

# Visualisation de modèles CAO dans une application de réalité virtuelle immersive

D. Paillot<sup>1</sup>, F. Merienne<sup>1</sup>, M. Neveu<sup>2</sup>, J.P. Frachet<sup>1</sup>.

(1) : Institut Image-ENSAM  
2, rue Thomas DUMOREY – BP 123  
71321 Chalon sur Saône  
33 (0)3 85 90 98 60 / 33 (0)3 85 90 98 61  
E-mail :  
{paillot,merienne,frachet}@cluny.ensam.fr

(2) : Université de Bourgogne – LE2I  
UFR Sciences et Techniques – Bât. Mirande  
Aile de l'Ingénieur – BP 47870  
21078 Dijon Cedex  
33 (0)3 80 39 58 45 / 33 (0)3 80 39 58 46  
E-mail : mneveu@u-bourgogne.fr

**Résumé :** *Les applications de réalité virtuelle se développent de plus en plus. La quasi totalité des secteurs industriels peuvent aujourd'hui tirer profit de la réalité virtuelle. Les attentes des utilisateurs sont en augmentation, plus d'interactivité avec la scène graphique, meilleur rendu, meilleure fluidité de l'application,... Bien que les performances des calculateurs continuent d'augmenter, il est toujours nécessaire d'optimiser les scènes 3D. Le principal et premier problème pour optimiser les scènes reste les modèles qui seront utilisés dans l'application. L'origine de ces modèles, leur qualité et leur conversion en un format compatible avec ceux utilisés par les applications de réalité virtuelle restent les problématiques majeures. Nos travaux ont pour objectif de visualiser en temps réel des modèles issus de la CAO dans une salle immersive de type CAVE™. Ces travaux couvrent deux axes de recherche principaux. Le premier concerne la mise en conformité, la simplification des modèles surfaciques en modèles triangulés. Le deuxième axe de recherche se consacre à la visualisation et à son optimisation pour l'utilisation d'un périphérique particulier : salle immersive.*

**Mots-clés :** Triangulation, simplification, visualisation réaliste, temps réel, salle immersive.

## 1. Introduction.

Les cartes graphiques actuelles ne savent gérer que des polygones, principalement des triangles. Pour cette raison les applications de réalité virtuelle utilisent des modèles polygonaux plutôt que des modèles surfaciques. Dans la mesure où les modèles doivent être créés pour une application de réalité virtuelle spécifique, ils sont parfois générés directement dans ce format. Mais bien souvent, il est souhaitable d'utiliser des données issues d'autres corps de métiers (design, conception assistée par ordinateur, ...). Cette réutilisation évite d'avoir à redessiner les modèles et de bénéficier à tous moments de la dernière mise à jour des modèles. Cependant, cela impose un travail de préparation, de réparation, d'optimisation pour chacun de ces modèles. Cette mise en conformité est nécessaire pour rendre les modèles utilisables par les applications et de les optimiser pour une application de visualisation. Bien que les capacités des calculateurs se soient accrues, il est toujours incontournable de réduire le nombre de polygones d'une scène de réalité virtuelle. Les applications développées sont principalement des applications de revue de projet. La finalité n'est pas d'étudier la fonctionnalité ou la définition structurelle, mais de visualiser les pièces dans leur environnement. La visualisation sera réalisée dans une salle immersive de type MoVE™ (Modular Virtual Environment). Ce périphérique particulier offre une place unique à l'utilisateur. Il est immergé au cœur de la scène 3D, alors que tous les autres périphériques existants placent l'utilisateur en dehors de la scène. L'immersion est totale et réelle. Cette configuration amène à repenser les applications de visualisation. Actuellement, les logiciels offrent des solutions pour les périphériques classiques. Leur fonctionnement pourrait être modifié pour s'implémenter correctement et être encore plus performant dans la gestion du nombre de polygones et de la qualité visuelle offerte. En particulier, la notion de vision périphérique semble être utilisable pour les salles immersives. Comme l'utilisateur est équipé d'un capteur de position, l'application connaît à tout moment ce que regarde l'utilisateur. L'application peut ainsi en temps réel dégrader les modèles situés à la périphérie du champ visuel et affiner ceux situés dans la partie centrale de la vision.

Le chapitre 2 de l'article présentera les enjeux, les problèmes rencontrés et les solutions développées pour apporter une réponse au transfert des modèles. Le troisième chapitre abordera la visualisation et

particulièrement la vision périphérique, ses apports éventuels ainsi que sa mise en œuvre. La conclusion et les perspectives feront l'objet de la quatrième partie.

