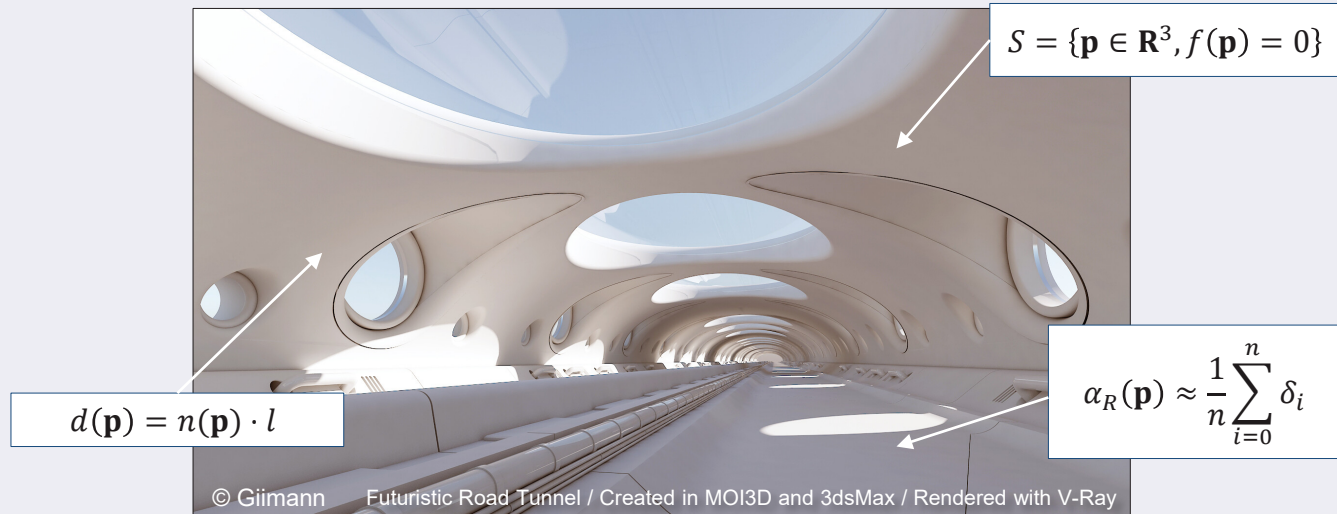


Computer Graphics

From mathematics ...



... to the screen

E. Galin
Université Lyon 1

Computer Graphics

Overview

Deformations

Curves

Surfaces

Introduction

Classification

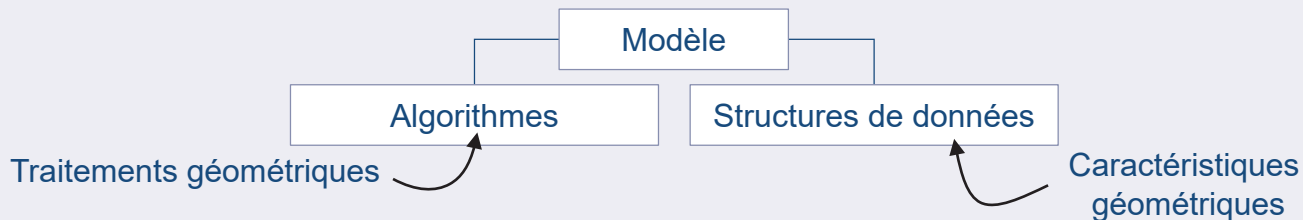
Surfaces

Volumes

Conclusion

Modélisation géométrique

Description de la forme et des propriétés géométriques et topologiques de l'objet
Modèles volumiques ou surfaciques



