



IUT de Provence
Site d'Arles
Département Informatique

Licence professionnelle
Imagerie numérique

Rapport de stage

Stage du 13 Mars au 16 juin 2006



Ajout de fonctionnalités à un outil de
visualisation d'objets 3D

Enseignant Tuteur : **Mr Romain Raffin**
Maître de stage : **Mr Éric Coiro**

REMERCIEMENTS

Je tiens à remercier Monsieur Eric COIRO, ingénieur au sein de l'Office National d'Etudes et de Recherches Aérospatiales, pour m'avoir confié ce sujet de stage qui fut très intéressant. Je le remercie aussi de m'avoir laissé beaucoup de libertés dans le développement, et de m'avoir fait confiance.

Je remercie également Monsieur Philippe SAGNIER, responsable de l'Unité MVA, pour m'avoir accueilli durant ces quatorze semaines de stage au sein du Département Optique Théorique et Appliquée du centre de Salon de Provence.

Je remercie Monsieur Alain LESAIN, responsable informatique du site, pour avoir mis à ma disposition le matériel nécessaire lorsque j'en ai eu besoin.

Je remercie aussi l'ensemble des stagiaires pour les moments agréables que j'ai passé en leur compagnie tout au long de ce stage, et plus particulièrement Fabrice, Emmanuel, et Anousak pour leurs précieux conseils en Tcl/Tk.

Pour finir, je remercie l'ensemble du personnel pour son accueil sympathique et chaleureux, qui a facilité mon intégration au sein de l'ONERA.

SOMMAIRE

REMERCIEMENTS	2
SOMMAIRE	3
INTRODUCTION	4
PRESENTATION DU CANDIDAT	5
I. PRESENTATION DE L'ONERA	6
1. Activités	6
2. Situation géographique	8
3. Le centre de Salon de Provence	8
II. Le Département Optique Théorique et Appliquée (DOTA)	9
1. Les domaines d'actions stratégiques :	9
2. L'unité MVA (Modélisation de dispositifs optroniques et Validations Associées)	10
III. CADRE JURIDIQUE DE L'ACTIVITE DE L'ENTREPRISE	11
1. Financements	11
2. Dépenses d'investissement entre 1999 et 2003	11
3. Budget d'investissement	12
4. L'ONERA et son environnement	12
5. Les contrats informatiques	12
IV. TRAVAIL RÉALISÉ	13
1. Introduction	13
2. Outil de Rendu	14
3. Outil de Visualisation	17
a. Introduction	17
b. Structure de données	18
d. Outils pour l'amélioration des BDD	23
e. Ajout d'un objet à la BDD	32
4. Description du degré d'aboutissement et d'intégration des tâches réalisés	33
V. REFLEXIONS	34
VI. CONCLUSION	35
VII. ANNEXES	36
1. Glossaire	36
2. Parties de l'interface graphique développées	38
a. Fenêtre de sélection et déplacement	38
b. Fenêtre de translation	38
c. Fenêtre de modification de la BDD	39

INTRODUCTION

Afin de valider cette année de licence, je viens d'effectuer un stage de quatorze semaines au sein de l'ONERA. Le but de ce stage a été de développer des fonctionnalités complémentaires pour un visualisateur d'objets 3D.

Cet outil de visualisation, qui a pour nom OVNI (Outil de Visualisation et de modificationN Interactif), a initialement été développé par un groupe d'étudiants de l'IUT d'Arles. Le développement a ensuite été repris par une stagiaire et complété par un ingénieur de l'ONERA.

Cette application permettait de visualiser des fichiers d'objets 3D particuliers, appelés BDD. Cependant, certaines BDD se sont révélées être erronées, et les utilisateurs de l'outil de visualisation ont souhaités pouvoir corriger ces objets.

Le but de ce stage a donc été d'ajouter des outils spécifiques à l'outil de visualisation, afin qu'il soit possible de corriger les BDD erronées.

Précision : Les termes suivis d'un * sont définis dans le glossaire qui se trouve en annexe.

PRESENTATION DU CANDIDAT

Après avoir obtenu un baccalauréat scientifique à Limoges, je me suis orienté vers un DUT informatique option Imagerie Numérique.

Étant très intéressé par les jeux vidéo et l'informatique, je souhaitais m'orienter vers une formation qui puisse me permettre de travailler dans le milieu des jeux.

Cependant, le DUT m'a permis de découvrir d'autres domaines très intéressants, comme la 3D, la programmation, ou encore la programmation d'applications en 3D temps réel.

Après avoir obtenu le DUT, j'ai me suis orienté vers une licence informatique Imagerie Numérique, afin de compléter et de parfaire mes connaissances dans ce domaine. Je souhaitais aussi avoir plus de connaissances dans le domaine de la programmation de logiciels pour la synthèse d'image.

Cette année de licence m'a permis de découvrir qu'un bon nombre d'entreprises, pas forcément en rapports avec le domaine du jeu vidéo, développaient des applications utilisant la 3D en temps réel, ou encore des applications de visualisation d'objets 3D.

I. PRESENTATION DE L'ONERA

1. Activités

L'ONERA, Office National d'Etudes et de Recherche Aérospatiales, est un organisme d'état placé sous la tutelle de la DGA*. Depuis sa création en 1946, les ingénieurs de l'ONERA imaginent, testent, simulent, modélisent et expérimentent des concepts nouveaux ; ils forgent les connaissances qui seront employées dans les 10 à 20 prochaines années dans une large palette de disciplines. L'ONERA a ainsi pour mission d'orienter et conduire les recherches et de les valoriser pour l'industrie aérospatiale. Pour cela, il participe à la conception, à la réalisation et à la mise en œuvre des moyens nécessaires ainsi qu'à la conduite des essais. Il gère aussi le premier parc européen de souffleries.



L'Airbus A380 en maquette dans la soufflerie F1 de l'ONERA

L'ONERA prend en compte les enjeux collectifs actuels majeurs : le transport aérien et spatial de demain exige de lutter contre le bruit, de sauvegarder l'environnement terrestre (pollution) et spatial (débris), et d'accompagner la croissance du trafic aérien. Continuer d'accroître les performances des avions, des hélicoptères et des lanceurs, augmenter leur cadence d'exploitation, tout en réduisant les nuisances sur l'environnement, c'est le défi que relève l'ONERA.

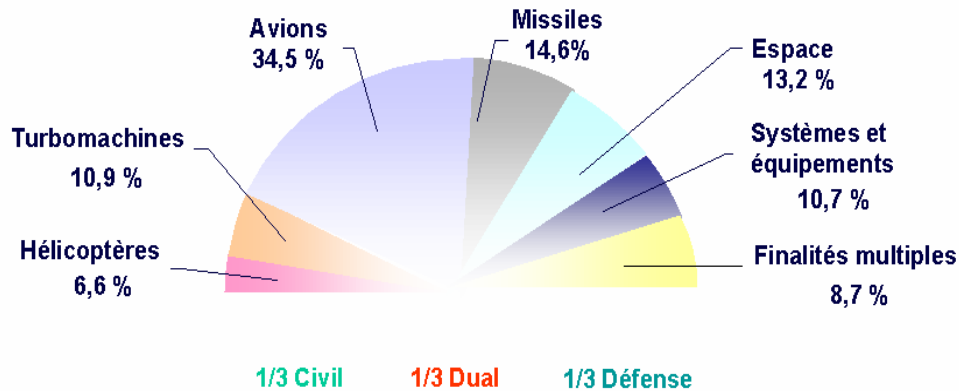
Il répond aussi aux nouveaux besoins de la défense ; il conçoit les moyens de surveillance et de veille, ainsi que les systèmes de défense de demain, garants de souveraineté à long terme, pour la France et l'Europe.

Fidèles à leur mission de former les scientifiques de demain, 300 ingénieurs de l'ONERA partagent leur savoir en enseignant et en animant des séminaires dans les universités et les grandes écoles.

Organisme à caractère industriel et commercial, l'ONERA finance ses activités par des conventions de recherches conclues aussi bien avec des services de l'état ou des organismes publics (CNES*, CEA*...) qu'avec des entreprises du secteur privé (EADS, Dassault Aviation...). Par ailleurs, l'ONERA compte aussi parmi ses partenaires plusieurs écoles et universités françaises ainsi que de nombreux établissements étrangers tels que le Van Kármán Institut ou la NASA*.

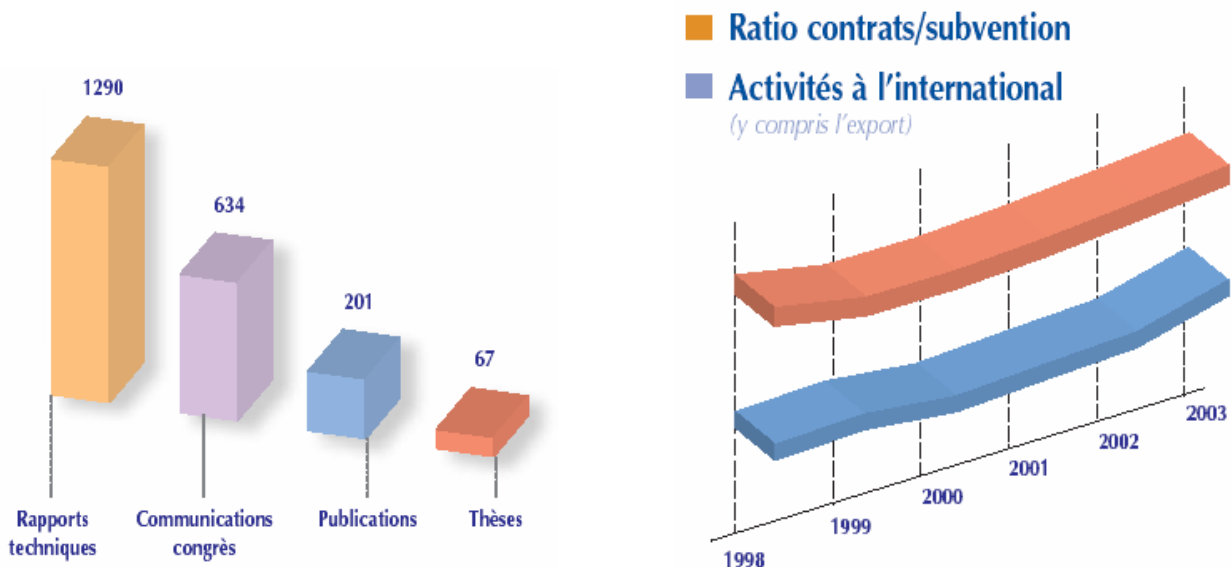
C'est ainsi que par la diversité de ses partenaires et de ses activités, l'ONERA constitue aujourd'hui un véritable pont entre la recherche scientifique et l'industrie ; c'est pourquoi son nom se trouve associé aux grands programmes aéronautiques tels que Mirage, Concorde, Ariane, Rafale ou encore Tigre.

Une recherche finalisée civile, de défense et duale



Un portefeuille d'activités équilibré, couvrant l'ensemble des besoins de l'industrie aéronautique et spatiale, réparti par tiers entre recherche de défense, civile et duale.

Quelques chiffres



D'un point de vue production scientifique en 2003, on peut remarquer qu'un grand nombre de rapports techniques a été produit (1290), ensuite viennent les communications en congrès (634), puis les publications et enfin les thèses.

