



Synthèse d'images par lancer de rayon : algorithmes et architecture

Ray-Casting in image synthesis : algorithms and architecture

CH. BOUVILLE, R. BRUSQ, J.L. DUBOIS, I. MARCHAL

C.C.E.T.T. B.P 59 - 35510 CESSON-SEVIGNE

RESUME

La synthèse d'image par lancer de rayon connaît depuis peu un développement spectaculaire essentiellement dans le domaine de la production audiovisuelle et de la C.A.O. L'avantage majeur de cette méthode, est de procurer un cadre très général dans lequel de nombreux problèmes peuvent être résolus simplement.

Après un rappel des principes de la synthèse d'image par lancer de rayon, cette communication aborde le problème du modèle photométrique en cherchant à exploiter au mieux les fondements optiques de cette approche. A titre d'exemple, il est également décrit une méthode de visualisation des surfaces de révolution avec des possibilités de déformation.

Ce procédé de synthèse d'image est encore difficilement exploitable en production audiovisuelle à moins de disposer d'un ordinateur très puissant. Cependant, les algorithmes de synthèse d'image par lancer de rayon se prêtent très bien au traitement parallèle, et, dans la suite, il est montré qu'une machine d'architecture appropriée peut pallier à cet inconvénient tant sur le plan des coûts que celui des performances

SUMMARY

Ray-casting in image synthesis has recently known impressive developments in the area of audiovisual production and C.A.D. The major advantage of this method is to provide a very general framework in which many problems can be easily solved. After a recall of the ray-casting principles, this paper approaches the lighting model question trying to take advantage of the optical basis of this technique. As an example, a method for displaying surfaces of revolution is also described.

Ray-casting techniques are still difficult to work in audiovisual applications without a powerful computer. However, ray-casting algorithms lend themselves quite well to parallel processing and, in the following, it is shown that an appropriate machine architecture can palliate these drawbacks from the cost as well as the performance point of view.



I - INTRODUCTION

La synthèse d'images par lancer de rayon est sans conteste à l'origine des images les plus étonnantes qui aient jamais été réalisées en infographie. Le principe est pourtant connu depuis longtemps puisqu'il a été utilisé dès 1950 en optique pour la conception des lentilles mais c'est en 1968 que cette technique a été utilisée pour la première fois en infographie [1]. Comparativement à la plupart des algorithmes utilisés en synthèse d'images, cette méthode est relativement facile à mettre en oeuvre mais cette simplicité est obtenue aux dépens du temps de calcul qui est sensiblement plus long que celui des méthodes classiques.

Au premier abord, on serait donc tenté d'écarter à priori cette méthode lorsqu'il n'est pas nécessaire de produire certains effets (reflexions multiples, transparence, ombres portées) pour lesquels il est indispensable d'y recourir. La réalité est en fait plus complexe car pour comparer l'efficacité des algorithmes, il faut non seulement faire intervenir la qualité des images produites mais aussi le modèle géométrique à partir duquel l'image est générée. En effet, de récents travaux ont montré, qu'en modélisation de solide, les résultats obtenus étaient comparables sinon meilleurs que ceux des procédés classiques [2], [3], [4].

II - PRINCIPES

1) Algorithmes de base

Le principe de la synthèse d'images par lancer de rayon est extrêmement simple : il consiste à reconstruire l'image projetée d'une scène en analysant le parcours des rayons lumineux parvenant sur la fenêtre de visualisation matérialisée par l'écran (Fig. 1). Pour une image numérique, il suffit donc en principe de suivre le trajet inverse des rayons lumineux en partant du point d'observation et en passant par chaque pixel de l'image d'où le nom de lancer de rayon ("ray-casting ou ray-tracing en anglais").

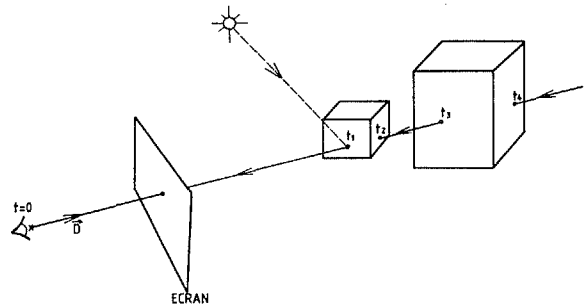


FIG. 1 Principe du lancer de rayon

Pour chacun de ces rayons, la première surface intersectée sera la surface visible. Il suffit donc de calculer la luminance en chacun des points d'intersection visibles pour générer l'image. En fait, le calcul de la luminance nécessite lui-même le lancer d'autres rayons comme nous le verrons par la suite. Ces rayons auront la particularité d'avoir comme point de départ le point d'intersection considéré. Nous les appellerons rayons secondaires pour les distinguer des rayons primaires qui sont issus du point d'observation.

Les algorithmes de base en synthèse d'images par lancer de rayon sont donc essentiellement des algorithmes de calcul d'intersection de droite avec des surfaces. Ces calculs occupent la majeure partie du temps de traitement et ils doivent donc faire l'objet d'une attention toute particulière.

Dans ces calculs, l'équation du rayon s'écrit :

$$x^i = tx_0^i + D^i \quad \text{où } i = 1, 2, 3, t \geq 0$$

$M \begin{bmatrix} x_1^i \\ x_2^i \\ x_3^i \end{bmatrix}$ est un point du rayon, $O \begin{bmatrix} 1 \\ x_0^1 \\ x_0^2 \\ x_0^3 \\ x_0^0 \end{bmatrix}$ est l'origine du rayon

et $\vec{D} \begin{bmatrix} D^1 \\ D^2 \\ D^3 \end{bmatrix}$ est le vecteur directeur du rayon

Le problème revient à calculer la valeur de t pour chacun des points d'intersection. Pour les rayons primaires, le point visible sera celui qui a la valeur de t la plus petite.



La plupart des algorithmes d'intersection n'exploite pas la cohérence c'est-à-dire le fait qu'à deux pixels contigus, il correspond généralement deux valeurs de t voisines pour les intersections des rayons primaires et secondaires. Le seul algorithme connu exploitant la cohérence est décrit en [3]. Son principe repose sur un balayage ligne par ligne de l'écran. Les résultats obtenus sont remarquables cependant il ne s'applique qu'aux objets polygonaux et aux rayons primaires.

