

Synthèse d'Image : Evolution et Applications

Kadi Bouatouch
IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
e-mail : kadi@irisa.fr

Mots Clés: *synthèse d'image, modélisation, simulation, éclairage, animation, parallélisation.*

Résumé

La synthèse d'image a connu une évolution fantastique ces deux dernières décennies et son champ d'application est très varié. Toutes les méthodes et tous les algorithmes dérivés de la synthèse d'image sont maintenant utilisés dans des domaines autres que celui de l'image tels que : infrarouge, thermique, téléphone.... Cet article ne prétend pas décrire de manière exhaustive ce vaste domaine, il veut juste présenter les principes de base utilisés ainsi que les différentes applications.

Abstract

Image synthesis has seen a significant evolution during the last two decades and is now applied in various scientific fields. All the methods and algorithms employed in image thesis are now applied to various other scientific areas such as: infrared, thermal engineering, telephone... This paper does not claim to be exhaustive but aims at giving the main bases on which relies the image synthesis field. It also makes list of the different applications of this latter.

1 Principes et Définitions

La synthèse d'image a connu une évolution significative durant ces deux dernières décennies. En effet, il y a peu de temps c'était un ensemble de techniques permettant de produire différents effets (perspective, ombres, transparence, textures...) qui reproduisent plus ou moins fidèlement les effets lumineux réels. Elle s'appuyait uniquement sur des modèles empiriques et son seul but était d'obtenir une représentation de ces effets sans chercher à les reproduire fidèlement. Le but recherché était alors le *réalisme*.

Depuis une dizaine d'années la synthèse d'image prend une direction plus scientifique en s'appuyant sur différents domaines de la science tels que la physique et les mathématiques... L'objectif visé est de reproduire le plus précisément possible la réalité. Il s'agit donc de *photoréalisme*.

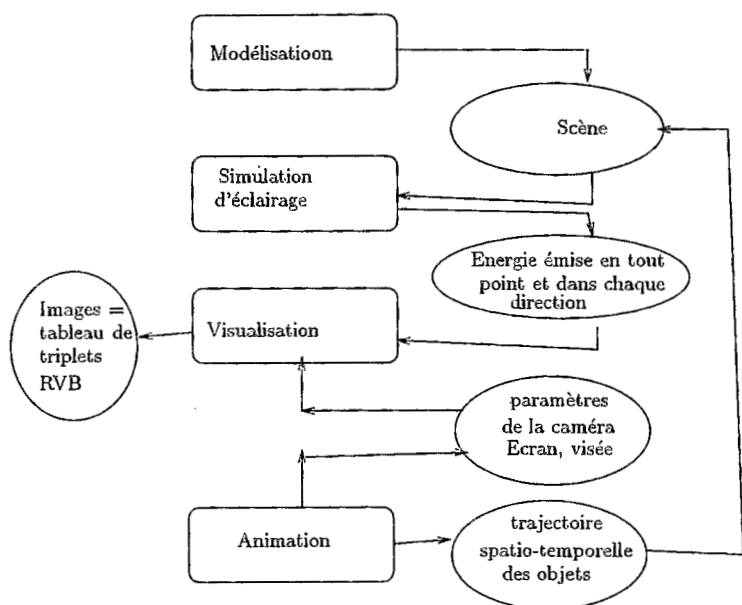


FIG. 1 - Différentes étapes de la synthèse d'image

Le processus de création d'une image de synthèse comprend quatre parties : la modélisation géométrique, la simulation des interactions lumière-matière (ou technique de rendu ou simulation d'éclairage), la visualisation et l'animation (création de séquences d'images animées).

La modélisation géométrique consiste d'une part à créer des objets et à les placer dans un repère cartésien, d'autre part à leur associer des paramètres géométriques (rayon et centre pour une sphère, liste de sommets pour un polygone, normale...) et des paramètres d'aspect (réflectivité, transparence, textures, couleurs...).

Après l'étape de modélisation, des sources de lumière sont placées dans l'environnement créé (ou scène) et la simulation de l'éclairage de cet environnement est réalisée. Il s'agit en fait de calculer toutes les interactions entre les sources de lumière et les objets de la scène telles que : multiples réflexions, diffusion (dans un milieu tel que fumée, brouillard...) transmission (à travers des objets transparents), reflets. Le résultat est l'énergie lumineuse émise en chaque point de chaque surface de l'environnement et dans toutes les directions..

Une fois la simulation de l'éclairage terminée, le processus de visualisation peut commencer. Il consiste à placer dans la scène un observateur (point de vue), appelé aussi caméra, et un écran virtuel (grille de petites surfaces carrées appelées pixels) sur lequel seront projetés les objets de la scène et enfin fixer une direction de vue formée par l'observateur et un point de visée choisi par l'utilisateur. L'énergie lumineuse reçue par l'observateur à travers chaque pixel est déterminée et convertie en trois composantes RVB (rouge, vert, bleu). L'image de synthèse obtenue est représentée alors par un tableau à deux dimensions

de triplets RVB.

L'animation consiste à décrire le mouvement des objets de la scène, de la caméra, des sources de lumière et les changements d'aspect de ces objets dans le temps. Tout objet mobile est caractérisé par une trajectoire spatio-temporelle. Le résultat du module d'animation est une séquence d'images de synthèse (50 par seconde). En effet tous les cinquantièmes de seconde une image est calculée en tenant compte de la nouvelle position des objets mobiles.

Ces différentes étapes de la synthèse d'image sont illustrées par la figure 1.

Dans la suite de cet article, nous présentons ces différentes étapes tout mettant l'accent sur leur évolution au cours des deux dernières décennies. Ensuite une section sera consacrée à leur parallélisation. Enfin nous terminerons par une section montrant les différentes applications visées par la synthèse d'image. L'animation étant un domaine assez large, elle ne sera pas traitée dans ce document pour des raisons de limitation en terme de nombre de pages. Le lecteur pourra se reporter à [13] pour plus de détails.

2 Modélisation

Depuis la fin des années 60 nous assistons à l'utilisation sans cesse croissante des systèmes de modélisation géométrique dans les domaines de la CAO (Conception Assistée par ordinateur) et de la CFAO (Conception et Fabrication Assistée par Ordinateur). Ces logiciels sont utilisés dans diverses applications telles que : dessin industriel, animation, architecture, simulation...

La modélisation géométrique est un ensemble de méthodes utilisées pour décrire la forme ou les caractéristiques géométriques d'un objet.

Dans un système de modélisation, appelé *modeleur*, un objet peut être représenté par plusieurs modèles [20] : fil de fer, surfacique, solide.

