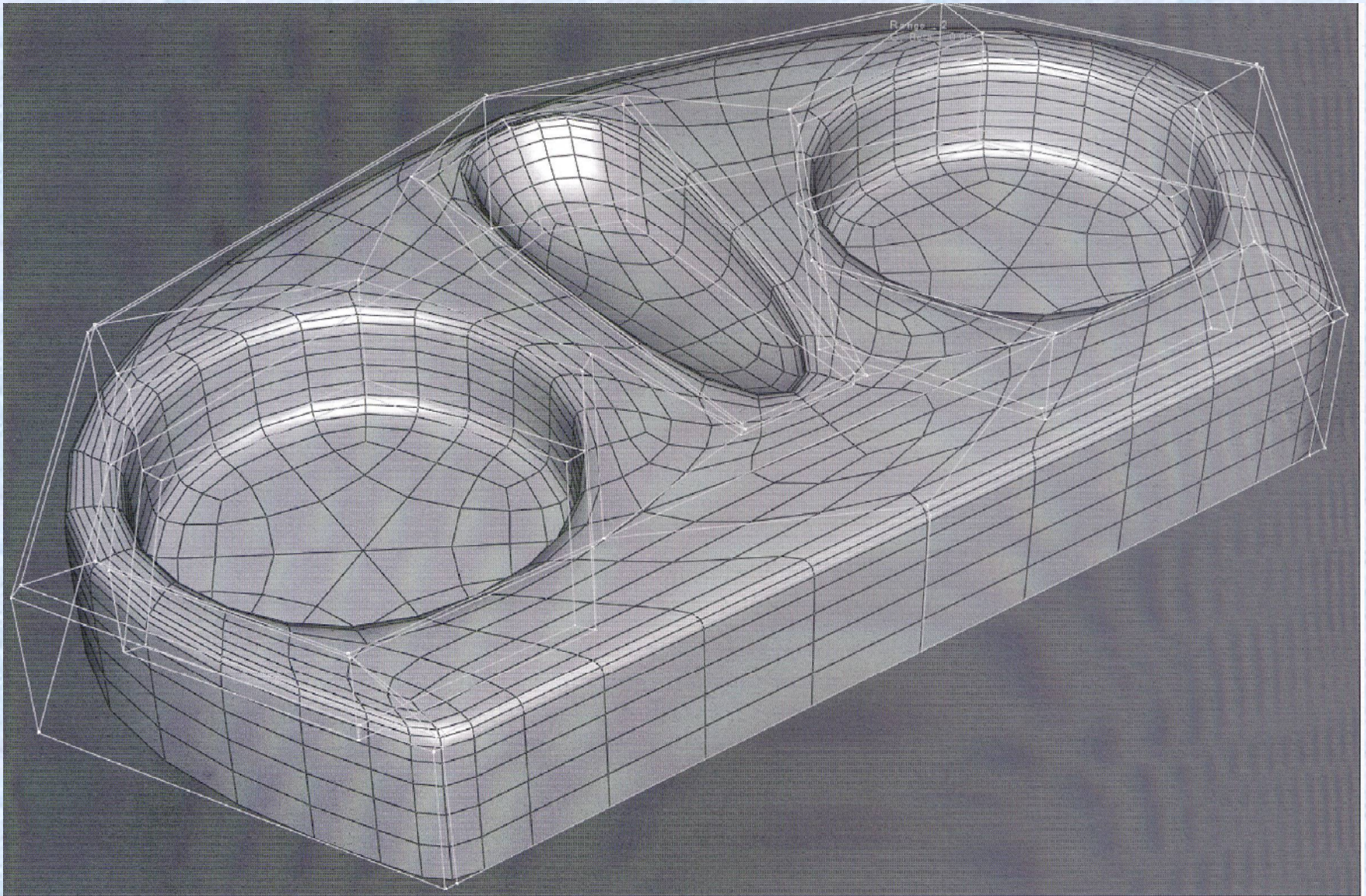


# Modélisation solide



Tiré de Olivier Drion, *Amapi 7 Ateliers graphiques*. Eyrolles, 2003, p. 141.

# Représentation et construction d'objets

## OBJECTIFS

- Énumérer les principaux modes de représentation d'objets géométriques.
- Décrire les principales propriétés de ces modes de représentation d'objets
- Montrer leur potentiel de modélisation.

## PLAN DU CHAPITRE

- ★ Introduction
- ★ Description de l'enveloppe de l'objet → {  
Modèle en fil de fer  
Représentation à partir de surfaces frontières
- ★ Familles d'objets paramétrisés
- ★ Modèles de composition → {  
Modèles basés sur la notion de demi-espaces  
Modèles C.S.G.
- ★ Modèles de décomposition → {  
Énumération spatiale  
QUADTREE, OCTREE

# INTRODUCTION

-  Un modèle d'un objet est une représentation “ idéalisée ” ou simplifiée d'un objet permettant de l'observer plus facilement.
-  La construction d'un modèle pour représenter la structure géométrique d'un objet est intéressante à plusieurs points de vue :
  -  certaines caractéristiques du modèle peuvent être étudiées plus facilement que celles de l'objet lui-même;
  -  l'objet peut ne pas exister;
  -  l'objet ne peut être observé directement;
  -  l'objet ne peut être observé sans engendrer des coûts déraisonnables ou sans contrôle de l'expérience.

# PROBLÈMES RENCONTRÉS DANS LE PROCESSUS DE MODÉLISATION

- ★ Besoin d'un modèle **complet** :

un modèle qui offre une description complète de la géométrie d'un objet, permettant de répondre aux différentes questions géométriques pertinentes.

- ★ Construction d'un modèle **correct** :

nous devons être capable de détecter et de corriger des anomalies lors de la construction du modèle d'un objet.

Ex. : Un modèle d'objet basé sur des facettes polygonaux convexes.

Nous devons nous assurer que cette restriction est toujours satisfaite après une opération quelconque.

- ★ **Complexité** de l'objet à modéliser :

- ➡ La représentation sous forme polyédrique d'une main est très difficile.

- ➡ L'augmentation du potentiel de modélisation afin de représenter des formes davantage complexes entraîne des problèmes lors des calculs géométriques.

# Objectifs à atteindre pour représenter un objet

- ✧ précision de l'image;
- ✧ possibilité de visualiser, d'analyser ou de manipuler l'objet selon n'importe quelle direction d'observation;
- ✧ capacité de recueillir toutes les informations pertinentes décrivant l'objet nécessaires à chaque application;
- ✧ représentation non ambiguë de l'objet;
- ✧ réduction du nombre de paramètres décrivant l'objet;
- ✧ simplification de calcul de certaines mesures;
- ✧ approche systématique de construction d'objets à partir de formes connues.

Les objectifs précédents ne sont pas tous atteints à travers les modèles que nous verrons; des choix doivent être faits.

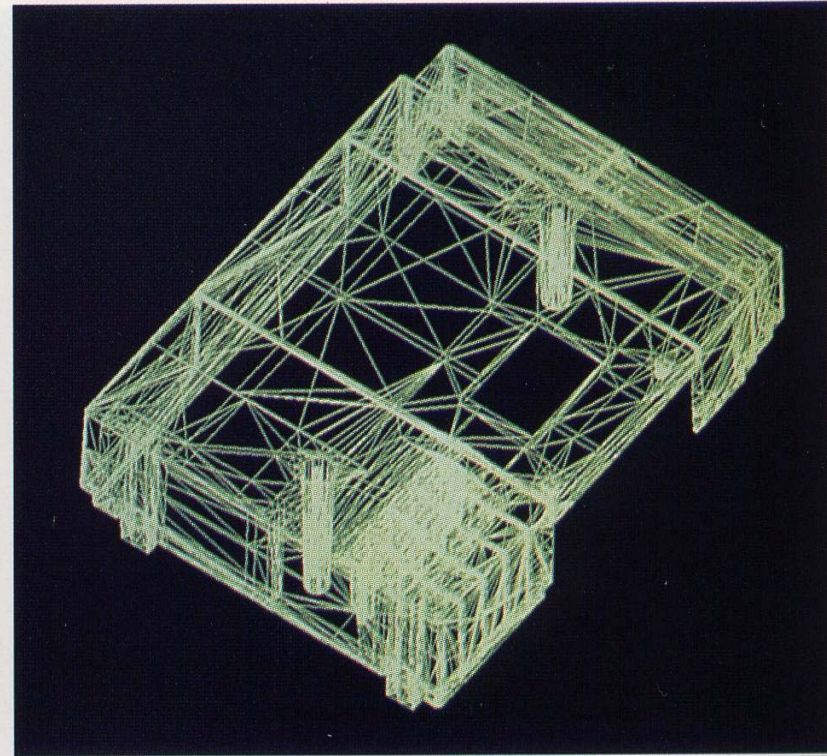
# Modèles frontières

(connus aussi sous le nom de « b-reps »)

Il s'agit d'un mode de représentation indirect d'un solide en décrivant l'enveloppe du solide à l'aide de ses frontières : sommets, arêtes et facettes.

