



HAL
open science

Gestion de données pour la visualisation de maquettes numériques en environnement virtuel immersif: application au design automobile

Damien Paillot

► **To cite this version:**

Damien Paillot. Gestion de données pour la visualisation de maquettes numériques en environnement virtuel immersif: application au design automobile. Computer Science [cs]. Arts et Métiers ParisTech, 2004. English. NNT : 2004ENAM0031 . pastel-00001563

HAL Id: pastel-00001563

<https://pastel.hal.science/pastel-00001563>

Submitted on 17 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ecole Nationale Supérieure d'Arts et Métiers
Centre de Cluny

THÈSE

présentée pour obtenir le titre de

DOCTEUR

de

L'ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS

Spécialité : Informatique

par

Damien PAILLOT

**GESTION DE DONNEES POUR LA VISUALISATION DE
MAQUETTES NUMERIQUES EN ENVIRONNEMENT VIRTUEL IMMERSIF.
APPLICATION AU DESIGN AUTOMOBILE**

soutenue le 29 novembre 2004 devant le jury composé de

Mr.	J.C.	LEON	<i>Professeur à l'INPG, Grenoble</i>	<i>Président</i>
Mr.	C.	CHAILLOU	<i>Professeur à Polytech'Lille, Lille</i>	<i>Rapporteur</i>
Mme.	S.	COQUILLART	<i>Directeur de Recherche INRIA, Grenoble</i>	<i>Rapporteur</i>
Mr.	F.	MERIENNE	<i>Professeur à l'ENSAM, Cluny</i>	<i>Examineur</i>
Mr.	M.	NEVEU	<i>Professeur à l'Université de Bourgogne, Dijon</i>	<i>Examineur</i>
Mr.	S.	THIVENT	<i>Responsable des systèmes de Réalité Virtuelle à la Direction Informatique, PSA Peugeot-Citroën, Vélizy</i>	<i>Examineur</i>

GESTION DE DONNEES POUR LA VISUALISATION DE MAQUETTES NUMERIQUES EN ENVIRONNEMENT VIRTUEL IMMERSIF. APPLICATION AU DESIGN AUTOMOBILE.

RESUME: Les techniques de la réalité virtuelle constituent de nouveaux outils pour les métiers de la conception de produits manufacturés et offrent des gains en terme d'innovation, de temps et d'argent. L'enjeu industriel est l'intégration de ces techniques dans le cycle de conception du produit. Ces techniques doivent ainsi s'interfacer avec les différents corps de métiers et pouvoir être en phase avec le rythme des mises à jour des modèles numériques. La problématique technologique posée est relative à la gestion des données géométriques produites par les bureaux d'études en conception assistée par ordinateur. Ces données sont exhaustives, très nombreuses et ne sont pas manipulables dans un dispositif de réalité virtuelle en raison des contraintes de temps de calcul. Une maquette numérique, possédant une quantité d'informations maîtrisée et dédiée à la réalité virtuelle, est alors nécessaire. Ces travaux ont pour objectif d'apporter des méthodes et outils permettant l'utilisation des techniques de l'immersion virtuelle par les métiers du design automobile. Une chaîne de conversion, simplification et optimisation des données en fonction des critères des métiers du design automobile est proposée pour la création d'une maquette numérique dédiée à l'immersion virtuelle. La gestion en temps réel des données de cette maquette numérique est effectuée en fonction de critères de visualisation de la scène dans un contexte d'immersion virtuelle.

Mots-clés: *réalité virtuelle, maquette numérique, traitement de données 3D, conversion de modèles CAO, automatisation, système visuel humain, optimisation, niveaux de détail*

DATA MANAGEMENT FOR DIGITAL MOCK-UP VISUALIZATION IN IMMERSIVE VIRTUAL ENVIRONMENT. APPLICATION TO CAR DESIGN.

ABSTRACT: Virtual Reality techniques constitute new tools for the trades of the design of manufactured products and offer profits in term of innovation, time and of money. The industrial stake is the integration of these techniques in the product design process. These techniques must thus be interfaced with the various trade associations and be able to be in phase with the flow of the digital models updates. The technological problems relate to the management of the geometrical data produced by the engineering and design departments in computer-aided design. These data are exhaustive, very numerous and are not easy to handle in a device of virtual reality because of the time constraints of calculation. A digital mock-up, having a quantity of information controlled and dedicated to virtual reality, is then necessary. This work aims to bring methods and tools allowing the use of the techniques of the virtual immersion by the trades of the automotive design. A chain of conversion, simplification and optimization of the data according to the criteria of the trades of the automotive design is proposed for the creation of a digital mock-up dedicated to the virtual immersion. Management in real time of the data of this digital mock-up is carried out according to criteria of visualization of the scene in a context of virtual immersion.

Keywords: *Virtual reality, digital mock-up, 3D models treatment, CAD data conversion, automation, human visual system, optimization, level of details.*



Remerciements

Je tiens à remercier très sincèrement Marc NEVEU d'avoir accepté la direction de cette thèse et de m'avoir guidé tout au long de ce long travail.

Je tiens à exprimer ma plus profonde sympathie à Frédéric MERIENNE qui a encadré mes travaux. Je le remercie non seulement pour la qualité scientifique de son encadrement, mais aussi pour la disponibilité dont il a fait preuve à mon égard. Que ces quelques lignes soient au-delà d'un remerciement le début d'une amitié.

Je suis extrêmement reconnaissant envers tous les membres de mon jury pour l'attention qu'ils ont bien voulu porter à ce travail.

Jean-Claude LEON pour m'avoir fait l'honneur de présider ce jury.

Sabine COQUILLART et le professeur Christophe CHAILLOU pour avoir accepté la charge de rapporteur.

Stéphane THIVENT, responsable des systèmes de Réalité Virtuelle au sein du groupe PSA Peugeot-Citroën, pour sa disponibilité et ses conseils avisés.

Ces travaux s'inscrivent dans le cadre du projet PERF-RV, je tiens à remercier Jean LORISSON et Jean-Paul FRACHET pour leur temps passé à mettre en place cette thèse.

Les résultats de cette thèse sont étroitement liés aux développements réalisés avec le logiciel Simpoly[®], que Lionel FINE trouve ici toute ma reconnaissance pour ses nombreuses heures passées au téléphone à tenter de m'expliquer le fonctionnement de Simpoly[®].

Je tiens à saluer la disponibilité de Cécile BALLAZ pour l'écoute qu'elle a su m'offrir me permettant d'appréhender le domaine de la psycho physique ainsi que pour ses nombreuses relectures du chapitre relatif à la vision humaine.

Que Samuel BOIVIN trouve ici l'expression de mes sincères remerciements pour l'aide qu'il m'a apportée dans le développement de mes travaux.

Ces remerciements ne pourraient se terminer sans que j'exprime mes plus profonds remerciements à Magali, mes parents et tous ceux qui m'ont supporté et soutenu durant ces trois années.

Enfin, je tiens à remercier le Conseil Régional de Bourgogne pour sa participation au montage financier de cette thèse.



Table des matières

Table des matières.....	7
Table des figures	13
Introduction.....	17
Chapitre I Problématique technologique.....	23
I.1 - Métiers de la conception.	24
I.1.1 - Différentes natures de modèles surfaciques.	24
I.1.2 - Création d'un véhicule.....	26
I.1.3 - Méthodes de travail.	27
I.1.4 - Méthodes et outils : les besoins.	28
I.2 - Réalité Virtuelle.....	29
I.2.1 - Définition.....	29
I.2.2 - Origines.....	31
I.3 - Réalité Virtuelle et métiers de la conception.	31
I.4 - Traitement des données : des modèles surfaciques au rendu final.	34

I.4.1 - Méthodes existantes.....	34
I.4.2 - Du modèle surfacique au modèle RV.....	36
I.4.3 - Fonctionnement d'une carte graphique.....	38
I.5 - Démarche proposée.....	40
I.6 - Bilan.....	41
Chapitre II Préparation des données	45
II.1 - Fonctions à remplir par la préparation des données.....	46
II.2 - Méthodes existantes pour la préparation des données.....	48
II.2.1 - Triangulation.....	48
II.2.2 - Mise en cohérence.....	48
II.2.3 - Suppression des éléments non visibles.....	51
II.2.3.1 - Algorithme du peintre.....	52
II.2.3.2 - Elimination des parties arrières (<i>Backface culling</i>).....	53
II.2.3.3 - Arbre binaire de partitionnement de l'espace (<i>BSP-Tree</i>).....	54
II.2.3.4 - Tampon de profondeur (<i>Z-Buffer</i>).....	56
II.2.3.5 - Algorithme des <i>P-Buffers</i>	57
II.2.4 - Familles d'algorithmes de décimation.....	57
II.3 - Outils existants pour la préparation des données.....	61
II.3.1 - Simpoly [©]	61
II.3.1.1 - Mise en cohérence implémentée par Simpoly [©]	61
II.3.1.2 - Décimation implémentée par Simpoly [©]	62
II.3.2 - QSlim [Garland 97].....	65
II.3.3 - DeltaGen de Real Time Technology.....	66
II.3.4 - MeshDev [ROY 03].....	66
II.4 - Développement de nouvelles fonctionnalités avec Simpoly [©]	68
II.4.1 - Création et gestion des directions de visualisation.....	69
II.4.2 - Elimination des parties arrières (<i>Back face culling</i>).....	70
II.4.3 - Arbres de partitionnement de l'espace (<i>BSP</i>).....	71
II.4.4 - Détection des faces cachées (<i>P-Buffers</i>).....	71
II.4.4.1 - Implémentation.....	71

II.4.4.2 - Performances de l'algorithme.	75
II.4.4.3 - Fonctionnement sans l'interface graphique.....	79
II.4.4.4 - Tests et résultats : cas d'une portière.	79
II.4.4.5 - Tests et résultats : cas d'une planche de bord.	82
II.5 - Comparaisons des différentes solutions.	85
II.5.1 - Simply [©] vs QSlim.....	85
II.5.2 - Comparaison entre l'algorithme des <i>P-Buffers</i> et DeltaGen.....	89
II.6 - Conclusion.	93
Chapitre III Visualisation.....	95
III.1 - Enjeux de la visualisation.	96
III.2 - Optimisation et niveaux de détails.....	97
III.2.1 - Différents types de niveaux de détails.	97
III.2.2 - Gestion des niveaux de détails.....	98
III.3 - Fonctionnement du système visuel humain.	100
III.4 - Vision périphérique et applications informatiques.....	103
III.5 - Acuité visuelle et périphériques de visualisation.....	106
III.6 - Protocole de tests.	109
III.6.1 - Contexte.....	109
III.6.2 - Mise en œuvre.	110
III.7 - Résultats.....	115
III.8 - Mise en œuvre dans une application de Réalité Virtuelle.....	118
III.9 - Conclusion.	121
Chapitre IV Conclusions et perspectives	123
IV.1 - Analyse des objectifs.....	124
IV.1.1 - Prétraitement.	125
IV.1.2 - Visualisation.	126
IV.2 - Analyse critique.....	127
IV.2.1 - Prétraitement.	127
IV.2.2 - Visualisation.	128

IV.3 - Perspectives.....	129
Bibliographie	133
Annexes	141
Annexe 1 - Le Sensorama, Morton HEILIG 1962.....	142
Annexe 2 - Spécifications techniques : MoVE.....	143
Annexe 3 – Algorithme de détection des faces cachées.....	145
Annexe 4 - Suppression des faces cachées : DeltaGen vs Simpoly.....	146
Annexe 5 – Détail des sujets présents lors du test de vision périphérique.	147

Table des figures

Figure I-1 : Schéma Opérationnel de Développement (source PSA).....	27
Figure I-2 : Différentes natures de modèles.....	37
Figure I-3 : Différents paramètres de triangulation.....	38
Figure I-4 : Du modèle surfacique à la visualisation en temps réel.....	40
Figure II-1: Traitements à appliquer au modèle surfacique.	46
Figure II-2 : Problème de cohérence des modèles triangulés.....	49
Figure II-3 : Illustration de la phase de couture.....	50
Figure II-4 : Illustration des parties cachées.....	51
Figure II-5 : Algorithme du peintre mis en défaut par des faces sécantes.....	52
Figure II-6 : Calcul de la normale d'une facette.....	54
Figure II-7 : Scène initiale - 5 faces - 5 normales.....	55
Figure II-8 : Etapes de construction d'un <i>BSP-Tree</i>	55
Figure II-9 : Différents opérateurs de suppression et d'insertion d'entités.	58
Figure II-10 : Séquences d'arêtes confondantes.	62
Figure II-11 : Processus de suppression d'un sommet.	62
Figure II-12 : Types d'enveloppes utilisées pour le contrôle de la retriangulation.	64
Figure II-13 : Contraction de sommets (QSlm de GARLAND).....	65

Table des figures

Figure II-14 : Mesure de la déviation géométrique (MeshDev).	67
Figure II-15 : Illustrations de la déviation géométrique entre deux modèles.	68
Figure II-16 : Répartition de vecteurs.	69
Figure II-17 : Résultat de l'algorithme : orientation des normales.	70
Figure II-18 : Image calculée en remplaçant les couleurs par les index de faces.	73
Figure II-19 : Caméra à projection orthographique.	74
Figure II-20 : Caméra de type perspective.	74
Figure II-21 : Durée de traitement.	77
Figure II-22 : Pourcentage de polygones visibles conservés.	77
Figure II-23 : Durée de traitement en fonction de la résolution du <i>P-Buffer</i>	78
Figure II-24 : Pourcentage de conservation des faces visibles en fonction de la résolution.	78
Figure II-25 : Illustration de la suppression des parties cachées, cas de la portière.	80
Figure II-26 : Pourcentage de réduction modèle par modèle.	81
Figure II-27 : Identification des espaces libres.	82
Figure II-28 : Illustration de la suppression des parties cachées, cas de la planche de bord.	83
Figure II-29 : Pourcentage de réduction modèle par modèle.	83
Figure II-30 : Grille d'aération avant et après la suppression des faces cachées.	84
Figure II-31 : Durée de décimation de QSlim et de Simpoly [®]	86
Figure II-32 : Ecart maximum entre les modèles générés et le modèle de référence.	87
Figure II-33 : Ecart moyen entre les modèles générés et le modèle référence.	87
Figure II-34 : Variance entre les modèles générés et le modèle référence.	88
Figure II-35 : Répartition du nombre de faces par modèle (cas de la portière).	90
Figure II-36 : Illustration de l'impact des carreaux surfaciques.	92
Figure III-1 : Gestion des niveaux de détails par la distance.	98
Figure III-2 : Gestion des niveaux de détails avec un filtre d'hystérésis.	98
Figure III-3 : Choix des niveaux de détails piloté par la taille de projection.	99
Figure III-4 : Limites du champ visuel d'après [ARTHUR 00].	100
Figure III-5 : Anatomie de l'œil.	102
Figure III-6 : Répartition des cônes (cones) et des bâtonnets (rods), [OSTERBERG 35].	102
Figure III-7 : Variation de l'acuité visuelle en fonction de l'excentricité.	102

Figure III-8 : Illustration de la vision périphérique (source Hérault).	103
Figure III-9 : Champ visuel total / champ visuel sollicité par une application standard.....	104
Figure III-10 : Influence de la résolution sur l'acuité visuelle.....	107
Figure III-11 : Lunettes stéréoscopiques (Crystal Eyes de Stereographics).....	107
Figure III-12 : Influence des lunettes sur le champ visuel.	108
Figure III-13 : Champ visuel des lunettes à vision large.....	108
Figure III-14 : Système de maintien en position de la tête des sujets.....	111
Figure III-15 : Choix des positions angulaires pour le test.	112
Figure III-16 : Modèles utilisés pour le tests de vision périphérique.....	113
Figure III-17 : Répartition des modèles présentés en fonction des excentricités.	114
Figure III-18 : Couples de modèles proposés au sujet.....	114
Figure III-19 : Variations géométriques des modèles utilisés pour le test.	115
Figure III-20 : Pourcentage de bonnes réponses en fonction de l'excentricité.....	116
Figure III-21 : Ecart géométriques moyens calculés pour un taux de détection de 75%.	116
Figure III-22 : Erreur géométrique moyenne en fonction de l'excentricité.....	117
Figure III-23 : Représentation de l'excentricité pour le choix des LODs.....	118
Figure III-24 : Illustration des vecteurs de l'objet utilisés pour calculer l'excentricité.	119
Figure III-25 : Nombre de polygones par niveaux de détails.	119
Figure III-26 : Illustration de deux positions privilégiées.....	120
Figure IV-1 : Du modèle surfacique à la visualisation en temps réel.....	125
Figure IV-2 : Pilotage du prétraitement par l'application finale.	130

Introduction

L'industrie automobile utilise de manière systématique la *Conception Assistée par Ordinateur* (CAO) pour concevoir ses produits depuis les premières esquisses de style jusqu'à l'implantation des moyens de production dans les usines. Cette production 3D permet de réaliser des opérations de validation et de prise de décision sur des sous-ensembles complexes à l'aide de systèmes de réalité virtuelle plus ou moins évolués selon les besoins.

Les moyens graphiques sont de plus en plus performants, il en résulte une nouvelle problématique de restitution réaliste qui apparaît dans le monde industriel comme cela a pu être le cas il y a quelques années dans l'univers du jeu vidéo. Si l'objectif premier d'un affichage le plus réaliste possible avec une performance matérielle bornée est le même pour l'industrie du jeu vidéo et le secteur automobile en première approche, il en est tout autrement en ce qui concerne les moyens mis en œuvre pour arriver au résultat, ainsi que les attentes des utilisateurs vis-à-vis de ces images. Là où un joueur est avant tout intéressé par l'atmosphère de la scène dans laquelle il évolue, le styliste ou l'ingénieur se focalise sur l'exactitude des données affichées. Les procédés de production des scènes graphiques sont donc très différents. La conception d'un jeu vidéo se rapproche de la réalisation d'une superproduction cinématographique. La production va comprendre la mise en place d'éléments 3D créés de toutes pièces afin de coller à l'esprit du jeu.

L'industrie automobile souhaite utiliser ces procédés graphiques pour réduire ses circuits de décision; le temps de production des données nécessaires doit donc être réduit au maximum. A cela s'ajoute le fait que les équipes chargées de réaliser les scènes 3D pour validation ne sont pas des spécialistes dédiés à chaque tâche de la production 3D. Il en résulte des processus de production qui sont différents dans leurs objectifs et leurs moyens.

Dans le domaine industriel, les ingénieurs utilisent depuis longtemps l'outil informatique pour concevoir les nouveaux produits. La validation des étapes de conception se fait dans un premier temps numériquement (par la visualisation du produit). La validation définitive se fait par la réalisation d'un prototype physique. La visualisation lors du processus de conception s'effectue généralement sur un écran d'ordinateur. Cette visualisation est appropriée pour un ensemble de petite dimension et avec peu d'éléments. Dans le cas d'assemblages complexes et de grandes dimensions, il faut mettre en œuvre des moyens supplémentaires et adaptés pour être en mesure de proposer une visualisation de qualité. Cette nouvelle étape du processus de conception n'a pas pour vocation de supprimer le prototype physique. Elle permet de tester plus de configurations, effectuer plus d'itérations, plus d'anticipation dans le schéma de développement. Cette liberté a une incidence directe sur la créativité et l'innovation. Cette liberté est d'autant plus intéressante que le coût d'un prototype virtuel est plus faible que celui d'un prototype physique.

La technologie de visualisation et les logiciels actuels proposent des visualisations stéréoscopiques et, en temps réel, d'assemblages aussi complexes que des automobiles. L'utilisation de la stéréoscopie permet de s'approcher du réalisme des prototypes physiques dans des délais plus courts et des coûts plus faibles. Cette solution est donc complémentaire au prototype physique. Elle l'est d'autant plus si l'obtention de l'expérimentation (une expérimentation est l'association d'un logiciel et de son contenu) se fait de manière simple et rapide. Or, il se trouve que la préparation et la génération d'une expérimentation représentent beaucoup de travail et que peu d'outils sont mis à disposition pour optimiser ces travaux.

L'objectif de nos travaux est de proposer une méthode et des outils permettant de visualiser des modèles CAO (appelés aussi modèles surfaciques) en temps réel dans une expérimentation

stéréoscopique fonctionnant en temps réel. Ces travaux ont été développés en partenariat avec le groupe PSA Peugeot-Citroën.

Le premier chapitre décrit les méthodes et les outils industriels, il donne aussi une définition de la Réalité Virtuelle. Les techniques, les logiciels et les matériels qui sont employés pour obtenir la visualisation finale sont issus du domaine de la Réalité Virtuelle. Après avoir défini les deux domaines, les apports de la Réalité Virtuelle au monde industriel sont présentés. Puis, le mode de fonctionnement et les principes de base de la visualisation sont détaillés et mis en évidence par l'analyse d'outils logiciels commercialisés. Une méthode permettant de relier l'expérimentation aux modèles surfaciques est proposée. Cette méthode est décomposée en deux phases : la première phase est une phase de préparation des données, la seconde phase prend en charge la visualisation en temps réel.

Le chapitre 2 est consacré à la préparation des données. Les modèles sont visualisés dans un environnement différent de celui qui a servi à leur conception. Il faut les convertir pour les adapter au logiciel final. Ces modèles convertis devront être optimisés pour respecter les contraintes de visualisation en temps réel. Le chapitre fait état des différentes fonctions à remplir pour obtenir des modèles correctement préparés. Les solutions existantes sont recensées et comparées. Un algorithme a été développé et implémenté pour optimiser les modèles en vue d'applications de visualisation. Le chapitre se clôt par des tests et mesures de performances de la solution ainsi qu'un comparatif avec une autre solution logicielle commerciale. Les modèles simplifiés et optimisés seront intégrés dans des logiciels de visualisation. Ces logiciels possèdent leur propre gestion du temps réel. Ils font appel à des techniques leur permettant de maintenir un nombre minimum d'image par seconde. Ces techniques visent principalement à dégrader davantage les modèles.

Le chapitre 3 traite de la visualisation et de sa gestion en temps réel. Les enjeux de cette gestion sont présentés dans un premier temps. Les expérimentations s'adressent aux sens et plus précisément à la vue. Les caractéristiques physiologiques du système visuel humain sont étudiées et analysées pour comprendre son fonctionnement afin d'adapter les traitements à la perception humaine. Des tests de psychologie cognitive ont été réalisés dans un dispositif de visualisation de

type salle immersive. Suite à ces tests, une nouvelle gestion des modèles virtuels permettant d'accroître les performances globales de l'expérimentation est introduite.

Le chapitre 4 est une synthèse de la méthode globale définie et des outils développés et mis en œuvre. Les objectifs ainsi que les solutions apportées sont repris, analysés et critiqués. Ce chapitre se termine par les perspectives et la continuité possible de ces travaux.

Chapitre I

Problématique technologique

Ce chapitre a pour objectif de décrire le contexte dans lequel s'inscrit la problématique des travaux présentés dans ce document ainsi que les différentes techniques employées pour répondre au problème posé. Il s'agit de proposer une visualisation en temps réel et de qualité des modèles numériques issus de l'industrie manufacturière et particulièrement de l'industrie automobile. La compréhension du processus de conception d'un nouveau véhicule est nécessaire pour la mise en place d'une solution ayant des liens forts avec ce processus. Ce chapitre débute par une présentation du cycle de conception d'un nouveau véhicule. Les métiers de la conception sont abordés par leur mode de fonctionnement et par leurs éventuels besoins en terme d'optimisation. C'est dans cette phase du processus industriel, la conception, que le groupe PSA Peugeot-Citroën souhaite intégrer la Réalité Virtuelle. C'est pourquoi le chapitre se poursuit par les définitions existantes de la Réalité Virtuelle, de ses origines et de ses différentes applications. Nous verrons ensuite comment la Réalité Virtuelle peut s'intégrer aux métiers de la conception et plus particulièrement à celui du design et du style. En s'appuyant sur un état de l'art des différentes méthodes et outils actuels pour visualiser les modèles surfaciques en temps réel, une

méthode et des outils sont proposés permettant de créer à partir de modèles numériques une application de réalité virtuelle optimisée pour la visualisation de ces modèles en temps réel. Les exemples qui serviront à illustrer nos propos proviennent du groupe PSA Peugeot-Citroën.

I.1 - Métiers de la conception.

I.1.1 - Différentes natures de modèles surfaciques.

L'outil informatique et plus particulièrement les logiciels de CAO font partie du quotidien des acteurs des métiers de la conception. Il n'existe pas a priori de définition standard pour les modèles surfaciques issus des logiciels de CAO. Cependant, il est possible de définir les constituants formant ces modèles :

- *les éléments ponctuels*, éléments de représentation les plus simples de dimension 0 qui n'interviennent pas directement dans la définition géométrique du modèle. Mais ils servent à définir les courbes ou surfaces plus complexes (extrémités de courbes, coins de surfaces, ...),

- *les éléments linéiques*, éléments de dimension 1, utilisés dans la définition de courbes ou segments. Ils interviennent dans la représentation des modèles filaires, premier type de représentation de modèles 3D,

- *les éléments surfaciques*, éléments de dimension 2, introduits pour des représentations de formes lisses (capot, aile de voiture, portière, ...). Deux niveaux de description peuvent être distingués :
 - les descriptions de surfaces (plans, cylindres, sphères, ...) par des fonctions implicites :

$$f(x, y, z) = 0$$

- les descriptions paramétriques [LEON 91], [FARIN 00] : ce type de représentations est basé sur l'utilisation de formes à pôles telles que les carreaux de Bézier [BEZIER 87], B-Splines, NURBS (Non Uniform Rational B-Splines). La définition paramétrique de la surface S d'un objet 3D s'exprime à l'aide d'une équation S dépendant de deux paramètres u et v variant souvent dans l'intervalle $[0, 1]$:

$$S = \{(x, y, z) \in \mathbb{R}^3 / \exists (u, v) \in [0, 1]^2 \text{ avec } S(u, v) = (x, y, z)\}$$

Les modèles décrits de cette manière ne possèdent pas suffisamment d'informations pour définir une géométrie réelle d'un objet 3D. Ils ne renseignent pas l'intérieur ou l'extérieur de la pièce, ils ne définissent que la surface limite.

- *les solides*, éléments de dimension 3. La notion de topologie est intégrée, définissant les notions d'intérieur et d'extérieur qui se caractérisent par les conditions :
 - de fermeture,
 - d'orientation des surfaces bornant le domaine volumique,
 - de non auto intersection des surfaces.

Les approches les plus répandues sont :

- la modélisation CSG (*Constructive Solid Geometry*) [REQUICHA 80], qui se base sur des opérations élémentaires (union, soustraction, intersection) de formes géométriques simples (cylindres, sphères, extrusions, ...).
- la modélisation B-Rep (*Boundary representation*), qui définit des modèles à partir de frontière. Cette notion lui permet d'assigner un *intérieur* et un

extérieur au modèle. Ces frontières divisent l'espace en deux ensembles : l'un fermé (le solide), l'autre infini (le reste de l'espace). La représentation B-Rep décrit les frontières du solide comme un assemblage de faces orientées formant une surface fermée. Chaque face est à son tour limitée par des contours bornés par des sommets ou arêtes. Cette notion d'orientation est importante car elle permet de déterminer le lieu de la matière. Les modèles polyédriques sont une représentation de la modélisation des B-Rep. Ils utilisent une liste de faces (généralement des triangles), constituées d'arêtes pour définir le modèle. Ce type de représentation est couramment utilisé pour les applications de réalité virtuelle, la diffusion multimédia des scènes 3D, ... Ces modèles sont peu utilisés dans le monde de la conception car ils ne sont qu'une approximation des surfaces réelles.

I.1.2 - Création d'un véhicule.

La création d'un nouveau véhicule par le groupe Peugeot-Citroën répond au Schéma Opérationnel de Développement (SOD). Ce schéma représente la logique d'ensemble et les principales étapes de la conception, du développement et de l'industrialisation. Il est le cadre de référence pour les projets véhicule. Un projet véhicule est composé de :

1. **l'avance de phase programmée** : phase de construction globale du programme et de dimensionnement du véhicule.
2. **la phase de définition préliminaire** : phase de définition du style et des composants du véhicule.
3. **la phase de développement** : phase de conception détaillée, de tests et de mise en production du véhicule.

Ces trois phases sont illustrées dans la Figure I-1.

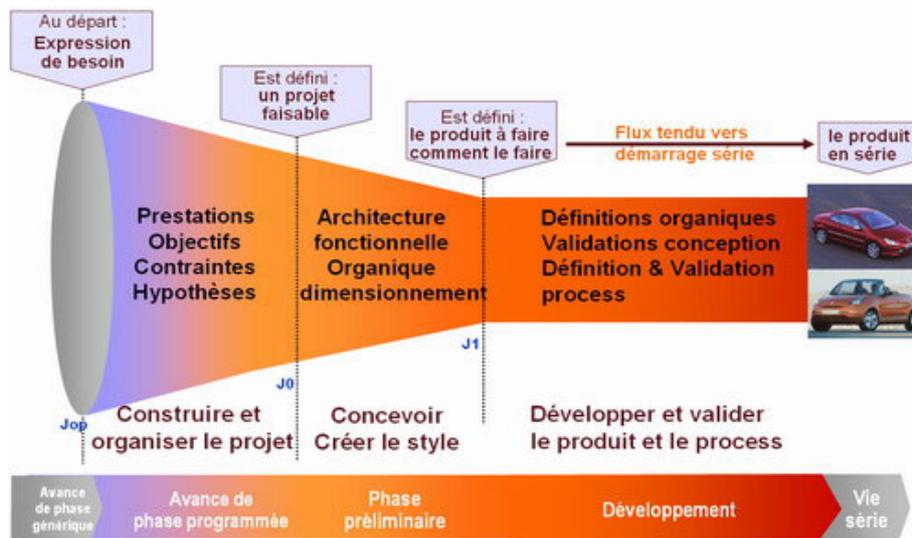


Figure I-1 : Schéma Opérationnel de Développement (source PSA).

I.1.3 - Méthodes de travail.

La création d'un véhicule résulte de la contribution de plusieurs corps de métiers (styliste, carrossier, architecte, mécanicien, électricien, électronicien, ergonomes, fournisseur, ...). Chacun de ces métiers possède ses habitudes, ses règles, ses méthodes de travail, ses logiciels, son vocabulaire. Le développement d'un nouveau produit se fait au travers d'une organisation de type plateau. Ces plateaux permettent de regrouper les différents corps de métiers concernés par le produit. Le produit est alors conçu simultanément avec les contraintes de chacun.

Les concepteurs créent et dessinent les modèles numériques du véhicule via leur logiciel orientés métiers. Ces modèles sont des modèles surfaciques. Ils définissent de manière mathématique les courbures, formes et topologie des modèles. Ils servent à la réalisation des modèles physiques utilisés pour l'assemblage du futur du véhicule. Leur évolution au cours du développement du véhicule est constante. Des mises à jours quotidiennes peuvent avoir lieu. Les revues de projet décisionnelles doivent s'appuyer sur les dernières versions de ces modèles pour être pertinentes.

I.1.4 - Méthodes et outils : les besoins.

Il est nécessaire de trouver des environnements permettant la mutualisation des différents métiers intervenant dans le processus de conception du produit. Les logiciels utilisés par chaque métier doivent permettre l'export du travail réalisé vers des formats d'échange. Cependant, ces passerelles d'export ne sont jamais complètement compatibles et de nombreuses informations sont perdues lors des conversions vers les formats d'échange. Certains logiciels tels que CATIA V5 de Dassault System ou ProE de PTC offrent la possibilité à différents corps de métiers de travailler sous la même plate-forme logicielle. Ces logiciels doivent être par ailleurs très ouverts aux autres logiciels de CAO car la conception d'un véhicule fait intervenir des entreprises extérieures (sous traitant) possédant leur propre CAO. Il est nécessaire de faire cohabiter toutes ces sources de modèles surfaciques.

En dehors de ces problèmes de format et d'échange de données, les logiciels doivent permettre de capitaliser l'information et le savoir faire. En effet, un nouveau produit est avant tout une réutilisation de l'existant, une voiture ressemble beaucoup à la précédente. Il faut pouvoir mémoriser le savoir-faire, les modèles, les actions menées lors de la conception. Cette capitalisation, si elle est réutilisée à bon escient, permet d'être plus performant et plus compétitif. Pour une nouvelle voiture, si certaines parties sont forcément nouvelles (style, habillage intérieur, évolution technologique majeure, ...) d'autres sont réutilisables. Il peut s'agir soit des pièces ou éléments physiquement présents dans le véhicule (rétroviseur, climatisation, blocs moteurs, ...) soit il peut s'agir de méthodes de fabrication, d'implantation d'atelier, de cadencement, Le produit suit une évolution depuis son concept, son développement, sa réalisation et sa fin de vie. Cette évolution est appelée cycle de vie du produit (*Product Life Cycle Management : PLM*). Le PLM permet de gérer les différentes configurations d'un même véhicule (options, direction à gauche ou à droite, ...) mais aussi de connaître la maturité du projet (validation de jalon). A ce stade, la visualisation d'éléments constituant le véhicule participe à l'avancement du produit dans son cycle de vie. Des expérimentations permettent de valider des jalons, d'effectuer des choix sur telle ou telle configuration, ...

Pour des expérimentations de visualisation, il faut pouvoir supporter les différents formats natifs et pouvoir interagir avec les modèles et les assemblages. Ces techniques de visualisation et d'interaction sont étroitement liées à celles de la Réalité Virtuelle. La Réalité Virtuelle a entre autre la vocation de proposer un environnement de visualisation, d'interaction et de temps réel. Elle apporte cependant un nouvel environnement de travail avec ses contraintes. Elle permet l'intégration des utilisateurs dans les applications de réalité virtuelle, un large éventail d'interactions possibles et une liberté de créativité accrue. En revanche, il faut être en mesure de proposer des applications dont la qualité est suffisante pour que les revues de projet puissent être décisionnelles. En fait il faut quantifier la précision des modèles visualisés et calculer le rendu le plus réaliste possible. Dans ces travaux, nous nous adressons aux métiers du style. Les attentes de ces métiers vis à vis d'une application de réalité virtuelle sont très strictes et exigeantes. La précision des modèles présentés doit être maîtrisée. Sans contrôle et connaissance de cette précision, aucune décision ne saurait être prise. Pour finir, la mise en situation des modèles doit être la plus fidèle possible. C'est à dire que la fonctionnalité doit pouvoir être éprouvée. Par exemple dans le cas d'un ouvrant, il faut donner la possibilité de manipuler cet assemblage.

