

---

## RENDU DE SURFACES PAR DEFORMATION DE TEXELS

**Babahenini Mohamed Chaouki**

**Département d'Informatique, Université Mohamed Khider de Biskra**  
[babahenini@wissal.dz](mailto:babahenini@wissal.dz)

**Hamida Ammar**

**Département d'Informatique, Université Mohamed Khider de Biskra**

**Hemidi Djallel**

**Département d'Informatique, Université Mohamed Khider de Biskra**

**Djedi Noureddine**

**Département d'Informatique, Université Mohamed Khider de Biskra**  
[djedi\\_nour@yahoo.fr](mailto:djedi_nour@yahoo.fr)

***Résumé** – Le besoin accru de réalisme, de convivialité et d'efficacité a conduit à l'étude de nouvelles méthodes d'habillage où la génération de textures, le plaquage et le rendu global se conjuguent pour permettre la création et la représentation multi-échelle de scènes à géométrie répétitives complexes.*

*Nous présentons dans cet article une méthode générale pour l'habillage et le rendu d'une surface tridimensionnelle représentant la scène par une texture volumique. Cette surface doit initialement être discrétiser afin de créer un maillage surfacique optimal permettant ainsi de construire un maillage volumique qui sera considéré comme une peau épaisse. La manipulation interactive de ce maillage par la perturbation de ses paramètres conduira à le coiffer.*

*Une fois le coiffage réaliser, la méthode que nous proposons passe à l'étape de l'habillage qui s'effectue dans le cadre d'un rendu global utilisant un lancer de rayon conique courbé et prenant en charge l'aspect multi-échelle de la texture volumique.*

***Mots Clés** – Texture volumique, Habillage, Lancer de faisceaux, Texel, Maillage volumique, déformation, Mip map, Niveau de détail.*

## **1. INTRODUCTION :**

L'un des grands problèmes en synthèse d'images est le traitement de scènes très complexes. Ces scènes posent plusieurs défis tel que :

- le grand nombre de détails à conserver qui requiert une méthode d'accès aux informations efficace et rapide,
- le temps de calcul nécessaire pour afficher toutes ces informations qui doit demeurer raisonnable,
- et enfin le résultat visuel obtenu qui doit être une bonne approximation (une image) de notre information (les objets de la scène).

Les textures volumiques apportent une solution aux problèmes posés par les scènes qui exhibent une grande répétitivité. Ces scènes peuvent toutes être construites à partir d'un nombre relativement restreint de primitives qui sont répétées et déformées, ce qui élimine le problème de l'accès rapide aux informations, qui sont typiquement peu nombreuses. Cependant, le problème d'un rendu efficace et exhibant peu d'aliassage demeure entier. Pour cela il est nécessaire de donner aux informations une importance proportionnelle à leur contribution, d'où l'idée de représenter les mêmes informations à plusieurs échelles.

Une fois cette représentation volumique, multi-échelle, basée sur l'information de réflectance au lieu de la géométrie établie, elle doit être exploitée afin d'habiller et faire le rendu des scènes tridimensionnelles. Pour cela, il faut établir une fonction de plongement de l'espace de la texture vers l'espace tridimensionnel. Cette fonction consiste à rattacher les coordonnées de la texture aux points de la surface à habiller, ce qui pose des problèmes de distorsions, et de discontinuités [5][6]. Il faut ensuite passer à l'étape de rendu qui doit être efficace et doit se faire dans un cadre global.

La méthode que nous proposons dans cet article repose essentiellement sur celle établie par Neyret [1,2,3,4], et Ratib [7]. Néanmoins, nous avons introduit des optimisations dans le calcul du maillage surfacique et généraliser la notion de mip-map [8] en l'intégrant dans l'habillage et le rendu de la surface, ce qui a permis de prendre en charge l'aspect niveau de détail avec lequel a été défini la texture volumique.

## **2. PRESENTATION DES TEXTURES VOLUMIQUES :**

L'utilisation des textures a compensé la pauvreté des modèles géométriques pour représenter la réalité. Elles ont permis de simuler l'aspect externe des objets en prenant en charge la complexité qu'ils peuvent exhiber, et les détails qu'ils peuvent contenir.

Le modèle de textures volumiques, introduit par Kajiya et Kay [9] pour le rendu de la fourrure constitue une approche orientée texture pour représenter des géométries répétitives complexes recouvrant une surface à la manière d'une peau épaisse. Pour cela ils utilisent un volume cubique contenant un échantillon de référence de cette géométrie encodée sous forme d'un volume de voxels. Les instances de ce volume plaquées sur une surface sont appelées *texels*. Ce volume de référence est déformé de façon à être plaqué sur la surface, à la manière d'une texture 2D classique.

Le volume cubique de référence est constitué de voxels. Chaque voxel contient formellement une densité; un ensemble de 3 vecteurs donnant l'orientation locale de la surface; et une fonction indiquant comment la lumière se réfléchit.

Un texel contient donc une information spatiale (densité) et une information sur le comportement local vis à vis de la lumière (réflectance). Cette fonction est constante dans un modèle de fourrure : le volume de référence représente un échantillon constitué de cylindres (les poils), il est ensuite instancié et déformé pour suivre le sens du poil. Le calcul de la fonction d'illumination analytique sera amélioré par Goldman.

Dans [9] les auteurs voulaient représenter une texture bien particulière, la peluche sur un ours. Le texel et la fonction de réflectance utilisés sont très spécifiques : le texel, qui doit matérialiser un ensemble de poils, contient des cylindres perpendiculaires à la base du volume, et la fonction de réflectance utilisée est cylindrique, n'autorisant que des objets cylindriques dans le volume de référence.

