

IFT3355: Infographie  
Modélisation des solides

© Pierre Poulin

Dép. I.R.O.

Université de Montréal

## Modélisation de solides

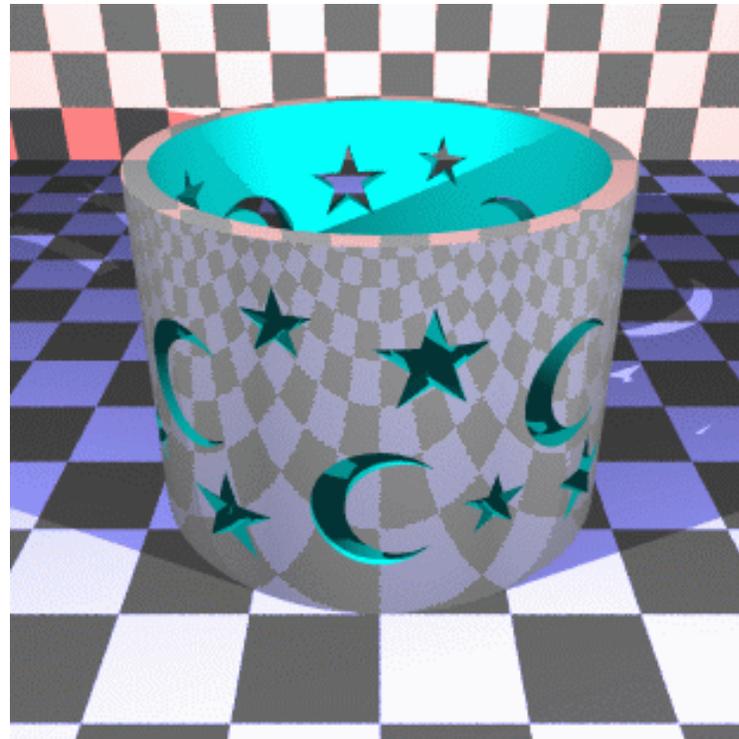
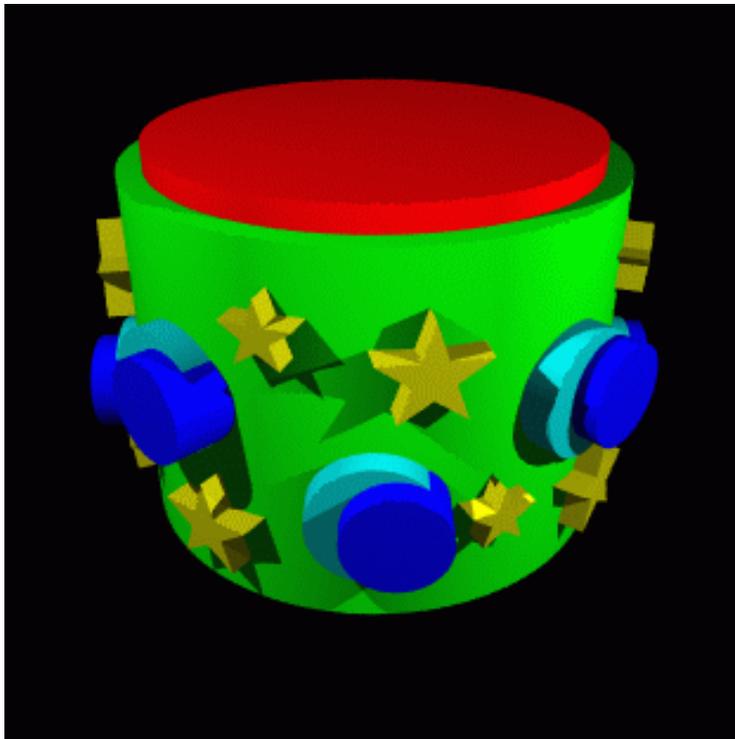
- Dans certaines applications, il est important de pouvoir caractériser un point par rapport à un objet:
  - intérieur *ou* extérieur *ou* sur la surface
- En considérant un objet comme un volume, on peut utiliser ses propriétés:
  - interférence entre objets (collisions, déformations)
  - volume et masse (moments d'inertie)
  - méthodes d'éléments finis (forces dans l'objet)

