

MÉMOIRE D'HABILITATION

présenté par :

Jean-Marc Hasenfratz

pour obtenir le diplôme

d'Habilitation à Diriger des Recherches
de

L'UNIVERSITÉ JOSEPH FOURIER
(GRENOBLE I)

Spécialité : Informatique

Insertion en temps réel d'un animateur
dans un monde virtuel

Date de la soutenance : 10 octobre 2005

Composition du jury :

Président : Claude Puech, Pr. Université Joseph Fourier, Grenoble
Rapporteurs : Alan Chalmers, Pr. University of Bristol, United Kingdom
Bernard Péroche, Pr. Université Claude Bernard, Lyon
Xavier Pueyo, Pr. Universitat de Girona, España
Examineurs : Pascal Guitton, Pr. Université de Bordeaux
François Sillion, DR INRIA, Artis, Grenoble

Habilitation préparée au sein du projet Artis
Projet commun entre le CNRS, l'INRIA, l'INPG et l'UJF

À ma mère

Remerciements

“Il était une fois un petit alsacien en DEA à Strasbourg auquel on proposa de faire un thèse à Limoges – il était nul en géo et ne savait même pas où était cette ville ! Son futur Directeur fut franc : «je ne peux rien te promettre, il n’y a pas d’école doctorale en Informatique à Limoges, tu n’auras donc pas de bourse du Ministère, je ferai mon possible pour que tu puisses faire des vacances. La seule chose que je peux te promettre, c’est une machine pour toi seul, une Indy». Le petit étudiant fit son stage de DEA à Limoges, le sujet de recherche lui convenait : “*Le lancer de faisceaux en synthèse d’images*”. Il fit un dossier de demande de bourse qu’il présenta à la Région Limousin, il fut boursier. C’était parti pour trois années de travail sous la direction de Djamchid Ghazanfarpour auquel il est toujours reconnaissant pour son enseignement et grâce auquel, il peut aujourd’hui écrire ces pages. Le 5 janvier 1998, le petit doctorant soutient sa thèse entouré de toute sa famille et d’amis de toujours. Un jour, par hasard, il répondit à une annonce parue sur la liste AFIG : «Cherchons Post-Doc pour travailler sur projet européen dans le domaine de la radio-sité au sein de l’équipe iMAGIS». «Oh là là, jamais ils ne vont vouloir de moi, ils sont bien trop forts, mais bon, je vais tenter ma chance quand même, sait-on jamais». «Bonjour, moi c’est Claude Puech, moi c’est François Sillion, moi c’est Georges Drettakis [...] on attend votre réponse». Le petit bonhomme n’en croyait pas ses oreilles, il allait travailler à Grenoble, dans l’équipe iMagis dans une superbe ambiance. De temps à autre, il y avait la “salade du chef” ou bien un “vénérable” pour faire des jeux de mots à la cafète – bon, je saute plusieurs années – Septembre 2000, le petit bonhomme devient Maître de Conférences à l’IUP Commerce et Vente. Janvier 2003, le projet *Artis* naît. Je prends de plus en plus de café avec Claude et François, ils m’apprennent le “métier”, je les harcèle de questions, ils me guident. Printemps 2005, je rédige ce mémoire et continue à grandir...”

Remerciements

C'est ici l'occasion de remercier les personnes qui m'ont permis de présenter ces travaux.

Merci à Claude Puech pour m'avoir accueilli dans son équipe iMAGIS, pour m'avoir suggéré de présenter cette habilitation, pour avoir accepté de présider le jury, pour ses nombreux conseils avisés... Je voudrais plus particulièrement te remercier, Claude, pour ta disponibilité et ta simplicité. Il me semble que tu es un "exemple" pour nombre d'entre nous.

Merci à François Sillion pour m'avoir permis de travailler avec lui dans différents projets européens, pour m'avoir permis d'effectuer mes recherches sur un thème nouveau dans l'équipe, pour m'avoir soutenu dans les moments de doute et pour avoir su canaliser les caprices d'un "Monsieur plus". Merci "Vénérable" !

Merci à Messieurs Alan Chalmers, Bernard Péroche et Xavier Pueyo, rapporteurs et Pascal Guitton, membres du jury, de m'avoir fait l'honneur et le plaisir d'avoir évalué mon travail.

Merci aux co-auteurs des différents papiers sans qui aucune publication n'aurait été possible. Je regrette que l'ordre des auteurs ait une telle importance dans ces publications puisque toujours, chacun d'entre eux s'y est investi pleinement.

Merci à Marc Lapierre pour avoir si efficacement concrétisé pendant près de 3 ans toutes les idées qui me traversaient l'esprit et sans qui les projets CYBER et CYBER-II n'auraient pu aboutir

Merci à toute l'équipe de l'IUP Commerce et Vente pour m'avoir permis d'effectuer ma recherche dans de bonnes conditions et pour la bonne ambiance que j'y retrouve semaines après semaines.

Merci à mes correctrices, Angéline, Marie-Paule et Patricia, sans lesquelles ce mémoire compterait un nombre de fautes d'orthographe qui nécessiterait certainement un *long int* pour le stocker !

Merci à Patricia pour sa disponibilité, son professionnalisme, l'aide au quotidien dans un système administratif parfois «étrange» !, mais aussi pour son amitié et nos discussions non professionnelles qui me rappellent de temps à autre qu'il n'y a pas que la Recherche dans la vie !

Merci aussi à tous ceux qui ne sont pas nommément cités mais qui ont participé, de près ou de loin, à mon travail : les collègues de travail d'Artis, ceux d'Evasion, ceux faisant partie des ACL..., les stagiaires, les doctorants...

Enfin, c'est à mon Ange que j'adresse mes remerciements les plus tendres. Elle, dont le soutien est inconditionnel, me permet de surmonter les batailles que je livre à de vieux démons. Merci à toi, ma femme.

Avant propos

Ce document englobe mes travaux de recherche depuis 2001 même si exceptionnellement, je ferai référence à mes travaux et publications effectués lors de mon Post-Doc. Les projets auxquels j'ai participé et qui ont constitué le cadre de mes recherches sont les suivants :

- **ARCADE** : “*Making Radiosity Usable*”, projet européen ESPRIT Reactive Long Term Research Programme, project #24944, octobre 1997-septembre 2000
<http://artis.imag.fr/Projects/Arcade/>
 - **SIMULGEN** : “*Realistic Simulation of Light for General Environments*”, projet européen ESPRIT Open Long Term Research Programme, Project #25772, octobre 1997 - septembre 1998
<http://iia.udg.es/Simulgen/1stphase.htm>
 - **DEREVE** et **DEREVE-II** : “*Rendu réaliste et non-photométrique, temps réel, de larges scènes animées*”, projet Région Rhône-Alpes, DEREVE 1999-2002, DEREVE-II 2003-2006.
 - **GrImage** : “*Grid and Image, PC Cluster for Vision, Visualization and Virtual Reality*”, 2003 - 2007
<http://www.inrialpes.fr/sed/grimage/>
 - **Cyber** : “*Insertion temps réel d'un animateur dans un monde virtuel*”, Action Concertée Incitative (ACI) «Jeunes Chercheurs» du Ministère de la Recherche, février 2002 - janvier 2005
<http://artis.imag.fr/Projects/Cyber/>
 - **Cyber-II** : “*Insertion temps réel d'un animateur dans un monde virtuel*”, Action Concertée Incitative (ACI) «Masse de données» du Ministère de la Recherche, septembre 2003 - août 2006
<http://artis.imag.fr/Projects/Cyber-II/>
-

Le fait d'avoir été le responsable de l'ACI «Jeunes Chercheurs» CYBER m'a permis de développer mon propre projet de recherche. J'ai pu construire, développer et orienter ma recherche dans le sens qui me semblait le plus intéressant. L'ACI «Masse de données» CYBER-II est la continuité de la première ACI, mais avec des ambitions très nettement supérieures.

Ces deux ACI sont au coeur de ma recherche, c'est à travers elles que j'ai encadré, doctorants, ingénieurs et stagiaires, et publié un certain nombre d'articles. C'est donc tout naturellement que ce mémoire d'Habilitation à Diriger des Recherches porte sur le sujet des deux ACI : *“Insertion en temps réel d'un animateur dans un monde virtuel”*.

Sommaire

Introduction	1
1 Description du projet CYBER	1
2 Organisation du mémoire	2
Plate-forme CYBER	5
1 Introduction	5
2 La salle	6
3 Les caméras	7
4 L'éclairage	9
5 Les calculateurs	10
6 Contributions	11
7 Et maintenant ?	11
Acquisition des silhouettes	13
1 Introduction	13
2 Acquisition des silhouettes dans CYBER	14
3 Extraction de fond	14
Fond uniforme	15
Fond quelconque	15
4 Détermination des silhouettes	16
Silhouettes sous forme de segments	17
Silhouettes sous forme de bitmap	17
Transmission des silhouettes	17
5 Contributions	20
6 Et maintenant ?	20

Modélisation	23
1 Introduction	23
2 Modélisation dans le projet CYBER.....	24
3 Calibrage des caméras.....	25
4 Modèles non-articulés.....	26
Approche volumique	26
Approche surfacique	27
Précision des silhouettes	28
5 Modèles articulés.....	29
6 Synchronisation	30
7 Parallélisation	32
8 Contributions.....	33
9 Et maintenant ?.....	34
Rendu réaliste	35
1 Introduction	35
2 Rendu réaliste dans le projet CYBER.....	36
3 Ombres	37
Importance des ombres	37
Ombres douces.....	39
4 "Texturage" omnidirectionnel.....	41
Jonction entre les textures	42
Problèmes de visibilité	43
Comment positionner les caméras ?.....	43
Cohérence des couleurs dans le temps.....	45
5 Contributions.....	45
6 Et maintenant ?.....	46
Rééclairage.....	46
Qualité des images	47
Rendu "non photoréaliste"	48
Interaction avec le monde virtuel	49
1 Introduction	49
2 L'interaction dans le projet CYBER	49
3 Temps réel et latence	50
4 Exemple d'interaction	51
5 Contributions.....	51
6 Et maintenant ?.....	52

Conclusion	53
1 Contributions	53
2 Et maintenant ?	54
Articles	57
1 A Praticle Analysis of Clustering Strategies for Hierarchical Radiosity	59
2 Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer	75
3 Real-Time Capture, Reconstruction and Insertion into Virtual World of ...	91
4 A survey of Real-Time Soft Shadows Algorithms	101
5 A Real-Time System for Full Body Interaction	125
6 Omnidirectional texturing of human actors from multiple view video seq.	137
7 Fast Voxel Carving Using OpenGL Facilities	145
8 Soft Shadow Maps: Efficient Sampling of Light Source Visibility	153
Bibliographie	167

Introduction

Depuis des années, nous imaginons qu'un jour nous allons pouvoir nous déplacer dans des mondes "virtuels", que tout en restant à Grenoble, nous allons pouvoir visiter la pyramide de Chéops ou la maison de Salvador Dalí à Figueres... Nous voudrions déambuler dans les couloirs du Louvre, nous arrêter devant Mona Lisa, décrocher ce tableau et l'admirer sous tous les angles... Nous voudrions faire le tour du futur tramway de Bordeaux, entrer à l'intérieur et imaginer une ballade en centre ville sous la lumière chaleureuse de l'automne ou un soir au coucher du soleil... Nous voudrions...

Tous ces rêves ont un point commun, il faut que l'on puisse se voir évoluer dans un monde réel ou imaginaire, y trouver sa place et interagir avec lui. Pour cela, une puissance de calcul très importante et du matériel spécifique (caméras hautes résolutions, réseaux hauts débits, très grands écrans, etc.) sont nécessaires et c'est seulement depuis peu que nous en disposons. Nous pouvons donc enfin nous atteler à la tâche. Sous ce côté rêveur, je vais présenter mes travaux de recherche effectués ces quatre dernières années. Ils ont été réalisés au travers des projets CYBER.

1 Description du projet CYBER

Le projet CYBER se place dans le cadre de la réalité virtuelle/augmentée. Il s'agit "d'insérer" une personne dans un monde virtuel. Cette personne doit **être totalement libre** de ses mouvements, elle ne doit donc pas porter d'équipement particulier comme des capteurs. La personne doit pouvoir **être accompagnée d'autres personnes et/ou d'objets quelconques** (voir *Figure 1*), nous ne nous limitons pas à l'insertion "d'éléments" connus *a priori* (par la suite, nous parlerons d'un "animateur" pour désigner une ou plusieurs personnes avec ou sans objets sup-



Figure 1: Deux animateurs et un ballon sont à insérer dans le monde virtuel.

plémentaires). L'animateur doit pouvoir **interagir** avec le monde virtuel, il faut donc un traitement **temps réel** (nous considérons que 25 images par seconde est la limite inférieure de vitesse d'affichage). L'animateur inséré dans le monde virtuel doit s'intégrer parfaitement, il faut donc que le rendu soit le plus **réaliste** possible en particulier, il faudra traiter les ombres portées et les conditions d'éclairage.

En pratique, un ensemble de caméras filme l'animateur sous différents angles. À partir de ces flux vidéo, nous pouvons reconstruire la géométrie 3D de l'animateur. Ces différents flux sont aussi utilisés au moment de l'insertion de l'animateur dans le monde virtuel afin que l'on puisse tourner autour de lui. Des ombres portées de l'animateur dans le monde virtuel sont calculées et affichées. L'ensemble des processus est présenté ci-contre (voir *Figure 2*).

2 Organisation du mémoire

Nous allons prendre le projet CYBER comme prétexte pour aborder mes thèmes de recherche de ces dernières années. Nous commencerons par une étude relativement technique mais importante sur l'élaboration de la **plate-forme** matérielle avant d'aborder les problèmes d'**acquisition des silhouettes** de l'animateur et de sa **modélisation**. Nous traiterons ensuite du **rendu réaliste** de l'animateur et des **interactions** possibles avec le monde virtuel.

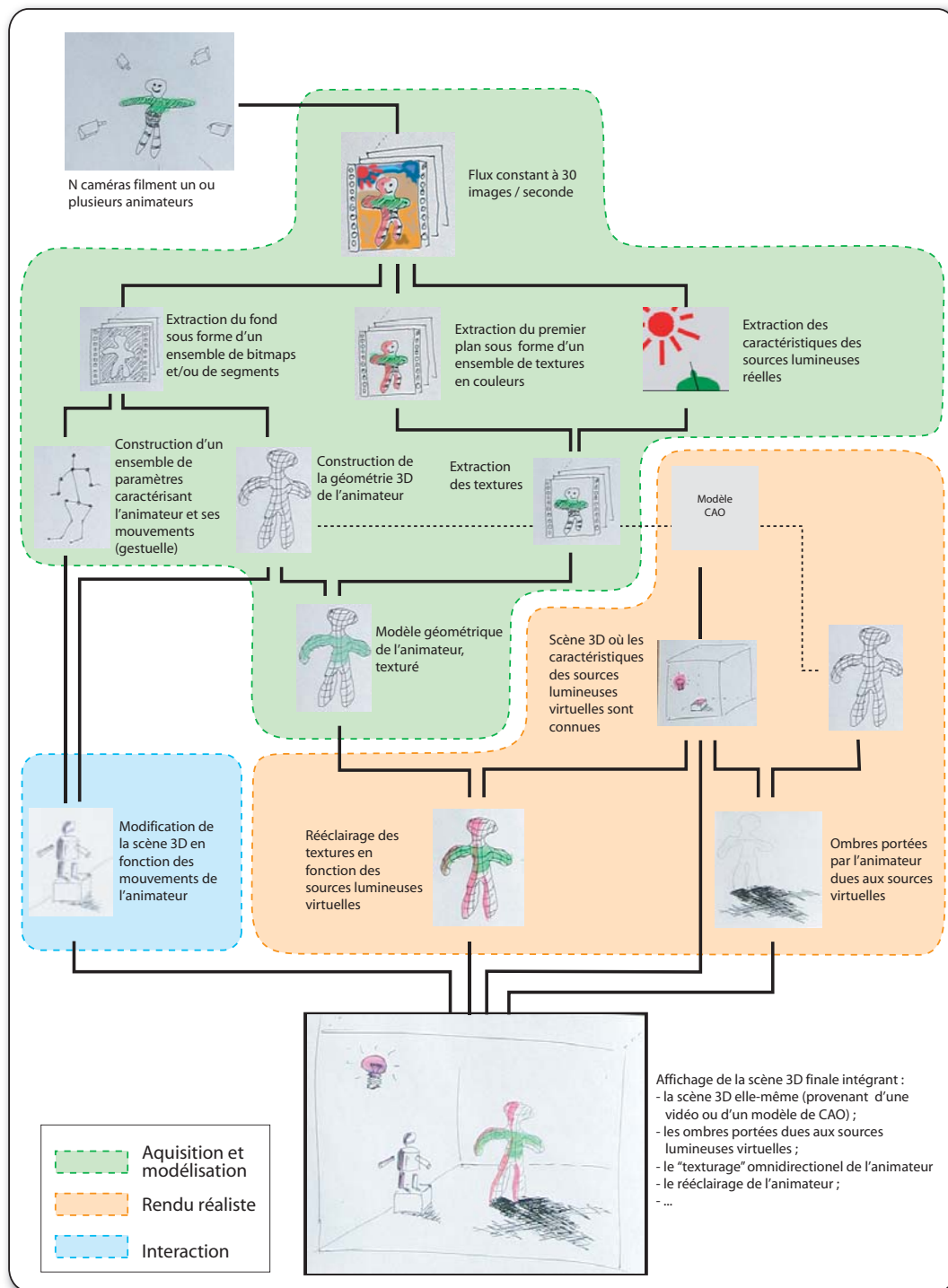


Figure 2: Schéma général du projet CYBER

Plate-forme CYBER

1 Introduction

Le côté technique est généralement passé sous silence lors des recherches, le plus souvent parce que son importance est faible et que l'on considère cette partie comme un "simple problème d'ingénieur". Dans notre cas, les choix matériels font partie intégrante de notre recherche. En effet, au commencement du projet, début 2002, il a fallu définir les besoins matériels en termes d'occupation du bâtiment, de type d'éclairage, de caméras, etc. Les choix effectués allaient contraindre nos recherches, les solutions choisies devaient donc être pérennes. Cette période d'analyse des besoins a pris du temps mais nous a permis par la suite d'avoir à disposition une plate-forme évolutive et répondant à tous nos besoins. Marc Lapierre, ingénieur sur les ACI CYBER nous a énormément aidé dans cette tâche.

Notons que la première ACI a démarré en février 2002 et que la deuxième se terminera en août 2006, soit une durée totale du projet de 56 mois. La présentation correspond donc à la situation actuelle¹ sachant que le projet évolue régulièrement au cours des mois.