2. Situation géographique

L'Office s'implique dans toutes les techniques nécessaires à la conception d'un engin aérospatial à savoir : Aérodynamique, Mécanique du vol, Energétique, Résistance des structures, Physiques, Matériaux, Optique, Acoustique, Electromagnétisme, Electronique, Systèmes, Robotique et Traitement de l'information.

Ces activités sont regroupées sous quatre branches :

Mécanique des Fluides et Energétique, Matériaux et Structures, Physique et Traitement de l'Information et Systèmes.

Ces quatre branches totalisent 17 départements répartis sur l'ensemble du territoire français à travers les 8 sites que compte l'ONERA et dans lesquels travaillent environ 2000 personnes, dont plus de 1000 scientifiques.



Répartition géographique des sites de l'ONERA

3. Le centre de Salon de Provence

Le centre de recherche conjoint ONERA/Ecole de l'Air situé sur la base aérienne 701 de Salon de Provence regroupe 3 unités de recherche de l'ONERA :

- L'unité Pilotage, Simulation et Expérimentation en vol du département Commande des Systèmes et Dynamiques du vol (DCSD).
- L'unité Modélisation de dispositifs optroniques et Validations Associées (MVA) du département Optique Théorique et Appliquée (DOTA).
- L'unité Imagerie SAR du département Electromagnétisme et Radar (DEMR).



Centre de Salon de Provence

II. Le Département Optique Théorique et Appliquée (DOTA)

Ce département fédère les compétences de l'ONERA dans le domaine de l'optique et de l'optronique. Son activité est essentiellement tournée vers le développement de capacités de détection, d'identification et de localisation de menaces ou de cibles en environnement difficile d'une part, l'amélioration de l'efficacité des systèmes d'autoprotection et/ou de contre mesures optroniques d'autres part.

Les gains potentiels en terme de réduction des coûts, encombrement des équipements et durcissement face aux agressions laser sont pris en compte.

Le DOTA réfléchit aux actions tournées vers la "rupture conceptuelle" (incluant les technologies innovantes). Il se donne pour objectif d'être mobilisateur de talents pour faire bénéficier la DGA et l'industrie de la meilleure expertise possible.

Le programme d'études et de recherches du département prépare le futur, en cohérence avec le modèle de capacités technologiques de la DGA dans le domaine de l'aéronautique et de l'espace, au travers d'études amont et de la participation à la conception et/ou la réalisation de démonstrateurs (JOANNA, LOLA, Alerte avancée,...). Les opticiens s'impliquent par ailleurs dans des tâches de prospective au sein de la DGA (filiale optronique, structure POLOQ,...) et nouent de très nombreuses coopérations avec un effort d'ouverture à l'étranger.

1. Les domaines d'actions stratégiques :

DAS1: Modélisation de la scène optronique et validation

DAS2: Conception, réalisation et qualification d'instruments optiques de hautes performances

DAS3: Imagerie Optique à Haute Résolution

DAS4: Applications des Lasers



2. L'unité MVA (Modélisation de dispositifs optroniques et Validations Associées)

Activités :

Les compétences de MVA ont un caractère multidisciplinaire. Ainsi, MVA est impliquée dans la modélisation de systèmes optroniques complexes, dans des activités à caractère expérimental et dans des études conjointes avec d'autres unités du DOTA.

Axes d'études – compétences :

Les principales compétences de MVA portent sur les domaines suivants :

Radiométrie infrarouge

Etalonnage de caméras infrarouges

Systèmes laser

Guerre optronique

Modélisation de la thermique de scène optronique

Caractérisation thermique d'éléments de scènes optronique

Modélisation de capteur infrarouge

Modélisation du rayonnement infrarouge d'avions

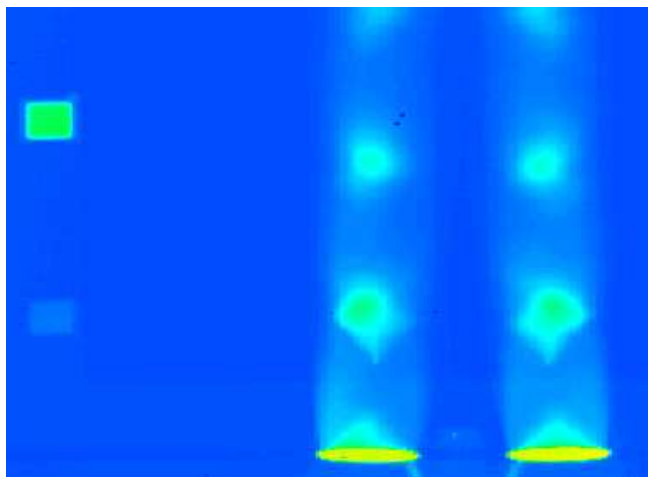
Principaux projets :

PEA OGT : Optimisation du Guidage Terminal de missiles à imagerie infrarouge

PEA EVASIGO : EVALuation par la Simulation de la Guerre Optronique

PEA MIRADOR : Moyen Infrarouge Aéroporté pour l'alerte et la désignation d'Objectif Rapide anti-missile

DE SIRA : Développement Exploratoire sur la Signature Infrarouge d'Aéronef



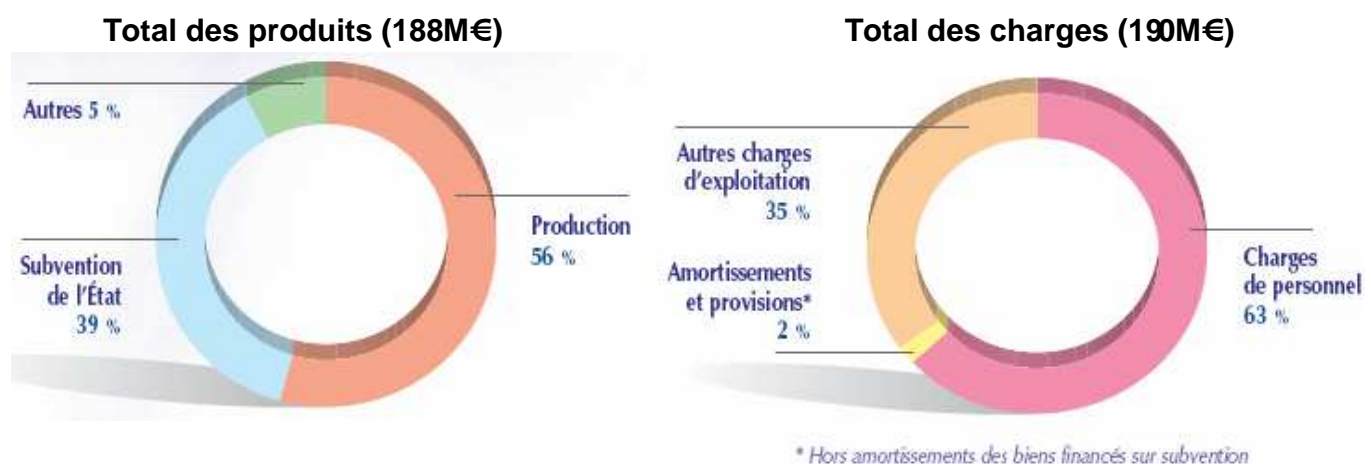
PEA Etude d'une méthodologie de production de gabarits de signature infrarouge et d'évaluation terrain des caméras visibles et infrarouges

CESAME : Centre Etatique de Simulation Avancée en électroMagnétismE

III. CADRE JURIDIQUE DE L'ACTIVITE DE L'ENTREPRISE

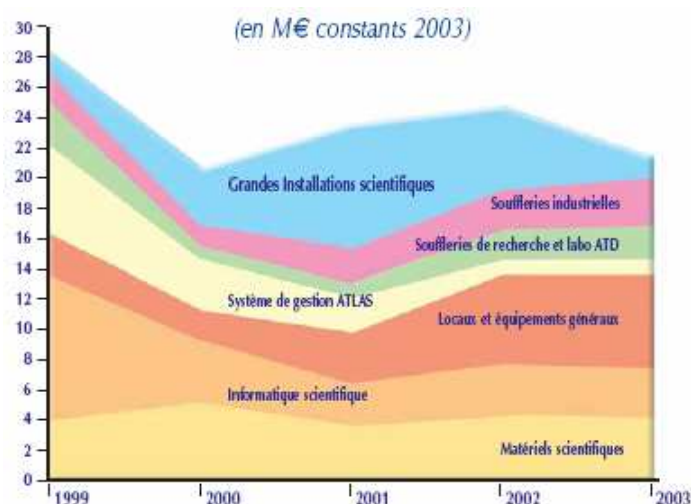
L'ONERA est un office national de recherche ce qui signifie que ce n'est pas une entreprise et qu'elle fait partie du secteur public.

1. Financements



Les revenus de l'ONERA proviennent pour 3/5^{ème} des prestations d'études et de recherche avec les clients étatiques ou industriels et pour 2/5^e des subventions de l'État. Sa politique tarifaire a pour objectif d'autofinancer l'activité contractuelle, la subvention de l'État finançant la recherche fondamentale. Ses investissements sont financés par une subvention spécifique.

2. Dépenses d'investissement entre 1999 et 2003



La majeure partie des dépenses d'investissement se fait dans l'achat de matériels scientifiques, dans les locaux et les équipements généraux mais surtout dans les grandes installations scientifiques.

3. Budget d'investissement

La source principale de financement de l'ONERA en 2003 est la DGA*. Elle représente 76.27% de la totalité du budget. Ensuite c'est la DGAC* avec 11.89%, puis ce sont les fonds propres de l'ONERA (7.59%) et enfin le financement des collectivités territoriales avec 4.25%.

4. L'ONERA et son environnement

◆ Industrie aérospatiale :

EADS, EADS/ST, Astrium, Eurocopter, Airbus, MBDA, Snecma, Dassault, Sagem, Thales, SNPE...

◆ Donneurs d'ordres et agences de programmes :

DGA*, DPAC*, CNES*, esa*, UE*, MRNT*

◆ Recherche publique et enseignement :

SUPAERO*, CNRS*, JAXA*, CIRA*, INTA*, FOI*, USAirForce, NASA*, NLR*, EREA*, DLR*, TSAGI*, CNRT*, ENS, Universités, Grandes Ecoles...

L'ONERA est leader mondial dans de nombreux domaines cependant il n'est pas seul. Dans l'industrie aérospatiale d'autres acteurs sont présents comme Thalès ou Eurocopter.

Il finance ses activités par des contrats ou plus exactement par des conventions de recherches. Ces contrats se font avec des donneurs d'ordres et des agences de programmes tels que la DGA*, le CNES*...

Enfin, il compte parmi ses partenaires de nombreux établissements étrangers ou non, tels que la NASA*, mais aussi plusieurs écoles et universités françaises.

5. Les contrats informatiques

L'ensemble du personnel de l'ONERA ainsi que tous les intervenants extérieurs travaillant dans ses locaux sont soumis au devoir de réserve, de confidentialité, concernant la protection des informations confiées par des industriels. Ils doivent respecter le règlement intérieur en ce qui concerne l'utilisation des moyens informatique de l'ONERA.

Cette annexe s'intéresse principalement à plusieurs points :

- Accès aux ressources informatiques et services Internet
- Règle d'utilisation, de sécurité et de bon usage
- Conditions de confidentialité
- Respect de la législation concernant les logiciels
- Préservation de l'intégrité des systèmes informatiques
- Usage des services Internet
- ... etc...

