

# Rendu réaliste et prototypage virtuel interactif

## Rendering and interactive virtual prototyping

par Mathias PAULIN\*, Olivier BALET\*\*, Cyril KARDASSEVITCH\*, Mathieu ROBERT\*, Jean-Pierre JESSEL\*, René CAUBET\*

\* I.R.I.T., Université Paul Sabatier, 118, Route de Narbonne 31062 Toulouse Cedex, France e-mail : paulin@irit.fr

\*\* CS-CISI, 13, Rue Villet, Z.I. du Palays 31029 Toulouse, France e-mail : Olivier.Balet@cisi.cnes.fr

### *résumé et mots clés*

L'application des techniques de réalité virtuelle à l'architecture permet de diminuer les coûts de conception du projet par prototypage virtuel. Afin de rendre le prototype développé exploitable pour la prise de décision, une simulation convenable de l'éclairage doit être effectuée. Le besoin important d'interactivité dans la manipulation du prototype, soit pour des déplacements de l'utilisateur dans le monde virtuel, soit pour des modifications géométriques apportées à ce monde, nécessite un éclairage très rapide de la scène. Actuellement, les modèles d'éclairage utilisés, dans ce type d'application, sont des approximations empiriques des modèles physiques dont le seul avantage est la rapidité d'évaluation. Les textures sont largement utilisées pour augmenter le réalisme des scènes virtuelles mais le niveau de simulation de l'éclairage reste grossier.

Le modèle de radiosit , mod le physique fond  sur l' valuation des transferts radiatifs dans une sc ne, permet de d terminer, de fa on ind pendante du point de vue, la valeur du signal lumineux en tout point de la sc ne, en tenant compte de toute la sc ne (illumination globale). Le principal inconv nient de ce mod le  tant le temps de calcul n cessaire   sa r solution, son utilisation dans une application de prototypage passe par la d finition de techniques de r solution adapt es. Nous proposons dans cet article une plate-forme logicielle, organis e autour de deux noyaux de simulation - *Visualisation-Interaction* et *Simulation de l' clairage* -, permettant d'utiliser le mod le de radiosit  lors de la conception par prototypage virtuel d'environnements architecturaux. Nous pr sentons plus particuli rement la conception et l'implantation d'un module de simulation de l' clairage, appel  * clairagiste*, dont l'objectif est d'assurer le r alisme maximum   chaque instant. Le module  clairagiste, d fini comme une entit  de l'environnement de programmation d'applications de r alit  virtuelle VIPER, utilise une mod lisation multi- chelle de la radiosit  et repose sur une m thode de r solution d riv e de la radiosit  hi rarchique.

Radiosit , r alit  virtuelle, prototypage virtuel coop ratif, interaction 3D temps r el.

### *abstract and key words*

Application of virtual reality techniques to architectural design allows to decrease the conception cost by virtual prototyping. In order to make the prototype usable for the decision process, an accurate lighting simulation must be done. The important need of interactivity in the manipulation of the prototype, for virtual walk-through or for modifying the prototype, implies a time efficient lighting simulation in the virtual world. Generally, lighting simulation used in this type of application used empirical models whose only advantage is the real-time simulation. Texture mapping is a technique widely used to increase the quality of the simulation but without reaching the needed accuracy for architectural design.

The radiosity model, based on the radiative transfers theory, allows to determine the lighting of a scene in a view-independent way. This model's limitation is its complexity. Also, its use in a virtual prototyping application needs the definition of adapted evaluation method. In this paper we propose a software environment, organised around two simulation kernels - *Visualisation-interaction* and *Lighting simulation* - that allows the use of the radiosity model during the design of an architectural environment. We present more precisely the design and the implementation of the lighting simulation module, called *Eclairagiste*, which objective is to ensure the maximum lighting accuracy at each instant. The *Eclairagiste* module, defined as an entity of the virtual reality programming environment VIPER, uses a multi-resolution representation of the radiosity function and relies on resolution method derived from hierarchical radiosity.

Radiosity, virtual reality, collaborative virtual prototyping, 3D real time interaction.

# 1. introduction

De nos jours, l'utilisation de maquettes physiques fait partie intégrante des projets d'ingénierie à grande échelle dans de nombreux secteurs d'activités tels que les industries aérospatiales, automobiles ou du bâtiment. Ces maquettes restent bien souvent l'unique moyen pour identifier et analyser de nombreux problèmes dès les phases d'avant-projet. Pour les décideurs et les ingénieurs, cette phase de prototypage constitue une étape inévitable pour confirmer et valider visuellement leurs idées et leurs concepts, les choix faits à ce niveau d'avancement du projet pouvant par la suite altérer considérablement le résultat final ainsi qu'être la cause de pertes de temps et d'argent.

Cependant, le coût de fabrication des prototypes physiques ainsi que leur manque de flexibilité constituent deux inconvénients majeurs qui ont nécessité l'étude et la réalisation de solutions informatiques pour le prototypage de maquettes numériques. Ce type d'opération est de nos jours réalisé à l'aide de logiciels de Conception Assistée par Ordinateur (CATIA, EUCLID, Pro-Engineer) dont les capacités restent encore trop limitées pour permettre une navigation et une manipulation interactive des objets modélisés. Ces outils ont pour premier objectif de fournir une description extrêmement précise des objets et des scènes modélisés : c'est le modèle généré par ces logiciels qui reste la référence tout au long du processus de création d'un produit, depuis sa conception jusqu'à sa fabrication. C'est pour cette raison que ces outils sont mal, voire pas du tout adaptés au rendu et à l'interaction temps réel, la précision des modèles produits rendant impossible (ou très difficile) leur manipulation directe dans un système de réalité virtuelle. Il faut alors, avec de tels systèmes, beaucoup de temps et d'imagination pour déceler les erreurs de design. Il est de plus nécessaire de disposer d'une grande expertise pour pouvoir les utiliser. Or les décideurs et les concepteurs, intervenant dans les choix stratégiques et techniques en phase d'avant-projet, ne disposent généralement pas d'une telle expertise; ce sont pourtant leurs décisions qui orienteront le projet.

C'est pourquoi de grands projets d'ingénierie ont souvent nécessité le développement spécifique d'outils de prototypage virtuel interactifs [Ellis96] utilisant les techniques d'interaction 3D. Nous pouvons citer, par exemple, le ISS VR Demonstrator utilisé par Rolls-Royce pour vérifier le concept de prototypage et de maintenance de moteurs virtuels [Greenfield96], le système FlyThru conçu par Boeing pour le design du 777 [Abarnabel96] ou encore le système i3d développé conjointement par le CERN et le CRS4<sup>1</sup> pour l'aide à la conception et à la construction du Large Hadron Collider du CERN [Balaguer96].

Le projet de recherche PROVIS (PROtotype VIRTuel de Systèmes) [Torguet97] [Balet97], lancé en 1995 par le Centre National d'Etudes Spatiales et réalisé par CS-CISI avec la collaboration de l'IRIT pour les aspects distribués, étudie de nouvelles

solutions pour simplifier le travail des décideurs et des concepteurs de systèmes qui, dès les phases d'avant-projet, doivent disposer d'outils leur permettant d'appréhender et de vérifier intuitivement la pertinence de leurs choix (encombrement des structures, accessibilité des composants, etc.).

L'objectif principal de ce projet est d'étudier les techniques d'interaction homme-machine 3D appropriées pour une utilisation du système pouvant durer plusieurs heures. Nous avons pour cela choisi une interface matérielle (figure 1) du type *desktop VR* à laquelle nous avons adjoint un système de suivi de la tête de l'utilisateur, intégré aux lunettes de visualisation stéréoscopique, afin de pouvoir générer les effets de parallaxe. Ce type de configuration permet de travailler longtemps sans engendrer de fatigue musculaire notable; seule la vision stéréoscopique prolongée peut entraîner une gêne visuelle à la longue.