Pour un grand nombre de surfaces utilisées couramment en modélisation, il existe un repère pour lequel le calcul d'intersection est particulièrement simple. On définira donc un repère local pour chaque primitive et à chaque calcul d'intersection, on opérera un changement de repère sur l'équation du rayon. Comme la valeur de t est indépendante du repère, il n'est pas nécessaire d'opérer une transformée inverse.

L'application de ce principe permet de traiter très facilement le cas des objets à base de plans ou de surfaces quadriques (ellipsoïde, cône, cylindre) le cas des prismes et des surfaces de révolution est un peu plus complexe [5]. Nous verrons dans la suite comment représenter les surfaces de révolution déformables. Pour les surfaces bi-cubiques (Bézier, B-Spline...) il existe deux algorithmes : l'un repose sur les propriétés de subdivision des bi-cubiques [6], l'autre effectue un calcul direct des intersections en ramenant le problème à la recherche des racines d'un polynôme [7]. Le cas des surfaces algébriques peut être traité de manière similaire [8]. En ce qui concerne les surfaces fractales, Kajiyama [5] a développé un algorithme basé sur la subdivision récursive des triangles ne nécessitant pas un calcul préliminaire de tous les triangles composant la surface.

2) Composition des intersections

En modélisation de solide, les objets sont généralement composés de solides primitifs (paral-

lélépipède, cylindre, sphère, cône, etc...) assemblés par des opérateurs booléens ($\cup, \cap, -$). Les objets sont décrits par un arbre binaire où chaque noeud correspond à un opérateur d'assemblage et où les noeuds terminaux sont des solides primitifs. Ce type de modèle est relativement difficile à traiter autrement que par lancer de rayon où le problème est résolu très simplement en combinant les intersections par des opérations booléennes figurant les opérateurs d'assemblage [2]. Cette opération est indépendante du calcul d'intersection et la combinaison des intersections peut être appliquée à tous les solides quelque soit la méthode de calcul d'intersection. Des résultats très intéressants ont pu ainsi être obtenus pour le traitement de ce type de modèle ([2],[3]).

3) Structure de données

L'application brutale du principe de base conduit à des temps de calcul prohibitifs. Il existe plusieurs manières d'améliorer l'efficacité de la méthode. Nous avons vu que l'exploitation de la cohérence était une voie intéressante malheureusement elle ne s'applique qu'aux rayons primaires et aux surfaces polygonales. Or la génération d'images synthétiques de qualité introduit beaucoup plus de rayons secondaires que de rayons primaires.

De manière générale, c'est par une organisation appropriée de la base de données qu'on obtient les meilleurs résultats, le but étant de réduire au minimum les calculs d'intersection inutiles. La méthode la plus couramment utilisée ([9],[6],[2]) consiste à hiérarchiser la base de données en lui donnant une structure arborescente. A chaque noeud de l'arbre, on déterminera un volume d'extension approprié (parallélépipède, sphère ou ellipsoïde) englobant l'objet composite défini par ce noeud. Cette structure arborescente existe déjà en modélisation de solide : il s'agit en fait d'un arbre binaire dont les noeuds correspondent aux opérateurs booléens et les branches aux opérands. Pour les rayons primaires, ces volumes d'extension peuvent être directement projetés sur l'écran [2] ou encore, comme



exposé en référence [4], il est possible d'exploiter la cohérence en déterminant des intervalles d'inclusion pour chaque ligne de balayage.

4) Mise en oeuvre matérielle

La mise en oeuvre matérielle des algorithmes de base dépend beaucoup du type d'application recherché. Au C.C.E.T.T., où les études sont orientées vers la synthèse d'images pour l'audio-visuel et la télématique, le lancer de rayon est utilisé pour la production d'images très réalistes incluant les effets d'ombre portée, de réflexions multiples et de transparence avec réfraction. Ceci implique le calcul d'un nombre très important de rayons secondaires, or il est beaucoup plus difficile d'exploiter la cohérence pour les rayons secondaires que pour les rayons primaires et ceci aussi bien dans les calculs d'intersection que dans la détermination des volumes d'extension. Par contre, le traitement parallèle est particulièrement approprié puisque le calcul de chaque pixel peut être mené indépendamment des autres.

5) Principaux avantages

L'intérêt principal de cette méthode est de procurer un cadre très général dans lequel de nombreux problèmes peuvent être résolus et notamment :

- l'assemblage de solides par des opérateurs booléens,
- les réflexions multiples,
- la transparence avec réfraction,
- les ombres portées.

D'autre part, la représentation des surfaces par lancer de rayon ne nécessite aucune approximation polygonale et les fondements optiques de la méthode permettent d'exploiter des modèles photométriques très élaborés comme nous le verrons dans la suite. Ces deux avantages expliquent la très grande qualité des images produites par lancer de rayon.

III - MODELE PHOTOMETRIQUE

Quand un rayon primaire intersecte des objets de la scène, l'intensité du pixel correspondant se calcule à partir des propriétés optiques du premier des éléments de surface intersectés. Plus exactement, il s'agit de la luminance de cet élément, grandeur radiométrique qui mesure l'intensité énergétique dans une direction par une unité de surface. Les flux énergétiques renvoyés par une surface peuvent provenir soit d'une source lumineuse (ou plusieurs) soit des objets environnants qui, eux-mêmes, en renvoyant une partie de la lumière qu'ils reçoivent, se comportent comme des sources de lumière. De manière générale, le flux lumineux provenant d'une surface se décompose en un flux réfléchi et un flux transmis. Le flux réfléchi est dû aux propriétés réfléchissantes de la surface et le flux transmis à ses propriétés de transparence. Dans la suite, nous analyserons en détail comment évaluer ces deux composantes.

1) Flux réfléchi :

Dans le cas d'une source lumineuse, la luminance dL_r de l'élément de surface peut s'écrire de la manière suivante : $dL_r = RE_i$ où R est la réflectance bidirectionnelle liée à la nature du matériau et E_i représente l'éclairement, c'est-à-dire, la densité surfacique de flux reçu. Les calculs s'effectuent simultanément avec les trois composantes primaires : rouge, vert et bleu. Le flux réfléchi est dû à l'éclairage direct par les sources lumineuses mais aussi à l'effet des surfaces rayonnantes. L'éclairage direct est très bien déterminé par l'approche théorique de Torrance et Cook [11]. A ce niveau, on peut traiter deux problèmes assez simplement : celui des ombres portées et celui des corps transparents, colorés ou non, placés entre la source lumineuse et les objets. Pour le problème des ombres portées, il suffit de lancer un rayon, cette fois partant de l'élément de surface, vers la source lumineuse (Fig 2).

