu'étant donné que les paramètres temporels et spatiaux sont fortement couplés, des discontinuités apparaissent lors du passage entre les splines composant la courbe – par exemple si deux points sont très proches, la caméra mettra plus de temps pour parcourir cette distance que si ces deux points

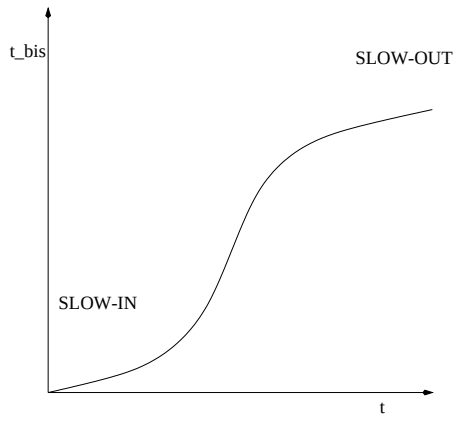


FIG. C.2 – *Exemple de courbe de réajustement de type slow-in/slow-out*

étaient plus éloignés, et ce quelque soit le réajustement du paramètre t –. Kochanek propose une solution à ce problème, en tenant compte du pas d'interpolation associé à chaque élément de courbe :

– *Résolution du problème de discontinuité des vitesse.* Kochanek propose de pondérer les vecteurs tangents en fonction du nombre de points N_i pris sur chaque élément de courbe C_i. Les vecteurs tangents deviennent :

$$adjusted\ DS_i = DS_i.\frac{2.N_i}{N_{i-1} + N_i} \tag{C.4}$$

$$adjusted\ DD_i = DD_i.\frac{2.N_{i-1}}{N_{i-1} + N_i} \tag{C.5}$$

Annexe D

Définition d'un modèle de robot : les paramètres de Denavit-Hartenberg

Dans cette annexe, une présentation des paramètres de Denavit-Hartenberg est proposée. Ces paramètres servent à décrire une chaîne articulaire de façon unique et non-équivoque. On ne considère ici que des chaînes cinématiques *simples* et *ouvertes*. Le lecteur pourra se référer à l'article [Denavit 55] pour plus de détails sur cette méthode.

Dans cette méthodologie, on considère que la chaîne articulaire est composée d'éléments rigides (les liens), connectés entre eux deux par deux au travers de liaisons. Considérons deux liens, seules les liaisons ayant des points de contact compris dans une surface sont considérées, par opposition aux liaisons ponctuelles (exemple : une boule glissant sur une table ; la boule et la table sont des éléments rigides mais ils ne sont en contact qu'en un point). Seuls deux types de liaisons sont modélisés au travers des paramètres de Denavit-Hartenberg : les liaisons *rotoïdes* , et les liaisons dites *prismatiques*.

- les liaisons *rotoïdes* (voir Fig D.1(a)), où la surface de contact est une surface de révolution (définie par une courbe et un axe de rotation dans le plan de la courbe). Ce type de liaison permet une rotation d'un lien autour de l'axe de la liaison.
- les liaisons *prismatiques* (voir Fig D.1(b)), où la surface de contact est un cylindre généralisé (défini par une surface fermée et une ligne ; la liaison est définie comme le volume balayé par la surface le long de la trajectoire que constitue la ligne, sans changement d'orientation). Ce type de liaison permet une translation d'un lien par rapport à un autre le long de la direction de la ligne génératrice.

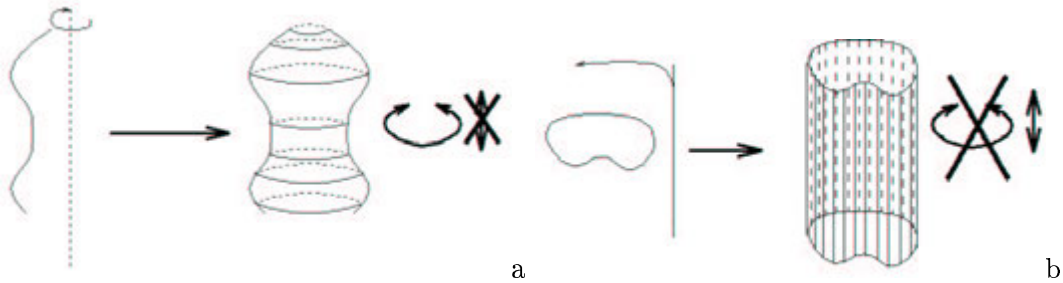


FIG. D.1 – *Les deux types de liaisons : (a) rotoïde (b) prismatique*

Pour décrire de manière exhaustive une chaîne articulaire, il faut être capable de pouvoir décrire ses liens ainsi que ses liaisons. Nous allons maintenant voir comment décrire ces éléments.

Description d'un lien Considérons un lien avec deux liaisons rotoïdes des deux côtés. Les deux liaisons possédent un axe de révolution (voir Fig D.2(a)). Soit **a** la distance entre les deux axes de rotation. Cette longueur représente la longueur du lien, et est appelée *normale commune*. L'angle formé par les deux axes de rotation α est appelé angle de torsion. Dans le cas où une liaison (ou les deux) est de type prismatique, on raisonne par

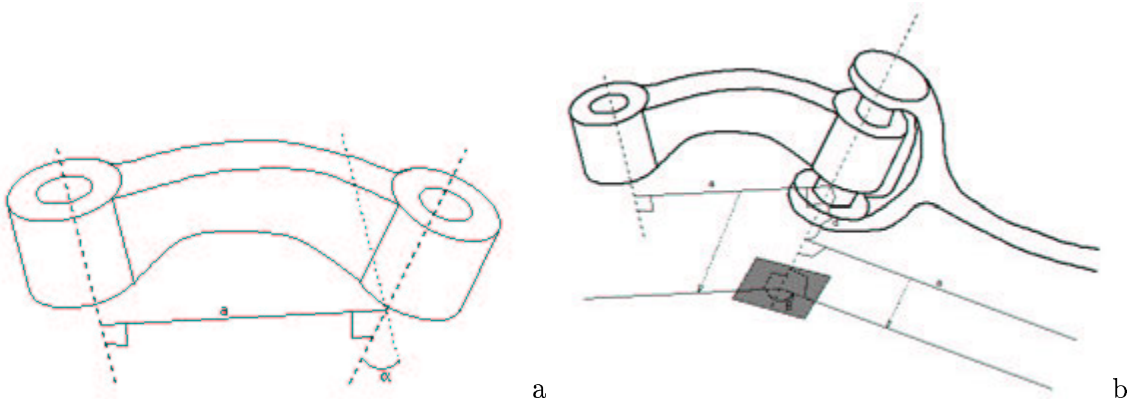


FIG. D.2 – *((a) Description d'un lien (b) description d'une liaison*

rapport à deux lignes parallèles aux lignes génératrices des deux liaisons. Dans ce cas, $a = 0$ et on mesure seulement α.

