



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

**UNIVERSITE IBN KHALDOUN**  
**MEMOIRE**

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE  
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

**MASTER**

Spécialité : Génie Informatique

Par :

**Meziani Imane**

Sur le thème

---

**Un système de Navigation 3D**

---

Soutenu publiquement le 30 /10/ 2018 à Tiaret devant le jury composé de :

|                     |     |                         |             |
|---------------------|-----|-------------------------|-------------|
| Mr. Djaafri Laouni  | MCA | Université Ibn Khaldoun | Président   |
| Mr. Alem Abdelkader | MAA | Université Ibn Khaldoun | Encadreur   |
| Mr. Daoud Med Amine | MCA | Université Ibn Khaldoun | Examinateur |

Année universitaire 2017/2018

## *Remerciement*

*Nous tenons à remercier Dieu le tout puissant ; qui nous a donné la volonté, la santé et la patience tout au long de nos études pour atteindre ce stade.*

*Nous remercions très vivement notre encadreur Mr : Alem abdelkader pour son aide, ses conseils, et surtout pour sa patience.*

*Nos remerciements sont aussi exprimés aux membres du Jury, qui feront l'honneur d'évaluer et de juger notre modeste travail*

*Nous remercions aussi tous ceux qui nous ont aidé de près ou de loin à l'élaboration de ce travail.*

## *Dédicace*

*Je dédie ce modeste travail :*

*A mes chers parents pour leur patience, leur soutien, et leurs sacrifices que  
dieu les protège.*

*Je le dédie à toute ma famille qui a toujours été là pour me soutenir et  
m'aider à surmonter les difficultés.*

*A mes chers frères : Mohamed et sa femme Khadidja, Youcef et Djellali.*

*A mes chères sœurs : Fatima et Nouria.*

*Mes nièces : Yasmine , Rania et hanaa.*

*A mes chères amies.*

*Je le dédie à tous ce qui m'a donné leur moindre coup de poussé pour le  
réaliser.*

*Imane*

# Table des matières

## Tables des matières

|   |    |
|---|----|
| Introduction générale .....   | 08 |
| Chapitre I : Traitement d'images et Vision artificielle.....  | 09 |
| I.1 Introduction.....   | 10 |
| I.2 Qu'est-ce qu'une image.....   | 10 |
| I.3 Le traitement d'image.....  | 10 |
| I.3.1 Acquisition d'une image.....  | 11 |
| I.3.2 Caractéristiques d'une image numérique.....   | 11 |
| I.3.2.1 Dimension.....  | 11 |
| I.3.2.2 Résolution.....   | 11 |
| I.3.2.3 Bruit.....  | 12 |
| I.3.2.4 Histogramme.....  | 12 |
| I.3.2.5 Luminance.....  | 12 |
| I.3.2.6 Contraste.....  | 12 |
| I.3.2.7 Images à niveaux de gris.....   | 13 |
| I.3.2.8 Images en couleurs.....   | 13 |
| I.3.3 Outils fondamentaux.....  | 13 |
| a) L'histogramme.....   | 13 |
| b)La convolution.....   | 14 |
| c) La transformée de Fourier.....   | 16 |
| I.4. La Vision artificielle: .....  | 16 |
| I.4.1 Vision et système de vision .....   | 16 |
| I.4.2 Qu'est-ce que la vision artificielle ? .....  | 17 |
| I.4.3 Où se situe la vision par ordinateur? .....   | 17 |
| I.4.4 Techniques utilisées en vision par ordinateur.....  | 18 |
| I.4.4.1 La segmentation.....  | 18 |
| I.4.4.2 La reconstruction 3D .....  | 19 |
| I.4.5 Les applications de la vision par ordinateur.....   | 19 |
| I.5. Conclusion .....   | 20 |
| Chapitre II : La navigation 3D.....   | 21 |
| II.1 Introduction.....  | 22 |
| II.2. Rappels mathématiques et géométriques.....  | 22 |
| II.2.1 Repères.....   | 22 |
| II.2.2 Transformations.....   | 23 |
| II.2.2.1 Translation.....   | 23 |
| II.2.2.2 Rotation.....  | 23 |
| II.2.2.3 Homothétie.....  | 24 |
| II.2.3 Les matrices.....  | 24 |
| II.2.4 Rappel sur quelques transformations géométriques en 2D et en 3D.....                               | 25 |
| II.3 Qu'est-ce que la 3D.....   | 26 |
| II.3.1 Généralités sur la 3D.....   | 26 |
| II.3.1.1 Pourquoi l'utiliser ? .....  | 26 |
| II.3.1.2 Comment expliquer que sur un écran en deux dimensions nous pouvons obtenir une sortie 3D ? ..... | 26 |
| II.3.2 Comment fonctionne la 3D.....  | 27 |
| II.3.3 Les domaines d'utilisation de la 3D.....   | 28 |
| II.3.3.1 Utilisation de la 3D dans la chimie.....   | 28 |
| II.3.3.2 L'utilisation de la 3D dans le domaine de l'astronomie.....                                      | 29 |
| II.4. Visualisation des scènes.....   | 30 |
| II.4.1 Deux méthodes de rendu 3D : différences et utilités.....   | 30 |
| II.4.2 Les bases de la 3D temps réel.....   | 32 |

|   |    |
|---|----|
| II.4.3 Le pipeline 3D.....                            | 32 |
| II.5. La Réalité Virtuelle.....                       | 35 |
| II.5.1 Interaction.....                               | 35 |
| II.5.2 Les techniques d'interaction 3D.....           | 35 |
| II.5.2.1 Qu'est-ce qu'une navigation .....            | 36 |
| II. Les composants de navigation .....                | 36 |
| II. Les types de navigations .....                    | 37 |
| II. 5.2.1.1 La navigation libre.....                  | 37 |
| II.5.2.1.2 La navigation assistée.....                | 38 |
| II.5.2.2 La sélection.....                            | 39 |
| II.5.2.3 La manipulation.....                         | 39 |
| II.6. Navigation 3D.....                              | 40 |
| II.6.1 Les outils de navigation 3D.....               | 41 |
| II.7. Conclusion.....                                 | 42 |
| Chapitre III : IMPLEMENTATION.....                    | 43 |
| III.1 Introduction.....                               | 44 |
| III.2 Choix du langage.....                           | 44 |
| III.3 Choix d'environnement .....                     | 44 |
| III .4 OpenGL.....                                    | 45 |
| III 4.1 Les types de projection OpenGL.....           | 46 |
| III 4.1.1 La projection en perspective .....          | 46 |
| III 4.1.2 La projection parallèle orthographique..... | 47 |
| III.5 Navigation dans un environnement 3D.....        | 47 |
| III.5.1 La problématique .....                        | 47 |
| III.5.2 Implémentation .....                          | 47 |
| III.6 La gestion de l'orientation.....                | 48 |
| III.6.1 Les contrôles .....                           | 48 |
| III.6.1.1 Rotation de la souris.....                  | 48 |
| III.6.1. 2 Gestion du déplacement au clavier.....     | 48 |
| III.6.1. 3 Gestion du regard à la souris.....         | 48 |
| III.6.2 Bibliothèque utilisée .....                   | 49 |
| III.7 Description de la scène.....                    | 49 |
| III.8 Conclusion.....                                 | 52 |
| Conclusion générale et perspectives.....              |    |
| Résumé.....   |    |
| Bibliographie.....                                    |    |

## Liste des figures

### CHAPITRE I

|                   |  |    |
|-------------------|--|----|
| <b>Figure I.1</b> | : égalisation d’histogramme.....                 | 14 |
| <b>Figure I.2</b> | : calcul de convolution.....                     | 15 |
| <b>Figure I.3</b> | : Filtrage d’image par convolution.....          | 15 |
| <b>Figure I.4</b> | : Le fonctionnement d’un système de vision.....  | 16 |
| <b>Figure I.5</b> | : Les différentes disciplines de l’imagerie..... | 17 |
| <b>Figure I.6</b> | : La segmentation d’image.....                   | 18 |
| <b>Figure I.7</b> | : La reconstruction 3D d’image.....              | 19 |

### CHAPITRE II

|                     |   |    |
|---------------------|---|----|
| <b>Figure II.1</b>  | : Différents repères couramment utilisés.....                                     | 22 |
| <b>Figure II.2</b>  | : Translation, rotation, homothétie : les trois transformations élémentaires..... | 24 |
| <b>Figure II.3</b>  | : Fonctionnement de la 3D.....  | 28 |
| <b>Figure II.4</b>  | : Exemple d’utilisation de la 3D – La molécule de Chlorophylle A.....             | 29 |
| <b>Figure II.5</b>  | : Exemple d’utilisation de la 3D – Jupiter et ses satellites.....                 | 29 |
| <b>Figure II.6</b>  | : Principe du lancer de rayons.....   | 32 |
| <b>Figure II.7</b>  | : Les différentes étapes du pipeline 3D.....                                      | 33 |
| <b>Figure II.8</b>  | : Les deux repères (main droite et gauche) utilisés en 3D.....                    | 34 |
| <b>Figure II.9</b>  | : Exemple de techniques de navigation libre.....                                  | 38 |
| <b>Figure II.10</b> | : Exemples de techniques de navigation assistée.....                              | 38 |
| <b>Figure II.11</b> | : Représentation d’une caméra 3D.....   | 40 |

### CHAPITRE III

|                     |                             |    |
|---------------------|-----------------------------|----|
| <b>Figure III.1</b> | : pyramide de clipping..... | 46 |
| <b>Figure III.2</b> |                             |    |
| <b>Figure III.3</b> |                             |    |

## Liste des tableaux

### CHAPITRE II

**Tableau II.1** :L'utilisation des matrices dans les transformations géométriques 2D et 3D.....25

## Introduction Générale

Depuis déjà quelques années, les évolutions technologiques en micro-informatique se succèdent à une vitesse impressionnante pour répondre principalement au besoin toujours croissant de puissance, requise pour les applications multimédias et notamment celles relativement récentes exploitant une interface 3D. En effet, non seulement les machines actuelles ont des microprocesseurs permettant d'effectuer de nombreuses opérations en virgules flottantes nécessaires à la 3D (pour les transformations, pour les projections, etc.) mais aussi, ils embarquent des cartes graphiques bon marché dans lesquelles les fonctions 3D les plus courantes sont câblées (c'est-à-dire prises en charge par les cartes elles-mêmes). Jusqu'à présent, les performances de ces cartes ont quasiment été doublées tous les ans et les premières cartes de troisième génération, intégrant un module pour l'éclairage et les transformations, n'infléchissent pas, loin s'en faut, cette tendance. Les innovations dans le domaine de l'accélération 3D concernent même les microprocesseurs eux-mêmes qui intègrent des instructions spécifiques à la 3D pour les calculs matriciels par exemple [1].

Notre projet de fin d'étude consiste en la mise en place d'un système de navigation en 3D qui peut servir comme outil de simulation visuelle.

La première partie de ce rapport décrit deux grandes disciplines liées à la vision : le traitement d'image et la vision par ordinateur ainsi que les notions de base nécessaires à la compréhension de ces deux domaines.

Dans la deuxième partie, nous verrons le concept de la navigation et ses techniques, Puis nous verrons les outils, s'appuyant généralement sur les API utilisées pour l'édition et/ou le rendu d'une scène 3D. Ces outils sont les modeleurs qui aident à la création d'un « monde » 3D et les plugins, capables d'interpréter certains fichiers décrivant une scène et d'afficher leur contenu.

La troisième partie est consacrée à l'implémentation d'un système de navigation 3D, nous présentons l'interface graphique de notre application .Enfin, nous clôturons ce mémoire par une conclusion générale et quelques perspectives.

# Chapitre I

## Chapitre I

Traitement d'images et Vision artificielle

### Traitement d'images et Vision artificielle

## **I.1 Introduction**

Dans ce premier chapitre, nous abordons les notions de base nécessaires à la compréhension du domaine de l'imagerie, ainsi que les aspects et les disciplines liés à ce domaine. Ensuite, nous présenterons les différentes phases de traitements impliquées dans les processus appliqués dans ce domaine. En particulier, l'étude de ce domaine comporte des phénomènes complexes. Car, il se trouve à la croisée des chemins entre les mathématiques, le traitement du signal et l'intelligence artificielle. De ce fait, nous examinerons les techniques les plus utilisées dans ces différents axes de recherches.

## **I.2 Qu'est-ce qu'une image**

Commençons par nous intéresser à la définition de l'image. Une image est plutôt difficile à décrire d'une façon générale. Il s'agit d'une représentation du monde. En informatique, l'image est représentée sous format numérique, l'image «réelle » va être transformée en une image numérique par différents outils de transformation (caméra, scanner, satellite, etc). Par conséquence, le mot numérique nous conduit à une modélisation de l'image sous forme d'une fonction mathématique pour faciliter les traitements par la suite. [2]

## **I.3 Le traitement d'image :**

Afin d'extraire des informations et améliorer l'aspect visuel de l'image, un ensemble de méthodes et techniques opérants sur celles-ci est nécessaire. Autrement dit, le traitement d'image consiste à appliquer des transformations mathématiques sur des images dans le but d'améliorer leur qualité. On peut citer par exemple : la création d'effets spéciaux au cinéma, l'amélioration de la qualité des photos en corrigeant les conditions de prise de vue, la détection de lettres ou de visages, l'identification de zones cancéreuses en imagerie médicale, la compression JPEG, etc.

Tout système de traitement d'image peut être vu comme une combinaison de deux phases, l'acquisition et le traitement proprement dit. La qualité des résultats dépend de l'algorithme mis en place et de son adéquation au problème posé, et aussi, de la qualité initiale des images traitées.

Avant d'aborder le sujet de traitement d'image il faut savoir d'abord la façon par laquelle est représentée l'image numérique ainsi que sa structure. Dans ce domaine on considère que l'image est une fonction mathématique de  $R \times R$  dans  $R$  où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de

gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de  $R \times R \times R$  dans  $R$ . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de  $R \times R$  dans  $R \times R \times R$ . [2]

### **I.3.1 Acquisition d'une image**

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Afin de rendre l'image lisible et manipulable par un système informatique, il est nécessaire de lui faire subir une transformation. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales :

- les caméras numériques.
- les scanners.

À ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'inter corrélation de deux éléments voisins : plus cette inter corrélation est faible, meilleure est l'image.

### **I.3.2 Caractéristiques d'une image numérique**

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

#### **I.3.2.1 Dimension**

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

#### **I.3.2.2 Résolution**

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour

désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution.

### **I.3.2.3 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.

### **I.3.2.4 Histogramme**

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleurs) dans le cas d'une image trop claire ou d'une image trop foncée. Cependant, il ne faut pas considérer l'histogramme comme une caractéristique fondamentale de l'image dans la mesure où on peut le transformer radicalement sans changer significativement l'image.

