

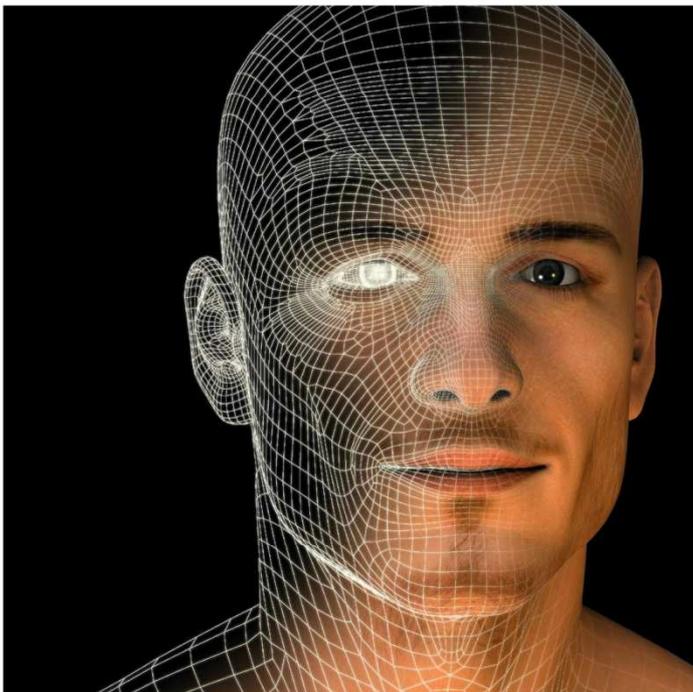
République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de La Recherche  
Scientifique



Université Mouloud MAMMERY de Tizi-Ouzou  
Faculté de Génie électrique et Informatique  
Département Informatique

Mémoire de fin de cycle en vue de l'obtention du diplôme de  
Master en Système Informatique

Conception et réalisation d'une application de  
modélisation automatique des objets 3D  
(Cas de Visage)



Dirigé Par :

Dr : AIT ADDA

Dr : TALEB

Realisé Par :

AZZOUZ Amine

BEN BELKACEM Yahia

2014/2015

# Remerciement

*Nous remercions le bon dieu pour le courage, la patience qui nous ont été utiles tout au long de notre parcours. Nous tenons très sincèrement à remercier notre promoteur **Mr TALEB HAKIM** de nous avoir guidé et pour ses pertinents conseils.*

*Nos plus vifs remerciements vont aussi **M<sup>elle</sup> AIT ADA SAMIA** qui nous a encore encadrer et aidée à terminer notre projet, nous tenons à lui exprimer notre profonde reconnaissance pour le temps précieux qu'elle nous a consacrée, ses conseil, ses encouragements qu'elle a mise à notre disposition.*

*Les membres du jury pour l'honneur qu'il nous fait en acceptant de juger notre travail.*

*Ces remerciements ne seraient pas complets si on n'a pas pensé à les destiner, avec notre profonde reconnaissance, à nos parents qui nous ont offert un environnement favorable mener à terme notre travail.*

*Enfin nous remercions tous ceux qui ont contribués de près ou de loin afin que notre travail puisse voir le jour.*

# Dédicaces

*Je dédie ce modeste travail à :*

*A mes très chers parents qui m'ont beaucoup  
soutenu durant mon travail.*

*A Mon cher frère*

*A mes très chères Sœurs*

*A mes Proches*

*A tous mes Amis(es)*

*A Mon Binôme Amine*

*A ma très chère Kahina*

*A toutes la promotion d'Informatique 2014|2015*

« YAHIA »

# Dédicace

Je didier ce travaille a

Mes Parents :

Ma Très Chère Mère

Mon Très Chèr Père

Ma Sœur (Amel) & Mon Frère (Imad)

Ma Famille & Spécialement Mon Oncle (Rezki)

Mes Amis & Collègues (Ahmed & Salim)

Mon Minôme Yahia

Mes Enseignants

Lila & Sonia

Amine



# Sommaire

<b>Introduction Générale .....</b>	<b>1</b>
<b>Chapitre I</b>	
<b>Introduction .....</b>	<b>2</b>
<b>1. Définition d'une image.....</b>	<b>2</b>
<b>1.1. Image analogique.....</b>	<b>2</b>
<b>1.2. Image numérique.....</b>	<b>2</b>
<b>1.3. Caractéristique d'une image numérique.....</b>	<b>3</b>
<b>1.3.1. Le pixel.....</b>	<b>3</b>
<b>1.3.2. Le contour .....</b>	<b>4</b>
<b>1.3.3. Résolution.....</b>	<b>4</b>
<b>1.3.4. Région.....</b>	<b>4</b>
<b>1.3.5. Dimension .....</b>	<b>4</b>
<b>1.3.6. La Luminance .....</b>	<b>4</b>
<b>1.3.7. Le bruit .....</b>	<b>4</b>
<b>1.3.8. Le contraste .....</b>	<b>4</b>
<b>1.4. Types d'images numériques .....</b>	<b>5</b>
<b>1.4.1. Image matricielle .....</b>	<b>5</b>
<b>1.4.2. Image vectorielle .....</b>	<b>5</b>
<b>2. Types d'images par couleur .....</b>	<b>6</b>
<b>2.1. Image couleur .....</b>	<b>6</b>
<b>2.1.1. Les Modèles couleur .....</b>	<b>7</b>
<b>2.2. Images en niveau de gris .....</b>	<b>8</b>

2.3.	Image binaire .....	10
3.	Formats d'Images .....	10
3.1.	Format BMP .....	10
3.2.	Format PNG .....	10
3.3.	Format GIF .....	10
3.4.	Format JPEG .....	11
4.	Traitement d'image .....	11
4.1.	Définition .....	11
4.2.	Les filtres usuels en traitement d'images .....	12
5.	La Vidéo .....	13
5.1.	Définition de la Vidéo .....	13
5.2.	Formats de vidéo .....	13
5.2.1.	AVI .....	13
5.2.2.	MPEG .....	14
5.2.3.	WMV .....	14
6.	Tracking (tracking 2D) .....	1
<b>Conclusion .....</b>		<b>15</b>
 <b>CHAPITRE II</b>		
<b>Introduction .....</b>		<b>16</b>
1.	Les étapes de production d'un Objet 3D .....	16
1.1.	Modélisation 3D .....	16
1.1.1.	Modélisation Polygonale .....	17
1.1.1.1.	Construction de maillages polygonaux .....	17
1.1.1.2.	Mode d'Utilisation .....	18
1.1.1.3.	Les Avantages et les Inconvénients .....	19

1.1.2. Modélisation par Courbes (NURBS).....	20
1.1.3. Modélisation par Subdivision de surface .....	20
1.1.4. Modélisation par Géométrie .....	21
1.1.4.1.Utilisation .....	22
1.1.5. Modélisation Volumique.....	23
1.1.5.1.Voxel .....	23
1.2. Animation .....	24
1.2.1. Les techniques d'Animation 3D .....	25
1.2.1.1.L'Animation d'images clés ou keyframe .....	25
1.2.1.2.L'Animation avec des trajectoires .....	25
1.2.1.3.L'animation de marionnettes virtuelles .....	25
1.2.1.4.L'animation de formes .....	26
1.2.2. L'hybridation du mouvement .....	27
1.2.3. L'animation procédurale .....	28
1.3. Rendu .....	28
1.3.1. Moteur de rendu .....	29
1.3.2. Types de moteurs .....	29
2. Formats de sauvegarde de l'Objet 3D .....	31
3. Visualisation de scène .....	31
3.1.La vue perspective .....	31
3.2.La vision orthonormée .....	31
4. Quelques outils de base de la modélisation .....	32
4.1.Les modificateurs : .....	32
4.2.Les matériaux .....	43
5. Exemple de modélisation d'un visage 3D .....	45

<b>Conclusion .....</b>	<b>54</b>
-------------------------	-----------

## **CHAPITRE III**

<b>Introduction .....</b>	<b>55</b>
---------------------------	-----------

<b>1. Les étapes de production .....</b>	<b>55</b>
--	-----------

<b>1.1.Récupération des vidéos .....</b>	<b>55</b>
--	-----------

<b>1.2.Parcours et extraction de points 3D .....</b>	<b>56</b>
--	-----------

<b>1.3.Création de script de construction d'objet .....</b>	<b>58</b>
---	-----------

<b>1.4.Importation &amp; exécution de fichier script .....</b>	<b>59</b>
--	-----------

<b>2. Optimisation des étapes de production .....</b>	<b>59</b>
---	-----------

<b>2.1.Récupération des vidéos .....</b>	<b>59</b>
--	-----------

<b>2.1.1. Positionnement des caméras .....</b>	<b>59</b>
--	-----------

<b>2.1.2. Lancement de balayage .....</b>	<b>59</b>
---	-----------

<b>2.2.Parcours et extraction de points 3D .....</b>	<b>60</b>
--	-----------

<b>2.3.Création de script de construction d'objet .....</b>	<b>62</b>
---	-----------

<b>3. Boite à outils .....</b>	<b>64</b>
--------------------------------	-----------

<b>4. Choix de la bibliothèque .....</b>	<b>65</b>
--	-----------

<b>5. Les critères d'une bibliothèque parfaite .....</b>	<b>66</b>
--	-----------

<b>6. Algorithmes de réalisation .....</b>	<b>67</b>
--	-----------

<b>7. Environnement technique de développement .....</b>	<b>68</b>
--	-----------

<b>7.1.Les Logiciels utilisés .....</b>	<b>68</b>
---	-----------

<b>7.2.Langage utilisé</b> .....	<b>70</b>
<b>7.3.Environment de développement</b> .....	<b>71</b>
<b>8. Présentation de quelques interfaces</b> .....	<b>73</b>
<b>Conclusion</b> .....	<b>75</b>
<b>Conclusion Générale</b> .....	<b>76</b>

# Liste des Figures

<b>Chapitre I :</b>	
<b>Figure I.1</b> : Représentation d'une image numérique.	<b>Page 3</b>
<b>Figure I.2</b> : Représentation d'une image par des pixels.	<b>Page 3</b>
<b>Figure I.3</b> : Présentation Image couleur.	<b>Page 6</b>
<b>Figure I.4</b> : Représentation des trois couleurs primaires.	<b>Page 7</b>
<b>Figure I.5</b> : Représentation des quatre couleurs primaires.	<b>Page 8</b>
<b>Figure I.6</b> : Présentation de modèle HSB.	<b>Page 9</b>
<b>Figure I.7</b> : Image en niveau de gris.	<b>Page 9</b>
<b>Figure I.8</b> : Image binaire.	<b>Page 10</b>
<b>Figure I.9</b> : Logo de logiciel Photoshop.	<b>Page 11</b>
<b>Figure I.10</b> : Logo de logiciel GIMP.	<b>Page 12</b>
<b>Chapitre II :</b>	
<b>Figure II.1</b> : Subdivision de surface.21	<b>Page 21</b>
<b>Figure II.2</b> : Les Opérations booléennes pour la modélisation géométrie.22	<b>Page 22</b>
<b>Figure II.3</b> : arbre binaires représentant la construction de l'objet en géométrie.23	<b>Page 23</b>

<b>Figure II.4</b> : Une série de voxels dans une pile avec un seul boxel.23	<b>Page 23</b>
<b>Figure II.5</b> : Une matrice de voxels en nuances de gris.23	<b>Page 23</b>
<b>Figure II.6</b> : Effet rendu mental ray.	<b>Page 30</b>
<b>Figure II.7</b> : Eclairage.	<b>Page 30</b>
<b>Figure II.8</b> : Projection orthographique et perspective.	<b>Page 32</b>
<b>Figure II.9</b> : Effet modificateur révolution.	<b>Page 33</b>
<b>Figure II.10</b> : Effet du modificateur Courbure.	<b>Page 34</b>
<b>Figure II.11</b> : Lissage.	<b>Page 35</b>
<b>Figure II.12</b> : Lissage sur une zone.	<b>Page 35</b>
<b>Figure II.14</b> : Effet du modificateur Optimiser.	<b>Page 37</b>
<b>Figure II.15</b> : Effet du modificateur ride.	<b>Page 38</b>
<b>Figure II.16</b> : Effet du modificateur Etirement.	<b>Page 38</b>
<b>Figure II.17</b> : Effet du modificateur Symétrie.	<b>Page 39</b>
<b>Figure II.18</b> : Effet du modificateur Onde.	<b>Page 40</b>
<b>Figure II.19</b> : Effet du modificateur Extrusion.	<b>Page 41</b>
<b>Figure II.20</b> : Effet du modificateur Effiler.	<b>Page 41</b>
<b>Figure II.21</b> : Effet du modificateur interpolateur.	<b>Page 42</b>

<b>Figure II.22</b> : Exemple de Cible d'interpolation.	<b>Page 43</b>
<b>Figure II.23</b> : L'interface Editeur de matériaux.	<b>Page 45</b>
<b><i>Chapitre III</i></b>	
<b>Figure III.1</b> : Diagramme des étapes de production	<b>Page 55</b>
<b>Figure III.2</b> : Positionnement des caméras.	<b>Page 56</b>
<b>Figure III.3</b> : Exemple du balayage.	<b>Page 56</b>
<b>Figure III.4</b> : Déterminer le centre du point vert	<b>Page 57</b>
<b>Figure III.5</b> : Illustration de deux points 2D.	<b>Page 57</b>
<b>Figure III.6</b> : Construction du point 3D.	<b>Page 58</b>
<b>Figure III.7</b> : Photo avant filtre.	<b>Page 61</b>
<b>Figure III.8</b> : Postérisation niveau 2.	<b>Page 61</b>
<b>Figure III.9</b> : Postérisation niveau 20.	<b>Page 61</b>
<b>Figure III.10</b> : Exemple de masque.	<b>Page 63</b>
<b>Figure III.11</b> : Interface Autodesk 3ds max.	<b>Page 69</b>
<b>Figure III.12</b> : Interface Photoshop CC.	<b>Page 70</b>
<b>Figure III.13</b> : Interface NetBeans.	<b>Page 71</b>
<b>Figure III.14</b> : Interface Eclipse.	<b>Page 72</b>

<b>Figure III.15</b> : Interface d'accueil de l'application.	<b>Page 73</b>
<b>Figure III.16</b> : extrait de fichier MaxScript '.ms'.	<b>Page 74</b>
<b>Figure III.17</b> : Interface de logiciel 3D.	<b>Page 74</b>
<b>Figure III.18</b> : Interface de résultat de l'Objet 3D.	<b>Page 75</b>

# *Introduction Générale*

Tous les jours, le monde Informatique nous invite à découvrir les projets les plus innovants. L'imagination et le temps seront les seules limites pour plus de révolution face à cette innovation.

Parmi ces innovations qui ont bouleversé le monde informatique, nous trouvons le graphique. Ce domaine qui joue un rôle particulier parmi les « technologies de l'imagerie », permet de créer des contenus numériques sous forme d'objets géométriques 3D, éventuellement bougeant et se déformant au cours du temps, puis rendant visibles en prenant des sortes de photographies de ces objets mathématiques, pouvant être joués en séquence pour montrer un contenu animé.

Mettre la création et la simulation graphique dans la main de tous (via des tablettes permettant de dessiner et de manipuler simplement du contenu 3D, et de le mettre en mouvement) constituerait une véritable révolution des méthodes de travail dans les domaines de l'ingénierie, du design et des sciences.

La 3D a envahi notre quotidien d'une manière explosive au point que beaucoup confondent entre la technique 3D relief (utilisée au cinéma et sur certaines télévisions) qui, grâce à des lunettes spéciales, donne l'impression d'avoir un objet qui se rapproche, et la création d'images de synthèse destinées à être regardées sur un écran.

Faire de la 3D demande d'être minutieux et patient, notre projet s'inscrit dans le cadre de modélisation d'objet 3D, précisément modélisation de visage.

Au cours de nos recherches sur la réalisation d'un produit 3D, nous avons remarqué que la modélisation de tous les projets réalisés dans cette optique se faisait d'une manière manuelle, cette méthode prend un temps considérable (La modélisation d'un visage de base nécessite aux moins quatre heures) et encore nécessite une acquisition d'un grand savoir-faire.

Face à ces contraintes, nous avons pensé à une solution, non complexe, moins couteuse, fiable, rapide, qui consiste à automatiser la modélisation 3D à partir de deux vidéos (Spécialement dans notre cas : modélisation de visage).

Pour mener à bien notre projet, nous avons entrepris notre étude sur trois chapitres présentés comme suit :

- **Chapitre I** : Généralité sur le traitement d'image : présentation des notions fondamentales de traitement d'image.
  
- **Chapitre II** : Modélisation 3D : description des concepts généraux de la 3D et de différentes techniques de modélisation, animation et rendu d'objet 3D.
  
- **Chapitre III** : Conception et Réalisation : description des différentes étapes suivies afin de concevoir l'application et description de l'environnement et des outils de travail, implémentation et test de l'application et son fonctionnement.

# Chapitre I

**Généralité sur le traitement d'image**

## **Introduction :**

L'image est considérée comme l'un des moyens les plus importants que l'homme utilise pour communiquer et exprimer ses motivations.

Le traitement d'une image est l'ensemble des méthodes et techniques opérant sur celle-ci, qui permettent de corriger les dégradations subies lors de l'acquisition ou d'extraction des informations en vue d'une interprétation visuelle ou automatique. Dans notre cas, on va extraire des informations qui vont nous servir à produire un objet 3D.

Ainsi dans ce chapitre, nous rappellerons quelques notions fondamentales de traitement d'images qui serviront à notre étude (image et vidéo).

### **1. Définition d'une image :**

D'une manière générale, on définit une image comme étant un support d'informations. Elle représente une scène qui a été captée par un capteur approprié (appareil photographique, soit par satellite etc...).

Elle peut être considérée comme étant un signal variant dans l'espace suivant les directions horizontal X et vertical Y. Le traitement s'effectue en appliquant tous les outils mathématiques de traitement de signal à deux dimensions. [1]

On compte deux types d'images, c'est qu'on définira ci-dessous, mais nous nous intéressons, dans notre cas d'étude, sur les images de type numérique

#### **1.1. Image analogique :**

Une image analogique peut-être mathématiquement représentée comme une distribution continue d'intensité lumineuse dans un plan  $x$  O  $y$ .

#### **1.2. Image numérique :**

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter. [2]

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques.

Une fois une image en niveaux de gris a été capturée et numérisée, elle est stockée comme un tableau à deux dimensions (matrice) en mémoire de l'ordinateur. [2]



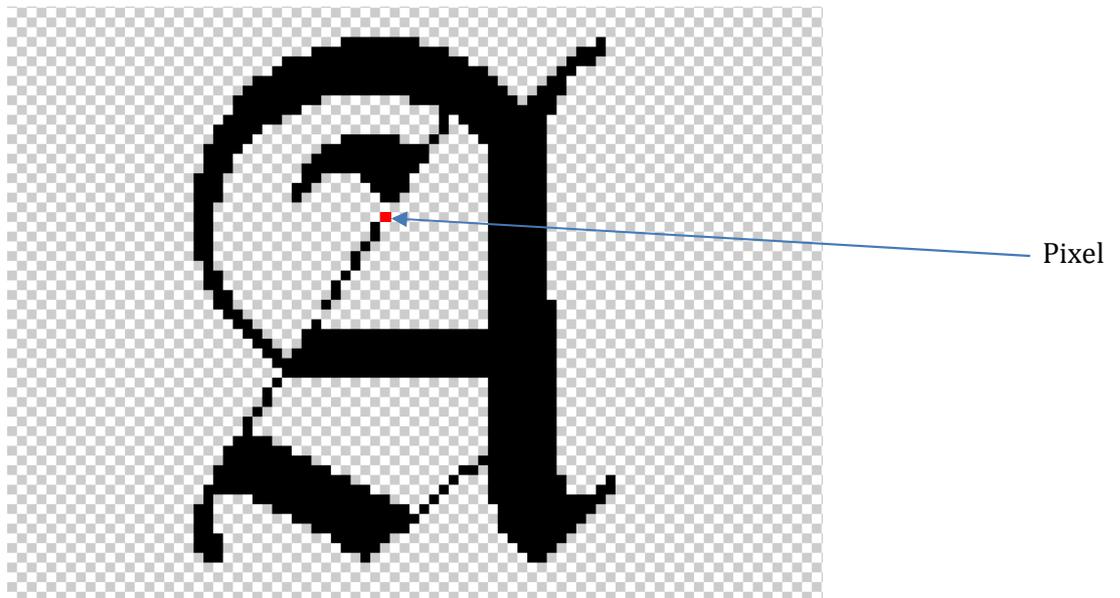
**Figure 1.1. Représentation d'une image numérique.**

### **1.3. Caractéristique d'une image numérique :**

#### **1.3.1. Le pixel :**

Un pixel est l'abréviation de l'expression anglaise (Picture Élément) qu'est le plus petit élément constitutif d'une image numérique. C'est la plus petite unité d'information que peut manipuler les matériels et logiciels d'affichage ou d'impression. C'est une unité de surface permettant de mesurer une image numérique. [3]

La lettre A, par exemple, peut être affichée comme un groupe de pixel dans la figure ci-dessous :



**Figure 1.2. Représentation d'une image par des pixels.**

### **1.3.2. Le contour :**

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

### **1.3.3. Résolution :**

On utilise le mot résolution pour désigner le nombre de pixels affichable horizontalement ou verticalement sur un moniteur, plus grand est ce nombre, meilleur est la résolution. Elle est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

### **1.3.4. Région :**

Ensemble connexe de pixels ayant une ou plusieurs propriétés communes.

### **1.3.5. Dimension :**

Elle définit la taille de l'image qui se présente sous forme d'une matrice dont les éléments représentent des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonne nous donne le nombre total de pixels dans une image.

### **1.3.6. La Luminance :**

La luminance est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'air apparente de celle-ci. Une bonne luminance est caractérisé par :

- ✓ Des images lumineuses (brillantes).
- ✓ Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir.
- ✓ L'absence de parasite. [4]

### **1.3.7. Le bruit :**

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

### **1.3.8. Le contraste :**

Le contraste est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Une augmentation du contraste a pour effet que les couleurs sombres deviennent encore plus sombres et les couleurs claires encore plus claires.

Les différences entre les couleurs claires et les couleurs sombres sont donc accentuées.

La loi de Weber-Fechner définit le contraste  $C_w$  comme : [10]

$$C_w = \frac{L_{Zone} - L_{Fond}}{L_{Fond}}$$

$L_{zone}$  et  $L_{fond}$  désignent respectivement la luminance de l'objet et du fond directement aux abords de l'objet. Ce contraste mesure la perception que l'on a d'un objet de luminance  $L$  placé sur un fond de luminance  $L_{fond}$ .

Le *contraste de Michelson*  $C_m$  (compris entre 0 et 1) est défini comme : [10]

$$C_m = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

#### 1.4. Types d'images numériques :

On distingue deux types d'images : les images vectorielles et les images matricielles.

**1.4.1. Image matricielle :** une image numérique est en fait une matrice (un tableau) de points appelés pixels (pour Picture éléments). Chaque pixel a donc des composantes spatiales dans le plan réel et est représenté par une couleur.

##### ✓ Images 2D

Dans le cas des images à deux dimensions (le plus courant), les points sont appelés pixels. D'un point de vue mathématique, on considère l'image comme une fonction de  $R \times R$  dans  $R$  où le couplet d'entrée est considéré comme une position spatiale.

##### ✓ Images 2D + temps (vidéo), images 3D, images multi résolution

Lorsqu'une image possède une composante temporelle, on parle d'animation.

Dans le cas des images à trois dimensions les points sont appelés des voxels. Ils représentent un volume.

Ces cas sont une généralisation du cas 2D, la dimension supplémentaire représentant respectivement le temps, une dimension spatiale ou une échelle de résolution. D'un point de vue mathématique, il s'agit d'une fonction de  $R \times R \times R$  dans  $R$ .

**1.4.2. Image vectorielle :** Le principe est de représenter, autant que cela est possible de le faire, les données de l'image par des formes géométriques qui vont pouvoir

être décrites d'un point de vue mathématique. Ceci est commode pour les dilatations de l'image en conservant sa qualité.