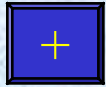
Modèles simplifiés : les modèles en « fil de fer »

- Ils permettent de représenter uniquement le contour des objets.
- Les objets sont représentés à partir d'un ensemble de segments de droite reliés éventuellement par leurs extrémités.
- Les extrémités des segments de droite et les liens existant entre eux sont stockés.

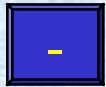


**FIGURE 19.5** Looking at the STL file with the 3D Viewer<sup>®</sup> from 3D Systems. (Courtesy of Protogenic, Inc., and EFI Electronics Corporation.)

# Modèles en « fil de fer »

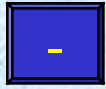


- Ce modèle est simple : facilite les transformations de base et de visualisation.
- Permet à peu de frais d'avoir une représentation géométrique globale de l'objet.

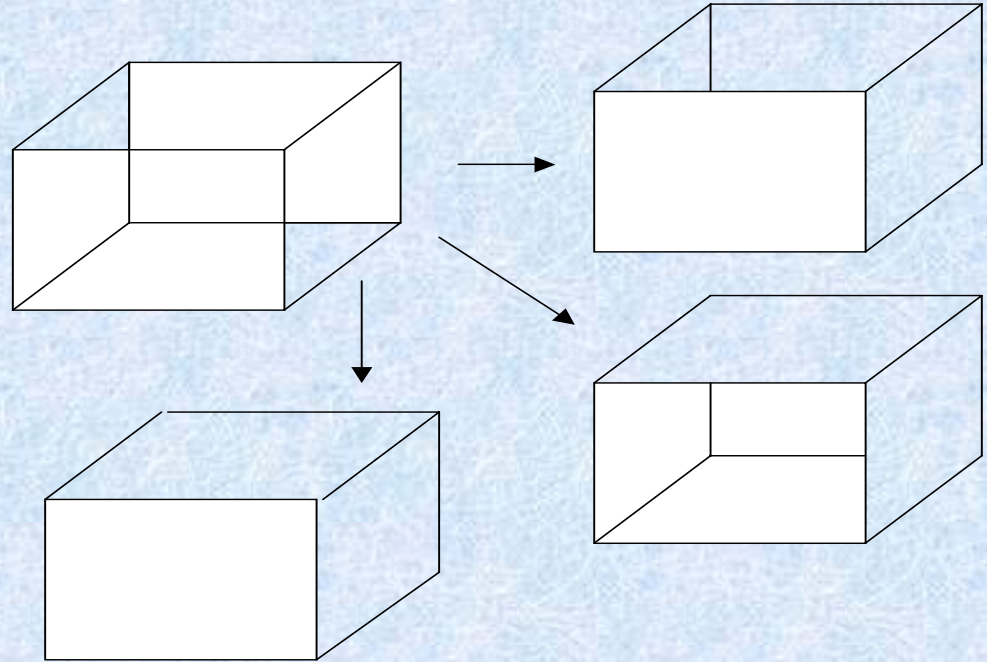


- Capacité de modélisation très limitée : faible degré de réalisme.
- Le calcul de certaines mesures de l'objet peut être difficile ou impossible par manque d'informations sur l'objet.
- La quantité imposante de données nécessaires pour décrire l'objet.

# Modèles en « fil de fer »

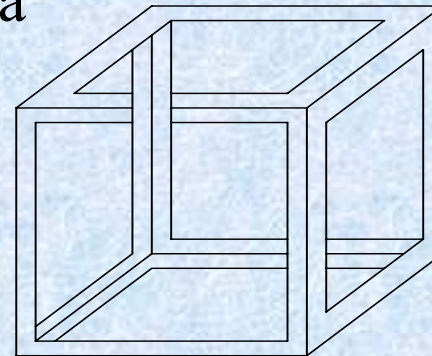


- Il peut y avoir ambiguïté
  - dans l'interprétation de la représentation
  - ou
  - donner lieu à un modèle impossible.



Plusieurs interprétations d'un même objet.

- Ces ambiguïtés dans l'interprétation de la représentation rendent difficiles la résolution du problème d'élimination des lignes cachées.

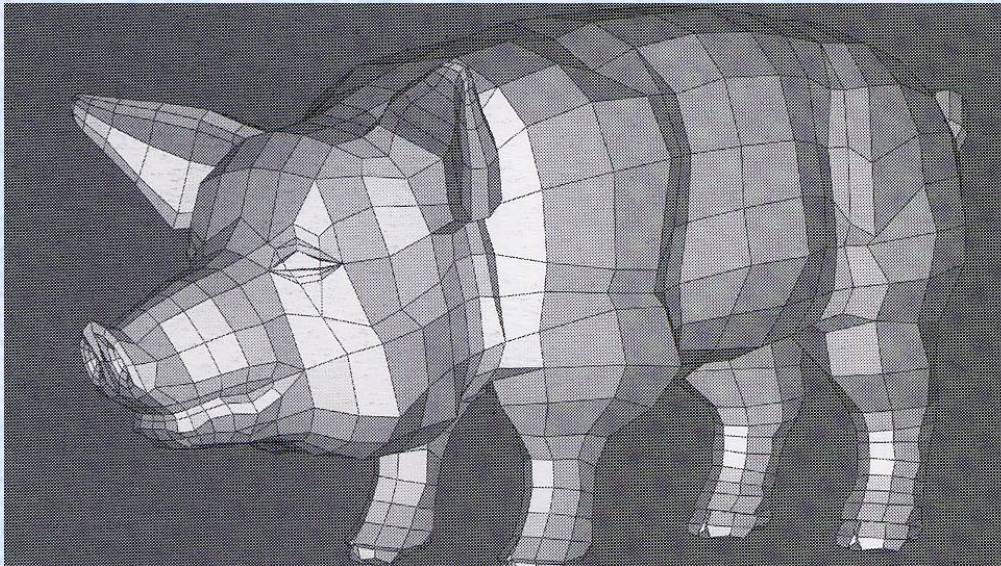


Modèle impossible

# Modèles frontières

## Les modèles définis à partir de surfaces frontières

- Il s'agit de décrire l'enveloppe de l'objet à partir de la description de ses surfaces frontières : surfaces courbes, frontières planes, polygonales, polygonales convexes, triangulaires.
- La plupart du temps, chaque surface frontière courbe est divisée en facettes  
Ex. : des facettes polygonales  
et chaque facette est représentée par l'ensemble des arêtes et des sommets qui la délimitent.



# Modèles définis à partir de surfaces frontières

## Conditions à respecter :

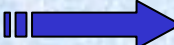
- L'ensemble des facettes forme une figure fermée.
- Les facettes ne s'interceptent pas sauf à des arêtes ou sommets communs.
- Les surfaces frontières ne s'interceptent pas elles-mêmes.
- • Le nombre de facettes peut être élevé (approximation polygonale).
- Le modèle ne fournit pas d'information quant à l'intérieur de l'objet.
- L'évaluation d'une mesure de l'objet peut être difficile à faire.
- Les conditions précédentes sont difficiles à vérifier.