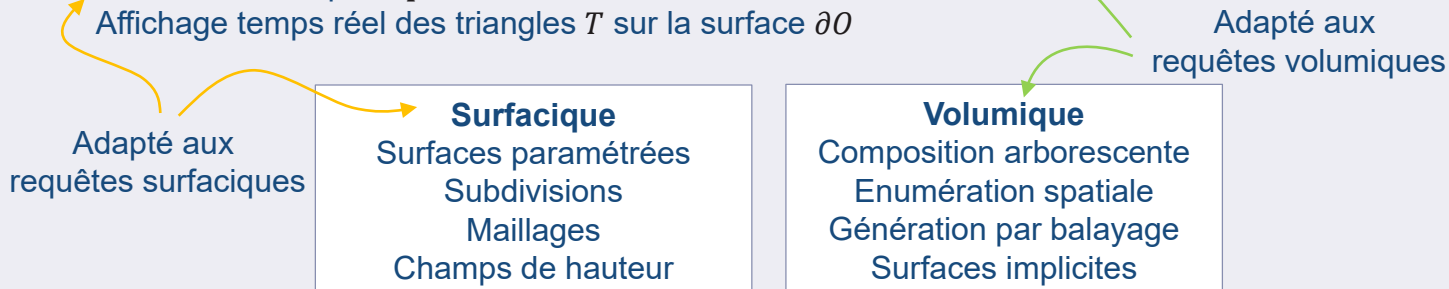
Traitements géométriques élémentaires

Intersection avec une droite $\Delta \cap O$ ou entre objets $O \cap O$

Position d'un point par rapport au modèle $p \in O$

Recherche d'un point p sur la surface ∂O

Affichage temps réel des triangles T sur la surface ∂O



Surfacique
Surfaces paramétrées
Subdivisions
Maillages
Champs de hauteur

Volumique
Composition arborescente
Enumération spatiale
Génération par balayage
Surfaces implicites

Il n'existe **pas** de modèle universel



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Computer Graphics

Surface models

Surfaces paramétrées

Classification

Surfaces

Volumes

Conclusion

Définition

On définit $S = \{\mathbf{p}(u, v), (u, v) \in \Omega\}$

Différents types de surfaces selon les fonctions $\mathbf{p}(u, v)$

Construction

$\mathbf{p}(u, v)$ est défini par des polynômes

Construction à partir de points de contrôle \mathbf{p}_{ij}

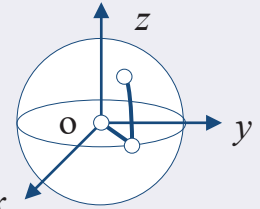
$$\mathbf{p}(u, v) = \sum_{(i,j) \in [0,n]^2} B_i^n(u) B_j^n(v) \mathbf{p}_{ij}$$

$$\mathbf{p}(u, v) = (u^3 \ u^2 \ u \ 1) \mathbf{M} [\mathbf{p}_{ij}] \mathbf{M} (v^3 \ v^2 \ v \ 1)^t$$

Propriétés

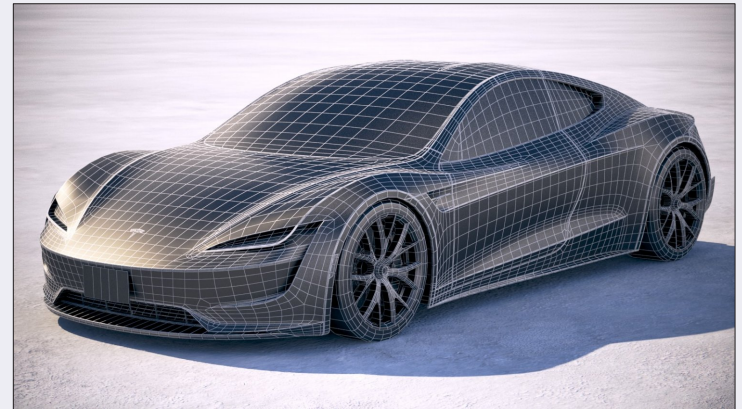
Régularité des carreaux

Conversion en maillage trivial



Sphère paramétrée

$$\mathbf{p}(\theta, \phi) = (\cos \theta \cos \phi, \sin \theta \cos \phi, \sin \phi)$$
$$(\theta, \phi) \in [0, 2\pi] \times [-\pi/2, \pi/2]$$



