

---

# Chapitre 3

## Les niveaux de détails et la simulation comportementale:

---

### 3. Les niveaux de détails

#### 3.1. Introduction :

La simulation de scènes en synthèse d'images est une technique maîtrisée, largement utilisée pour les effets spéciaux cinématographique et les jeux vidéo. La possibilité de rendre un mouvement réaliste constitue la partie essentielle de la plupart des environnements virtuels et des applications de simulation visuelle.

En effet, malgré le développement des machines graphiques, la visualisation en temps réel de modèles complexes reste une véritable problématique. Il existera toujours des scènes dont la complexité dépasse les capacités de calcul d'une machine aussi puissante soit-elle. Ceci est d'autant plus vrai que les besoins des utilisateurs évoluent autant – si ce n'est plus vite – que les progrès en informatique [Bel98].

Il faut pour cela mettre en œuvre une optimisation dont la multirésolution fait partie. La multirésolution ou la représentation d'un modèle avec des niveaux de détails plus ou moins fins selon les besoins de l'utilisateur et le type d'application. La méthode consiste à simplifier la géométrie des objets, à en produire plusieurs représentations puis à visualiser le modèle à l'aide d'un algorithme qui réponde aux critères de l'utilisateur. Nous présenterons, avec plus de détails, les enjeux et les difficultés de cette approche dans la première partie de ce chapitre.

La deuxième partie est consacrée à la présentation de l'exploitation de la technique de niveaux de détails dans la simulation comportementale.

## **3.2. Les niveaux de détails :**

Réaliser un niveau de détail d'un élément consiste à en construire une version moins complexe géométriquement, c'est-à-dire comportant moins de facettes [Val99]. Il s'agit donc d'afficher des versions plus au moins dégradées de l'objet selon certains critères tel que la distance entre la caméra et l'objet ou la taille de l'objet à l'écran.

La complexité des environnements virtuels influence le temps de traitement informatique et d'affichage. Plus on ajoute de complexité visuelle à une scène, plus long sera le délai induit au niveau du rendu. Pour bien préciser l'utilité de l'aspect niveaux de détails dans le rendu des scènes en synthèse d'image on va commencer par la définition de la complexité dans une scène que ce soit statique ou dynamique.

### **3.2.1. La complexité dans une scène statique :**

Une scène géométrique est définie par un ensemble de données comme les maillages polygonaux, les textures ou les sources de lumière. La complexité d'une scène dans l'absolu est fonction de la quantité de chacun des divers éléments la composant [Val99]. Lorsqu'on a une scène trop complexe, il devient nécessaire d'en simplifier certaines parties. Il faut rejeter les éléments non visibles ou cachés par d'autres éléments dans le processus de calcul d'image. En ce qui concerne les éléments visibles (au moins partiellement), la suppression est également envisageable mais à utiliser avec précaution car certains éléments peuvent être importants pour l'application. En revanche, la dégradation de l'élément, c.à.d. son remplacement par un élément plus simple (avec moins de détails) est très utile.

### **3.2.2. La complexité dans une scène dynamique:**

La complexité d'une scène animée dépend du nombre d'entités animées, du type de mouvement qu'on veut avoir et du taux de réalisme qu'on souhaite obtenir.

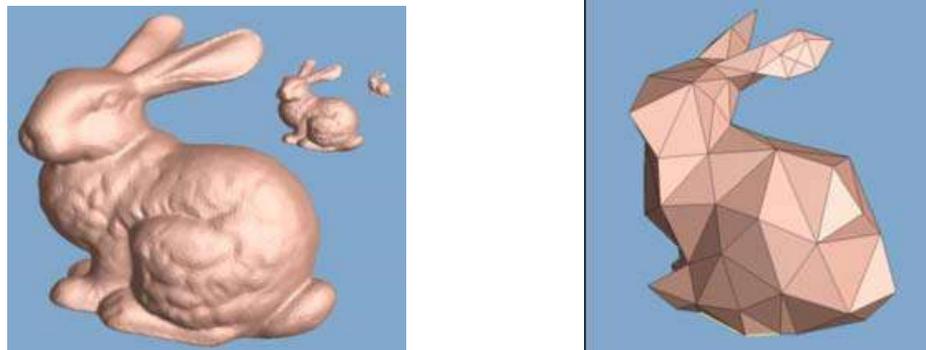
Le principe le plus utilisé pour gérer la complexité en animation consiste à créer des NDDs de modèle d'animation, c'est-à-dire une famille de modèles d'animation avec des coûts de calcul différents et permettant d'animer une même partie de la scène ou une même entité, de telle sorte qu'on utilise une version dégradée pour les entités éloignées de la caméra qui ne nécessite pas beaucoup de détails, et une version plus détaillée pour les

entités les plus proche de la caméra pour qu'on puisse avoir un niveau de réalisme en conséquence.

### 3.2.3. Les types de LODs

#### 3.2.3.1. Niveaux de détails discrets

C'est le cas le plus simple pour la représentation des LODs des maillages, Il consiste en une collection de maillages de différentes tailles. Chaque maillage représente un objet avec une résolution différente. Chacune de ces représentations est définie d'une façon indépendante des autres. Toutes les représentations sont générées dans une étape de prétraitement et sont stockées. Une représentation est choisie au besoin (au moment du traitement) selon certain critère de sélection [Con01].



**Figure 3.1: Niveaux de détails discrets**

- **Avantage :**
  - L'avantage le plus important des LODs statiques est la simplicité de la programmation. Il y a une séparation entre l'algorithme de simplification et celui du rendu, ce qui implique un programme simple contenant seulement la sélection du LOD le plus approprié.
- **Inconvénient :**
  - Chaque représentation doit être stockée indépendamment, la taille de la mémoire nécessaire pour le stockage des représentations augmente avec le nombre de LODs stockés, en pratique il y a toujours des contraintes pour la taille requise, ce qui signifie qu'on ne peut pas stocker plusieurs LODs. [Ben 04].

### 3.2.3.2. Niveaux de détail continus

Dans le cas de LODs continus, la structure de données est créée de telle sorte que n'importe quel LOD peut être extrait au moment du traitement. Une telle structure de données est appelée maillage *multirésolution*. Selon la complexité de la structure de donnée, Il est possible d'obtenir un petit ou un grand nombre de maillage à différentes résolutions [Con01].

Le nombre de LODs qui peuvent être générés n'est pas fixé à priori. On peut citer quelques propriétés de la multirésolution continue :

- Elle doit être capable de générer la résolution désirée dans un temps très court (temps réel).
- La transition entre les différentes résolutions ne doit pas être perceptible.

A cause du nombre important des représentations possibles d'un modèle original, la création et le stockage de chacune d'elles sont impossibles. Les informations communes de ces représentations sont donc utilisées pour créer une seule représentation unifiée. A partir de cette dernière, le LOD désiré est extrait au moment de traitement selon les paramètres de vue.