Description d'une liaison Nous avons défini précédemment la notion de normale commune. La distance entre deux normales communes de deux liens adjacents est appellée **d**. Cette distance est la variable associée aux liaisons prismatiques. La variable associée aux liaisons rotoïdes θ représente l'angle formé par la projection des deux normales communes sur un plan normal à l'axe de la liaison (voir Fig D.2(b)).

La description de la chaîne cinématique s'effectue en décrivant pour chaque liaison un quintuplet regroupant le type de liaison, a, d, θ et α. Deux types de notations existent, une notation "simple" et une notation "étendue", décrivant deux ordres distincts de numérotation des axes (considérant soit la base de la chaîne, soit son effecteur). Denavit et Hartenberg ont développé un cadre mathématique utilisant les systèmes de transformations homogènes pour calculer la position de l'effecteur de la chaîne dans le repère de base de celle-ci. A partir des descriptions établies auparavant, les matrices homogènes associées à chaque lien sont calculées dans le repère du lien précédent (ou suivant selon l'ordre). Appliquer ces transformations en cascade permet d'obtenir la position de n'importe quelle liaison dans le repère de base de la chaîne.

Nous avons établi au cours de cette thèse plusieurs modèles de chaînes articulaires correspondant à différents ensembles de parties du corps. A titre d'exemple nous donnons la

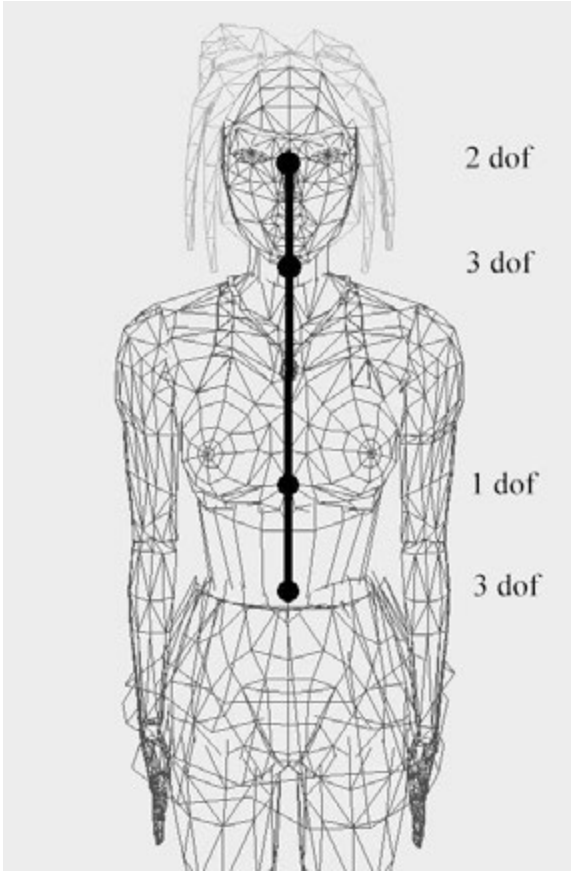


FIG. D.3 – *Chaines articulaires pour un robot à neuf degrés de liberté*

représentation en paramètres de Denavit-Hartenberg d'un modèle de chaîne comptant neuf degrés de liberté (trois pour le bassin, un pour le buste, trois pour le cou et deux pour les yeux, voir figure D.3) :

	Paramètres de Denavit-Hartenberg				
	type de liaison	θ	d	a	α
bassin1	0	0	0	0	$-\pi/2$
bassin2	0	$-\pi/2$	0	0	$\pi/2$
bassin3	0	0	0	longueur(bassin,buste)	0
buste1	0	0	0	longueur(bassin,cou)	0
cou1	0	0	0	0	$\pi/2$
cou2	0	$\pi/2$	0	0	$\pi/2$
cou3	0	$\pi/2$	longueur(cou,yeux)	0	$\pi/2$
oeil1	0	π	0	0	$-\pi/2$
oeil2	0	$\pi/2$	0	0	$\pi/2$

FIG. D.4 – *Paramètres de Denavit Hartenberg pour une chaîne à neuf degrés de liberté. Le type de liaison est rotoïde (0) ou prismatique (1)*

Table des figures

1.1 Exemple de tâche visuelle : “voir le tableau de Monet centré dans l'image perçue” (a) état initial (b) état final 17
1.2 Différents cadres applicatifs et leurs besoins relativement à la notion d'autonomie et à la notion d'interactivité 18

2.1 Principe du keyframing 25
2.2 Boucle d'asservissement visuel 28
2.3 Tâche de positionnement relative à un rectangle (a) vue de la caméra et vue du dessus correspondante, la caméra est en vert (b) vitesse de la caméra en translation, (c) en rotation (d) erreurs dans l'image relative aux positions des quatre points 32
2.4 Erreurs dans trois expériences de tracking (a) sans correction (b) avec un scalaire $\alpha = 0.9$ (c) avec une correction totale de l'erreur 33
2.5 Poursuite d'un objet dans une scène, vue de la caméra et vue du dessus . . 34
2.6 Courbes associées à la poursuite d'un objet : (a) distance à l'objet (15 est la valeur souhaitée) (b) erreurs entre la position désirée dans l'image et la position réelle 36
2.7 Courbe paramétrique : variation du paramètre t 37
2.8 Résultats associés au suivi d'une trajectoire (a) images de l'exemple rendues sous Maya (b) erreur associée à la tâche principale (c) variations de la distance au point que l'on cherche à rattraper (d) visualisation de la trajectoire de la caméra décrivant une courbe de Kochanek 39
2.9 Différentes situations d'occultation : (a) occultation par un objet en mouvement (b) occultation provoquée par un déplacement de l'objet (c) occultation provoquée par un déplacement de la caméra 42
2.10 Différents types de volumes englobants : (a) associé à un objet (b) associé à la caméra et à l'objet d'intérêt 43
2.11 Volume augmenté à cause d'un déplacement de l'objet 44
2.12 Différents types d'intesections avec le volumes englobants : (a) évitement d'occultations (b) évitement d'obstacles 44
2.13 Volume augmenté à cause d'un déplacement de la caméra 46
2.14 Déplacement de la caméra lors d'un changement de direction de l'objet . . . 47
2.15 Impossibilité de déplacer la caméra latéralement : il faut diminuer la distance entre la caméra et l'objet 48
2.16 Evitement d'occultations en restant en ligne avec l'objet d'intérêt 49