Ce chapitre peut être lu comme un manuel pour bien construire son studio virtuel. Certes, ce n'est pas de la recherche fondamentale mais c'est une bonne manière de s'immerger dans le sujet. Nous allons étudier les principaux éléments de la plate-forme : la salle, les éclairages, les caméras et les ordinateurs. Nous allons considérer, pour chacun d'eux, les contraintes à respecter, nos choix et leurs motivations et résumer l'expertise acquise.

1. La rédaction de ce mémoire a été terminée le 12 juillet 2005.

2 La salle

La salle a été le premier souci car il n’existait dans le bâtiment aucun espace prévu à cet effet. Nous avons donc transformé une salle de réunion (voir *Figure 1*) en “salle de capture pour la réalité virtuelle”.

Les contraintes à respecter étaient les suivantes :

- La salle devait être suffisamment grande pour avoir du recul pour placer les caméras de sorte à filmer la totalité du volume où évoluerait l’animateur (8 m^3).
- Le plafond devait être suffisamment haut pour placer des éclairages et des caméras au-dessus de l’animateur.
- Les objets spéculaires étaient à supprimer pour éviter des problèmes de reflets des éclairages et de saturation des images.
- Il fallait placer un écran et un vidéo projecteur pour que l’animateur puisse se voir évoluer (voir *Figure 2*).
- Il fallait définir une zone pour un public (chercheurs travaillant sur des expérimentations ou visiteurs) qui ne soit pas dans le champ des caméras.
- Il fallait pouvoir supprimer la lumière du jour pour ne pas dépendre d’un éclairage naturel changeant, difficile à gérer lors de l’extraction de fond.



Figure 1: Salle de réunion avant réaménagement.



Figure 2: Animateur et écran de projection.

À partir de ces contraintes, nous avons dessiné un plan de cette nouvelle salle (voir *Figure 3*).

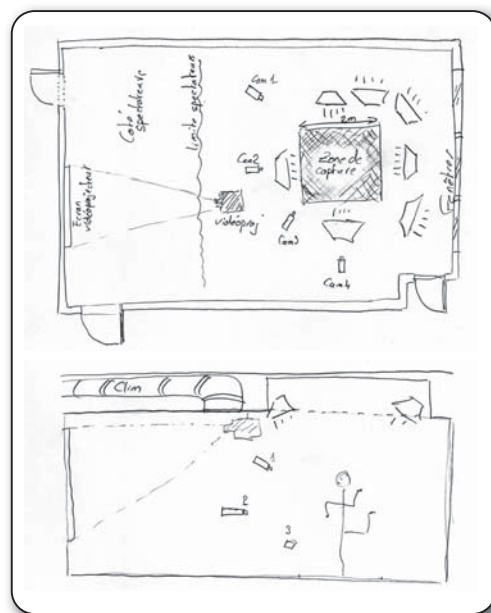


Figure 3: Plan de la salle.

Les choix effectués ont été les suivants :

- Nous avons rehaussé au maximum le plafond (voir *Figure 4*).
- Des rampes ont été installées au plafond pour y fixer les éclairages et pouvoir les déplacer facilement (voir *Figure 5*).
- Un vidéo projecteur a été accroché au plafond (voir *Figure 5*).
- Des stores ont été installés pour supprimer l'influence de l'éclairage extérieur.
- Les cadres en aluminium des fenêtres ont été repeints avec une peinture mate pour éviter les reflets.

Cette salle était parmi les premières du genre dans un laboratoire de recherche, nous n'avions donc que très peu d'exemple sur lesquels s'inspirer. Avec le recul, certains choix seraient à corriger.

Expertise :

- Le sol est une moquette très spéculaire ce qui ne facilite pas l'extraction de fond, il faudrait une surface mate ;
- Les stores que nous avons posés sont bleu foncé ce qui prête à confusion lorsque l'on dit que nous faisons de l'extraction sur fond quelconque. De plus, l'animateur porte souvent des habits sombres qui se détachent mal du bleu foncé. Une couleur beaucoup moins présente dans les vêtements aurait été plus judicieux. Dans notre cas, nous aurions pu utiliser le même jaune que les bandes sur le mur.



Figure 4: Rehaussement d'une partie du plafond.

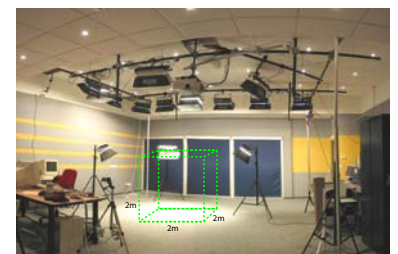


Figure 5: Salle réaménagée permettant une zone de capture de $8m^3$.

3 Les caméras

Le choix des caméras a probablement été la décision la plus difficile à prendre. En effet, il existait à l'époque que très peu de laboratoires de recherche disposant d'une plate-forme similaire de laquelle s'inspirer et les informations techniques n'étaient pas disponibles.

Les contraintes à respecter étaient les suivantes :

- Le débit des caméras devait être d'un minimum de 25 images par seconde pour espérer faire du temps réel.
- La focale devait être compatible avec la taille de la salle et du volume de capture.
- Le coût unitaire devait être modéré puisqu'il nous fallait un minimum de quatre caméras.
- La partie logicielle ("pilote") devait exister et être fiable.
- Les images ne devaient pas être entrelacées pour améliorer la qualité de l'extraction de fond.

- La connectique devait pouvoir atteindre au minimum 5 mètres.

Début 2002, peu de constructeurs proposaient des caméras numériques en couleurs. Nous avons donc le choix entre utiliser des caméras analogiques avec des cartes d'acquisition ou nous lancer dans l'achat de caméras numériques.

Le choix a été le suivant :

- Nous avons choisi des caméras numériques pour éviter les problèmes d'entrelacement des images.
- Nous avons opté pour les caméras au standard IEEE 1394 (ou Firewire) utilisées sous Windows pour l'existence et la fiabilité des pilotes.
- Nous avons donc acheté quatre caméras Sony DFW-VL500, 640x480 YUV(4:2:2)¹, avec un zoom 12x capable de produire des images à 30fps (voir Figure 6).



Figure 6: Caméra numérique Sony DFW-VL500.

Expertise :

Il existe actuellement un choix plus grand de caméras numériques aussi bien chez Sony que Marlin ou d'autres (voir Tableau 2.1). De plus, la nouvelle interface *camera link* permet des débits plus importants. Enfin, les pilotes sous Linux sont disponibles et fiables.

TABLEAU 2.1. Principaux modèles de caméras numériques disponibles en avril 2005

Modèle	Interface	Résolution	Fps	Ko/img	Débit ¹
SONY DFW VL500	IEEE 1394	640x480	30	600	140 Mb/s
SONY DFW X700	IEEE 1394	1024x768	15	1 536	180 Mb/s
SONY DFW sX900	IEEE 1394	1280x960	7.5	2 400	140 Mb/s
AVT MARLIN 046C	IEEE 1394	780x580	37	884	255 Mb/s
AVT MARLIN 145B2	IEEE 1394	1392x1040	10	2 828	220 Mb/s
JAI CV-M7+CL	camera link	1380x1030	24	1 735	325 Mb/s

1. Encodage YUV(4:2:2) : Y représente la luminance, U et V respectivement les composantes rouge et bleue (chrominances). En mode 4:2:2 nous utilisons 4 valeurs Y suivi de 2 valeurs U et 2 valeurs V pour coder la couleur. Il existe différentes manières de compresser ces données soit sur 12 bits (MPEG) soit sur 16 bits (Cinepak). Pour plus d'informations, voir le site <http://alcazar.xbecom.com/videogarage/spec/YUV.html>

1. Pour les interfaces ieee1394, le débit est calculé sur la base d'images au format YUV4:2:2 avec une moyenne de 16 bits par pixel. Le format *camera link* quant à lui utilise ici 10 bits par pixel.

4 L'éclairage

Pour éclairer la salle, nous avons pris contact avec une société construisant des plateaux de télévision. Nous avons aussi étudié attentivement la documentation "*Catalogue Concepteur de Lumière*" proposée sur le site www.balcar.com. Nous avons pu ainsi comprendre comment utiliser les éclairages pour limiter les ombres portées. L'idée est de saturer de lumière les murs et le sol pour "brûler" les ombres portées de l'animateur.

Les contraintes à respecter étaient les suivantes :

- L'éclairage devait être homogène et sans variation dans le temps pour que l'extraction de fond fonctionne correctement.
- Le nombre de sources lumineuses devait être suffisamment important pour éclairer toute la salle et réduire au maximum les ombres portées.
- Vu le nombre de sources lumineuses, celles-ci devaient dégager un minimum de chaleur.

Nous nous sommes très rapidement dirigés vers le matériel utilisé par les professionnels sur les plateaux de télévision.

Les choix effectués ont été les suivants :

- Un ensemble de 7 Balcar Duolite DMX, 125 W, 3800 cd accrochés au plafond pour éclairer les murs (voir *Figure 7*).
- Un ensemble de 4 Balcar Quadlite DMX 250 W, 8300 cd (voir *Figure 8*) accrochés au plafond pour éclairer l'animateur et le sol.
- Une autre série de 3 Balcar Quadlite DMX 250 W, 8300 cd fixés sur des pieds et braqués sur le sol au niveau de la zone de capture pour diminuer au maximum les ombres portées.

Expertise :

Une très grande quantité de lumière est effectivement indispensable pour diminuer les ombres portées. Notons qu'il est quasiment impossible de supprimer la totalité des ombres portées, particulièrement au niveau des pieds. Le choix de la température des lampes utilisées est important. En effet lors de différents tournages de films, nous avons eu des problèmes pour obtenir une bonne balance des blancs. Ceci parce que nos éclairages sont de type "lumière du jour" alors que celui du vidéo projecteur



Figure 7: Balcar accrochés à la rampe et Balcar sur trépieds..



Figure 8: Balcar Quadlite, 125 W, 3800 cd.

utilisé pour projeter le monde virtuel est plutôt de type tungstène. Le choix des tubes doit donc être en accord avec les autres sources de lumières potentielles (principalement les vidéos projecteurs).

5 Les calculateurs

Le choix des calculateurs a été relativement simple puisque nous avons décidé du type de caméras à utiliser (numériques avec des interfaces IEEE 1394). De plus, nous disposions d'un supercalculateur SGI Onyx.

Les contraintes étaient les suivantes :

- Chaque machine, associée à une caméra, devait être capable de traiter en temps réel le flux vidéo ainsi que l'extraction de fond ;
- Une machine devait être capable de recevoir via un réseau à 1Gb/s l'ensemble des silhouettes vues par la caméra, d'effectuer la reconstruction, de traiter les interactions et d'effectuer le rendu de l'ensemble de la scène.

Le choix a été le suivant :

- Un ensemble de 4 PC standards associés aux 4 caméras.
- Le supercalculateur SGI Onyx 3400 (voir *Figure 9*) pour l'intégration 3D — les PC de l'époque n'étaient pas assez puissants pour remplacer une telle machine.

L'ensemble des machines était connecté à un réseau 1 Gbits/s (voir *Figure 10*).



Figure 9: Supercalculateur SGI Onyx 3400
(photo © INRIA/R. Lamoureux)

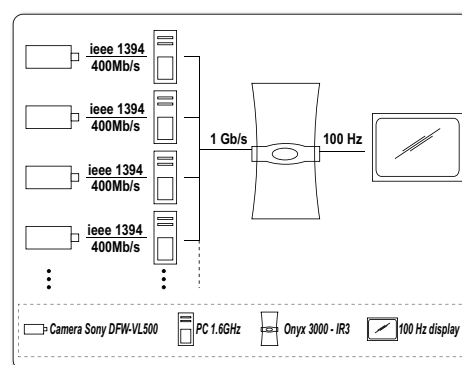


Figure 10: Architecture centrée sur d'un seul gros calculateur.

Expertise :

L'architecture utilisée a permis d'obtenir rapidement de bons résultats. Il est sûr qu'au vu de l'évolution des PC en puissance de calcul et en trai-

tement graphique, l'utilisation d'un supercalculateur n'est plus justifiée. Une grappe de PC semble un choix bien moins onéreux et plus facile à faire évoluer dans le temps.

6 Contributions

Dans ce chapitre, nos contributions sont principalement d'ordre technique. Nous n'avons pas eu à nous poser des questions de "recherche fondamentale". Néanmoins, nous avons testé un ensemble de configurations matérielles et proposé une solution cohérente pour construire une plate-forme d'acquisition fonctionnelle, fiable et pérenne. Il est évident que le matériel évolue très vite, en particulier les caméras et les ordinateurs, mais les contraintes sont toujours celles que nous avons énoncées. L'ossature générale reste donc valable et correspond à notre contribution.

7 Et maintenant ?

Les objectifs actuels sont plus ambitieux. Nous voulons plus de précision dans la reconstruction et un affichage de meilleure qualité. Ceci nous a amené à utiliser une grappe de plus d'une dizaine de PC, un mur d'images composé de 4x3 vidéos projecteurs (voir *Figure 11*) correspondant à une résolution de l'ordre de 4800x3000 pixels à la place des 1024x780 pixels habituels (voir *Figure 12*). Cette évolution matérielle fait partie du projet GrImage (Grid and Image) [GrImage]. Une description complète de cette infrastructure est proposée dans [HM04].



Figure 11: 12 vidéo projecteurs (photo © INRIA/Jim Wallace).



Figure 12: "Mur d'images" de 2x2.7m (photo © INRIA/Jim Wallace).

Acquisition des silhouettes

Définition *Silhouette* : contour représentant le profil de l'animateur du point de vue d'une des caméras.

1 Introduction

Dans ce chapitre, nous allons aborder deux thèmes de recherches importants : l'**extraction d'objets sur un fond quelconque** et le **calcul de leurs silhouettes** depuis différents point de vues.

L'extraction d'objets sur un fond quelconque consiste à différencier la région de l'image contenant l'information pertinente de l'arrière-plan de l'image. Les méthodes existantes reposent sur l'hypothèse d'un arrière-plan statique qui peut être appris *a priori* (voir *Figure 1*). La possibilité d'utiliser un fond quelconque à la place des fonds bleus ou verts habituels est très importante pour diverses raisons :

- elle apporte une plus grande liberté pour les tournages en disposant d'un plateau qui n'est plus limité à la surface peinte ou recouverte de tissus bleu ou vert ;
- le coût financier est moindre ;
- de nouvelles applications deviennent possibles comme le suivi de joueurs sur un terrain de jeu, les jeux vidéos où le joueur serait filmé chez lui et directement incrusté...
- le suivi et la surveillance vidéo (dans des bâtiments, sur des autoroutes...).

Nous allons aborder dans ce chapitre l'acquisition des silhouettes dans le projet CYBER, l'extraction de fond et la détermination des silhouettes.



Figure 1:Extraction sur fond quelconque (photomontage).

Nous présenterons nos contributions et les approches sur lesquelles nous voudrions travailler.

2 Acquisition des silhouettes dans CYBER

L'extraction sur fond quelconque et la détermination de la silhouette sont deux points très liés dans notre application. Techniquement, nous disposons, en entrée, des flux d'images d'une scène dynamique contenant un personnage et nous devons produire en temps réel un ensemble de silhouettes (voir *Figure 2*).

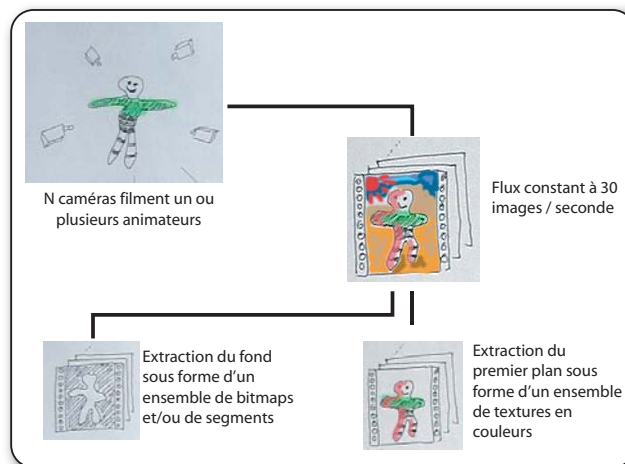


Figure 2: Module "Acquisition". À partir de n caméras, nous construisons n silhouettes ainsi que n images texturées.

Nous ne nous limiterons pas à un seul format de silhouette afin de pouvoir explorer plusieurs types de reconstructions, par contre, nous respecterons la contrainte temps réel.

Nous expliquerons par la suite pourquoi la précision de l'extraction de fond n'a pas besoin d'être très grande lors du processus de construction du modèle géométrique de l'animateur (voir *Chapitre 4—Modélisation, 4.3 Précision des silhouettes, page 28*). Cette extraction servira aussi à habiller le modèle géométrique. Nous verrons que dans ce cas, ce sera plutôt la résolution et la qualité de l'image (son piqué) qui seront importantes (voir *Chapitre 5—Rendu réaliste, 6.2 Qualité des images, page 47*).

3 Extraction de fond

Lorsque l'on se place dans le cadre de productions cinématographiques, nous trouvons soit des fonds bleus soit des fonds verts [Par02]. En

effet, ce sont le plus souvent des acteurs que l'on veut isoler du fond. Les couleurs bleues ou vertes facilitent alors la séparation de la couleur chair de la peau avec celle du fond.

3.1 Fond uniforme

Nos premiers travaux effectués dans ce domaine l'ont été au cours du stage de DEA [Fai01]. Nous avons travaillé sur l'algorithme de "chroma key" [SB96], [BL99]. Cette étude nous a permis de nous familiariser avec les techniques d'extraction de fond (voir *Figure 3*). En fait, nous avons vérifié qu'il était relativement facile de faire une extraction sur fond uniforme à condition de maîtriser l'éclairage (suffisamment puissant, bien orienté, etc.). L'avancée scientifique a été réalisée au niveau de l'optimisation de calcul pour obtenir des résultats en temps réel.

L'extraction sur fond uniforme est une technique éprouvée depuis des années, de nombreux systèmes professionnels temps réel existent sur le marché et remplissent leur fonction à merveille. Généralement, tous les traitements sont directement effectués par de l'électronique dédiée. La publication de A.R. Smith et J.F. Blinn [SB96] permet de se faire une idée très précise sur le sujet avec des références aux différents brevets sur cette technique.

3.2 Fond quelconque

L'extraction sur fond quelconque reste du domaine de la recherche. La littérature sur le sujet est importante. Nous pouvons retenir la comparaison entre différentes méthodes faite dans [TKBM99]. D'autres références intéressantes sur le sujet sont [WFSM02], [CPP03], [YPLL04].