- **Avantages:**

- Les LODs dynamiques exécutent quelques simplifications comme un prétraitement, mais reportent le calcul du reste du travail au système de visualisation au moment du traitement. Cela permet d'incorporer plus de critères de sélection de LODs.
- Les LODs dynamiques ajustent les détails graduellement et avec augmentation, en réduisant le passe visuel.

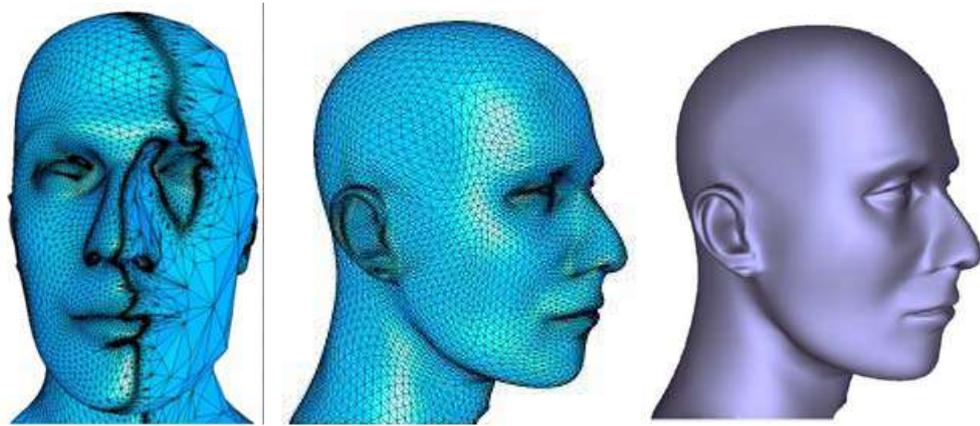
- **Inconvénient :**

- Requièrent un espace mémoire et un temps de calcul supérieur à celui des LODs statiques.

### 3.2.3.3. Les LODs dépendants du point de vue

C'est une extension des LOD continus qui intègre un critère de simplification qui est dépendant du point de vue. Ainsi cette représentation est anisotropique: différentes zones du même objet sont visualisées à des niveaux de détail différents.

Les parties de l'objet proches du point de vue peuvent apparaître à une résolution plus haute que les parties éloignées, cette méthode permet de réaliser des dégradations plus importantes qu'avec les méthodes précédentes, mais à un coût de calcul relativement élevé.



**Figure 3.2 : Niveaux de détails dépendants du point de vue**

En pratique :

En pratique les LOD discrets sont les plus utilisés. Les LOD continus n'apparaissent que dans quelques rares applications spécifiques à cause du coût de calcul de la géométrie en cours d'exécution [Lrc+ 03].

### **3.2.4. Création des niveaux de détails :**

#### **3.2.4.1. Objectifs algorithmiques**

En traversant les NDDs des modèles du plus complet au plus dégradé, on constate [val99] :

- Une réduction de l'espace des trajectoires possibles.
- Une réduction des coûts de calcul.
- Une moins bonne prise en compte des contraintes extérieures.

Idéalement, les algorithmes de créations des niveaux de détails doivent posséder plusieurs qualités [Kru97]:

- Pouvoir créer une suite de représentations de plus en plus simplifiées.

- Pouvoir mesurer et contrôler de manière intuitive le degré de simplification. Cela passe par la définition d'une mesure de l'approximation.
  - Tenir compte des parties significatives des objets pour les préserver dans les représentations simplifiées. Intuitivement, il s'agit de trouver les parties de l'objet que l'observateur percevra le plus facilement.
  - Pouvoir faire varier le degré de simplification à travers l'objet pour, par exemple, simplifier davantage les parties peu importantes ou lointaines d'un objet.
- **Difficultés :**

D'après G. Debunne dans [Deb00], il y a deux problèmes principaux dans la mise en place des approches basées sur le principe de NDD :

    - Les différentes résolutions que l'on veut mélanger doivent avoir, dans une certaine mesure, le même comportement dynamique, c'est-à-dire approximer le même mouvement au cours du temps. Si tel n'était pas le cas, la simulation serait chaotique, son comportement dépendant des résolutions employées alors que c'est précisément ce que l'on veut cacher à l'utilisateur.
    - Les différentes résolutions doivent pouvoir cohabiter, c'est-à-dire que les transitions entre les niveaux ne doivent pas être visibles. Ces transitions sont d'une grande importance et de leur discrétion dépend le résultat. Cette notion est à différencier de la précédente, car même avec un comportement dynamique identique, deux résolutions ne peuvent pas donner un bon résultat si elles ne cohabitent pas. Deux mouvements, même similaires, doivent par exemple être en phase à l'endroit où l'on passe de l'un à l'autre. On devra aussi veiller à éviter les clignotements dus à une oscillation entre deux résolutions ainsi que les apparitions soudaines de détails qui ne faisaient pas partie de la résolution grossière précédente, ou tout autre indice visible dépendant de l'application.

#### 3.2.4.2. Simplification orienté géométrie :

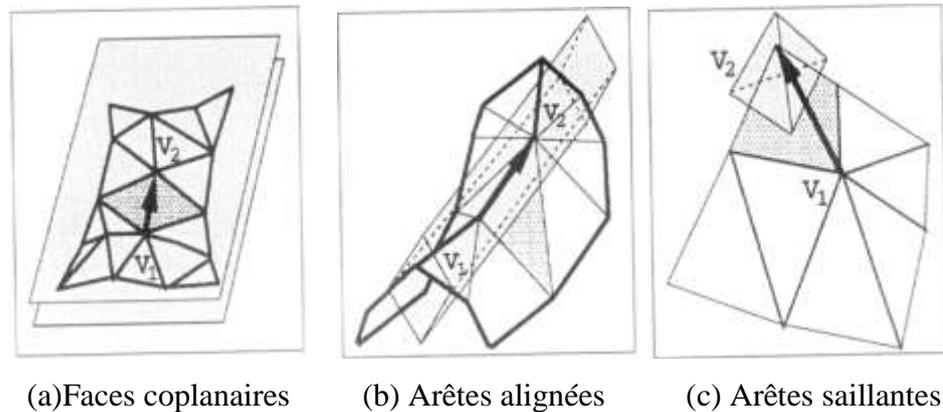
Pour simplifier les objets polygonaux il existe deux types de méthodes. Les premières cherchent à simplifier la représentation géométrique des objets, en réduisant le nombre de polygones par exemple. On parle alors d'algorithmes de simplification polygonale. Les deuxièmes utilisent une représentation totalement différente, par exemple une boîte texturée par une image issue d'une représentation détaillée de l'objet. [Kru 97]

### 3.2.4.2.1. Simplification géométrique polygonale:

L'objectif d'un algorithme de simplification polygonale est de prendre un modèle détaillé avec un nombre important de polygones et de générer un modèle plus simple avec un nombre inférieur de polygones qui apparaît semblable au modèle d'origine, tout en conservant leurs importantes caractéristiques visuelles. L'avantage de la représentation simplifiée est qu'elle peut être rendue plus rapidement qu'un modèle original [Con01].