## 2. Types d'images par couleur :

### 2.1. Image couleur : [11]

Ce sont des images codées en utilisant, en général, le codage des trois couleurs fondamentales (Rouge, Vert, Bleu), on parle alors d'images RVB. Une image couleur RVB contient trois plans de couleurs : rouge, vert et bleu. Chaque plan est codé comme une image au niveau du gris avec des valeurs allant de 0 à 255.



**Figure I.3 : Image couleur**

Il existe différents types d'images couleurs en fonction de nombre de bits utilisés pour le stockage de l'information couleur :

- **Images à 24 bits : [11]**

Le codage de la couleur est réalisé sur 3 octets, chaque octet représente la valeur d'une quantité couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans RVB.

- **Images à 256 couleurs :**

Pour que l'information couleur soit codée sur un octet (pour gagner de la place) au lieu de trois, on utilise une palette (table) de couleurs attachée à l'image. Chaque pixel véhiculera un chiffre compris entre 0 et 255. A chacun de ses chiffres va correspondre une couleur, définie par son code RVB et stockée dans une palette.

Lors de la visualisation de l'image, la correspondance se fait entre le numéro de la couleur affectée à chaque pixel et le code couleur RVB correspondant.

### 2.1.1. Les Modèles couleur :

Il existe de nombreuses façon de percevoir les couleurs d'une image, l'espace de représentation le plus connu est l'espace RGB (Red, Green Blue dont la notation en français correspondante est RVB).

#### a. Le Modèle RGB :

Le modèle RGB est le modèle couleur le plus répandu et qui est utilisé, entre autres par la plupart des caméras et des écrans vidéo. Cet espace tridimensionnel est construit à partir d'une base de trois stimuli couleurs primaires que sont le rouge, le vert et le bleu. La figure suivante est une représentation colorimétrique de cet espace

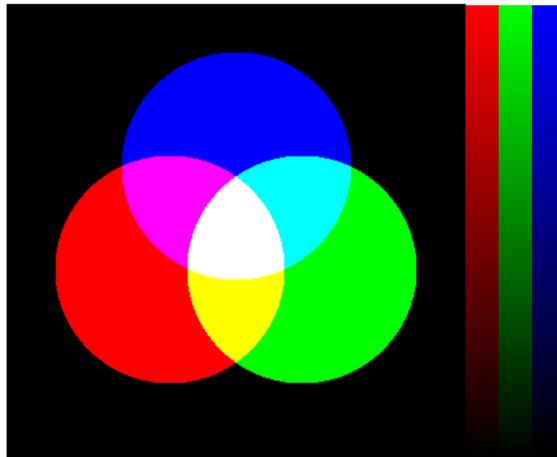


Figure I.4 : Représentation des trois couleurs primaires.

#### b. Le Modèle CMYK :

Le CMJN (cyan, magenta, jaune, noir) (en anglais CMYK, cyan, magenta, yellow, key) appelé plus techniquement quadrichromie est un standard colorimétrique pour l'imprimerie permettant de reproduire un large spectre de couleurs. Ces couleurs sont créées en mélangeant trois encres de bases : le cyan, le magenta et le jaune auxquels on ajoute le noir.

L'absence de ces trois composantes appelant couleurs soustractives donne du blanc tandis que leur ajout donne du noir.

Il est utilisé pour la conception d'un graphisme des projets de pressage ou gravure de CD-ROM et DVD.



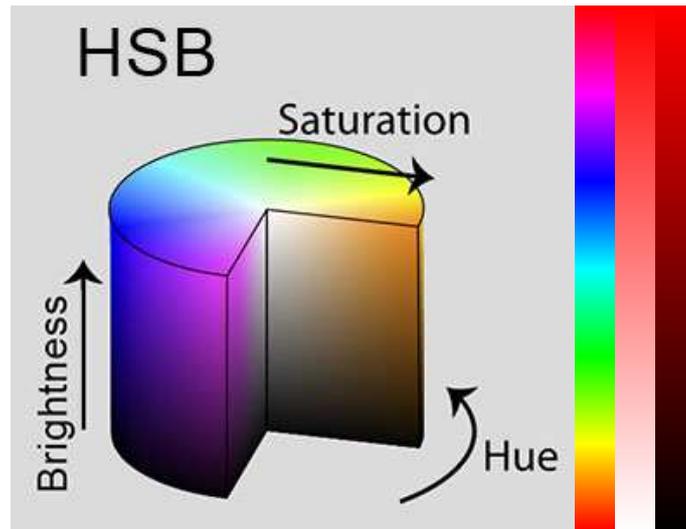
**Figure I.5 : Représentation des quatre couleurs primaires.**

**c. Le Modèle HSB :**

Le modèle TSV pour Teinte Saturation Valeur (en anglais HSV pour Hue Saturation Value ou HSB pour Hue Saturation Brightness) est un système de gestion des couleurs en informatique, particulièrement utilisé dans les applications graphiques.

Il fait partie d'une famille de systèmes basés sur la perception des couleurs, basés sur les trois composantes définies par une approche psychologique et perceptuelle de la couleur : teinte, saturation et valeur.

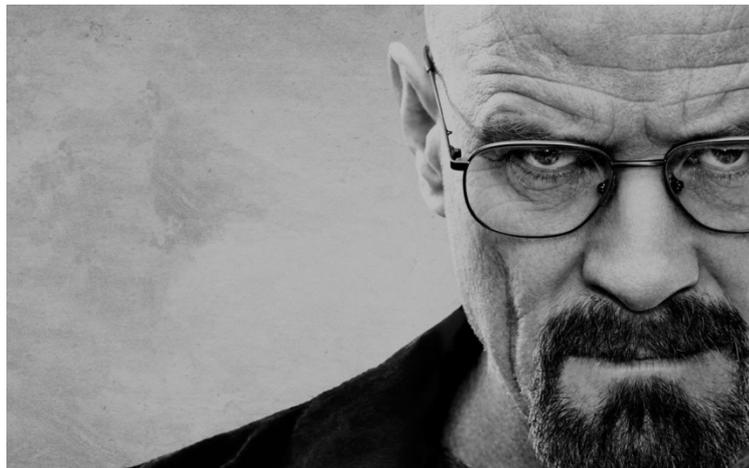
- ✚ La teinte est codée suivant l'angle qui lui correspond sur le cercle des couleurs.
- ✚ La saturation est l'« intensité » de la couleur.
- ✚ La valeur est la « brillance » de la couleur.



**Figure I.6.Représentation du modèle HSB.**

## **2.2. Images en niveau de gris :**

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Pour représenter les images en niveaux de gris, Nous pouvons attribuer à chaque pixel de l'image une valeur correspondante à la quantité de lumière renvoyée. Cette valeur est comprise entre 0 et 255.



**Figure I.7 : Image en niveau de gris.**

### 2.3. Image binaire :

L'image binaire est représentée sous forme d'une matrice dont les éléments valent 0 ou 1. Le niveau 0 représente le noir absolu et le niveau 1 représente le blanc. Dans ce cas un seul bit est utilisé pour coder le niveau du gris. [1]



Figure I.8.Représentation d'une image binaire.

### 3. Formats d'Images :

Le format est la structure utilisée pour représenter une image. Il existe plusieurs formats :

#### 3.1. Format BMP (BitMaP) :

Le format BMP est la représentation d'un pixel et son map en mémoire. Il s'agit d'un format créé par Microsoft. Le poids /taille d'une image au format BMP est très lourde.

#### 3.2. Format PNG :

Format libre de droits. Convient pour tout type d'image, la transparence mais pas les animations. Il existe en 8 bits (type GIF, 256 couleurs) et en 24 bits (type JPEG, millions de couleurs, mais pas de compression).

#### 3.3. Format GIF (GraphicsInterchange Format) :

Développé par CompuServe, GIF est un format de compression utilisant l'algorithme de Compression LZW, on le rencontre aussi sur l'Internet (extension **.gif**). Il possède une palette de couleurs associée et occupe peu d'espace disque mais est limité à 256 couleurs. Il est codé sur 8 bits, les images RGB en milliers de couleurs doivent d'abord être converties en 256 couleurs avant d'être exportées en format GIF.

### **3.4. Format JPEG (Joint Photographic Experts Group) :**

Les images JPEG sont des images de 24 bits peuvent afficher un spectre de 16 millions de couleurs. Ce format accepte les images RVB et CMJN. Le format JPEG présente un bon rapport compression et qualité de l'image. De ce fait, il est fortement utilisé sur le WEB.

Après avoir passé en revue la définition d'image, ses caractéristiques et types, nous présentons dans ce qui suit leur traitement et application.

## **4. Traitement d'image :**

### **4.1. Définition :**

Le traitement d'images est l'ensemble des opérations qui entraînent la modification des images numériques.

Ces modifications portent généralement sur la couleur des pixels. Plusieurs techniques sont utilisées pour manipuler ces données, le but étant l'amélioration de ces données afin d'obtenir une plus grande lisibilité de l'image.

On trouve plusieurs logiciels gratuits ou payants spécifiques pour le traitement des images matricielles ou vectorielles. On peut citer (**GIMP, Adobe Photoshop**) pour le traitement des images bitmap et autres.



**Figure I.9. Adobe Photoshop.**

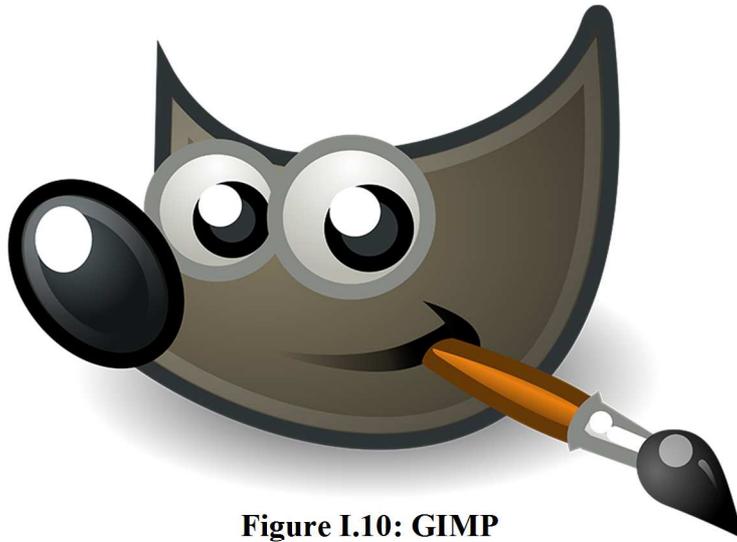


Figure I.10: GIMP

#### 4.2. Les filtres usuels en traitement d'images :

Les techniques de filtrage constituent un volet important des systèmes de traitement d'image.

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. N'entrent pas dans la catégorie du filtrage toutes les transformations de l'image d'origine : zoom, découpage, projections...

##### a) Le filtrage global :

Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Dans cette catégorie on trouve, par exemple, les opérations sur les histogrammes.

##### b) le filtrage local :

Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire. [5]

Puisque notre solution part sur le fait que les images que nous traitons sont restituées depuis des vidéos, qui n'est qu'un enregistrement d'un point laser scannant l'objet réel à modéliser en 3D, nous proposons alors dans la suite de ce chapitre de définir le concept vidéo.

### 5. La Vidéo :

#### 5.1. Définition de la Vidéo :

La **vidéo** regroupe l'ensemble des techniques permettant l'enregistrement ainsi que la restitution d'images animées, accompagnées ou non de son, sur un support électronique et non de type photochimique.

Le mot « vidéo » vient du latin « *video* » qui signifie : « je vois ». C'est l'apocope de vidéophonie ou vidéogramme.

Les affichages informatiques disposent de résolutions spécifiques et de modes de balayage tout aussi spécifiques. Les micro-ordinateurs 8 bits et les premiers 16 et 32 bits étaient destinés à un branchement sur un appareil de télévision, leur sortie vidéo était donc en 740/50 ou 525/60. Les normes utilisées sur PC sont différentes :

- EGA 640×350×16c à 60 Hz
- VGA 640×480×16c à 60 Hz

La fréquence de balayage est comprise entre 50 Hz et plus de 120 Hz. Il n'est pas possible de brancher un ordinateur directement sur un téléviseur, car les fréquences de balayage sont différentes, cela peut entraîner la destruction du téléviseur, donc l'encodeur couleur (PAL, SECAM ou NTSC) est nécessaire pour réaliser un enregistrement vidéo d'une image informatique.

#### 5.2. Formats de vidéo :

##### 5.2.1. AVI (Audio Video Interleave) :

Le format **AVI** est le format de fichier vidéo créé par Microsoft. Sigle de « Audio Vidéo Interleave », son nom d'origine du format est en réalité " Vidéo for Windows ".

Ce format prévoit une définition d'image maximale de 320 x 240 pixels et une cadence maximale de 30 images/seconde, mais en contrepartie, la compression et la décompression des images sont effectuées par logiciel de lecture que l'on utilise.

L'avantage est que les applications multimédias faisant appel au format AVI peuvent toucher une large base d'utilisateurs disposant d'un PC en configuration standard. Cependant les fichiers sont très lourds et demande une configuration matérielle assez performante.

### 5.2.2. MPEG (Motion Picture Expert Group) :

Dans le format MPEG, on distingue plusieurs méthodes d'encodage (compression et décompression de données, qu'on appelle aussi codec), qui permettent d'optimiser le rapport qualité/taille. De plus, la notion de définition est directement liée à la taille du fichier. Pour 10 minutes de vidéo, on constate :

- **MPEG1** format 360 X 240 (pixels), environ 100 Mo/10 mn, appelé aussi format VCD (vidéo CD)
- **MPEG2** format 540 X 360 (pixels), environ 500 Mo/10 mn, appelé aussi format SVCD (super vidéo CD)
- **MPEG2** format 760 X 540 (pixels), environ 700 Mo/10 mn, appelé aussi format DVD (digital versatildisk)

On constate la lourdeur considérable de ces fichiers, qu'il est très difficile d'associer à internet.

### 5.3. WMV / .wmv

**5.3.1. H264 / MPEG 4 AVC** : C'est le Codec utilisé pour la haute définition.

## 6. Tracking (tracking 2D) :

Le principe du tracking 2D est de **recupérer des informations de transformation** (position (points), rotation et échelle) d'un élément faisant partie d'une vidéo et de les convertir en animation utilisable.

Notons qu'il existe plusieurs technique pour le traitement de vidéo, mais nous intéressant à la technique du tracking, car elle répond parfaitement à l'objectif qui vise, comme on l'a déjà dit, à réaliser une production d'un objet 3D.

Pour ce faire, on utilisera deux vidéos qui sont en format (avi / mp4), en appliquant le tracking sur la position d'un point laser.

### **Conclusion :**

Dans ce chapitre nous avons présenté des généralités sur le traitement d'images puis nous avons abordé les concepts généraux concernant l'image (leurs différents types et formats ...etc.) et la vidéo (leurs formats).

Le principe de traitement de vidéo est le même avec celui de l'image, car une vidéo est définie avec une séquence d'image et un son (le son est facultatif).

Dans le chapitre suivant nous allons détailler les différentes techniques de modélisation 3D et les concepts généraux sur la 3D.

# Chapitre II

Modélisation 3D

## Chapitre II

### Modélisation 3D

#### **Introduction :**

Trois dimensions ou tridimensionnel ou 3D sont des expressions qui caractérisent l'espace qui nous entoure, tel que perçu par notre vision, en termes de largeur, hauteur et profondeur, et la modélisation 3D qui est devenue à l'air actuel indispensable. Elle est très utilisée en art numérique de l'industrie du film, de jeux vidéo, et également dans de nombreuses autres applications : visualisation architecturale, visualisation médicale, simulations diverses, économiseurs d'écrans...etc

La modélisation 3D est utilisée essentiellement pour simuler des scènes imaginaires, des périodes ancestrales ou des phénomènes scientifiques qui révèlent de la fiction et de la précision. Son avantage principal est qu'elle permet d'obtenir une très grande qualité d'image et un réalisme poussé.

Rappelons que notre travail consiste en la production d'un objet 3D à partir des images extraites depuis des prises vidéos capté sur des figures.

En conséquence dans ce deuxième chapitre, nous dressons les étapes de production de l'objet 3D que nous classons en trois. La première étape consiste à la description de différentes techniques de modélisation de l'objet 3D, la deuxième étape comporte les techniques de l'animation d'un objet 3D et la dernière étape comprend les différentes techniques de rendu. [6], [7], [8]

#### **1. Les étapes de production d'un Objet 3D :**

Pour réaliser un produit 3D on suit trois étapes : **Modélisation, Animation, Rendu.**

##### **1.1. Modélisation 3D :**

La modélisation 3D est l'étape en infographie tridimensionnelle qui consiste à créer des objets trois dimensions par ajout, soustraction et modifications de ses constituants, en utilisant un logiciel de modélisation 3D.

Il existe différents techniques de modélisation 3D on citera :

### 1.1.1. Modélisation Polygonale :

Dans l'infographie 3D , la modélisation polygonale est une approche pour la modélisation des objets en représentant ou un alignement de leurs surfaces en utilisant des polygones . La modélisation polygonale est bien adapté à ligne de balayage rendu et est donc la méthode de choix pour les graphiques en temps réel de l'ordinateur .

#### 1.1.1.1. Construction de maillages polygonaux :

En utilisant des sommets et de faces et en peut utiliser une variété d'outils, une grande variété de logiciels graphiques 3D paquets.

Bien qu'il soit possible de construire un maillage en spécifiant manuellement sommets et de faces, il est beaucoup plus commun de construire mailles en utilisant une variété d'outils. Une grande variété de logiciels graphiques 3D paquets sont disponibles pour une utilisation dans la construction de maillages polygonaux.

Il existe deux méthodes :

#### a) La modélisation de la boîte :

La plus populaire qui utilise deux outils simples.

- **L'outil de subdivision** : divise faces et arêtes en petits morceaux en ajoutant de nouveaux sommets. Par exemple, un carré serait subdivisé en ajoutant un sommet dans le centre et une sur chaque bord, créant ainsi quatre petits carrés.
- **L'outil d'extrusion** : est appliqué à un visage ou un groupe de visages. Il crée une nouvelle face de la même taille et la forme qui est relié à chacun des bords existants par une face. Ainsi, effectuer l'opération **d'extrusion** sur un visage carré serait créé un cube reliée à la surface à l'emplacement du visage.

#### b) modélisation de gonflage (modélisation d'extrusion)

Dans cette méthode, en crée une forme 2D qui retrace le contour d'un objet à partir d'une photographie ou d'un dessin. Ensuite en utilise une seconde image de l'objet à partir d'un angle différent et expulse la forme 2D en 3D, de nouveau après le contour de la forme. Cette méthode est particulièrement commune pour créer des visages et des têtes. En général, en modélisent la moitié de la tête, puis dupliquer les sommets,

inverser leur emplacement par rapport à un plan, et relier les deux morceaux ensemble. Cela garantit que le modèle sera symétrique.

Un autre procédé commun de créer un maillage polygonal est en reliant ensemble des primitives différentes, qui sont des maillages polygonaux prédéfinis créés par l'environnement de modélisation. Primitives communs incluent :

- Cubes
- Pyramides
- Cylindres
- Primitives 2D, tels que des carrés, des triangles, et les disques
- Primitives spécialisées ou ésotériques, comme l' Utah Théière ou Suzanne, de Blender singe mascotte.
- Sphères sont couramment représentées dans une des deux façons suivantes :
  - ✓ **Icosphères** sont icosahedrons qui possèdent un nombre suffisant de triangles pour ressembler à une sphère.
  - ✓ **sphères UV** sont composés de quads, et la grille ressemblent vu sur certains globes - quads sont plus grandes à proximité de«équateur» de la sphère et inférieure près des "pôles", éventuellement se terminant en un sommet unique.

### 1.1.1.2. Mode d'Utilisation :

Il y a un très grand nombre d'opérations qui peuvent être effectuées sur des maillages polygonaux. Certains d'entre eux correspondent approximativement aux manipulations du monde réel d'objets 3D, tandis que d'autres ne le font pas.

Les opérations de maillage polygonal sont les suivantes :

- Créations - Créer une nouvelle géométrie de quelque autre objet mathématique**
  - Loft* - générer un maillage en balayant une forme le long d'un chemin
  - Extruder* - même que mezzanine, sauf le chemin est toujours une ligne
  - Revolve* - générer un maillage en tournant (rotation) une forme autour d'un axe
  - Marching cubes* - algorithme pour construire un maillage d'une fonction implicite
- Créations binaires - Créer une nouvelle maille à partir d'une opération binaire de deux autres mailles**

Ajouter - plus booléenne de deux mailles

Soustraire - soustraction booléenne de deux mailles

Intersection - intersection booléenne

Union - Union booléenne de deux mailles

Fixez - joindre une maille à l'autre (en enlevant les surfaces intérieures)

Chanfrein - créer une surface biseautée qui reliait deux surfaces en douceur

**c. Déformations - déplacer uniquement les sommets d'un maillage**

Déformer - déplacer systématiquement sommets (selon certaines fonctions ou des règles)

Déformer pondéré - Déplacement de sommets sur la base de poids localisées par sommet

Morph - déplacer les sommets en douceur entre une source et cible maille

Bend - déplacer sommets pour "plier" l'objet

Twist - déplacer sommets à "tordre" l'objet

**d. Manipulations - modifier la géométrie du maillage, mais pas nécessairement topologie**

Déplacer - introduire la géométrie supplémentaire basé sur une "carte de déplacement" de la surface

Simplifiez - retirer systématiquement et sommets moyenne

Subdiviser - lisser un maillage de cours en subdivisant le maillage (Catmull-Clark, etc.)

Convex Hull - générer un autre maillage qui entoure minimalement un maillage donné

Couper - créer un trou dans une surface de maille

Piquez - fermer un trou dans une surface de maille

**1.1.1.3. Les Avantages et les Inconvénients :**

**Les Inconvénients :**

- Les polygones sont incapables de représenter avec précision les surfaces courbes.
- L'utilisation de modèles complexes a un coût réduit de la vitesse. Dans la conversion de ligne de balayage, chaque polygone doit être convertie et affichée, indépendamment de la taille.

**Les Avantages :**

- Polygones sont plus rapides que d'autres représentations (Bien que d'une carte graphique moderne peut montrer une scène très détaillée à un taux de trame de 60 images par seconde ou plus).
- Utilisé en production cinématique

**1.1.2. Modélisation par Courbes (NURBS) :**

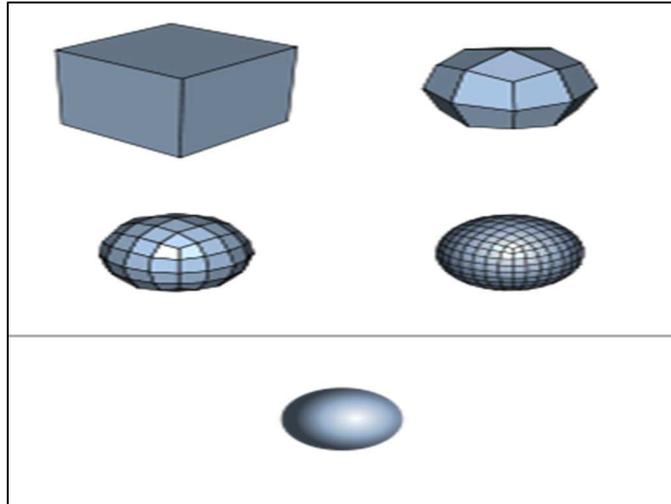
La modélisation par NURBS (Nouniform rational basic spline/Spline basique rationnelle non uniforme) consiste en un réseau de courbes créé grâce à des points de contrôles (control vertices). L'interpolation des courbes entre ces points peut se faire automatiquement selon un algorithme NURBS, par la manipulation de tangentes de courbe de Bézier, ou encore par modification des paramètres d'interpolation.

La modélisation par courbe se base sur un maillage adaptatif, conçu pour adapter ses subdivisions à la complexité des courbes dans une région donnée. Une surface parfaitement plane aura un nombre de subdivision très faible ou nul. C'est la technique de modélisation la plus précise. C'est la raison pour laquelle elle est massivement utilisée en architecture et en CAO (Conception Assisté par Ordinateur) industrielle, lorsque le souci de précision prime (notamment lorsque les modèles 3D servent de référence pour les machines-outils).