2.17 (a) extrait de l'animation associée au problème du pilier (b) translations de la caméra (c) rotations de la caméra (d) distance à l'objet d'intérêt (e) trajectoire 3D de la caméra 51
2.18 Environnement de type musée: trajectoire de la caméra pour différentes stratégies 52
2.19 Progression dans un couloir: vue de dessus et vue de la caméra 53
2.20 Traversée d'un musée. Le processus d'évitement des occultations et des obstacles n'est pas considéré. La trajectoire résultante comprend de nombreuses pertes de la cible et des collisions avec l'environnement 54
2.21 Traversée d'un musée: vue de la caméra et vue du dessus. Dans cette animation le processus d'évitement des obstacles et des occultations est considéré. Les volumes englobants apparaissent en jaune 55
2.22 Contrôle des conditions d'éclairement. (a) source lumineuse statique, caméra mobile (b) caméra statique, source lumineuse mobile 56
2.23 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 58
2.24 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 . . 59
2.25 Illumination de la Venus de Milo: trajectoires de la source lumineuse et de la caméra 59

3.1 Pyramide de l'animation d'humanoïdes de synthèse 61
3.2 Différents types de chaîne articulaires impliquées dans le processus d'attention visuelle (a) neuf degrés de liberté (b) cinq degrés de liberté 69
3.3 Remise à jour de la chaîne articulaire 70
3.4 Tâche de positionnement relative à une sphère (a) vue d'ensemble et vue du personnage (b) configuration articulaire (c) vitesses calculées (d) erreurs dans l'image relative à la sphère 72
3.5 Suivre des yeux un objet qui bouge: (a) vue globale de la scène (b) vision subjective de l'humanoïde 73
3.6 Evolution de l'erreur (axe X: itérations, axe Y: valeurs de l'erreur le long des deux axes) 73
3.7 Asservissement sur quatre points: (a) vue globale de la scène (b) vision subjective de l'humanoïde 74
3.8 Tâche secondaire d'évitement de butées 75
3.9 Evitement des butées à l'aide de la tâche secondaire: différentes valeurs du paramètre β (a) $\beta = 0.1$, (b) $\beta = 1$, (c) $\beta = 10$ (d) la méthode calculant un gain par axe 78
3.10 Evitement des butées articulaires (a) sans butées (b) avec butées; évolution des axes permettant d'assurer la tâche (en vert les yeux, en rouge et bleu le cou et le bassin) (c) sans butées (d) avec butées 79
3.11 Exemple de fonction α 81
3.12 Morphing d'un point vers quatre points 82

3.13 Transition d'un point à une sphère 83
3.14 Choisir une bonne interpolation spatiale (a) mauvaise (b) bonne 85
3.15 Transition entre deux tâches de même nature (asservissement sur une sphère) avec une interpolation spatiale correcte 85
3.16 Erreur moyenne sur l'ensemble de l'animation 86
3.17 Transition d'une primitive de type "point" à une primitive de type "sphère" (a) vue d'ensemble (b) image perçue par l'humanoïde 86
3.18 Regarder de près un objet qui bouge: (a,b,c) l'humanoïde se rapproche de la balle (d,e,f) pas de coté pour maintenir la distance souhaitée 89
3.19 Exemple de gestion de la locomotion avec une chaîne articulaire comportant douze degrés de liberté (dont trois en translation) 90
3.20 Coopération cou/yeux pour le suivi visuel d'une cible. Comparaison entre l'étude de Robinson [Robinson 64] (a) et les résultats obtenus avec notre système d'animation (b) 92
3.21 Divergence au cours du temps, les deux captures d'écran sont prises à l'itération 3000 de la simulation (a) sans contrainte sur le réalisme (b) avec . . . 92

4.1 Boucle Perception–Décision–Action pour un acteur synthétique 102
4.2 Relation automate à états finis et automates parallèles hiérarchisés (a) simple (b) hiérarchisé (extrait de [Koga 98]) 105
4.3 *PaT-net* décrivant le comportement d'un jeu de cache-cache 105
4.4 Automate modélisant un comportement simple : observer tous les tableaux d'une pièce 109

5.1 *Cinemasaurus* par John Funge (extraits de [Funge 99]) 113
5.2 Boucle perception-décision-action pour un agent de cinématographie virtuelle 114
5.3 Exemple de travelling sur un personnage 116
5.4 Notions de distance en cinématographie : (a) différents plans de coupe. Les lignes de découpage sont en rouge (b) un exemple de primitives utilisées pour réaliser un gros plan 116
5.5 Enchaînement entre différents plans. Les points rouges montrent les objectifs visuels dans l'image (a) plan d'ensemble (b) plein cadre (c) gros plan (d) un autre plan d'ensemble avec contre-plongée 117
5.6 Plan d'ensemble avec contre-plongée. On donne à la caméra un mouvement vertical avec maintient de la contrainte plan d'ensemble 117
5.7 Caméras et modèle décisionnel 118
5.8 Plans possibles pour filmer un dialogue entre deux acteurs, sur le principe du champ contre-champ interne ou externe 119
5.9 (a) Automate simple de gestion d'un dialogue entre deux personnages (b) Primitives d'asservissement pour la réalisation d'un champ contre-champ (c,d) exemples de plans associés aux deux états *Focal perso* ; les objectis visuels sont en jaune dans l'image 120
5.10 Extraits de storyboards (a) *Garde à vue* de Claude Miller (b) *Himalaya* de Claude Valli 121

5.11 Compétition de deux comportements de caméra (a) automate de dialogue entre deux personnages (b) automate filmant un acteur évoluant dans une pièce 124

6.1 Schéma d'un œil humain 126
6.2 Les deux systèmes de vision 126
6.3 Contrôle de l'axe. La tâche d'observation se fait selon la théière. La ligne droite doit être maintenue verticale dans l'image (a) mauvaise configuration (b) bonne configuration 131
6.4 Battement des paupières (a) l'objet graphique paupière (délimité par un trait noir) possède un centre de rotation confondu avec celui de l'oeil (b) tête de sara avec les paupières à moitié fermées, lui donnant un air maussade 132
6.5 Tâche de surveillance pendant un comportement de marche (a) automate associé au comportement visuel (b) zones où l'on effectue le tirage aléatoire des cibles 133
6.6 Exemples de carte de saillance sur des images réelles (extraites de [Itti 98]). Les cercles jaunes montrent les zones saillantes et les traits rouges l'ordre de détermination de celles-ci 135
6.7 Utilisation du principe de saillance 136
6.8 Interaction entre les processus attentionnels endogène et exogène 139
6.9 Extrait de l'animation où nonos avance dans un environnement inconnu . . 140
6.10 Différents résultats de l'utilisation d'un banc de filtres de gabor sur différentes images synthétiques. Pour chacune des images, quatre niveaux d'échelle sont utilisés avec huit orientations de filtres différentes 142
6.11 Chaîne de traitement aboutissant à la création de la carte de saillance . . . 143
6.12 Calcul d'une suite de quatre zones de fixation, les carrés blanc montre les zones possédant un maximum d'énergie, et les carrés rouges montrent les zones découpées de l'image 143
6.13 Automate servant à la synchronisation du calcul des zones saillantes et de leurs observations. L'état **Pause** sert à relâcher la ressource des yeux si un comportement plus prioritaire intervient. 145
6.14 Extraits de l'animation relative à l'utilisation d'une carte de saillance dans un environnement urbain 146
6.15 Différentes cartes de saillance établies lors de la promenade de Sara. La première colonne montre la vue subjective associée au personnage. La deuxième montre les cartes de profondeur associées, et la troisième le résultat du calcul des zones saillantes, avec l'ordre d'obtention des fixations 147