Neyret [1, 2, 3, 4] a généralisé ce modèle, en proposant une fonction de réflectance paramétrable (l'ellipsoïde) qui est capable de modéliser de nombreux types de formes, et a introduit une approche multi-échelles similaire au *MIP mapping* utilisant les octrees. Le niveau de détails affiché est ainsi modulé en fonction de la distance qui sépare le texel de l'observateur,

Noma [10] utilise le principe des textures volumiques de Kajiya pour le rendu spécifique d'arbres. L'opacité est stockée en chaque voxel, de manière discrète en échantillonnant la géométrie depuis une série de directions autour de la sphère, il interpole ensuite les valeurs lors du rendu. Pour la fonction d'illumination, il calcule en chaque voxel la moyenne des normales des feuilles et la moyenne des cosinus des angles entre ces normales et les trois axes X,Y,Z. Lors du rendu, il interpole ces valeurs en fonction des angles entre la lumière et les axes X,Y,Z.

### 3. HABILLAGE ET RENDU DE LA SURFACE :

Une fois le maillage réalisé, notre modèle passe à l'étape d'habillage de la surface qui s'effectue dans le cadre d'un rendu global des texels qui est réalisé à l'aide d'un algorithme de lancer de rayons conique. Cette technique est choisie car les texels s'y intègrent naturellement.

- Le processus du rendu doit acquérir les données suivantes :
  - Les volumes de références.
  - Les sources lumineuses (positions et intensités).
  - L'ensemble des boîtes qui représente la peau volumique.
- Pour chaque pixel, le tracer de rayons s'effectue à partir de deux types de calcul :
  - des calculs géométriques : qui s'intéressent aux intersections entre un rayon et un objet.
  - des calculs optiques : qui s'intéressent aux opérations de mélange des couleurs.

Le lancer de rayons s'effectue donc dans deux espaces différents :

- Le lancer de rayons classique dans l'espace scène : qui calcul l'intersection du rayon avec l'objet.
- Le lancer de rayons courburé dans l'espace texture : qui s'intéresse aux calculs optiques.

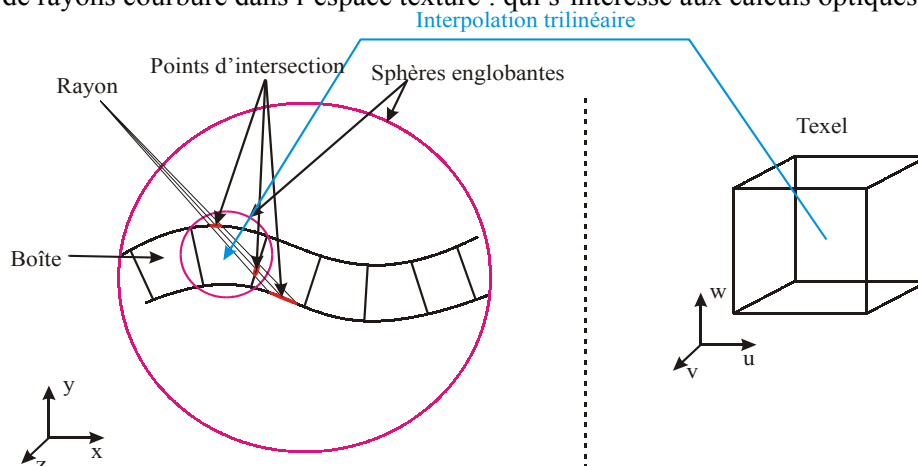


figure 1 : Habillage de la surface.

### 4. LE LANCER DE RAYONS DANS L'ESPACE SCENE

Le lancer de rayons dans l'espace scène permet de calculer l'intersection entre le rayon et l'objet. Pour cela nous devons suivre les étapes suivantes :

#### 4.1. Intersection du rayon avec la peau volumique

Elle consiste à effectuer d'une part l'intersection du rayon avec les faces des boîtes qui construisent la peau et d'autre part l'intersection du rayon avec la surface de l'objet (la face basse de la boîte).

- ❖ Intersection avec les faces des boîtes : Pour minimiser le temps de calcul concernant la recherche de la bonne face, on procède comme suit :
  - Pour toute face appartenant à une des boîtes du maillage on effectue un test afin de savoir si le rayon lancé intersecte cette face.
  - Etant donnée un rayon défini par son origine  $O$ , sa direction  $\vec{D}_R$  et un segment droit  $AB$  de la face, le produit mixte  $\langle \vec{OA}, \vec{D}_R, \vec{AB} \rangle$  est positif si le rayon passe "à gauche" de  $AB$ . Alors un rayon intersecte une face si ce critère est vérifié sur tout le contour de celle ci dans le sens trigonométrique [13].
  - Si c'est le cas, il faut calculer l'intersection.
- ❖ Méthode de calcul du point d'intersection avec une face :
  - Dans le cas où on n'a pas une forte déformation des boîtes ; le calcul s'effectue comme suit :
    - On peut considérer que le vecteur  $\vec{d}$  est une normale à un point quelconque sur la face.

- S (la distance entre la face et le centre de la scène) est calculé à partir de :  $S = P^t \vec{d}$ , où il suffit de remplacer P par un des points définissant la face (A, B, C ou D), ce qui permet de déduire l'abscisse  $\lambda$  où :

$$\lambda = \frac{S - O^t \vec{d}}{D_R \cdot \vec{d}}$$

- Les coordonnées du point d'intersection sont trouvées en remplaçant  $\lambda$  dans la formule  $P = O + \lambda \vec{D}_R$ .