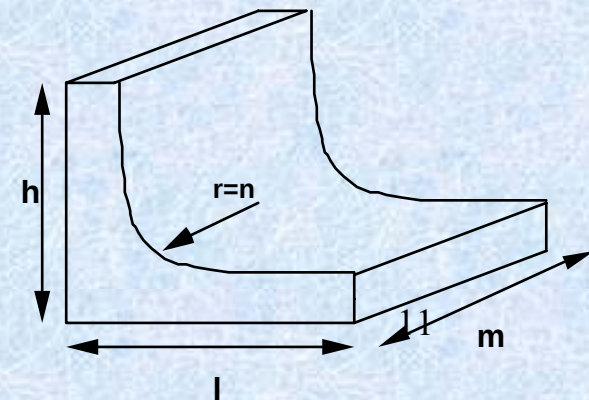


- Ils ne sont pas ambigus comme cela pouvait être le cas précédemment.
- Le modèle permet de résoudre le problème d'élimination des parties cachées.
- Il permet d'appliquer un modèle d'illumination et/ou de génération de texture.

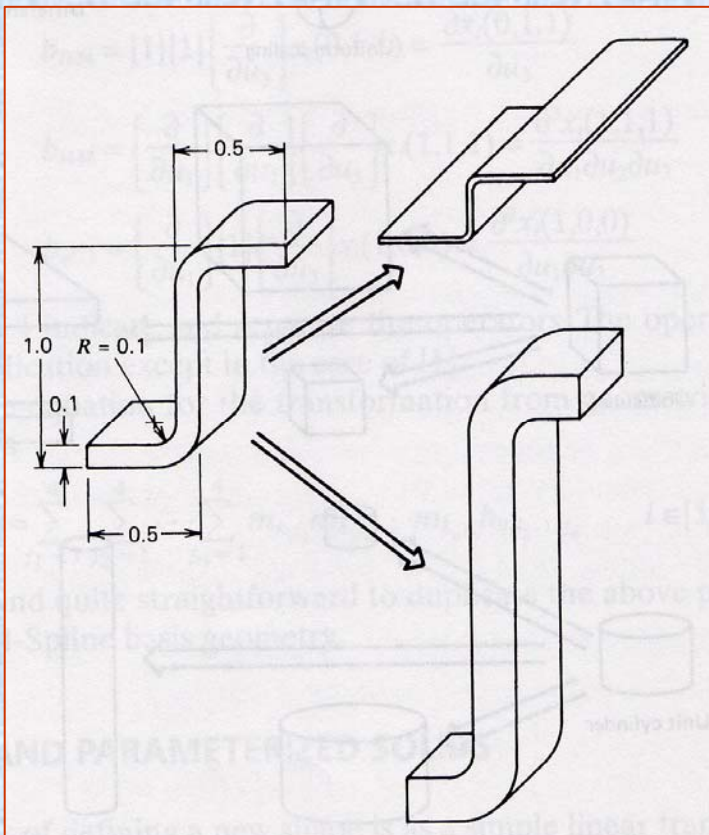
# Famille d'objets paramétrisés

- Il s'agit de décrire la famille à laquelle cet objet appartient et de définir les paramètres permettant d'identifier de façon unique un objet de cette famille.  
Ex. : Pour générer une ellipse, il suffit de fixer les valeurs des paramètres décrivant la classe des coniques.
- Les paramètres d'un objet sont en général des caractéristiques géométriques : le volume, l'aire, l'axe principal, la hauteur, la largeur, etc.
- Un objet paramétrisé peut être aussi bien une forme connue (cube, sphère, etc.) qu'une forme spécialisée selon le besoin de l'application.

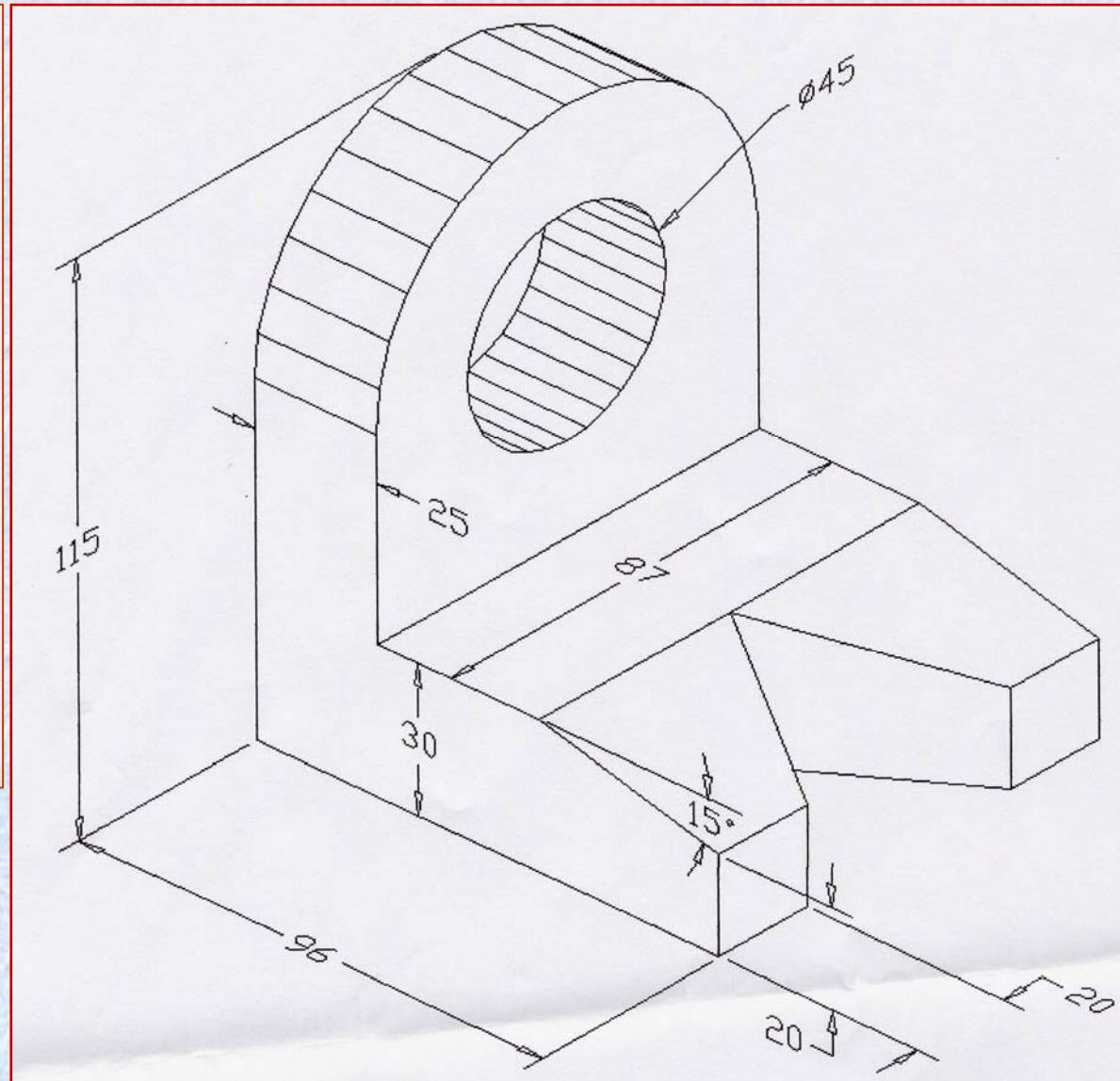
Forme paramétrisée par  $l$ ,  $h$ ,  $r$  et  $m$  



# Famille d'objets paramétrisés



M. E. Mortenson, Geometric Modeling.  
Wiley, 1997, p. 262.



Pièces mécaniques

# Famille d'objets paramétrisés

- C'est un modèle spécialisé facilitant la description de pièces souvent utilisées mais trop limité à cause de la faible flexibilité des objets paramétrisés.  


Exemple :

Modélisation d'un visage humain.
- L'apparition d'un nouvel objet nécessite la génération d'une nouvelle classe, ce qui peut s'avérer difficile, voire impossible à réaliser.
- Définir des paramètres simples en nombre limité pour caractériser un objet complexe n'est pas une mince tâche.
- Donne lieu à des algorithmes efficaces en synthèse d'images.

## 2 alternatives :

restreindre notre potentiel de modélisation

ou

augmenter substantiellement le nombre de formes paramétrisées.

- ∴ Il s'agit d'un mode de représentation secondaire dans les modeleurs  
lesquels s'appuient sur d'autres représentations.