**1.1.3. Modélisation par Subdivision de surface :**

Cette méthode consiste à accélérer le processus grâce à la subdivision automatique d'une partie de la surface, regroupe un peu des deux méthodes classiques (polygonale et NURBS) ; Ceci permet d'ajouter des détails à certains endroits uniquement, sans se soucier du nombre de faces comprises sur la globalité de l'objet.

Cette figure ces dessus va montrer la subdivision de surface.



**Figure II.1 : Subdivision de surface**

#### **1.1.4. Modélisation par Géométrie :**

En infographie, la géométrie de construction de solides (CSG en anglais: "Constructive Solid Geometry") est une branche de la modélisation des solides (ou modélisation 3D). cette technique concerne la représentation d'un objet solide comme combinaison d'objets solides simples (cylindre, sphère, cône, tore, etc...) à l'aide d'opérateurs géométriques booléens (union, intersection, soustraction).

La modélisation géométrique des solides procède de 2 méthodes :

1. La CSG (dite aussi "modélisation solide" ou "modélisation volumique"),
2. La B-Rep ("BoundaryRepresentation" dite aussi "modélisation surfacique").

##### **1.1.4.1. Utilisation :**

Il existe deux opérations (booléennes et géométriques).

##### **a. Les Transformations géométriques :**

Ce sont les transformations géométriques classiques :

- Translation
- Rotation
- Homothétie

**b. Les Opérations booléennes :(Union, Différence, Intersection)**

- **Union (addition) :**

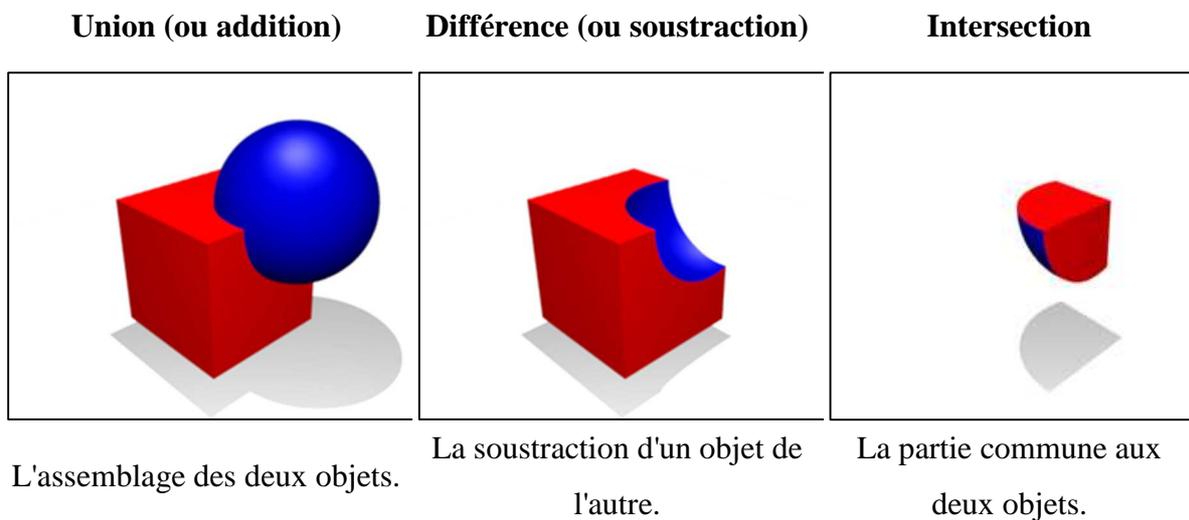
Le résultat est l'assemblage des deux objets ou plus.(Voir la figure 2).

- **Différence (ou soustraction) :**

Le résultat est le premier objet moins la partie commune avec le second.(Voir la figure 2).

- **Intersection :**

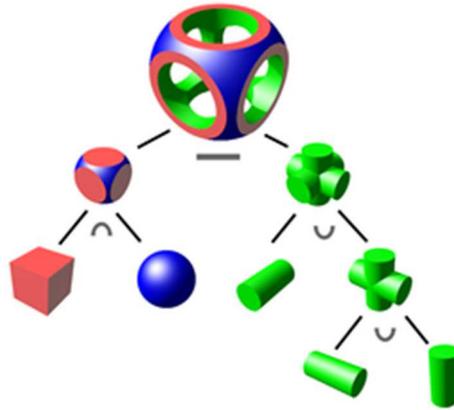
Le résultat est la partie commune aux deux objets.(Voir la figure 2)



**Figure II.2 : Les Opérations booléennes pour la modélisation géométrie.**

**Structure :**

La géométrie de construction de solides est stockée sous une forme arborescente (arbre) qui décrit : l'opération et les éléments manipulés. L'utilisation d'arbre permet de faire très facilement des modifications.



Les objets en Géométrie de construction de solides peuvent être représentés par des arbres binaires, dans lesquels les feuilles représentent les primitives, et les nœuds représentent les opérations.

∩: intersection

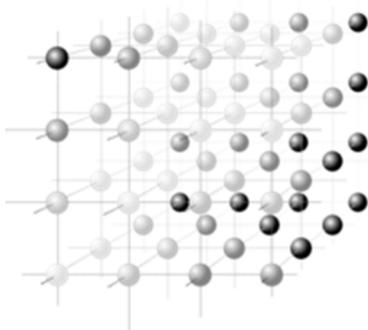
∪: union

−: différence

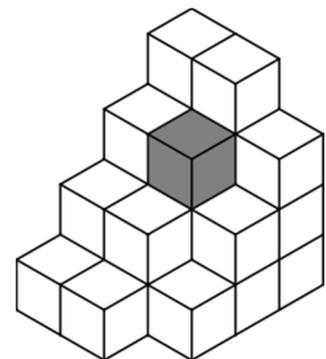
**Figure II.3 :** arbre binaires représentant la construction de l'objet en géométrie

**1.1.5. Modélisation Volumique :** cette méthode utilise des voxels

**1.1.5.1. Voxel :(volumetric pixel)** est un pixel en 3D.



**Figure II.4 :** Une matrice de voxels en nuances de gris.



**Figure II.5 :** Une série de voxels dans une pile avec un seul voxel

❖ **Avantages**

La 3D permet souvent de diminuer (ou d'éviter) l'étape coûteuse (et fastidieuse) de la maquette. Ceci grâce aux caractéristiques suivantes :

- ✓ les visualisations possibles : Il est possible de visualiser l'objet sous différents angles : en peu définit les directions d'observation qu'il désire, et ceci d'une manière simple.
- ✓ l'ouverture du modèle : permet la connexion avec des logiciels spécifiques en vue de réaliser différentes applications ou opérations telles que calculs de structures, élaboration de gammes d'usinage, programmation de machines-outils à commande numérique...

### ❖ **Inconvénients**

Les inconvénients ne sont pas absents, surtout en ce qui concerne

- ✓ Les coûts (Logiciels/Matériels)
- ✓ Letemps de production
- ✓ Les applications 3D ont une gamme différente du 2D. Le choix de ces applications doit être sérieusement étudié sous peine de se révéler très pénalisant pour l'utilisateur, essentiellement en temps et en motivation.

Après avoir vue les techniques de modélisation en a choisi la technique modélisation polygonale par apport à ces avantages qui est la facilité d'utilisation et l'efficacité et moindre erreur, dans notre travail.

### • **Remarque :**

On peut utiliser un scanner pour modéliser un objet. Sauf que le scanner est excessivement couteux et n'est pas opté pour des grands objets (Construction, Véhicules ...).

Après avoir vue les déférentes technique de modélisation, les avantages, mode d'utilisation et efficacité, notre choix c'est porté à utiliser la technique de Modélisation Polygonale pour notre travail, par ce qu'elle répond bien à notre objectif.

### **1.2. Animation 3D :**

Est une technique d'animation numérique équivalant à l'animation en volume dans un monde virtuel. Les animations 3D sont des œuvres artistiques complexes dont dépend la maîtrise de multiples procédés. L'illusion se façonne autour de la création d'un univers et de personnages à la fois surprenants et d'une réalité presque palpable. Derrière la puissance des ordinateurs et des logiciels.

### **1.2.1. Les techniques d'Animation 3D :**

#### **1.2.1.1. L'Animation d'images clés ou keyframe :**

L'animation d'images clés, également appelée animation par « keyframe », est une technique qui permet de définir les images clés d'un objet 3D, ou d'un paramètre de celui-ci, et de calculer automatiquement par interpolation de l'information contenue dans les images clés, toutes les images intermédiaires d'une séquence animée.

#### **1.2.1.2. L'Animation avec des trajectoires :**

Une trajectoire est une ligne (droite ou courbe) qui représente le parcours à suivre par un objet 3D déterminé. Les trajectoires sont composées de points et chaque point représente une position dans l'espace et un instant donné dans le temps. Animer avec des trajectoires permet de penser le mouvement en termes de « va d'ici à là et suit cette trajectoire ». Cette technique permet à l'animateur de définir des mouvements rapides nécessitant des rotations et translations en même temps.

L'animation avec des trajectoires est spécialement utilisée pour définir avec précision le mouvement des personnages 3D volants ou aquatiques parce qu'ils se déplacent suivant un chemin dans un espace tridimensionnel.

Elle est utilisée pour animer les déplacements de la caméra virtuelle le long d'une trajectoire prédéfinie dans l'espace 3D.

#### **1.2.1.3. L'animation de marionnettes virtuelles :**

Une marionnette virtuelle est un personnage 3D doté d'une structure hiérarchique ou d'un squelette virtuel lui permettant de bouger et d'accomplir des mouvements aussi complexes que la marche.

Les marionnettes virtuelles sont manipulées de la même façon qu'une marionnette traditionnelle grâce à une structure ou « armature » articulée interne, invisible à l'œil nu. L'animation de marionnettes virtuelles requiert la mise en place d'une structure hiérarchique et/ ou d'un squelette virtuel.

### **1) Les structures hiérarchiques :**

L'assemblage des marionnettes virtuelles sous forme d'une structure hiérarchique permet de lier les différents modèles 3D qui la constituent et de concevoir la logique de transmission du mouvement. Les liens établis entre les différents modèles 3D sont dits hiérarchiques dans la mesure où chaque élément 3D situé au plus haut niveau de l'organisation, contrôle tous ceux qui se trouvent sous leur influence et non l'inverse. Le lien hiérarchique permet d'établir les relations de pouvoir ou de dépendance entre chaque modèle 3D de la structure et détermine le comportement des objets lorsqu'ils sont animés. Cette technique de construction qui permet d'exprimer la dépendance d'éléments vis-à-vis d'autres éléments.

Convient particulièrement bien à l'animation de structures articulées. Une structure articulée est constituée de plusieurs objets assemblés et interconnectés entre eux. Les modèles 3D faisant partie d'une structure hiérarchique ont des niveaux de priorité ou d'importance bien définis. Pour désigner leur position hiérarchique, on utilise quelques termes empruntés à la généalogie. De telle sorte que le modèle 3D contrôlant, se trouvant tout en haut de la hiérarchie, s'appelle « parent » et les éléments contrôlés, se trouvant en dessous, s'appellent « enfants » et « petits enfants ». La construction et l'animation de structures articulées avec des liens hiérarchiques sont à la base de la création des personnages animés en 3D.

### **2) Les squelettes virtuels :**

Le squelette virtuel est constitué d'arêtes (les os) et de joints (les articulations), tel qu'un squelette humain. La simulation d'un squelette permet d'animer des marionnettes 3D anthropomorphes et zoomorphes modélisées avec un maillage d'un seul tenant (polygonaux ou à base de surfaces soudées). Une fois la marionnette virtuelle modélisée, il reste à lui associer un squelette construit en fonction de sa morphologie.

#### **1.2.1.4. L'animation de formes :**

L'animation de formes est une technique permettant la déformation voire la métamorphose entre deux ou plusieurs objets 3D. Elle consiste à définir, à chaque image clé, une forme de départ et une forme de fin. L'ordinateur se charge ensuite de calculer les formes intermédiaires par interpolation. L'animation résultante est la chorégraphie de cette transformation. Elle permet de créer soit de simples déformations, soit de radicales

transformations morphologiques. Lorsque l'interpolation est réalisée entre différentes formes d'un même personnage 3D.

- ✓ La déformation
- ✓ La métamorphose

### **1.2.2. L'hybridation du mouvement :**

L'hybridation du mouvement résulte du couplage de deux techniques numériques distinctes : la « capture de mouvement » et la « synthèse de mouvement ». Par conséquent, les mouvements résultants de cette hybridation ont deux sources différentes : l'une réelle, issue de la capture numérique des mouvements d'êtres humains (acteurs, danseurs, artistes divers) et l'autre abstraite, issue des modèles logico-mathématiques (cinématique, dynamique, comportementaux, etc.). L'hybridation numérique de ces deux mouvements d'origines distincts se réalise à travers un processus computationnel. Le résultat est une animation hybride.

#### **1) La rotoscopie tridimensionnelle :**

La « rotoscopie tridimensionnelle » désigne un ensemble de méthodes inventé pour l'enregistrement du mouvement d'un acteur réel et la transmission de ce même mouvement à un personnage virtuel. La « rotoscopie tridimensionnelle » a précédé la capture de mouvement.

La rotoscopie, c'est-à-dire, le principe qui consiste à recueillir les informations des mouvements d'un acteur réel pour animer ensuite un personnage inanimé.

#### **2) La capture de mouvement (motion capture ou mocap en anglais) :**

Le processus de capture de mouvement consiste à enregistrer les mouvements d'un acteur réel au moyen de combinaisons électromécaniques, de capteurs électromagnétiques, de marqueurs optiques et à les convertir sous forme de coordonnées XYZ. Ces coordonnées tridimensionnelles sont ensuite transmises au personnage virtuel afin que ses mouvements soient dotés de toutes les subtilités d'interprétation de l'acteur.

La capture de mouvement est un procédé d'acquisition de données tridimensionnelles de mouvement extraites du monde réel. Cette technique est utilisée dans les secteurs médicaux, industriels et créatifs. Dans le domaine de l'animation, la capture de mouvement est utilisée pour l'animation réaliste de personnages 3D.

### **1.2.3. L'animation procédurale :**

L'animation procédurale englobe toutes les techniques qui permettent d'animer automatiquement des objets 3D durs (solides), mous (tissus, vagues, etc.) ou flous (nuages, fumée, etc.) à partir de la description de leurs propriétés physiques (masse, poids, inertie, etc.) et de la simulation des forces (gravité, friction, etc.) auxquelles ils vont être soumis dans un espace-temps.

L'animation procédurale, en simulant les phénomènes physiques auxquels les objets réels sont soumis, permet d'insuffler un grand réalisme aux mouvements des objets virtuels. Parmi les principales techniques d'animation procédurale citons : la dynamique du mouvement et les systèmes de particules.

#### **❖ Les systèmes de particules :**

Un système de particules est une technique d'animation permettant de générer automatiquement le mouvement et l'apparence d'« objets flous » ou « fuzzy objets » dont les formes irrégulières et complexes évoluent continuellement dans le temps. Cette technique est particulièrement utilisée pour la modélisation et l'animation des phénomènes naturels tels que le vent, la neige, le feu et l'eau.

### **1.3. Rendu :**

Le rendu est un processus informatique calculant l'image 2D (équivalent d'une photographie) d'une scène créée dans un logiciel de modélisation 3D comportant à la fois des objets et des sources de lumière et vue d'un point de vue précis. Ce processus peut faire appel à différentes techniques de calcul d'éclairage plus ou moins élaborées, dont les plus connues sont le rendu au lancer de rayon et en radiosités ; le rendu est généré par un/des programme(s) qu'on appelle Moteur de rendu 3D, intégré(s) ou non au logiciel de modélisation.

Se décompose à son tour en plusieurs phases :

- 1) Calcul de l'éclairage (on parle d'illumination)
- 2) Projection dans le plan d'observation.
- 3) Dessin à proprement parler avec application éventuelle de textures

### **1.3.1. Moteur de rendu :**

Les moteurs de rendu 3D sont des logiciels ou des cartes graphiques spéciales (matériel) qui calculent une ou plusieurs images 3D en y restituant non seulement la projection 3D mais surtout tous les effets d'éclairage (ombres, réflexions, etc.).

C'est ce que l'on appelle le rendu. L'opération assez complexe, dépend du type de moteur de rendu. Le moteur de rendu 3D analyse les éléments d'une image numérisée (couleurs, intensité et type de la lumière, ombres et leurs combinaisons, etc), image censée être vue par une "caméra" virtuelle dont les coordonnées x y z déterminent l'angle de vue et la position des objets.

### **1.3.2. Types de moteurs :**

#### **a) Moteurs de rendu logiciels :**

Les moteurs logiciels sont des programmes qui vont eux-mêmes calculer l'image, avec l'aide du **processeur** central (*CPU*) et de la **mémoire vive** (*RAM*).

#### **b) Moteurs de rendu accélérés matériellement :**

Les moteurs matériels sont intégrés à des cartes graphiques haut de gamme. Ces moteurs sont nettement plus rapides que les moteurs logiciels, étant donné qu'ici le processeur central (*CPU*) et la mémoire vive (*RAM*) ne font plus rien et laissent tout le travail au processeur graphique (*GPU*), dans lequel les calculs sont implantés. Ces cartes sont évidemment plus chères que les cartes graphiques habituelles.

#### **c) Le moteur de rendu Mental Ray :**

Le rendu mental ray de mental images est un rendu à usage général capable de générer des simulations physiquement correctes d'effets d'éclairage tels que des réflexions et des réfractions par lancer de rayons, des réverbérations et une illumination globale.

Mental ray est devenu un standard du rendu photoréaliste dans l'industrie du cinéma, des effets visuels et de la conception.

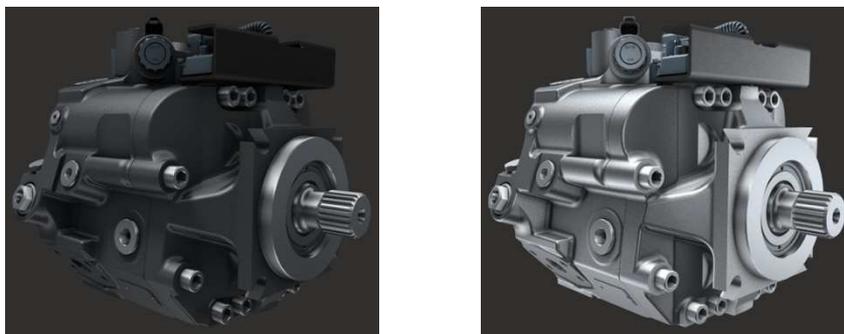


**Figure II.6 : Effet rendu mental ray**

Qui dit rendu dit **éclairage**. Il existe plusieurs types d'éclairages : l'éclairage "simple", et l'éclairage avancé, autrement dit, la GI (**Global Illumination**). La GI demande énormément de temps de calcul comparé à l'éclairage simple, mais on obtient un résultat beaucoup plus réaliste.

Avec la GI, on peut reproduire des effets de la physique comme les caustiques (quand la lumière passe à travers un verre d'eau, ou mieux, le petit "maillage" qu'il y a sur le sable, sous l'eau bien chaude des caraïbes...).

Tous les rendus en 3D seront une affaire de lumière : c'est elle qui donnera une crédibilité à la scène.



**Figure II.7 : Eclairage.**

## 2. Formats de sauvegarde de l'Objet 3D :

Il existe plusieurs formats disponibles pour stocker les données des scènes 3D. Les plus populaires sont :

- .3ds, .max, Qui est associé avec 3D Studio Max
- .blend, qui est associé à Blender
- .dae
- .dxf, .dwg, .dwf, associé à AutoCAD
- .ply utilisé pour stocker des données de scanners 3D
- .obj

**Remarque :** On peut également sauvegarder les données sur des fichiers script comme dans notre cas d'étude (MaxScript sous extension '.ms').

## 3. Visualisation de scène :

Chaque fenêtre 3D supporte deux systèmes de projection : la projection orthographique et la projection en perspective.

### 3.1. La vue perspective :

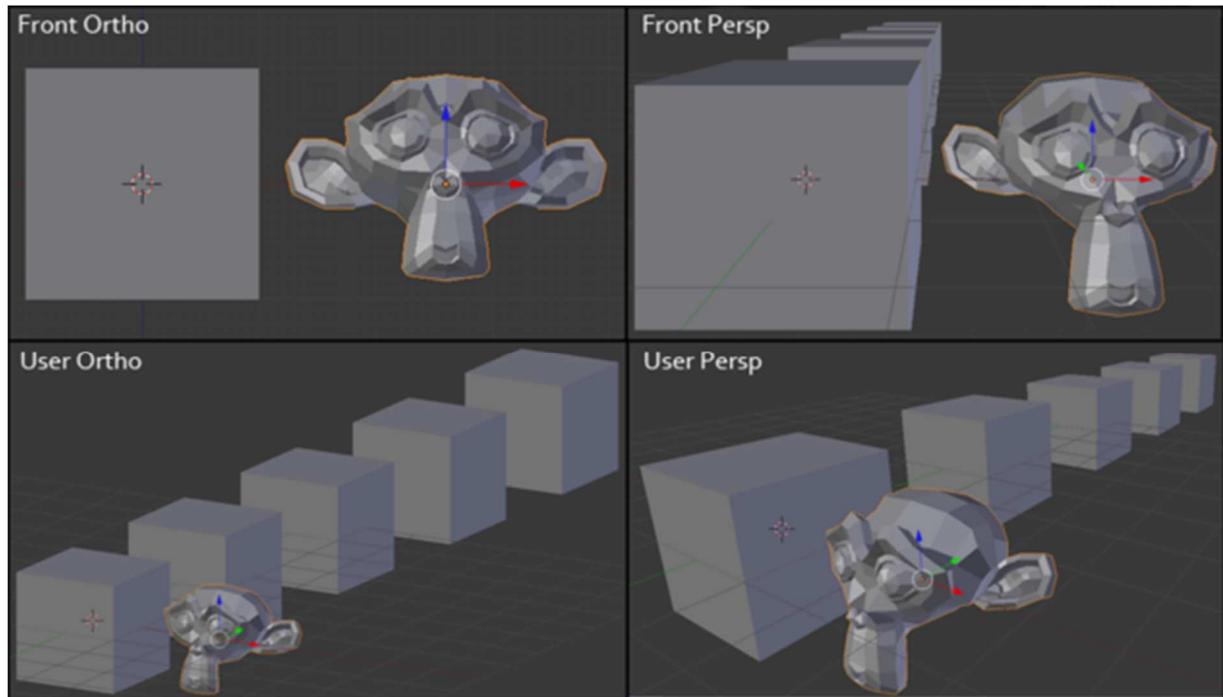
La vision en perspective est la vue la plus réaliste ; plus on regarde de loin, plus les objets apparaissent petits.

Une vue en perspective est géométriquement construite de cette façon : vous observez une scène en 3D et vous êtes placé au point O. La scène en perspective 2D est construite en plaçant un plan, une feuille de papier où la scène 2D doit être représentée comme vue du point O, perpendiculairement à la direction du regard. Pour chaque point P de la scène 3D une ligne PO est tracée passant par O et P. Le point S intersection de cette ligne PO et le plan est la projection en perspective de ce point. En projetant tous les points P de la scène, vous obtenez une vue en perspective.

### 3.2. La vision orthonormée :

La vision orthonormée est très utile et plus technique car elle permet de dessiner précisément et de mieux percevoir les distances et proportions des objets : c'est comme regarder la scène d'un point éloigné à l'infini.

Dans une projection orthographique, vous avez un sens d'observation, mais pas de point d'observation O. La ligne est alors tirée à partir du point P de sorte qu'elle soit parallèle à la direction du regard. L'intersection S entre la ligne et le plan est la projection orthogonale du point P. Et en projetant tous les points P de la scène, vous obtenez la vue orthographique.



**Figure II.8 : Projection orthographique et perspective.**

Notre œil est habitué à la visualisation en perspective, les objets éloignés paraissant plus petits. La projection orthographique semble souvent un peu bizarre au premier abord, parce que les objets restent de la même taille quelle que soit leur distance. C'est comme regarder la scène à partir d'un point infiniment éloigné. Néanmoins, la visualisation orthographique est très utile (elle est utilisée par défaut dans la plupart des applications 3D spécialement dans les fenêtres secondaires), car elle fournit une approche plus "technique" de la scène, ce qui la rend plus facile à dessiner et à juger de ses proportions.

