

Infographie : Fiche

Sommaire

- Infographie : Fiche
 - Sommaire
 - 1. Modélisation géométrique
 - 1.1. Introduction
 - 1.2. Modèles explicites
 - 1.2.1. Modèle B-Rep
 - 1.2.1.1. Principe
 - 1.2.1.2. Modèle winged-edge
 - 1.2.1.3. Modèle half-edge
 - 1.2.1.4. Subdivision d'un triangle
 - 1.2.2. Les énumérations spatiales - Voxel
 - 1.2.2.1. Principe
 - 1.2.2.2. Formats de fichier
 - 1.2.2.3. Jeux
 - 1.3. Modèles implicites
 - 1.3.1. Octree
 - 1.3.1.1. Principe
 - 1.3.2. Instanciation de primitives
 - 1.3.2.1. Principe
 - 1.3.3. Constructive Solid Geometry (CSG)
 - 1.3.3.1. Principe
 - 1.3.3.2. Equation
 - 1.3.4. Surfaces isoparamétriques
 - 1.3.4.1. Principe
 - 1.3.4.2. Métaballes.
 - 1.3.5. Fractales
 - 1.3.5.1. Principe
 - 1.3.5.2. Cas particulier : ensemble de Mandelbrot
 - 1.3.5.3. Modélisation : Grammaire
 - 1.3.5.4. Terrains par subdivision de surface
 - 1.4. Outils de modélisation
 - 1.4.1. Opérateurs ensemblistes
 - 2. Algorithmes de visualisation
 - 3. Modèles d'éclairage
 - 4. Animation
 - Annexe
 - Logiciels de modélisation géométrique

1. Modélisation géométrique

1.1. Introduction

Le but est de fabriquer des objets

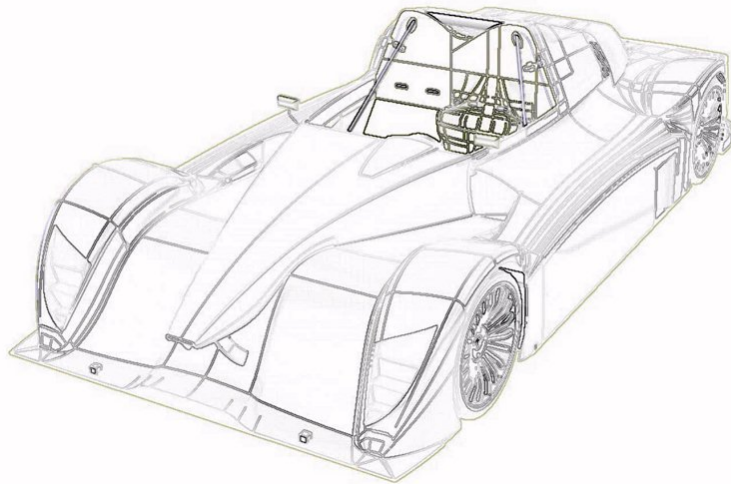
Modèle : simplification du monde réel.

1.2. Modèles explicites

Pas de calcul intermédiaire avant l'exploitation

- **B-Rep** (*Boundary REPresentations*)
 - Modèle à frontières / à enveloppe
 - Vectoriel : (2D) courbe - (3D) surface
- **Voxel** (*VOlume ELement map*)
 - Modèle à évolution spatiale
 - Map : (2D) pixel - (3D) voxel
- Contraintes de choix et d'utilisation :
 - Précision de la représentation
 - Domaine de représentation
 - Efficacité mémoire / temps de calcul

1.2.1. Modèle B-Rep



1.2.1.1. Principe

Un objet = Une liste (tableau, etc) de **carreaux de surface** (= *patch*).

- **patch** = restriction paramétrique d'une modélisation mathématique de surface
 - Modélisation mathématique de surface explicite : $z = f(x, y)$
 - Modélisation mathématique de surface implicite : $f(x, y, z) = 0$

- Modélisation mathématique de surface paramétrique (ex : Bézier /

$$\text{Hermite} \rightarrow \text{NURBS}) : \begin{cases} P_x = Q_x(u, v) \\ P_y = Q_y(u, v) \\ P_z = Q_z(u, v) \end{cases}$$

- Stockage : modèle point-face
- Visualisation : modèle point-arrête-face (table de points, table d'arrêtes, table de faces)