## 2. Conversion de modèles surfaciques en modèles polygonaux.

### 2.1. Problématique.

L'utilisation des modèles issus de la CAO offre l'avantage de travailler avec des modèles constamment en phase avec le cycle de vie du produit. Cependant cette réutilisation n'est pas directe. Il est impossible de visualiser les modèles issus de la CAO dans une application de réalité virtuelle sans les convertir. Les étapes de conversion sont actuellement bien connues :

- Triangulation,
- Cohérence,
- Réduction du nombre de polygones,
- Suppression des éléments non visibles.

Ces quatre étapes sont présentées dans les sous-chapitres suivants.

### 2.2. Triangulation.

La triangulation consiste à générer un modèle polygonal à partir d'un modèle surfacique. Pour ce faire, on utilise un critère appelé erreur cordale (cf. Figure 1). Cette erreur porte souvent le nom de SAG dans les logiciels. Elle correspond à la distance maximale autorisée entre la facette générée et la courbe.



Figure 1 : Erreur cordale.

Cette erreur conditionne le nombre de polygones du modèle final. Plus cette erreur sera faible, plus le modèle polygonal sera fidèle au modèle initial mais plus le nombre de polygone risque d'être élevé. Il faut trouver la valeur optimum entre nombre de polygones et finesse géométrique. Le nombre de polygones peut être sensiblement élevé puisqu'une étape de simplification aura lieu ultérieurement.

### 2.3. Mise en cohérence.

Lors de la visualisation, il est fréquent de voir apparaître des trous ou discontinuités dans les modèles (cf. Figure 2(b)). La mise en cohérence consiste à rendre le modèle polygonal conforme pour une visualisation correcte du modèle. Ces discontinuités sont dues à la non-concordance des sommets entre eux (cf. Figure 2 (a)). Cette non-concordance crée des discontinuités dans le maillage apparaissant comme des trous.

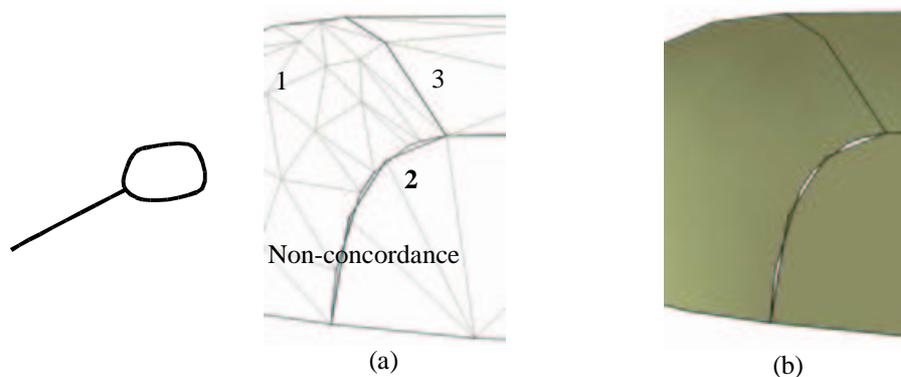


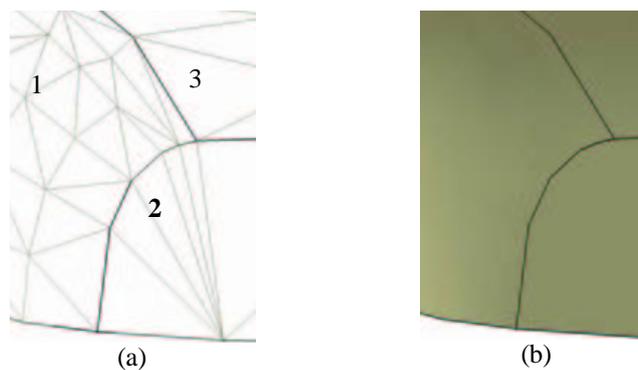
Figure 2 : Problème de cohérence des modèles triangulés.

Ce problème de discontinuïté peut être r solu de deux mani res diff rentes.

La premi re solution consiste   travailler sur le mod le surfacique. Les mod les surfaciques sont un assemblage de carreaux. La Figure 2(a) montre un mod le surfacique avec ses trois carreaux. Ces carreaux ont en commun des fronti res, gr ce   ses fronti res le mod le surfacique poss de une unit  et permet d'obtenir un rendu de bonne qualit  sans trou ou discontinu t .

Ces carreaux sont triangul s un par un. La connaissance des fronti res communes est alors perdue. Chaque carreau poss de sa propre triangulation. Cela aura pour effet de ne pas avoir de coh rence au niveau du mod le final. Les Figure 2(a) et (b) illustrent le probl me de continuit  de triangulation le long des fronti res. Les carreaux 1 et 2 ont subi chacun leur propre triangulation. Comme la notion de fronti re est perdue, la fronti re commune entre le carreau 1 et le carreau 2 sera triangul e de deux mani res diff rentes. Les sommets de la fronti re appartenant au carreau 1 n'ont aucune raison a priori de concorder avec les sommets de la fronti re appartenant au carreau 2. La cons quence de ce manque de coh sion est une apparition de trous et de discontinu t s lors du rendu des pi ces (cf. Figure 2(b)).