- Dans le cas où il existe une forte déformation des boîtes, le calcul s'effectue par la division de la face en deux triangles (ABC) et (ACD), et l'application des mêmes étapes de calcul précédant pour un triangle.

L'ensemble des boîtes constituant la peau obtenues à partir du maillage volumique de la surface et les points d'intersection avec les boîtes (le point d'entrée et le point de sortie) permettent de définir le chemin du rayon par :

- Le tri croissant des points d'entrée dans chaque boîte, selon l'abscisse d'intersection avec le rayon.
- Le calcul de l'ouverture du rayon correspondant à chaque boîte.
- Le passage de l'espace scène à l'espace texel, pour chaque point d'intersection.

A la fin de cet étape, on dispose du trajet du rayon dans la peau représenté par l'ensemble des boîtes intersectées, des points d'intersection et des ouvertures dans l'espace objet, on est donc prêt à poursuivre le rayon dans l'espace texel.

#### 4.2. Calcul de l'ouverture du rayon sur la face intersectée

Le calcul de l'ouverture permet de choisir la résolution adéquate à l'intérieur de l'octree et de minimiser l'aliassage, si l'ouverture est importante, alors la résolution utilisée sera plus basse, donc le filtrage sera plus important.

L'ouverture correspond à la projection inverse d'un pixel de l'écran sur un plan parallèle à l'écran. Plus on s'éloigne de l'écran, plus grande sera la région du texel projetée sur un seul pixel.

- ❖ Calcul de l'ouverture : Si on considère le pixel comme un cercle, l'ouverture du rayon est donc une mesure de cette région circulaire du texel. L'équation de calcul de l'ouverture du cône dans l'espace scène est donnée par :

$$\frac{\sqrt{3}}{H} \cdot \frac{d}{f} (\vec{D} \cdot \vec{a})$$

Où

H: la résolution verticale de l'image.

d : la distance du point d'impact.

$\vec{D}$  : la direction du rayon.

f : la distance focale de la caméra.

$\vec{a}$  : la distance pointée par la caméra.

Une fois l'ouverture calculé on doit la ramener dans l'espace du texel, en la divisant par le min des côtés du texel s'il on préfère une image nette avec un peu d'aliassage ou par le max des côtés du texel si l'on préfère un léger flou. Connaissant l'épaisseur du rayon à cet endroit, on estime la taille de voxel qui correspond le mieux à cette ouverture, ce qui est exactement du mip-mapping (généralisé ici au cas 3D).

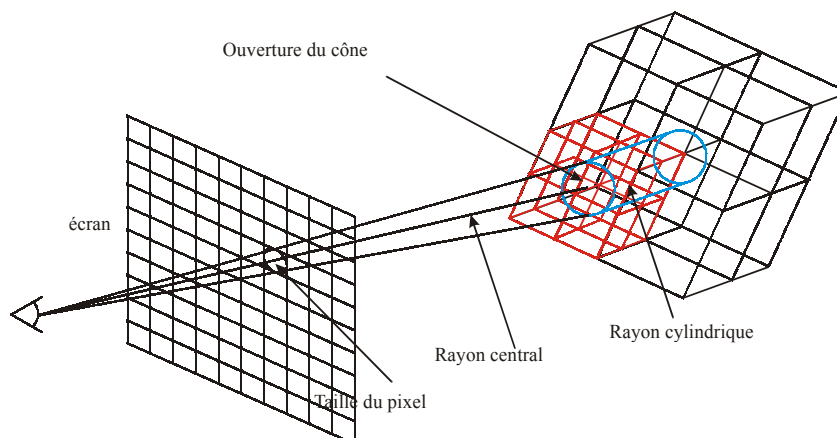


figure 2 : Traversée du texel.

### 4.3 Le passage de l'espace scène à l'espace texel et l'inverse

Après avoir calculer les points d'intersection (de coordonnées (x, y, z)) du rayon avec la peau volumique; on doit passer dans l'espace texel (de coordonnées (u, v, w)) où le rayon poursuit sa trajectoire. Ce passage entre les deux espaces s'effectue par:

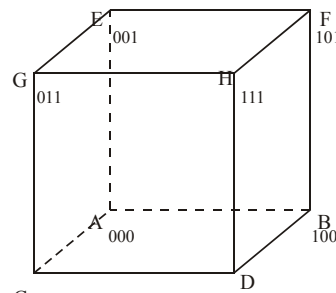
- Le passage de l'espace objet (où la texture est définie) à l'espace scène : on utilise une interpolation trilineaire donnée par la formule suivante :

$$(f,g,h)=T(x,y,z) = (1-w)*((1-v)*((1-u)*V000+ u*V100)+ v*((1-u) * V010 + u* V110)) + w* ((1-v)*((1-u)*V001+ u*V101) + v*((1-u)*V011+ u*V111)).$$