# Modèles de composition

- Considèrent les solides comme des ensembles de points 3D.
- Débutent avec des ensembles simples qui peuvent être représentés directement à l'aide de primitives (quadriques, polyèdres, etc.).
- Des objets plus complexes sont obtenus en combinant des ensembles simples entre eux à l'aide d'opérations ensemblistes.

## 1er cas : Modèles basés sur la notion de demi-espaces

- Nous pouvons définir une fonction caractéristique d'un ensemble A de points 3D :

$$g_A(x, y, z) : (x, y, z) \rightarrow \{0, 1\}$$

qui nous indique si un point  $(x, y, z)$  appartient ou non à A.

- Si  $g_A(X) = 1$ , alors  $(x, y, z) \in A$  ; si  $g_A(X) = 0$  alors  $(x, y, z) \notin A$ .

# Modèles basés sur la notion de demi-espaces

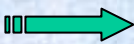
- Pour des ensembles de points 3D, la représentation de  $g_A$  est aussi difficile que celle de  $A$ .
- Par contre, il existe des classes d'objets qui peuvent être représentées à partir d'un ensemble de points 3D décrit par une fonction  $f(x, y, z)$  (demi-espace).
- L'objet est décrit comme étant l'ensemble des points  $(x, y, z)$  tels que
$$f(x, y, z) \geq 0,$$
le complément de cet objet comme étant les points  $(x, y, z)$  tels que
$$f(x, y, z) < 0.$$
- Un solide est représenté à partir d'une expression ensembliste de demi-espaces:
$$\text{Objet} = \bigcup_i \bigcap_j D_{ij} \text{ où } D_{ij} = \text{un demi-espace.}$$
- Toutes les surfaces d'intérêt ne sont pas des demi-espaces:  
surfaces bicubiques.

# Modèles basés sur la notion de demi-espaces

- **Exemple**  $C = D_1 \cap D_2 \cap D_3$  où
$$\begin{aligned} D_1 &\equiv x^2 + y^2 - r^2 \leq 0 \\ D_2 &\equiv z \geq 0 \\ D_3 &\equiv z - h \leq 0. \end{aligned}$$
- Très souvent, on opte pour des demi-espaces linéaires :
$$a x + b y + c z + d \leq 0$$
ou quadratiques (demi-espaces sphériques, cylindriques, ...).
- La capacité de modélisation dépend des familles de demi-espaces disponibles et des techniques nous permettant de les combiner ensemble.
- Les demi-espaces sont rarement des ensembles finis. Ainsi, une expression ensembliste de demi-espaces n'est pas nécessairement une représentation valide de solide.
- Aucune définition ambiguë :  
chaque combinaison valide de solides donne un solide mais sa représentation n'est pas nécessairement unique.

# Modèles basés sur la notion de demi-espaces

Comment évaluer les expressions ensemblistes de demi-espaces ?

- Il s'agit d'adapter l'algorithme de tracés de rayons (*ray tracing*) en synthèse d'images.
- Pour générer une image, définissons pour chaque *pixel* de l'écran une demi-droite issue de l'observateur et passant par ce *pixel*.
- Le calcul principal consiste à évaluer l'intersection d'une demi-droite avec chaque solide de la scène, puis, à retenir le point le plus près de l'observateur.
- L'intersection d'une demi-droite avec un solide de la scène se ramène à évaluer l'arbre binaire représentant l'expression ensembliste de demi-espaces.  Problème unidimensionnel

Contrainte : L'intersection d'une demi-droite avec un demi-espace doit être relativement simple.

# Modèles basés sur la notion de demi-espaces

Exemple : Intersection d'une demi-droite avec un polyèdre convexe.

Soit un polyèdre convexe  $\{X \mid a_i^t X + b_i \geq 0, i = 1, 2, \dots, m\}$ ,

un rayon :  $P + \lambda \mathbf{d}, \quad \lambda \geq 0$

où  $P$  désigne la position de l'observateur,

l'intersection nous donne :

$$\{\lambda \geq 0 \mid a_i^t (P + \lambda \mathbf{d}) + b_i \geq 0, i = 1, 2, \dots, m\}$$

ou encore,

$$\{\lambda \geq 0 \mid (a_i^t \mathbf{d}) \lambda \geq -a_i^t P - b_i, i = 1, 2, \dots, m\}$$

ou encore,

$$\left\{ \lambda \geq 0 \mid \begin{array}{ll} \text{Max} & \{(-a_i^t P - b_i) / a_i^t \mathbf{d}\} \leq \lambda \leq \text{Min} \{(-a_i^t P - b_i) / a_i^t \mathbf{d}\} \\ a_i^t \mathbf{d} > 0 & a_i^t \mathbf{d} < 0 \\ i = 1, 2, \dots, m & i = 1, 2, \dots, m \end{array} \right\}$$

# Modèles de composition

## 2<sup>ième</sup> cas : Modèles CSG (« *constructive solid geometry* »)

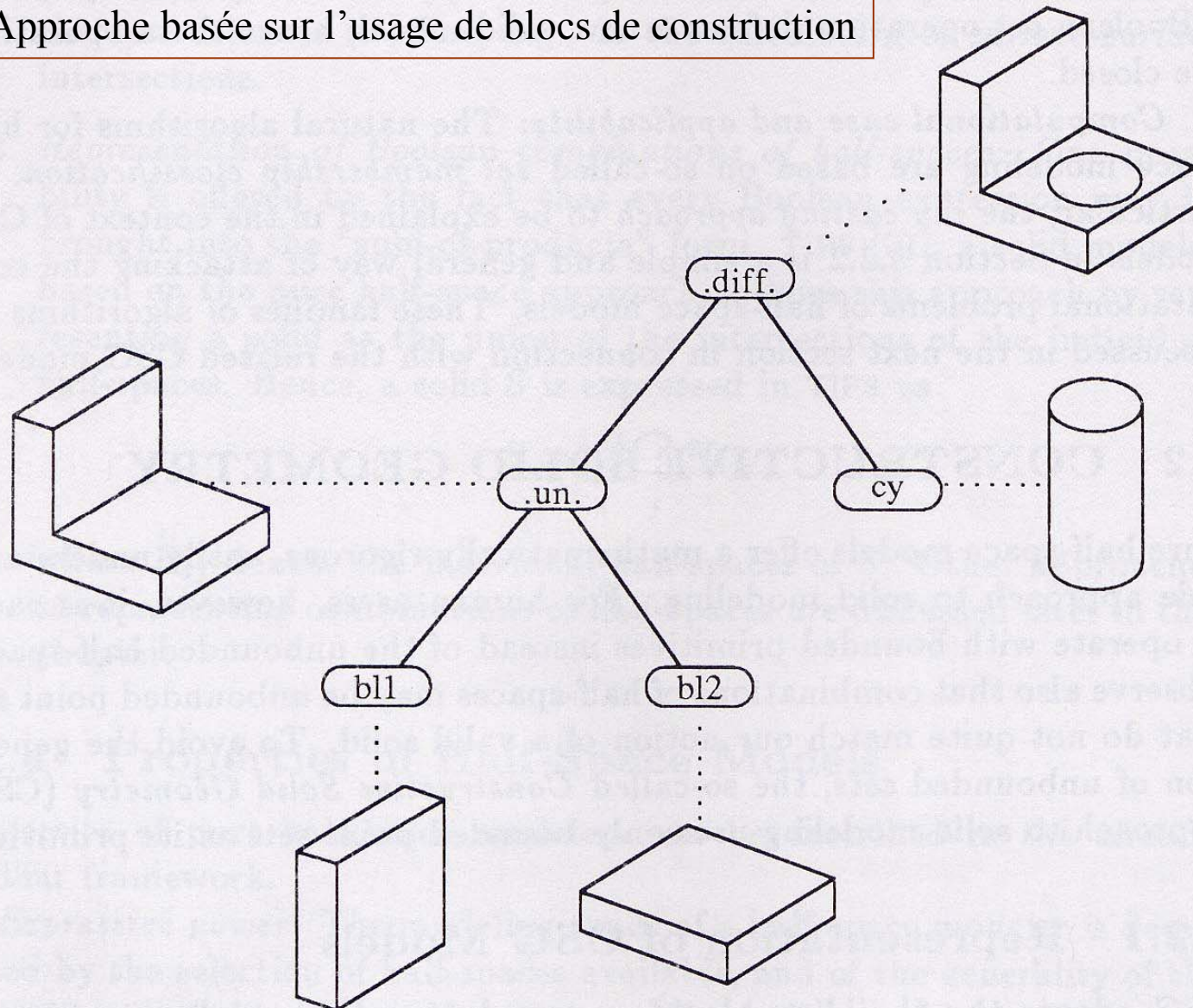
- Contrairement aux modèles avec demi-espaces, il s'agit de manipuler des objets élémentaires bornés : polyèdre, cylindre, sphère, etc.