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalin>

Maillages géométriques

Classification

Surfaces

Volumes

Conclusion

Maillages triangulaires

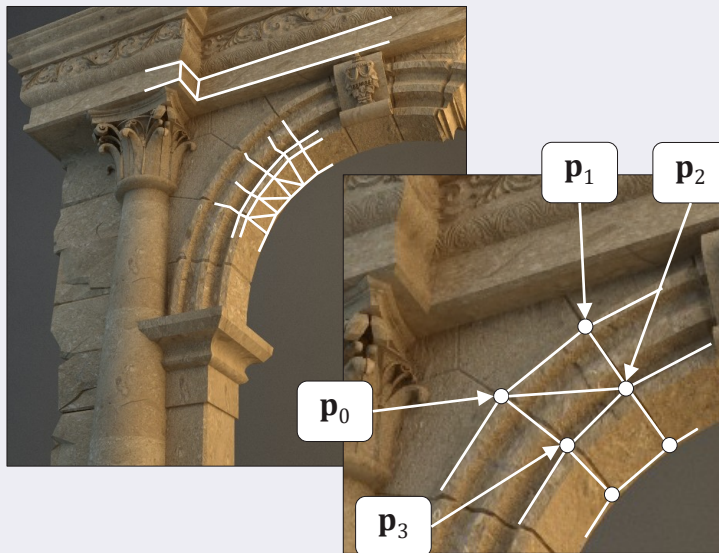
Géométrie G : sommets (normales)

Topologie T : connectivité entre sommets, arêtes, faces

Structure minimale

Sommets connectés en triangles : T fixe

Triplets $\{a, b, c\}$ ou double triplet $\{a, n_a, b, n_b, c, n_c\}$ pour chaque triangle



Géométrie G

p_0

p_1

p_2

p_3

Topologie T

0 3 2

1 0 2

```
class Mesh {  
    std::vector<Vector> p ;           // Vertexes  
    std::vector<int> t ;             // Indexes  
};
```

Topologie

Classification

Surfaces

Volumes

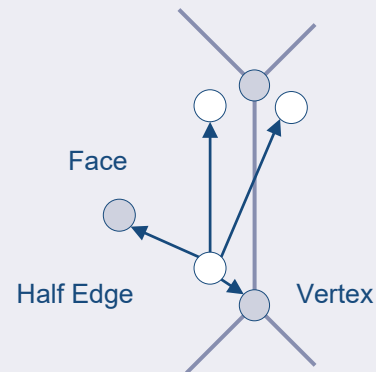
Conclusion

Structure de données

Arête ailée (winged edge), demi arête (half edge)
Opérateurs garantissant une topologie cohérente

```
class HalfEdge {  
    Vertex* vertex; // Sommet de la demi-arête  
    Edge* pair; // Demi arête opposée  
    Face* face; // Face  
    HalfEdge* next; // Demi arête suivante  
};  
class Vertex {  
    Vector p; // Coordonnées  
    HalfEdge* edge; // Une des demi arêtes  
};  
class Face {  
    HalfEdge* edge; // Une des demi arêtes  
};
```

$$V + F - E = 2 (G + 1)$$



Requêtes

Itération sur les demi arêtes adjacentes à une face, parcours des arêtes d'une face

```
HalfEdge* edge = face → edge;  
do {  
    // do something with edge  
    // or face  
    edge = edge → next;  
} while (edge != face → edge);
```

```
HalfEdge* edge = vert → edge;  
do {  
    // do something with edge, edge->pair  
    // or edge->face  
    edge = edge → pair → next;  
} while (edge != vert → edge);
```



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Maillages faible résolution

Classification

Surfaces

Volumes

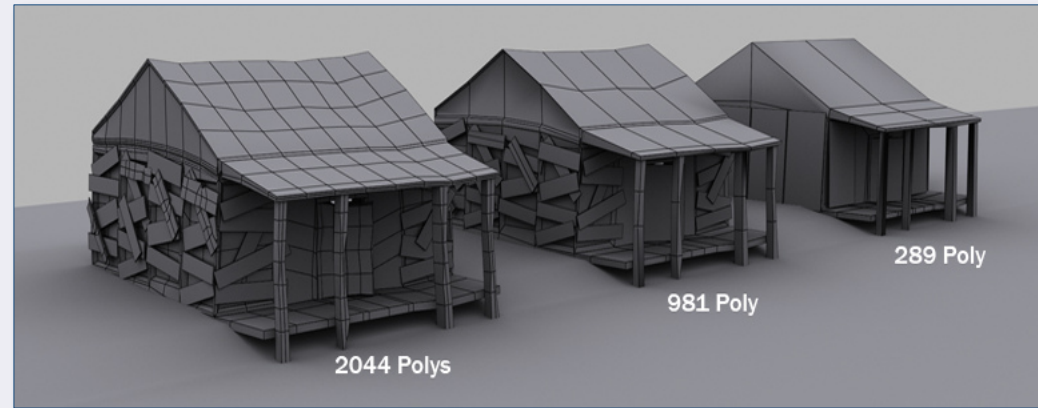
Conclusion

Fondamentaux

Rupture des frontières entre modélisation et texture

Maillages non conformes, facettes sans raccordements

Utilisation intensive de *billboards*



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Surfaces de subdivision

Classification

Surfaces

Volumes

Conclusion

Définition

Surface limite S : subdivisions successives d'un maillage de contrôle C
Schémas de raffinement de maillage