1.2.1.2. Modèle winged-edge

- **Point** : $x \ y \ z$
- **Arrêtes** :
 - $p_{\text{départ}} \ p_{\text{fin}}$
 - $f_{\text{gauche}} \ f_{\text{droite}}$
 - $a_{\text{gauche procédurale}} \ a_{\text{gauche suivante}}$
 - $a_{\text{droite procédurale}} \ a_{\text{droite suivante}}$
- **Face** : Un numéro d'arrête

1.2.1.3. Modèle half-edge

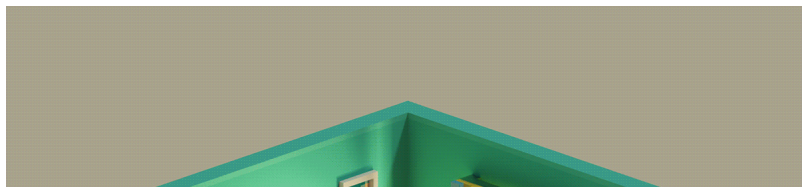
- **Point** : $x \ y \ z$
- **Arrêtes** :
 - $p_{\text{départ}} \ p_{\text{fin}}$
 - f
 - a_{jumelle}
 - $a_{\text{précédente}}$
 - a_{suivante}
- **Face** : Un numéro d'arrête

1.2.1.4. Subdivision d'un triangle

Procédure classique : diviser un triangle en 4 triangles.

Attention : On coupe les arrêtes, il faut donc subdiviser les faces voisines ou créer des arrêtes fictives

1.2.2. Les énumérations spatiales - Voxel





1.2.2.1. Principe

Enumérations spatiales → tableau 3D. (Version 3D des Pixmaps)

- Souvent issus de processus de mesure (scan)
 - Information de densité :
 - 0 si vide
 - 1 si plein
 - Possibilité de réaliser des opérations de filtrage (*"Telle gamme de densité correspond aux os, telle gamme de densité correspond aux muscles"*).
 - Structures de données : 1 tableau 3D $[i][j][k]$ + taille et quantité des voxels.

Domaine de représentation le plus complet mais par conséquent le plus lourd en mémoire.

- Permet de représenter des objets issus de **scan** ou des objets "**naturels**" (ex. nuages).
- Utilisé en impression 3D.

1.2.2.2. Formats de fichier

Exemple : Un objet de 1 m^3 avec des voxels de $1\text{ mm}^3 \Rightarrow 10^9$ voxels. C'est énorme.

- **DICOM (Digital Imaging and Communications in Medicine)** : Utilisé par les médecins pour archiver les scans.
 - Pile d'images 2D (*slices*)
- Formats pour artistes → spécifiques à chaque logiciel (ex. Magivoxel).

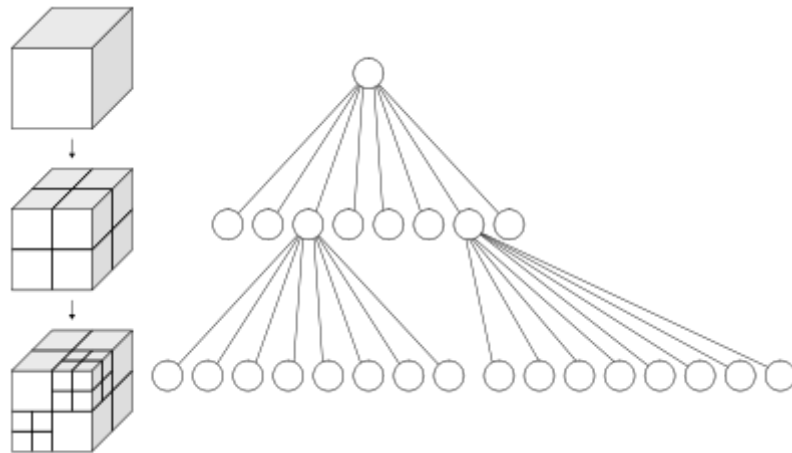
1.2.2.3. Jeux

- **Minecraft** (MCEdit)

1.3. Modèles implicites

Calcul intermédiaire avant l'exploitation

1.3.1. Octree



1.3.1.1. Principe

Octree = arbre de subdivision de l'espace (d'ordre **8**).

- Version 2D : **Quadtree** (d'ordre **4**).
- **Noeud** (cube) → plein / vide / complexe. Si complexe, on subdivise.

Format compact des énumérations spatiales.

- Utilise la **cohérence spatiale** des objets.