$\langle \text{objet à modéliser} \rangle ::= \langle \text{primitive} \rangle \mid \langle \text{objet} \rangle \langle \text{opération} \rangle \langle \text{objet} \rangle$   
 $\mid \langle \text{objet} \rangle \langle \text{transformation} \rangle$   
 $\langle \text{opération} \rangle ::= \text{union} \mid \text{intersection} \mid \text{différence}$   
 $\langle \text{transformation} \rangle ::= \text{translation}$   
 $\mid \text{rotation}$   
 $\mid \text{changement d'échelle.}$

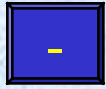
- On met à la disposition de l'utilisateur un nombre fini de primitives simples dont la grandeur, la forme, la position et l'orientation sont définies par l'utilisateur.
- Pour ce qui est des primitives non élémentaires, l'utilisateur doit spécifier uniquement le système de référence de l'objet.

# Modèles CSG

Approche basée sur l'usage de blocs de construction



# Modèles CSG



- Ce modèle est incomplet car il nécessite des algorithmes pour évaluer l'arbre de construction.
- L'évaluation de l'arbre de construction peut exiger des temps de calculs importants.
- Lorsque l'arbre CSG est mal balancé, les algorithmes proposés sont inefficaces.
- Il est difficile de construire un modèle CSG pour décrire des objets complexes n'intégrant aucune propriété géométrique.  
Ex. : la représentation d'un visage humain.



- Aucune ambiguïté dans le modèle; cela donne lieu à des objets valides.
- L'opération de modélisation est simplifiée; l'animateur doit spécifier seulement l'arbre de construction.

# Évaluation des arbres de construction des modèles CSG

- L'algorithme de *tracés de rayons* s'applique aussi aux modèles CSG en autant que le calcul d'intersection entre un rayon et chaque primitive du modèle soit relativement simple.  
Ex. de primitives : polyèdres, formes implicites :  $f(x, y, z) = 0, \dots$
- Au lieu d'opter pour une approche approximative lors de l'évaluation de l'arbre de construction, on peut évaluer de manière exacte chaque composante de l'arbre de construction.

## Exemple :

- Chaque feuille de l'arbre est un polyèdre convexe.
- Chaque sommet intermédiaire de l'arbre est un ensemble de polyèdres convexes disjoints.
- Pour évaluer chaque sommet intermédiaire de l'arbre, on peut adapter un algorithme de découpage 3D.

# Modèle de décomposition

- Les objets sont représentés comme un ensemble d'objets ou cellules élémentaires.
- Un mode de représentation de solides par partitionnement spatial où chaque solide est décomposé en un ensemble de solides adjacents, sans intersection, qui sont plus primitifs que le solide original, bien que nécessairement du même type.

1er cas : énumération exhaustive des cellules élémentaires

- Considérer l'ensemble **infini** de tous les points faisant partie d'un solide.



énumération impossible

- L'espace est décomposé en cellules identiques, souvent appelées *voxels*, arrangées en une grille fixe et régulière faisant partie ou non du solide.

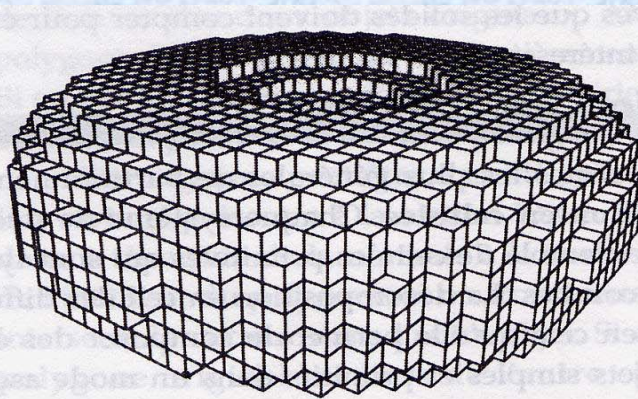


Subdivision régulière de l'espace.

# Énumération spatiale

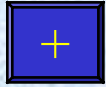
- Le type de cellule le plus courant est le cube.
- Ces cubes ne se recouvrent pas; ils ont même dimension et même orientation.
- Chaque cube peut être défini uniquement à partir d'un de ses sommets.
- En associant à chaque objet une région régulière dont l'objet fait partie, l'objet est représenté à partir d'un tableau binaire 3D.

un élément du tableau  $\equiv \begin{cases} 1 \Leftrightarrow & \text{le cube associé représente une sous-région du solide} \\ 0 \Leftrightarrow & \text{le cube ne fait pas partie du solide.} \end{cases}$

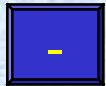


Tore représenté par énumération spatiale. [Par A.H.J. Christensen, SIGGRAPH'80 Conference Proceedings, Computer Graphics (14)3, July 1980. (Association for Computing Machinery, Inc.)]

# Énumération spatiale

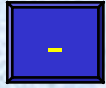


- Permet de construire facilement et efficacement de nouveaux objets à l'aide d'opérations ensemblistes en se basant sur ces tableaux binaires.
- Permet d'évaluer efficacement différentes mesures de l'objet.



- Ce modèle est une représentation approximative de l'objet à moins que celui-ci ne coïncide exactement avec la grille ce qui est rarement le cas.
- On peut réduire la taille des *voxels* afin d'augmenter la précision de la représentation du solide.  
Cela exige une quantité d'espace mémoire imposante.
- Cette approche ne peut être exploitée directement : pour construire un objet, on ne peut exiger l'énumération de l'ensemble de ces cellules.

# Énumération spatiale



- Une façon plus pratique de procéder consiste à passer d'une autre représentation à celle par énumération des cellules élémentaires.



Modèle frontière, modèle de composition, modèle de familles paramétrisées, etc.

- A) Il faut d'abord définir une région (parallélépipède) renfermant les objets de la scène.
- B) Il faut ensuite construire la grille binaire qui correspond à chaque objet de la scène du modèle frontière ou de familles paramétrisées.
- C) Dans le cas d'un modèle de composition, il faut appliquer l'étape B) avec les feuilles de l'arbre, puis, évaluer l'arbre de construction à l'aide des grilles binaires de ces feuilles.

# Énumération spatiale

## 2 ième cas : subdivision irrégulière d'une région dont l'objet à modéliser fait partie

- Les techniques d'énumération spatiale sont simples, générales et permettent de développer une grande variété d'algorithmes.
- Mais elles exigent une grande quantité d'espaces mémoires et une représentation très approximative des solides.
- Pour remédier à cet inconvénient, nous nous basons sur le principe suivant:

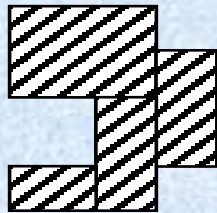
Les cubes élémentaires voisins d'un cube faisant partie de l'objet ont de bonnes chances d'en faire partie aussi.



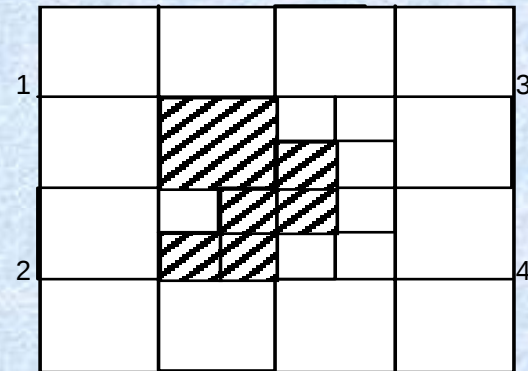
Le # de cubes nécessaires pour représenter un objet devrait dépendre de la surface de l'objet et non du volume de l'objet.

- Il existe plusieurs méthodes basées sur ce principe:  
notamment les **OCTREE** et les **QUADTREE**.