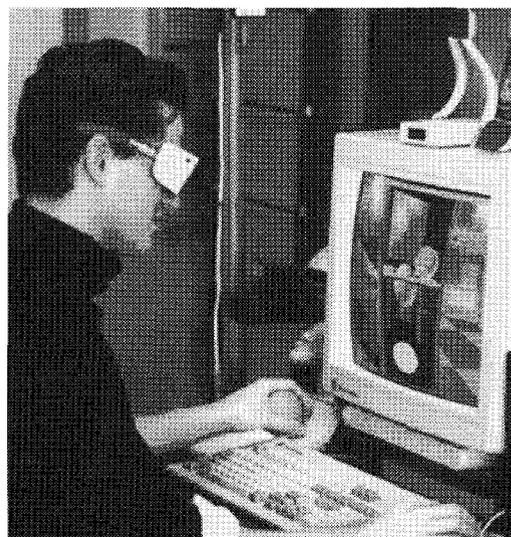


Figure 1. – Interface homme-machine.

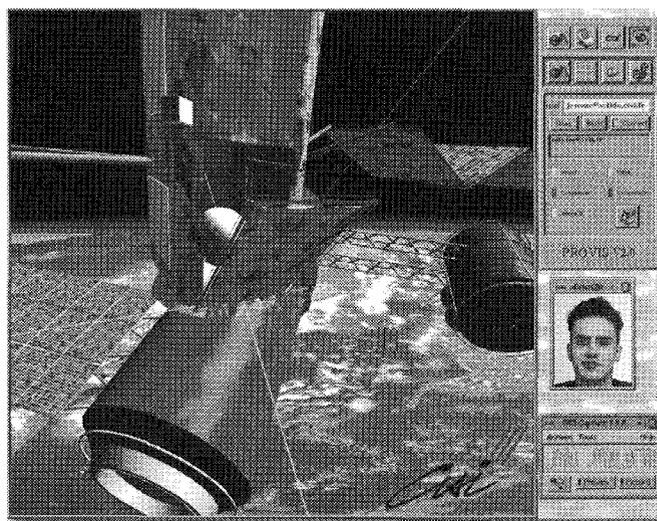


Figure 2. – Exemple de session PROVIS.

1. Centro di Ricerca, Sviluppo e Studi Superiori in Sardegna.

Une première liste de fonctionnalités fut spécifiée avec les utilisateurs finaux du département mini-satellites. Notre système devait ainsi pouvoir permettre :

- la gestion d'une bibliothèque de prototypes et de composants de satellites;
- l'assemblage interactif de ces composants à l'aide de liaisons mécaniques;
- la manipulation interactive, à des fins de validation, des mécanismes ainsi construits;
- la validation du choix des composants en termes de consommation électrique ou de poids;
- la gestion de la documentation multimédia en ligne associée aux composants.

Étant donnée la dispersion géographique des équipes impliquées dans les projets du CNES ainsi que la difficulté de communication entre les différents corps de métier, nous proposons une solution basée sur une architecture distribuée afin de pouvoir supporter le prototypage virtuel coopératif synchrone, la maquette virtuelle servant de support de communication. L'idée sous-jacente à cette proposition était de fournir, dans le même temps, un système qui puisse servir de support aux processus d'ingénierie concurrente adoptés par de nombreux industriels.

PROVIS a tout d'abord été conçu pour le prototypage interactif de satellites. À l'utilisation, la proposition d'appliquer les mécanismes mis en œuvre à de nombreux autres domaines industriels fut émise. Nous présentons, dans cet article, la configuration PROVIS destinée au milieu de l'architecture et du bâtiment. Ces domaines d'activités ont la particularité de nécessiter un rendu graphique très réaliste basé sur des modèles d'éclairage incompatibles avec les techniques d'interaction 3D en temps réel, ces dernières s'appuyant généralement sur un moteur de rendu intégrant une approximation grossière du modèle physique de l'éclairage. Au début des années 80, l'introduction des méthodes d'illumination globale en synthèse d'images, telles que le lancer de rayons [Wittd80] ou la méthode des radiosité [Goral84], a permis un saut qualitatif sur le plan de l'éclairage pour le rendu d'environnements réalistes. Cependant, malgré de nombreux travaux concernant l'optimisation de ces méthodes [Fujimoto86] [Cohen86] [Hanrahan91] [Smits92] [Sillion94], le temps de calcul nécessaire à la résolution de ces modèles reste trop important pour une simulation interactive de l'éclairage. La réponse apportée à l'accroissement de la complexité des modèles d'éclairage consiste en l'utilisation de transferts lumineux multi-échelle entre groupes abstraits de surfaces [Smits94], en l'utilisation de textures pour représenter la luminance [Segal92] [Soler98], ou en des méthodes orientées vers des environnements dynamiques [Shaw97] [Drettakis97], propres à faciliter l'interactivité et à rapprocher radiosité et animation. Gagnant en rapidité, ces méthodes dégradent un peu la qualité photométrique de la simulation de l'éclairage; un contrôle quantitatif de l'erreur permet cependant d'obtenir un compromis acceptable.

Ces approches se retrouvent dans les applications interactives (promenade ou prototypage virtuel) pour accélérer le rendu et conserver le maximum d'interactivité. De manière à n'afficher que les données les plus pertinentes à un instant donné, la scène est découpée logiquement et des liens de connectivité permettent de faciliter et fluidifier la navigation. Ici aussi, pour ne pas s'encombrer de détails inutiles, les objets sont présents avec plusieurs niveaux de détail en fonction de leur importance visuelle. Toujours dans l'optique de substituer à une géométrie endémique une représentation visuellement équivalente, l'exploitation intensive de textures permet d'éviter de représenter des détails complexes ou, pour pallier les déficiences des modèles d'éclairage sommaires, de se substituer à une simulation réaliste le placage de prises de vues réelles.

Dans cet article, nous présenterons tout d'abord le système PROVIS ainsi que la plate-forme logicielle VIPER (VIRtuality Programming EnviRONment) [Torguet98] utilisée pour son développement. Nous présenterons ensuite les principes de l'illumination globale, puis notre implantation d'un modèle de radiosité orienté temps réel pour une application interactive. L'intégration des fonctionnalités de rendu réaliste dans le système PROVIS est facilitée par les similitudes conceptuelles évoquées ci-dessus et par l'adoption des mécanismes de distribution des calculs et des données proposés par VIPER. Le système final peut alors fournir à des concepteurs, par exemple des architectes, une visualisation réaliste et interactive de l'éclairage de leurs maquettes virtuelles.

## 2. architecture générale

PROVIS est construit au-dessus de la plate-forme logicielle VIPER développée par l'Institut de Recherche en Informatique de Toulouse [Torguet98]. Cette bibliothèque permet la réalisation d'applications de réalité virtuelle distribuée et repose, pour cela, sur deux concepts principaux.

Le concept d'entité [Balet96] permet de gérer de façon uniforme l'ensemble des composants qui constituent l'environnement. Il peut s'agir ici d'objets ayant une représentation graphique visible aussi bien que d'objets invisibles définissant des services de

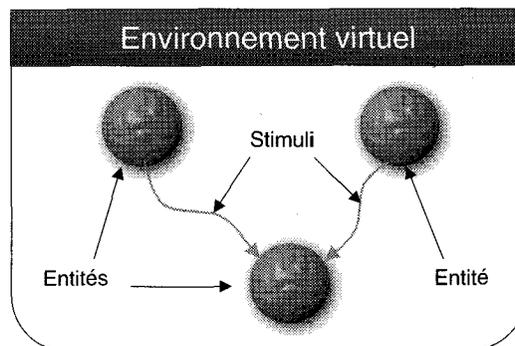


Figure 3. – Structure d'un environnement virtuel.