La simplification doit être faite de telle sorte que la forme générale du modèle soit préservée. Quelques particularités des algorithmes de simplification qui doivent être précisés sont [Kbg+97]:

- **Faces coplanaires (planer area)** : sont identifiées en examinant les normales des polygones adjacents, ces polygones peuvent être fusionnés pour former un polygone plus grand. C'est le type de simplification le plus facile. **Figure (3.3.a)**
- **Arêtes alignées (Sharp edges)** : sont trouvées en comparant les angles entre les normales des facettes adjacentes, elles peuvent être simplifiées en fusionnant les arêtes connectées qui sont presque colinéaires. **Figure (3.3.b)**
- **Arêtes saillantes (Pointed edges)** : doivent être préservées, elles peuvent être détectées en utilisant la courbure locale autour d'un sommet. **Figure (3.3.c)**



**Figure 3.3 : Formes caractéristiques**

### 3.2.4.2.2. Opérateurs de simplification :

Les algorithmes qui entrent dans cette catégorie sont indépendants les uns des autres mais partagent cependant un certain nombre de caractéristiques. Astheimer et

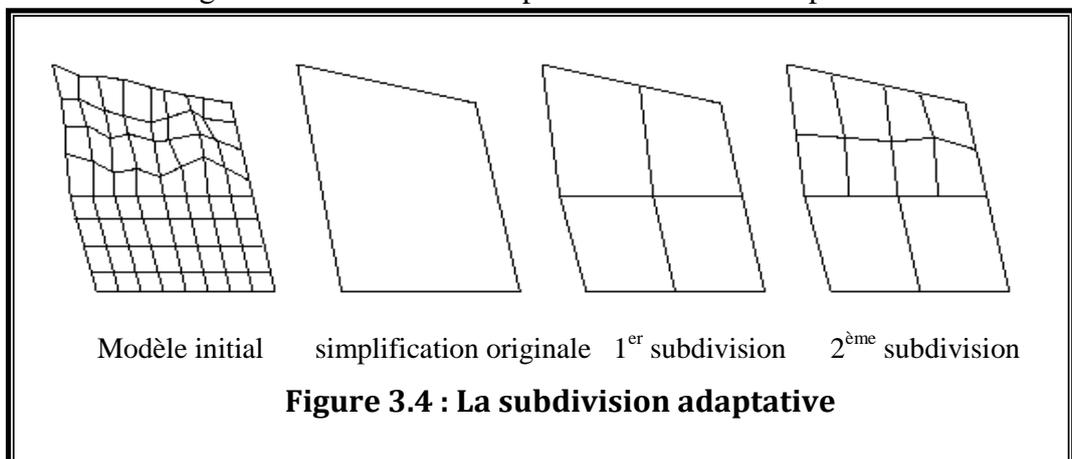
Pöche présentent dans [AP94] une liste, définie de manière intuitive, d'opérateurs de simplification :

- **Normalisation** : élimine les points et arêtes définis plusieurs fois.
- **Simplification des Sommets** : tous les points à l'intérieur d'un volume (cube d'une grille uniforme ou sphère autour d'un point) sont regroupés. Ainsi, les amas de points et les petites facettes sont combinés.
- **Simplification des arêtes** : tous les bords d'une longueur inférieure à une longueur donnée sont éliminés.
- **Simplification basée sur les angles** : les bords qui contiennent un angle inférieur à une certaine valeur sont éliminés.
- **Simplification basée sur la surface d'une facette** : les facettes dont la surface est inférieure à une certaine valeur sont éliminées. Les trous qui en résultent peuvent être bouchés par triangulation en utilisant, par exemple, le barycentre des points retirés.
- **Simplification basée sur la coplanarité d'un ensemble de facettes** : les facettes dont les normales sont presque parallèles sont éliminées (il faut aussi boucher le trou). Il faut noter qu'il est nécessaire de connaître les relations de voisinage en facettes.

En se basant sur ces opérateurs, il est possible de définir une classification des méthodes de simplification polygonale:

#### 3.2.4.2.3. Subdivision Adaptative (Adaptative Subdivision) :

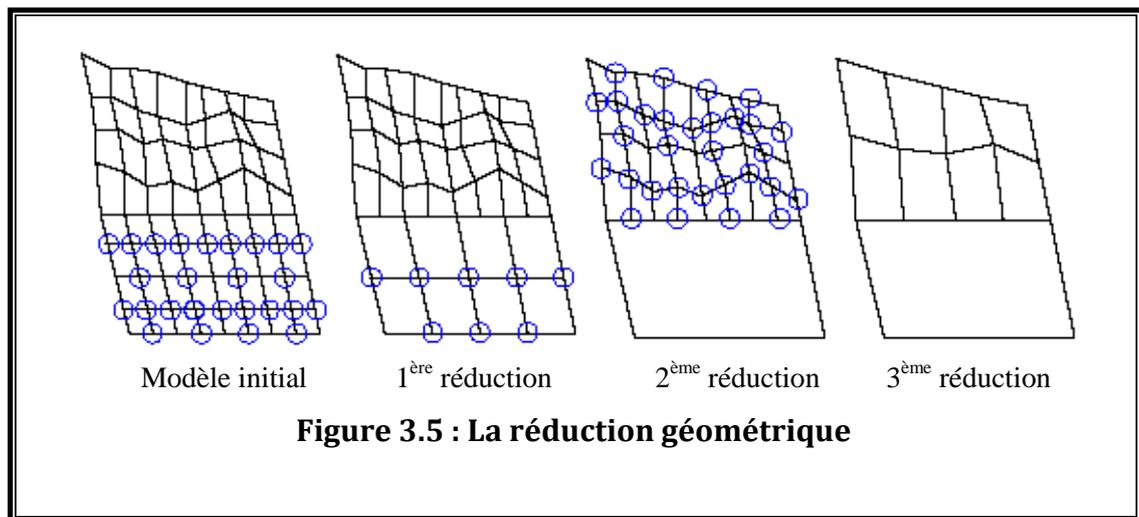
Cette méthode commence avec un modèle de base très simple et le subdivise récursivement en ajoutant des détails à certains endroits du modèle à chaque étape. L'algorithme s'arrête lorsque le modèle approxime le modèle original à un degré spécifié par l'utilisateur. La figure 3.4 montre un exemple de subdivision adaptative.



Cette méthode est assez peu utilisée car il n'est pas facile de construire la simplification initiale dans le cas général [Kru99].