7.1 Principe de fonctionnement 150
7.2 Schéma représentant les différents éléments utilisés par le système d'animation 151
7.3 Suivi du visage et animation d'un agent communicant pour France Telecom R&D. (a) Vue d'ensemble du système (b) Captures d'écran présentant l'application. L'incrustation vidéo représente l'image perçue par la webcam, et la croix en rouge représente le résultat du suivi 153
7.4 Graphe d'états d'une fiction non linéaire 154

7.5 Présentation de la démonstration de fiction interactive (a) générique de début (b) Jean (c) Sara 155
7.6 Schéma structurel des différents modules utilisés par l'application 157
7.7 Extrait de la démonstration Imagina. (a,b) arrivée de Jean (c,d) début du dialogue, Sara parle (e,f) Jean parle (g) Sara parle (h,i) ellipse 158
7.8 Extrait de la démonstration Imagina. (a,b,c,d) dialogue entre Jean et Sara (e,f,g) interruption du dialogue par la sonnerie d'un téléphone ; l'image (g) est en cinémascope 159

A.1 LookAt/LookFrom : $\mathbf{P}$ représente la position désirée de la caméra et $\mathbf{P_d}$ la position du point d'intérêt 169

C.1 Différentes formes de courbe après réglages des paramètres intrinsèques : (a) t = 0, c = 0, b = 0 (b) Déplacement des points terminaux T et Q (c) t = 0, c = 1, b = 0 174
C.2 Exemple de courbe de réajustement de type slow-in/slow-out 175

D.1 Les deux types de liaisons : (a) rotoïde (b) prismatique 177
D.2 ((a) Description d'un lien (b) description d'une liaison 178
D.3 Chaines articulaires pour un robot à neuf degrés de liberté 179
D.4 Paramètres de Denavit Hartenberg pour une chaîne à neuf degrés de liberté. Le type de liaison est rotoïde (0) ou prismatique (1) 179

Bibliographie

[Allen 93] P.K. Allen, A. Timcenko, B. Yoshimi, P. Michelman. – Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE trans. on Robotics and Automation*, 9(2):152–165, avril 1993.

[Amaya 96] K. Amaya, A. Bruderlin, T. Calvert. – Emotion from motion. – *Proc. of Graphics Interface '96*, pp. 222–229, mai 1996.

[Arijon 84] D. Arijon. – *Grammaire du langage filmé.* – Editions Dujarric, Paris, 1984.

[Arnaldi 89] B. Arnaldi, G. Dumont, G. Hégron, N. Magnenat-Thalmann, D. Thalmann. – Animation control with dynamics. – *Proc of IEEE Computer Animation '89*, pp. 113–123, 1989.

[Arnaldi 91] B. Arnaldi, G. Dumont, G. Hégron. – Animation of physical systems from geometric, kinematic and dynamic models. – *Proc. of Working Conf. on Modeling in Computer Graphics*, pp. 37–53, Tokyo, Japan, avril 1991.

[Badler 90] N. Badler, L. Webber, J. Esakov, J. Kalita. – *Animation from Instructions.* – Rapport de recherche, University of Pennsylvania, mars 1990.

[Badler 95] N. Badler, B. Webber, W. Becket, C. Geib, M. Moore, C. Pelachaud, B. Reich, M. Stone. – Planning and parallel transition networks: Animation's new frontiers. – *Proc. of the Conf. on Computer Graphics and Applications (Pacific-Graphics-95)*, pp. 101–117, 1995.

[Badler 97a] N. Badler. – Real-time virtual humans. – *Proc. of the Conf. on Computer Graphics and Applications (Pacific-Graphics-97)*, pp. 4–14, Los Alamitos, California, octobre 1997.

[Badler 97b] N. Badler, B. Reich, B. Webber. – Towards personalities for animated agents with reactive and planning behaviors. *Lecture Notes in Artificial Intelligence, Creating Personalities for synthetic actors*, (1195):43–57, 1997.

[Badler 99a] N. Badler, D. Chi, S. Chopra. – Virtual human animation based on movement observation and cognitive behavior models. – *Proc. of IEEE Computer Animation´99*, pp. 128–137, Lausanne, Suisse, mai 1999.

[Badler 99b] N. Badler, M. Palmer, R. Bindiganavale. – Animation control for

real-time virtual humans. *Communications of the ACM*, 42(8):64–73, août 1999.

[Baerlocher 98] P. Baerlocher, R. Boulic. – Task-priority formulations for the kinematic control of highly redundant articulated structures. – *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'98*, Victoria, Canada, octobre 1998.

[Baerlocher 00] P. Baerlocher, R. Boulic. – Kinematic control of the mass properties of redundant articulated bodies. – *Proc. of IEEE Int. Conf. on Robotic and Automation*, pp. 2557 – 2562, San Francisco, avril 2000.

[Beer 91] R. Beer. – *Intelligence as Adaptive Behaviour; An Experiment in Computational Neuroethology.* – Editions Samsara, Londres, 1991.

[BenIsrael 74] A. Ben-Israel, T. Greville. – *Generalized Inverses: Theory and Applications.* – Wiley-Interscience, 1974.

[Bindigavanale 98] R. Bindigavanale, N. Badler. – Motion abstraction and mapping with spatial constraints. – *Proc. of Int. Workshop on Modelling and Motion Capture Techniques for Virtual Environments (CAPTECH'98)*, pp. 70–82, Genève, Suisse, novembre 1998.

[Blinn 88] J. Blinn. – Where am I? what am I looking at? *IEEE Computer Graphics and Applications*, 8(4):76–81, Juillet 1988.

[Blumberg 95] B. Blumberg, T. Galyean. – Multi-level direction of autonomous creatures for real-time virtual environments. – *Proc. of SIGGRAPH 95, in Computer Graphics Proceedings*, pp. 47–54, Los Angeles, Californie, août 1995.

[Blumberg 97] B. Blumberg. – Go with the flow: Synthetic vision for autonomous animated creatures. – *Proc. of the 1st Int. Conf. on Autonomous Agents*, pp. 538–539, New York, février 1997.

[Boulic 96] R. Boulic, R. Mas, D. Thalmann. – A robust approach for the control of the center of mass with inverse kinetics. *Computers and Graphics*, 20(5):693–701, septembre–octobre 1996.

[Breton 00] G. Breton, D. Pelé, C. Bouville. – Faceengine : a 3d facial animation engine for real time applications. – *ACM Web3D Symposium*, Paderborn, Germany, février 2000.