La solution   ce niveau est de coudre les diff rents carreaux surfaciques entre eux. Cette couture permet de rassembler sous une seule entit  les diff rents carreaux surfaciques initiaux. Cette  tape peut se faire dans le logiciel de conception du mod le. A titre d'exemple, une simple pi ce plastique d'int rieur de v hicule peut atteindre 200 carreaux. Il n'est pas possible de tous les s lectionner et de les coudre en une seule op ration. Il faut bien souvent traiter le mod le morceau par morceau. Cette t che devient vite longue et fastidieuse. Le r sultat de la triangulation pr sent  sur la Figure 3 montre un exemple de concordance souhait e pour les sommets ainsi que la qualit  de la continuit  du rendu.



**Figure 3 : Mod le cousu.**

La deuxi me solution consiste   intervenir apr s la triangulation. A partir d'un crit re de distance, les sommets et ar tes dont la distance qui les s pare est inf rieure   ce crit re seront fusionn s. Cette action est plus rapide et moins contraignante   r aliser que la premi re solution. Cependant,   ce niveau il n'existe plus de r f rence. Le mod le surfacique initial n'est plus connu. Il est possible d'avoir des d rives non souhait es sur la g om trie ou topologie du mod le.

Quelle que soit la m thode utilis e, la qualit  du mod le triangul  sera primordiale pour la suite. En particulier, le r sultat de la simplification sera fonction de la coh rence du mod le initial.

#### 2.4. R duction du nombre de polygones.

Face   ce probl me de r duction du nombre de polygones, de nombreux algorithmes ont  t  d velopp s. Chacun d'entre eux poss de sa sp cificit . Plusieurs techniques de g n ration de niveaux de d tails ont  t  d finies par Heckbert et Garland [HG94]. Il est possible de d finir trois cat gories d'algorithmes pour la cr ation de niveaux de d tails. La premi re cat gorie regroupe les algorithmes qui font appel   des simplifications polygonales en g n rant de nouveaux mod les avec des niveaux de d tails diff rents. La seconde concerne les algorithmes qui remplacent les objets par des textures. Ces algorithmes ne nous seront pas utiles ici, puisque nous nous int ressons   l'immersion 3D. Les textures ne sont pas tr s appropri es pour la vision en relief. La troisi me cat gorie repr sente les algorithmes qui agissent sur la sc ne et qui remplacent un ensemble d'objets par une repr sentation plus simple.

Nous ne nous intéressons dans un premier temps qu'à la première catégorie. Dans cette catégorie plusieurs approches existent :

1. Subdivision adaptative (*Adaptative Subdivision*) : cette méthode consiste à construire un modèle de base très simple qui sera ensuite subdivisé. L'algorithme s'arrête lorsque l'écart entre le modèle initial et le modèle subdivisé est inférieur au critère spécifié par l'utilisateur. Cette méthode est peu utilisée à cause de la complexité à définir le modèle de base.
2. Réduction géométrique (*Geometrical Removal*) : cette méthode s'appuie sur le modèle d'origine et supprime des faces ou des sommets. L'algorithme s'arrête lorsque le degré de satisfaction imposé par l'utilisateur est atteint. La majorité des algorithmes respecte la topologie des modèles initiaux. La plupart des algorithmes récents fonctionnent avec cette méthode.
3. Échantillonnage (*Sampling*) : cette méthode effectue un échantillonnage de la géométrie du modèle initial (en choisissant arbitrairement des sommets ou en l'englobant dans une grille tridimensionnelle et en échantillonnant chaque boîte de la grille). L'algorithme essaye de créer un modèle simplifié qui soit proche des données échantillonnées. Le contrôle se fait par le nombre de points ou la taille de la grille.

De manière générale, parmi les méthodes de décimation de polyèdres bon nombre d'entre elles sont contrôlées par des critères d'arrêts directement liés au nombre de sommets désirés par l'utilisateur [SZL92], [HG94], [RO96]. Ce critère n'est dans notre cas d'étude pas satisfaisant. En effet, l'écart géométrique généré doit être connu et même maîtrisé. D'autres méthodes permettent un contrôle de la déviation entre le maillage résultant et le maillage initial [DFMP96], [G96], [PS97]. Ces techniques sont d'utilisation plus délicate ou bien concernent des modèles particuliers (modèles de terrain). Les paramètres de contrôle (distance et angle) ne sont pas faciles à définir par les utilisateurs.

