



Imagerie Numérique

M4105Cin – Moteurs 3D avancés

1. Niveaux de détails

DUT INFO 2ème année 2018-2019

Sébastien THON

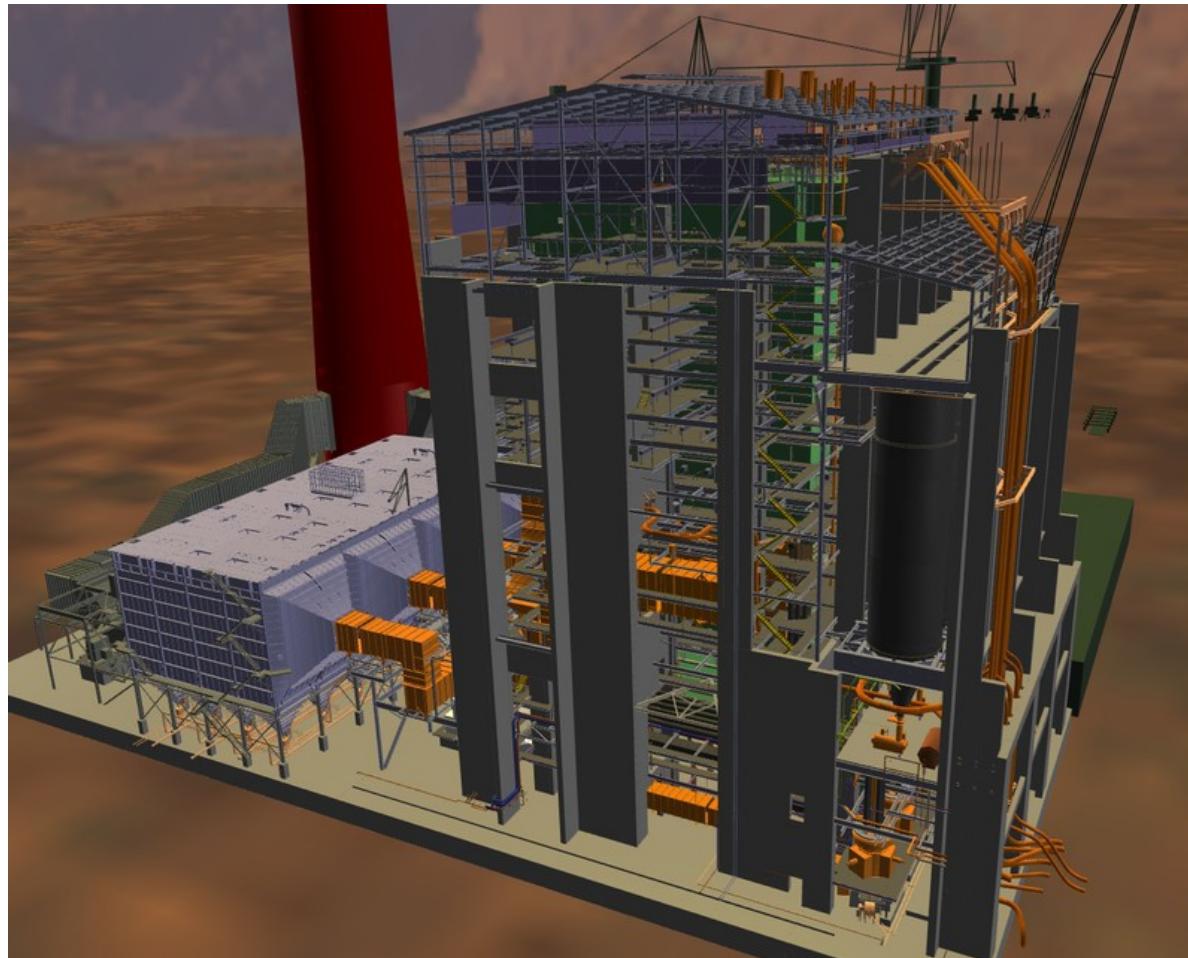
IUT d'Aix-Marseille Université, site d'Arles
Département Informatique

Introduction

De nombreux domaines utilisent des objets 3D très volumineux, comportant de plus en plus de polygones :

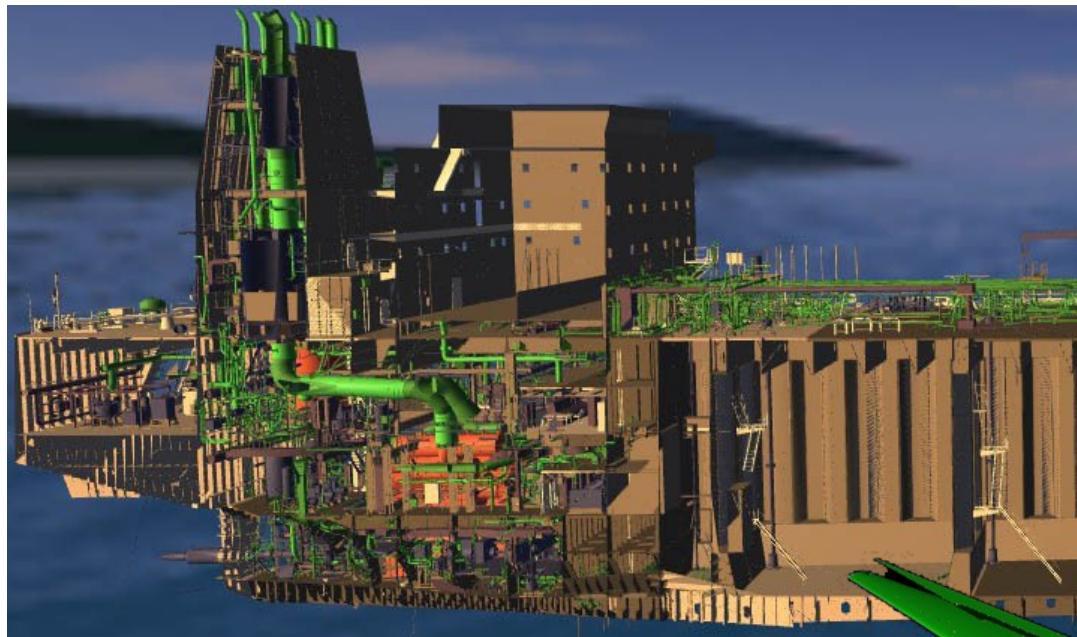
- Architecture
- Conception Assistée par Ordinateur
- Objets scannés
- Géomatique
- Visualisation scientifique
- Visualisation médicale
- Jeux vidéo
- Cinéma
- Etc.

Architecture

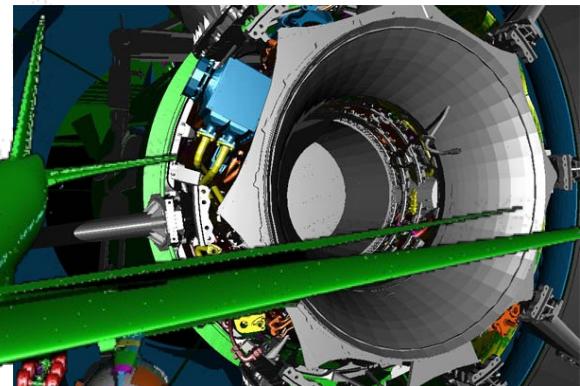
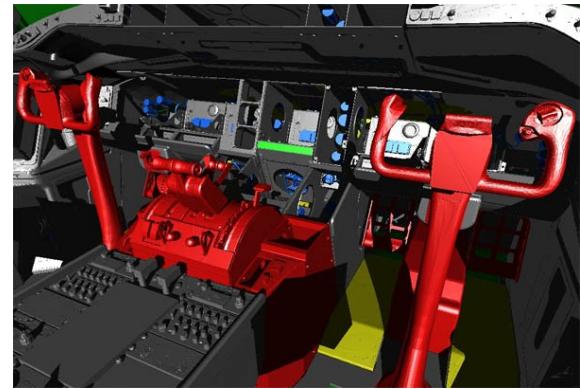


12 millions de triangles

Conception Assistée par Ordinateur (CAO)

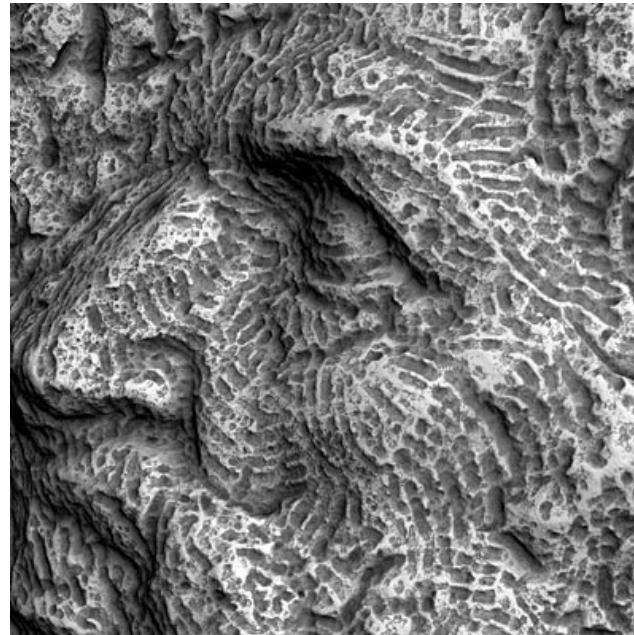


82 millions de triangles



*350 millions de triangles
(source: Boeing)*

Objets scannés



386 millions de triangles

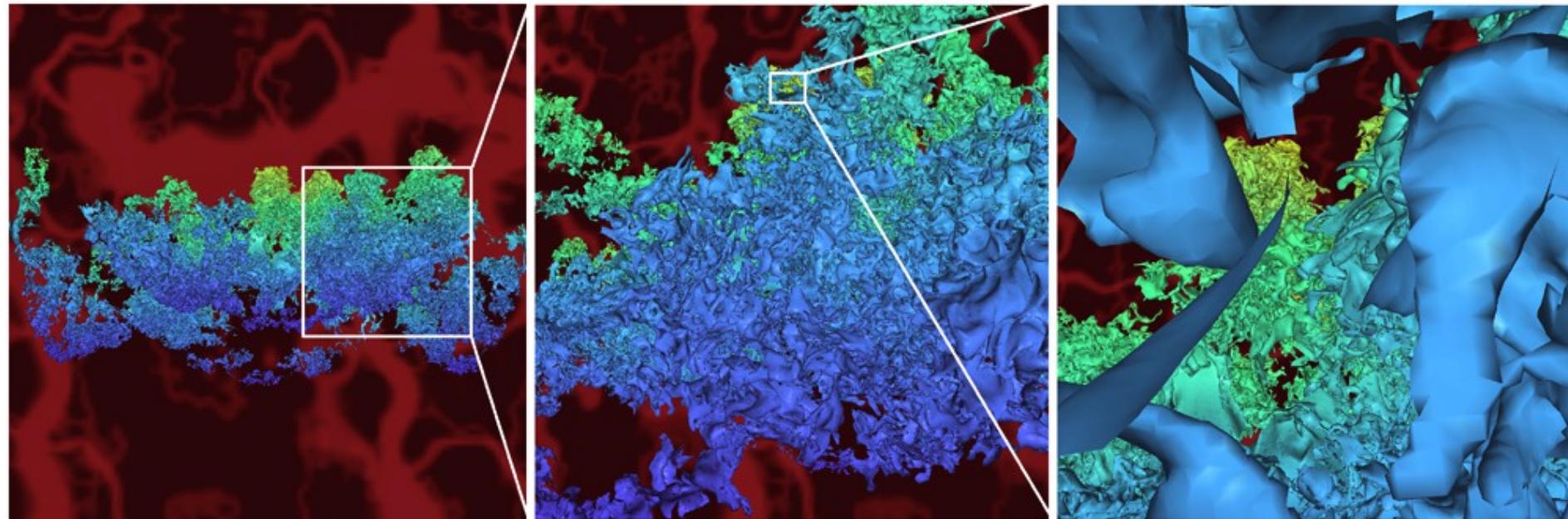
The Digital Michelangelo Project

<http://graphics.stanford.edu/projects/mich/>

En fonction de la résolution souhaitée et des dimensions de l'objet à scanner, un scanner laser peut produire un nuage de plusieurs millions de points, et donc un maillage triangulaire de plusieurs millions de triangles.



Visualisation scientifique



Visualisation 3D (environ 100 millions de triangles) de l'isosurface d'une turbulence simulée en mécanique des fluides.

Les simulations réalisées dans de nombreux domaines scientifiques (mécanique des fluides, météorologie, etc.) produisent des données que l'on cherche à visualiser en 3D. Pour obtenir des résultats précis, des échantillons très fins sont utilisés, ce qui peut produire des surfaces composées de millions de triangles.

Jeux vidéo

Les jeux vidéo proposent des scènes de plus en plus complexes et détaillées, avec des décors, des objets et des personnages comportant de plus en plus de polygones.



Half Life (1998)
~ 600 triangles

Half Life 2 (2004)
~ 6000 triangles

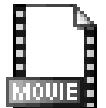


Scènes naturelles



Scène composée de 5 arbres (plusieurs millions de triangles chacun) et de 28000 tournesols (35000 triangles chacun), soit plus d'1 milliard de triangles!

Pour représenter de manière réaliste la complexité de la géométrie de scènes naturelles, il faudrait des milliards de triangles.



Problème :

Impossible d'afficher en **temps réel** des scènes 3D trop détaillées, comportant trop de polygones. Ce nombre de polygones dépend de la puissance du matériel (CPU, carte graphique).

Première solution :

Ne pas demander le rendu des objets qui ne sont pas visibles, pour réduire les calculs effectués par le pipeline d'OpenGL.

Plusieurs méthodes possibles :

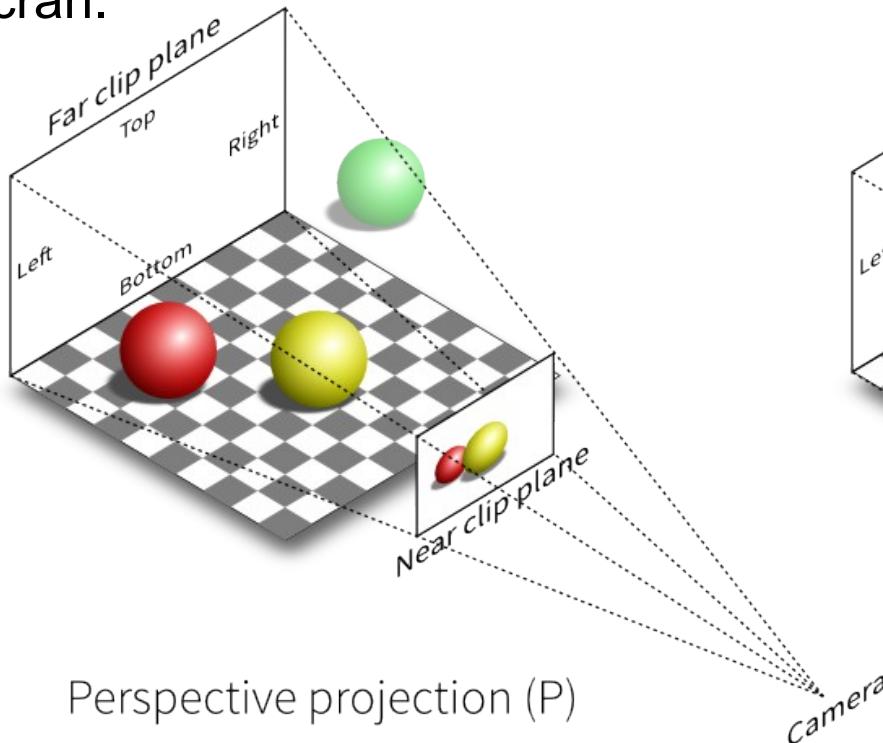
1) Structuration hiérarchique de la scène 3D

Les mondes virtuels sont souvent composés d'objets structurés (ex: un immeuble composé d'appartements).

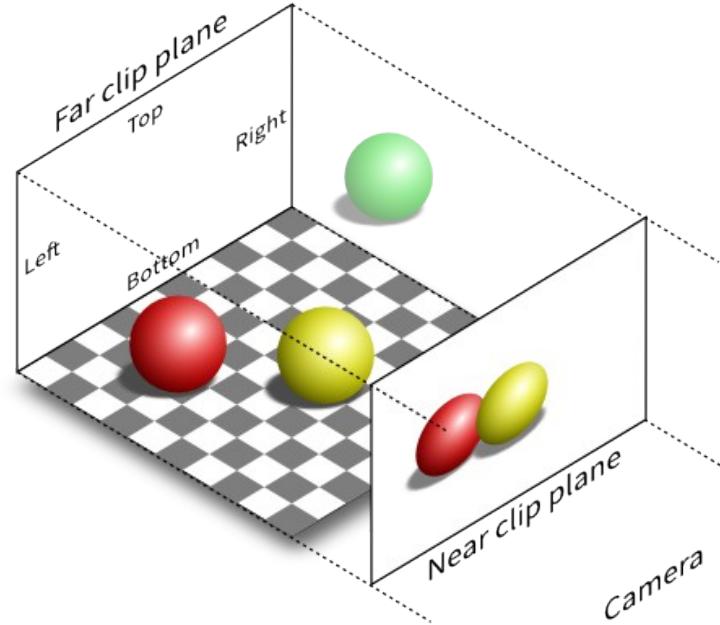
- Utiliser cette structure pour ne considérer à un instant donné que la zone où se trouve l'utilisateur.
- N'afficher que les objets 3D se trouvant dans cette zone
- Réduit les temps de calcul

2) Champ de vision (« *view frustum culling* »)

Ne pas demander l'affichage des objets qui ne se trouvent pas dans la pyramide de vision (« *View frustum* »), car ils ne seront pas affichés à l'écran.



Perspective projection (P)



Orthographic projection (O)

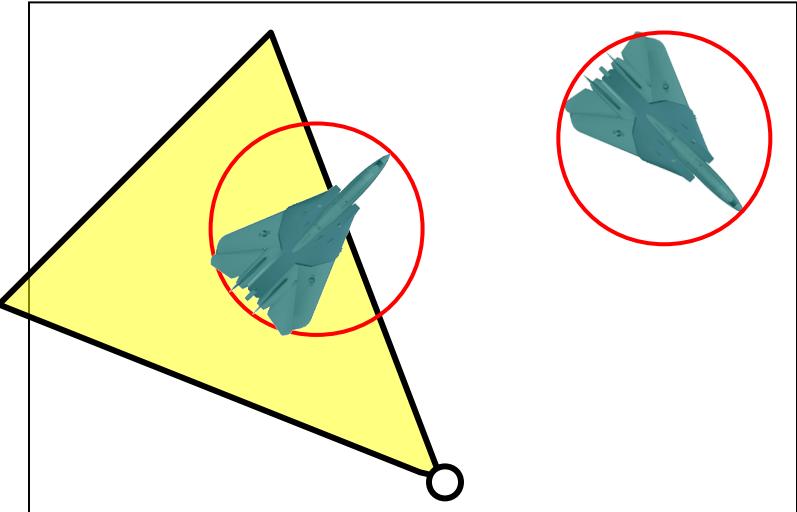
```
gluPerspective( 45.0f, (float)width/height, 1.0f, 1000.0f );
```

near

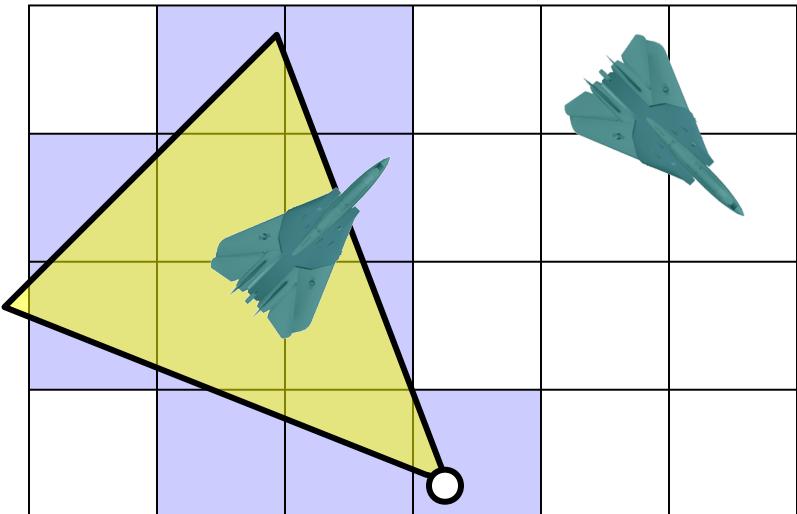
far

Algorithme naïf : tester l'intersection de chaque triangle de chaque objet avec la pyramide de vue → trop lent.

→ Accélérer en testant l'intersection de la pyramide de vue avec un **volume englobant de l'objet** (sphère, boîte, enveloppe convexe, etc.)



Peut être combiné avec une partition de l'espace, uniforme ou adaptative (quadtree, octree, kD-tree, BSP tree, etc.)