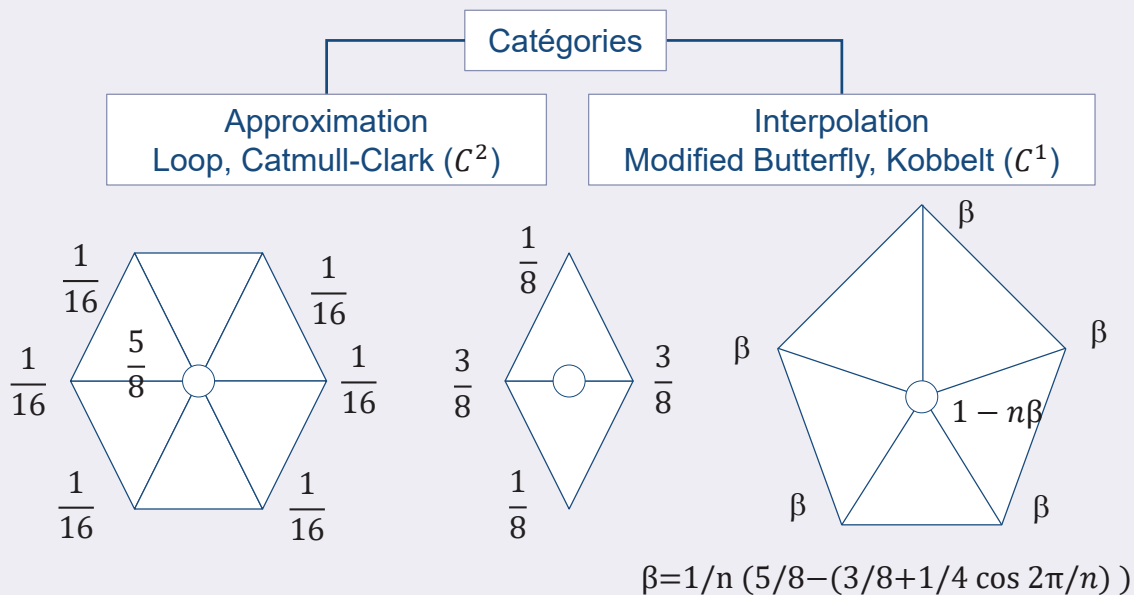
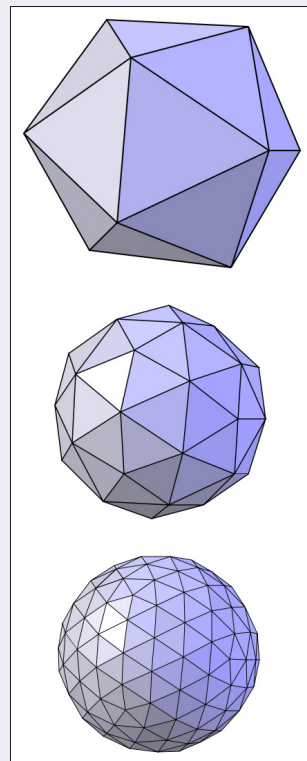


Schéma de Loop



E. Catmull and J. Clark. Recursively generated B-Spline surfaces on arbitrary topological meshes. *Computer Aided Design*, **10**(6):350–355, 1978.