## Opérations booléennes (régulières)

- Appliquées sur les volumes :
  - union, intersection, différence
- Ces opérations sont non-fermées sur un solide
  - il peut en résulter un volume, une surface, une courbe, un point, le vide
- Les opérations booléennes dites *régulières* sont fermées sur un solide, donc résultent toujours en un volume ou le vide

# CSG

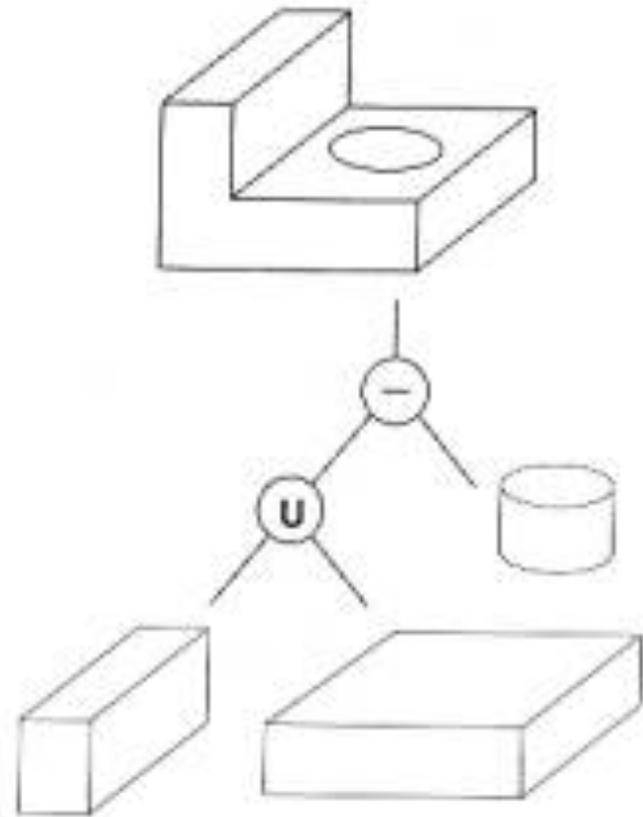
- Combine des solides avec des opérations booléennes (*Constructive Solid Geometry*)



Dodgson

# CSG

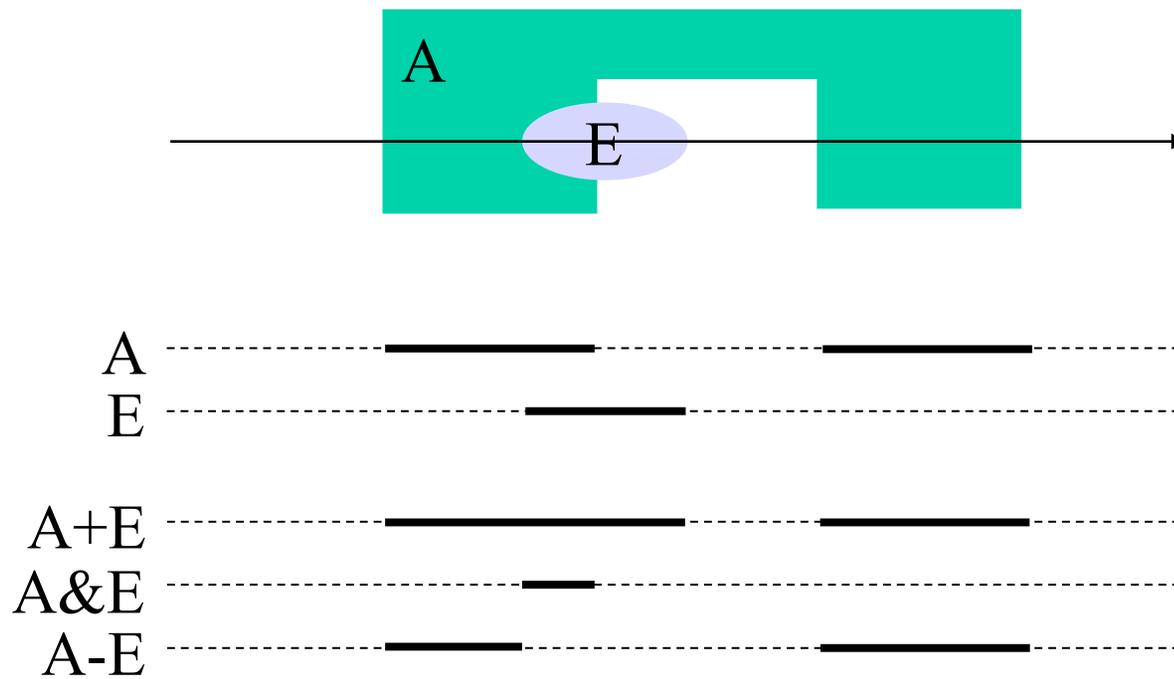
- Objet est défini comme un arbre:
  - feuilles:
    - objets simples (primitives)
  - noeuds internes:
    - opérateurs booléens
    - transformations