Le règlement sur l'utilisation des moyens informatique de l'ONERA traite, la protection industrielle (confidentialité), les contrats informatiques et la responsabilité de l'utilisateur, ainsi que la sécurité informatique et les sanctions encourues en cas de non respect de la législation.

IV. TRAVAIL RÉALISÉ

1. Introduction

Le code de calcul du rayonnement des avions (CRIRA) développé à l'ONERA, permet le calcul de la signature infrarouge d'aéronefs. Des outils de calculs géométriques sont intégrés dans le code, afin de gérer notamment les calculs d'images. Par ailleurs, CRIRA a besoin d'un logiciel de visualisation d'objets 3D afin de vérifier la conformité des bases de données géométriques des aéronefs.

Concernant le code CRIRA, les calculs de rendu sont actuellement réalisés avec des outils commerciaux. L'ONERA souhaite s'en affranchir afin de pouvoir mieux maîtriser la chaîne de calculs et disposer des codes sources. Le premier objectif du stage a été de développer un outil de rendu faisant parti de cette chaîne de calculs et permettant de générer une image à partir des caractéristiques d'un capteur.

Un outil de visualisation d'aéronefs a été développé en interne. Les aéronefs visualisés sont décrits par des points, et des facettes avec des propriétés physiques associées. Ces informations sont stockées dans un fichier dénommé base de données géométrique (BDD).

Certaines fonctionnalités du visualisateur devant être améliorées afin de répondre aux besoins, le second objectif du stage a été de mettre au point de nouvelles fonctionnalités afin de permettre :

- l'ajout d'un objet simple à l'aéronef avec les fonctionnalités nécessaires pour l'orienter, le déplacer, et le mettre à l'échelle de manière interactive ;
- la sélection graphique d'une ou d'un ensemble de facettes afin de pouvoir modifier leurs propriétés (orientation des normales, attribut de groupes, ...) ;
- l'amélioration de la géométrie des bases de données géométriques actuelles : correction de défauts, raccord d'objets supplémentaires. Pour cela, un outil sera mis au point afin de pouvoir fusionner des points voisins ou confondus.

Les contraintes techniques suivantes ont été définies par l'entreprise :

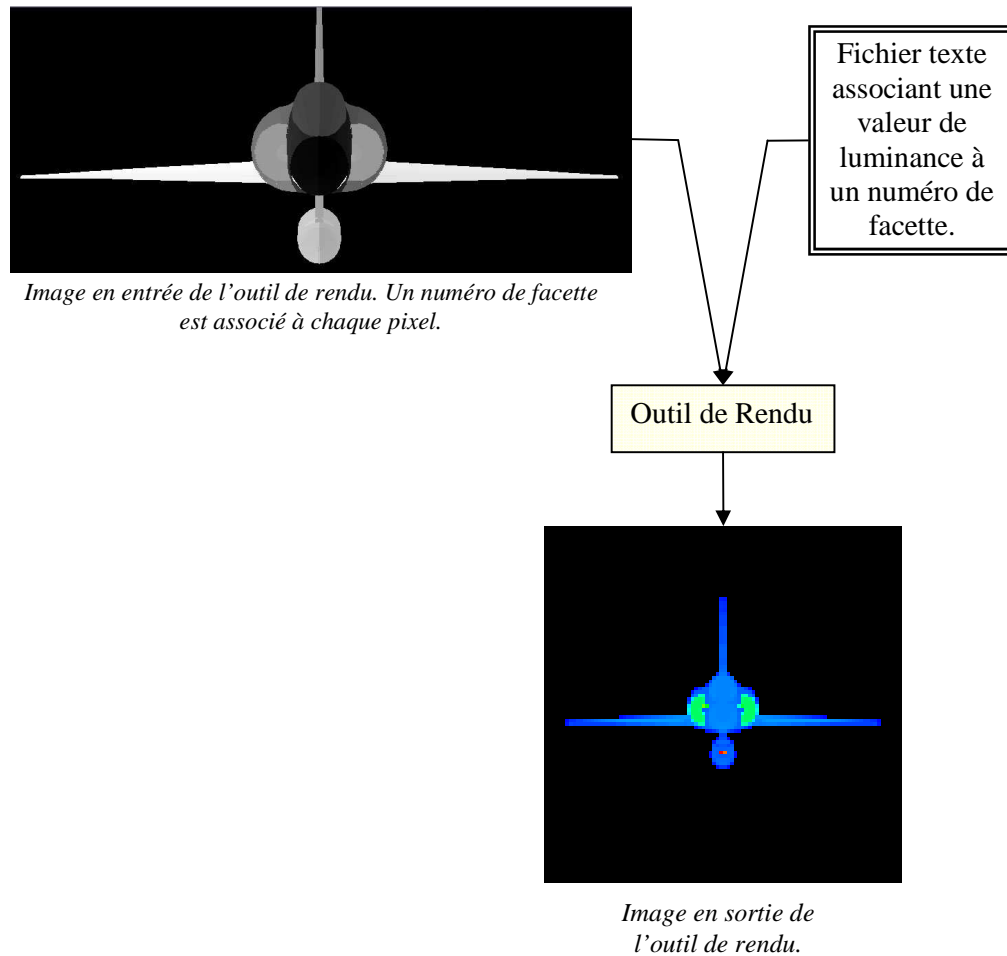
- l'outil de visualisation doit pouvoir être utilisé sur station SUN et PC.
- l'utilisation du langage C ; langage de programmation non orienté objet. Les utilisateurs de cet outil souhaitent garder la possibilité de comprendre le code source, afin de pouvoir y apporter des modifications par la suite.

Aucune contrainte technique n'a été définie pour le développement de l'outil de visualisation, alors que l'outil de visualisation définissait les contraintes techniques suivantes :

- le temps réel ; améliorer et ne pas alourdir le code de l'outil de visualisation afin que l'affichage des BDD reste fluide ;
- l'utilisation des langages Tcl/Tk, C, et de la librairie graphique OpenGL, pour continuer le développement de l'outil de visualisation.

2. Outil de Rendu

Le but de cet outil est de créer, à partir d'une image suréchantillonnée et d'un fichier texte, une image (image capteur) où chaque pixel est associé à une valeur de luminance. L'image suréchantillonnée, représentant une BDD sous un certain angle de vue, est une image brute qui contient un numéro de facette associé à chaque pixel. Le fichier texte contient des valeurs de luminances associées à un numéro de facette. L'organigramme suivant schématise les entrées et sorties de l'outil de rendu.



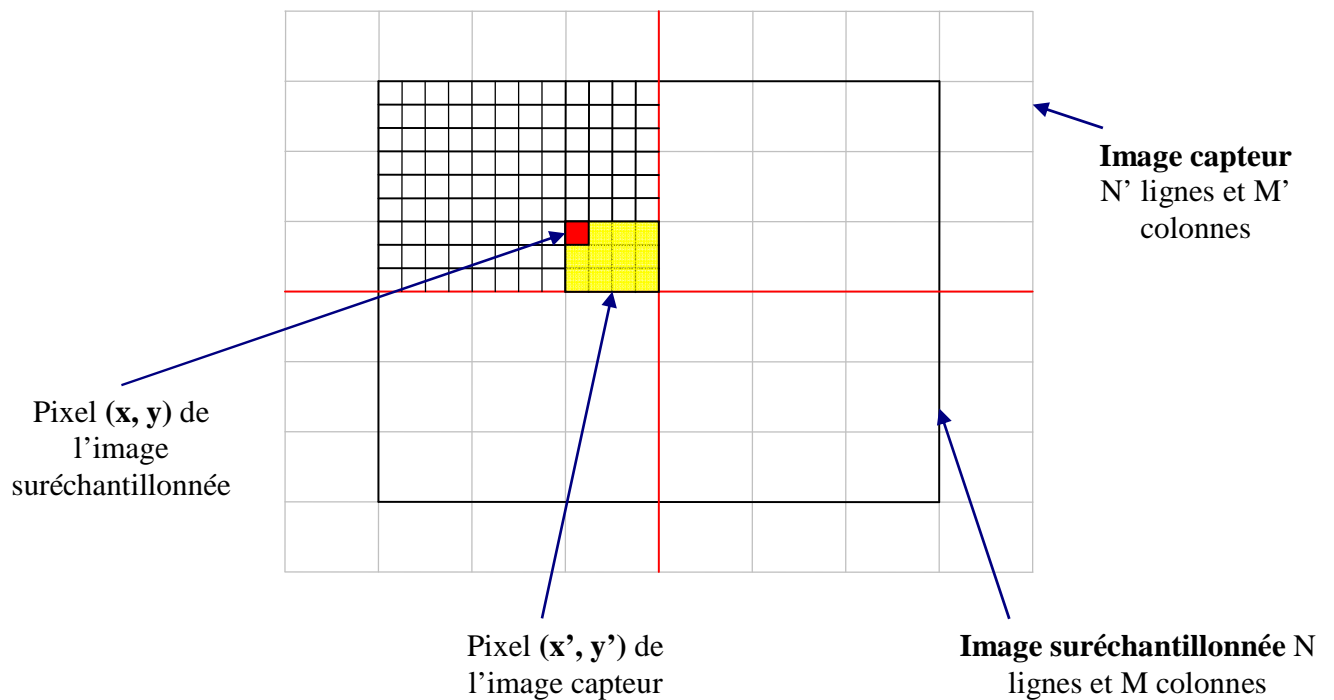
Un pixel sur l'image suréchantillonnée est donnée par le couple (x, y) , et la taille de l'image est donnée par N et M ; N pour le nombre de ligne, et M pour le nombre de colonnes.

Un pixel sur l'image capteur est donné par le couple (x', y') , et la taille de l'image est donnée par N' et M' ; N' pour le nombre de ligne et M' pour le nombre de colonnes.

La taille d'un pixel de l'image capteur est définie à partir d'un nombre de pixels de l'image suréchantillonnée, à l'aide de deux coefficients entiers k et l :

$$x' = k * x \quad \text{et} \quad y' = l * y$$

Le schéma ci-dessous reprend tous ces paramètres.



Dans cet exemple, les valeurs de k et l sont fixées :

$$x' = k * x = 4x \quad \text{et} \quad y' = l * y = 3y$$

Rappelons que l'outil de rendu doit fournir une image brute de $N' \times M'$ pixels, associant une luminance à chaque pixel, à partir d'une image suréchantillonnée brute associant un numéro de facette à un pixel.

Les fichiers textes, appelés fichiers de luminance et fournis par la chaîne de calculs, permettent d'associer une luminance à un numéro de facette, et sont placés dans le répertoire où l'outil s'exécute. L'outil de rendu doit créer autant d'images capteur que de fichiers de luminances.

Il est aussi important de noter que, lors du traitement, l'image suréchantillonnée et l'image capteur sont centrées en $(0,0)$, ce qui revient à aligner les centres des deux images, comme on peut le voir sur le schéma ci-dessus.

L'explication de tous ces paramètres d'entrées est nécessaire afin de comprendre l'algorithme principal, donné ci-dessous.

Début

Lire l'image suréchantillonnée et récupérer les numéros de facette associés à un pixel dans un tableau.

Créer un tableau vide qui recevra les valeurs de l'image capteur.