Si on détecte une intersection avec un corps opaque l'élément n'est pas éclairé par cette source. Avec ce

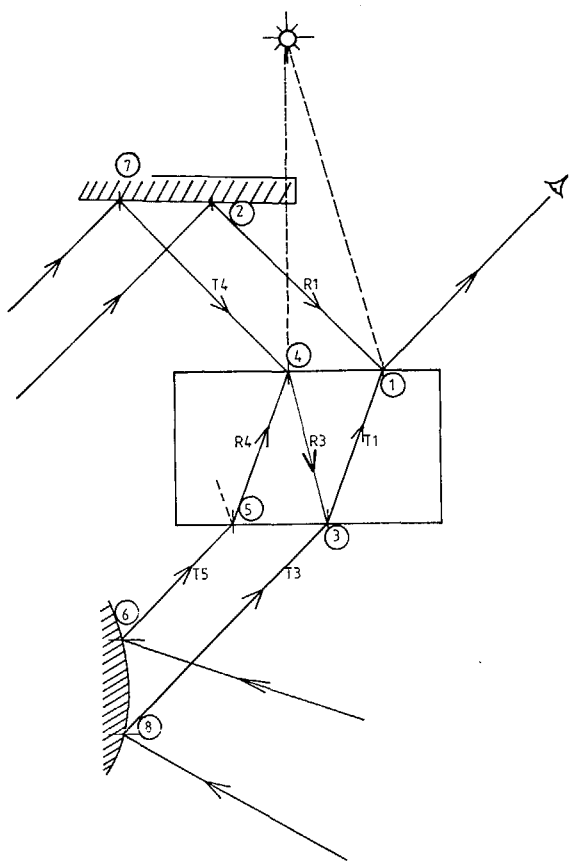


FIG. 2 Calcul des différentes composantes du flux lumineux

même lancer de rayon, on peut aussi détecter la présence d'un corps transparent. Dans ce cas, l'éclaircissement sera atténué en fonction de l'absorption du matériau (qui est sélective dans le cas coloré) mais aussi lors des transitions entre milieu d'indice différent. Il est à noter que le lancer de rayon, en calculant toutes les intersections successives, permet un calcul immédiat de l'épaisseur.

La réflectance bidirectionnelle permet une analyse des propriétés de réflexion des matériaux: aspect mat ou brillant, rugueux ou lisse, plastique, métallique, couleur...

La réflexion peut se faire selon une direction privilégiée, celle qui correspond à l'image réfléchie de la source éclairante. Cette réflexion dite spéculaire, s'observe particulièrement dans le cas des objets polis et métalliques. A l'opposé, les corps rugueux et mats renvoient la lumière qu'ils frappent dans toutes les directions. La réflexion est

dite diffuse et la surface a même apparence lumineuse quelle que soit la direction. La plupart des matériaux ne sont ni parfaitement diffusants, ni parfaitement spéculaires, mais présentent des caractéristiques intermédiaires. On peut ainsi écrire si s représente la part relative de la réflexion spéculaire, et d celle de la réflexion diffuse.

$$R = sR_s + dR_d \text{ avec } s + d = 1$$

Torrance et Sparrow [12] font intervenir la notion de microfacettes. Chaque surface d'un objet est constituée d'une multitude de microfacettes ayant chacune les propriétés d'un miroir et une orientation prenant des valeurs aléatoires autour d'une valeur moyenne qui correspond à l'orientation de la surface. La réflexion spéculaire correspond à la lumière qui est immédiatement réfléchiée sur les microfacettes. La réflexion diffuse prend son origine d'une part des réflexions multiples entre les microfacettes, surtout si le corps est rugueux, et d'autre part de la dispersion interne. Dans le cas d'un diffuseur parfait, la luminance reste constante quelle que soit la direction de la lumière incidente (loi de Lambert). Si R_d est donc une constante, la composante spéculaire de la réflectance bidirectionnelle est égale à [11].

$$R_s = \frac{F \cdot D \cdot G}{\pi \cos \theta \cos \phi}$$

où l'angle θ est l'angle d'incidence du flux énergétique sur la surface, et ϕ celui formé par la normale à la surface et la direction d'observation. La surface n'est pas un miroir parfait, seulement une partie du flux atteignant les facettes est réfléchiée. Le terme de Fresnel F , qui apparaît dans cette formule, permet de tenir compte de ce phénomène. Un corps est d'autant plus réfléchissant que sa valeur de F est proche de 1.

La fonction D permet d'attribuer un degré de rugosité à un matériau. Le principe est le suivant : un corps est d'autant plus lisse que les normales de ses microfacettes sont proches de la normale à la surface. Blinn propose 3 modèles dans [13]. La disposition géométrique des microfacettes peut être telle que certaines d'entre elles masquent le flux incident ou arrêtent le flux réfléchi de facettes adjacentes. La fonction d'atténuation G qui en



résulte a été étudiée par Torrance-Sparow [12] et Blinn [13].

Le calcul de la luminance due aux sources secondaires que sont les objets environnants reste en principe identique dans le cas de la lumière réfléchie, avec une composante diffuse et une composante spéculaire. Il est cependant préférable d'apporter quelques simplifications pour limiter le nombre de calculs. Il faut également remarquer que certaines approximations comme celles introduites par l'éloignement des sources, ne tiennent plus ici. D'abord, la somme des flux énergétiques provenant des divers objets et tombant sur l'élément de surface produit une composante diffuse sensiblement uniforme sur tout l'objet. Ensuite, on peut supposer, dans le cas spéculaire, que l'énergie provient surtout d'une direction privilégiée déterminée par les lois de l'optique géométrique :

- la direction d'incidence du flux est dans le plan formé par la normale à la surface et la direction d'observation.
- l'angle d'incidence est égal à l'angle formé par la normale et la direction d'observation.

2) Flux transmis

Le flux transmis se décompose également en deux composantes: celle due à l'éclairage direct par les sources lumineuses et celle due aux surfaces rayonnantes. Cependant, à la différence des corps opaques, la plupart des objets transparents rencontrés dans la nature sont généralement d'aspect très lisse. L'influence des sources directes est dans la plupart du temps très faible puis la lumière se diffuse très peu sur les surfaces lisses. Elle n'est en fait importante que dans les zones où la source est directement ou indirectement vue par l'observateur. Cette composante peut se calculer de manière tout à fait similaire à celle de la lumière réfléchie, il suffit de remplacer le coefficient de réflexion F par le coefficient de transmission $(1 - F)$.