Les hypothèses de travail sont généralement les suivantes : l'arrière plan ne bouge pas (seules des ombres peuvent s'ajouter) et les caméras sont calibrées une fois pour toute. Le principe d'une extraction sur fond quelconque se décompose en trois étapes (voir *Figure 4*) :

- Les caméras "apprennent" l'arrière plan en filmant quelques secondes l'environnement "vide". Notez qu'une seule image du fond pour chacune des caméras ne suffit pas. En effet, les capteurs électroniques oscillent dans le temps et les couleurs mesurées ne sont donc pas constantes. Il faut donc filmer quelques secondes pour moyennner ces couleurs.
- Les nouveaux éléments (objets, personnes...) sont introduits sur la zone de tournage.

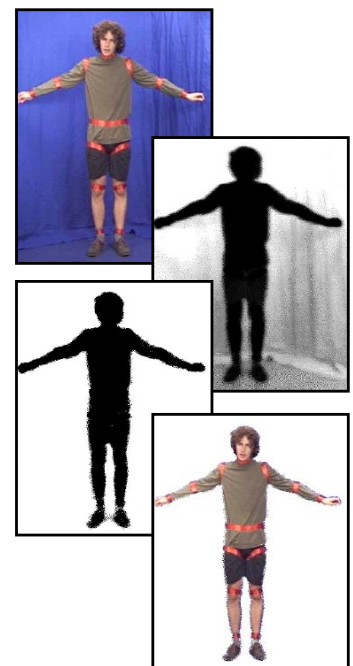


Figure 3: Image sur fond bleu, composante alpha, alpha filtré et image résultante.

- Une “différence” entre les images apprises et ce qui est filmé, est calculée. Elle permet de retrouver les nouveaux éléments. Le problème est de ne pas prendre en compte les ombres portées dues à l’ajout d’éléments dans la scène pour extraire exactement les contours des nouveaux objets ou personnes.

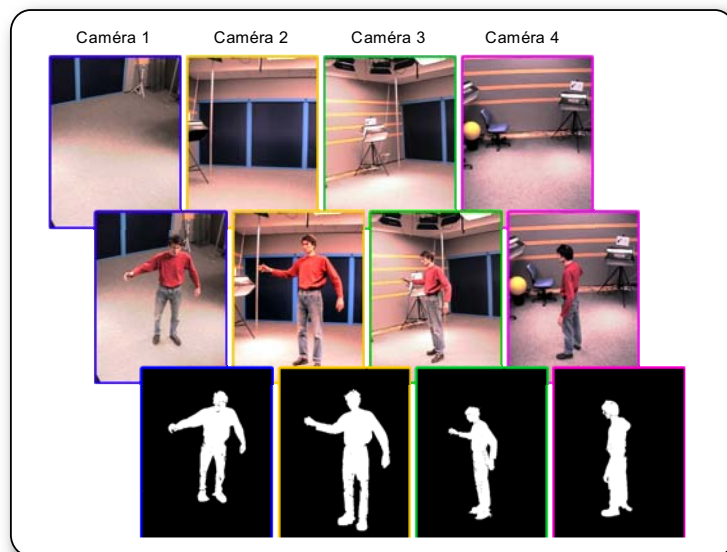


Figure 4:Extraction sur fond quelconque à partir de 4 caméras disposées autour de l'animateur.

Dans le cas du projet CYBER, nous pouvons alléger les contraintes. En effet, il n’est pas nécessaire d’avoir une extraction très performante puisque le bruit disparaît au moment de la reconstruction volumique comme nous le verrons pas la suite (voir 4.3 “Précision des silhouettes”, page 28).

Les détails techniques de l’implémentation de l’extraction sur fond quelconque ont été publiés (voir “Real-Time Capture, Reconstruction and Insertion into Virtual World of ...”, paragraphe “4.2 Background Substraction”, page 95).

4 Détermination des silhouettes

Une fois l’extraction de fond effectuée, nous disposons d’un ensemble d’images noir et blanc correspondant, pour chacune des caméras, aux masques des nouveaux éléments. Nous verrons dans le chapitre suivant (voir Chapitre 4–Modélisation, page 23) que, suivant l’approche utilisée, nous utiliserons des contours sous forme de segments ou des représentations sous forme de bitmaps.

4.1 Silhouettes sous forme de segments

Il y a plusieurs raisons pour vouloir travailler avec des silhouettes sous la forme de segments plutôt que directement avec les bitmaps. Ainsi la taille de la structure de données est plus petite et donc moins coûteuse à transmettre sur le réseau. Les algorithmes de reconstruction du modèle 3D peuvent travailler avec des polygones.

Il existe de nombreux algorithmes de segmentation d'images, dans le cas de CYBER, nous utilisons [DR95].

4.2 Silhouettes sous forme de bitmap

Le résultat direct de l'extraction de fond est une bitmap (avec des valeurs pouvant être comprises entre 0 et 1) à laquelle on peut associer ou non des valeurs de transparence. Comme nous le verrons par la suite (voir *Chapitre 4—Modélisation, 4.1 Approche volumique, page 26*), il peut être très intéressant de travailler directement sur ces images pour construire le modèle géométrique de l'animateur.

4.3 Transmission des silhouettes

Dans le cas du projet CYBER, nous utilisons une architecture distribuée où les informations transitent sur un réseau dédié. Même si celui-ci est à très haut débit (1 Gbit/seconde), il peut être rapidement saturé lorsque l'on augmente le nombre de caméras. Il est donc intéressant de faire une étude sur la proportion de surface occupée par l'animateur et par extrapolation du nombre de caméras que l'on peut utiliser simultanément sans saturer le réseau.

Nous nous plaçons dans l'hypothèse où il n'y a qu'un seul animateur à filmer et à insérer dans le monde virtuel. Nous allons quantifier la surface occupée par un animateur et le poids des images à transmettre. Nous distinguerons les bitmaps pouvant être utilisées lors du processus de reconstruction du modèle géométrique et les images en couleurs servant à l'habillage du modèle dans l'étape de rendu réaliste. Par extrapolation, nous évaluerons le nombre de caméras que l'on peut espérer utiliser réellement.

4.3.1 Proportion de surface occupée par un animateur

Considérons les résultats de l'extraction de fond effectuée lors d'une de nos manipulations (voir *Figure 5*). Nous pouvons constater que la surface occupée par l'animateur (pixels blancs) varie suivant la caméra entre

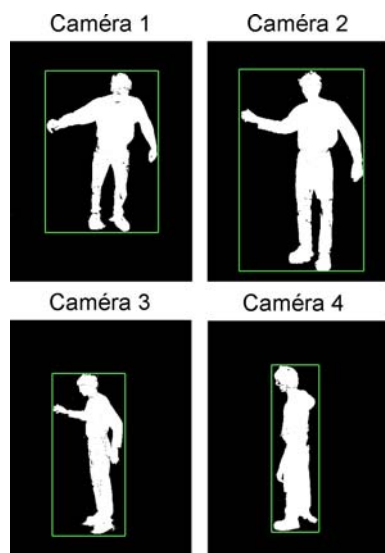


Figure 5: Quatre images avec les boîtes englobant les silhouettes (en vert).

10% et 21% de la surface totale de l'image (voir *Tableau 3.1*, colonne "Silhouette").

Une première optimisation très classique est de considérer uniquement la boîte englobante de la silhouette. Seuls les positions, tailles et contenus de ces boîtes seront transmis sur le réseau. En considérant le même exemple que précédemment (voir *Figure 5*), les surfaces des boîtes englobantes occupent de 17% à 56% de l'image complète (voir *Tableau 3.1*, colonne "Boîtes englobantes"). Cette optimisation est triviale à mettre en place et permet de gagner dans la plupart des cas plus de 50% de bande passante (voir *Tableau 3.1*). Nous pouvons aussi constater que le poids d'une bitmap correspond à moins de 10% du poids de l'image en couleur.

TABLEAU 3.1. Proportion maximale de la surface de la boîte englobante et de sa silhouette

	Image 640x480	Boîte englobante	Silhouette
Caméra 1	100%	41%	15%
Caméra 2	100%	56%	21%
Caméra 3	100%	26%	10%
Caméra 4	100%	17%	10%
Maximum	100%	56%	21%
Bitmap¹	37.5 Ko	21 Ko	7.88 Ko
Couleur²	600 Ko	336 Ko	126 Ko

1. Poids d'une bitmap avec un bit par pixel

2. Poids d'une image au format YUV:4:2:2 (16 bits par pixel)

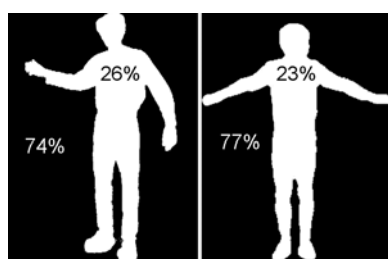


Figure 6: À gauche : l'animateur occupe toute la hauteur de l'image, à droite : l'animateur occupe toute la largeur de l'image.

Considérons les cas maximisant la surface occupée par l'animateur (voir *Figure 6*) correspondant aux configurations suivantes :

- une caméra est face à l'animateur ;
- un animateur ayant une taille de la tête au pied correspondant à la hauteur de l'image ;
- un animateur écartant ses bras de sorte à couvrir toute la largeur de l'image.

Dans ces cas, un quart seulement (26% et 23%) de la surface de l'image est occupé par l'animateur. Ces cas restent néanmoins purement théoriques puisque le champ de vision des caméras est nettement plus large que celui utilisé dans cet exemple extrême. En effet, l'animateur doit pouvoir élever les bras puis les écarter sans que l'on ait besoin de changer la focale des caméras. Nous reprendrons donc dans la suite, l'extrait de séquence précédent (voir *Figure 5*).

4.3.2 Nombre de caméras utilisables simultanément

Etudions à présent les débits nécessaires en fonction du nombre et du type de caméras utilisées et des données transmises (images entières, boîtes englobantes ou contenu des silhouettes). Une première approche consiste à envoyer les résultats bruts, en couleur, sur le réseau, c'est-à-dire sans aucune compression. Ceci a l'avantage d'éviter de passer du temps en compression/décompression. Bien sûr, si besoin était, il faudrait étudier soigneusement le type de compression à utiliser. Considérons parmi les caméras proposées (voir *Tableau 2.1, page 8*) uniquement celles permettant un débit temps réel, c'est-à-dire supérieur à 20 images par seconde (voir *Tableau 3.2*).

TABLEAU 3.2. Modèles de caméras numériques soutenant un débit de plus de 20 images par seconde.

Modèle	Résolution	Fps	Ko/img	Débit [Mb/s]
SONY DFW VL500	640 x 480	30	600 Ko	140 Mb/s
AVT MARLIN 046C	780 x 580	37	884 Ko	255 Mb/s
JAI CV-M7+CL	1380 x 1030	24	1 735 Ko	325 Mb/s

Dans le graphe ci-dessous (voir *Figure 7*), nous avons représenté les débits réseau nécessaires en fonction du nombre de caméras. Nous avons supposé transmettre les images dans leur résolution maximale, transmettre uniquement la boîte englobante (en supposant qu'elle couvrait au maximum 57% de l'image complète) et transmettre uniquement le contenu des silhouettes (en supposant qu'il couvrait au maximum 21% de l'image complète).

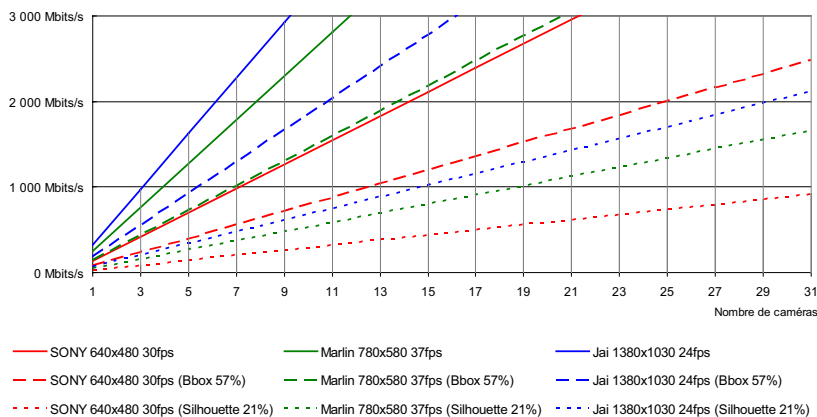


Figure 7: Débits des caméras en fonction de leur type et de leur nombre et des données transmises (images entières, boîtes englobantes ou contenu des silhouettes)

Ces résultats sont intéressants puisqu'ils montrent clairement les limites d'un réseau à 1 Gbits/s. Ainsi, transmettre les images dans leur résolution maximale n'est plus possible lorsque l'on désire travailler avec plus de sept caméras. Au contraire, le simple fait d'utiliser les boîtes englobantes permet de travailler théoriquement avec 7 caméras en haute résolution (1380x1030 pixels) et 13 caméras en 640x480 pixels à 30 Hz. Finalement, si l'on transmet uniquement le contenu des silhouettes, nous pouvons espérer utiliser quinze caméras en haute résolution et plus de trente en basse résolution (640x480 pixels).

Cette étude est bien sûr théorique, elle se base sur l'analyse des images produites à un instant d'une séquence avec un seul animateur et avec seulement quatre caméras, mais elle permet de fixer un peu mieux les idées et les limites d'un réseau à haut débit (1 Gbits/s).

5 Contributions

Nous avons développé un algorithme d'extraction sur fond quelconque. Celui-ci est robuste et fonctionne en temps réel [HLGB03] (voir *"Real-Time Capture, Reconstruction and Insertion into Virtual World of ..."*, page 91), [HLS04] (voir *"A Real-Time System for Full Body Interaction"*, page 125). Cet algorithme permet d'obtenir les silhouettes de l'animateur sous la forme de bitmap ou de segments.

Nous avons aussi proposé une étude théorique pour quantifier l'information utile à transmettre sur le réseau afin de fixer un cadre de recherche pour les prochaines extensions de la plate-forme (voir 4.3 *"Transmission des silhouettes"*, page 17).

6 Et maintenant ?

Une approche sur laquelle je souhaiterais travailler est l'augmentation de la résolution des silhouettes. En effet, nous avons besoin de précision pour distinguer par exemple les doigts des mains pour obtenir une interaction fine avec le monde virtuel. De plus, nous travaillons avec des "écrans" ayant des résolutions de plus en plus grandes comme des "murs d'images" (voir *Chapitre 2—Plate-forme CYBER, 7. Et maintenant ?*, page 11) comportant près 4800x3000 pixels [Grlmage]. Il devient alors difficile d'utiliser des images d'un animateur provenant de caméras ayant une résolution sensiblement plus faible (actuellement la résolution maximale serait d'environ 1300x1000 pixels – voir *Tableau 2.1*, page 8). Nous pouvons bien sûr parier sur l'augmentation de la résolution des capteurs numériques mais de la même manière, nous pouvons parier sur l'augmentation des dimensions et résolutions des "murs d'images". Il y aura donc toujours une course à la résolution comme il y a une course à la puissance.

Pour gagner en résolution, soit nous achetons systématiquement les caméras ayant la plus haute résolution et probablement les plus chères ce qui “résout” ponctuellement le problème, soit nous adoptons une attitude de chercheur pour trouver une solution pérenne. C’est bien sûr cette dernière qui m’intéresse et pour cela, je voudrais essayer de coupler plusieurs caméras ayant des résolutions convenables et des prix abordables. Ainsi un bloc de deux ou quatre caméras d’une résolution d’environ 1300x1000 permettrait d’obtenir respectivement des images d’environ 1300x2000 pixels ou 2600x2000 pixels – en pratique la résolution serait légèrement moindre pour prendre en compte le recouvrement des régions (voir *Figure 8*). Chacune de ces caméras ne filmerait qu’une partie du volume à capturer.

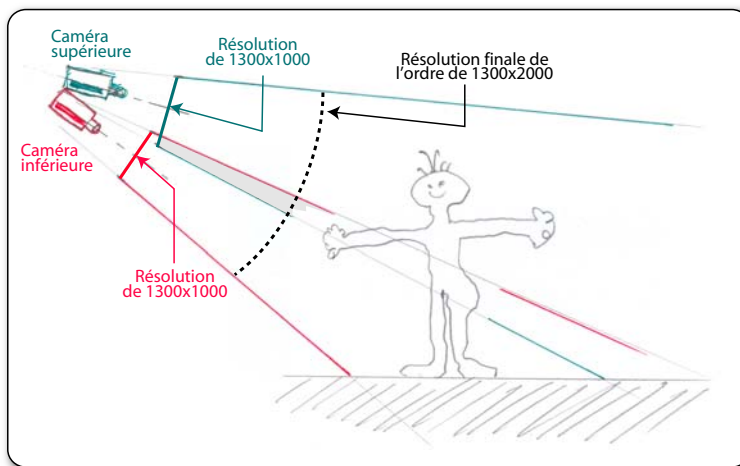


Figure 8: Utilisation d’un bloc de deux caméras pour doubler la résolution.

Le problème scientifique serait alors de développer des algorithmes de calibrage des caméras extrêmement précis pour minimiser les problèmes au niveau du recouvrement des régions filmées. Des travaux dans ce domaine existent déjà mais dans un cadre très différent du nôtre. En effet, [WJVT+05], proposent de regrouper une centaine de caméras de faible résolution (640x480) pour obtenir des images allant jusqu’à 6900x3500 pixels. Néanmoins, l’installation d’un tel système est très complexe et d’un prix prohibitif.

Si l’on utilise les systèmes permettant d’obtenir des images de haute résolution, il n’est pas sûr que les algorithmes d’extraction de fond passent à l’échelle et restent temps réel. Enfin, comme nous l’avons étudié précédemment (voir 4.3 “Transmission des silhouettes”, page 17), nous aurons certainement des problèmes de saturation du réseau.

Modélisation

1 Introduction

Un *modèle* de l'animateur est nécessaire pour être incrusté dans le monde virtuel. Le modèle peut être simple : les studios virtuels utilisent par exemple un plan vertical sur lequel la texture du présentateur est plaquée. La carte graphique se charge alors d'afficher ce quadrilatère texturé et la géométrie du monde virtuel. Les problèmes simples de visibilité sont donc traités automatiquement par le processeur graphique. Un animateur pourra se placer derrière un pilier sans problème particulier, par contre, il ne pourra pas enlacer ce même pilier. En effet, le polygone sur lequel est projetée la texture de l'animateur (voir *Figure 1*) ne peut pas représenter des parties de l'animateur à la fois devant et derrière le pilier. D'autre part, des effets comme les ombres portées ou le déplacement de la caméra virtuelle autour du modèle sont interdits sous peine de voir la supercherie (voir *Figure 1*). Cette approche est donc très limitée.

Des modèles plus évolués sont par conséquent nécessaires. Nous considérons ici deux catégories d'approches en fonction des connaissances *a priori* disponibles sur les objets à modéliser : les **modèles non-articulés** pour lesquels l'objet est supposé être constitué de plusieurs parties non-connexes de types inconnus et les **modèles articulés** pour lesquels l'objet est supposé être constitué de parties de types connus reliées par des articulations. Cette classification est bien évidemment non exhaustive mais correspond à deux thèmes de recherche que nous avons explorés.