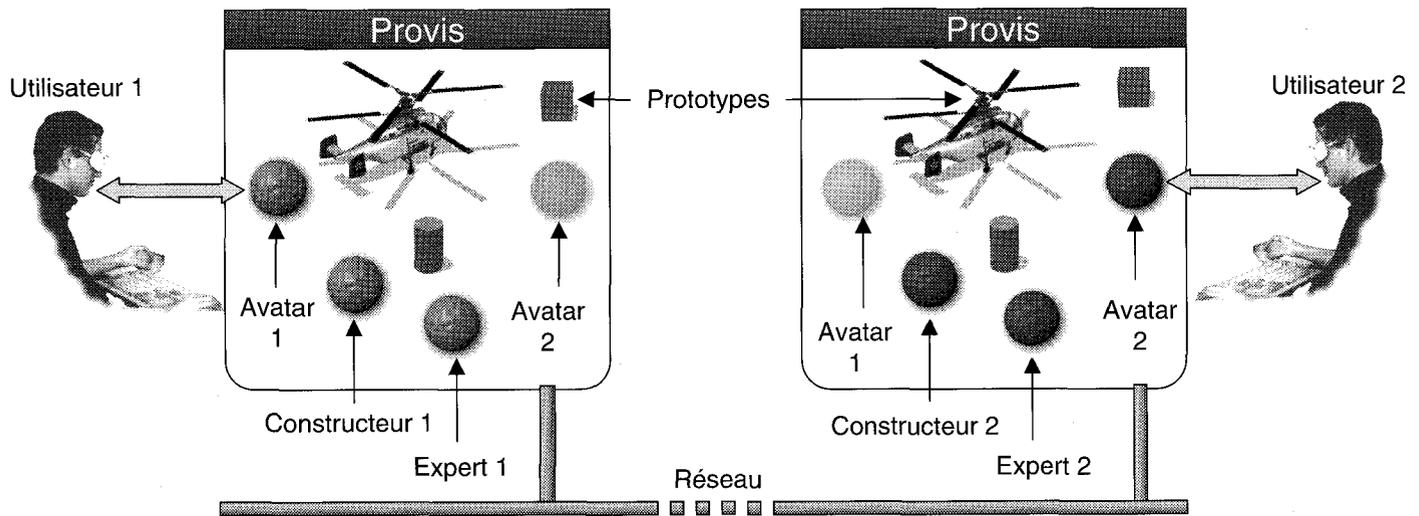


Figure 4. – Architecture générale d'une session PROVIS.

l'application développée. Parmi les objets pouvant avoir une représentation graphique, on peut citer les éléments de décors de l'environnement, les éléments ayant une activité autonome ainsi que les clones [Mouli94] qui sont les objets représentant les utilisateurs de l'application. La notion de clone, ou avatar<sup>2</sup> dans la littérature, permet de modéliser de manière homogène le plongement d'un utilisateur dans un environnement de travail tridimensionnel. Chaque entité possède un comportement plus ou moins complexe décrivant les fonctionnalités dont elle dispose ainsi que son activité propre.

Le concept de stimulus permet de modéliser les échanges de données entre entités. Les stimuli sont donc des vecteurs d'informations circulant de manière transparente au sein de l'environnement virtuel. Il existe d'ailleurs plusieurs types de stimuli correspondant chacun à un type d'information échangeable (son, image, ordre, forme, etc.). Le lecteur pourra se reporter à [Torguet98] pour les détails de l'implantation de la bibliothèque VIPER ainsi que pour une description détaillée des mécanismes proposés pour la distribution des données et la communication des stimuli au travers d'un réseau informatique.

PROVIS est défini, dans sa version de base, par quatre principaux types d'entité (figure 4), à savoir, les avatars, le Constructeur, l'Expert et les prototypes.

## 2.1. les avatars

Les entités avatars représentent les utilisateurs qui partagent la session de travail. L'une d'entre elles est ainsi le clone de l'utilisateur

2. Le mot Avatar vient du sanskrit *avatarā* (littéralement descente de Vishnu sur Terre) qui est le nom donné à chacune des réincarnations de Vishnu dans le religion hindoue.

local. La connexion d'un utilisateur au système PROVIS engendre automatiquement la création d'un tel clone défini par un nom d'utilisateur, des droits d'accès, une URL, une fenêtre de visualisation ainsi qu'une fenêtre de gestion des communications toutes locales à la station utilisée. Cet avatar comporte plusieurs modules comportementaux ainsi qu'un ensemble de capteurs et d'effecteurs lui permettant d'émettre des ordres vers d'autres entités, de visualiser l'environnement 3D et d'entrer en communication avec les autres utilisateurs. Un avatar local possède, de plus, des capteurs directement connectés aux périphériques du système.

## 2.2. le constructeur

L'entité Constructeur est en charge du dialogue naturel multimodal avec l'utilisateur, via son avatar, afin de lui fournir un ensemble de commandes de haut niveau lui permettant de gérer, manipuler et assembler les prototypes. C'est donc cette entité qui définit les fonctionnalités du système PROVIS. Une entité de type Constructeur accompagne chaque avatar localement et n'est donc pas distribuée dans l'environnement virtuel; elle n'a qu'une existence locale à la station de travail.

## 2.3. l'expert

L'entité Expert est en charge de vérifier la validité et la cohérence du prototype construit en fonction des règles métier du domaine d'application. Elle est donc supposée contenir la connaissance d'un expert de ce domaine. En pratique, nous nous sommes limités à l'implantation de fonctionnalités permettant simplement de mettre en lumière les concepts proposés.

## 2.4. les prototypes

Les entités prototypes sont en fait les éléments géométriques que manipulent les utilisateurs. On parlera indifféremment, par la suite, de composants ou d'objets pour désigner les briques de base d'un prototype. Il n'existe conceptuellement aucune différence entre un prototype et un composant. Un prototype est défini par une forme géométrique au sens VRML du terme, c'est à dire un ensemble de géométries correspondant à ses différents niveaux de détails. Ce type d'entité est référencé par une adresse URL et dispose d'un attribut documentation ainsi que d'un profil de protection utilisé pour la gestion des droits d'accès.

Comme nous l'avons précisé précédemment, un outil tel que PRO-VIS doit pouvoir fournir un rendu réaliste d'un prototype pour une utilisation dans le milieu de l'architecture et du bâtiment. Nous proposons, pour cela, la définition d'une entité supplémentaire, appelée l'éclairagiste, qui est en charge de fournir un tel rendu tout en préservant le caractère temps réel du système. Nous décrivons, dans les sections suivantes, l'organisation de cette entité fondée sur une résolution hiérarchique du modèle de radiosité.

# 3. l'entité éclairagiste

## 3.1. définition de l'entité

### 3.1.1. objectifs

L'objectif de l'entité éclairagiste est de résoudre le problème de la simulation de l'éclairage dans un environnement architectural. Pour cela un modèle d'illumination globale est mis en œuvre. Les modèles d'illumination globale, par opposition aux modèles d'illumination directe, consistent à évaluer toutes les interactions de la lumière avec les objets constituant la scène. La résolution d'un modèle d'illumination globale étant un processus coûteux en calcul, nous devons mettre en œuvre un algorithme de résolution adapté permettant de fournir, au plus vite, la solution la plus précise possible au problème de l'éclairage. Un modèle d'illumination globale utilisable est le modèle de radiosité. Le choix de ce modèle est guidé par son principal avantage : il fournit une solution associant la valeur de la radiosité aux sommets des polygones de la scène indépendamment de la position de l'observateur. Cette propriété du modèle de radiosité est très intéressante pour le domaine d'application visé : lorsque les conditions d'éclairage ne varient pas, seule la phase d'affichage de la scène doit être exécutée si l'observateur se déplace. Cependant, l'obtention d'une solution précise nécessite un très grand nombre de polygones (dus aux maillages des objets), et génère une structure grande consommatrice de ressources lors du rendu. L'utilisation de textures multi-échelle afin de représenter les niveaux de détails de la radiosité sur chaque surface permet de libérer le processeur

géométrique. D'autre part, les textures de luminance permettent d'éviter naturellement certains artefacts inhérents aux subdivisions géométriques (sommets en T, problèmes d'interpolation le long des arêtes, ...). Ainsi, la phase de résolution de la radiosité sur une surface est modifiée de façon à fournir, pour chaque surface, une texture multi-échelle de radiosité. C'est cette texture qui sera utilisée pour l'affichage de la solution en utilisant la géométrie initiale de la scène.