La composante due aux surfaces rayonnantes peut s'évaluer en ne calculant que le flux provenant d'une direction privilégiée comme nous l'avons fait pour la réflexion.

Cette direction privilégiée est celle de la réfraction qui peut se calculer, elle aussi, par les lois de Snell. Il s'agit alors de lumière transmise et on doit retenir $1-F$ et non plus F à chaque passage d'un milieu à un autre. Il faut aussi dans ce cas tenir compte de l'absorption du corps transparent. A partir de l'élément de surface, on est donc amené à lancer un rayon réfléchi, et lorsque le corps est transparent, un rayon réfracté. S'il y a intersection avec un objet, on fait intervenir dans les calculs la luminance de ces points d'intersections. La connaissance de celle-ci oblige de nouveaux lancers de rayon. Pour un rayon initial provenant du point d'observation, un arbre binaire sera ainsi construit (Fig 3): les noeuds correspondront aux intersections et les branches correspondront au rayon réfléchi et au rayon réfracté. Les calculs devront ensuite remonter de proche en proche dans l'arbre jusqu'à la première intersection

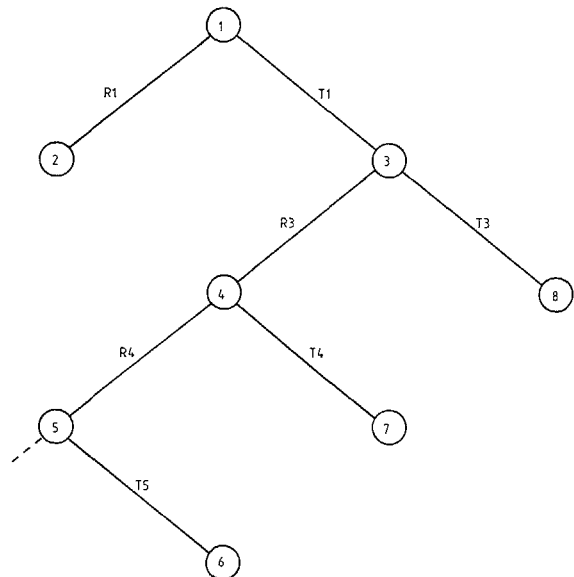


FIG 3 Arbre binaire de composition des flux réfléchis et transmis

IV - Visualisation des surfaces de révolution

1) Introduction

Parmi les nombreux objets représentés en modélisation de solide, les surfaces de révolution sont fréquemment utilisées soit comme solide à part entière, soit comme primitive en combinaison avec d'autres objets.

Rappelons tout d'abord que l'on désigne par surface de révolution, le volume engendré par la rotation d'une courbe génératrice (C) quelconque (ouverte ou fermée) autour d'un axe de révolution (fig 4). Suivant la forme de la courbe (C), ce volume est fermé par 1 ou 2 disques plans désignés par P1 et P2 sur la figure 4.

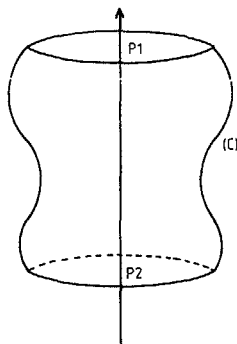


FIG 4 Une surface de révolution

Parmi les différents algorithmes développés pour traiter ce type d'objet et qui se ramènent souvent à des approximations par des patches ou des polygones, J. Kajiya [6] a présenté un algorithme qui exploite très bien la symétrie de l'objet, mais qui a pour inconvénient de nécessiter un prétraitement de la courbe génératrice. Ce prétraitement consiste à déterminer une autre représentation de la génératrice: le "strip-tree" dont la structure hiérarchique facilite le calcul d'intersection. Ceci est relativement gênant dans les applications audiovisuelles où l'on applique souvent des déformations aux objets, ce qui conduit à renouveler à chaque fois le calcul du "strip-tree". D'autre part la représentation des génératrices sous forme de "strip-tree" conduit à des volumes de données assez importants sauf si on se limite aux arcs de parabole. C'est pour éviter ces problèmes que nous avons développé un algorithme où la courbe est représentée à l'aide de sections cubiques.

2) Description de l'algorithme

2-1 Modélisation de la génératrice

De la même manière que pour chaque type de primitive, on définit la surface de révolution dans un repère local (Fig 6) dans lequel l'axe z est l'axe de révolution et le rayon maximal vaut 1. Chaque section de génératrice (C) est décrite de façon paramétrique dans un plan contenant l'axe de révolution par les équations :

$$r = r(u)$$
$$z = g(u)$$

ou $r(u)$ et $g(u)$ sont des polynômes d'ordre 3. La modélisation est de type B. Spline. Celle-ci présente en effet l'avantage de permettre la déformation locale des objets par le seul déplacement des points de contrôle. Dans cette représentation, chaque section de cubique est générée par 4 points de contrôle mais ces points sont partagés avec la cubique précédente et la cubique suivante produisant ainsi des courbes plus "lisses". Cependant lors des calculs d'intersection, il est préférable d'utiliser les courbes de Bézier qui permettent de calculer des volumes d'extension plus proches de l'objet, les 2 points de contrôle extrêmes se trouvant dans ce cas sur la courbe elle-même (Fig 5).

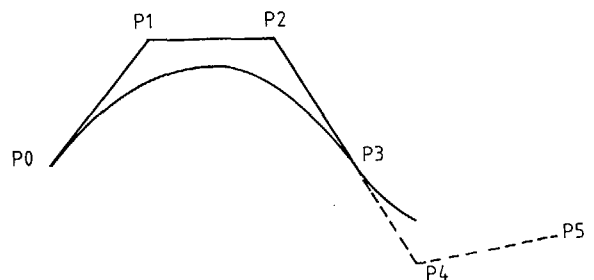


FIG 5 Une courbe de BEZIER

C'est pourquoi, nous utilisons la représentation B. Spline pour générer les courbes et nous convertissons ensuite les points de contrôle B. Spline en points de contrôle de Bézier ce qui revient à un changement de base effectué simplement à l'aide de matrices.

2-2 Introduction des surfaces de révolution dans la base de données.