L'ensemble de ces contraintes pour réaliser une application de visualisation requiert des connaissances particulières aux niveaux matériels informatiques, logiciels ou des différents métiers mis en œuvre. La création et la manipulation de telles applications semblent définir un nouveau métier dans l'entreprise au même titre qu'il existe des stylistes, ingénieurs de conception, ergonomes, ...

I.2 - Réalité Virtuelle.

I.2.1 - Définition.

Le nom de **Réalité Virtuelle** aurait été créé selon de nombreux auteurs par Jaron LANIER en 1989. Des approches syntaxiques ont tenté de définir la réalité virtuelle, les auteurs de l'article "*La réalité virtuelle*" [Computer Science 93] la présentent ainsi : "... *Le paradoxe contenu dans le terme de*

"réalité virtuelle" explique le galvaudage de celui de "virtualité". "Virtuel" signifie "qui a force de réalité". Une réalité virtuelle est donc une "fausse réalité vraie", ...".

Au delà des considérations sémantiques, le sénateur Claude HURIET dans son rapport d'information [HURIET 97-98] associe la notion de virtuel à la notion d'acteur. C'est en effet, ce qui apparaît de façon récurrente dans la littérature : on ne peut pas parler de réalité virtuelle sans parler d'immersion. C'est ce que Claude Cadoz dans son livre *"Les réalités virtuelles"* [CADOZ 94] résume par *"être et ne pas être"*. Il faut avoir l'impression d'être dans un monde recréé mais sans l'être physiquement. Afin d'arriver à *être et ne pas être*, Claude Cadoz parle de **conviction de la réalité**. Mais cette immersion n'est pas autosuffisante. En effet, Morton HEILIG, dont le *Sensorama* (cf. Annexe 1 - Le Sensorama, Morton HEILIG 1962.) est considéré comme la première application de réalité virtuelle, révèle lui-même que *pour être vraiment de la réalité virtuelle*, il manque la notion d'**interactivité**. Cette notion d'interactivité est souvent associée au terme de temps réel. L'interactivité ne sera possible, confortable et pertinente que si les actions de l'utilisateur sont répercutées dans les plus brefs délais au niveau de l'application ; l'idéal étant de pouvoir effectuer les mises à jour de l'application avant que l'utilisateur grâce à ses sens ne puisse déceler ce délai.

A ce titre, Didier Verna (ENST) [Priméca 01] ne parle pas de temps réel. Il évoque les **composantes cognitives** et les notions de **"prise cognitive"** sur des objets. En l'absence de ces "prises cognitives", une maquette aussi précise soit elle ne touche pas l'imaginaire de l'utilisateur.

En accord avec la définition que donnent Grigore Burdéa et Philippe Coiffet dans leur livre intitulé : *"La Réalité Virtuelle"* [BURDEA 93] la réalité virtuelle serait la combinaison de : l'immersion, l'interaction et l'intégration.

Cette approche de la perception est par ailleurs abordée dans le rapport de Fréjus et Drouin pour EDF en 1997 [FREJUS 97] *"Réalité virtuelle et processus cognitifs"*. Ils constatent que "les recherches actuelles ont, en effet, abordé l'homme sous l'angle de la perception et non de la cognition, c'est à dire de l'appropriation des informations perçues."

I.2.2 - Origines.

Les origines de la réalité virtuelle sont difficilement définissables. En effet, des applications que l'on qualifierait de réalité virtuelle existaient avant que le terme de réalité virtuelle ne soit inventé.

La première application reconnue de tous est le Sensorama (cf. Annexe 1 - Le Sensorama, Morton HEILIG 1962.). Ce dispositif permet à un utilisateur d'être immergé visuellement et auditivement dans une scène réelle, préalablement filmée. L'utilisateur perçoit également des odeurs et des vibrations. Reposant sur des techniques non informatiques, le Sensorama est resté à l'état de prototype. Selon Morton HEILIG, le Sensorama offrait bien de l'immersion mais pas d'interactivité.

Howard RHEINGOLD dans son livre "*La réalité virtuelle, quand l'illusion a toutes les apparences de la réalité*" [RHEINGOLD 93] nous présente ces pionniers. On y retrouve Morton Heilig au côté de Frederick Brooks de l'Université de Caroline du Nord ou de Ivan Sutherland. Frederick Brooks est à l'origine d'un programme de recherche visant à créer un dispositif combinant la visualisation et le retour d'effort. Ivan Sutherland du Massachusetts Institute of Technology, est concepteur d'un dispositif d'affichage stéréoscopique monté sur tête avec capteurs de position. Ces recherches datant toutes des années 60 nous montrent que les bases de la réalité virtuelle existent déjà depuis plusieurs décennies.

Depuis cette époque beaucoup d'applications et de recherches ont été menées, certaines seront abordées ultérieurement avec la présentation du matériel utilisé pendant ces travaux de recherche.

I.3 - Réalité Virtuelle et métiers de la conception.

La Réalité Virtuelle semble ainsi être en mesure d'apporter l'environnement de visualisation immersif, temps réel et interactif requis pour les revues de projets des métiers de la conception. A ce titre, les éditeurs de logiciels offrent dans leurs solutions standard un modèle de réalité virtuelle adapté à leur logiciel. Le choix s'effectue en fonction des compatibilités de format de fichiers, des

possibilités et facilités de génération d'applications interactives (création de comportement sur des objets, fluidité, ...). En revanche, l'utilisation de la réalité virtuelle impose une préparation des données. Cette préparation est soit intégrée dans le logiciel natif soit effectuée par un autre logiciel. Dans le premier cas, les solutions offertes ne sont pas satisfaisantes, pour des raisons de qualité visuelle et d'interactivité avec la scène qui seront expliquées d'avantage ultérieurement. Dans le second cas la préparation est souvent lourde, longue et fastidieuse mais permet d'avoir un meilleur contrôle sur la qualité de la prestation finale. A titre d'exemple, une application de revue de projet du véhicule complet 807/C8 du groupe PSA Peugeot-Citroën a nécessité 4 mois de travail pour traiter les quelques 400 modèles visibles du véhicule. Il est à noter que ce travail serait à refaire entièrement en cas d'évolution des modèles constituant le véhicule. Aucune capitalisation des actions menées pour préparer les modèles n'a été mise en place. Les premières expérimentations développées étaient des outils de communication sur des véhicules finis et en production.

Afin que le lecteur puisse se repérer facilement, quelques termes qui permettront de comprendre les différences entre modèles surfaciques (modèles CAO par exemple) et modèles utilisés pour la Réalité Virtuelle (appelés **modèle RV** dans la suite du document) sont définis. Ces définitions sont partagées par une grande majorité des acteurs industriels du domaine. Le premier terme défini est celui de maquette numérique. La **maquette numérique** est constituée de modèles surfaciques issus de la CAO ou des métiers du style et design. Ce sera la référence pour la qualité des modèles. Il est possible de parler de référence puisque les modèles sont définis de manière exacte car les surfaces sont générées par des équations mathématiques. Or, ce type de modèle n'est absolument pas compatible avec les logiciels de réalité virtuelle, qui pour fonctionner en temps réel, ne peuvent pas gérer ce type de représentation. Les modèles doivent être convertis. Ces modèles une fois adaptés forment avec un outil de visualisation dédié à la Réalité Virtuelle, la **maquette virtuelle**. Il est important de noter que la maquette virtuelle est optimisée pour une application donnée. Les modèles ne sont pas préparés de la même manière s'ils servent à une application de visualisation, à une application de montage/démontage ou à une simulation de calcul (acoustique vibratoire, aérodynamique, ...). De plus, la maquette virtuelle est souvent diffusée sur des périphériques de visualisation spécialisés (salle immersive, *workbench*, écran échelle 1, casques, ...). Le passage d'une maquette à l'autre est actuellement le point

bloquant pour utiliser la Réalité Virtuelle au niveau des métiers de la conception. La lourdeur des tâches à effectuer pour préparer les modèles impose des temps de livraison des applications trop longs pour respecter les délais de validation des étapes de conception. Tant que la génération de la maquette virtuelle ne sera pas en phase avec le cycle de conception, il sera difficile de donner à la Réalité Virtuelle le statut d'outil de conception.

L'usage des outils de Réalité Virtuelle est à positionner dans le cadre des processus de développement du groupe PSA Peugeot-Citroën. Trois modes d'utilisation principaux ont été identifiés :

1. en tout numérique : pour faire le tri entre plusieurs hypothèses, pour les classer, éliminer les moins prometteuses et décider sur lesquelles concentrer les efforts : en phase d'avant-projet, très tôt dans le processus, en remplacement de prototypes physiques qui n'existent pas toujours à cette étape.
2. en anticipation du prototype physique : pour détecter des anomalies avant le lancement de sa réalisation et fiabiliser celle-ci.
3. en accompagnement du prototype physique : de nombreux jalons de développement imposent la présence de prototypes physiques pour juger d'encombrement, d'impression d'ensemble. Les modèles virtuels permettent à moindre frais de présenter les alternatives, les variantes, les dérivés, de mettre les véhicules dans des scènes de vie virtuelles. Physique et virtuel se complètent alors.

La totalité des outils et méthodologies accompagnant ces modes d'utilisations n'est actuellement pas disponible. Le principal verrou technologique est la simplification des chaînes de traitement pour convertir les modèles surfaciques en modèles RV. Des solutions sont cependant mises en place pour répondre aux besoins des utilisateurs. Elles sont complexes et font appel à de multiples logiciels ; ce qui rend leur utilisation difficile.

Cette recherche d'outils et de méthodes par le groupe PSA Peugeot-Citroën s'inscrit dans un projet d'investissement d'une salle immersive. Les équipes de réalité virtuelle du groupe ont pu tester pendant un an la salle immersive de l'ENSAM appelée MOVE™ (cf. Annexe 2). Cette salle

immersive place l'utilisateur au cœur de la scène 3D. Son champ visuel peut être entièrement sollicité. Ainsi, il est envisageable de placer un styliste ou designer au volant d'un véhicule virtuel de lui montrer les modèles qu'il vient de concevoir pour qu'il puisse juger de leur pertinence, forme, ...

I.4 - Traitement des données : des modèles surfaciques au rendu final.

I.4.1 - Méthodes existantes.

Le passage de la base de données des modèles surfaciques à une application de réalité virtuelle est un travail long et fastidieux. La littérature propose plusieurs chaînes de traitements afin de faciliter le travail des utilisateurs.

[GRAF 02] présente une chaîne de conversion comprenant les phases suivantes:

1. Récupération des données,
2. Conversion des données (triangulation),
3. Réduction des données (décimation),
4. Mise en cohérence (gap, orientation des normales, ...).

Les modèles inclus dans la scène sont regroupés dans un arbre (ou graphe de scène). Cet arbre contient les modèles mais il contient aussi les matrices de position et orientation de chacun des modèles ou groupe de modèles. Ces matrices sont indispensables pour une représentation correcte de la scène. La base de cette chaîne de conversion s'appuie sur l'idée qu'il faut dupliquer l'arborescence utilisée par la maquette numérique vers l'application de réalité virtuelle. Les modèles virtuels ainsi créés sont correctement mis en position par rapport à leur référence surfacique. Dans le cas où cet arbre n'est pas exporté, il faut s'assurer du positionnement et de l'orientation corrects des modèles.

Par ailleurs, l'auteur insiste sur la semi automatisation de cette chaîne. L'automatisation n'étant pas possible sans connaissance du rôle des modèles dans la scène. Trop de critères entrent en jeu pour permettre l'automatisation complète, l'utilisation finale de la maquette, le comportement des utilisateurs, Il préconise donc l'implémentation d'un '*protocole manager*' qui permet de sauvegarder les actions de l'utilisateur et de pouvoir les répéter si nécessaire. Cependant, il ne fait pas état de la manière dont sont obtenus les modèles RV : la conversion des modèles surfaciques en modèles RV n'est pas traitée. Nous verrons ultérieurement la complexité et les problèmes générés par cette conversion.

[BRUNETTI 00] ajoute à cette chaîne une étape de suppression d'éléments inutiles. Cette étape permet d'alléger la scène finale. Elle est intéressante dans notre cas, car l'application développée est liée à la visualisation, seules les parties a priori visibles des modèles sont à considérer. Les modèles surfaciques sont conçus pour permettre la réalisation des modèles physiques. Ils représentent aussi bien la partie visible que les éléments structurels ou de fixation. Il est possible de supprimer toutes les parties mécaniques (raidisseurs, nervures, ...) et les éléments de fixation (passage de vis, alésage fonctionnel, ...). Brunetti fait aussi état de l'importance d'un mode automatique (sans l'intervention d'un opérateur et sans l'utilisation de l'interface graphique du logiciel de traitement) pour répliquer les séquences de la chaîne. Une dimension supplémentaire intégrée par Brunetti est la prise en compte des matériaux et textures pour les modèles RV afin de proposer aux utilisateurs des modèles virtuels proches de ce qu'aurait été un prototype physique fabriqué avec les matériaux finaux.

[WHYTE 00] aborde le problème de la conversion des modèles surfaciques en modèles RV en définissant trois approches différentes. La première consiste à créer une librairie d'objets de base optimisés et réutilisable à chaque nouvelle application. Cette approche a été mise en place pour des projets architecturaux dans lesquels un certain nombre d'éléments peuvent être réutilisés, mais elle est peu adaptée au domaine automobile. La seconde s'apparente à celles présentées par Graf et Brunetti, et enfin la troisième centralise les données surfaciques ainsi que des informations sur la façon dont sont conçus et utilisés les modèles. Cette approche permet de capitaliser le savoir-faire de préparation des modèles. Whyte s'intéresse aussi au format d'échange des modèles. La multiplicité des logiciels impose de trouver un format d'échange commun. Il

existe beaucoup de formats transversaux. Les principaux formats cités par Whyte sont : le format DXF développé par Autodesk, le format IGES (*Initial Graphics Exchange Format*) et le format STEP (*Standard for Exchange of Product Data*). D'autres formats peuvent être cités, la liste est longue et il est souvent nécessaire d'avoir recours à des formats différents avant de parvenir au modèle final.

Au-delà de ces méthodes développées dans des laboratoires de recherche, les éditeurs de logiciels proposent des solutions intégrées, telles que 4D Navigator pour CATIA V4, DMU pour CATIA V5 ou DVMockUp pour le format Pro/Eng. Ces solutions ne sont pas satisfaisantes d'un point de vue qualité visuelle et / ou interactivité. Même si les logiciels proposent des outils d'optimisation et de gestion des données pour des applications en temps réel, les contrôles et les ouvertures pour le développement sont faibles. A titre d'exemple le module DMU pour CATIA V5 ne s'inscrit dans aucune des méthodes précédemment citées. L'optimisation des modèles (i.e. la décimation) est réalisée de manière arbitraire. Pour maintenir un taux de rafraîchissement satisfaisant, le logiciel supprime des triangles au hasard sur les modèles présents à l'écran. L'impact immédiat est l'apparition de discontinuités (de trous) sur les modèles visualisés. Ce type de solution n'est pas acceptable pour les revues de styles. C'est pourquoi il est nécessaire d'avoir recours à des outils logiciels spécifiques aux applications de réalité virtuelle et à des chaînes de préparation des données.

I.4.2 - Du modèle surfacique au modèle RV.

Les modèles surfaciques sont constitués d'éléments définis par des équations mathématiques (cf. Figure I-2(a)) ou formes polynomiales. Les modèles RV quant à eux sont des modèles dits polyédriques car les applications de réalité virtuelle ne savent gérer en temps réel que des polygones (généralement des triangles¹, cf. Figure I-2(b)). La conversion d'un modèle surfacique en un modèle polyédrique s'appelle la triangulation. Elle consiste en la discrétisation des surfaces définissant les modèles CAO. La discrétisation de modèles CAO n'est pas une nouveauté créée

¹ Un triangle est la forme la plus simple possible des polygones. Utiliser des formes plus évoluées (quadrilatères, ...) pour définir les modèles génère des problèmes de cohérence et de conservation de primitive lors des phases de traitement. Ces problèmes seront abordés ultérieurement.

pour les applications de visualisation. Les modèles polyédriques sont présents dans le cycle de conception de produit à d'autres niveaux. Par exemple, les machines à commandes numériques pour l'usinage de pièce calculent leur trajectoire grâce à des modèles polyédriques. Des formats de modèles et d'échanges existent donc déjà.

Cette opération de discrétisation est pilotée par un ou plusieurs paramètres. Le nombre et le choix des paramètres dépendent du résultat escompté. La Figure I-3 illustre les trois paramètres principaux. Le premier est l'écart maximal acceptable entre la courbe de référence et les éléments discrets générés (dans notre cas les triangles), appelé la SAG. Le second est la taille maximum des éléments de base, appelé la STEP. Cette valeur permet d'avoir un contrôle sur la taille des arêtes des triangles générés afin d'éviter d'avoir des triangles trop effilés. Les triangles effilés sont facilement décelables lors du rendu en temps réel. Ils apportent souvent des discontinuités de rendu. Le troisième, α , est l'angle entre deux faces partageant une même arête. Cette valeur permet de respecter la courbure du modèle de référence. Plus ces valeurs seront faibles plus le modèle polyédrique sera proche géométriquement du modèle de référence mais le nombre de triangles sera plus important. D'autres paramètres peuvent être pris en compte comme la surface des triangles, les angles aux sommets, la nature du triangle (isocèle, équilatéral), ... Plus le nombre de paramètres est important plus la génération est compliquée et coûteuse en temps de calcul. Généralement, les paramètres utilisés sont la SAG et la STEP. Les problèmes liés à l'organisation et au nombre de polygones seront abordés ultérieurement.

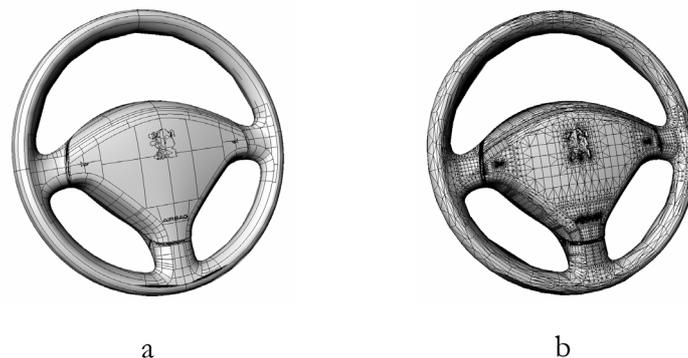


Figure I-2 : Différentes natures de modèles.
(a) modèle surfacique / (b) modèle polyédrique.

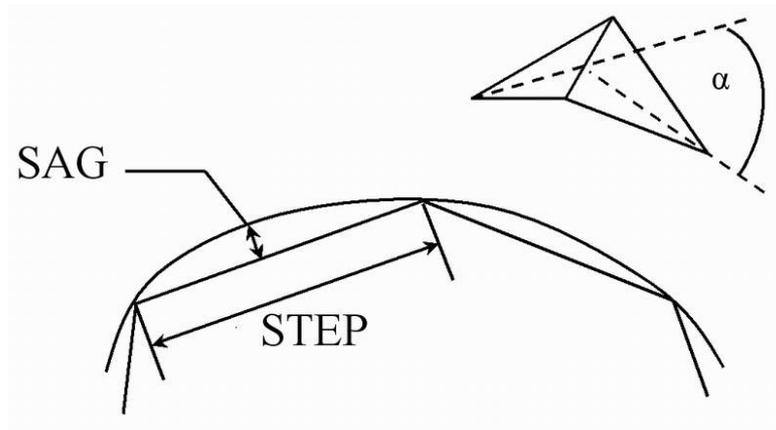


Figure I-3 : Différents paramètres de triangulation.

Ces modèles polyédriques sont ensuite gérés en temps réel par les applications de réalité virtuelle. Chacun des triangles composant les modèles polyédriques est envoyé à la carte graphique. La carte graphique prend en charge l'ensemble de ces triangles et génère les images qui seront affichées sur le périphérique de visualisation.

I.4.3 - Fonctionnement d'une carte graphique.

La carte graphique de l'ordinateur reçoit les triangles à afficher. Des algorithmes sont implémentés sur ces cartes pour traiter les ordres et informations émis par le logiciel sollicitant la carte. Les fonctionnalités des cartes actuelles sont en augmentation. Les cartes sont de plus en plus performantes. Les implémentations finales et matérielles varient d'une carte à l'autre et d'un fabricant à l'autre. Cependant, le principe de base permettant de transformer un polygone d'une scène en 3 dimensions en une séquence de pixels est quasiment le même. Pour illustrer ce principe, nous présentons celui de la machine OpenGL [WOO 04]. Nous avons choisi la machine OpenGL car la bibliothèque de développement OpenGL est un élément de base très employée dans le domaine de l'informatique graphique.

Il est possible de distinguer 9 étapes dans le processus d'affichage d'un polygone :

1. **Transformation matricielle** : passage du repère de l'objet au repère global,
2. **Calcul lumineux** : calcul de la couleur à chaque sommet,
3. **Transformation matricielle** : passage du repère de l'objet au repère normalisé de la caméra,
4. **Transformation matricielle** : passage du repère normalisé de la caméra aux coordonnées de la fenêtre de visualisation,
5. **Pixellisation** : passage des primitives (triangles) à un ensemble de pixels, chaque pixel est affecté d'une couleur,
6. **Plaquage de texture** : calcul d'une nouvelle couleur en fonction de la couleur actuelle et celle de la texture,
7. **Opération de mélange des pixels** : mélange des couleurs en fonction de la profondeur, du brouillard et de la transparence éventuels,
8. **Faces cachées (Z-Buffer)** : test de profondeur, il sert à connaître les faces visibles,
9. **Affichage** : envoi de l'image calculée au périphérique de visualisation

Le processus global peut être décomposé en deux parties. La première qui regroupe les étapes 1 à 4 est appelée *geometry engine*. Cette partie prend en charge l'ensemble des transformations géométriques appliquées à la scène. La deuxième qui regroupe les étapes 5 à 9 est appelée *raster manager*. Cette partie gère les éléments d'un point de vue pixel. Lorsqu'une application ne fonctionne pas à une cadence suffisante, il faut identifier laquelle des parties constitue le goulot d'étranglement en terme de temps de calculs.

Le processus détaillé ci avant montre que tous les polygones de la scène subissent les étapes 1, 2, 3, 4, 5, 6, 7 et 8 qu'ils soient cachés, partiellement cachés ou visibles. Le nombre de polygones contenus dans une scène est un élément important pour la fluidité finale de l'application. L'optimisation et le choix des polygones à afficher sont des contraintes majeures pour la réalisation d'une application de réalité virtuelle. Les cartes graphiques ont des performances en constante augmentation. Cependant elles possèdent toujours une limite. Elles sont de plus en plus sollicitées pour des rendus particuliers (effet de peinture métallisée, réflexions, ...), les temps de calculs pour l'affichage s'en trouvent forcément affectés. Il est difficile de connaître

exactement la limite du nombre de polygones pour une scène. Cette limite dépend du matériel (processeur, mémoire, carte graphique) mais aussi du contenu et des demandes inhérentes à la scène (type de rendu, interaction, ...).

Tout ceci montre l'importance de maîtriser et de contrôler le nombre de polygones d'une scène.

I.5 - Démarche proposée.

La conversion des modèles surfaciques en modèles polyédriques manipulables en temps réel est incontournable car les cartes graphiques ne savent actuellement que des polygones. Ces opérations sont très coûteuses en temps de calculs, donc inconcevables pour des scènes volumineuses. Par ailleurs, l'application de réalité virtuelle n'a pas que cette tâche à effectuer. Il faut aussi gérer la scène, les interactions éventuelles, les calculs de rendu, ... Il est nécessaire d'extraire la phase de préparation des données de l'application finale. La Figure I-4 présente une vue d'ensemble de la démarche proposée pour permettre de visualiser un modèle surfacique dans une application de réalité virtuelle.

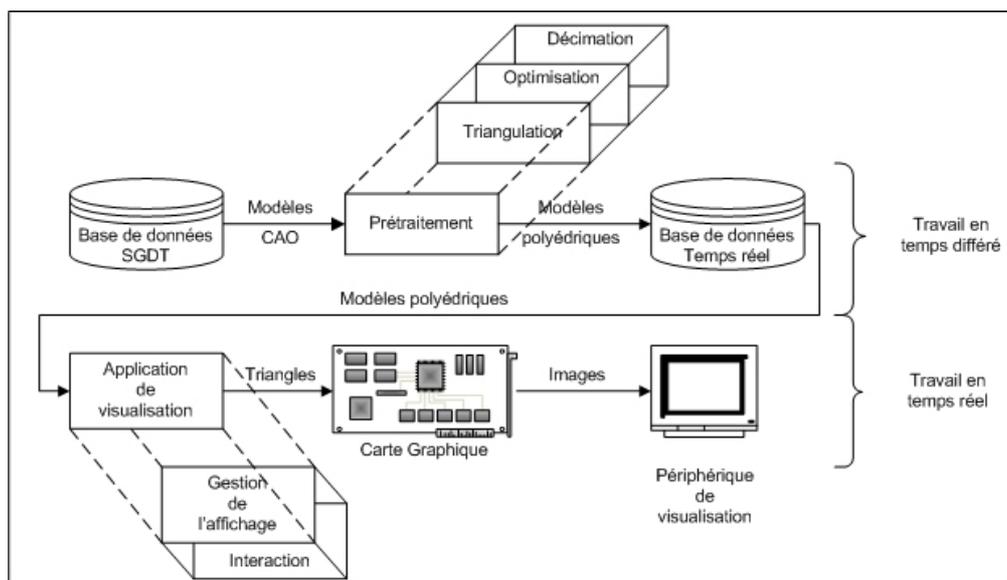


Figure I-4 : Du modèle surfacique à la visualisation en temps réel.

La Figure I-4 fait apparaître une base de données dans laquelle sont stockés les éléments utilisés par l'application temps réel. Cette base de données est alimentée en amont par une base de données contenant les modèles surfaciques ayant subi un prétraitement. Celui-ci souvent long est réalisé en temps différé. Il est une succession d'étapes : triangulation, optimisation et décimation qui sont présentées dans les chapitres suivants. C'est à ce niveau que la qualité des modèles est contrôlée, le nombre de polygones créés par modèle est une conséquence du prétraitement.

Il est à noter que la notion de qualité géométrique des modèles va de pair avec l'augmentation du nombre de polygones. Pour contrôler ce nombre de polygones, nous avons deux leviers à notre disposition. Le premier se situe au niveau du prétraitement puisque c'est à ce stade que sont créés les polygones. Le second se situe au niveau de l'application de visualisation où il est possible de définir une gestion adéquate de la scène et des modèles (i.e. des polygones) pour optimiser le nombre de polygones envoyés à la carte graphique.

I.6 - Bilan.

La Réalité Virtuelle peut permettre aux métiers de la conception d'être plus performants, plus innovants. Cet apport pose cependant un certain nombre de problèmes. Il faut pouvoir alimenter les applications de réalité virtuelle avec des modèles issus des métiers de la conception. Des méthodes de conversions sont connues et peuvent être mises en place. Elles sont souvent longues et fastidieuses. Les modèles issus de la conception évoluent quasiment quotidiennement. Ces évolutions doivent être répercutées à la même cadence dans les applications finales, ce qui est impossible avec le rythme d'évolution des modèles. Les éditeurs de logiciel de CAO ont compris l'enjeu de la Réalité Virtuelle. Leurs logiciels sont de plus en plus ouverts vers des solutions intégrant des fonctionnalités de réalité virtuelle. Ces solutions ne sont pas satisfaisantes car la diversité des métiers concernés est telle que les logiciels ne peuvent pas gérer l'ensemble des contraintes des différents métiers. Or, il faut pouvoir prendre en compte ces contraintes pour définir l'application finale.

La Réalité Virtuelle impose l'emploi de modèles polyédriques pour ses applications. Les métiers de la conception mettent à disposition des modèles surfaciques. Il faut définir et mettre en place une passerelle permettant la conversion et l'optimisation des modèles issus de la conception. Le chapitre 2 traite de la phase de préparation des modèles. L'ensemble des modèles issus de cette préparation représente un nombre de polygones. Il est possible que ce nombre soit supérieur à celui admissible par les calculateurs des applications temps réel. Il est nécessaire de prévoir une gestion appropriée de ces modèles et de leurs triangles pour garantir la fluidité de l'application. Le chapitre 3 est consacré à la gestion des données géométriques pendant la visualisation.

Chapitre II

Préparation des données

Le chapitre précédent montre l'importance d'une étape de préparation des données. Les objectifs sont de mettre à disposition de l'application de réalité virtuelle des modèles polyédriques optimisés. Ces modèles doivent remplir plusieurs conditions. La première est le respect de la géométrie et de la topologie des modèles de référence. La seconde est la qualité du rendu en temps réel. La qualité des images calculées par le moteur de rendu est dépendante de la qualité des modèles polyédriques. L'application doit pouvoir assurer un nombre minimum d'images par seconde pour permettre l'interaction avec la scène. Cette condition est, nous le verrons ultérieurement, très liée au nombre de polygones des modèles. Finalement, la dernière condition est de capitaliser les actions menées sur les modèles en préparation afin d'automatiser au maximum les traitements. Cette automatisation permet de réduire les temps de préparation des modèles et d'être en phase avec le cycle de vie du produit.

Ce chapitre débute par un inventaire des différentes actions à mener sur les modèles polyédriques qui sont : mise en cohérence du polyèdre, suppression des triangles non visibles, décimation, capitalisation. La deuxième partie du chapitre est un inventaire des différentes méthodes disponibles pour répondre aux besoins de la préparation de données. La troisième partie est consacrée aux outils existants. La quatrième partie illustre les développements réalisés lors de ces travaux. Ces travaux sont comparés dans la cinquième partie aux outils existants. Le chapitre se clôt par un bilan sur la mise en œuvre de la chaîne de traitement des modèles.

II.1 - Fonctions à remplir par la préparation des données.

La préparation des données ne se limite pas à la triangulation des modèles surfaciques. Nous avons vu qu'il est nécessaire d'optimiser ces modèles et de garantir une qualité de rendu minimum pour la pertinence de l'application finale. La Figure II-1 illustre la succession des étapes nécessaires à la conversion d'un modèle CAO en un modèle polyédrique optimisé pour une application de visualisation en temps réel.

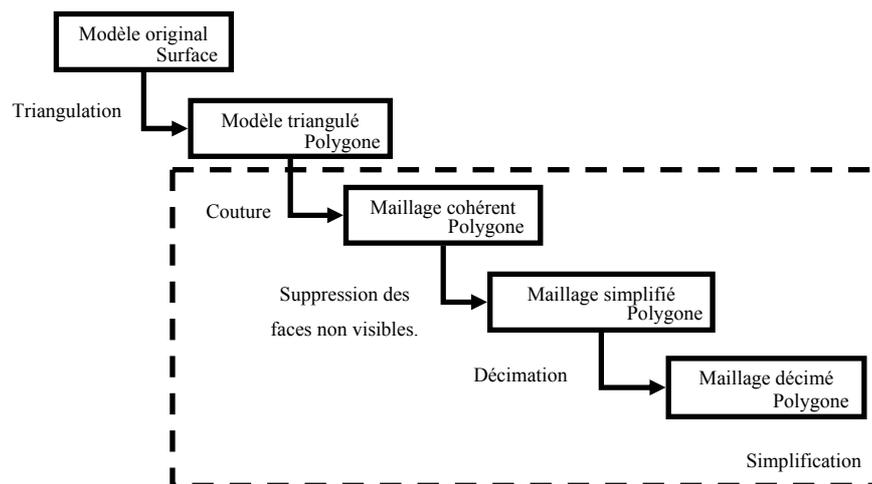


Figure II-1: Traitements à appliquer au modèle surfacique.

1. Triangulation :

La triangulation est l'étape de conversion des modèles surfaciques en modèles polygonaux. Le critère pilotant cette étape est la précision souhaitée par l'utilisateur, plus la précision sera grande plus le nombre de polygones sera important.

2. Couture :

La couture permet de mettre en cohérence les modèles (respect de la géométrie et de la topologie des modèles). Cette étape garantit en partie la qualité du rendu final des modèles.

3. Suppression des faces non visibles :

C'est une optimisation dépendante des conditions d'utilisation, les faces qui ne sont jamais visibles peuvent être supprimées. Le critère pilotant cette étape est directement lié à l'application finale, donc aux métiers concernés.

4. Décimation :

C'est une seconde optimisation. Il s'agit de réduire le nombre de polygones par dégradation du modèle (perte de précision). Le critère pilotant cette étape est soit un écart géométrique maximum, soit un nombre maximum de polygones pour le modèle dégradé.

La littérature et les éditeurs de logiciels proposent des solutions pour une, voire plusieurs des étapes précédentes. Le nombre de logiciels nécessaires à la préparation des données doit être limité afin de réduire le risque de perdre des données lors des conversions de fichiers.

La suite du chapitre présente les différentes étapes précitées, ainsi que la méthode choisie pour mettre en œuvre cette chaîne de traitement.

II.2 - Méthodes existantes pour la préparation des données.

II.2.1 - Triangulation.

Cette étape convertit une surface mathématique en maillage polyédrique. Plusieurs techniques existent basées sur la triangulation de Delaunay ([BOISSONNAT 84], [BOISSONNAT 92], [EDELSBRUNNER 92]) ou sur le graphe de Voronoï ([ATTALI 94], [AMENTA 98]).