#### **4. Quelques outils de base de la modélisation :**

##### **4.1. Les modificateurs :**

Éditer les maillages c'est bien, mais c'est long et fastidieux. Et en plus, on ne peut pas revenir sur les modifications qu'on a apportées aux objets, sans compter qu'on ne peut pas non plus animer ces maillages. La solution, ce sont les modificateurs !

Les modificateurs comme leur nom l'indique, modifient la structure géométrique d'un objet, en la déformant d'une certaine façon ; c'est effectuer des transformations réversibles sur cet objet. Par exemple, lorsque vous appliquez un modificateur Effiler à l'extrémité d'un cylindre, les sommets proches de l'extrémité se rapprochent.

Une fois que votre modificateur est appliqué, il faut aller le régler dans la pile. La zone où on sélectionne les modificateurs s'appelle la pile, c'est tout simplement parce que quand on a un certain nombre de modifier appliqués, la liste forme une pile (on empile les modificateurs les uns sur les autres comme des filtres).

Il est possible d'effectuer les opérations suivantes via la pile :

- Masquer les effets du modificateur (ampoule)
- Supprimer un modificateur (corbeille)
- Modifier l'ordre d'empilement

On citera quelques modificateurs :

### a. Le modificateur Révolution

Le modificateur Révolution crée un objet 3D en faisant pivoter une forme ou une courbe NURBS autour d'un axe.



**Figure II.9 : Effet modificateur révolution.**

### b. Le modificateur Courbure

Le modificateur Courbure permet d'arrondir la sélection courante jusqu'à 360 degrés sur un seul axe, ce qui produit une courbure uniforme dans la géométrie d'un objet. Ainsi

contrôler l'angle et la direction de la courbure sur chacun des trois axes, ou limiter la courbure à une section de la géométrie.



**Figure II.10 : Effet du modificateur Courbure.**

**c. Le modificateur Lissage :**

Le modificateur Lissage rapide, à l'instar du modificateur Lissage maillage, permet de lisser la géométrie de votre scène.

Le modificateur lissage rapide permet de subdiviser la géométrie tout en interpolant les angles des nouvelles faces aux coins et aux arêtes, et d'appliquer un groupe de lissage unique à toutes les faces de l'objet. Il arrondit les coins et les arêtes comme s'ils avaient été limés ou conçus pour être lisses. Utiliser les paramètres de Lissage rapide pour contrôler la taille et le nombre des nouvelles faces et la manière dont elles affectent la surface de l'objet.

Les différences entre ces deux modificateurs sont les suivantes :

- Lissage rapide est beaucoup plus rapide et moins gourmand en mémoire que Lissage maillage. Lissage rapide possède une option Normales explicites non disponible avec Lissage maillage.
- Lissage rapide offre un sous-ensemble limité des fonctions de Lissage maillage. Lissage rapide utilise en particulier une méthode de lissage unique (NURMS) qui peut être

appliquée à l'intégralité d'un objet, ne possède aucun niveau de sous-objet et produit un objet à maillage triangulaire.

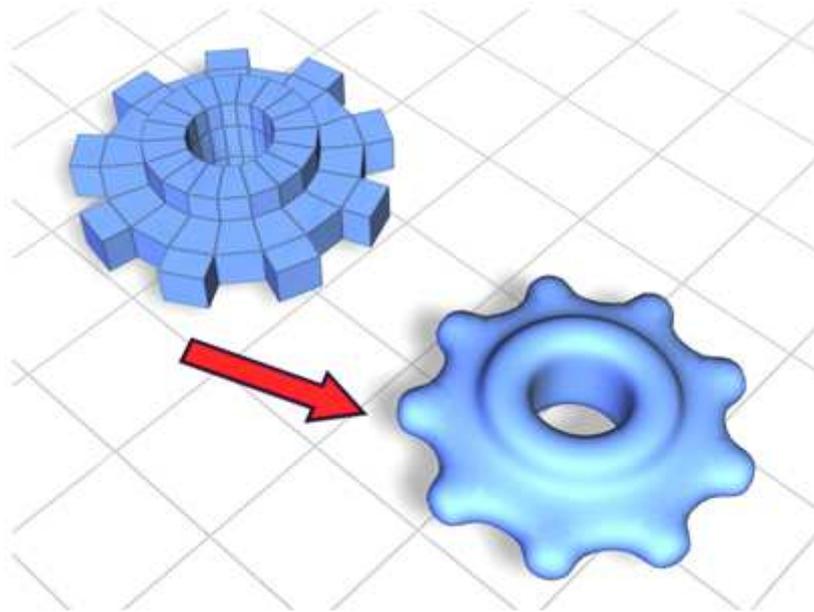


Figure II.11 : Lissage.

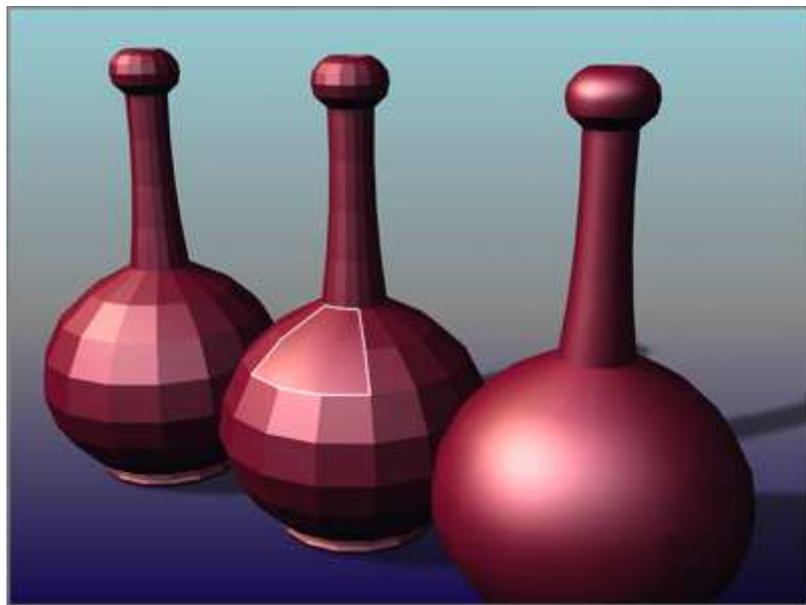


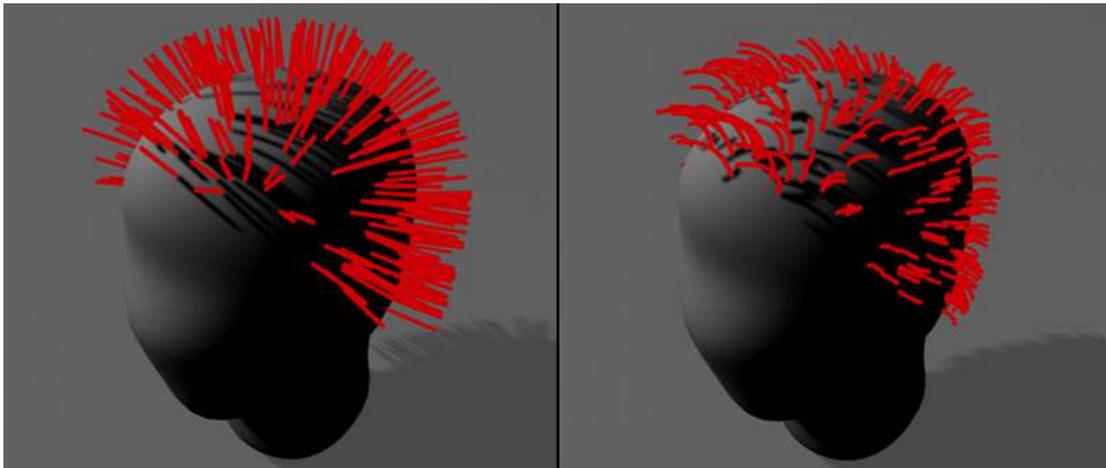
Figure II.12 : Lissage sur une zone.

**d. Le modificateur chevelure et pelage**

Le modificateur Chevelure/pelage constitue la base de la fonction Chevelure/pelage. Vous l'appliquez à tout objet à partir duquel obtenir une pousse de la chevelure/du pelage, soit

de type maillage soit de type spline. En cas d'objet maillage, la pousse s'effectue depuis toute la surface, sauf si vous sélectionnez un sous-objet. Il permet d'appliquer un style aux guides et configurer les paramètres de la taille, de la coloration, des boucles, des frisures, etc.

Les rendus Chevelure/pelage sont effectués par "recouvrements", qui sont des projections en 2D de l'espace 3D que la chevelure ou le pelage occupe. Chaque recouvrement est affiché dans la Fenêtre image rendu dès qu'il est créé. De même, vous pouvez définir la mémoire maximale utilisée par un recouvrement au moment du rendu.



**Figure II.13 : Effet de modificateur chevelure et pelage.**

### **e. Le modificateur Optimiser :**

Le modificateur Optimiser permet de réduire le nombre de faces et de sommets d'un objet. Cela simplifie la géométrie et accélère le rendu tout en conservant une image acceptable. Un indicateur Avant/Après qui informe précisément sur l'effet produit par chaque opération de réduction.



**Figure II.14 : Effet du modificateur Optimiser.**

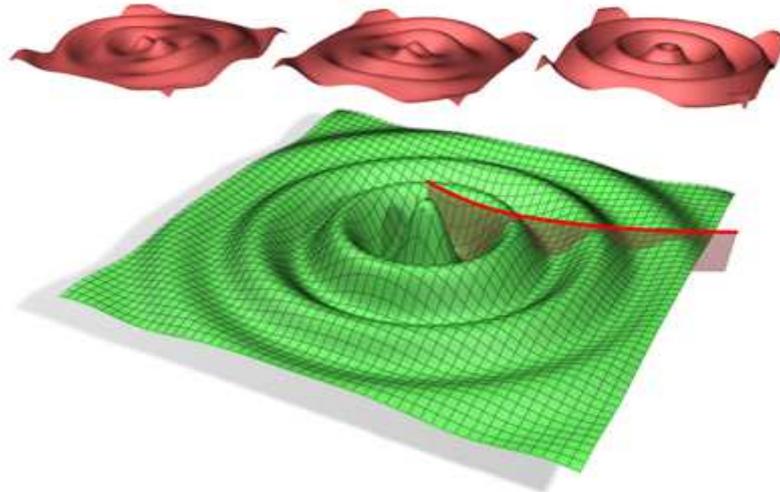
Le modificateur Optimiser permet de sélectionner des objets et de les optimiser de manière interactive.

La fonction Optimiser est un outil d'optimisation qui permet de réduire le nombre de sommets (et donc le nombre de faces) d'un objet tout en préservant son apparence. Grâce à ces options, on peut mettre à jour les informations relatives aux matériaux, aux correspondances et aux couleurs de sommet dans le modèle optimisé.

Optimiser des objets peut réduire les exigences mémoire d'une scène, simplifier la modélisation (les objets présentant moins de faces) et accélérer le rendu et l'affichage des fenêtres.

#### **f. Modificateur ride :**

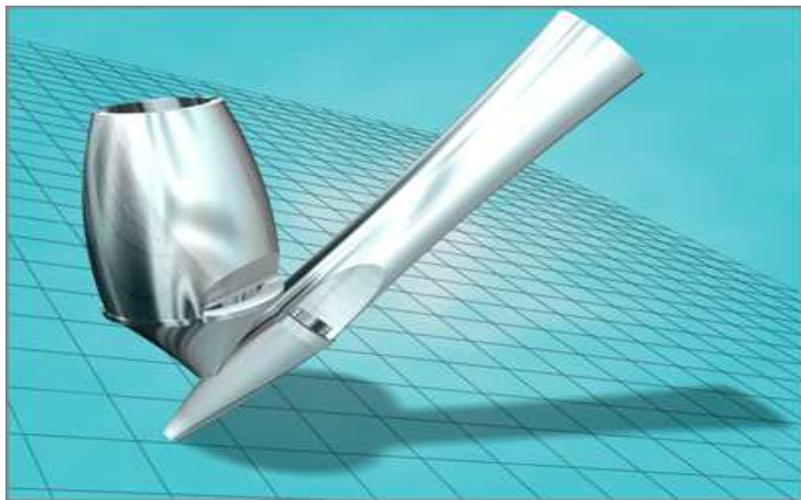
Ce modificateur produit un effet de ride concentrique dans la géométrie d'un objet. Le modificateur Ride utilise un gizmo et un centre standard auxquels il est possible d'appliquer des transformations pour augmenter le nombre de variations de ride.



**Figure II.15 : Effet du modificateur ride.**

**g. Le modificateur Etirement :**

Le modificateur Etirement simule l'effet d'étirement et d'écrasement traditionnellement utilisé en animation. L'étirement applique une mise à l'échelle le long d'un axe d'étirement sélectionné et une mise à l'échelle contraire le long des deux autres axes secondaires. La valeur du changement d'échelle sur les axes secondaires varie selon la distance par rapport au centre de l'effet de l'échelle. Le changement est maximal au centre et s'atténue à mesure que l'on s'en éloigne.



**Figure II.16 : Effet du modificateur Etirement.**

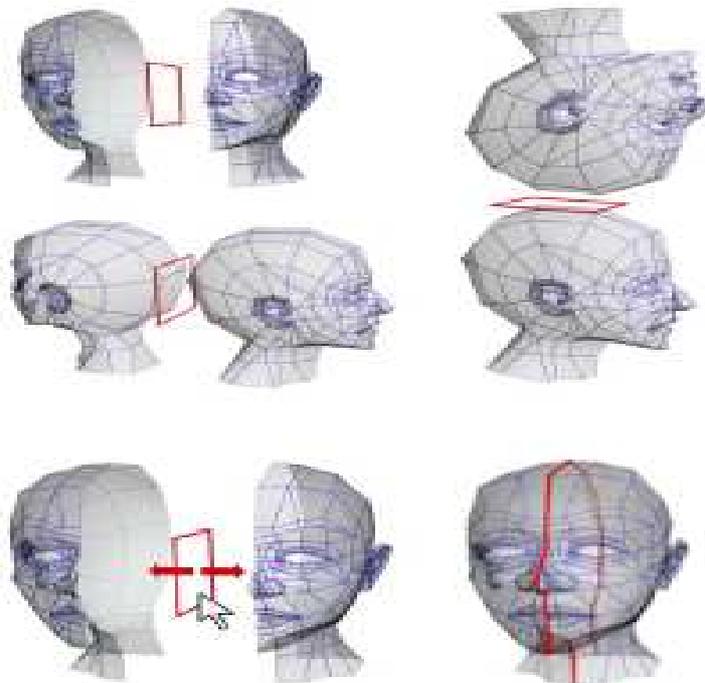
**h. Le modificateur Symétrie :**

Le modificateur Symétrie s'applique à tout type de géométrie et est particulièrement utile pour la modélisation de personnages ou la construction de bateaux ou d'avions.

Ce modificateur est unique dans le sens où il permet d'effectuer trois tâches de modélisation courantes :

- Créer une image symétrique d'un maillage autour du plan X, Y ou Z.
- Sectionner un maillage, en supprimant des parties si nécessaire.
- Souder automatiquement des sommets le long d'une couture commune.

Lorsque le modificateur Symétrie est appliquée à un maillage, toutes les modifications apportées à la moitié d'origine du maillage sous le modificateur Symétrie dans la pile se reproduisent de manière interactive sur l'autre moitié.

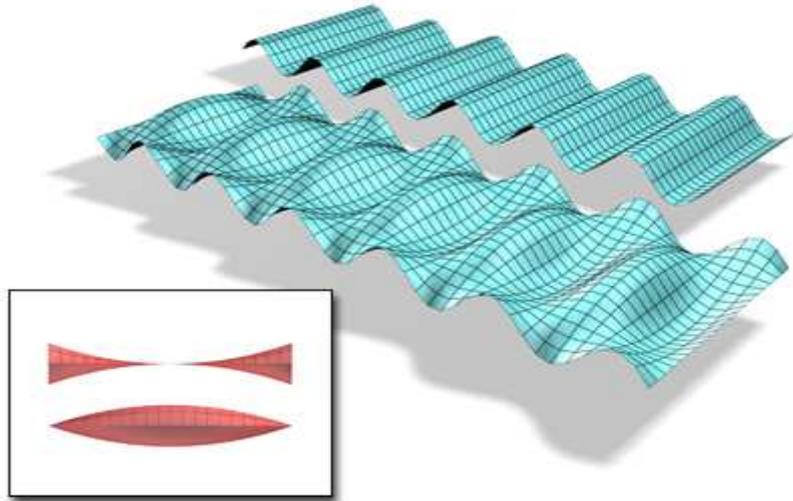


**Figure II.17 : Effet du modificateur Symétrie.**

**i. Le modificateur Onde :**

Le modificateur Onde permet de générer un effet d'onde dans la géométrie d'un objet. Le modificateur Onde utilise un gizmo et un centre standard auxquels il est possible d'appliquer des transformations pour renforcer l'effet d'onde.

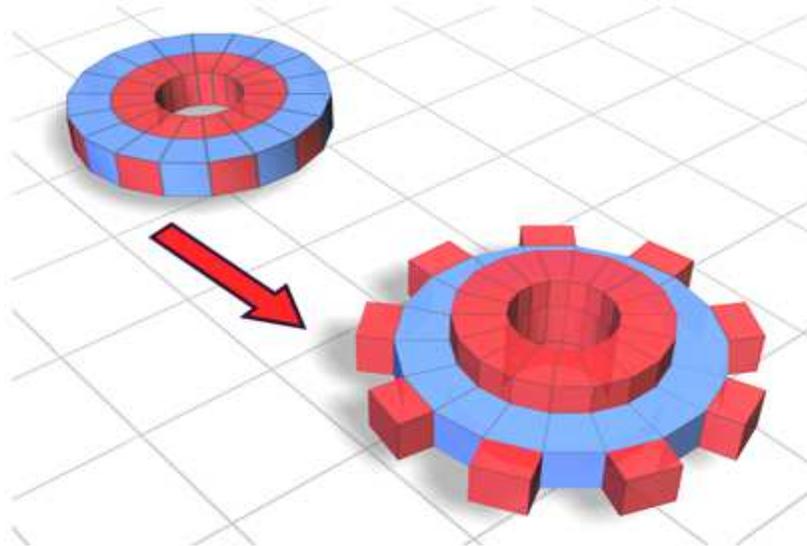
La déformation spatiale Onde a des fonctionnalités similaires et est utilisée pour appliquer des effets à un grand nombre d'objets.



**Figure II.18 : Effet du modificateur Onde.**

**j. Le modificateur Extrusion :**

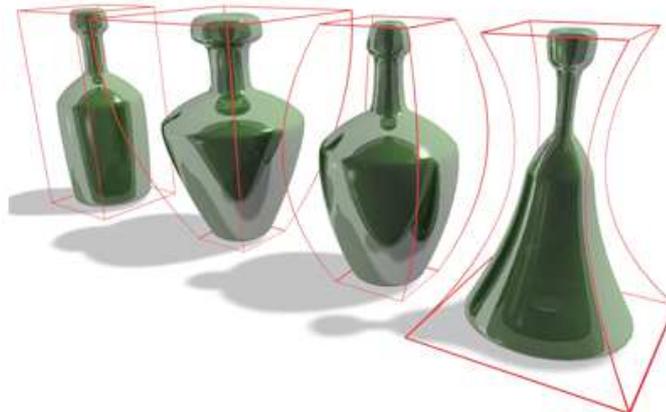
Le modificateur Extrusion face extrude les faces sur leurs normales, créant ainsi de nouvelles faces sur les côtés de l'extrusion reliant les faces extrudées à leurs objets. Comme avec la plupart des modificateurs, ceci affecte la transmission vers le haut de la pile de la sélection de la face courante. Plusieurs différences distinguent le modificateur et la fonction Extrusion face dans un maillage éditable, en particulier le fait que tous les paramètres du modificateur Extrusion face peuvent être animés.



**Figure II.19 : Effet du modificateur Extrusion.**

#### **h. Le modificateur Effiler :**

Le modificateur Effiler permet de produire un contour effilé en mettant à l'échelle les deux extrémités de la géométrie d'un objet, l'une étant agrandie et l'autre réduite. Vous pouvez contrôler le volume et la courbe de l'effilement sur deux jeux d'axes. Vous pouvez également limiter l'effilement à une partie de la géométrie.



**Figure II.20 : Effet du modificateur Effiler.**

#### **k. Le modificateur Interpolateur :**

Le modificateur Interpolateur permet de changer la forme d'un modèle de maillage, de patch ou de surface NURBS. Il gère la transformation des formes (splines) et des FFD de l'espace universel aussi les matériaux.

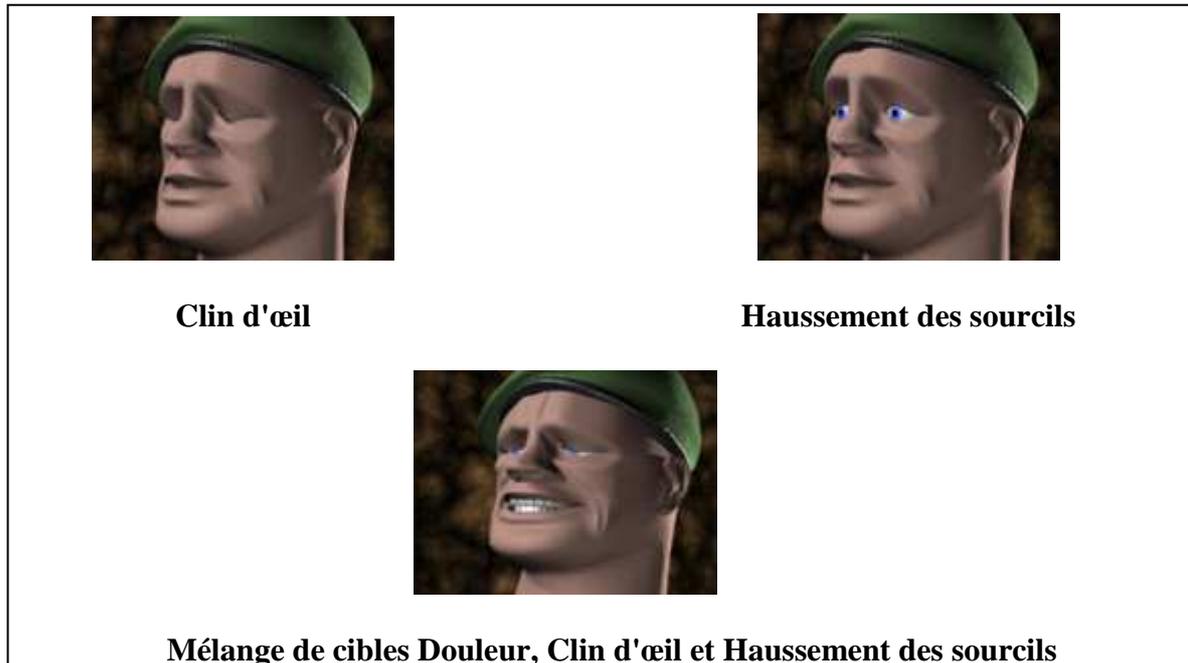
Bien que l'interpolation serve généralement à représenter les mouvements des lèvres et les expressions faciales d'un personnage 3D, cette technique peut s'employer pour changer la forme de n'importe quel modèle 3D. Le modificateur fournit 100 canaux pour les cibles et les matériaux d'interpolation. Vous pouvez mélanger des pourcentages de canaux et utiliser le résultat du mélange pour créer une nouvelle cible.



**Figure II.21 : Effet du modificateur interpolateur.**

Cibles d'interpolation pour les expressions :

Créez pour le personnage autant de cibles d'expression que nécessaire. Joie, tristesse, surprise, colère, tous ces sentiments peuvent avoir leur propre cible. Selon la personnalité du personnage, certaines cibles (la terreur, par exemple) peuvent être inutiles. Des cibles représentant des narines qui frémissent, une contraction des muscles de la mâchoire, un battement au niveau des tempes, etc. peuvent suffire à donner à votre personnage un air énervé. Chaque canal d'interpolation peut également contenir un matériau (voici quelques représentations : clin d'œil, haussement des sourcils par exemple).



**Figure II.22 : Exemple de Cible d'interpolation.**

Dans notre travail en a utiliser le modificateur symétrie pour notre objet 3D.