Les principes de simplification de polyèdres reposent sur l'emploi d'opérateurs de fusion d'arêtes [H96] ou bien de remaillage d'un contour après suppression de sommet [PS97], [V97a]. Les opérateurs de fusion d'arêtes n'engendrent qu'un nombre limité de configurations de maillage du contour du nœud supprimé ce qui réduit la qualité de la restitution de la forme du polyèdre simplifié.

#### 2.4.1. Analyse multirésolution de maillage (Eck [EDRDHLS95]).

La méthode consiste à reproduire une représentation à résolutions multiples d'un maillage polygonal quelconque. Les représentations à résolutions multiples sont basées sur une technique introduite par De Rose et al. [DRLW93] appelée analyse multirésolution. Elles consistent en un maillage de base simple et une séquence de corrections locales sous formes de coefficients d'ondelettes.

#### 2.4.2. Maillages progressifs (Hoppe [H96]).

Une représentation en maillage progressif d'un maillage arbitraire  $M'$  contient un maillage moins détaillé  $M^0$  et une séquence de  $n$  enregistrements de détails qui précise comment reconstruire de manière incrémentale  $M'$  à partir de  $M^0$  :  $M' = M^n$ . Un des points très importants de cette technique est que le degré de simplification est contrôlé par le choix du nombre de facettes désiré. De plus, il est possible de générer tous les niveaux de détails en un seul traitement.

#### 2.4.3. Approximation de polyèdre triangulaires (Ronfard [RR96]).

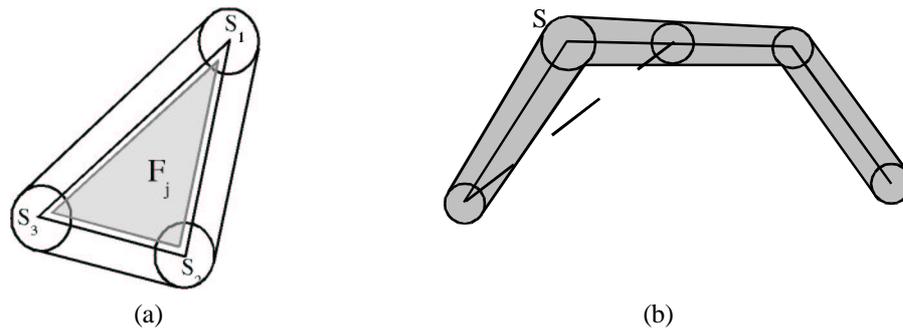
Cet algorithme, présenté à Eurographics'96, est issu des techniques de segmentation d'images. Il utilise un opérateur de fusion de régions sur des modèles triangulés. Cet opérateur retire une arête en fusionnant à la fois ses extrémités et plusieurs triangles. Cette méthode est capable de supprimer les détails les plus petits (et ce uniquement à base d'un raisonnement géométrique), elle produit tous les niveaux de détails en une seule passe. En revanche, l'implémentation actuelle est un peu lente.

#### 2.4.4. Simplification multicritères (Véron [V97b]).

Les principes de simplification de polyèdres retenus sont basés sur un critère de distance entre le polyèdre initial et le polyèdre simplifié.

L'approche utilisée est une suppression répétée d'un nœud du polyèdre suivie par un remaillage du contour en fonction des courbures locales du polyèdre.

L'algorithme fonctionne à partir de zones d'erreurs. Ces zones sont géométriquement représentées par des sphères centrées sur les nœuds du polyèdre initial et constituent une enveloppe entourant le polyèdre initial. La Figure 4 représente dans le plan le concept de zones d'erreurs. La Figure 4(a) illustre le placement des sphères par rapport aux sommets ainsi que l'enveloppe d'erreur générée. La Figure 4(b) illustre le principe de suppression, ici, le sommet S ne pourra pas être supprimé car la nouvelle arête est placée en dehors de l'enveloppe.

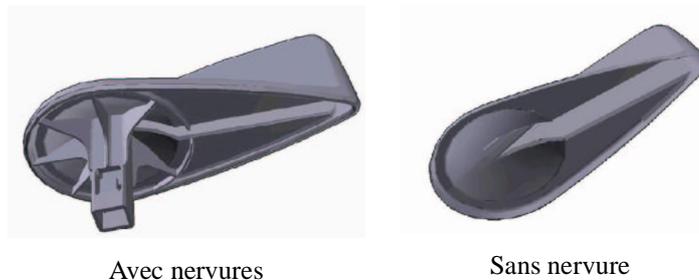


**Figure 4 : Définition des zones d'erreurs.**

La dimension des zones d'erreur (rayon des sphères) est le seul paramètre fourni par l'utilisateur pour effectuer la simplification.