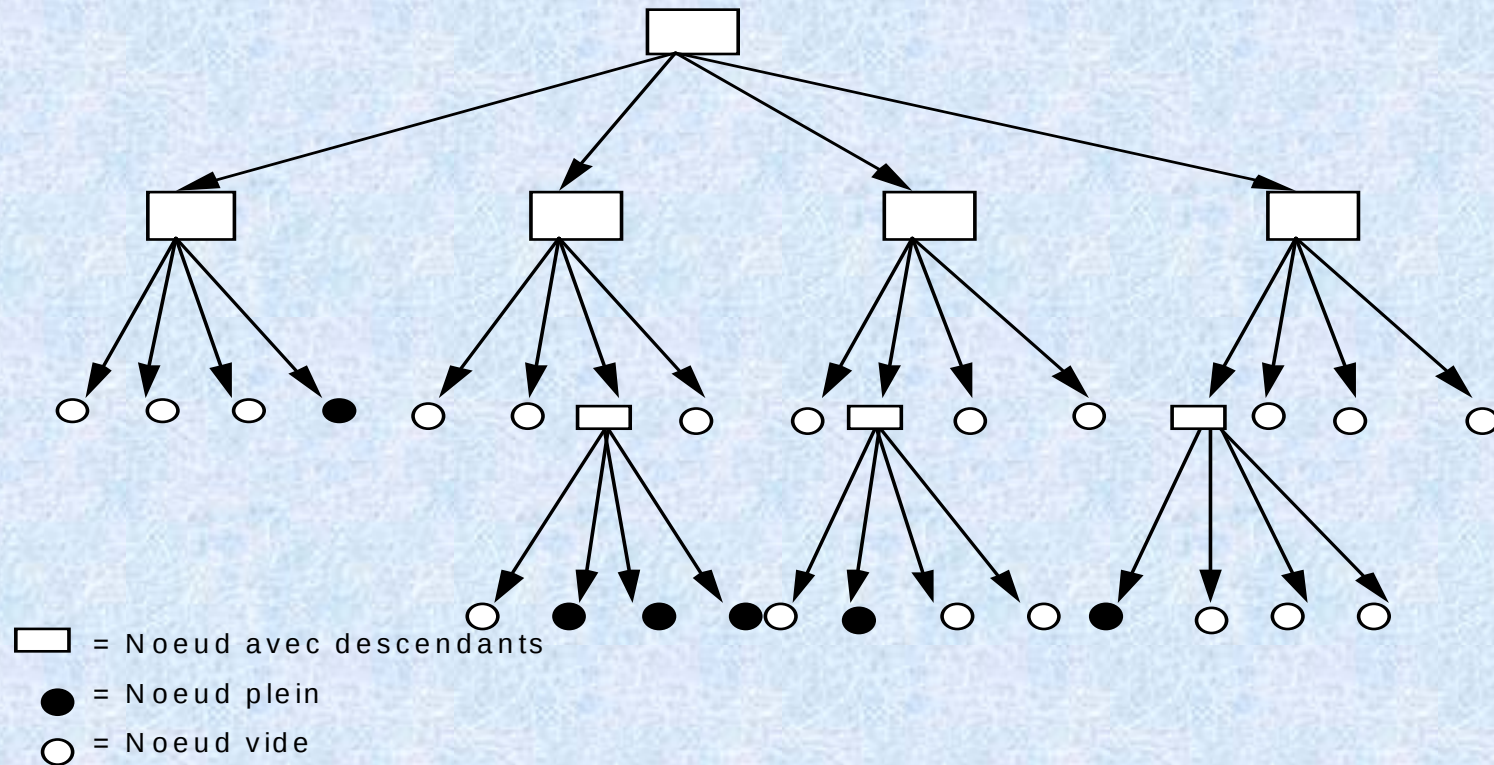
# Représentation QUADTREE en 2D



(a)



(b)



# Représentation QUADTREE

- Racine : l'espace initial englobant l'objet (un carré renfermant les objets de la scène).
- Feuilles : des blocs (carrés) qui sont
  - complètement remplis (faisant partie de l'objet) ou
  - complètement vides (ne faisant pas partie de l'objet).
- Subdivision réursive de l'espace contenant l'objet 2D en 4 régions carrées, chaque carré en 4 régions carrées, etc.
- Le processus prend fin dans l'un des 3 cas suivants:
  - Nous aboutissons à une cellule vide ne contenant aucune partie de l'objet.
  - Nous aboutissons à une cellule entièrement remplie.
  - Le niveau de résolution est atteint.
- La structure de données correspondante à ce modèle est un arbre où les sommets ont 0 ou 4 descendants (« QUADTREE »).

## Représentation QUADTREE

```
const niveau_maximum_de_recurzivite = 10;

enum etat { homogene, heterogene } ;

struct Information_QUADTREE
{
    enum etat Statut ;
    union
    {
        int Couleur ;                // sommet homogène
        struct Information_QUADTREE * Quatre_Enfants;
                                        // sommet hétérogène
    } ;
} ;

struct objet
{
    float vx; float vy; float longueur;
                                // définition d'un carré renfermant l'objet.
    struct Information_QUADTREE * racine;
};
```

# Représentation QUADTREE

Note :

Dans chaque sommet du QUADTREE, on pourrait conserver le niveau de récursivité et la définition du carré associé.

On choisit plutôt de les déduire au fur et à mesure que l'on parcourt la structure d'arbre par souci d'économie d'espace.

## Construction d'un QUADTREE

Soient  $x, y$  les coordonnées du sommet inférieur gauche,  
la longueur d'un côté  $c$  du carré renfermant l'objet,  
le niveau de récursivité  $n$  ( $= 0$ ),  
une pile  $S$  vide,

créer un nouveau sommet `Information_QUADTREE` pointé par  $p$ ,  
initialiser le pointeur vers la racine de l'objet à l'aide de  $p$ ,  
initialiser  $vx, vy$  et longueur de l'objet à l'aide de  $x, y$  et  $c$ ,  
insérer un sommet dans  $S$  renfermant les données  $n, p, x, y$  et  $c$ .

## Représentation QUADTREE

Tant et aussi longtemps que la pile S n'est pas vide

{ enlever un élément de S avec comme données n, p, x, y et c;  
si le carré défini par x, y et c est rempli  
alors

le sommet pointé par p est homogène,

la couleur associé à ce sommet est celle de l'objet,

sinon si le carré défini par x, y et c est vide  
alors

le sommet pointé par p est homogène,

la couleur associé à ce sommet est la couleur  
de fond,

sinon

si n égale le niveau maximum de récursivité  
alors

le sommet pointé par p est homogène,

la couleur associé à ce sommet est choisie

sinon /\*\*\* voir la diapositive suivante <sup>32</sup>\*\*\* /

# Représentation QUADTREE

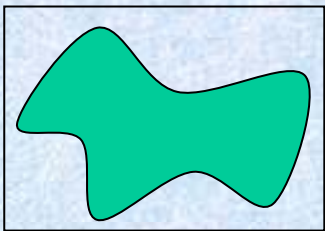
## Cas hétérogène :

le sommet pointé par  $p$  est hétérogène,  
créer un vecteur de 4 sommets de type `Information_QUADTREE`  
le sommet pointé par  $p$  renferme l'adresse de ce vecteur,  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du premier sommet de ce vecteur,  
     $x$ ,  $y$  et  $c/2$ ;  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du deuxième sommet de ce vecteur,  
     $x$ ,  $y + c/2$  et  $c/2$ ;  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du troisième sommet de ce vecteur,  
     $x + c/2$ ,  $y$  et  $c/2$ ;  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du quatrième sommet de ce vecteur,  
     $x + c/2$ ,  $y + c/2$  et  $c/2$ .