[Brogan 98] D. Brogan, R. Metoyer, J. Hodgins. – Dynamically simulated characters in virtual environments. *IEEE Computer Graphics and Applications*, 18(5):58–69, septembre–octobre 1998.

[Brooks 91] R. Brooks. – Intelligence without representation. *Artificial Intelligence*, 47:139–159, mai 1991.

[Cassell 00] J. Cassell, J. Sullivan, S. Prevost, E. Chirchill (édité par). – *Embodied Conversational Agents.* – The MIT Press, Cambridge, Massachusetts, 2000.

[Cassell 01] J. Cassell, H. Vilhjálmsson, T. Bickmore. – BEAT: The behavior expression animation toolkit. – *Proc. of SIGGRAPH 01, in Computer Graphics Proceedings*, pp. 477–486, 2001.

[Chang 95] T. Chang, R. Dubey. – A weighted least-norm solution based scheme

for avoiding joints limits for redundant manipulators. *IEEE trans. on Robotics and Automation*, 11(2):286–292, avril 1995.

[Chauffaut 97] A. Chauffaut, S. Donikian. – Gasp: a general animation and simulation platform. – *Proc. of the 14th IMEKO World Congress (ISMCR'97)*, Tampere, Finland, juin 1997.

[Chaumette 90] F. Chaumette. – *La relation vision-commande: théorie et application à des tâches robotiques.* – Thèse de Doctorat, Thèse de l'université de Rennes I, IRISA, July 1990.

[Chaumette 00] F. Chaumette, E. Marchand. – A new redundancy-based iterative scheme for avoiding joint limits. application to visual servoing. – *IEEE Int. Conf. on Robots and Automation (ICRA'2000)*, San Francisco, Californie, 2000.

[Choi 00] K. Choi, H. Ko. – Online motion retargetting. *The Journal of Visualization and Computer Animation*, 11(5):223–235, avril 2000.

[ChopraKhullar 99a] S. Chopra-Khullar. – *Where to look? Automating Certain Visual Attending Behaviors of Human Characterisitics.* – Phd thesis, University of Pennsylvania, mars 1999.

[ChopraKhullar 99b] S. Chopra-Khullar, N. Badler. – Where to look? automating attending behaviors of virtual human characters. – *Proc. of the 3rd Int. Conf. on Autonomous Agents (AA-99)*, pp. 16–23, New York, mai 1999.

[Christianson 96] D. Christianson, S. Anderson, L. He, D. Salesin, D. Weld, M. Cohen. – Declarative camera control for automatic cinematography (video). – *Proc. of the 13th Nat. Conf. on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pp. 148–155, Menlo Park, août 1996.

[Cozot 96] R. Cozot. – *Environnement de simulation des systèmes physiques.* – Thèse de doctorat, Université de Rennes I, 1996.

[Crétual 98] A. Crétual, F. Chaumette. – Image-based visual servoing by integration of dynamic measurements. – *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA-98)*, pp. 1994–2001, Piscataway, mai 1998.

[Denavit 55] J. Denavit, R. S. Hartenberg. – A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22:215–221, 1955.

[Didday 75] R. Didday, M. Arbib. – Eye movements and visual perception: A "two visual system" model. *International Journal of Man-Machine Studies*, 7(4):547–569, 1975.

[Donikian 98] S. Donikian, A. Chauffaut, T. Duval, R. Kulpa. – Gasp: from modular programming to distributed execution. – *Proc. of Computer Animation '98*, Philadelphie, USA, juin 1998. IEEE Computer Society Press.

[Donikian 01] S. Donikian. – HPTS: a behaviour modelling language for autonomous agents. – *Proc. of the 5th Int. Conf. on Autonomous Agents*, pp. 401–408, Montréal, Canada, mai 2001.

[Drucker 94a] S. Drucker. – *Intelligent Camera Control for Graphical Environnemts.* – PhD. Thesis, S.M. MIT, Juin 1994.

[Drucker 94b] S. Drucker, D. Zeltzer. – Intelligent camera control in a virtual environnement. – *Proc. of Graphics Interface '94*, pp. 190–199, Banff, Alberta, Canada, mai 1994.

[Espiau 92] B. Espiau, F. Chaumette, P. Rives. – A new approach to visual servoing in robotics. *IEEE trans. on Robotics and Automation*, 8(3):313–326, juin 1992.

[Faloutsos 01] P. Faloutsos, M. van de Panne, D. Terzopoulos. – The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers and Graphics*, 25(6):933–953, décembre 2001.

[Feraud 01] R. Feraud, O. Bernier, J.-E. Viallet, M. Collobert. – A fast and accurate face detector based on neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:42–53, 2001.

[Firby 91] R. Firby, M. Swain. – Symbolic action, behavioral control and active vision. – *Proc. of the 13th Annual Cognitive Science Society*, pp. 698–702, 1991.

[Flandin 01] G. Flandin. – *Modélisation probabiliste et exploration visuelle autonome pour la reconstruction de scènes inconnues.* – Thèse de Doctorat, Thèse de l'université de Rennes I, IRISA, novembre 2001.

[France 99] L. France, A. Girault, J. D. Gascuel, B. Espiau. – Sensor modeling for a walking robot simulation. – *Proc. of Computer Animation and Simulation '99*, pp. 189–198, Italie, septembre 1999.

[Funge 98] J. Funge. – *Making them behave, cognitive models for computer animation.* – PhD. Thesis, University of Toronto, 1998.

[Funge 99] J. Funge, X. Tu, D. Terzopoulos. – Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. – *Proc. of SIGGRAPH 99, Computer Graphics Proceedings*, pp. 29–38, Los Angeles, 1999.

[Gibet 94] S. Gibet, P. Marteau. – A self-organised model for the control, planning and learning of nonlinear multivariable systems using a sensori-feedback. *Journal of Applied Intelligence*, 4:337–349, 1994.

[Gibet 98] S. Gibet, J. Richardson, T. Lebourque, A. Braffort. – Corpus of 3D natural movements and sign language primitives of movement. – *Proc. of the Int. Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction (GESTURE-97)*, vol. 1371 of *LNAI*, pp. 111–122, Berlin, septembre 1998.

[Gibet 01] S. Gibet, T. Lebourque, P. Marteau. – High-level specification and animation of communicative gestures. *Journal of Languages and Computing*, 12:657–687, 2001.

[Gibson 79] J. Gibson. – *The Ecological Approach to Visual Perception.* – Houghton Mifflin, Boston, Massachusetts, 1979.

[Gleicher 92] M. Gleicher, A. Witkin. – Through-the-lens camera control. – *Proc. of SIGGRAPH 92, in Computer Graphics Proceedings*, pp. 331–340, Juillet 1992.

[Gregory 93] R. Gregory. – *Le cerveau, un inconnu.* – Oxford Press, Londres, 1993.

[Hacisalihzade 92] S. S. Hacisalihzade, L. W. Stark, J. S. Allen. – Visual perception and sequences of eye movement fixations: A stochastic modeling approach. *IEEE Trans. on Systems, Man, and Cybernetics*, 22(3):474–481, mai–juin 1992.