On peut accélérer l'affichage d'une scène 3D en n'affichant pas tout :

View-frustum culling

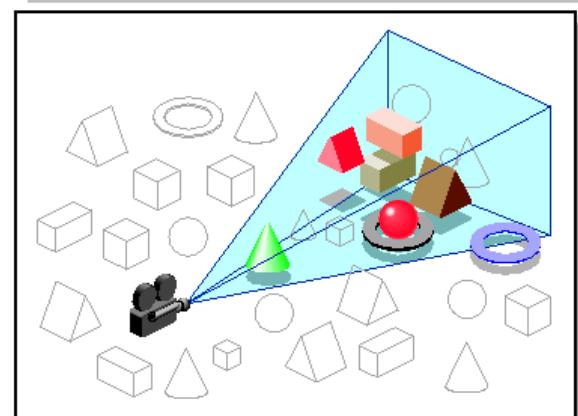
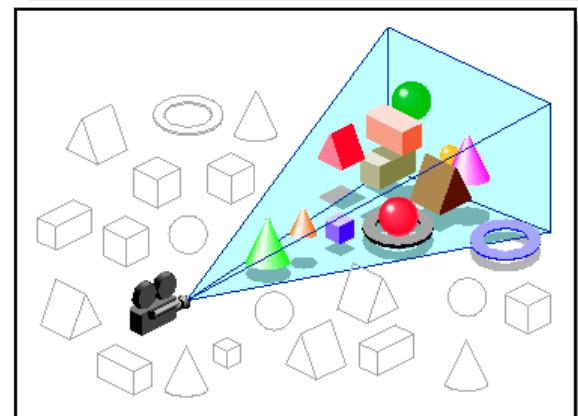
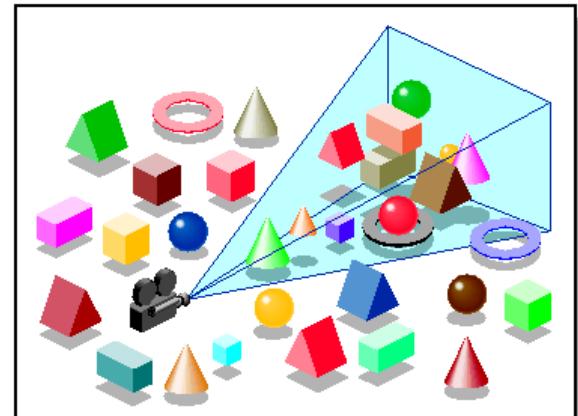
On ne demande l'affichage que des objets qui se trouvent dans la pyramide de vue (view frustum), car seuls eux seront affichés à l'écran.

Occlusion culling

Identifie les polygones de la scène qui seront masqués par d'autres objets plus proches de la caméra, et n'en demande pas le rendu.

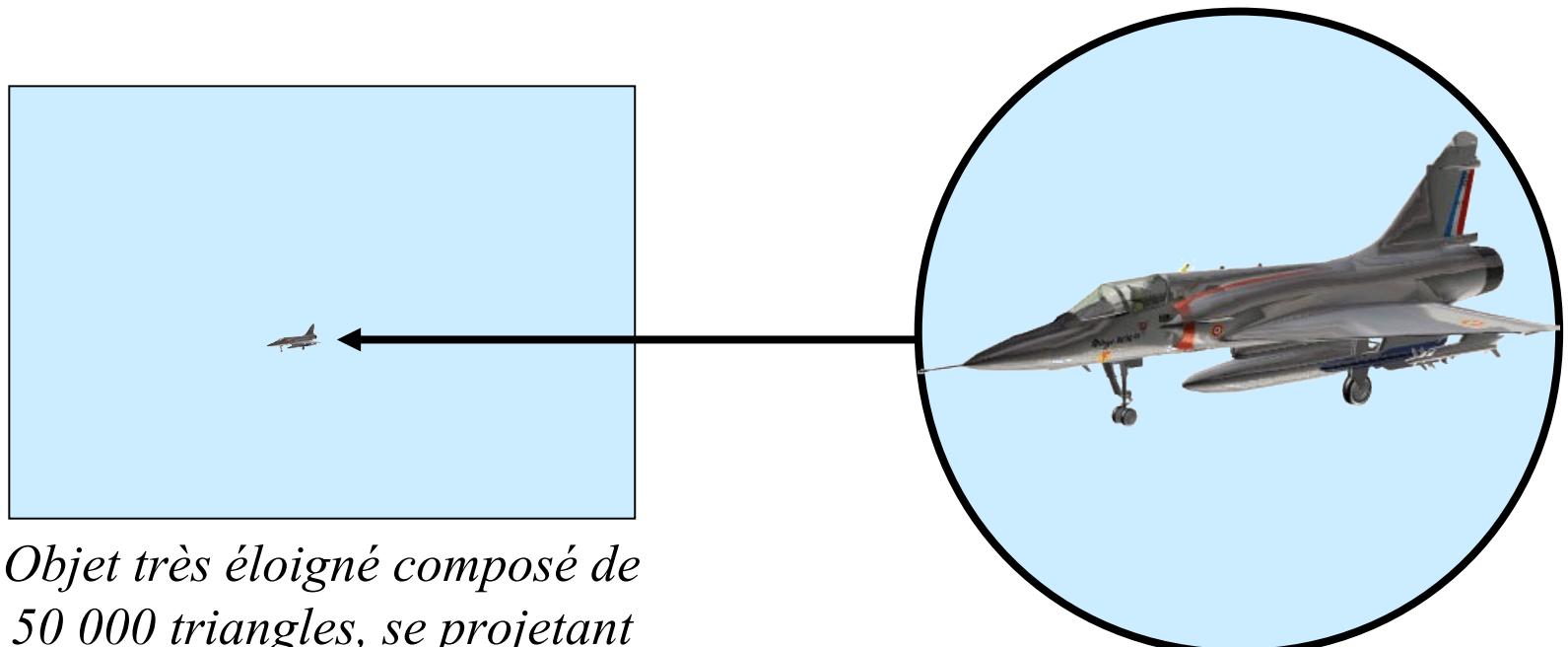
Small feature culling

N'affiche pas les objets qui se projettent sur moins d'un pixel à l'écran.



Mais après avoir enlevé tous les objets ne se trouvant pas dans le champ de vision, il peut encore en rester un grand nombre à afficher.

Ex: il peut rester un objet lointain comportant un nombre très important de polygones.

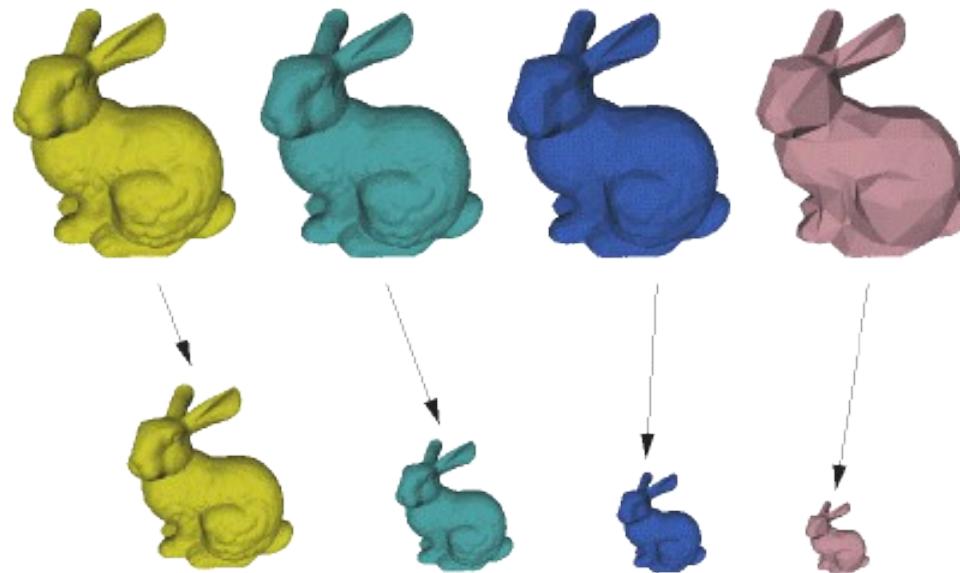


Objet très éloigné composé de 50 000 triangles, se projetant sur quelques pixels à l'écran

Deuxième solution :

On ne remarque les détails d'un objet que si il est proche. Si il est éloigné, on n'en voit que la structure grossière.

→ substituer à l'objet des représentations plus ou moins détaillées, comportant plus ou moins de polygones, en fonction de la distance à laquelle il est vu.



→ Utilisation de **niveaux de détail**
(**LOD** : « *Levels Of Detail* »)

Techniques de niveaux de détail utilisées :

- Simplification de maillage
- Simplification structurelle d'objets
- Simplification de scène

1. Simplification de maillage

Buts :

- Simplifier la représentation géométrique des objets en réduisant le nombre de polygones.
- Conserver la forme générale.

Diverses approches :

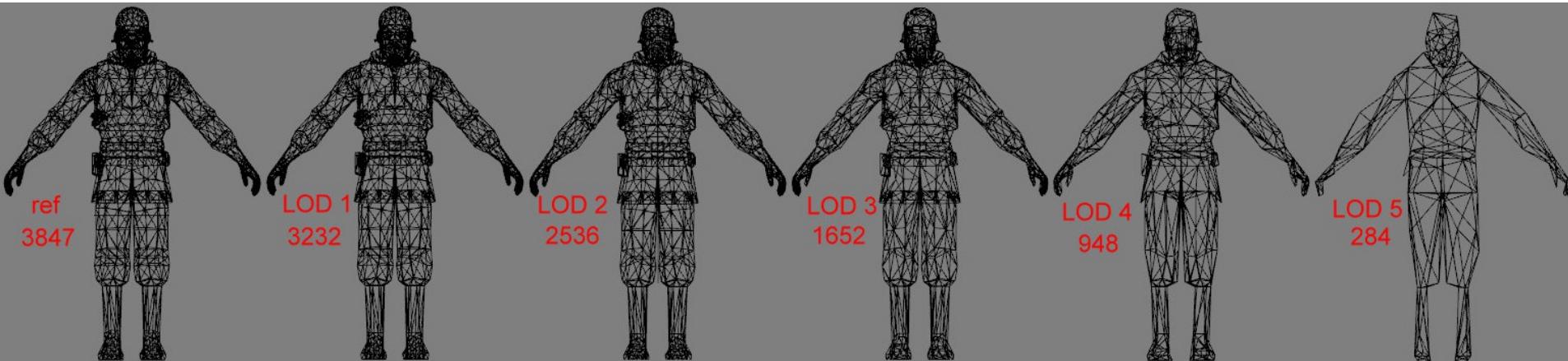
- Simplification à niveaux de détails statiques.
- Simplification à niveaux de détails continus.

1.1 Simplification à niveaux de détails statiques

Construire à l'avance différents niveaux de représentation d'un objet 3D, plus ou moins détaillés (manuellement ou avec un algorithme).

Lors du rendu, choisir le bon modèle à afficher en fonction de la distance.

Technique très souvent utilisée dans les jeux, permet aussi de s'adapter à la puissance de la machine.





Mesh Reduction



Original (100%): 8000 polys

LOD #1 (50%): 4000 polys

LOD #2 (25%): 2000 polys

Réduction du nombre de polygones

Texture Rebaking



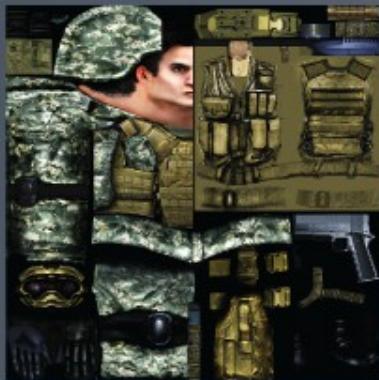
Original (100%): 8000 polys



LOD #1 (50%): 4000 polys



LOD #2 (25%): 2000 polys



Original: 1024x1024 map



LOD #1 (50%): 512x512 map



LOD #2 (12.5%): 128x128 map

Réduction de la taille des textures de chaque niveau de détail de personnage.

Avantages

- Rapidité de sélection d'un niveau de détail.
- Permet le rendu de scènes très complexes en ajustant le nombre de polygones à afficher.

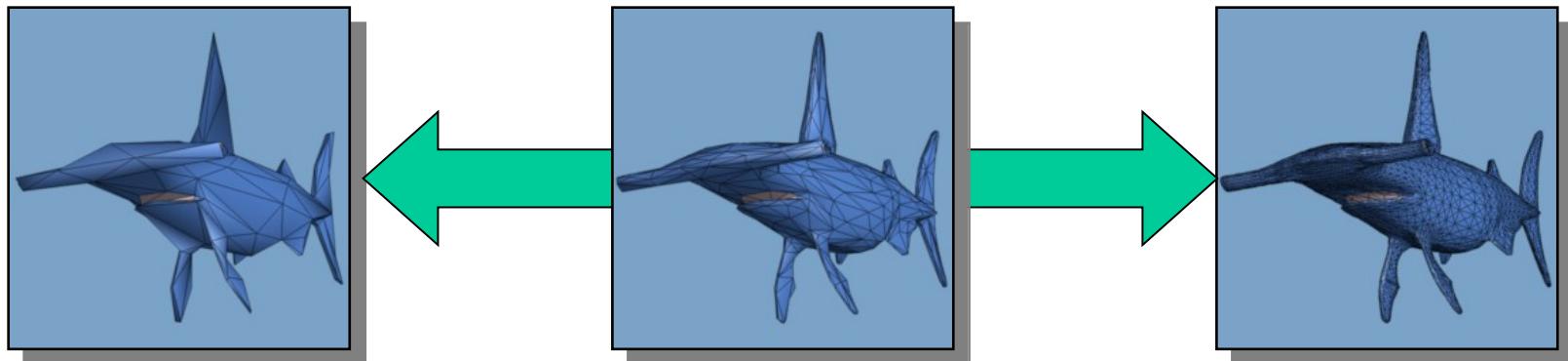
Inconvénients

- Problèmes de transition (« *popping* » : effet de saut).
- Stockage en mémoire de plusieurs objets.

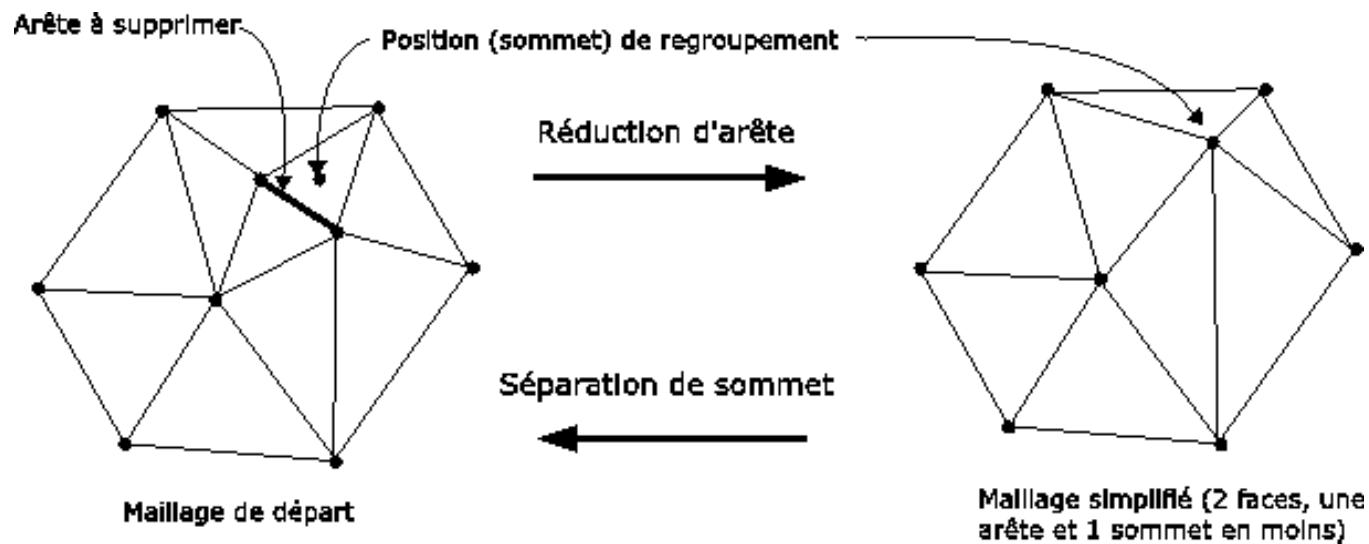
1.2 Simplification à niveaux de détails continus

Génération séparée de maillages → problème de transition brutale entre les différents niveaux de détails.

Solution : structure générant au vol un maillage adapté au niveau de détail que l'on souhaite → structure à niveaux de détails continus (CLOD, « *Continuous Level Of Detail* »).

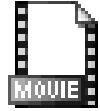
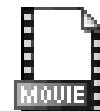


Il existe plusieurs méthodes, basées sur le principe de réduction des arêtes (\rightarrow élimination de sommets) pour simplifier le maillage. Elles diffèrent sur le choix des arêtes à réduire, ou le moment de le faire.



La méthode de maillage progressif (« *Progressive mesh* ») de Hugues Hoppe est implémentée dans DirectX.

Lire : <http://research.microsoft.com/~hoppe/pm.ppt>

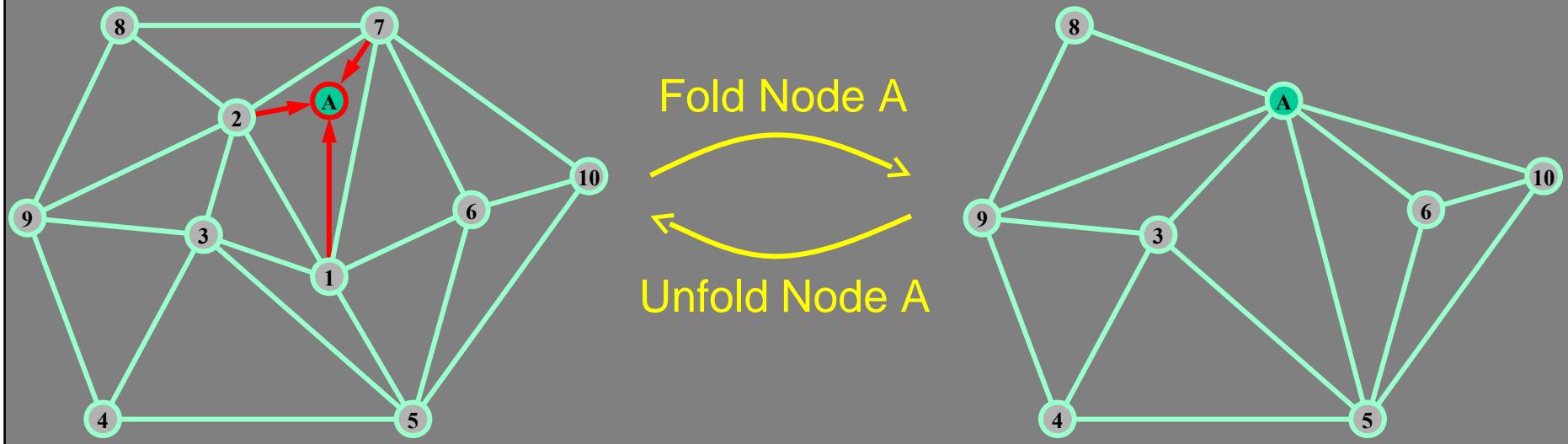


Opérateurs de simplification

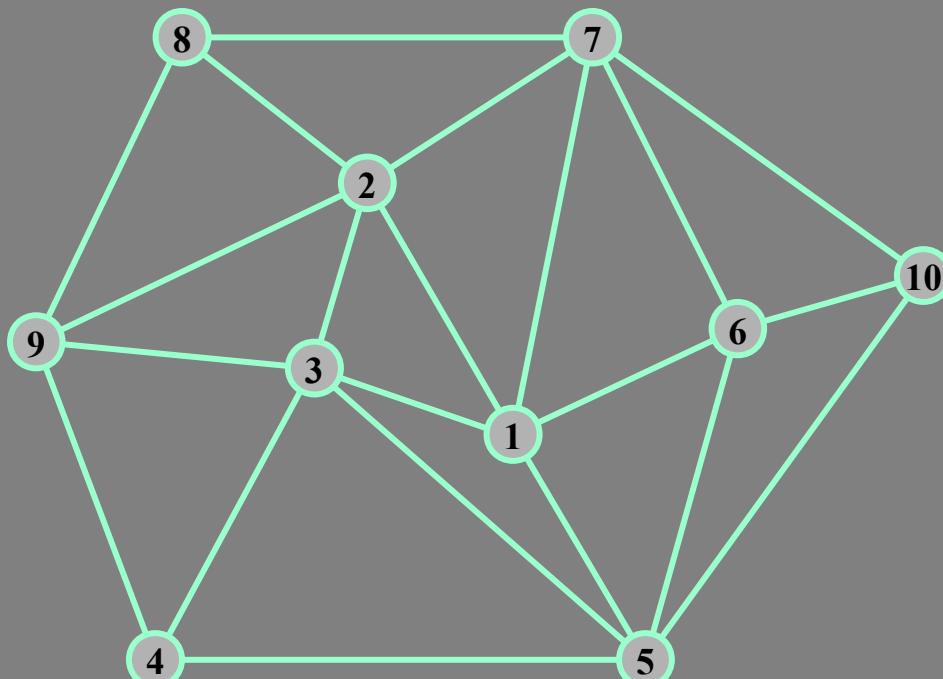
- **Normalisation** : élimine les points et les arêtes définis plusieurs fois.
- **Simplification des sommets** : tous les points à l'intérieur d'un volume (cube, sphère) centré autour d'un point sont regroupés → Les amas de points et les petites facettes sont combinés.
- Suppression de facettes dont la surface est trop petite.
- ...

Structure de données possible : arbre de sommets

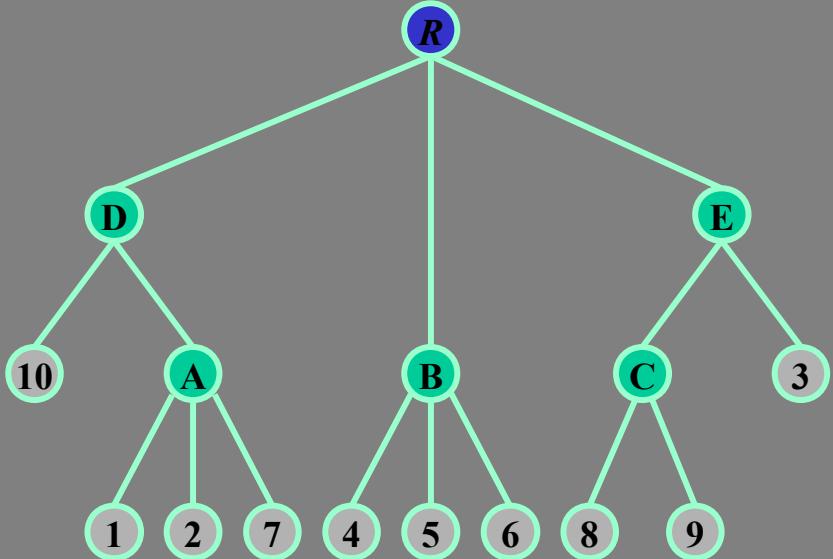
Permet de réduire l'objet ou au contraire de l'étendre.