#### 2.5. Suppression des éléments non visibles.

Cette dernière étape dans le processus de conversion est une étape d'optimisation. L'objectif est toujours de limiter le nombre de polygones aux polygones utiles à l'application considérée. Dans notre application, il s'agit de visualisation. Les pièces utilisées sont des pièces issues des bureaux d'études, elles comportent toutes les informations utiles à leur fabrication et à la fonction ultime. Beaucoup d'éléments utiles pour le fonctionnement ne le sont pas pour la visualisation. Par exemple, les nervures et raidisseurs de pièces plastiques, les trous de fonctionnement sont des éléments qui pour leur définition géométrique vont utiliser des polygones mais qui n'ont pas d'apport réel dans une application de visualisation. Afin de diminuer le nombre de polygones dans la scène, il semble nécessaire de les supprimer (cf. Figure 5).



**Figure 5 : Suppression des parties non visibles.**

## 2.6. Mise en œuvre.

La chaîne complète de conversion d'un modèle surfacique en un modèle polygonal peut être décrite comme sur la Figure 6. Cette chaîne représente la première solution, la mise en cohérence est effectuée au niveau du modèle surfacique.

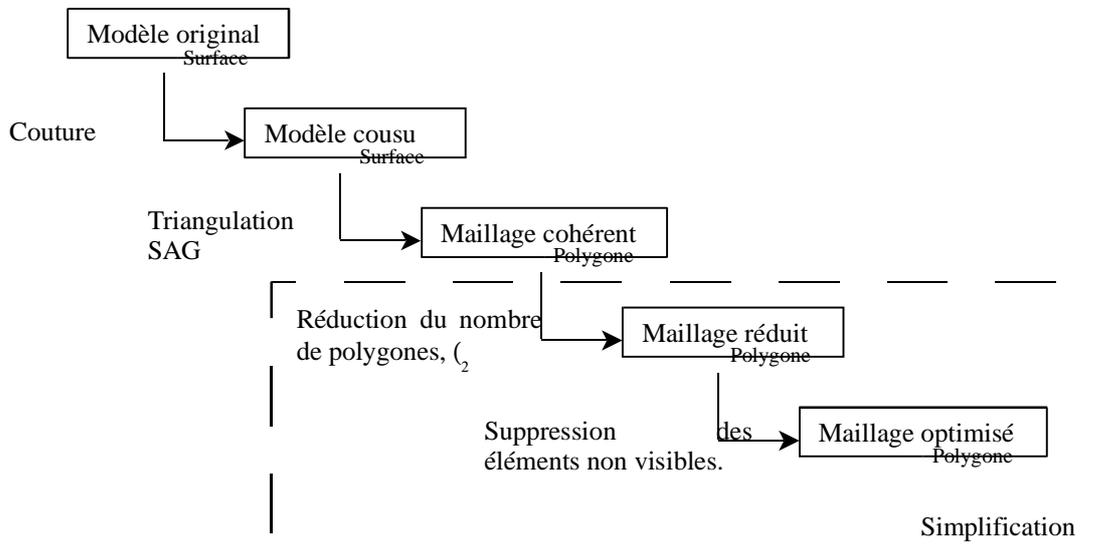
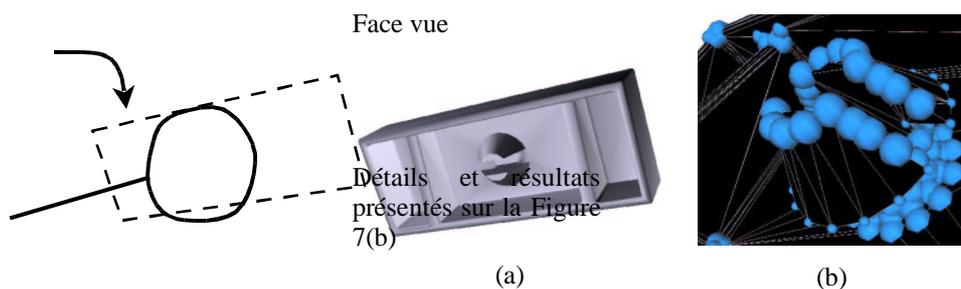


Figure 6 : Chaîne de conversion modèle surfacique / modèle polygonal.

Les étapes de couture et de triangulation sont réalisées par le modelleur CAO. Pour la simplification, nous avons fait le choix de travailler en collaboration avec le laboratoire 3S (Sols Solides et Structures) dirigé par Jean-Claude LEON à Grenoble et la société Géo-Numéric sur la base du logiciel *Simpoly*<sup>®</sup>. Ce logiciel est le fruit de plusieurs thèses réalisées au sein du laboratoire 3S. Nous avons effectué ce choix suite à l'étude des algorithmes existants. Il s'est avéré qu'ils ne répondaient pas entièrement à notre demande. Principalement parce que nous souhaitions pouvoir piloter la simplification non uniformément sur tout le modèle. Les travaux de P. VERON [V97b-VL98] présentent une approche différente de la simplification. Leur algorithme permet de simplifier de manière non uniforme.