Lors de modifications des conditions d'éclairage, l'entité éclairagiste doit fournir, au bout d'un temps fixé par l'utilisateur, une nouvelle solution. Cette contrainte nécessite l'évaluation des régions de l'espace importantes pour l'utilisateur afin d'effectuer en priorité la résolution dans ces régions ainsi qu'un contrôle permanent de l'état d'avancement de la résolution au cours du temps. Lorsque le temps alloué à la résolution est terminé, l'éclairagiste fournit une solution partielle et en fournira d'autre à la même fréquence tant que la résolution n'est pas terminée. L'entité éclairagiste est donc totalement autonome.

### 3.1.2. connexion de l'entité avec l'environnement

La connexion de l'entité éclairagiste avec les autres entités de la simulation est réalisée par la définition d'un capteur de forme, recevant les modifications apportées à la scène et d'un effecteur de forme, permettant de fournir les textures de radiosité à appliquer sur les objets (figure 5). Les informations en entrée sont fournies sous forme d'un arbre VRML97 [VRML97] où chaque objet est nommé de façon unique.

Lors de l'initialisation de la simulation, le capteur de forme reçoit la scène décrite par l'arbre VRML complet la définissant. L'entité initialise ensuite le processus de résolution de la radiosité, soit en calculant la solution initiale, soit en la chargeant depuis un fichier. Par la suite, le capteur ne reçoit plus qu'un arbre de modification réduit contenant les nœuds subissant cette modification.

La gestion de l'effecteur est plus simple. En effet, chaque fois qu'une texture de radiosité est déterminée, elle est émise dans l'espace des stimuli à destination des objets concernés. Elle peut

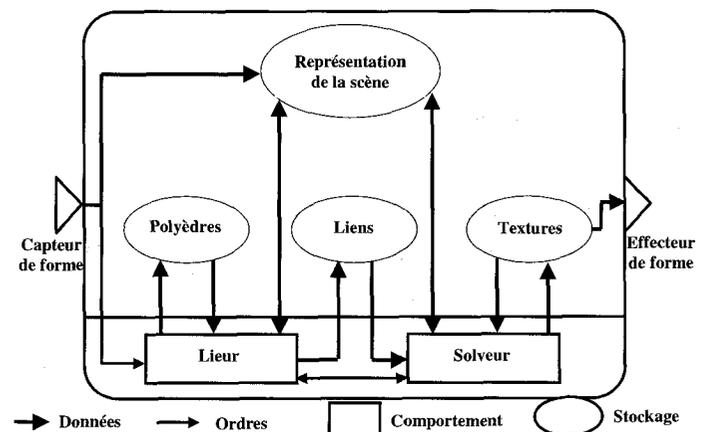


Figure 5. – Représentation schématique de l'entité éclairagiste.

ainsi être visualisée immédiatement par les avatars des utilisateurs connectés à la simulation. En pratique, afin d'améliorer les performances et de limiter le volume de données transitant entre les différentes entités, l'entité éclairagiste modifie directement la texture en mémoire texture grâce à l'utilisation du mécanisme de renommage de textures disponible en OpenGL. Ce mécanisme permet d'améliorer grandement les performances de l'affichage, les textures n'étant chargées en mémoire texture qu'une seule fois. Le volume d'informations échangées entre les entités est considérablement réduit puisque seul le nom logique de la texture est émis à destination des objets de la scène.

### 3.2. le modèle de radiosit 

#### 3.2.1. expression du mod le

Nous ne faisons ici qu'une pr sentation du mod le de radiosit  adapt e   notre objectif. Le lecteur soucieux d'obtenir une description plus d taill e du mod le de radiosit  pourra se reporter aux ouvrages de Sillion et Puech [Sillion94] pour les aspects th oriques ou de Cohen et Wallace [Cohen93] pour les aspects pratiques.

Afin de mod liser les transferts lumineux   l'int rieur d'une sc ne, Kajiyama [Kajiyama86] propose d'exprimer l'intensit  lumineuse par une  quation, connue sous le nom d' quation du rendu :

$$I(x, x') = \varepsilon(x, x') + \int_S \rho(x, x', x'')g(x, x')I(x', x'')dx'' \quad (1)$$

Le domaine d'int gration S est l'ensemble de toutes les surfaces constituant la sc ne. Les diff rents termes de cette  quation s'interpr tent de la mani re suivante :

$I(x, x')$  repr sente la luminance de  $x'$  vue de  $x$ .

$g(x, x')$  repr sente la visibilit  entre les points  $x$  et  $x'$ . Ce terme  $g$  prend comme valeurs :

$$\begin{aligned} &0 \text{ lorsque } x \text{ n'est pas visible depuis } x' \\ &\frac{1}{\|x - x'\|^2} \text{ lorsque } x \text{ est visible depuis } x'. \text{ (figure 6)} \end{aligned}$$

$\varepsilon(x, x')$  repr sente l' mission propre du point  $x'$  en direction de  $x$

$\rho(x, x', x)$  repr sente la r flectance bidirectionnelle au point  $x'$  pour les directions incidentes  $\vec{x''x'}$  et  $\vec{x'x}$

$dx$  est l' l ment de surface autour du point  $x$ .

La r solution de cette  quation int grale dans un environnement quelconque est trop co teuse pour  tre envisageable. Aussi, nous pouvons effectuer quelques hypoth ses simplificatrices sur l'environnement. Tout d'abord, si les objets de la sc ne sont lambertiens, i.e. la r flectance ne d pend pas de la direction d' clairage ni de la direction de r flexion, cette  quation donnant l'intensit  lumineuse peut  tre r ecrite en une  quation donnant la radiance  mise (radiosit ) en tout point de la sc ne. La radiosit 

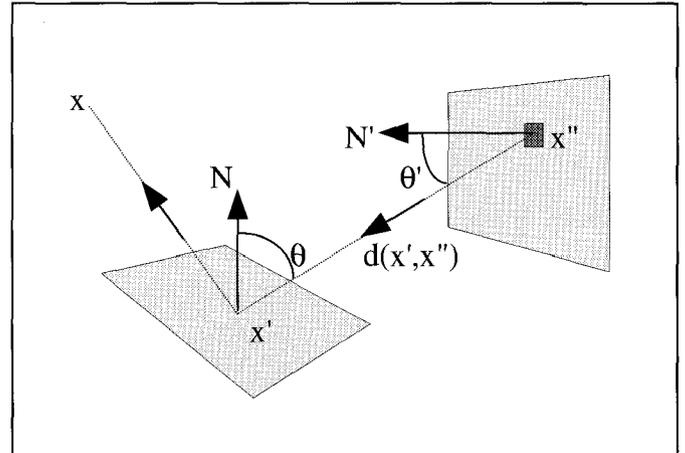


Figure 6. - G om trie pour l' quation du rendu.

en tout point d'un environnement diffus est donc donn e par l' quation int grale suivante :

$$L(x) = L_e(x) + \int_S K(x, x')L(x')dx \quad (2)$$

avec l'expression du noyau d'interaction  $K$  :

$$K(x, x') = \rho(x) \frac{\cos \theta \cdot \cos \theta'}{\|x - x'\|} \text{vis}(x, x') \quad (3)$$

avec  $\text{vis}(x, x')$  la fonction de visibilit  entre les points  $x$  et  $x'$  et  $L_e(x)$  l' mission propre au point  $x$ .