### **I.3.2.5 Luminance**

C'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'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise 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; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

### **I.3.2.6 Contraste**

C'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. Si  $L_1$  et  $L_2$  sont les degrés de luminosité respectivement de deux zones voisines  $A_1$  et  $A_2$  d'une image, le contraste  $C$  est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

### **I.3.2.7 Images à niveaux 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. Donc, pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

### **I.3.2.8 Images en couleurs**

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, etc) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.).[2]

## **I.3.3 Outils fondamentaux**

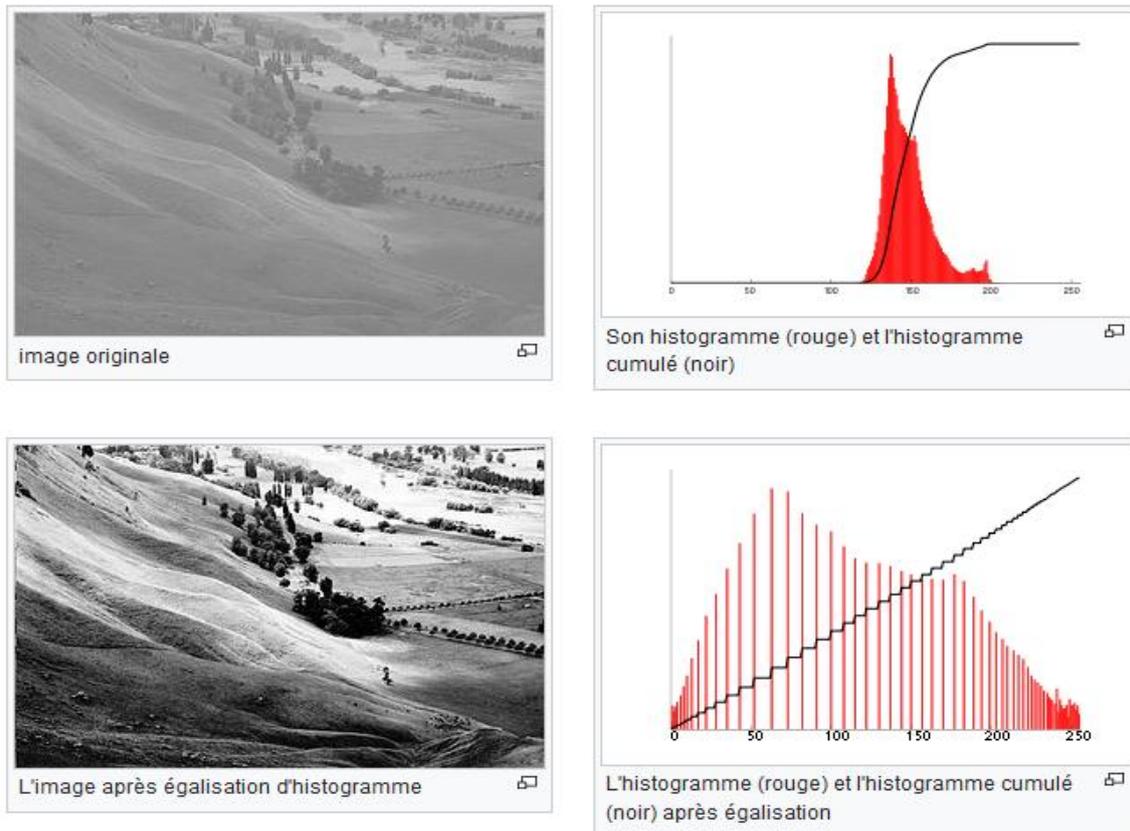
### **a) L'histogramme**

Il n'est pas considéré seulement comme une caractéristique de l'image mais aussi comme un outil de base pour l'étude des capteurs ou de la dynamique d'une scène, il est utilisé par certains opérateurs d'analyse. Cet outil peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci, Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image.

De nombreux traitements peuvent être effectués sur l'image en se basant sur l'histogramme :

- Normalisation : exploiter toute la dynamique de codage.
- Égalisation : équilibrer la dynamique de codage et augmenter le contraste.

- Segmentation : simplifier l'image en regroupant les pixels selon leurs valeurs.



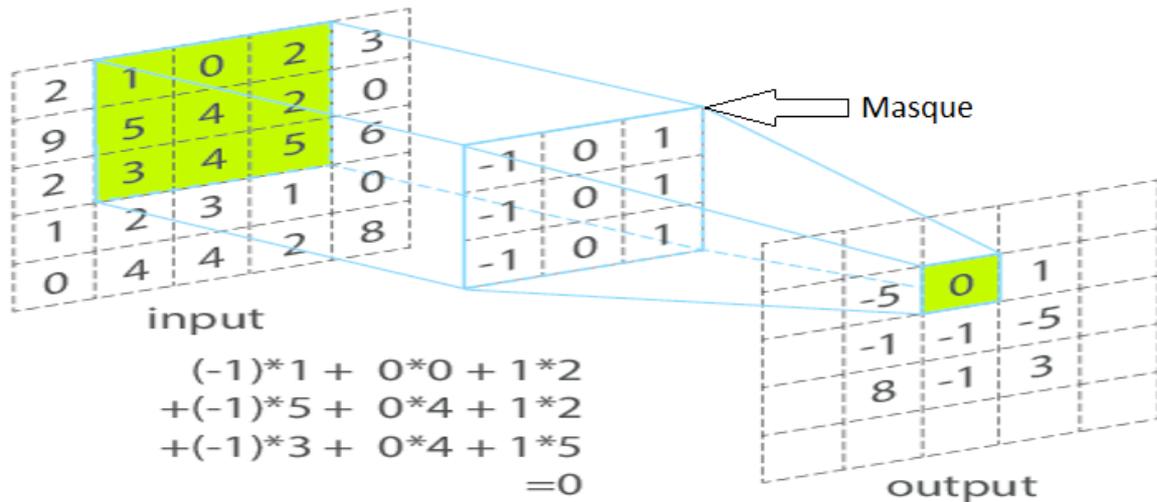
**Figure I.1 : égalisation d'histogramme**

La figure ci-dessous montre un exemple d'égalisation d'histogramme grâce à une transformation des niveaux de gris dont le principe est d'équilibrer le mieux possible la distribution des pixels dans la dynamique (idéalement, on cherche à obtenir un histogramme plat). [3]

### b) La convolution

C'est l'opérateur de base du traitement linéaire des images. Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, ce n'est que plus tard qu'on a fourni des justifications physiques et fait le lien théorique avec les filtres et le traitement du signal. [3]

La convolution consiste à recalculer la valeur d'un pixel donné d'une image de départ (input) en se basant sur la valeur du pixel lui-même et sur la valeur des pixels environnants le pixel à recalculer (**Figure I.2**).

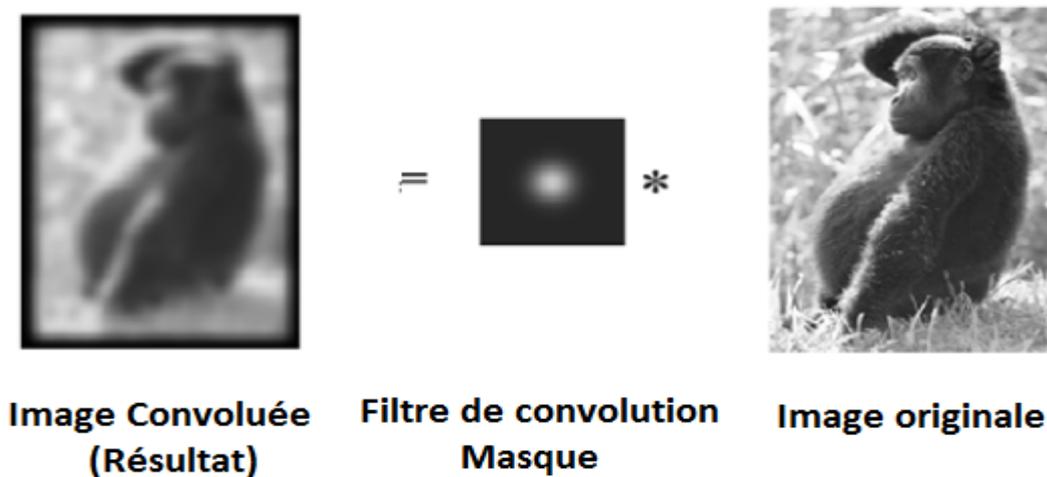


**Figure I.2 : calcul de convolution.**

Chaque pixel environnant est affecté d'un coefficient de calcul, l'ensemble des coefficients utilisés sont regroupés dans un tableau de valeurs (appelé aussi "matrice"). Ce tableau est appelé le masque.

La valeur finale du pixel dans l'image finale (output) correspond à un calcul prenant en compte tous les pixels environnants du pixel de départ et le pixel lui-même, combinés un à un aux coefficients du masque.

La figure suivante illustre le principe du filtrage des images (signal bidimensionnel) par convolution.



**Figure I.3 : Filtrage d'image par convolution.**

### c) La transformée de Fourier

La transformée de Fourier permet d'explorer la composition fréquentielle de l'image, de lui appliquer des opérateurs de filtrage. Lors de cette transformation, le signal est décomposé sur un ensemble de signaux de base qui sont cosinus, sinus ou l'exponentielle imaginaire et forment une base orthogonale (qui permet de supprimer les redondances d'informations). Malgré que son utilisation en tant qu'outil analytique et algorithmique se soit estompée en traitement d'images au profit d'approches mieux adaptées à la localisation spatiale des fréquences (ondelettes), elle reste un outil théorique et pédagogique intéressant en traitement du signal. [3]

## I.4. La Vision artificielle

### I.4.1 Vision et système de vision ?

L'espace qui nous entoure a une structure basée principalement sur les objets. Lorsque l'on demande à une personne de décrire ce qu'elle voit, elle n'éprouve aucune difficulté à discerner et nommer les objets qui l'entourent : téléphone, table, livre, etc. Et pourtant, l'information qui est réellement disponible sur la rétine de ses yeux n'est ni plus ni moins, une collection de points (environ un million !). En chaque point ou pixel (picture element) il y a tout simplement une information qui donne une indication quant à la quantité de lumière et la couleur qui proviennent de l'espace environnant et qui ont été projetées à cet endroit de la rétine. Le téléphone, la table ou le livre n'existent pas sur la rétine. Guidé à la fois par l'information codée dans l'image (ou la rétine) et par ses propres connaissances, le processus visuel construit des percepts.

Le téléphone ou le livre sont les réponses finales, résultant d'un processus d'interprétation qui fait partie intégrante du système de vision (Figure I.4). De plus, il n'y a pas de correspondance terme à terme entre l'information sensorielle (la lumière et la couleur) et la réponse finale (des objets 3D).

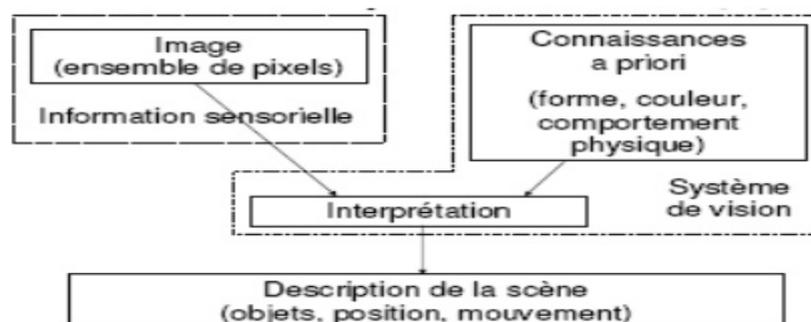


Figure I.4 : Le fonctionnement d'un système de vision.

Comme il est illustré dans la figure ci-dessus, un système de vision doit fournir les connaissances nécessaires afin de permettre une interprétation non ambiguë. [4]

#### I.4.2 Qu'est-ce que la vision artificielle ?

En utilisant un système de vision, la vision artificielle (ou « vision par ordinateur ») est cette discipline qui consiste à convertir une image en données objets ou, plus explicitement, à identifier les objets contenus dans l'image par l'extraction et l'analyse de caractéristiques abstraites (features) à partir des pixels, suivant un processus de reconnaissance de forme similaire à celui opéré par l'humain. En effet, ce domaine est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle «voit» lorsqu'on la connecte à une ou plusieurs caméras. [5]

#### I.4.3 Où se situe la vision par ordinateur?

Dans ce vaste domaine d'imagerie, le traitement d'image représente des traitements de bas niveau sur l'image pour obtenir une autre, tandis que la discipline de vision par ordinateur intervient dans une phase d'interprétation, et de compréhension de l'image pour donner une description à notre scène réelle (la modélisation géométrique) en appelant à l'intelligence artificielle (traitement bas niveau).[6]

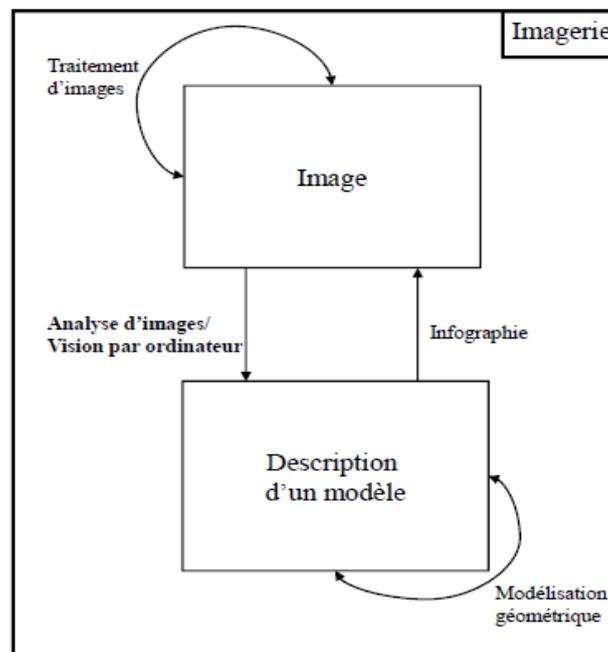


Figure I.5 : Les différentes disciplines de l'imagerie.

Le passage entre le modèle géométrique et l'image, fait appel à une autre discipline plus moderne que les deux autres (traitement d'image et vision artificielle), il s'agit de **l'infographie** (la génération d'images artificielles à partir de données objets).

#### **I.4.4 Techniques utilisées en vision par ordinateur**

Deux grandes approches peuvent être envisagées dans ce domaine (La vision par ordinateur): la segmentation et la reconstruction 3D. Les processus appliqués dans ces deux aspects utilisent des calculs mathématiques complexes à travers les techniques suivantes :

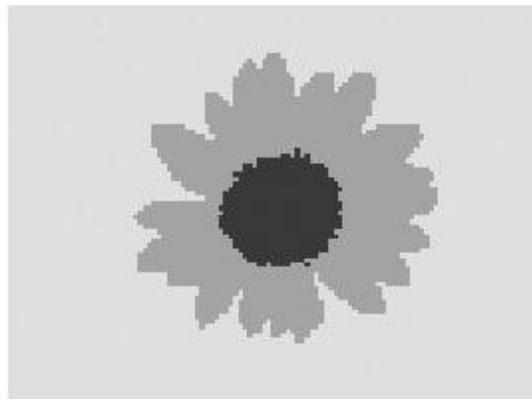
- Transformée de Hough
- Transformée de Randon
- Transformée de Fourier [7]

##### **I.4.4.1 La segmentation**

C'est un processus ayant pour objet de regrouper ensemble des observations ayant des caractéristiques communes. Autrement dit, la segmentation est le découpage d'une image en différentes régions et/ou contours.