## CSG - Affichage

- Affichage en transférant sous représentation surfacique, telle *B-rep*
  - précalcul complexe
  - représentation finale sous forme de polygones
- Affichage par lancer de rayons
  - évalue l'arbre de composition pour chaque point d'intersection
  - approximatif (valide seulement aux rayons)

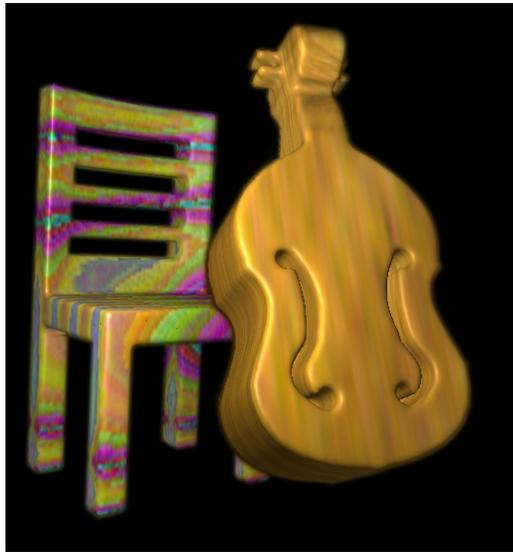
# CSG par lancer de rayons



## Division de l'espace

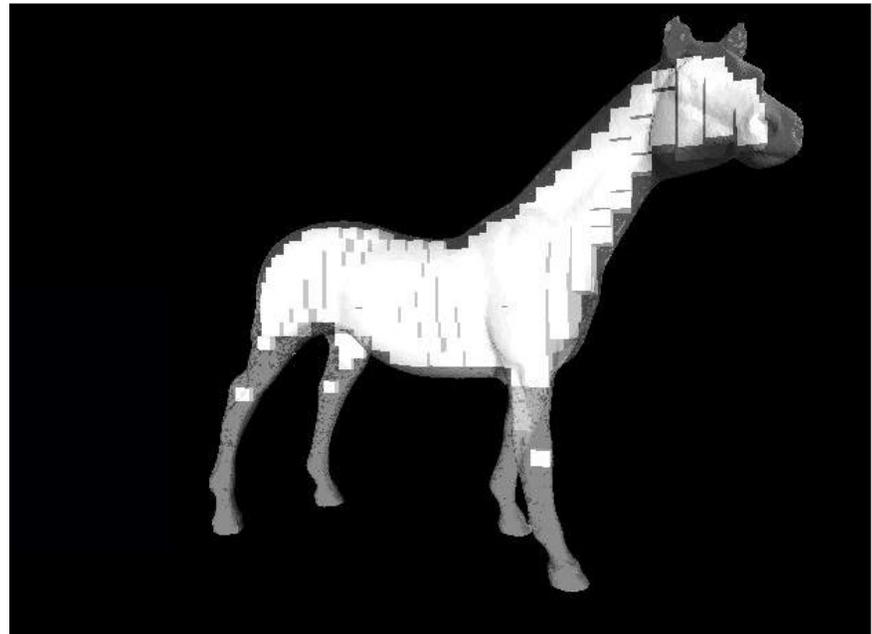
- Décomposition en *voxels*
  - objet est représenté par la liste des voxels occupés
- Décomposition en *octree*
  - nombre de noeuds est proportionnel à la *surface* de l'objet
  - *bin-tree*: alterne l'axe de subdivision
  - *kd-tree*: déplace librement le plan de subdivision
- Décomposition en *BSP tree*