Pour chaque rendu à effectuer

 Lire les fichiers de luminances et stocker les valeurs dans un tableau

Pour chaque pixel de l'image capteur

 Pour les pixels de l'image suréchantillonnée appartenant au pixel de l'image capteur, effectuer la moyenne des luminances correspondant au numéro de facette associé au pixel.

 Stocker la moyenne dans le tableau représentant l'image capteur.

Fin Pour

 Créer l'image de rendu à partir du tableau

 Créer une image TGA pour avoir un aperçu de l'image binaire créée

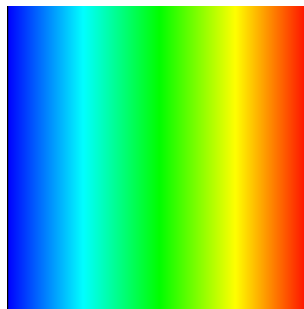
Fin Pour

Fin

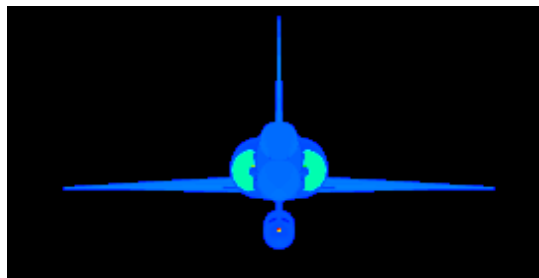
Pour contrôler le contenu de chaque image capteur créée, une image au format TGA est générée pour chaque image capteur.

Le format d'image choisi est le format TGA car une image utilisant une palette de couleurs au format TGA est assez simplement créée, et de plus, ce format est reconnu par un grand panel de logiciels de visualisation d'images sous SUN et Windows.

La palette de couleur suivante est utilisée dans chaque image pour associer une couleur à une luminance.



La palette ne contenant que 256 couleurs, les valeurs des luminances sont interpolées puis arrondies, car le but de cette image est de donner un aperçu, comme le monte l'image suivante.



3. Outil de Visualisation

a. Introduction

Le développement de l'outil de visualisation a été initialisé en 2004 par six étudiants de l'IUT d'Arles, dans le cadre d'un projet tuteuré, pour créer de nouveaux outils développés en C et OpenGL afin de remplacer les modules existants qui utilisent le logiciel SPECRAY de la société OKTAL.

Le travail réalisé a été repris en stage, afin de tester l'outil de visualisation sur la plateforme Sun de l'ONERA, et de compléter les fonctionnalités du visualisateur. Ce travail a ensuite été complété et optimisé par un ingénieur de l'ONERA.

L'outil initial a été développé en C, utilisant OpenGL comme librairie graphique, et la librairie GTK pour la création de l'interface graphique.

Après de nombreux problèmes avec l'installation de la librairie GTK sur la machine SUN, la librairie Tcl/Tk a remplacé GTK pour la création de l'interface graphique. Tcl/Tk offre une compatibilité ascendante, ce qui permet aux applications développées avec d'anciennes versions de Tcl/Tk, d'être compatibles avec de nouvelles versions.

Tcl (Tool Command Language) est un langage interprété, disponible gratuitement et qui peut être utilisé sous de très nombreux systèmes d'exploitation. Ce langage possède une extension, Tk (pour Toolkit) qui permet de créer des widgets*.

L'autre avantage de ce langage est qu'il peut être facilement interfacé avec de nombreux langages, comme le langage C.

L'interface codée en Tcl/Tk peut appeler des fonctions du code C si celles-ci respectent la syntaxe suivante :

```
int ET_COMMAND_nom_fonction(ET_TCLARGS)
{
    // Code en C...
    return TCL_OK; ou return TCL_ERROR;
}
```

L'utilisation de l'application mktclapp permet ensuite de générer les fichiers nécessaires à une compilation normale de l'application.

Plus précisément, mktclapp utilise les fichiers suivants en entrée :

- Source de l'interface graphique en Tcl/Tk, dans un fichier ayant l'extension **.tcl** (par exemple **application.tcl**)
- Fichier(s) contenant le code C des fonctions appelées dans le fichier **application.tcl**. L'extension de ces fichiers est **.c**

A partir de ces fichiers mktclapp génère un fichier de configuration et des fichiers sources exclusivement en C. Les fichiers obtenus peuvent être utilisés normalement par un compilateur, tel que gcc, pour compiler correctement l'application.

Pour permettre l'utilisation d'OpenGL, un widget spécial est utilisé : ToGL. Ce widget permet de définir les fonctions de bases utilisées par OpenGL, c'est-à-dire :

- Fonction d'initialisation ;
- Fonction d'affichage ;
- Fonction de redimensionnement.

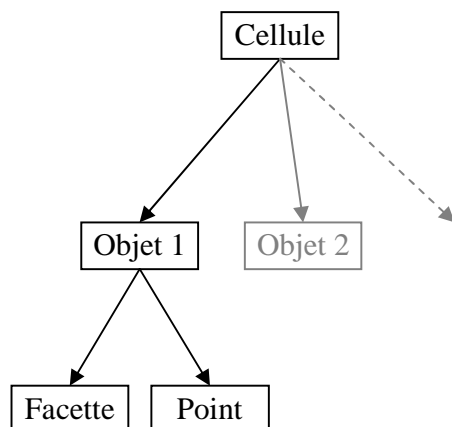
Ainsi, une zone d'affichage utilisant la librairie OpenGL va pouvoir être ajoutée dans l'application.

Après une rapide présentation des différents langages utilisés, et de la façon donc l'application est compilée, nous allons nous intéresser aux différentes fonctionnalités développées.

Des modifications sur certaines parties du programme ont été nécessaires afin de pouvoir réaliser les différentes fonctionnalités demandées. Le choix de ces modifications sera expliqué dans les fonctionnalités concernées.

b. Structure de données

Lorsqu'une BDD est ouverte par l'application, les informations contenues dans le fichier permettent d'initialiser la structure de données suivante :



Cellule : cette structure possède un tableau contenant tous les **objets** constituant la BDD, ainsi qu'un entier permettant de connaître le nombre d'objets.

Objet : cette structure caractérise un objet à partir des éléments suivants :

- nombre de **points** ;
- tableau de points ;
- nombre de **facettes** ;
- tableau de facettes ;
- numéro de l'objet ;
- nom de l'objet.

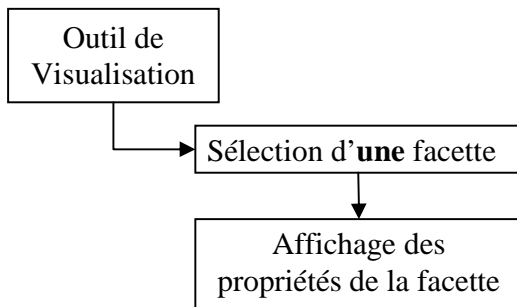
Facette : cette structure caractérise une facette à partir des éléments suivant :

- nombre de sommets constituant la facette ;
- tableau contenant les indices des sommets de la facette ;
- la normale de la facette ;
- le groupe auquel appartient la facette.

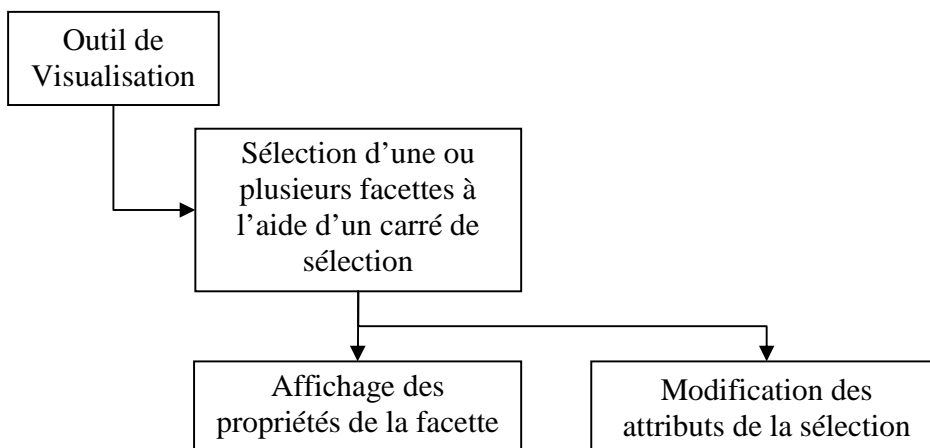
Point : cette structure caractérise un point en trois dimensions à l'aide de trois réels.

c. Modification d'attributs de facettes

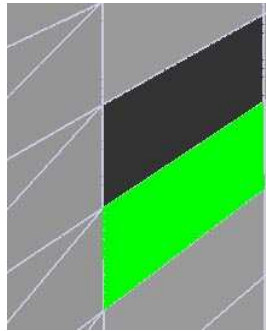
Une fonctionnalité de l'outil interactif permet d'afficher les propriétés d'une facette sélectionnée à la souris. Les propriétés affichées sont le numéro de groupe et le sens de la normale. Cette fonctionnalité est représentée par le schéma suivant.



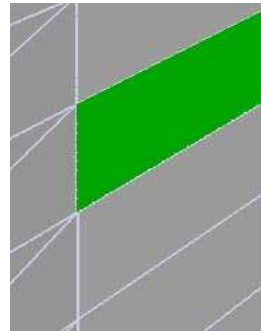
Cette fonctionnalité a été complétée afin de pouvoir sélectionner un groupe de facettes, et afin de modifier les attributs de la sélection. Le schéma suivant représente les modifications apportées.



Sur certaines BDD erronées, il est difficile de voir si la normale de la facette est correctement orientée. Une fonctionnalité a été développée pour faciliter la distinction entre une facette mal orientée et une facette correctement orientée. Lorsqu'une facette est sélectionnée, la couleur de la sélection est soit vert clair si la facette est considérée comme visible, soit vert foncé si la facette est considérée comme non visible. Les images suivantes illustrent ce principe.



facette correctement orientée



facette mal orientée

La fonctionnalité d'OpenGL utilisée pour effectuer la sélection est le picking* par nom. Rappelons que le picking* permet de trouver les objets se situant dans une région restreinte de l'écran.

Le picking par nom consiste à dessiner tous les éléments de la scène en leur attribuant un nom. A partir de coordonnées (du pointeur de la souris par exemple), OpenGL renvoie une pile d'éléments entrés en intersection. Ces éléments sont reconnaissables grâce au nom qu'il leur a été attribué. Cette fonctionnalité a l'avantage d'être très rapide car tout est réalisé par la carte graphique, grâce à OpenGL.

Sélection d'une facette

Le picking* permet donc de récupérer les facettes se trouvant près du pointeur de la souris. Sachant que la profondeur de toutes les facettes intersectées est donnée, il est aisé de récupérer la facette la moins profonde. Il n'y a donc aucun problème pour sélectionner une seule facette.

Cette solution existait déjà dans l'application. La fonction gérant le picking permettait de trouver la facette la moins profonde, et sauvegardait le numéro de la facette sélectionnée dans une variable globale. Cette facette était ensuite redessinée d'une autre couleur par-dessus la facette existante.