#### 3.2.4.2.4. Réduction Géométrique (Geometry Removal):

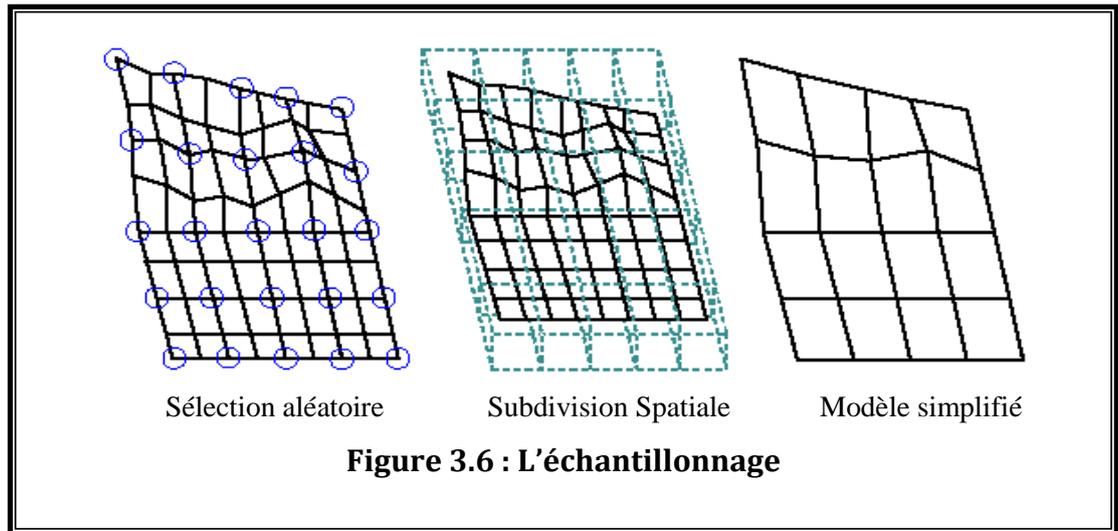
Cette méthode part du modèle d'origine et le simplifie en retirant des faces ou des sommets récursivement. L'algorithme s'arrête lorsqu'il ne peut plus retirer de géométrie et lorsque le modèle satisfait un degré d'approximation spécifié par l'utilisateur. La majorité de ces algorithmes ne retirent des faces ou des sommets que si cela ne cause pas de violation de la topologie. La figure 3.5 montre un exemple de réduction géométrique.



Cette technique est utilisée fréquemment dans les algorithmes de simplification polygonale les plus récents [Kru99].

#### 3.2.4.2.5. Echantillonnage (Sampling):

Cette méthode fonctionne de manière opposée à la méthode précédente au sens où elle cherche à trouver les primitives à conserver plutôt que celles à retirer. Elle effectue un échantillonnage de la géométrie du modèle original, soit en choisissant arbitrairement un certain nombre de sommets, soit en englobant le modèle dans une grille tridimensionnelle, et en échantillonnant chaque boîte de la grille. L'algorithme essaye alors de créer un modèle simplifié qui soit proche des données échantillonnées. L'utilisateur peut contrôler le degré d'approximation en changeant le nombre de points ou la taille de la grille. La figure 3.6 montre un exemple d'échantillonnage.



Cette méthode, historiquement assez importante, n'est pas fréquemment utilisée dans les algorithmes récents. Il est en effet plus difficile de trouver l'ensemble des sommets à préserver, plutôt que de tester chaque sommet pour savoir s'il peut être retiré [Kru99].

Pour se construire une idée plus approfondie sur les trois catégories de la méthode de simplification polygonale le lecteur se reportera à [Red97] et [Kru99] pour voir les différentes techniques publiées dans chacune de ces catégories.

Nous avons relativement détaillé cette technique (Simplification polygonale) parce que pratiquement tous les systèmes de réalité virtuelle utilisent les polygones pour le rendu de leurs scènes.

#### 3.2.4.2.6. Simplification structurelle, polygone texture :

Ces méthodes changent la structure de représentation des objets. Par exemple il est possible de remplacer un objet polygonal par une boîte englobante texturée à l'aide d'une image produite à partir d'une version détaillée de l'objet. Les LODs créés avec cette méthode sont appelés des imposteurs. Ces méthodes produisent souvent des imposteurs qui dépendent du point de vue et posent des problèmes pour les calculs d'éclairage dans le cas où les projections d'ombres sont nécessaires (le résultat est correct dans le cas d'un éclairage ambiant). Certaines méthodes proposent de remplacer les objets par un polygone texturé à l'aide d'une image de l'objet original calculée avec un haut niveau de détails [Kru 97].

### 3.2.4.3. Simplification orientée scène :

La plupart des algorithmes orientés scène cherche à simplifier des régions de la scène plutôt que les objets eux-mêmes. Ainsi dans une phase de pré-calcul ils décomposent récursivement la scène en zones 3D (en utilisant un octree ou une grille 3D) pour obtenir une description hiérarchique, chaque niveau étant plus détaillé que le précédent. Ensuite à chaque nœud de la hiérarchie une représentation simplifiée de la sous hiérarchie est produite. Cette représentation sera utilisée si les critères de sélection sont satisfaits.

Toutes ces méthodes obligent à opérer au sein de l'algorithme de sélection des LODs un parcours d'arbre pour chaque image afin de déterminer le niveau de détails à utiliser. De plus, elles ne sont vraiment efficaces que sur des scènes très profondes sans occlusion due par exemple à des murs [Kru 97].

#### 3.2.4.3.1. Boite colorée :

Cet algorithme est proposé par [Cdl+95] il décompose la scène à l'aide d'un octree. Il fait ensuite correspondre à chaque nœud un cube chaque face ayant une couleur représentative de la couleur de l'ensemble des facettes comprises dans le sous arbre. Cet arbre construit a priori est utilisé dynamiquement. En cela l'algorithme peut être vu comme simplifiant une région de la scène plutôt qu'un objet. L'algorithme de rendu parcourt l'arbre et détermine la taille dans l'espace image de la boîte englobante à chaque nœud. Si cette taille est inférieure au pixel alors le cube est utilisé et les facettes du sous arbre sont ignorées. Si la taille est supérieure l'algorithme descend récursivement l'arbre.

Toutefois, cet algorithme fait apparaître des trous entre les cubes et il augmente la taille globale de l'objet. De plus le rendu est plus sombre qu'avec l'objet original.

#### 3.2.4.3.2. Fusion hiérarchique :

Cet algorithme est proposé par [Lue 96]. Il décompose la scène en utilisant un octree. Pour chaque nœud, il fusionne l'ensemble des sommets dans le sous arbre en un sommet représentatif (estimé en fonction de la courbure locale). Ensuite, lors de l'affichage il parcourt l'arbre et détermine en fonction de la taille dans l'espace image quel nœud doit être utilisé. Cette méthode a l'avantage de pouvoir donner à différentes parties d'un objet des LODs différents. Ceci est particulièrement intéressant pour les grands objets.

### 3.2.4.3.3. Modèles d'illumination :

En utilisant des techniques d'illumination différentes pour l'éclairage et l'ombrage, on peut obtenir différents niveaux de détails pour les objets. Ceci signifie que nous pouvons utiliser par exemple moins de polygones et un algorithme d'illumination amélioré pour obtenir une représentation semblable à celle avec plus de polygones.

Néanmoins, certaines de ces techniques exigent un nombre important d'opérations de calcul pour être implémentées, ainsi leur utilisation n'est pas toujours appropriée. Quelques modèles d'éclairage incluant les modèles de Phong et de Gouraud produisent des résultats plus réalistes que ceux produits par le modèle de Lambert [Con01].