Dans le modèle *fil de fer* l'objet est représenté par un ensemble de segments de courbe ou de droite. Par exemple un polyèdre est représenté par ses arêtes.

Ensuite un autre modèle dit *surfacique* est apparu. Dans ce cas l'objet est constitué d'un ensemble de morceaux de surface (ou carreaux) définis par le type de surface support et la description de ses bords (contours). Tout au début de la synthèse d'image, ces carreaux étaient des surfaces planes, ce qui nécessitait la conversion des objets non plans en un ensemble de carreaux plans. Mais dès les années 70 un autre type de surface a été introduit [2], il s'agit des surfaces gauches ou surfaces paramétrées. Ces surfaces, très utilisées actuellement, permettent de modéliser des objets complexes tels que les carrosseries de voiture, les ailes d'avion... Il existe différents types de surface, chacune ayant ses avantages et inconvénients. On peut citer les surfaces de Bézier, de Coons, B-splines, Beta-splines, NURBS (B-spline rationnelle non uniforme) [21]. Mis à part les surfaces de Bézier, ces surfaces sont définies par morceaux, ce qui permet des déformations locales, déformations très fréquentes et indispensables en CAO.

Depuis 1980 [26], les modèles solides sont utilisés et plus particulièrement le modèle CSG (Constructive Solid Geometry). Etant donné un ensemble de primitives solides (parallélépipèdes, cônes, cylindres, tores...) dont on paramètre les dimensions (hauteur, largeur, longueur, rayon...), d'un ensemble de transformations géométriques (translation,

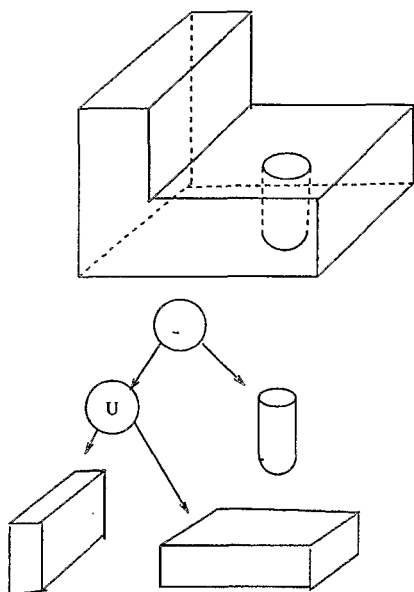


FIG. 2 - *Objet CSG*

rotation, mise à l'échelle...), d'un ensemble d'opérateurs ensemblistes (union, intersection, différence) on peut construire des objets CSG. En fait un objet CSG est le résultat d'opérations ensemblistes sur des primitives solides de base. La représentation interne (structure de données) d'un solide CSG est un arbre binaire où chaque nœud non terminal est une opération ensembliste et les feuilles sont des primitives solides (voir figure 2).

Les modèles classiques de conception d'objets spatiaux sont fondés sur l'utilisation de méthodes impératives, associées à une démarche de conception ascendante, dans laquelle l'utilisateur spécifiera une à une les valeurs des paramètres des modèles qu'il utilise (construction explicite des sommets, des arêtes et des facettes d'un volume). Tous ces systèmes ont en commun d'être très près du modèle géométrique, et donc très loin des considérations du concepteur, du moins dans les premières phases de la conception. Ces approches n'offrent en outre qu'une seule solution à l'utilisateur : celle de définir complètement tous les composants, avant de pouvoir les assembler, alors qu'il serait souhaitable de pouvoir manipuler des objets partiellement définis. De plus, ce genre d'approche trouve rapidement ses limites dès que le concepteur pense à la forme de ses objets, en termes plus sémantiques. Une autre approche consiste en la définition de modèles dynamiques, d'un niveau d'abstraction plus élevé que le niveau géométrique, permettant la description de scènes de manière plus qualitative en termes de propriétés et contraintes sur les objets et leurs relations ; elle s'intéresse en outre non plus seulement à la représentation de connaissances sur les objets mais aussi sur le processus de conception en lui-même. Ces approches plus déclaratives (car elles manipulent des données à la fois qualitatives et

quantitatives), permettent une spécification plus progressive, “par raffinements successifs”, laissant au logiciel le soin de proposer des solutions, de gérer les informations incomplètes, voire incohérentes. On parlera dans cas de modélisation déclarative [10].

3 Simulation d'éclairage ou Technique de Rendu

Pour atteindre le photoréalisme, les efforts de recherche ont porté principalement sur l'élaboration de modèles d'illumination permettant de rendre les effets de transparence, les reflets, les pénombres ainsi que les multiples réflexions. Un modèle d'illumination permet de calculer l'énergie lumineuse (sous forme d'un spectre dépendant de la longueur d'onde ou de 3 composantes RVB) en tout point de la scène et dans chaque direction, en prenant en compte les multiples réflexions et réfractions (un objet recevant de la lumière d'une source la réfléchit ou la réfracte vers un autre objet qui la réfléchit ou la réfracte à son tour vers d'autres objets et ainsi de suite).

Un modèle est dit local s'il ne tient compte que de l'apport des sources lumineuses (éclairage direct) et global s'il prend en compte aussi l'apport de tous les objets de la scène de manière indirecte (éclairage indirecte).

Avant de présenter les différentes méthodes de simulation, nous définissons quelques termes techniques utilisés dans la suite de ce document.

– Réflexion diffuse

Une lumière incidente arrivant sur un objet mat pénètre dans celui-ci puis ressort avec la même amplitude quelque soit la direction d'observation (de réflexion), c'est la réflexion diffuse.

– Réflexion spéculaire

Une lumière incidente arrivant sur un objet lisse (réfléchissant) est réfléchi par la surface de celui-ci mais uniquement dans certaines directions autour de la direction de réflexion idéale. Cette dernière est telle que l'angle de réflexion est égale à l'angle d'incidence. Par exemple un miroir est quasiment parfaitement spéculaire, c'est à dire que la réflexion diffuse est presque nulle et qu'il n'y a qu'une seule direction de réflexion spéculaire qui est celle de la réflexion idéale. De même un objet parfaitement mat (c'est à dire parfaitement diffus) a une réflexion spéculaire nulle. Notons que les objets peuvent être diffus et spéculaires.

– Emissance d'une surface

C'est l'énergie lumineuse totale émise par unité de temps (flux) et par unité de surface, elle est appelée aussi *radiosité*.