[Hager 95] G. Hager, W. Chang, A.S. Morse. – Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination. *IEEE Control Systems Magazine*, 15(1):30–39, February 1995.

[Hager 97] G. Hager. – A modular system for robust positioning using feedback from stereo vision. *IEEE trans. on Robotics and Automation*, 13(4):582–595, août 1997.

[Hashimoto 93] K. Hashimoto. – *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, Singapour, 1993.

[He 96] L.-W. He, M.F. Cohen, D.H. Salesin. – The virtual cinematographer: a paradigm for automatic real-time camera control and directing. – *Proc. of SIGGRAPH 96, in Computer Graphics Proceedings*, pp. 217–224, Nouvelle Orleans, Août 1996.

[Hégron 93] G. Hégron, B. Arnaldi. – *Computer Animation and deformation control.* – Novembre 1993.

[Hong 97] L. Hong, S. Muraki, A. Kaufman, D. Bartz, T.He. – Virtual voyage: Interactive navigation in the human colon. – *Proc. of SIGGRAPH 97, in Computer Graphics Proceedings*, Annual Conference Series, pp. 27–34, 1997.

[Hutchinson 96] S. Hutchinson, G. Hager, P. Corke. – A tutorial on visual servo control. *IEEE trans. on Robotics and Automation*, 12(5):651–670, octobre 1996.

[Itti 98] L. Itti, J. Braun, D.. Lee, C. Koch. – A model of early visual processing. – *Advances in Neural Information Processing Systems*, vol. 10. The MIT Press, 1998.

[Jardillier 98] F. Jardillier, E. Languénou. – Screen-space constraints for camera movements: the virtual cameraman. – *Proc. of EUROGRAPHICS '98*, pp. 176–186, 1998.

[Kallmann 98] M. Kallmann, D. Thalmann. – Modeling objects for interaction tasks. – *Proc. of Eurographics Workshop on Computer Animation and Simulation'98*, Lisbonne, septembre 1998.

[Kallmann 99] M. Kallmann, D. thalmann. – A behaviourial interface to simulate agent-object interactions in real time. – *Proc. of Computer Animation ´99*, pp. 138–146, Genève, Suisse, 1999.

[Karp 93] P. Karp, S. Feiner. – Automated presentation planning of animation using task decomposition with heuristic reasoning. – *Proc. of Graphics Interface '93*, pp. 118–127, Toronto, Ontario, Canada, mai 1993.

[Khatib 86] O. Khatib. – Real-time avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–99, Janvier 1986.

[Kochanek 84] D. Kochanek, H. Bartels. – Interpolating splines with tension, continuity and bias control. *Proc. of SIGGRAPH 84, in Computer Graphics Proceedings*, 18(3):33–41, Juillet 1984.

[Koga 98] Y. Koga, G. Annesley, C. Becker, M. Svihura, D. Zhu. – On intelligent digital actors. – Imagina'98 White papers, 1998.

[Kosecká 94] J. Kosecká, L. Bogoni. – Application of discrete events systems for modeling and controlling robotic agents. – *Proc. of the Int. Conf. on Robotics and Automation*, vol. 3, pp. 2557–2562, Los Alamitos, CA, USA, mai 1994.

[Kuffner 99] J. Kuffner, J.C Latombe. – fast synthetic vision, memory, and learning models for virtual humans. – *Proc. of Computer Animation '99*, pp. 118–127, Genève, Suisse, mai 1999.

[Kuo 93] A. Kuo, F. Zajac. – Human standing posture : Multi-joint movement strategies based on biomechanical constraints. *Journal of Progress in Brain Research*, 97, 1993.

[Lamarche 01] F. Lamarche, S. Donikian. – The orchestration of behaviours using resources and priority level. – *Proc. of the Eurographics Workshop on Computer Animation and Simulation (EGCAS-01)*, pp. 171–182, Vienne, Autriche, septembre 2001.

[Lamarche 02] F. Lamarche, S. Donikian. – Automatic orchestration of behaviours through the management of resources and priority levels. – *Proc. of the Int. Conf. on Autonomous Agents and Multi Agent Systems (AAMAS'02)*, Bologne, Italie, juillet 2002.

[Latombe 91] J.C. Latombe. – *Robot Motion Planning.* – Kluwer Academic Publishers, 1991.

[Lebourque 99] T. Lebourque, S. Gibet. – High-level specification and control of communication gestures : the GESSYCA system. – *Proc. of Computer Animation (CA'99)*, pp. 128–139, Genève, mai 1999.

[Lee 99] J. Lee, S. Shin. – A hierarchical approach to interactive motion editing for human-like figures. – Alyn Rockwood (édité par), *Proc. of SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 39–48, New-york, août 1999.

[Maes 95] P. Maes, T. Darrell, B. Blumberg, A. Pentland. – The alive system: Full-body interaction with autonomous agents. – *Computer Animation'95*, pp. 11–18, Geneva, Switzerland, avril 1995. IEEE.

[Malis 98] E. Malis, F. Chaumette, S. Boudet. – Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing. – *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA-98)*, pp. 1352–1359, Piscataway, mai 1998.

[Mallot 97] H. A. Mallot. – Behavior-oriented approaches to cognition : theoretical perspective. *Theory in biosciences*, 116:196–220, 1997.

[Marcelja 80] S. Marcelja. – Mathematical description of the responses of simple cortical cells. *Journal of Optical Society of America*, 70:1297–1300, 1980.

[Marchand 96] E. Marchand, F. Chaumette, A. Rizzo. – Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. *Proc of IEEE/RSJ Int. Conf on Intelligent Robots and Systems, IROS'96*, 14(3):1083–1090, novembre 1996.

[Marchand 98] E. Marchand, G.D. Hager. – Dynamic sensor planning in visual servoing. – *IEEE Int. Conf. on Robotics and Automation ICRA '98*, vol. 3, pp. 1988–1993, Louvain, Belgique, Mai 1998.

[Marchand 99] E. Marchand. – Visp: A software environnement for eye-in-hand visual servoing. – *IEEE Int. Conf. on Robotics and Automation ICRA '99*, Détroit, Michigan, Mai 1999.

[Mateas 97] M. Mateas. – *An Oz-Centric Review of Interactive Drama and Believable Agents.* – Technical report, School of Computer Science, Carnegie Mellon University, 1997.

[McCallum 97] A. McCallum. – *Reinforcement Learning with Selective Perception and Hidden State.* – Rapport de Recherche n TR611, University of Rochester, Computer Science Department, juillet 1997.

[Menardais 03] S. Menardais. – *Modèle adaptatif multi-niveaux de mouvements humains à partir d'acquisitions.* – Thèse de doctorat, Université de Rennes 1, IRISA, 2003.