Une approximation de la solution de cette  quation peut  tre obtenue facilement en discr tisant les surfaces de la sc ne en  l ments polygonaux et en faisant l'hypoth se que la radiosit   $L(x)$  sur l' l ment  $i$  peut  tre approch e par la combinaison lin aire d'un ensemble de fonctions de bases  $\phi_{i,\alpha}(x)$  d finies sur les  l ments :

$$L(x) = \sum_{i,\alpha} L_{i,\alpha} \phi_{i,\alpha}(x) \quad (4)$$

Si ces fonctions de bases sont non recouvrantes ( $\phi(x) = 0, \forall x \notin \text{ l ment}$ ) et si l' mission propre peut aussi  tre exprim e de mani re similaire  $L_c(x) = \sum_{i,\alpha} E_{i,\alpha} \phi_{i,\alpha}(x)$ , l' quation (2) est bien approch e par :

$$\begin{aligned} \sum_{i,\alpha} L_{i,\alpha} \phi_{i,\alpha}(x) &= \sum_{i,\alpha} E_{i,\alpha} \phi_{i,\alpha}(x) \\ &+ \sum_{j,\beta} \int_{S_j} K(x, x') L_{j,\beta} \phi_{j,\beta}(x') dx' \end{aligned} \quad (5)$$

La r solution de cette  quation devant fournir les coefficients  $L_{i,\alpha}(x)$  pour chaque fonction de base et chaque  l ment, la

projection de l'équation (5) (intégration sur l'élément  $i$  du produit de cette équation avec chaque fonction de base) nous donne :

$$\sum_{\alpha} (L_{i,\alpha} - E_{i,\alpha}) O_{i,\alpha,\alpha'} = \sum_{j,\beta} K_{i,\alpha;j,\beta} L_{j,\beta} \quad (6)$$

où les coefficients d'interactions sont donnés par

$$K_{i,\alpha;j,\beta} = \int_{A_i} \phi_{i,\alpha}(x) \int_{A_j} K(x,x') \phi_{j,\beta}(x') dx' \quad (7)$$

et les coefficients de recouvrement par :

$$O_{i,\alpha,\alpha'} = \int_{A_i} \phi_{i,\alpha}(x) \phi_{i,\alpha'}(x) dx \quad (8)$$

Nous obtenons ainsi autant d'équations (6) que de fonctions de base  $\phi_{i,\alpha}$ . Si les fonctions de base sont orthogonales, les coefficients  $O_{i,\alpha,\alpha'}$  sont nuls sauf si  $\alpha = \alpha'$ . Nous avons alors :

$$O_{i,\alpha,\alpha'} = \delta_{\alpha,\alpha'} \cdot N_{i,\alpha} \quad \text{et} \quad N_{i,\alpha} = \int_{S_i} \phi_{i,\alpha}^2(x) dx \quad (8)$$

Dans ce cas, l'équation (5) se réduit à :

$$L_{i,\alpha} = E_{i,\alpha} + \frac{1}{N_{i,\alpha}} \sum_{j,\beta} K_{i,\alpha;j,\beta} L_{j,\beta} \quad (9)$$

On peut vérifier que, si l'on choisit une base de fonction réduite à la fonction constante :

$$\phi_{i,\alpha}(x) = \phi_i(x) = 1 \quad \text{si } x \in S_i \quad \text{et} \quad 0 \quad \text{si } x \notin S_i$$

on a  $N_i = A_i$  et  $K_{i,j} = \rho_i A_i F_{ij} \cdot F_{ij}$  représente la portion de flux lumineux émise par l'élément  $j$ , interceptée par l'élément  $i$  et appelée *facteur de forme*. L'équation (9), multipliée par  $\pi$ , correspond alors à l'équation de radiosité communément utilisée. Des exemples de bases de fonctions plus évoluées sont donnés dans [Bekaert96].

Le calcul des coefficients  $L_{i,\alpha}$  sur chaque élément se fait de la manière suivante :

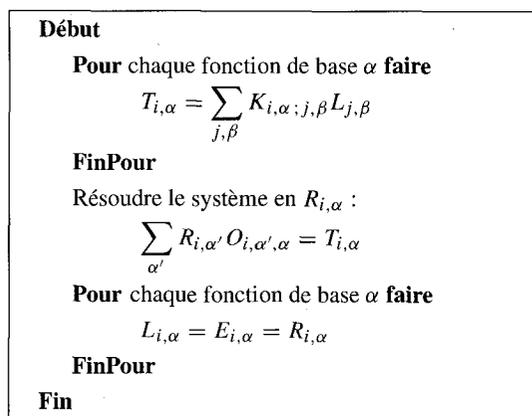


Figure 7. – Algorithme de calcul des  $L_{i,\alpha}$ .

Il est à noter que les coefficients  $O_{i,\alpha,\alpha'}$  peuvent être calculés une seule fois puisqu'ils ne dépendent que des fonctions de base choisies et que les coefficients d'interactions  $K_{i,\alpha;j,\beta}$ , dépendant de la géométrie de la scène, sont évalués lors de la construction du système d'équations.

### 3.2.2. résolution du modèle

La résolution du modèle de radiosité, représenté par l'équation (9) du paragraphe précédent, se déroule en plusieurs étapes :

- échantillonnage de la scène et calcul des coefficients d'interactions.

- résolution des équations de radiosité; construction des textures de radiosité.

Les coefficients d'interaction sont stockés dans une structure, nommée *lien*, représentant l'interaction entre deux éléments de surfaces (figure 8). Un lien permet de connaître non seulement les coefficients  $K_{i,\alpha;j,\beta}$  correspondant au couple d'élément  $(i, j)$  mais aussi les relations de visibilité entre ces éléments et une évaluation de l'erreur commise sur l'interaction, correspondant à la qualité d'approximation de la fonction de radiosité induite par cette interaction. Cette évaluation permet de décider si la liaison entre les éléments peut être considérée comme acceptable ou non. Si cette erreur est trop importante, les éléments sont subdivisés et un nouveau processus de liaison entre les éléments est effectué. Le maillage de la scène est ainsi dynamique. A chaque élément de surface est associée une hiérarchie d'éléments, représentée par un quadtree construit au fur et à mesure de la construction des équations de radiosité.

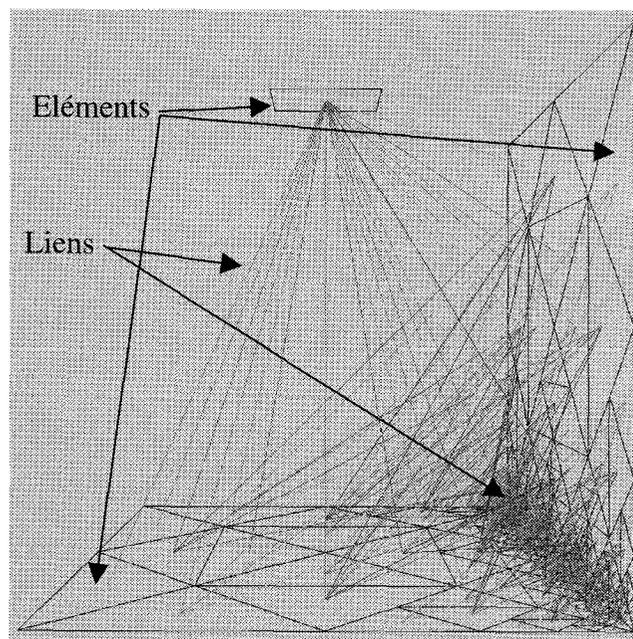


Figure 8. – Liaison entre 3 surfaces.

L'évaluation de l'erreur commise sur le lien se fait de la manière suivante :

Considérons la luminance  $L_i(x)$  de l'élément  $i$  due à l'élément de surface  $S_j$  :

$$L_i(x) = \int_{S_j} K(x, x') L(x') dx' = \sum_{\beta} L_{j,\beta} K_{j,\beta}(x) \quad (10)$$

avec :

$$K_{j,\beta}(x) = \int_{S_j} K(x, x') \phi_{j,\beta}(x') dx' \quad (11)$$

Si la base de fonction est orthogonale, cette luminance est approchée, comme précédemment, par :

$$\tilde{L}_i(x) = \sum_{\alpha} \left( \sum_{\beta} L_{j,\beta} \frac{K_{i,\alpha;j,\beta}}{N_{i,\alpha}} \right) \phi_{i,\alpha}(x) \quad (12)$$

L'erreur commise peut être correctement estimée en prenant la norme  $L_1$  de la différence entre (12) et (10), représentant la variation entre la fonction de radiosité et son approximation par décomposition sur la base de fonctions  $\phi_{i,\alpha}(x)$  :

$$Err = \max_{x \in S_i} \left| \sum_{\beta} L_{j,\beta} \left( \sum_{\alpha} \frac{K_{i,\alpha;j,\beta}}{N_{i,\alpha}} \phi_{i,\alpha}(x) - K_{j,\beta}(x) \right) \right| \quad (13)$$

Dans la pratique, afin de diminuer les coûts de calcul lors de l'évaluation de l'erreur, seule une fonction de base  $\beta$  est utilisée : la fonction constante.