#### **4.2. Les matériaux :**

De base, tous les objets créés se voient assigner une couleur au hasard qui fait office de très mauvaise texture, pour pallier cette contrainte, il faut créer un matériau que l'on appliquera ensuite aux objets. La gestion des matériaux se fait à l'aide de l'éditeur de matériaux.

Les matériaux sont définis dans l'éditeur de matériaux en respectant les valeurs de degré de réflexion (La réflexion permet de régler la manière dont la lumière va rebondir sur le matériau ; pas de réflexion signifie un objet perçu noir, une réflexion maximale implique un objet presque blanc). Il est possible d'utiliser des matériaux prédéfinis dans les bibliothèques de matériaux du logiciel 3ds Max Design, mais il est souvent plus intéressant de créer les matériaux propres au projet développé, suivant les paramètres désirés.

L'interface Editeur de matériaux se compose d'une barre de menus en haut, de champs échantillon (les sphères) sous la barre de menus et de barres en bas, à côté des champs échantillon. Pour des liens de renvoi à des rubriques décrivant ces éléments d'interface ainsi que des présentations des matériaux et maps.

L'éditeur de matériaux est composé de 2 parties :

- Les échantillons (ce sont les sphères en haut, qu'on appelle aussi *slot*) et les options (les boutons autour des échantillons)
- Les panneaux de réglage

Une **Map** est une texture applicable à votre matériau. La plus utilisée est le **Bitmap**, qui vous permet de choisir une image sur votre ordinateur. Le **Gradient** va créer sur votre texture un dégradé allant du noir au blanc. Le **Noise** est très utile pour générer rapidement une texture aléatoire en niveaux de gris.

Il existe une infinité de façons de coller un matériau sur un objet, nous citons le mapping qui consiste justement à positionner le matériau à la surface de l'objet.

Les matériaux déterminent la façon dont un objet reflète ou transmet la lumière. Dans un matériau, les maps vous permettent de simuler des textures, des dessins appliqués, des réflexions, des réfractions, entre autres effets. (Les maps peuvent également être utilisées en tant qu'environnements ou projections émanant de lumières.) L'éditeur de matériaux est une boîte de dialogue qui vous permet de créer, modifier ou appliquer des matériaux dans votre scène.

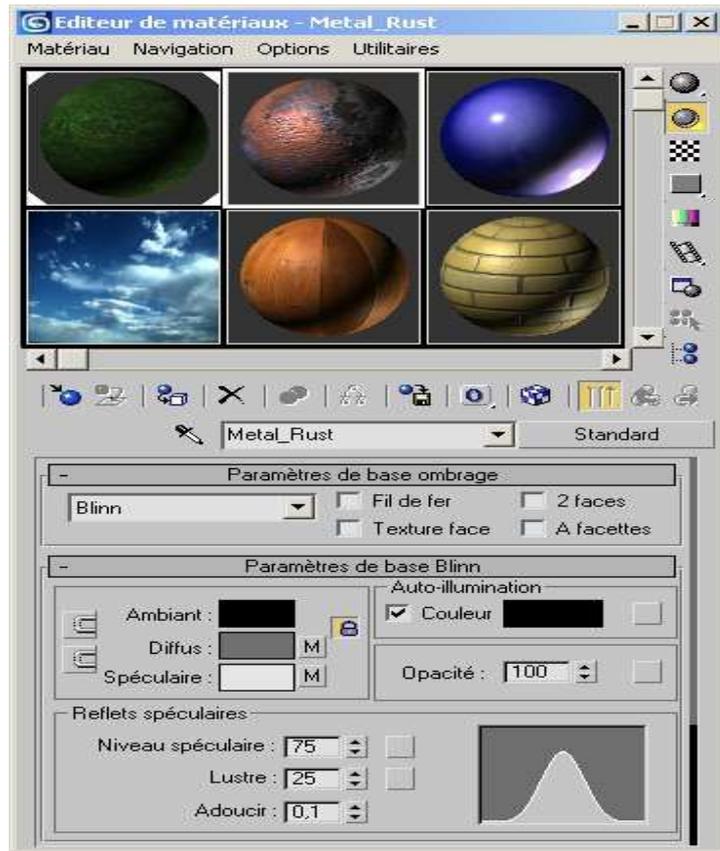
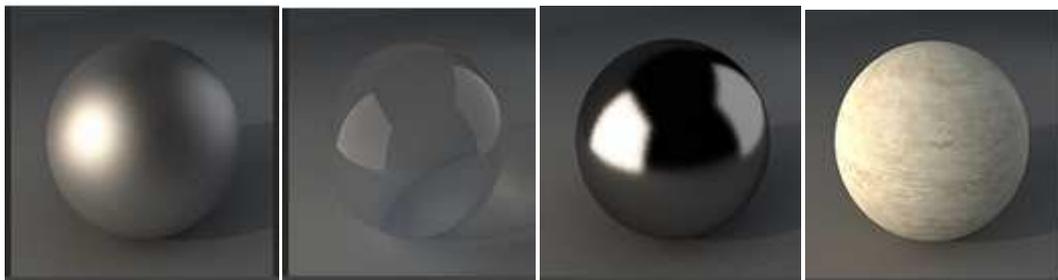


Figure II.23 : L'interface Editeur de matériaux.

❖ Quelques matériaux de base :



Chrome

Glace

Argent

Marbre

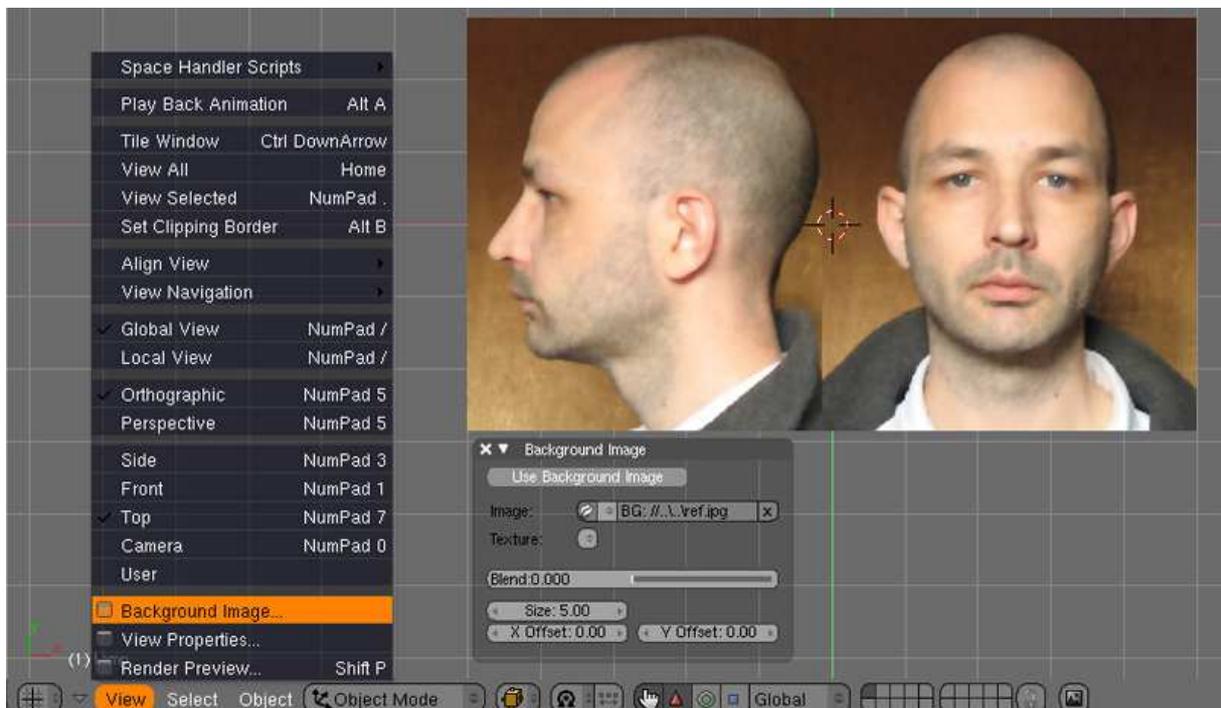
5. Exemple de modélisation d'un visage 3D :

Pour commencer, la modélisation d'un visage demande deux ou trois photos de personnage, vu de face, vu de gauche et vu de haut si nécessaire. Si le personnage est imaginaire les images sont dessinées par un artiste en 2D. Les trois images s'appellent (Blueprint) serviront de modèle approximatif à l'objet final. **Figure II.24**



**Figure II.24**

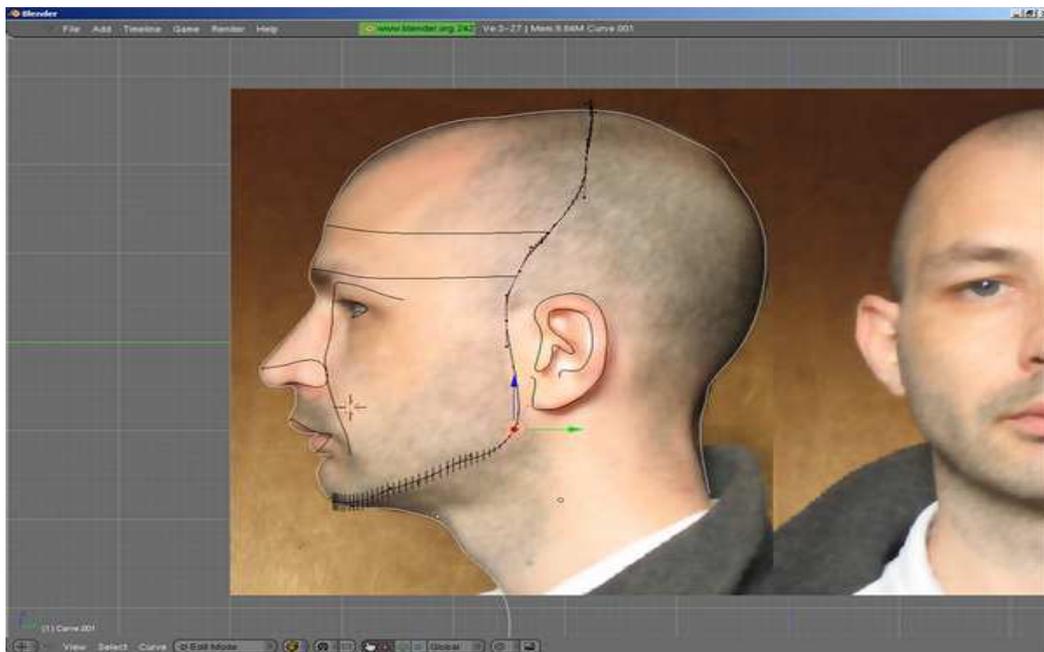
On commence par préparer la scène, en créant des plans sur lesquels on va affecter les fonds ou les Blueprint qu'on a. **Figure II.25**



**Figure II.25**

Ensuite, nous allons faire un gabarit approximatif à partir des vues coté/face à l'aide de courbe de Bézier.

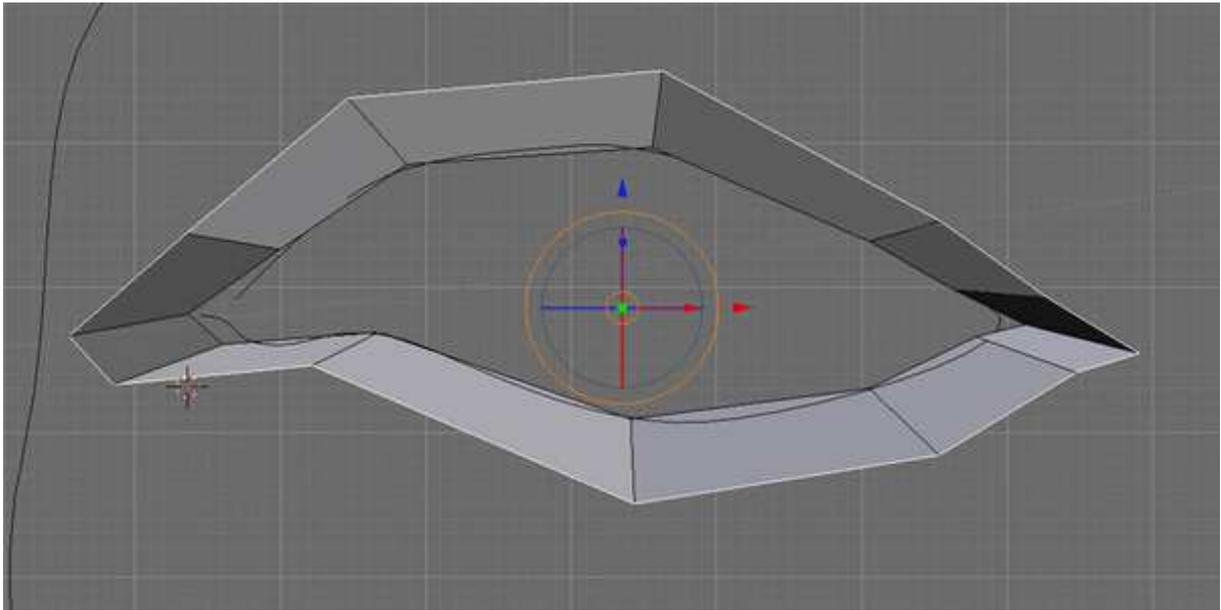
Ce gabarit est loin d'être indispensable à la modélisation en elle-même, mais nous servira à placer les contours et les traits principaux du visage avant la modélisation elle-même, comme une espèce de "croquis" en 3D (c'est une petite astuce qui permet de gagner énormément de temps pour la modélisation). **Figure II.26**



**Figure II.26**

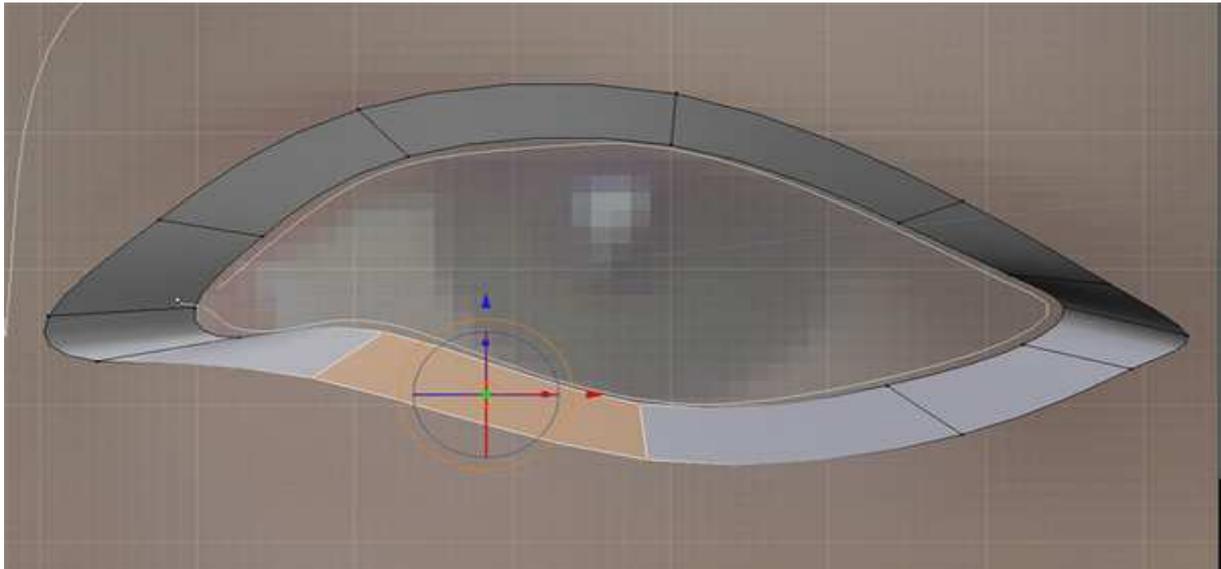
Pour la modélisation il n'y a bien évidemment pas qu'une seule façon de procéder, et ce qui va suivre tient plus d'un compte rendu de travail que d'une technique universelle. A chacun de faire en fonction de ses préférences et de sa sensibilité. **Figure II.27**

Après, on crée un troisième plan, qui sert comme primitive éditable pour notre objet. On commence par extruder en suivant le contour (curve) de l'œil en vue de face/coté. On ferme en sélectionnant les deux vertices de jonctions. On continue l'extrusion en suivant les blueprint, puis on subdivise en cas de besoin, on bouge chaque sommet par rapport à nos repères. On rajoute des sommets où ça demande plus de détails. La procédure demande de l'imagination dans l'espace, précision et l'œil artistique.



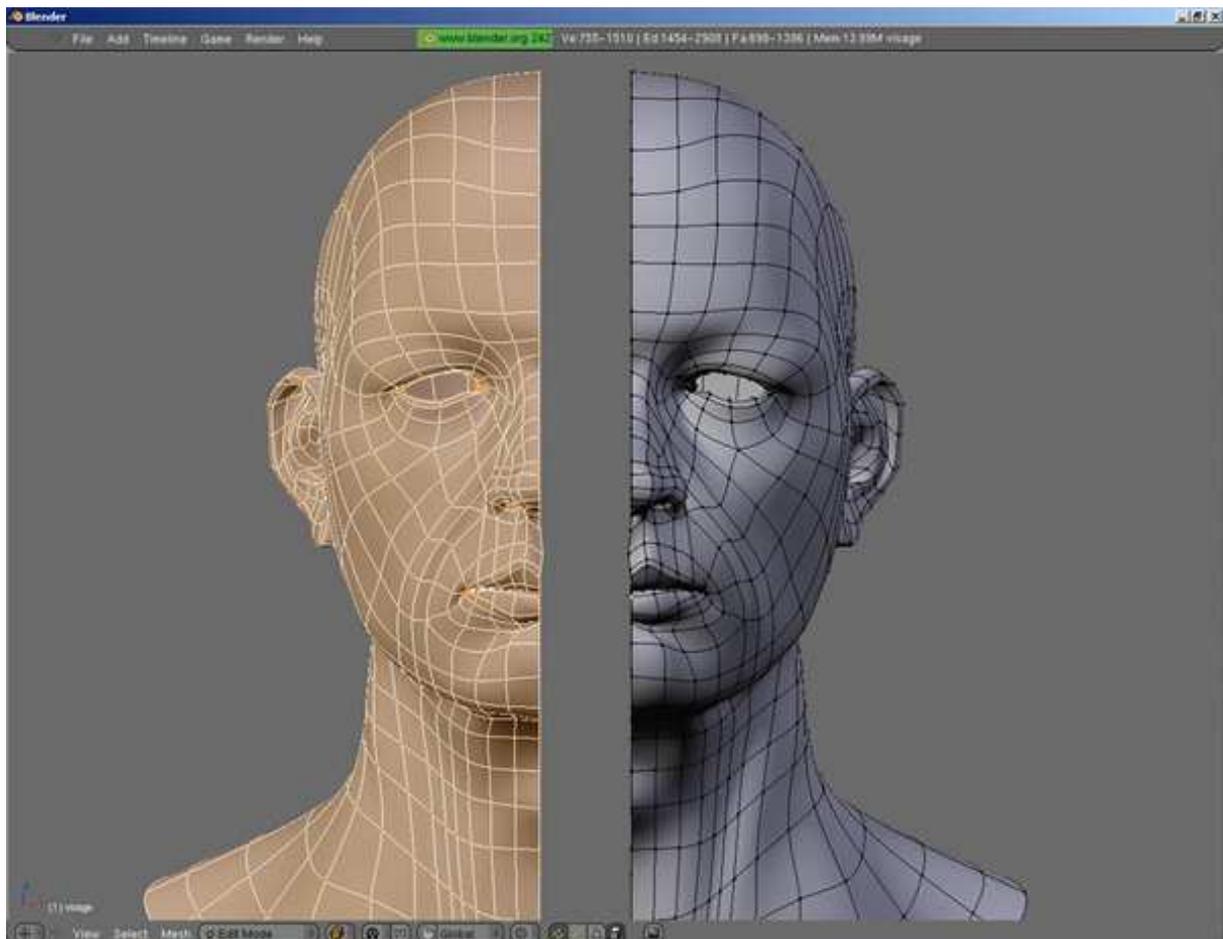
**Figure II.27**

On applique le modificateur lissage qui déformera obligatoirement le positionnement de départ des vertices (les vertices ou les sommets sont des points dans l'espace : ils définissent la structure du visage), donc là encore c'est un choix de modélisation que de partir d'une maille avec un modificateur. A savoir que chaque vertice déplacée aura obligatoirement une action sur ses voisins immédiats. Il faudra donc sans cesse repositionner les vertices pour coller au plus près à la référence (gabarit et/ou background image). **Figure II.28**



**Figure II.28**

Ensuite après avoir réalisé la coque de base de visage, on duplique cette moitié par symétrie puis on raccorde les deux pour obtenir le visage complet. **Figure II.29**

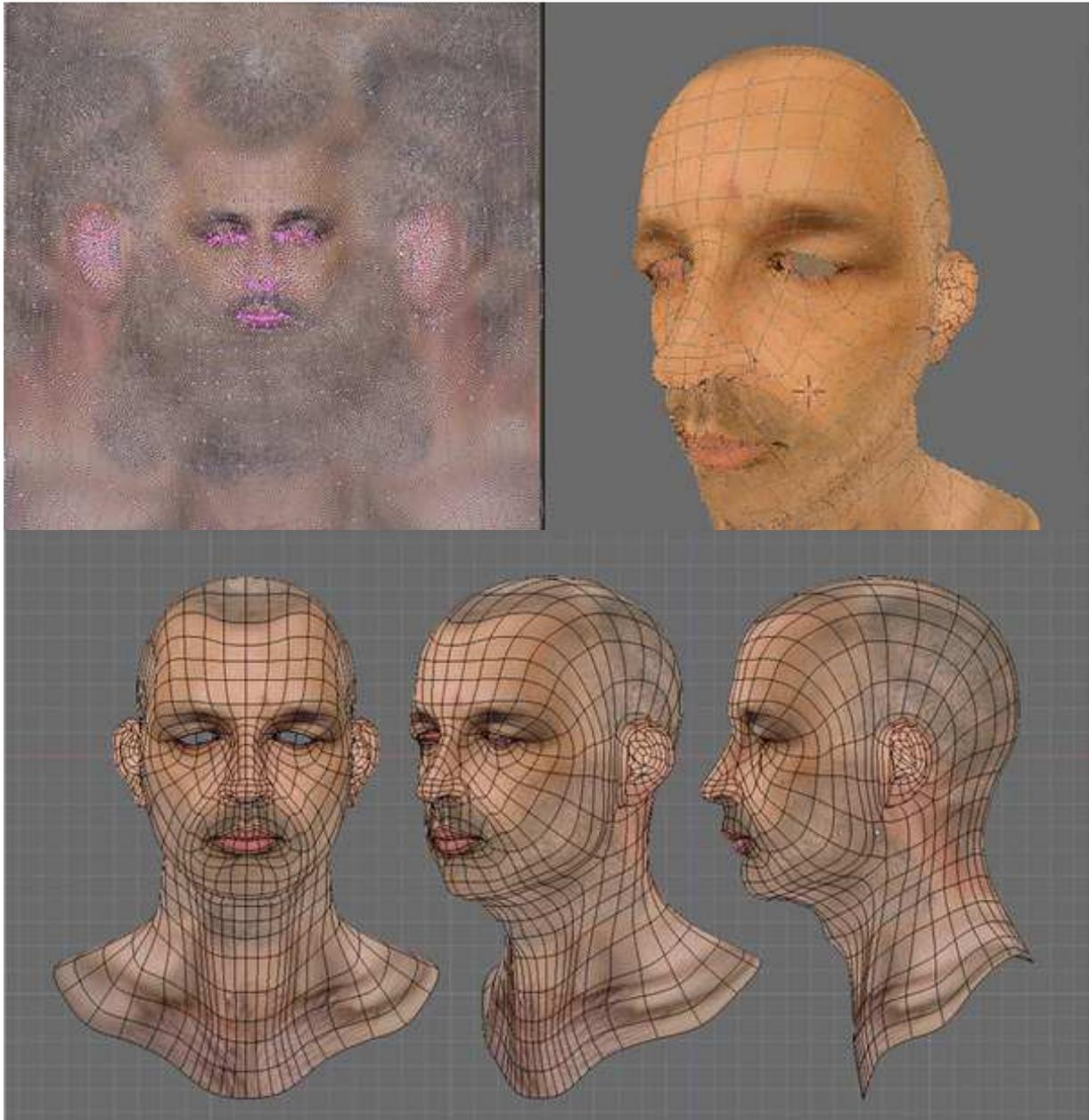


**Figure II.29**

On peut donc passer directement à la fabrication de la texture (colormap) elle-même.