**Image originale**



**Image segmentée**

**Figure I.6 : La segmentation d'image**

Cette opération vise à sous-diviser l'image en constituants distincts (des objets) afin de faciliter la tâche de reconnaissance des objets contenus dans celle-ci.

### I.4.4.2 La reconstruction 3D

C'est un processus ayant pour objet de retrouver la forme 3D d'objets vus par une ou plusieurs caméras. Pour certaines applications, l'objectif est d'estimer la distance des objets à l'observateur.

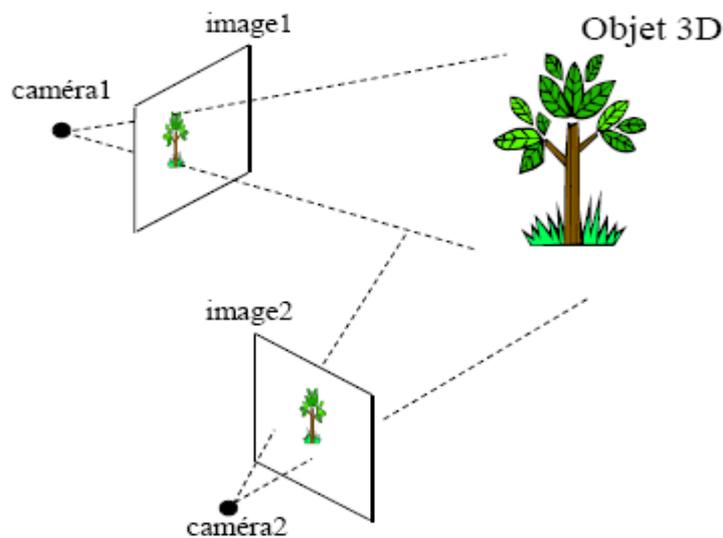


Figure I.7 : La reconstruction 3D d'image.

La reconstruction 3D d'objets à partir de différentes vues 2D est un problème souvent étudié dans les systèmes de vision par ordinateur, la robotique, la productique, l'infographie, et d'autres applications. Plusieurs méthodes ont été élaborées selon le nombre de vues existantes. On vise, à travers l'application de ces méthodes, de retrouver la profondeur des points visibles sur l'image, qui est perdue durant le processus de projection qui produit l'image (capture de l'image en utilisant la caméra). [6]

### I.4.5 Les applications de la vision par ordinateur

La vision par ordinateur a réalisé des progrès énormes grâce aux avancées de l'électronique et de l'informatique, ainsi qu'aux avancées méthodologiques. Ces dernières ont rendu possible l'utilisation de la vision en plusieurs applications:

- **La robotique** : la vision active est très utilisée en robotique où on utilise le retour d'informations du contrôle pour améliorer les traitements sur l'image et orienter les robots mobiles.
- **La vision industrielle** : le contrôle visuel sur une chaîne de fabrication en inspectant les objets, de façon automatique, afin de permettre le déroulement du processus industriel.
- **Reconnaissance d'écriture (OCR)** : de nombreux logiciels de conversion PDF-WORD utilise la vision par ordinateur pour la reconnaissance des lettres. [8]

## **I.5. Conclusion**

Dans ce chapitre, Nous avons présenté les deux grandes disciplines de l'imagerie ainsi que les méthodes et les approches les plus utilisées dans chacune. Cependant, le traitement d'image et la vision sont des disciplines relativement jeunes (~années 60) et qui évoluent rapidement. Elles sont en plein expansion et donnent lieu chaque année à une profusion de travaux, académiques, technologiques et industriels.

Chapitre II  
**Chapitre II**

La Navigation 3D  
**La Navigation 3D**

## II.1 Introduction

Il est généralement admis que l'interaction 3D regroupe trois catégories de tâches : la tâche de navigation, la manipulation et la sélection [9]. Ces trois tâches sont essentielles au pilotage d'une application 3D. Dans ce chapitre nous nous concentrerons à une tâche propre à l'interaction 3D qu'est la navigation car elle constitue le thème central de ce travail, avant tous nous illustrons par un rappel mathématique et géométrique. D'une part, nous présentons les fondements de la 3D, nous y voyons les intérêts de son utilisation et les domaines principaux dans lesquels ces interfaces sont utilisées. D'autre part, nous décrivons comment sont obtenues les images sur un écran 2D à partir d'une scène décrite en coordonnées 3D. [10]

## II.2. Rappels mathématiques et géométriques

### II.2.1 Repères

Afin de positionner un point 3D dans l'espace, il est nécessaire de définir un système de coordonnées. Le plus couramment employé est le système de coordonnées cartésiennes, où chaque point de l'espace est repéré par rapport à un repère défini par une origine  $O$  et trois vecteurs non coplanaires  $x ; y ; z$  (la plupart du temps, ces vecteurs seront orthonormés).

Dans les applications 3D, quatre repères sont principalement utilisés, illustrés sur la **Figure II.1**

**a -Le repère monde :** Il permet de définir l'origine de l'environnement 3D ainsi que son orientation globale. Généralement, l'axe  $Y$  est dirigé vers le haut et le plan formé par les axes  $X$  et  $Z$  est alors utilisé pour représenter le sol de l'environnement 3D.

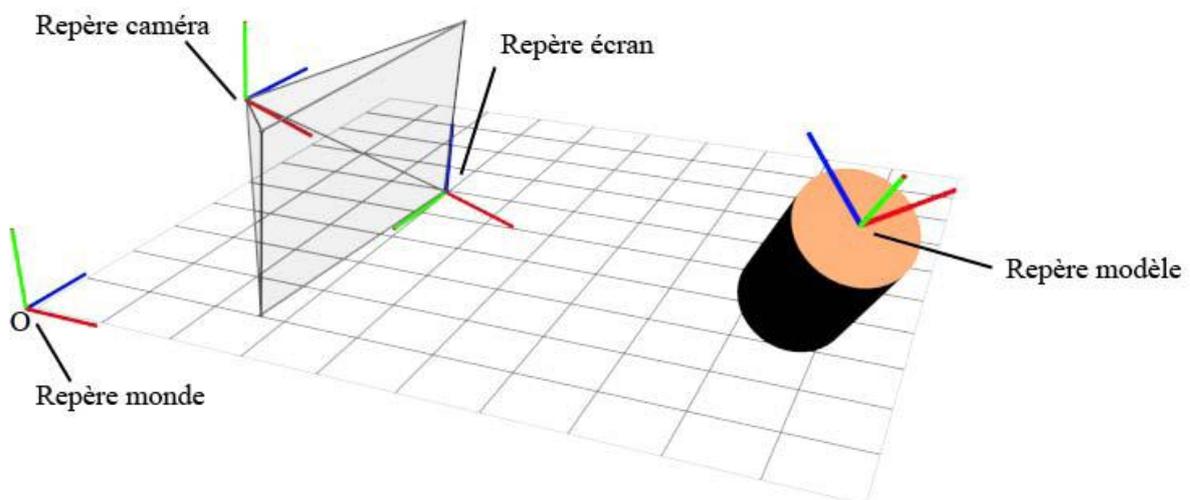


Figure II.1 : Différents repères couramment utilisés.

**b- Le repère caméra :** Ici, le point de vue courant est considéré comme l'origine du repère, et possède donc les coordonnées (0; 0; 0). Généralement, l'axe Z est utilisé pour désigner la direction de la vue ; on parle alors d'axe des profondeurs.

**c- Le repère modèle :** Chaque modèle de la scène 3D possède son propre repère : la position du point de pivot de l'objet définit l'origine du repère, et l'orientation du modèle sert à définir la base.

**d- Le repère écran :** Il permet de connaître la position d'un pixel à l'écran, ainsi que sa profondeur. Cette donnée correspond à la valeur du pixel dans le tampon de profondeur, aussi appelé ZBuffer. La phase consistant à passer d'un repère 3D au repère écran s'appelle la rasterisation ou projection. Il est également possible, à partir des coordonnées écran d'un pixel, de retrouver la position 3D du point correspondant.

## II.2.2 Transformations

Le placement d'un modèle 3D dans l'espace est effectué en spécifiant les coordonnées de son point de pivot. Ce point correspond généralement au centre de gravité de l'objet, mais peut correspondre à n'importe quel point de l'espace 3D, et n'appartient pas forcément au modèle. Ce point sera ensuite utilisé pour calculer les transformations à appliquer au modèle 3D.

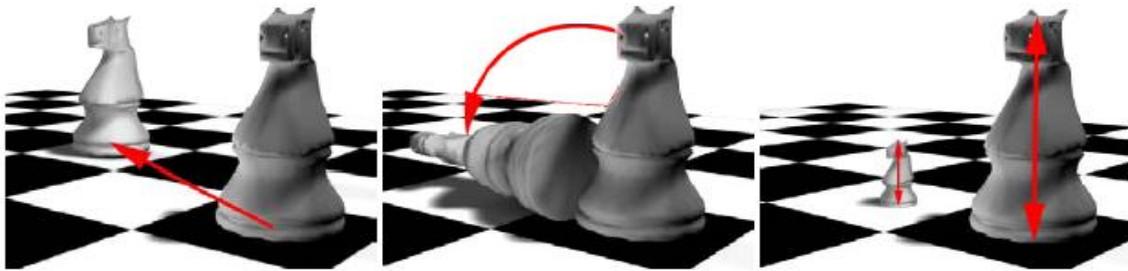
Les transformations géométriques permettent d'associer à toute figure initiale, une figure image (figure finale).

### II.2.2.1 Translation

Une fois la position d'un modèle 3D déterminée, la transformation la plus élémentaire à appliquer est la **translation**. Cette opération permet de déplacer un point 3D. La distance et la direction de la translation sont représentées par un vecteur.

### II.2.2.2 Rotation

Une seconde transformation 3D est essentielle : la **rotation**. Elle permet d'orienter un modèle dans l'espace. Une rotation peut être effectuée de deux manières : autour d'un axe de rotation ou autour d'un point de pivot (on parle alors de rotation centrale). L'axe de la rotation est déterminé par un point appartenant à cet axe ainsi qu'un vecteur donnant la direction de cet axe. La rotation est alors déterminée par l'angle de rotation à appliquer autour de cet axe.



**Figure II.2 : Translation, rotation, homothétie : les trois transformations élémentaires.**

Lors d'une rotation centrale, il y a plusieurs manières de noter la transformation à appliquer. Il est courant d'utiliser les angles d'Euler, qui sont les angles utilisés dans le repère sphérique décrit auparavant. Il est également possible de représenter une rotation en utilisant la notion de quaternion [11]. Les quaternions fournissent une notation mathématique utile pour la représentation de l'orientation et des rotations d'un modèle 3D. L'un des avantages de cette représentation est qu'il est facile de composer les rotations, et de retrouver les angles d'Euler ou la matrice de transformation associées.

### II.2.2.3 Homothétie

Enfin, la taille d'un modèle 3D est également une caractéristique essentielle. Il est ici aussi possible d'utiliser un triplet pour représenter la taille, chaque coordonnée du triplet déterminant la taille du modèle selon l'axe correspondant. La transformation associée au changement de taille d'un modèle se nomme l'**homothétie**, mise à l'échelle ou redimensionnement. Celle-ci peut être appliquée selon les trois axes : on parle alors d'homothétie centrale.

Si la transformation est appliquée sur deux axes seulement, l'homothétie est dite planaire. Enfin, une homothétie n'affectant qu'un axe est dite axiale. []

### II.2.3 Les matrices

Une matrice est un tableau de nombres ordonnés en lignes et en colonnes entourés par des parenthèses.1

$$A = (a_{i,j}) = \begin{pmatrix} a_{1,1} & 0 & \cdots & \cdots & 0 \\ a_{2,1} & a_{2,2} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n} \end{pmatrix}$$

C'est une sorte de super-vecteur qui permet de faire pas mal d'opérations intéressantes en mathématique. Un vecteur est en général utilisé en 3D pour gérer les points, les directions, les normales, etc. Les matrices permettent de faire bien plus que ça. Elles servent principalement

à convertir des données géométriques en données numériques car c'est plus facile de travailler avec des nombres.

## II.2.4 Rappel sur quelques transformations géométriques en 2D et en 3D

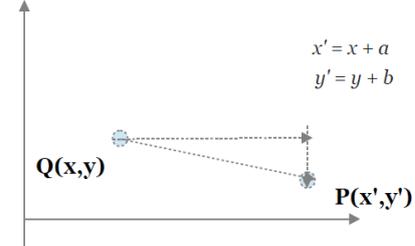
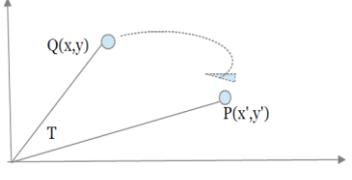
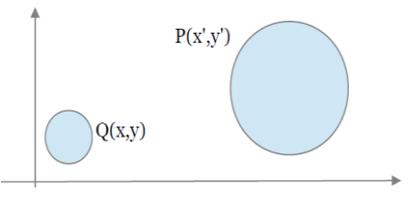
| Les transformations | Opération en 2D   | Opération en 3D  |
|---------------------|---|--|
| <b>Translation</b>  | <p>Consiste à faire "glisser" un ensemble de points sur un "rail" (un vecteur).</p>  $\begin{matrix} x' \\ y' \\ z' \\ 1 \end{matrix} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ |  |
| <b>Rotation</b>     | <p>Consiste à faire pivoter un ensemble de points d'un angle Thêta par rapport à un point.</p>  $P = \begin{pmatrix} \cos(T) & -\sin(T) \\ \sin(T) & \cos(T) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$  | <ul style="list-style-type: none"> <li>• autour de l'axe X</li> </ul> $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(T) & -\sin(T) & 0 \\ 0 & \sin(T) & \cos(T) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ <ul style="list-style-type: none"> <li>• autour de l'axe Y</li> </ul> $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(T) & 0 & \sin(T) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(T) & 0 & \cos(T) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ <ul style="list-style-type: none"> <li>• autour de l'axe Z</li> </ul> $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(T) & -\sin(T) & 0 & 0 \\ \sin(T) & \cos(T) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ |
| <b>Homothétie</b>   | <p>Consiste simplement à agrandir ou à réduire une forme géométrique.</p>  $P = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$   | $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$  |

Tableau II.1 :L'utilisation des matrices dans les transformations géométriques 2D et 3D[12]

## II.3 Qu'est-ce que la 3D

La 3 Dimension, ou tridimensionnelle, caractérise l'espace en termes de largeur, hauteur et profondeur. En fait, tout ce qui nous entoure est 3D : une table, un animal, nous-mêmes, etc Mais nous la désignons en réalité comme étant la représentation des images de synthèse dans le numérique, ainsi que les images en relief. [13]

### II.3.1 Généralités sur la 3D

#### II.3.1.1 Pourquoi l'utiliser ?

La plupart des opérations d'entrée/sortie à partir des périphériques standards se font en deux dimensions : les entrées par des dispositifs 2D comme la souris ou le *joystick* qui comptent deux degrés de liberté (deux translations) et les sorties sur imprimante ou moniteur qui ne peuvent que coucher des images à plat. Des périphériques spécialisés existent cependant pour interagir réellement en 3D ou pour obtenir une sortie 3D. Ces périphériques 3D se sont en effet multipliés et l'on compte désormais parmi eux les gants de réalité virtuelle et les souris 3D offrant les six degrés de liberté (trois translations et trois rotations). La plupart des dispositifs de visualisation 3D sont cependant encore réservés aux domaines de la recherche, des technologies de pointe ou des simulateurs militaires. C'est le cas notamment des systèmes de représentation holographique comme le *MIT holovideo display* [14], des caves équipées d'écrans offrant une vue panoramique, des *workbenches* [15] où les images sont projetées sur une table et des imprimantes 3D créant des volumes.