– Luminance d'une surface en un point et dans une direction donnée

C'est le flux émis par unité de surface et dans une direction donnée. Pour un objet parfaitement diffus, la radiosité d'une surface en un point est égale au produit de la luminance en ce point (qui est la même quelque soit la direction) par π . Rappelons que le processus de création d'image de synthèse calcule pour chaque pixel la luminance atteignant l'observateur provenant d'un point de la scène à travers ce pixel.

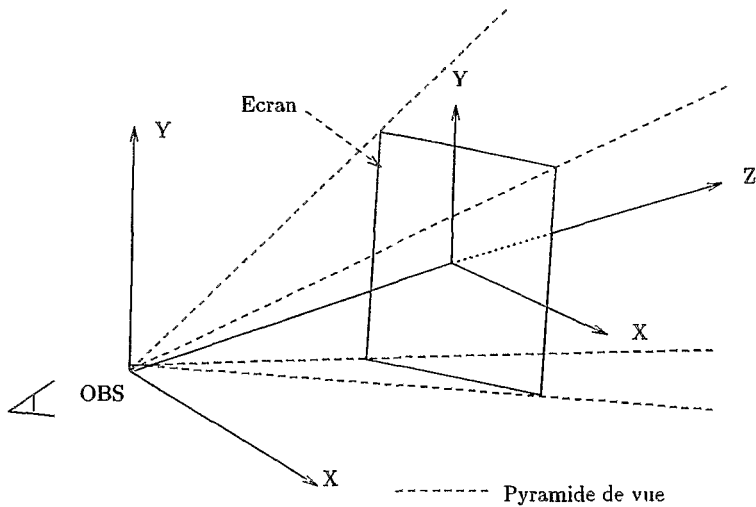


FIG. 3 - Repère de l'observateur et Pyramide de vue

A une luminance est associée une distribution spectrale $L(\lambda)$ qui est fonction de la longueur d'onde λ . $L(\lambda)$ est appelée *luminance spectrale*. Cette dernière est ensuite convertie en 3 composantes RVB (Rouge, Vert, Bleu) par le module de visualisation.

- Composante diffuse directe et globale

La réflexion diffuse due à une source lumineuse est appelée *composante diffuse directe*. Celle due aux autres objets de la scène après multiples réflexions porte le nom de *composante diffuse indirecte*. La somme de ces deux composantes est appelée *composante diffuse globale*.

Différentes méthodes de rendu existent, elles sont décrites ci-après.

3.1 Méthodes projectives

Les premières méthodes de simulation d'éclairage (ou méthodes de rendu) sont apparues au début des années 70 [12]. Nous les appelons *méthodes projectives*. Elles utilisent un modèle d'illumination local, ce qui réduit considérablement le réalisme puisque l'éclairage indirecte dans un environnement représente la plus grosse partie de l'éclairage. Pour ces méthodes les objets sont modélisés par des polygones plans. Les différents traitements impliqués par ces méthodes pour générer une image sont :

- transformation de tous les polygones du repère absolu (défini lors de l'étape de modélisation) vers le repère lié à l'observateur (OBS, X, Y, Z) tel que le montre la figure 3 ;
- Le découpage par rapport à une pyramide de vue, c'est à dire élimination de tous les polygones ne se trouvant pas dans le champ visuel ;

- élimination des polygones ou parties de polygones cachées, soit avant ou après projection en perspective des polygones sur l'écran virtuel ;
- Remplissage des polygones, c'est à dire détermination des pixels contenues dans la projection d'un polygone et calcul de leurs composantes RVB à l'aide d'un modèle d'illumination local et des informations de texture ;
- calcul des ombres portées.

3.2 Méthode du Lancer de Rayon

3.2.1 La méthode

Afin d'augmenter le réalisme une autre méthode a été proposée en 1980 par Whitted [35], Il s'agit de la méthode de synthèse d'image par lancer de rayon. Cette méthode a longtemps été considérée comme celle permettant de prendre en compte tous les effets et d'obtenir le meilleur niveau de réalisme. Cette technique simule le fonctionnement d'un appareil photographique, en parcourant le trajet inverse de la lumière (figure 4). La méthode consiste à lancer des rayons à partir d'un observateur à travers les points d'un écan (pixels) vers les objets d'une scène. Le programme calcule alors les intersections de chaque rayon avec chaque objet, et détermine la plus proche. Les contributions des sources lumineuses à la luminance (ou couleur) du pixel sont calculées en lançant des rayons à partir du point d'intersection en direction des sources lumineuses. S'il se trouve un objet opaque entre le point d'intersection et la source lumineuse, alors le point se trouve à l'ombre. En fonction des propriétés physiques de réflexion et de transparence de l'objet rencontré, un nouveau rayon est lancé dans la direction de réflexion si l'objet est réfléchissant ainsi qu'un autre rayon dans la direction de réfraction si l'objet est transparent. Ces nouveaux rayons permettent d'obtenir des reflets et des effets de transparence. L'éclairément du point d'intersection est alors calculé en appliquant un modèle d'illumination qui tienne compte de tous ces rayons. Les rayons réfléchis permettent de calculer les reflets d'objets sur d'autres objets tandis que les rayons transmis permettent de prendre en compte les objets transparents.

3.2.2 Accélération

Le calcul d'intersection étant très coûteux, il est indispensable de le réduire en utilisant une structure de données qui permettrait d'éviter des calculs inutiles tels que l'intersection avec des objets ne se trouvant pas sur le trajet d'un rayon. Cette structure de données est hiérarchique et peut être par exemple un arbre binaire dont les nœuds sont des volumes englobants de sous-scènes. Le volume englobant d'un objet ou d'une sous-scène doit impliquer un faible coût de calcul d'intersection avec un rayon. Cette structure pourrait être construite en subdivisant la scène en deux sous-scènes, calculer le volume englobant de chaque sous-scène et créer les nœuds associés. Ce processus est appliqué récursivement à chaque sous-scène jusqu'à ce que les sous-scènes contiennent un nombre d'objets inférieur à un certain seuil. Notons que le volume englobant correspondant à la racine de l'arbre binaire englobe toute la scène et que les feuilles englobent un faible nombre d'objets. Lors