L'objectif de leur outil de simplification est de générer des modèles pour de la simulation de calculs. Ils avaient besoin de simplifier différemment les zones d'un modèle en fonction de leur implication dans les calculs de résistance. Par ailleurs, la majeure partie des algorithmes de simplification existants utilise comme critère de pilotage un pourcentage de réduction. L'utilisateur indique s'il souhaite une réduction de X%, l'algorithme respectera ce critère et générera un modèle. L'écart géométrique, l'erreur commise, sera une conséquence du taux de réduction. *Simpoly*<sup>®</sup> propose de piloter la simplification avec l'erreur géométrique maximale tolérée par l'utilisateur. Le taux de réduction devient une conséquence du calcul. Dans notre cas, il est impératif de contrôler finement cet écart géométrique. Les modèles ne doivent pas avoir un écart géométrique supérieur au millimètre pour que l'application soit crédible. La simplification non uniforme est pilotée par une carte de taille. Cette carte de taille est une répartition de sphères dans l'espace qui permet d'indiquer pour chaque sommet sa latitude à disparaître ou à se déplacer (cf. 2.4). Actuellement, les modules existants génèrent des cartes de tailles dans le l'objectif d'effectuer des calculs de résistance.

Nous développons les modules qui permettent de piloter l'algorithme de simplification dans l'objectif de générer des modèles pour des applications de visualisation. Notre objectif est de conserver les détails visibles du modèle et de simplifier les zones non visibles. La simplification des zones non visibles peut aller jusqu'à la modification géométrique de certaines parties. Le problème est d'identifier ces différentes zones. Le premier algorithme [PMFNTF02] implémenté travaille à partir de l'angle solide entre les normales des facettes et la direction de vue du modèle. Pour un angle solide compris entre  $- \pi/2$  et  $+ \pi/2$  la facette sera déclarée comme visible. La Figure 7 illustre cet algorithme. La pièce traitée est vue par la face dessinée en pointillée sur la Figure 7(a).



**Figure 7 : Orientation des normales.**

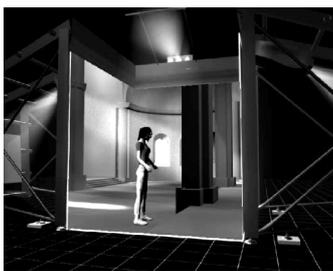
Cet algorithme n'est pas assez robuste. Le clip de fixation n'a pas été complètement détecté (cf. Figure 7(b)). Nous nous orientons actuellement sur des algorithmes de partitionnement de l'espace ou de profondeur. Nous recherchons à connaître les faces cachées par les autres. Avec cette connaissance, il sera aisé de pouvoir supprimer les sommets des faces non visibles. La priorité dans le traitement des sommets sera faite par l'importance de la concavité entre les faces. Plus la concavité sera grande, plus il semble difficile de voir ce qui se passe à l'intérieur de cette concavité.

Cette préparation des modèles est l'étape incontournable avant la visualisation dans une application de réalité virtuelle. Tout ce travail a pour objectif de pouvoir donner au calculateur le temps de traiter la totalité des polygones contenus dans la scène dans le temps qui lui est imparti. Les applications sont en stéréo et en temps réel. Il faut que le calculateur puisse calculer entre 30 et 50 images par seconde pour un confort maximal.

### 3. Visualisation et vision.

#### 3.1. Matériel utilisé.

Les applications sont portées dans la salle immersive MoVE™. Cette salle immersive de type CAVE™ est reconfigurable et possède 4 faces (deux faces latérales, la face avant et le sol). Comme le montre la Figure 8, l'utilisateur est physiquement immergé dans la scène 3D.

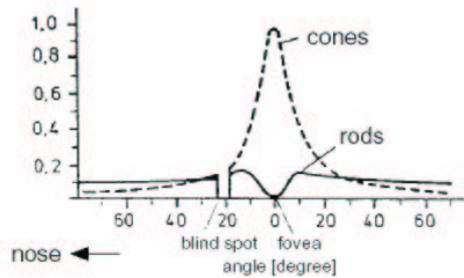


**Figure 8 : MoVE.**

Habituellement l'utilisateur perçoit les images de l'application sur un écran ou dans un casque, dans le cas d'un écran l'utilisateur a toujours dans son champ visuel des éléments réels (bureaux, meubles, ...), dans le cas d'un casque la totalité du champ visuel n'est pas couvert. Dans la salle immersive, l'utilisateur est physiquement au cœur de la scène 3D.