Le processus de liaison initiale permet donc de déterminer, pour un élément de surface, une hiérarchie d'éléments de surface avec, à chaque niveau, un ensemble de liens quantifiant la luminance calculée à ce niveau de détail. Lors de la résolution, un mécanisme de propagation des informations d'énergie entre chaque niveau de la hiérarchie doit être mis en œuvre afin d'assurer une représentation multi-résolution de la fonction de radiosité. Ce mécanisme, appelé *Push-Pull* (Reconstruction/Décomposition) a pour but de déterminer, en descendant la hiérarchie, l'approximation de la fonction de radiosité de l'élément courant en restreignant le domaine de définition de la fonction de radiosité sur l'élément du niveau supérieur à cet élément (phase *Push*) puis, en remontant (phase *Pull*), à effectuer le travail inverse revenant à étendre le domaine de définition de la fonction de radiosité d'un élément.

La résolution des équations de radiosité selon une méthode hiérarchique repose sur l'algorithme suivant : (voir figure 9).

Afin de rendre cette résolution compatible avec une modification interactive de la scène, et plus particulièrement avec le déplacement d'un ou de plusieurs objets, une réorganisation des étapes de calcul et l'utilisation de structures de données adéquates sont nécessaires. En effet, la construction de l'ensemble de liens initial est, de loin, l'opération la plus coûteuse de l'algorithme.

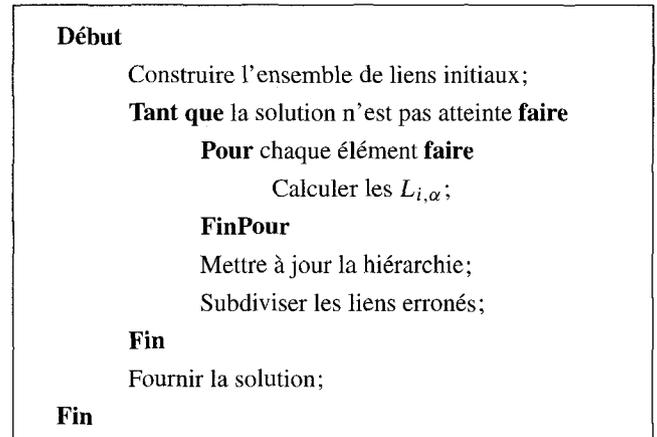


Figure 9. – Algorithme général de la radiosité hiérarchique.

Le calcul de l'ensemble des liens initial pour chaque instant rendrait impossible la manipulation interactive de la scène. Cependant, dans le cadre de scènes dynamiques, l'ensemble de liens initial à l'instant  $t$  peut se déduire de l'ensemble de liens terminal à l'instant  $t - 1$ . Aussi, nous pouvons mettre en œuvre un algorithme de modification incrémental des liens modifiant l'ensemble de liens terminal d'une itération pour former l'ensemble de liens initial de la suivante. De la même manière, lors de la résolution du système d'équations formé par les liens, la solution à l'instant  $t - 1$  est une bonne approximation de la solution à l'instant  $t$  et sera utilisée comme solution initiale de la résolution itérative du système.

L'objectif de l'entité éclairagiste étant de fournir en permanence une solution la plus précise possible au problème de l'éclairage de la scène, la boucle de résolution ne peut être conservée en l'état. En effet, si la scène est modifiée pendant la résolution, celle-ci doit être interrompue et reprise en fonction de la nouvelle configuration géométrique. Si la scène n'est pas modifiée, une nouvelle solution approchée doit être fournie à chaque itération. Afin d'atteindre cet objectif, l'entité éclairagiste doit dialoguer avec le système de Visualisation-Interaction afin de connaître les modifications éventuelles de la scène et de fournir la solution dès qu'elle est disponible. Pour cela, l'algorithme de résolution est modifié comme indiqué (figure 10). La boucle infinie englobant le traitement permet de donner la priorité au traitement des modifications de la scène par l'utilisateur. Ainsi, si la scène a été modifiée, l'ensemble des liens actifs, correspondant aux liens terminaux de l'instant  $t$ , est réévalué afin de prendre en compte ces modifications et une nouvelle solution approchée est calculée. Si la scène n'a pas été modifiée, la convergence du système est évaluée, les liens erronés éventuellement affinés et une nouvelle solution est fournie.

Nous présentons plus en détail, dans les paragraphes suivants, les opérations de construction de l'ensemble de liens à utiliser pour l'itération suivante ainsi que la détermination et la représentation de la solution courante.

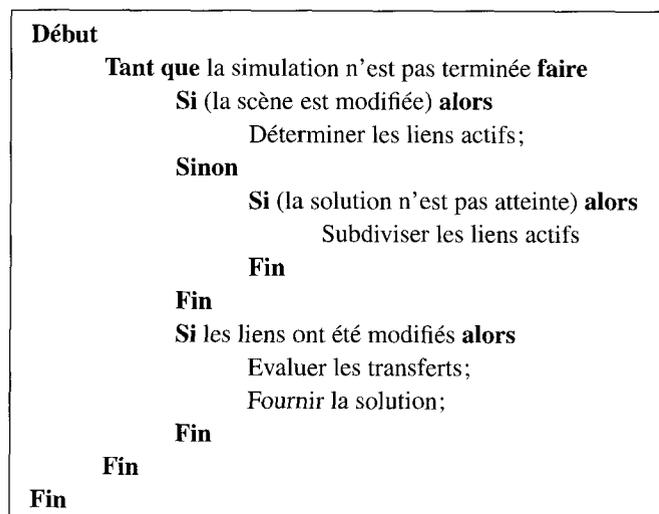


Figure 10. – Algorithme modifié de résolution de la radiosité.

### 3.2.3. construction des liens actifs

L'étape de construction de l'ensemble des liens à utiliser pour la résolution est une combinaison des étapes de construction de liens initiaux et de subdivision des liens existant de l'algorithme classique de résolution. Son but est de définir l'ensemble de liens devant être utilisés pour fournir la solution de l'itération suivante par un mécanisme de construction et/ou de subdivision non récursive de l'ensemble de liens actifs. Pour cela, nous utilisons une structure de données similaire à celle décrite par [Drettakis97] et fondée sur la définition d'un polyèdre de visibilité entre chaque couple d'objet de la scène. Ces objets pouvant être des surfaces ou des groupes de surfaces (*clusters*).

Ce polyèdre, adaptation de la notion de *shaft* [Haines91], est construit comme étant l'englobant le plus proche des deux objets et contient les informations de visibilité entre ces objets (figure 11). Il est aussi la racine d'une arborescence de tuyaux de transferts, chaque tuyau étant la matérialisation du volume englobant le plus proche de la hiérarchie de lien qu'il contient. Les tuyaux, représentant la partie du polyèdre de visibilité dans laquelle les transferts sont effectifs, permettent d'obtenir une solution grossière aux équations de radiosité en les assimilant à des liens  $\beta$ , définis par Smith dans [Smits94]. C'est la solution initiale pour le processus itératif de construction et de résolution du système d'équation des radiosité. Le tuyau, de par son inclusion dans le polyèdre de visibilité, permet aussi la détermination rapide de l'ensemble de liens susceptibles d'être modifiés par le mouvement d'un objet.

Les informations de visibilité stockées dans le polyèdre initial sont classées en deux catégories : *Exactes* et *Approchées*. Les informations de visibilité exactes sont représentées par les graphes d'aspect et les projections inverses des deux objets [Stewart94],

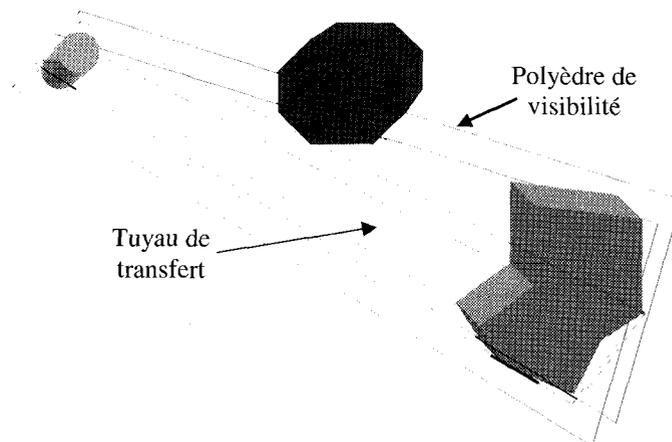


Figure 11. – Polyèdre de visibilité et hiérarchie de tuyaux de transferts.

les informations approchées sont représentées par une image représentant la vision d'un objet depuis l'autre.