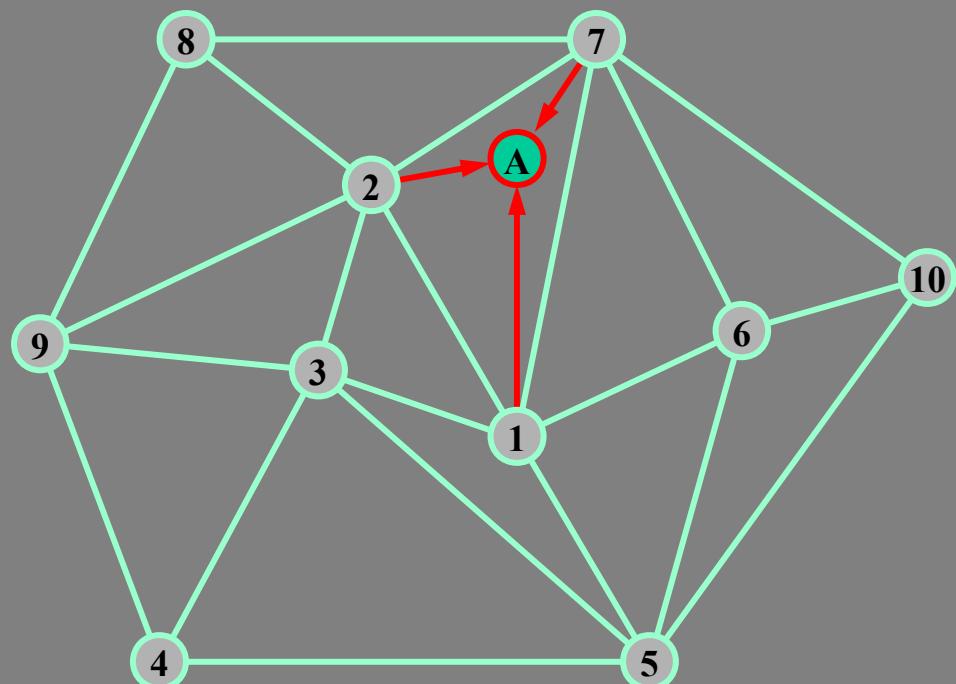
On code dans un arbre les points qui peuvent être contractés en un seul.