Toutes les configurations actuelles bien que ne possédant pas ces périphériques embarquent cependant des possibilités 3D. 3D étant l'abréviation de trois dimensions,

#### II.3.1.2 Comment expliquer que sur un écran en deux dimensions nous pouvons obtenir une sortie 3D ?

C'est à ce point que la définition de la 3D pour l'informatique est quelque peu différente de celle du monde physique puisqu'elle intègre également, en plus de la manipulation d'objets 3D, la manipulation d'images 2D (projections à l'écran d'objets décrits dans l'espace) donnant l'impression de volume. Mais précisons dès à présent que toutes les méthodes de création et de visualisation d'un contenu 3D dont nous parlerons ne permettent donc que de donner à l'utilisateur l'illusion qu'il évolue dans un « monde 3D ». La plus grande immersion à un prix abordable est obtenue avec **les casques de réalité virtuelle** ou **les lunettes** grâce au rendu stéréoscopique. Dans les deux cas, deux images sont calculées, une par œil. La seule différence entre ces deux images provient d'une légère translation de la prise de vue sur la scène pour simuler l'écartement des yeux et donc la vision depuis chacun des yeux. Pour le

casque, les deux images sont affichées simultanément sur deux écrans à cristaux liquides différents (un par œil). Avec les lunettes, le rendu est effectué sur le moniteur. L'effet stéréoscopique est obtenu par l'affichage alterné des images calculées. Pour percevoir le relief, les lunettes servent d'obturateur.

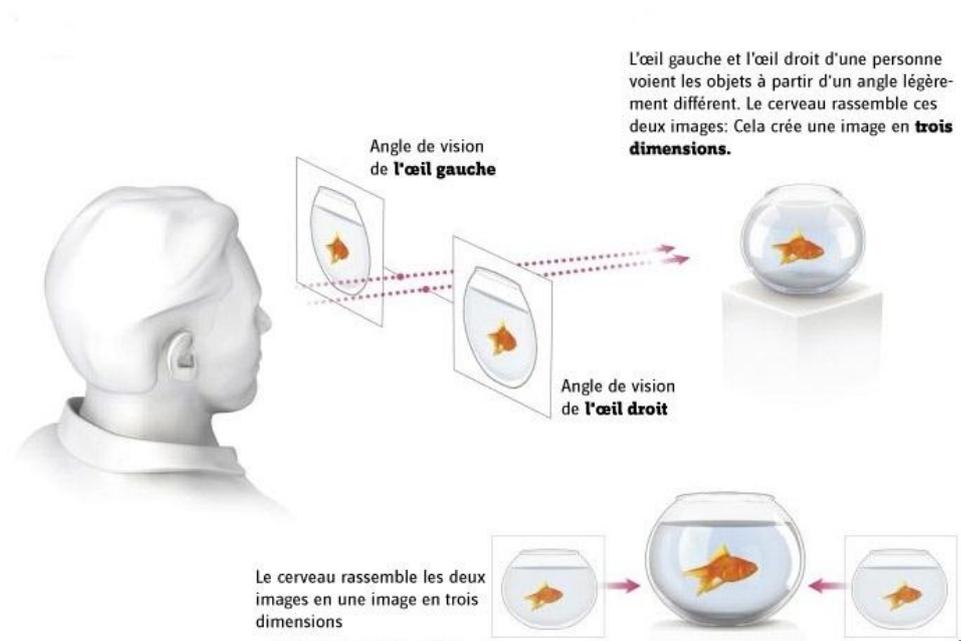
Elles sont, en effet, synchronisées avec l'affichage et lorsque l'image pour un œil est affichée, l'autre œil est obturé pour ne pas la voir. Une fréquence double – 50 images par seconde – est nécessaire pour satisfaire la tolérance individuelle de chacun des yeux.

La principale différence entre le monde physique que nous percevons en 3D ou la représentation holographique et la fausse 3D plus communément utilisée est l'immobilité des objets lorsque nous bougeons la tête en face de l'écran. Ces systèmes (mis à part lorsque l'on utilise un casque avec capteurs de position) ne permettent pas, par une translation du regard, de découvrir plus d'information sur le volume d'un objet. Cependant, l'illusion est suffisante pour donner à l'utilisateur le sentiment d'être plongé dans un univers 3D ; d'autant plus s'il utilise un environnement immersif visuel et sonore.

Tous les modes de représentation et les techniques de conception basés sur la 3D servent dans la plupart des applications à fournir une compréhension accrue d'un modèle comportant en réalité les trois dimensions. Leur but principal est de rendre l'environnement de travail (autrement dit l'interface entre l'homme et la machine) plus réaliste et visuel. Pour la simulation et la conception assistée par ordinateur (C.A.O.), les deux premières utilisations industrielles de la 3D, le but est de permettre la manipulation et la simulation de comportements selon certains paramètres. Dans le cas de la C.A.O., les pièces mécaniques sont créées, assemblées et testées avant leur mise en fabrication. Les modèles 3D des pièces, associées aux paramètres physiques de simulation (vitesses de rotation, frottements par exemple) permettent d'obtenir les usures au bout d'une durée donnée ou leur temps théorique de bon fonctionnement. [1]

### **II.3.2 Comment fonctionne la 3D**

Le positionnement parallèle de nos deux yeux nous permet de voir notre environnement en relief. Ils fournissent à notre cerveau deux images légèrement décalées étant donné que la perspective de ces deux images 2D est légèrement différente. Cette petite différence suffit à notre cerveau pour établir une image spatiale en 3D à partir de ces deux images 2D. [16]



**Figure II.3: Fonctionnement de la 3D.**

### **II.3.3 Les domaines d'utilisation de la 3D**

En ce qui concerne l'utilisation de la 3D dans les films, le but recherché est de créer un maximum d'émotions chez le spectateur en le plongeant directement dans une action irréaliste ou difficile à mettre en œuvre par des effets spéciaux physiques ou mécaniques. Pourtant, la 3D n'a pas qu'un rôle ludique puisqu'elle est utilisée dans les laboratoires de recherche, les centres de conception, pour toutes les études de design, en architecture et également de plus en plus dans l'art numérique, comme on peut le constater en regardant toute chronologie des événements ayant marqués l'IHM.

#### **II.3.3.1 Utilisation de la 3D dans la chimie**

Un des premiers domaines dans lequel la 3D a été employée est la simulation et la visualisation de phénomènes réels. Dans les laboratoires de chimie, la 3D est utilisée pour obtenir des représentations de molécules complexes manipulables à souhait. La 3D est un outil très puissant permettant la représentation de l'infiniment petit. Même si une molécule peut être perçue au microscope, l'étude de sa structure est rendue bien plus simple par sa représentation 3D.

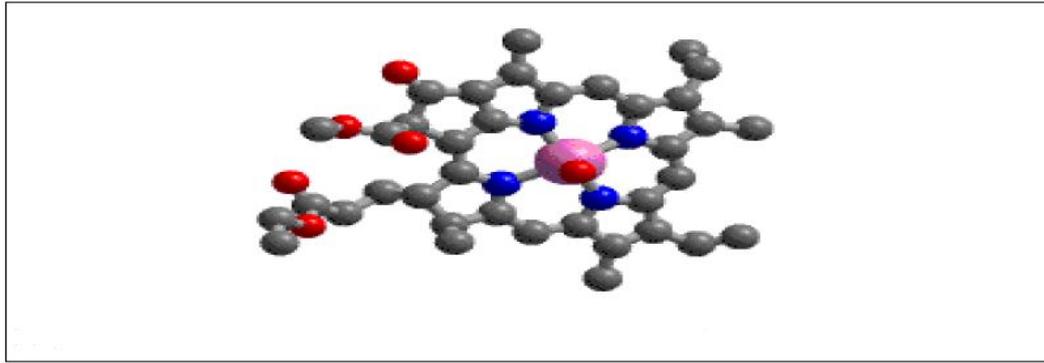


Figure II.4 : Exemple d'utilisation de la 3D – La molécule de Chlorophylle A.

### II.3.1.2 L'utilisation de la 3D dans le domaine de l'astronomie

En astronomie, règne de l'infiniment grand, la 3D est également largement utilisée. Il est possible de suivre le mouvement des astres ou tout simplement d'obtenir une carte animée du ciel vue d'un point quelconque du temps et de l'espace. On peut ainsi faire des promenades virtuelles dans le système solaire ou dans l'univers. Par simulation, il est possible de visualiser les dégâts occasionnés par une météorite. Une des premières animations 3D était pour visualiser l'orbite d'un satellite artificiel autour de la terre par E. Zajac au BTL (*Bell Telephone Laboratory*). À encore plus grande échelle, il est également possible de visualiser les collisions de galaxies.

La 3D est aussi utile en cartographie pour représenter les informations relevées par les satellites ou par les topographes sur le terrain. À partir des fichiers de données de ces relevés topographiques accessibles à tous (par les fichiers DEM (*Digital Elevation Model*) ou bientôt sur le site *Web* de l'Institut Géographique National). [1]

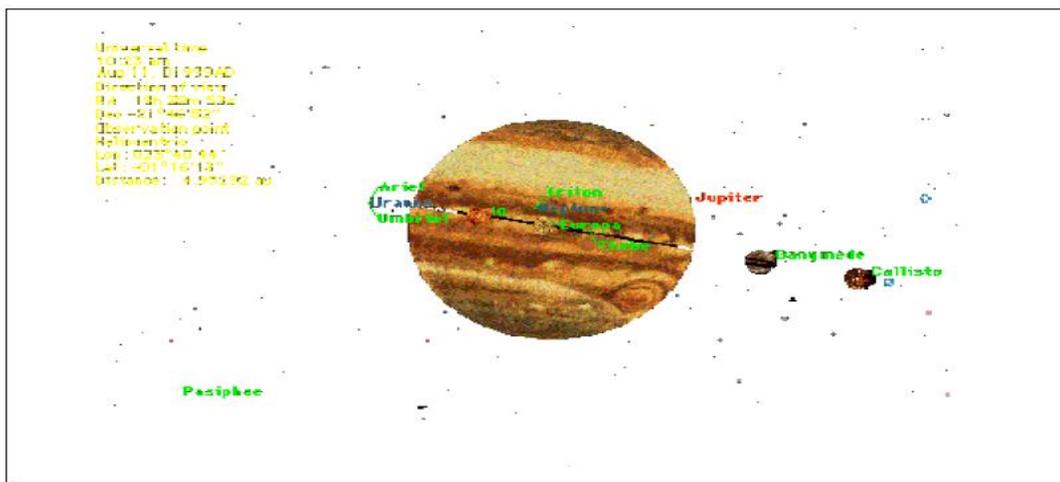


Figure II.5: Exemple d'utilisation de la 3D – Jupiter et ses satellites.

## II.4. Visualisation des scènes

### II.4.1 Deux méthodes de rendu 3D : différences et utilités

Il existe différentes manières de créer des images de synthèse à partir de la spécification d'une scène en 3 dimensions. Chronologiquement, est d'abord apparu un modèle à base de polygones et plus précisément de triangles. Un second modèle, présenté en 1976 par James Blinn, est le lancer de rayons. Enfin, les méthodes utilisant la radiosité et celles dites mixtes sont apparues plus récemment.

Nous verrons en détail comment sont obtenues les images par *raytracing* (lancer de rayons) dont les méthodes cherchent à reproduire le trajet des rayons de lumière depuis l'œil de l'observateur jusqu'aux objets de la scène, éclairés par des sources lumineuses. Nous parlerons plus particulièrement de *raytracing* arrière (*back ward raytracing*) qui consiste à suivre les rayons depuis la caméra jusqu'aux objets. Il existe également des méthodes de *Forward Raytracing* où le trajet est fait à l'envers depuis les différents points de l'écran vers la caméra (comme dans la réalité). La méthode dite de radiosité, quant à elle, calcule la luminosité et non la couleur de chacun des points. Ces calculs de luminosité sont basés, contrairement à ceux du lancer de rayons, sur des modèles physiques rigoureux de distribution d'énergie dans un environnement.

Pour les deux méthodes de rendu 3D que nous décrivons ici (*raytracing* et rendu par facettes), les premières étapes de la chaîne de production menant de la description de la scène à l'image finale affichée à l'écran sont les mêmes. Cette chaîne débute par la partie conception de l'image. Pour un rendu sur écran, seulement cette étape est véritablement 3D puisque le créateur de la scène décrit le positionnement des objets dans l'espace. Chacun des objets composants la scène est alors habillé par une couleur (dans le cas le plus simple) ou d'une texture qui représente à l'utilisateur le matériel de l'objet, voire également sa rugosité et son relief. Puis, sont définies les différentes lumières éclairant la scène et les caméras desquelles sont prises les vues.

La deuxième étape vise à transformer la description de la scène en 3 dimensions en une image en 2 dimensions en fonction des différents éléments de la scène et du point de vue duquel elle est regardée. C'est pour cette étape que le rendu diffère entre les deux méthodes.

Pour le modèle à base de triangles, il convient d'abord de transformer les différents éléments en ensembles de facettes. Tous les objets de la scène, y compris ceux qui sont courbes seront donc constitués de triangles qui sont le point d'entrée du pipeline graphique dont nous parlerons plus loin.

Afin que l'aspect courbe d'un objet soit respecté, deux solutions sont envisageables : décomposer l'objet en un assez grand nombre de facettes afin d'obtenir un ensemble convainquant ou utiliser une méthode de lissage par apparence telle que celle de Gouraud ou de Phong (cette dernière méthode n'est cependant pas encore implémentée dans la plupart des API 3D).

Le pipeline graphique de la méthode par facettage effectue de nombreuses opérations (pour les transformations, la détermination des faces visibles, etc.) et les modèles de lumière, de coloriage ou de texturage sont souvent plus rudimentaires que pour le lancer de rayons. En effet, à l'heure actuelle ce mode de rendu par triangles est utilisé pour permettre une interaction temps réel. Ainsi, la nécessité de pouvoir calculer et afficher une image rapidement après une modification d'un paramètre de la scène interdit l'utilisation de méthodes trop coûteuses en temps. Cependant, grâce à des fréquences rapides d'affichage des images successives, l'œil humain tolère les défauts générés par cette méthode.

### **Lancer les rayons**

De l'autre côté, les méthodes de lancer de rayons ne permettent pas de réaliser des images en un temps suffisamment court pour permettre une interactivité avec l'utilisateur. Cependant, la qualité des images obtenues, fixes ou animées (par calculs d'images successives), est parfois stupéfiante.

L'avenir de la 3D en temps réel réside probablement dans l'utilisation de cette méthode avec des ordinateurs bien plus puissants, embarquant un matériel spécialisé, capables de calculer une image en un temps inférieur au 25ème de seconde. Cette fréquence d'échantillonnage étant connue comme celle à partir de laquelle l'œil humain ne perçoit plus les changements d'images.