### 3.2.5. Classification:

L'art de l'utilisation des niveaux de détails consiste à trouver le juste équilibre entre le réalisme et la vitesse de calcul. Ainsi, nous cherchons à décrire l'importance d'un objet dans une scène afin de choisir la finesse de sa représentation [Fts+ 93] stipule que la gestion des différents LODs doit être faite de manière prédictive, basée sur la complexité de la scène plutôt que réactive, basée sur le temps de calcul de l'image précédente. Ainsi, il est possible de garantir un temps de rendu inférieur à une certaine limite. A cette fin les auteurs définissent un certain nombre de facteurs qui permettent de choisir un LOD parmi d'autres.

#### 3.2.5.1. La sélection d'un NDD:

Différentes techniques sont utilisées pour choisir un niveau spécifique de détail. Ces techniques essayent d'éviter la transaction inutile entre les NDDs en réduisant les détails seulement quand ils ne sont pas perceptibles, offrant le bénéfice d'améliorer la réaction du système sans aucune perte dans le coût de transition entre les niveaux de détails.

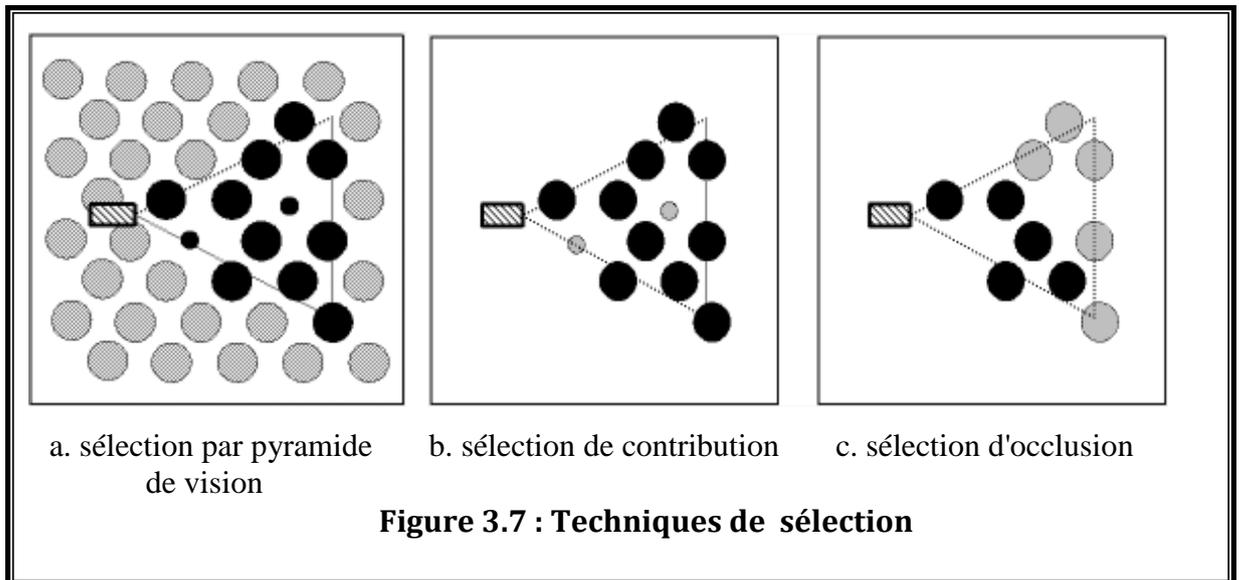
Z. Constantinescu dans [Con01] a classifié ces méthodes dans les groupes suivants :

- Le premier groupe se concentre à l'enlèvement des détails qui ne doivent pas être rendus par le matériel d'affichage graphique. Dans ce cas, nous avons différentes techniques de sélection, lesquelles se basant sur certains critères, elles éliminent quelques parties du modèle.

- Le deuxième groupe consiste à éliminer les détails qui ne peuvent pas être rendus par le matériel. Parmi les méthodes qui utilisent ce principe nous citons celles basées sur la distance (§3.2.5.1.2) et sur la taille de l'objet (§ 3.2.5.1.3).
- Dans le troisième groupe nous avons les algorithmes qui éliminent les détails qui ne peuvent être perçus par l'observateur. Ces méthodes sont celles basées sur la vision périphérique (§ 3.2.5.1.4), la profondeur du champ de vision (§ 3.2.5.1.5) et la vitesse (§ 3.2.5.1.6). Il existe aussi une autre méthode dont le but principal est de maintenir un taux d'affichage constant, indépendamment de la complexité du modèle.

### 3.2.5.1.1. Sélection (culling):

Il existe des situations où l'utilisateur ne voit jamais quelques parties de l'objet. Dans ces cas, ces parties ne sont pas représentées par le système graphique. Dans ce qui suit, quelque cas de ces situations sont présentées [Cab97].



- La méthode de sélection par pyramide de vision (view frustum culling) enlève tous les objets qui sont à l'extérieur de la pyramide de vision de l'observateur. Un tel cas est présenté dans la figure (3.7.a). Tous les objets qui sont à l'extérieur de la pyramide sont écartés (ils sont présentés en gris dans cette figure). Cela améliore la performance de la représentation graphique en éliminant le travail inutile dans les premières étapes du rendu.

- La deuxième méthode est la sélection de contribution (contribution culling). Dans cette méthode les objets dont la contribution visuelle est inférieure à un certain seuil sont écartés, même s'ils sont visibles. Les facteurs définissant la contribution sont généralement la taille de l'objet, la taille de la fenêtre d'affichage et la résolution de l'écran. Cette situation est présentée dans la figure (3.7.b) dont quelques petits objets (en gris) sont écartés, bien qu'ils sont à l'intérieur de la pyramide de vision. Cette méthode réduit l'exactitude fournie par la scène originale, mais elle peut réduire sensiblement sa complexité et spécialement quand nous avons plusieurs objets de petite taille.
- La technique de sélection la plus puissante est la sélection d'occlusion (occluding culling). Dans ce cas (présenté dans la figure 3.7.c) la sélection est obtenue par la découverte et l'enlèvement des objets qui sont cachés ou occlus par d'autres objets. Il est aussi possible, d'écarter les objets qui sont partiellement occlus.

#### 3.2.5.1.2. Distance :

La deuxième technique prend en considération la distance de l'objet par rapport à l'observateur. La mesure est basée sur la distance Euclidienne entre le point de vue et un point prédéterminé à l'intérieur de l'objet. Cette approche est basée sur la théorie suivante: "plus l'objet est loin du point de vue, moins de détails de ces composants sont visibles". Cela signifie que nous pouvons choisir une représentation moins complexe sans trop affecter l'exactitude de l'image [Con01].

- **Avantages :**

- La simplicité : tout ce qui est exigé est de vérifier si une distance excède un seuil prédéterminé.
- L'efficacité : cette technique est efficace car un seul calcul doit être effectué, c'est le calcul de la distance entre le point de vue et l'objet.