# Division de l'espace



Kaufmann, SUNY

voxels



octree

# Représentations des solides - Difficultés

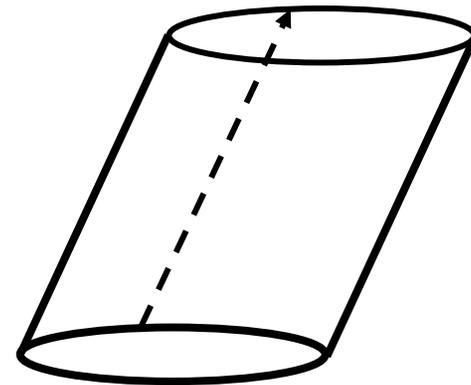
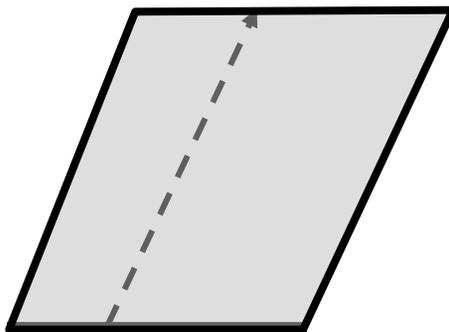
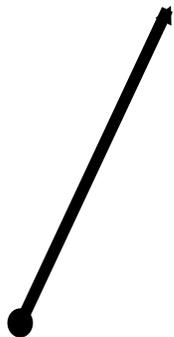
- Représentation robuste
  - instabilités dues à la précision machine des calculs numériques
- Transferts des propriétés après les opérations
- Domaine d'application (type d'objets, précision)
- Efficacité (pré-évaluation)
- Espace mémoire requis par la représentation
- Interface usager
- Transferts d'une représentation à une autre

# Représentation surfacique

- Malgré l'intérêt indéniable envers le traitement volumique des objets, la plupart des représentations d'objets en infographie sont surfaciques
  - polygones
  - surfaces bilinéaires et bicubiques
  - surfaces de balayage
  - surfaces de subdivision
  - *B-rep*
  - points

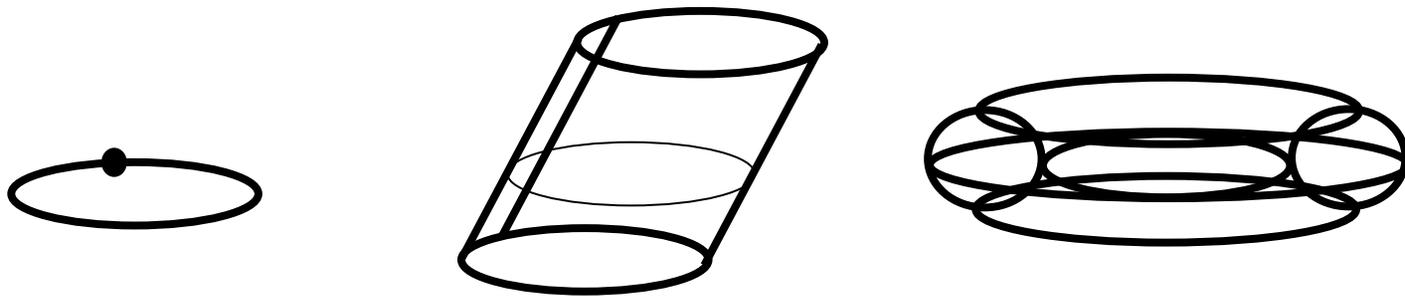
# Balayage

- Balaye un objet le long d'une trajectoire  
(*sweep representation*)
- Attention aux *self-intersections*
- 1. Segment : *extrusion*