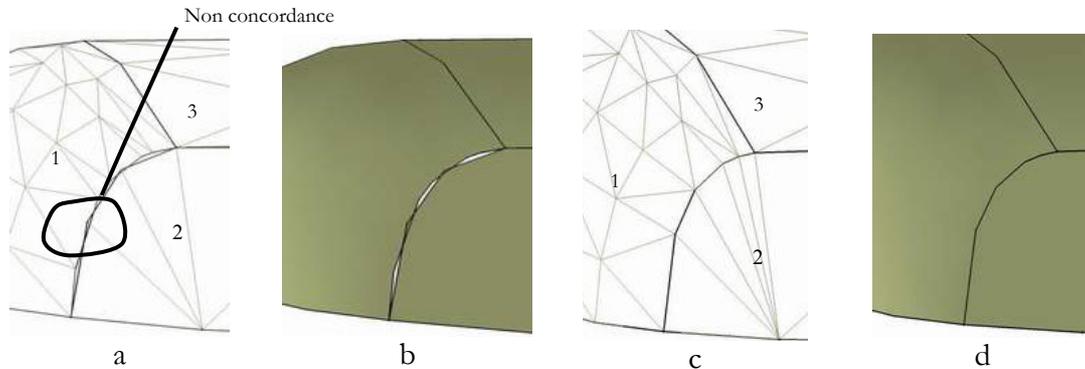
Le pilotage des algorithmes de triangulation se fait par différents critères. Les principaux critères sont la distance maximum tolérable entre les polygones générés et la surface de référence (SAG) ou encore la longueur des arêtes des polygones (STEP). A ces critères viennent s'ajouter d'autres valeurs autorisant le contrôle de la qualité du maillage final. Par exemple, il est possible de contraindre la nature des triangles (isocèle, équilatérale) ou l'angle maximum entre deux faces ayant une arête commune. Ces contraintes confèrent au maillage une allure et des courbes plus fidèles au modèle de référence. Cependant, l'augmentation du nombre des contraintes se fait au détriment du nombre de triangles générés.

La triangulation est appliquée à chaque carreau surfacique composant le modèle initial. La cohérence existante entre les frontières des carreaux surfaciques est perdue lors de la triangulation. Cette perte de cohérence engendre des discontinuités sur le polyèdre. Une étape corrective, appelée couture, est alors à mettre en place pour palier à ce problème.

II.2.2 - Mise en cohérence.

La construction des modèles surfaciques permet de connaître l'intérieur et l'extérieur de la pièce. Ces modèles sont alors dits fermés. Cette notion est importante pour le processus de conception puisqu'elle permet de définir le volume, la masse, l'inertie des modèles... Ces paramètres sont incontournables pour le résultat final qu'est le véhicule. L'équivalent de cette notion pour les modèles polyédriques est la cohérence. Un modèle polyédrique est dit cohérent s'il est possible de différencier l'intérieur de l'extérieur du modèle. Cette cohérence est souvent perdue lors de la triangulation. Les modèles surfaciques sont construits par la juxtaposition de plusieurs carreaux surfaciques. Lors de la triangulation, chaque carreau est traité de manière

indépendante. Cette approche implique que les frontières qui étaient communes dans le modèle surfacique se retrouvent avec des séquences de sommets différentes dans le modèle polyédrique.



1, 2 et 3 carreaux surfaciques provenant du modèle CAO.

Figure II-2 : Problème de cohérence des modèles triangulés.

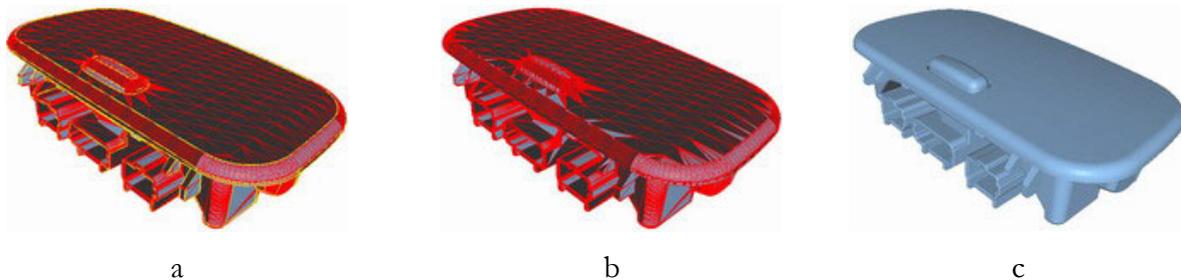
(a) non cohérence en mode filaire, (b) non cohérence du rendu,

(c) cohérence en mode filaire, (d) cohérence du rendu

Ces problèmes ont été abordés par Butlin [BUTLIN 96]. Il dresse un état des problèmes rencontrés avec les modèles CAO et les algorithmes de triangulation. Des méthodes pour assainir ces modèles sont proposées. Les principales actions sont la suppression des entités surnuméraires (sommets, arête, face) et la fusion des entités séparées d'une distance définie au préalable.

La Figure II-2 (a) montre la jonction entre 3 carreaux surfaciques qui ont été triangulés. Le modèle serait cohérent si les sommets appartenant à la frontière d'un carreau coïncidaient avec les sommets de la frontière commune avec un autre carreau. Cette non concordance a pour effet direct de provoquer des discontinuités lors du calcul du rendu (cf. Figure II-2(b)). Ces discontinuités sont en fait des trous entre des groupements de triangles. La présence de ces trous n'est pas acceptable pour une application de revue de projet. La solution pour réparer les modèles est de souder les points entre eux. La Figure II-2(c) illustre la réparation (ou couture) du modèle. Cette couture consiste à rapprocher ou fusionner les points les plus proches. Cette étape est pilotée par un critère représentant la distance maximum séparant deux points éventuellement fusionnables. Parallèlement, cette opération est couplée à des algorithmes de suppression de sommets, arêtes et faces doubles. Ces entités doubles apparaissent lors de la triangulation.

L'élimination de ces entités doubles est primordiale pour le succès de la phase de mise en cohérence. La Figure II-3 représente les différentes étapes de la mise en cohérence.



**Figure II-3 : Illustration de la phase de couture.
Appliquée à un plafonnier (source PSA).**

La Figure II-3(a) montre un modèle issu de la triangulation. Les lignes jaunes de la Figure II-3(a) représentent les bords libres du modèle qu'il est nécessaire de supprimer. Un bord libre est l'ensemble des arêtes d'une même surface n'appartenant qu'à un seul triangle. Un modèle fermé ne possède pas de bord libre. Le modèle de la Figure II-3(a) s'il était décomposé en sous modèles cohérents (suivant les bords libres) générerait 50 modèles. La Figure II-3(b) est le résultat obtenu après la phase de couture et de suppression des entités doubles. Le modèle résultant ne présente plus de bords libres. Cependant, certaines faces ne sont pas correctement orientées. Le modèle étant désormais fermé, il est possible de détecter l'intérieur de l'extérieur donc d'orienter les faces vers l'extérieur. La Figure II-3(c) est le résultat final, le rendu du modèle ne fait pas apparaître de discontinuités et/ou de face mal orientée.

Actuellement, cette phase est réalisée manuellement. C'est un utilisateur qui scrute le modèle à la recherche des entités doubles pour les supprimer et qui *recond* le modèle. Cette méthode ne permet pas de capitaliser les actions. Lors de l'évolution du modèle pendant le cycle de développement, il sera nécessaire de recommencer le travail depuis le début. Ces répétitions discréditent fortement l'utilité de la réalité virtuelle dans le cycle de conception d'un véhicule.

II.2.3 - Suppression des éléments non visibles.

A ce niveau de la chaîne de prétraitement, les modèles polygonaux sont fermés et cohérents. L'objectif à présent est de réduire le nombre de polygones. Rappelons que nous devons proposer aux utilisateurs des modèles pour une revue de projet. Les modèles seront vus et assemblés tels qu'ils le seront lors de leur mise en place finale dans le véhicule. Seul l'extérieur des assemblages est visible et utile pour notre revue de projet. Les modèles CAO contiennent toutes les informations nécessaires à leur réalisation : raidisseurs, patte de fixation, passage de vis ... L'aspect extérieur, mais aussi ses parties cachées qui ont un rôle fonctionnel sont représentés. Tous ces éléments requièrent des polygones pour être dessinés et comme ce nombre est limité, il faut pouvoir supprimer un maximum de polygones qui ne seront jamais vus.

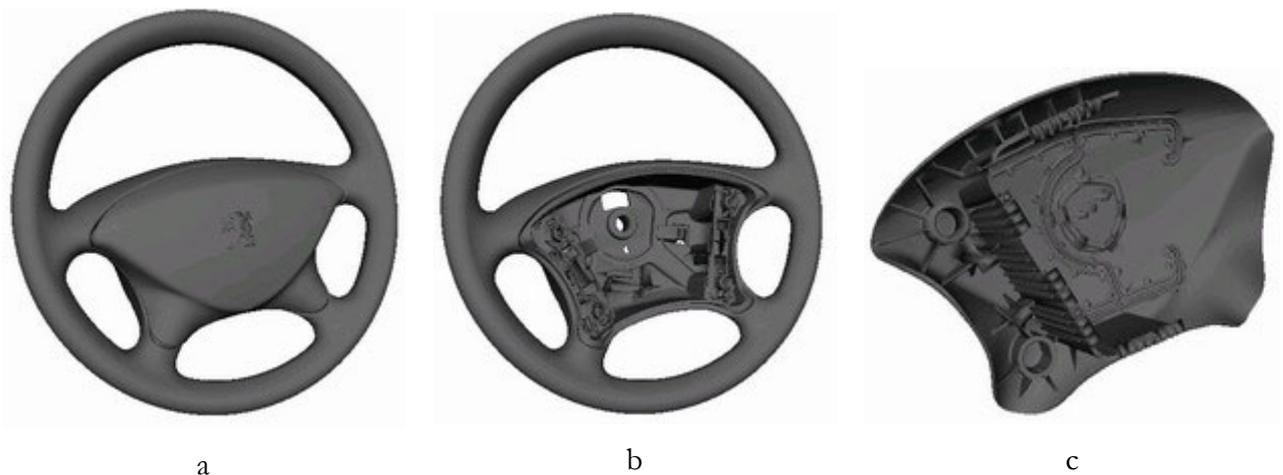


Figure II-4 : Illustration des parties cachées.

Appliquée au volant et au support de l'air bag (source PSA).

La Figure II-4(a) montre un assemblage de deux modèles : le volant et le support de l'air bag. Ces deux modèles sont représentés sur les Figure II-4(b) et (c). Lors d'une revue de projet d'intérieur de véhicule ces modèles ne sont jamais vus démontés. Les polygones servant à définir l'arrière du support et l'intérieur du volant ne sont pas utiles et peuvent être supprimés.

Pour cela, il faut pouvoir :

- identifier ces polygones,
- supprimer ces polygones,
- automatiser ce traitement.

Ce problème se rapproche du problème fondamental en image de synthèse appelé indifféremment élimination des parties cachées ou détermination des surfaces visibles. Les premiers algorithmes datent des années 60. Les principaux algorithmes fonctionnant au niveau de la scène ou des modèles (pyramide de visualisation, obstruction d'objets par d'autres, ...) ont été abordés. L'objectif n'est pas de faire une revue des familles d'algorithmes permettant la suppression des parties cachées mais de s'intéresser à ceux travaillant au niveau des polygones.

II.2.3.1 - Algorithme du peintre.

Le nom de l'algorithme vient de la similitude de la technique employée avec celle d'un peintre. Un peintre commence par couvrir sur sa toile les éléments de l'arrière plan et remonte jusqu'au premier plan. L'algorithme trace les polygones dans l'ordre, grâce à un tri sur la profondeur de leur isobarycentre, du plus éloigné au plus proche. Cet algorithme très rapide fonctionne très bien avec les objets convexes mais est plus long avec les objets concaves et est mis en défaut avec des faces sécantes.

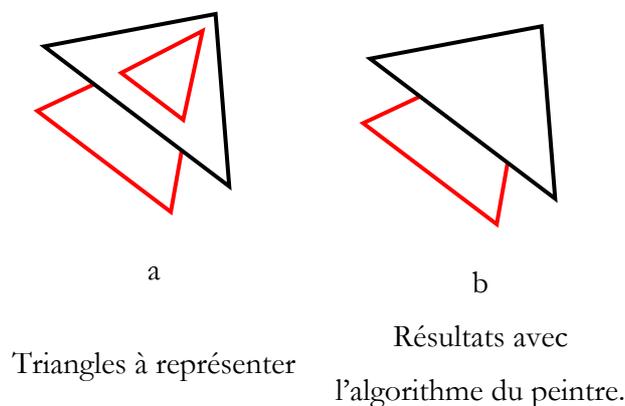


Figure II-5 : Algorithme du peintre mis en défaut par des faces sécantes.

II.2.3.2 - Elimination des parties arrières (*Backface culling*).

Cet algorithme permet de retirer les faces non visibles par l'observateur (indépendamment des autres faces). Ce traitement se fonde sur une observation simple : si la normale d'une face \vec{n} et la direction de vue \vec{v} de l'observateur sont connues, il est possible de déterminer si cette face sera vue. En effet, toute face dont la normale pointe dans une direction opposée à la direction de vue sera occultée par l'objet lui-même. De manière plus mathématique, si la normale à une face fait un angle de moins de 90° avec la direction de vue alors la face est dissimulée. Pour détecter une telle configuration, il suffit de réaliser le produit scalaire entre \vec{n} et \vec{v} et de regarder son signe. Un produit scalaire négatif ou nul signifie que la face est visible, tandis qu'un produit scalaire positif indique qu'elle est occultée. Appliquer cet algorithme implique que l'orientation des faces soit cohérente (définition claire d'intérieur et d'extérieur) : la normale à une face doit pointer vers l'extérieur de l'objet. Elle peut alors être calculée à l'aide du produit vectoriel de deux arêtes consécutives de la face.

Les notations sont celles de la Figure II-6.

Le vecteur normal est obtenu par le produit vectoriel $\vec{N} = \overrightarrow{P_1P_2} \wedge \overrightarrow{P_1P_3}$. Le produit scalaire dont le signe va déterminer si la face est visible ou non est le produit :

$$\vec{V}_1 \cdot \vec{N} = \vec{V}_1 \cdot (\overrightarrow{P_1P_2} \wedge \overrightarrow{P_1P_3})$$

On peut aussi l'exprimer à l'aide des vecteurs \vec{V}_1 , \vec{V}_2 et \vec{V}_3 :

$$\vec{V}_1 \cdot \vec{N} = \vec{V}_1 \cdot ((\vec{V}_2 - \vec{V}_1) \wedge (\vec{V}_3 - \vec{V}_1))$$

D'où:

$$\vec{V}_1 \cdot \vec{N} = \vec{V}_1 \cdot (\vec{V}_2 \wedge \vec{V}_3)$$

La face sera visible si et seulement si le produit mixte précédent est négatif ou nul.

De manière générale, cet algorithme est rapide et il peut s'appliquer à tout objet convexe. Il est suffisant lorsque la scène n'est composée que d'un seul objet convexe. Dès que la scène contient plusieurs objets qui peuvent se cacher mutuellement, il ne suffit plus mais reste une étape préliminaire à des algorithmes plus évolués. En revanche, il est inefficace pour des objets concaves.

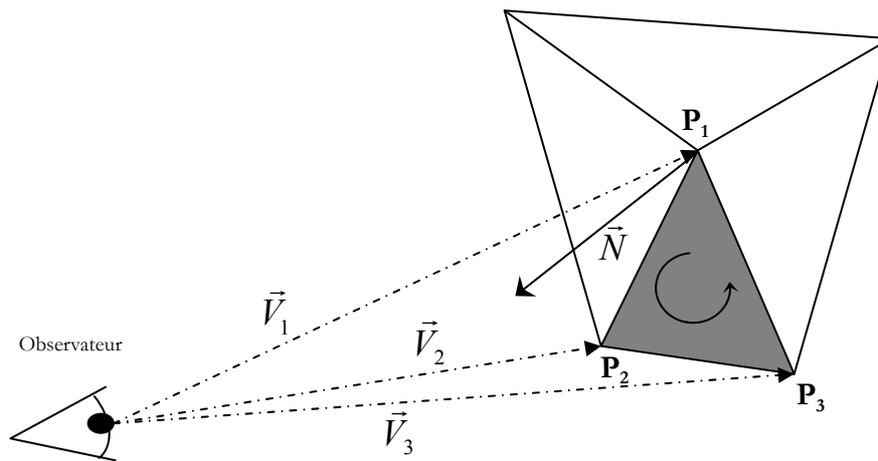


Figure II-6 : Calcul de la normale d'une facette.

II.2.3.3 - Arbre binaire de partitionnement de l'espace (*BSP-Tree*).

L'idée de base de cet algorithme consiste à partitionner l'espace en régions à l'aide de plans définis par les facettes polygonales de la scène ([NAYLOR], [SCHUMACKER 69], [SUTHERLAND 73], [FUCHS 80]). Cette partition détermine l'ordre de visibilité des facettes en fonction de la région dans laquelle se trouve l'observateur. Ainsi, pour une facette donnée, l'affichage prendra en compte d'abord les facettes situées dans le demi-espace ne contenant pas l'observateur, la facette traitée, puis les facettes situées dans le même demi-espace que l'observateur. En itérant ce processus pour les demi-espaces définis précédemment, il est possible de définir un ordre d'affichage pour toutes les facettes constituant la scène.

Pour cela un arbre binaire de partitionnement de l'espace (*Binary Space Partition Tree* : *BSP-Tree*) est défini. Il va représenter la subdivision récursive de l'espace en une hiérarchie de demi-espaces. Une face choisie "arbitrairement" sera la racine de l'arbre et une normale orientée lui est associée. Cette normale permettra de distinguer les deux demi-plans générés par cette face. Le demi-plan

dans lequel est dessinée la normale sera appelé arbitrairement "devant" de la face et "derrière" de la face l'autre demi-plan. L'ensemble des faces situées devant la face racine sera rangé dans le sous-arbre gauche tandis que l'ensemble des faces de derrière sera installé dans le sous-arbre droit. Si la droite générée par la face traverse une autre face, celle-ci est coupée en deux et chacune des demi-faces est rangée dans le sous-arbre correspondant. Dans chacun des demi-espaces, une face racine est choisie "arbitrairement" et le processus est réitéré. La Figure II-7 et la

Figure II-8 détaillent les étapes du processus.

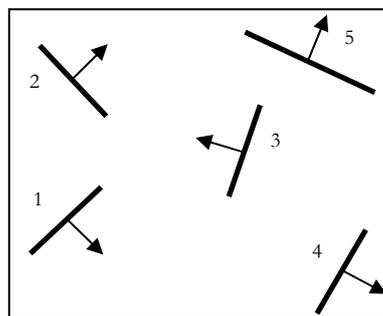


Figure II-7 : Scène initiale - 5 faces - 5 normales.

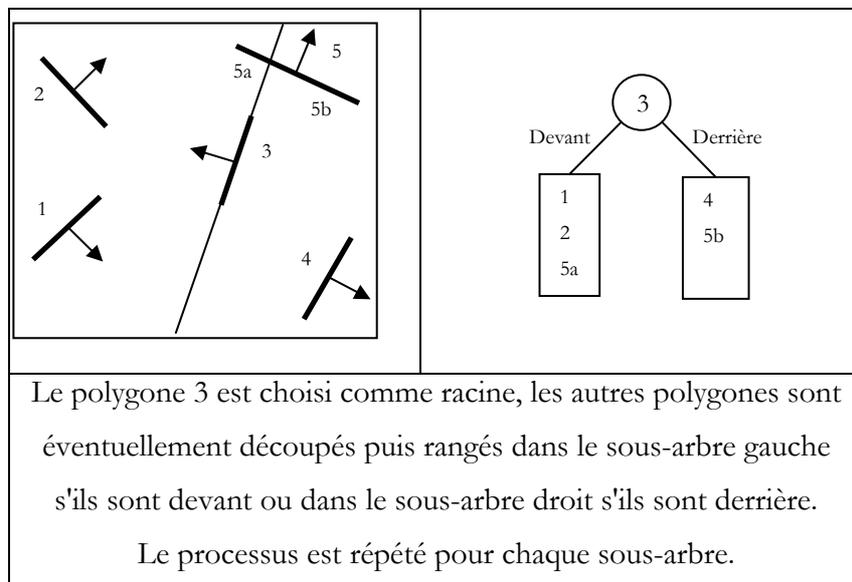


Figure II-8 : Etapes de construction d'un *BSP-Tree*.

Ces arbres sont très performants pour l'ordonnancement des polygones. Cependant, ils sont mal appropriés pour les scènes dynamiques (déplacement d'objets dans la scène) car les arbres sont construits une seule fois pour l'application. Les changements éventuels ne seront pas pris en compte. Néanmoins, des travaux d'amélioration des *BSP-Tree* existent en particulier des travaux sur des *BSP-Tree* dynamiques [TORRES 90]. D'autre part ces arbres renseignent sur la position des faces les unes par rapport aux autres (i.e. devant ou derrière) mais ne statuent pas sur la visibilité d'une face (i.e. une face située derrière une autre est-elle visible ou non).

II.2.3.4 - Tampon de profondeur (*Z-Buffer*).

L'algorithme du *Z-Buffer* ([CATMULL 74]) consiste à stocker pour chaque pixel de l'écran, non seulement sa couleur mais aussi sa distance par rapport à l'observateur dans la scène (*Z-Buffer*). Il procède de la manière suivante : les profondeurs de tous les pixels de l'image sont d'abord initialisées à la plus grande distance possible. Les éléments de la scène sont alors successivement traités dans un ordre quelconque. Pour traiter un élément (dans notre cas des polygones), on détermine les pixels de l'image sur lesquels il se projette. En chacun de ces pixels, la distance de l'élément par rapport à l'observateur est calculée. Si celle-ci est inférieure à la valeur actuelle de la profondeur du pixel, alors l'élément est plus proche de l'observateur en ce pixel que l'élément précédemment déterminé.

Malgré sa lourdeur et ses exigences en mémoire, cet algorithme est très fréquemment utilisé car il permet de traiter un grand nombre de configurations, y compris le cas des facettes qui s'interpénètrent. En outre, il a l'avantage de pouvoir être directement programmé au niveau de la carte graphique, ce qui rend son exécution plus rapide. Néanmoins quelques problèmes se posent. Le premier concerne des objets qui partagent une arête, autrement dit qui ont des faces adjacentes. L'arête prendra la couleur de l'objet traité en premier. Une solution consiste à utiliser le test de visibilité dans l'algorithme du *Z-Buffer*. De cette manière, les faces dont les normales sont dans la direction de l'observateur sont traitées en dernier. Une autre possibilité pourrait consister à réaliser un tri des objets selon l'algorithme du peintre puis d'appliquer le *Z-Buffer* sur les objets triés du plus loin au plus près. Cette technique permet aussi de gérer la transparence ce qui n'est pas le cas pour le *Z-Buffer* classique.

Bien que cet algorithme soit câblé, c'est-à-dire pris en charge par le *hardware* de la carte graphique, l'exploitation du résultat reste longue. De plus, le résultat de l'algorithme du *Z-Buffer* étant une image, il faut accepter d'avoir une fenêtre dédiée à cette image ouverte en permanence en parallèle de l'application de base. C'est la récupération de l'image en mémoire graphique qui est coûteuse en temps. Face à ce problème, les *P-Buffers* ont été développés.

II.2.3.5 - Algorithme des *P-Buffers*.

La technique des *P-Buffers* [SUNG 92] [WEGHORST 84] [BOIVIN 01] permet d'utiliser les accélérations matérielles des cartes graphiques tout en offrant la possibilité à l'utilisateur de récupérer le résultat au niveau de la mémoire et non plus dans une image envoyée sur un écran. Les temps d'accès sont considérablement réduits. Ainsi, chaque nouvelle amélioration algorithmique de détection des faces cachées implémentée dans les cartes graphiques profitera à cette méthode. Utiliser la technique des *P-Buffers* revient à utiliser les algorithmes de *Z-Buffer* sans vouloir afficher l'image calculée. Il est dans ce cas possible d'exploiter les notions de profondeur et de polygones visibles depuis une position de caméra (ou d'utilisateur).

II.2.4 - Familles d'algorithmes de décimation.

Les techniques de décimation n'ont pas été créées pour l'optimisation des modèles polyédriques destinés à la réalité virtuelle. Elles ont des origines aussi variées que : la modélisation d'un objet 3D à usiner (définition de la trajectoire), le contrôle dimensionnel (mesure d'écart entre un modèle de référence et un modèle polyédrique mesuré point par point), la gestion de bases de données diverses (médecine, jeux vidéo, scanner 3D,...), ...

Ces différents domaines utilisent les modèles polyédriques. Ils ont besoin d'outils permettant la manipulation de ces modèles. Les techniques développées autour des modèles polyédriques dépendent des contraintes et des critères métiers de chaque domaine. Cependant, il est possible de classer ces méthodes selon soit le principe général de fonctionnement des opérateurs soit les critères contrôlant le fonctionnement des opérateurs.

[VERON 97] et [PUPPO 97] proposent des études plus approfondies de ces méthodes.

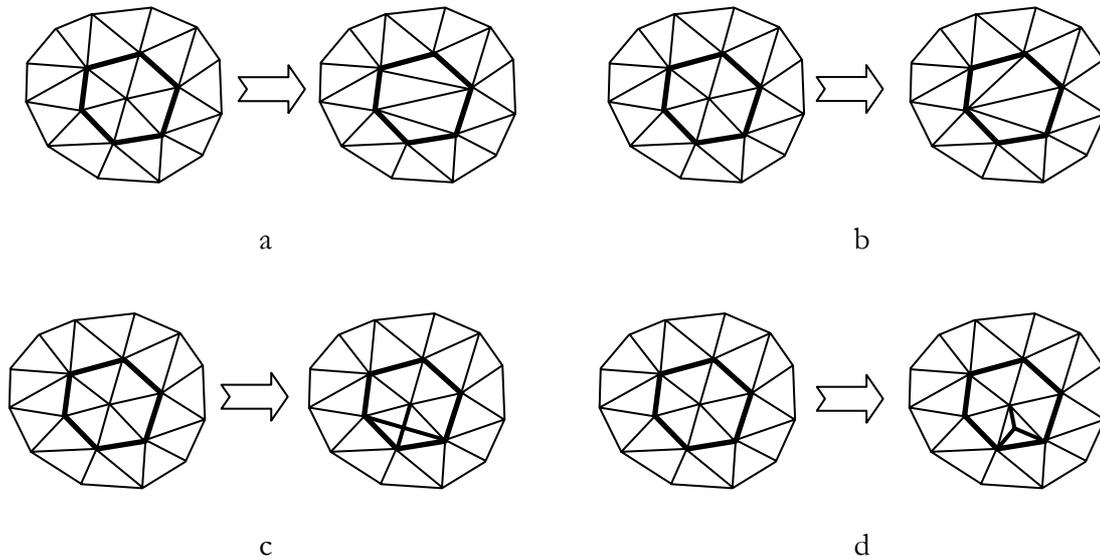


Figure II-9 : Différents opérateurs de suppression et d'insertion d'entités.

- a) opérateur de suppression de sommets, b) opérateur de fusion d'arêtes
c) opérateur de subdivision d'arêtes, d) opérateur d'introduction de sommets.**

Les principes de fonctionnement des opérateurs peuvent être de trois types différents :

- *les techniques basées sur la suppression contrôlée d'entités géométriques.* Le principe repose sur un procédé itératif de suppression d'entités (faces, arêtes, sommets) une à une. Le procédé s'arrête lorsque le polyèdre obtenu répond aux critères d'arrêt spécifiés. A ce stade plusieurs techniques cohabitent : il y a les techniques de suppression de sommets (cf. Figure II-9a comme [VERON 97] ou [SCHROEDER 92]), les techniques de fusion d'arêtes (cf. Figure II-9b comme [HOPPE 97] ou [ALGORRI 96]) ou les techniques opérant des suppressions de faces coplanaires à une tolérance près ([HAMMANN 94]),
- *les techniques basées sur le remaillage ou l'insertion de nœuds.* Le principe est d'accroître le nombre de sommets d'un polyèdre. Deux techniques sont possibles. La première consiste à générer un modèle grossier puis d'introduire des points dans ce maillage grossier pour s'approcher petit à petit du maillage initial. Les sommets ajoutés sont soit issus de subdivision de faces (cf. Figure II-9c) [ECK 95] soit issus du maillage initial. La seconde, dite d'enrichissement, fonctionne par introduction de sommets selon des critères de

courbures permettant d'augmenter la précision de la discrétisation dans les zones à forte courbure [TURK 92],

- *les techniques basées sur le regroupement de sommets.* Leur principe consiste à fusionner plusieurs sommets suivant différents critères (géométriques, numériques, ...) et à remplacer ces sommets par un seul sommet représentatif du groupe de sommets fusionnés. L'avantage de ces méthodes est le faible besoin en ressources informatiques. Cependant, la qualité du polyèdre n'est pas garantie [ROSSIGNAC 93].

Indépendamment de la technique choisie, il faut pouvoir contrôler le résultat obtenu. Ce résultat est évalué en fonction de la capacité du modèle final à restituer la géométrie ou la topologie du modèle initial.

La mesure de la restitution de la géométrie est une mesure délicate. La géométrie d'un modèle est souvent appréciée visuellement donc de manière subjective. Chaque méthode de simplification propose ces critères de contrôle pour assurer la qualité géométrique finale. Deux approches différentes se distinguent :

- *approches assurant un contrôle quantitatif de la géométrie.* L'erreur est caractérisée par la distance en chaque point entre les deux modèles. Suivant la méthodologie de simplification, il est possible de définir deux catégories. La première utilise des critères basés sur le calcul de la distance entre les nœuds simplifiés et le polyèdre résultat. Ces critères permettent d'évaluer, itération après itération, l'erreur locale commise lors de chaque simplification [DE HAEMER 91], [DE FLORIANI 96]. La seconde catégorie utilise des critères basés sur la définition d'enveloppe géométrique autour du polyèdre. Le polyèdre final sera validé si ce dernier est à l'intérieur de cette enveloppe. Plusieurs types d'enveloppes existent, Varshney [VARSHNEY 94] définit une enveloppe globale, elle est la même pour tout le polyèdre. Guézic [GUEZIEC 96] propose de définir des valeurs différentes pour chaque face du polyèdre. Enfin, Véron [VERON 97] situe la définition de l'enveloppe au niveau des sommets. Cette gradation dans le type de définition des enveloppes permet d'avoir un contrôle plus fin sur la décimation.

- *approches n'assurant pas un contrôle quantitatif de la géométrie.* Même si ces approches ne mettent pas en œuvre des critères quantitatifs, elles cherchent à assurer une certaine qualité des résultats. Plusieurs approches peuvent être identifiées. Une première catégorie d'approches est basée sur le taux de réduction du nombre de polygones [ROSSIGNAC 93], [HE 96] et [SCHAUFLENER 95]. Une seconde catégorie utilise également le taux de réduction comme critère mais le fonctionnement est affiné grâce à des critères minimisant les erreurs d'approximation [SCHROEDER 92], [HECKBERT 92] et [TURK 92]. Une troisième catégorie utilise des principes de décimation dont l'objectif est de conserver au mieux la géométrie sans contrainte de taux de réduction. Ces techniques sont basées soit sur des regroupements de faces pilotés par des critères mettant en œuvre des seuils d'angles entre les faces ou les arêtes [ALGORRI 96], [HINKER 93], soit sur l'optimisation d'une fonction objectif définie par l'écart entre le polyèdre initial et le polyèdre simplifié [HOPPE 97], [ECK 95].

La restitution de la topologie est validée si le voisinage de chaque entité ne change pas de dimension lors de la simplification. Toutes les méthodes n'ont pas le même comportement en ce qui concerne la restitution de la topologie. Trois approches existent :

- *les approches conservant la topologie du polyèdre initial.* La plupart des approches contrôlent la restitution de la topologie en faisant intervenir des critères locaux pilotant les relations d'adjacences entre les différentes entités du polyèdre [ALGORRI 96], [DE FLORIANI 96], [DE HAEMER 91], [ECK 95] et [VERON 97].
- *Les approches autorisant un changement piloté de la topologie du polyèdre initial.* L'utilisateur peut avoir besoin de changer la variété géométrique d'un modèle, par exemple, idéaliser une pièce (passer d'une poutre percée de trous sur son axe longitudinal à une poutre pleine) [HE 96] et [VERON 97].
- *Les approches n'assurant pas le respect de la topologie du polyèdre initial.* Certaines approches telles que [ROSSIGNAC 93] ou [SCHAUFLENER 95] ne garantissent pas la restitution de la topologie. Le seul critère utilisé est le taux de réduction du nombre de polygones. Le résultat final peut être très éloigné du polyèdre initial.

II.3 - Outils existants pour la préparation des données.

Les études des méthodes nous ont permis de découvrir et de tester plusieurs outils. Parmi ces outils un grand nombre permet de gérer la mise en cohérence et la décimation. En revanche, la suppression des parties cachées étant très liée à l'application finale, peu de solutions répondaient à notre problème. Lors de nos recherches, nous avons pu tester le logiciel Simpoly[®]. Ce logiciel commercialisé par la société GéoNuméric (Grenoble) est un projet encadré par Jean-Claude LEON de l'équipe Conception Intégrée du Laboratoire Sols, Solides et Structures (L3S) de l'Institut National Polytechnique de Grenoble (INPG). Ce logiciel est le fruit de plusieurs thèses. Il a pour vocation de générer des modèles polyédriques conformes pour des calculs par éléments finis. Il intègre déjà des fonctionnalités de mise en cohérence et de décimation. Bien que le domaine d'application du logiciel Simpoly[®] ne soit pas le même que celui de la visualisation, les contraintes sur les modèles polyédriques sont similaires.

II.3.1 - Simpoly[®].

II.3.1.1 - Mise en cohérence implémentée par Simpoly[®].

Cette étape supprime l'ensemble des entités surnuméraires (sommets, arêtes, faces). Cette détection est pilotée par une valeur représentant la distance maximum entre deux entités candidates à une fusion. Après avoir fusionné ces entités surnuméraires, il est possible que des séquences d'arêtes soient confondantes sans pour autant que les sommets les composant le soient. La Figure II-10 illustre ce problème. Les zones de fusion définies par le critère ne permettent pas de fusionner les sommets de la séquence d'arêtes rouge avec les sommets de la séquence d'arêtes noire. Pourtant, les séquences pourraient être fusionnées. Un opérateur de distribution du critère permet de détecter ce genre de problème et de fusionner les arêtes.

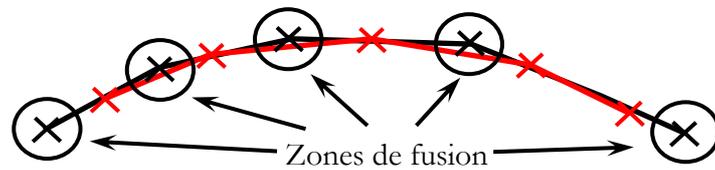


Figure II-10 : Séquences d'arêtes confondantes.

II.3.1.2 - Décimation implémentée par Simpoly[®].

Le fonctionnement de l'algorithme de décimation de Simpoly[®] est un processus itératif de suppression de sommets et de retriangulation locale (cf. Figure II-11).