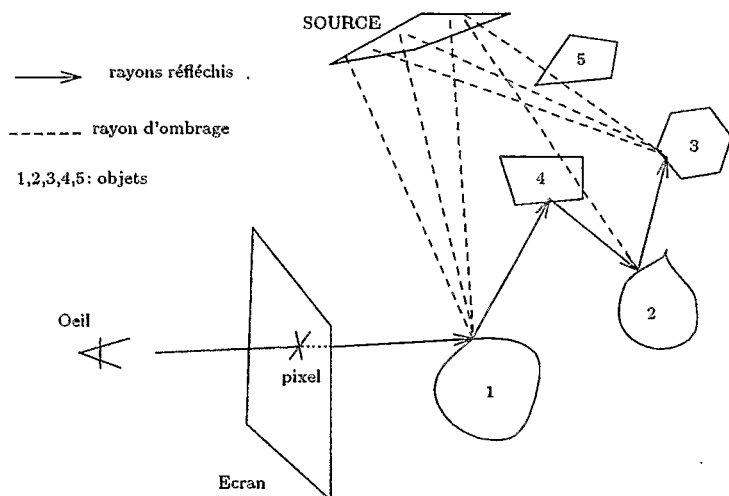


FIG. 4 - Lancer de Rayon

du calcul d'intersection entre un rayon et la scène, cette structure est parcourue de la racine vers les feuilles. En chaque nœud N atteint on teste s'il y a intersection entre le volume englobant associé et le rayon. Si oui on réitère ce processus sur ces deux fils jusqu'à ce qu'on atteigne les feuilles ; sinon on abandonne complètement le sous-arbre dont la racine est N . Dans ce dernier cas toutes les feuilles du sous-arbre de N ne sont pas prises en compte par le calcul d'intersection, ce qui réduit considérablement les temps de calcul [14].

Dans le cas de scènes complexes l'arbre décrivant la hiérarchie de volumes englobants devient très profond et par conséquent son parcours devient très coûteux en temps de calcul. Il est alors indispensable de réduire ce coût en exploitant au maximum la relation de voisinage qui existe entre les objets de la scène. Cette relation est appelée *Cohérence Spatiale*. Cette dernière est obtenue en subdivisant l'espace contenant la scène en régions 3D de petite taille ne contenant qu'un petit nombre d'objets. Cet espace est en général un parallélépipède. Un rayon qui pénètre dans une région n'intersecte que les objets se trouvant dans cette région. Lorsqu'un rayon n'intersecte aucun objet d'une région, on détermine la région suivante se trouvant dans la direction de ce rayon puis on effectue l'intersection entre ce rayon et les objets de cette nouvelle région [3].

3.2.3 Discussion

Etant donné la simplicité de sa mise en œuvre, la méthode du lancer de rayon reste encore très attractive pour beaucoup de chercheurs et d'industriels. En effet elle peut s'appliquer à des domaines différents de celui de la synthèse d'image. Néanmoins cette méthode reste coûteuse en terme de temps de calcul, c'est pourquoi des versions parallèles ont été proposées dans la littérature tel que le montre la section 5.2.

3.3 Radiosité

3.3.1 Principe

La méthode du lancer de rayon, bien qu'elle permette d'augmenter le réalisme en traitant des objets spéculaires et transparents, présente l'inconvénient de ne pas calculer précisément la composante diffuse globale. Par contre cette composante est évaluée de manière précise par la méthode de radiosité.

La méthode de radiosité [19, 31] considère que tous les objets d'une scène sont parfaitement diffus (mats), et sont subdivisés en petits polygones (appelés aussi carreaux) de telle façon que la radiosité (ou émittance, ou flux émis par unité de surface) en tout point d'un carreau est constant. Cette méthode s'adapte bien à la simulation d'éclairage d'un ensemble architectural.

Le but va être de déterminer l'émittance (radiosité) de chaque carreau de surface en supposant connue l'émittance de tous les carreaux visibles depuis le careau examiné, ceci afin de pouvoir évaluer l'énergie incidente. Pour le carreau d'indice i , les conditions d'équilibre énergétique s'expriment par l'équation de radiosité qui devient :

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^{j=N} F_{ji} B_j A_j \quad (1)$$

avec :

- B_i : émittance totale de la surface A_i (Radiosité) ;
- E_i : émittance de la surface A_i due à l'auto-émissio(pour les sources de lumière) ;
- ρ_i : coefficient de réflexion ou réflectivité ;
- F_{ij} : facteur de forme géométrique (fraction du flux émis par le carreau i effectivement reçue par le carreau j) ;
- A_i : surface du carreau i ;
- N : nombre de carreaux dans la scène.

Toutes les grandeurs photométriques (B_i, E_i) définies dans cette équation sont supposées constantes sur tout un carreau. Du fait de la propriété de réciprocité, nous avons :

$$F_{ij} A_i = F_{ji} A_j.$$

Dans ces conditions l'équation (1) devient :

$$B_i = E_i + \rho_i \sum_{j=1}^{j=N} F_{ij} B_j.$$

Cette équation exprime simplement que le flux lumineux B_i émis en chaque point d'un réflecteur diffus est la somme du flux produit par auto-émission (dans le cas d'une source) et des fractions de flux reçu de toutes les autres surfaces ($F_{ij} B_j$) multipliées par le

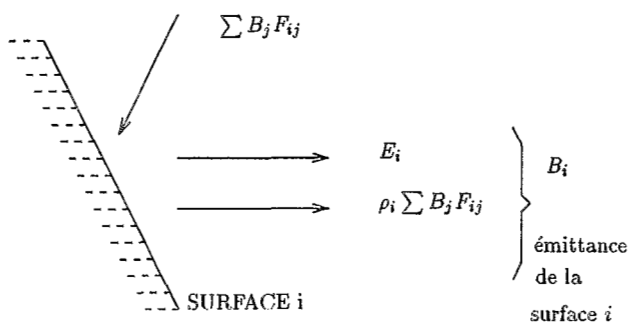


FIG. 5 - Réflexion diffuse globale

coefficient de réflexion ρ_i , ce dernier terme représentant le flux émis par réflexion (figure 5). La somme de ces deux flux par unité de surface donne l'émittance (ou radiosité) de la surface.

En écrivant cette relation pour chacun des carreaux, nous obtenons un système d'équations linéaires qui exprime les conditions d'équilibre énergétique global. Ce système peut s'écrire sous la forme :

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & \rho_2 F_{2N} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & \dots & -\rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

La synthèse d'image selon ce principe peut donc se décomposer en quatre étapes :

- subdivision des surfaces en carreaux élémentaires ;
- calcul des facteurs de forme :
- résolution du système linéaire :
- application d'une technique de rendu (Z-Buffer ou lancer de rayon).

Dans le cas d'une approche spectrale, il y a autant de systèmes d'équations de radiosité que de longueurs d'onde échantillons. Ce nombre se réduit à 3 dans le cas d'une approche trichromatique, ρ_i représente alors dans ce cas la couleur d'un objet.

Remarquons que ce n'est qu'à la dernière étape que nous avons besoin du point de vue, ce qui permet de restituer rapidement l'image en cas de modification de la vue.