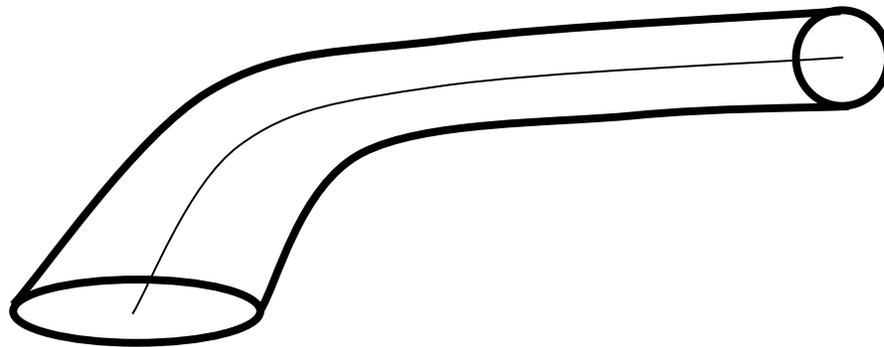


# Balayage

- 2. Rotation : cylindre généralisé



- 3. Courbe : balayage généralisé



## *B-rep (Boundary representation)*

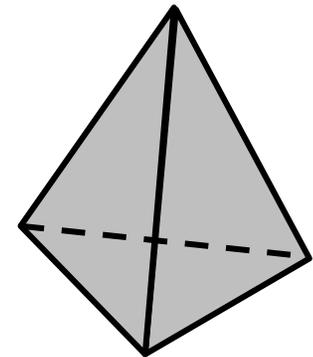
- Un objet est représenté par sa surface:
  - sommets, segments, faces
- *2-manifolds*:
  - topologie d'un disque sur un plan
  - aucune inter-pénétration des faces
- Polygones et polyèdres simples (*2-manifolds*)
  - peut être déformé en une sphère
- Représentation utilisée surtout dans les systèmes CAD

## *B-rep*

- Un *polyèdre simple* satisfait la formule d'Euler:

$$V - E + F = 2$$

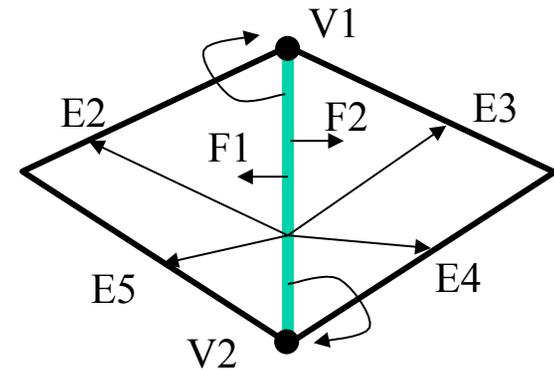
- nécessaire mais pas suffisant, puisqu'on peut créer un ensemble de faces qui satisfait la formule d'Euler sans former un polyèdre simple
- Représentation simple (*NFF*):
  - coordonnées de chaque sommet
  - liste d'index de sommets
    - sens horaire de l'intérieur du polyèdre
  - liste de faces