- Le passage dans l'autre sens, nécessite d'inverser ce système non linéaire. Pour cela, on peut utiliser des méthodes numériques qui se basent sur un processus itératif où le temps de calcul est prohibitif. Pour remédier à ce problème nous proposons d'utiliser le Jacobien de la déformation T pour déterminer la direction du rayon dans chaque cellule. Soit:

$$\left. \begin{array}{l} V000 \rightarrow A \\ V100 \rightarrow B \\ V010 \rightarrow C \\ V011 \rightarrow D \\ V001 \rightarrow E \\ V101 \rightarrow F \\ V011 \rightarrow G \\ V111 \rightarrow H \end{array} \right\} \text{Les coordonnées des huit points de la boîte en espace scène.}$$

$$J = \begin{bmatrix} \frac{\delta f}{\delta u} & \frac{\delta g}{\delta u} & \frac{\delta h}{\delta u} \\ \frac{\delta f}{\delta v} & \frac{\delta g}{\delta v} & \frac{\delta h}{\delta v} \\ \frac{\delta f}{\delta w} & \frac{\delta g}{\delta w} & \frac{\delta h}{\delta w} \end{bmatrix} = \begin{bmatrix} J[0] \\ J[1] \\ J[2] \end{bmatrix}$$



donc le jacobien de la déformation est:

$$\begin{aligned} J[0] &= (1-w)* ((1-v)* \overrightarrow{AB} + v* \overrightarrow{CD}) + w*((1-v)* \overrightarrow{EF} + v* \overrightarrow{GH}) \\ J[1] &= (1-w)* ((1-u)* \overrightarrow{AC} + u* \overrightarrow{BD}) + w*((1-u)* \overrightarrow{EG} + u* \overrightarrow{FH}) \\ J[2] &= (1-v)* ((1-u)* \overrightarrow{AE} + u* \overrightarrow{BF}) + v*((1-u)* \overrightarrow{CG} + u* \overrightarrow{DH}) \end{aligned}$$

### 5. PRINCIPE DU PARCOURS (DÉFORMATION DU RAYON)

Le calcul d'intersection de rayons non linéaires avec la géométrie est relativement complexe. Par contre, lorsque les données sont spatialement structurées (ex : volume en voxels), le rayon peut être suivi pas à pas pour déterminer les pixels couverts.

Soit une déformation qui transforme un point  $M_o$  en  $M_s=T(M_o)$ , où T représente la transformation trilineaire et les indices 'o' et 's' désignent l'espace objet et l'espace scène respectivement. Il est difficile d'avoir la transformation inverse  $T^{-1}$  permettant d'effectuer le passage de l'espace scène vers l'espace objet, mais si on sait définir une boîte englobante autour de l'objet dans laquelle la transformation est bijective, le rendu peut se faire en calculant les points d'intersection du rayon linéaire avec la boîte déformée dans l'espace de la scène, puis en ramenant ces points dans l'espace de l'objet, afin de suivre le rayon dans cet espace.

#### 5.1. Résolution numérique

Le jacobien de la transformation  $J_T$  peut être calculé en tout point, et le jacobien  $J_{T^{-1}}$  de la transformation inverse et  $J_T^{-1}$ . Un vecteur  $V_o$  au point  $M_o$  est transformé par T en  $V_s=J_T(M_o)*V_o$ .

Cela nous permet de résoudre numériquement le problème de l'inverse de la transformation T, en utilisant l'algorithme suivant :

**InvT(X)**

**Début**

$$U \leftarrow (0.5, 0.5, 0.5)$$

**Jusqu'**à convergence

$X_0 \leftarrow T(U)$

$dX \leftarrow X - X_0$

**Si**  $\|dX\| < \text{eps}$  **alors** fin

$\text{InvJ} \leftarrow \text{inv}(J(U))$

$dU \leftarrow \text{InvJ} * dX$

$U \leftarrow U + dU$

Retourner U

**Fin**

On pourrait alors construire un algorithme itératif type Newton pour trouver l'intersection du rayon courbé avec une surface. Les données exigent une structuration permettant de suivre le rayon cellule par cellule, le problème étant alors juste de trouver le point de sortie de la cellule courante. Ceci nous ramène au problème du tracé de droite, c'est à dire la détermination des intersections entre un rayon et une grille régulière.

Le rayon étant défini par son origine  $X_0$  et sa direction D, il nous faut alors trouver son intersection avec l'une des tranches parallèles aux axes (parois), définie par l'isovaleur  $u_i=c$ . Cela donne l'algorithme suivant :

**Atteint\_Tranche** (i, c,  $X_0$ , D)

**Début**

$X \leftarrow X_0$

**Jusqu'**à convergence

$U \leftarrow \text{InvT}(X)$

**Si**  $|U[i]-c| < \text{eps}$  **alors** fin

$\text{InvJ} \leftarrow \text{inv}(J(U))$

$dU \leftarrow \text{InvJ} * D$

**Si**  $dU[i] \neq 0$  **alors**

$d \leftarrow (c - U[i]) / dU[i]$

**Sinon**

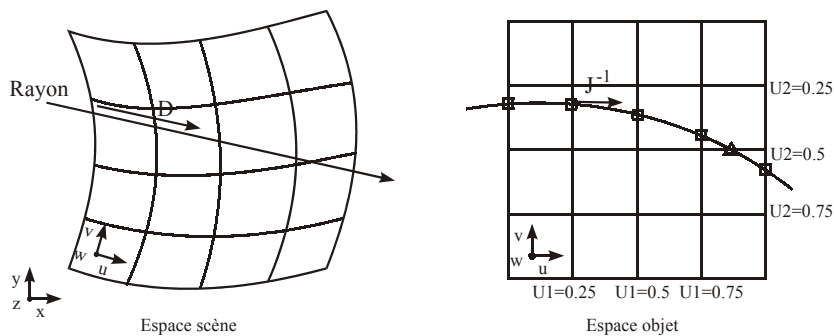
$d \leftarrow 1$

**Finsi**

$X \leftarrow X + d * D$

Retourner U

**Fin**



**figure 3** : résolution numérique de l'intersection rayon courbé – grille.

Pendant la traversée récursive de l'octree, il faut calculer pour chaque voxel les points d'intersection avec les parois, afin de trouver les voxels enfants traversés par le rayon.