Comme nous le verrons par la suite, un modèle non-articulé de l'animateur peut être obtenu en temps réel. Si de plus, le modèle est volumique, il permettra l'ajout d'ombres portées (voir *Chapitre 5–Rendu réaliste, 3. Ombres, page 37*), un habillage omnidirectionnel de l'animateur (voir *Chapitre 5–Rendu réaliste, 4. “Texturage” omnidirectionnel, page 41*) et des inte-

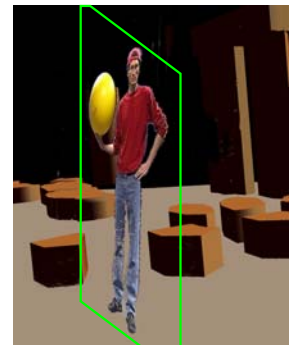


Figure 1 : Haut : la texture de l'animateur est plaquée sur un polygone. Bas : les ombres portées ne sont pas en accord avec l'image insérée.

rations avec le monde virtuel (voir *Chapitre 6—Interaction avec le monde virtuel, 4. Exemple d’interaction, page 51*).

Un modèle articulé permettra quant à lui une reconnaissance de la gestuelle, son interprétation et des interactions complexes avec le monde virtuel. Nous avons peu travaillé sur ce type de modélisation, c’est pourquoi seule une tentative de construction d’un squelette est proposée (voir *4. “Modèles non-articulés”, page 26*).

Dans ce chapitre, nous allons aborder les thèmes suivants : la modélisation dans le cadre du projet CYBER, le calibrage des caméras, la construction de modèles non-articulés et la construction de modèles articulés. Nous évoquerons aussi les problèmes dus à la synchronisation ou non des caméras ainsi que les possibilités de parallélisation de la construction.

2 Modélisation dans le projet CYBER

Dans le cadre du projet CYBER, la modélisation recherchée doit être avant tout effectuée en temps réel et si possible de bonne qualité.

Le module “d’extraction” fournit n silhouettes sous forme d’images ou de segments. Il s’agit alors, à partir de ces informations reçues en entrée, de construire un module “Modélisation” offrant une représentation articulée et/ou géométrique de l’animateur (voir *Figure 2*).

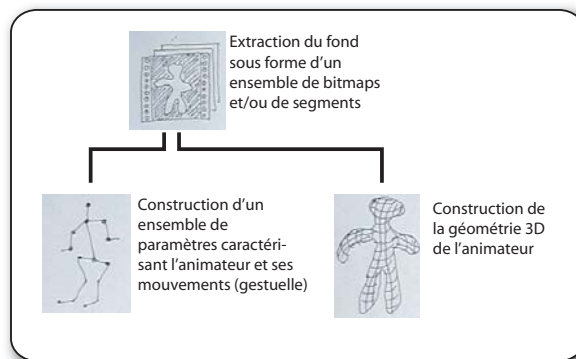


Figure 2: Module “Modélisation”. À partir de silhouette, nous calculons un squelette et/ou un modèle géométrique.

Le nombre de caméras ainsi que leurs positions sont des paramètres importants dans la modélisation. Il semble évident que plus nous disposerons de caméras et plus leurs positions seront judicieuses, plus la modélisation sera précise et sans ambiguïté. Dans le cas du projet CYBER, nous avons commencé par travailler avec quatre caméras. Nous les avons positionnées de manière intuitive de sorte à découper au mieux les silhouettes

de l'animateur (voir *Figure 3*). Dans le cas où les caméras ne servent qu'à la reconstruction, il n'est pas nécessaire d'en placer face à face puisque les silhouettes seront presque identiques. En effet, l'effet de perspective est négligeable en comparaison de la précision des silhouettes calculées. Nous avons donc positionné nos caméras tous les 45° en les répartissant en hauteur. Notons que quatre caméras seulement permettent d'obtenir une modélisation très correcte.

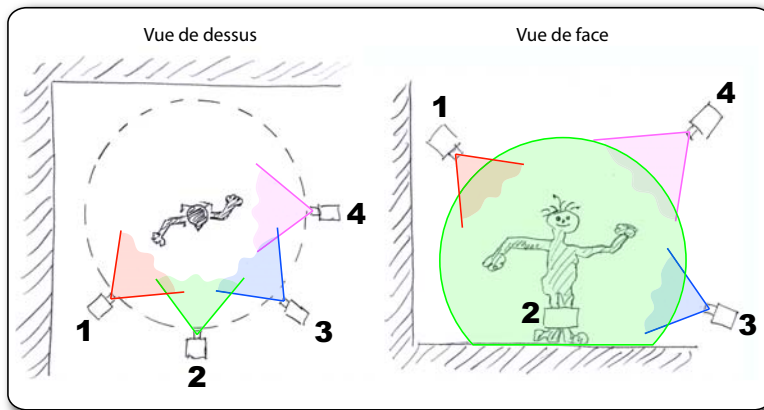


Figure 3: Position des quatre caméras pour obtenir une reconstruction acceptable.

Cette discussion sur les caméras, leurs positions et leurs nombre sera complétée par la suite puisqu'elle doit tenir compte du fait que ces caméras peuvent être utilisées pour un rendu "réaliste" (voir *Chapitre 5—Rendu réaliste*, 4.3 *Comment positionner les caméras ?*, page 43).

3 Calibrage des caméras

Le calibrage des caméras consiste à déterminer les caractéristiques propres des caméras, les focales, les distorsions..., ainsi que leurs positions et orientations respectives. Ces informations sont indispensables pour modéliser l'élément filmé. Le calibrage de caméras est largement étudié dans le domaine de la vision par ordinateur. De nombreuses solutions existent en repérant des points particuliers sur une surface plane, en général un damier [Tsa86], [Zha00]. Dans le cas de CYBER, nous avons utilisé la bibliothèque du domaine publique [OpenCV]. Le calibrage est effectué de manière très simple en utilisant un damier d'un mètre de côté composé de carreaux de 10 cm (voir *Figure 4*).



Figure 4: Application permettant de piloter les caméras (focale, diaphragme...) et de calculer les matrices de calibrage.

4 Modèles non-articulés

Lorsque plusieurs points de vue sont disponibles, il est possible de construire l'enveloppe visuelle [Lau94] de l'objet en calculant l'intersection des faisceaux d'observation associés aux silhouettes. Cette approximation est plus ou moins précise en fonction du nombre de points de vue disponibles. Plusieurs approches existent pour déterminer cette enveloppe visuelle :

- Les approches **volumiques** [CKBH00], [Dye01], [Lok01], [SCMS01] : l'idée est de sculpter un volume dans l'espace. À partir d'un volume initial composé de cellules élémentaires, les *voxels*, nous éliminons les cellules se projetant, dans une, ou plusieurs, des différentes images considérées, à l'extérieur de la silhouette.
- Les approches **surfaciées** [CG99], [MBM01], [BF03], [FB03] : l'idée est de déterminer la surface de l'enveloppe visuelle correspondant aux différents points de vue et délimitant le volume d'intersection des faisceaux d'observations. Ces approches peuvent être fondées sur des algorithmes de calcul d'intersection de polyèdres ou de polygones.

Bien que populaires, les approches basées sur les enveloppes visuelles présentent des inconvénients. En premier lieu, elles ne produisent pas un modèle de la surface observée mais une enveloppe de cette dernière et, l'approximation que cela constitue n'est pas toujours très précise, en particulier lorsque peu de caméras sont disponibles. En deuxième lieu, les parties concaves d'un objet n'apparaissent pas dans l'enveloppe visuelle du fait qu'elles ne sont pas présentes dans les silhouettes de l'objet. De récentes approches volumiques [Kut00], [KS00] proposent de considérer l'information photométrique en réponse à ce dernier problème.

Un état de l'art des différentes approches existantes est proposé dans [HLS04] (voir "*A Real-Time System for Full Body Interaction*", paragraphe "2. Previous works", page 128)

Nous allons à présent étudier plus en détails les deux approches, volumique et surfacique et discuter de l'importance limitée de la précision des silhouettes pour la reconstruction de nos modèles.

4.1 Approche volumique

L'approche volumique consiste à diviser régulièrement le volume dans lequel évolue l'animateur en cubes ou *voxels*. L'idée est alors de sculpter cette grille en supprimant, pour chaque caméra, les voxels n'appartenant pas à la silhouette vue par celle-ci (voir "*Real-Time Capture, Reconstruction*

and *Insertion into Virtual World of ...*”, paragraphe “5. Shape Estimation”, page 96). Cette technique est généralement appelée “voxel carving” (voir Figure 5).

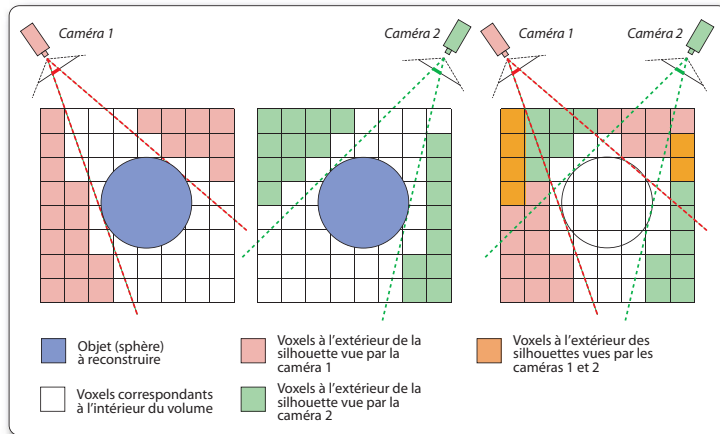


Figure 5: Principe (en 2D) du “voxel carving” pour une sphère.

Une fois les voxels formant l’animateur trouvés (voir Figure 7), nous utilisons un algorithme de “marching cubes” (voir “*Real-Time Capture, Reconstruction and Insertion into Virtual World of ...*”, paragraphe “5.3 Marching cube on binary voxels”, page 98) pour lisser la surface (voir Figure 6).

Pour obtenir des résultats en temps réel, nous avons optimisé l’algorithme de “voxel carving” en utilisant directement les silhouettes sous forme de bitmap et en effectuant le plus d’opérations possibles sur la carte graphique. L’approche que nous avons proposée fonctionne sur tout type de carte proposant les fonctionnalités OpenGL. La description générale est proposée dans (voir “*Real-Time Capture, Reconstruction and Insertion into Virtual World of ...*”, paragraphe “5.1 Hardware Assisted Voxelisation”, page 96). Une description beaucoup plus technique comportant une discussion pour d’autres applications que celle du projet CYBER a été soumise à publication (voir “*Fast Voxel Carving Using OpenGL Facilities*”, page 145).

4.2 Approche surfacique

Dans une approche de reconstruction surfacique, chaque silhouette obtenue pour chacune des positions des caméras sert à construire un faisceau. L’intersection de l’ensemble des faisceaux est calculée et forme l’enveloppe visuelle de l’objet (voir Figure 8).

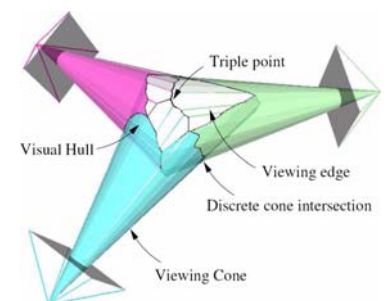


Figure 8: Reconstruction surfacique d’une sphère à partir de trois caméras (extrait de [FMBR04]).



Figure 7: À gauche : vue prise par une des quatre caméras. À droite : reconstruction voxelique.

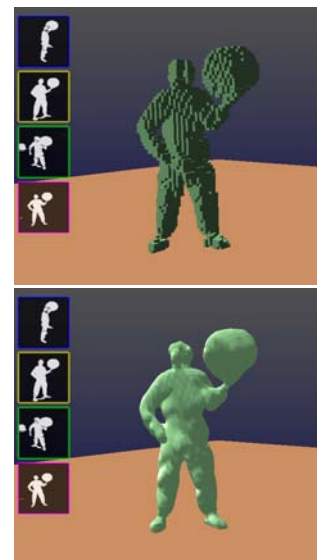


Figure 6: En haut : résultat brut du “voxel carving”, en bas : résultat après lissage par “marching cubes”. Les quatre vignettes sur la gauche correspondent aux quatre silhouettes utilisées pour la reconstruction.



Figure 9: Reconstruction surfacique d'une personne (extrait de [ABFM+04]).

Une description d'un certain nombre d'approches surfaciques est décrite dans [CG99]. De nouvelles solutions sont proposées dans [MBM01], [BF03], [FB03].

Une reconstruction surfacique temps réel d'un animateur a été développée sur une grappe de PC dans [ABFM+04]. Les résultats obtenus avec quatre caméras et le module d'extraction de fond développé dans le projet CYBER sont proposés ci-dessous (voir Figure 9).

4.3 Précision des silhouettes

Lors du processus d'extraction de fond, il est possible que du bruit ou un mauvais réglage des seuils de détection entraînent des erreurs. On peut ainsi obtenir des images avec des zones ne correspondant pas à la présence de l'animateur (voir Figure 10).

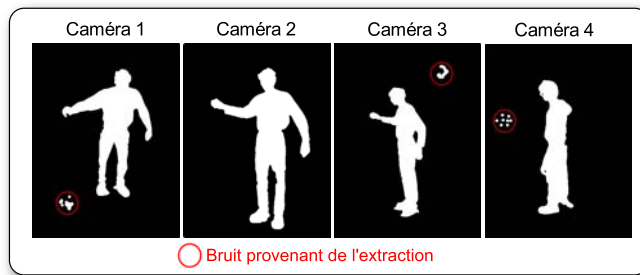


Figure 10: Les images provenant des caméras 1, 2 et 3 sont bruitées (cf. cercle rouge). Les régions bruitées ne correspondant pas à une même position dans l'espace, ces erreurs n'auront aucune conséquence.

À partir du moment où ce bruit ne correspond pas, sur toutes les images d'extraction, à une même région dans l'espace, ce bruit ne portera pas à conséquence (voir Figure 11).

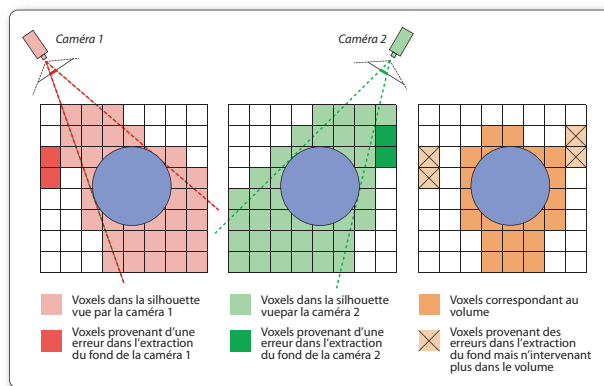
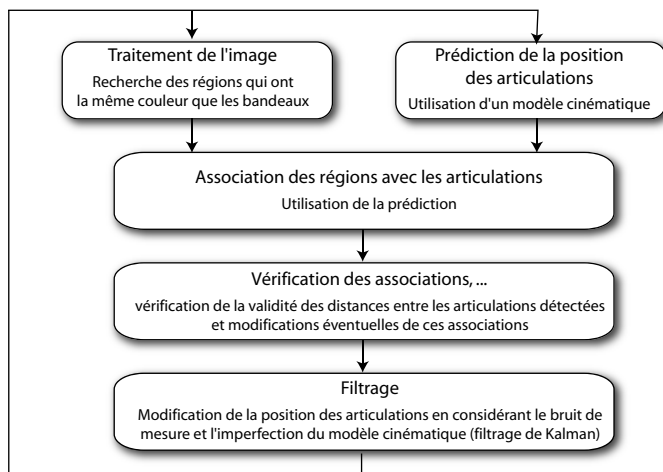


Figure 11: Certaines erreurs lors de l'extraction disparaissent au moment de la reconstruction.

Ainsi, des images légèrement bruitées, permettent d'obtenir un modèle de bonne qualité. Ceci vaut quel que soit l'approche, surfacique ou volumique.

5 Modèles articulés

Dans cette partie, nous allons présenter les travaux effectués lors d'un stage de DEA en 2001 [Fai01]. Un des buts du stage était d'être capable de filmer, avec une seule caméra, un animateur et de reconstruire en temps réel son squelette simplifié à 14 articulations (voir Figure 12). Comme nous ne voulions pas placer de capteurs sur l'animateur, pour ne pas le gêner dans ses mouvements, nous avons travaillé sur le suivi des 14 articulations. Pour simplifier la reconnaissance de ces points particuliers, nous avons utilisé 14 bandeaux de différentes couleurs (voir Figure 13). L'algorithme de suivi des articulations était le suivant :



Les résultats étaient plutôt bons, le suivi des bandeaux couplé avec une prédiction de leurs positions permettait de construire, en temps réel, le squelette de l'animateur.

Ces premiers travaux ont été effectués avant que la salle de capture ne soit construite ; la salle utilisée était petite et l'éclairage était limité. Il arrivait que des pixels rouges apparaissent au niveau des yeux ou des lèvres pendant quelques images. Ces pixels étaient alors confondus avec un bandeau rouge. Il est néanmoins intéressant de constater que l'algorithme mis en place retrouvait le squelette exact et s'adaptait très bien.

Les limites de l'approche ont été atteintes et nous en avons retenu les points suivants que nous développerons par la suite :

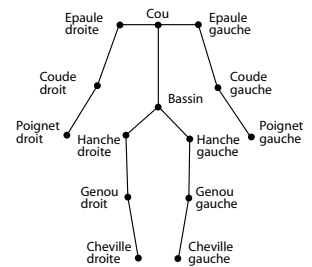


Figure 12: Modèle humain comportant 14 articulations.

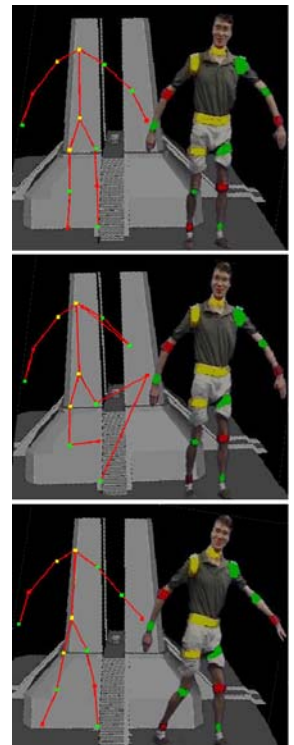


Figure 13: Trois images extraites d'une séquence d'environ dix images. L'algorithme de reconstruction du squelette peut se tromper lorsqu'il perd un bandeau mais rattrape automatiquement son erreur.