Une des caractéristiques importantes de la méthode de radiosité est l'indépendance du point de vue pour l'essentiel des calculs. Ne dépendent du point d'observation que les calculs de projection, d'élimination des parties cachées et d'ombrage (luminance des pixels). Tous ces traitements peuvent être exécutés en moins d'une seconde sur la plupart

des stations de travail graphiques de hautes performances. L'exploitation des résultats est donc possible dans des applications interactives ne nécessitant que les déplacements du point de vue, ce qui est beaucoup plus difficile avec la technique du lancer de rayon classique.

Différentes expérimentations ont démontré [17] la précision et la fidélité du rendu en comparant des scènes réelles à leurs équivalents simulés.

3.3.2 Radiosité Progressive

L'inconvénient de la méthode de radiosité présentée ci-dessus est qu'elle nécessite le calcul de tous les facteurs de forme. Ceci est donc très coûteux, aussi bien en temps qu'en espace mémoire. A titre indicatif, pour une scène composée de 100000 carreaux, la mémoire nécessaire est de 40 giga-octets pour des réels codés en simple précision. Pour remédier à cela, Cohen et al. ont proposé une nouvelle formulation de l'algorithme de radiosité, dite par raffinement progressif [9]. Dans la méthode classique, on collecte pour chaque carreau l'énergie qu'il reçoit des autres carreaux. Par contre, la radiosité par raffinement progressif consiste à inverser le processus. On calcule alors la radiosité de tous les carreaux j due à l'émission du carreau i :

$$B_j = \rho_j B_i F_{ij} \frac{A_i}{A_j} \quad (2)$$

L'algorithme consiste alors à sélectionner un carreau et à distribuer son énergie aux autres carreaux en appliquant l'équation (2). Pour cela, on attribue à chaque carreau, en plus de sa radiosité, une delta radiosité correspondant à l'énergie qu'il possède mais qu'il n'a pas encore émis. Initialement, la delta radiosité est égale à l'auto-émittance E_i . En théorie, le programme itère jusqu'à ce que tous les carreaux aient une delta radiosité nulle. En pratique, on décide que la convergence vers la solution exacte a été atteinte lorsque l'énergie restant à émettre est inférieure à un seuil donné. La vitesse de convergence dépend beaucoup des propriétés réfléchives de la scène : plus ρ_j est proche de 1, plus le nombre d'itérations est élevé. Toutefois, comme aucun matériau ne réfléchit 100% de l'énergie qu'il reçoit, l'algorithme converge obligatoirement. Par ailleurs, la convergence sera plus rapide si on fait émettre en priorité les carreaux possédant la plus forte énergie. Cette méthode ne nécessite le stockage que d'une colonne de facteurs de forme, et permet de lancer un calcul d'image à chaque itération, ce qui se prête bien à une approche interactive.

Plusieurs algorithmes parallèles de la radiosité progressive ont été mis en œuvre tel que le montre la section 5.2.

3.4 Lancer de Rayon et Radiosité

La méthode de radiosité se limite à la réflexion diffuse seulement. L'extension de cette approche au cas de la réflexion spéculaire ne peut donner qu'une solution partielle. Une alternative intéressante consiste à compléter les calculs d'émittance par un post-traitement à base de lancer de rayon pour obtenir une précision suffisante dans le rendu des reflets [34, 28, 30, 6, 29, 5]. En effet, La mise en œuvre d'un modèle d'illumination global peut être réalisée en deux phases. La première consiste à appliquer la méthode de radiosité pour

évaluer la composante diffuse globale de la luminance d'un point de la scène. La deuxième évalue par lancer de rayon la composante spéculaire globale.

La qualité du rendu peut encore être améliorée en tenant compte des caractéristiques d'absorption et de diffusion des milieux traversés par la lumière (fumée, brume ...) [27].

3.5 Discussion

Ces travaux montrent l'importance de la modélisation des phénomènes lumineux dans la synthèse d'image réaliste. En s'appuyant de plus en plus sur des modèles inspirés de la photométrie et de la physique, ce domaine de la synthèse d'image prend une orientation résolument scientifique. On est en mesure de simuler d'une manière précise l'éclairage intérieur ou extérieur d'un ensemble architectural. Le seul inconvénient est le temps de calcul qui reste très élevé. Mais l'augmentation continue de la puissance de calcul des ordinateurs et l'avènement de machines parallèles sur le marché pallieront cet inconvénient dans un futur proche.

4 Visualisation

Comme il a été mentionné précédemment le processus de synthèse d'image fournit pour chaque pixel la luminance d'un point de la scène vue par l'observateur à travers ce pixel. Cette luminance dépend de la longueur d'onde (notons que dans le cas d'une approche trichromatique 3 longueurs d'onde sont utilisées : Rouge, Vert, Bleu). Appelons cette luminance $L(\lambda)$, λ étant la longueur d'onde. Rappelons que le spectre visible correspond à l'intervalle des longueurs d'onde $[380nm, 780nm]$. Pour pouvoir visualiser ces luminances sur un écran vidéo il va falloir les convertir en 3 composantes Rouge, Vert, Bleu (R,V,B).

Cette conversion est donnée par :

$$R = \int_{380nm}^{780nm} L(\lambda)\bar{r}(\lambda)d\lambda$$

$$V = \int_{380nm}^{780nm} L(\lambda)\bar{v}(\lambda)d\lambda$$

$$B = \int_{380nm}^{780nm} L(\lambda)\bar{b}(\lambda)d\lambda$$

Les fonctions $\bar{r}(\lambda)$, $\bar{v}(\lambda)$, $\bar{b}(\lambda)$ sont appelées fonctions de mélange trichromatiques. Elles dépendent du moniteur utilisé.

Pour pouvoir à la fois calculer précisément les luminances et les couleurs RVB associées, il est important de considérer un nombre de longueurs d'onde suffisant et des valeurs de longueur d'onde adéquates tel qu'il est montré dans [18].

Chaque composante R, V, B est codée sur 8 bits. Une composante a donc une valeur comprise entre 0 et 256. Le calcul de ces composantes pose en fait un problème qui reste insoluble. En effet, la luminance maximale affichée sur un moniteur est de l'ordre de 100 candelas par mètre carré alors que la luminance du ciel peut être de l'ordre de quelques milliers de candelas par mètre carré. Ce qui se traduit par des valeurs de R, V et B

dépassant largement 256. La solution la plus simple, adoptée couramment, est de tronquer les valeurs à 256 ou effectuer une mise à l'échelle. Cette solution donne des résultats très insuffisants. Une autre solution, plus intéressante, est de prendre en compte les problèmes de perception visuelle [32].