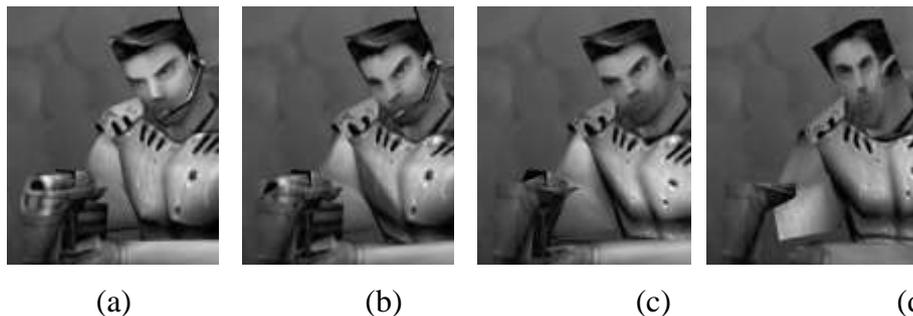
- **Inconvénients :**

- L'inconvénient principal est lié à la manière de choisir un point du volume de l'objet. Si nous choisissons un point fixe dans l'objet, la distance réelle entre le point le plus proche de l'objet et le point de vue peut changer selon l'orientation de

l'objet. Pour les objets grands ou très proches, le choix de ce point n'est pas toujours facile.

➤ Le deuxième inconvénient peut apparaître lorsqu'on fait un changement d'échelle d'un objet, ou quand on change la résolution d'affichage, dans ce cas le seuil de la distance originale sera invalide et doit lui aussi changer.

Cette technique a été largement employée dans les simulateurs de vol et dans la visualisation de terrain.



Le niveau du détail varie en fonction de la distance. Le modèle original (a) contient plus de 600 polygones (à l'exclusion de l'arme), tandis que (b), (c), et (d), le nombre de sommets a réduit à 75%, à 50%, et à 25%. © 1999-2001 Epic Games, Inc. de copyright.

**Figure 3.8 : Le niveau de détail en fonction de la distance**

### 3.2.5.1.3. Taille (Size) :

Une troisième technique profite de la capacité réduite de l'œil à percevoir les objets comme la diminution de la taille de ces objets. Elle prend en considération la taille des objets représentés, les objets les plus petits étant représentés avec moins de détails, tandis que les objets les plus grands sont représentés avec plus de détails [Con01]. Cette technique peut être considérée comme une autre façon d'implémentation de la distance dans les niveaux de détails, en effet, plus un objet est loin, plus sa taille apparente sera réduite.

- **Avantage :**

➤ fournit une mesure pour déterminer la visibilité de particularités dans un objet, indépendamment de la résolution d'affichage ou de changement d'échelle de l'objet. Dans cette technique on n'a plus besoin de choisir un point pour le calcul de la distance.

- **Inconvénient :**

- Comparée à la méthode précédente, elle est plus coûteuse en temps de calculs, parce que nous devons d'abord faire une projection des objets dans l'espace de vision et calculer ensuite leurs tailles avec les coordonnées de vue. Cependant, il existe quelques implémentations très efficaces pour cette méthode dans la littérature.

#### 3.2.5.1.4. Vision périphérique (Eccentricity) :

Une autre technique, pour choisir un NDD, profite de la capacité réduite de l'œil à percevoir les objets en dehors du centre du champ de vision (la périphérie). Cela suggère de diviser le champ de vision en deux secteurs: un correspondant au centre du champ de vision de l'œil pour les régions détaillées, et l'autre correspondant à la périphérie du champ de vision de l'œil pour les régions moins détaillées.

Des expériences ont montré que nous pouvons réduire la complexité visuelle dans la périphérie sans affecter la performance de la tâche de recherche visuelle. Cela suggère qu'un système de gestion de NDD utile puisse être mis en œuvre en employant une approche de dégradation périphérique. Cette technique de dégradation périphérique exige la connaissance de la direction actuelle de regard fixe de l'utilisateur. Pour cela, il faut tracer le mouvement de la tête et/ou des yeux. Il est aussi possible d'assumer que l'utilisateur regardera vers le centre d'affichage et que les objets sont donc dégradés par rapport à leur déplacement de ce point [Con01].

- **Avantage :**

- Offre un outil efficace pour réduire le temps de calcul dans les systèmes de réalité virtuelle.

- **Inconvénient :**

- La sélection d'un NDD optimal pour chaque objet dans la scène est difficile. Pour cela, plusieurs travaux sont consacrés à développer un mécanisme formel pour choisir le NDD le plus approprié.



**Figure 3.9 : Les niveaux de détails dégradés en vision périphérique**

#### **3.2.5.1.5. Profondeur de champ (depth of field) :**

La représentation d'un objet peut être choisie en se basant sur la profondeur du centre du champ de vision de l'observateur. Les objets dans les frontières ou derrière ce champ sont non concentrés. Ils sont donc représentés avec moins de détails, tandis que ceux dans le centre du champ de vision sont plus détaillés. [Con 01]

#### **3.2.5.1.6. Vitesse (Velocity):**

Comme le précise Reddy dans [Red94], un objet en mouvement est moins bien perçu par le système visuel humain qu'un objet fixe, il peut donc être simplifié. Si un objet tourne rapidement sur lui-même, les détails de sa surface seront moins perceptibles. On pourra alors choisir une représentation moins détaillée. Un autre cas est le mouvement par rapport à l'observateur. Si un objet passe rapidement dans le champ de vision de l'utilisateur, celui-ci le verra légèrement flou. On pourra ainsi également utiliser une représentation moins détaillée [Kru97].

Différentes techniques peuvent être utilisées pour le choix du niveau de détail, la proportion de la vitesse de l'objet à la taille de cet objet, par exemple [Con01].

#### **3.2.5.1.7. Incidence**

Dans certain cas, un objet peut avoir une forme particulière selon l'angle de vue. Par exemple, une porte ou un tube, vu de profil, se réduisent à des représentations très simples [Kru 97].

### 3.2.5.1.8. Taux d'affichage d'une image fixe :

Pour avoir une bonne interactivité, il faut maintenir un taux élevé et cohérent d'affichage de l'image. Ce taux sera maintenu indépendamment de la complexité de la scène.

Cette technique inclut, habituellement un planificateur dont le travail est d'analyser le système de chargement et d'assigner l'évaluation des NDDs à chaque objet. Il existe différentes approches pour cette technique: les systèmes réactifs, les systèmes prédictifs, les systèmes préemptifs [Con01].

- Le système réactif ajuste simplement le détail en se basant sur le taux d'affichage de l'image précédente. Si l'affichage de la dernière image a été achevé après une certaine limite, les détails seront réduits, autrement ils peuvent être augmentés.
- Le système prédictif estime la complexité de l'image et choisi le NDD de telle sorte qu'il n'excède pas une certaine limite.
- Dans le système préemptif, les objets du modèle sont rendus dans l'ordre de priorité. L'utilisateur met le taux d'affichage désiré, et si le temps permis pour une image s'est écoulé, l'affichage de cette image sera arrêté et on commence par l'image suivante. Ainsi, si le taux d'affichage est choisi et l'ordinateur n'est pas assez puissant, donc moins de détails du modèle seront perçus durant la navigation

### 3.2.5.1.9. Tâche :

Une des caractéristiques importantes de toute représentation est qu'elle doive être utile à une tâche. Il est alors envisageable d'adapter la représentation à la tâche en cours. Par exemple, en utilisant des niveaux de détails dégradés pour les objets qui n'interviennent pas directement dans la tâche. La dégradation des objets repose donc sur l'importance associée à l'objet considéré, notamment dans le cas de scènes interactives et dynamiques où il ne faut pas dégrader un objet important à la compréhension d'une action, par exemple.