Comme nous avons pu le constater, une surface de révolution peut-être décomposée en plusieurs



sections correspondant aux sections cubiques de la génératrice. On peut imaginer de combiner ces sections à l'aide d'un opérateur "union de surface" (+S) (à ne pas confondre avec l'opérateur U) qui assemble des solides.

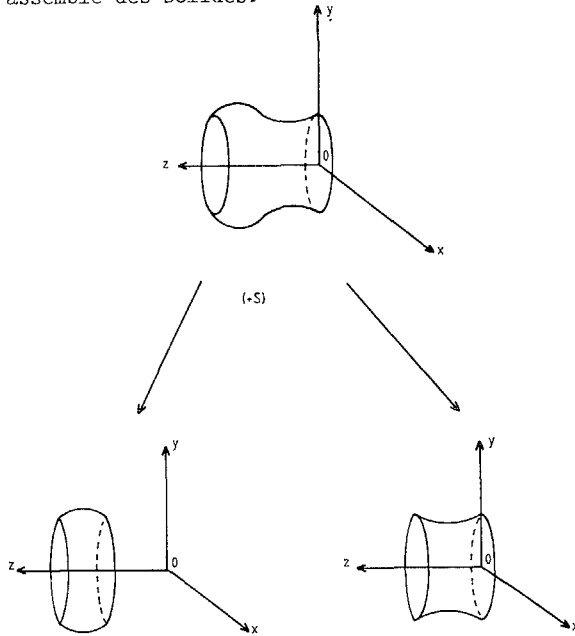


FIG 6 Composition des sections de surface de révolution

Afin de donner à la base de données, une structure d'arbre binaire très adaptée au "lancer de rayon" et permettre de réduire le nombre des calculs d'intersection, chaque surface de révolution est décomposée en un arbre binaire avec l'opérateur (+S) aux noeuds. Cet arbre est introduit dans l'arbre binaire CSG à la place occupée par une primitive. Des volumes d'extension sont calculés facilement à chaque noeud à l'aide des points de contrôle de Bezier.

2-3 Calcul d'intersection

Une section de surface de révolution est décrite sous forme paramétrique par les équations :

$$\begin{cases} x^2 + y^2 = r^2(u) \\ z = g(u) \end{cases} \quad 0 \leq u \leq 1$$

D'autre part, on a vu que le rayon est défini par une équation du type :

$$x^i = txo^i + D^i \quad (1)$$

En substituant x , y et z par leur valeur donnée dans (1), et en éliminant t des équations ainsi formées, on obtient une équation de la forme :

$$A_1 g^2(u) + A_2 g(u) + A_3 - r^2(u) = 0$$

Cette équation est une équation en u de degré 6 puisque $g(u)$ et $r(u)$ sont des polynômes d'ordre 3. En résolvant cette équation, on obtient des racines u_1, u_2, \dots, u_n appartenant à l'intervalle $[0, 1]$ qui vont nous permettre de déduire les valeurs de t correspondantes et donc les coordonnées des points d'intersection du rayon avec la section de surface de révolution. Ces points d'intersection sont classés par valeurs de t croissantes afin de déterminer le point visible qui est celui qui se trouve le plus proche de l'observateur.

Pour déterminer l'intensité du pixel selon le modèle d'illumination décrit précédemment il faut calculer la valeur de la normale au point d'intersection. Celle-ci est une droite qui passe par l'axe de révolution et qui peut-être déduite de la normale à la génératrice dans le plan (r, z) .

On peut écrire :

$$N = (x_i, y_i, -r_i \left[\frac{dr}{du} / \frac{dz}{du} \right]_{u = u_i})$$

où x_i, y_i, z_i sont les coordonnées du point d'intersection, $r_i = \sqrt{x_i^2 + y_i^2}$ et u_i est la valeur du paramètre u en ce point.

2-4 Calcul des racines de l'équation en u

Le choix de la méthode utilisée pour calculer les racines de l'équation de degré 6 est très important d'une part dans la perspective d'une implantation future en hardware et d'autre part afin de minimiser les temps de calcul. C'est pour cette raison que nous avons développé un algorithme qui procède en 2 étapes :

1- Trouver le nombre de racines réelles $\in [0, 1]$ et donner à chacune d'elles une valeur approximative. Cette étape est réalisée grâce à un algorithme décrit par Collins et Loos [14] basé sur la règle des signes de Descartes. Cet algorithme permet notamment de déterminer très rapidement si l'équation n'a pas de racines.

2- Calculer la valeur exacte des racines. Ceci est accompli au moyen de la méthode itérative de Laguerre qui est une méthode similaire à celle de Newton mais qui converge plus sûrement en particulier dans le cas des racines très voisines. La valeur de départ pour le processus itératif étant fourni par l'étape précédente, on a généralement besoin que de 3 à 5 itérations.



V - Architecture matérielle

1 - Introduction

Comme exposé plus haut, plusieurs modèles de mise en oeuvre matérielle des algorithmes "Ray-Casting" peuvent être implantés suivant le type d'application envisagé. Au C.C.E.T.T. nous nous plaçons dans un cadre de production d'images réalistes avec ombres portées, réflexions multiples, transparences, etc... Nous excluons des méthodes exploitant la cohérence au niveau des rayons primaires, et nous traitons globalement les rayons primaires et secondaires relatifs à chaque pixel; mais nous séparons au maximum les calculs de chaque pixel les uns des autres.

L'inconvénient d'une telle approche est le manque d'optimisation des calculs au niveau du pixel. Mais de cette manière, le problème est envisagé dans son ensemble et un minimum d'approximations sont introduites dans l'image produite, ce qui a des répercussions importantes sur la qualité.

2-1 Etude de l'implantation matérielle

2-1 Nature des calculs

Le processus de synthèse d'images par lancer de rayon présente une suite importante de calculs, et en particulier des calculs d'intersection de droite et de surfaces et des opérations de changement de repères. Ces calculs doivent être réalisés avec une précision suffisante pour éviter les dégradations dues aux accumulations d'erreurs d'arrondis. Par exemple, les valeurs de "t" aux différents points d'intersection devront être obtenues avec suffisamment de précision pour éviter les ambiguïtés de position relative de deux surfaces voisines. Cependant cette précision doit demeurer cohérente avec la résolution du support d'affichage (taille du pixel). Par ailleurs, l'excursion des valeurs est importante. Par exemple les coordonnées des points d'intersection dans le repère propre à la primitive sont comprises entre 0 et 1, alors qu'elles peuvent prendre des valeurs très grandes dans le repère lié à l'écran.