Sélection de plusieurs facettes

Sélectionner plusieurs facettes implique que l'on puisse enregistrer plusieurs numéros de facettes afin de savoir celles qui doivent être dessinées d'une autre couleur. Le nombre de facettes sélectionnées dépendant de l'utilisateur, le meilleur moyen de sauvegarder ces numéros de facette est d'utiliser une liste d'entiers.

Une gestion de liste d'entiers a donc été développée, et propose les fonctions suivantes :

- création de la liste ;
- ajout d'un entier dans la liste ;
- suppression d'un entier dans la liste ;
- chercher un élément dans la liste ;
- taille de la liste ;
- vider la liste.

La suppression de la liste se fait avec la fonction free(), qui permet de libérer la mémoire allouée en C.

Principe

Lorsque l'utilisateur désire sélectionner plusieurs facettes, un carré de sélection (dont la taille dépend de la position du pointeur de la souris) est tracé à l'écran, et toutes les facettes présentes dans ce carré sont sélectionnées.

Dans ce cas là, la profondeur des facettes ne peut être utilisée. En effet si on garde la facette la moins profonde, une seule facette est retenue, et on ne peut sélectionner plusieurs facettes.

Solutions envisagées

La première solution envisagée a été d'utiliser le culling lorsque le picking est effectué. Le culling est une fonctionnalité d'OpenGL qui permet d'afficher seulement les facettes considérées comme visibles. Pour définir si une facette est visible ou non, OpenGL utilise une formule qui se base sur les coordonnées des points de la facette, et l'ordre de parcours des points de la facette.

Il peut arriver, avec des BDD erronées, que les facettes les plus proches soient considérées comme non visible, et les facettes les plus éloignées comme visible. Dans ce cas, si le culling est utilisé, les mauvaises facettes seront sélectionnées.

Pour palier à cet inconvénient, une deuxième solution a été envisagée, l'utilisation de la fonction OpenGL `glReadPixel()`. Cette fonction permet entre autre de récupérer la valeur de profondeur pour un pixel. Ainsi on cherche la valeur de profondeur la plus grande dans la zone sélectionnée, et toutes les facettes ayant une profondeur inférieure ou égale à cette valeur seront sélectionnées.

Le principal inconvénient de cette solution est la vitesse d'exécution : la fonction `glReadPixel` est une fonction qui demande beaucoup de ressources, ce qui induit une grande chute du nombre d'images affichées en une seconde. L'affichage n'étant plus fluide, cette fonction ne peut être utilisée en temps réel.

La solution finale consiste donc à proposer trois modes de sélection, pour permettre à l'utilisateur de pouvoir corriger l'orientation des normales dans tous les cas :

- Sélection unique : sélection de la facette la moins profonde ;
- Sélection des facettes avant : sélection des facettes considérées comme visibles ;
- Sélection des facettes arrières : sélection des facettes considérées comme non visibles.

Une fonction, permettant d'étendre la zone de sélection complète le mode de sélection unique. En effet, les facettes voisines, ayant une normale de même signe que la facette sélectionnée au départ, sont ajoutées à la sélection. Cela permet de modifier les attributs d'un plus grand nombre de facettes dans les cas où la sélection des facettes avant n'est pas possible.

La sélection des facettes est gérée grâce aux trois fonctions suivantes :

- une fonction appelée lorsque le bouton central est appuyé. Elle permet d'activer le mode sélection, de vider la liste de sélection, et d'obtenir les coordonnées de la position de départ du rectangle de sélection ;
- une fonction appelée quand la souris est en mouvement. Elle passe à la fonction C gérant le picking les coordonnées actuelles du pointeur de la souris. Le picking va être effectué dans une région de sélection rectangulaire, définie à l'aide du point de départ et du point courant ;
- une fonction appelée lorsque le bouton central est lâché, qui désactive le mode de sélection.

Une contrainte a été ajoutée pour la sélection de plusieurs facettes ; seules les facettes appartenant à un même objet peuvent être sélectionnées. Lorsqu'une première facette est sélectionnée, le numéro d'objet est enregistré dans une variable globale. Le numéro d'objet des facettes pickées est ensuite comparé au numéro d'objet enregistré, et seules les facettes ayant un numéro identique sont ajoutées dans la liste de sélection.

Le mode de sélection a été étendu afin de pouvoir aussi sélectionner des points et des objets.

Sélection de points

Avec la structure de donnée actuelle, aucune distinction n'est possible entre les points. Le calcul d'une normale pour chaque point serait nécessaire pour déterminer si un point est visible ou non.

La sélection de points reprend le mode de sélection unique, pour sélectionner un seul point, ainsi que le mode de sélection des "facettes avant", pour sélectionner un groupe de points. Les points sélectionnés sont ajoutés à la liste de points, pour être dessinés d'une couleur différente.

Sélection d'objets

La sélection d'objet utilise uniquement le mode de sélection unique. En effet, le numéro d'objet auquel appartient la facette sélectionnée est récupéré, et toutes les facettes ayant le même numéro sont dessinées d'une autre couleur.

Maintenant que des sélections de points, de facettes, et d'objets peuvent être effectuées, les outils permettant de modifier la sélection vont être présentés.

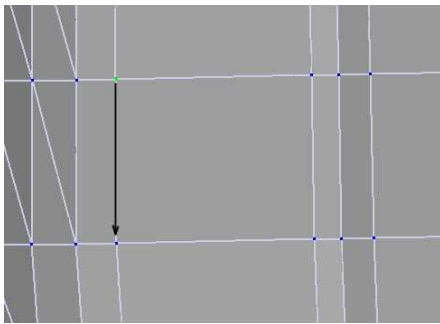
d. Outils pour l'amélioration des BDD

Comme nous l'avons vu précédemment, les bases de données d'aéronefs peuvent présenter des défauts de construction (facettes non jointives, trous, ...). Le développement de fonctionnalités permettant de corriger ces défauts est donc nécessaire.

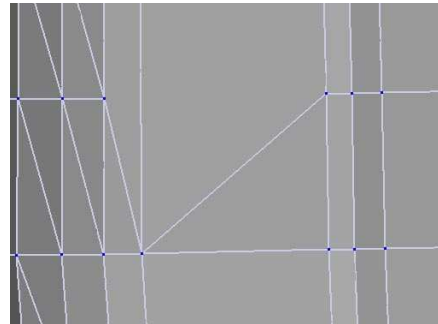
Afin de pouvoir modifier localement la forme des facettes et modifier les points, il doit être possible de fusionner deux points et de subdiviser une facette. Pour réaliser ces deux fonctions, plusieurs outils ont été développés.

Fusion de deux points

La fusion de deux points consiste à amener le point à fusionner sur un point de destination. Pour cela, l'utilisateur doit cliquer dans un premier temps sur le point à fusionner, et dans un deuxième temps sur le point de destination pour fusionner les deux points. Ainsi la facette qui contenait le point à fusionner, contient à sa place le point de destination.

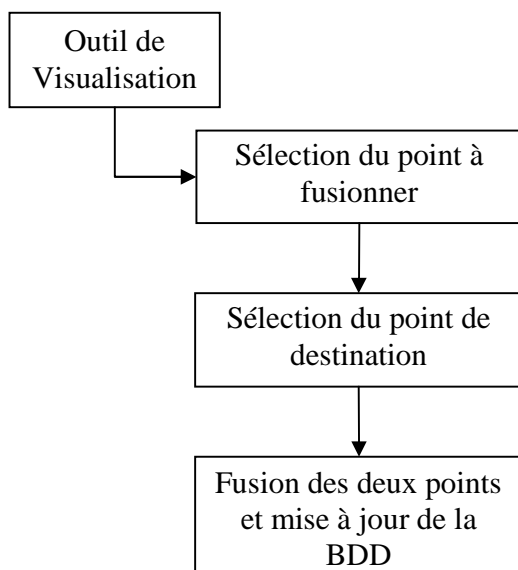


Maillage avant la fusion des deux points



Maillage après la fusion des deux points

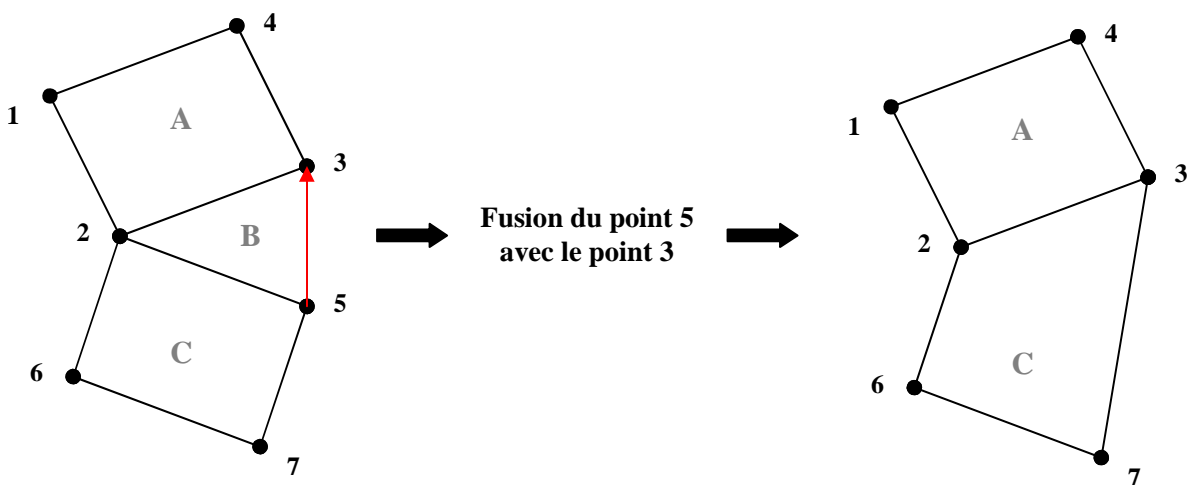
Le principe de cet outil est schématisé ci-dessous.



Nous avons vu que la structure de données est telle qu'il existe un tableau de points pour chaque objet ainsi qu'un tableau de facettes, où chaque facette contient un tableau de sommets, contenant lui-même les indices des points constituant la facette.

Lors de la fusion de deux points, l'indice du point à fusionner doit être remplacé par l'indice du point de destination dans le tableau de sommets des facettes concernées. Cependant plusieurs cas de figure se distinguent une fois que les points ont été fusionnés.

Facette constituée de trois sommets



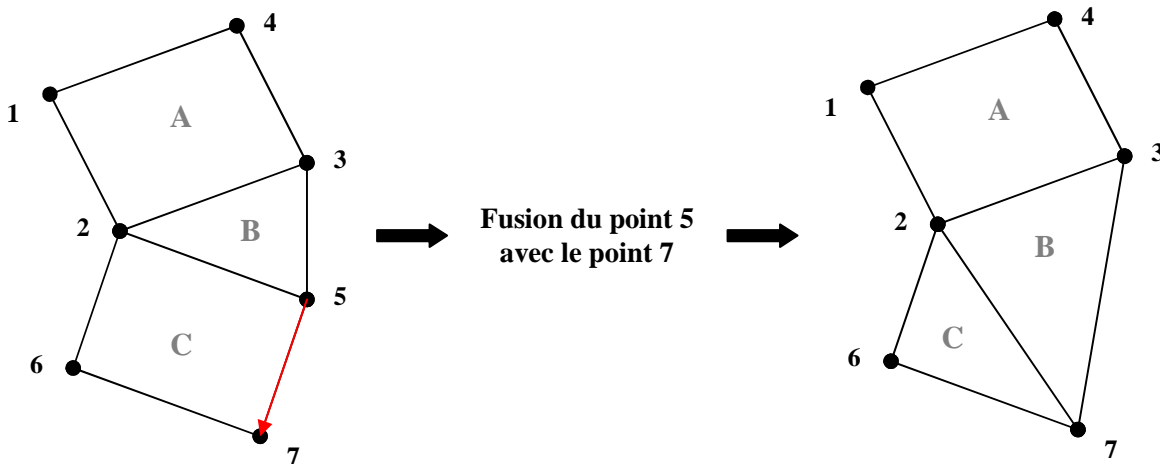
Dans le tableau de sommets de la facette B, le point 5 est remplacé par le point 3. Le point 3 se trouve deux fois dans le tableau de sommet de la facette B. Une fois la deuxième occurrence du point 3 supprimée, la facette B ne possède plus que deux sommets. Comme une facette doit au minimum être constituée de trois sommets, la facette B est supprimée.