L'algorithme de rendu par la méthode du lancer de rayons est beaucoup plus simple que celui de la méthode par facettes. On appelle cette méthode ainsi car son principe est de former un rayon partant de l'œil de l'observateur (considéré comme une caméra ponctuelle) et passant par un point précis de l'écran que l'on place virtuellement entre l'observateur et la scène. La couleur du pixel de l'écran par lequel passe le rayon sera alors de la couleur du matériel du premier objet frappé par le rayon. La détermination des objets traversés par le rayon se fait par des calculs d'intersection peu complexes mais assez coûteux, ce qui explique la lenteur de ce procédé. Pour obtenir l'image complète de la scène vue depuis un point précis, il suffit de « balayer » tous les pixels de l'écran et de faire le calcul précédent pour chacun des rayons formés. **La Figure II.6** schématise le parcours d'un rayon depuis la caméra jusqu'à la scène. Les deux rayons réfléchi et absorbé servent à calculer respectivement les réflexions et les

objets vus en transparence. Le rayon vers la source lumineuse sert, quant à lui, à calculer les ombres influant sur la couleur de l'objet. [1]

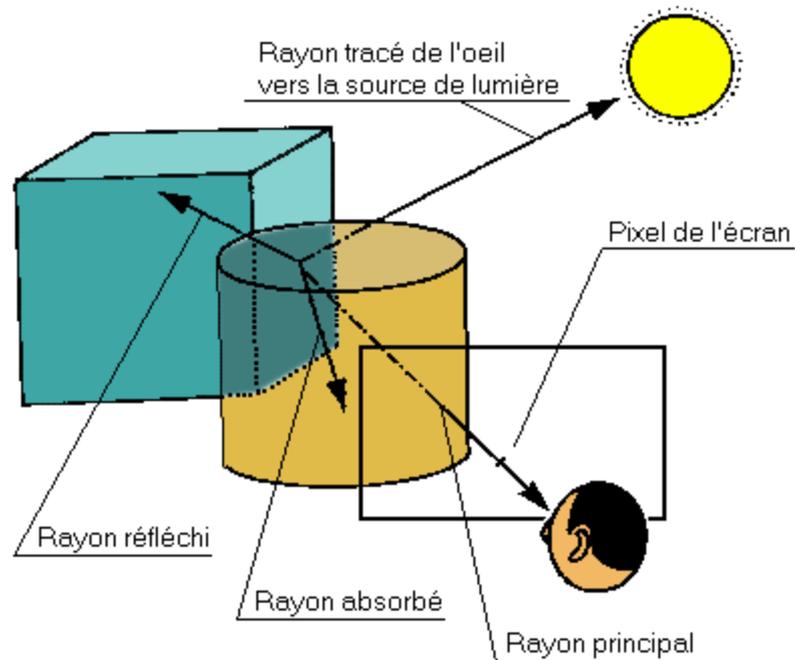


Figure II.6 : Principe du lancer de rayons. « Qualité plutôt que quantité »

## II.4.2 Les bases de la 3D temps réel

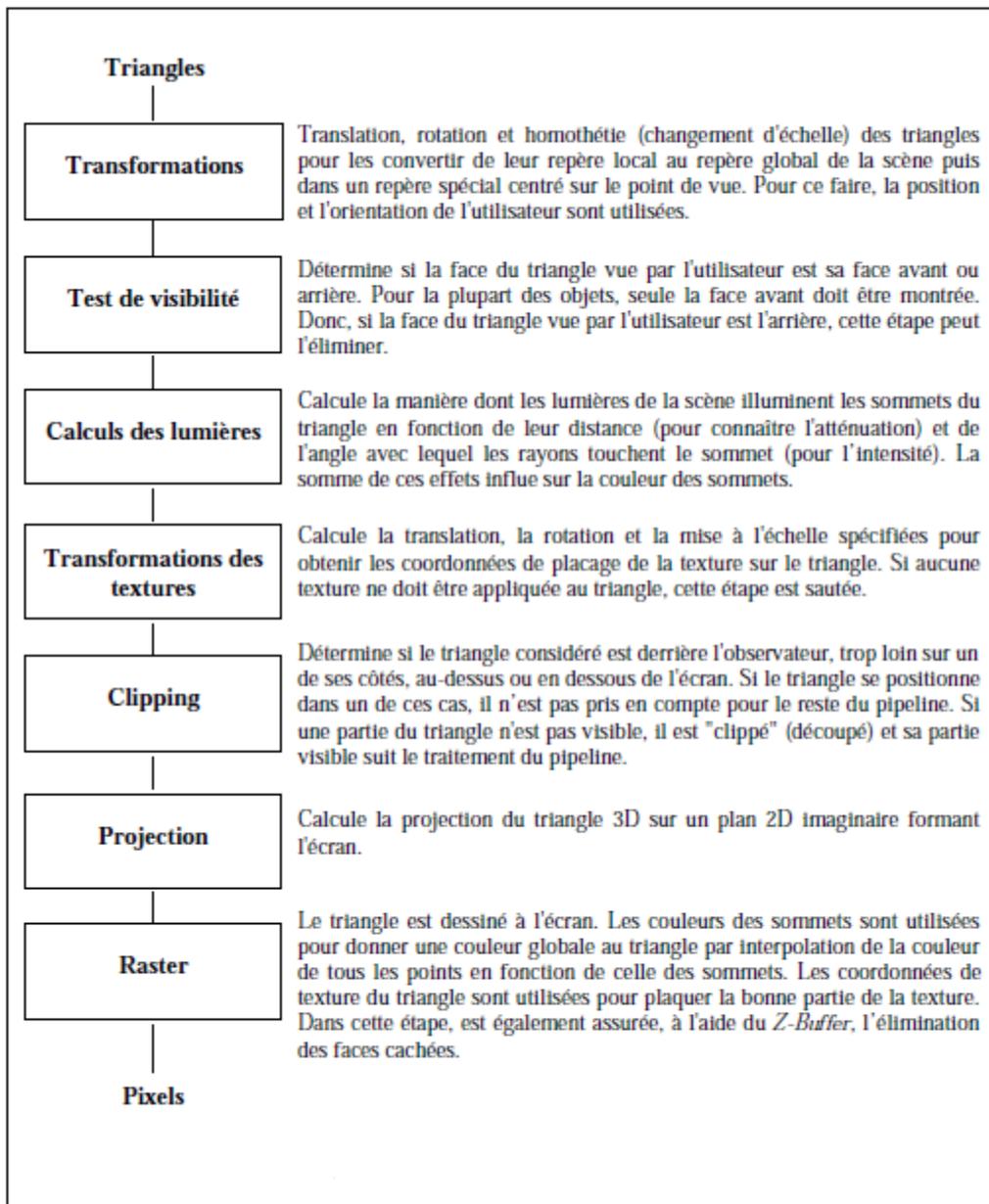
Avant de présenter les différentes manières de créer et d'afficher une scène 3D, nous souhaitons rappeler les fondements de la 3D temps-réel. Ils sont rassemblés dans la chaîne de production d'une image ou pipeline graphique (ou encore pipeline 3D). Il convient donc de le présenter, ainsi que les différentes étapes de celui-ci.

### II.4.2 Le pipeline 3D

C'est l'ensemble des opérations nécessaires pour afficher une scène constituée d'objets 3D regardés depuis une position et avec une orientation données. À chaque fois que l'état de la scène change (c'est-à-dire à chaque mouvement de la caméra, à chaque déplacement d'un objet, etc.), elle doit être redessinée. Pour cela, la description en mémoire vive des objets de la scène doit être traduite en points 2D à l'écran. Ce processus est le travail du pipeline 3D (plus communément appelé moteur 3D). Quel que soit le moteur 3D utilisé, *plugin* ou API, un pipeline 3D effectue les différents calculs nécessaires à l'affichage d'une scène 3D sur un écran.

Les **éléments d'entrée** de ce pipeline sont **des triangles** plus pratiques que les quadrilatères ou autres polygones pour les calculs car ils ne peuvent être ni vrillés (dont les sommets ne

sont pas coplanaires) ni concaves – donc plus faciles à tracer. Cependant, afin de nous autoriser à créer des polygones complexes, la plupart des moteurs 3D effectuent une triangulation des différentes faces avant de les envoyer au pipeline graphique. Ainsi, ils nous permettent de réfléchir en termes de polygones et non uniquement en termes de triangles mais évitent néanmoins le problème de remplissage des faces concaves. La Figure suivante illustre les différentes étapes de calculs menant à l’affichage d’un triangle.



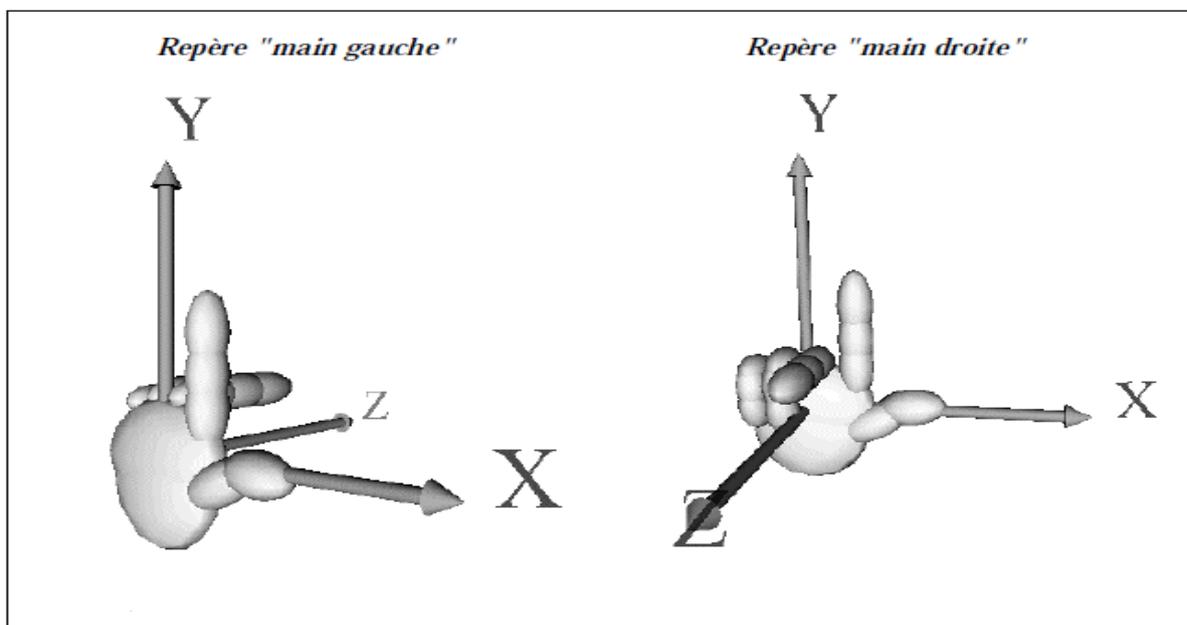
**Figure II.7: Les différentes étapes du pipeline 3D.**

Avec une scène de petite taille, ou optimisée, le pipeline calcule le rendu efficacement et effectue un rafraîchissement rapide. On obtient par conséquent une animation fluide. Au contraire, avec une scène comportant de nombreux objets ou pour une scène non optimisée, le

pipeline risque de bloquer dans une, voire plusieurs étapes. Il existe cependant des pipelines graphiques garantissant 25 images par seconde quelle que soit la scène ; ceci en arrêtant leurs traitements au-delà du 25ème de seconde si cela est nécessaire. D'autres techniques, moins radicales, existent également.

Nous expliquons ici succinctement les calculs effectués à chacune des étapes du pipeline graphique

Avant tout, il convient de parler du mode de représentation utilisé pour placer un point dans l'espace. Les calculs nécessaires pour la transformation et la projection d'un point dépendent de cette représentation. Le système de coordonnées que l'on utilise pour la 3D est cartésien. Selon les outils utilisés, ce référentiel sera « main droite » (comme avec OpenGL ou VRML) ou « main gauche » (comme avec Direct3D). Comme nous pouvons le voir dans **la Figure II.8**, que le système soit direct ou indirect, pour chacun d'eux les axes X et Y sont les mêmes (X de gauche à droite et Y de bas en haut de l'écran). La façon la plus naturelle de se représenter un système de coordonnées cartésien est d'utiliser ses deux mains (d'où les noms « main droite » et « main gauche ») comme cela est schématisé dans la figure. Pour chacune des mains, il suffit de faire pointer le pouce (donnant l'axe X) vers la droite, l'index (l'axe Y) vers le haut. Le majeur, perpendiculaire aux deux doigts précédents (et à la paume de la main) nous donne automatiquement la direction positive de l'axe Z.



**Figure II.8: Les deux repères (main droite et gauche) utilisés en 3D.**

La transformation des coordonnées d'un système de coordonnées à l'autre se fera en prenant l'opposée de la coordonnée Z. Quant à la différence d'interprétation que l'on peut donner à cette opposition de l'axe Z, pour le repère « main gauche » on parlera de profondeur pour la

coordonnée Z d'un point alors que, pour le repère « main droite », on y verra plus volontiers une altitude.

Toute primitive graphique complexe est d'abord transformée en un ensemble de triangles. L'apparence obtenue par approximation en un nombre plus ou moins de triangles permet d'avoir soit un rendu plus rapide soit un rendu de meilleure qualité. Chacun des triangles subit alors les différentes étapes du pipeline graphique dont nous donnons pour les plus importantes le détail des calculs.

## **II.5. La Réalité Virtuelle**

La réalité virtuelle RV est une nouvelle technologie qui se situe à la croisée de plusieurs disciplines telles que l'infographie, la conception assistée par ordinateur (C.A.O), la simulation, la téléopération, l'audiovisuel ou le travail collaboratif. Elle utilise de nombreux périphériques matériels et des techniques logicielles pour chaque domaine applicatif. [17]

D'autres définitions de la RV d'un point de vue fonctionnel, technique et philosophique sont détaillées dans [18] et [19].

Quelle que soit la définition de cette réalité dite virtuelle, sa finalité reste la même.

C'est pour cette raison que Fuchs et associés la définissent à travers sa finalité : *«La finalité de la réalité virtuelle est de permettre à une personne (ou plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel.»* [18]

### **II.5.1 L'interaction**

Depuis l'apparition de la RV, les chercheurs se sont particulièrement intéressés à l'interaction qui peut être considérée comme la composante motrice de tout système interactif. L'interaction peut être définie comme étant un langage de communication entre l'homme et la machine. Ce langage correspond à l'ensemble des actions/réactions réciproques entre l'homme et l'ordinateur par l'intermédiaire d'interfaces sensorielles, d'interfaces motrices et de techniques d'interaction [20].

## **II.5.2 Les techniques d'interaction 3D**

L'interaction 3D donne aux utilisateurs le pouvoir de réaliser différentes tâches au sein d'un monde virtuel. Une tâche peut être définie comme la fonction que l'utilisateur exécute à un moment donné. Le résultat de l'exécution de cette fonction se traduira par les changements de l'état du système dans lequel l'utilisateur évolue.