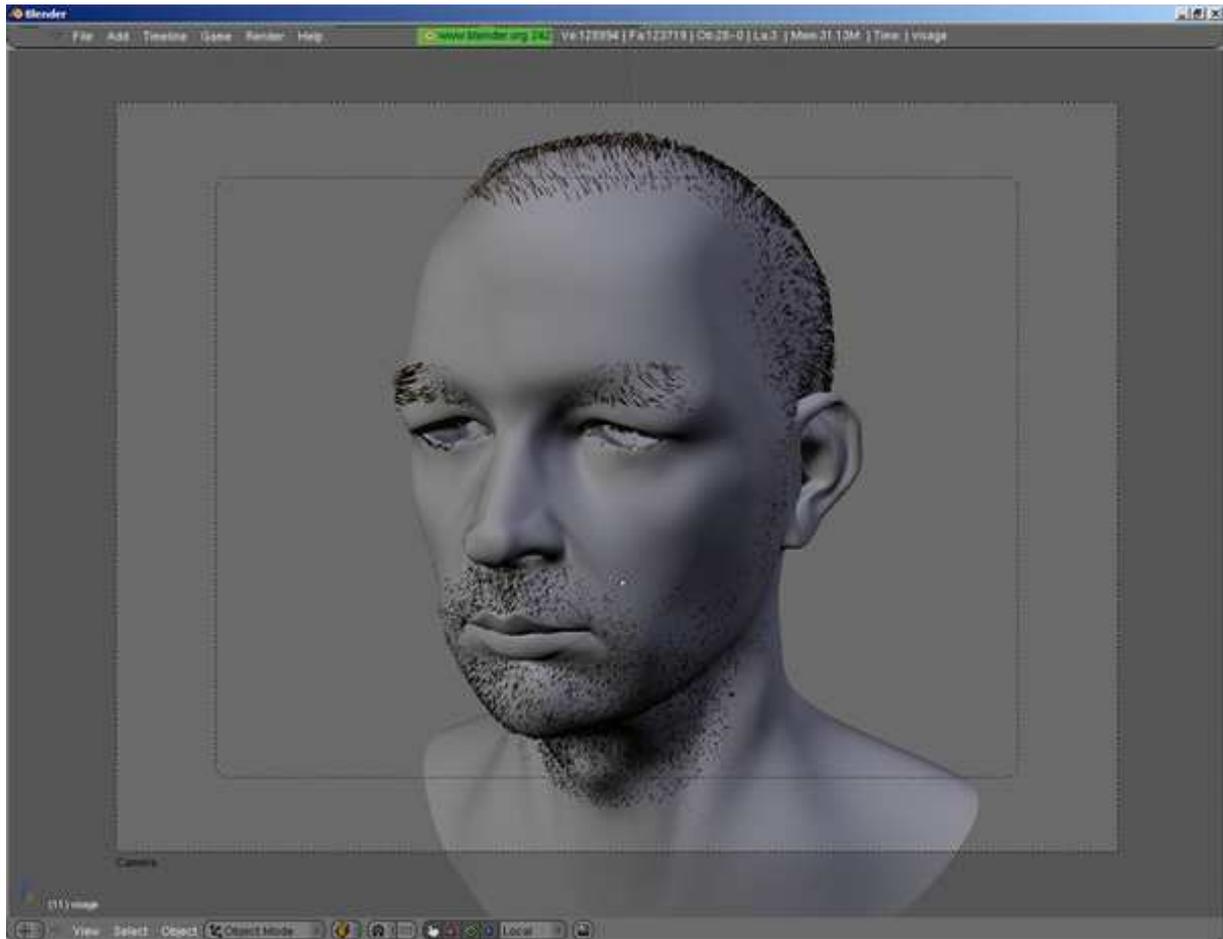
Une fois la map de couleur en place, on s'en servira comme base pour créer la texture de relief et celle des brillances. **Figure II.30**





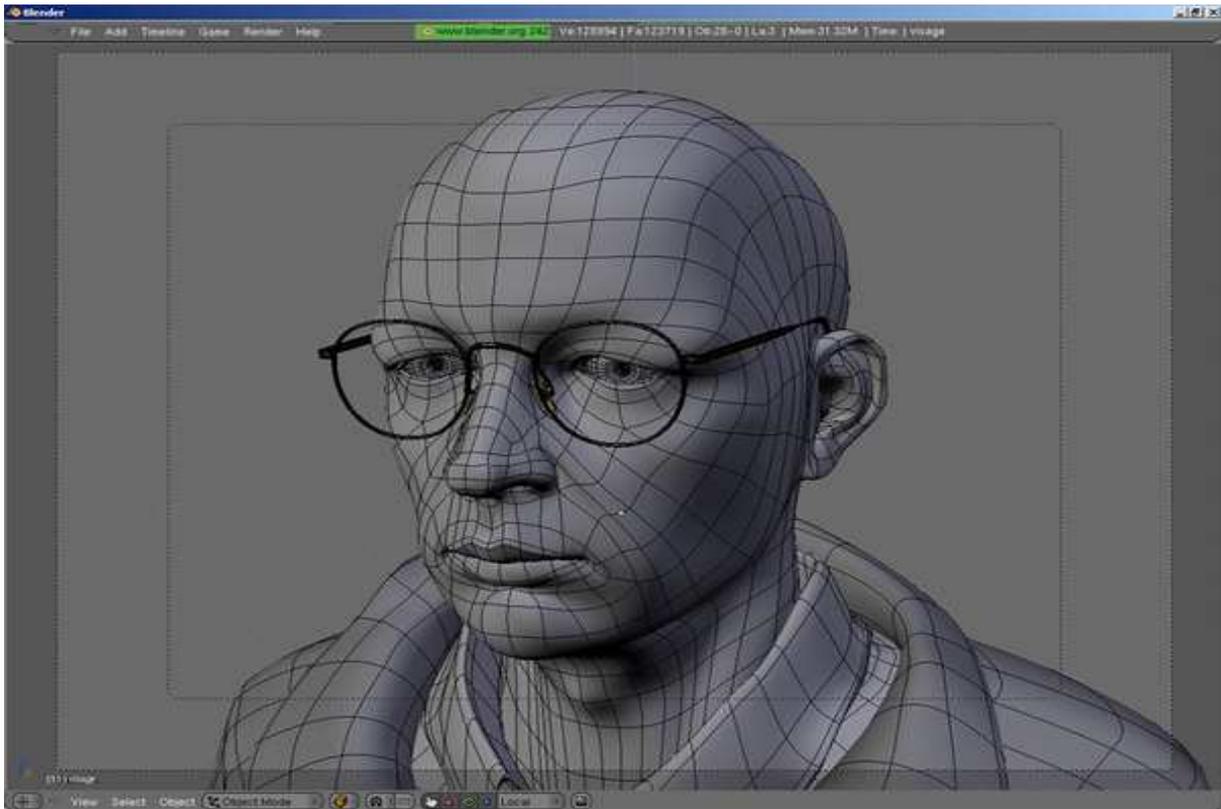
**Figure II.30**

On applique le modificateur chevelure sur des zones bien précises pour obtenir une pousse de la chevelure, il sert à ajouter des poils sur la tête, barbe et paupière. **Figure II.31**



**Figure II.31**

On passera à apporter quelques précisions sur la finalisation du modèle, à savoir l'ajout des yeux. **Figure II.32**



**Figure II.32**

Enfin, c'est le moment du rendu de la scène, en utilisant un moteur de rendu. **Figure II.33**



**Figure II.34**

Avec ce bref exemple qu'on s'inspirera à modéliser les objets en particulier les visages en 3D.

### **Remarque :**

La partie lissage peut se faire avant la symétrie comme dans le cas cité précédemment, mais également il peut se faire après la symétrie, puis on rajoute des petits détails sur le nez, la bouche, les zones autour de l'œil et des poils à la barbe, sourcil, paupière et autres à l'aide des différentes fonctions de modificateurs. Ensuite, on rajoute deux sphères pour les yeux. En finale, on affecte les matériaux et les motifs.

### **Conclusion :**

La 3D est utilisée dans différents domaines (Cinéma, Jeux, Simulation Scientifique ...). Dans ce chapitre, nous avons présenté l'ensemble des étapes de réalisation d'un produit 3D dans le domaine d'infographie à savoir la modélisation, l'animation et le rendu.

Dans le cas d'une animation 3D, la forme à animer doit être suffisamment légère pour pouvoir bouger de façon non saccadée, tout en conservant une bonne qualité. Notons que nous avons insisté en détail sur la modélisation polygonale ; méthode que nous utiliserons dans notre application, qui est jugée plus efficace par rapport aux autres méthodes existantes vu sa facilité d'utilisation et sa permission de réduire le nombre de polygones, pour obtenir une animation fluide, et préserver au maximum la qualité de définition pour le rendu.

# Chapitre III

**Conception et réalisation**

### Introduction

Dans cette partie, nous allons spécifier d'une manière bien détaillée les étapes de réalisation de notre application. Au cours de nos recherches sur la réalisation d'un produit 3D, nous avons constaté que la modélisation 3D qui se fait d'une manière manuelle, prend un temps considérable (La modélisation d'un visage de base nécessite aux moins quatre heures) et encore nécessite une acquisition d'un grand savoir-faire.

Pour remédier à cette complexité, nous avons pensé à une solution, non complexe, moins couteuse, fiable, rapide, qui consiste à automatiser la modélisation 3D à base de deux vidéos captant un visage humain considéré comme le cas d'étude pour notre réalisation.

Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou une démarche bien définie.

### 1. Les étapes de production :

Pour la réalisation de notre application, nous proposons une méthode de modélisation composée de différentes étapes ainsi résumé par la figure ci-dessous et détailler dans les paragraphes qui suivent :

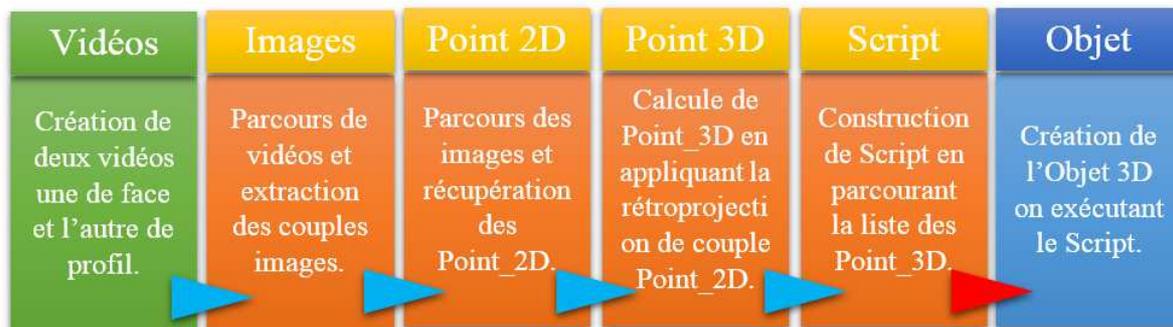


Figure III.1 : Etapes de production

#### 1.1.Récupération des vidéos :

##### a. Positionnement des caméras :

Pour commencer, on aura besoin de deux caméras de référence, l'une de profil et l'autre de face, tel indiqué sur la figure III.2.



Figure III.2 : Positionnement des caméras.

**b. Lancement de balayage**

Après avoir posé les caméras en bonne position, on utilisera l'outil laser pour balayer la surface du visage horizontalement du haut vers le bas.



Figure III.3 : Exemple du balayage.

**1.2. Parcours et extraction de points 3D :**

**a. Extraction de séquence d'images :**

On parcourt les deux vidéos, images par image pour extraire simultanément une image de chaque une (une de face et l'autre de profil).

**b. Parcours des images :**

On parcourt les images pixel par pixel jusqu'à ce qu'on retrouve le point vert du laser pour chaque couple d'image extraite. Le point vert repéré est constitué de pixels compris dans une

zone en forme d'une matrice de (20 sur 20) pixels, on parcourt ainsi cette matrice afin de calculer le centre du point laser.

On détermine le centre du point vert par rapport aux bornes gauches et basses de l'image, on obtient de la sorte un point 2D selon les coordonnées suivantes :

$$\text{Point\_2D}(\frac{\Delta X}{2} + X_{min}, \frac{\Delta Y}{2} + Y_{min})$$

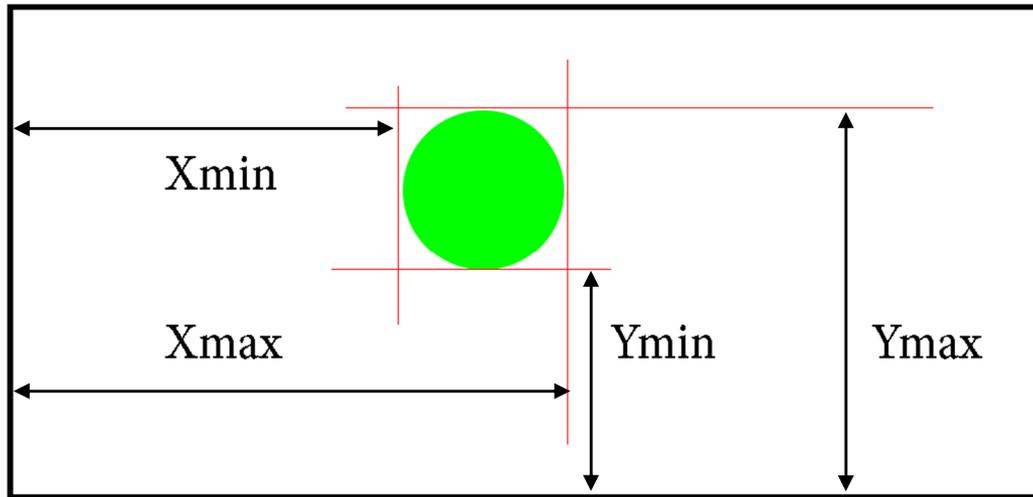


Figure III.4 : Déterminer le centre du point vert.

On aura en finale, deux points 2D, l'un de face et l'autre de profil, qu'on nommera respectivement **P2D\_F**, **P2D\_P**.

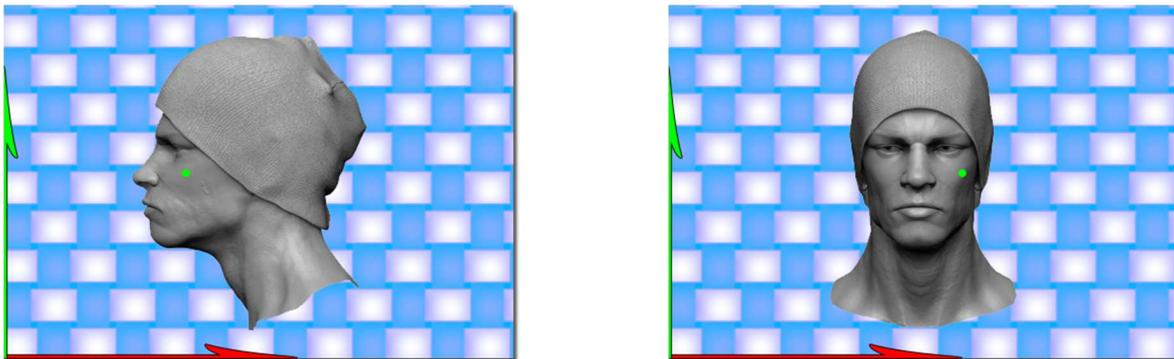


Figure III.5 : Illustration de deux points 2D.

On utilisera ces deux points extraits des deux images pour construire un point 3D, tel que ses coordonnées seront les suivantes :

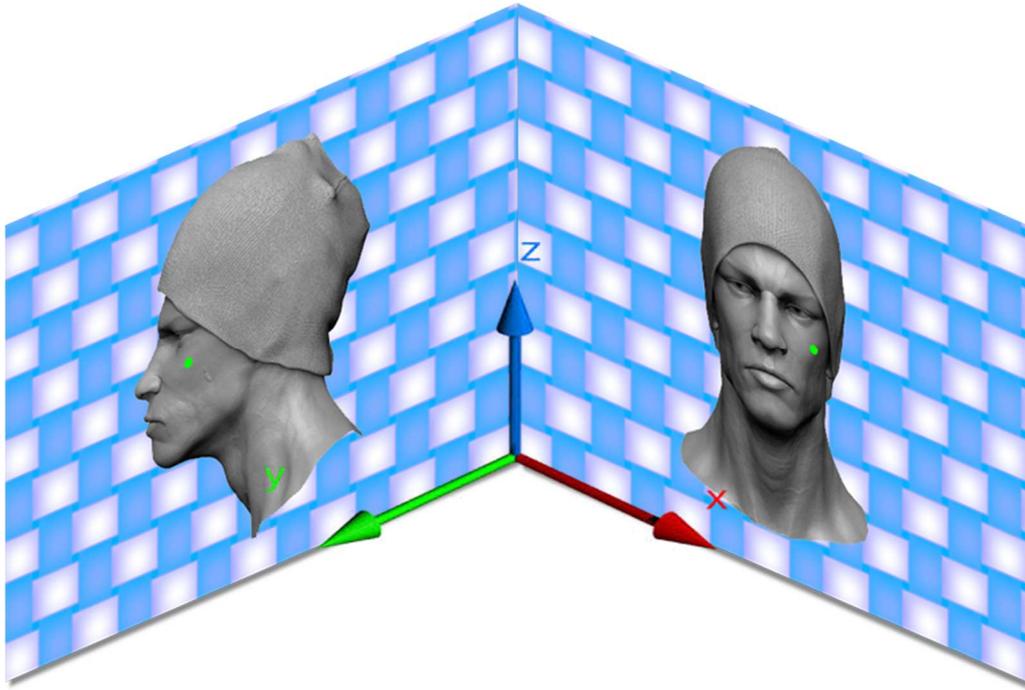
$$\text{Point\_3D} (\text{P2D\_F} . X, \text{LI} - \text{P2D\_P} . X, \text{P2D\_F} . Y).$$

**LI** = Largeur de l'Image.

La coordonnée X du point 3D est la coordonnée X de point 2D de face (P2D\_F.X).

La coordonnée Y (profondeur) du point 3D est la coordonnée X de point 2D de profil (P2D\_P.X).

La coordonnée Z du point 3D est la coordonnée Y de l'un des points 2D, dans notre cas on a choisi le point de face (P2D\_F.Y).



**Figure III.6 : Construction du point 3D.**

### 1.3.Création de script de construction d'objet :

Après avoir réuni les points 3D, représentant le point laser détecté, dans une matrice, on passera à la construction de script (MaxScript) suivant la syntaxe ci-dessous puis l'enregistrer sous extension '.ms'.

#### La syntaxe:

**Mesh vertices :#([x1,y1,z1], [x2,y2,z2], [x3,y3,z3], ...) faces :#( [p1\_1,p2\_1,p3\_1], [p1\_2,p2\_2,p3\_2], [p1\_3,p2\_3,p3\_3], ...) materialIDS :#(1,2,3, ...)**

Cette instruction sert à construire un maillage en trois sous instructions :

### 1) Construction des points (Sommets) :

**Mesh vertices :#()** est une instruction qui désigne les coordonnées de tous les points (x,y,z) qu'utilisera notre maillage. Les sommets auront comme identifiant l'indice de leur ordre de création.

### 2) Construction des Surfaces (Polygones) :

**faces :#()** est une instruction qui prend en paramètres plusieurs trio d'identifiant de points qui formeront des polygones triangulaires.

### 3) Affectation des identifiants aux Polygones :

**materialIDS :#()** est une instruction qui permet d'affecter des identifiants pour chaque polygone triangulaire.

#### 1.4.Importation & exécution de fichier script :

Il y'a plusieurs façons d'importer et exécuter un script sur 3Ds MAX. La plus simple est la suivante : cliquer sur la barre de menus *MaxScript >Exécuter Script* puis sélectionner un script de l'extension '*.ms*' puis l'ouvrir. L'objet sera construit automatiquement sur la scène 3Ds MAX.

## 2. Optimisation des étapes de production :

### 2.1.Récupération des vidéos :

#### 2.1.1. Positionnement des caméras :

Pour avoir un résultat optimal, lors de la prise des vidéos il est conseillé de placer les caméras le plus loin possible du sujet et si possible se servir d'un zoom ou mieux d'un téléobjectif, le but étant de se rapprocher le plus possible d'une vue orthonormée.

La qualité des vidéos joue un rôle pour le résultat final, c'est pour cela qu'une très bonne qualité de vidéo est exigée pour une meilleure détection de la position du laser quel que soit sa finesse.

#### 2.1.2. Lancement de balayage :

Lors du balayage, il est préférable de commencer du milieu haut du visage vers l'extérieur et éteindre le laser tout au long du retour pour ne pas recalculer les points déjà balayé.

Pour le choix du laser, il faut prendre compte de certaines exigences de sélection comme le diamètre du pointeur qui doit être fin (afin de réduire les erreurs) et coloré (exemple : vert pour le distinguer des points du visage et le mettre en évidence et aussi faciliter le tracking).

### **2.2.Parcours et extraction de points 3D :**

Pour optimiser le temps de parcours, on doit minimiser le nombre de pixels parcourus, pour cela, on donne la possibilité à l'utilisateur de délimiter la zone de recherche en dessinant un cadre sur la moitié du visage à modéliser (un de face et l'autre de profil).

On applique aussi le filtre Postérisation sur une image pour la conversion d'une gradation continue, avec des changements brusques d'un ton à l'autre et ceci conduit à réduire trois fois le nombre de contrôle sur un pixel ; tel qu'on effectue trois contrôles sur un pixel pour vérifier s'il est vert par contre avec le filtre on détecte la couleur dès le premier test.

Dans notre cas, on aura une image de 1200×720 pixels :

*-Avant l'optimisation* on a effectué 3 tests pour chaque pixel, en moyenne 1 296 000 tests par image et le temps de réponse était de 50 mn : 06s.

*-Après l'optimisation* (cadre de taille 263×440 pixels et filtre postérisation) :

On a réussi à réduire le temps de réponse jusqu'à 00 mn :29 s avec une moyenne de contrôle de 39 490 tests par image. Sachant que nos vidéos comportent 724 trames chacune.

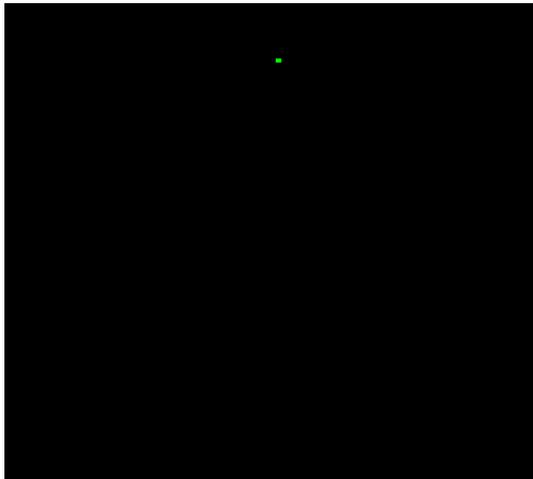
Notons que ce résultat sera plus satisfaisable sur une machine plus puissante que celle que nous avons utilisé et qui a les performances suivantes : (3Go RAM, processeur Intel(R) core(TM) 2 Duo 2.0 GHz).

Généralement, les équipements utilisés pour ce genre d'application (production 3D) sont spécifiques et adéquats en termes d'outils et puissance de calcul.

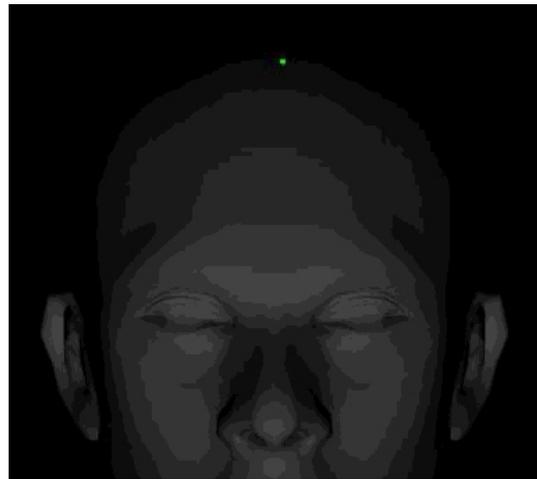
Le cadre et filtre ne sont pas effectués automatiquement par notre application, c'est par l'outil de retouche 'Photoshop' qu'on a réussi à réduire la taille des trames et filtrer la vidéo.