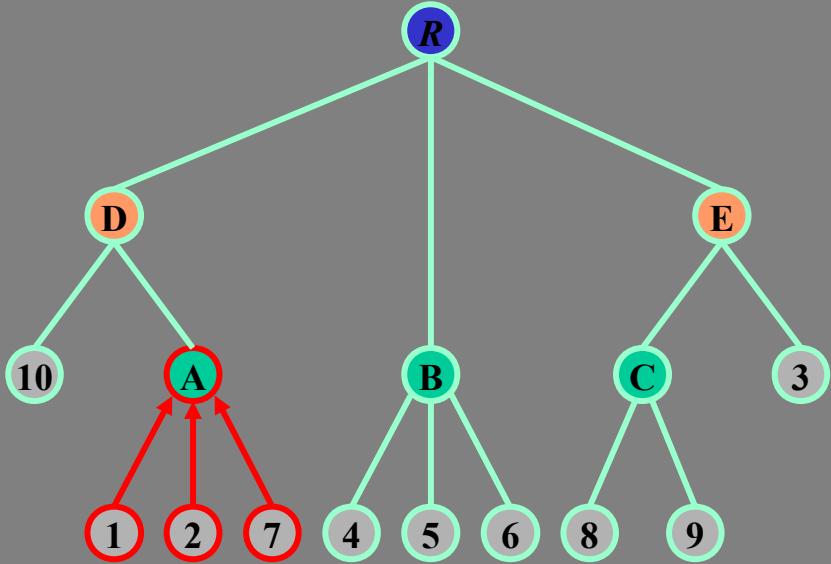
Triangles dans la liste active



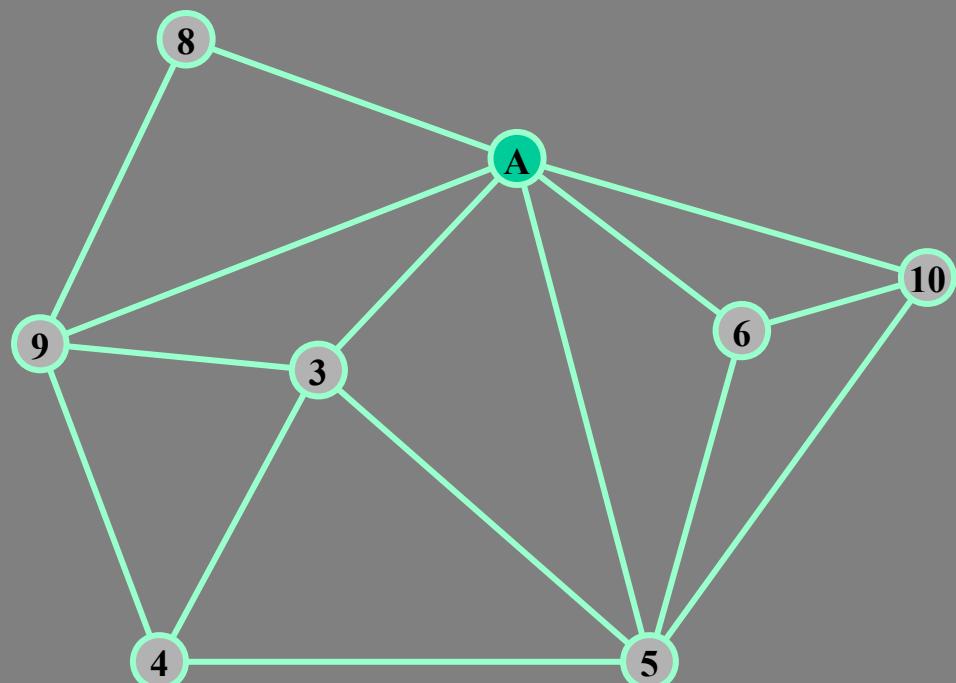
Arbre de sommets



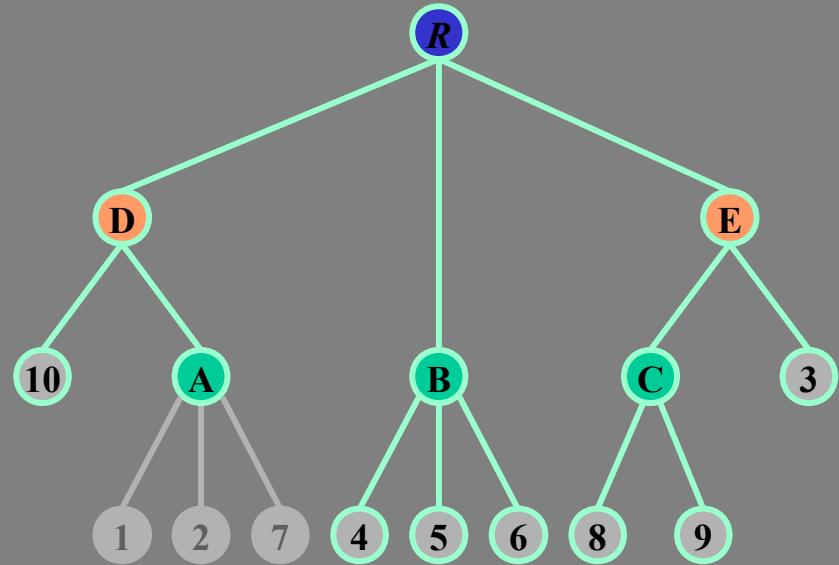
Triangles dans la liste active



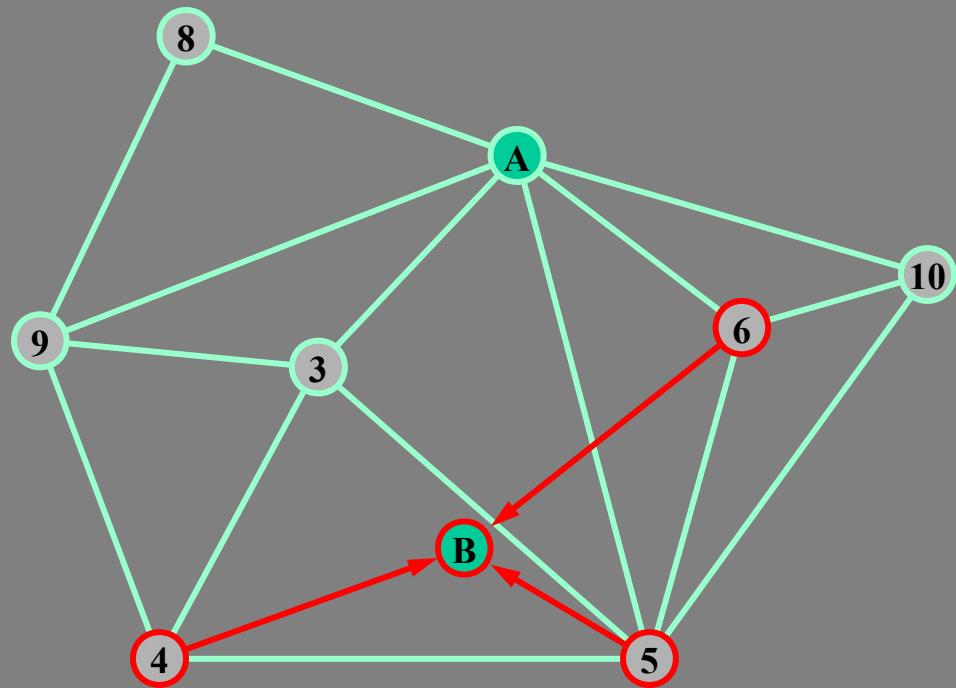
Arbre de sommets



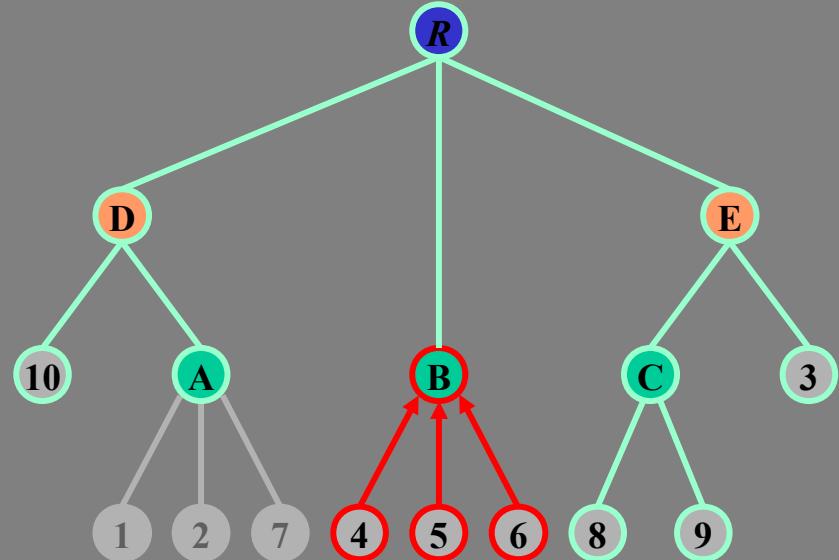
Triangles dans la liste active



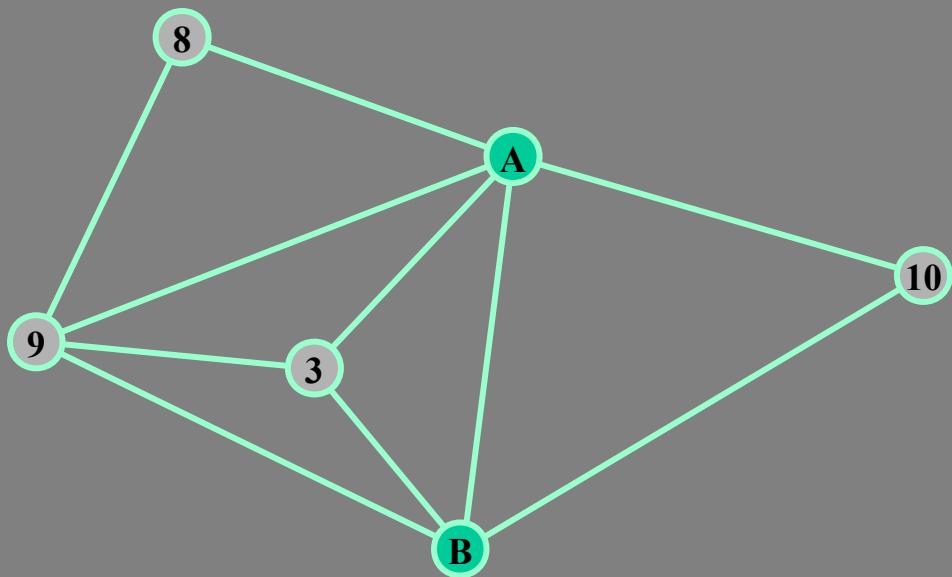
Arbre de sommets



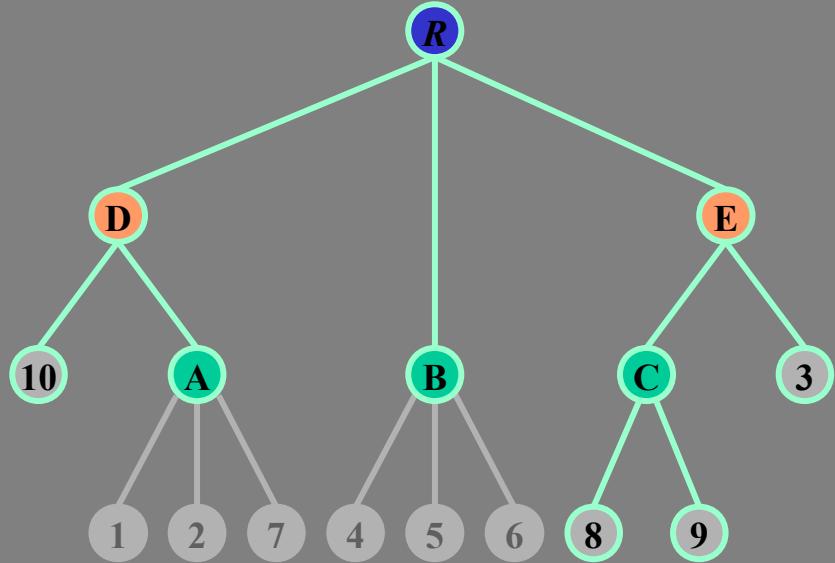
Triangles dans la liste active



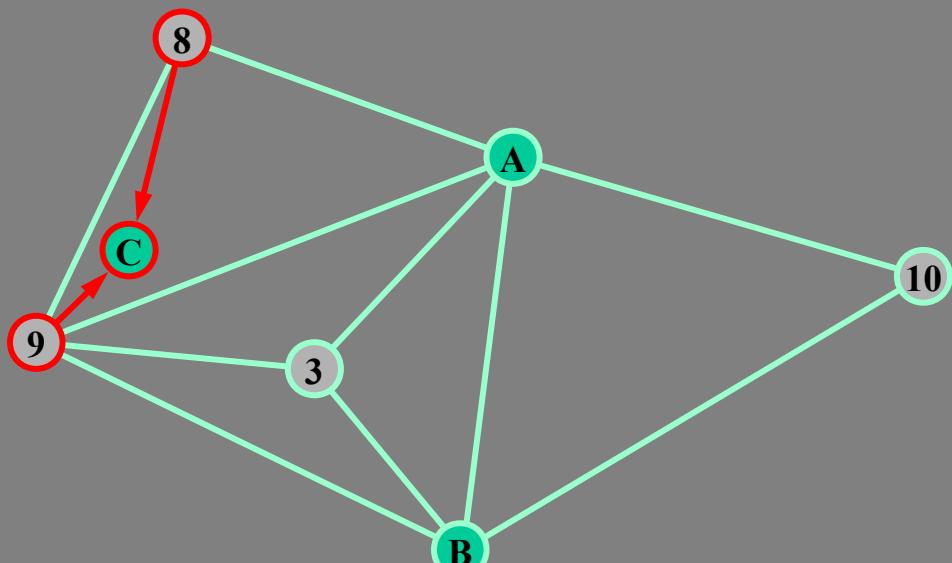
Arbre de sommets



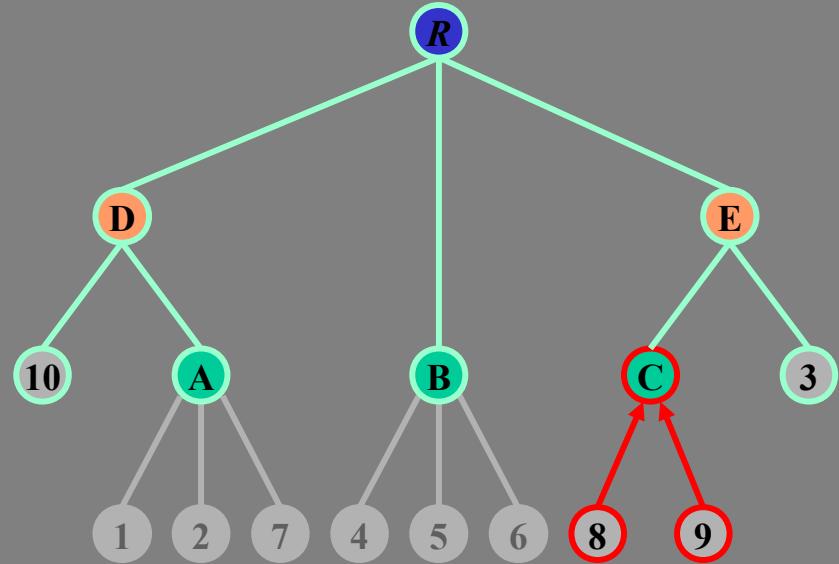
Triangles dans la liste active



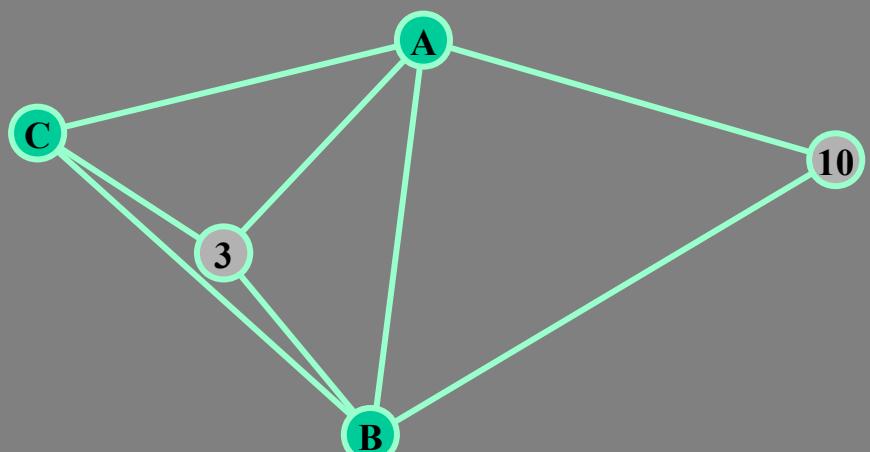
Arbre de sommets



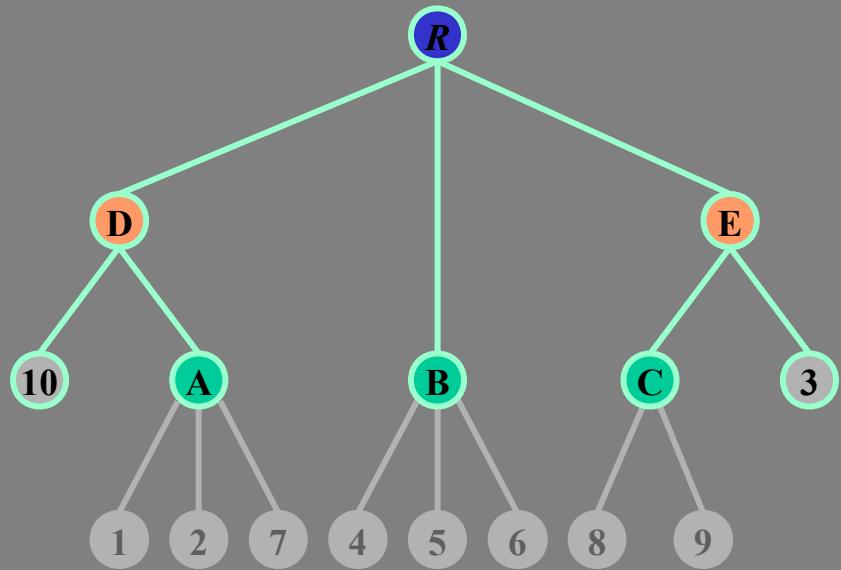
Triangles dans la liste active



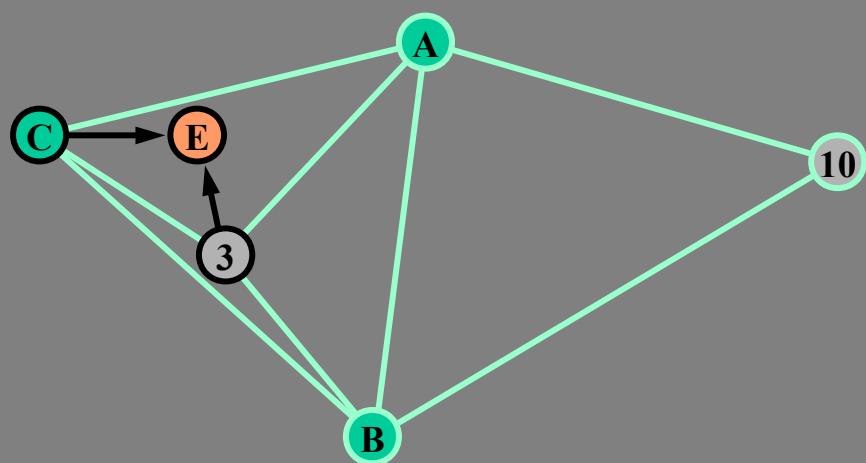
Arbre de sommets



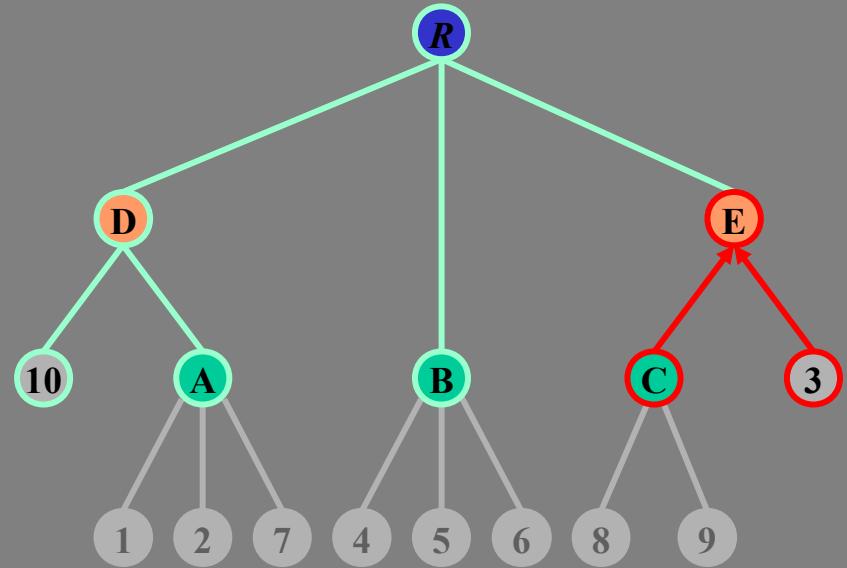
Triangles dans la liste active



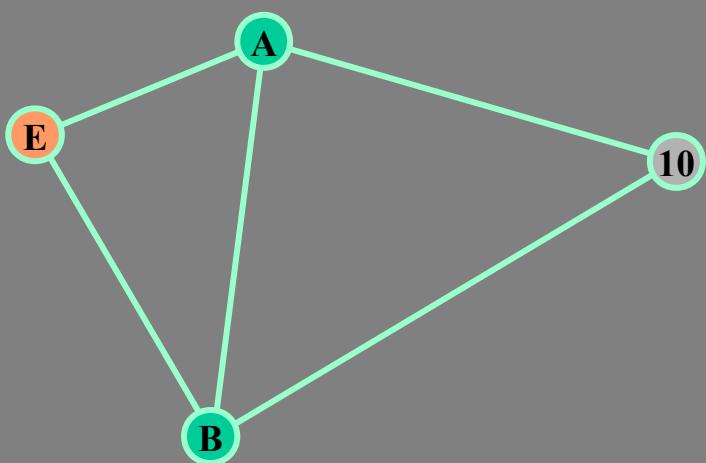
Arbre de sommets



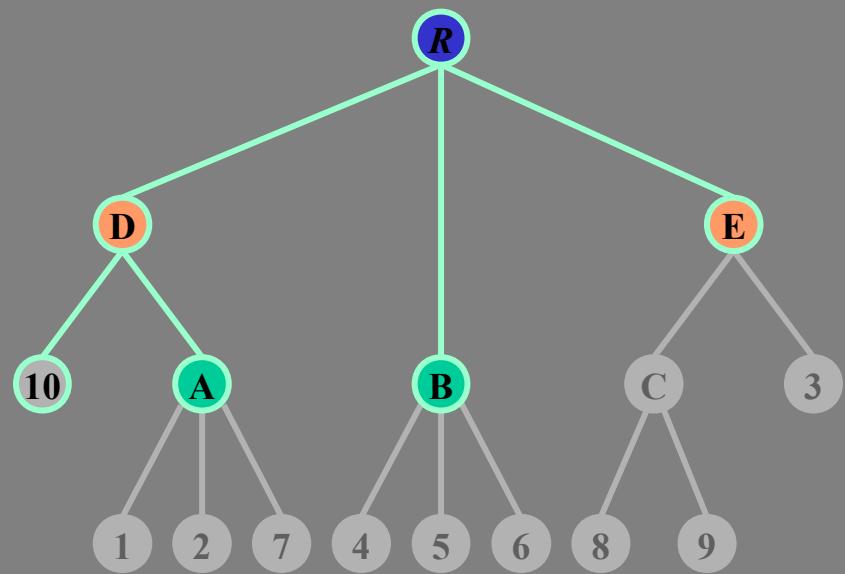
Triangles dans la liste active



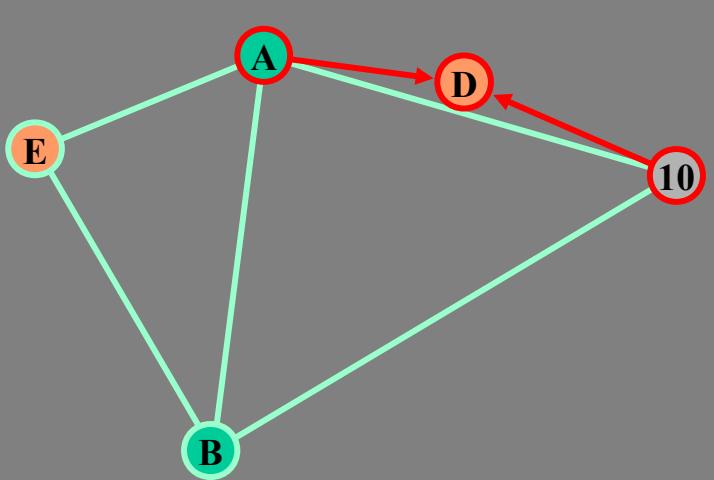
Arbre de sommets



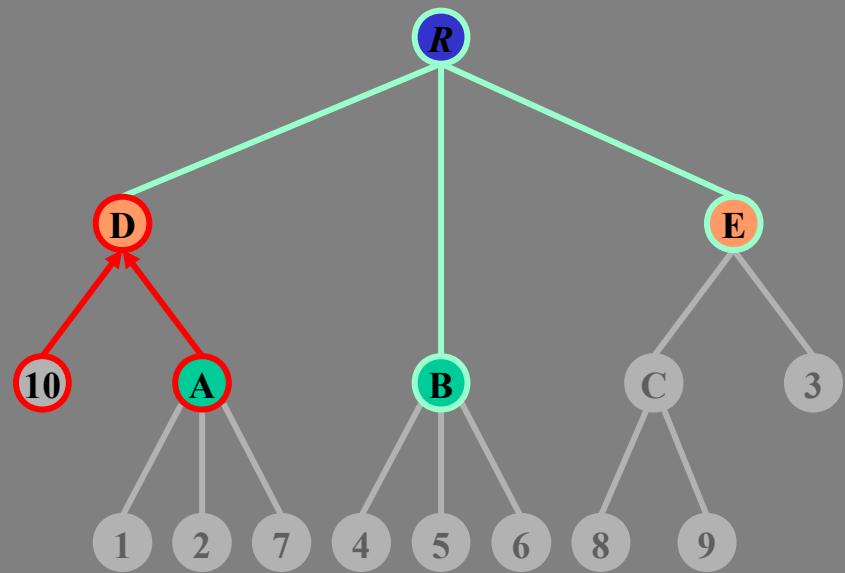
Triangles dans la liste active



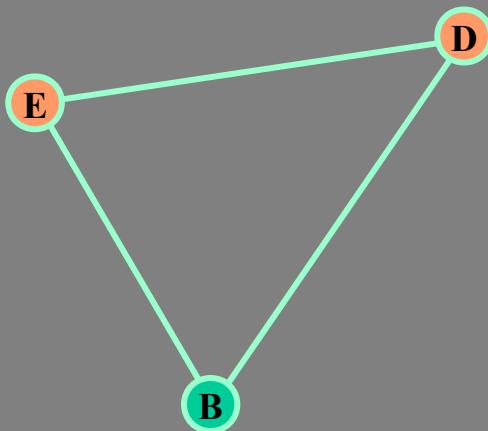
Arbre de sommets



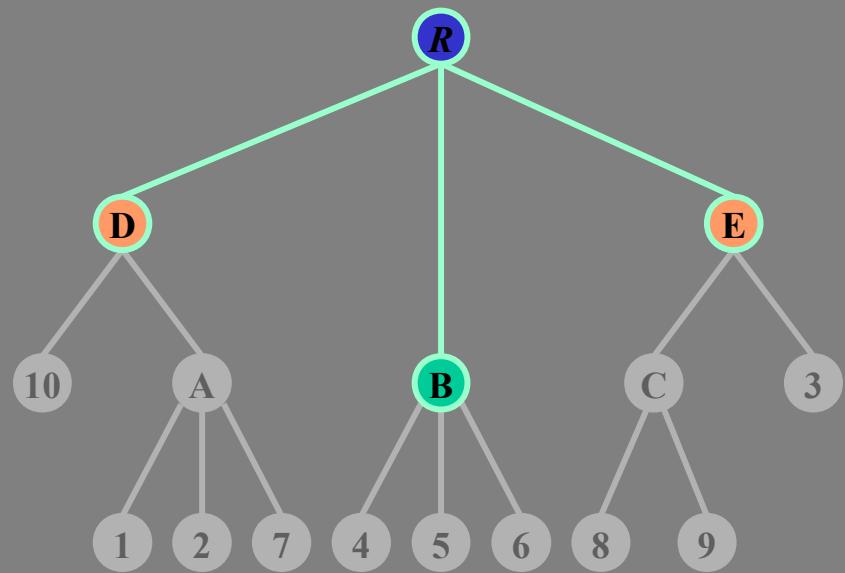
Triangles dans la liste active



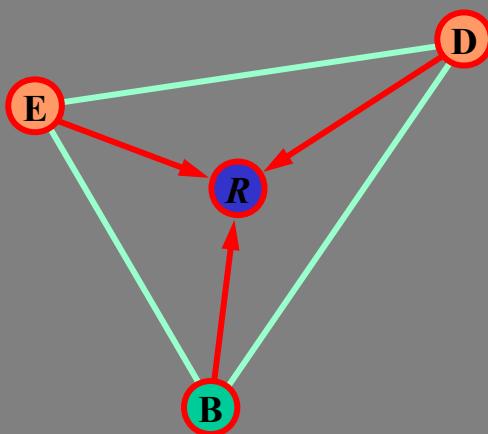
Arbre de sommets



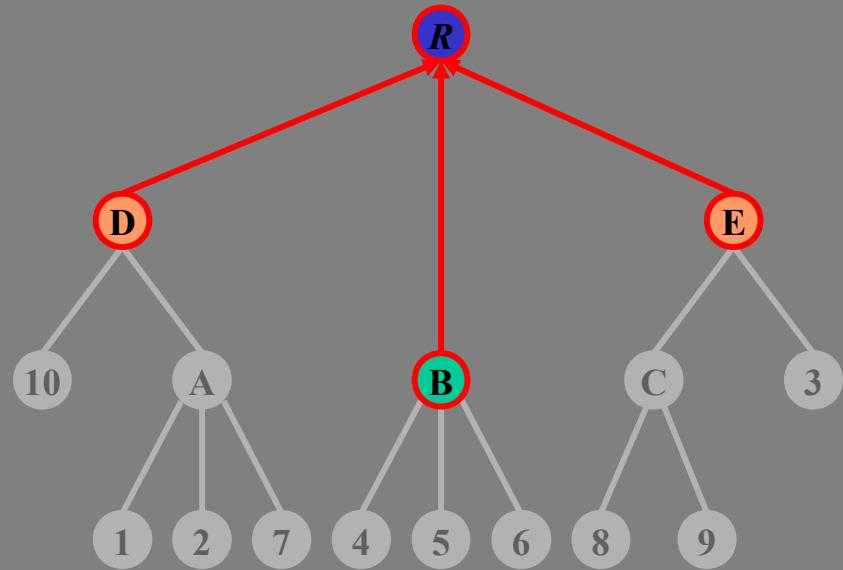
Triangles dans la liste active



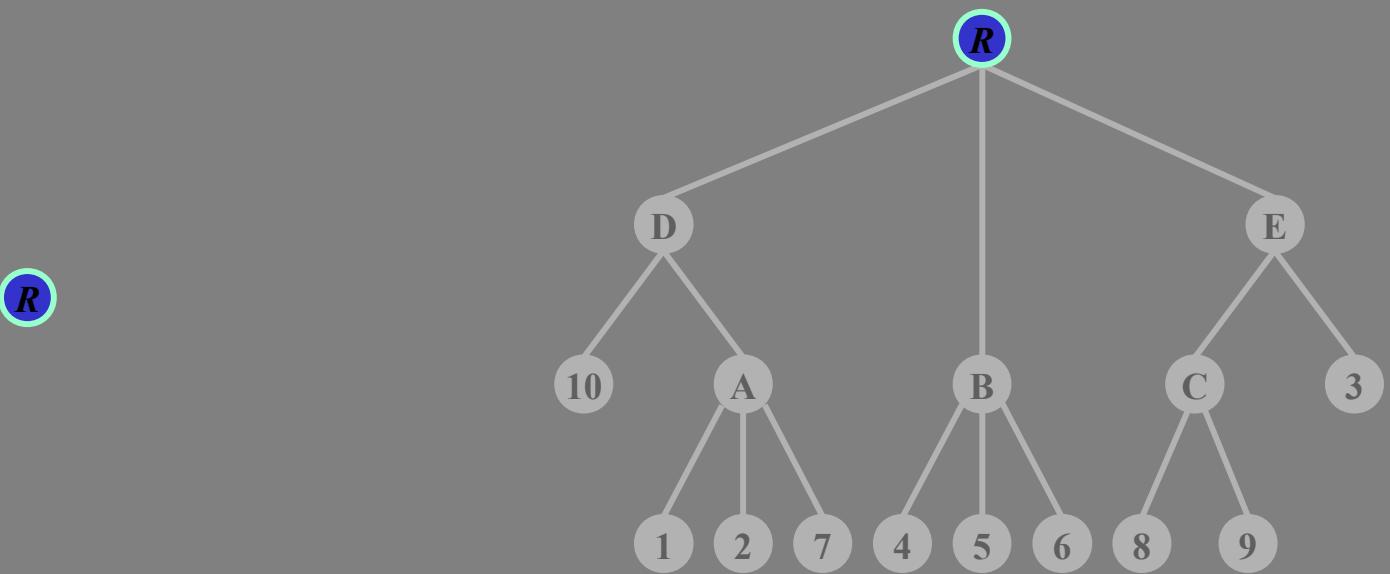
Arbre de sommets



Triangles dans la liste active



Arbre de sommets



Triangles dans la liste active

Arbre de sommets

Avantages

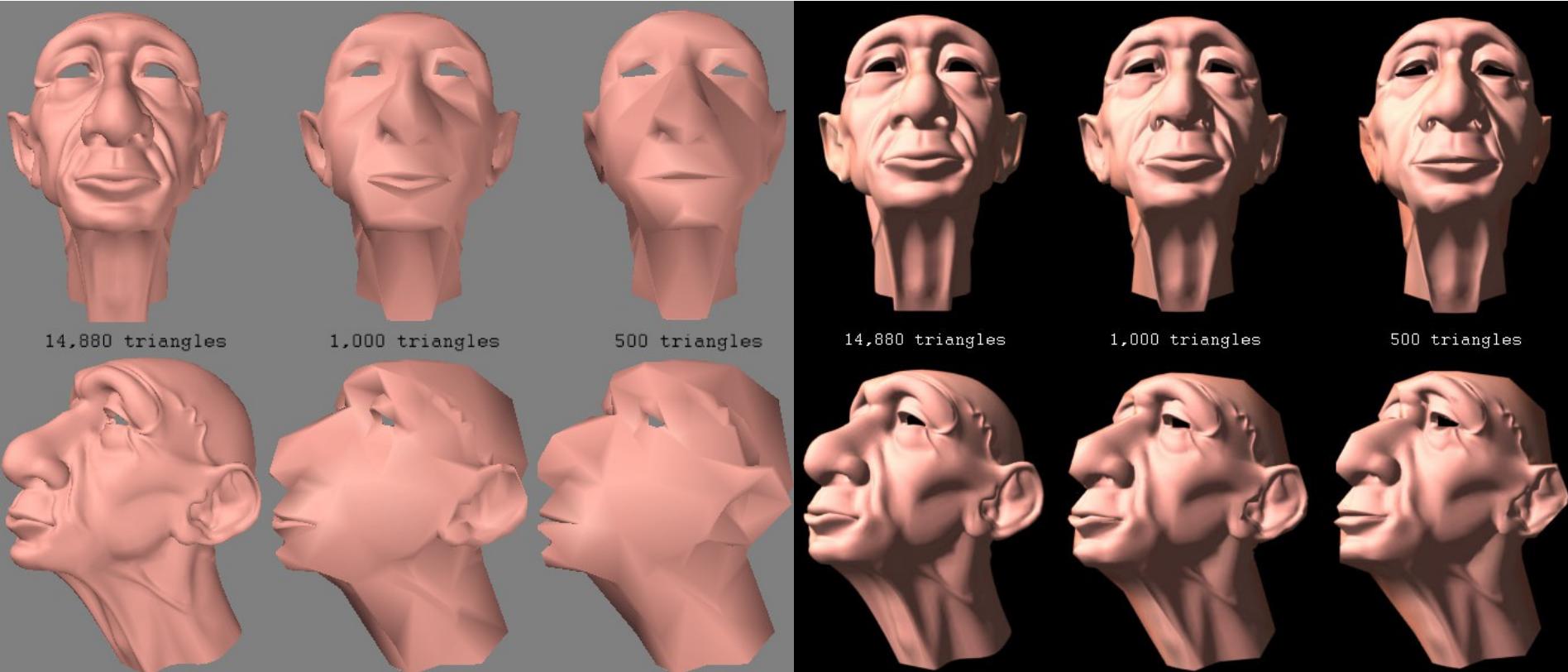
- Pas de problèmes de saut entre niveaux de détails.
- Pas nécessaire de stocker en mémoire les différents maillages car ils sont générés au vol directement.

Inconvénients

- Temps de calcul.

1.3 Amélioration du maillage simplifié

On peut améliorer le maillage simplifié en lui donnant l'apparence d'un maillage plus détaillé grâce au bump-mapping.



2. Simplification structurelle d'objets

Ces méthodes changent la structure de représentation des objets.

Ex: 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.

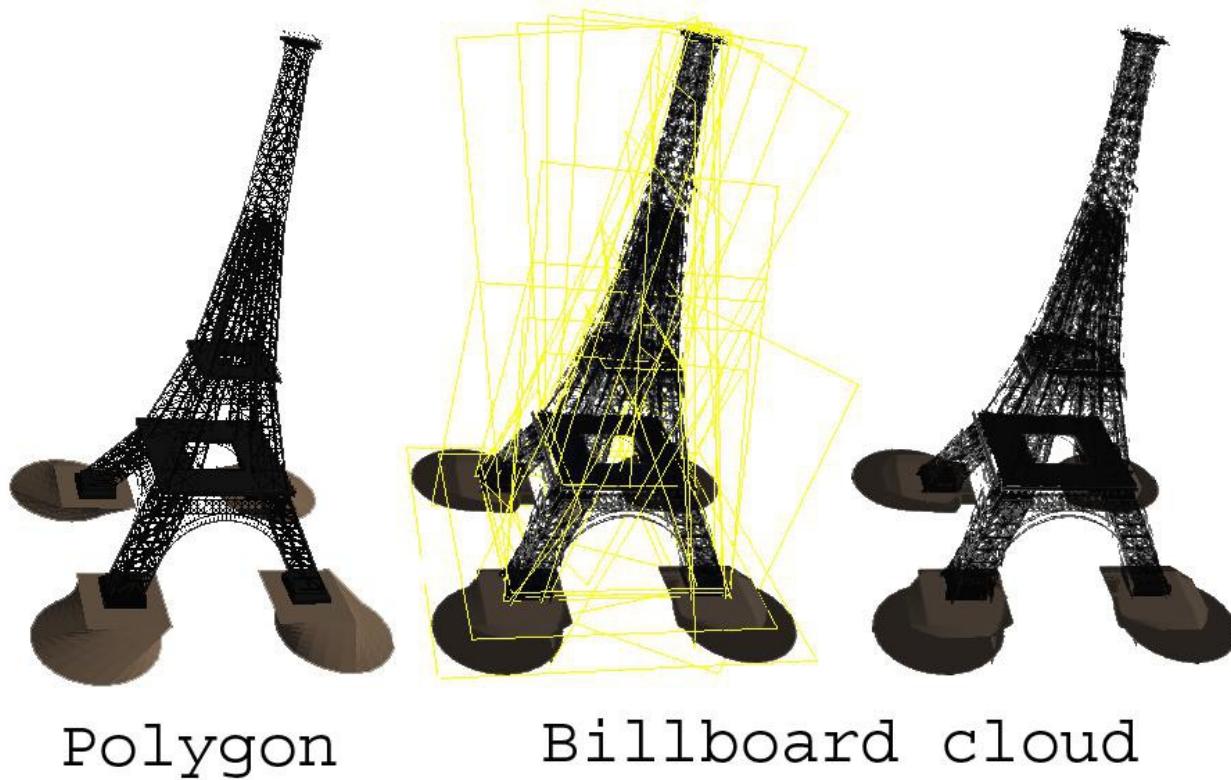
LOD ainsi créés = « **imposteurs** » (« *impostors* »)

Problèmes :

- dépendent souvent du point de vue.
- éclairage.