Nous pouvons définir l'interaction en RV comme la traduction des actions des utilisateurs dans le monde réel en des tâches spécifiques dans l'espace virtuel. En effet, les chercheurs tentent de plus en plus de reproduire dans un environnement virtuel des gestes identiques à ceux de la vie quotidienne. Par exemple : se déplacer pour découvrir un lieu inconnu ou pour aller dans un endroit bien précis, toucher, prendre ou manipuler des objets. Pour chacun de ces gestes, plusieurs techniques spécifiques ont été développées ces dernières années. [10]

### **II.5.2.1 Qu'est-ce que la navigation**

Comme dans le monde réel, l'utilisateur a besoin de se déplacer dans la scène virtuelle 3D pour réaliser certaines tâches. L'action « se déplacer » va de bouger sa tête pour observer un objet de plus près ou d'un autre angle, à se promener dans un parc ou visiter un musée, voire se déplacer d'une planète à une autre. La navigation désigne l'ensemble des méthodes qui permettent de connaître la position d'un objet par rapport à un système de référence, ou par rapport à un point fixe déterminé. Certains auteurs considèrent la navigation comme la capacité de bouger à l'intérieur d'une scène générée par l'ordinateur [21]. D'autres encore la définissent comme l'ensemble des déplacements de l'utilisateur à l'intérieur de l'espace virtuel [22]. Autrement dit, elle permet aux utilisateurs d'explorer, de rechercher et/ou de manœuvrer dans l'espace virtuel.

#### **II.5.2.1.1 Les composants de navigation**

Généralement, deux composantes principales sont explorées dans la navigation :

le déplacement et la recherche d'itinéraire [23].

1- Le déplacement représente la composante motrice de la navigation. Il se rapporte aux déplacements physiques de l'utilisateur d'un endroit à un autre.

2-La recherche d'itinéraire correspond à la composante cognitive de la navigation. Elle permet aux utilisateurs de se repérer dans l'environnement et de choisir une trajectoire pour se déplacer [18]. Dans ce cas, l'utilisateur se pose des questions telles que : "où suis-je ?", "où doit -je aller ?", "comment arriver à cet endroit ?".

Lors de la conception de toute technique de navigation, il faut prendre en considération plusieurs facteurs. Car, ces facteurs influent sur la qualité de la technique de navigation :

- La vitesse de déplacement (le temps d’achèvement d’une tâche de navigation) ;
- La précision d’exécution;
- La conscience spatiale (connaissance de l’utilisateur de sa position et de son orientation dans l’environnement virtuel pendant et après la navigation) ;
- La facilité d’apprentissage (la capacité d’un utilisateur débutant à s’approprier la technique de navigation) ;
- La facilité d’utilisation ;
- La capacité de l’utilisateur à collecter des informations sur l’environnement pendant le voyage ;
- La présence virtuelle (le sentiment de l’utilisateur d’être immergé dans l’environnement virtuel) ;
- Le confort de l’utilisateur.

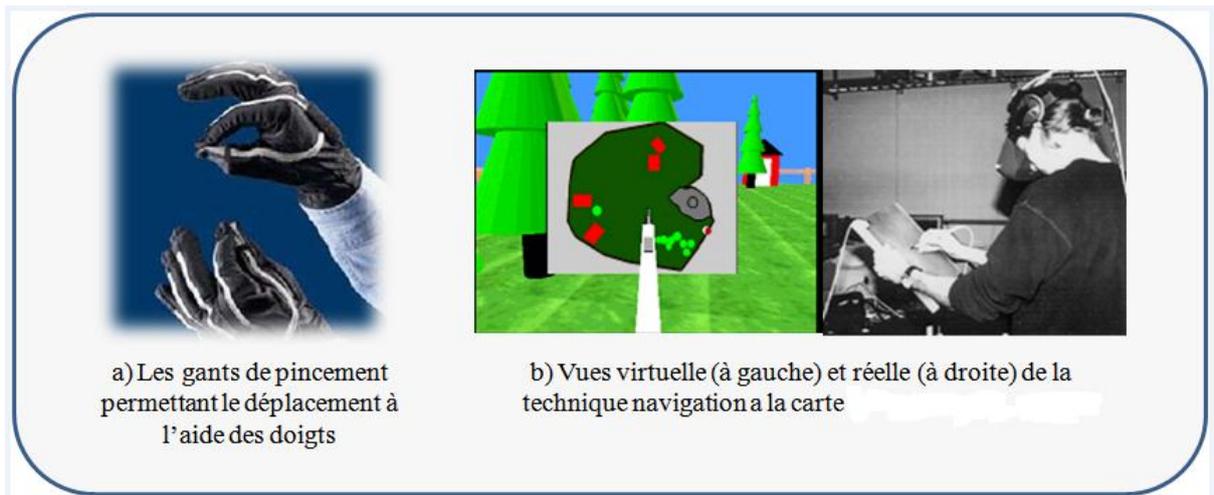
#### **II.5.2.1.2 Les types de navigations**

En fonction de la connaissance de destination (connue ou bien inconnue), la navigation peut être libre ou assistée. Dans le cas d’une navigation libre, l’utilisateur ne connaît pas précisément sa destination. Il se déplace dans l’espace dans le but de découvrir et d’explorer l’univers virtuel. Dans le cas d’une navigation assistée, l’utilisateur avance dans l’espace virtuel pour aller à un endroit bien précis qui représente la cible du déplacement. Pour cela, l’utilisateur est aidé pendant le processus de navigation afin de se repérer et trouver la bonne trajectoire.

Dans ce qui suit, nous allons voir un ensemble de métaphores de navigation. Nous classons les différentes techniques de navigation en deux grandes familles : les techniques de navigation libre et les techniques de navigation assistée.

##### **a) -La navigation libre**

Dans la plupart des cas d’étude sur l’interaction 3D, le concept utilisé pour le développement de nouvelles techniques est inspiré de l’interaction de l’être humain avec son environnement réel. Par exemple, marcher est le moyen le plus naturel et le plus utilisé pour se déplacer d’un endroit à un autre dans la vie quotidienne. Ce mode de navigation est basé sur la marche réelle, cette dernière à la particularité d’être simple et naturelle. L’utilisateur se déplace librement à l’intérieur d’un cyberspace en marchant physiquement selon les contraintes de place et en tournant sa tête pour changer sa direction. Plusieurs techniques d’interaction basées sur la métaphore de la marche réelle existent aujourd’hui et chacune d’elles utilise un périphérique matériel spécifique pour permettre à l’utilisateur de marcher physiquement sur place.



**Figure II. 9 : Exemple de techniques de navigation libre.**

Lorsque l'utilisateur navigue dans le but d'explorer et de découvrir l'espace virtuel, son trajet se dessine au fur et à mesure de son avancée dans l'environnement. Pour cela, l'utilisateur doit disposer des moyens qui lui permettent de démarrer et d'arrêter le processus de navigation, de modifier la vitesse et la direction du déplacement.

**b)- La navigation assistée**

Lorsque l'utilisateur connaît sa cible, c'est-à-dire la destination de son déplacement, il est possible de l'assister afin de le décharger de certaines tâches pendant le processus de navigation. Pour cela, l'utilisateur peut se servir d'une liste ou d'un menu, dans lequel il peut sélectionner son choix. Certaines applications utilisent un monde en miniature pour sélectionner la destination sur la maquette du monde virtuel. La maquette joue le rôle de carte 3D détaillée, dans laquelle l'avatar de l'utilisateur se déplace et s'oriente selon la destination déterminée. Dans ce cas, le chemin à suivre est déterminé automatiquement par le système (voir la figure 10-a).



**Figure II.10 : Exemples de techniques de navigation assistée.**

D'autres applications utilisent une autre méthode qui permet à l'utilisateur de dessiner le chemin souhaité à l'aide d'un stylo sur une plaque en plexiglas (voir la figure 10-b).

Cependant, il existe d'autres applications qui utilisent un rayon virtuel déformable pour dessiner librement le chemin à suivre pendant la navigation. Le but de cette interaction à deux mains est d'éviter les obstacles qui se dressent devant l'utilisateur et de modifier finement la direction et la vitesse courante. Le chemin parcouru est représenté par une courbe 3D. Une fois la destination connue, il suffit juste de déplacer l'utilisateur jusqu'à l'endroit Souhaité.

### **II.5.2.2 La sélection**

L'action « sélectionner un objet » est une tâche très courante dans la vie quotidienne. En effet, pour pouvoir manipuler un objet, l'être humain a toujours besoin de le prendre dans sa main ou de le désigner parmi d'autres objets. Le processus de sélection en RV s'inspire souvent de la sélection dans le monde réel. La tâche de sélection, appelée aussi tâche d'acquisition d'une cible[24], représente la désignation d'un objet ou d'un ensemble d'objets afin d'accomplir un objectif donné au sein de l'environnement virtuel [25]. Mais comment indiquer au système qu'un objet a été sélectionné ? La validation de sélection est l'action qui suit la tâche de désignation. Elle peut être indiquée de plusieurs façons suivant la technique de sélection utilisée et l'environnement dans lequel l'utilisateur évolue. Par exemple, on peut presser un bouton, utiliser un geste ou encore utiliser une commande vocale, mais la validation pourrait être faite automatiquement si le système d'interaction prend en considération les intentions des utilisateurs.

### **II.5.2.3 La manipulation**

La navigation ou la sélection sont des tâches qui permettent à l'homme d'avoir l'illusion d'habiter un monde virtuel, de voyager au sein de celui-ci et même d'atteindre et toucher les objets appartenant à ce dernier. Dans la plupart de ces cas, l'homme reste toujours un spectateur immergé dans son environnement virtuel.

La tâche de manipulation quant à elle permet à l'utilisateur d'être un acteur capable de changer les propriétés de l'environnement virtuel. Elle représente la composante active de tout système interactif. Elle peut être définie comme un processus complexe qui consiste à modifier les propriétés d'un objet ou d'un ensemble d'objets appartenant à l'univers virtuel. Ces propriétés sont, par exemple, la position, l'orientation, la couleur, l'échelle et la texture.

Il existe deux types de manipulations en RV : la manipulation directe et la manipulation indirecte. Le type de manipulation dépend de la technique utilisée. La manipulation directe est celle pour laquelle l'utilisateur agit directement sur l'objet appartenant à l'environnement

virtuel. La manipulation indirecte, quant à elle, est celle pour laquelle l'utilisateur agit sur les composantes du monde virtuel par l'intermédiaire d'entités graphiques (boutons, menus 3D) ou matérielles.

La tâche de manipulation est intimement liée à la tâche de sélection, car on ne peut pas manipuler un objet sans l'avoir sélectionné préalablement [10].

## II.6. Navigation 3D

La navigation dans une scène 3D représente certainement une part importante de l'interaction 3D. Elle est invoquée en permanence dans tous les types d'application 3D, que ce soit dans la modélisation afin d'observer les différentes parties d'un modèle, la RV immersive afin de se déplacer dans l'environnement 3D ou dans les jeux vidéo afin de suivre le personnage virtuel. Dans beaucoup d'applications 3D, on parlera de caméra 3D afin de désigner la position et l'orientation de la vue courante. Naviguer dans la scène correspond donc à manipuler la caméra 3D, la déplacer et la pivoter. Techniquement parlant, il est courant de représenter une caméra 3D à l'aide de trois composantes : une position dans l'espace  $P$ , représentée par les trois coordonnées  $(x; y; z)$ , ainsi que deux vecteurs : un vecteur donne la direction de vue **Dir** et un autre, appelé vecteur up ou vecteur haut, noté **Up**, permet de désigner la verticalité (Figure 8).

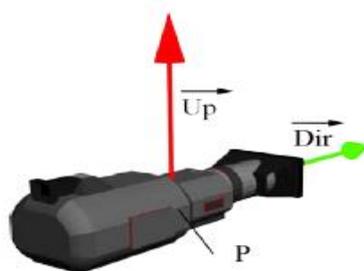


Figure II.11 : Représentation d'une caméra 3D.

Les déplacements au sein de la scène 3D peuvent être effectués de manière égocentrique, c'est à dire en considérant que le point de vue appartient à la scène, ou de manière exocentrique, en considérant la scène 3D dans son intégralité et en gardant une vision globale de celle-ci. Lors de déplacements égocentriques, les mouvements sont calculés et réalisés par rapport au repère de la caméra, tandis que les déplacements exocentriques font référence à un repère extérieur au point de vue [27].

## II.6.1 Les outils de navigation 3D

La plupart des outils de navigations partent d'un seul principe, La navigation en 3D s'effectue grâce à un objet caméra. La surface visualisable d'une vue 3D est déterminée par la localisation de la caméra (position de la caméra dans la vue 3D) et le point sur lequel elle est dirigée (point 3D vers lequel la caméra est orientée). Toutes les méthodes de navigation 3D opèrent en manipulant les localisations du point d'observation et de la cible dans l'espace  $x,y,z$ .

L'une des manières de visualiser la relation entre le point d'observation et la cible consiste à la représenter sous forme d'une ligne le long de laquelle notre regard va se porter. Au début de la ligne se trouve la caméra (point d'observation), à la fin le centre de notre vue (cible). Les deux extrémités de la ligne sont situées dans l'espace  $x,y,z$  de la vue 3D.

La ligne point d'observation-cible doit suivre des règles basées sur l'outil de navigation 3D actif.



Il existe trois combinaisons de manipulation de la relation point d'observation-cible :

- **Point d'observation mobile, cible fixe**

Le point d'observation peut bouger, mais le centre de la vue reste fixe par rapport à une localisation 3D. C'est comme si nous marchions autour d'une bouche d'incendie tout en gardant notre regard fixé sur sa partie supérieure.

- **Point d'observation et cible mobiles**

Les deux extrémités de la ligne peuvent bouger indépendamment. C'est comme si nous regardions du hublot d'un avion au moment de l'atterrissage.

- **Point d'observation fixe, cible mobile**

Le centre de la vue 3D bouge, mais le point d'observation reste fixe par rapport à une localisation 3D. Cela revient à rester debout sans bouger, en penchant uniquement la tête.

Nous pouvons consulter nos données 3D par des outils tels que ArcGlobe dans ArcScene. Différentes options sont disponibles pour la navigation dans notre vue 3D.

Les outils de navigation 3D nous permettent de parcourir la vue 3D de manière interactive. Certains de ces outils sont similaires à ceux que nous utilisons dans ArcMap. D'autres sont spécifiques à la navigation en 3D. Quelques outils sont spécifiques à ArcGlobe uniquement.

Nous pouvons également parcourir la vue 3D sans outils interactifs à l'aide des workflows suivants :

- Zoom sur l'étendue totale des données ;
- Zoom sur l'étendue d'une couche ;
- Utilisation de l'outil Rechercher et zoom sur une entité ;
- Utilisation d'une requête de sélection et zoom sur un ensemble de sélection ;
- Création et consommation d'un ensemble de géo signets 3D pour la vue 3D. [28]

## **II.7. Conclusion**

Nous avons abordé dans ce chapitre les différents concepts liés à la navigation 3D. Ainsi que, les techniques de navigations les plus utilisées dans les applications d'interactions 3D, notamment en réalité virtuelle, afin de rendre la scène virtuelle plus proche de celle du monde réel. En réalité, La théorie de navigation 3D n'est pas limitée seulement à ces modèles élémentaires mais ils existent d'autres approches visant à ajouter plus de réalisme aux interactions avec l'utilisateur pour lui rendre plus à l'aise.