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Surfaces de subdivision

Classification

Surfaces

Volumes

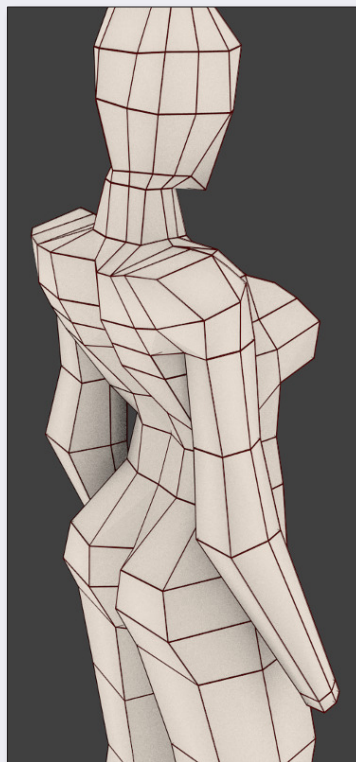
Conclusion

Modélisation

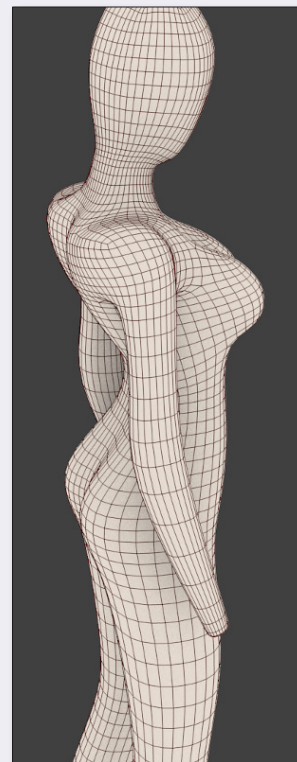
Création de formes lisses (organiques) à partir d'un squelette de contrôle

Animation

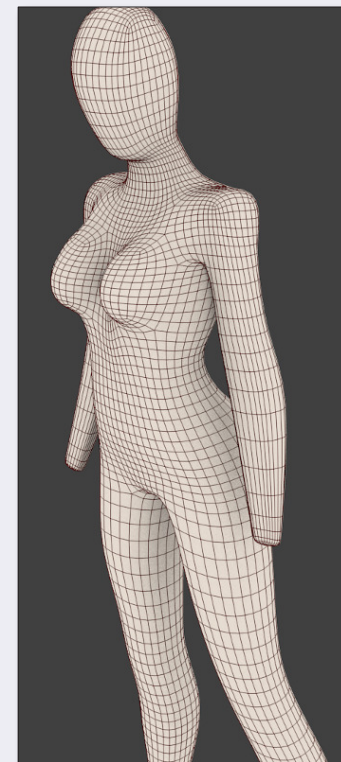
Il suffit d'animer les points du maillage grossier



Maillage grossier



Subdivision de niveau 2



Comparaison

Critères

Opérations sur I

Classification

Surfaces

Volumes

Conclusion

Summary



Feature	Polygonal Mesh	Implicit Surface	Parametric Surface	Subdivision Surface
Accurate	No	Yes	Yes	Yes
Concise	No	Yes	Yes	Yes
Intuitive specification	No	No	Yes	No
Local support	Yes	No	Yes	Yes
Affine invariant	Yes	Yes	Yes	Yes
Arbitrary topology	Yes	No	No	Yes
Guaranteed continuity	No	Yes	Yes	Yes
Natural parameterization	No	No	Yes	No
Efficient display	Yes	No	Yes	Yes
Efficient intersections	No	Yes	No	No



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Computer Graphics

Volumetric models

Enumération spatiale

Classification

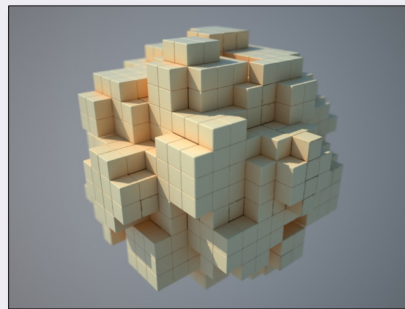
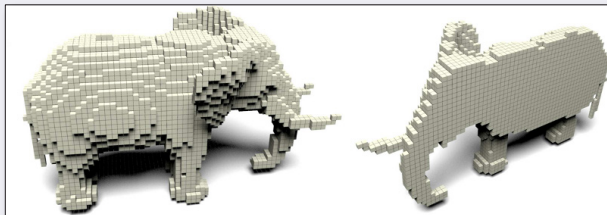
Surfaces

Volumes

Conclusion

Structure régulière

Subdivision de l'espace en grille régulière,
Stockage du type de matériau dans les voxels



Structure coûteuse en mémoire : $O(n^3)$ cellules et accès efficace en $O(1)$
Traitements locaux efficaces, coûteux sur l'ensemble du modèle

```
class Voxel
{
protected:
    int n ;
    unsigned char* array ;
    Vector a, b ; // End vertices of cubic domain
public:
    // ...
    int Index (int, int, int) const ;
    unsigned char At(const Vector&) const ;
};
```

```
int Voxel::Index (int x,int y, int z) const {
    return x*n*n + y*n + z*n ;
}
unsigned char Voxel::At (const Vector& p) const {
    // Real coordinates inside the cubic region [0,1]
    double x=(p[0]-a[0])/(b[0]-a[0]); // Same for y , z
    int ix=int (x*n); // Same for y , z

    // Should check range here
    return array[ Index (ix,iy,iz) ];
}
```