- Un mauvais éclairage introduit des couleurs erronées comme des yeux ou des lèvres rouges. Ces “fausses couleurs” peuvent alors être confondues avec des bandeaux.
- Une seule caméra ne permet pas de suivre en permanence tous les bandeaux, même si la prédiction autorise la perte de bandeaux durant quelques images.

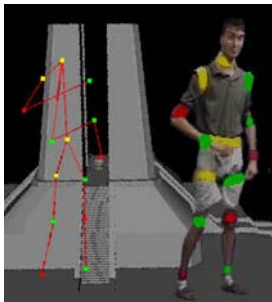


Figure 14: Les contraintes anatomiques ne suffisent pas toujours pour assurer une bonne reconstruction du squelette.

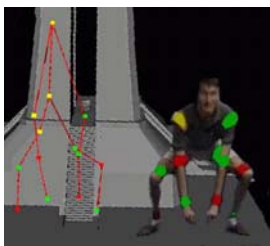


Figure 15: La disparition d'un bandeau rend le squelette incohérent.

En effet, considérons trois images prises sur une série de dix (voir Figure 13). La première et la dernière images correspondent respectivement au début et à la fin de la séquence. Ces deux images étant de bonne qualité, le squelette est cohérent. Dans l'image du milieu, des pixels rouges apparaissent au niveau du visage et la cheville droite n'est pas suffisamment éclairée pour que son bandeau soit reconnu. Le système associe alors la cheville droite au genou gauche, le genou gauche au poignet gauche et le poignet gauche au pixel rouge du visage. Le squelette ne correspond plus à une position possible. Nous avons donc ajouté des contraintes morphologiques en fixant les distances entre les articulations. Malheureusement, certaines positions sont mal interprétées (voir Figure 14). Il est aussi possible que les bandeaux ne soient plus visibles, la reconstruction devient alors incohérente (voir Figure 15).

En conclusion, la construction d'un modèle articulé tridimensionnel est un problème complexe à résoudre, d'autant plus que nous ne disposons que d'une seule caméra. Nous avons divisé le problème de la reconstruction 3D en deux parties. Nous avons cherché à déterminer la position des articulations dans l'image. Pour cela, nous avons fixé des rubans de couleur au niveau des articulations de l'animateur. Un filtrage de Kalman a permis de suivre les articulations. Une fois que l'on connaissait la position 2D des articulations, les contraintes anatomiques nous ont permis de déterminer les positions des articulations dans l'espace. Depuis ces recherches, nous n'avons plus travaillé sur la reconnaissance du squelette. Nous avons préféré nous concentrer sur les modèles non-articulés et sur le rendu réaliste.

6 Synchronisation

En théorie, la synchronisation des caméras doit être parfaite pour obtenir une reconstruction exacte. En effet, si les images ne sont pas prises exactement au même instant, l'objet reconstruit risque d'être incomplet comme nous pouvons le constater avec l'exemple d'une balle filmée par deux caméras non synchronisées (voir Figure 16). Le moyen le plus sûr pour obtenir une telle synchronisation est d'utiliser un générateur d'impulsions relié aux prises de synchronisation externe des caméras. Les caméras numériques que nous utilisons ont un débit de 30 Hz non synchronisé, malheureusement ce débit tombe à 15 Hz lorsque l'on active la synchronisation matérielle. Notons qu'un très grand nombre de caméras numériques réagissent de la même manière.

Pour passer outre cette limitation, nous avons utilisé une synchronisation logicielle (voir "*A Real-Time System for Full Body Interaction*", paragraphe "4.2.3 Synchronisation", page 131). Pour cela, nous utilisons un système de cache et de datation des images provenant des caméras (voir "*A Real-Time System for Full Body Interaction*", paragraphe "3.2. Software Architecture", page 129).

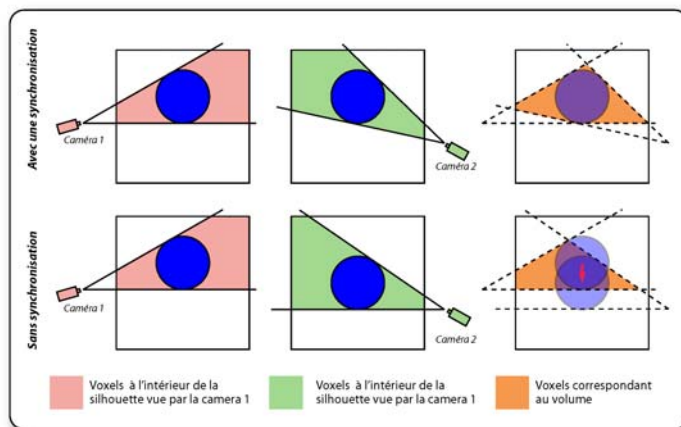


Figure 16: Ligne du haut : les deux caméras sont synchronisées, la reconstruction est exacte. Ligne de bas : la balle bleue a le temps de se déplacer entre les images prises par les deux caméras, la reconstruction est incomplète, seule une partie de la balle est reconstruite.

Étudions à présent les conséquences d'une telle synchronisation logicielle. Lors de l'acquisition des images à partir des différentes caméras non synchronisées fonctionnant à 30 Hz, nous pouvons assurer que le décalage maximum est strictement inférieur à 33 ms (voir Figure 17). Si l'objet ne bouge pas, la reconstruction est bien sûr exacte. Théoriquement, les problèmes peuvent arriver lorsque l'objet bouge rapidement. Néanmoins, nous n'avons constaté aucun problème "choquant" et les résultats étaient plutôt réalistes. Ceci peut s'expliquer par le fait qu'il faut prendre en compte le temps d'exposition (indépendant de la fréquence d'acquisition) pour chaque caméra, et le fait qu'un objet en mouvement entraîne un flou de bougé. La silhouette produite est alors plus grande que la silhouette réelle de l'objet et corrige ainsi l'erreur théorique. Finalement, la synchronisation logicielle entraîne, en théorie, une reconstruction incomplète de l'objet mais en pratique, le flou de bougé, inhérent au temps d'obturation, moyenne cette erreur et la reconstruction reste de bonne qualité.

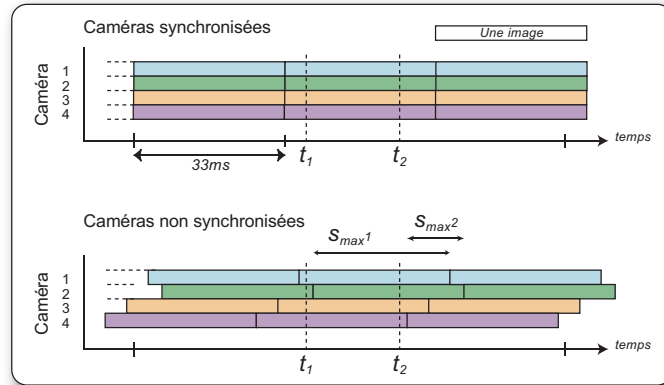


Figure 17: Meilleur et pire cas de synchronisation. Soit D_i le délai entre t_k et la dernière image reçue depuis la caméra i ($0 \leq D_i < 33$ ms). Le décalage $S_{i,j}$ entre les caméras i et j est $||D_i - D_j||$. Dans le cas synchronisé, toutes les silhouettes sont reçues en même temps : $S_{i,j} = 0$ ms quel que soit t_k . Dans le cas non-synchrone, les images sont décalées. À t_1 le décalage maximum est $S_{max1} = S_{1,2} \sim 30$ ms, à t_2 le décalage maximum $S_{max} = S_{2,4} \ll 33$ ms : S_{maxk} est donc toujours inférieur à 33 ms.

7 Parallélisation

La modélisation géométrique étant un processus coûteux en temps et l’affichage devant s’effectuer à 25 images par seconde minimum, nous avons été amené à travailler sur la parallélisation des traitements. Au début de nos travaux, nous disposions de PC peu performants et d’une SGI Onyx (voir *Chapitre 2—Plate-forme CYBER, 5. Les calculateurs, page 10*). Nous connaissions les performances de l’Onyx pour l’avoir “malmené” lors de nos travaux sur la radiosité (voir *“Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer”, page 75*). Nous savions que son architecture permettait de traiter en temps réel les différents flux vidéos, d’effectuer la reconstruction du modèle et l’affichage de la scène.

La distribution des processus a été la suivante. Sur chaque PC, un processus gère l’extraction de fond et le filtrage puis transmet les silhouettes à l’Onyx via une liaisons réseau haut débit (1 Gb/s). Sur l’Onyx, un processus par PC a en charge un cache avec les différentes silhouettes datées et transmet au processus de reconstruction la dernière silhouette reçue. En pratique, nous n’avons pas constaté de perte d’image due à un traitement trop lent. Enfin, le processus principal effectue la reconstruction du

modèle, modifie la scène en fonction des interactions et s'occupe du rendu final (voir *Figure 18*).

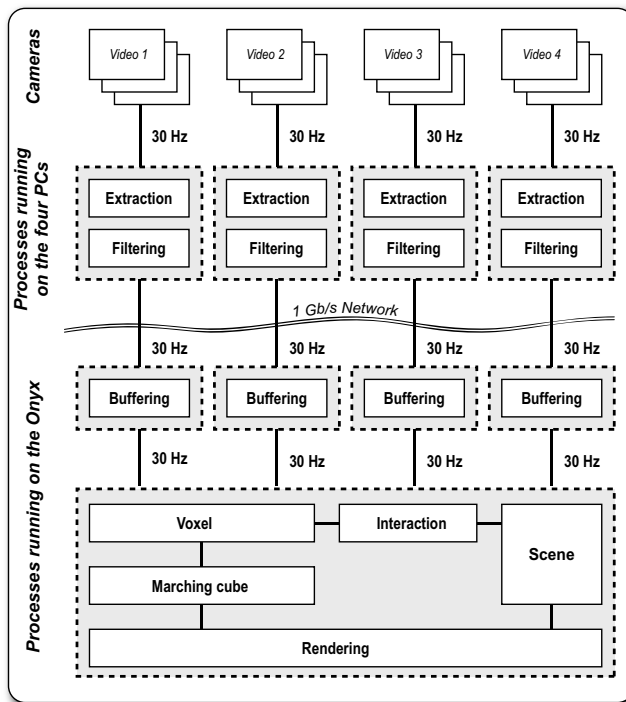


Figure 18: Répartition des différents processus.

Cette parallélisation fonctionne très bien et a permis d'obtenir tous les résultats que l'on a publiés. Cependant, la puissance de l'Onyx est limitée et son prix est prohibitif (à l'achat et à la maintenance). À l'heure actuelle, lorsque l'on a besoin de puissance de traitement, nous nous tournons plutôt vers des architectures évolutives et nettement moins onéreuses comme les grappes de PC.

8 Contributions

Nous avons commencé par travailler sur un modèle articulé à partir d'un simple flux vidéo et de marqueurs sur l'animateur (bandeaux de couleurs). Cette étude a montré les limites d'une telle approche cependant elle a permis de montrer que l'on pouvait extraire un squelette 3D à condition d'introduire des contraintes morphologiques [Fai01].

Dans [HLGB03] (voir *Chapitre 8—Articles, 3. Real-Time Capture, Reconstruction and Insertion into Virtual World of ..., page 91*), nous avons proposé une modélisation temps réel à base de voxels et de seulement quatre caméras.

Nous avons développé des algorithmes temps réel pour reconstruire un modèle non-articulé de l'animateur. En particulier, nous avons proposé une nouvelle méthode de “voxel carving” utilisant pleinement les fonctionnalités matérielles disponibles sur la plupart des cartes graphiques (voir *Chapitre 8–Articles, 7. Fast Voxel Carving Using OpenGL Facilities, page 145*).

Nous avons montré qu'une synchronisation logicielle permettait une reconstruction de qualité [HLS04] (voir *Chapitre 8–Articles, 5. A Real-Time System for Full Body Interaction, page 125*).

Enfin, notre bonne connaissance de l'architecture DSM (“*Distributed Shared Memory*”), disponible par exemple sur les SGI Onyx [SH00] (voir *Chapitre 8–Articles, 1. A Practicle Analysis of Clustering Strategies for Hierarchical Radiosity, page 59*) nous a permis de proposer une architecture logicielle efficace pour une modélisation temps réel.

9 Et maintenant ?

La précision de la reconstruction du modèle géométrique de l'animateur dépend principalement des caméras : leur résolution, leur nombre et leur position. L'utilisation de quatre caméras est un minimum ; nous arrivons à obtenir un modèle volumique correct pour un animateur seul. Par contre, si nous désirons insérer plusieurs animateurs, il sera nécessaire d'augmenter sensiblement le nombre de caméras pour diminuer les angles morts. Nous serons confrontés aux problèmes de débit réseau (voir *Chapitre 3–Acquisition des silhouettes, 4. Transmission des silhouettes, page 17*) et de positionnement des caméras (voir *Chapitre 5–Rendu réaliste, 4. Comment positionner les caméras ?, page 43*). C'est probablement ce dernier point qui sera le plus intéressant scientifiquement. Comment trouver les positions optimales (ou les moins mauvaises) des caméras ? Comment varie la précision du modèle en fonction du nombre et de la position des caméras ?

Nous avons cherché à extrapoler un squelette de l'animateur. L'approche présente un certain nombre de lacunes. Depuis, plusieurs travaux ont été publiés [TMSS02], [CTMS03] pour trouver le squelette d'un seul animateur. Mais est-il possible de reconnaître les squelettes de plusieurs animateurs ? d'un animateur jouant avec une balle ? Cet axe de recherche reste ouvert.

Rendu réaliste

1 Introduction

Le “rendu réaliste” consiste, dans notre cas, à afficher sur un périphérique graphique une scène virtuelle et un ou plusieurs éléments réels sans que l’on puisse remarquer d’incohérences. Nous voulons que l’intégration du réel dans le virtuel soit la plus convaincante possible.

Dans le monde de l’audiovisuel et des émissions de télévision en particulier, nous pouvons constater que le réalisme n’est pas encore d’actualité. Prenons pour exemple l’émission *Thalassa* diffusée par France 3 en 2003 (en 2005 rien n’a vraiment changé). En regardant l’émission, nous remarquons immédiatement que le décor n’est pas réel, que Georges Pernoud est filmé en studio sur fond bleu et qu’il est simplement incrusté sur un décor virtuel.

Etudions de plus près deux images extraites de l’émission pour comprendre ce qui “choque” lors de la diffusion (voir *Figure 1*).

- Nous avons l’impression que l’animateur flotte sur le sol. En fait, ceci est dû au manque d’ombres portées comme nous le verrons par la suite.
- Quelle que soit l’image projetée sur l’écran en arrière plan, l’aspect de l’animateur ne change pas (cela est d’autant plus visible lors de la diffusion). En effet, une image à dominante rouge projetée sur l’écran en arrière plan devrait se traduire par un rougissement de l’animateur sur son côté droit.
- A l’inverse, le teint de Georges Pernoud est trop chaud par rapport au monde virtuel bleu et donc froid dans lequel il évolue.



Figure 1: Extrait de l’émission *Thalassa* sur France 3.

- Les plans proposés par le producteur sont peu variés (flagrant lors de la diffusion). Ceci est dû au fait que pour obtenir une incrustation sans distorsion, la caméra de plateau doit être face à l'animateur.

Dans ce chapitre “Rendu réaliste”, nous allons reprendre ces différents problèmes et apporter un ensemble de solutions. Dans l'ordre, nous allons traiter les points suivants :

- Rendu réaliste dans le projet CYBER ;
- Ombres portées ;
- “Texturage” omnidirectionnel de l'animateur.

2 Rendu réaliste dans le projet CYBER

Le module “Rendu réaliste” (voir Figure 2) est celui faisant appel au plus de ressources : la géométrie de l'animateur, les flux vidéo, la géométrie de la scène...

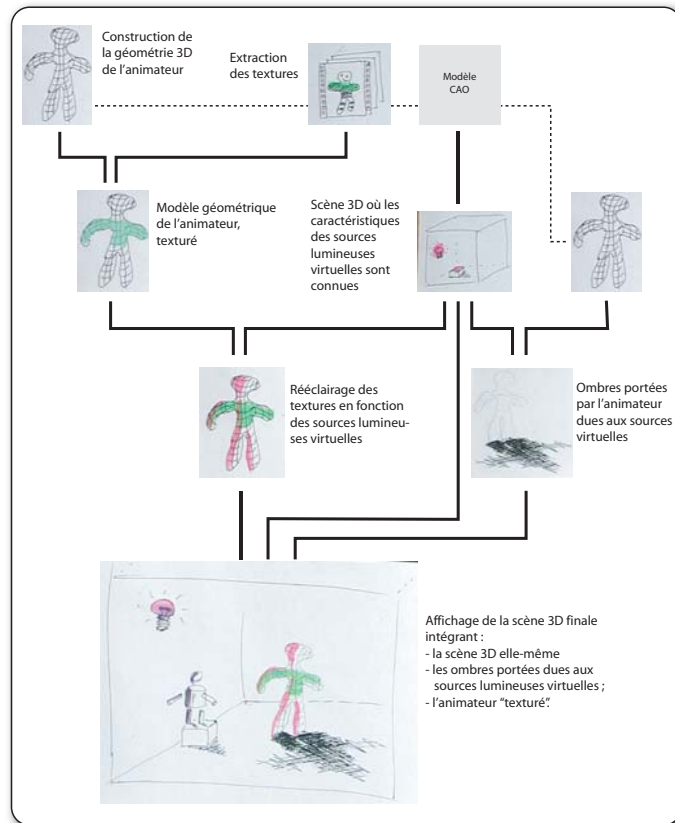


Figure 2: Module “Rendu réaliste”.

Les avancées scientifiques réalisées dans le cadre du projet CYBER doivent pouvoir être présentées à des visiteurs. Nous verrons par la suite (voir 4.3 "Comment positionner les caméras ?", page 43) que vouloir utiliser cette plate-forme à la fois pour des expérimentations scientifiques et des démonstrations impose des contraintes supplémentaires.