Pour toutes ces raisons, le calcul en flottant s'im-

pose. Nous n'avons pas déterminé avec exactitude la précision nécessaire, car celle-ci dépend de nombreux facteurs et en particulier de la nature des images elles-mêmes. Le calcul flottant standard simple précision (32 bits) semble une bonne base d'étude.

2-2 Choix d'architecture

Sans viser la synthèse en vue d'animation au rythme TV, notre souci est de ramener les temps de calcul d'une image dans des limites raisonnables (unités mn) pour des prix non prohibitifs. Or, pour diminuer le temps de calcul il faut soit augmenter la performance du calculateur réalisant les processus de synthèse, soit paralléliser des calculateurs de performance donnée, soit encore étudier des architectures spécifiques collant de très près à chacune des phases de l'algorithme de synthèse.

Dans la ligne "calculateur performant", le GRAY-1 est très utilisé aux USA pour réaliser la synthèse d'images par Ray-Casting. Les images obtenues sont saisissantes mais les temps de calcul et donc les coûts de production sont très importants. Dans la ligne "architectures spécifiques", un certain nombre de travaux peuvent être cités : [19], [20], [21].

Notre démarche au C.C.E.T.T. se situe dans la ligne "parallélisation de calculateurs de performance donnée" et en particulier mise en parallèle de microprocesseurs. Si une telle approche peut paraître aujourd'hui peu performante, on peut cependant penser que, l'évolution technologique aidant, les performances des microprocesseurs allant grandissant, ce type de matériel multimicroprocesseurs donnera rapidement des résultats intéressants.

2-3 Mise en oeuvre de la synthèse d'image par Ray-Casting sur une architecture multi-microprocesseurs

2-3-1 Découpe fonctionnelle du processus général de synthèse (Figure 8)

2-3-1-1 Structure de données

Le processus général de synthèse réalise l'affichage d'une image à partir d'une base de données.

Comme exposé plus haut, la base de données est

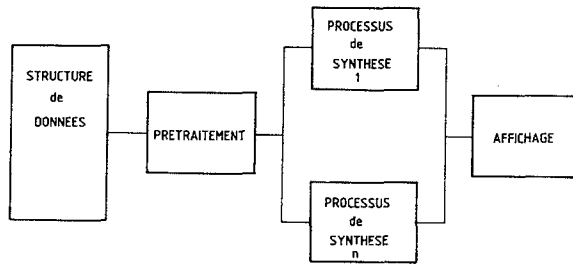


FIG 8 Processus général de synthèse. Découpage Fonctionnel

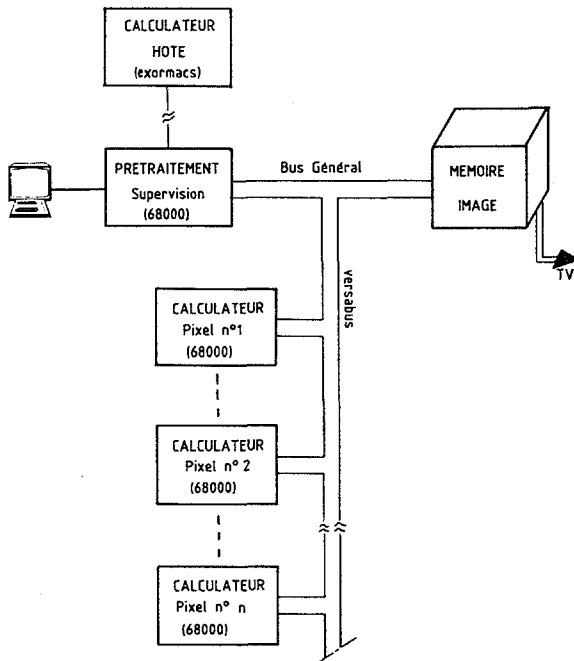


FIG 9 Structure matérielle : Principe

hiérarchisée et structurée en arbre binaire. L'image à afficher est la racine de l'arbre et les feuilles sont constituées par les primitives (solides géométriques simples ou surfaces plus complexes). A chaque noeud de l'arbre (ou composite) correspond une phase de recombinaison des solides primitifs. Les feuilles de l'arbre comportent les données suivantes :

- relation de voisinage (noeud père)
- nature de la primitive

- transformations directe et inverse du repère local de la primitive au repère global de la scène
- identification de surface (couleur, texture, etc).

Les noeuds composites comportent les données suivantes :

- relations de voisinages (noeud père et noeuds fils)
- l'opération (+, -, \cap) à appliquer entre les deux fils.

La structure de données à l'entrée du processus général de synthèse présente donc cette structure arborescente de la scène à afficher, plus un certain nombre d'éléments liés à l'observation de la scène : matrices de transformation donnant la position de l'observateur et des sources lumineuses et la position de l'écran par rapport au repère global de la scène.

2-3-1-2 Décomposition du processus général de synthèse

Précédant le processus de synthèse proprement dit, on distingue une phase de prétraitement consistant en un ensemble de calculs préliminaires.

En vue de déterminer les calculs d'intersection inutiles, on calcule à chaque noeud de l'arbre un volume d'extension englobant l'objet correspondant : primitive de base, objets intermédiaires, scènes globale. Pour chaque objet, deux volumes sont calculés : un volume parallélépipédique utilisé pour les rayons primaires et un volume sphérique utilisé pour les rayons secondaires.

Dans cette phase de prétraitement, la suite des calculs est la suivante :

- concaténation des transformations de repères : calcul pour chaque primitive, de la matrice de transformation du repère local au repère écran ainsi que son inverse :
- calcul des volumes parallélépipédiques et du rectangle englobant leur projection sur l'écran;
- calcul des volumes sphériques;

Le rectangle résultant des projections des volumes parallélépipédiques et de leur combinaison à l'aide des opérateurs de chaque noeud composite délimite sur l'écran une zone de pixels pour lesquels les calculs sont à réaliser avec plus de précision.



Les volumes sphériques sont utilisés avec les rayons secondaires pour éviter là aussi les calculs d'intersection inutiles.

Le processus de synthèse est lancé pour chacun des pixels de la zone utile. Il a accès à la structure de données et au module d'affichage.