5 Parallélisme

5.1 Le Lancer de Rayon

La parallélisation de la technique du lancer de rayon a fait l'objet de plusieurs travaux de recherche. Parmi les différents types d'architecture de machine parallèle, c'est l'architecture MIMD qui a été retenue vue la structure très complexe de l'algorithme du lancer de rayon. Puisque l'objectif est de produire des images de scènes très complexes (plusieurs dizaines de milliers d'objets), la base de donnée décrivant la scène doit être impérativement répartie sur tous les processeurs afin de réduire le coût de l'algorithme de lancer de rayon implanté dans chaque processeur et de satisfaire aux contraintes de limitation de la taille mémoire de chaque processeur d'une architecture MIMD.

Il existe principalement deux techniques de répartition de la base de données : la première utilise la hiérarchie de volumes englobants associée à la scène, et la deuxième repose sur une subdivision spatiale de l'espace de la scène.

Une fois la hiérarchie de volumes englobants construite, la première technique duplique le haut de cette hiérarchie dans chaque processeur. Appelons *HDH* le haut de cette hiérarchie. Une feuille *F* de *HDH* est en fait un pointeur vers le processeur chargé du sous-arbre (de toute la hiérarchie) dont la racine est *F*. De cette manière, chaque processeur contient dans sa mémoire locale : *HDH*, un sous-arbre de la hiérarchie et la partie de la base de donnée associée à ce sous-arbre. Lorsqu'un processeur doit traiter un rayon, il parcourt la structure *HDH*. Si ce parcours conduit à un pointeur vers ce processeur, ce dernier effectue le calcul d'intersection entre ce rayon et sa base de donnée locale, sinon il le transmet au processeur concerné. Le parcours de la structure *HDH* permet donc à chaque processeur de connaître le processeur chargé du traitement d'un rayon.

La deuxième technique de répartition de la base de donnée consiste à effectuer une subdivision spatiale de la scène en régions *3D*, chacune d'elles est affectée à un processeur et ne contient qu'une partie de la base de donnée. Des processeurs voisins contiennent des régions adjacentes et communiquent entre eux localement par messages.

La première technique de répartition offre l'avantage de requérir une capacité mémoire beaucoup moins importante comparativement à la deuxième technique. Mais cet avantage est obtenu au dépens d'un coût de calcul plus important dû essentiellement au parcours de la structure *HDH* et des sous-arbres associés aux bases de donnée locales.

C'est pour cette raison que beaucoup de travaux se sont orientés vers des algorithmes parallèles utilisant une subdivision spatiale. Puisque la subdivision spatiale nécessite une taille mémoire importante, la base de donnée doit donc être répartie sur l'ensemble des processeurs.

Certains algorithmes ont été mis en œuvre sur la machine *iPSC* [4, 22, 23] et sont basés sur la répartition de la base de données, décrivant la scène, sur tous les processeurs, ainsi

que sur le fait que les communications inter-processeurs s'effectuent par messages. L'espace de la scène est subdivisé en autant de régions que de processeurs. A chaque processeur est affectée une région dont il est responsable. Quand un rayon quitte la région associée à un processeur, il le transmet au processeur chargé de la région suivante dans la direction du rayon.

L'importance et l'originalité de ces algorithmes ont été de mettre en évidence les vrais problèmes liés à une machine à mémoire distribuée. Ces problèmes concernent l'équilibrage des charge des processeurs, l'interblocage, la terminaison et le placement de tâches.

5.2 La radiosité

Les algorithmes de radiosité parallèles sont difficiles à classer car de nombreux critères peuvent être retenus: architecture visée (SIMD, MIMD) et type de mémoire (partagée ou distribuée), type de parallélisme (par le contrôle, par les données, systolique), niveau auquel le parallélisme est exploité, méthode employée pour le calcul des facteurs de forme, etc... Le problème le plus important rencontré lors de la parallélisation d'un algorithme de radiosité est la contrainte d'accès aux données. En effet, le calcul des facteurs de forme d'un carreau et la mise à jour des radiosités nécessitent potentiellement l'accès à l'intégralité de la scène. La section suivante recense les approches envisageables pour lever cette contrainte. Un autre élément à prendre en compte est le niveau auquel le parallélisme est exploité dans l'algorithme. En effet, un gros grain permet de diminuer les coûts de gestion mais génère moins de parallélisme qu'un grain plus fin. La section 5.2.2 examine les différents niveaux de parallélisme que l'on peut extraire de l'algorithme de radiosité.

5.2.1 Architectures et Schémas de Parallélisation

Une Architecture Parallèle à Mémoire Partagée (APMP) permet de supprimer la contrainte d'accès aux données. Dans ces machines, chaque processeur est relié aux mémoires locales par un réseau d'interconnexion. L'espace d'adressage étant unique, les processeurs ont alors accès à l'intégralité des données composant la scène. Il ne reste alors à résoudre que le problème de la mise à jour parallèle des variables partagées ce qui peut être fait en utilisant des sections critiques. Baum et al. [1] ont obtenu de bon résultats sur une telle machine en utilisant les z-buffers cablés dont elle disposait. Malheureusement, les APMP ne sont pas disponibles avec un grand nombre de processeurs car la capacité des bus, reliant les processeurs, est limitée.

Les Architectures Parallèles à Mémoire Distribuée (APMD) n'ont pas ce problème. Ce sont des machines facilement extensibles, autorisant un parallélisme massif. Par contre, les processeurs ne peuvent accéder qu'à leur mémoire locale. Pour résoudre la contrainte d'accès aux données, trois approches peuvent être envisagées.

La première approche consiste à dupliquer les données dans chacune des mémoires locales des processeurs. La taille des scènes que l'on peut traiter est alors limitée à la taille de la mémoire disponible sur chaque noeud. Dans ce cas, une approche maître-esclave est souvent utilisée [8, 16, 25]. En effet, la scène étant dupliquée, chaque modification (mise à jour des radiosités, modification dans la subdivision en carreaux, changement de la géométrie de la scène, ...) doit être effectuée sur chaque processeur, ce qui génère un

trafic de messages extrêmement important. Il est donc plus raisonnable de spécialiser l'un des processeurs (le maître) et de lui confier le maintien de la cohérence de la scène. Les autres processeurs (les esclaves) se contentent alors d'exécuter les travaux commandés par le maître et de lui retourner les résultats.

La seconde approche répartit équitablement les données : chaque processeur dispose alors dans sa mémoire locale d'une partie de la scène. Deux schémas de parallélisation peuvent alors être envisagés.