Pour trouver le voxel-enfant traversé par le rayon, il suffit de comparer le point d'intersection avec le centre du voxel. Pour chaque axe, il existe deux possibilités (au-dessus ou au-dessous). Chacune des  $2^3 = 8$  possibilités correspond donc à l'un des huit voxels enfants [7].

## **5.2. Incidence sur l'illumination**

On procède au calcul de l'illumination locale lorsque le rayon a atteint un point de l'objet, ce qui ne peut se faire directement dans l'espace objet : l'illumination dépend des normales. Le modèle d'illumination de Phong qui utilise les produits scalaires entre la normale locale et les vecteurs d et L dans l'espace scène ne peut pas être évalué directement dans l'espace objet. Pour calculer le modèle de Phong avec les données disponibles dans l'espace objet, il faut soit réévaluer la normale à partir de la transformation du voisinage de  $M_0$ , soit modifier le produit scalaire de telle sorte qu'il donne le même résultat que le produit scalaire canonique dans l'espace de la scène.

Etant donné un vecteur  $V_o$  et la normale  $N_o$  en  $M_o$  dans l'espace objet, on a  $\langle N_o, V_o \rangle = \langle N_o, J^{-1} \cdot J \cdot V_o \rangle = \langle J^{-1t} \cdot N_o, V_s \rangle$ , cela donne la normale dans l'espace de la scène  $N'_s = J^{-1t} \cdot N_o / \| J^{-1t} \cdot N_o \|$ , c'est à dire que l'on convertit la normale locale  $N_o$  en  $N'_s$  avant d'effectuer le produit scalaire canonique de  $n$  avec un vecteur  $V$  dans l'espace scène.

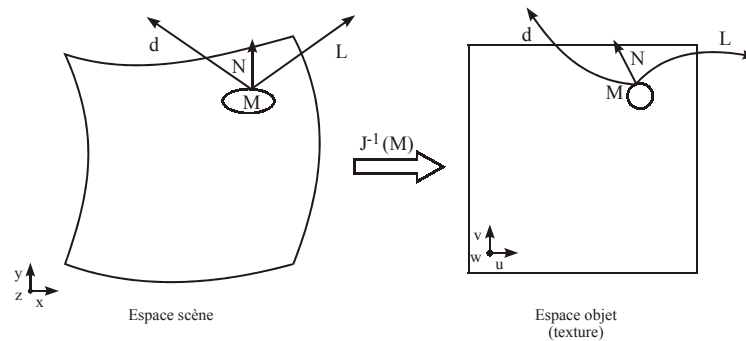


figure 4 : Transformation du voisinage et des vecteurs.

### 5.3. Accélération dans le calcul du rendu

Vu le très grand nombre de calculs d'intersections rayon-objet, on utilise deux méthodes d'accélération dans la recherche de la première face intersectée par le rayon :

- Une sphère englobante de la surface qui est utilisée pour éliminer tous les rayons qui n'intersectent pas l'objet et attribue la couleur de fond à ce rayon, ainsi qu'une sphère englobante pour chaque boîte associée à sa face qui est utilisée pour minimiser le temps de parcours dans la liste des boîtes.
- Une grille 3D qui correspond à une « voxelisation » de l'espace à l'intérieur de la boîte englobante d'un objet, les voxels contiennent la liste des facettes présentes dans leur espace. Ceci accélère fortement le lancer de rayons, dans la mesure où seules les facettes contenues dans les voxels qui sont sur la trajectoire du rayon, doivent être testées. Mieux encore, le rayon peut parcourir pas à pas cet espace structuré, comme on le fait pour le tracé de droite en 2D [13,14]

### 6. RENDU D'UN TEXEL INTERSECTE PAR UN RAYON

Quand un rayon de la scène intersecte un texel, on procède au calcul de l'intensité du rayon  $R$  traversant ce texel. Un rendu volumique est alors effectué et l'intensité est obtenue par intégration de la fonction de réflectance (NDF), (selon le modèle d'illumination locale de *Phong*), sur le trajet du rayon à l'intérieur du volume pour chaque source lumineuse de la scène modulée par l'atténuation cumulée sur le rayon, ainsi que par l'atténuation due au blocage des sources lumineuses. Dans notre cas cette équation se traduit par la somme des illuminations locales à chaque voxel traversé par le rayon, en cumulant aussi l'atténuation à chaque voxel.

Une fois le rayon  $R$  transformé, nous obtenons un cône  $R_T$  en espace texel. Nous pouvons alors calculer les voxels traversés par ce cône. L'idée de la traversée est de trouver les points d'intersection de  $R_T$  courbé avec les parois des voxels, et de retourner chaque voxel intersecté dont la taille correspond à l'ouverture de  $R_T$  [15].

Si l'ouverture du rayon ne correspond pas exactement à la taille des voxels illuminés, il faut calculer l'intensité pour le niveau fin (taille du voxel < ouverture) et le niveau supérieur, ensuite interpoler entre les deux intensités pour obtenir l'intensité qui correspond à l'ouverture (Mip-mapping). La traversée est terminée une fois que le rayon devient opaque. Dans le cas contraire il faut traverser le texel voisin.