1.3.2. Instanciation de primitives



1.3.2.1. Principe

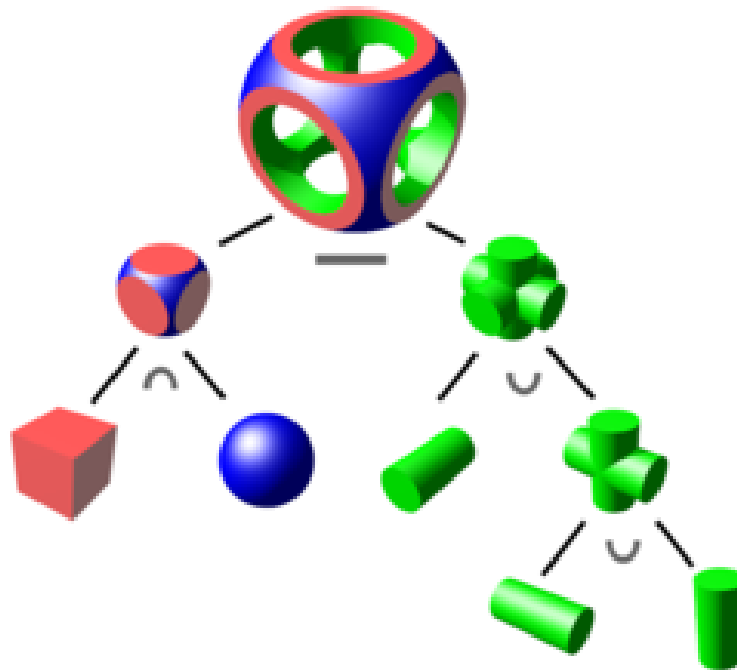
Instanciation = répétition d'un objet.

Primitive = Catalogue de composants (ex. cube, sphère, cylindre, etc).

Modèle = liste de primitives paramétrées.

- **Paramètres** : transformations géométriques et autres paramètres (ex. nombre de trous dans un cube).
- **Domaine de représentation** : dépend du catalogue.
- **Précision** : dépend du catalogue.
- **Encombrement** : réduit.

1.3.3. Constructive Solid Geometry (CSG)

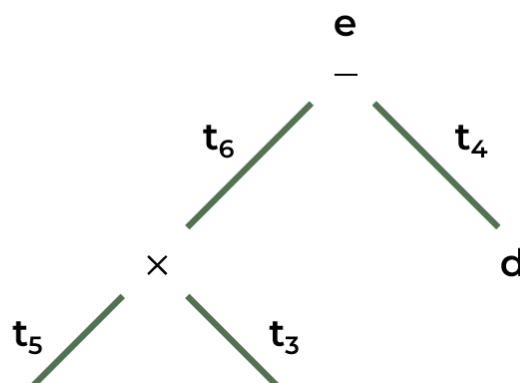


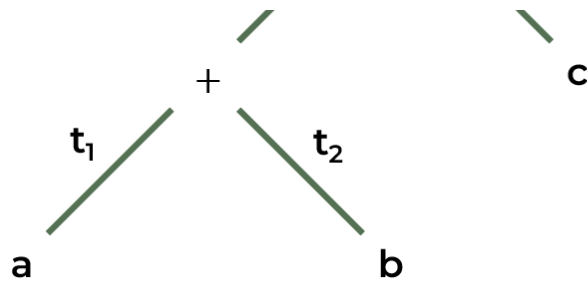
1.3.3.1. Principe

Modèle : Expression de calcul d'arbre d'ordre 2.

- **Racine** : Objet complet.
- **Feuilles** : primitives (plan/boîte, sphère, cylindre, cone).
- **Noeud** : opération (union, intersection, différence, etc).