$$4 - 6 + 4 = 2$$

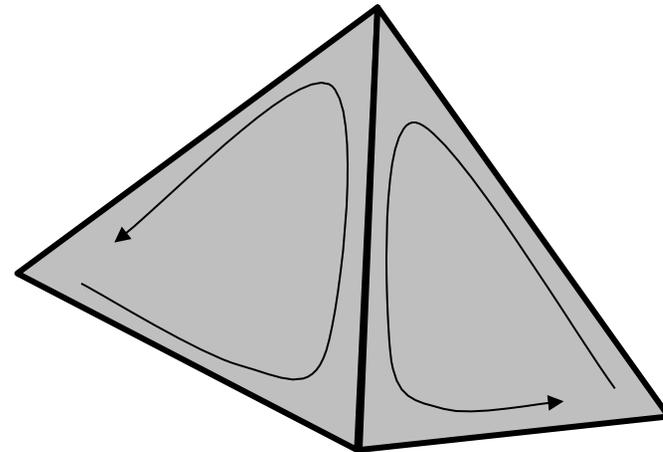
## *B-rep et winged-edge*

- Représentation simple rend parfois certains calculs ou l'accès à certaines informations plus coûteuses, e.g., trouver les voisins d'une face
- Représentation *winged-edge*
- Un segment pointe à:
  - ses deux sommets
  - ses deux faces
  - les autres segments connectés à ses sommets



## *B-rep et winged-edge*

- Un sommet pointe à:
  - un segment qui part de lui
- Une face pointe à:
  - un de ses segments

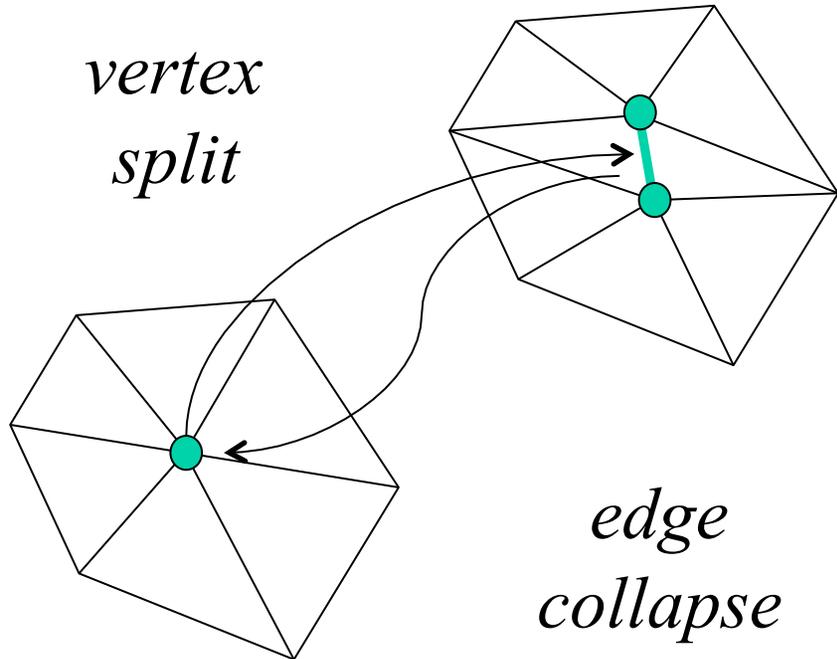


- Les structures et *surtout les opérations* sont plus complexes

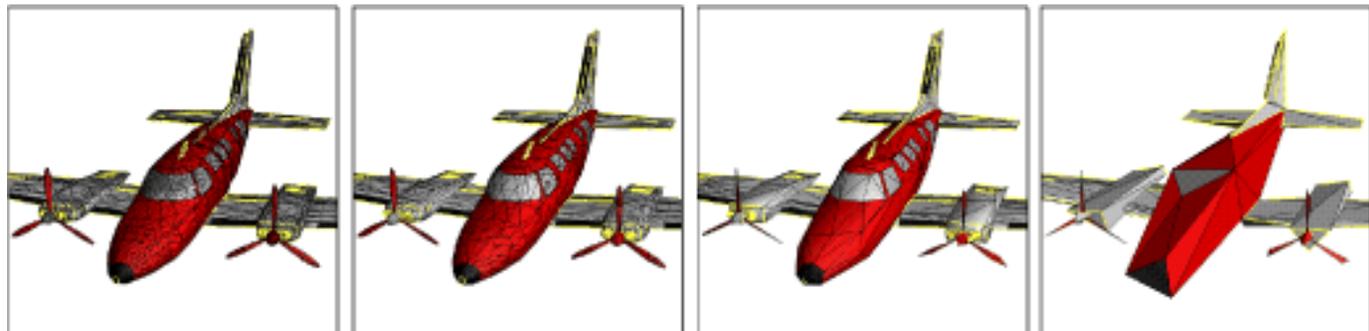
# Applications

- Paramétrisation de surface de maillage
- Maillages progressifs pour la simplification de maillages
- *Strips*