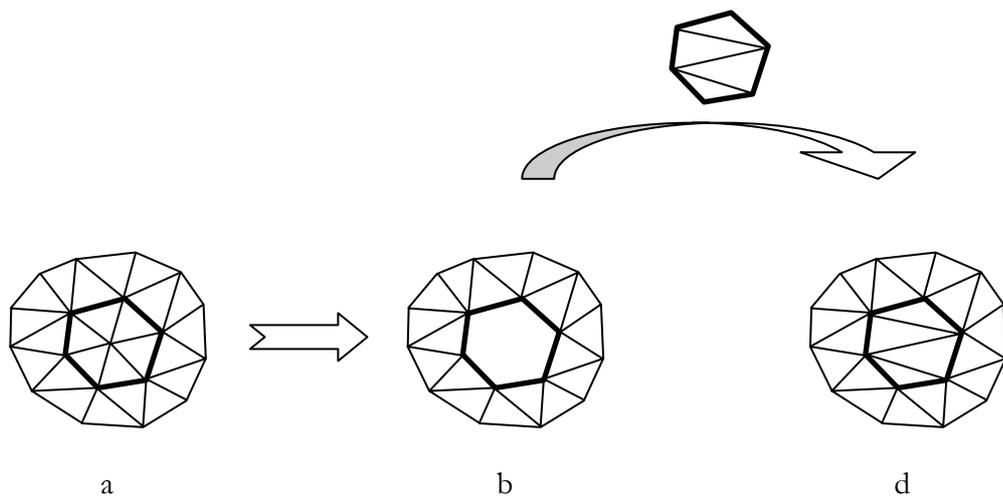


Figure II-11 : Processus de suppression d'un sommet.

a. triangulation initiale, b. trou créé dans la triangulation,
c. remaillage construit, d. triangulation finale.

La difficulté majeure consiste à choisir le sommet à supprimer puis de reconstruire un maillage local adéquat (cf. Figure II-1 b., c. et d.).

Les traitements nécessaires à la décimation des sommets d'un polyèdre se décomposent en quatre étapes :

1. *choix du sommet à supprimer* : le choix du sommet s'appuie sur le calcul d'énergie de déformation². Les sommets sont classés par ordre croissant de la valeur absolue de cette énergie calculée sur l'étoilé³. Les zones planes ou cylindriques seront traitées en priorité (courbures faibles dans une ou deux directions).
2. *création du maillage local* : le polygone contournant le sommet à supprimer est extrait et sert de base pour le remaillage local, deux critères sont utilisés pour la reconstruction du remaillage :
 - un critère dit d'équilatéralité : les faces recrées doivent être les plus équilatérales possibles.
 - un critère dit des hauteurs : l'algorithme cherche à minimiser l'écart entre les arêtes supprimées et les arêtes recrées.
3. *validation de la triangulation* : la triangulation est contrôlée grâce aux zones de tolérances indiquées par l'utilisateur. Ces zones de tolérances définissent une enveloppe autour du modèle. La retriangulation pour être valide doit se situer à l'intérieur de l'enveloppe définie (cf. Figure II-12). Une enveloppe est définie par une répartition de sphère autour du modèle. La taille des sphères est fixée par l'utilisateur.
4. *mise à jour du polyèdre* : une fois la retriangulation validée, le sommet et son étoilé sont supprimés et remplacés par la retriangulation.

² Energie de déformation : par analogie, c'est l'énergie qu'il faut développer pour déformer une plaque mince pour lui donner l'allure du maillage. Il est basé sur le calcul de la courbure gaussienne discrète ramené à un invariant, pour des informations complémentaires le lecteur est invité à se référer à la thèse de Fine ([FINE 01]).

³ Etoilé : ensemble des faces au contact d'un sommet donné.



Figure II-12 : Types d'enveloppes utilisées pour le contrôle de la retriangulation.

a. enveloppe définie avec une seule taille de sphère d'erreur.

b. enveloppe définie avec deux tailles de sphères d'erreurs.

L'étape de décimation peut s'avérer nécessaire lorsque les modèles traités par la mise en cohérence et la suppression des parties cachées contiennent encore trop de faces pour permettre à l'application de réalité virtuelle de fonctionner en temps réel. La décimation sert alors à générer des déclinaisons des modèles de référence avec des précisions géométriques de plus en plus faibles. Pour une faible précision, le modèle généré sera grossier (moins fidèle au modèle de référence) mais il comportera moins de polygones.

L'intérêt majeur de cet algorithme est d'être piloté par un critère d'erreur géométrique (dans l'unité du modèle). Ce critère d'erreur géométrique permet d'assurer la précision du modèle obtenu. Cette information est utile lors de la visualisation des modèles en temps réel. Les utilisateurs coutumiers de la manipulation d'un paramètre tel que la précision sauront juger de la validité de la scène qui leur est présentée. Ce critère est à comparer au critère de pourcentage classiquement utilisé dans les techniques de décimation. Ce critère a pour objectif de garantir à l'utilisateur le nombre final de polygones. Cette technique présente l'avantage de connaître a priori le nombre de polygones du modèle décimé mais ne permet pas de contrôler la qualité géométrique du modèle généré. Cette absence de contrôle de la précision peut être acceptable

dans des domaines où la précision n'est pas déterminante (ex. les jeux vidéo) mais ne peut pas l'être pour des domaines comme celui de la mécanique ou de la conception.

II.3.2 - QSlim [Garland 97].

QSlim est un exécutable mettant en œuvre un algorithme de décimation, il fait partie d'une palette d'outils permettant la visualisation et le contrôle de polyèdres. Ce logiciel de décimation est disponible au téléchargement (sources et exécutable), il est basé sur une technique de regroupement de sommets (cf. Figure II-13). Il fonctionne à partir de **polyèdres cohérents constituant une variété *manifold***. Les sommets doivent coïncider avec d'autres sommets et une arête ne peut pas être commune à plus de deux faces. L'utilisation de cet algorithme nécessite d'avoir préparé et contrôlé en amont les modèles.

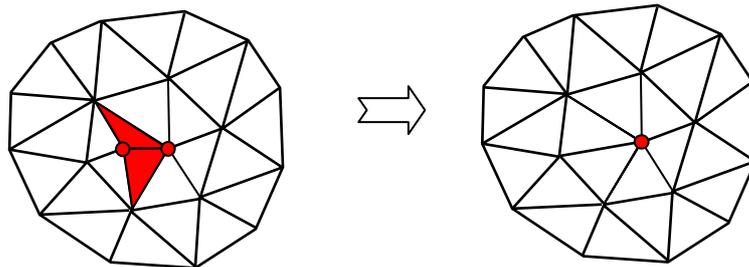


Figure II-13 : Contraction de sommets (QSlim de GARLAND).

La décimation se fait par itération sur le polyèdre avec un contrôle d'erreur. L'algorithme définit des ellipsoïdes⁴ sur chaque sommet, la mobilité du sommet est limitée par cet ellipsoïde. De plus, l'algorithme permet de fusionner des sommets n'ayant pas de lien entre eux mais qui sont spatialement très proches. Cette fonctionnalité permet d'être plus performant dans la décimation et de conserver une cohérence du modèle.

⁴ Les ellipsoïdes définissent le lieu minimisant l'erreur quadratique pour un sommet donné

II.3.3 - DeltaGen de Real Time Technology.

Ce logiciel commercial est un des rares à proposer une solution comprenant un import de modèles surfaciques, un algorithme de suppression de surfaces cachées et un algorithme de décimation. La possibilité de charger un modèle surfacique (modèle CATIA, IGES, ...) est un avantage certain. En effet, rares sont les solutions logicielles faisant cohabiter modèles surfaciques et polyédriques. Il charge et affiche les modèles en conservant les carreaux surfaciques. Toutes les actions sont réalisées au niveau du modèle surfacique. La triangulation n'intervient que plus tard dans le cycle de préparation des modèles. Cependant, il n'est pas possible de sauvegarder le modèle au format natif et de le récupérer dans le modeleur initial pour d'éventuelles modifications. Le logiciel offre la possibilité de mettre en cohérence les modèles, de les trianguler avec un contrôle de type SAG, de générer des déclinaisons des modèles. DeltaGen offre la possibilité de détecter et supprimer les zones cachées selon un ou plusieurs points de vue. L'opérateur doit positionner son assemblage dans la direction de visualisation souhaitée puis demander au logiciel de détecter les carreaux surfaciques visibles, puis positionner l'assemblage selon une nouvelle direction de visualisation ajouter les carreaux surfaciques visibles à ceux déjà sélectionnés et ainsi de suite jusqu'à avoir parcouru l'ensemble des directions de visualisation. Ensuite, il est possible de supprimer les parties non sélectionnées.

II.3.4 - MeshDev [ROY 03].

Pour effectuer une comparaison entre différents algorithmes de décimation, il est nécessaire d'avoir un outil permettant la mesure de la qualité des algorithmes. MeshDev est un exécutable développé par Mickaël ROY au Laboratoire d'Electronique Informatique et Image (LE2I). Ce logiciel mesure des écarts entre deux modèles polyédriques. Ces écarts peuvent être de différentes natures : géométriques, normales, couleurs, textures. Les déviations sont visualisables par une carte de couleur plaquée sur le modèle 3D de référence ou par le calcul de l'écart moyen, de l'écart maximum et de la variance. Il est ainsi facile de visualiser les zones à fort gradient. Ce logiciel sera utilisé pour quantifier la déviation géométrique entre deux modèles générés par deux algorithmes différents.

Définition :

Soient deux maillages M_i et M_j , leurs surfaces respectives S_i et S_j , et un point p_i appartenant à S_i , la déviation géométrique $d_g(p_i, S_j)$ du point p_i à la surface S_j est mesurée avec la distance point à surface :

$$d_g(p_i, S_j) = \min_{p_j \in S_j} d(p_i, p_j)$$

avec $d(p_i, p_j)$ la distance euclidienne entre deux vecteurs. La déviation géométrique est donnée par la distance entre le point p_i et le plus proche point p_j sur la surface S_j (cf. Figure II-14).

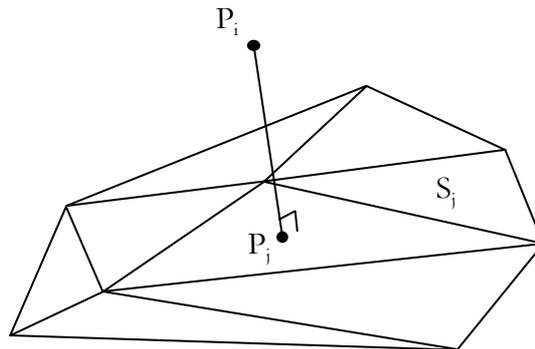


Figure II-14 : Mesure de la déviation géométrique (MeshDev).

La Figure II-15 (a) et la Figure II-15 (b) représentent des illustrations d'un modèle de référence (69541 faces) et d'un modèle décimé (500 faces). L'utilisation de MeshDev avec ces deux modèles permet de générer le modèle de la Figure II-15 (c). Ce modèle est le même que le modèle de référence. Les couleurs affectées aux sommets du polyèdre sont dépendantes de la déviation géométrique calculée en ces points. La couleur rouge correspond à un écart maximum et la couleur bleue à un écart minimum.

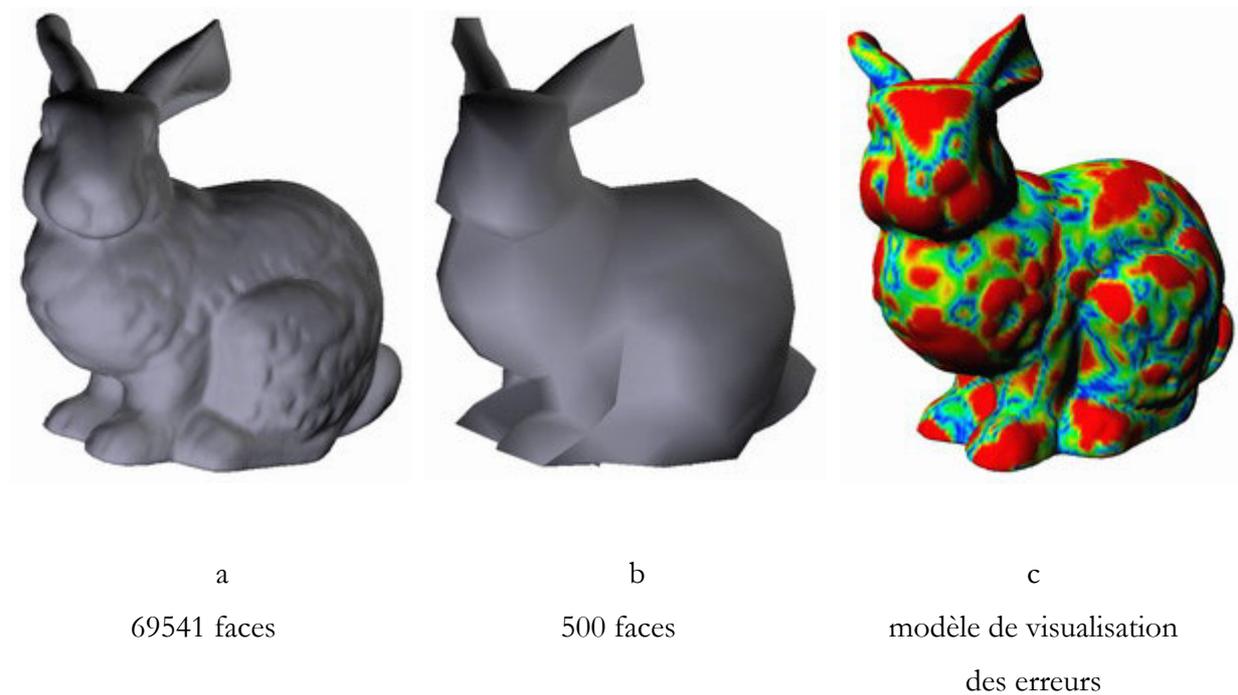


Figure II-15 : Illustrations de la déviation géométrique entre deux modèles.

II.4 - Développement de nouvelles fonctionnalités avec Simpoly[®].

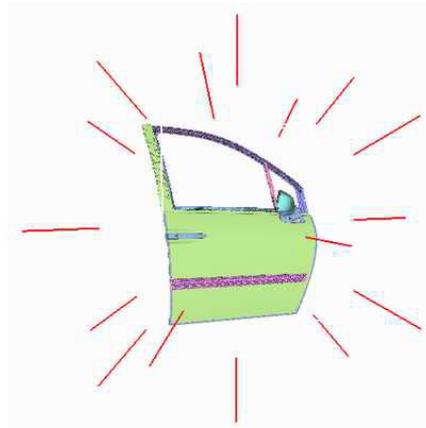
Le logiciel possède deux des trois étapes nécessaires à la préparation des données. La structure et l'exploitation du logiciel offrent la possibilité de développer de nouveaux modules. S'interfacer avec un logiciel existant permet de bénéficier d'un développement avancé sur les fonctionnalités de base comme la création d'un environnement graphique pour l'application, le chargement des divers formats de fichiers, la gestion des polyèdres (structure regroupant sommets, arêtes, faces), les fonctions d'accès aux entités des polyèdres, ... Ainsi, nous avons pu intégrer et tester des algorithmes de suppression de faces cachées.

Trois approches ont été implémentées dans Simpoly[®]. La première est la technique d'élimination des parties arrières (*Back face culling*), la seconde a permis de tester et de comprendre le fonctionnement des arbres de partitionnement de l'espace (BSP), la troisième a vu la mise en œuvre des *P-Buffers*.

II.4.1 - Création et gestion des directions de visualisation.

En étudiant les différentes solutions existantes et possibles pour la détection des parties cachées, nous avons constaté qu'elles étaient forcément toutes dépendantes de la direction de visualisation des modèles. Le premier développement réalisé pour le logiciel a été d'implémenter un système de gestion de directions de visualisation. Pour créer des directions, il suffit à l'utilisateur de positionner et orienter le modèle dans l'interface graphique pour que la vue qu'il a du modèle à travers la caméra du logiciel soit la direction de visualisation qu'il désire définir. Un simple bouton permet de valider et de mémoriser cette direction. Il peut par la suite en définir autant qu'il le souhaite afin de simuler l'ensemble des positions de visualisation possibles. Il a la possibilité d'en supprimer. Ces directions sont représentées par des vecteurs dans l'interface graphique du logiciel.

La Figure II-16 illustre une répartition de directions de visualisation (18 vecteurs au total) autour d'une portière. Les directions ont été positionnées par l'utilisateur de manière à couvrir au maximum l'assemblage. En effet, la portière lors d'une revue de projet pourra être vue sous tous les angles (ex. lors d'une animation d'ouverture).



**Figure II-16 : Répartition de vecteurs.
Appliquée au cas d'une portière de 807-C8 (PSA).**

Ces directions sont des vecteurs, c'est à dire sans origine. Le type d'application finale nous a amené à ajouter des directions de visualisation avec une origine. Ces directions servent pour simuler le point de vue depuis un point donné (ex. siège conducteur, zone difficile d'accès, ...).

II.4.2 - Elimination des parties arrières (*Back face culling*).

Cet algorithme est basé sur l'utilisation de la simplification non uniforme du logiciel Simpoly[®]. La simplification non uniforme est déterminée par une répartition spatiale de sphères. Les sphères sont définies en chaque sommet ou une opération d'interpolation permet d'affecter une sphère à chaque sommet en réalisant une interpolation sur les rayons des sphères les plus proches. Le rayon de la sphère représente la latitude pour un sommet à se déplacer. Un sommet affecté d'une sphère de grand rayon a de grandes chances d'être supprimé lors du processus de décimation. Le principe retenu pour l'implémentation de l'élimination des parties cachées est de distribuer des sphères de grand rayon pour les sommets appartenant à des faces cachées et de petits rayons pour les faces visibles. Par itération sur les directions de visualisation définies par l'utilisateur, les sommets sont affectés de sphères avec le rayon correspondant à leur état (visibles ou cachées).

Figure II-17 illustre les résultats obtenus avec une pièce concave. Le vecteur rouge indique la direction de visualisation de la pièce. Le résultat est illustré sur le pion de fixation. Les faces formant la base du pion sont orientées vers l'observateur. Ces faces ne seront par conséquent pas supprimées alors qu'elles ne seront jamais visibles. Ce résultat n'est pas satisfaisant en terme de nombre de faces détectées car les modèles utilisés comportent un grand nombre de faces dans le cas de celles du pion de fixation.



Figure II-17 : Résultat de l'algorithme : orientation des normales.

II.4.3 - Arbres de partitionnement de l'espace (*BSP*).

Ces algorithmes sont robustes et très rapides. Ils sont très souvent utilisés pour la gestion des polygones de grandes bases de données, par exemple villes ou terrains. Ils permettent de localiser et segmenter rapidement l'espace à partir d'un point donné. Ils renseignent sur la position d'une face par rapport à une autre (i.e. devant ou derrière). Ainsi, pour optimiser une navigation dans une grande base de donnée, il est possible de connaître facilement et rapidement les polygones situés devant l'utilisateur (donc à afficher) et ceux situés derrière l'utilisateur (donc à ne pas afficher). Les BSP semblent être une solution pour la détection des faces cachées. Il est possible de connaître quels sont les polygones situés devant ou derrière un polygone donné mais il n'est pas possible de savoir si un polygone cache partiellement ou entièrement un autre. Dans notre cas, il est primordial d'avoir cette information. L'efficacité du traitement en dépend directement. Dans le cas de la gestion de grandes bases de données, cette information est moins importante et l'utilisation des BSP est accompagnée des techniques classiquement utilisées en temps réel (cône de visualisation, orientation des normales, ...). L'utilisation de ces techniques pour notre problème nécessiterait de définir les directions de visualisation de l'assemblage et de faire des calculs de projection. Ce qui se rapproche de la technique des P-Buffers. C'est pourquoi les développements ont été arrêtés pour les BSP.

II.4.4 - Détection des faces cachées (*P-Buffers*).

II.4.4.1 - Implémentation.

La notion de visibilité des faces est dépendante des notions de points de vue et de directions de visualisation. En partant de ce principe, une méthode de détection des faces visibles sur ces points de vue et directions de visualisation a été construite. L'algorithme des *P-Buffers* calcule une image grâce à une caméra *OpenGL*⁵. Les pixels présents sur cette image sont des points projetés des éléments (triangles) de la scène. Un pixel de l'image correspond à une partie d'un triangle de la scène. Ainsi une relation entre le pixel et les triangles permet de répertorier les triangles visibles depuis la position et l'orientation de la caméra. Les seuls paramètres que contient un pixel sont

⁵ Les différents types de caméras *OpenGL* seront détaillés dans la suite du chapitre.

ses couleurs (Rouge – R, Vert – V et Bleu – B) et éventuellement un niveau de transparence (Alpha – A). Le niveau de transparence n'est pas utilisé dans notre algorithme de codage pour ne pas ajouter d'ambiguïté sur les profondeurs des triangles. Les triangles sont référencés par un indice de face dans la structure de maillage. En codant l'indice d'une face en couleur RVB et en dessinant cette face dans le *P-Buffer*, l'image calculée devient une carte des indices des faces visibles. Le parcours de cette image pixel par pixel permet en décodant la couleur de chaque pixel de connaître l'indice des faces visibles. Cette technique s'apparente à celle des *Items-Buffers* [WEGHORST 84]. Les *Items-Buffers* utilisés dans les algorithmes de calcul de rendu par lancer de rayons dressent une carte de la scène vue depuis la caméra. Chaque pixel de la carte contient l'indice de l'objet visible en ce point. Cette technique n'est pas assez précise car l'information ne doit pas se situer au niveau du modèle mais au niveau du polygone.

Le codage des indices de faces se fait par le calcul suivant :

$$R = \text{indice} \& 255$$

$$V = (\text{indice} \gg 8) \& 255$$

$$B = (\text{indice} \gg 16) \& 255$$

ou $\& 255$ est un masque permettant de ne conserver que la partie entière du résultat de la division de l'indice par 255 et ou $\gg 8$ ($\gg 16$) permet de diviser l'indice par 256 (65536), i.e. 8 bits (16 bits).

Le décodage des couleurs se fait par le calcul suivant :

$$\text{Indice}_{\text{Pixel } i} = R_{\text{Pixel } i} + (V_{\text{Pixel } i} \ll 8) + (B_{\text{Pixel } i} \ll 16)$$

ou $\ll 8$ ($\ll 16$) permet de multiplier par 256 (65536), i.e. 8 bits (16 bits).

Par défaut, toutes les faces du polyèdre sont cachées (potentiellement supprimables), le parcours de l'image rendue par l'algorithme du *P-Buffer* donne le statut de face visible (non supprimable) à toutes les faces détectées.

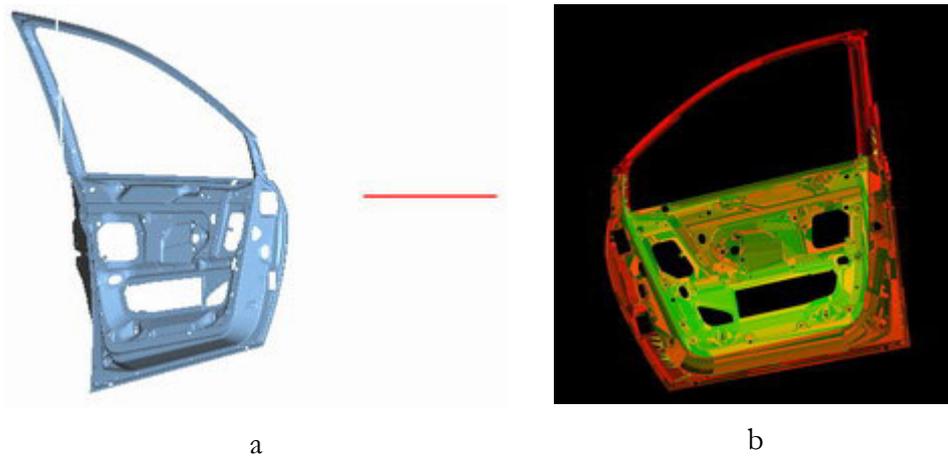


Figure II-18 : Image calculée en remplaçant les couleurs par les index de faces.

La Figure II-18 (b) montre le résultat d'une image calculée par la méthode des *P-Buffers* en codant l'indice des faces sur les canaux RVB de chaque pixel. L'image produite est calculée suivant la direction rouge représentée sur la Figure II-18 (a). Cette opération est à réitérer sur l'ensemble des points de vues différents de la scène. Le choix, le nombre et le type des directions de visualisation sont des éléments discutés dans la suite du document.

Le résultat de l'image calculée est dépendant du choix de la caméra choisie. La bibliothèque *OpenGL* propose deux types de caméras : les caméras à projection orthographique et les caméras à projection en perspective.

Les caméras à projection orthographiques (cf. Figure II-19) réalisent une projection normale à la direction de visualisation de la scène.

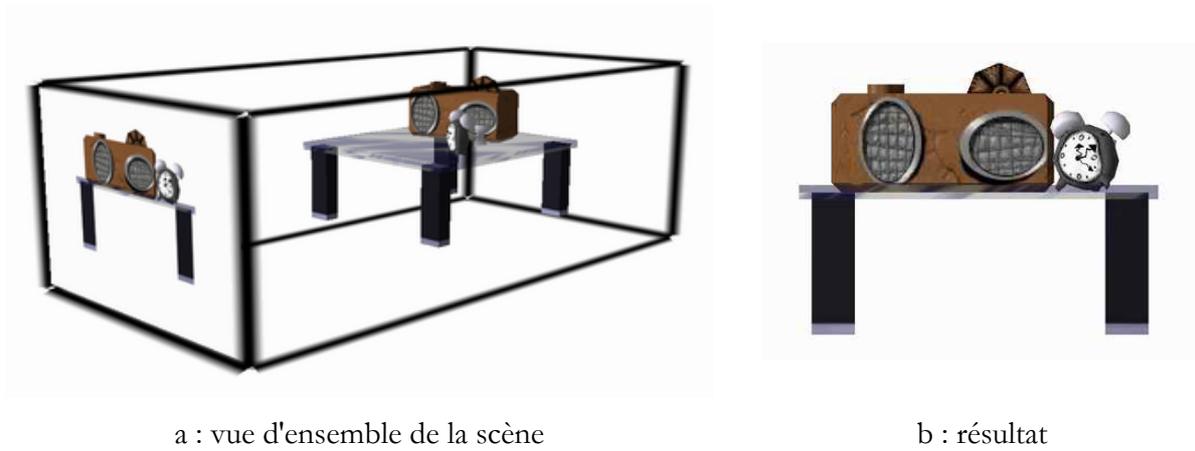


Figure II-19 : Caméra à projection orthographique.

Dans le cas de la scène présentée sur la Figure II-19, il est impossible de différencier les pattes avant et arrière de la table. La perspective est perdue. Ce mode de projection ne correspond pas à notre vision. Afin que l'utilisateur puisse définir correctement la façon dont sera vue la pièce et la façon dont il voit l'ensemble à l'écran, il faut utiliser un autre type de caméra.

Les caméras à projection en perspective permettent de conserver cette notion de perspective (cf. Figure II-20).

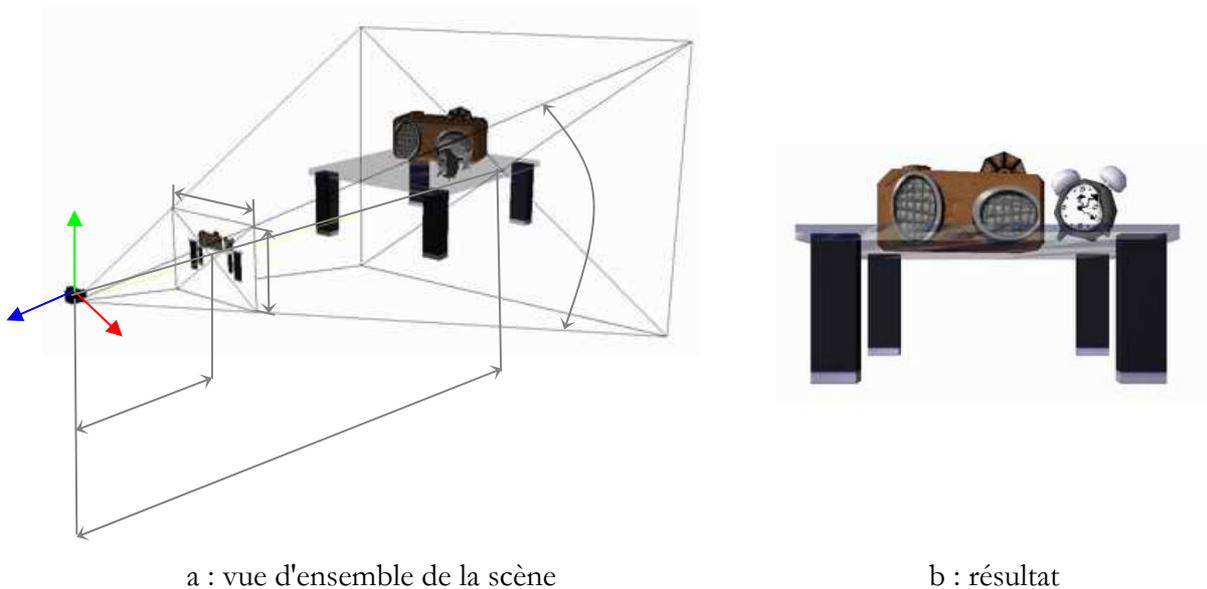


Figure II-20 : Caméra de type perspective.

La direction de visualisation définie par l'utilisateur correspond à l'opposé du vecteur Z de la Figure II-20. Le vecteur Y est choisi de manière aléatoire. Le vecteur X est calculé en fonction des vecteurs Y et Z . C'est dans ce repère (X, Y, Z) que le point d'origine, le point de visée, la distance Z_{Proche} et Z_{Loin} sont calculés pour chaque direction de visualisation.

Le point d'origine est calculé de manière à ce que la totalité de l'assemblage soit vu par la caméra. Aucun polygone ne pourra se trouver en dehors du cône de visualisation de la caméra. Cette vérification est importante car les modèles visualisés sont susceptibles d'évoluer géométriquement et topologiquement. Il faut s'assurer d'avoir le recul minimal pour englober la totalité de la scène. Le point de visée est placé au centre de la boîte englobante de l'ensemble.

Il faut pouvoir simuler la visualisation depuis un point fixe (ex. regard du conducteur dans un véhicule, ...). L'utilisateur peut avoir besoin de définir la vision du conducteur ou du passager pour des raisons d'ergonomie ou tout simplement pour visualiser un endroit difficilement accessible (ex. sous la planche de bord). Cette possibilité est mise en œuvre lors de définition de la direction de visualisation. L'utilisateur choisit de créer une direction avec ou sans origine. La caméra créée pour le calcul de l'image du *P-Buffer* est positionnée à l'endroit indiqué par l'utilisateur. L'image reflète le point de vue exact de l'utilisateur dans l'application.

II.4.4.2 - Performances de l'algorithme.

L'algorithme de détection des faces repose sur le calcul d'une série d'images depuis une série de directions de visualisation. Les résultats et performances de l'algorithme sont dépendants de la résolution des images générées, du nombre des modèles ainsi que du nombre et du positionnement des directions de visualisation. Le fonctionnement global de l'algorithme de détection est détaillé en Annexe 3.

Le fonctionnement et les performances sont illustrés grâce à l'étude d'un cas concret. Il s'agit de l'assemblage complet d'une portière des véhicules 807/C8 Peugeot-Citroën. Cet assemblage comporte 54 modèles. Tous ces modèles ont été extraits de la base de données de modèles CAO, puis triangulés très finement afin d'avoir un grand nombre de polygones pour respecter au mieux les courbes avant d'être mis en cohérence. Cet assemblage est par conséquent la copie de la

portière réelle dans toute sa complexité, présence des joints, présence des pièces toujours cachées (tapis d'insonorisation, moteur lève-vitre, ...). Le nombre total de polygones avant la détection des faces cachées est de : 1 452 360 polygones. L'influence des différents paramètres sur les temps de calculs et les résultats sont étudiés ci après.

Le temps de traitement est fonction du nombre de directions de visualisation et du nombre de modèles. Il dépend du produit :

$$\text{Temps} = f(\text{Nombre de directions}) * (\text{Nombre de modèles}) * (\text{Nombre de modèles})$$

Afin de pouvoir tester les interférences des modèles entre eux, il est nécessaire pour chaque direction de visualisation et pour chaque modèle de dessiner la totalité des modèles présents dans l'assemblage.

La Figure II-21 illustre la variation du temps de traitement en fonction du nombre de directions de visualisation. Elle montre par la même occasion l'ordre de grandeur des temps de traitement. Ces temps peuvent paraître longs, ils sont dépendants du nombre initial de polygones. Dans le cas de la portière, ce nombre est très élevé (1 452 360 triangles) par la présence de pièces géométriquement très complexes (joints de vitre).

La Figure II-22 montre l'impact en terme de pourcentage de conservation de faces visibles en fonction du nombre de directions de visualisation. Il varie de 9% pour 6 directions à 19% pour 58 directions. Plus le nombre de directions est important, meilleure est la couverture, moins de polygones visibles sont supprimés par oubli.

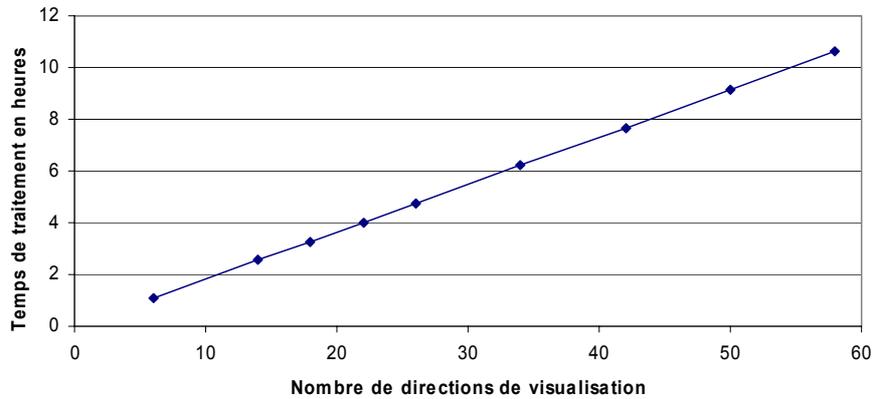


Figure II-21 : Durée de traitement.