### 3.2.5.1.10. Densité :

Il semble naturel de penser que si un objet est présenté au sein d'une scène complexe, ses détails seront moins perceptibles car l'œil humaine devra d'abord filtrer la grande quantité d'informations qui l'entoure. Nous pouvons ainsi suggérer que si la

densité d'objets est élevée, alors chacun de ces objets pourra être représenté avec moins de détails. La validité de cette hypothèse doit toutefois être confirmée par des expérimentations ergonomiques [Kru99].

### 3.2.6. Transition entre Niveaux de Détails :

Le basculement entre deux NDDs ou deux résolutions d'un même objet génère deux problèmes principaux :

- **Problème de coût:** un des objectifs d'utiliser les niveaux de détails est le gain dans le temps de calcul, mais si le basculement entre les NDDs est trop fréquent, ce gain sera perdu en calcul nécessaire de basculement. Pour pallier ce problème, il faut gérer deux seuils : un par sens de basculement. Les deux seuils étant assez éloignés, les phénomènes d'hystérésis dus à des allers-retours entre différents NDDs disparaissent. Cette technique permet donc de borner la fréquence de basculement.
- **Problème de continuité:** si deux images successives utilisent deux représentations différentes d'un objet, il est possible d'échanger immédiatement les représentations entre les deux images. Toutefois, cela peut provoquer des sauts perceptibles par l'utilisateur si les deux représentations sont utilisées alternativement dans plusieurs images successives. Pour résoudre ce problème, deux techniques peuvent être utilisées :
  - La première consiste à gérer les deux représentations en mémoire et à cacher progressivement la première pour afficher progressivement la deuxième en faisant varier la transparence des objets. Mais ceci est coûteux, puisque deux représentations du même objet sont présentes en mémoire et sont calculées à un même moment.
  - La seconde technique dite des géomorphes consiste à contraindre la construction des NDDs pour permettre une interpolation purement géométrique entre deux NDDs successifs.

### 3.2.7. Bilan :

A ce jour, il n'existe pas une méthode de simplification polygonale qui soit capable de simplifier tous les objets, quelque soit leurs caractéristiques. Par exemple, il existe une limitation importante concerne la définition du taux d'approximation désiré. Certains systèmes définissent un taux d'erreur qu'il est parfois difficile d'estimer.

D'autres, plus simple, permettent de choisir le nombre de polygones (ou de sommets) dans l'objet simplifié

Par contre, les algorithmes de simplification orienté scène sont assez primaires et posent certains problèmes .Par exemple, que faire si un objet se déplace dans une scène? De plus, ils ont tendance à produire des résultats assez peu satisfaisants du point de vue visuel, car ils peuvent produire des discontinuités entre les polygones et faire apparaître des trous, ce qui restreint leur usage.

Mais, ils ont des avantages : ils s'intègrent bien dans les algorithmes de gestion de scène au sens large, comme l'élimination des objets non visibles (object culling).De plus, ils permettent de faire de la gestion de LODs progressive, en dégradant les détails sur les parties éloignées d'un objet [Kru 97].

### **3.2.8. Conclusion**

Après la définition du principe de la multirésolution et les différentes techniques pour la génération des niveaux de détails et les critères de sélection de LODs ainsi que la présentation des problèmes générés par le basculement entre ces niveaux et les solutions proposées pour résoudre ces problèmes, nous consacrons la deuxième partie de ce chapitre à la présentation de l'utilisation de la technique de niveaux de détails dans la simulation comportementale et les différents travaux exploitant cette technique dans la simulation du comportement des humains virtuels .

## **4. Intégration de la technique de niveaux de détails dans la simulation comportementale**

Un grand monde virtuel habité par des humains virtuels nécessite beaucoup de ressources informatiques. La simulation devrait exécuter un nombre très élevé d'événements en même temps ou dans un temps très proche, afin de réaliser une simulation crédible, rapide et en temps réel, quelque soit la taille du monde simulé ; et dans les limites des ressources informatiques utilisées.

Dans des jeux d'ordinateur, le problème est habituellement résolu en ne simulant pas les régions du monde qui ne sont pas observées par l'utilisateur (joueur). Cependant, ceci rend souvent l'environnement anormal ; nous pouvons voir quelques contradictions (par exemple, des monstres réapparaissent quand nous revenons à un certain endroit pour la deuxième fois).

En revanche, notre but était de maintenir la simulation aussi crédible que possible. En combinaison avec la condition d'utiliser un PC simple même pour des mondes virtuels très complexes, nous devons penser à une meilleure manière pour réduire la taille de la simulation. La technique de niveau de détail au niveau comportemental (LOD IA) est utilisée.

La technique de niveau de détail (LOD) est utilisée couramment dans les infographies pour la même raison – la réduction du coût informatique-. Les jeux d'ordinateur contiennent beaucoup d'objets mobiles qui consomment beaucoup de ressources informatiques.

L'idée de réduire la quantité d'instructions graphiques est basée sur le fait qu'il n'est pas important de rendre chaque aspect de la représentation visuelle du monde aux endroits où l'utilisateur ne peut pas les voir correctement (par exemple, ils sont trop loin). Ces objets sont rendus selon leur position par rapport à l'œil de l'observateur.

L'utilisation de la technique de niveau de détail LOD au niveau comportemental consiste à effectuer un transfert du domaine des infographies des ordinateurs vers le domaine de l'intelligence artificielle. Cette dernière est basée sur une simple idée qui consiste en ce qui suit:

Il existe souvent peu de places, dans le monde artificiel à un moment donné de la simulation, où le comportement des humains virtuels est important, celles de moindre importance n'ont pas besoin de simuler le comportement des humains virtuels avec précision, leurs simulation sera dégradée.

Si le comportement des humains virtuels est simulé partiellement, les demandes de la simulation en ressources informatiques peuvent être réduites significativement [Kkp+06].

#### **4.1. Les travaux relatifs**

Une idée très robuste pour l'emploi de la technique des LOD au niveau comportemental, en utilisant les machines à états définis hiérarchiques est présentée dans [Cha03], mais elle représente seulement un modèle qui ne semble pas avoir été encore exploré.

Sullivan et al. ont utilisé la technique de LOD pour le comportement conversationnel, à l'aide de l'outil de génération de comportement « BEAT » en utilisant la technique de passage de rôle [Scv+02]. Les règles et les rôles peuvent être vus comme

des morceaux d'un programme posé au-dessus d'un humain virtuel de base. Si l'humain virtuel n'est pas visible, le rôle ne lui est pas passé, et par conséquent seulement le comportement de base est effectué. Contrairement à l'approche proposée, cette technique simplifie seulement le comportement conversationnel.