1.3.3.2. Equation

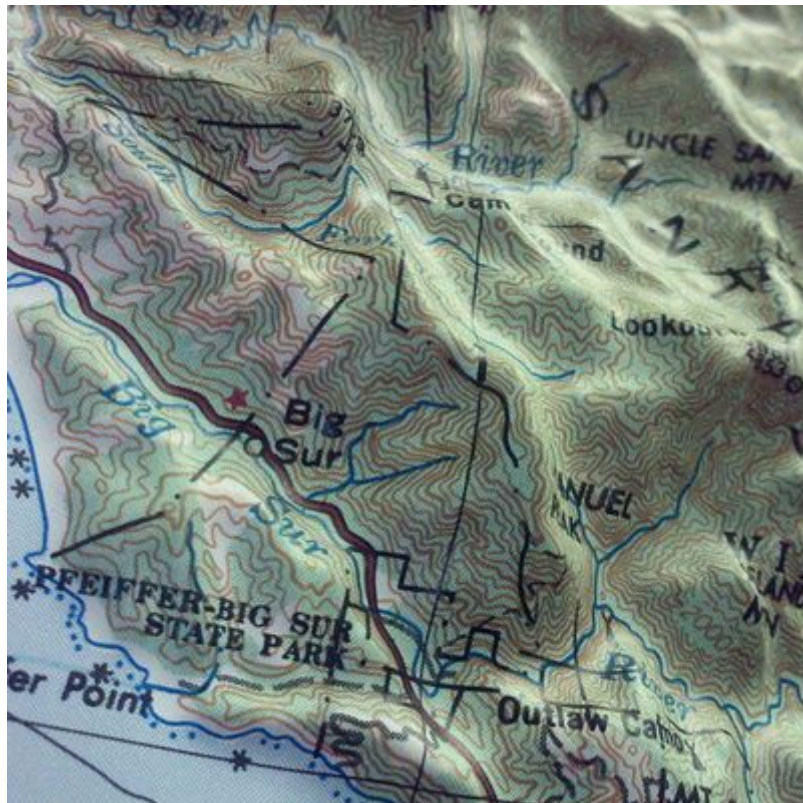




$$e = t_6(t_5(t_1(a) + t_2(b)) \times t_3(c)) - t_4(d)$$

Avec t_i = transformation géométrique.

1.3.4. Surfaces isoparamétriques



1.3.4.1. Principe

Surfaces isoparamétriques = surfaces paramétriques.

- Surface de valeur : $f(x, y, z) = \text{seuil}$
 - Ex. par rapport à un point : $f(x, y, z) = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} = \text{seuil}$

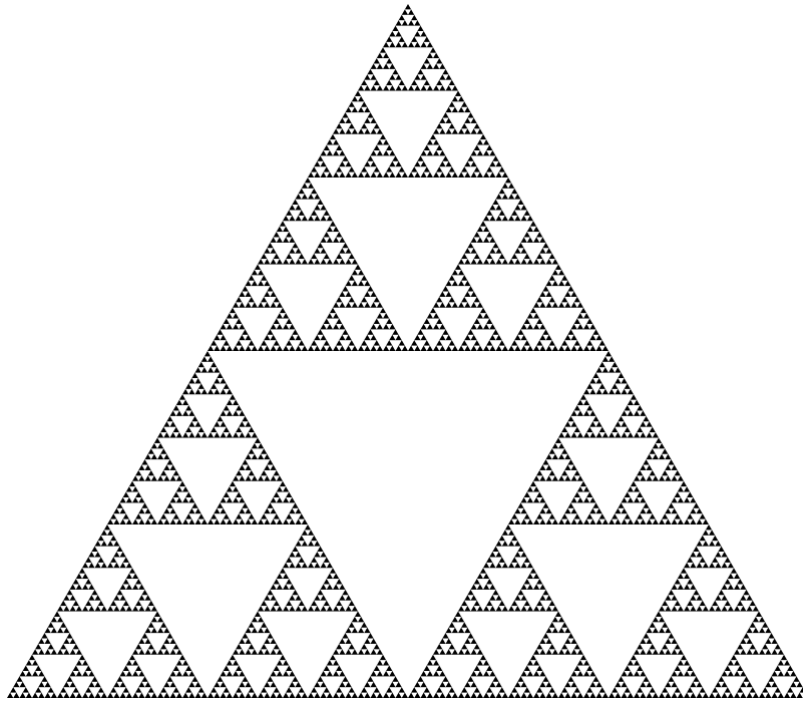
1.3.4.2. Métaballes.

- **Métaballe** (*blob*) : permet de créer des formes organiques ou de représenter des fluides.

Fonctions génératrices (distance) : $\sum_n f_n(x, y, z) < \text{seuil}$

Métablle + Valeur aléatoire + Seuil.