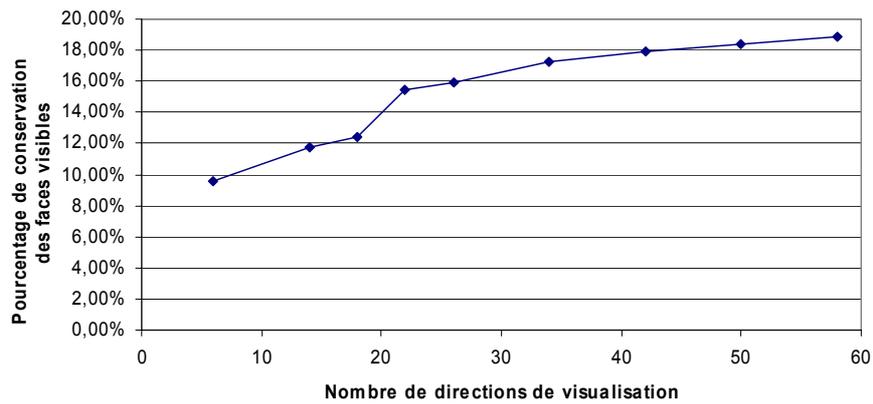


Figure II-22 : Pourcentage de polygones visibles conservés.

Pour la suppression des faces cachées, l'algorithme part du principe que toutes les faces sont supprimables. Toutes les faces identifiées pendant l'exploitation de l'image du *P-Buffer* sont préservées lors de la phase de destruction des faces. Une face plus petite qu'un pixel ou une face non visible depuis les directions de visualisation définies sera effacée. La résolution de l'image calculée a par conséquent une importance sur le résultat final.

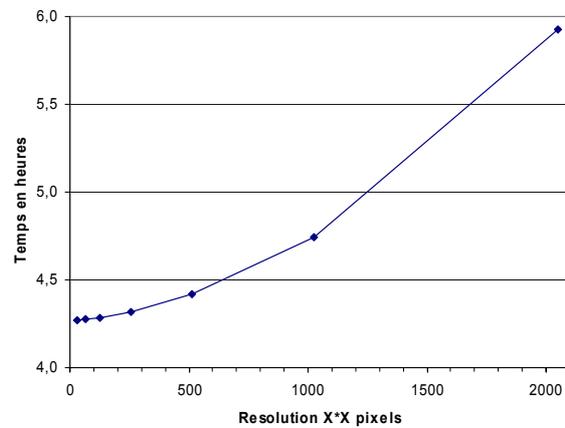


Figure II-23 : Durée de traitement en fonction de la résolution du *P-Buffer*.
(Courbe établie pour 26 directions de visualisation).

La Figure II-23 montre l'impact du choix de la résolution de l'image du *P-Buffer* sur le temps de traitement. L'augmentation de la résolution s'accompagne d'une augmentation du temps de traitement. En effet, plus de pixels sont à traiter avec une résolution plus élevée. Ces résultats sont à considérer avec ceux de la Figure II-24. Les modèles obtenus simplifiés avec une basse résolution sont très grossiers. Le taux de conservation des faces visibles est de l'ordre de 3% (cf. Figure II-24) pour la résolution la plus basse (32 pixels par 32 pixels).

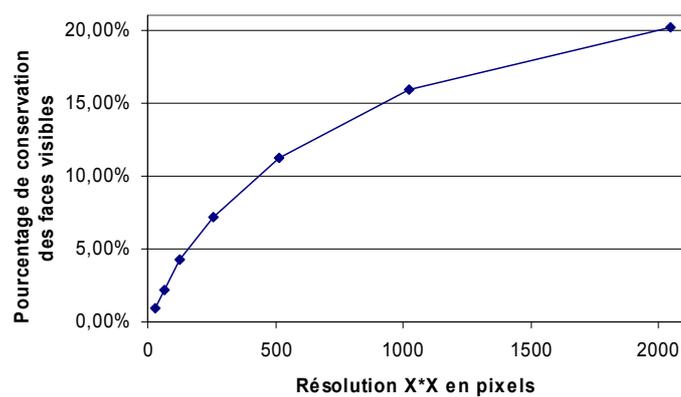


Figure II-24 : Pourcentage de conservation des faces visibles en fonction de la résolution.
(Courbe établie pour 26 directions de visualisation).

II.4.4.3 - Fonctionnement sans l'interface graphique.

Un modèle CAO peut être modifié quotidiennement; il faut que la durée de traitement de ce modèle pour obtenir la version optimisée pour le temps réel ne soit pas supérieure à la durée de remise à jour du modèle lui-même. Ce critère assure une conformité des modèles virtuels vis-à-vis des modèles CAO. La chaîne de traitement ne peut pas être automatisée complètement puisque des paramètres subjectifs entrent en jeu (validation de la mise en cohérence, définition des directions de visualisation, ...). Les évolutions sont continues et de faibles envergures. La précision de génération du polyèdre est la même tout au long du cycle de vie du modèle, celui-ci est vu sous les mêmes angles et la précision requise au final reste identique. Si les paramètres de mise en cohérence, si les directions de visualisation et si les paramètres de décimation sont sauvegardés lors du premier traitement du modèle, il est envisageable de les répercuter sur les évolutions futures du modèle. Cette opération est rendue possible grâce au fonctionnement en mode automatique (sans interface graphique) de Simpoly©. Les paramètres sont sauvegardés dans des fichiers textes afin d'être réutilisés lors du fonctionnement automatique. Ce procédé permet de traiter les modèles modifiés la journée. La présence d'un opérateur n'est plus nécessaire.

II.4.4.4 - Tests et résultats : cas d'une portière.

L'assemblage considéré est le même que précédemment. L'illustration des résultats sera faite pour un nombre de directions de visualisation égal à 26. Nous avons privilégié le taux de conservation des faces par rapport au temps de traitement. Le temps de traitement n'est pas une contrainte majeure pour le prétraitement car cette opération est réalisée pendant les phases de faible activité, principalement la nuit.

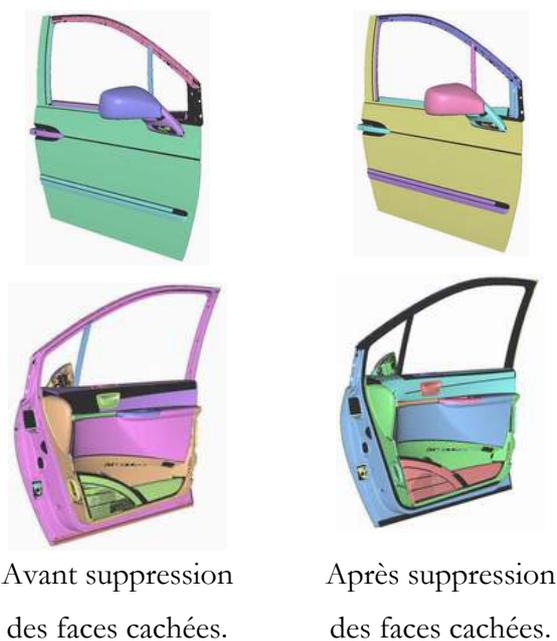


Figure II-25 : Illustration de la suppression des parties cachées, cas de la portière.

La Figure II-25 représente des captures d'écran du logiciel Simpoly[®]. Ces captures illustrent le respect de la géométrie extérieure par l'algorithme. Seules les parties cachées ont été modifiées voire supprimées, les parties visibles sont identiques à celles du modèle initial.

Le Tableau II-1 recense les résultats du traitement. Le taux de réduction est très important (proche des 84%). Ce taux est justifié par le nombre important d'interfaces entre les différents modèles. Ces interfaces ont un rôle mécanique et non pas visuel, par exemple la structure interne de la portière sera complètement effacée.

	Nb. de faces initiales	Nb. de faces supprimées	Nb. de faces finales	Pourcentage de réduction	Pourcentage de réduction moyen
Portière	1 452 360	1 221 452	230 908	84%	67%

Tableau II-1 : Résultats du traitement de la portière par l'algorithme de suppression des faces cachées.

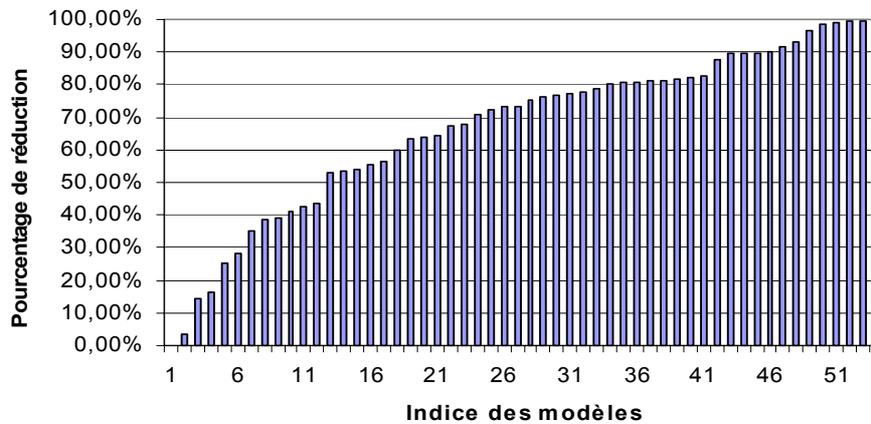


Figure II-26 : Pourcentage de réduction modèle par modèle.

L'axe des abscisses représente chacun des modèles constituant la portière, ils sont représentés par un numéro d'indice

(Les résultats sont organisés de manière croissante pour plus de lisibilité).

La Figure II-26 illustre le pourcentage de réduction modèle par modèle. Le taux moyen de réduction est de 67%. Ce chiffre montre bien la quantité d'information non visible contenue dans les modèles.

Tout assemblage mécanique possède des jeux de fonctionnement et des jeux de montage. Ces jeux sont des espaces entre les modèles qui permettent à un assemblage d'être monté en prenant en compte entre autres les contraintes de fabrication (précision dimensionnelle), les dispersions (variations aléatoires dues aux procédés), ... Dans le cadre du fonctionnement de l'algorithme, les directions de visualisation cherchent et identifient les faces visibles. Si un espace est laissé libre entre deux modèles, même s'il est très petit, les directions de visualisation peuvent aller identifier un polygone situé derrière les modèles. Ce polygone à l'échelle de la vision humaine n'est peut être pas visible pourtant il a été détecté comme visible par l'algorithme. Ce comportement s'applique aussi à tous les espaces vides de l'assemblage. Par exemple, la Figure II-27 b. montre un grossissement d'une zone de l'assemblage de la portière présentant 4 évidements. Ces trous laissent passer les directions de visualisation qui vont identifier et garder un certain nombre de polygones sur des modèles situés à l'intérieur de la portière.

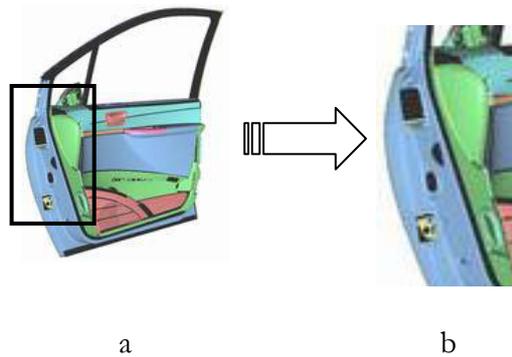


Figure II-27 : Identification des espaces libres.

II.4.4.5 - Tests et résultats : cas d'une planche de bord.

Cet assemblage comporte 54 modèles. Il a été choisi car il possède des pièces de grandes dimensions (visière planche de bord) et des pièces beaucoup plus petites (grilles d'aérateurs). 19 directions de visualisations ont été mises en place pour le traitement. Elles ont été réparties en façade afin de simuler la vision des passagers du véhicule. Sur ces 19 directions, 3 directions possèdent une origine, elles ont été disposées au niveau du volant pour accéder et visualiser la partie inférieure du volant. Le temps de traitement a été de 6 heures avec une résolution de 1024*1024 pour l'image du *P-Buffer* avec un ordinateur de type Pentium III cadencé à 1 giga Hz avec 1 giga de mémoire. Le pourcentage de réduction est là aussi très important : 76% (cf. Tableau II-2). Le résultat est présenté sur la Figure II-28.

	Nb. de faces initiales	Nb. de faces supprimées	Nb. de faces finales	Pourcentage de réduction	Pourcentage de réduction moyen
Planche de bord	1774837	1351764	423073	76%	66%

Tableau II-2 : Résultats du traitement de la planche de bord par l'algorithme de suppression des faces cachées.



Avant suppression des parties cachées.



Après suppression des parties cachées.

Figure II-28 : Illustration de la suppression des parties cachées, cas de la planche de bord.

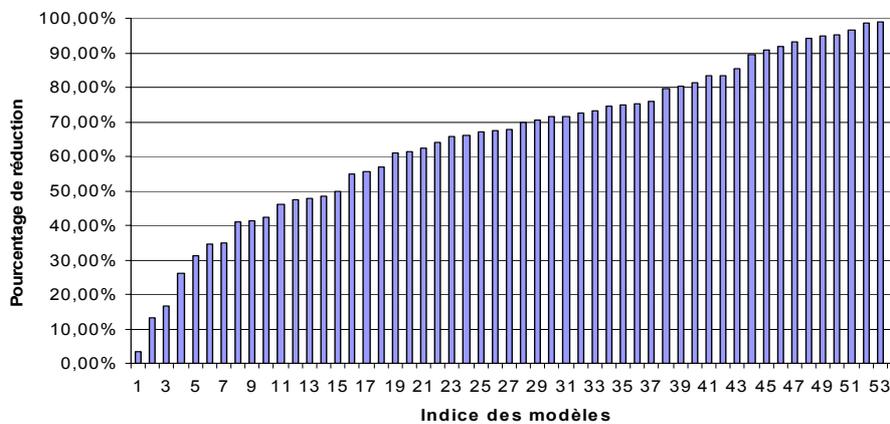


Figure II-29 : Pourcentage de réduction modèle par modèle.

L'axe des abscisses représente chacun des modèles constituant la portière, ils sont représentés par un numéro d'indice

Les résultats sont organisés de manière croissante pour plus de lisibilité.

La Figure II-29 montre que le taux moyen de réduction est de 66% ce qui est très élevé. La géométrie extérieure est exactement la même que celle du modèle original. Elle montre aussi les ouvertures entre les modèles. Elles sont générées par l'absence de modèles (ex : levier de vitesse). Cette absence est due à la difficulté d'extraire ou de trouver le modèle dans la base de données. C'est un cas qui peut arriver en production. Il suffit qu'un partenaire de l'entreprise ne remette pas les modèles en temps voulu et l'assemblage est incomplet. Il faut cependant pouvoir continuer à travailler et à traiter le reste de l'assemblage.

Une autre ouverture très visible est l'écart entre le volant et la planche de bord. Cet espace permet le réglage du volant en hauteur. Habituellement cette ouverture est moins visible car les éléments situés derrière la planche de bord rendent cette zone sombre. Les modèles eux-mêmes peuvent posséder des ouvertures. C'est le cas des grilles d'aération. Ces grilles laissent passer les directions de visualisation pour aller *voir* les polygones situés derrière la façade de la planche de bord. Ces modèles sont géométriquement très complexes et requièrent un grand nombre de polygones pour leur définition. Il n'est pas rare de voir ces modèles remplacés par des textures dans des applications de réalité virtuelle afin de gagner en performances. Le traitement qu'il leur est imposé ici permet de conserver ces modèles en tant que modèles 3D. Les Figure II-30 a. et Figure II-30 b. montrent que le traitement n'a aucune influence sur la perception du modèle dans son ensemble. L'arrière du modèle (cf. Figure II-30 c. et d.) est considérablement dégradé. Le taux de réduction du modèle est de 71%. Un travail manuel aurait pu permettre d'avoir un taux encore meilleur, mais il aurait été certainement long et non capitalisable.



Figure II-30 : Grille d'aération avant et après la suppression des faces cachées.

- a. vue de face avant, b. vue de face après simplification,
- c. vue decôté avant, d. vude côté après simplification.

II.5 - Comparaisons des différentes solutions.

II.5.1 - Simpoly[®] vs QSlim.

Afin de positionner l'algorithme de décimation de Simpoly[®], nous l'avons comparé à celui de QSlim. Les tests ont été effectués sur un volant, la décimation d'une telle pièce est difficile de part sa forme géométrique (forte courbure, concavité). Le modèle initial contient 34601 faces. Pour effectuer la comparaison, une séquence de décimation a été réalisée avec l'algorithme de Simpoly[®]. Les critères de pilotage de l'algorithme ont pris les valeurs : 0.1, 0.5, 1, 1.3, 1.5, 1.7 et 2. Ces décimations ont généré 7 modèles possédant : 19361 faces, 5619 faces, 3226 faces, 2737 faces, 2451 faces, 2246 faces et 2041 faces. Deux méthodes ont été utilisées pour aboutir à ces modèles : la première dite absolue consiste à utiliser toujours le modèle de référence comme modèle de départ pour la décimation, la seconde dite relative consiste à utiliser le modèle i pour générer le modèle $i+1$. Comme l'algorithme de QSlim est piloté par le nombre final de polygones, les séquences de décimation des modèles avec QSlim ont été exécutées grâce aux nombres de polygones des modèles issus des décimations de Simpoly[®]. Suite à ces générations de modèles, les deux algorithmes ont pu être comparés sur un critère de rapidité et de qualité géométrique. En ce qui concerne Simpoly[®], le respect de la consigne de décimation sera aussi étudié. La consigne pour QSlim est forcément respectée puisque la seule consigne est le nombre final de polygones.

Le Tableau II-3 montre clairement que l'algorithme de Garland est beaucoup plus rapide que celui de Simpoly[®] quelque soit la méthode utilisée (relative ou absolue). Cependant, la méthode relative de Simpoly décroît fortement (cf. Figure II-31) pour des séquences de décimation de plusieurs modèles. Cette décroissance est importante car il sera nécessaire de générer des déclinaisons de chaque modèle, l'explication et la justification de ces déclinaisons seront exposées à la fin de ce chapitre.

Nb de faces	QSlim		Simpoly [®]		
	Absolue (secondes)	Relative (secondes)	Absolue (secondes)	Relative (secondes)	Précision géométrique imposée
19361	7.64	7.64	35	35	0.1
5619	7.20	3.17	23	17	0.5
3226	7.20	1.01	19	5	1
2737	7.16	0.56	17	2	1.3
2451	7.25	0.58	16	2	1.5
2246	7.26	0.41	16	2	1.7
2041	7.42	0.31	16	2	2
Total	51.13	13.76	142	65	

Tableau II-3 : Comparaison des temps de décimation de QSlim et de Simpoly[®].

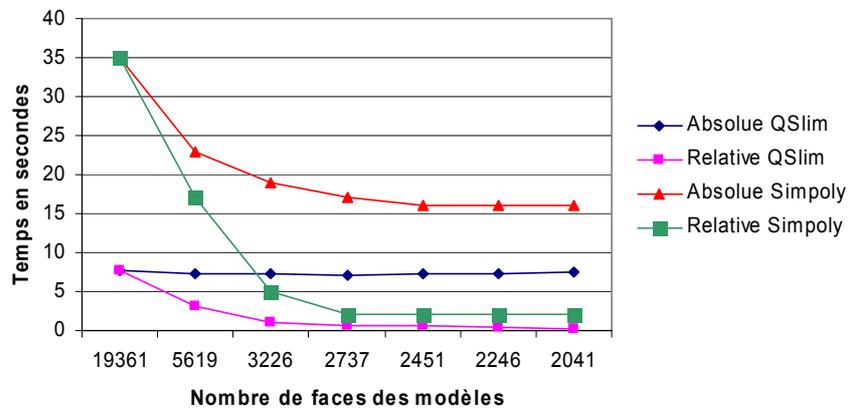


Figure II-31 : Durée de décimation de QSlim et de Simpoly[®].

QSlim ne donne pas d'indication quant à la précision géométrique des modèles, la mesure de l'écart géométrique entre le modèle référence et les modèles générés a été mesurée grâce au logiciel *MeshDev*.

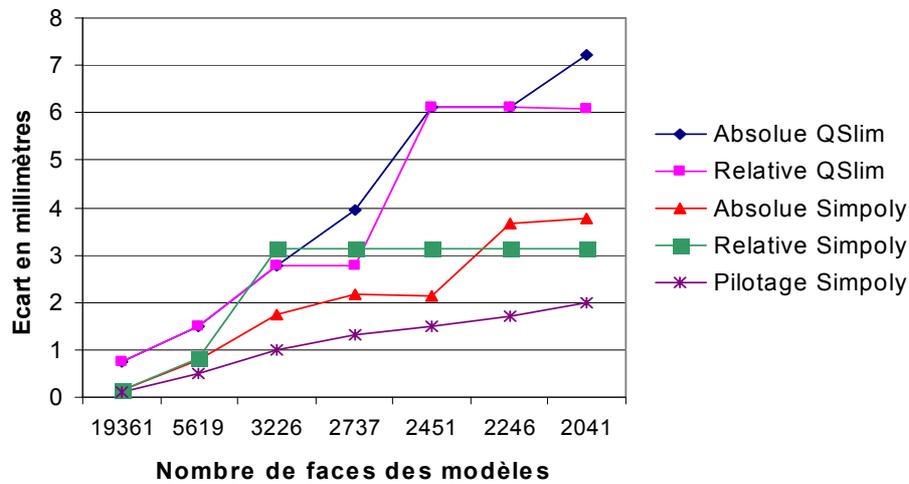


Figure II-32 : Ecart maximum entre les modèles générés et le modèle de référence.

Les résultats de mesure ont été portés sur la Figure II-32 pour l'écart maximum et sur la Figure II-33 pour l'écart moyen. Les deux méthodes de deux algorithmes ont été représentées ainsi que les critères de pilotage de Simpoly[®]. Cette courbe permet de constater que l'algorithme de Simpoly[®] n'a pas respecté parfaitement ce critère pour l'écart maximum. Cette dérive est une dérive localisée à un nombre limité de points, puisque l'écart moyen pour tous les modèles (cf. Figure II-33) est largement en dessous de la valeur de pilotage de l'algorithme de Simpoly[®].

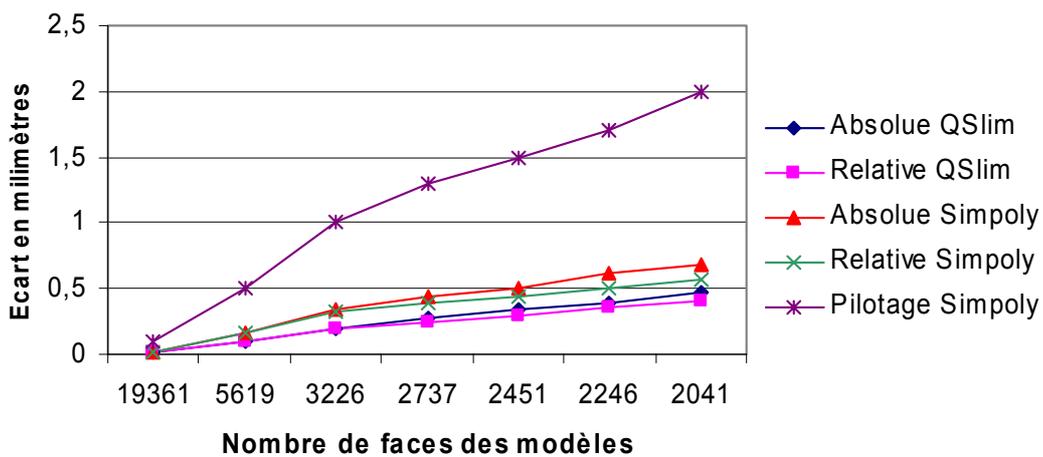


Figure II-33 : Ecart moyen entre les modèles générés et le modèle de référence.

La Figure II-33 montre par la même occasion un écart moyen plus faible pour les solutions proposées par QSlim que par celles proposées par Simpoly[®].

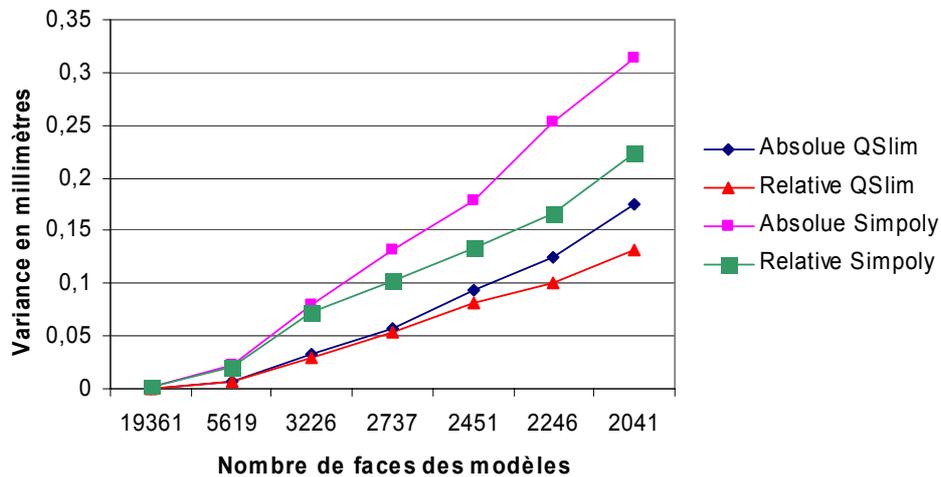


Figure II-34 : Variance entre les modèles générés et le modèle référence.

La Figure II-34 trace la valeur de la variance calculée entre les différents modèles. QSlim semble proposer une décimation de meilleure qualité que Simpoly[®].

Bien que QSlim offre des performances supérieures à celles de Simpoly[®] pour la décimation, le choix de Simpoly[®] a été fait pour plusieurs raisons. Il est possible de distinguer deux catégories de raisons, la première regroupe les raisons de fond :

- la décimation proposée par Simpoly[®] supprime des sommets du maillage initial, les sommets subsistants après la décimation sont identiques à ceux présents avant la décimation. QSlim intègre dans son fonctionnement des déplacements de sommets. Cette technique permet d'avoir des décimations plus poussées. Le risque est de s'éloigner de la géométrie initiale,
- la décimation dans Simpoly[®] peut ne pas être uniforme sur l'ensemble du polyèdre, il est possible de décimer plus fortement dans certaines zones identifiées que dans d'autres,

la seconde regroupe les raisons conjoncturelles et pratiques :

- intégration de la mise en conformité des modèles : QSlim requiert des modèles cohérents pour fonctionner correctement, cela sous-entend de posséder un autre logiciel pour mettre en cohérence les modèles à décimer. La solution développée avec Simpoly[®] intègre cette fonctionnalité, ce qui permet de travailler les modèles dans un seul environnement,
- fonctionnement via une interface graphique ou en mode automatique, QSlim ne possède pas d'interface graphique; celle-ci est importante pour le positionnement et le contrôle des directions de visualisation,
- décimation pilotée par un écart géométrique et non par un nombre final de polygones comme l'est QSlim,
- logiciel commercialisé, donc maintenu et avec un support, ce point est important dans un contexte industriel, QSlim est un outil développé dans le cadre de travaux de recherche sans structure commerciale.

II.5.2 - Comparaison entre l'algorithme des *P-Buffers* et DeltaGen.

Le logiciel DeltaGen de la société Real Time Technology permet de travailler des modèles surfaciques pour générer des modèles polyédriques. Il est compatible avec de nombreux formats de modèles surfaciques type CATIA, IGES, Les modèles issus des logiciels de conception n'ont pas à être convertis avant l'utilisation de DeltaGen. Lors du chargement des fichiers, une triangulation par défaut est appliquée à chacun des modèles pour permettre le fonctionnement du logiciel et de ses fonctionnalités. Le logiciel offre la possibilité de mettre en cohérence les modèles, de supprimer les parties cachées et les décimer. Dans cette partie seule la suppression des parties cachées a été testée. Il est à noter qu'aucune capitalisation n'est prévue pour les actions menées, tout le travail sera à recommencer lors de l'évolution des modèles de l'assemblage.

Le test a été effectué avec l'assemblage de la portière déjà utilisé. Pour des raisons de gain de temps de traitement et d'import des modèles (faible stabilité des fonctions de chargement des modèles surfaciques), seulement 30 modèles ont été conservés. Ces modèles correspondent aux modèles composant l'extérieur de la portière. Ils ont subi une suppression des parties cachées avec le logiciel DeltaGen et avec le logiciel Simpoly[®]. Les résultats sont présentés dans le Tableau II-4. La Figure II-35 représente le nombre de faces de chacun des 30 modèles composant la portière; trois courbes sont dessinées: la première représente le nombre de faces des modèles avant simplification, la seconde illustre le résultat avec la simplification générée par le logiciel DeltaGen et la troisième est la solution obtenue par le logiciel Simpoly[®].

	Nombre de faces		
	Initial	Après simplification	% de réduction
Delta Gen	464805	261445	45%
Simpoly	464805	137569	70%

Tableau II-4 : Résultats de simplification DeltaGen / Simpoly.

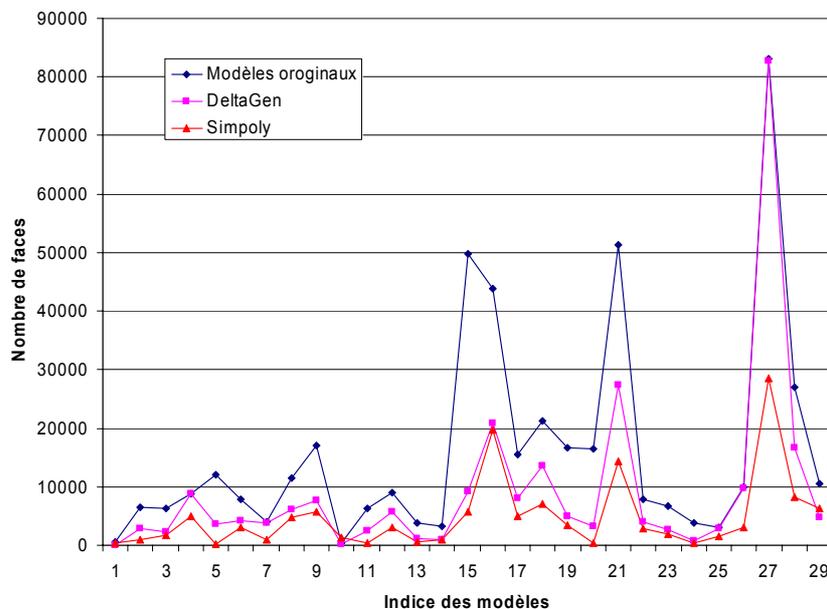


Figure II-35 : Répartition du nombre de faces par modèle (cas de la portière).

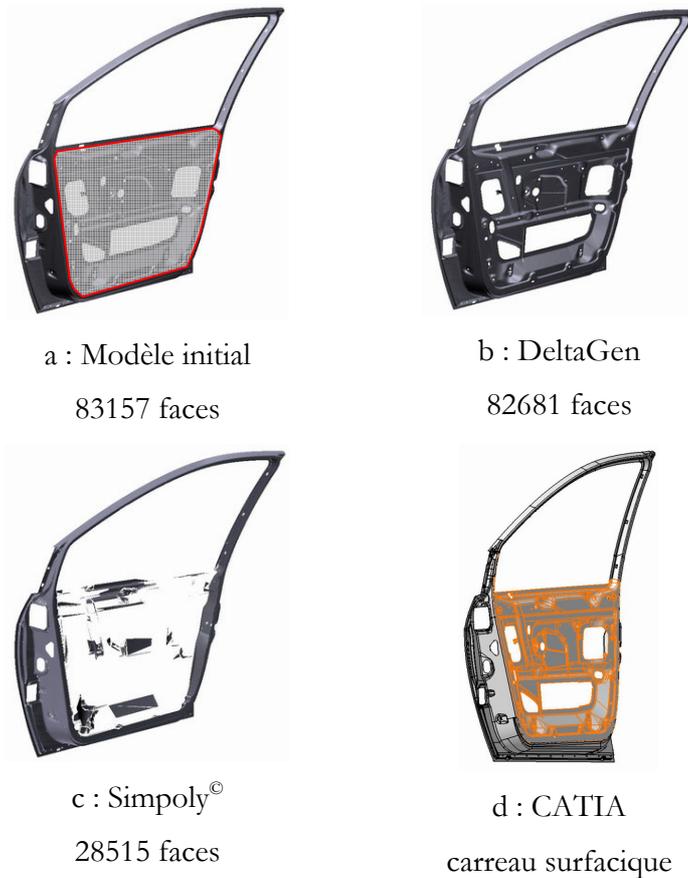
La solution proposée par Simpoly[®] est plus efficace en terme de nombre de polygones restants. De plus, lors du test, le logiciel DeltaGen a posé des difficultés; il a été impossible de sauvegarder et d'exporter correctement les modèles simplifiés. Un format intermédiaire a dû être utilisé pour exploiter les modèles.

Les courbes de la Figure II-35 montrent que le modèle indicé 27 n'a presque pas été simplifié par DeltaGen. Ce modèle est celui présenté sur la Figure II-36. Il s'agit de la doublure de porte. Cet élément reçoit d'un côté la tôlerie (habillage extérieur) de l'autre côté la garniture intérieure. Ainsi, la zone délimitée en rouge sur le modèle initial de la Figure II-36 (a) représente une zone qui peut être entièrement supprimée puisque jamais visible lors d'une revue de visualisation. Or, le logiciel DeltaGen a conservé presque tous les polygones de cette zone (cf. Figure II-36 b.). La raison est que le logiciel fonctionne au niveau des carreaux surfaciques. Si une petite partie d'un carreau est visible alors son intégralité est conservée. Dans le cas de la doublure de portière, la conception du modèle est telle que les carreaux composant la zone à supprimer débordent de celle-ci et sont partiellement visibles de l'extérieur. La totalité du carreau est gardée puisque identifiée comme visible (cf. Figure II-36(d)).

La conséquence directe est le faible taux de simplification du modèle (cf. Figure II-36(b)). En contrepartie, la solution mise en œuvre avec Simpoly fonctionne au niveau du polygone. La Figure II-36(c) illustre le résultat obtenu avec la méthode Simpoly(c). Le nombre de polygones supprimés est bien plus important, cependant il subsiste encore des polygones dus aux ouvertures entre les modèles.