L'autre facette concernée est la facette C, lors du remplacement du point 5 par le point 3, on se retrouve dans le cas des facettes à quatre sommets.

Facette constituée de quatre sommets

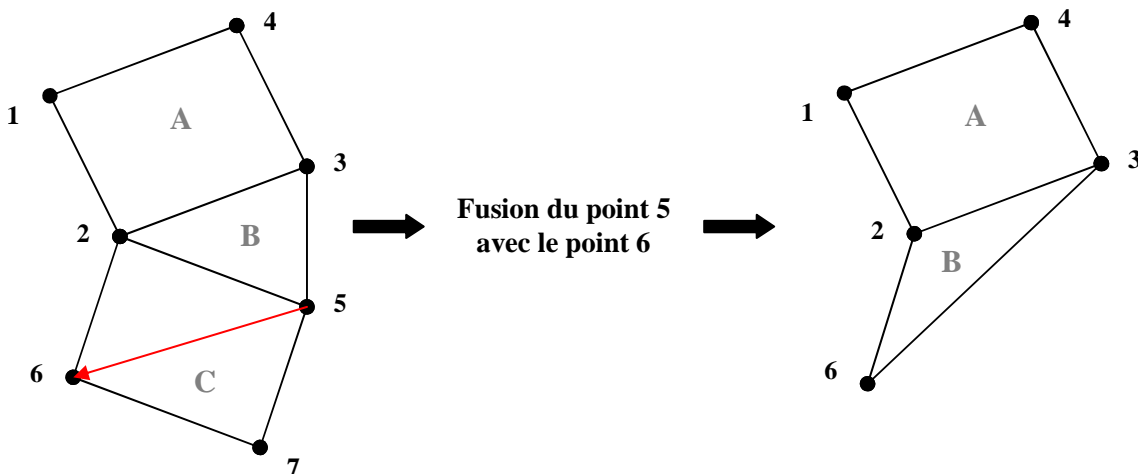
Reprenons le contexte précédent. La facette C possède aussi le point 5, et lorsque le point 5 est remplacé par le point 3, cela n'a aucune influence sur le nombre de points de la facette C. Ceci est dû au fait qu'un point de la facette C est fusionné avec un point d'une autre facette. Deux autres cas se distinguent si deux points de la facette C sont fusionnés : la fusion de deux points successifs d'une même facette, et la fusion de deux points non successifs d'une même facette.

- fusion de deux points successifs d'une même facette

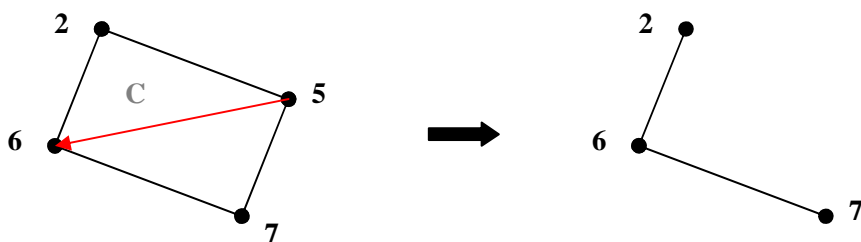


La fusion du point 5 avec le point 7, qui appartiennent tous les deux à la facette C, influe sur le nombre de points de la facette C. Une fois le sommet 5 remplacé par le sommet 7 dans le tableau de sommet de la facette C, le point 7 est en double. La deuxième occurrence du sommet 7 est enlevée du tableau, et comme la facette C a un nombre de sommets supérieur ou égal à trois, elle n'est pas supprimée.

- fusion de deux points non successifs d'une même facette



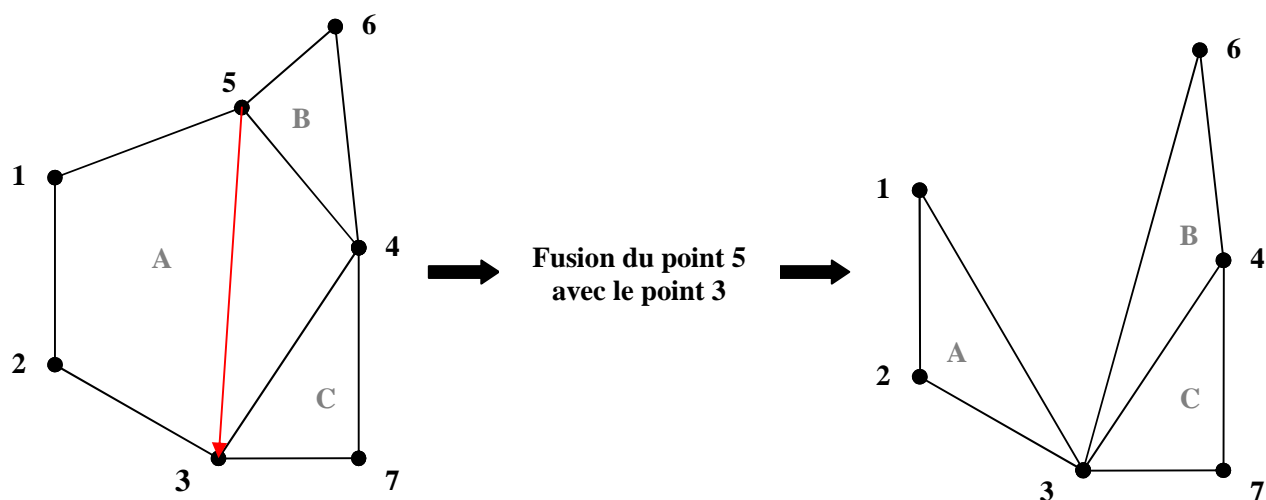
La fusion du point 5 avec le point 6, qui appartiennent tous les deux à la facette C pose un cas plus complexe. Le fait de remplacer le sommet 5 par le sommet 6 dans le tableau de sommet n'est pas suffisant. De plus, la deuxième occurrence du sommet 6 ne doit pas être supprimée, pour garder la forme de la facette qui est la suivante :



La facette C a été remplacée par deux segments : $[2,6]$ et $[6,7]$. La surface de la facette C étant nulle, elle peut être supprimée.

Il existe un dernier cas, il s'agit de la fusion de deux points non successifs d'une même facette, pour une facette possédant plus de quatre sommets.

- fusion de deux points non successifs d'une facette de plus de quatre sommets



Dans ce cas, le point 5 est fusionné avec le point 3. Prenons la liste des sommets de la facette A avant la fusion : 1, 2, 3, 4, 5

On relève les positions du sommet à fusionner et du sommet de destination, dans la liste.

Le sommet 3 est en 3^{ème} position et le sommet 5 en 5^{ème} position.

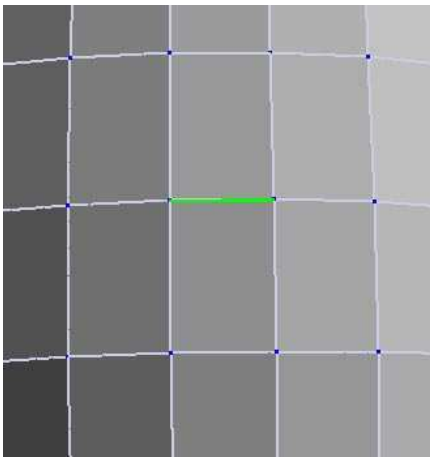
Si il y a plus d'un sommet intercalé entre le sommet 3 et 5, alors une facette peut être créée. Dans la figure ci-dessus, les sommets 1 et 2 sont intercalés entre les sommets 3 et 5. Comme le sommet 5 est remplacé par le sommet 3, on peut recréer une facette avec les sommets 1, 2 et 3.

Par exemple, prenons une facette à six sommets. Après fusion de deux points successifs, on peut soit obtenir une facette de quatre sommets, ou soit obtenir deux facettes de trois sommets. Cela dépend du nombre de sommets intercalés entre le sommet de fusion et le sommet de destination.

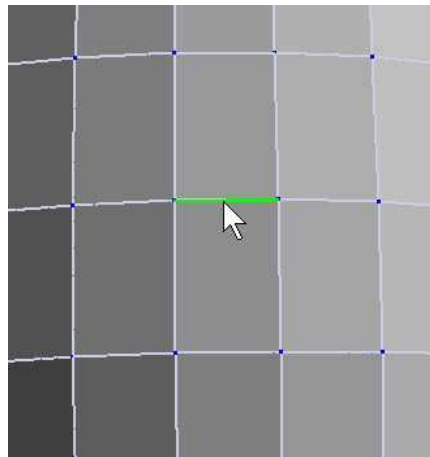
Ajout d'un point

L'ajout de points permet d'ajouter un point sur une arête. Pour que l'ajout d'un point puisse se faire dans n'importe quel cas, un point doit pouvoir être ajouté à une ou n arêtes, car une arête peut appartenir à une facette ou à plusieurs facettes. Dans le cas où une arête appartient à plusieurs facettes, toutes les facettes concernées doivent être mises à jour.

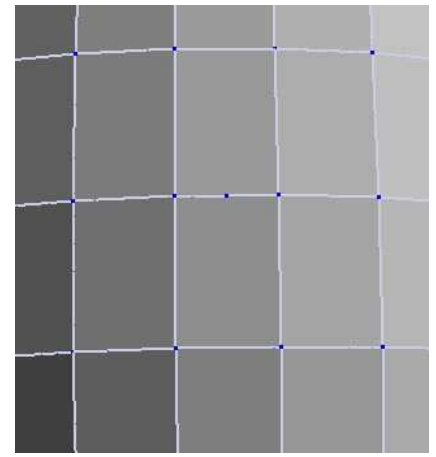
Pour ajouter un point, l'utilisateur doit pointer une arête avec le pointeur de la souris. L'arête choisie apparaît en surbrillance, et l'utilisateur clique sur l'endroit de l'arête où il souhaite ajouter le point.



L'arête en surbrillance a été pointée par la souris

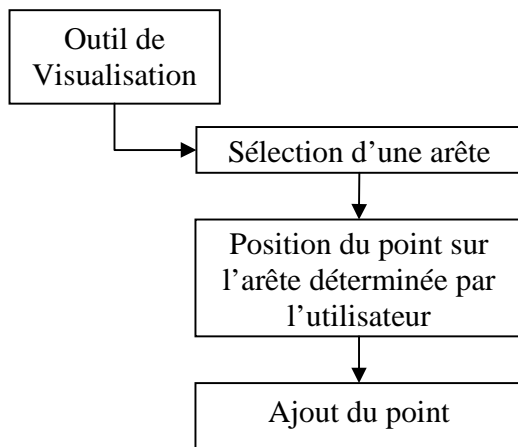


La position de la souris détermine la position du point sur l'arête



Le point a été bien ajouté à l'arête

Le fonctionnement de cet outil est schématisé de la manière suivante.