La suite des calculs pour chaque rayon primaire est la suivante :

- vérifier qu'il traverse au moins un volume d'extension de primitive (sinon le processus est arrêté),
- si oui, calculer les points d'intersection du rayon avec les différentes primitives dont les volumes d'extension sont traversés,
- classier les segments du rayon limités par ces points d'intersection "in" ou "out" suivant que les segments sont intérieurs ou extérieurs aux primitives,
- combiner les segments conformément aux opérateurs des noeuds composites.
- déterminer la première surface rencontrée par le rayon primaire.

A partir de ce point d'intersection, deux rayons secondaires sont relancés pour déterminer l'intensité lumineuse en ce point, un rayon réfléchi et un rayon réfracté (si la surface est transparente).

De plus, pour déterminer l'ombre portée en ce point par d'autres objets de la scène, on relance autant de rayons secondaires qu'il y a de sources lumineuses.

Comme pour le rayon primaire, les calculs d'intersection des rayons secondaires avec les primitives ne sont réalisés que si les volumes d'extension sont traversés. Ensuite, après classification et combinaison, le processus détermine s'il y a ombre portée, s'il y a un nouveau point d'intersection avec une autre surface occasionnant une seconde réflexion, ou si l'impact lumineux est direct.

En fin de processus, l'intensité lumineuse correspondant à chaque pixel est transmise au processus d'affichage.

2-3-2- Introduction du parallélisme

On s'aperçoit qu'après détermination par le module de prétraitement de la zone de pixels de l'écran dont le calcul est réellement à faire, le processus de synthèse agit de façon indépendante pour chacun des pixels et leurs rayons associés.

On peut donc augmenter la rapidité du processus général de synthèse en lançant en parallèle plusieurs processus sur des pixels différents.

Les deux points de conflit sont l'accès à la structure de données et l'accès au module d'affichage. Or il apparaît que l'accès au module d'affichage ne se fait qu'une seule fois par pixel. Un simple contrôle d'accès de type partage temporel d'un bus suffit donc à ce niveau eu égard aux performances des processeurs utilisés.

Par contre, la structure de données est accédée à plusieurs reprises au cours du processus de synthèse. Mais elle est en fait peu volumineuse et sa duplication au niveau de chaque processus de synthèse ne pose pas de problème.

Nous noterons ce parallélisme élémentaire "parallélisme de premier niveau" par opposition à d'autres niveaux de parallélisme à l'intérieur même du processus de synthèse.

L'exploitation de ce premier niveau conduit à une architecture de machine multimicroprocesseur. Chacun des microprocesseurs réalise un processus de synthèse appliqué à un pixel ou à un groupe de pixels. Les problèmes rencontrés sont à ce niveau des problèmes classiques d'affectation de zones de mémoire d'image aux processeurs de calcul (ici : affectation de groupes de pixels aux processus de synthèse).

On peut noter un certain nombre de travaux dans ce domaine : [15], [16], [17], [18].

2-3-3- Réalisation matérielle (Fig 9)

La maquette d'étude est réalisée à l'aide de microprocesseurs 68000. Bâtie autour du bus VERSABUS, elle comprend :

- 4 cartes processeurs KVM02,
- une mémoire d'image 768x575 à 24 bits par pixel, lue par un processeur vidéo au standard TV 625 lignes.
- le logiciel est développé sur calculateur EXORMACS en assembleur et en PASCAL.



3 - Conclusion

Les résultats obtenus montrent la validité de l'approche "parallélisme de premier niveau". En effet, malgré la modestie des performances de chaque processeur et malgré l'absence de coprocesseur mathématique, les temps de calcul obtenus (7 mn pour la scène de la photo N°4) sont encourageants. Et l'évolution des microprocesseurs et coprocesseurs associés permet d'envisager des résultats de plus en plus intéressants.

Cependant, cette approche a ses limites : limites des bus, conflits de partages de ressources,... et il faut optimiser les calculs au niveau de chaque pixel. Pour ce faire, une découpe fonctionnelle plus fine du processus de synthèse permet d'envisager d'autres niveaux de parallélisme débouchant sur des architectures spécifiques [19][21]. Et c'est dans ce sens également que l'évolution technologique (microprocesseurs microprogrammables VLSI performants,...) nous incite à poursuivre les investigations.

VI - Résultats et conclusion

Les trois premières photographies ont été réalisées sur un VAX 11-750 couplé à un processeur graphique AYDIN CONTROL 5216. La résolution est de 1024 x 1024 en X, Y et de 16 bits en Z avec 8 bits pour la composante verte et 4 bits pour le rouge et le bleu. Les deux premières photographies illustrent les possibilités du modèle photométrique décrit en [10]. La troisième a été générée par l'algorithme de visualisation des surfaces de révolution décrit précédemment. La courbe génératrice est une B. Spline avec 18 points de contrôle. Le modèle photométrique utilisé ne traite que la composante diffuse (loi de Lambert). Le temps C.P.U. nécessaire pour générer cette image a été de 00 mn.

En conclusion, la synthèse d'image par lancer de rayon continuera sans doute à connaître les développements importants dans les applications audiovisuelles. Sur le plan algorithmique, bien des améliorations sont encore possibles car cette méthode a en fait un passé relativement récent et les publications sont, à ce jour, encore peu nombreuses.

La qualité des images produites peut encore être améliorée, surtout en ce qui concerne l'anticréne-

lage, mais c'est surtout en ce qui concerne le matériel qu'il faut s'attendre à une évolution sensible avec l'utilisation des architectures parallèles beaucoup mieux adaptées et beaucoup plus économiques que les architectures classiques pour cette application. A court terme, les possibilités du "temps réel" sont exclues, le domaine étant pour l'instant réservé aux systèmes travaillant sur des facettes planes. Par contre, des matériels pouvant générer une trame en haute définition dans un temps de l'ordre de la minute, verront sans doute le jour d'ici deux ans, ce qui entraînera une baisse sensible du coût de production d'images synthétiques de qualité.