La solution proposée par DeltaGen est moins performante que celle mise en œuvre dans Simpoly[®] pour la suppression des parties cachées. A ce titre, les modèles issus de DeltaGen ont été traités par Simpoly afin de savoir s'il était possible de simplifier davantage, le taux de réduction mesuré est de 52% soit un total de 126061 polygones au final. L'écart entre le résultat de la simplification par Simpoly[®] seul (soit 137569 polygones) et celui de la simplification

DeltaGen-Simpoly (soit 126061 polygones) provient de la différence entre les directions de visualisation choisies. Dans un cas Simpoly est utilisé sur les modèles initiaux et dans le second cas il est utilisé sur des modèles ayant déjà subi une simplification avec des directions définies par un autre utilisateur dans le logiciel DeltaGen. Le nombre final de polygones ne peut donc pas être exactement le même.



**Figure II-36 : Illustration de l'impact des carreaux surfaciques.
Cas de la doublure de portière.**

II.6 - Conclusion.

Ce chapitre présente la préparation des modèles en vue d'une utilisation par une application de réalité virtuelle. Les contraintes sont d'utiliser les modèles surfaciques comme modèles de référence et de générer des modèles optimisés pour des applications de réalité virtuelle. Une chaîne de traitement répondant à ce besoin a été mise en place. Elle se décompose en trois étapes qui sont :

- mise en cohérence des polyèdres,
- suppression des faces cachées,
- décimation.

L'étude des solutions et outils existants nous a amené à choisir le logiciel Simpoly[®] pour la mise en cohérence et la décimation. En s'appuyant sur la structure de données de Simpoly[®] pour la gestion des polyèdres, un nouveau module permettant la détection des faces cachées a été développé. Il fournit des résultats très satisfaisants en terme de taux de réduction. Les résultats sont directement liés à la répartition et au nombre de directions de visualisation.

De plus, cette chaîne de traitement autorise la capitalisation des actions menées par les utilisateurs. L'export des déclinaisons des modèles est ainsi assuré de manière automatique.

Les modèles obtenus en sortie de la chaîne sont correctement orientés, sans discontinuité et décimés avec une précision géométrique donnée. Cette précision est imposée par les métiers qui seront utilisateurs de la future maquette virtuelle. Elle permet de réduire le nombre initial de polygones mais ne garantit pas que le total du nombre de polygones contenus dans la scène soit inférieur aux capacités de la machine pour permettre un rendu en temps réel. Comme il n'est pas envisageable de dégrader davantage les modèles, une deuxième optimisation doit être développée pour la phase de visualisation.

Chapitre III

Visualisation

L'objectif de ces travaux est de proposer une visualisation réaliste en temps réel aux stylistes pour des revues de projets d'intérieur et d'extérieur de véhicule. Pour atteindre cet objectif, les modèles sont préparés, *optimisés*. Cette optimisation est indispensable pour tendre vers des applications en temps réel. L'optimisation présentée ci-avant concerne la qualité géométrique des modèles. Elle est réalisée grâce à des critères objectifs, des critères numériques, chiffrables. Le résultat de l'optimisation est un nombre de polygones. Ce nombre est une conséquence des opérations de simplification des algorithmes. Il se peut que pour satisfaire aux critères de tolérance géométrique acceptable par les utilisateurs, le nombre de polygones soit trop élevé pour que l'application puisse les gérer en temps réel. Un compromis est à trouver entre la gestion en temps réel des données et la qualité des modèles affichés.

Le fonctionnement de l'application et la gestion du contenu peuvent être optimisés également. Ce chapitre est consacré à la compréhension des enjeux de la visualisation, au fonctionnement de

la vision humaine et à l'implémentation informatique d'un modèle simplifié de cette vision pour une optimisation du compromis qualité de rendu / calcul temps réel.

La littérature ainsi que les éditeurs de logiciels proposent beaucoup d'études et de solutions pour l'optimisation de la visualisation de modèles polyédriques. Ces solutions sont principalement développées pour des environnements et des périphériques de visualisation de type écran d'ordinateur. Or, le contexte de travail est différent. L'utilisateur est immergé au cœur de la scène 3D. Il est entouré par les modèles virtuels. Cette situation permet de reconsidérer les gestions actuelles de visualisation en temps réel.

III.1 - Enjeux de la visualisation.

La visualisation, dernier maillon dans la chaîne présentée dans ce document, joue un rôle capital. Elle permet d'orchestrer et de mettre en scène les différents modèles préparés. Cette mise en œuvre passe par le rendu des modèles mais aussi par l'interactivité possible avec la base de données 3D. Il est nécessaire d'avoir une fluidité suffisante pour le confort de l'utilisateur et la validité de l'application. Les modèles ont été préparés dans un souci de qualité géométrique et de conformité esthétique. Le nombre de polygones obtenu peut être trop élevé par rapport aux capacités du calculateur. Il est nécessaire d'optimiser encore la scène pour atteindre la fluidité requise. Clark [CLARK 76] fut le premier à proposer une optimisation des scènes comportant un grand nombre de polygones. Le principe repose sur la corrélation entre la capacité de l'utilisateur à percevoir des détails et la précision géométrique des modèles proposés. Ainsi, il n'est pas nécessaire d'afficher un modèle avec des détails géométriques très fins si la position de cet objet ne permet pas à l'utilisateur de percevoir ces détails. En proposant un modèle dégradé (i.e. moins de polygones), il est possible d'accroître les performances de l'application sans amoindrir la perception de l'utilisateur. Ces modèles dégradés s'appellent des *niveaux de détails*. La définition, les différents types de niveaux de détails et leur mise en œuvre seront détaillés ultérieurement. L'utilisation d'une telle technique impose aussi de bien connaître et comprendre le fonctionnement du système visuel humain.

III.2 - Optimisation et niveaux de détails.

III.2.1 - Différents types de niveaux de détails.

Le gain en performance d'une application passe généralement par la mise en œuvre de niveaux de détails. Un niveau de détail d'un modèle est ce même modèle décliné en une ou plusieurs versions dégradées (i.e. moins de polygones). Cette dégradation est ajustée de façon à ne pas amoindrir la perception et la compréhension de la scène par l'utilisateur. Plusieurs types de niveaux de détails existent, il est possible de les regrouper en trois catégories :

- *Les niveaux de détails discrets* : cette approche proposée initialement par Clark [CLARK 76] consiste en la génération de plusieurs versions dégradées d'un modèle. Ces versions sont alors affichées par l'application en temps réel. La méthode des niveaux de détails discrets est très performante en terme de temps de calcul. Cependant, la simplification des modèles doit être uniforme puisque aucune condition de visualisation n'est référencée. Cette technique est aussi appelée méthode isotropique ou niveaux de détails indépendants du point de vue.
- *Les niveaux de détails continus* : cette technique s'oppose à la précédente à deux niveaux : premièrement cette technique est réalisée en temps réel, deuxièmement les changements d'un niveau de détail à un autre sont quasiment imperceptibles puisque le traitement est effectué à chaque image. La variation est donc très faible. Cette technique, introduite par Hoppe [HOPPE 96], est aussi appelée niveaux de détails progressifs. Le terme *continu* est employé pour faire le parallèle avec la technique dite discrète.
- *Les niveaux de détails dépendants du point de vue* : cette technique [LUEBKE 01] est une extension de la précédente. Elle permet de simplifier de manière non uniforme un même modèle. Cette possibilité est très intéressante dans le cas de grands modèles (ex. terrains, ville, ...).

III.2.2 - Gestion des niveaux de détails.

Indépendamment de la technique des niveaux de détails choisie, la gestion de ces derniers est cruciale pour le gain de performances et pour la qualité de la perception de la scène. Le critère le plus fréquemment utilisé pour le choix d'un niveau de détails est celui de la distance séparant l'utilisateur de l'objet (cf. Figure III-1).

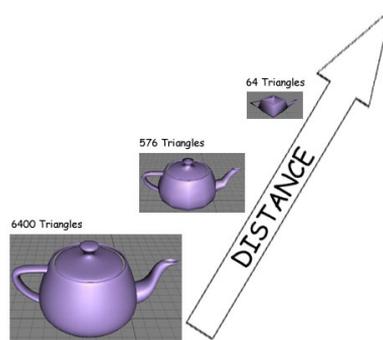


Figure III-1 : Gestion des niveaux de détails par la distance.

Ce critère induit un scintillement lors du passage d'un niveau à un autre. L'utilisation d'un filtre de type hystérésis (cf. Figure III-2) permet de supprimer le scintillement d'un objet situé proche de la distance de changement de niveau de détail.

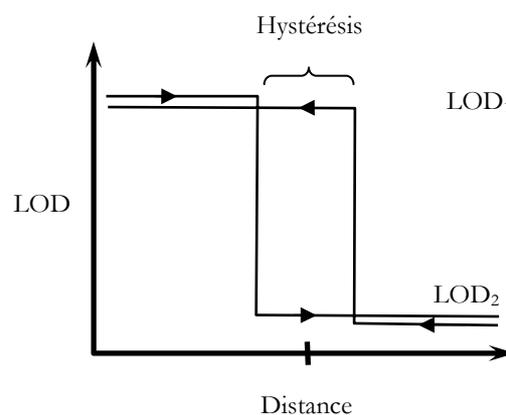


Figure III-2 : Gestion des niveaux de détails avec un filtre d'hystérésis.

Le modèle LOD₁ est de meilleure qualité que le modèle LOD₂.

Le choix du niveau de détail d'un modèle peut aussi se faire en fonction de la taille de la surface de la projection de l'objet sur l'écran (cf. Figure III-3). Plus la surface est grande plus l'objet a d'importance dans la scène donc plus il a besoin de détails. Des recherches ont été menées [LUEBKE 97] afin d'affiner le calcul de la taille de la projection et le choix de la forme englobante (boîte, sphère, ...). D'autres types de choix de niveaux de détails sont développés en fonction de l'application finale ([LUEBKE 03]).

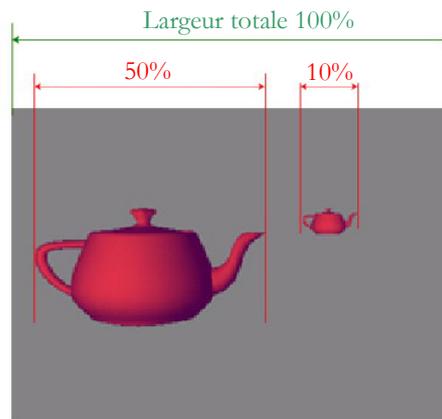


Figure III-3 : Choix des niveaux de détails piloté par la taille de projection.

La manière d'utiliser ces méthodes de gestion de niveaux de détails dans un contexte technologique donné (calculateur graphique, dispositif de visualisation) doit prendre en compte les facteurs de perception. En effet, la gestion des niveaux de détails peut s'appuyer sur le fonctionnement du système visuel humain pour déterminer la quantité minimum d'information nécessaire (i.e. nombre de triangles) à afficher pour que la perception de la scène soit maximale. Aujourd'hui peu d'applications prennent en compte les facteurs humains. Selon Tory et Moller [TORY 04] seulement 23% des articles *IEEE Transactions on Visualization and Computer Graphics* traitent des facteurs humains alors que les applications développées sont des applications de visualisation. Ce chiffre a été établi sur la période 1995 à 2002. C'est pourquoi, il semble important de comprendre le fonctionnement du système visuel humain.

III.3 - Fonctionnement du système visuel humain.

La vision humaine est une vision stéréoscopique. La stéréoscopie provient de la différence de point de vue entre les deux yeux. Chaque œil reçoit et transmet au cerveau une image différente de la même scène. La conjugaison de ces deux images par le cerveau permet de reconstruire la scène en trois dimensions et ainsi de percevoir la profondeur. Avoir deux images différentes n'est pas le seul critère pour voir une scène en 3 dimensions. En effet, il est possible de percevoir la profondeur dans des images en 2 dimensions. Dans ce cas, d'autres notions (taille, ombres, perspectives, ...) entrent en jeu et permettent au cerveau de reconstruire une image en 3D (cf. [FUCHS 03]).

Le champ visuel humain couvre un volume défini par un angle d'ouverture de 180° horizontalement et pour un angle de 120° verticalement (cf. Figure III-4).

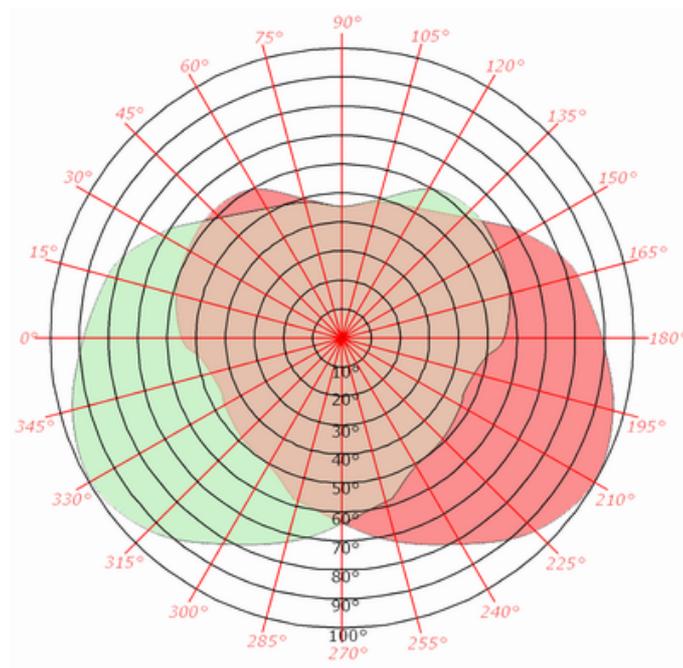


Figure III-4 : Limites du champ visuel d'après [ARTHUR 00].

Zone verte : œil droit, zone rouge : œil gauche.

La vision se décompose en deux fonctions : *le transport du flux lumineux* (la cornée, l'iris, la pupille et le cristallin) et *l'acquisition et le traitement de ce flux lumineux*. L'acquisition est assurée par la rétine (composée de cônes et de bâtonnets). Un premier niveau de traitement est assuré par les photorécepteurs qui tapissent la rétine. Les informations sont véhiculées jusqu'au cerveau par les fibres du nerf optique (cf. Figure III-5).

Il y a dans chaque œil environ 120 millions de bâtonnets répartis sur la majeure partie de la rétine. Les bâtonnets sont environ 100 fois plus sensibles à la lumière que les cônes. Ils ne réagissent qu'à l'intensité lumineuse et servent à la vision scotopique (vision nocturne). Les bâtonnets n'interviennent pas dans la vision des couleurs. C'est pourquoi seules les formes et non les couleurs sont perçues dans la pénombre. Les cônes, au nombre d'environ 7 millions, servent à la vision photopique (vision diurne) et à la discrimination des couleurs. Ceux-ci sont plus courts et plus épais que les bâtonnets. Ils réagissent 4 fois plus vite. Les cônes ont besoin de plus de lumière que les bâtonnets pour être excités. La fonction et la répartition de ces cônes et bâtonnets (cf. Figure III-6) définissent en partie l'acuité et la résolution du système visuel (source <http://www.essilor/RESSOURCES-SCIENTIFIQUES/ARCHIVES/Anatomie/anatomie01.html>).

L'acuité visuelle n'est pas uniforme sur l'ensemble du champ visuel pour deux raisons principales. La première est une conséquence de la répartition non uniforme des cônes et des bâtonnets. A ce titre, il est important de considérer plusieurs zones de la rétine. La **macula**, tache jaune, apparaît au centre de la chambre postérieure comme une fine excavation. C'est dans cette zone que se trouve le maximum de cônes. Cette zone permet donc une vision très précise. Elle mesure 3 mm dans le grand axe et 2 mm dans le petit axe. La **fovéa** est une région de la rétine située dans la macula, près de l'axe optique de l'œil. Elle est la partie centrale de la macula. Elle mesure 1300 à 1500 microns. Elle contient 400 000 cônes. Elle est de la plus haute importance pour la vision. C'est elle qui donne la vision la plus précise, en éclairage diurne. L'acuité sera maximale à cet endroit, et très faible ailleurs (cf. Figure III-6). Cette faiblesse est compensée par la rapidité de rotation des yeux (~900 degré/s). Dans une vision encore plus centrale on trouve la **fovéola**. Elle mesure 300 à 400 microns de diamètre et contient 25 000 cônes. Plus au centre on trouve une zone ponctuelle qui s'appelle le **bouquet de cônes centraux**. Il mesure 100 microns

et contient 2500 cônes. La seconde concerne l'interface entre les photorécepteurs (cônes et bâtonnets) et les fibres du nerf optique. Au niveau de la fovéa, chaque photorécepteur est relié directement à une fibre du nerf optique. Ailleurs, plusieurs photorécepteurs sont reliés à une seule fibre par l'intermédiaire des cellules bipolaires, horizontales et endocrines. Cette variation de densité rend l'acuité visuelle encore plus faible en vision périphérique.

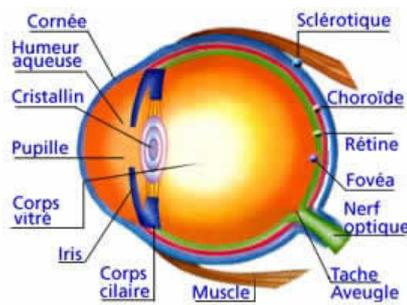


Figure III-5 : Anatomie de l'œil.

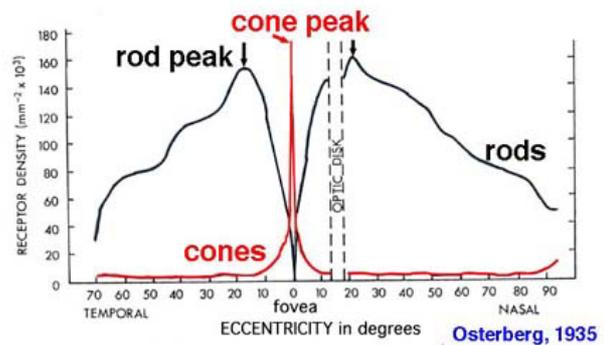


Figure III-6 : Répartition des cônes (cones) et des bâtonnets (rods), [OSTERBERG 35].

La Figure III-7 montre la variation de l'acuité visuelle en fonction de l'excentricité. Elle met en évidence la faiblesse de l'acuité visuelle en vision périphérique et sa finesse en vision centrale.

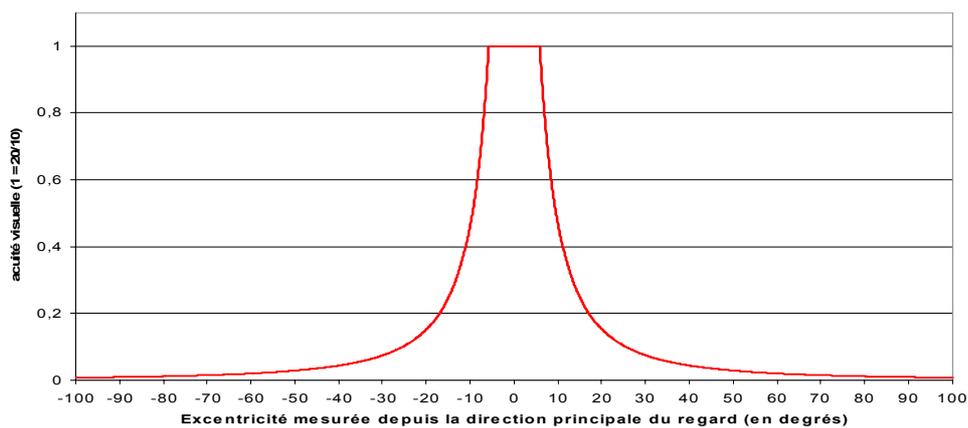


Figure III-7 : Variation de l'acuité visuelle en fonction de l'excentricité d'après le modèle de Reddy [REDDY 97].



Figure III-8 : Illustration de la vision périphérique (source Hérault).

Sere [SERE 00] en collaboration avec Hérault propose lui aussi une carte de l'acuité visuelle en fonction de l'excentricité. Cette carte indique clairement les faiblesses du système visuel humain en ce qui concerne la vision périphérique. La Figure III-8 illustre la théorie qui affirme que l'œil ne perçoit pas les détails en périphérie mais aussi qu'il n'est pas sensible à la couleur. En revanche, la mobilité des objets est un phénomène très rapidement détecté. La perception d'un objet est un phénomène très complexe. Elle est la combinaison de beaucoup de facteurs. Il faut chercher à comprendre son fonctionnement dans le cadre d'applications informatiques.

III.4 - Vision périphérique et applications informatiques.

La vision d'un environnement en 3 dimensions bien qu'il soit virtuel pose plusieurs problèmes. Par exemple, l'utilisateur a tendance à focaliser sur les supports de projection plutôt que sur les objets eux mêmes. En effet, l'accommodation est effectuée sur l'écran de projection. Une

disparité entre accommodation et focalisation provoque une gêne et fatigue oculaire. Ce phénomène est un facteur de gêne et de fatigue pour les utilisateurs.

Pour une application classique fonctionnant sur un écran d'ordinateur, le champ visuel de l'utilisateur n'est pas complètement sollicité (cf. Figure III-9).



Figure III-9 : Champ visuel total / champ visuel sollicité par une application standard.

Champ visuel total : cône rouge

Champ visuel sollicité par l'application : cône bleu.

Image générée avec le logiciel et les sources Virtools.

Notre objectif est de développer des applications devant fonctionner dans une salle immersive. L'utilisateur se trouve au cœur de la scène 3D, il est entouré par les modèles virtuels. Son champ visuel est alors complètement sollicité. Dans ce cas, il semble intéressant d'exploiter les faiblesses de la vision périphérique pour la gestion des niveaux de détails. Deux objets identiques situés à égale distance de l'observateur mais avec une position angulaire différente ne seront pas vus avec la même acuité. Il est possible sans gêner la perception de moins bien définir l'objet le plus éloigné de l'axe de vision principal. La revue de projet d'intérieur de véhicule est un exemple de situation où les modèles sont situés à une distance quasiment constante de l'utilisateur et sont répartis sur un angle de 360° autour de l'utilisateur.

Plusieurs études et développements ont été menés pour mettre en œuvre ce phénomène de décroissance d'acuité visuelle. Funkhauser et Séquin [FUNKHOUSER 93] ont incorporé cette notion dans leur visite virtuelle architecturale en partant du principe que l'utilisateur regardait toujours au centre de l'écran. La distance des objets par rapport au centre de l'écran vient influencer le niveau de résolution des objets. Cette approche est très limitative dans la liberté accordée aux utilisateurs. Hitchner [HITCHNER 93] propose un modèle basé sur un niveau d'intérêt pour l'utilisateur. Ce niveau est calculé par la formule : $\text{Intérêt} = \gamma_{static} / \text{Distance}$, où la distance à la direction du regard de l'utilisateur est mesurée en projection 2D et γ_{static} est un facteur d'échelle. De la même manière, Ohshima et al. [OHSHIMA 96] ont développé un système utilisant un capteur de position de la tête. La décroissance de l'acuité visuelle est calculée par la formule :

$$f(\theta) = \begin{cases} 1 & \text{pour } 0 \leq \theta \leq \alpha \\ \exp\left(-\frac{\theta - \alpha}{c_1}\right) & \text{pour } \alpha < \theta \end{cases} \quad \text{Eq : III-1}$$

Avec θ l'angle entre la direction du regard et le centre de l'objet, α l'angle entre le centre de l'objet et l'arête la plus proche de la direction du regard, c_1 est un facteur d'échelle (les auteurs l'ont fixé à 6.2 degrés). Ces études s'appuient sur l'excentricité pour piloter le choix des niveaux de détails. Cette courbe a été calculée pour des casques de visualisation.

Selon Tory et Moller [TORY 04] seulement 2.8% des auteurs des articles *IEEE Transactions on Visualization and Computer Graphics* ont mené une campagne de tests avec des utilisateurs en situation permettant ainsi de valider les hypothèses émises sur la perception et le fonctionnement du système visuel humain. Pour ces raisons, nous avons souhaité réaliser une série de tests dans la salle immersive afin de définir la courbe de variation de l'acuité visuelle en fonction de l'excentricité.

III.5 - Acuité visuelle et périphériques de visualisation.

Les modèles et les valeurs présentés pour la vision humaine sont définis pour la vision réelle, c'est à dire sans interface entre l'œil et la scène regardée. Dans le cas d'applications informatiques, le périphérique apparaît comme un filtre pour la quantité d'informations parvenant jusqu'à l'œil. Un objet ne sera plus un phénomène continu mais une image discrète (matrice de pixels). Cette discrétisation a pour influence de réduire considérablement l'acuité visuelle. Par exemple, dans le cas de la salle immersive MoVE™, la résolution par écran (3 mètres par 2,4 mètres) est de 1280 pixels par 1024 pixels. Cette résolution impose une taille de pixel de l'ordre de 2,3 millimètres. L'angle sous lequel sont vus ces pixels depuis une distance d'un mètre cinquante est de :

$$\alpha = \text{Arc tan}\left(\frac{2.3}{1500}\right) = 0.09 \text{ deg} = 5.27 \text{ min d'arc}$$

d'où:

$$\text{acuité} = \frac{10}{5.27} = 1.89 \text{ dixième}$$

Figure III-10 montre la réduction de l'acuité visuelle induite par la résolution des projecteurs de la salle immersive.

La visualisation d'une scène en stéréoscopie active induit l'utilisation de lunettes spéciales. L'application génère des images dédiées à chacun des deux yeux. Les projecteurs affichent ces images les unes après les autres à une cadence de 110 hertz. Les lunettes équipées de verres à cristaux liquides s'opacifient à la même fréquence que celle des projecteurs. Ainsi, quand les projecteurs affichent l'image pour l'œil gauche, le verre de l'œil gauche est ouvert et celui de l'œil droit est fermé et réciproquement. Ces lunettes ont souvent des montures très épaisses et très larges (cf. Figure III-11). La raison est de pouvoir isoler le champ visuel de l'environnement, limitant ainsi les nuisances et les perturbations visuelles extérieures à l'application.

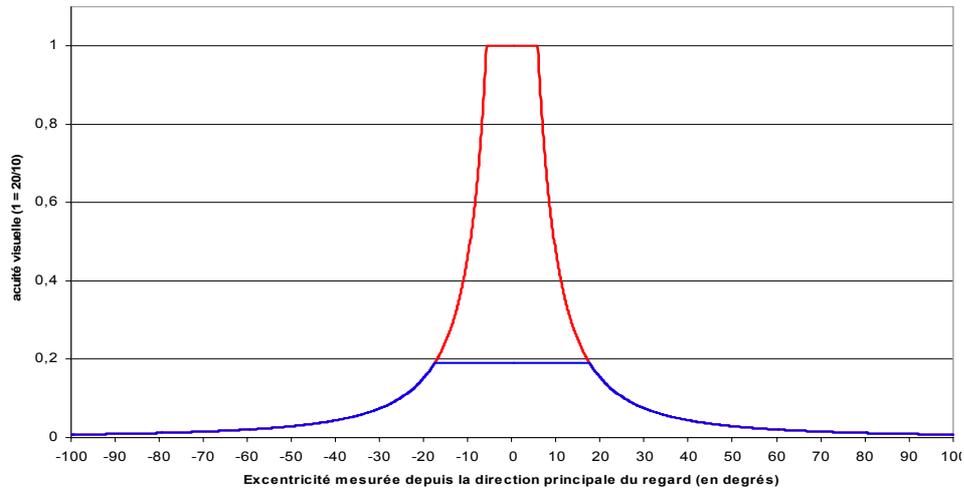


Figure III-10 : Influence de la résolution sur l'acuité visuelle.

Courbe rouge : acuité visuelle normale,
courbe bleue : acuité maximum possible avec une résolution de 1280 par 1024
pour un écran de 3 mètres par 2,4 mètres situé à 1.5 mètre.



Figure III-11 : Lunettes stéréoscopiques (Crystal Eyes de Stereographics).

Le champ visuel est considérablement limité par ces lunettes. Il est de 80° horizontalement et de 55° verticalement réduisant considérablement la couverture du champ visuel (cf. Figure III-12).

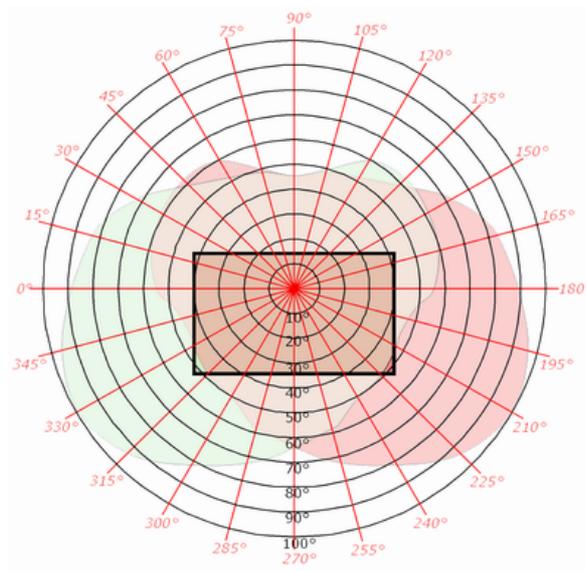


Figure III-12 : Influence des lunettes sur le champ visuel.
La zone couverte par les lunettes Crystal Eyes
se situe à l'intérieur du rectangle noir

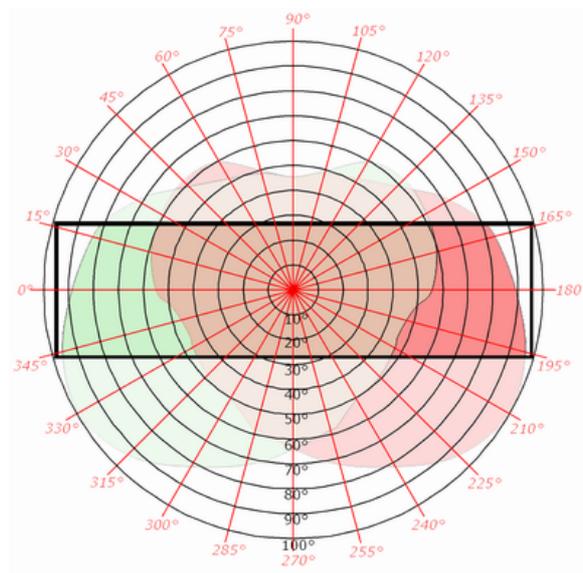


Figure III-13 : Champ visuel des lunettes à vision large.
La zone couverte par les lunettes à vision large
se situe à l'intérieur du rectangle noir

Le champ visuel d'un utilisateur placé dans la salle immersive est entièrement sollicité. Afin de profiter de cette configuration particulière, des lunettes à vision large ont été développées dans le cadre de ces travaux (cf. Figure II-13). Elles donnent accès à un champ visuel de 180° horizontalement et de 50° verticalement. Elles permettent de tester pleinement le champ visuel horizontal.

III.6 - Protocole de tests.

III.6.1 - Contexte.

L'objectif de ces tests est de définir la variation de l'acuité visuelle en fonction de l'excentricité des modèles par rapport à la direction principale du regard. En fonction de cette variation, il sera possible de proposer à l'utilisateur des niveaux de détails plus fortement dégradés pour des objets situés en périphérie du champ visuel. Il faut établir un lien entre les critères de décimation des modèles et leur mise en œuvre c'est à dire leur perception par l'utilisateur. L'intérêt est de proposer à l'utilisateur des modèles suffisamment dégradés pour diminuer le nombre global de polygones à afficher sans pour autant amoindrir sa perception et compréhension de la scène.

Un protocole de test a été réalisé en collaboration avec Mme C. BALLAZ, maître d'enseignement et de recherche au Laboratoire de Psychologie Cognitive de l'Université de Genève. Cette collaboration a permis de définir un protocole de tests conforme aux règles en matière de psychophysique et de cognition.

Les tests ont lieu dans la salle immersive (MoVE™), la position et l'orientation de la tête sont connues à chaque instant. Ces données ne nous renseignent pas sur la direction du regard. Ce qui peut être problématique pour l'étude que nous avons à mener. C'est pourquoi, nous avons ajouté une cible dans le protocole pour obliger l'utilisateur à regarder toujours dans la même direction, dans le cas présent droit devant lui. A cette condition la direction de la tête et celle du regard peuvent être assimilées. Par ailleurs, il a été démontré que lorsqu'une personne souhaite changer de point de vue, ce sont les yeux qui bougent si l'angle entre le point de départ et le point

d'arrivée est inférieur à 25° , au-delà de ces 25° la tête se met en mouvement [AL 67]. Lors de la mise en œuvre des résultats du protocole, il faudra s'assurer qu'il existe une zone d'au moins 25° autour de la direction principale de visualisation afin de s'affranchir de cette mobilité des yeux. Il pourrait être intéressant de mettre en place un protocole de tests prenant en compte les déplacements des yeux. Cette mise en œuvre est complexe pour deux raisons : premièrement, la mobilité des yeux est de l'ordre de 900° par seconde ce qui implique un temps de réponse très faible pour les algorithmes du test et une impossibilité pour les projecteurs actuels de gérer une telle fréquence. Deuxièmement, les utilisateurs de la salle immersive sont déjà équipés de lunettes stéréoscopiques, il faudrait leur ajouter le système de capture de mouvement des yeux sur la tête en plus des lunettes.

Le test mis en œuvre cherche à caractériser l'acuité visuelle que possède un utilisateur lors d'une revue d'intérieur de véhicule. Les modèles entourent l'utilisateur et sont situés à une distance quasiment constante. Le choix du modèle pour le test s'est porté sur un volant. Selon les utilisateurs des premières expérimentations réalisées dans le groupe PSA, le volant est un élément incontournable dans une revue de projet d'intérieur de véhicule. Le moindre défaut constaté sur le volant d'une application de revue d'intérieur de véhicule donne une mauvaise impression quant à la qualité globale de la scène. De plus, le volant est un modèle difficile à décimer de part sa géométrie complexe (selle de cheval, concavité, ...). Le modèle utilisé pour générer les niveaux de détails n'a pas subi le prétraitement défini dans le chapitre précédent. Le polyèdre utilisé a été mis en conformité manuellement. Ce choix permet de différencier les deux étapes de la chaîne et de ne pas avoir d'influences de la première étape sur les conclusions de la seconde.

III.6.2 - Mise en œuvre.