3 Ombres

L'animateur peut produire deux types d'ombres portées, les ombres réelles et les ombres virtuelles. Les premières sont dues aux éclairages du studio. Elles sont toujours présentes même en disposant l'éclairage de manière optimale. Elles peuvent être supprimées très facilement lorsque l'on utilise une technique de "chromakey". Si le fond n'a pas une couleur uniforme, il est nécessaire de "jouer" sur les contrastes des caméras, les positions et la puissance des éclairages, pour limiter ce type d'ombre. Le second type d'ombre provient des sources lumineuses du monde virtuel. Elles doivent être calculées et prises en compte lors du rendu final (voir Figure 3). Ceci est possible uniquement lorsque l'on dispose d'un modèle géométrique 3D. Dans un contexte de production télévisuelle, nous ne disposons généralement pas d'un tel modèle, les ombres portées ne peuvent donc pas provenir des sources lumineuses du monde virtuel (cf. l'exemple de l'émission Thalassa présenté en introduction de ce chapitre)

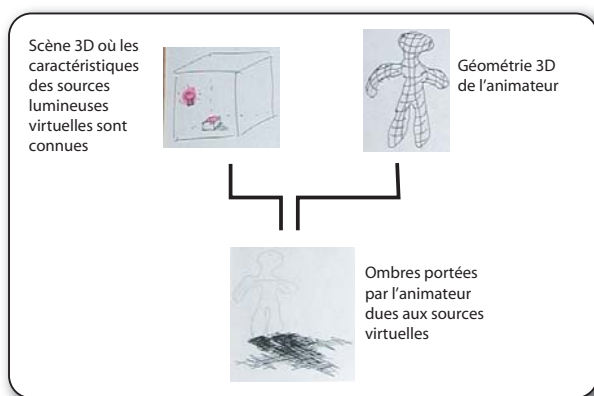


Figure 3: Les ombres de l'animateur dues aux sources appartenant au monde virtuel doivent être calculées et prises en compte dans le rendu final.

3.1 Importance des ombres

Les ombres sont étudiées depuis bien longtemps, les premières caractérisations écrites se trouvent dans le *Codex Urbinas* de Léonard de Vinci de 1490. En fait, ceci n'est pas surprenant puisque les ombres portées

sont indispensables à la compréhension d'une image. En effet, elles apportent quatre informations :

- la position (voir *Figure 4*) ;
- la forme de l'objet (voir *Figure 5*) ;
- la forme de la surface recevant l'ombre (voir *Figure 6*) ;
- l'existence d'un objet qui n'est pas visible dans l'image (voir *Figure 7*).



Figure 4: L'ombre apporte une information visuelle sur la position du robot par rapport au sol.



Figure 5: L'ombre apporte une information sur la géométrie cachée. De gauche à droite, nous pouvons remarquer, grâce aux ombres portées, que le robot tient un anneau, tient la "teapot" ou ne tient rien dans sa main gauche.



Figure 6: L'ombre portée permet de mieux comprendre la forme du sol.



Figure 7: L'ombre indique que le robot se trouve hors du champ visuel.

Une discussion plus complète sur l'importance des ombres portées est proposée dans "l'état de l'art" sur les ombres douces en temps réel que nous avons publié (voir "*A survey of Real-Time Soft Shadows Algorithms*", paragraphe "1. Introduction", page 104).

3.2 Ombres douces

Produire des ombres douces (voir *Figure 8*) est un sujet traité depuis longtemps en synthèse d'image. Par contre, le fait d'ajouter la contrainte temps réel est relativement récent, pour preuve le «grand» nombre de publications sur le sujet à la conférence *Siggraph 2003*. Cette effervescence est en fait directement liée aux possibilités de programmation des cartes graphiques de dernières générations.

Nous avons présenté un “état de l’art” sur les ombres douces temps réel lors de la conférence Eurographics en 2003 [HLHS03]. Cette présentation a été complétée par les publications présentées à cette même conférence pour donner lieu à une publication exhaustive sur le sujet [HLHS03b] (voir “*A survey of Real-Time Soft Shadows Algorithms*”, page 101).

Dans cette publication, nous avons entre autres expliqué comment sont produites les ombres douces, pourquoi elles sont importantes et les effets que peuvent produire plusieurs sources lumineuses de couleurs différentes. Nous avons mis en avant un certain nombre de problèmes généralement “oubliés” dans les algorithmes d’ombres douces temps réel (voir “*A survey of Real-Time Soft Shadows Algorithms*”, paragraphe “2.4 Important issues in computing soft shadows”, page 106)¹. Nous reprendrons quelques uns de ces problèmes.

Lorsqu’une source lumineuse est beaucoup plus grande que l’objet éclairé, son ombre peut être quasiment douce (voir *Figure 9*) et presque aucune zone d’ombre dure n’est produite. Cet effet est rarement obtenu avec les algorithmes temps réel parce que la plupart du temps, ils considèrent le centre de la source pour produire les régions de pénombre et/ou considèrent uniquement les polygones les plus proches de la source lumineuse.



Figure 9: Lorsque la source lumineuse est sensiblement plus grande que l’objet qu’elle éclaire, il n’est plus possible de considérer son centre pour essayer de produire une ombre douce réaliste.

1. Un certain nombre de pages Internet ont été écrites pour servir de support à des cours, elles contiennent en particulier des images et des vidéos montrant l’importance des ombres – <http://artis.imag.fr/Research/RealTimeShadows/>



Figure 8: Ombres dures et douces produites par trois sources de tailles différentes.

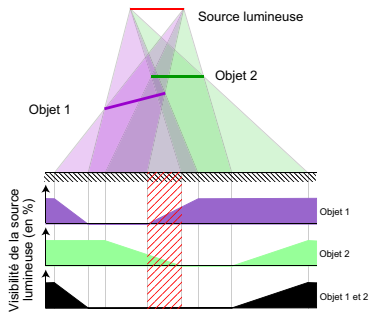


Figure 10: L'ombre de deux objets n'est pas une simple combinaison des deux ombres prises séparément.

Le calcul de l'ombre de deux objets n'est pas une simple combinaison des deux ombres prises séparément (voir *Figure 10*). Ceci impose donc de traiter tous les objets d'une scène en même temps.

Dans cet "état de l'art", nous avons proposé une classification des différentes approches en fonction de leur rapidité, de leur qualité visuelle, du type de sources lumineuses qu'elles considèrent, du type de scènes utilisables et de la nécessité de disposer de fonctionnalités matérielles particulières [HLHS03b] (voir "*A survey of Real-Time Soft Shadows Algorithms*", *Table 1*, *page 119*).

Nous avons travaillé sur d'autres approches pour produire des ombres douces temps réel qui prennent en compte les problèmes présentés ci-dessus. Nous avons soumis à publication une nouvelle approche pour produire ces ombres douces en utilisant massivement les possibilités de programmation des dernières générations de cartes graphiques (voir "*Soft Shadow Maps: Efficient Sampling of Light Source Visibility*", *page 153*). Cet algorithme discrétise les surfaces des objets portant des ombres pour calculer la proportion de lumière retenue par chaque élément de surface. En pratique, la complexité de cette approche est en $O(n)$ où n est le nombre de polygones des objets portant des ombres douces (voir *Figure 11*).

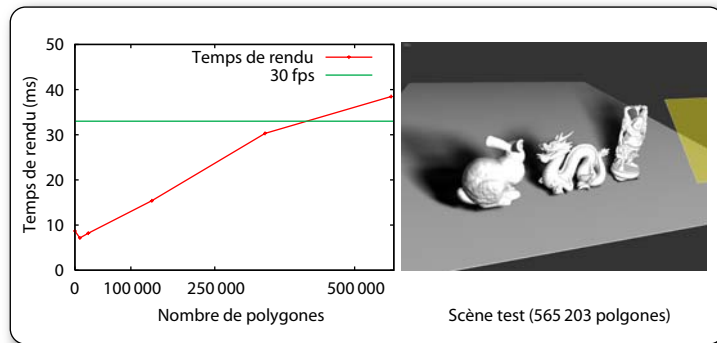


Figure 11: Influence du nombre de polygones sur la vitesse du rendu.

De plus, cet algorithme permet d'obtenir des ombres douces réalistes même pour des sources lumineuses plus grandes que les objets éclairés (voir *Figure 12*). Ceci grâce à un traitement en deux passes : la première

traite les polygones orientés vers la source lumineuse et la deuxième traite les autres polygones portant des ombres (voir *Figure 12*).

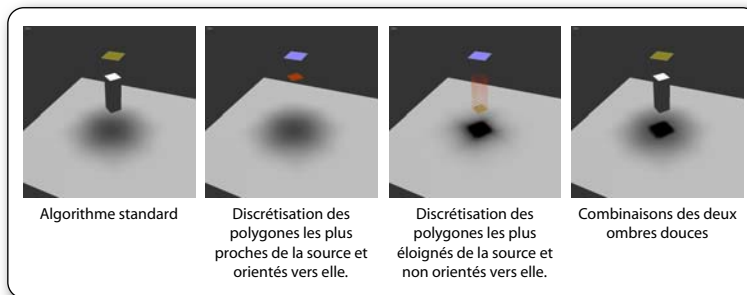


Figure 12: Les algorithmes standards ne fonctionnent pas toujours (à gauche). Notre approche en deux “passes” produit des ombres réalistes (à droite).

4 “Texturage” omnidirectionnel

Pour que le rendu soit réaliste, il faut que l’on retrouve les couleurs et les textures réelles de l’animateur dans le monde virtuel. En effet, il n’est pas envisageable d’afficher la simple reconstruction géométrique sans l’habiller. Nous allons donc utiliser les flux vidéo provenant des caméras ayant servis dans la phase de modélisation (voir *Figure 13*) pour “texturer”¹ la géométrie de l’animateur. De plus, pour que l’on puisse se déplacer tout autour du modèle et le voir sous tous les angles, l’habillage doit être complet, c’est-à-dire omnidirectionnel. Nous utiliserons dans cette partie les modèles et textures acquis dans le cadre du projet GrImage [GrImage].



Figure 13: Position de six caméras servant à la construction du modèle 3D et au “texturage”.

L’étape de “texturage omnidirectionnel” (voir *Figure 14*) étant liée aux données produites lors de l’acquisition, nous allons rappeler les hypothèses de travail :

- La géométrie de l’animateur est disponible sous la forme d’un ensemble de polygones, mais nous ne disposons pas forcément des informations d’adjacence des faces.
- Les caméras sont calibrées et nous connaissons leurs positions (à quelques centimètres près). Elles sont synchronisées (de manière matérielle ou logicielle).

1. Dans la suite du chapitre, je prendrai la liberté de ne plus utiliser “texturage” (ou “texture”) pour ne pas alourdir la lecture. Lorsqu’il n’y aura pas de confusion possible, j’utiliserai le terme *habiller*.

- Pour chaque caméra, nous disposons d'un flux d'images couleurs correspondant à l'intérieur de la silhouette.

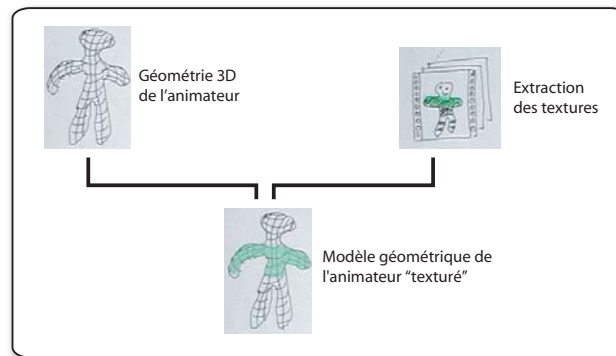


Figure 14: Module "Texturage omnidirectionnel". À partir du modèle géométrique reconstruit et des flux vidéos nous "habillons" l'animateur.

Les travaux sur l'habillage du modèle géométrique ont débuté grâce à deux stages de DEA en 2003 et 2004. Ces travaux ont permis de faire un état des lieux des approches existantes et d'en implémenter un certain nombre. Nous avons ainsi pu mettre en évidence les problèmes à résoudre :

- Comment assurer une bonne jonction entre les textures provenant de caméras différentes ?
- Comment positionner et choisir les caméras ?
- Comment traiter les problèmes de visibilité ?
- Comment conserver une cohérence temporelle des couleurs ?
- Comment tenir compte de l'éclairage provenant du monde virtuel ?

Ces travaux de recherche étant en cours, nous n'avons pas encore le recul suffisant pour proposer la meilleure approche. Par contre, nous pouvons faire le point sur les problèmes présentés ci-dessus et proposer des solutions ou des perspectives de recherche. Nos premiers résultats seront publiés en septembre 2005 [OH05] (voir "Omnidirectional texturing of human actors from multiple view video seq.", page 137).



Figure 15: La jonction entre des textures provenant de caméras différentes peut être visible. Ici, nous pouvons observer un «ségrage» noir dû à ce problème.

4.1 Jonction entre les textures

Pour habiller l'ensemble du modèle 3D, nous utilisons les différents flux vidéo à disposition. Deux polygones partageant une même arête peuvent, en théorie, être texturés avec des images provenant de caméras différentes. D'une part, la jonction entre les images utilisées peut alors être visible et nuire au réalisme (voir Figure 15). D'autre part, le modèle poly-

gonal peut être composé de triangles très allongés (voir *Figure 16*) ce qui augmente ces artefacts.

Plusieurs approches ont déjà été proposées dans la littérature [CTMS03], [DYB98], principalement en "mélangeant" les couleurs des polygones partageant une même arête (voir "*Omnidirectional texturing of human actors from multiple view video seq.*", paragraphe "2.2 Multi-view texture mapping", page 147). Le principal problème est de définir correctement le "mélange". En effet, nous pouvons faire une simple moyenne entre les textures des différentes caméras. Cette solution ne donne pas de bons résultats en particulier lorsque l'on superpose trois textures ou plus. En effet, le mélange d'un trop grand nombre de textures introduit du flou dans l'image résultante. Il faut donc attribuer judicieusement des poids à chacune des textures.

4.2 Problèmes de visibilité

Le choix des caméras utilisées pour texturer l'avatar est relativement complexe. En effet, pour un polygone donné, nous pourrions utiliser la caméra la "plus orthogonale" au polygone. Malheureusement, cette approche simpliste ne fonctionne pas. En effet, si un autre polygone se situe entre la caméra choisie et le polygone à texturer, ce dernier ne sera pas entièrement visible depuis la caméra mais sera néanmoins texturé avec ce qu'elle voit (voir *Figure 17*). Le choix des caméras servant à habiller l'avatar doit donc obligatoirement prendre en compte les problèmes de visibilité (voir *Figure 18*). La plate-forme d'acquisition est actuellement équipée de six caméras, à terme, nous devrions en disposer d'une dizaine, voire plus. Une augmentation du nombre de caméras permet d'augmenter le nombre de point de vue et donc d'améliorer la couverture visuelle de l'animateur. On peut ainsi espérer filmer l'animateur sous "toutes ses coutures".

4.3 Comment positionner les caméras ?

Nous avons vu qu'avec peu de caméras, et à condition de les positionner judicieusement, nous pouvions obtenir une reconstruction satisfaisante de l'animateur (voir *Chapitre 4—Modélisation, 2. Modélisation dans le projet CYBER, page 24*). Par contre, ce faible nombre de caméras ne convient pas à l'habillage du modèle. En effet, certaines régions de l'animateur sont "à l'ombre" de toutes les caméras, nous ne disposons alors d'aucune information de texturage. Nous avons étudié le nombre de caméras visibles depuis la surface du modèle en travaillant avec une configuration utilisant six caméras (voir *Figure 19*). La conclusion est simple, il faut beaucoup plus de caméras pour couvrir tous les "angles morts" dus aux mouvements de l'animateur. Dans notre exemple, le simple fait de



Figure 16: Modèle géométrique de l'animateur. Certains polygones sont très allongés.

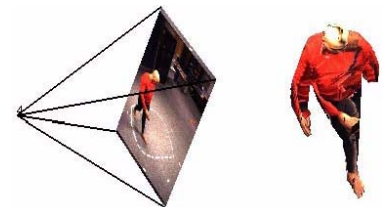


Figure 17: Problème de visibilité, la caméra utilisée (à gauche sur la figure) pour texturer le bras droit de l'avatar est la même que celle utilisée pour texturer la jambe droite. Il en résulte une projection erronée du bras sur la cuisse de l'avatar.



Figure 18: Autre exemple où un objet (le damier) se trouve entre la caméra et les polygones à texturer (le visage).

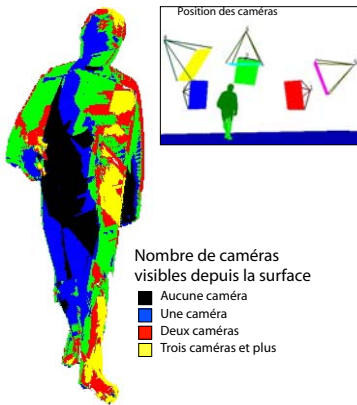


Figure 19: Nombre de caméras visibles depuis la surface du modèle.

lever le bras droit cache le flanc droit (régions noires voir Figure 19) de l’animateur de toutes les caméras, nous ne pouvons donc plus le texturer.

Il est donc nécessaire d’augmenter sensiblement le nombre de ces caméras. Le problème est alors de les positionner au mieux en fonction de ce que l’on veut filmer : un ou plusieurs animateurs avec ou sans objets supplémentaires (comme des balles), etc.

D’autres contraintes sont à prendre en compte : la position de l’écran sur lequel est projetée la scène et la position du public (chercheurs, visiteurs...). En effet, si l’écran de projection est dans le champ d’une des caméras, nous ne pourrons plus appliquer l’algorithme d’extraction sur fond statique quelconque. Ainsi, il est nécessaire de bien orienter la caméra 4 de notre exemple (voir Figure 20) pour ne pas risquer de filmer l’écran. La solution est de placer la caméra en hauteur et de la faire pointer vers le bas à la limite de l’écran, mais nous risquons d’avoir des angles morts.

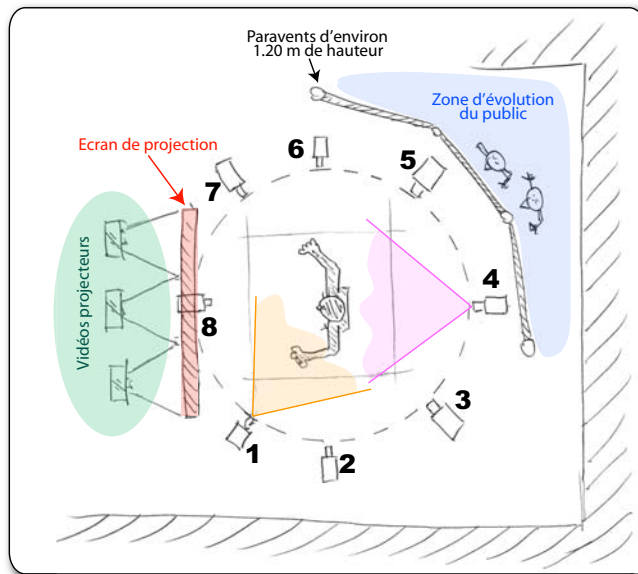


Figure 20: Positions des caméras. La caméra 1 risque de filmer le public, la caméra 4 risque de filmer l’écran de projection.

Le problème est le même pour tous les objets ou personnes en mouvement. Ainsi, du public pourrait nuire à la bonne extraction de fond. La caméra 1 de notre exemple (voir Figure 20) risque de filmer ce public. Une solution est alors d’utiliser des paravents pour augmenter les surfaces statiques du fond tout en autorisant des personnes à circuler à proximité de la zone de capture (voir Figure 20).