Ces polyèdres permettent une recherche rapide de l'ensemble des liens à modifier lors de la manipulation de la scène. En effet, lors du déplacement d'un objet, seuls les liens contenus dans les polyèdres reposant sur cet objet ainsi que les liens contenus dans des polyèdres affectés par le mouvement de l'objet sont à mettre à jour. Lors de la modification de l'ensemble de liens définissant le système d'équations, il peut arriver, suite à un mouvement particulier d'un objet, qu'une arborescence de lien soit supprimée à l'instant  $t$  puis reconstruite à l'instant  $t + i$ . Aussi, lors de la modification de l'ensemble de liens, la suppression des liens devenus inutiles ne se fait que dans certains cas, en fonction de leur marque. Le système de marquage que nous employons permet de savoir quels types d'objets sont en relation, quelle est la validité du lien et quelle est sa probabilité d'être modifié à l'itération suivante. Ce marquage est une extension de ceux proposés par [Drettakis97] et [Shaw97].

Le résultat de la phase de construction des liens est un ensemble de liens marqués actifs et résultant soit de la suppression ou de l'apparition d'une relation entre deux objets, soit de la subdivision d'un ensemble de liens existants pour augmenter la précision de la solution. Il est à noter que les parcours des arborescences de polyèdres et de liens sont faits en largeur d'abord afin de répartir plus équitablement le travail effectué sur l'ensemble de la hiérarchie.

Les concepts décrits dans cette section ont été implantés et les temps de calculs sont satisfaisant compte tenu de l'objectif d'interactivité que nous nous sommes fixés. L'identification de l'ensemble de liens actifs après modification de la scène s'effectue en moins de 1 seconde pour une scène de complexité moyenne (10000 éléments). Ce temps de calcul contient la mise à jour éventuelle des informations de visibilité représentées par un maillage des discontinuités (de l'ordre de 0.1 seconde pour une telle scène) [Bonnafous98] et la modification de l'arborescence

de liens suite à l'évolution de la scène (de l'ordre de 0.8 secondes) [Dupuy97].

### 3.2.4. mise à jour de la solution – textures de radiosité multi-échelle

Après la phase de modification du système d'équations (modification de l'ensemble de liens), la phase de résolution proprement dite est effectuée. Cette phase de résolution consiste en un parcours en largeur d'abord de l'arbre de liens et produit, pour chaque élément, un n-uplet formé des coefficients  $L_{i,\alpha}$  décrits à la section 3.2.1. Ces coefficients définissent, à chaque niveau de subdivision, la fonction de radiosité. Cependant, ils ne sont pas exploitables directement car il ne prennent en compte que les transferts effectués à ce niveau. Aussi, une phase de mise à jour (appelée *Push-Pull*) dont l'objectif est de déterminer, à tous les niveaux, l'approximation de la radiosité totale doit être mise en œuvre. Dans notre application, cette phase possède un double objectif :

- l'obtention d'une approximation de la radiosité à tous les niveaux.
- La création des textures qui seront utilisées pour la visualisation de la scène.

Pour atteindre ce double objectif, nous associons, à chaque élément de la géométrie initiale, une image dont la taille dépend de l'aire de l'élément. Nous avons ainsi, au niveau 0 de la hiérarchie une correspondance directe entre le domaine de définition de la fonction de radiosité et l'image permettant de la représenter. Chaque ajout d'un niveau de subdivision dans l'arbre des éléments entraîne la création d'une nouvelle texture de telle sorte que chaque élément de la subdivision (définissant un sous-domaine de l'élément précédent) recouvre une partie de la texture (figure 12).

Lors de la réalisation du *Push-Pull*, nous effectuons le même traitement sur les deux arborescences, les éléments et la texture. Lors de la descente (*Push*) dans l'arbre, la fonction de radiosité et la texture sont obtenues en sommant, à chaque niveau, la valeur stockée sur le niveau et la valeur venant du niveau supérieur. Lors

de la remontée (*Pull*), les valeurs de la fonction de radiosité et la texture du niveau supérieur sont obtenues par convolution des informations du niveau inférieur.

## 4. conclusion

Nous avons vu que les méthodes extrapolées de la radiosité primitive pour permettre la prise en compte des environnements de plus en plus complexes sont facilement adaptables à des environnements dynamiques. Ces méthodes de radiosité hiérarchiques assurent des convergences rapides vers une solution approchée dont l'écart à la solution optimale est maîtrisée. De même, les raffinements ou les mises à jour nécessaires en cas de modification de la scène peuvent être très rapides. Le problème crucial est alors de détecter les liens mis en cause par ces changements et de les traiter, en leur ayant éventuellement attribué une priorité.

La mise à jour de la photométrie de la scène, légèrement plus lente que la visualisation, est donc réalisée par une tâche indépendante de la visualisation interactive, de façon asynchrone et intermittente (quand c'est nécessaire). C'est ainsi que l'on assure à tout instant la qualité photométrique optimale disponible. L'exécution de ces traitements parallèlement au rendu est rendue possible par l'utilisation des concepts de VIPER, qui sont déjà à la base de PROVIS. L'intégration du calcul de l'illumination globale, sous la forme d'une entité particulière au sein de PROVIS, permet de faire bénéficier le prototypage virtuel des améliorations substantielles de la qualité du rendu visuel, facteur primordial dans des domaines comme l'architecture ou l'urbanisme. Il devient alors possible d'interagir, même à plusieurs, avec l'environnement lors de visites virtuelles d'ensembles architecturaux, mais aussi, c'est bien le but du prototypage virtuel, de pouvoir visualiser rapidement (interactivement) et avec un rendu réaliste, le résultat d'un nouveau choix de conception, de modélisation géométrique, de matériaux utilisés ou de luminaires.

Les travaux menés ainsi que les résultats obtenus au cours du projet de recherche PROVIS sont actuellement exploités dans le cadre du projet Esprit n°26285 intitulé CAVALCADE (*Collaborative Virtual Construction And Design*). Piloté par CS-CISI et soutenu par l'Union Européenne, ce projet fournira un produit industriel pour le prototypage virtuel coopératif. De plus amples informations concernant le projet CAVALCADE sont disponibles à l'adresse suivante :

<http://www.cisi.fr/cavalcade> 5.1

## BIBLIOGRAPHIE

- [Abarnabel96] R. M. Abarnabel, E. Brechner et W. McNeely, «FlyThru the Boeing 777», Presentation at the Digital Bayou, SIGGRAPH, 1996.
- [Balaguer96] J.F. Balaguer, «VRML for LHC Engineering», Proceedings of the Eurographics Workshop on Virtual Environments, 1996.

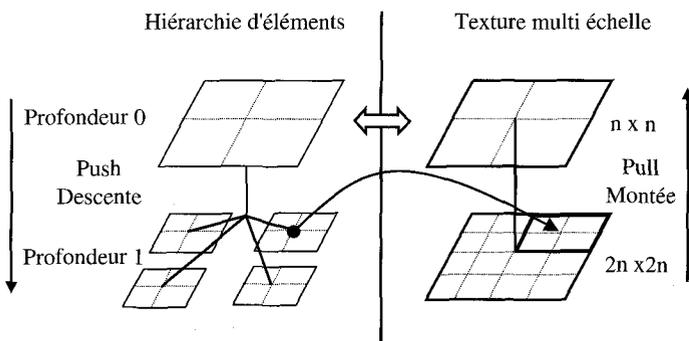


Figure 12. – Association éléments-texture.