Les tests ont été menés grâce à un panel de trente personnes (23 hommes et 7 femmes), d'âge moyen 29,3 ans (écart type 8.3, voir détails de la population en Annexe 5). Les tests se sont déroulés dans la salle immersive (MoVE™) dont les spécifications techniques sont données en Annexe 2. Le sujet est assis dans un fauteuil au centre de la salle immersive. Un système mécanique maintient la tête de l'utilisateur, il lui est impossible de tourner la tête horizontalement ou verticalement (cf. Figure III-14). Le système de maintien a été conçu et réalisé lors de ces

travaux. Une phase d'initialisation mesure la hauteur des yeux du sujet. Cette hauteur permet de positionner une cible que le sujet devra fixer pendant toute la durée du test. Le sujet est équipé de lunettes stéréoscopiques à vision large (180° d'ouverture horizontale et 50° d'ouverture verticale) développées spécialement dans le cadre de ces travaux. Il tient en main une télécommande équipée de deux boutons pour les réponses à donner.



Vue d'ensemble



Vue de côté



Vue de face

Figure III-14 : Système de maintien en position de la tête des sujets.

Principe :

Dix positions angulaires sont choisies le long des 180° disponibles du champ visuel : cinq positions à gauche de la direction principale du regard et leurs symétriques (cf. Figure III-15). Les modèles sont affichés à une distance d'un mètre cinquante du sujet.

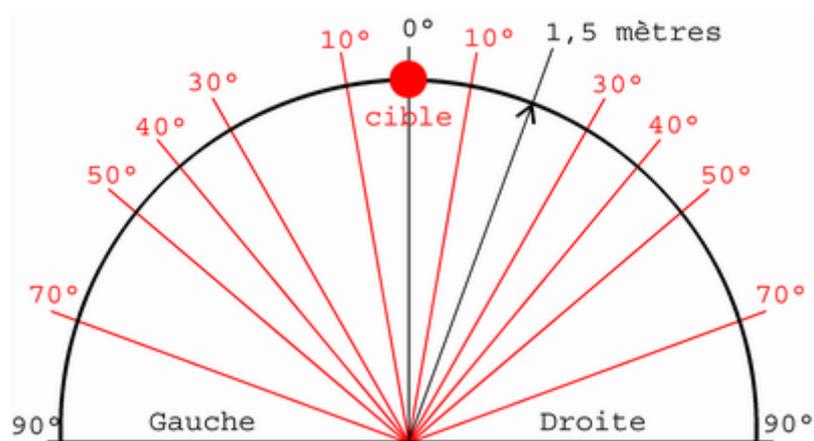


Figure III-15 : Choix des positions angulaires pour le test.
L'angle 0° correspond à la direction du regard et de la tête.

Le modèle étalon et les déclinaisons du volant utilisés pour le test sont présentés sur la Figure III-16.

Les cinq déclinaisons du modèle étalon ne sont pas présentées sur les dix positions. La Figure III-17 donne la liste des modèles présentés en fonction des positions angulaires.

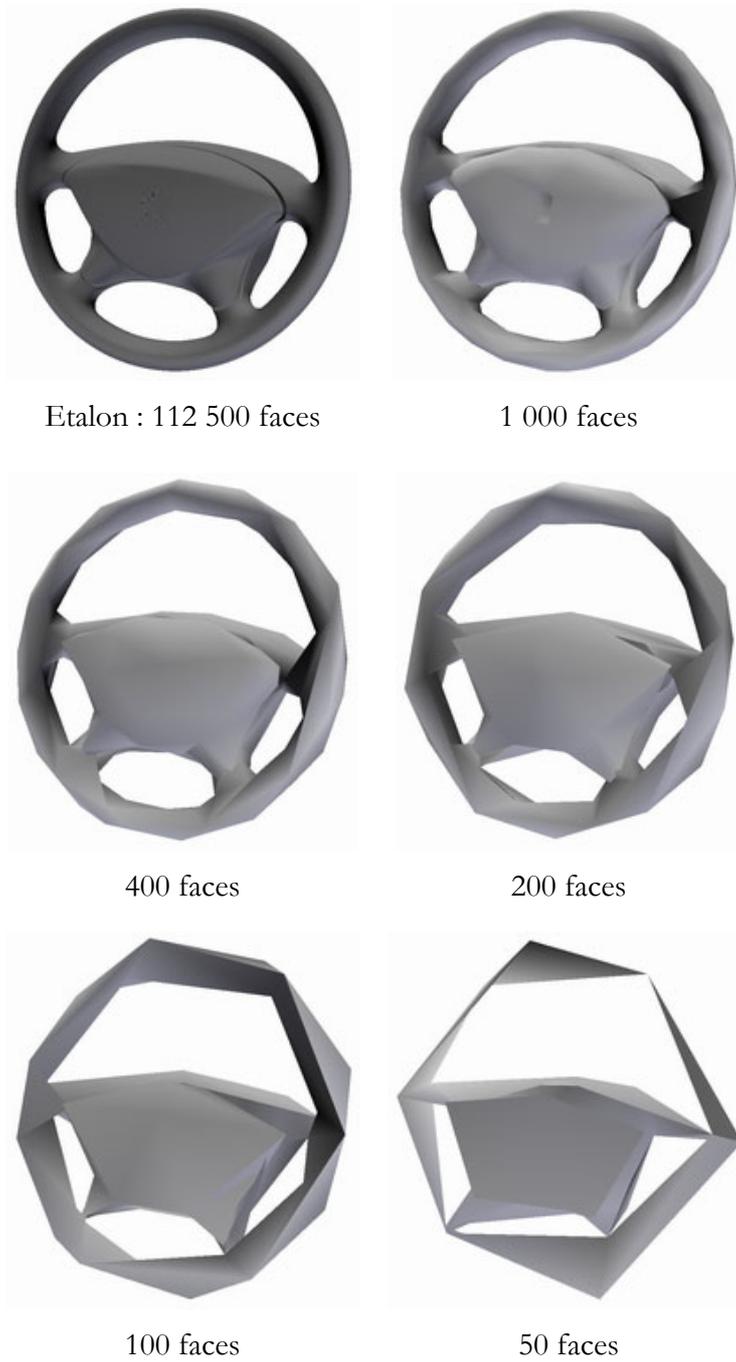


Figure III-16 : Modèles utilisés pour le tests de vision périphérique.

Modèles	Gauche					Droite				
	70°	50°	40°	30°	10°	10°	30°	40°	50°	70°
1000 faces										
400 faces										
200 faces										
100 faces										
50 faces										

Figure III-17 : Répartition des modèles présentés en fonction des excentricités.
Les positions testées correspondent aux cases grisées.

Chacune des versions dégradées du modèle (1000, 400, 200, 100 et 50 faces) est présentée simultanément à la version étalon (112500 faces), l'une au dessus de l'autre. Elles sont réorientées afin d'être visualisées sous le même angle par le sujet. Pour une position donnée, pour un modèle dégradé donné à tester, quatre couples sont définis (cf. Figure III-18; il y a autant de couples présentés contenant un modèle à tester que de couples étalon/étalon).

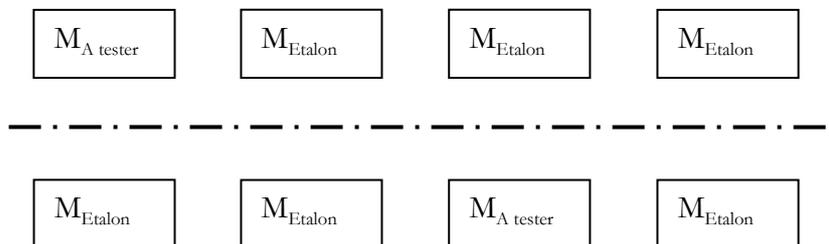


Figure III-18 : Couples de modèles proposés au sujet.
L'axe de symétrie représente la hauteur des yeux du sujet.

Le sujet doit déterminer le plus rapidement possible si les modèles qui lui sont présentés sont identiques ou différents et répondre grâce aux boutons de la télécommande (Bouton de droite pour identique, bouton de gauche pour différent). Il n'y a pas de délai minimal pour répondre, mais pour chaque présentation la réponse est nécessaire à la poursuite du test. Le sujet est soumis à 136 présentations (17 positions par côté * 2 côtés (gauche/droite) * 4 couples par position). L'ordre des couples présentés est généré de manière aléatoire. Les variables indépendantes testées

sont l'excentricité, l'erreur géométrique du modèle à tester, la position et le maintien en position du sujet. Les variables dépendantes sont l'exactitude des réponses du sujet et la durée des réponses du sujet pour chaque question.

Le matériel utilisé pour les tests est une grappe de deux PC BiXéon 3 giga Hz équipés d'une carte graphique NVIDIA FX2000G.

III.7 - Résultats.

Les réponses des trente sujets ont été traitées pour connaître le pourcentage de bonnes réponses. Le nombre de polygones d'un modèle n'est pas une donnée explicite. Il est difficile de connaître la qualité d'un modèle à partir du nombre de polygones. En revanche, la précision d'un modèle, c'est-à-dire l'écart géométrique de ce modèle par rapport à un modèle de référence est une donnée explicite. Pour pouvoir déduire de ces résultats une relation entre l'excentricité et l'écart géométrique, il faut déterminer pour chaque modèle l'écart géométrique avec le modèle étalon. Pour ce faire, l'algorithme *MeshDev* (développé par M. ROY, Le2i) a été utilisé, les résultats sont reportés sur la Figure III-19.

Modèles	Nb polygones	Ecart minimum (en mm)	Ecart maximum (en mm)	Ecart moyen (en mm)	Variance
1	1 000	7,69E-07	1,17086	0,201739	0,0326392
2	400	3,01E-05	1,87454	0,538627	0,116288
3	200	2,30E-05	3,14892	0,578089	0,294947
4	100	2,17187E-04	4,85386	1,17107	0,932494
5	50	3,13086 E-04	15,4244	2,37445	9,20746

Figure III-19 : Variations géométriques des modèles utilisés pour le test.

Le modèle étalon comporte 112 553 faces.

La Figure III-20 liste les pourcentages de bonnes réponses des sujets. Les sujets, lors du test sont en situation de choix forcé puisqu'ils n'avaient que deux réponses possibles ('modèles identiques' ou 'modèles différents') et qu'une réponse leurs était demandée systématiquement pour mettre fin à la présentation et parvenir à la fin du test. Dans cette situation de choix forcé les sujets ne peuvent pas ne pas répondre ou répondre '*je ne sais pas*'. En terme de probabilité, un

sujet qui répond sur la simple base du hasard se situera donc autour de 50% de bonne réponse. Selon Bonnet [BONNET 86], le seuil de perception est atteint dès lors qu'au moins 75% de réponses correctes sont enregistrées.

Nombre de faces	Ecart moyen	Gauche					Droite				
		70°	50°	40°	30°	10°	10°	30°	40°	50°	70°
1000	0,201739					42%	37%				
400	0,538627		43%	52%	48%	67%	60%	45%	43%	43%	
200	0,578089		52%	67%	83%	92%	97%	82%	73%	67%	
100	1,17107	55%	78%	97%	97%			93%	92%	88%	60%
50	2,37445	70%	95%	98%	95%			100%	97%	97%	67%

Figure III-20 : Pourcentage de bonnes réponses en fonction de l'excentricité.

Le calcul du seuil à 75% se fait par le biais d'une régression linéaire sur les résultats de la Figure III-20. Les échantillons sont approximés par une droite d'équation : $y = ax + b$ avec :

$$a = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2} \quad \text{et} \quad b = \frac{\sum x_i y_i - a \sum x_i^2}{\sum x_i} \quad \text{Eq : III-2}$$

L'équation Eq : III-2 permet de calculer les seuils pour un taux de réponses bonnes égal à 75%. Les résultats sont reportés sur la Figure III-21 et sur la Figure III-22.

Seuil	Gauche					Droite				
	70°	50°	40°	30°	10°	10°	30°	40°	50°	70°
75%	2,78	1,43	1,06	1,03	0,50	0,50	1,04	1,13	1,20	3,88

Figure III-21 : Ecart géométrique moyen calculés pour un taux de détection de 75%.

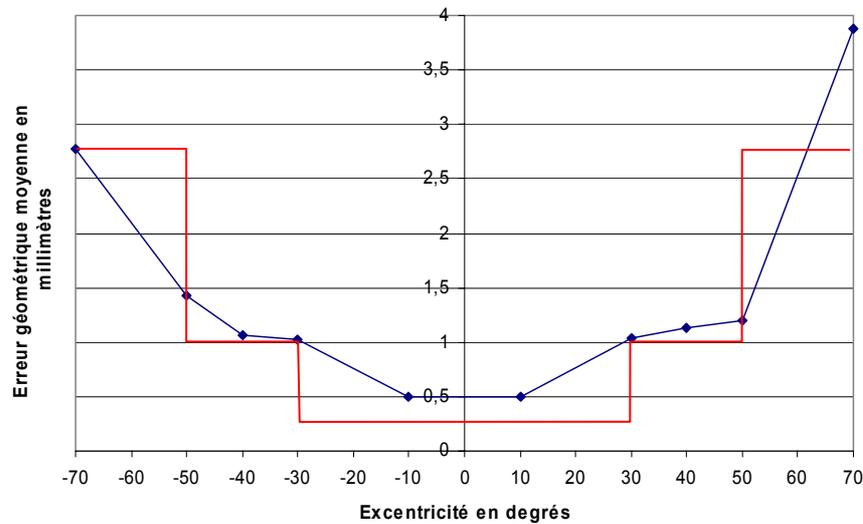


Figure III-22 : Erreur géométrique moyenne en fonction de l'excentricité.

**Courbe bleue : valeurs calculées par régression linéaire,
courbe rouge : segmentation de l'espace.**

La courbe bleue de la Figure III-22 illustre et valide la faiblesse de l'acuité visuelle en périphérie du champ visuel. Il semble intéressant de vouloir dégrader les modèles situés en périphérie du champ visuel. En respectant cette courbe, l'utilisateur percevra la scène dégradée au même titre que la scène non dégradée. La courbe rouge de la Figure III-22 représente une segmentation de l'espace défini à partir de la courbe bleue de cette même figure. Elle prend en compte la mobilité relative des yeux par rapport à la tête. Sachant que les yeux peuvent tourner de 25° avant que la tête ne se mette en mouvement, la première zone englobe ces 25° de part et d'autre de la direction principale. C'est pourquoi la première zone couvre l'espace de -30° à $+30^\circ$. Le seuil choisi correspond à la valeur minimum de l'écart géométrique moyen dans cette zone multiplié par un coefficient de sécurité. La deuxième est établie par le palier présent entre 30° et 50° . La troisième et dernière zone gère le reste de l'espace soit de 50° à 90° . Le choix des niveaux de détails s'effectuera ainsi :

$$\begin{cases} |excentricité| \leq 30^\circ \Rightarrow LOD \text{ dont } l'écart_{\text{moy}} \leq 0.25 \text{ mm} \\ 30^\circ < |excentricité| \leq 50^\circ \Rightarrow LOD \text{ dont } l'écart_{\text{moy}} \leq 1 \text{ mm} \\ 50^\circ < |excentricité| \leq 90^\circ \Rightarrow LOD \text{ dont } l'écart_{\text{moy}} \leq 2.75 \text{ mm} \end{cases}$$

III.8 - Mise en œuvre dans une application de Réalité Virtuelle.

La mise en œuvre de ce mode de gestion a nécessité de nouveaux développements. Elle a été réalisée pour le logiciel Virtools. L'utilisateur placé en immersion dans la salle est équipé d'un capteur de position et d'orientation. Ce capteur permet à tout instant de localiser la tête de l'utilisateur et de connaître son orientation. Le choix du niveau de détail à afficher se fait selon la valeur angulaire calculée entre le vecteur représentant la direction de la tête et le vecteur reliant la tête de l'utilisateur et un objet de la scène (cf. Figure III-23).

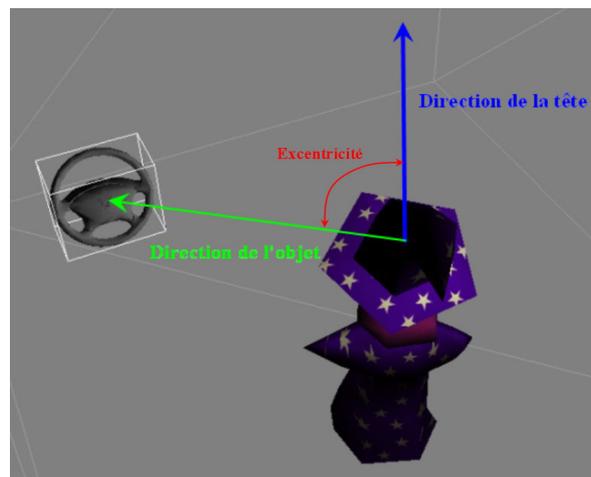


Figure III-23 : Représentation de l'excentricité pour le choix des LODs.

Le point qui sert d'extrémité pour le vecteur de l'objet est pris au centre de la boîte englobante. La prise en compte d'un seul vecteur objet est problématique pour les pièces longues. En effet, elles peuvent couvrir plusieurs zones. Il faut être en mesure de pouvoir détecter toutes les zones et en particulier la zone qui demande le plus de finesse. Afin d'augmenter la précision de mesure,

des vecteurs vers l'objet ont été ajoutés. Il s'agit des vecteurs pointant vers les sommets de la boîte englobante (cf. Figure III-24). Pour chaque vecteur, l'excentricité est mesurée. De toutes ces excentricités, la plus faible, c'est à dire la plus restrictive, est conservée et utilisée pour le choix des LODs.

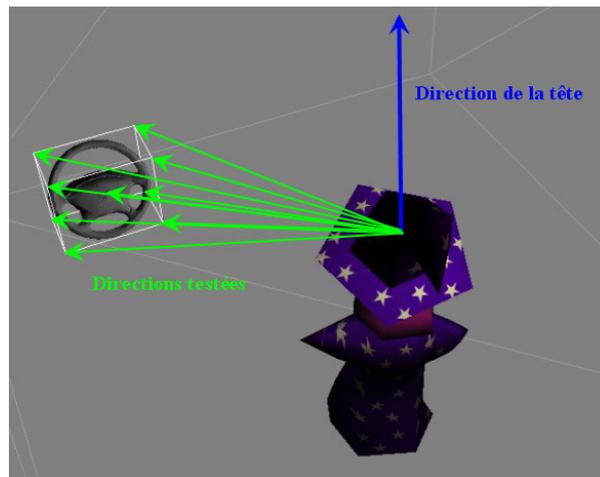


Figure III-24 : Illustration des vecteurs de l'objet utilisés pour calculer l'excentricité.

Des tests ont été effectués avec l'assemblage de la planche de bord présentée précédemment. Le nombre de polygones initial avant décimation est de 400 000 polygones. Les niveaux de détails ont été générés avec les critères définis par les tests de vision. Le nombre de polygones restant est indiqué sur la Figure III-25.

	Nombre de polygones	Ecart géométrique moyen (mm)
Avant décimation	400 000	
LOD_0	100 000	0.25
LOD_1	43 000	1
LOD_2	13 000	2.5

Figure III-25 : Nombre de polygones par niveaux de détails.

L'application affiche un taux de rafraîchissement de 7 images par seconde pour une scène comportant les modèles avant décimation (soit 400 000 polygones) et un taux de rafraîchissement de 30 images par seconde avec les modèles les plus grossiers (soit 13 000 polygones).

Avec la méthode proposée, le nombre de polygones de la scène varie constamment en fonction de la direction de regard de l'utilisateur. Pour mesurer les performances obtenues avec cette gestion des niveaux de détails, l'utilisateur est positionné à la place du conducteur. La Figure III-26 montre deux points de vue différents. Le premier dirigé vers le volant (Figure III-26 a.) offre à l'utilisateur un volant bien défini (44 000 faces) dans sa vision centrale correspondant au LOD₀. Le second dirigé vers l'extrémité droite de la planche de bord (Figure III-26 b.) offre à l'utilisateur un volant très dégradé (2 800 faces).

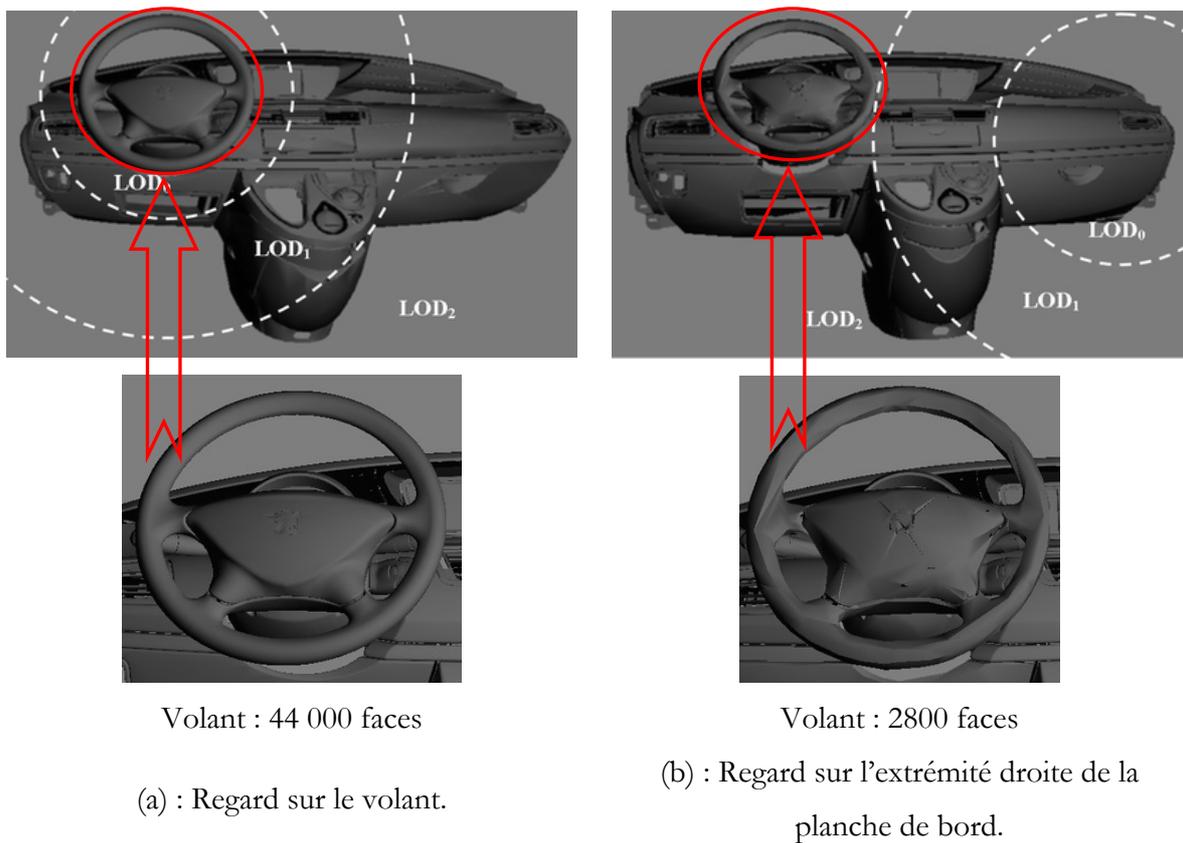


Figure III-26 : Illustration de deux positions privilégiées.

Le taux de rafraîchissement moyen mesuré dans cette application est de 23 images par seconde pour un nombre moyen de polygones de 94 000.

Les résultats obtenus montrent l'apport de cette gestion des niveaux de détails en terme d'augmentation des performances. Les utilisateurs qui ont testé cette nouvelle gestion n'ont pas décelé de différence avec l'application sans la gestion des niveaux de détails (avec 400 000 polygones) hormis l'augmentation du rafraîchissement. Ainsi, la dégradation du modèle du volant dans la position (b) n'a pas été perçue par l'utilisateur. La stratégie sélectionnée pour la décimation des modèles a été prise pour un respect maximum de la géométrie et de la courbe d'acuité visuelle. Il se trouve que lors d'une revue de projet, les utilisateurs sont concentrés sur une tâche précise. Cette concentration a pour effet de diminuer davantage leur perception en périphérie. Dans ce cas, il est possible de dégrader encore plus les modèles.

III.9 - Conclusion.

La visualisation, dernier maillon de la chaîne d'export et de visualisation, peut aussi être soumise aux optimisations. L'étude du système visuel humain met en évidence un certain nombre de faiblesses. Les tests menés dans la salle immersive MoVETM confirment cette étude. Ils ont par ailleurs permis de définir une loi permettant à une application de Réalité Virtuelle de piloter sa gestion de niveaux de détails. Cette gestion permet d'accroître le taux de rafraîchissement de l'application sans diminuer la perception de l'utilisateur. Le protocole de tests défini lors de ces travaux s'est focalisé sur le seul axe horizontal. D'autres paramètres seront à tester : l'axe vertical, l'éloignement par rapport à l'utilisateur, l'influence du contraste, de la luminosité [WATSON 04], des textures, ...

Chapitre IV

Conclusions et perspectives

Les techniques de la Réalité Virtuelle présentent un intérêt certain pour les métiers de l'industrie. Les gains financiers réalisables grâce à ces nouvelles technologies sont tangibles. Au-delà des considérations financières, la créativité et l'innovation sont elles aussi favorisées et accrues. La liberté de création et d'expression offerte par la Réalité Virtuelle ouvre de nouveaux champs exploratoires aux designers. Les solutions qui, auparavant, étaient écartées car trop originales et sans certitude de leur faisabilité peuvent, aujourd'hui, être étudiées et éventuellement développées. Des méthodes et outils doivent accompagner les logiciels et matériels spécialisés en Réalité Virtuelle pour les intégrer dans les process industriels. Ces travaux ont pour objectif de proposer des solutions pour permettre aux designers du groupe PSA Peugeot-Citroën d'utiliser les techniques de la Réalité Virtuelle comme outil de création et de travail.

Après avoir présenté de manière différenciée chacune des parties de la chaîne d'export des modèles surfaciques vers une application fonctionnant en temps réel, ce chapitre analyse les résultats obtenus. Cette chaîne est une chaîne complète et semi-automatique. Elle débute par la

prise en charge des modèles triangulés dans le modèleur originel et s'achève par la visualisation en temps réel des modèles optimisés pour des applications de réalité virtuelle immersives. Cette réalisation rassemble plusieurs thématiques différentes. Cette pluridisciplinarité nous a amenés à faire le choix d'utiliser un certain nombre d'algorithmes existants. En revanche, une technique de détection et de suppression des faces cachées a été développée. Le logiciel prenant en charge le prétraitement permet de gérer toutes les actions à mener sur les polyèdres avant de les envoyer vers l'application finale. Cette unicité de logiciel est suffisamment rare pour être soulignée. Habituellement, plusieurs logiciels sont nécessaires pour cette phase de préparation. Pour visualiser les modèles, la quantité, le type d'informations et la manière dont ces informations sont perçues ont été étudiés. Cette étude a permis de proposer une gestion des niveaux de détails adaptée à la salle immersive.

Bien que la chaîne soit aujourd'hui en mesure de fournir des modèles optimisés pour le temps réel, des optimisations sont à apporter et les perspectives sont nombreuses.

Par exemple, les directions de visualisation peuvent être définies de manière réelle grâce à un système de capture de position. Un utilisateur muni de capteurs dans un environnement représentant l'assemblage final peut valider le nombre et les directions de visualisation adéquates. Ce bouclage permet de ne pas oublier de positions.

IV.1 - Analyse des objectifs.

L'objectif de ces travaux est de proposer des méthodes et outils pour générer des applications de visualisation en environnement virtuel immersif à partir de modèles surfaciques. Ces applications doivent répondre à des critères d'interactivité, de fluidité mais aussi à des critères de qualité des modèles visualisés et de rendu. Ces conditions sont indispensables pour que la Réalité Virtuelle trouve sa place au sein du process de conception.

La Figure IV-1 représente les différentes étapes nécessaires à la création et à la gestion du contenu des applications temps réel. Cette figure définit une méthode dans laquelle sont implémentés des algorithmes. Le choix des algorithmes est expliqué et justifié dans les chapitres

précédents. Toutefois, il est possible de changer les algorithmes dans la chaîne tant que les contraintes imposées par la qualité requise des modèles, par l'automatisation, ... sont respectées.

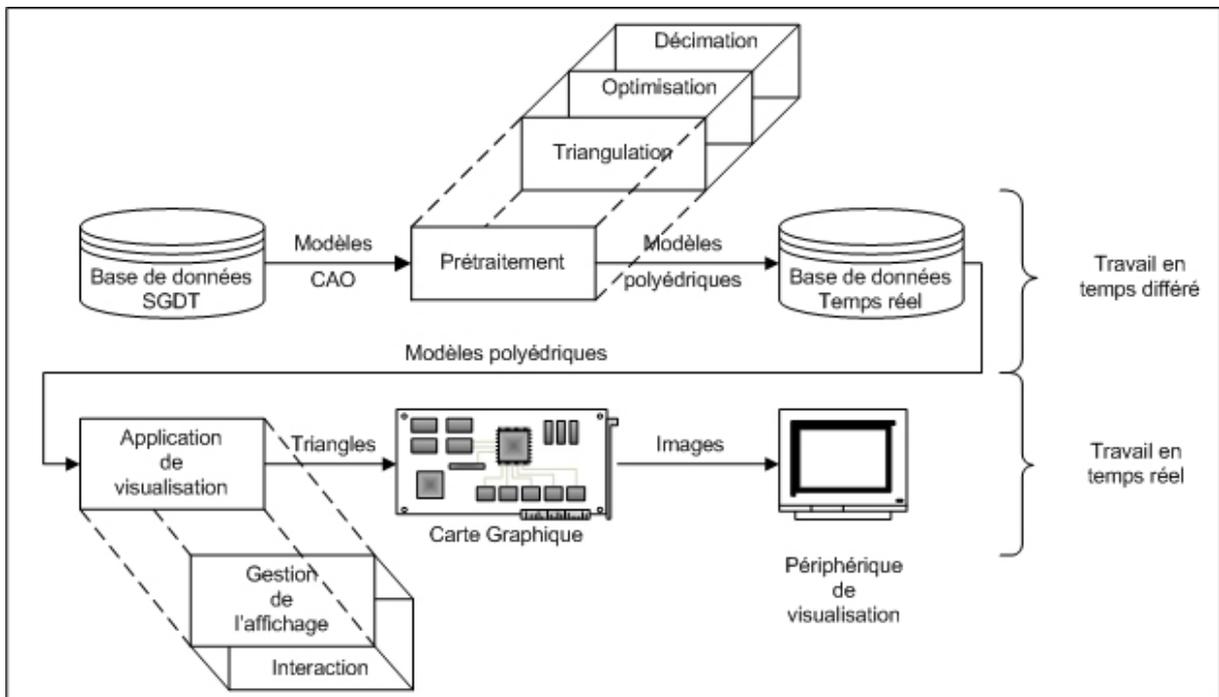


Figure IV-1 : Du modèle surfacique à la visualisation en temps réel.

IV.1.1 - Prétraitement.

Cette étape, la première, est la principale de cette chaîne. Elle est aussi la plus longue et la plus fastidieuse. Les modèles surfaciques, utilisés comme référence, sont conçus indépendamment de leur utilisation pour la Réalité Virtuelle. Les méthodes de conversion des modèles doivent inclure des traitements spécifiques aux origines des modèles. Par exemple, il faut supprimer des informations en 2D (schémas, textes, cotations, ...) contenues dans le format de fichiers d'un modèle 3D, ou encore plusieurs modèles 3D différents peuvent être rassemblés dans un seul fichier, dans ce cas, il est nécessaire de les extraire. Ces traitements spécifiques sont à mettre en œuvre avec la connaissance des métiers concernés par la revue de projet, cela demande une bonne connaissance des métiers impliqués ainsi que de leurs méthodes de travail et outils logiciels utilisés. Puis, les modèles surfaciques sont triangulés par l'intermédiaire des filtres d'export des différents logiciels de création utilisés. Ils sont souvent générés avec beaucoup de polygones

pour minimiser l'erreur créée lors de la discrétisation du modèle. La conséquence directe est le grand nombre de polygones créés. A ce niveau, le résultat est un ensemble de polygones non organisés. C'est pourquoi les fonctionnalités de mise en cohérence sont indispensables pour redonner aux modèles la consistance qu'ils avaient avant la triangulation (modèle volumique fermé). Une première optimisation est ensuite réalisée. Les applications ciblées sont des revues de projet. Les modèles sont visualisés assemblés. Les éléments d'assemblage (pattes de fixation, rivets, ...), les parties mécaniques (raidisseurs, nervures, ...) ne sont jamais visibles. Ces éléments sont souvent géométriquement complexes et de petites dimensions. Ils représentent par conséquent un grand nombre de polygones. L'optimisation consiste à identifier et à supprimer ces polygones inutiles à la visualisation. La méthode développée s'appuie sur l'utilisation des cartes de profondeur (*P-Buffers*). Ces cartes sont habituellement des images (les pixels contiennent des informations de couleur) la méthode mise en œuvre stocke dans les pixels les indices des polygones effectivement visibles depuis un point de vue donné. Grâce à l'ensemble des points de vue, tous les polygones visibles sont identifiés et conservés alors que les autres sont supprimés. Les taux de réduction du nombre de polygones sont très élevés (plus de 60%). Les modèles issus de cette optimisation ne peuvent plus être des modèles fermés. Les opérateurs de détection d'intérieur et d'extérieur de modèles ne peuvent plus s'appliquer. Il faut que les modèles soient orientés avant l'étape de suppression des parties cachées pour que le rendu lors de l'application ne présente pas de discontinuité. Les modèles sont sauvegardés dans une base de données temps réel qui servira aux applications de visualisation. Les différentes étapes mentionnées pour le prétraitement sont toutes automatisables. Lors du premier prétraitement des modèles, les paramètres des différents algorithmes sont sauvegardés et réutilisés pour les nouvelles versions des modèles.

IV.1.2 - Visualisation.