4.4 Cohérence des couleurs dans le temps

L'animateur étant en mouvement, les silhouettes vues depuis chaque caméra changent à chaque image, la géométrie de l'avatar n'est donc jamais la même. En particulier, le nombre et la forme des triangles qui composent l'avatar varient en permanence. Ceci introduit des discontinuités temporelles. L'effet le plus visible est une variation de couleur de certains polygones due à un changement de caméra utilisée pour leurs "texturages" (voir *Figure 21*).

Les polygones ayant une durée de vie d'une seule image, il n'est pas possible de les suivre dans le temps et il n'est donc pas possible de se fonder sur une numérotation des polygones. Ceci reviendrait à utiliser une approche dans "l'espace objet". La solution que nous étudions actuellement se place dans "l'espace image". Nous cherchons à suivre dans le temps la couleur des pixels affichés. L'idée est alors de pondérer la couleur des pixels à l'instant t en fonction des couleurs de ces mêmes pixels à l'instant $t-1, t-2...t-n$. Ainsi, nous pouvons espérer éviter les sauts de couleur. Bien sûr, la pondération sera un choix crucial pour le bon fonctionnement du processus. Notons que tous les changements brusques de couleur d'un pixel ne doivent pas être lissés, ils peuvent correspondre à un mouvement réel de l'animateur. On peut donc envisager d'utiliser des pondérations dynamiques dans le temps en fonction de critères comme des seuils, des mouvements brusques de la caméra, etc.

Une approche que nous voulons étudier, est le suivi des points dans le temps pour prédire leurs positions. Nous pourrions nous inspirer de [VBK05] même si l'approche décrite utilise un modèle voxelique et n'est pas temps réel. Une autre technique est présentée dans [TCMS03]. Cette solution aurait l'avantage de traiter les discontinuités temporelles mais aussi, dans une certaine mesure, le manque d'information dans les angles morts. En effet, par extrapolation, nous pouvons envisager de combler les vides avec les couleurs prédites.

5 Contributions

La première contribution a été de mettre en évidence ce qui "choque" dans les émissions télévisuelles utilisant des studios virtuels (voir 1. "Introduction", page 35). À partir de ces constats, nous avons principalement travaillé sur le point le plus critique, à savoir les ombres portées. Nous avons commencé par publier un état de l'art exhaustif des algorithmes temps réel d'ombres douces [HLHS03], [HLHS03b] (voir "A survey of Real-Time Soft Shadows Algorithms", page 101). Nous y avons proposé une classification des algorithmes pour faciliter le choix de l'approche à utiliser en fonction des besoins du programmeur.



Figure 21: Deux images prises en tournant très légèrement la caméra. Nous constatons un changement net des textures.

Forts de cette expérience et des problèmes scientifiques que nous en avons dégagés, nous avons proposé une nouvelle approche pour des ombres douces temps réel en discrétisant les objets pour évaluer la proportion de lumière retenue par chaque élément de surface (voir *“Soft Shadow Maps: Efficient Sampling of Light Source Visibility”*, page 153).

Enfin, nous travaillons actuellement sur le texturage omnidirectionnel de la géométrie 3D de l’animateur. Nous avons mis en évidence un certain nombre de points sur lesquels nous concentrer. Les premiers résultats ont fait l’objet d’une publication [OH05] (voir *“Omnidirectional texturing of human actors from multiple view video seq.”*, page 137).

6 Et maintenant ?

6.1 Rééclairage

Pour obtenir un rendu réaliste, les éléments insérés dans le monde virtuel (dans notre cas, un animateur) ne doivent pas ressortir, ils doivent parfaitement s’intégrer. Dans notre cas, c’est l’apparence de notre animateur qui doit être prise en compte ainsi que les modifications d’éclairage qu’il peut entraîner sur le reste de la scène. En effet, lorsqu’il passe à proximité d’une source lumineuse, il doit bien sûr projeter une ombre mais aussi changer de couleur en fonction des caractéristiques des sources lumineuses présentes. Ainsi, si une source a une dominante rouge, l’animateur doit rougir. C’est ce que l’on va appeler le “rééclairage” (voir *Figure 22*).

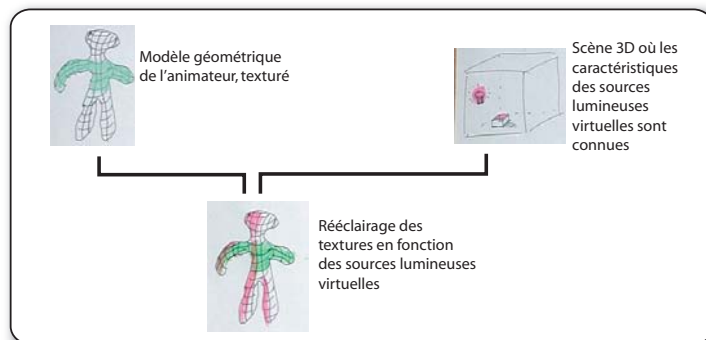


Figure 22: Module de “Rééclairage”. Nous utilisons le modèle géométrique ainsi que les caractéristiques des sources lumineuses du monde virtuel pour “rééclairer” l’animateur.

Nous pouvons distinguer deux types d’échanges lumineux à simuler dans nos mondes virtuels : les éclairages directs ou locaux et les éclairages indirects ou globaux.

Éclairage local

Les *effets locaux* sont produits directement par les sources de lumière : éclairage direct, ombres portées et auto-ombrage. Nous avons principalement travaillé sur l'aspect "ombre douce" qui est le plus critique. Il reste néanmoins encore beaucoup de travail à faire. Nous continuons à explorer de nouvelles approches aussi bien dans "l'espace image" que dans "l'espace objet". Parmi les problèmes à résoudre, le plus difficile est probablement de traiter les "auto-ombrages" (ombre d'un objet portée sur lui-même) d'une manière physiquement exacte (voir *Figure 23*). En effet, nous savons produire des effets d'auto-ombrage plausibles [Assa03] mais pas réel, c'est-à-dire ne respectant que très peu les lois optiques.

Les sources lumineuses généralement utilisées en synthèse d'images sont de simples quadrilatères avec une distribution énergétique homogène. Cependant, ce type de sources est une violente simplification. En effet, nous trouvons dans le monde réel des éclairages "statiques" comme des néons, des spots... mais aussi des éclairages "dynamiques" comme un feu de cheminée, un ciel nuageux... Ces sources peuvent être géométriquement complexes (voir *Figure 24*) ou ne pas être "modélisables" sous la forme d'un ensemble de polygones (comme des nuages), il faut donc trouver une modélisation adéquate. Ces modèles peuvent correspondre à une description énergétique ou à une géométrie très simple simulant le même éclairage. Pour proposer de nouveaux modèles, nous pouvons nous appuyer sur nos travaux dans le domaine de la radiosité et en particulier sur le "clustering" [HDSD99], [SH00] (voir "*A Practicle Analysis of Clustering Strategies for Hierarchical Radiosity*", page 59).

Éclairage global

Les *effets globaux* proviennent d'échanges subtils entre l'environnement et l'animateur plongé dans le monde virtuel. Par exemple, lorsqu'un objet est éclairé, il réfléchit une partie non négligeable de cette lumière, modifiée par sa matière, vers les objets qui l'entourent. L'apparence de ces objets est alors modifiée par cette lumière colorée. C'est ce que l'on appelle communément l'éclairage indirect. Il est très important pour ajouter du réalisme dans la scène [SP93], [TL04] (voir *Figure 25*). Pour tenir compte de ces échanges globaux, nous pouvons là encore nous appuyer sur nos connaissances en matière de radiosité [HDSD99], [SH00] (voir "*A Practicle Analysis of Clustering Strategies for Hierarchical Radiosity*", page 59).

6.2 Qualité des images

Nous utilisons actuellement des caméras de résolution modérée qui ne sont pas appropriées à un rendu sur un mur d'images par exemple. Il nous faut donc augmenter la qualité des images. Nous pouvons nous tourner vers des solutions comme celles proposées par [WJVT+05] en utilisant un ensemble de 128 caméras mais ce système est très coûteux et peu envisa-

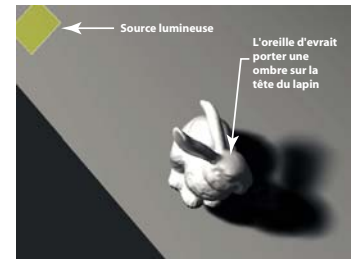


Figure 23: L'auto-ombrage n'est pas pris en compte dans cet exemple.

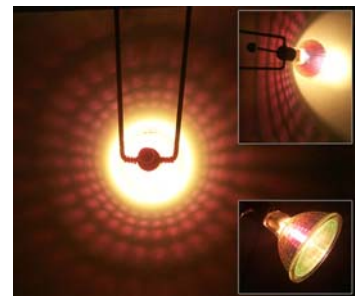


Figure 24: Exemple de lampe de bureau que l'on veut modéliser et utiliser dans le monde virtuel.

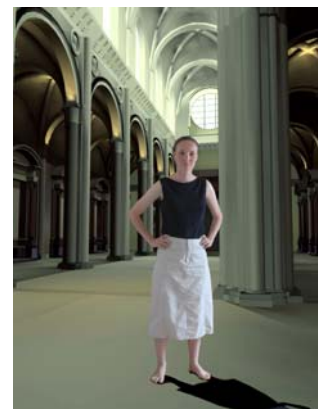


Figure 25: L'ambiance de l'environnement virtuel n'est pas prise en compte lors de l'incrustation ce qui rend l'insertion choquante (photomontage)

geable dans notre contexte. Je préfère imaginer une solution à base de blocs de quelques caméras (deux ou quatre) pour augmenter la qualité des images (voir *Chapitre 3–Acquisition des silhouettes, 6. Et maintenant ?, page 20*).

6.3 Rendu “non photoréaliste”

Nous avons présenté en introduction le “rendu réaliste” comme l’affichage d’une scène virtuelle et d’éléments réels sans que l’on puisse remarquer d’incohérences. En fait, la frontière avec le rendu “non photoréaliste” est très proche. En particulier, nous pouvons imaginer sans difficulté intégrer notre modèle 3D de l’animateur dans une scène qui aurait un rendu dans un style donné, comme de l’aquarelle, du dessin à la plume, etc. (voir *Figure 26*). Mais c’est une recherche à long terme puisqu’il faut tout d’abord résoudre le problème de navigation dans des modèles rendus de manière non photoréaliste (même si des travaux sont en cours pour aller dans ce sens [KDMF03], [DFR04]).



© wang-wang@vip.sina.com



© HLM Design International Ltd



© Carlos Marrero, Fort Myers, Florida USA



© Javier Sandoval, Construsol Digital Services, Mexico

Figure 26: Exemple de personnages adoptant le même rendu non-photoréaliste que le monde virtuel dans lequel il évolue. (Ces images proviennent de la “Piranesi Gallery” et ont été créées à l’aide du logiciel “Piranesi” <http://www.informatix.co.uk/piranesi/>)

Interaction avec le monde virtuel

1 Introduction

Dans ce chapitre, l'**interaction avec le monde virtuel** concerne uniquement la modification de la géométrie de la scène ou du modèle reconstruit. En particulier, les interactions de type “échanges lumineux” ne sont pas discutées ici, elles font partie du chapitre précédent (voir *Chapitre 5—Rendu réaliste, 6.1 Rééclairage, page 46*).

L'interaction (géométrique) a été traitée dans nos recherches à des fins plutôt ludiques. En effet, ma spécialité étant le “rendu d'images de synthèse” plutôt que “l'interaction homme/machine”, nous avons exploré très “naïvement” les possibilités de notre plate-forme. Cependant, les interactions entre l'animateur et le monde virtuel dans lequel il est plongé contribuent très largement à la sensation d'immersion. Il est donc important de développer les recherches dans ce sens.

Nous allons donc reprendre les hypothèses dans le cadre du projet CYBER, étudier les problèmes de latence et proposer quelques exemples d'interaction avec le monde virtuel que nous avons développés.

2 L'interaction dans le projet CYBER

Dans le projet CYBER, la contrainte a toujours été le temps réel afin de pouvoir interagir avec le monde virtuel. Nous avons aussi interdit tout capteur intrusif qui pourrait gêner l'animateur dans ses mouvements. À terme, nous devrions reconnaître le squelette de l'animateur et utiliser cette structure pour effectuer des interactions (voir *Figure 1*). À l'heure actuelle nous ne disposons que d'une géométrie sous forme de polygones ou de voxels sans information morphologique. Nous ne sommes donc

pas capables de distinguer un pied d'une main ! Les interactions sont donc limitées.

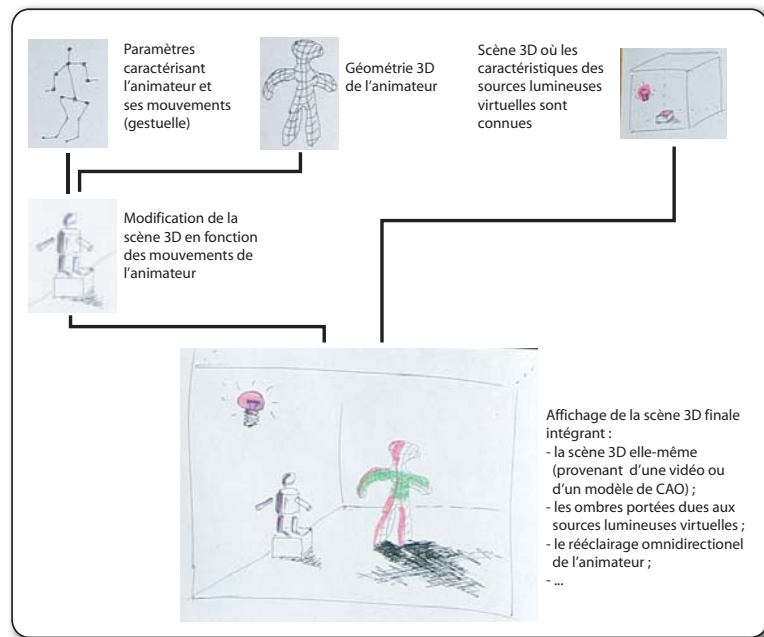


Figure 1: Module "Interaction" dans le projet CYBER.

3 Temps réel et latence

Pour que l'animateur puisse interagir avec le monde virtuel, il est indispensable que l'ensemble des processus soit exécuté en temps réel. Dans notre implémentation [HLS04] (voir "*A Real-Time System for Full Body Interaction*", page 125), nous arrivons à traiter l'ensemble des informations : acquisition, construction du modèle, rendu et interaction en moins de 30 ms (voir Figure 2).

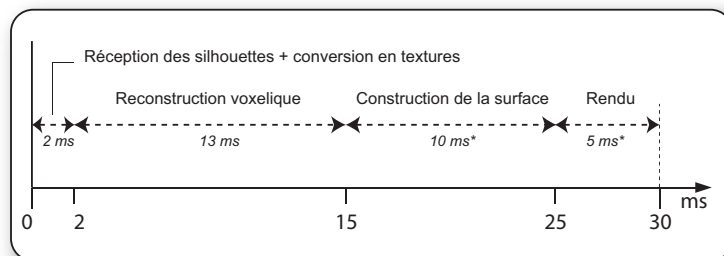


Figure 2: Temps passé dans chaque processus. (*) Le temps de construction du modèle et son rendu dépendent de la complexité de sa surface – les chiffres proposés sont des bornes supérieures.

La latence du système est un point crucial de l'interaction. En effet, si l'animateur bouge son bras et qu'il faut quelques secondes avant de voir le mouvement dans le monde virtuel, aucune interaction ne sera possible. Nous avons donc optimisé les transmissions des informations pour obtenir une latence constante d'environ 4 images, c'est-à-dire 0.12 s (voir "*A Real-Time System for Full Body Interaction*", paragraphe "5. Interaction", page 132).

4 Exemple d'interaction

Nous nous sommes fondés sur le modèle voxelique pour effectuer nos interactions. Le principe est très simple, il suffit qu'un certain nombre de voxels de la grille soient occupés par l'animateur pour déclencher une action à cette position. Nous avons développé plusieurs interactions, cependant l'apport scientifique étant faible, nous n'en présenterons que deux [HLS04] (voir "*A Real-Time System for Full Body Interaction*", page 125).

"Slider" virtuel

L'interaction la plus évoluée consiste à simuler un "slider" virtuel. Plus le "slider" est haut et plus le nombre de balles tombant est important (voir Figure 3). Le jeu est alors de les faire rebondir sur une raquette (réelle) ou n'importe quelle partie du corps (voir Figure 4).

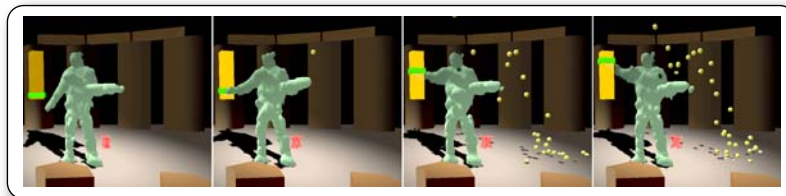


Figure 3: Manipulation d'un "slider" virtuel pour régler le flux de balles.

Effets spéciaux

Le deuxième type d'interaction est basique, il s'agit de toucher un objet de la scène (ici un 'x' rouge) pour provoquer une "explosion" de l'animateur en utilisant un système de particules (voir Figure 5).

5 Contributions

Nous avons proposé une chaîne complète, temps réel avec une latence très faible permettant de traiter l'acquisition vidéo, d'effectuer la construction du modèle, de calculer des ombres portées et d'interagir avec des éléments du monde virtuel [HLGB03], [HLS04] (voir *Chapitre 8—Articles*,



Figure 4: L'animateur fait rebondir les balles virtuelles sur sa raquette réelle.

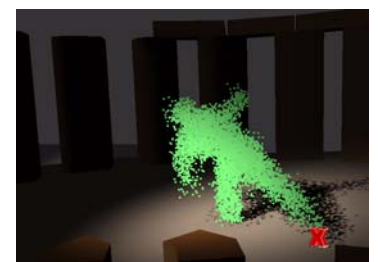


Figure 5: Explosion de l'animateur en utilisant un système de particules.

3. Real-Time Capture, Reconstruction and Insertion into Virtual World of ..., page 91).

Nous avons présenté un certain nombre d'interactions pour lesquelles il n'est pas nécessaire de connaître la morphologie de l'animateur [HLS04] (voir *Chapitre 8–Articles, 5. A Real-Time System for Full Body Interaction*, page 125)

6 Et maintenant ?

Pour développer l'interaction avec le monde virtuel, il sera nécessaire de connaître la morphologie de l'animateur, probablement en retrouvant son squelette [TMSS02], [CTMS03]. Nous pourrions ainsi connaître la position des mains et envisager des interactions plus fines. Cependant, ceci risque d'être difficile si nous voulons conserver la possibilité de travailler sans connaissance *a priori* des objets et/ou animateurs que l'on désire capturer et insérer dans le monde virtuel.