### 3.2. Système visuel humain.

L'œil humain n'est pas sensible de la même manière à tout son environnement. Le récepteur de l'œil, la rétine, est composé d'une zone, la fovéa, représentant 1% de la surface de la rétine. La densité de cônes dans cette zone est de 180000 par mm<sup>2</sup> alors qu'elle est de 5000 sur le reste de la rétine. Les cônes servent principalement dans la vue diurne et apportent des informations précises sur la position et la couleur des objets. Les bâtonnets quant à eux sont utilisés pour une vision nocturne ou de faible luminosité. On s'aperçoit donc que la zone où l'information est maximale est faible. Cette faiblesse est compensée par le mouvement rapide des yeux. L'acuité visuelle est la capacité pour un œil humain à distinguer les détails (i.e. résolution). Elle est évaluée à une minute d'arc. Cette acuité visuelle a été établie pour la ligne de vue qui correspond à la fovéa.

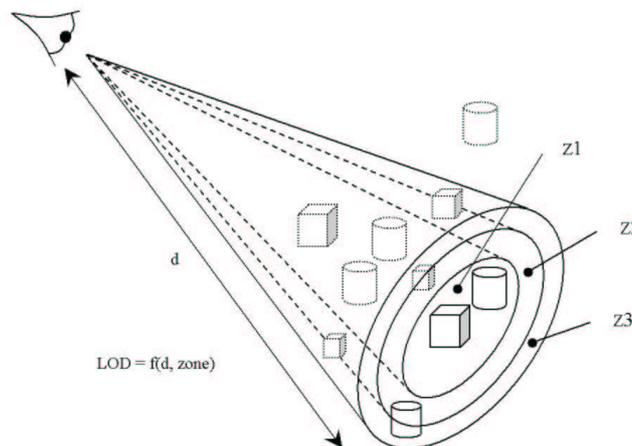


**Figure 9 : Résolution spatiale des cônes et bâtonnets en fonction du rayon angulaire par rapport à la fovéa (source [HVPO]).**

La Figure 9 montre que l'acuité visuelle maximale ( $A_v=1$ ) ne représente que 1% du champ de vision, c'est à dire une surface d'environ 2° de rayon angulaire. L'acuité visuelle diminue très vite en fonction de l'écartement de la fovéa, elle est divisée par 2 à 2° du centre de la fovéa, par 4 à 5° et par 10 pour des angles variant de 10 à 20° [SUPELEC-YRR94]. Ces constatations confirment qu'une simplification plus importante des modèles situés à la périphérie du champ visuel ne modifiera pas a priori la perception visuelle de la scène globale.

### 3.3. Mise en œuvre.

Le fait de simplifier d'avantage les modèles situés sur la périphérie du champ visuel permettra de diminuer le nombre de polygones à calculer pour chaque image et ainsi accroître les performances de l'application. Les logiciels actuels gèrent les niveaux de détails de manière identique. Il est fonction de la distance séparant l'œil de l'utilisateur à l'objet considéré. Dans notre cas, nous avons vu qu'un objet qui serait proche de l'œil mais situé à 15° de l'axe de vision principal serait vu avec 10 fois moins de détails qu'un objet situé sur l'axe. Un pilotage des niveaux de détails ne fonctionnant qu'avec la distance œil-objet ne serait pas satisfaisant. Il serait possible d'optimiser la scène afin d'augmenter les performances de l'application. Nous proposons de définir une autre fonction de choix des niveaux de détails (cf. Figure 10).



**Figure 10 : Cônes de vue.**

Le niveau de détail d'un objet ne doit plus être choisi uniquement en fonction de son éloignement par rapport à l'œil, mais aussi en fonction de sa position angulaire.

Cette approche est soumise à la condition que la direction du regard coïncide avec la direction de la tête de l'utilisateur qui est connue (capteur de position). Cette approximation est acceptable puisque l'utilisateur est muni de lunettes stéréoscopiques qui lui obturent une grande partie du champ visuel. On constate que les utilisateurs ont tendance à tourner la tête plutôt que le regard. L'assimilation de la direction de la tête et du regard est dans ce cas valide.

L'approche envisagée actuellement est d'utiliser les arbres binaires de partitionnement de l'espace (*BSP : Binary Space Partition*). Ce partitionnement de l'espace permet de savoir en temps réel en fonction de la direction de la caméra dans quelle zone de l'espace se situe un objet. En fonction de l'appartenance de l'objet à une zone, le niveau de finesse du modèle est choisi et affiché.