1.3.5. Fractales



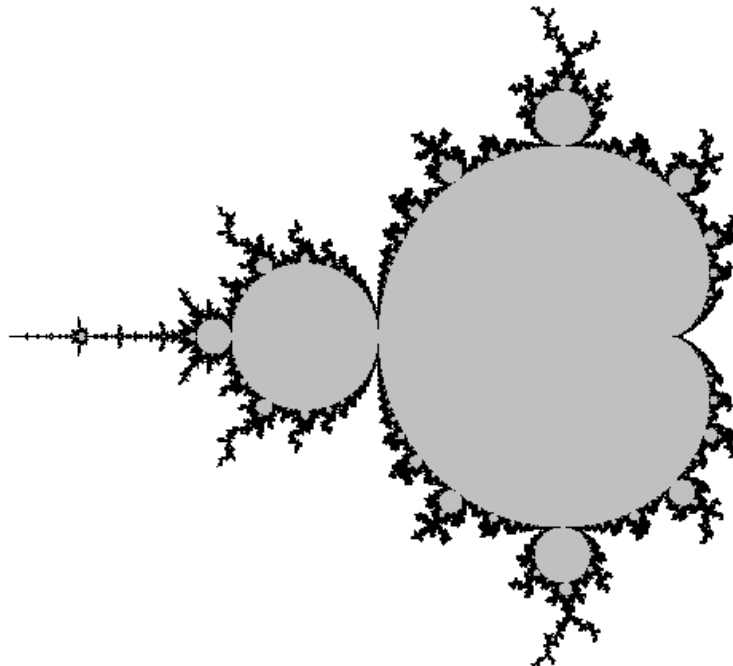
1.3.5.1. Principe

Fractale = objet dont la structure est répétitive à différentes échelles.

Modèle = objet initial + procédure de transformation.

- Volume nul, surface infinie.
- Construction itérative/réursive.

1.3.5.2. Cas particulier : ensemble de Mandelbrot



- Etude de la suite complexe $\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$ avec c l'objet initial.
- Divergent si $\exists i \in \mathbb{N}$ tel que $|z_i| \geq 2$.

1.3.5.3. Modélisation : Grammaire

1. alphabet
2. règles de transformation

Exemple :

1. alphabet :

- $\Sigma = \{A, B, [,]\}$

2. règles de transformation :

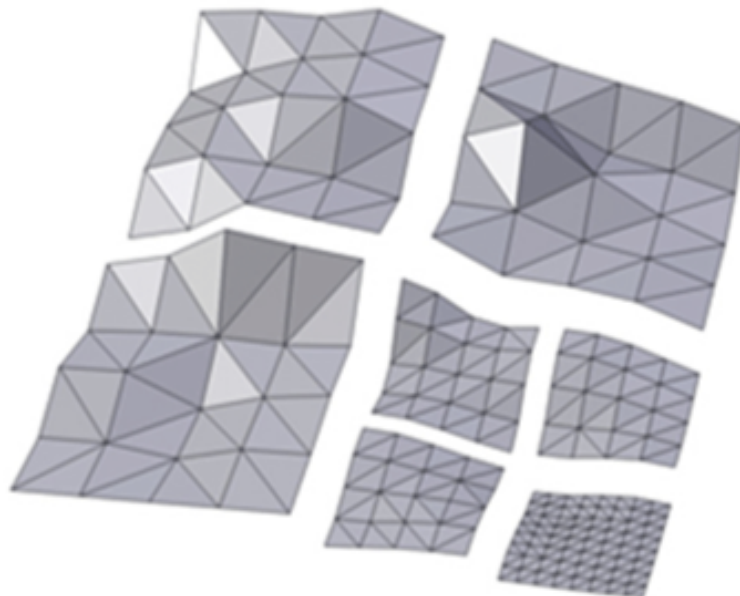
- $A \rightarrow AB$
- $B \rightarrow A[AB]A[AB]$
- $[\rightarrow [BB$
- $] \rightarrow]AB$

3. exemple de procédé :

1. $ABBA$
2. $ABA[AB]A[AB]A[AB]A[AB]AB$
3. $ABA[AB]A[AB]AB[BB...$
4. ...

1.3.5.4. Terrains par subdivision de surface

Subdivision de surface + hauteur aléatoire à chaque division.



- Utilisé pour créer des terrains de synthèse.
- Procédé que nous pouvons utiliser pour obtenir en 3D un **B-Rep** ou en 2D une image en niveau de gris.

1.4. Outils de modélisation

1.4.1. Opérateurs ensemblistes

Opérations sur des volumes.

- $+$: union
- \times : intersection
- $-$: différence

$$\circ a - b = a \times -b$$

Intérieur et extérieur d'un objet.

- **Voxel** : pas de problème.
- **B-Rep** : calculs d'intersection sur les enveloppes.

2. Algorithmes de visualisation

3. Modèles d'éclairage

4. Animation

Annexe

Logiciels de modélisation géométrique

- **AutoCAD** : (CAD \rightarrow Computer Aided Design) dessin technique (2D) est devenu **3DS** (3D artistique)
- **Maya** (~3 500€)
- **Blender** (gratuit)