#### 6.1 Rendu des voxels optimaux traversés par le rayon

Pour calculer le rendu des voxels optimaux traversés par le rayon, on procède au calcul de l'atténuation locale  $Att_v$ , puis on effectue un lancer de rayons d'ombrage vers les sources lumineuses pour le calcul de l'atténuation de chaque rayon et on doit enfin intégrer le modèle d'illumination locale sur la NDF du voxel.

a) Le calcul de l'atténuation locale  $Att_v$  : Elle est obtenue en combinant la visibilité dans la direction du rayon et la transparence du matériau du voxel, pondérée par la distance traversée à l'intérieur du voxel :

$$Att_v = 1 - ((1 - \text{Visibilité}) * \text{FacteurDistance} * (1 - \text{Transparence})). [7]$$

Où :

- FacteurDistance qui représente la longueur du segment de rayon traversant ce voxel.

- $\text{Transparence} = 1 - (1 - \text{matériau}) * (1 - \text{densité- voxel} * dl)$  elle représente le degré de transparence du voxel.
- $\text{Visibilité} = (1 - \text{Transparence}) * T_{\text{total}} * \text{occultation}$ .
- $T_{\text{total}}$  : est l'atténuation totale du rayon.

b) Le calcul de l'atténuation des rayons d'ombrage : Pour chaque source lumineuse, un rayon d'ombrage est lancé vers celle-ci. Ce rayon doit, d'abord traverser le texel courant avant de continuer son trajet dans la scène.

c) L'intégration du modèle d'illumination locale sur la NDF : Une fois l'atténuation d'un rayon pour chaque source lumineuse calculée, il faut intégrer le modèle d'illumination locale sur la partie visible de l'ellipsoïde représentant la NDF du voxel, puis calculer l'intensité contenant les deux composantes spéculaire et diffuse de la couleur du voxel courant.

- *Génération des points visibles et leurs normales* : Il faut en premier lieu calculer l'ellipse apparente (la partie visible qui correspond à la direction de l'observateur), en utilisant la forme quadratique de l'ellipsoïde et le vecteur de direction de l'observateur, puis on effectue l'échantillonnage de cette ellipse apparente sur les deux directions  $x$  et  $y$  (16 échantillons par exemple).
- Calcul des normales : La NDF est originellement positionnée dans l'espace texel ; il est nécessaire de la transformer dans l'espace scène pour calculer l'illumination. Nous effectuons cette étape en transformant chaque normale obtenue de l'ellipsoïde par la transposition du jacobien de la déformation inverse du texel.
- Le calcul de l'illumination d'un échantillon : Une fois ces calculs effectués, on passe au calcul de l'illumination d'un échantillon (le calcul de l'intensité diffuse et spéculaire  $I_{\text{diffuse } i}$ ,  $I_{\text{spéculaire } i}$  pour chaque source ):

- $I_{\text{diffuse } i} = I_{\text{di}} * \frac{K_d}{n} = C_{\text{di}}$

Où  $I_{\text{di}}$  : est l'intensité diffuse de la lumière  $i$ .

$K_d = N \cdot L_i$  : est la contribution diffuse de la lumière  $i$ , tel que  $N$  est la normale au point  $M$ , et  $L_i$  est le vecteur entre le point  $M$  et la position de la lumière  $i$ .

$n$  : est la norme de  $N$ .

Le terme  $C_{\text{di}}$  pour une source  $i$ , ne sera calculé que si  $N \cdot L_i$  est positive.

- $I_{\text{spéculaire } i} = I_{\text{si}} * \left(\frac{K_s}{n}\right)^{\text{matériau}} = C_{\text{si}}$

Où  $I_{\text{si}}$  : est l'intensité spéculaire de la lumière  $i$ .

$K_s = \frac{D}{K_d \times N_{H_i}}$  : est la contribution spéculaire de la lumière  $i$ , tel que  $N_{H_i}$  est la norme d'un vecteur  $H_i = D + L_i$ .

matériau : est un facteur de rugosité.

Le terme  $C_{\text{si}}$  pour une source  $i$ , ne sera calculé que si  $N \cdot H_i$  est positive.

La couleur de l'échantillon courant (diffuse et spéculaire) est donnée par :

$$C' = \sum_{\text{Lumière } i} A_i \times \frac{(I_{\text{diffuse } i} + I_{\text{spéculaire } i})}{d_i^2}$$

Où :

$A_i$  : est l'atténuation du rayon d'ombrage lancé vers la source  $i$ .

$$d_i = \|L_i\|$$

Une fois la couleur de chaque échantillon est calculée, on peut calculer la couleur (diffuse + spéculaire) du voxel de la manière suivante:

$$C'' = \frac{\sum_{\text{échantillons}} N \cdot D \times C'}{\sum_{\text{échantillons}} N \cdot D}$$

Après obtention de la couleur (diffuse + spéculaire) du voxel, on peut calculer la couleur  $C_v$  d'un voxel en ajoutant la composante ambiante:  $C_v = (K_a * I_a + C'') * (1 - Att_v)$ .