Chapitre III  
**Chapitre III**

IMPLEMENTATION  
**IMPLEMENTATION**

## III.1 Introduction

L'apprentissage de la navigation est une étape incontournable au bon fonctionnement d'une application 3D. Deux fonctions sont nécessaires pour naviguer dans une scène virtuelle, la rotation et la translation.

Pendant la programmation nous avons utilisé l'IDE permettant de programmer en C++, on parle bien sûr du Visual studio C++ avec l'API OpenGL.

## III.2 Choix du langage

Nous avons préféré d'utiliser un langage dont on se sert pour concevoir des jeux vidéo et parmi ces langages il y'a le C++.

Le C++ est un très bons choix, car il est couramment utilisé dans le monde du jeu vidéo. Du fait qu'il dispose d'une multitude de bibliothèques permettant de réaliser des jeux2D et 3D (SDL, Ogre3D, irrlicht). Malheureusement, ce n'est pas le langage le plus simple (même si ce point de vue est relatif) et celui-ci demande une grande rigueur.

## III.3 Choix d'environnement

**Visual studio** C++ est un environnement de développement C++ complet et gratuit

Nous avons choisi **Visual studio** C++ (2010) parce qu'il répond bien à nos exigences et offre :

- Un Compilateur C et C++ (Win32)
- Un Débugger intégré
- Une Colorisation de la syntaxe paramétrable
- Un Travail en mode source ou en mode projet
- Une Création de fichier Setup (pour une installation simplifiée)
- De Nombreuses bibliothèques intégrées.



### III.4 OpenGL

**OpenGL** (Open Graphics Library) est une bibliothèque graphique très complète qui permet aux programmeurs de développer des applications 2D, 3D assez facilement. [29]

Il s'agit d'un ensemble normalisé de fonctions de calcul d'images 2D ou 3D lancé par Silicon Graphics en 1992.

Cette interface de programmation est disponible sur de nombreuses plateformes où elle est utilisée pour des applications qui vont du jeu vidéo jusqu'à la CAO (Conception Assistée par Ordinateur) en passant par la modélisation.

OpenGL permet à un programme de déclarer la géométrie d'objets sous forme de points, de vecteurs, de polygones, de bitmaps et de textures. Elle effectue ensuite des calculs de projection en vue de déterminer l'image à l'écran, en tenant compte de la distance, de l'orientation, des ombres, de la transparence et du cadrage.

L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plateformes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D et certaines applications 2D vectorielles.

De plus, cette bibliothèque est également utilisée dans l'industrie du jeu vidéo où elle est souvent en rivalité avec la bibliothèque de Microsoft : Direct3D. Une version nommée

OpenGL ES a été conçue spécifiquement pour les applications embarquées (téléphones portables, agenda de poche, consoles de jeux, etc).

### III.4.1 Les types de projection OpenGL

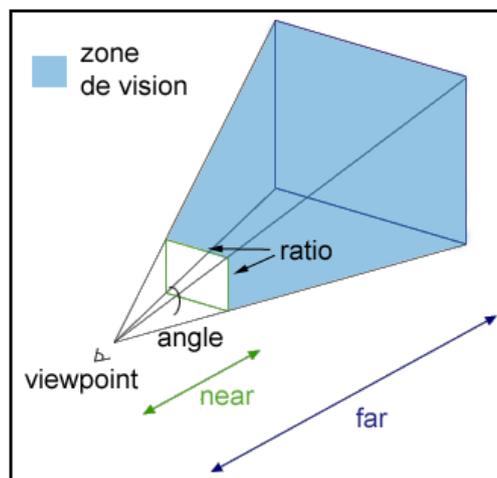
En général, une projection transforme les points dans une dimension  $N$  vers des points dans une dimension inférieure à  $N$  par exemple  $N=3$  vers  $N=2$ . Le passage le plus important dans la 3D est la **projection**. Pour passer d'un monde décrit en 3D à une fenêtre avec des pixels 2D il nous faut perdre une dimension en projetant la scène 3D sur le plan de la fenêtre (l'écran d'affichage). OpenGL nous permet de choisir le mode d'affichage des scènes dessinées.

Deux choix sont offerts: **la projection en perspective** ou bien **la projection parallèle orthographique**.

#### III.4.1.1 La projection en perspective

En perspective, le centre de projection se trouve à une distance finie des objets. Cette projection présente un bon modèle de l'œil humain ou d'un appareil photo et respecte l'effet de perspective : plus un objet est loin plus il est petit. En effet, ce mode préserve les lignes mais pas les angles ni les distances.

La perspective est définie par la pyramide ci-dessous :



*Figure III.1: pyramide de clipping*

En spécifiant les paramètres illustrés dans la pyramide, ce modèle donne une image la plus près de la réalité en montrant les 3 dimensions de l'objet. De ce fait, la méthode de projection la plus utilisée en 3D est la perspective. Dans notre application aussi nous utiliserons ce mode de projection.

### **III.4.1.2 La projection parallèle orthographique**

Dans ce mode de projection, le centre de projection est à l'infini, on préserve les lignes, les distances, les angles. Mais, on perd toute information concernant la profondeur. Cependant l'œil humain ne voit pas comme ça.

## **III.5 Navigation dans un environnement 3D**

### **III.5.1 La problématique**

Notre projet consiste à mettre en œuvre une application de navigation dans un environnement 3D en utilisant OpenGL.

D'abord, nous commencerons par comprendre comment il est possible de faire de la 3D sur un écran 2D, ce qu'il nous faut préparer pour « dessiner » en 3D, Ensuite nous dessinerons un cube et verrons les problèmes que cela pose.

### **III.5.2 Implémentation**

Afin de répondre à notre problématique, nous avons divisé l'implémentation de notre application en deux phases :

- Phase 1 : La construction de la scène virtuelle 3D.
- Phase 2 : La navigation dans la scène virtuelle 3D.

D'une part, la première phase contient le dessin de la scène 3D en utilisant les fonctionnalités offertes par OpenGL.

D'autre part, dans la deuxième phase, nous allons implémenter une caméra contrôlable en C++ à l'aide d'une classe. Cette classe contient les fonctions nécessaires pour gérer deux types d'événements :

- le mouvement de la souris : pour changer l'orientation de la caméra ;
- le clavier : pour déplacer la camera.

## III.6 La gestion de l'orientation

### III.6.1 Les contrôles

Pour pouvoir se déplacer dans notre monde 3D, nous devons utiliser le clavier et la souris. Grâce à OpenGL, nous allons pouvoir lier un événement à un déplacement, ces événements seront divisés en deux catégories. Nous utiliserons :

- La souris : pour orienter la caméra en fonction du mouvement effectué
- Le clavier : pour déplacer la caméra selon 2 axes (l'axe X : gauche, droite et l'axe Z : avant et arrière).

#### III.6.1.1 Rotation à la souris

En maintenant le bouton gauche de la souris enfoncé, les mouvements de la souris feront tourner la scène :

- un mouvement horizontal de la souris donne une rotation horizontale de la scène (donc autour de sa verticale).
- un mouvement vertical de la souris donne une rotation verticale de la scène.

#### III.6.1.2 Gestion du déplacement au clavier

La souris nous permet d'orienter la caméra par rapport aux axes X, Y, Z locaux, le clavier quant à lui va nous permettre de déplacer la caméra selon l'orientation actuelle :

- 2 touches nous permettent de faire **avancer/reculer** la caméra ;
- 2 touches nous permettent de faire **straffer** (déplacement latéral) la caméra.

Nous avons choisi d'utiliser les touches **EZ**, utilisées de façon presque standard pour ce genre de mouvement sur un clavier AZERTY :

#### III.6.1.2 Gestion du regard à la souris

Pour pouvoir regarder tout autour de nous, nous allons utiliser les mouvements de la souris pour orienter la caméra :

- un mouvement horizontal de la souris fait tourner la caméra horizontalement autour de la verticale du monde (regard à gauche et à droite) ;
- un mouvement vertical de la souris fait tourner la caméra verticalement (le regard se lève ou se baisse).

## III 6.2 Bibliothèque utilisée

Nous avons utilisé plusieurs bibliothèques dont certaines développées à partir d'OpenGL afin d'apporter des fonctionnalités qui ne sont pas disponibles dans la bibliothèque OpenGL elle-même :

- **<math.h>**

Pour calculer des fonctions mathématiques courantes (sinus, cosinus, etc.) on a ajouté de nombreuses fonctions mathématiques. Notamment dans le calcul des coordonnées des vecteurs selon lesquelles notre caméra va être déplacée.

- **<gl/glaux.h>**

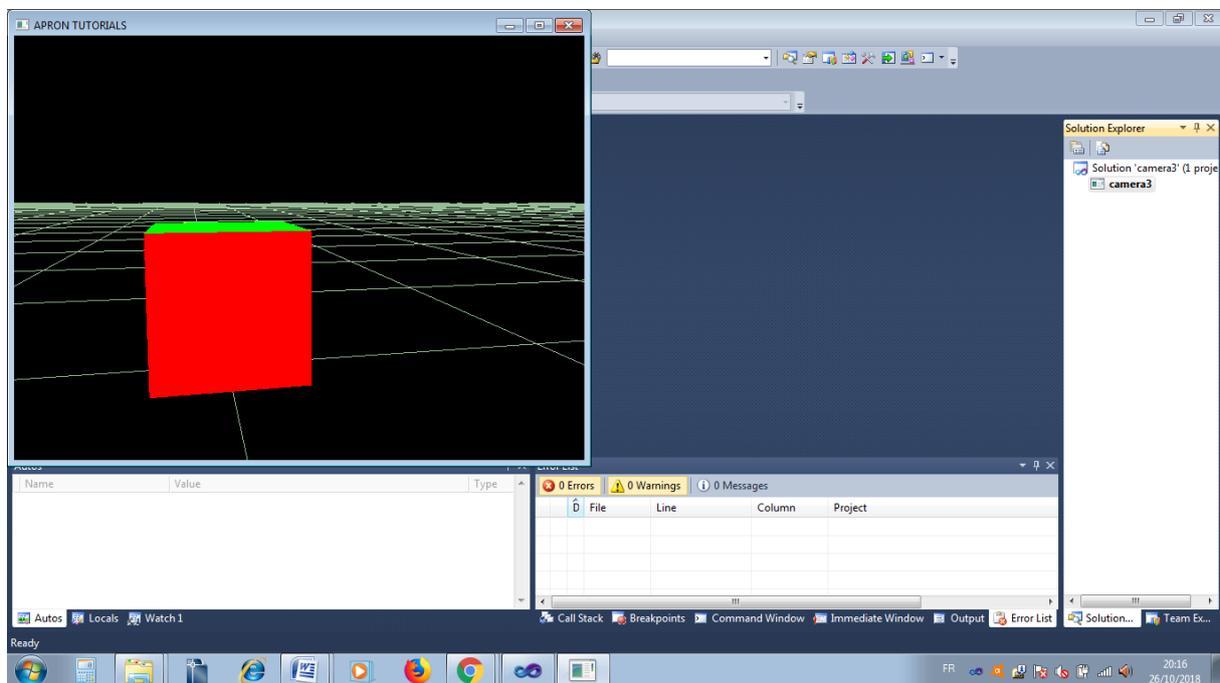
Pour lire les fichiers .bmp

Pour les opérations relatives au système tel que les fonctions de gestion des événements clavier et souris.

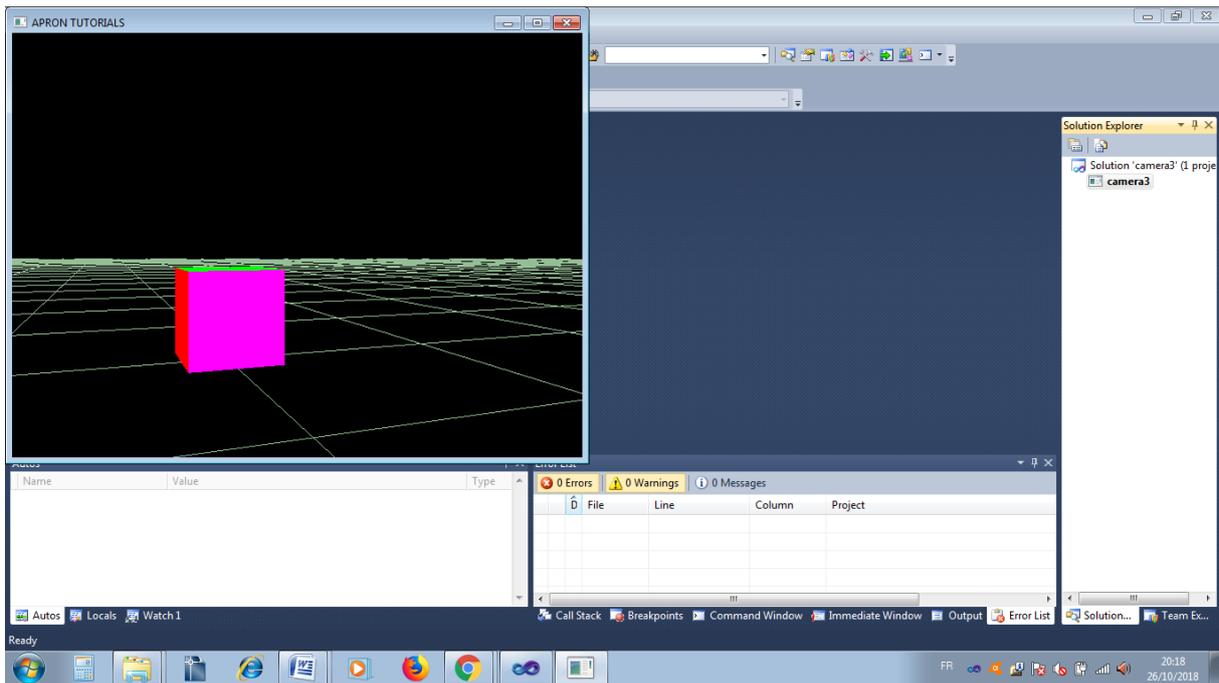
## III .7 Description de la scène

Donc, comme un exemple nous avons choisi de dessiner un cube.