[Monzani 00] J. Monzani, P. Baerlocher, R. Boulic, D. Thalmann. – Using an intermediate skeleton and inverse kinematics for motion retargeting. – *Proc. of Eurographics 2000*, vol. 19(3), Interlaken, Suisse, 2000.

[Morasso 86] P. Morasso, V. Tagliasco (édité par). – *Human Movement Understanding.* – North-Holland, Amsterdam, 1986.

[Moray 93] N. Moray. – Designing for attention. *Attention, Selection, Awareness, and Control: A tribute to Donald Broadbent*, pp. 53–72, 1993.

[Moreau 98a] G. Moreau. – *Modélisation du comportement pour la simulation interactive : application au trafic routier multimodal.* – Thèse de doctorat, Université de Rennes I, Irisa, novembre 1998.

[Moreau 98b] G. Moreau, S. Donikian. – From psychological and real-time interaction requirements to behavioral simulation. – *Proc. of the Eurographics Workshop on Computer Animation and Simulation (EGCAS-98)*, Lisbon, septembre 1998.

[Multon 98] F. Multon. – *Contrôle du mouvement des humanoïdes de synthèse.* – Thèse de doctorat, Université de Rennes 1, IRISA, 1998.

[Newell 90] Newell. – *Unified theories of cognition.* – Harward university press, 111–157p., Londre, 1990.

[Niebur 96] E. Niebur, C. Koch. – Control of selective visual attention: Modeling the where pathway. – *Advances in Neural Information Processing Systems*, vol. 8, pp. 802–808. The MIT Press, 1996.

[Noser 95] H. Noser, O. Renault, D. Thalmann, N. Magnenat-Thalmann. – Navigation for digital actors based on synthetic vision, memory and learning. *Computer & Graphics*, 19(1):7–19, 1995.

[Noser 97] H. Noser, D. Thalmann. – Sensor based synthetic actors in a tennis game simulation. – *Computer Graphics International'97*, pp. 189–198, Hasselt, Belgium, juin 1997. IEEE Computer Society Press.

[Nougaret 98] J.L. Nougaret, F. Multon, B. Arnaldi. – Coarse-to-fine design of feedback controllers for dynamic locomotion. *The Visual Computer*, 13(9–10):435–455, 1998. – ISSN 0178-2789.

[Pissard-Gibollet 93] R. Pissard-Gibollet. – *Conception et commande par asservissement visuel d'un robot mobile.* – Thèse de doctorat, École des Mines de Paris, décembre 1993.

[Popovič 99] Z. Popovič, A. Witkin. – Physically based motion transformation. – Alyn Rockwood (édité par), *Proc. of SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 11–20, N.Y., août 1999. ACM Press.

[Rabie 96] T. Rabie, D. Terzopoulos. – Motion and color analysis for animat perception. – *Proc. of the 13th Nat. Conf. on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pp. 1090–1097, Menlo Park, août 1996.

[Reynolds 87] C. W. Reynolds. – Flocks, herds, and schools : A distributed behavioural model. *Computer Graphics*, 21:25–34, 1987.

[Rimey 92] R. Rimey, C. Brown. – Task-oriented vision with multiple Bayes nets. – A. Blake, A. Yuille (édité par), *Active Vision*, pp. 217–236, Cambridge, MA, 1992. The MIT Press.

[Robinson 64] D. Robinson. – The mechanics of human saccadic eye movements. *Journal of Physiology*, 174:245–264, 1964.

[Romeny 02] B. Ter Haar Romeny. – *Front-End Vision and Multiscale Image Analysis: Introduction to Scale-Space Theory.* – Kluwer Academic Publishers, 2002.

[Rose 96] C. F. Rose, B. Guenter, B. Bodenheimer, M. F. Cohen. – Efficient generation of motion transitions using spacetime constraints. – *Proc. of SIGGRAPH 96, in Computer Graphics Proceedings*, pp. 147–154, New Orleans, Louisiane, août 1996.

[Samson 91] C. Samson, M. Le Borgne, B. Espiau. – *Robot Control: the Task Function Approach.* – Clarendon Press, Oxford, Royaume Uni, 1991.

[Sanza 99] C. Sanza, C. Destruel, Y. Duthen. – Evolution of autonomous actors in an interactive real-time environment. – *Proc. of the Int. Conf. on Visual Computing (ICVC99)*, pp. 117–122, Kothrud, India, 1999.

[Tak 00] S Tak, O. Song, H. Ko. – Motion balance filtering. *Computer Graphics Forum*, 19(3):45–61, août 2000.

[Terzopoulos 97] D. Terzopoulos, T. Rabie, R. Grzeszczuk. – Perception and learning in artificial animals. – *Proc. of the 5th Int. Workshop on Artificial Life : Synthesis and Simulation of Living Systems (ALIFE-96)*, pp. 346–353, Cambridge, mai 1997.

[Terzopoulos 98] D. Terzopoulos, X. In, K. Joshi, C. Reynolds, T. Simpson. – Behavioral modeling and animation (panel): past, present, and future. – *Conference abstracts and applications: SIGGRAPH 98, July 14–21*, pp. 209–211, Orlando, USA, juillet 1998.

[Thalmann 90] D. Thalmann, O. Renault, N. Magnenat Thalmann. – A vision-based approach to behavioural animation. *The Journal of Visualization and Computer Animation*, 1(1):18–21, août 1990.

[Thalmann 95] N. Magnenat Thalmann, D. Thalmann. – Virtual actors living in a real world. – *Proc. of IEEE Computer Animation (CA'95)*, avril 1995.

[Thomas 99] G. Thomas. – *Environnements virtuels urbains : modélisation des informations nécessaires à la simulation de piétons.* – Thèse de doctorat, Université de Rennes I, Irisa, 1999.

[Thomas 00] G. Thomas, S. Donikian. – Virtual humans animation in informed urban environments. – *Proc. of IEEE Computer Animation (CA'00)*, pp. 112–121, Los Alamitos, Californie, mai 2000.

[Tolani 96] D. Tolani, N. Badler. – Real-time inverse kinematics for the human arm. *Presence*, 5(4):393–401, 1996.

[Tomlinson 00] B. Tomlinson, B. Blumberg, D. Nain. – Expressive autonomous cinematography for interactive virtual environments. – *Proc. of the 4th Int. Conf. on Autonomous Agents*, pp. 317–324, Barcelone, Catalogne, Espagne, juin 2000.

[Tu 94] X. Tu, D. Terzopoulos. – Artificial fishes: Physics, locomotion, perception, behavior. – *Proc. of SIGGRAPH 94, in Computer Graphics Proceedings*, pp. 43–50, Orlando, Floride, juillet 1994.

[Turner 91] R. Turner, F. Balaguer, B. Gobbetti, D. Thalmann. – Physically-based interactive camera motion control using 3D input devices. – *Proc. of Computer Graphics International (CGI'91*, pp. 135–145, Tokyo, Japon, mai 1991.