## 6.2 Cumulation des couleurs et des atténuations

A partir de cette étape on peut calculer la couleur du rayon traversant le texel, tel que :

$$C_{\text{rayon}} = \sum_{\text{Voxels Traversés}} Att \times C_v$$

$$Att = \prod_{\text{Voxels Traversés}} Att_v$$

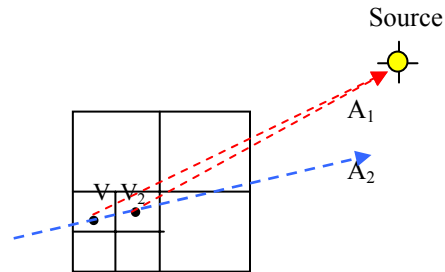
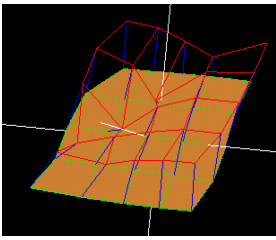
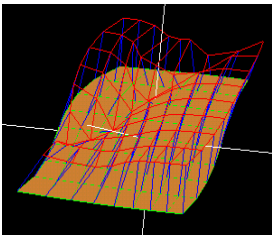
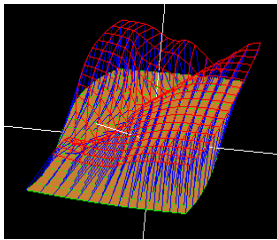


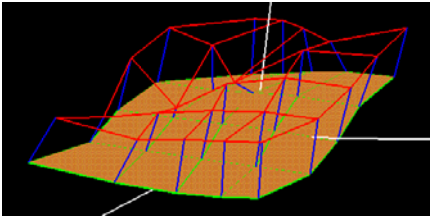
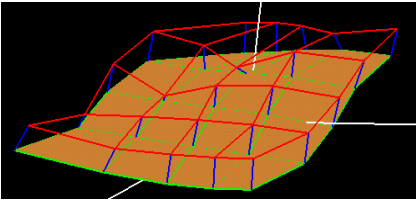
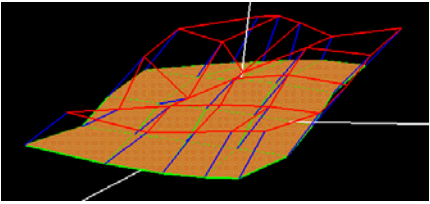
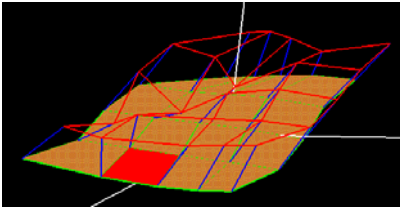
figure5 : Calcul de l'illumination C et de l'atténuation A du texel

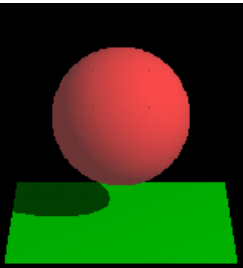
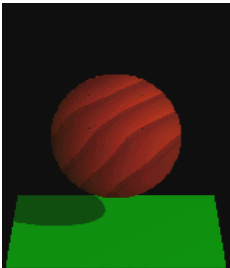
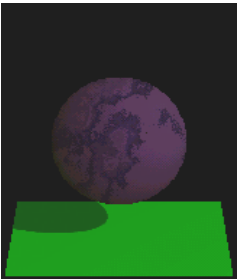
$A_1$  et  $A_2$  représentent l'atténuation d'un rayon d'ombrage lancé à partir de  $V_1$  et  $V_2$  respectivement.

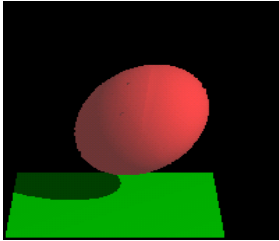
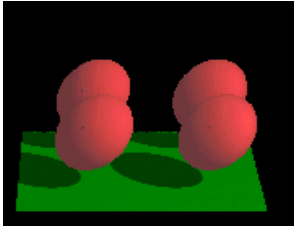
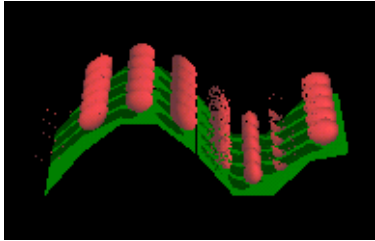
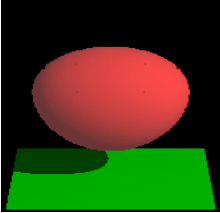
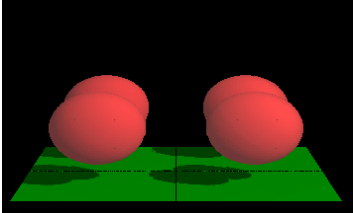
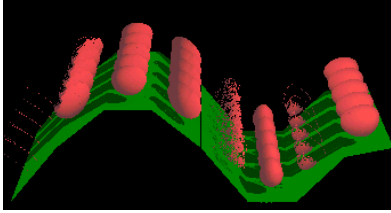
**7. Résultats:**

Après avoir implémenté un système permettant d'une part de générer des textures volumiques multi-échelles intégrant l'aspect illumination et d'autre part d'habiller puis effectuer le rendu de façon interactive, nous présentons dans cette partie quelques résultats générés :

			
Erreur arc	30 %	25 %	10 %
Erreur aire	80 %	40 %	20 %
Nb de sommets	25	81	289
Nb de patches	16	64	256
Temps de calcul	260 ms	330 ms	450 ms
Espace mémoire	2.148 KO	6.96 KO	24.84 KO

	
a : état initial.	b : manipulation des hauteurs ; ici, les amplitudes sont diminués uniformément.
	
c : perturbation de l'orientation des vecteurs normales ; orientation vers la droite.	d : Une manipulation locale d'une boite ; la sélection d'un patch, et le changement de l'orientation de deux vecteurs.
Coiffage de la peau.	