Le premier schéma, appelé parallélisation par le contrôle, consiste à distribuer les boucles. Un ensemble de tâches est créé, chacune prenant en charge un sous-ensemble de l'itération. Pour mener à bien sa tâche, un processeur peut avoir besoin de données qu'il ne possède pas. Il doit alors demander au processeur qui possède la donnée manquante de lui en envoyer une copie. Dans cette approche, la gestion des communications est un élément important. Elle peut être prise en charge par l'utilisateur [7], ce qui complique sensiblement la mise en œuvre, ou peut être gérée par le système : c'est le concept de mémoire virtuelle partagée. Le deuxième schéma de parallélisation est obtenu en faisant exécuter par un processeur les calculs qui portent sur les données qu'il possède. C'est la parallélisation par les données. Dans le cadre de la radiosité on peut par exemple transmettre à chaque processeur une surface de projection sur laquelle il projette les carreaux qu'il possède. Cette approche est utilisée par [33] et [11] sur des machines SIMD.

La dernière approche, appelé parallélisme en pipeline ou parallélisme systolique, décompose l'algorithme original en un ensemble de fonctions. Chaque fonction est associée à un processeur et les données transitent alors de processeur en processeur [24].

5.2.2 Niveaux de parallélisation dans la radiosité

Une classification peut être obtenue en regroupant les algorithmes de radiosité parallèle en fonction du ou des niveaux auxquels le parallélisme est exploité :

- Niveau 1 : plusieurs carreaux émettent en parallèle. Chaque processeur a la charge d'un carreau émetteur dont il calcule les facteurs de forme. Trois cas peuvent alors se présenter :
 - a : tous les processeurs ont accès aux radiosités des carreaux. Dans ce cas, chaque processeur peut effectuer la distribution de l'énergie de son carreau émetteur et sélectionner le prochain carreau à faire émettre.
 - b : chaque processeur gère une partie des radiosités. Les facteurs de forme calculés par un processeur sont alors envoyés aux autres processeurs qui effectuent en parallèle la mise à jour des radiosités.
 - c : un seul processeur gère les radiosités. Les autres processeurs se contentent alors de calculer un vecteur de facteurs de forme et envoient ce vecteur au processeur maître qui se charge de mettre à jour les radiosités et de déterminer les prochains carreaux émetteurs.
- Niveau 2 : les processeurs coopèrent pour calculer les facteurs de forme d'un carreau donné,

– Niveau 3 : les delta facteurs de forme sont calculés en parallèle.

Ces trois niveaux correspondent en fait aux boucles que l'on trouve dans l'algorithme de radiosité progressive. Le grain de parallélisme obtenu découle directement du niveau retenu. Il faut noter que la parallélisation au niveau 1 modifie la sémantique de l'algorithme séquentiel dans la mesure où l'ordre d'émission des carreaux diffère suivant le nombre de processeurs employés. Par ailleurs, la sélection des carreaux émetteurs nécessite une "vision globale" des radiosités des carreaux. C'est donc le niveau 1-c qui est le plus souvent mis en œuvre [8, 25, 16]. Cependant, Chalmers et al. [7] exploitent le niveau 1-b sans avoir cette vision globale. Chaque processeur sélectionne son carreau émetteur parmi les carreaux qu'il possède, ce qui diminue la vitesse de convergence. Le niveau 3 n'existe que si un z-buffer est utilisé pour le calcul des facteurs de forme. Baum et Winget [1] exploitent ce niveau en utilisant le pipeline de processeurs composant le z-buffer câblé de la machine qu'ils utilisent.

5.2.3 Discussion

La solution qui consiste à dupliquer les données de la scène dans chacune des mémoires locales des processeurs, si elle est inévitable lorsqu'on vise un réseau de stations de travail n'est pas appropriée dans le cadre d'une architecture parallèle. L'idéal est de disposer d'une machine MIMD sur laquelle est implantée une MVP (mémoire virtuelle partagée) telle que KOAN [15]. Un tel dispositif permet d'alléger sensiblement le travail de l'utilisateur puisqu'il le décharge de l'accès aux objets distants. Un autre élément à souligner est le coût d'exploitation du parallélisme. Pour limiter ceux-ci, il est indispensable de privilégier un gros grain, même si celui-ci a des répercussions sur les possibilités d'équilibrage de charge entre les processeurs. Une mémoire virtuelle partagée étant bien adaptée à une parallélisation par le contrôle, un choix intéressant est de paralléliser la boucle d'émission des carreaux afin d'avoir le grain le plus élevé. Par ailleurs, il est préférable de rejeter les approches qui conduisent à un goulot d'étranglement potentiel. C'est donc le niveau 1-a qui peut être retenu dans un premier temps : chaque processeur effectue seul la totalité de l'itération. Ceci est rendu possible grâce à la MVP qui permet d'avoir, sur chaque processeur, une vision globale et cohérente des données réparties. Cependant, le coût du maintien de cette cohérence peut amener à effectuer la mise à jour des radiosités en parallèle, ce qui correspond au niveau 1-b de notre classification.

6 Applications

6.1 Synthèse d'Image

La synthèse d'image peut être utilisée comme un outil permettant de produire des images spectaculaires de scènes tridimensionnelles à des fins de production de films ou publicité, à des architectes pour visualiser l'immeuble qu'ils ont conçu. Dans ce cas les modèles utilisés (d'illumination, de réflexion...) sont simples et empiriques.

Par contre, elle peut être utilisée en tant qu'outil de simulation. Dans ce cas elle constitue un outil de simulation d'éclairage. Les applications dans le domaine de l'éclairagisme

d'un logiciel de simulation d'éclairage sont multiples : décoration intérieure (influence des matériaux sur l'ambiance lumineuse), vérification de la conformité aux normes en matière de confort visuel pour les bâtiments publics (écoles, bureaux, ateliers, musées), optimisation de l'éclairage dans les locaux de grande dimension (salles de sport, grandes surfaces de vente) et l'éclairage urbain (éclairage routier efficace sans risques d'éblouissement, éclairage de bâtiments et édifices historiques). Les travaux en synthèse d'image peuvent facilement s'appliquer à d'autres types de rayonnement non visibles tels que le rayonnement infra-rouge.

6.2 Animation

Trois types d'application existent :

- les applications commerciales,
- la simulation,
- Les applications scientifiques.

Le but recherché par une application commerciale est la qualité artistique liée à la séquence d'images générées. Le réalisme du mouvement est secondaire, l'attrait visuel est primordial pour le public. Comme applications citons : films publicitaires, génériques d'émissions de télévision, réalité virtuelle... Ces applications sont utilisées par des graphistes ou artistes.