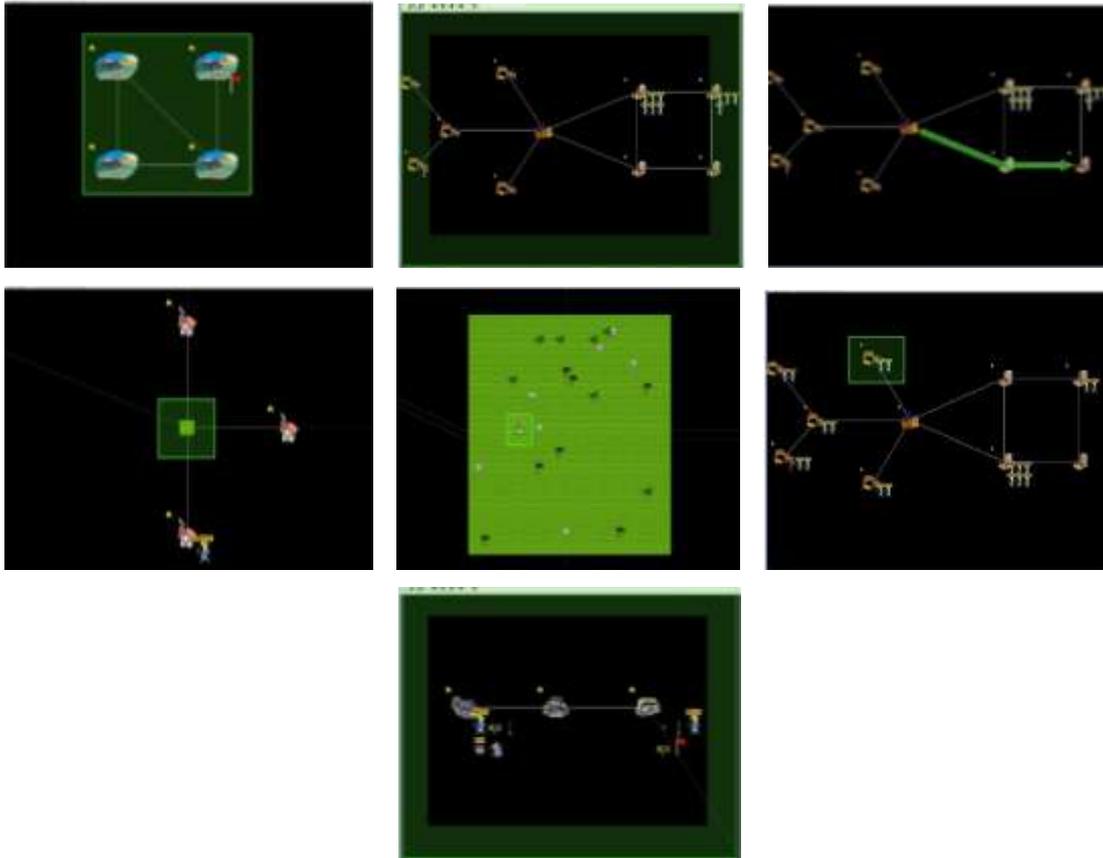
**Figure3.10 : Une conversation normale dans un groupe**

Une approche plus robuste pour la planification du temps machine pour chaque humain virtuel est présentée dans [WM00]. Cependant, comme les scripts comportementaux non planifiés ne sont pas exécutés, cette approche peut être insérée dans le domaine de la simulation « on/off ».

Un autre projet (IVE<sup>1</sup>: Intelligent Virtual Environnement) propose la simulation de grands mondes virtuels extensibles habités par des humains virtuels, en utilisant la technique de niveau de détail au niveau comportemental. Ce projet réduit la qualité de la simulation, en employant une simulation précise au niveau des localisations les plus importantes et une simulation approximative au niveau des localisations les moins importantes, en prenant en compte les ressources informatiques disponibles. Ce projet simplifie la simulation complète<sup>2</sup> [Kkp+06].

<sup>1</sup> Le code du projet, un scénario de test et la documentation peuvent être téléchargés à partir du site <http://mff.modry.cz/ive>.

<sup>2</sup> Il simplifie le comportement et la topologie du monde artificiel.



**Figure 3.11 : Un scénario du projet IVÉ**

Le tableau 3.1 montre une vue d'ensemble des travaux précédents utilisant la technique LOD IA, caractérisée par les critères suivants : le critère de détermination de LOD, le nombre de niveaux de détails, l'usage de LOD et les comportements simulés par l'IA.

<b>Auteurs</b>	<b>Le critère de détermination de LOD</b>	<b>Le nombre de LOD</b>	<b>L'usage de LOD</b>	<b>Les comportements simulés par IA</b>
Chenney et al. [Caf01]	Le potentiel de la visibilité	02	Mettre à jour le mouvement	La navigation, l'évitement de collision
O'Sullivan et al [Scv+02]	La distance	Non spécifié	La géométrie, les animations, l'évitement de collision, les gestes, les expressions faciales et la sélection d'action	La navigation, l'évitement de collision, le dialogue complexe entre les agents

Brockington [Bro02]	Distance	05	La recherche, la navigation, la sélection d'action dans un combat	La navigation, l'évitement de collision, les interactions de combat complexes
Niederberger and Gross [NG05]	La distance et la visibilité	21	La planification, l'évitement de collision, la recherche du chemin et les décisions de groupe	La navigation, l'évitement de collision et la planification du chemin
Pettré et al [Pcm+06]	La distance et la visibilité	05	La géométrie, la mise à jour de mouvement, l'évitement de collision, la navigation	La navigation, l'évitement de collision, la recherche de chemin
Brom et al. [Bsp07]	La distance simplifiée	04	La sélection d'action (avec l'arbre AND-OR), la simplification environnementale	La navigation, les interactions complexes avec les objets et les autres agents
Paris et al. [Pgo09]	La distance	03	La navigation et l'évitement de collision	La recherche de chemin, la navigation et l'évitement de collision
Lin et Pan [LP07]	distance	Non spécifié	La géométrie et les animations	La locomotion
Osborne and Dickinson [OD10]	La distance	Non spécifiée	La navigation, flockage, les décisions de groupe	La navigation
M.Wibner et al [Wka10]	La distance et la visibilité	08	La mise à jour du mouvement, l'évitement de collision, la navigation et l'exécution de l'action	La navigation, l'évitement de collision, les interactions basées sur les désires avec les agents et les objets intelligents, les dialogues

Tableau 3.1 : Comparaison des différentes approches de LOD IA

## **4.2. Conclusion:**

Ce chapitre est divisé en deux axes principaux, la multirésolution et l'utilisation de la technique de niveau de détail dans la simulation comportementale.

Le chapitre suivant comportera une description détaillée de notre modèle proposé et une présentation des différents outils utilisés pour l'implémentation de notre modèle.