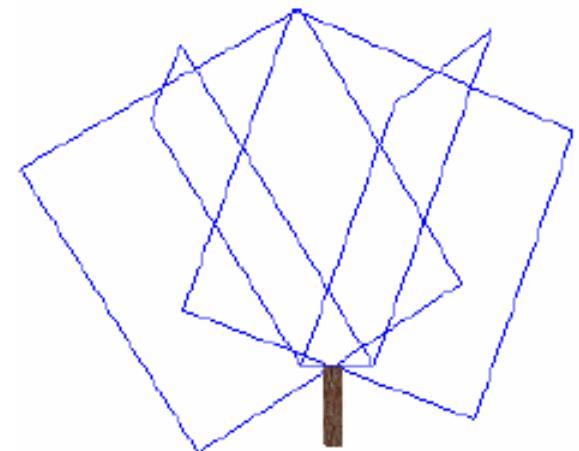
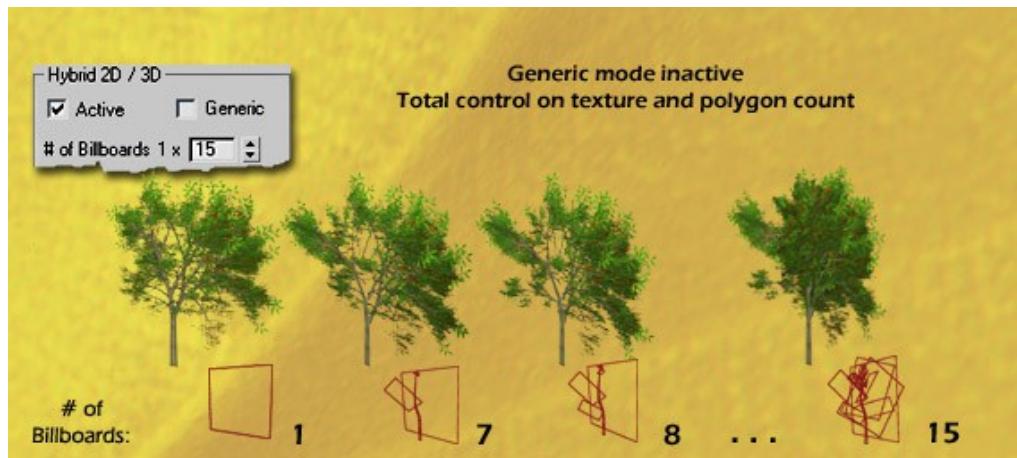
2.1 Billboard clouds

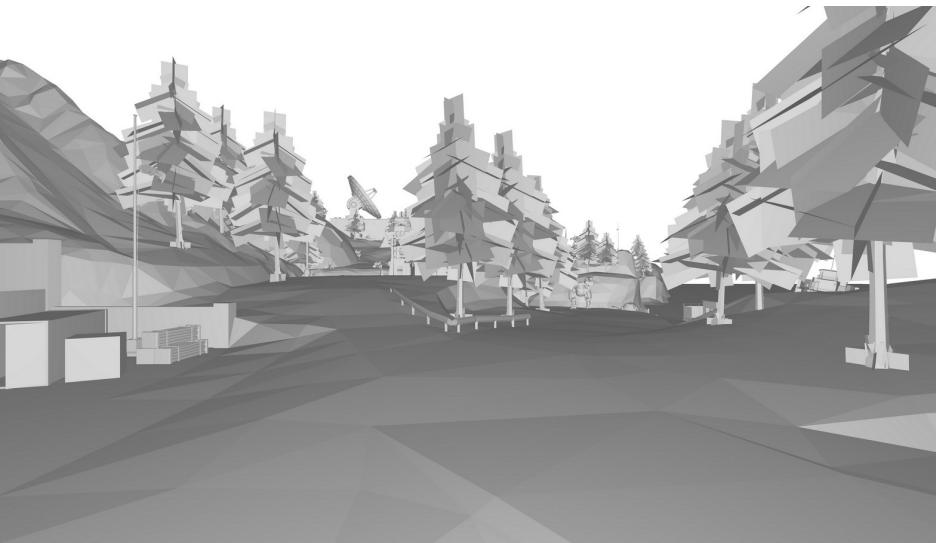
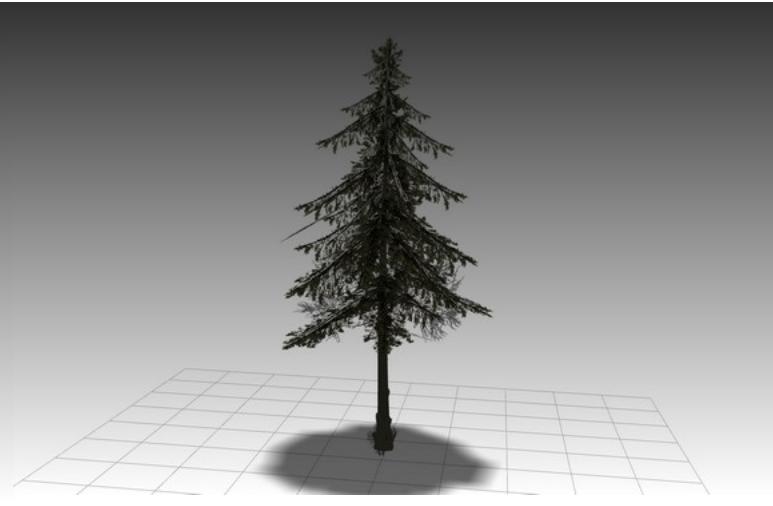
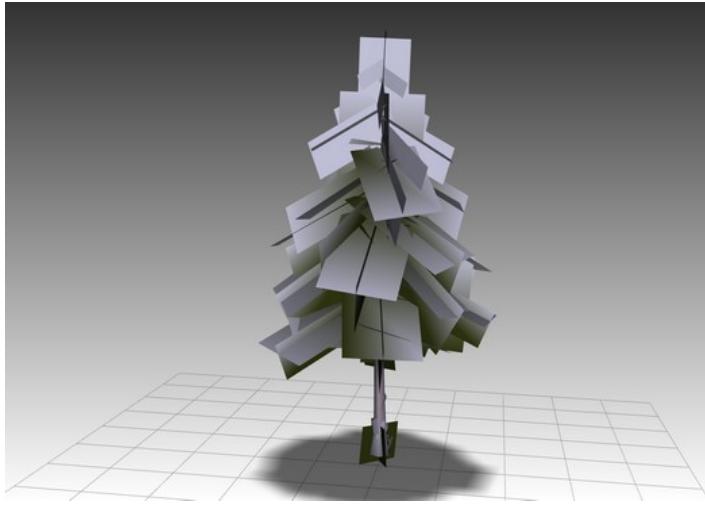
Un objet 3D est simplifié en un ensemble de polygones sur lesquels sont plaqués des textures semi-transparentes (= « *billboards* »). Il existe de nombreuses variantes, selon les manières de déterminer l'ensemble le plus réduit possible de billboards qui vont approcher l'objet initial.



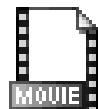
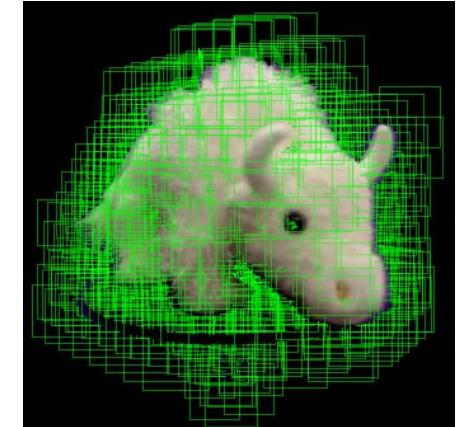
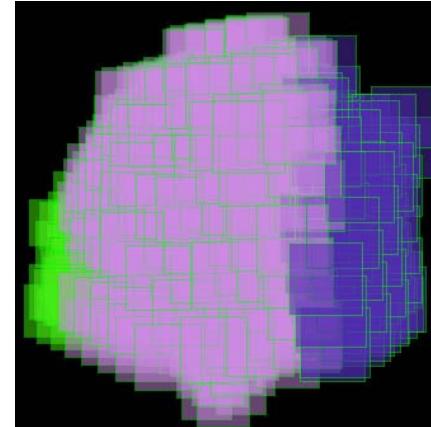
Exemple : Arbres

Arbres représentés avec plus ou moins de facettes et de quadrilatères texturés en fonction de la distance au point de vue.



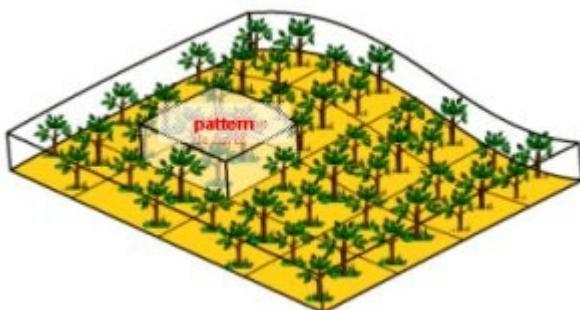


Exemple : objets à la géométrie complexe

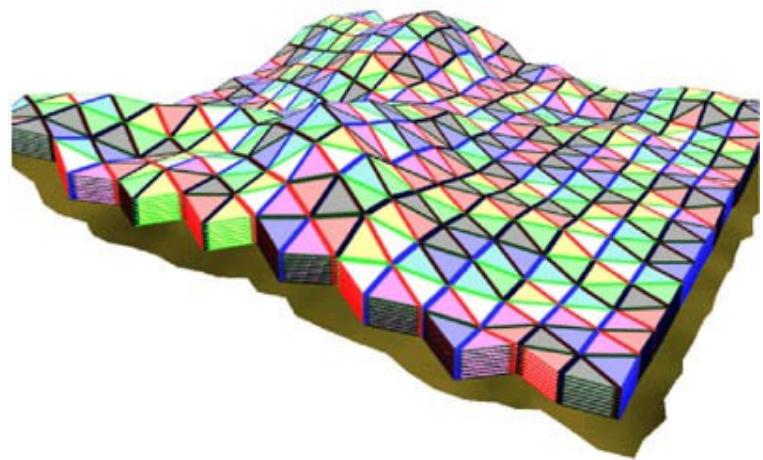


Microfacet billboarding (Shuntaro Yamazaki, 2002)
http://www.dh.aist.go.jp/members/shuntaro_e.php

Autre technique : prismes contenant des tranches de texture d'arbre



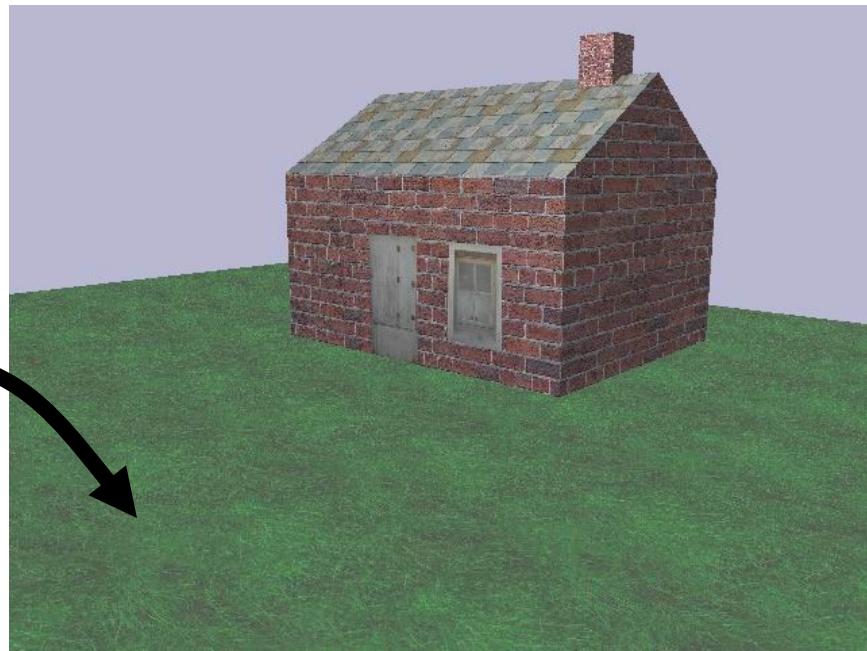
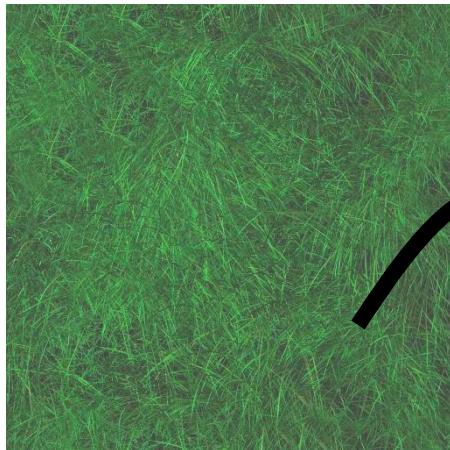
(Imagis, Grenoble)



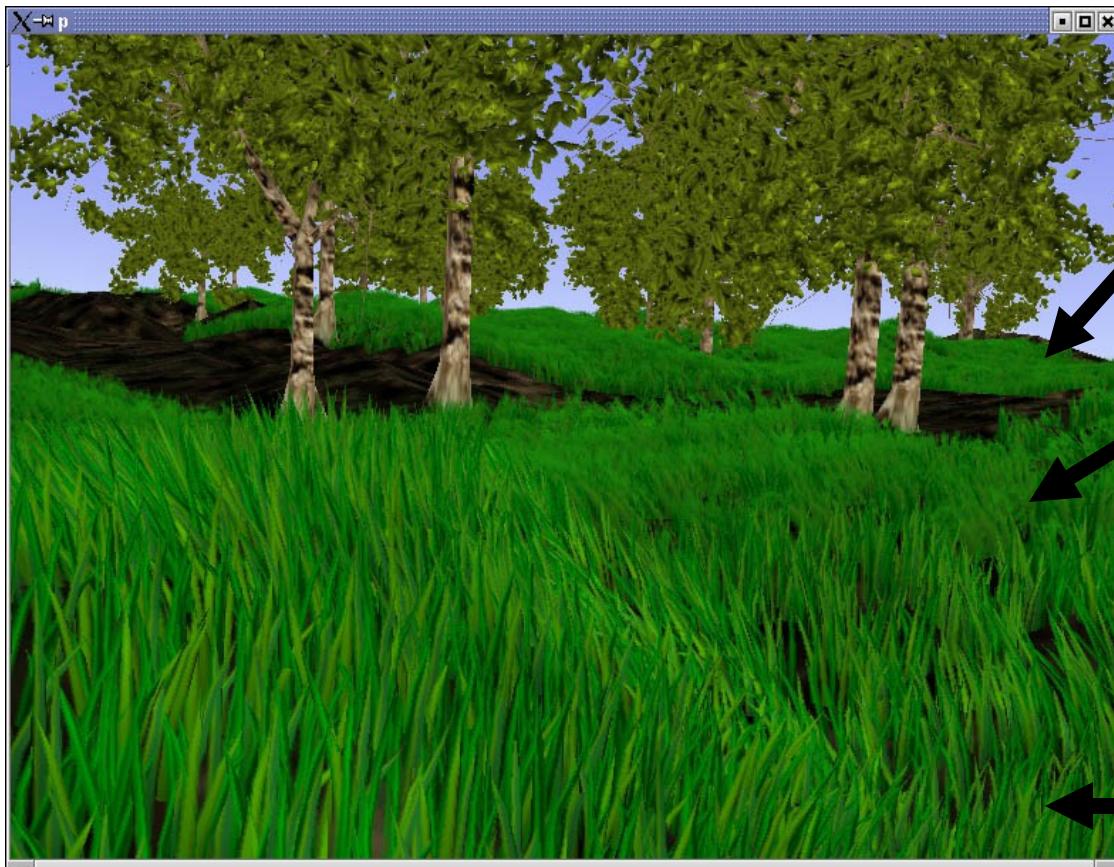
Exemple : herbe

Scène comportant de l'herbe : plusieurs millions de brins
→ impossible de les représenter tous de manière géométrique.

D'autre part, un simple plaquage de texture 2D n'est pas suffisamment réaliste.
→ il faut une solution intermédiaire.

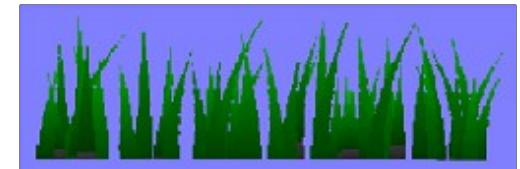


Niveaux de détails :

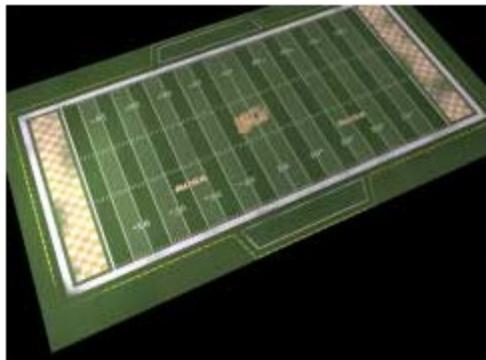


Au loin : simple texture verte.

A mi-distance : texture semi-transparente de brins d'herbe plaquée sur des polygones.



De près : brins d'herbe représentés individuellement en 3D.



Rendu temps réel d'étendue herbeuse avec éclairage dynamique

Combinaison de géométrie à l'avant plan et de volumes découpés en tranches (BTF = *Bidirectionnal Texture Function*)

Prise en compte dynamique de l'éclairage, d'ombres portées et de l'antialiassage.

Scène composée de 627 millions de brins d'herbe

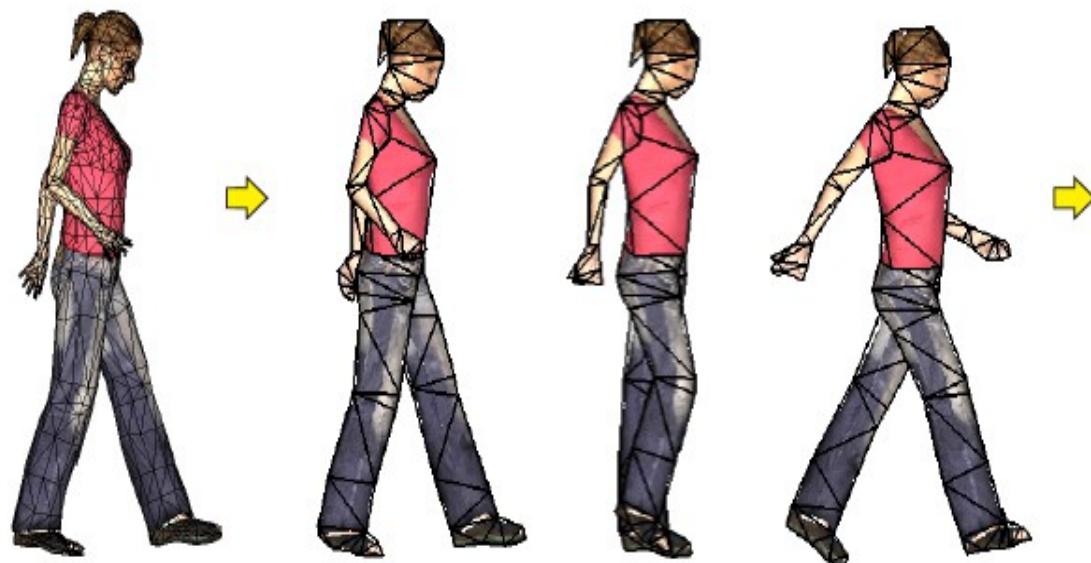
<http://www.kevinboulanger.net/grass.html>



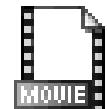
2.2 Imposteurs animés

Exemple : *PolyPostors* : représentation de personnages d'apparence 3D avec des polygones 2D texturés.

→ Permet de représenter de très grandes foules en utilisant des modèles 3D à l'avant plan et des imposteurs à l'arrière plan.