- [Bekaert96] Ph. Bekaert et Y.D. Willems, «HIRAD : A Hierarchical Higher Order Radiosity Implementation», Proceedings of the 12th Spring Conference on Computer Graphics, Bratislava, Slovakia (June 1996).
- [Balet96] O. Balet, P. Torguet, V. Gaildrat et R. Caubet, «Autonomous Entities In Distributed Virtual Environments», Proceedings of Multimedia Modeling '96, Published in Towards the information super highways, World Scientific, p. 457-471, 1996.
- [Balet97] O. Balet, H. Luga, Y. Duthen et R. Caubet, «PROVIS : A Platform For Virtual Prototyping And Maintenance Tests», Proceedings of IEEE Computer Animation '97, IEEE Computer Society Press, p. 39-47, 1997.
- [Bonnafous98] Vincent Bonnafous, «Maillage de discontinuités dynamique», DEA de l'université Paul Sabatier, IRIT, Toulouse, Juin 1998.
- [Cohen86] M. F. Cohen, D. P. Greenberg, D. S. Immel et P. J. Brock, «An efficient radiosity approach for realistic image synthesis», IEEE Computer Graphics & Applications 6, no. 2, p. 26-35, 1986.
- [Cohen93] M. F. Cohen et J. R. Wallace, «Radiosity and realistic image synthesis», AP Professional, 1993.
- [Drettakis97] G. Drettakis et F. X. Sillion, «Interactive Update Of Global Illumination Using A Line-Space Hierarchy», Computer Graphics (SIGGRAPH'97 Proceedings), p.57-64, August 1997.
- [Dupuy97] Yann Dupuy, «Eclairage global des environnements animés», DEA de l'université Paul Sabatier, IRIT, Toulouse, Septembre 1997.
- [Ellis96] G. Ellis, «They're Not Making 'Em Like They Used To : Virtual Reality Saves Time and Money in Manufacturing and Construction », Iris Universe, Summer, 1996.
- [Fujimoto86] A. Fujimoto, T. Tanaka et K. Iata, «ARTS : Accelerated Ray-Tracing System», IEEE Computer Graphics & Applications, p. 16-26, Avril 86.
- [Greenfield96] D. Greenfield, «Virtual Prototyping at Rolls-Royce», Intelligent Systems, Report 13(1), 1996.
- [Haines91] E. A. Haines et J. R. Wallace, «Shaft culling for Efficient Ray-Traced Radiosity.», Photorealistic Rendering in Computer Graphics, Proceedings of the Second Eurographics Workshop on rendering, Springer-Verlag, New-York, 1994, p. 122-138.
- [Hanrahan91] P. Hanrahan, D. Salzman et L. Aupperle, «A rapid hierarchical radiosity algorithm.», Proceedings of SIGGRAPH'91, Computer Graphics, vol. 25, n°4, p. 197-206, 1991.
- [Kajiya86] J.T. Kajiya, «The rendering equation», Proceedings of SIGGRAPH '86, Computer Graphics, vol 20, n°4, p. 143-150, 1986.
- [Kardasse97] C. Kardassevitch, M. Paulin, JP. Jessel et R. Caubet, «A Hierarchical Radiosity Platform Using Efficient Data Structures and VRML 97», Proceedings of Compugraphics'97, Vilamoura, Portugal, 1997.
- [Shaw97] E. Shaw, «Hierarchical Radiosity for Dynamic Environments», Computer Graphics forum, Vol 16, n°2, p. 107-118, 1997.
- [Segal92] M. Segal, C. Korobkin, R. Van Widenfelt, J. Foran et P. Haerberli, «Fast Shadows and Lighting Effects Using Texture Mapping», Computer Graphics, Vol. 18, n°2, p. 249-252, July 1992.
- [Sillion94] F. X. Sillion et C. Puech, «Radiosity and Global Illumination», Morgan Kaufmann Publishers, San Francisco, 1994. ISBN 1-55860-277-1
- [Sillion95] F. X. Sillion, «Clustering and volume scattering for hierarchical radiosity calculations». Photorealistic Rendering Techniques, 1995, p. 105-118.
- [Smits94] B. Smits, J. Arvo et D. Greenberg, «A Clustering algorithm for radiosity in complex environments», Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings) , n°4, p. 435-442.
- [Smits92] B. Smits, J. Arvo et D. H. Salesin, «An importance-driven radiosity algorithm», Computer Graphics (ACM SIGGRAPH'92 Proceedings), no. 4, 1992, p. 273-282.
- [Soler98] C. Soler et F.X. Sillion, «Fast Calculation of Soft Shadow Textures Using Convolution», to appear in the SIGGRAPH'98 Conference Proceedings
- [Stewart94] A. J. Stewart et S. Ghali, «Fast Computing of Shadow Boundaries Using Spatial Coherence and Backprojections», Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings), p. 231-238..
- [Torguet97] P. Torguet, O. Balet et R. Caubet, «A Software Architecture For Collaborative Virtual Prototyping», Proceedings of CompuGraphics'97, p. 310-319, 1997.
- [Torguet98] P. Torguet, «VIPER : Un modèle de calcul réparti pour la gestion d'environnements virtuels», Thèse de doctorat, Université Paul Sabatier, Toulouse, 1998.
- [VRML97] The VRML Consortium Incorporated, «The Virtual Reality Modeling Language», ISO/IEC 14772-1 :1997, <http://www.vrml.org/Specifications/VRML97/>.

*Manuscrit reçu le 8 juin 1998.*

## Rendu réaliste et prototypage virtuel interactif

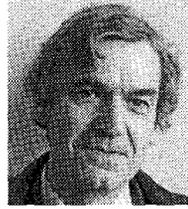
### LES AUTEURS

#### Mathias PAULIN



Mathias Paulin est Maître de Conférences en informatique à l'Université Paul Sabatier de Toulouse depuis septembre 1997. Il a obtenu son doctorat en informatique à l'Université Paul Sabatier de Toulouse en décembre 1995 sur le thème «Algorithmes pour la radiosité : parallélisme et échantillonnage». Ses domaines d'activités concernent la modélisation de la réflectance, la modélisation de l'éclairage et les outils pour le rendu réaliste interactif. e-mail : Mathias.Paulin@irit.fr

#### René CAUBET



René Caubet est professeur d'informatique à l'Université Paul Sabatier, à l'IRIT. Il est le directeur de l'Equipe Synthèse d'Images et Réalité Virtuelle de l'IRIT. Il est responsable de la formation doctorale «Informatique de l'Image et du Langage», et vice président de l'Université. Ses domaines d'intérêts incluent la modélisation, le rendu réaliste et la réalité virtuelle. e-mail : caubet@irit.fr

#### Cyril KARDASSEVITCH



Cyril Kardassevitch prépare un doctorat en informatique à l'Université Paul Sabatier, à l'IRIT. Ses domaines d'intérêt sont les modèles de radiosité hiérarchiques, le clustering, VRML97, et la combinaison des modèles d'illumination globale et la réalité virtuelle. e-mail : kardasse@irit.fr

#### Mathieu ROBART



Mathieu Robart prépare un doctorat en informatique à l'Université Paul Sabatier, à l'IRIT. Ses domaines d'intérêts sont les modèles physiques d'interaction matériaux - lumière dans l'illumination globale, les méthodes de stockage et de compression par ondelettes. e-mail : robart@irit.fr

#### Jean-Pierre JESSEL



Jean Pierre Jessel est Maître de Conférences à l'Université Paul Sabatier, à l'IRIT. Il a obtenu son doctorat en informatique sur la modélisation, la radiosité, le parallélisme. Ses travaux de recherche concernent la réalité virtuelle distribuée, l'animation temps réel, les interactions 3D. e-mail : jessel@irit.fr

#### Olivier BALET



Olivier Balet a obtenu son doctorat en informatique à l'Université Paul Sabatier de Toulouse sur le thème de l'utilisation de la réalité virtuelle pour le prototypage coopératif. Il est responsable «Recherche et Technologie» pour la réalité virtuelle au sein de la société CS-CISI. Ses domaines d'activités recouvrent la réalité virtuelle distribuée, l'interaction homme machine, le prototypage virtuel ainsi que le travail coopératif. e-mail : Olivier.Balet@cisi.cnes.fr