Il se peut que le nombre de polygones contenus dans la maquette virtuelle (i.e. base de données temps réel) soit supérieur aux capacités du matériel pour assurer une gestion de l'application en temps réel. Des techniques existent pour augmenter les performances d'applications de réalité virtuelle contenant un grand nombre de polygones. Les revues de projet concernées (revue d'intérieur de véhicule) dans ces travaux sont prévues pour fonctionner dans

une salle immersive. Ce périphérique confère aux utilisateurs une position particulière. En effet, il est au cœur de la scène 3D, les modèles virtuels l'entourent. Son champ visuel est complètement sollicité ce qui n'est pas le cas avec des périphériques classiques (écran, casques, ...). L'étude du système visuel humain a donné lieu à la rédaction d'un protocole de test en collaboration avec des spécialistes en psychologie cognitive. Ces tests ont pour objectif de cartographier l'acuité du champ visuel humain. Les résultats ont permis d'implémenter une technique de gestion des niveaux de détails adaptée au périphérique de visualisation et au type d'application. L'étude du système visuel humain, la rédaction et la réalisation du test ont requis plusieurs mois de travail. Le protocole de tests rédigé est une base de travail pour d'autres tests afin d'explorer davantage le système visuel humain (sensibilité au mouvement, contraste, couleur, ...). Gérer des niveaux de détail implique de les générer. La carte d'acuité du champ visuel renseigne sur la capacité de l'œil à percevoir ou non les détails géométriques des modèles. Cette acuité a été ramenée à des écarts géométriques qui permettent de décimer les modèles issus du prétraitement pour en faire des niveaux de détails discrets. La génération des niveaux de détails est intégrable dans l'étape de prétraitement.

IV.2 - Analyse critique.

IV.2.1 - Prétraitement.

Ces travaux présentent une chaîne de conversion complète et semi-automatique. Elle a été testée et éprouvée avec des modèles industriels issus des bases de données du groupe PSA Peugeot-Citroën. Bien que des opérateurs de mise en cohérence et de réparation des modèles triangulés soient intégrés à cette chaîne, tous les modèles ne sont pas correctement traités. Certains modèles ont besoin d'être retravaillés manuellement. La raison principale est le manque de maîtrise de la triangulation. Cette étape est réalisée par les logiciels de création des modèles surfaciques. Il est impossible d'intervenir dans cette phase. Les logiciels commerciaux ne sont pas suffisamment ouverts pour permettre de modifier ou même de comprendre leur fonctionnement. Les algorithmes du prétraitement doivent se satisfaire du résultat fourni par ces logiciels. Dans le

cas d'un résultat non satisfaisant le modèle doit être retravaillé manuellement pour le rendre consistant et cohérent. Cette consistance et cette cohérence sont perdues lors de la phase de suppression des faces cachées. Les polygones des modèles ne sont plus attachés les uns aux autres. Des polygones ou des ensembles de polygones sont isolés et désassemblés du corps principal du modèle. La conséquence directe est la difficulté de gestion de ces polygones. Les algorithmes habituellement utilisés pour l'orientation des modèles, la décimation, ... peuvent être mis en défaut. Là aussi, une intervention manuelle sera de rigueur pour appliquer le traitement déficient au modèle. L'algorithme de suppression des parties cachées fonctionne grâce à une répartition de directions de visualisation. Ces directions sont définies actuellement de manière subjective. Il n'y a pas d'optimisation quand au nombre et orientation de ces directions. Or, ce nombre est important pour la durée globale du prétraitement. Dans un premier temps, la durée du prétraitement n'est pas la contrainte première. En effet, le prétraitement est réalisé pendant les périodes de faible activité. Dans un second temps, il est envisageable de réduire les durées des algorithmes pour aboutir à un prétraitement temps réel intégré à l'application de visualisation. Néanmoins, le prétraitement permet d'obtenir des modèles cohérents et optimisés pour alimenter la base de données temps réel.

IV.2.2 - Visualisation.

L'application a été optimisée grâce à des tests de psychologie cognitive portant sur l'étude de l'acuité visuelle le long de l'axe horizontal du champ visuel humain. Cette étude a permis de compléter la méthode globale par une optimisation dépendante de l'acuité visuelle. Les tests pourront se décliner selon l'axe vertical et selon d'autres facteurs intervenant dans la perception (couleur, luminosité, contraste, mobilité des objets, ...). La chaîne et les tests mis en œuvre ne prennent en considération que l'aspect géométrique. La dimension couleur et matière des modèles n'a pas été abordée car les modèles surfaciques fournis par les stylistes ou les ingénieurs ne contiennent pas ces informations. Leur définition intervient très en aval dans le processus de conception.

IV.3 - Perspectives.

L'étape de prétraitement donne des taux de réduction du nombre de polygones très élevés. Ces taux de réduction sont très liés au choix et aux nombres de directions de visualisation. Des travaux du type de ceux de Plemenos [PLEMENOS 96] permettent de définir des trajectoires de caméras dans une scène virtuelle pour offrir le meilleur angle de vue possible à l'utilisateur. Ces techniques sont utiles pour des applications possédant un mouvement de caméra. Dans notre cas, l'utilisateur est libre de regarder où bon lui semble. Cette liberté est quand même limitée par l'application et son contenu. En effet, pour une revue d'intérieur de véhicule l'utilisateur ne cherchera pas à démonter la planche de bord ou à regarder ce qu'il y a derrière. Il peut être intéressant de pouvoir relier la génération des directions de visualisation avec l'application finale. Il est possible de définir plusieurs niveaux d'interactions entre la génération des directions de visualisation et l'application :

- *génération des directions de visualisation à partir d'une application de visualisation allégée* : c'est à dire avec des modèles polyédriques très simplifiés (boîte englobante ou modèle très grossier). Ainsi, l'utilisateur peut se positionner dans la scène et définir lui-même les directions de visualisation qu'il juge utiles et nécessaires.
- *génération des directions de visualisation en temps réel à partir de l'application finale* : ce fonctionnement nécessite que l'application finale soit en mesure de supprimer en temps réel les parties cachées à partir de la direction de la tête définie par le capteur de position. Cette solution va de pair avec une augmentation des performances des machines et des puissances de calcul. Le choix de définir les directions de visualisation en temps réel implique de générer les niveaux de détails aussi en temps réel, ce qui demande davantage de puissance de calcul.

L'idée majeure de ces propositions est de pouvoir lier au maximum le prétraitement à la visualisation. Le système pourra alors fonctionner en boucle fermée (cf. Figure IV-2) avec un contrôle permanent sur la qualité des modèles affichés.

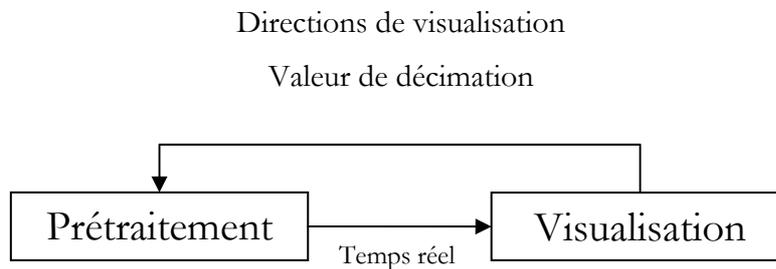


Figure IV-2 : Pilotage du prétraitement par l'application finale.

Ce mode de fonctionnement laisse place à de nouveaux développements tels que le travail collaboratif. C'est à dire proposer en temps réel plusieurs points de vue correctement calculés. Ainsi, deux utilisateurs travaillant sur la même scène possèderaient chacun une version correcte des modèles par rapport à leur propre champ de vision. Actuellement, un seul utilisateur possède cette adéquation champ visuel - niveaux de détails, il s'agit de l'utilisateur équipé du capteur de position.

Les applications de réalité virtuelle doivent être interactives pour favoriser l'immersion et l'appropriation de l'univers 3D. Actuellement, le degré d'interaction consiste à pouvoir naviguer dans la scène, déclencher des interactions sur les éléments de la scène (manipulation des ouvrants, changement de configuration de la sellerie, ...) voire manipuler des objets (déplacement, changement d'échelle, ...). Il est très rare de trouver des applications proposant de modifier la géométrie des modèles temps réel. Il serait intéressant d'implémenter un outil permettant aux stylistes d'intervenir sur la géométrie des modèles. Il serait alors envisageable de modifier en temps réel et dans l'environnement 3D les flancs de carrosserie, le galbe d'une portière, ... Cette liberté d'interaction sur les modèles temps réel serait d'autant plus pertinente si les modifications faites sur le modèle temps réel pourraient être automatiquement répercutées sur le modèle surfacique. Dans l'état actuel de la chaîne de traitement et de l'export des modèles surfaciques, cette notion n'est pas envisageable car lors de la triangulation, il y a perte d'informations sur les surfaces originelles. En effet, il est actuellement impossible de reconstruire les carreaux

surfaciques à partir des triangles du polyèdre. La résolution de ce verrou technologique offre des perspectives intéressantes à ce titre.

Pour aller plus loin dans la relation modèle surfacique - modèle temps réel, il serait utile de pouvoir récupérer des informations sur les liaisons cinématiques des modèles entre eux. Cette connaissance permettrait de pouvoir générer automatiquement les animations et déplacements relatifs des modèles.

Le rendu des scènes virtuelles en temps réel est un point faible de la pertinence des applications. En effet, les stylistes aimeraient voir des rendus de très haute qualité. Là encore, l'augmentation des performances des machines devraient aller dans le sens de l'amélioration des rendus. Cependant, il existe des techniques permettant d'augmenter la sensation de réalisme sans pour autant surcharger les calculateurs. Les cartes des normales permettent de donner une impression de relief sur des polyèdres ne possédant pas ce relief. Par exemple, l'aspect des plastiques des planches de bords pourrait être rendu de cette manière. De même, des boutons de petites tailles pourraient être remplacés par de simples polygones texturés avec une carte des normales. Cette perspective est d'autant plus intéressante qu'il est possible de générer les cartes des normales lors des phases de décimation. Cette intégration est susceptible de se faire rapidement. De même, quand les textures et matériaux seront définis, des solutions existent pour conserver ces attributs lors des phases de décimation. Par ailleurs, les interfaces de Réalité Virtuelle (visualisation, interaction) ne sont pas encore suffisamment évoluées pour permettre une immersion virtuelle de très haute qualité, c'est à dire offrant des sensations comparables au réel. Les évolutions technologiques en la matière sont attendues.

L'optimisation de l'immersion virtuelle nécessite une adéquation soignée entre la perception de l'utilisateur, l'application visée et le dispositif de rendu utilisé (visualisation, calcul, interaction). La problématique technologique résultante fait intervenir un large panel de compétences scientifiques qu'il est nécessaire d'harmoniser.

Les résultats prometteurs permettent d'envisager une intégration rapide des techniques de la Réalité Virtuelle dans les métiers de la conception de produits manufacturés.

Bibliographie

[AL 67] : Y. AL, 1967. In: (ed.). *Eye movements and Vision*. New York: Plenum.

[ALGORRI 96] : M. E. ALGORRI et F. SCHMITT, *Mesh Simplification*, Proceedings of Eurographics'96, Poitiers, France, Vol. 15, pp 77-86, 1996.

[AMENTA 98] : N. AMENTA, M. BERN, M. KAMVYSSELIS, *A New Voronoi-Based Surface Reconstruction Algorithm*,. Proceedings of SIGGRAPH 98. pp. 415-422, 1998.

[ARTHUR 00] : K.W. ARTHUR, *Effects of Field Of View on Performance with Head-Mounted Displays*, Thesis Dissertation, University of North Carolina at Chapel Hill, department of Computer Science, 2000.

[ATTALI 94] : D. ATTALI, P. BERTOLINO et A. MONTANVERT, *Using polyballs to approximate shape and skeletons*, 12th ICPR, pp. 626-628, Octobre 1994.

[BEZIER 87] : P. BEZIER, *Courbes et surfaces*, Mathématiques et CAO, Volume 4, Edition Hermès, 1987.

[BOISSONNAT 84] : J.-D. BOISSONNAT, *Geometric structures for three-dimensional shape representation*, ACM Trans. Graph., 3 (4), pp. 266-286, 1984.

[BOISSONNAT 92] : J.-D. BOISSONNAT, B. GEIGER, *Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation*, Report 1697, INRIA Sophia-Antipolis, Valbonne, France, 1992.

[BOIVIN 01] : S. BOIVIN, *Simulation photo réaliste de scènes d'intérieur à partir d'images réelles*, thèse de doctorat, Ecole Polytechnique, janvier 2001.

[BONNET 86] : C. BONNET, *Manuel pratique de psychophysique*, Editions Armand Colin Paris, 1986.

[BUTLIN 96] : G. BUTLIN and C. STOPS, *CAD Data Repair, 5th International Meshing Roundtable*, Sandia National Laboratories, pp.7-12, October 1996.

[BURDEA 96] : G. BURDEA et P. COIFFET, *La réalité virtuelle*, 1993, Hermès.

[BRUNETTI 00] : G. BRUNETTI, P. BENOLKEN, J. RIX, *The Role of Product Data Technology in the Integration of PDM/CAD with Virtual Reality*, SMART Engineering, proceedings : STEP, Methods, Applications, Research & New Technologies, pp. 169-180, 2000.

[CADOZ 94] : C. CADOZ, Dominos, *Les réalités virtuelles*, 1994, Flammarion.

[CATMULL 74] : E. CATMULL, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, PhD thesis, Department of Computer Science, University of Utah, December 1974.

[CHEVALDONNE 03] : M. CHEVALDONNE, M. NEVEU, F. MERIENNE, N. CHEVASSUS and F. GUILLAUME, *Human Vision, application, and display considerations to determine the characteristics of images for a Virtual Reality application*, Virtual Concept 2003, Biarritz, pp. 372-379.

[CLARK 76] : J.H. CLARK, *Geometric Models for Visible Surface Algorithm*, Communication of the ACM. Vol. 19 (10), pp 547-554, 1976.

[Computer Science93] : Actes du colloque : Computer Sciences, Communication and Society : a technical and Cultural, septembre 1993. Challenge *La virtualité de la réalité*. Société suisse des Informaticiens & Société suisse de sociologie Neuchâtel.

[DE FLORIANI 96] : L. DE FLORIANI, O. MARZANO et E. PUPPO, *Multiresolution models for topographic surface description*, The Visual Computer, Vol. 14, N°4, pp 317-345, 1996.

[DE HAEMER 91] : M. J. DE HAEMER Jr et M. J. ZYDA, *Simplification of objects rendered by polygonal approximations*, Computers & Graphics, Vol. 15, N°2, pp 175-184, 1991.

[ECK 95] : M. ECK, T. DE ROSE, T. DUCHAMP, H HOPPE, M. LOUNSBERY et W. STUETZLE, *Multiresolution analysis of arbitrary meshes*, Computer Graphics Proceedings, Annual Conferences Series, SIGGRAPH'95, pp 173-182, 1995.

[EDELSBRUNNER 92] : H. EDELSBRUNNER, E. P. MÜCKE, *Three-dimensional alpha shapes*, Workshop on Volume Visualization, pp. 75-82, 1992.

[FARIN 00] : G. FARIN, D. HANSFORD, *The Essentials of CAGD*, published by A.K. Peters Ltd, 248 p., 2000.

[FINE 01] : L. FINE, *Processus et méthodes d'adaptation et d'idéalisation de modèles dédiés à l'analyse de structures mécaniques*, Doctorat de Mécanique de l'I.N.P.G., Grenoble, 2001.

[FREJUS 97] : M. FREJUS et A. DROUIN, rapport, *Réalité virtuelle et processus cognitifs : état de l'art et perspectives en formation*, EDF.

[FUCHS 80] : H. FUCHS, Z. KEDEM and B.F. NAYLOR, *On Visible Surface Generation by a Priori Tree Structures*, Computer Graphics Vol. 14(3), pp. 124-133, June 1980.

[FUCHS 03] : P. FUCHS, G. MOREAU, *Le traité de la réalité virtuelle, Volume 1 : Fondements et interfaces comportementales*, édition Presse des Mines, 2^{ème} édition, 2003.

[FUNKHOUSER 93] : T. A. FUNKHOUSER and C. H. SEQUIN, *Adaptive Display Algorithm for Interactive Frame Rate during Visualization of Complex Virtual Environments*, Proceedings of SIGGRAPH'93, pp. 247-254, 1993.

[GARLAND 97] : M. GARLAND and P. HECKBERT, *Surface Simplification Using Quadric Error Metrics*, SIGGRAPH'92

[GRAF 02] : H. GRAF, G. BRUNETTI, A. STORK; *CAD2VR or How to Efficiently Integrate VR into the Product Development Process*, CAD 2002, pp. 249-258, Dresden, Germany, 2002.

[GUEZIEC 96] : A. GUEZIEC, *Surface simplification inside a tolerance volume*, Rapport Technique RC20440, IBM Research Division, T J Watson Research Center, New York, USA, mai 96.

[HAMMANN 94] : B. HAMMANN, *A data reduction scheme for triangulated surface*, Computer Aided Geometric Design, Vol. 11, N°2, pp 197-214, 1994.

[HE 96] : T. HE, L. HONG, A. VARSHNEY et S. WANG, *Controlled topology simplification*, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, N°2, 1996.

[HECKBERT 92] : P. S. HECKBERT, *Introduction to global illumination*, SIGGRAPH'92, Global illumination course, Chicago, 1992.

[HINKER 93] : P. HINKER et C. HANSEN, *Geometric optimization*, IEEE Visualization'93 Proceedings, pp 189-195, 1993.

[HITCHNER 93] : L. E. HITCHNER and M. W. MCGREEVY, *Method for User-Based Reduction of Model Complexity for Virtual Planetary Exploration*, Proceedings of SPIE, The International Society for Optical Engineering, Vol. 1913, pp. 622-636, 1993.

[HOPPE 96] : H. HOPPE, *Progressive meshes*, Computer Graphics proceedings, SIGGRAPH'96, New Orleans, pp 99-108, 1996.

[HOPPE 97] : H. HOPPE, *View-depend refinement of progressive meshes*, Computer Graphics proceedings, SIGGRAPH'97, Los Angeles, pp 189-198, 1997.

[HURIET 97-98] : C. HURIET, sénateur, 1997-1998, Rapport d'informations n°169, *Images de synthèse et monde virtuel : technique et enjeux de sociétés*.

[LEON 91] : J.-C. LEON, *Modélisation et construction de surfaces pour la CFAO*, Editions Hermès, 1991.

[LUEBKE 97] : D. LUEBKE and C. ERIKSON, *View-dependent Simplification of Arbitrary Polygonal Environments*, Proceedings of SIGGRAPH'97, Los Angeles, pp 199-208, 1997.

[LUEBKE 01] : D. LUEBKE and B. HALLEN, *Perceptually Driven Simplification for Interactive Rendering*, Proceedings of 2001 Eurographics Rendering Workshop, pp 223-234, 2001.

[LUEBKE 03] : D. LUEBKE, M. REDDY, J.D. COHEN, A. VARSHNEY, B. WATSON and R. HUEBNER, *Level of Detail for 3D Graphics*, Edition M. KAUFMAN, 2003.

[NAYLOR] : B. F. NAYLOR, *A Tutorial on Binary Space Partitioning Tree*’, Spatial Labs Inc.

[OHSHIMA 96] : T. H. OHSHIMA, H. YAMAMOTO and H. TAMURA, *Gaze-Directed Adaptive Rendering for Interacting with Virtual Space*, Proceedings of IEEE Virtual Reality Annual International Symposium, pp. 103-110, 1996.

[OSTERBERG 35] : G. OSTERBERG, *Topography of the layer of rods and cones in the human retina*, Acta Ophthalmologica, 6 (Suppl.), 1935.

[PLEMENOS 96] : D. PLEMENOS and M. BENAYADA, *Intelligent display in scene modeling. New techniques to automatically compute good views*, GraphiCon’96, Saint Petersburg, July 1996.

[PUPPO 97] : E. PUPPO et R. SCOPIGNO, *Simplification, LOD and Multiresolution principles and Applications*, Eurographics'97, Budapest, septembre 1997.

[Priméca 01] : Colloque Priméca 2001, avril 2001, *Place de la réalité virtuelle dans la conception*, La Plagne.

[REDDY 97] : M. REDDY, *Perceptually Modulated Level Of Detail for Virtual Environments*, Ph.D. Thesis, University of Edinburgh, Scotland, 1997.

[REQUICHA 80] : A.G. REQUICHA, *Representations for Rigid Solids: Theory, Methods, and Systems*, ACM Computing Surveys (CSUR), Vol. 12, N° 4, pp 437-464, December 1980.

[RHEINGOLD 93] : H. RHEINGOLD, *La réalité virtuelle, Quand l'illusion a toutes les apparences de la réalité*, Dunod, 1993.

[ROSSIGNAC 93] : J. ROSSIGNAC et P. BORREL, *Multiresolution 3D approximations for rendering of complex scenes*, Falcidieno, B. et KUNII, T L editors, Geometric Modeling in Computer Graphic, Springer-Verlag, Gènes, Italie, pp 455-466, 1993.

[ROY 03] : M. ROY, S. FOUFOU and F. TRUCHETET, *Mesh Comparison using attribute deviation metric*, International Journal of Image and Graphics, Vol. 4, N°1 (2004), pp. 127-140, 2002.

[SCHAUFLER 95] : G. SCHAUFLER et W. STUERZLINGER, *Generating multiple levels of detail for polygonal geometry model*, Virtual Environments, Monte Carlo, pp 53-62, février 1996.

[SCHROEDER 92] : W. J. SCHROEDER, J.A. ZARGE et W.E. LORENSEN, *Decimation of triangle meshes*, Computer Graphics proceedings SIGGRAPH'92, Chicago, pp 65-70, 1992.

[SCHUMACKER 69] : R. A. SCHUMAKER, R. BRAND, M. GILLILAND and W. SHARP, *Study for Applying Computer - Generated Images to Visual Simulation*, AFHRL-TR-69-14, U.S. Air Force Human Resources Laboratory (1969).

[SERE 00] : B. SERE, C. MARENDAZ, J. HERAULT, *Nonhomogeneous resolution of images of natural scenes*, Perception, 2000, volume 29, pp 1403 - 1412.

[SUNG 92] : K. SUNG, *Are sampling buffer : Tracing rays with α -buffer hardware*, Computer Graphics Forum 11 (3) : 229-310, September 1992.

[SUTHERLAND 73] : I. E. SUTHERLAND, *Polygon Sorting by Subdivision: a Solution to the Hidden-Surface Problem*, unpublished manuscript, October 1973.

[TORRES 90] : E. TORRES, *Optimization of the Binary Space Partition Algorithm (BSP) for the Visualization of Dynamic Scenes.*, Eurographics '90, pp. 507-518, 1990.

[TORY 04] : M. TORY and T. MOLLER, *Human Factors in Visualization Research*, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, N)1, pp 72-84, 2004.

[TURK 92] : G. TURK, *Re-tiling polygonal surfaces*, Computer Graphics proceedings, SIGGRAPH'92, Chicago, pp 55-64, juillet 1992.

[VARSHNEY 94] : A. VARSHNEY, *Hierarchical geometric approximations*, Thèse de doctorat, Department of Computer Science, University of North Carolina, NC 27599-3175, Chapel Hill, 1994.

[VERON 97] : P. VERON, *Techniques de simplification de modèles polyédriques pour un environnement de conception mécanique*, Doctorat de Mécanique de l'I.N.P.G., Grenoble, 1997.

[WATSON 04] : B. WATSON, N. WALKER and L.F. HODGES, *Supra-Threshold Control of Peripheral LOD*, In Proceedings of SIGGRAPH 2004, Los Angeles, CA, 2004.

[WEGHORST 84] : H. WEGHORST, G. HOPPER and D. P. GREENBERG, *Improved computational methods for ray tracing*, ASM Transactions on Graphics, 3 (1), 52-69, January 1984.

[WHYTE 00] : J. WHYTE, N. BOUCLAGHEM, A. THORPE, R. McCAFFER, *From CAD to virtual reality: modelling approaches, data exchange and interactive 3D building design tools*, Automation in Construction, Volume 10, Issue 1, pp. 43-55, November 2000.

[WOO 04] : M. WOO, J. NEIDER, T. DAVIS et D. SHREINER, *OpenGL 1.4*, collection référence, édition Campus Press.

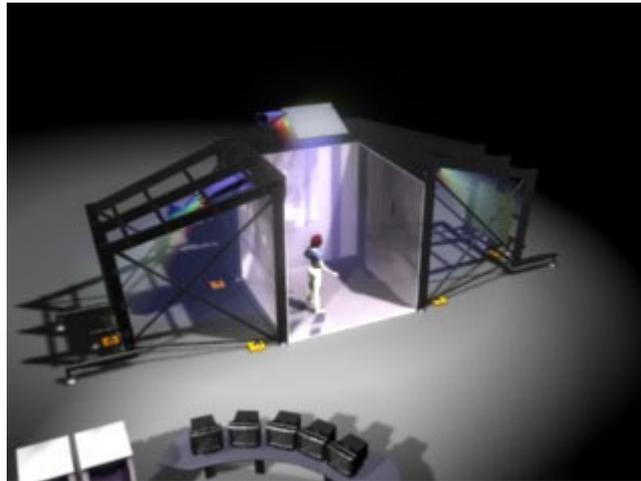
Annexes

Annexe 1 - Le Sensorama, Morton HEILIG 1962.



"... C'est donc à la base un théâtre à une place. On s'assied sur un siège, on glisse la tête dans l'habitacle et on regarde par deux optiques qui restituent une vision périphérique. Le siège est équipé d'un système vibratoire, la plate forme dessous est montée sur caoutchouc, un ventilateur, un dispositif pour les odeurs... le tout synchronisé avec l'image. Le tout est mécanique, les ordinateurs n'existaient quasiment pas quand je l'ai conçu... J'avais quatre films différents : une scène de séduction avec un danseur, une ballade en voiture, un voyage en hélico et une course à moto à travers New York - le siège vibrait avec la moto, on sentait les odeurs, le vent sur le visage... Central Park avec l'odeur d'herbe fraîchement coupée, le Brooklyn Bridge, Times square et ses pizzas, la 5^{ème} avenue, on avait la sensation physique d'être sur la moto et de pencher dans les virages... mais quelques éléments manquaient. D'abord l'effet environnemental était limité à 100°, j'aurais souhaité 180° mais c'était trop cher. Il n'y avait pas de variation de température, et je n'ai pas non plus inclus l'effet d'inclinaison du corps quand on prend un virage, j'avais peur qu'en public ce soit dangereux. Les gens m'appellent le père de la VR, comme Myron Kruger, mais je suis plutôt le grand-père. Pour être tout à fait franc, mon travail n'intègre pas la capacité du spectateur à contrôler l'environnement, à interagir avec lui; ..."

Annexe 2 - Spécifications techniques : MoVE.



Le ordinateur est une machine Silicon Graphics ONYX2 équipée de 8 processeurs R12000 cadencés à 300MHz, de 8 Go SDRAM ECC et de 4 *pipes* graphiques Infinite Reality 3 avec 2 Raster Manager par *pipe* et 256 Mo de texture pas Raster Manager. 12 millions de polygones (T-mesh, non éclairés, sans ombrage de Gouraud) peuvent être gérés par seconde. La bande passante vidéo est de 277 Mpixels/*pipe*. La performance moyenne est de 150-200 Mpixels/sec pour des pixels texturés, anti-crênelés et ombrés. Les différentes résolutions sont indiquées dans le tableau ci-dessous :

	Résolution	Fréquence	Taille pixel (mm)
Basse	1136x1024	110Hz	2.64
Haute	1320x1188	110Hz	2.27
Possibilité	1424x1282	96Hz	2.11

Les projecteurs sont des projecteurs CRT BARCOREALITY 909. Ils délivrent une puissance lumineuse de 1250 lumens à 10% de blanc et de 250 lumens ANSI en mode normal. La résolution maximum est de 3200x2560 pixels à 60Hz et la résolution courante est de 1600x1200 à 120Hz.

Les lunettes stéréoscopiques utilisées sont des lunettes CRYSTAL-EYES de Stereographics avec émetteur ELR. Les lunettes sont à base de cristaux liquides. Les émetteurs ELR sont placés de manière à assurer une bonne réception dans tout l'espace de la salle quelque soit l'orientation du ou des visiteurs :

- 4 émetteurs fixés sur les projecteurs et renvoyant son signal IR par les miroirs,
- 2 émetteurs à mi-hauteur aux deux piliers extérieurs orientés à 45 degrés vers l'intérieur.

Spécifications techniques des lunettes actives CRYSTAL-EYES :

- Transmittance : 32 %,
- Domaine dynamique : 1500 :1,
- Balayage : 80 à 160 Hz,
- Poids : 93 g.

Spécifications techniques des émetteurs longues portées ELR :

Portée : 6,60 m en haute émission/ 3,30 m en basse émission (standard)

Annexe 3 – Algorithme de détection des faces cachées.

```
Pour chaque direction de visualisation
  Pour chaque polyèdreI
    Calcul position caméra
    Pour chaque polyèdreJ
      Si polyèdreJ = polyèdreI
        Pour chaque face du polyèdreJ
          Couleur = f(Index_face)
          Dessin face
        Fin
      Fin
      Si polyèdreJ != polyèdreI
        Pour chaque face du polyèdreJ
          Couleur = noir
          Dessin face
        Fin
      Fin
    Fin
  Fin
  Pour chaque pixel
    Index = f(couleur)
    Si Index = noir
      Continue
    Fin
    Drapeau_face_vue(Index) = Vrai
  Fin
Fin
Pour chaque polyèdre
  Pour chaque face du polyèdre
    Si Drapeau_face_vue(Index) = Faux
      Suppression de la face
    Fin
  Fin
Fin
```

Annexe 4 - Suppression des faces cachées : DeltaGen vs Simpoly.

Indice des modèles	Nom des modèles	Modèles oroginaux	DeltaGen	Simpoly	DeltaGen / Simpoly
1	SUPPORT_RETRO_EXT_D.off	557	35	348	30
2	RET_G_ELE_RAB_DEGDAG_7000.off	6431	2899	1009	1214
3	RET_G_ELE_RAB_DEGDAG_6000.off	6320	2373	1730	1706
4	RET_G_ELE_RAB_DEGDAG_5000.off	8730	8730	5033	5040
5	RET_G_ELE_RAB_DEGDAG_4000.off	12109	3614	257	1040
6	RET_G_ELE_RAB_DEGDAG_3000.off	7847	4136	3092	2795
7	RET_G_ELE_RAB_DEGDAG_2000.off	3960	3911	976	1457
8	RET_G_ELE_RAB_DEGDAG_1000.off	11563	6137	4745	4157
9	RET_G_ELE_RAB_DEGDAG_0000.off	17012	7708	5664	4907
10	RENFO_CHOC_AVANT_PORT_D.off	396	110	302	103
11	PROTECTEUR_PORTE_D.off	6304	2418	1431	1171
12	PORTE_D.off	9038	5791	2970	2783
13	PANNEAU_PORTE_D_GARNI_9005.off	3916	1061	710	690
14	PANNEAU_PORTE_D_GARNI_7005.off	3319	964	867	779
15	PANNEAU_PORTE_D_GARNI_5005.off	49826	9283	5775	5527
16	PANNEAU_PORTE_D_GARNI_17005.off	43925	20830	19671	14272
17	PANNEAU_PORTE_D_GARNI_16005.off	15595	8009	5027	4771
18	PANNEAU_PORTE_D_GARNI_14005.off	21294	13689	7036	6805
19	PANNEAU_PORTE_D_GARNI_13005.off	16720	5027	3366	2956
20	PANNEAU_PORTE_D_GARNI_11005.off	16381	3237	478	440
21	PANNEAU_PORTE_D_GARNI_1005.off	51401	27291	14414	14138
22	MONT_VITR_FIXE_POR_D.off	7812	4019	2809	2141
23	LECH_EXT_VITR_PORT_D.off	6705	2683	1888	1488
24	FERMET_DOUBL_PORTE_D.off	3863	852	395	363
25	ENJOL_AR_ENCA_PORT_D.off	3060	2863	1513	1521
26	ENCADREMENT_PORTE_D.off	9963	9806	3017	2975
27	DOUBLURE_PORTE_D.off	83157	82681	28515	29498
28	COMM_INT_OUVE_PORT_D.off	27037	16572	8233	7371
29	COM_EXT_OUV_POR_DENS.off	10564	4716	6274	3923
	TOTAL	464805	261445	137545	126061

Annexe 5 – Détail des sujets présents lors du test de vision périphérique.

	Age	Sexe	Profession	Problèmes ophtalmologiques		Correction
				Gauche	Droite	
S1	27	M	Doctorant	Néant	Néant	Sans
S2	26	M	Post-Doctorat	Myope	Myope	Avec
S3	24	M	Etudiant IG	Myope	Myope	Avec
S4	26	F	Intermittent du spectacle	Néant	Néant	Sans
S5	25	M	Etudiant IG	Pb de rapidité de focalisation	Pb de rapidité de focalisation	Sans
S6	21	M	Informaticien	Néant	Néant	Sans
S7	26	M	Etudiant IG	Néant	Néant	Sans
S8	46	M	Ingénieur Informatique	Néant	Néant	Sans
S9	31	M	Ingénieur projet RV	Myope -2,25	Myope -1,5	Avec
S10	55	F	Service sociaux	Myope Astigmat	Myope Astigmat	Sans
S11	25	M	Etudiant IG	Myope	Myope et astigmat	Sans
S12	38	M	Enseignant	Néant	Néant	Sans
S13	27	M	Ingénieur Méthodes	Néant	Néant	Sans
S14	24	M	Etudiant	Néant	Néant	Sans
S15	26	M	Ingénieur IG	Néant	Néant	Sans
S16	25	F	Doctorant	Néant	Néant	Sans
S17	24	M	Analyste Programmeur	Néant	Néant	Sans
S18	27	M	Doctorant	Néant	Néant	Sans
S19	30	M	Ingénieur IG	Néant	Néant	Sans
S22	17	M	Etudiant	Néant	Néant	Sans
S20	19	F	Lycéenne	Myope	Myope, pas de mobilité latérale	Avec
S21	46	M	Professeur des Universités	Myope	Myope	Avec
S23	34	F	Infirmière	Myope	Myope	Avec
S24	27	F	Employée de banque	Myope -3,5	Myope -3,5	Avec
S25	35	M	Ingénieur Informatique	Néant	Néant	Sans
S26	24	M	Etudiant IG	Myope -0,75	Myope -1	Sans
S27	33	M	Conducteur d'engins	Néant	Néant	Sans
S28	27	F	Secrétaire	cristallin artificiel pas d'iris	Néant	Sans
S29	26	M	Doctorant	Myope -2	Myope -2	Avec
S30	37	M	Chef de Projet Infographiste	Néant	Néant	Sans