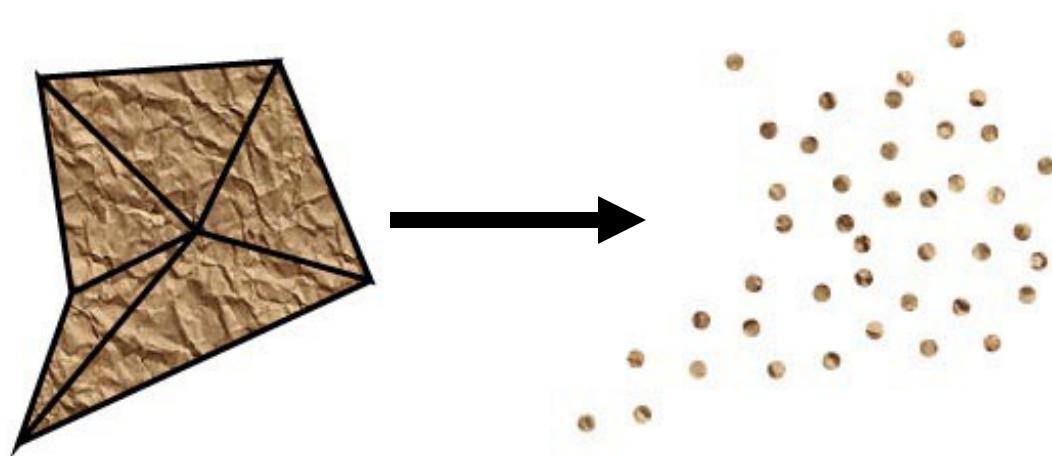
<http://www.jarmilakavanova.cz/ladislav/papers/ddd-i3d08/ddd-i3d08.htm>



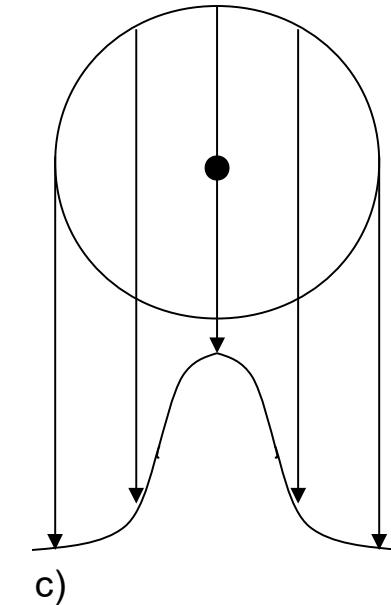
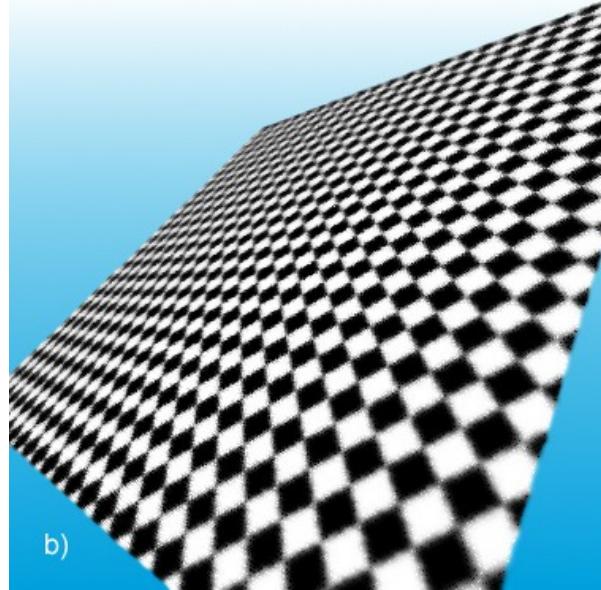
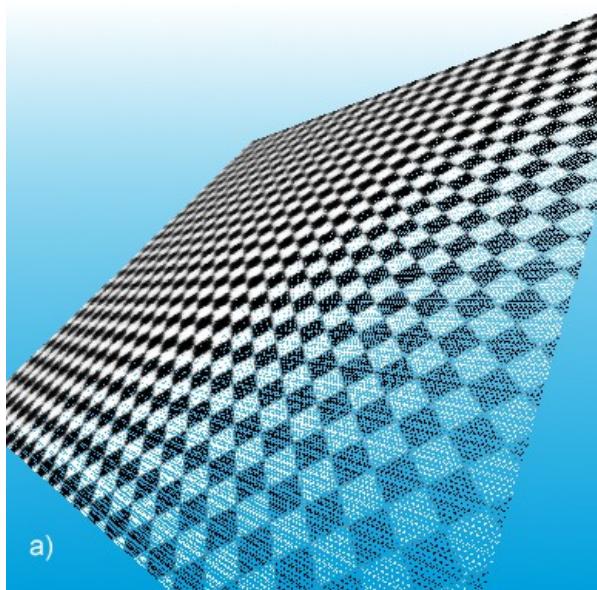
2.3 Point based rendering

Idée : de nos jours, les objets 3D sont si volumineux que leurs triangles projetés à l'écran sont très petits (occupent peu de pixels) → plutôt que d'afficher l'objet avec un maillage triangulaire, on va chercher à l'afficher sous la forme de points.

Des points sont échantillonnés sur la surface d'objets 3D. On stocke les coordonnées du point, sa couleur, la normale en ce point, éventuellement la courbure, etc.



Quand les échantillons sont projetés à l'écran, il peut y avoir des trous (a). Pour remplir ces trous (b), on effectue une interpolation des couleurs des échantillons affichés grâce à une convolution avec une gaussienne 2D (c).

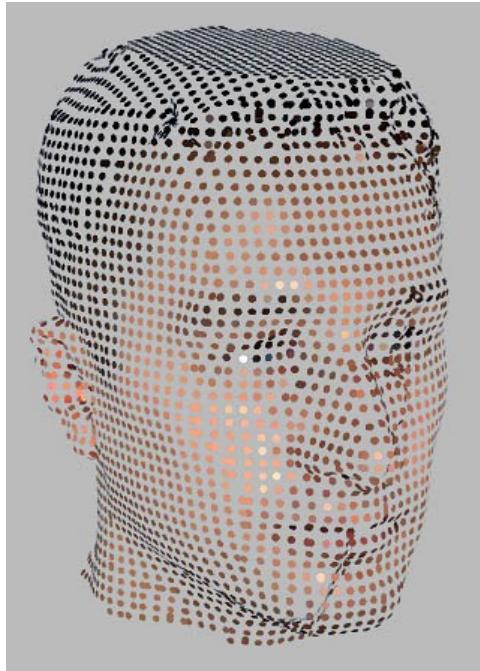


MOVIE
MOVIE
MOVIE



2.4 Splatting

Il s'agit d'une variante du point-based rendering. Plutôt que d'afficher un échantillon sous la forme d'un simple points, on utilise un élément de surface (surfel : *surface element*) : un quadrilatère, un disque, une ellipse, etc. Il faut suffisamment d'échantillons et des surfels suffisamment grands pour qu'il n'y ait pas de trous.



On code souvent chaque surfel avec sa position, sa taille, sa normale.

L'affichage par surfels peut être accéléré grâce au GPU.



<http://graphics.uni-bielefeld.de/research/points/>

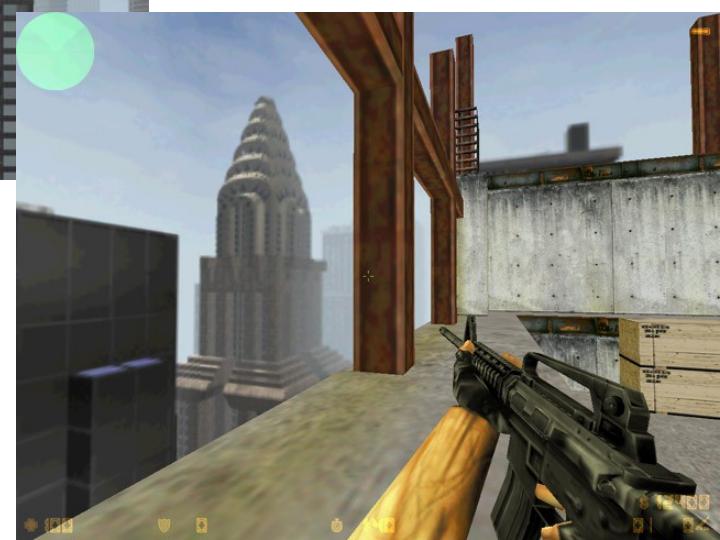
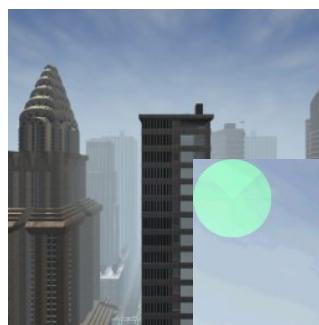
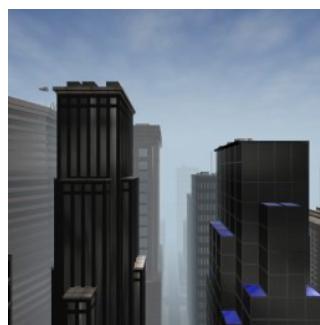
3. Simplification de scène

Ces méthodes cherchent à simplifier des régions de la scène plutôt que les objets eux-mêmes.

Utilisé pour la visualisation de très grandes scènes virtuelles (villes, paysages, etc.)

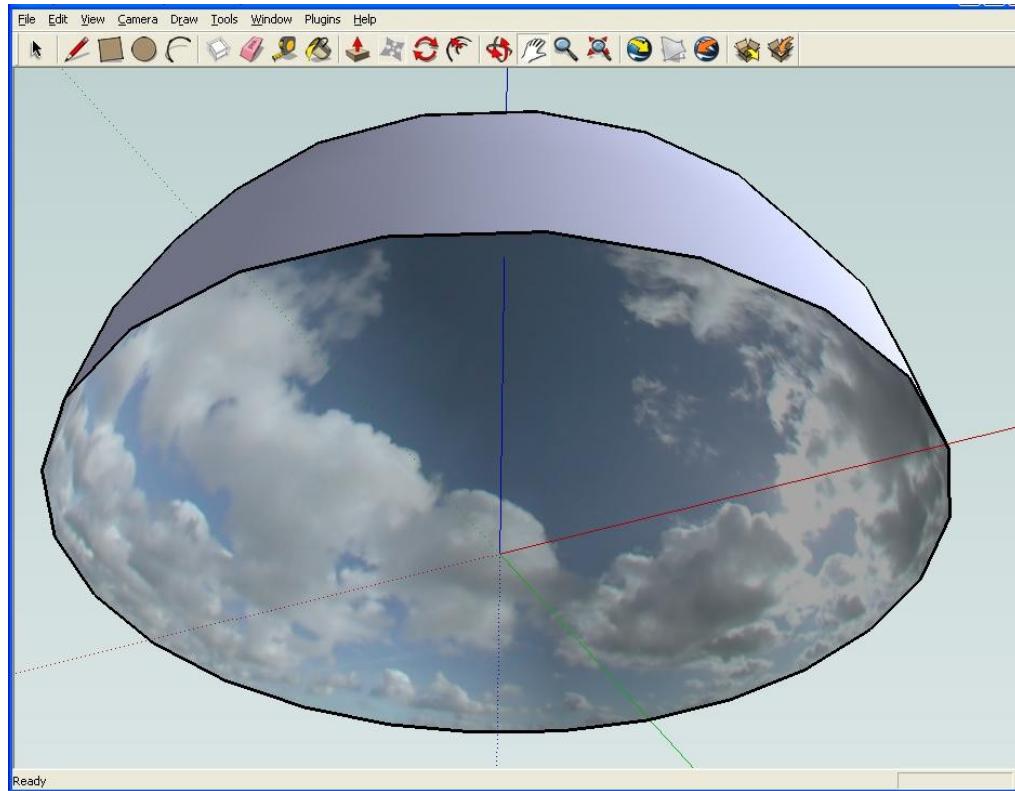
3.1 Skybox

On plaque des textures prédéfinies sur les faces internes d'un cube au milieu duquel se trouve la caméra. Permet de représenter une géométrie très importante mais lointaine.



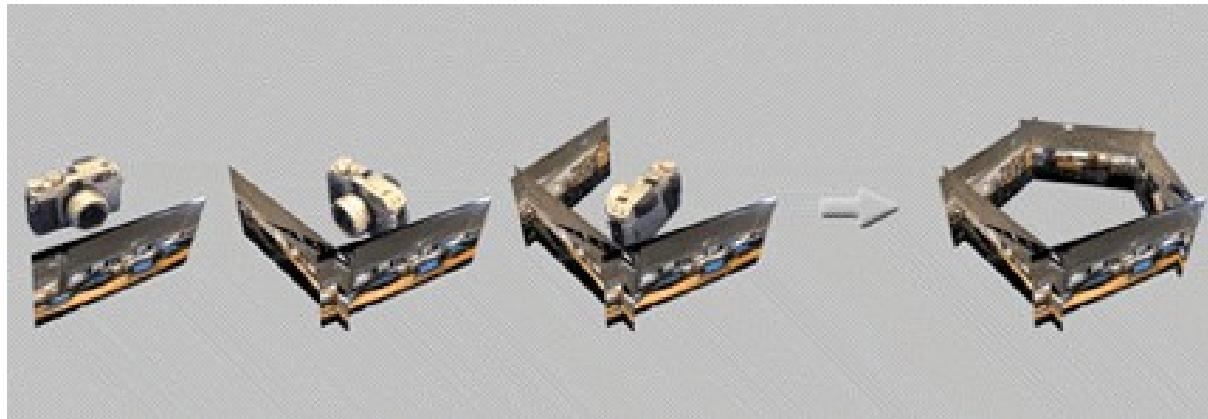
3.2 Skydome

Même principe que la skybox, sauf qu'on texture l'intérieur d'une hémisphère au lieu d'un cube. Une seule texture est nécessaire au lieu de 6. La caméra se trouve ensuite toujours au centre de ce dôme.



3.3 Quicktime VR

Technologie développée par Apple (1994) permettant de visualiser des scènes à partir d'un ensemble de photos (ou d'images de synthèse) prises à partir d'un point sur 360°. Une image panoramique est constituée à partir de ces photos. Lorsqu'elle est affichée, cette image est déformée sur l'intérieur d'une boîte, d'un cylindre ou d'une sphère pour donner l'illusion de la perspective.



Utilisé dans les jeux d'aventure de la société Cryo (moteurs Omni3D, Cryogen) : Atlantis, Versailles, etc.

<http://vrm.vrway.com/>

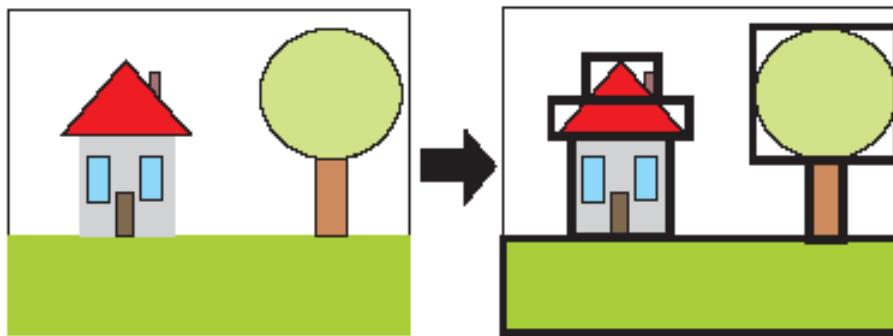
3.4 Google Street View

Ce principe de scène à 360° reconstituée à partir d'un ensemble de photos prises par plusieurs caméras depuis un même point de vue (par une voiture ou un pieton) est utilisé dans Google Street View.



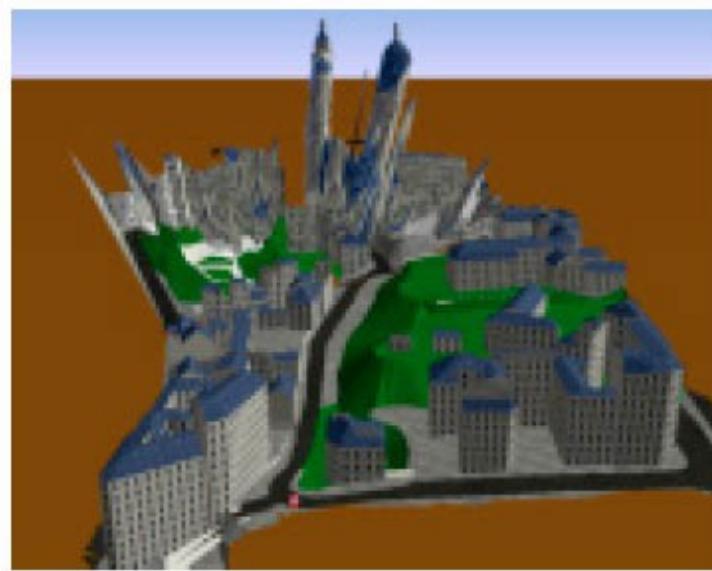
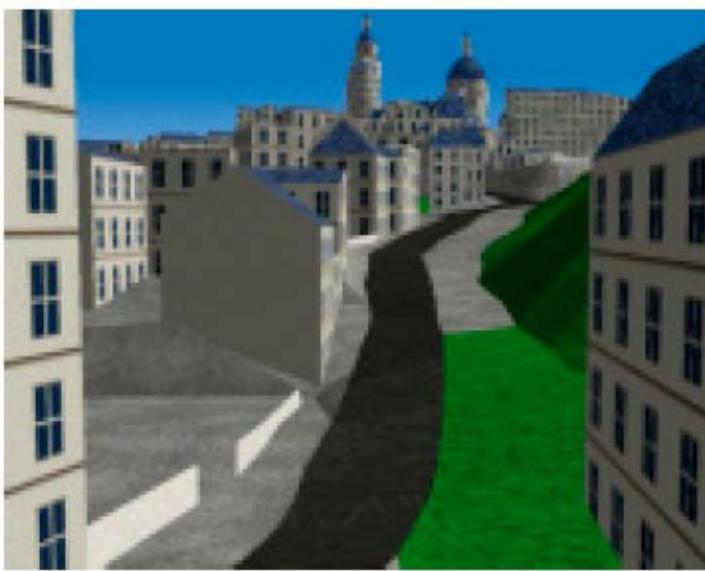
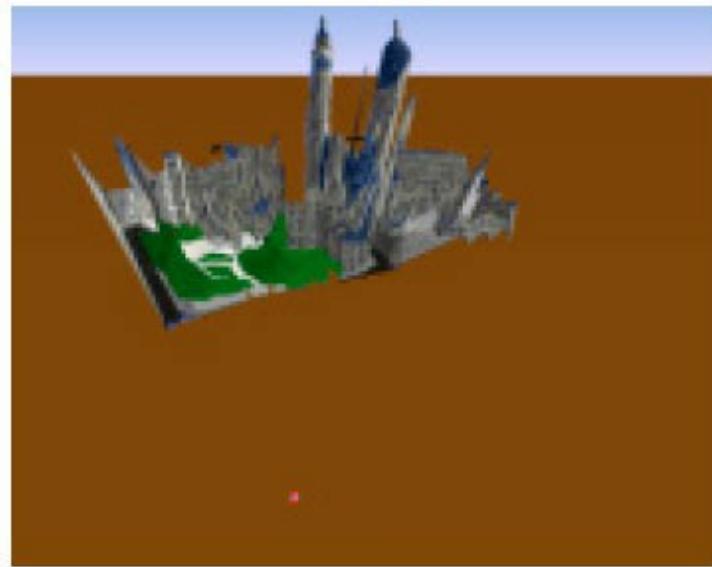
3.5 Villes

On positionne dans la scène un ensemble de polygones sur lesquels on plaque une texture provenant du rendu de la géométrie que l'on veut simuler.



Problèmes :

- Comment déterminer les zones de géométrie à remplacer par cette technique ?
- Transition entre le trompe-l'œil et la véritable géométrie lorsque le point de vue se déplace.

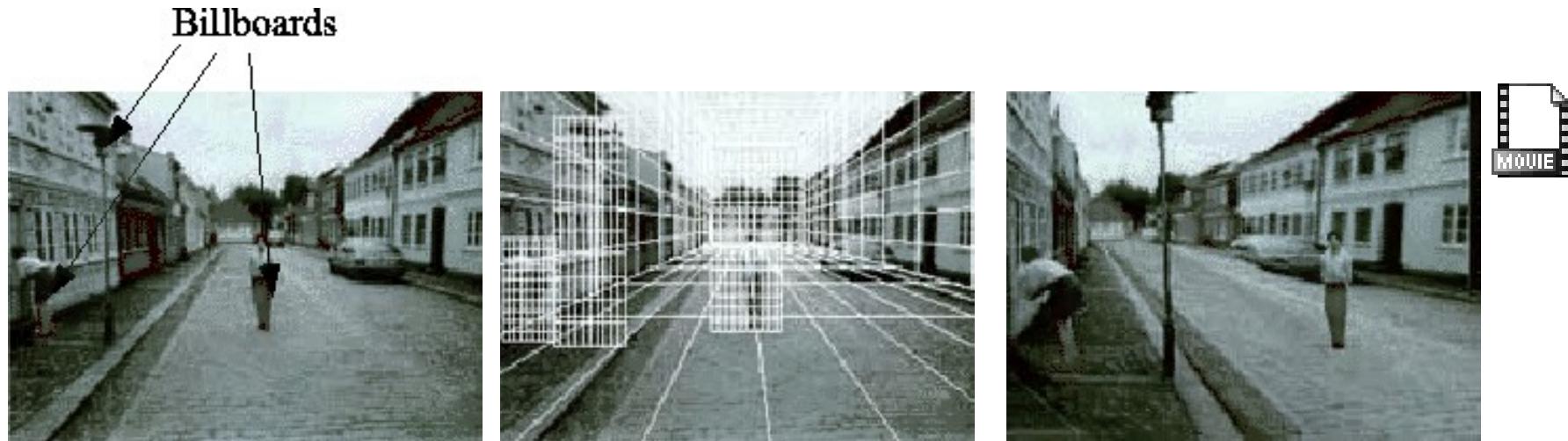


<http://www-sop.inria.fr/reves/publications/data/1997/SDB97/SDB97.pdf>

3.6 Déplacement dans une image 2D

Principe : déformer une image en fonction de son point de fuite, selon un maillage défini par l'utilisateur. Ce maillage sépare l'image en différentes zones (arrière plan, sol, bords, etc.). Des éléments de l'avant plan extraits de l'image sont utilisés comme billboards pour augmenter l'illusion de pénétrer dans l'image.

Très artistique, cet effet a été utilisé dans plusieurs clips vidéo et publicités.