# Applications – Simplification de maillages

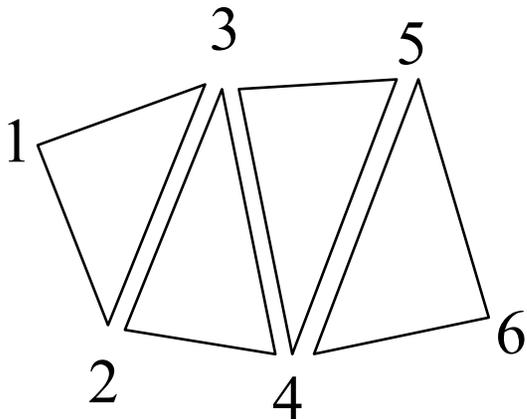


Jeff Somers

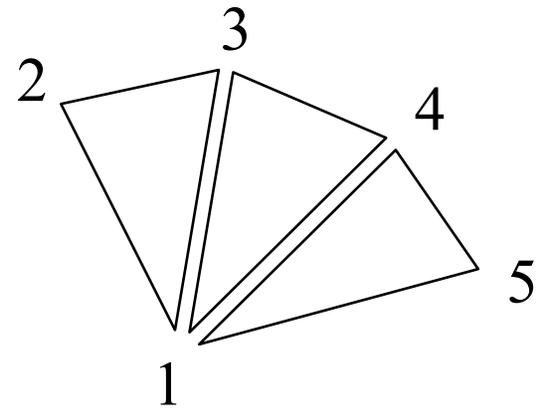


Hughes Hoppe

# Applications – *Triangle Strips*



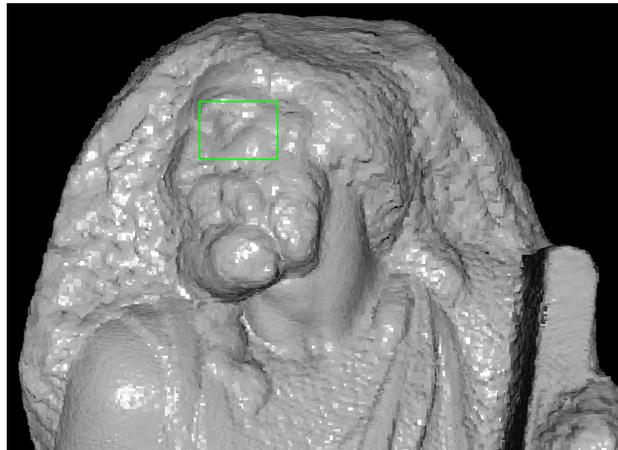
*Triangle strip*



*Triangle fan*

# Représentation par points

- Plus les objets deviennent complexes, plus le nombre de primitives de base nécessaires devient grand, et plus ces primitives deviennent petites
- La représentation par point utilise simplement des points avec couleur et normale pour représenter un objet complexe



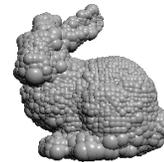
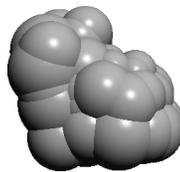
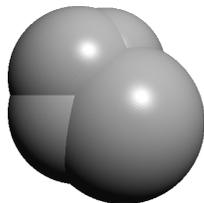
Rusinkiewicz



Stamminger

# Représentation par points

- + Les points peuvent représenter presque n'importe quel objet
- + Rendu d'un point est très efficace en hardware
- + Organisation hiérarchique des points (niveaux de détails) est facile et efficace pour le *culling*
- Il faut savoir échantillonner suffisamment les objets en fonction du point de vue et des effets (ombres, shading, textures, réflexions, réfractions)



# Modélisation d'objets complexes

- Grande variété d'objets à modéliser selon
  - le type des objets
    - surfaciques vs. volumiques
    - naturels vs. humains
  - les échelles de grandeur
  - le mouvement/comportement
    - les rigidités de surface
    - les degrés de liberté inhérents et souhaitables
    - les contrôles et les interfaces
  - le temps disponible