La structure de données utilise des points, des facettes, mais aucune arête. La création d'un tableau d'arêtes est donc nécessaire afin de pouvoir sélectionner une arête.

Le tableau d'arêtes est généré à partir des facettes de tout les objets, et en utilisant la structure suivante :

```

struct arete {
    struct point3d * a, * b ; //pointeurs sur les deux points de l'arête
    unsigned int num_objet ; //numéro d'objet auquel les deux points appartiennent
} ;

```

Le tableau d'arêtes doit être re-généré à chaque fois que des modifications ont été apportées à la BDD.

Remarque : La structure de données de l'application n'a pas été changée, car elle est utilisée par d'autres outils de calculs. La modifier impliquerait la vérification de tous les outils de calcul développés.

L'ajout de la gestion des arêtes permet d'effectuer du picking sur les arêtes dessinées. Une fois que l'utilisateur a choisi l'arête qu'il souhaitait modifier, il doit cliquer sur le bouton central pour indiquer à quel endroit il désire ajouter le point.

Pour ajouter le point au bon endroit sur l'arête, les coordonnées paramétriques de la droite (ab), définies par le point a de l'arête sélectionnée, et le vecteur \overrightarrow{ab} formé par les points a et b de l'arête, sont utilisées.

$$\text{Coordonnées paramétriques : } \begin{cases} x' = xa + t * \overrightarrow{xab} \\ y' = ya + t * \overrightarrow{yab} \\ z' = za + t * \overrightarrow{zab} \end{cases}$$

Ce système permet d'obtenir les coordonnées de n'importe quel point se situant sur la droite (ab).

Si $0 \leq t \leq 1$, alors le point se trouve sur le segment [ab].

La méthode suivante est utilisée pour définir la valeur de t en fonction de l'endroit où l'utilisateur a cliqué.

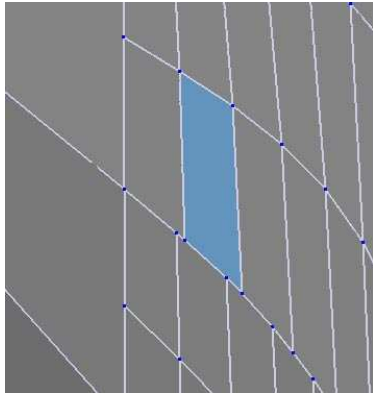
- Les coordonnées du pointeur de la souris sont enregistrées lorsque le clic a été effectué. Les coordonnées du pointeur de la souris sont données dans le repère écran, en x et y.
- Les coordonnées des deux points constituant l'arête sont converties dans le repère de l'écran en utilisant la fonction OpenGL gluUnProject().
- Soit le vecteur \overrightarrow{ab} défini par le point a et b de l'arête, et le vecteur \vec{v} défini par le point a de l'arête et le point correspondant à la position du pointeur de la souris. Les coordonnées des vecteurs \overrightarrow{ab} et \vec{v} sont définies dans le repère écran, ce qui permet d'utiliser la relation suivante : $\|\overrightarrow{ab}\| = k * \|\vec{v}\|$
- Dans la relation précédente, k est une constante, elle peut donc être remplacée par t. La relation

$$\text{est modifiée pour obtenir la valeur de t en fonction des normes des vecteurs } \overrightarrow{ab} \text{ et } \vec{v} : t = \frac{\|\overrightarrow{ab}\|}{\|\vec{v}\|}$$

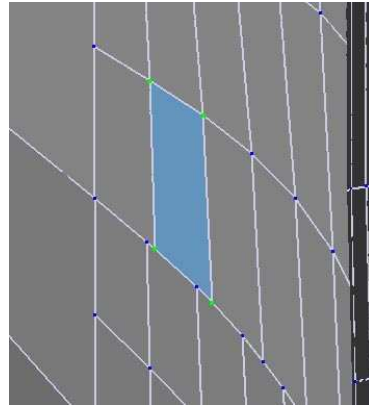
Les coordonnées du point à ajouter peuvent donc être calculées. Il ne reste plus qu'à ajouter le point dans le tableau de points de l'objet concerné, et à mettre à jour le tableau de sommets des facettes auxquelles le segment appartient.

Création d'une facette

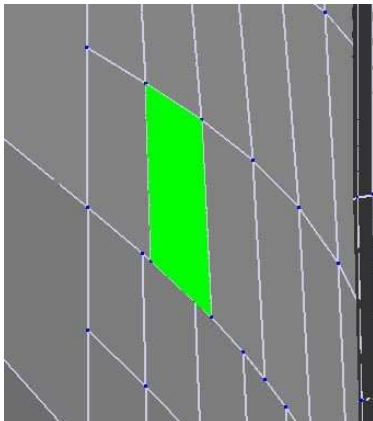
Pour créer une facette, l'utilisateur doit cliquer sur les points qu'il désire pour créer la facette. La facette est créée dès qu'il re-clique sur le premier point sélectionné.



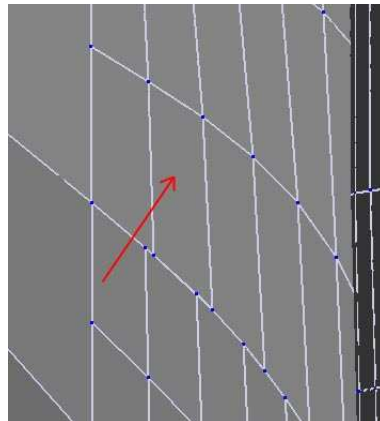
1. Il manque une facette au milieu de l'image



2. Les points de la facette sont sélectionnés. Une fois sélectionné, le point change de couleur.



3. On clique sur le premier point sélectionné pour créer la facette.
La facette créée est en surbrillance.



4. Après avoir désélectionné la facette, on peut voir que la facette a bien été créée.

Par définition, un minimum de trois points doit être sélectionné pour créer une facette. En deçà de cette limite, un message avertit l'utilisateur que la facette n'a pas été créée car le nombre de points est insuffisant.

Cet outil reprend la sélection de points évoquée précédemment. Cependant, une fonction spéciale de picking, exclusivement utilisée par cet outil, a été créée afin de ne pas alourdir la fonction de sélection existante.

Lors de l'utilisation de cet outil, l'utilisateur peut soit annuler la création de la facette, inverser la normale de la facette qu'il vient de créer, ou encore supprimer la facette qu'il vient de créer.

D'autres fonctionnalités ont été développées, afin de pouvoir modifier la BDD. Ces fonctionnalités sont brièvement décrites ci-dessous.

Les outils d'ajout de point, et de création de facette ont été développés afin de pouvoir subdiviser une facette. Cependant une troisième fonctionnalité est nécessaire pour subdiviser correctement une facette.

Suppression d'une facette

La fonction de suppression de facette supprime une facette dont le numéro, et le numéro de l'objet auquel elle appartient, est connu. Une fonction Undo est disponible afin de permettre d'annuler la suppression de la dernière facette.

Les étapes à franchir pour supprimer la facette sont :

- Sauvegarder la facette à supprimer ;
- Supprimer la n^{ième} facette du tableau de facettes appartenant à l'objet ;
- décaler les facettes suivant la n^{ième} facette, d'un rang vers la gauche dans le tableau de facettes, afin de mettre à jour les indices des facettes ;
- Mettre à jour le nombre de facettes dans la structure de l'objet ;
- Régénérer le tableau d'arêtes.

Par manque de temps, cette fonctionnalité autorise seulement la suppression d'une seule facette à la fois. La suppression d'un groupe de plusieurs facettes est envisageable car tous les éléments nécessaires sont présents dans le code.

Lorsque l'on édite des bases de données erronées, il est aussi très utile de pouvoir changer l'orientation de la normale d'une facette.

Inversion de la normale d'une facette

Pour inverser la normale d'une facette il suffit, d'inverser l'ordre de parcours de ses points, et de recalculer la normale de la facette. Cette fonctionnalité peut s'appliquer aussi bien à une seule facette qu'à un groupe de plusieurs facettes.

Une fonction permettant d'annuler l'inversion de la normale des facettes sélectionnées est aussi disponible. Avec le recul, ceci ne se révèle pas très utile, car une deuxième inversion suffit pour inverser les normales des facettes sélectionnées.

Recalculer toutes les normales

Un outil a été ajouté afin de pouvoir recalculer toutes les normales de tous les objets de la scène. Cet outil est surtout utilisé pour des BDD erronées.

L'algorithme de calcul de normales se sert de l'hypothèse suivante : Une facette est constituée au minimum de trois points qui ne sont pas alignés.

Début

On prend les deux premiers points de la facette pour avoir un premier vecteur

On déclare une variable de boucle, et on l'initialise à 3, ce qui donne : $i = 3$

Faire

On prend le 2^{ème} et le $i^{\text{ème}}$ point pour avoir un deuxième vecteur.

Si les trois points sont alignés, incrémenter i .

Tant Que i n'est pas le dernier point et que les trois points sont alignés

Calculer le produit vectoriel du premier et du deuxième vecteur.

Fin**Simplifier les sommets**

Cet outil été développé afin de diminuer le nombre sommets d'une BDD, dans le cas où plusieurs sommets sont très proches. La distance minimale est définie en dur dans le code, toutefois, elle pourrait être définie via l'interface afin de permettre à l'utilisateur d'adapter la distance minimale entre deux sommets. Néanmoins, la distance minimale ne doit pas être trop élevée afin de ne pas dégrader la BDD.

Le meilleur moyen d'expliquer le fonctionnement de cet outil est de donner l'algorithme général de traitement.

Début

Pour chaque objet de la liste d'objet

Pour chaque point de l'objet

Pour chaque point se trouvant après le point traité

On vérifie la distance entre les deux points

Si cette distance est inférieure à la distance minimale, le numéro du point proche est enregistré.

Fin Pour

Pour chaque point proche détecté

Pour chaque facette de l'objet

Remplacer l'indice du point proche par l'indice du point traité.

Décrémenter les indices des points dont l'indice est supérieur au point proche.

Fin Pour

Décalage des points du tableau de point d'un rang vers la gauche pour les points dont l'indice est supérieur au point proche.

Fin Pour

Fin Pour

Désallocation du tableau de point, et réallocation avec la nouvelle taille du tableau.

Vérifier la non présence de doublons dans les tableaux de sommets de chaque facette, et les corriger s'il y en a.

Fin Pour

Modifier la cellule.

Fin

e. Ajout d'un objet à la BDD

Le déplacement de la sélection a déjà été développé pour les points et les facettes. Cette fonctionnalité a été étendue pour être applicable sur un objet.

Déplacement de la sélection

Le déplacement a été développé pour les trois primitives de sélection : un point (ou un groupe de points), une facette (ou un groupe de facette), et un objet.