Conclusion

Durant ces dernières années et au travers des deux ACI CYBER et CYBER-II, j'ai travaillé sur plusieurs domaines. En premier lieu, les besoins matériels de notre plate-forme d'acquisition ont dû être établis de sorte qu'elle réponde à nos contraintes et qu'elle soit pérenne. Cette élaboration, puis la construction de cette salle dédiée, m'ont permis de me familiariser avec les éclairages de studio et les caméras numériques. Les autres domaines sur lesquels j'ai orienté ma recherche correspondent davantage à des problèmes scientifiques. Il s'agit, en effet, d'acquisition, de modélisation, de rendu réaliste et d'interaction. Je vais reprendre à présent mes contributions puis présenter mes axes de recherche à venir.

1 Contributions

L'**acquisition de silhouettes** a été le premier sujet de recherche. Nous avons pris le parti de ne pas travailler avec des fonds de couleur uniformes mais plutôt avec des fonds quelconques. Ces recherches ont rempli nos attentes puisque nous sommes capables d'extraire les silhouettes de l'animateur en temps réel. Il y a bien sûr quelques limitations pratiques. Il ne faut pas que l'animateur porte des habits dont la couleur se confond avec celle du fond de la salle. De plus, l'éclairage doit limiter au maximum les ombres portées (surtout au niveau des pieds) pour ne pas entraîner des erreurs de reconstruction. Nous avons discuté de la précision nécessaire lors de l'acquisition des silhouettes pour obtenir un modèle satisfaisant de l'animateur.

Nous avons abordé la **modélisation** sous deux angles : la création d'un modèle articulé et celle d'un modèle non-articulé. Le premier a permis de montrer clairement les limites lorsque l'on ne dispose que d'une seule caméra. En effet, des résultats temps réel peuvent être obtenus mais

notre approche utilisant des bandeaux de couleurs ne permet pas d'obtenir une reconnaissance fiable à tout moment. Concernant la modélisation non-articulée, nous avons proposé un certain nombre d'algorithmes temps réel permettant de produire un modèle volumique sous forme de voxels ou un modèle surfacique sous forme d'un ensemble de polygones. Les deux approches ont chacune leurs avantages et inconvénients. Dans les deux cas, le faible nombre de caméras utilisées s'avère être une limitation à la précision de la reconstruction. C'est pourquoi, nous proposons d'augmenter à la fois le nombre de caméras ainsi que leur résolution.

Le travail sur le **rendu réaliste** temps réel, nous a permis de publier un état de l'art des algorithmes produisant des ombres douces en temps réel. À partir de là, nous avons proposé une nouvelle approche temps réel, "basée image" et physiquement plus proche de la réalité. Nous travaillons sur l'habillage de la géométrie de l'animateur pour augmenter le réalisme. C'est actuellement notre principal sujet de recherche.

Nous avons élaboré une chaîne complète de traitements depuis l'acquisition jusqu'au rendu réaliste en **temps réel**. Il ne restait plus qu'à travailler sur l'**interaction** de l'animateur avec le monde virtuel. C'est ce que nous avons fait au travers de démonstrations très simples d'un "slider" virtuel, de jeux de balles ou d'effets spéciaux. Ces interactions utilisant un modèle voxelique, ont montré la faisabilité d'interagir avec le monde virtuel. Cependant, il faut bien reconnaître que tant que nous ne serons pas capables de reconnaître la morphologie de l'animateur, les interactions resteront limitées.

2 Et maintenant ?

Ces travaux ont apporté un certain nombre de réponses mais ont suscité des interrogations ouvrant la porte à de nouveaux axes de recherches. Ainsi, parmi les points sur lesquels je souhaite poursuivre mes recherches, nous trouvons le texturage omnidirectionnel de l'animateur. Ce sujet a été largement développé dans le chapitre sur le "*Rendu réaliste*". J'ajouterai simplement qu'il n'est peut-être pas nécessaire de passer par un modèle géométrique explicite pour effectuer ce rendu réaliste. Une approche utilisant les particularités des nouvelles cartes graphiques semble envisageable.

Toujours dans ce chapitre de "*Rendu réaliste*", nous avons présenté les effets de l'éclairage global et les interactions lumineuses animateur/monde virtuel. Le problème reste que ce type d'interactions lumineuses est très complexe et très difficile à réaliser en temps réel. Néanmoins, nous pouvons espérer qu'avec une modélisation adéquate des sources lumineuses, nous pourrions obtenir de bons résultats.

Enfin, si nous voulons interagir “physiquement” avec le monde virtuel, il nous faudra obligatoirement reconnaître les éléments reconstruits, que ce soit un ou plusieurs animateurs avec un ou plusieurs objets. Ce champ de recherche est certainement le plus vaste.

Articles

Liste des publications depuis 1999 et ayant un rapport avec le sujet traité dans ce mémoire.

- [HDSD99] Jean-Marc Hasenfratz, Cyrille Damez, François Sillion, George Drettakis, “*A Practical Analysis of Clustering Strategies for Hierarchical Radiosity*”, Computer Graphics Forum (Proc. of Eurographics '99), 18(3), pp. 221-232, Sep 1999

Voir page 59

- [SH00] François Sillion, Jean-Marc Hasenfratz, “*Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer*”, Third Eurographics Workshop on Parallel Graphics and Visualisation, Girona - Sep 2000, pp 61-74

Voir page 75

- [HLGB03] Jean-Marc Hasenfratz, Marc Lapierre, Jean-Dominique Gascuel, Edmond Boyer, “*Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors*”, Vision, Video and Graphics, pp. 49-56, 2003

Voir page 91

- [HLHS03] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, François Sillion, “*A survey of Real-Time Soft Shadows Algorithms*”, Computer Graphics Forum, 22(4), pp 753-774, Dec. 2003

Voir page 101

[HLS04] Jean-Marc Hasenfratz, Marc Lapierre, François Sillion, “*A Real-Time System for Full Body Interaction with Virtual Worlds*”, Eurographics Symposium on Virtual Environments, pp. 147-156, 2004

Voir page 125

[OH05] Alexandrina Orzan, Jean-Marc Hasenfratz, “*Omnidirectional texturing of human actors from multiple view video sequences*”, Proceedings of the Conference on Computer-Human Interaction, Sept. 2005

Voir page 137

[HLS05] Jean-Marc Hasenfratz, Marc Lapierre, François Sillion, “*Fast Voxel Carving Using OpenGL Facilities*”, **Rejected** to Journal of Graphics Tools, (?), 2005

Voir page 145

[ALHH+05] Lionel Atty, Marc Lapierre, Nicolas Holzschuch, Chuck Hansen, J-Marc Hasenfratz, Francois Sillion , “*Soft Shadow Maps: Efficient Sampling of Light Source Visibility*”, **Rejected** to Eurographics Symposium on Rendering (2005)

Voir page 153

1 A Practicle Analysis of Clustering Strategies for Hierarchical Radiosity

[HDS99] Jean-Marc Hasenfratz, Cyrille Damez, François Sillion, George Drettakis,
“*A Practicle Analysis of Clustering Strategies for Hierarchical Radiosity*”,
Computer Graphics Forum (Proc. of Eurographics '99), 18(3), pp. 221-232, Sep 1999

2 Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer

- [SH00] François Sillion, Jean-Marc Hasenfratz, “Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer”, Third Eurographics Workshop on Parallel Graphics and Visualisation, Girona - Sep 2000, pp 61-74

3 Real-Time Capture, Reconstruction and Insertion into Virtual World of ...

[HLGB03] Jean-Marc Hasenfratz, Marc Lapierre, Jean-Dominique Gascuel, Edmond Boyer, “Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors”, *Vision, Video and Graphics*, pp. 49-56, 2003

4 A survey of Real-Time Soft Shadows Algorithms

- [HLHS03] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, François Sillion, “A survey of Real-Time Soft Shadows Algorithms”, *Computer Graphics Forum*, 22(4), pp 753-774, Dec. 2003

5 A Real-Time System for Full Body Interaction

- [HLS04] Jean-Marc Hasenfratz, Marc Lapierre, François Sillion, “A Real-Time System for Full Body Interaction”, Eurographics Virtual Environments, pp. 147-156, 2004

6 Omnidirectional texturing of human actors from multiple view video seq.

- [OH05] Alexandrina Orzan, Jean-Marc Hasenfratz, “Omnidirectional texturing of human actors from multiple view video sequences”, Proceedings of the Conference on Computer-Human Interaction, Sept. 2005

7 Fast Voxel Carving Using OpenGL Facilities

- [HLS05] Jean-Marc Hasenfratz, Marc Lapierre, François Sillion, “Fast Voxel Carving Using OpenGL Facilities”, **Rejected** to Journal of Graphics Tools, (?), 2005

8 Soft Shadow Maps: Efficient Sampling of Light Source Visibility

[ALHHHS05]Lionel Atty, Marc Lapierre, Nicolas Holzschuch, Chuck Hansen, J-Marc Hasenfratz, Francois Sillion , “Soft Shadow Maps: Efficient Sampling of Light Source Visibility”, **Rejected** to Eurographics Symposium on Rendering (2005)

Bibliographie

- [ABFM+04] Jérémie Allard, Edmond Boyer, Jean-Sébastien Franco, Clément Ménier, Bruno Raffin, “*Marker-less Real Time 3D Modeling for Virtual Reality*”, Immersive Projection Technology (IPT'2004), May 2004.
- [Assa03] Ulf Assarsson, “*A Real-Time Soft Shadow Volume Algorithm*”, PhD thesis, Department of Computer Engineering, Chalmers University of Technology, October 2003.
- [BF03] Edmond Boyer, Jean-Sebastien Franco, “*A Hybrid Approach for Computing Visual Hulls of Complex Objects*”, Computer Vision and Pattern Recognition (CVPR'03), Volume. 1, pp. 695–701, June 2003
- [BL99] F. van den Bergh and V. Lalioti, “*Software Chroma Keying in an Immersive Virtual Environment*”, South African Computer Journal, (24), pp. 155–162, Nov. 1999.
- [CTMS03] Joel Carranza, Christian Theobalt, Marcus Magnor, Hans Peter Seidel, “*Free viewpoint video of human actors*”, Proceedings of ACM SIGGRAPH'03, 22(3), pp.569–577, 2003.
- [CG99] R. Cipolla and P.J. Giblin, “*Visual Motion of Curves and Surfaces*”, Cambridge University Press, 1999.
- [CKBH00] German K.M. Cheung, Takeo Kanade, Jean-Yves Bouguet, Mark Holler, “*A Real Time System for Robust 3D Voxel Reconstruction of Human Motions*”, Computer Vision and Pattern Recognition (CVPR'00), Vol. 2, pp. 714–720, 2000.
-

- [CPP03] Rita Cucchiara, Massimo Piccardi, Andrea Prati, , “*Detecting Moving Objects, Ghosts, and Shadows in Video Streams*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10), pp. 1337–1342, Oct. 2003.
- [CTMS03] Joel Carranza, Christian Theobalt, Marcus Magnor, Hans-Peter Seidel, “*Free-viewpoint video of human actors*”, ACM Trans. on Computer Graphics, 22(3), pp. 569–577, July 2003.
- [DFR04] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, “*Interactive Rendering of Suggestive Contours with Temporal Coherence*”, NPAR 2004, pp. 15–24, 2004.
- [DR95] I. Debled-Rennesson, J.-P. Réveilles, “*A linear algorithm for segmentation of digital curves*”, International Journal of Pattern Recognition and Artificial Intelligence, 9(6), pp. 635–662, Dec. 1995.
- [DYB98] Paul E. Debevec, Yizhou Yu, George D. Borshukov, “*Efficient view-dependent image-based rendering with projective texture-mapping*”, Eurographics Workshop on Rendering, pp. 105–116, 1998.
- [Dye01] Charles Dyer, “*Volumetric scene reconstruction from multiple views*”, In L. S. Davis, editor, Foundations of Image Understanding, pp. 469–489. Kluwer, 2001.
- [Fai01] Sylvain Faisan, “*Incrustation temps réel d’un personnage dans un monde virtuel*”, mémoire de stage de 3^{ième} année, diplôme d’ingénieur de l’ENSPS, 2001.
- [FB03] Jean-Sebastien Franco, Edmond Boyer, “*Exact Polyhedral Visual Hulls*”, British Machine Vision Conference (BMVC’03), Vol. 1, pp. 329–338, September 2003.
- [FMBR04] Jean-Sebastien Franco, Clement Menier, Edmond Boyer, Bruno Raffin, “*A Distributed Approach for Real-Time 3D Modeling*”, CVPR Workshop on Real-Time 3D Sensors and their Applications, Washington DC, July 2004.
- [GrImage] *GrImage (Grid and Image)*, GrImage is funded by INRIA and an Allocation Spécifique from the Ministère délégué à la Recherche et aux Nouvelles Technologies (via INPG), <http://www.inrialpes.fr/sed/grimage/>

-
- [HLHS03] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, François Sillion, “*A survey of Real-Time Soft Shadows Algorithms*”, STAR Eurographics, pp. 1–20, 2003.
- [HLHS03b] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, François Sillion, “*A survey of Real-Time Soft Shadows Algorithms*”, Computer Graphics Forum, 22(4), pp. 753–774, 2003.
- [HLGB03] Jean-Marc Hasenfratz, Marc Lapierre, Jean-Dominique Gascuel, Edmond Boyer, “*Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors*”, Vision, Video and Graphics, pp. 49–56, 2003.
- [HLS04] Jean-Marc Hasenfratz, Marc Lapierre, François Sillion, “*A Real-Time System for Full Body Interaction with Virtual Environments*”, Eurographics Symposium on Virtual Environments, pp. 147–156, 2004.
- [HDSD99] Jean-Marc Hasenfratz, Cyrille Damez, François Sillion, George Drettakis, “*A Practical Analysis of Clustering Strategies for Hierarchical Radiosity*”, Computer Graphics Forum (Proc. of Eurographics '99), 18(3), pp. 221–232, 1999
- [HM04] Bertrand Holveck, Hervé Mathieu, “*Infrastructure of the GrImage experimental platform: the video acquisition part*”, INRIA RT-0301, 50 pages, Nov. 2004.
- [KDMF03] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, Adam Finkelstein, “*Coherent stylized silhouettes*”, Proceedings of ACM SIGGRAPH 2003, 22(3), pp. 856–861, 2003.
- [KS00] K. Kutulakos and S. Seitz, “*A Theory of Shape by Space Carving*”, International Journal of Computer Vision, 38(3), pp. 199–218, 2000.
- [Kut00] K. Kutulakos, “*Approximate N-View Stereo*”, Proceedings European Conference on Computer Vision, pp. 67–83, 2000.
- [Lau94] A. Laurentini, “*Visual hull concept for silhouette-based image understanding*”, IEEE Trans. on PAMI, 16(2), pp. 150–162, 1994.

- [Lok01] Benjamin Lok, “*Online model reconstruction for interactive virtual environments*”, Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 69–72, ACM Press, 2001.
- [MBM01] Wojciech Matusik, Chris Buehler, and Leonard McMillan, “*Polyhedral visual hulls for real-time rendering*”, Eurographics Workshop on Rendering, pp. 115–126, 2001.
- [OH05] Alexandrina Orzan, Jean-Marc Hasenfratz, “*Omnidirectional texturing of human actors from multiple view video sequences*”, Conference on Computer-Human Interaction, 2005.
- [OpenCV] “*Open Computer Vision Library*”, <http://sourceforge.net/projects/opencvlibrary/>
- [Par02] David Parrish, “*Inspired 3d Lighting & Compositing*”, Muska & Lipman Publishing, ISBN 1931841497, 2002.
- [SB96] Alvy Ray Smith and James F. Blinn, “*Blue screen matting*”, In Proc. of the 23rd Annual Conf. on Computer Graphics and Interactive Techniques, ACM Press, New York, USA, pp. 259–268, 1996.
- [SCMS01] A. Slabaugh, B. Culbertson, T. Malzbender, and R. Scafe, “*A survey of methods for volumetric scene reconstruction from photographs*”, International Workshop on Volume Graphics, pp. 81–100, 2001.
- [SH00] François Sillion, Jean-Marc Hasenfratz, “*Efficient Parallel Refinement for Hierarchical Radiosity on a DSM computer*”, Third Eurographics Workshop on Parallel Graphics and Visualisation, Girona, pp. 61–74, Sep 2000.
- [SP93] François Sillion, Claude Puech, “*Radiosity and Global Illumination*”, Morgan Kaufmann Publishers, 1994.
- [TCMS03] Christian Theobalt, Joel Carranza, Marcus Magnor, Hans Peter Seidel, “*Enhancing Silhouette-Based Human Motion Capture with 3D Motion Fields*”, Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, pp. 185–193, 2003.

-
- [TMSS02] Christian Theobalt, Marcus Magnor, Pascal Schüler, Hans Peter Seidel, “*Combining 2D Feature Tracking and Volume Reconstruction for Online Video-Based Human Motion Capture*”, Proceedings of Pacific Graphics, pp. 96–103, 2002.
- [TKBM99] Kentaro Toyama, John Krumm, Barry Brumitt, Brian Meyers, “*Wallflower: Principles and Practice of Background Maintenance*”, ICCV, pp. 255–261, 1999.
- [TL04] Eric Tabellion, Arnauld Lamorlette, “*An Approximate Global Illumination System for Computer Generated Films*”, Proceedings of Siggraph 2004, 23(3), pp. 469–476, August 2004.
- [Tsa86] Roger Y. Tsai, “*An Efficient and Accurate Camera Calibration Technique for 3d Machine Vision*”, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 364–374, 1986.
- [VBK05] Sundar Vedula, Simon Baker, Takeo Kanade, “*Image-based spatio-temporal modeling and view interpolation of dynamic events*”, ACM Transactions on Graphics (TOG), 24(2), pp. 240–261, April 2005.
- [WFSM02] Dongsheng Wang, Tao Fengz, Heung-Yeung, Shumz Songde May, “*A Novel Probability Model for Background Maintenance and Subtraction*”, 15th International Conference on Vision Interface (ICVI’02), pp. 109–117, 2002.
- [WJVT+05] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville (Eddy) Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Marc Levoy, Mark Horowitz, “*High Performance Imaging Using Large Camera Arrays*”, Proceeding of SIGGRAPH 2005.
- [YPLL04] Tao Yang, Quan Pan, Stan Z. Li, Jing Li, “*Multiple Layer Based Background Maintenance in Complex Environment*”, Third International Conference on Image and Graphics (ICIG’04), pp. 112–115, 2004.
- [Zha00] Zhengyou Zhang, “*A Flexible New Technique for Camera Calibration*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 22(11), pp.1330–1334, 2000.