**Figure III.7 : Photo avant filtre.**



**Figure III.8 : Postérisation niveau 2.**



**Figure III.9 : Postérisation niveau 20.**

### 2.3.Création de script de construction d'objet :

Pour une construction optimale, on a opté pour l'ajout de quelques instructions qu'on définira ci-dessous :

Avant de commencer, on doit sélectionner le maillage en utilisant l'instruction :

```
max select all;
```

On déplace le point de pivot selon les points suivants, de sorte qu'il sera positionné au centre bas de la mâchoire.

```
$.pivot.x = $.min.x;
```

```
$.pivot.y = $.min.y;
```

```
$.pivot.z = $.min.z;
```

On transpose le maillage sur l'origine aux coordonnées (0,0,0)

```
move $ [-$.position.x,-$.position.y,-$.position.z] ;
```

Ensuite, on applique le zoom jusqu'à avoir l'objet au milieu des fenêtres pour bien le visualiser.

```
max zoomext sel all;
```

Enfin, on applique le modificateur symétrie (cf.chapitre II.4.1.) puis le lissage sur l'objet.

```
modPanel.addModToSelection (symmetry ()) ui:on;
```

```
modPanel.addModToSelection (TurboSmooth ()) ui:on;
```

Pour rendre l'objet fluide (moins de points) et permettre un lissage plus optimal, l'application propose un masque que l'utilisateur adaptera sur le visage de la 1ere image de face, ce masque a la même approche que le modificateur pro\_optimizer, seulement que cette technique est plus optimale et suit une topologie artistique.

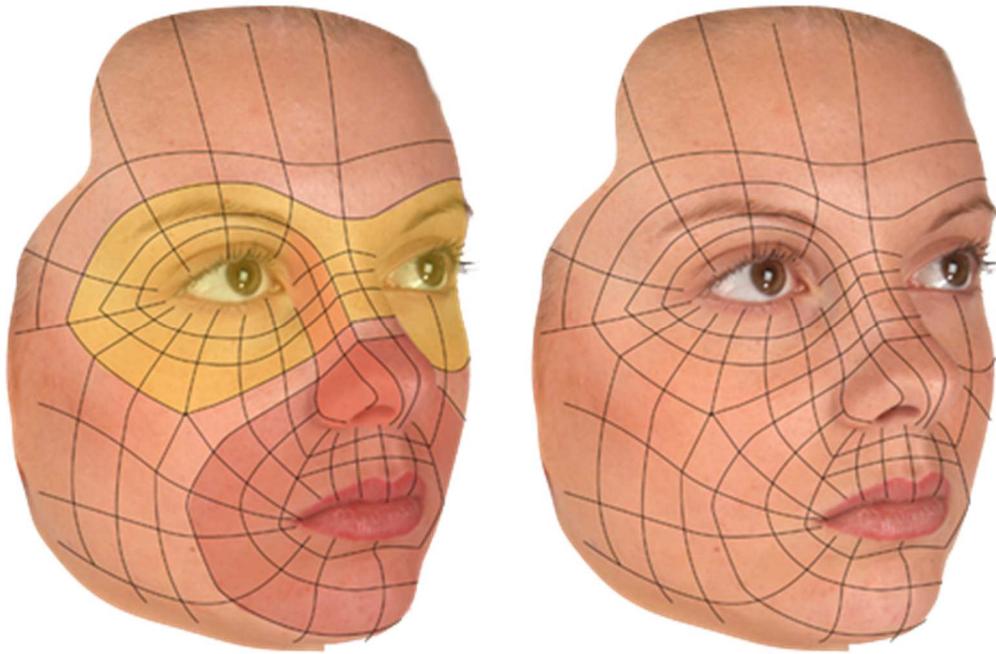


Figure III.10 : Exemple de masque.

Pour une meilleure optimisation, au lieu de construire un fichier script, on construira un fichier '.obj' qui suit la syntaxe suivante : [6]

**OBJ** est un format de fichier contenant la description d'une géométrie 3D.

Une surface polygonale est décrite par un ensemble de sommets (accompagné de coordonnées de texture et de normales en chaque sommet) et d'un ensemble de faces.

Description	Instruction
Un sommet	<b>v 1.0 0.0 0.0</b>
Une coordonnée de texture	<b>vt 1.0 0.0</b>
Une normale	<b>vn 0.0 1.0 0.0</b>

Tableau III.1 : Syntaxe d'instruction obj

Chaque face est ensuite définie par un ensemble d'indices faisant référence aux coordonnées des points, de texture et des normales définies précédemment.

Par exemple, la face suivante : « **f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3** », définit un triangle constitué des sommets d'indices v1, v2 et v3 dans la liste des sommets v. Chacun de ces

sommets possède une coordonnée de texture, identifiée par son indice dans la liste des coordonnées de texture vt, et une normale identifiée dans la liste des normales vn.

Lorsque plusieurs objets cohabitent dans le même fichier, la section définissant l'objet est définie par :

**o [nom de l'objet]**

À titre d'exemple, un cube est défini de la manière suivante : (L'exemple complet est sur l'annexe)

**# cube.obj**

**o cube**

**# 8 sommets : v 0.0 0.0 0.0**

**# 6 normales : vn 0.0 0.0 0.0**

**# 14 données de texture normales : vt 0.0 0.0**

**# 12 faces : f : 1/11/2 7/14/2 5/12/2**

### 3. Boîte à outils :

Pour la réalisation on utilisera les outils suivant :

- Un laser vert
- Un dispositif électronique pour le balayage avec le laser vert (On peut le remplacer avec un data show)
- Deux caméras (de bonne qualité de préférence (une camera HD))
- Bibliothèque d'extraction images à partir de vidéo.

Vu le manque de matériel électronique, et au lieu de prendre des vidéos réelles, on a simulé le balayage de laser avec une animation utilisant 3Dsmax sur un personnage 3D. Cette technique de simulation génère des certaines erreurs et manque de précision. Les vidéos obtenues lors de la simulation sont peu fiables car lors de calcul de rendu, le reflet de point vert n'apparait pas en bonne position au moment de son déplacement d'un polygone à un autre (changement de polygone).

### 4. Choix de la bibliothèque :

Lors du choix de bibliothèques a utilisé pour la réalisation de notre application, on a du rencontrer de nombreux problèmes car toutes les bibliothèques que nous avons testés ne répondaient pas efficacement à nos exigences, (chacunes d'elles possèdent des contraintes ou ne donne pas de résultat), c'est ce qu'on expliquera en détail dans ce qui suit. Noter que le choix de la bibliothèque est une étape primordiale pour la réussite de notre travail.

**JMF** : (Java Media Framework) : Elle permet d'extraire et d'accéder facilement aux images par index, mais la contrainte majeure de cette bibliothèque est la couleur et le rajout de point. Heureusement on a devancé ce problème avec une proposition d'une méthode qui parcourt des points essuient des images extraient depuis les deux vidéos qui représentaient l'objet à modéliser en 3D ; pour ensuite les comparer manuellement avec Photoshop et enregistrer les images qui ont des erreurs. [12]

**VLCJ** : Fournit un cadre de Java pour permettre une instance d'un lecteur multimédia VLC originaire d'être noyé dans une fenêtre AWT Java ou JFrame.

Elle permet d'obtenir plus de fluidité en lecture, et obtient également un cadre de niveau supérieur qui cache beaucoup la complexité de travailler avec libvlc.

VLCJ est principalement développé et largement testé sur Linux, elle fonctionne également aussi bien sur Windows et MacOSX. [13]

Les contraintes de cette bibliothèque sont :

- L'extraction qui passe sur une JFrame ou JPanel qui influe sur sa vitesse.
- Elle demande une bibliothèque native, tandis que l'exécutable du projet doit trainer des fichiers DLL de lecture VLC, pour toute les plateformes de version 32 et 64 bits, ce qui ne convient pas avec la portabilité de java.

**GStreamer** : est une bibliothèque logicielle de manipulation de sons et d'images (appelée aussi Framework multimédia) mais qui dépend sur d'autres bibliothèques jar ainsi que leurs versions. [14]

**FFmpeg** : Est une collection de logiciels libres destinés au traitement de flux audio ou vidéo (enregistrement, lecture ou conversion d'un format à un autre).

Le traitement prend plus de temps pour le résultat et qui est plus que la procédure manuelle, parce que l'extraction des images vers le disque demande deux accès disque pour chaque frame, et c'est la contrainte que nous avons trouvé pour cette bibliothèque. [15]

**Xuggler** : En général, elle permet la décompression, la modification et la re-compression des fichiers multimédia (ou flux) de Java. [16]

- **Les Inconvénients et contraintes :**

- L'extraction n'est pas facile : ne détecte pas automatiquement la fréquence de la vidéo.
- N'extrait pas la première et le dernier frame, la solution personnelle est de rajouter automatiquement une frame au début et une autre à la fin.

- **Les Avantages :**

- Extrait les frames avec leurs couleurs réel sur la vidéo, ce qui facilite la détection de point de laser.
- Ne demande aucune bibliothèque native, ce qui nous assure la portabilité.

Par contre les bibliothèques ci-dessous, lors de son utilisation, ne donnait aucun résultat :

**Jcodec** : est une source pure de mise en œuvre des codecs vidéo et audio et les formats ouverts java. [17]

**JMyron** : est une bibliothèque externe pour le traitement et qui permet la manipulation de l'image sans avoir à coder en dur tout. Pour utiliser JMyron, on doit télécharger le fichier JAR et de le mettre dans le chemin correcte .Ensuite importer la bibliothèque dans chaque esquisse de traitement où les objets JMyron seront utilisés. [18]

### 5. Les critères d'une bibliothèque parfaite :

- L'accès direct au frame [par index] (Comme JMF)
- Précision d'extraction de frame avec les bonnes couleurs (Comme Xuggler)
- Détection de la fréquence de la vidéo automatique (Comme VLCJ).

### 6. Algorithmes de réalisation :

Dans la plupart des cas, les objets et personnages sont construits suivant des modèles à trois dimensions (on définit des volumes avec une texture de surface), et les algorithmes informatiques traitent ces modèles pour générer l'image en simulant une lumière incidente (effet d'une lampe, laser...etc.) et en traitant les différents effets (notamment les ombres et réflexions).

On mesure l'efficacité d'un algorithme notamment par sa durée de calcul, par sa consommation de mémoire RAM (en partant du principe que chaque instruction a un temps d'exécution constant), par la précision des résultats obtenus, sa scalabilité (son aptitude à être efficacement parallélisé), etc.

Le but est d'évacuer la pensée du calcul, afin de le rendre exécutable par une machine numérique (ordinateur, ...). On ne travaille donc qu'avec un reflet numérique du système réel avec qui l'algorithme interagit. Notre algorithme se présente comme suit :

- 1- Préparer une liste qui contiendra les points 3D qui serviront pour construire le script final **List<List<Point>>L**.
- 2- Lire les fichiers et parcourir les vidéos images par image.
- 3- Récupérer les points 2D de chaque couple d'image (face/profil).
- 4- Test
  - Si le point est d'une couleur verte (celle du laser) : en appliquant la comparaison sous le système de couleur RGB tel que :  
$$(R < 100) (G \geq 200) (B < 100)$$
  - Sinon ignorer le point.
- 5- Calculer le centre de chaque point 2D.  
**Point2D getCenter(x,y,Image)**
- 6- On teste
  - Si le point 2D de l'image face existe alors on passe à l'image de profil où on cherchera l'existence du même point 2D.

- a- Si la condition est satisfaisante (existence du point 2D\_Profil) on construit le Point 3D(x,y,z).
- b- Sinon on construit le point 3D avec une profondeur 0, qu'on met dans une liste d'attente(x,y,0) jusqu'au prochain point et qui affectera sa valeur de profondeur  $z < 0$  pour tous les points mis en attente, puis on vide la liste .
- Sinon on ignore l'image.

7- Calculer les coordonnées de point\_3D en faisant appel à la fonction de l'étape 3.

### Point getPoint\_3D(ImageFace, Imageprofil)

- 8- On ajoute chaque point 3D créé à la liste initiale L.
- 9- On compare l'abscisse(x) du point p et l'abscisse du point(P\_Pred) précédent de la liste L si  $P.X < P\_PRE.X$  donc P est le premier point d'une nouvelle ligne ou d'une nouvelle liste L(i) dans la liste L, puis aller à l'étape 2 jusqu'à épuisement de tous les points.
- 10- On parcourt la liste L de point3D construite.
- 11- On établit des liaisons entre les points pour construire des Surfaces (Polygones) et affecter des identifiants aux Polygones selon l'ordre de création des points.
- 12- Construire le script final String MaxScript()

## 7. Environnement technique de développement :

### 7.1.Les Logiciels utilisés :

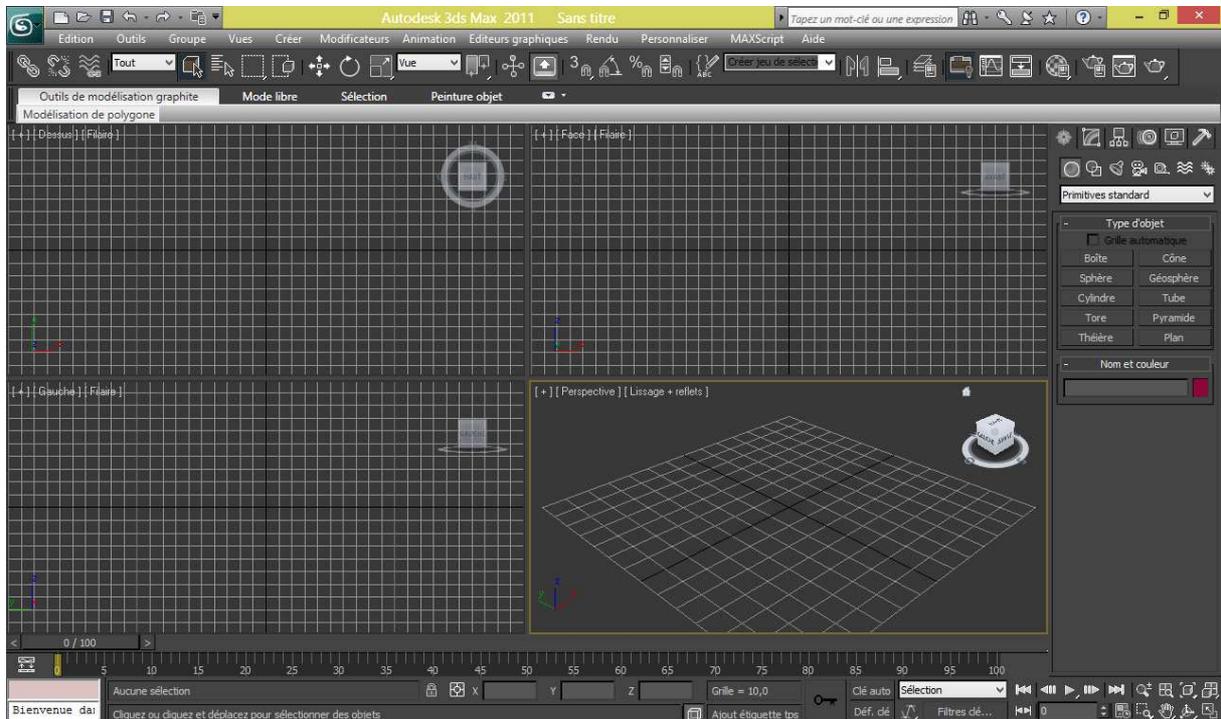
#### ❖ Autodesk 3ds max : [19]

Autodesk 3ds Max (ou simplement 3ds Max ou 3ds) est un logiciel de modélisation et d'animation 3D, développé par la société Autodesk. Avec Maya, Softimage XSI, Lightwave, Houdini et Blender, il est l'un des logiciels de référence dans le domaine de l'infographie 3D.



3Ds Max est conçu sur une architecture modulaire, compatible avec de multiples *plugins* (extensions) et les scripts écrits dans un langage propriétaire appelé Maxscript, et intègre de très nombreuses fonctionnalités, comme le moteur de rendu Mental Ray, Le lo-

logiciel 3ds Max s'est développé rapidement, en étant utilisé principalement dans les industries vidéoludique, cinématographique et automobile.



**Figure III.11 : Interface Autodesk 3ds max.**

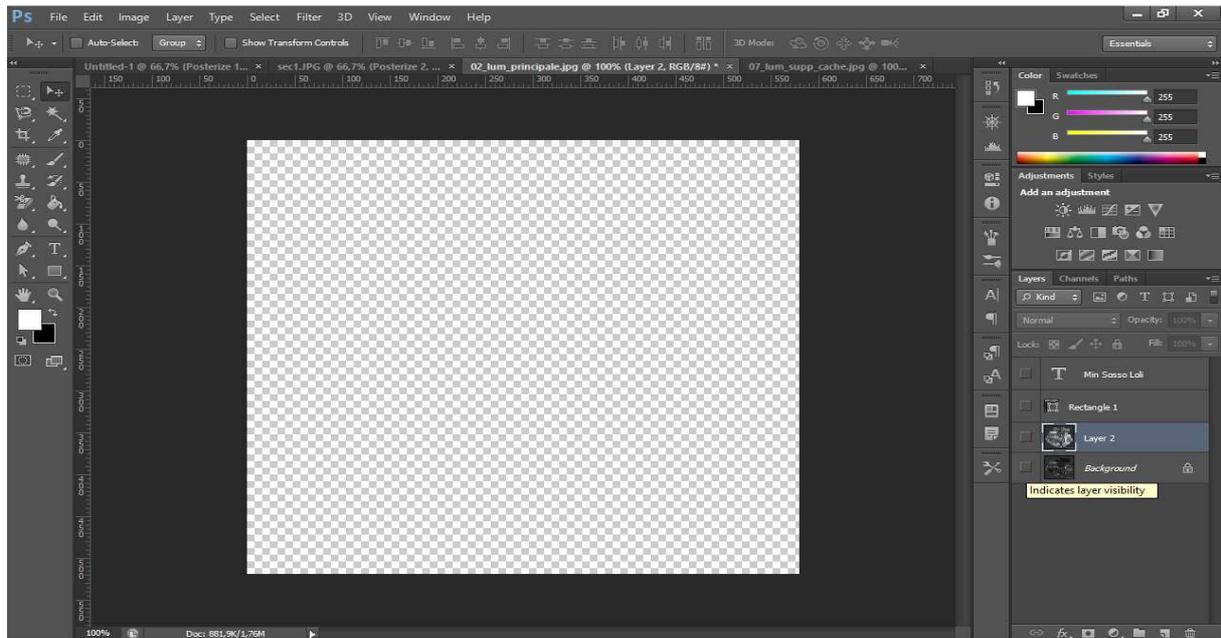
### ❖ Photoshop CC : [20]

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour les traitements de photographies numériques.

Photoshop travaille sur les images matricielles (également appelées "bitmap", à ne pas confondre avec le format d'enregistrement Windows bitmap) car les images sont constituées d'une grille de points appelés pixels. L'intérêt de ces images est de reproduire des graduations.

Photoshop possède son propre format de projet (PSD), qui est plus qu'un simple format de fichier. Le programme accepte également d'importer et d'exporter des fichiers d'image dans les formats les plus courants (GIF, JPEG, TIFF, PNG, ILBM, etc.).





**Figure III.12 : Interface Photoshop CC.**

### 7.2.Langage utilisé :

#### Java : [9]

Le langage java est apparu en fin 1995 début 1996 et développé par Sun Microsystems, il s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur. Destiné au départ à la programmation de centraux téléphoniques, la société Sun a eu l'idée de le recentrer sur les applications de l'internet et des réseaux.

#### Pourquoi Java :

- Portabilité excellente
- Langage puissant orienté objet
- Langage de haut niveau
- JDK très riche
- Nombreuses librairies tierces
- Très grande productivité
- Applications plus sûres et stables
- Nombreuses implémentations, JVM et compilateurs, libres ou non

- Haut niveau d'abstraction
- Le Java évite des problèmes de sécurité liée au dépassement de mémoire tampon

### 7.3. Environnement de développement :

L'environnement de programmation utilisé pour réaliser notre travail est «Eclipse, Netbeans».

#### A) Netbeans 8.0.1 :

NetBeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

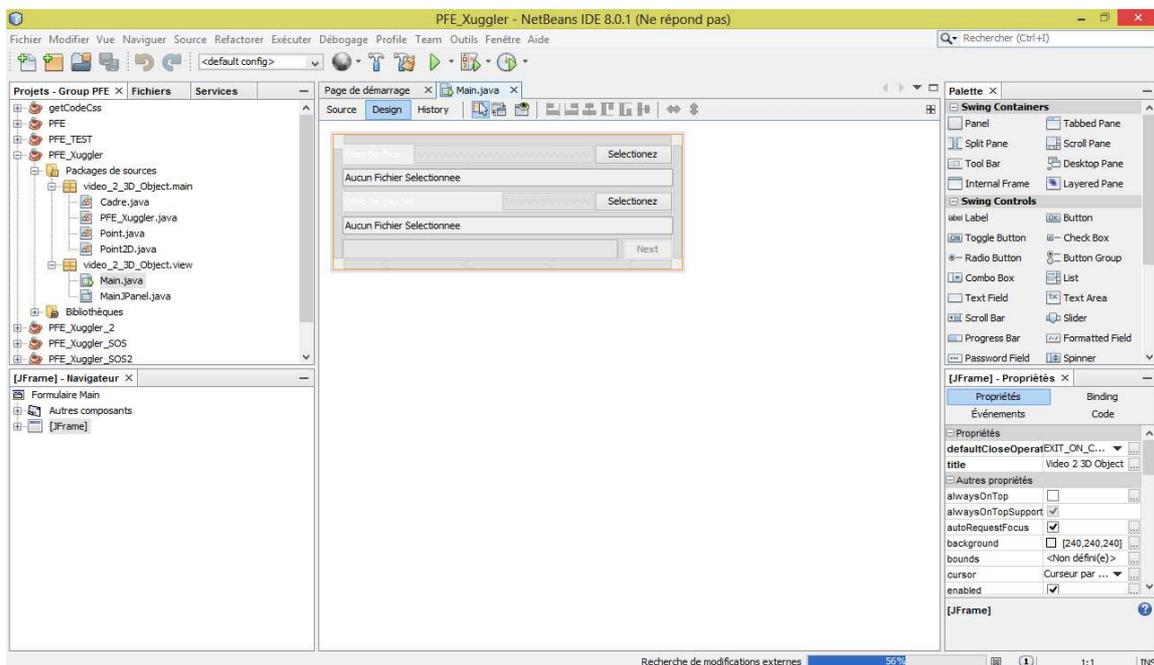


Figure III.13 : Interface NetBeans.