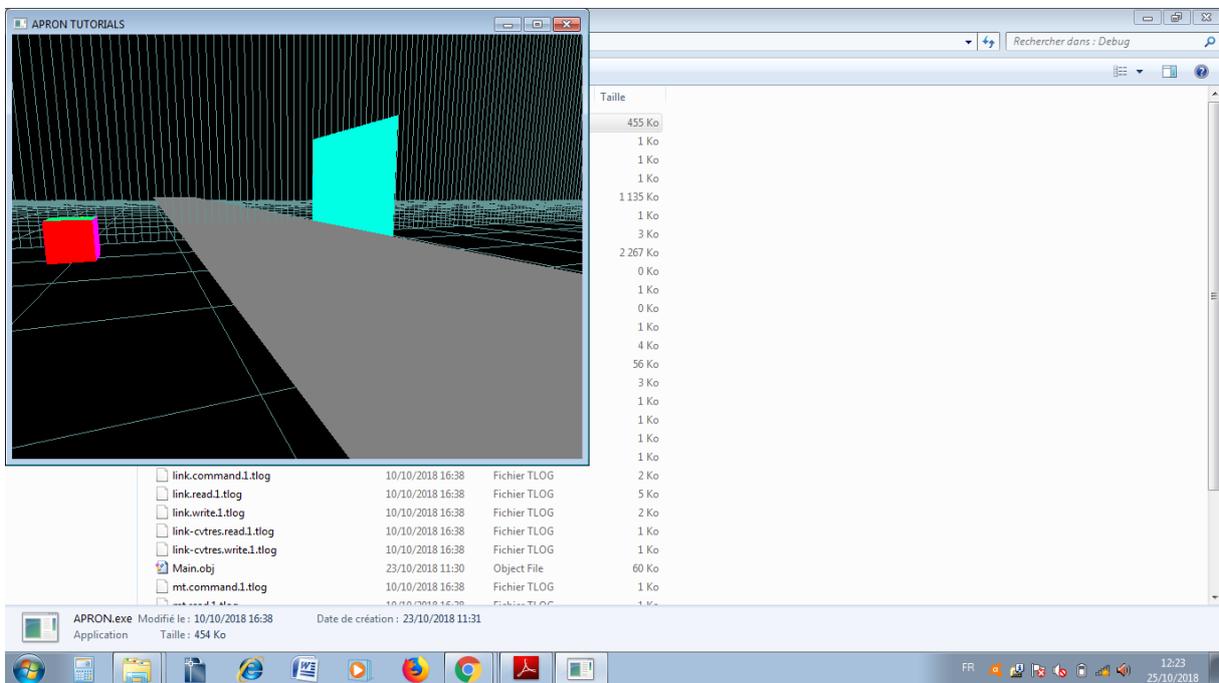
Les traits horizontaux et verticaux représentent le terrain de navigation.



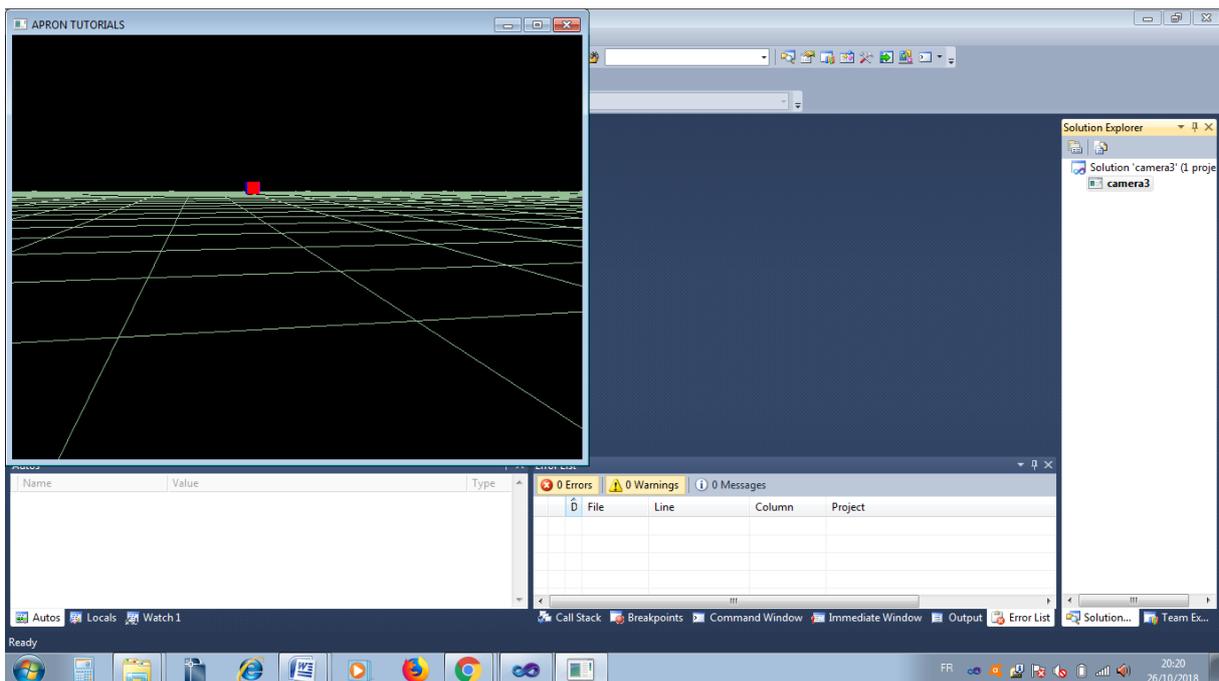
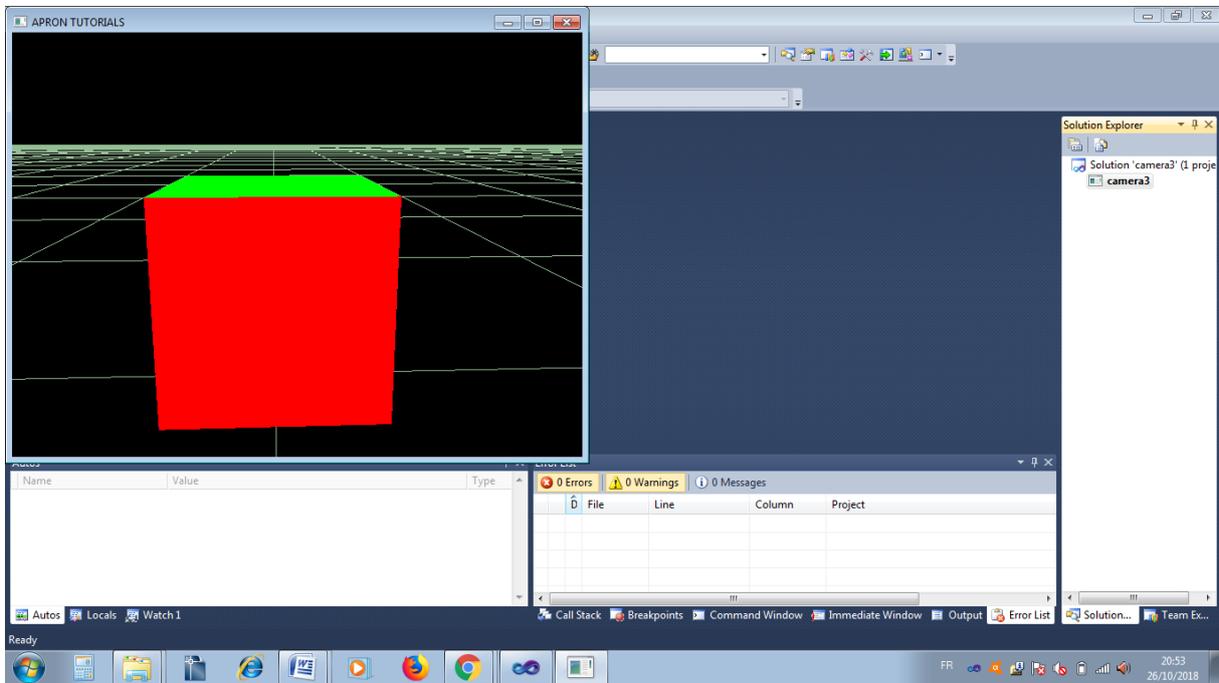
On peut orienter le cube à l'aide de la souris de tous les côtés



Nous pouvons dessiner n'importe quel objet sur la scène (un mur, une route, un bâtiment, etc).



En réalité, notre camera n'est pas déplaçable seulement selon les deux axes (X et Z), mais elle est équipée aussi de l'option « ZOOM » pour permettre à l'utilisateur de zoomer sur les objets.



### **III.8 Conclusion**

Dans ce dernier chapitre nous avons vu une étude de cas dans laquelle nous avons appliqué le principe de la navigation et ses techniques, dans la suite nous clôturons notre travail par une conclusion générale et quelques perspectives envisageables.

## **Conclusion générale et perspectives**

La simulation visuelle est un outil très fort pour examiner un objet ou un système entier, celle-ci doit être dynamique en ce sens que les images sont prises progressivement de plusieurs emplacements et suivant plusieurs orientations ce qui est appelé communément la navigation en trois dimensions.

En outre, elle doit remplacer d'une façon la plus proche possible la vision humaine et d'ailleurs elle s'y inspire. Ceci mène à conclure que les approches et méthodes utilisées en navigation 3D et en vision artificielle d'une façon générale sont compliquées comme la vision humaine l'est elle-même.

Dans ce projet nous avons présenté les notions de base sur la vision artificielle ainsi que le traitement d'images puisque ces deux domaines sont étroitement liés. Ensuite, nous avons détaillé beaucoup plus les concepts de la navigation en 3D et enfin, dans le troisième chapitre, nous avons présenté l'implémentation de notre application.

Lors de l'implémentation de cet outil de navigation, nous avons confronté beaucoup de problèmes, notamment dans la phase navigation. Car, il faut prendre en considération tous les paramètres qui influent sur le mouvement de la camera (vitesse de rotation, angle de vision, etc.), pour offrir à l'utilisateur une vision plus proche de celle du monde réel.

Afin d'apporter plus de réalisme à notre application, il serait intéressant d'ajouter d'autres fonctionnalités permettant aux utilisateurs de :

- choisir la trajectoire selon laquelle la navigation s'effectue (navigation assistée) ;
- lancer une navigation autonome dans la scène à travers une caméra en mouvement permanent ;
- basculer entre les modes de projection offerts (perspective ou orthogonale).

**Résumé :**

La navigation dans un environnement en trois dimensions consiste à simuler la sensation pour un utilisateur humain de se déplacer dans toutes les directions, autour et à travers tous les objets, ceci dans le but de découvrir des caractéristiques visuelles, de rétablir une disposition ou pour toute fin qui ne soit pas possible dans le cas d'un déplacement réel.

Dans ce cadre, notre projet de fin d'étude se résume en la mise en place d'un système de navigation en 3D.

**Mots clés :** navigation 3D, vision artificielle, traitement d'images, simulation.

**Abstract**

Navigation in a three-dimensional environment consists of simulating the sensation for a human user to move in all directions, around and through all objects, in order to discover visual characteristics, restore a disposition or for any end which is not possible in the case of a real displacement.

In this context, our project of end of study is summarized in the installation of a system of navigation in 3D.

**Keywords:** 3D navigation, artificial vision, image processing, simulation

## Bibliographie

- [1] mémoire présenté en vue d'obtenir le diplôme d'ingénieur C.N.A.M.En informatique par Alexandre TOPOL-conservatoire national des arts et métiers Paris.
- [2] Cours Chapitre 03 Généralités sur le traitement d'images
- [3] Cours Traitement d'images et vision artificielle ensta d9-2 *Antoine Manzanera* - Unité d'Électronique et d'Informatique
- [4] Vision par ordinateur outils fondamentaux Radu Horaud \_ CNRS Et Olivier Monga \_ INRIA
- [5] rapport\_veille\_gagnon\_151199 Vision par ordinateur
- [6] presentation\_IMN559\_2011\_2 pages Automne 2011 Vision par ordinateur IMN 559 Présentation Par Pierre-Marc Jodoin.
- [7] <http://dictionnaire.sensagent.leparisien.fr/Vision%20par%20ordinateur/fr-fr/>
- [8] vision1\_intro -vision par ordinateur introduction au domaine applications- Frédéric Devernay
- [9] Doug A. BOWMAN, Ernst KRUIJFF, Joseph J. LAVIOLA et Ivan POUPYREV : An introduction to 3D user interface design. Presence 10 (1), 2001b.
- [10] Interaction 3D en Réalité Virtuelle- Etat de l'art Nassima Ouramdane- Samir Otmane- Malik Mallem-*Laboratoire IBISC CNRS FRE 3190, Université d'Evry Val d'Essonne*
- [11] Nick BOBICK : Rotating objects using quaternions. Game Developer, 2(26):21–31, 1998.
- [12] Système de visualisation 3d avec opengl pour la navigation dans un environnement fermé (cas d'un labyrinthe)
- [13] <http://www.letendanceur.bzh/la-3d-dans-tous-ses-etats/>
- [14] J. A. Watlington, M. Lucente, C. J. Sparrell, V. M. Bove, I. Tamitani. A hardware architecture for rapid generation of electro-holographic fringe patterns. *In proceedings of SPIE*, 1995.
- [15] U. Obeysekare, C. Williams, J. Durbin, L. Rosenblum, R. Rosenberg, F. Grinstein, R. Ramaturi, A. Landsberg, W. Sandberg. Virtual Workbench – A non-immersive Virtual Environment for Visualizing and Interacting with 3D Objects for Scientific Visualization. *In proceedings of Visualization*, San Francisco, USA, 1996.
- [16] [https://www.conrad.fr/ce/fr/content/3d\\_infos/Alles-Wissenswertes-rund-um-das-Thema-3D-finden-Sie-bei-Conrad](https://www.conrad.fr/ce/fr/content/3d_infos/Alles-Wissenswertes-rund-um-das-Thema-3D-finden-Sie-bei-Conrad)
- [17] Burdea G., Coiffet P., La réalité virtuelle, Hermès-Paris, chapter 2, p. 243-251, 1993.

[18]-Fuchs P., Arnaldi B., Tisseau J., *La réalité virtuelle et ses applications*, Les Presses de l'Ecole des Mines de Paris, chapter 1, p. 3-52, 2003.

[19] Fuchs p., Moreau G., Burkhardt G.-M., Coquillart S., *le traité de la réalité virtuelle, volume 2 :L'interfaçage, l'immersion et l'interaction en environnement virtuel*, Edition Les Presses de l'Ecole des Mines de Paris, 2006.

[20] *Interaction en réalité virtuelle*, PhD thesis, Université Louis Pasteur de Strasbourg 1, 28 novembre, 2006.

[21] Rheingold H., « *Virtual Reality* », *Simon & Schuster, Inc.*, 1991.

[22] (Dumas et al., 1999). Dumas C., Plénacoste P., Chaillou C., « Définition d'un modèle d'interaction pour une interface de travail tridimensionnelle à partir d'expérimentation », In *Proceedings of IHM'99*, 1999.

[23] Bowman D. A., Koller D., Hodges L., « Travel in immersive virtual environments : An evaluation of view-point motion control techniques », *Proceedings of IEEE Virtual Reality Annual International Symposium*, vol. 7, p. 45-52, 1997.

[24] Zhai S., Buxton W., Milgram P., « The "Silk Cursor" : Investigating Transparency for 3D Target Acquisition », *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, vol. 1, p. 459-464, 1994.

[25] Bowman et al., 2005 Bowman A., Kruijff E., Laviola J. Poupyrev I., *3D User Interfaces : Theory and Practice*, Addison-Wesley, 2005.

[26] (Ware et Osborne, 1990) : Colin WARE et Steven OSBORNE : Exploration and virtual camera control in virtual three dimensional environments. In *Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 175–183. ACM Press New York, NY, USA, 1990.

[27] <https://desktop.arcgis.com/fr/arcmap/latest/extensions/3d-analyst/fundamentals-of-3d-navigation-in-arcscene.htm>

[28] <https://openclassrooms.com/fr/courses/167717-creez-des-programmes-en-3d-avec-opengl/164454-introduction-a-opengl#/id/r-164188>

[29] <https://openclassrooms.com/fr/courses/167717-creez-des-programmes-en-3d-avec-opengl/165319-enfin-de-la-3d-partie-1-2>