Pour déplacer un point, il suffit d'effectuer une translation du point. Pour déplacer un groupe de points, il est nécessaire d'appliquer cette translation à tous les points sélectionnés. Il est donc aisé de déplacer une facette ou un objet à partir du déplacement de point : tous les points de la facette ou de l'objet subissent la même translation.

La translation que subissent les points est définie par l'utilisateur à l'aide d'une boîte de dialogue, et peut être définie de manière très précise sur les trois axes x, y, et z.

La rotation et la mise à l'échelle d'un objet ont été demandées afin de pouvoir orienter et mettre à l'échelle un objet. Malheureusement, je n'ai pas eu le temps de réaliser ces deux fonctions.

Seule la fonction de déplacement d'un objet permet de placer correctement un objet dans la scène. La dernière fonctionnalité manquante est l'ajout d'un objet.

Ajout d'un objet

Cette fonctionnalité autorise l'ajout d'un objet à la scène courante, et permet ainsi de pouvoir compléter certaines BDD. Nous avons vu que lorsqu'une BDD est chargée, une cellule est initialisée avec les données contenues dans le fichier. Une fois la cellule créée, les objets sont insérés dans une liste d'objets.

L'ajout d'un objet consiste à insérer un ou plusieurs objets à la suite de la liste d'objets existante. Comme la sauvegarde du fichier est faite à partir des données contenues dans la cellule, il est nécessaire de mettre à jour le nombre d'objets et les objets contenus dans la cellule avant d'enregistrer le fichier. L'algorithme général d'ajout d'un objet est le suivant :

Début

- Ouvrir une fenêtre de sélection de fichiers BDD permettant à l'utilisateur de choisir l'objet à ajouter
- Créer une cellule temporaire
- Charger les données du fichier dans la cellule temporaire.

- Pour** chaque objet de la cellule temporaire

- Vérifier que l'objet a un numéro et un nom différent que les objets déjà présents dans la liste

- Si** le numéro ou le nom est identique

- Attribuer à l'objet de la cellule temporaire un numéro ou un nom différent

- Fin Si**

- Ajouter l'objet dans la liste d'objets de la scène

- Fin Pour**

Fin

4. Description du degré d'aboutissement et d'intégration des tâches réalisés

Tous les outils et fonctionnalités présentés jusqu'à présent ont été réalisés, à part la rotation et la mise à l'échelle d'un objet.

L'outil de rendu présenté en partie IV. 2. ainsi que plusieurs petites fonctions réalisées vont être intégrés à la chaîne de calcul CRIRA.

Concernant l'outil de visualisation, les outils et fonctionnalités développés vont pouvoir être utilisés afin de modifier les bases de données.

V. REFLEXIONS

Je regrette de ne pas avoir pu réaliser tous les objectifs fixés au début du stage. En effet la durée du stage se révèle être assez brève, et il est dommage qu'elle ne puisse pas être prolongée.

Il est vrai que ma gestion du temps n'a pu être pas été exemplaire. Certaines parties du code existant au début du stage ont du être revues, afin d'améliorer le code, et j'ai passé pas mal de temps à développer des outils qui ce sont révélés, au final, être inutilisables ou pas assez performants.

Par exemple, j'ai passé quelques jours à essayer de développer des algorithmes de picking* qui ce sont révélés inutilisables. J'ai aussi passé cinq jours à développer un outil permettant de détecter et corriger les facettes mal orientées dans une BDD. Devant le grand nombre de cas à prendre en compte, et le peu de temps en ma disposition, le développement de cet outil n'a pu être finalisé. Toutefois, plusieurs parties de cet outil ont été reprises dans le code, comme la fonctionnalité permettant de savoir si une facette est bien ou mal orientée.

Le code réalisé pendant se stage a pour but d'être le plus ouvert possible. En effet, les fonctions ont été développées dans la perspective que le code serait repris et complété par d'autres personnes.

VI. CONCLUSION

D'une part, ce stage m'a permis d'étendre mes connaissances dans plusieurs domaines, notamment dans l'utilisation du langage C pur, c'est-à-dire le langage C compilé avec un compilateur C (et non C++). J'ai ainsi pu me rendre compte de tout les avantages qu'apportait l'utilisation d'un langage orienté objet, comme le C++ ou le Java. J'ai également appris le langage Tcl, qui, combiné à Tk, permet d'effectuer simplement des interfaces graphiques.

J'ai aussi étendus mes connaissances en OpenGL, particulièrement en picking*, car une recherche des différentes manières de réaliser du picking avec OpenGL a été nécessaire, et je me suis documenté sur les différentes façons d'optimiser l'affichage.

D'autre part, ce stage m'a permis de voir quelles était les aptitudes à adopter afin de travailler en équipe ou sur un projet repris par plusieurs groupe de travail. Par exemple, l'importance de réaliser du code clair, afin qu'il puisse facilement être repris. Je me suis aussi rendu compte qu'il est difficile d'estimer avec exactitude le temps nécessaire pour effectuer une tâche.

Je retire un bilan positif de ce stage, car il m'a permis de mettre en pratique mes connaissances en informatique et en imagerie numérique dans une entreprise. De plus, une grande liberté m'a été donnée pour la réalisation des divers algorithmes.

J'espère avoir répondu aux attentes de mon tuteur de stage, et avoir suffisamment fait avancer le développement de l'outil de visualisation afin que des BDD puissent être modifiées convenablement.

VII. ANNEXES

1. Glossaire

CEA : Commissariat à l'Énergie Atomique. Le CEA est un organisme public de recherche dans les domaines de l'énergie, de la défense et des technologies de l'information et de la santé.

CIRA : Centre Italien de Recherche Aérospatiale

CNES : Centre National d'Études Spatiales.

CNRS : Centre National de la Recherche Scientifique

CNRT : Centre National de Recherche Technologique

DGA : délégation générale pour l'armement.

DGAC : Direction Générale de l'Aviation Civile.

DLR : Centre aérospatial allemand.

DPAC : Direction des Programmes Aéronautiques Civils

ENS : École Normale Supérieure

EREA : Association des établissements de recherche européens en aéronautique

ESA : Agence Spatiale Européenne

FOI : Centre de recherche suédois.

INTA : Institut National (espagnol) de Technique Aérospatiale.

JAXA : Japan Aerospace eXploration Agency.

JOANNA : Joint Optronic Airbone Navigation (attaque air-sol).

LOLA : Liaison Optique Laser Aéroportée. Le contrat LOLA a pour objectif, à terme, de permettre, à partir d'un théâtre d'opérations, la transmission d'informations en temps quasi réel entre un drone et sa base située à plusieurs milliers de kilomètres, grâce à des satellites géostationnaires utilisés comme relais.

MNRT : Ministère délégué à la Recherche et aux Nouvelles technologies.

NASA : National Aeronautics and Space Administration. Il s'agit de l'administration gouvernementale responsable du programme spatial des Etats-Unis d'Amérique.

NLR : Laboratoire national d'aérospatial des Pays-Bas

Picking : Il s'agit d'un mécanisme qui restreint le dessin à une région réduite de l'écran, généralement autour du pointeur de la souris, et qui va permettre de retrouver les objets qui sont "sous" le pointeur de la souris.

POLOQ : groupe de Prospective Orientée sur les Lasers et l'Optronique.

SUPAERO : École nationale supérieure de l'aéronautique et de l'espace.

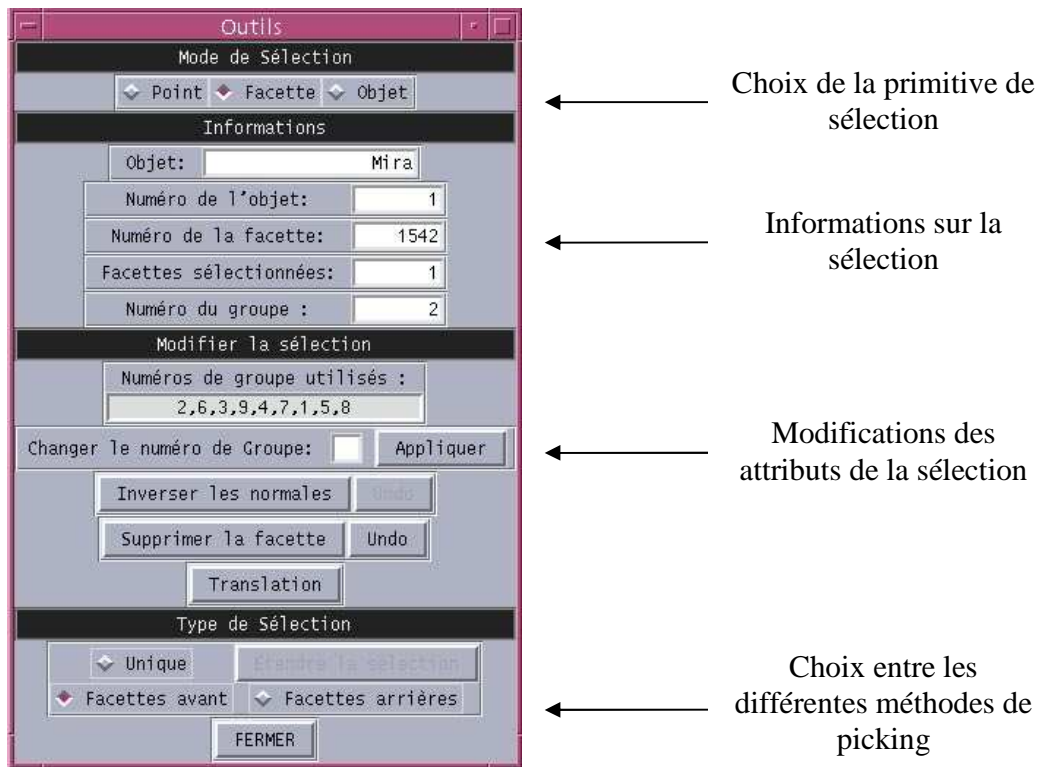
TSAGI : institut d'aérohydrodynamique russe

UE : Union Européenne.

Widget : Un widget est un élément d'une interface graphique, comme un bouton une zone de texte, un menu, etc...

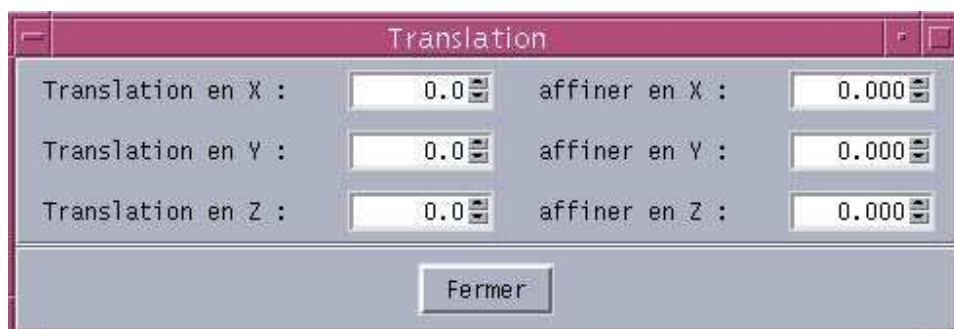
2. Parties de l'interface graphique développées

a. Fenêtre de sélection et déplacement



b. Fenêtre de translation

Cette boîte de dialogue permet à l'utilisateur d'effectuer une translation sur la sélection. La translation est appliquée en temps réel.



c. Fenêtre de modification de la BDD