#### 4. Conclusions et perspectives.

Les dispositifs de réalité virtuelle imposent des contraintes de temps et de qualité de visualisation très sévères. Aussi, des compromis sont nécessaires afin d'optimiser l'immersion virtuelle de l'utilisateur. L'étude des caractéristiques de la vision humaine montre qu'il n'est pas nécessaire d'afficher une image avec un rendu homogène dans l'espace. La notion de zones d'intérêts doit être développée en fonction de la distance de l'objet virtuel à l'œil mais également en fonction de l'écart à l'axe visuel principal. Par conséquent, il est opportun de prendre en compte ces caractéristiques lors du calcul de rendu des objets pour leur affichage dans la scène virtuelle. L'objet calculé pour l'affichage ne doit tout d'abord comporter que les faces utiles à la visualisation de l'objet. Cette première étape de simplification est déjà traitée par les mécaniciens pour leurs calculs de structures et peut être adaptée à la visualisation. L'objet calculé doit ensuite subir certaines opérations de simplification dont le degré doit être fonction de la distance de l'objet à l'œil ainsi que de la zone d'intérêt de l'objet. Les travaux présentés s'inscrivent dans le cadre du développement de méthodes et outils permettant d'optimiser l'immersion virtuelle eu égard à ces critères. Ces travaux prennent également en compte le contexte technologique fort du dispositif de visualisation MoVE utilisé. Il s'agit alors d'intégrer les méthodes en développement dans les outils logiciels disponibles ou en évaluation rapide sur le marché.

#### Références :

- [DFMP96] L. DE FLORIANI, P. MARZANO et E. PUPPO, *Multiresolution models for topographic surface description*, The visual computer, vol. 12, n°7, pp 317-345, 1996.
- [DRLW93] T. D. DE ROSE, M. LOUNSBERY et J. WARREN, *Multiresolution analysis for surface of arbitrary topological type*, report 93-10-05, Department of Computer Science, University of Washington, Seattle, WA, 1993.
- [EDRDHLS95] M. ECK, T. DE ROSE, T. DUCHAMP, H. HOPPE, M. LOUNSBERY et W. STUETZLE, *Multiresolution Analysis of Arbitrary Meshes*, SIGGRAPH '95, 1995.
- [G96] A. GUEZIEC, *Surface simplification inside a tolerance volume*, rapport technique RC 20440, IBM research division, T.J. Watson Research Center, USA, Mai 1996.
- [HG94] P. HECKBERT et M. GARLAND, *Multiresolution modeling for fast rendering*, Proceedings of Graphics Interface '94, pages 43-50, Banff, Alberta, Canada. Canadian Information Processing Society, 1994.
- [H96] H. HOPPE, *Progressive Meshes*, Computer Graphics, Vol. 30, Annual Conference Series, pp. 99--108, 1996.
- [HVPO] Human Visual Perception Overview : <http://www.stanford.edu/class/ee392c/lectures/chapter05.pdf>
- [PMFNTF02] D. PAILLOT, F. MERIENNE, J.-P. FRACHET, M. NEVEU, S. THIVENT, L. FINE, *Revue de projet immersive pour le style automobile*, Virtual Concept 2002, Biarritz, pages 92-97, 2002.
- [PS97] E. PUPPO et R. SCOPIGNO, *Simplification, LOD and Multiresolution Principles and Applications*, Eurographics'97, 1997.

- [RO96] K.J. RENZE et J.H. OLIVIER, *Generalized Unstructured Decimation*, IEEE Computer Graphics and Applications, pp 24-32, Nov. 1996.
- [RR96] J.R. RONFARD et P. ROSSIGNAC, *Full-range approximation of triangulated polyhedra*, Technical report RC 20423, IBM research division, T. J. Watson Research Center. Also in Eurographics'96, 1996.
- [SUPELEC] <http://www.supelec-rennes.fr/ren/perso/jweiss/tv/perception/percept3.html>
- [SZL92] W.J. SCHROEDER, J.A. ZARGE et W.E. LORENSEN, *Decimation of triangle meshes*, ACM SIGGRAPH 92, Chicago, pp 65-70, 26-31 juillet 1992.
- [V97a] P. VERON et J.-C. LEON, *Static polyhedron simplification using error Measurements*, Computer Aided Design, Vol. 29, n° 4, pp 287-298, Avril 1997.
- [V97b] P. VERON, *Techniques de simplification de modèles polyédriques pour un environnement de conception mécanique*, thèse de doctorat, INPG, Grenoble, 1997.
- [VL98] P. VERON, J.-C. LEON, *Shape preserving polyhedral simplication with bounded error*, Computer & Graphics, Vol. 22, N°5, pp. 565-585, 1998.
- [YRR94] Akitoshi YOSHIDA, Jannick P. ROLLAND, John H. REIF, *Design and Applications of a High Resolution Insert Head Mounted Display*, June 1994.