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalain>

Enumération spatiale

Classification

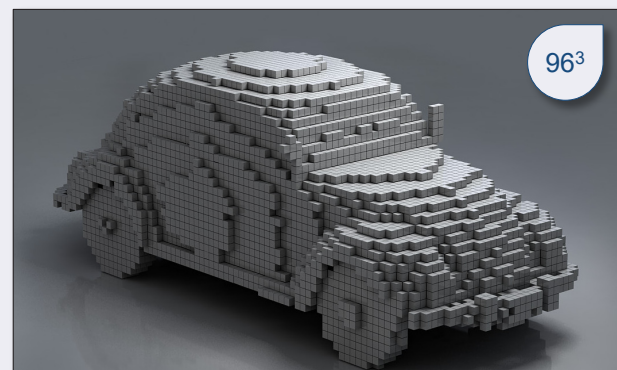
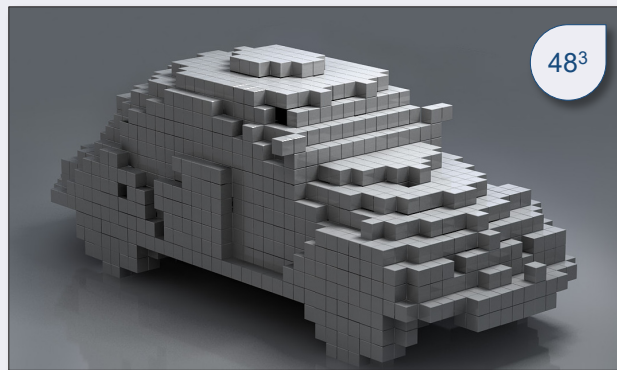
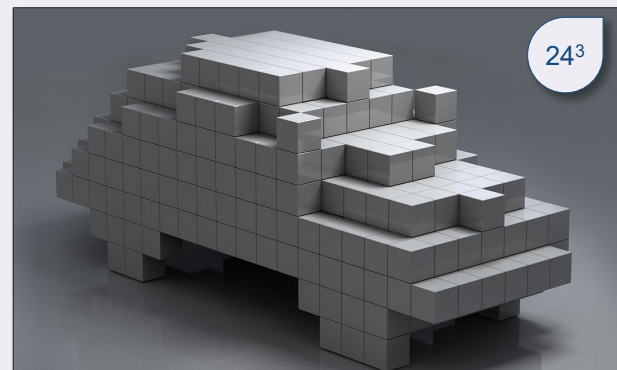
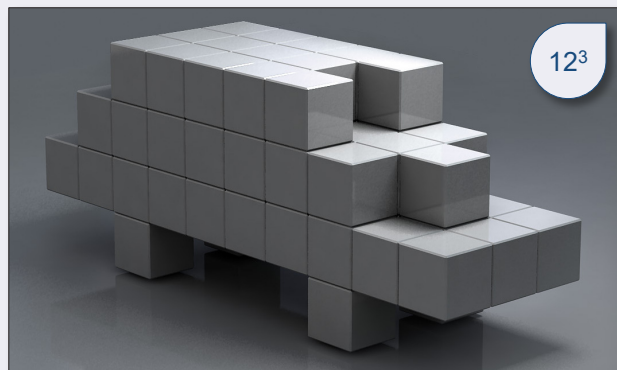
Surfaces

Volumes

Conclusion

Précision

Même objet à différentes résolutions



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Enumération spatiale

Classification

Surfaces

Volumes

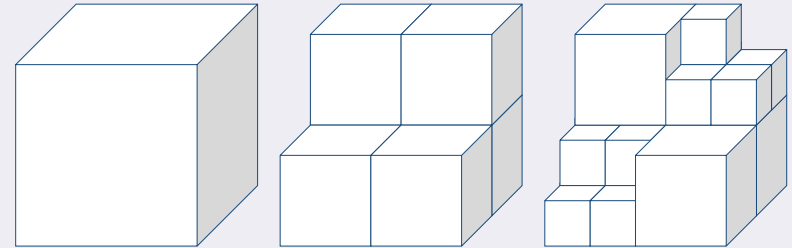
Conclusion

Améliorations

Compression des données
Algorithme run length encoding

Structure adaptative

Décomposition de l'espace en
sparse voxel octrees



Redondances et motifs fréquents

Directed acyclic graphs permettent de gérer des résolutions jusqu'à $[32 - 128k]^3$