- [1] APPEL A. "Some Techniques for shading machine renderings of solids" SJCC (1968) (AFIPS Conf. Proc.) Vol 32, pp 37 - 45 .
- [2] ROTH S.D. "Ray Casting for modeling solids" Computer Graphics and Image Processing. Vol 18, N°2 pp 109 - 144 .
- [3] ATHERTON P.R. " A scan-line hidden surface removal procedure for constructive solid geometry". Siggraph 83. Proc. Computer Graphics Vol 17, N°3 pp 109 - 144.
- [4] BRONSVOORT W.F, J.J Van WIJK, F.W. JANSEN "Two methods for improving the efficiency of ray casting in solid modeling" Computer Aided Design (1984) Vol 16, pp 51 - 55.
- [5] KAJIYA J.T. "New Techniques for Ray-Tracing Procedurally Defined Objects" Transaction on Graphics (1983) Vol 2, N° 3 (July 1983) pp 161 - 181 .
- [6] WHITTED J.T. "An improved Illumination Model for Shaded Display" Comm. ACM, Vol 23, N° 6 (June 1980), pp 343 - 349 .
- [7] KAJIYA J.T. "Ray Tracing Parametric Patches Computer Graphics"(1982) Vol 16, N° 3, pp 245 - 254.
- [8] HANRAHAN P. " Ray Tracing algebraic surface" Computer Graphics Vol 17, N° 3 (Jul.1983) pp 83 - 90 .
- [9] RUBIN S.M., J.T. WHITTED. "A 3 Dimensional Representation for Fast Rendering of Complex Scenes" Computer Graphics. (1980) Vol 14, N° 3, pp 110-116 .
- [10] BOUVILLE C, DUBOIS J.L , MARCHAL I. Generating high quality pictures by Ray-Tracing , Doc interne C:C.E.T.T.
- [11] COOK R.L; TORRANCE K.E, "A reflectance model for computer graphics" ACM Transactions on Graphics 1, 1 (Janv. 1982) PP 7 - 24 .
- [12] TORRANCE K.E, SPAROW E.M "Theory for off-specular reflection from roughened surfaces" Journal of the optical society of America 57, 9, (Sept 67).
- [13] BLINN J.F " Models of light reflection for computer synthesised pictures" Computer Graphics 11 2, (Jul. 1977) pp 192 - 198.
- [14] COLLINS G.E, R. LOOS "Real zeros of polynomials" Computing Suppl. 4, (1982) pp 83 - 94 .
- [15] FUCHS H "Distributing a visible surface Algorithm over Multiple Processors" Proc. ACM 77 Seattle. October 1977 pp 449 - 451 .
- [16] FUCHS H., JONSON B.W. "An expandable Multi-Processor Architecture for video graphics" ACM SIGGRAPH . Vol 7 pp 58 - 67 April 1979 .
- [17] JAMES CLARK "A high performance personal computer system" Stanford university.
- [18] R.F SPROULL, SOTISH GUPTA A VLSI Architecture for updating raster scan displays computer graphics Vol 15 N° 3 pp 71 - 78 .
- [19] C. BROWN "Special purpose computer Hardware for mechanical design systems proceedings" NCGA 81. pp 403 - 414 .
- [20] "Panel on Japanese computer graphics Part 1" Computer graphics 17, 3 (Jul 1983), pp 191-192.
- [21] GARDNER G.Y. "Computer Image Generation with efficient image storage" SPIE Vol.117, optical information storage (1979).

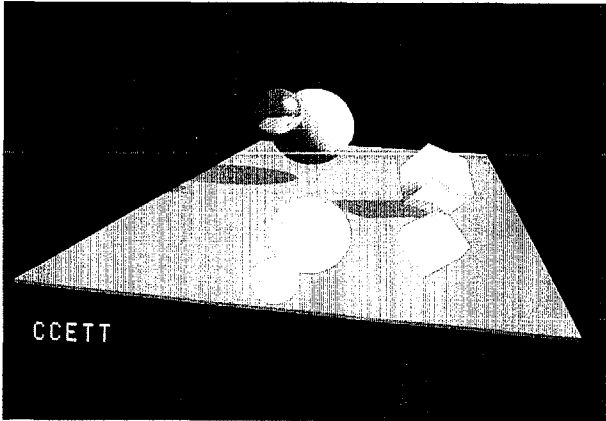


Photo N° 1

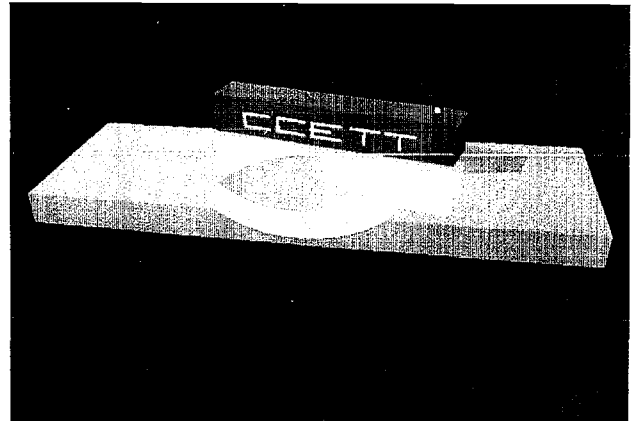


Photo N° 2

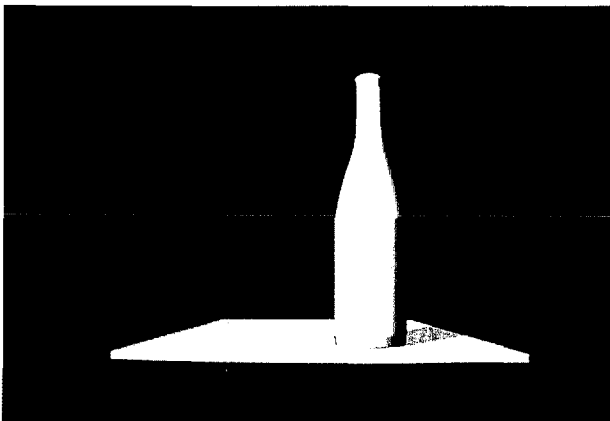


Photo N° 3

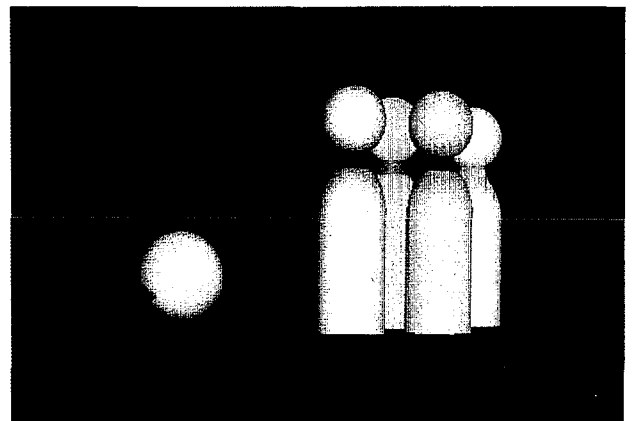


Photo N° 4