[Ware 90] C. Ware, S. Osborn. – Exploration and virtual camera control in virtual three dimensional environnements. – *Proc. of Symp. on Interactive 3D Graphics*, pp. 175–183, Mars 1990.

[Watt 92] A. Watt, M. Watt. – *Advanced animation and rendering techniques - Theory and practice.* – Addison-Wesley, New York, 1992.

[Webber 95] B. Webber, N. Badler. – Animation through reactions, transition nets and plans. – *Proc. of the Int. Workshop on Human Interface Technology*, Aizu, Japan, October 1995.

[Welman 93] C. Welman. – *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation.* – Master's thesis, Simon Frasier University, septembre 1993.

[Widyanto 91] T. Widyanto, A. Marriott, M. West. – Applying a visual perception system to a behavioral animation system. – *proc. of the Eurographics Workshop on Computer Animation and Simulation (EGCAS-91)*, pp. 89–98, Vienne, Autriche, 1991.

[Witkin 95] A. Witkin, Z. Popović. – Motion warping. – *Proc. of SIGGRAPH 95, in Computer Graphics Proceedings*, pp. 105–108, Los Angeles, Californie, USA, août 1995.

[Wolfe 96] J.M. Wolfe, G. Gancarz. – Guided search 3.0. – *Basic and Clinical Applications of Vision Science*, pp. 189–192, Dordrecht, Netherlands, 1996. Kluwer Academic.

[Yarbus 67] A. Yarbus. – *Eye movements and Vision.* – New-york : Plenum Press, 1967.

[Zeltzer 90] D. Zeltzer. – *Making Them Move : mechanics, control and animation of articulated figures*, chap. Task-level graphical simulation : Abstraction, representation, and control, pp. 3–33. – Norman I. Badler, Brian A. Barsky et David Zeltzer, 1990.

[Zordan 99] V. Zordan, J. Hodgins. – Tracking and modifying human motion with dynamic simulation. – *Proc. of SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 280–280, New-York, 1999.

Publications

Revues

N. Courty, E. Marchand. – Navigation et contrôle d'une caméra dans un environnement virtuel: une approche référencée image. *Technique et Science Informatiques, TSI*, 6(20):779–803, juin 2001.

E. Marchand, N. Courty. – Controlling a camera in a virtual environment. *The Visual Computer Journal*, 18(1):1–19, Février 2002. –

Congrès internationaux

E. Marchand, N. Courty. – Image-based virtual camera motion strategies. – *Proc. of Graphics Interface Conference, GI2000*, pp. 69–76, Quebec, Montréal, mai 2000.

N. Courty, E. Marchand. – Computer animation: a new application for image-based visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 223–228, Séoul, Corée du Sud, mai 2001.

N. Courty, E. Marchand, B. Arnaldi. – Through-the-eyes control of a virtual humanoïd. – *IEEE Int Conf. on Computer Animation 2001*, pp. 74–83, Séoul, Corée du Sud, novembre 2001.

Congrès et workshops nationaux

N. Courty, E. Marchand. – Navigation et contrôle d'une caméra dans un environnement virtuel : une approche référencée image. – *AFIG'99, 7èmes journées de l'association Française d'Informatique Graphique*, pp. 299–310, Reims, Novembre 1999.

Séminaires et rapports de recherche

E. Marchand, N. Courty. – *Visual Servoing in computer Animation. What you want to see is really what you get !* – Rapport de Recherche n1310, IRISA, mars 2000.

N. Courty, E. Marchand, B. Arnaldi. – Animation référencée vision d'un humanoïde de synthèse. – *13ème Journées des Jeunes Chercheurs en Robotique*, Rennes, septembre 2000.

N. Courty, F. Devillers, S. Donikian, F. Lamarche, D. Margery, S. Ménardais. – Towards Believable Autonomous Actors in Real Time Applications. *Conf. of Imagina 2002*, vol. 1, pp 154-163, Monte Carlo, Février 2002

Résumé

En synthèse d'images, la qualité d'une animation dépend souvent de spécifications complexes et/ou peu intuitives. Dans le cadre de cette thèse, nous nous sommes intéressés à l'étude de ce problème en appliquant une technique issue du domaine de la robotique : l'asservissement visuel. Cette technique permet de générer des mouvements 3D au travers d'une spécification 2D (dans l'image) d'une tâche visuelle. Deux grandes classes de problème ont été étudiés dans cette optique : le contrôle d'une caméra en environnement virtuel et la modélisation de l'attention visuelle pour les humanoïdes autonomes de synthèse. L'utilisation de l'asservissement permet dans les deux cas de spécifier simplement ce que l'on cherche à obtenir ; dans le cas de la caméra on spécifie ce que l'on cherche à observer, alors que dans le cas de l'humanoïde on spécifie ce qu'il doit observer. Grâce au principe de redondance, il nous a été permis de prendre en compte de manière dynamique et en temps réel des modifications de l'environnement, ou d'adresser des problèmes relatifs à l'animation. Ainsi dans le cadre de la caméra, les problèmes suivants ont été traités : suivi de trajectoires et planification, gestion des occultations et des collisions, problème de la « photo » ; dans le cadre de l'humanoïde : lien avec la locomotion, évitement des butées articulaires, gestion du réalisme. De plus, la validité de notre approche a été testée dans le cadre de l'animation comportementale, en modélisant à la fois des comportements type cinématographique pour la caméra, ainsi qu'un panel de comportements de perception pour l'humanoïde. Les résultats de cette étude ont été directement utilisés dans différents cadres applicatifs, dont l'animation d'un agent communicant réalisée par FranceTelecom R&D, ainsi qu'une démonstration de fiction interactive présentée lors d'Imagina 2002.

Mots clefs : Animation, asservissement visuel, redondance, simulation comportementale, cinématographie virtuelle, perception visuelle d'un humanoïde de synthèse

Abstract

Within computer graphics field, the quality of an animation often depends on complex or uneasy specifications. This thesis is concerned with the study of this problem through utilization of visual servoing. This technique has proved within the robotic field to be efficient to get 3D motions through 2D specifications of visual tasks (in the image space). Considering this context, two major classes of problem have been studied : camera control in a virtual environment and simulation of visual attention for synthetic humanoïds. Visual servoing offers easy specification of complex tasks. On behalf of redundancy formalism, some specific problems have been taken into account in interactive and real-time applications, like occultation avoidance or path following for the camera, and avoidance of joint limits, connection to locomotion or realism purpose for the humanoïd. Furthermore, the validity of our approach has been demonstrated within the context of behavioral simulation. Hence, we used this technique in the field of virtual cinematography and simulation of visual perception in the case of a synthetic humanoïd. Results of this framework have been directly used within different applications, like a conversationnal agent for France Telecom R&D, and a demonstration of interactive fiction presented at Imagina 2002.

Keywords: Animation, visual servoing, redundancy, behavioral simulation, virtual cinematography, visual perception for a synthetic humanoïd