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

S. Laine, T. Karras. Efficient sparse voxel octrees. *Symposium on Interactive 3D Graphics and Games*, 55–63, 2010.
S. Laine, T. Karras. Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics* **17**(8), 2011.
V. Kampe, E. Sintorn, U. Assarsson. High Resolution Sparse Voxel DAGs. *Proceeding of Siggraph*, 2013.

Construction arborescente

Classification

Surfaces

Volumes

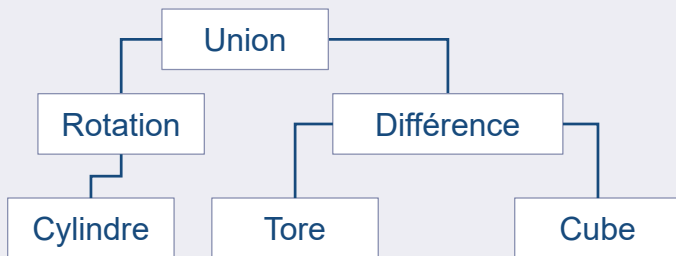
Conclusion

Structure

Arbre de construction combinant des primitives simples

Opérateur booléens

Transformation affines



Requêtes

Position d'un point par rapport au modèle $\mathbf{p} \in O$ par descente récursive de l'arbre jusqu'aux feuilles

Sphère $\mathbf{p} \in S$ si $\|\mathbf{p} - \mathbf{c}\| < r$

Test terminal

Union $\mathbf{p} \in A \cup B$ si $\mathbf{p} \in A$ et $\mathbf{p} \in B$

Récursion



eric.galin@liris.cnrs.fr

http://liris.cnrs.fr/~egalin

Construction arborescente

Classification

Surfaces

Volumes

Conclusion

Implémentation

Hiérarchie de classes avec polymorphisme

Chaque classe implémente ses méthodes spécifiques

```
class Node
{
public:
    virtual int Inside(const Vector&) const = 0;
};

class Sphere : public Node
{
protected:
    Vector c; // Center.
    double r; // Radius.
public:
    // ...
    int Inside(const Vector&) const;
};

class Union : public Node
{
protected:
    Node* a; // Left child
    Node *b; // Right child
public:
    // ...
};
```

```
int Sphere::Inside(const Vector& p) const
{
    // Center
    Vector n=c-ray[0];

    // Distance to center
    double pc=Norm (p-c);

    if ( pc<r) return 1;
    else return 0;
}
```

```
int Union::Inside(const Vector& p) const
{
    // Check children nodes
    if (a->Inside(p)) return 1;
    if (b->Inside(p)) return 1;

    return 0;
}
```



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Surfaces Implicites

Classification

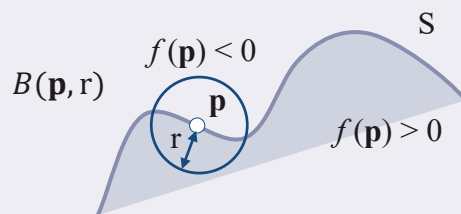
Surfaces

Volumes

Conclusion

Définition

On définit $S = \{\mathbf{p} \in \mathbf{R}^3 \mid f(\mathbf{p}) = 0\}$



Propriétés

Modélisation de formes lisses (raccordement)

Objets de topologie variable

Visualisation complexe

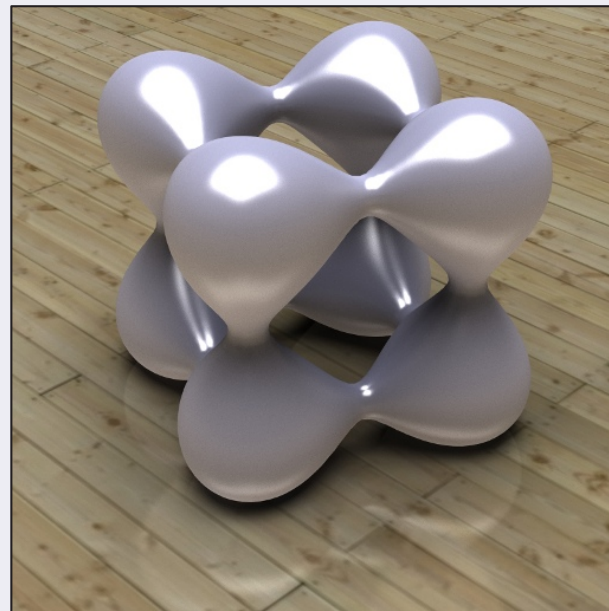
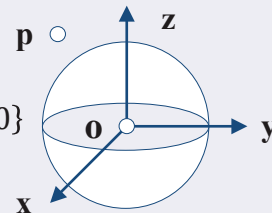
Catégories