Quant aux logiciels de simulation, ils utilisent à la fois l'image et le mouvement. L'objectif dans ce cas est de permettre la visualisation des mouvements calculés par ordinateur. Les résultats de simulation, effectués généralement à l'aide de calculs mathématiques complexes et coûteux en temps de calcul, fournissent de nombreuses informations qui ne peuvent être analysées à la seule vue de valeurs numériques défilant sur un écran. L'image permet par contre d'avoir une bonne vision globale du phénomène calculé. Ce type d'application recouvre des domaines comme l'analyse de structures mécaniques (véhicules, structure mécanique...), les simulateurs de véhicule et d'avion...

Les applications scientifiques sont souvent destinées à la représentation de phénomènes invisibles à l'œil humain tels que la représentation de molécules ou les organes humains

7 Conclusion

Ce tour d'horizon sur la synthèse d'image a permis de montrer que ce domaine est passé d'un stade empirique à celui des outils et méthodes rigoureux utilisés dans diverses branches de la science. En effet elle s'appuie sur l'informatique, la physique, la perception visuelle, la photométrie, la colorimétrie, les mathématiques... Autant de domaines qui ne peuvent que témoigner de l'intérêt et de la richesse scientifique de ce merveilleux domaine de recherche qu'est la synthèse d'image.

Références

- [1] Daniel R. Baum and James M. Winget. Real time radiosity through parallel processing and hardware acceleration. *ACM Workshop on interactive 3D Graphics*, pages 67–75, mar 1990.
- [2] Pierre Bézier. Mathematical and practical possibilities of unisurf. *Computer Aided Geometric Design*, 1974.
- [3] K. Bouatouch, M.O Madani, T. Priol, and B. Arnaldi. A new algorithm of space tracing using a CSG model. In *EUROGRAPHICS'87 Conference Proceedings*, pages 65–78, Amsterdam, The Netherlands, August 1987.
- [4] K. Bouatouch and T. Priol. Parallel space tracing : An experience on an ipsc hypercube. In *Computer Graphics International'88 Proceedings*, pages 170–188. Computer Graphics Society, May 1988.
- [5] Kadi Bouatouch and Pierre Tellier. A two-pass physics-based global illumination model. In *Proceedings of Graphics Interface 92*, Vancouver, Canada, May 1992.
- [6] C. Bouville and K. Bouatouch. A unified approach to global illumination models. In *PIXIM'89 Conference*, pages 250–263, September 1989.
- [7] Alan G. Chalmers and Derek J. Paddon. Parallel processing of progressive refinement radiosity methods. In *Second Eurographics Workshop on Rendering*, volume 4, Barcelone, may 1991.
- [8] Shenchang Eric Chen. A progressive radiosity method and its implementation in a distributed processing environment. Master's thesis, Cornell University, 1989.
- [9] Michael F. Cohen, Shenchang E. Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics*, 22(4):75–84, August 1988.
- [10] S. Donikian. *Une approche déclarative pour la création de scènes tridimensionnelles : application à la conception architecturale*. PhD thesis, Univ. de Rennes I, Rennes, 1992.
- [11] Steven M. Drucker and Peter Schröder. Fast radiosity using a data parallel architecture. *Third Eurographics Workshop on rendering*, 1992.
- [12] J.D. Foley, A. Van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics, Principles and Practice*. Addison Wesley Publishing Company, London, 1990.
- [13] G. Hégron and B. Arnaldi. Computer animation: Motion and deformation control. In *Eurographics'92 tutorial*, 1992.
- [14] J. Goldsmith and J. Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics & Applications*, pages pp 14–20, 1987.

- [15] Z. Lahjomri and T. Priol. Koan: a shared virtual memory for the ipsc/2 hypercube. In *CONPAR/VAPP92*, September 1992.
- [16] Eric Lepretre, Christophe Renaud, and Michel Meriaux. La radiosit  sur tranputers. *La lettre du tranputer et des calculateurs distribu s*, pages 49–66, dec 1991.
- [17] G. H. Meyer, H. Rushmeier, M. Cohen, D. Greenberg, and K. Torrance. An experimental evaluation of computer graphics imagery. *ACM transactions on graphics*, 5(1):30–50, January 1986.
- [18] Gary W. Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41:57–79, 1988.
- [19] M.F. Cohen and J.R. Wallace. Academic Press Professional, 1993.
- [20] M. E. Mortensen. *Geometric Modeling*. John Wiley & Sons, 1985.
- [21] Les Piegl. On nurbs: A survey. *IEEE Computer Graphics & Applications*, pages pp 55–71, 1990.
- [22] T. Priol and K. Bouatouch. Experimenting with a parallel ray-tracing algorithm on a hypercube machine. In *EUROGRAPHICS'88 Conference Proceeding*, pages 243–259, September 1988.
- [23] T. Priol and K. Bouatouch. Static load balancing for a parallel ray tracing on an MIMD hypercube. *The Visual Computer*, 5:109–119, Mars 1989.
- [24] Werner Purgathofer and Michael Zeiller. Fast radiosity by parallelization. *Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 173–184, jun 1990.
- [25] Rodney J. Recker, David W. George, and Donald P. Greenberg. Acceleration techniques for progressive refinement techniques. *ACM Workshop on Interactive 3D graphics*, pages 59–66, mar 1990.
- [26] A.A. Requicha. Representation for rigid solids : Theory, methods, and systems. *ACM Computing Surveys*, 12(4):437–464, December 1980.
- [27] H. K. Rushmeier and K. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics*, pages 293–302, August 1987.
- [28] M. Shao, Q. Peng, and Y. Liang. A new radiosity approach by procedural refinements for realistic image synthesis. *Computer Graphics*, 22(4):93–101, August 1988.
- [29] F. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics*, 25(4):187–196, August 1991.

- [30] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335-344, July 1989.
- [31] F. Sillion and C. Puech. Morgan Kaufmann Publishers, 1994.
- [32] Jack Tumblin and Holly E. Rushmeier. Tone reproduction for realistic computer generated images. Tech. Report GI -GVU-91-13, Graphics, Visualization & Usability Center, Coll. of Computing, Georgia Institute of Tech., 1991.
- [33] Amitabh Varshney and Jan F. Prins. An environment projection approach to radiosity for mesh-connected computers. *Third Eurographics Workshop on rendering*, 1992.
- [34] J. R. Wallace, M. F. Cohen, and D. P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics*, 21(4):311-320, August 1987.
- [35] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23:343-349, June 1980.