## B) Eclipse :

**Eclipse IDE** est un environnement de développement intégré libre (le terme *Eclipse* désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM) et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

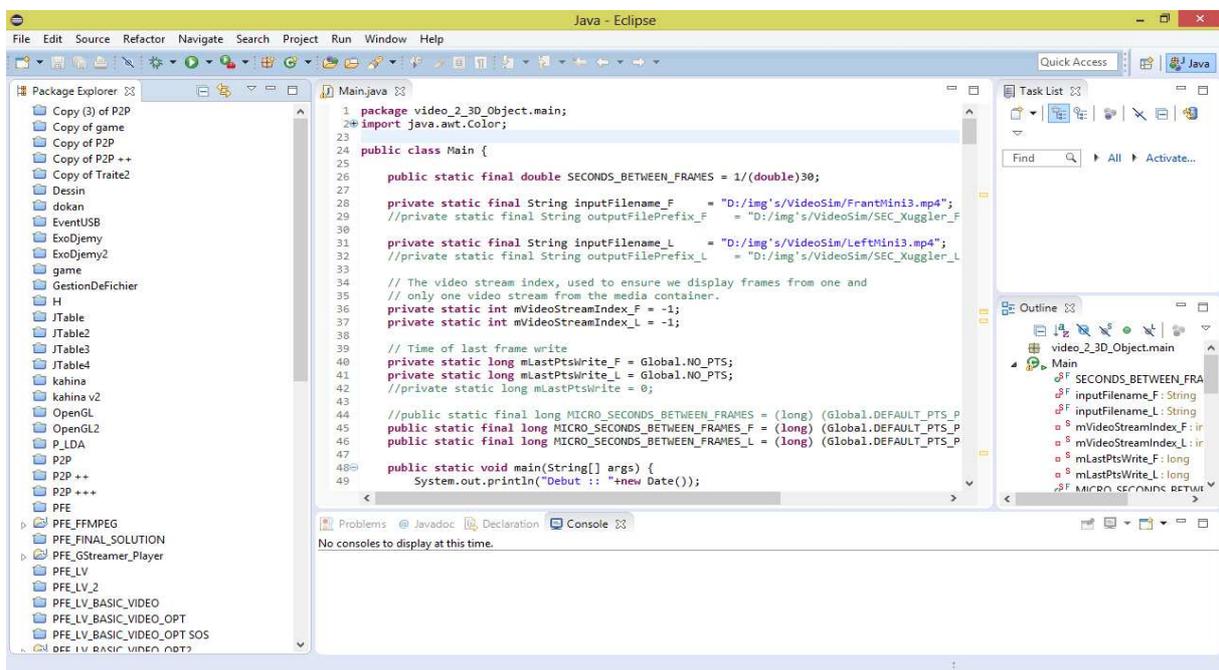


Figure III.14 : Interface Eclipse.

### 8. Présentation de quelques interfaces :

#### ➤ Interface d'accueil de L'application :

L'application est programmée avec le langage JAVA, elle prend deux vidéos en entre pour générer un fichier script en sortie, le fichier est enregistré automatiquement sur le disque dur dans le dossier des scripts 3Ds MAX.

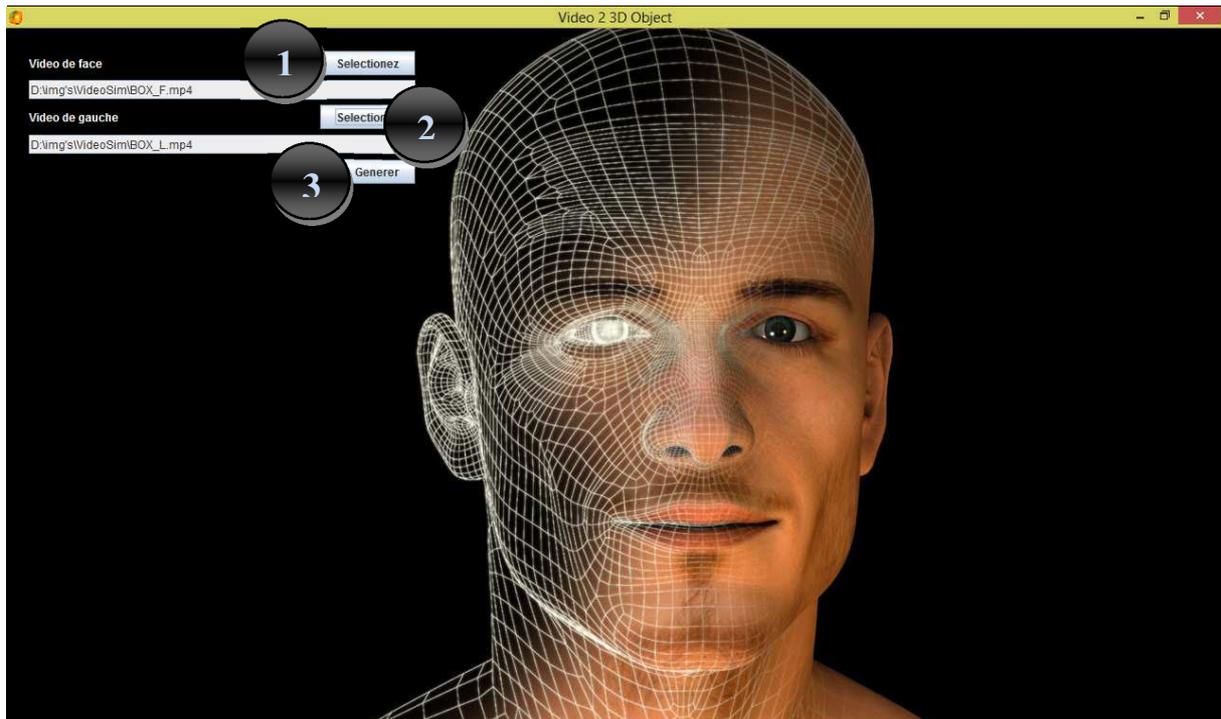


Figure III.15 : Interface d'accueil de l'application.

- 1** Sélectionner la vidéo de gauche (profil)
- 2** Sélectionner la vidéo de face
- 3** Cliquer sur Générer pour  
Générer le fichier script (.ms)  
Enregistrement automatique dans disque dur

### ➤ Interface de logiciel 3ds max :

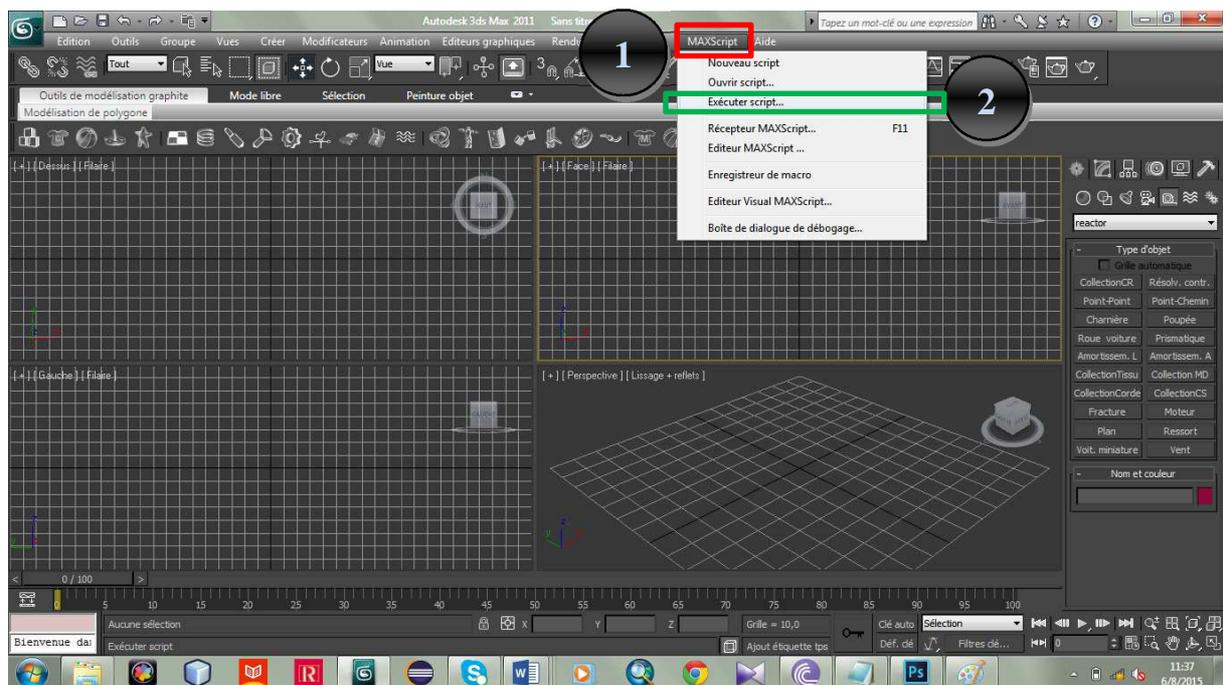
Pour avoir l'objet en 3D il faut cliquer sur la barre de menus MaxScript >Exécuter Script puis sélectionner un script de l'extension '.ms' puis l'ouvrir. L'objet sera construit automatiquement sur la scène 3Ds MAX.

```

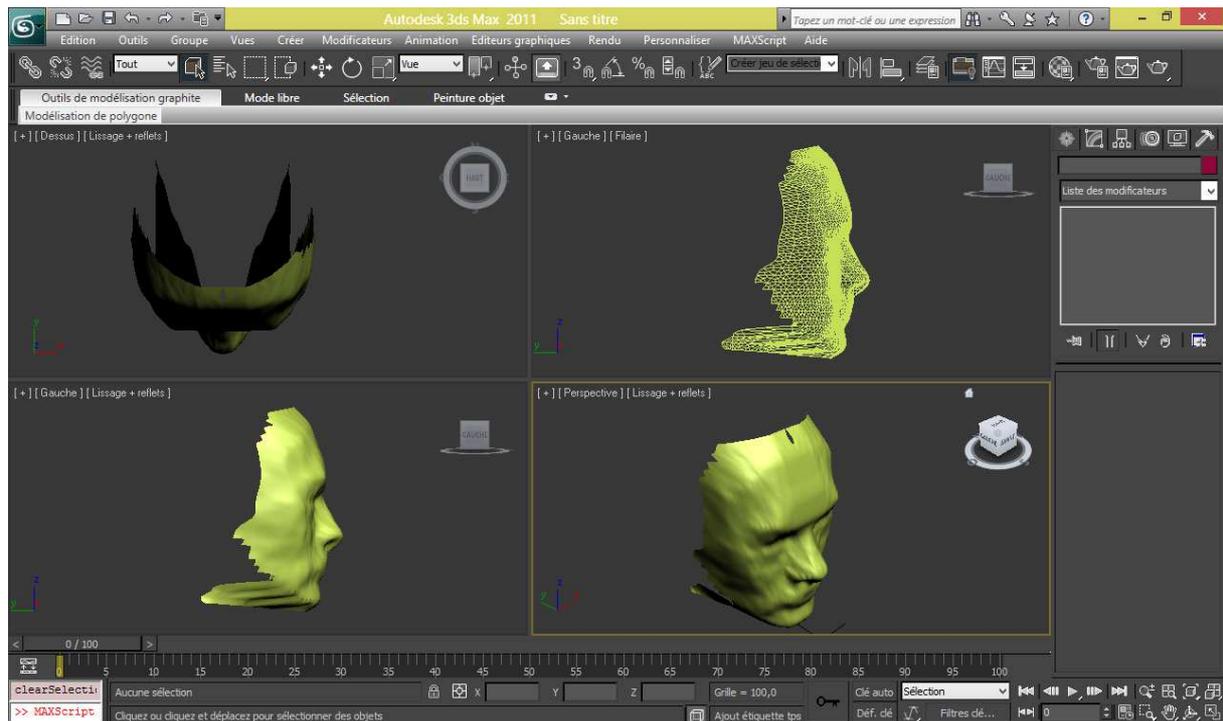
1 mesh vertices:#([23,103,417],[29,103,417],[35,103,417],[40,103,417],[46,103,417],[52,103,417],[58,103,416],[64,105,
416],[69,108,415],[75,110,415],[80,113,414],[86,116,413],[92,119,413],[97,123,412],[103,128,411],[108,132,410],[113,
136,409],[119,142,408],[124,149,407],[129,156,405],[129,90,405],[135,90,405],[141,90,405],[147,90,405],[152,90,405],[158,90
,404],[164,92,404],[170,95,403],[175,98,403],[181,100,402],[187,104,401],[192,107,401],[198,110,400],[103,114,399],[109,118,
398],[114,123,397],[119,129,396],[125,136,395],[130,144,394],[135,151,392],[140,162,390],[23,78,392],[29,78,393],[35,
78,392],[41,78,392],[47,78,392],[53,78,392],[59,78,392],[64,80,391],[70,82,390],[76,85,390],[82,88,390],[87,91,389],[
93,94,388],[98,97,388],[104,101,387],[109,106,386],[115,111,385],[120,116,384],[126,123,383],[131,131,382],[136,140,
380],[140,140,379],[145,167,376],[123,69,379],[29,69,379],[35,69,379],[41,69,379],[47,69,378],[53,69,378],[59,69,379
],[65,71,378],[70,72,378],[76,74,378],[82,76,377],[88,78,376],[94,81,376],[99,84,375],[105,89,375],[110,94,374],[116,
99,373],[121,104,372],[126,110,371],[132,119,370],[137,129,368],[141,139,366],[146,153,364],[150,173,361],[23,64,365
],[29,64,365],[35,64,365],[41,64,365],[47,64,365],[53,64,365],[59,64,365],[65,65,364],[71,67,364],[77,68,364],[82,70,
363],[88,72,363],[94,74,362],[100,76,362],[105,80,362],[111,83,361],[117,86,360],[122,91,359],[127,98,358],[132,108,
357],[137,118,355],[142,128,354],[147,141,352],[151,160,349],[154,180,347],[23,59,351],[29,59,351],[35,59,351],[41,59
,351],[47,59,351],[53,59,351],[59,59,351],[65,60,350],[71,62,350],[77,62,350],[83,64,350],[89,66,349],[94,68,348],[
100,72,348],[106,75,348],[111,78,347],[117,81,347],[123,84,346],[128,90,344],[133,97,344],[138,107,342],[143,118,341
],[148,129,340],[152,147,337],[155,175,333],[23,54,337],[29,54,337],[35,54,337],[41,54,337],[47,54,336],[53,54,336],[
59,54,336],[65,55,336],[71,56,336],[77,56,334],[83,58,335],[89,61,335],[94,64,334],[100,68,334],[106,71,333],[111,74,
333],[117,78,332],[123,83,331],[128,89,330],[133,96,330],[138,103,329],[144,110,328],[149,118,327],[152,142,324],[155
,172,320],[23,48,322],[29,48,322],[35,48,322],[41,48,322],[47,48,322],[53,48,322],[60,48,322],[65,49,322],[71,51,322
],[77,51,322],[83,52,321],[89,56,321],[95,59,320],[101,63,319],[106,68,319],[112,72,318],[117,77,318],[123,82,317],[
128,87,316],[133,94,315],[139,102,314],[144,110,313],[149,119,312],[153,138,310],[155,170,307],[23,43,307],[29,43,307
],[35,43,307],[42,43,307],[47,43,307],[53,43,307],[60,43,307],[65,44,308],[71,44,307],[77,46,307],[83,47,307],[89,49,
306],[95,52,306],[101,57,305],[106,63,304],[112,70,304],[117,75,303],[123,80,302],[128,86,302],[133,94,301],[139,102,
300],[144,111,299],[149,120,298],[153,140,296],[155,168,293],[23,41,293],[29,41,293],[35,41,293],[41,41,293],[47,41,
293],[54,41,293],[60,41,293],[66,41,293],[72,40,293],[78,42,293],[84,42,291],[90,43,292],[95,45,292],[101,50,291],[
107,56,290],[112,63,290],[118,69,289],[123,76,288],[129,84,287],[134,93,286],[139,102,285],[144,111,285],[149,121,284
],[152,144,281],[156,168,279],[23,39,278],[29,39,278],[35,39,278],[41,39,278],[48,39,278],[54,39,278],[60,39,278],[66
,39,278],[72,39,278],[78,39,278],[84,39,278],[90,39,277],[96,39,277],[102,43,277],[107,48,276],[113,54,276],[118,59

```

**Figure III.16 : Extrait de fichier MaxScript '.ms'.**



**Figure III.17 : Interface de logiciel 3D.**



**Figure III.18 : Interface de résultat de l'Objet 3D.**

### **Conclusion :**

Nous avons présenté dans ce chapitre, les étapes de conception et réalisation de notre projet et les différents outils utilisés pour le montage et la réussite de notre application, comme nous avons présenté quelques interfaces réalisées grâce à ces outils.

# *Conclusion Générale*

Notre projet consiste en une modélisation automatique d'un visage en 3D, la chose qui se fait en utilisant un matérielle, ce dernier est cher et aussi indisponible, c'est ce qui nous a poussés réellement à réaliser ce travail.

La réalisation de ce travail nous a permis d'accroître nos connaissances dans le domaine de la programmation java, d'acquérir une expérience dans le domaine de la 3D, elle nous a aussi permis aussi d'élargir nos connaissances ainsi que nos capacités d'analyse et d'interprétation.

Les difficultés que nous avons rencontré en réalisons ce travail consiste principalement aux difficultés trouver pendant la programmation et en manque de documentations concernant le domaine de 3D.

Enfin nous espérons que notre travail aura de grands intérêts pour le département informatique, nous espérant aussi qu'il va être un guide efficaces pour les prochaines promotions.

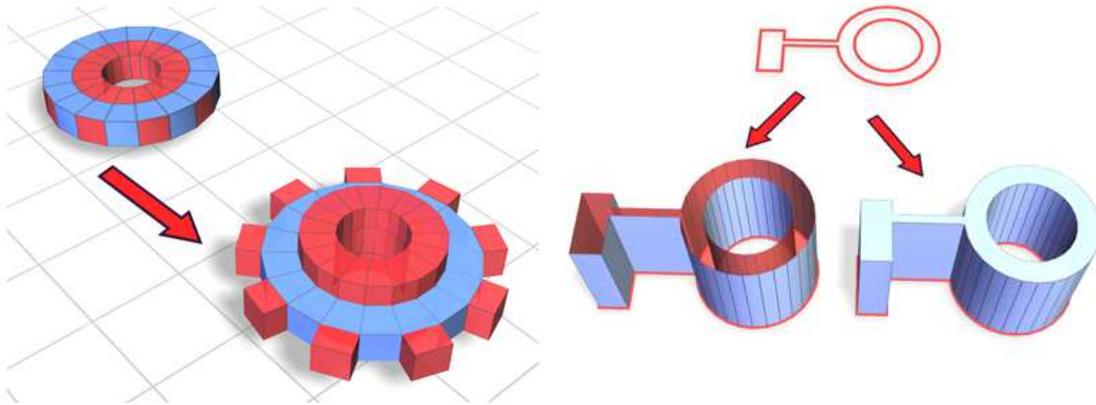
# Bibliographie

- [1]- (M. BERGOUNIOUX – Méthode mathématiques pour le Traitement d'image, (cours master université paris6, 2009).
- [2]- (Antoine MANZANERA- Les images numériques.)
- [3]- <http://fr.wikipedia.org/wiki/Pixel>
- [4]- (<http://fr.wikipedia.org/wiki/Luminance>).
- [5]- [http://xphilipp.developpez.com/articles/filtres/?page=page\\_3](http://xphilipp.developpez.com/articles/filtres/?page=page_3)
- [6]- [http://fr.wikipedia.org/wiki/Infographie\\_tridimensionnelle#Variantes](http://fr.wikipedia.org/wiki/Infographie_tridimensionnelle#Variantes) ;
- [7]- [http://www.wipo.int/wipo\\_magazine/fr/2013/02/article\\_0004.html](http://www.wipo.int/wipo_magazine/fr/2013/02/article_0004.html) ;
- [8]- [[url=http://dico.studiovitamine.com/definition,141,fr/3d,375,fr.html?id=177](http://dico.studiovitamine.com/definition,141,fr/3d,375,fr.html?id=177)] Définition du : 3D [URL]
- [9]- <http://info-rital.developpez.com/tutoriel/java>.
- [10]- <http://fr.wikipedia.org/wiki/Contraste>.
- [11]- M<sup>r</sup> SAM anis et M<sup>elle</sup> Djefel ouzna, ≤ segmentation d'image de documents numérisé : application aux manuscrits arabes anciens ≥, Mémoire ingénieur, UMMTO ,2009/2010.
- [12]- <http://www.oracle.com/technetwork/articles/javase/index-jsp-140239.html>
- [13]- <http://caprica.github.io/vlcj/>
- [14]- <http://gstreamer.freedesktop.org/>
- [15]- <https://www.ffmpeg.org/>
- [16]- <http://www.xuggle.com/xuggler/>
- [17]- <http://jcodecs.org/>
- [18]- <http://webcamxtra.sourceforge.net/>
- [19]- <http://www.autodesk.fr/products/3ds-max/overview>
- [20]- <http://www.adobe.com/fr/products/photoshop.html>

# Annexe

## Extrusion :

L'extrusion consiste à surélever une ou plusieurs faces (adjacentes ou non) ou un profil 2D le long d'une trajectoire et de créer les faces venant combler le vide occasionné par le déplacement de l'élément de départ. Par exemple, extruder un cercle donne un cylindre ouvert ou un tuyau qui suit la trajectoire.



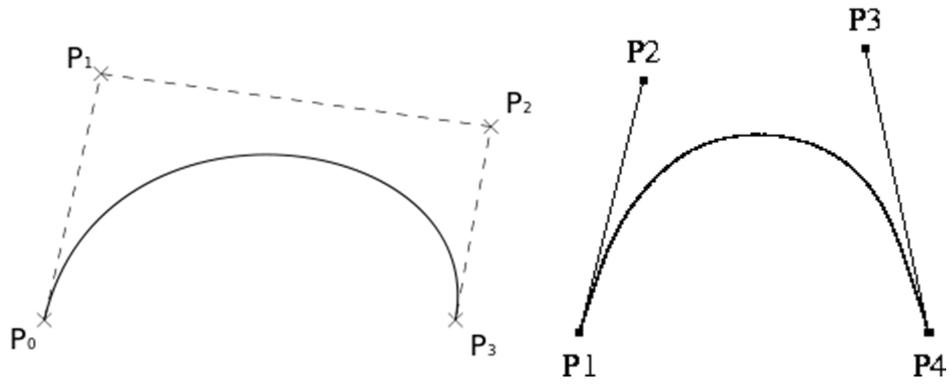
Exemple d'extrusion

## Courbe de Bézier :

Les **courbes de Bézier** sont des courbes polynomiales paramétriques décrites pour la première fois en 1962 par Pierre Bézier (ingénieur Arts et Métiers et Supélec à la régie Renault dans les années 1950) qui les utilisait pour concevoir des pièces d'automobiles à l'aide d'ordinateurs. Elles ont de nombreuses applications dans la synthèse d'images et le rendu de polices de caractères. Elles ont donné naissance à de nombreux autres objets mathématiques.

Il existait avant Bézier des courbes d'ajustement nommées splines, mais dont le défaut était de changer d'aspect lors d'une rotation de repère, ce qui les rendait inutilisables en CAO. Bézier partit d'une approche géométrique fondée sur la linéarité de l'espace euclidien et la théorie, déjà existante, du barycentre : si la définition est purement géométrique, aucun repère n'intervient puisque la construction en est indépendante, ce qui n'était pas le cas pour les *splines* (les *splines* conformes aux principes de Bézier seront par la suite nommées B-splines).

Les courbes de Bézier sont définies généralement avec des points de contrôle et un mode de lissage, ou des points de contrôle et des points de tangente.



**Courbes de Bézier**

**Exemple de Cube.obj :**

**# Nom de l'objet**

**o cube**

**# Sommets**

**v 0.0 0.0 0.0**

**v 0.0 0.0 1.0**

**v 0.0 1.0 0.0**

**v 0.0 1.0 1.0**

**v 1.0 0.0 0.0**

**v 1.0 0.0 1.0**

**v 1.0 1.0 0.0**

**v 1.0 1.0 1.0**

**# Normales**

**vn 0.0 0.0 1.0**

**vn 0.0 0.0 -1.0**

**vn 0.0 1.0 0.0**

vn 0.0 -1.0 0.0

vn 1.0 0.0 0.0

vn -1.0 0.0 0.0

vt 0.25 0.0

vt 0.5 0.0

vt 0 0.25

vt 0.25 0.25

vt 0.5 0.25

vt 0.75 0.25

vt 0.0 0.5

vt 0.25 0.5

vt 0.5 0.5

vt 0.75 0.5

vt 0.25 0.75

vt 0.5 0.75

vt 0.25 1.0

vt 0.5 1.0

# Faces

f 1/11/2 7/14/2 5/12/2

f 1/11/2 3/13/2 7/14/2

f 1/7/6 4/4/6 3/3/6

f 1/7/6 2/8/6 4/4/6

f 3/1/3 8/5/3 7/2/3

f 3/1/3 4/4/3 8/5/3

f 5/10/5 7/6/5 8/5/5

f 5/10/5 8/5/5 6/9/5

f 1/11/4 5/12/4 6/9/4

f 1/11/4 6/9/4 2/8/4

f 2/8/1 6/9/1 8/5/1

f 2/8/1 8/5/1 4/4/1