Surfaces implicites à squelettes

Surfaces variationnelles (nuages de points)

Sphère implicite

$$S = \{(x, y, z) \in \mathbf{R}^3 \mid 1 - x^2 - y^2 - z^2 = 0\}$$



B. Wyvill, A. Guy and E. Galin. Extending the CSG Tree (Warping, Blending and Boolean Operations in an Implicit Surface Modeling System). *Computer Graphics Forum*, 18 (2), 149-158, 1999.



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Comparaison

Classification

Surfaces

Volumes

Conclusion

Modèles

On définit $S = \{\mathbf{p} \in \mathbf{R}^3 \mid f(\mathbf{p}) = 0\}$

Memory

Shapes

Speed

Animation

Voxels

Implicits

Summary



	Voxels	Octree	BSP	CSG
Accurate	No	No	Some	Some
Concise	No	No	No	Yes
Affine invariant	No	No	Yes	Yes
Easy acquisition	Some	Some	No	Some
Guaranteed validity	Yes	Yes	Yes	No
Efficient	No	No	No	No

Summary

Feature	Polygonal Mesh
Accurate	No
Concise	No
Intuitive specification	No
Local support	Yes
Affine invariant	Yes



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Modeling

20 septem

Computer Graphics

Conclusion

Conclusion

Classification

Surfaces

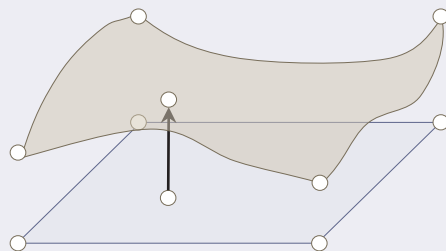
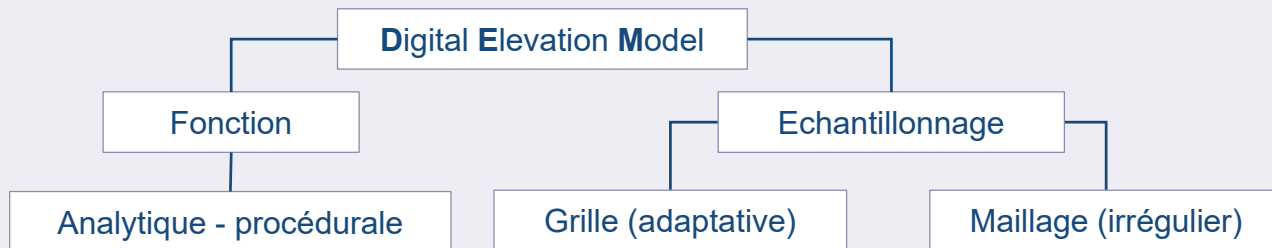
Volumes

Conclusion

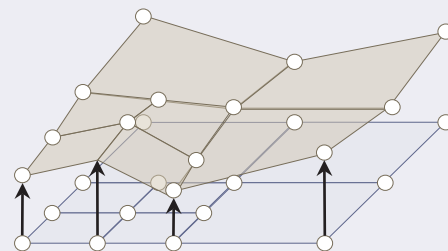
Choix d'un modèle

Il n'existe pas de modèle universel

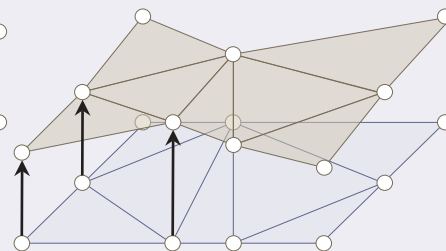
Stratégies pour représenter un terrain



Fonction ou procédure



Grille (régulière, adaptative)



Triangle Irregular Network