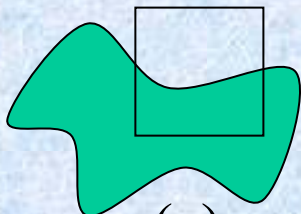
}

# Représentation QUADTREE

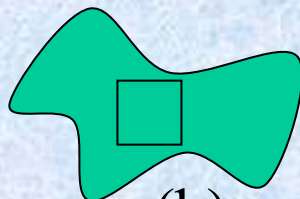
- Pour évaluer le statut d'un carré par rapport à l'objet 2D, on procède comme suit :
  - si l'une des 4 arêtes du carré intercepte la frontière de l'objet, alors la subdivision se poursuit si le niveau de résolution n'est pas atteint **(a)**
  - sinon choisissons un des 4 sommets du carré, disons P, si P fait partie de l'objet alors le carré est rempli **(b)**
  - sinon choisissons un point de l'objet, disons Q, si Q ne fait pas partie de la fenêtre alors le carré est vide **(c)**
  - sinon la subdivision se poursuit si le niveau de résolution n'est pas atteint **(d)**.



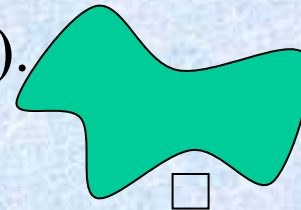
(d)



(a)



(b)



(c)

# Intersection de 2 QUADTREEs

Soient A et B, 2 objets représentés par des QUADTREEs,  
x, y les coordonnées du sommet inférieur gauche,  
la longueur d'un côté c du carré renfermant les 2 objets A et B,  
p un pointeur vers la racine de A,  
q un pointeur vers la racine de B,  
R un objet résultant de l'intersection de A et de B,  
une pile S vide,

créer un nouveau sommet Information\_QUADTREE pointé par r,  
initialiser le pointeur vers la racine de R à l'aide de r,  
initialiser vx, vy et longueur de l'objet R à l'aide de x, y et c,  
insérer un élément dans la pile S renfermant les données p, q et r,

Tant et aussi longtemps que la pile S n'est pas vide

{

enlever un élément de S avec comme données p, q et r;

# Intersection de 2 QUADTREES

si le sommet pointé par p et celui pointé par q sont homogènes,

alors (\*)

le sommet pointé par r est homogène,

la couleur du sommet pointé par r est la couleur de fond sauf si  
la couleur du sommet pointé par p et la couleur de celui pointé  
par q coïncident avec la couleur de l'objet.

sinon

si le sommet pointé par p (resp. q) est homogène et de couleur  
de fond

(\*\*) alors le sommet pointé par r est homogène,

la couleur du sommet pointé par r est la couleur de fond

sinon si le sommet pointé par p (resp. q) est homogène et de  
couleur de l'objet,

(\*\*\*) alors copier le sous-arbre pointé par q (resp. p) à partir  
du sommet pointé par r

(\*\*\*\*) sinon /\*\*\* voir la diapositive suivante \*\*\*\*\*/

# Intersection de 2 QUADTREES

## Cas hétérogène :

le sommet pointé par r est hétérogène,  
créer un vecteur de 4 sommets de type Information\_QUADTREE,  
le sommet pointé par r renferme l'adresse de ce vecteur,

ajouter à la pile S un sommet avec comme données

- l'adresse du 1<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par p,
- l'adresse du 1<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par q,
- l'adresse du 1<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par r;

ajouter à la pile S un sommet avec comme données

- l'adresse du 2<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par p,
- l'adresse du 2<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par q,
- l'adresse du 2<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par r;

ajouter à la pile S un sommet avec comme données

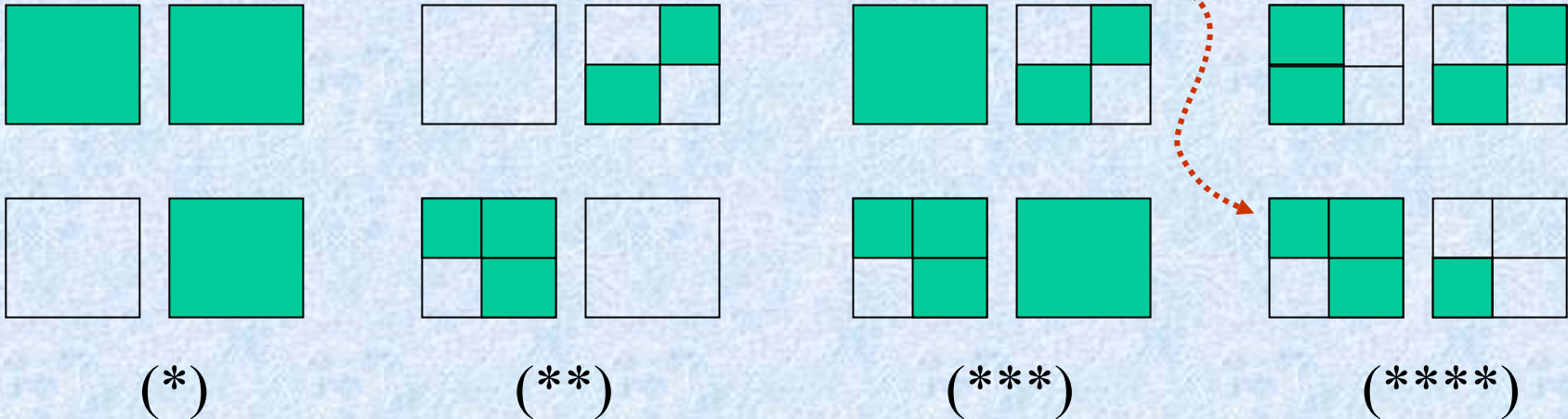
- l'adresse du 3<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par p,
- l'adresse du 3<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par q,
- l'adresse du 3<sup>e</sup> sommet du vecteur faisant partie du sommet pointé par r;

# Intersection de 2 QUADTREES

ajouter à la pile S un sommet avec comme données

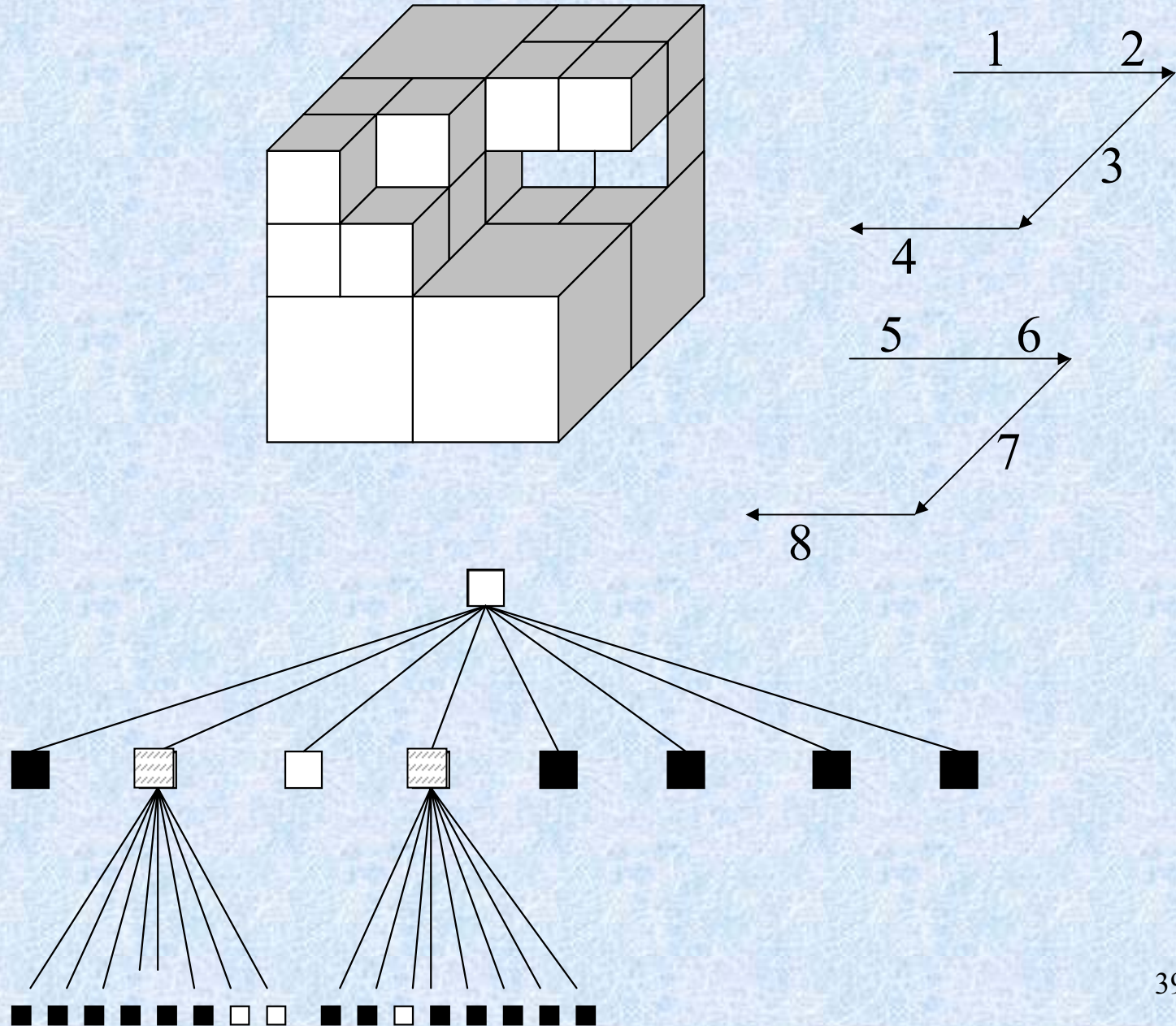
- l'adresse du 4e sommet du vecteur faisant partie du sommet pointé par p,
- l'adresse du 4e sommet du vecteur faisant partie du sommet pointé par q,
- l'adresse du 4e sommet du vecteur faisant partie du sommet pointé par r;

```
// si les 4 sommets du vecteur faisant partie du sommet pointé par r sont  
// homogènes et vides alors modifier le sommet pointé par r pour qu'il devienne  
// homogène et vide en adoptant une approche récursive ou en effectuant un  
// post-traitement.  
}
```