« Tour into the picture »

Youichi Horry, Ken-ichi Anjyo, and Kiyoshi Arai, 1997

http://www.cc.gatech.edu/classes/AY2000/cs4480_spring/cases/tip-siggraph98/html/tip000.htm

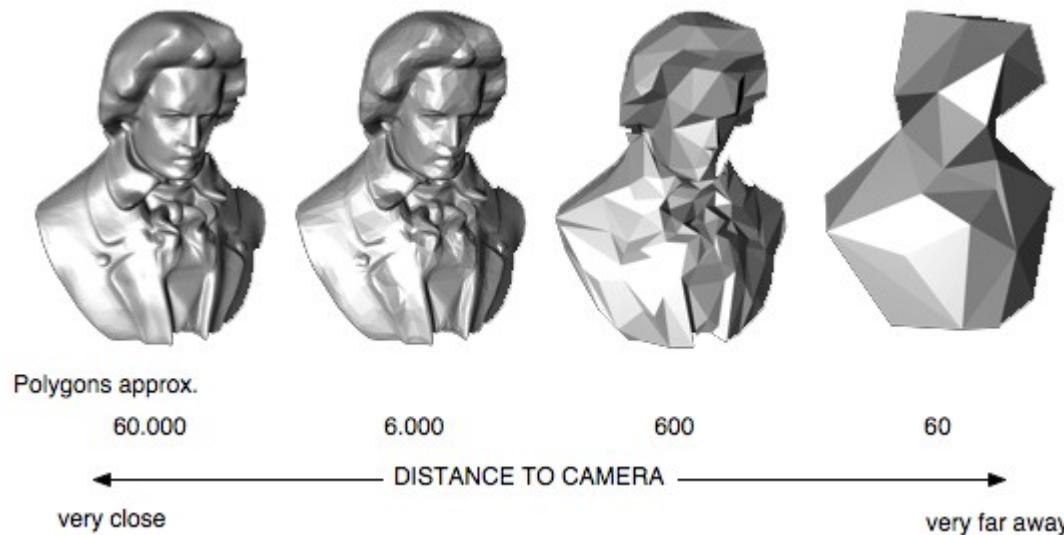


4. Gestion des niveaux de détails

Le choix des niveaux de détails à afficher peut dépendre de plusieurs paramètres :

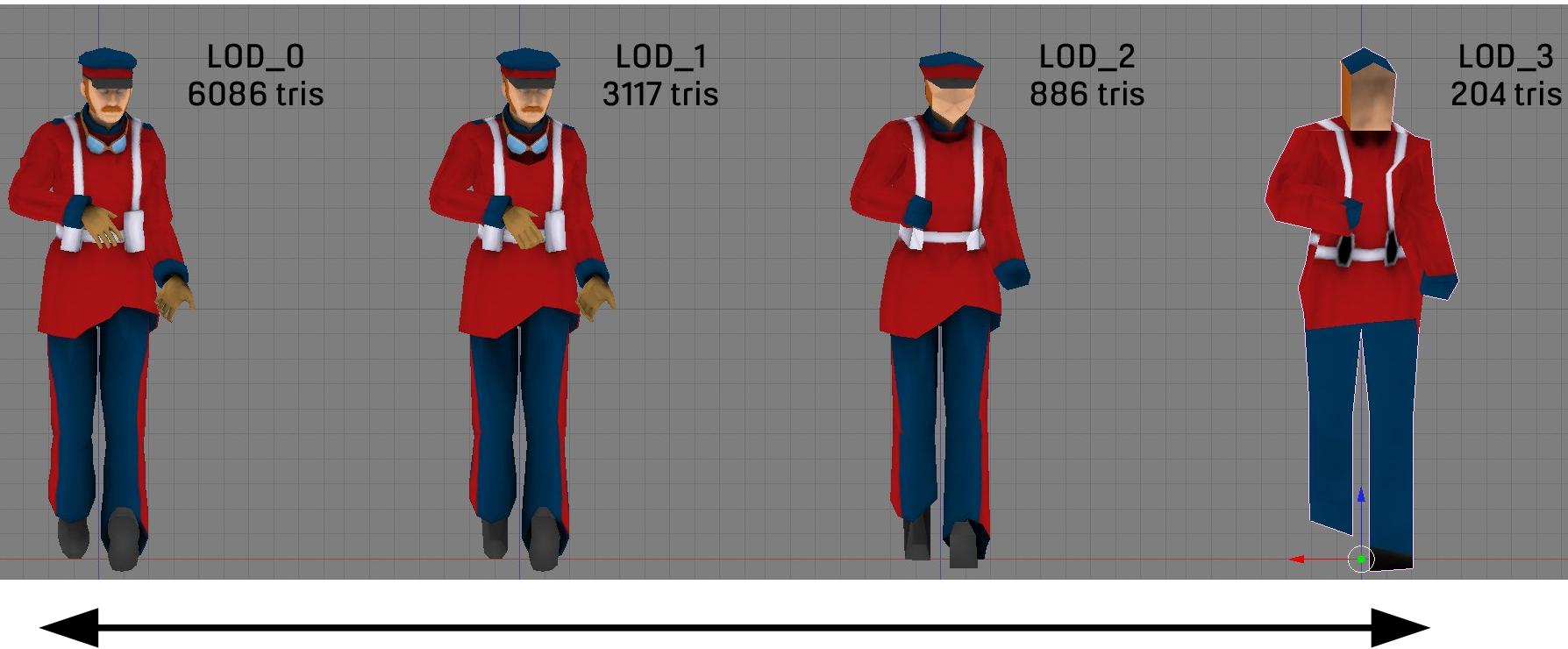
4.1 Distance

Plus un objet est éloigné, moins ses détails sont perceptibles. On pourra donc afficher une version plus grossière sans différence visuelle.



4.2 Puissance de calcul

En fonction de la puissance de la machine, du temps de calcul disponible, on affichera des objets à la géométrie plus ou moins complexe.



Machine puissante

Machine peu puissante

4.3 incidence

Selon l'angle de vue, un objet peut avoir une forme très simple qui ne nécessite pas d'afficher toute la géométrie.

Ex: l'arrière d'un bas-relief peut se résumer à une représentation très simple.



Avant



Arrière

4.4 Vitesse de l'objet par rapport à la caméra

Un objet en mouvement est moins bien perçu par le système visuel humain qu'un objet fixe.

Si un objet se déplace rapidement, les détails de sa surface seront moins perceptibles. On pourra alors choisir une représentation moins détaillée.

De plus, cette simplification peut être camouflée en appliquant un effet de « flou de bouger » (*motion blur*) qui ajoutera plus de réalisme au mouvement.



4.5 Vision périphérique

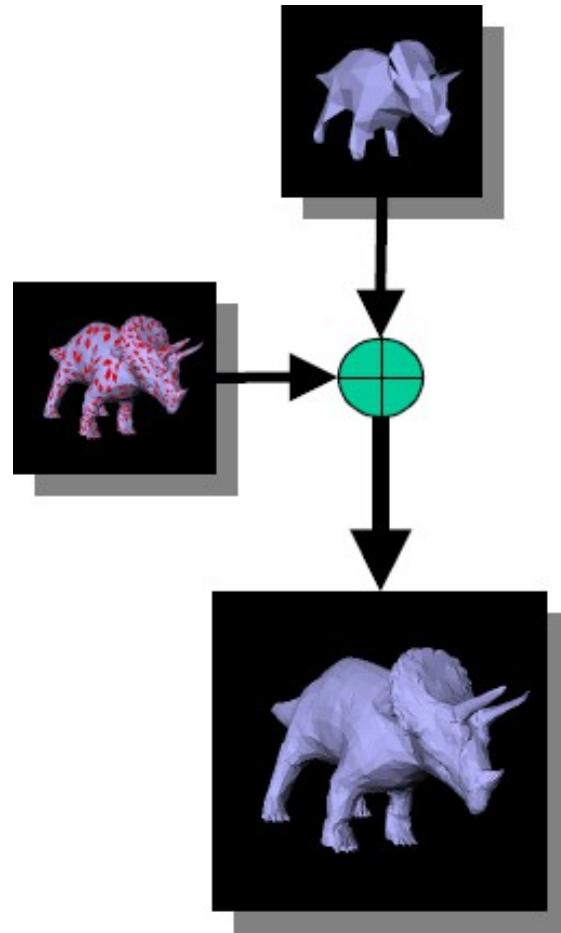
Possibilité de dégrader la qualité des objets en périphérie de l'image :

- Flou radial pour renforcer l'effet d'une accélération.
- Réalité virtuelle : l'attention de l'utilisateur n'est concentrée que dans la direction de son regard (casque doté d'eye tracking).



4.6 Transmission progressive

Les LOD peuvent être utilisés pour transmettre progressivement sur un réseau un objet ou une scène à la géométrie complexe.



Exemple d'utilisation de niveaux de détail :

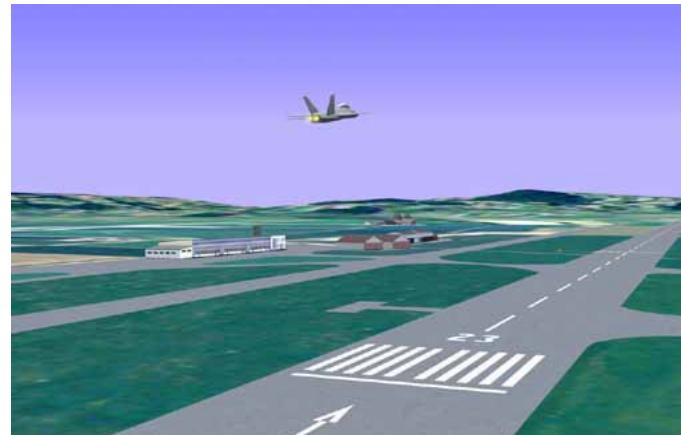
Représentation de terrains 3D

1. Définition d'un terrain 3D

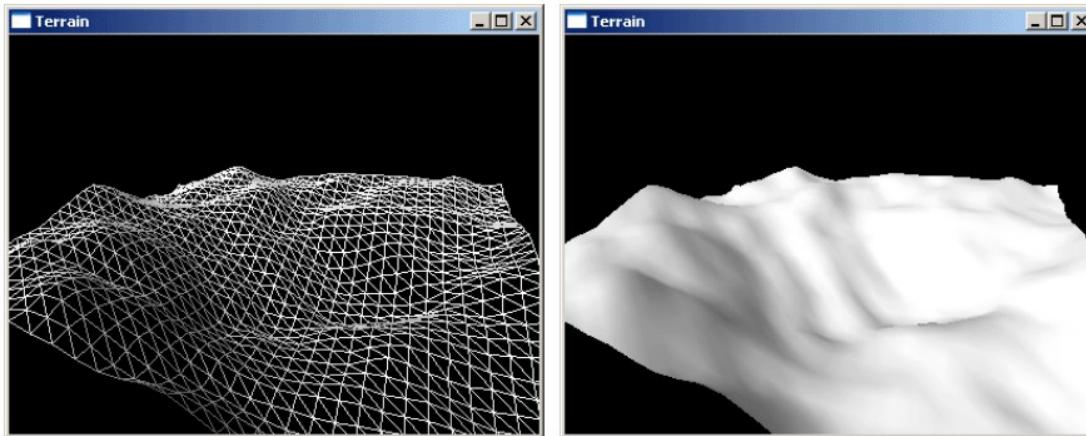
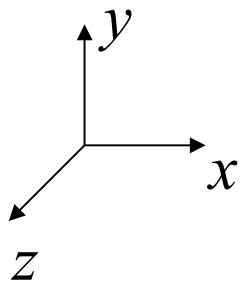
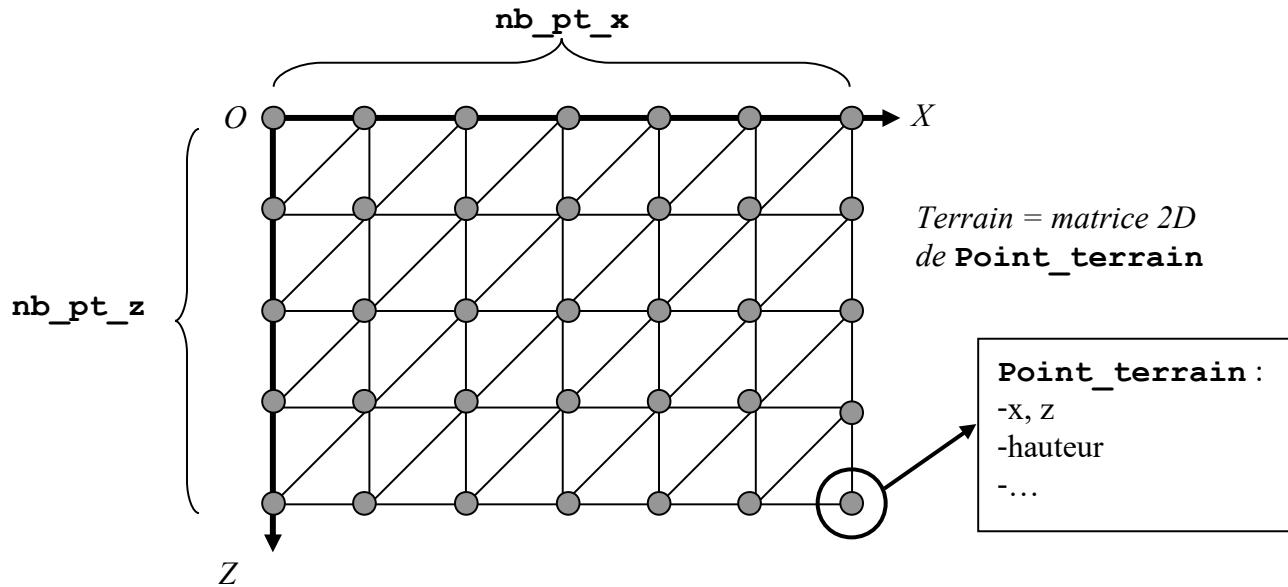
MNT : Modèle Numérique de Terrain
(DEM : *Digital Elevation Model*)

Nombreux domaines d'utilisation :

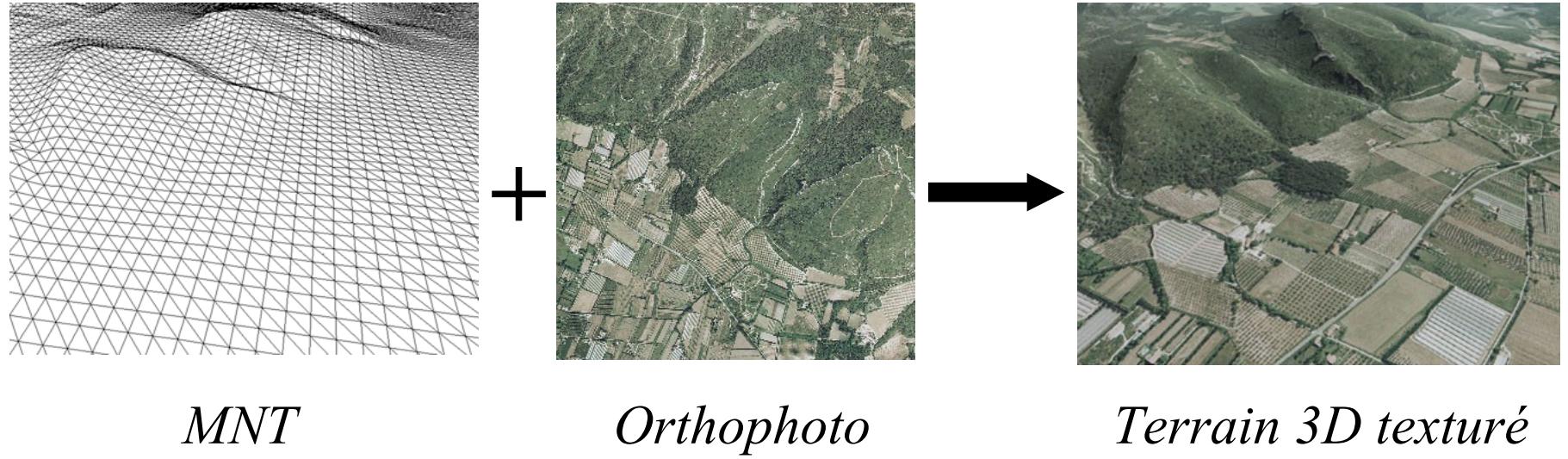
- simulateurs
- jeux vidéos
- cinéma
- visualisation scientifique
- météo
- ...



Un terrain est représenté avec une carte d'élévations. Selon le principe le plus simple, cette carte est uniforme.

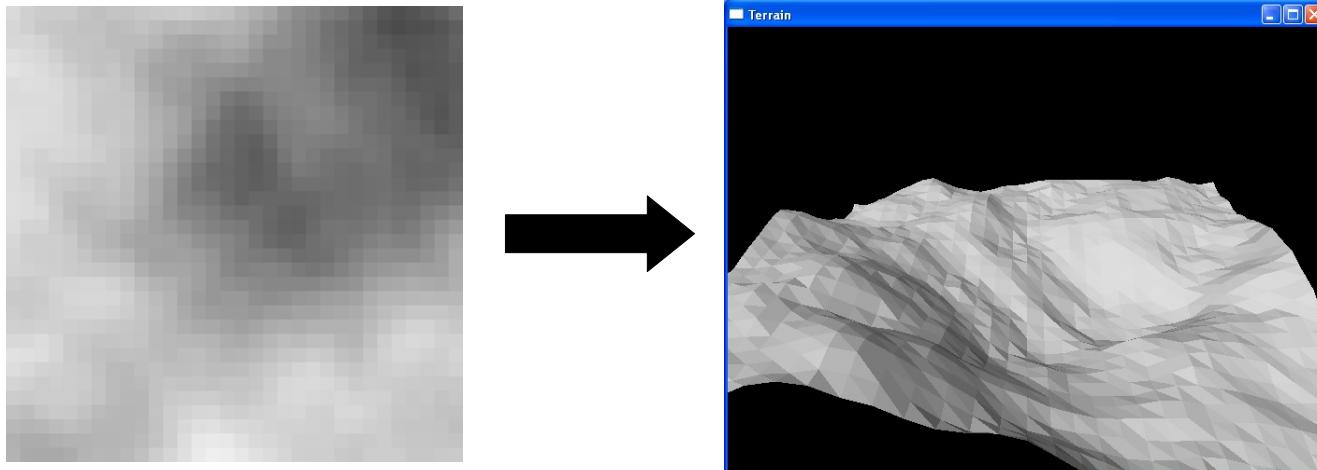


Plaquage de texture sur un MNT



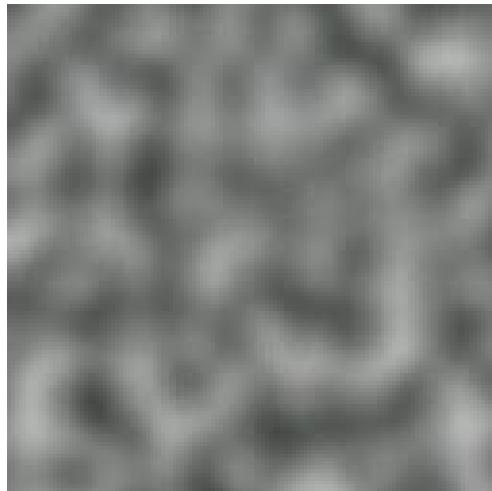
Comment obtenir une carte d'élévation

- A partir de données du monde réel, obtenues par mesures (relevés topographiques au sol, LIDAR, photogrammetrie, etc.).
Ex: BD ALTI de l'Institut Géographique National (IGN) ; données de la mission SRTM de la NASA ; etc.
- Modélisé à la main en 3D dans un logiciel d'images de synthèse.
- Modélisé à la main en dessinant une image 2D (généralement en niveaux de gris : un pixel correspond à un point de la carte, la valeur du niveau de gris donne l'altitude du point).

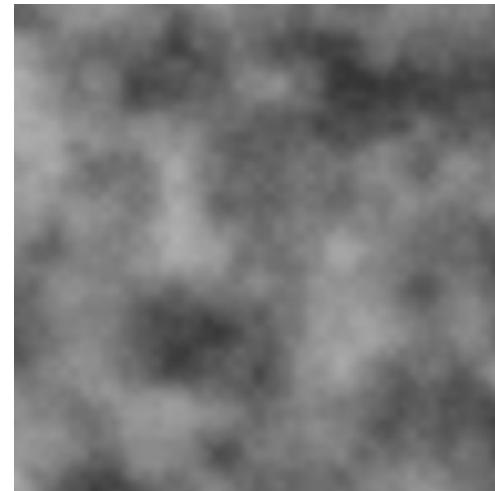


- Généré procéduralement avec un algorithme (valeurs pseudo aléatoires, bruit de Perlin, sommes de sinus, patchs de Bézier, ...).

Bruit de Perlin : fonction (2D, 3D, 3D+temps, etc.) qui interpole des valeurs précalculées pour donner une valeur qui évolue pseudo aléatoirement dans l'espace (plus éventuellement le temps).



Bruit de Perlin



Turbulence de Perlin, obtenue en additionnant plusieurs bruits à plusieurs échelles (octaves)

<http://www.noisemachine.com/talk1/>

Problème

Un terrain peut parfois représenter des millions de faces.

→ Impossible d'afficher en temps réel une zone de plusieurs km² (nécessaire pour des simulateurs, surtout de vol)



Grand Canyon

4,097 x 2,049 sommets ~ 16.7 millions de triangles

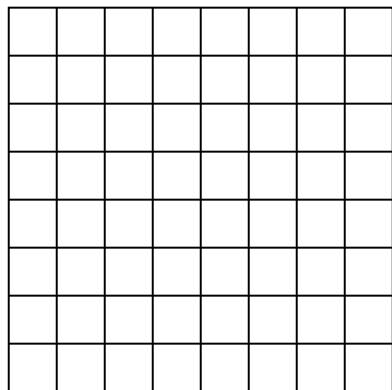
→ Il existe de nombreuses méthodes de représentation de terrains intégrant la notion de niveaux de détail.