		
Calcul d'un texel en introduisant la déformation et les bruits de Perlin		

		
Temps de calcul : 4'33"	Temps de calcul: 10'15"	Temps de calcul: 24'20"
		
Temps de calcul : 5'40"	Temps de calcul: 12'26"	Temps de calcul: 34'53"
Texel déformé	Habillage d'une surface plate	Habillage d'une surface fortement déformée
Résultats obtenus par un parcours courbé du rayon		

Les résultats obtenus sur un PC P3 800MHz montre la validité de nos choix: Les images obtenues par un parcours courbé du rayons sont d'une meilleure qualité que celles obtenues à l'aide d'un parcours linéaire, néanmoins le temps de calcul reste très élevé.

## 7. CONCLUSION ET PERSPECTIVES :

Nous avons présenté dans cet article une méthode multi-résolution pour l'affichage efficace et visuellement acceptable de scènes complexes exhibant une grande répétitivité. Cette méthode a pour but d'éliminer l'élément géométrique de répétitivité de ces scènes en le remplaçant par un volume contenant l'information de réflectance de cette géométrie à plusieurs niveaux de résolution. Cette information est définie à l'aide d'une fonction paramétrable (l'ellipsoïde) capable de modéliser plusieurs types de formes.

Le volume de référence ainsi défini consiste en une structure de subdivision spatiale hiérarchique, l'octree, dont les feuilles contiennent l'information de réflectance obtenue par échantillonnage de la géométrie. Les niveaux supérieurs de l'octree sont obtenus par filtrages successifs des niveaux inférieurs. Une scène composée de copies virtuelles de ce volume, les texels, peut alors être créée, à l'aide de la méthode présentée, en effectuant un maillage optimal de la scène. Les texels (éléments de la peau volumique recouvrant la scène) sont affichés par un rendu volumique, durant lequel des cônes traversent le volume de référence afin d'accumuler l'illumination dans les cellules de ce volume, selon la distance entre le texel et le point de vue.

Cette approche présente une solution au problème de la complexité géométrique des scènes, le volume de référence est une représentation approximative de l'élément de répétitivité, mais le degré de cette approximation peut être contrôlé de façon à obtenir les résultats désirés.

En perspective, nous projetons :

- Le traitement des cas où il existe de fortes déformations des boîtes, par l'introduction des rayons courbés qui doivent être utilisés dans le rendu volumique de l'octree dans l'espace boîte.
- L'amélioration du temps de calcul, par la parallélisation de l'algorithme de rendu d'une part et par une meilleure utilisation des fonctionnalités des cartes graphiques actuelles.
- Proposer des modèles de textures volumiques dédiés à des objets spécifiques, ce qui permettra d'augmenter le réalisme et améliorer le coût de calcul [16,17].
- L'utilisation d'une nouvelle structure pour la codification du volume de référence (P-grille, LDC, surfels), permettant de minimiser la capacité de stockage et la qualité du rendu.

---

**Bibliographie**

- [1] F. Neyret. A general and multiscale method for volumetric textures. In *Graphics Interface '95 Proceedings*, pages 83–91, May 1995.
- [2] F. Neyret. Synthesizing verdant landscapes using volumetric textures. In *Eurographics Workshop on Rendering '96*, pages 215–224, June 1996.
- [3] F. Neyret. Modeling animating and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics*, January 1998.
- [4] A. Meyer and F. Neyret. Interactive volumetric textures. In *Eurographics Workshop on Rendering 1998*, pages 157–168, June 1998.
- [5] B. Levy and J.L Mallet. Non-distorted texture mapping for sheared triangulated meshes. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, pages 343–352.
- [6] Jérôme Maillot, Hussein Yahia, and Anne Verroust. Interactive texture mapping. In J. T. Kajiya, editor, *Computer Graphics Proceedings*, volume 27, pp 27–34, August 93.
- [7] Karim Ratib. Texture volumique multi- échelle pour l’affichage de scènes complexe, Mémoire de Maître ès sciences en informatique, *Université de Montréal*, Décembre 1997.
- [8] Lance Williams. Pyramidal Parametrics. In *Computer Graphics (SIGGRAPH 83 Proceedings)* volume 17, pages 1-11, juillet 1983.
- [9] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *SIGGRAPH 1989, Computer Graphics Proceedings*, pages 271–280, July 1989.
- [10] T. Noma. Bridging between surface rendering and volume rendering for multiresolution display. In *Eurographics Workshop on Rendering 1995*,
- [11] P. J. Frey et Paul-louis George. Maillages applications aux éléments finis Edition Hermès, Septembre 1999.
- [12] M.C.Babahenini et N.Djedi. Habillage de surfaces de forme libre à l’aide de textures bidimensionnelles par minimisation de l’énergie de déformation. *Proceedings of 3rd International Symposium on Programming and Systems 97*, pp 85-104.
- [13] A. S. Glassner, J. Arvo, R. L. Cook, E. Haines, P. Hanrahan, P. Heckbert, and D. B. Kirk. *An Introduction to Ray Tracing*. Academic Press, London, 1989.
- [14] P. Poulin and A. Fournier. A model for anisotropic reflection. In *SIGGRAPH 1990, Computer Graphics Proceedings*, pages 273-282, August 1990.
- [15] J. Mhasenfratz et D. Ghazanfarpour «Une synthèse des variantes du lancer de rayon et du lancer de faisceaux ». *Revue international de CFAO et d’informatique graphique*. Vol. 13, Num 3- septembre 1998, pp. 235-264. *Computer Graphics Proceedings*, pages 273–282, August 1990.
- [16] A. Meyer and F. Neyret. Multiscale shaders for the efficient realistic rendering of pine-trees. In *Graphics Interface*, May 2000.
- [17] A. Meyer, F. Neyret, and P. Poulin. Interactive rendering of trees with shading and shadows. In *Eurographics Workshop on Rendering*, Jul 2001.