Exercices : union, différence de 2 QUADTREES.

# Représentation en OCTREE



# Représentation en OCTREE

- Racine : l'espace initial englobant les objets (un cube dont la dimension est une puissance de 2).
- Feuilles : des blocs (cubes) qui sont
  - complètement remplis (faisant partie de l'objet) ou
  - complètement vides (ne faisant pas partie de l'objet).
- Subdivision réursive de l'espace contenant l'objet 3D en 8 régions cubiques, chaque cube en 8 régions cubiques, etc.
- Le processus prend fin dans l'un des 3 cas suivants:
  - Nous aboutissons à une cellule vide ne contenant aucune partie de l'objet.
  - Nous aboutissons à une cellule entièrement remplie.
  - Le niveau de résolution est atteint.
- La structure de données correspondante à ce modèle est un arbre où les sommets ont 0 ou 8 descendants (« OCTREE »).

## Représentation en OCTREE

```
const niveau_maximum_de_recurivite = 10;

enum etat { homogene, heterogene } ;

struct Information_OCTREE
{
    enum etat Statut ;
    union
    {
        int Couleur ;                // sommet homogène
        struct Information_OCTREE * Huit_Enfants;
                                     // sommet hétérogène
    } ;
} ;

struct objet
{
    float vx; float vy; float vz; float longueur;
                                     // définition d'un cube renfermant l'objet.
    struct Information_OCTREE * racine;
};
```

# Représentation en OCTREE

## Construction d'un OCTREE

Soient  $x, y, z$  les coordonnées du sommet en bas, à gauche, en arrière,  
la longueur d'un côté  $c$  du cube renfermant l'objet,  
le niveau de récursivité  $n$  ( $= 0$ ),  
une pile  $S$  vide,

créer un nouveau sommet `Information_OCTREE` pointé par  $p$ ,  
initialiser le pointeur vers la racine de l'objet à l'aide de  $p$ ,  
initialiser  $v_x, v_y, v_z$  et longueur de l'objet à l'aide de  $x, y, z$  et  $c$ ,  
insérer un sommet dans  $S$  renfermant les données  $n, p, x, y, z$  et  $c$ .

Tant et aussi longtemps que la pile  $S$  n'est pas vide  
{  
  enlever un élément de  $S$  avec comme données  $n, p, x, y, z$  et  $c$ ;  
  si le cube défini par  $x, y, z$  et  $c$  est rempli  
  alors

    le sommet pointé par  $p$  est homogène,

    la couleur associé à ce sommet est celle de l'objet,

# Représentation en OCTREE

## Construction d'un OCTREE (suite)

sinon si le cube défini par  $x$ ,  $y$ ,  $z$  et  $c$  est vide  
alors

le sommet pointé par  $p$  est homogène,  
la couleur associée à ce sommet est la couleur  
de fond,

sinon

si  $n$  égale le niveau maximum de récursivité  
alors

le sommet pointé par  $p$  est homogène,  
la couleur associée à ce sommet est choisie

sinon

/\*\* voir la diapositive suivante \*\*\*/

## Construction d'un OCTREE (fin)

le sommet pointé par  $p$  est hétérogène,  
créer un vecteur de 8 sommets de type `Information_OCTREE`  
le sommet pointé par  $p$  renferme l'adresse de ce vecteur,  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du premier sommet de ce vecteur,  
     $x$ ,  $y$ ,  $z$  et  $c/2$ ;  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du deuxième sommet de ce vecteur,  
     $x + c/2$ ,  $y$ ,  $z$  et  $c/2$ ;  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du troisième sommet de ce vecteur,  
     $x$ ,  $y$ ,  $z + c/2$  et  $c/2$ ;  
...  
ajouter à la pile  $S$  un sommet avec comme données  
     $n+1$ , l'adresse du huitième sommet de ce vecteur,  
     $x + c/2$ ,  $y + c/2$ ,  $z + c/2$  et  $c/2$ .

}

# Représentation en OCTREE

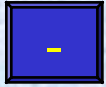
Pour évaluer le statut d'un cube par rapport à l'objet 3D, on procède comme suit :

si l'une des arêtes du cube intercepte la frontière de l'objet,  
alors la subdivision se poursuit si le niveau de résolution  
n'est pas atteint  
sinon choisissons un des sommets du cube, disons P,  
si P fait partie de l'objet  
alors le cube est rempli  
sinon choisissons un point de l'objet, disons Q,  
si Q ne fait pas partie du cube  
alors le cube est vide  
sinon la subdivision se poursuit si le  
niveau de résolution n'est pas  
atteint.

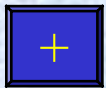
# Composantes d'un modeleur 3D basées sur la structure d'OCTREE

- un générateur d'OCTREE permettant de convertir une représentation quelconque d'un objet en une représentation par OCTREE ;
- une composante permettant d'effectuer des opérations ensemblistes sur des objets représentés par une structure d'OCTREE;
- une composante permettant d'effectuer des transformations ponctuelles ou des transformations visuelles à des objets représentés par une structure d'OCTREE;
- une composante permettant d'évaluer certaines mesures caractérisant l'objet telles que le volume ou la surface de l'objet ;
- une composante permettant de visualiser un objet représenté par une telle structure.
- etc.

# Représentation en OCTREE ou QUADTREE



- Ce modèle est aussi une représentation approximative de l'objet.
- Exige aussi une quantité d'espace mémoire imposante.
- La représentation de l'objet dépend de son emplacement :  
Une simple translation ou rotation de l'objet exige de reconstruire la structure d'OCTREE.



- Ce modèle permet d'effectuer avec facilité des opérations ensemblistes sur les objets.

# Variantes pour réduire l'espace mémoire exigée

## Subdivision binaire Arbres BSP (« Binary Space-Partitioning »)

Subdivision binaire (2 descendants) successivement dans les directions x, y et z au lieu de subdiviser en 8 cubes.



Légère économie.

## OCTREE linéaire

- Une adresse est définie à chaque sommet de l'OCTREE en numérotant les octants de 0 à 7 : l'adresse d'un sommet de niveau i correspond à une suite de i chiffres octaux.
- Un caractère spécial, disons « X » marque la fin d'un nombre dont la longueur est moindre que la résolution maximale.

### Gargantini

{0X, 10, 11, 12, 13, 14, 15, 30, 31, 33, 34, 35, 36, 37, 4X, 5X, 6X, 7X}

### Kawaguchi & Endo

Traversée en préordre de l'OCTREE en stockant les étiquettes « V » (sommet vide), « R » (rempli) et « ( » (intermédiaire).

(R(RRRRRRVVV(RRVRRRRRRRRR.