2. Méthodes de représentation

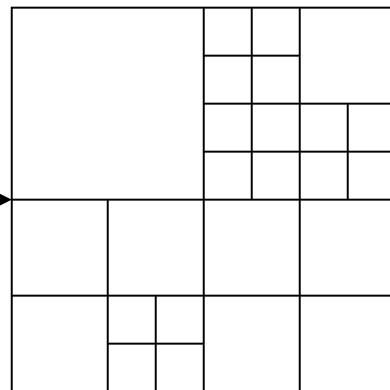
2.1 Quadtree

Objectifs :

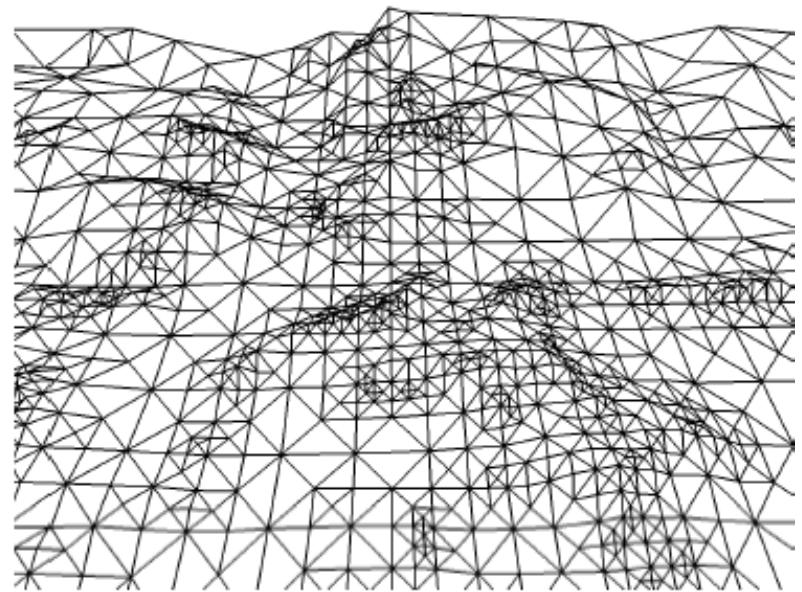
- Réduire le nombre de points.
- Déterminer rapidement les polygones des zones à afficher.
- Afficher des zones + ou – grossières (nœuds + ou – proches des feuilles) en fonction de la distance.



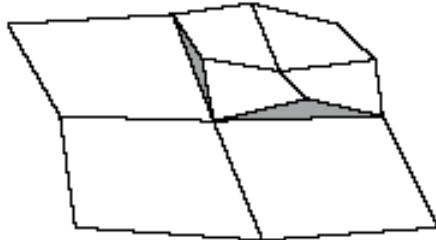
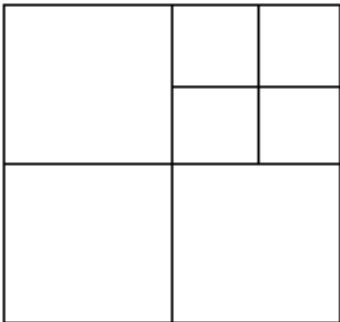
Uniforme



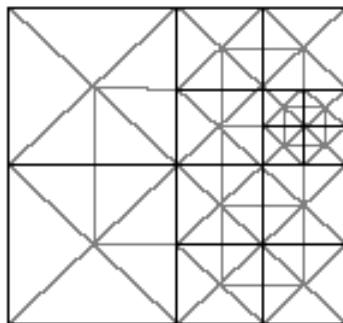
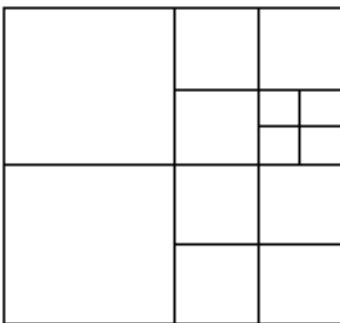
Quadtree



Génération des triangles



Problème de jonction en « T »
(*« T-cracks »*)



Solution : chaque carré est subdivisé en 8 triangles, 2 triangles par côté, à moins que le côté ne borde un carré plus grand. Dans ce cas, un seul triangle est créé le long de ce côté.

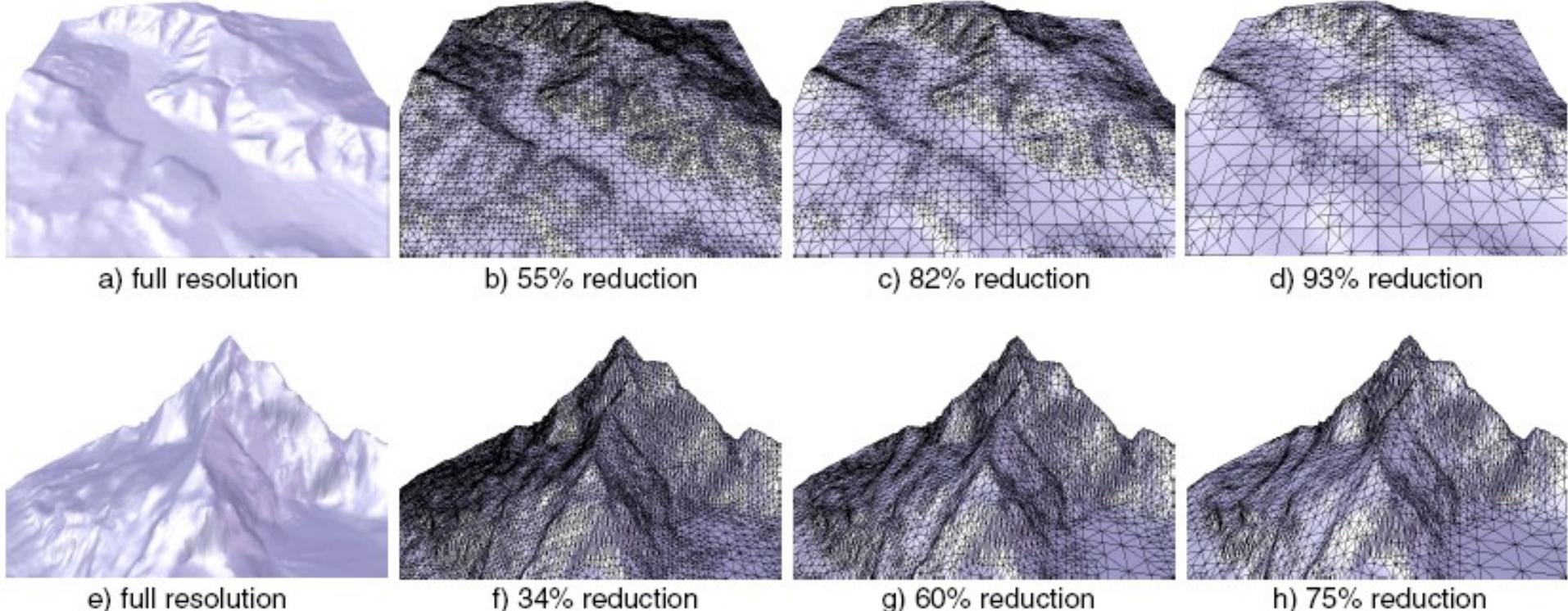


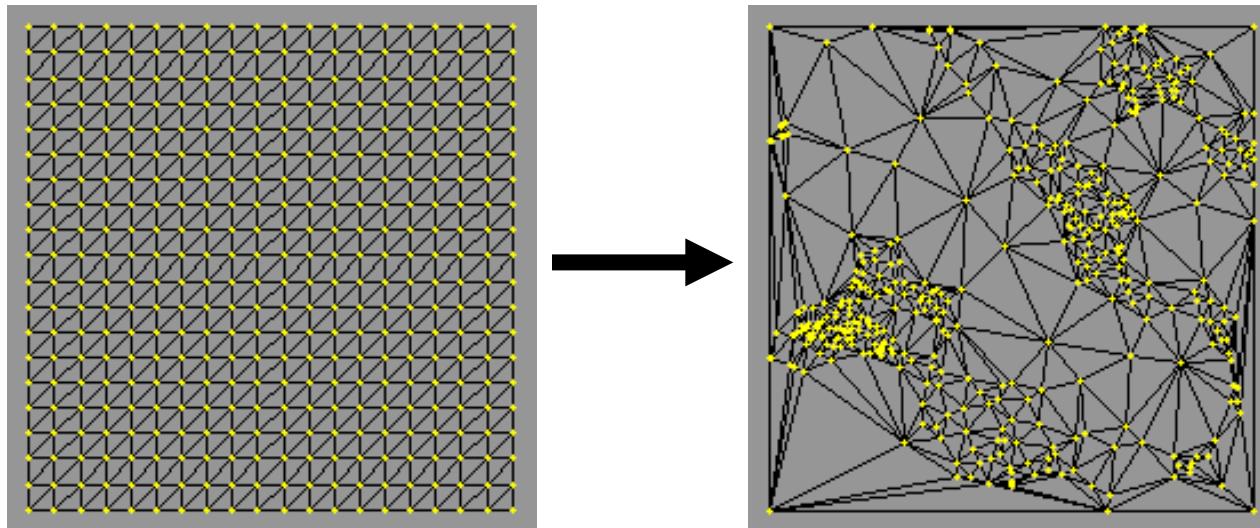
FIGURE 26. Top row of images shows the *Albis* terrain patch with a) 25921, b) 11636, c) 4664 and d) 1789 points. The bottom row shows the *Matterhorn* terrain patch with e) 25921, f) 17156, g) 10492 and h) 6367 points.

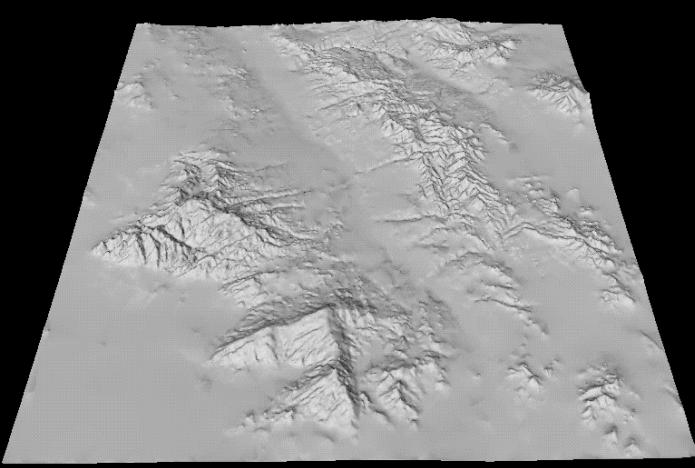
2.2 Triangulated Irregular Network (TIN)

La carte d'élévation n'est pas donnée par une matrice 2D uniforme d'altitudes mais par un réseau irrégulier de triangles.

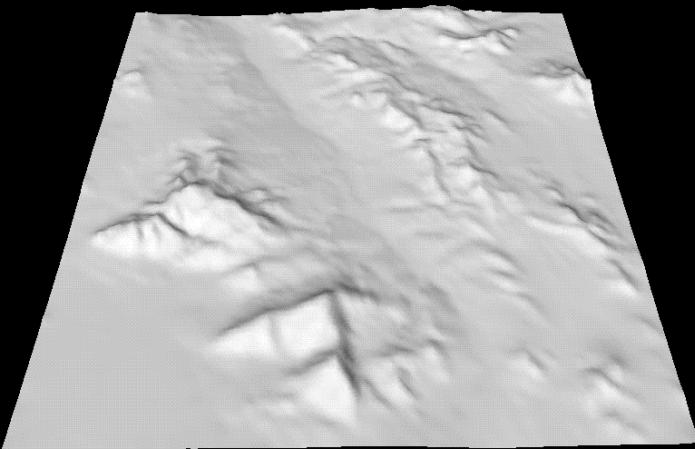
On aura plus de triangles dans des zones détaillées (changements de relief) que dans des zones plus homogènes (plates).

Cette triangulation peut être obtenue avec la **triangulation de Delaunay**.

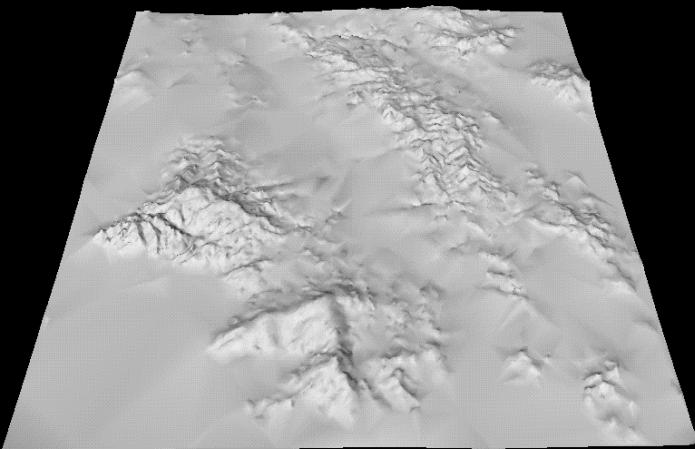




Énumération uniforme
1201 x 1201 points
2 880 000 triangles



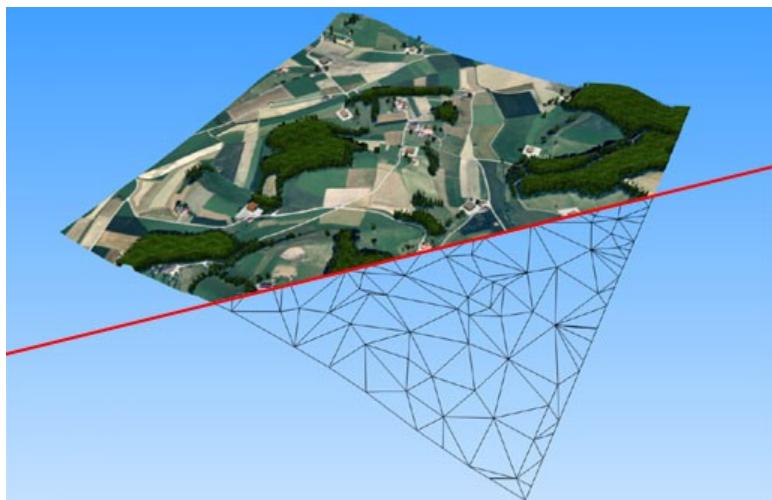
Énumération uniforme
120 x 120 points
28 322 triangles



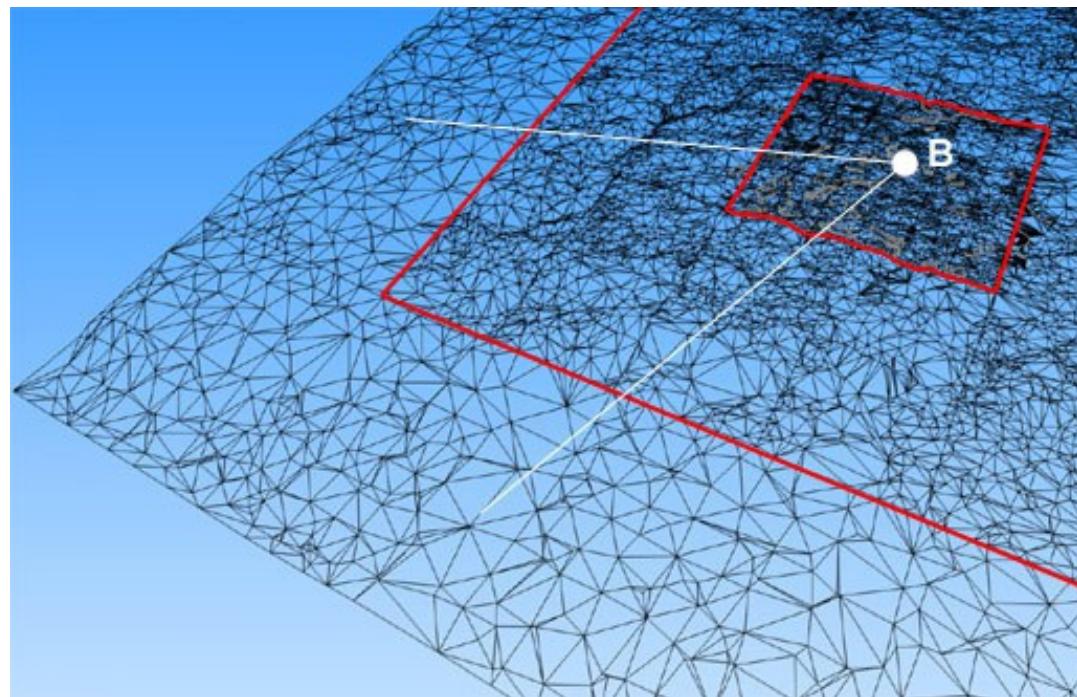
TIN
14400 points
28 700 triangles

2.3 Découpage du terrain en dalles avec LOD

En fonction de la distance par rapport à l'observateur, on utilise des dalles couvrant une surface plus ou moins importante et comportant plus ou moins de triangles.



Une dalle



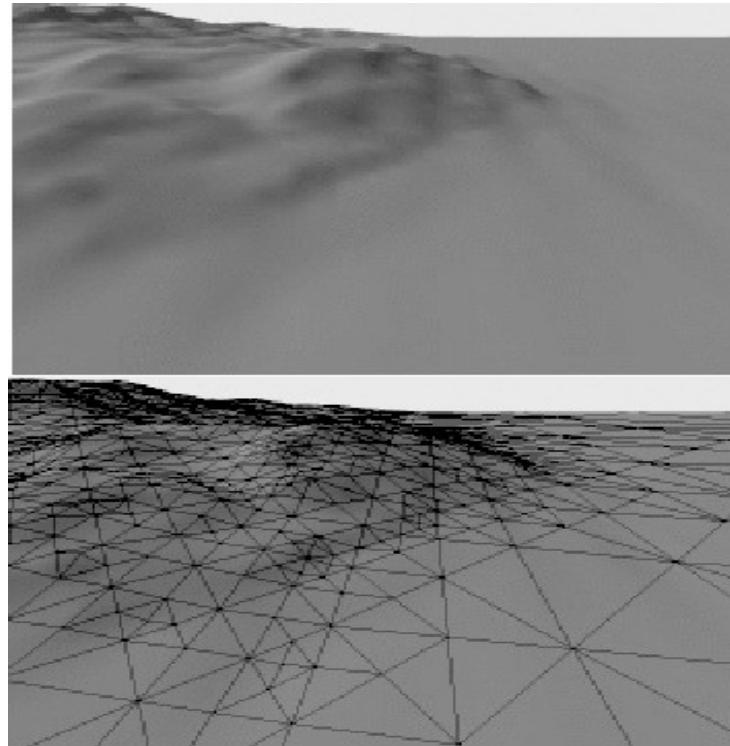
Le terrain est un assemblage de dalles

→ Problèmes de transition lorsque l'observateur se déplace, les dalles passent d'un LOD à un autre.

2.4 ROAM (Real-time Optimally Adapting Meshes)

Génération de triangles à la volée en fonction du point de vue.

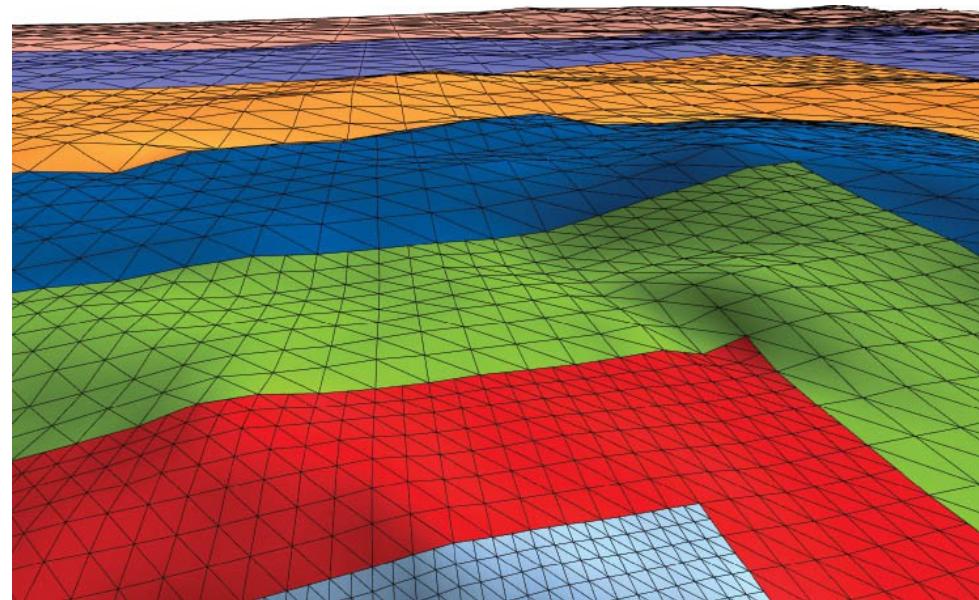
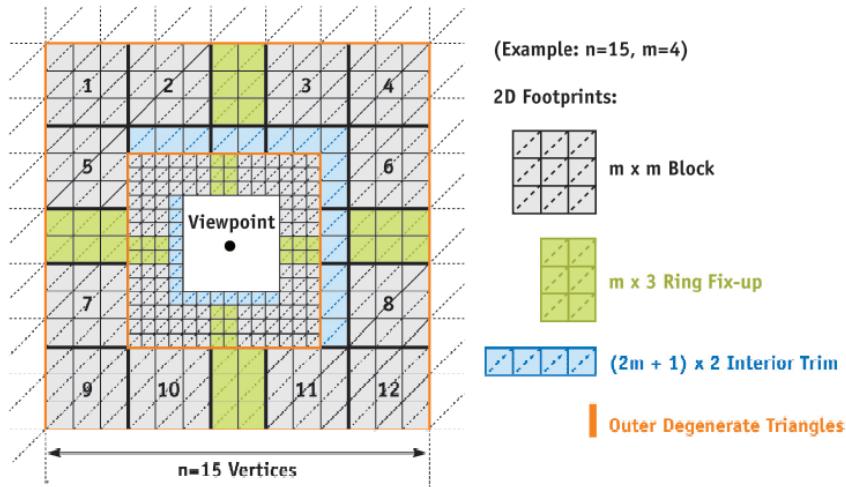
<http://www.ics.uci.edu/~pajarola/pub/UCI-ICS-02-01.pdf>



2.5 Geometry ClipMaps

Utilisation de niveaux de détails différents précalculés autour du point de vue.

<http://research.microsoft.com/~hoppe/gpugcm.pdf>



Il existe encore bien d'autres techniques, c'est un domaine de recherche scientifique très actif, dont les travaux sont ensuite utilisés dans de nombreux domaines (simulation, réalité virtuelle, jeu vidéo, ...)

Vous trouverez une liste assez exhaustive des techniques de représentation de terrain 3D sur :

<http://www.vterrain.org/LOD/Papers/>