## Modélisation d'objets complexes

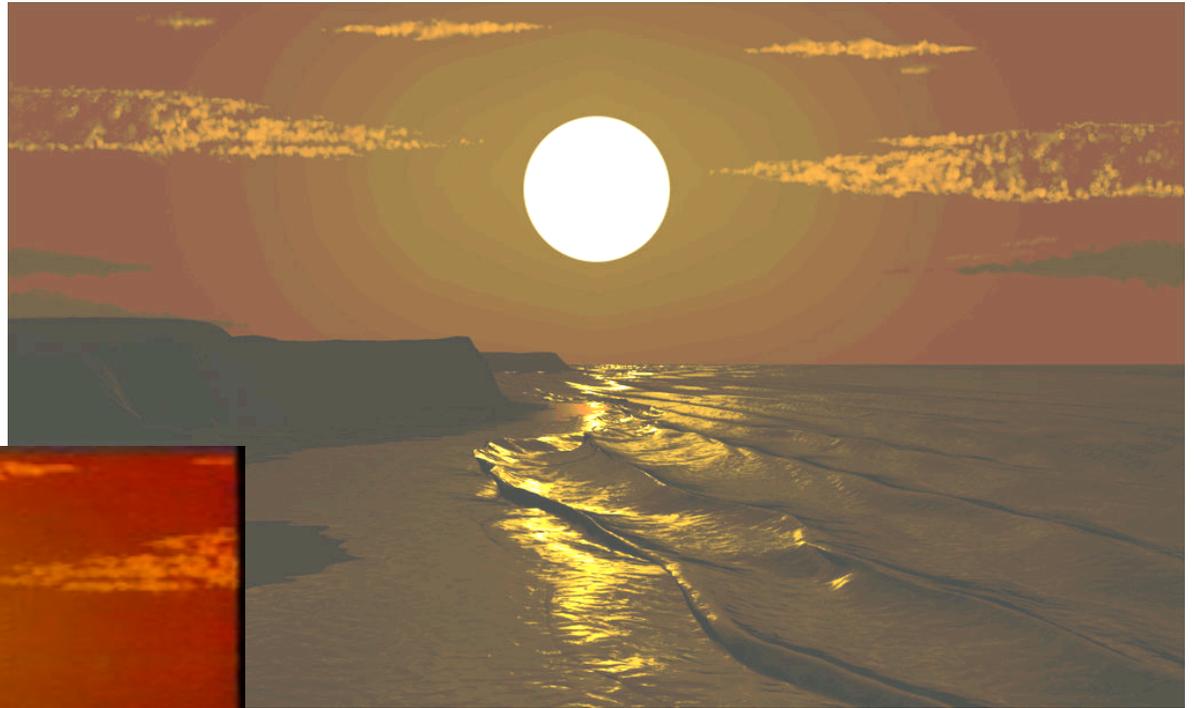
- Une grande variété d'objets dits *naturels* existent pour lesquels les objets (ou phénomènes) complexes peuvent être simulés par des modèles avec des paramètres simples
- Ces modèles permettent une augmentation automatique de leur base de données, dont les détails sont souvent irréalisables par un animateur

ex: arbres, montagnes, vagues, flammes, fumée

# Génération automatique - Modèles procéduraux

- Le modèle interagit (communique) avec son environnement (et autres objets):
  - polygonisation d'une sphère en fonction de sa projection sur l'écran, ou subdivision d'une patche de Bézier selon la proximité d'un objet
  - croissance d'une branche et de son feuillage selon l'ensoleillement, son poids, le vent
  - érosion d'une montagne selon le contenu de son sol et l'écoulement des cours d'eau

# Modèles procéduraux



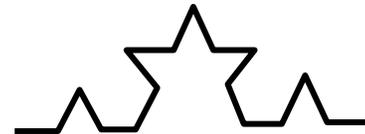
Reeves - Fournier



Autres vagues avec un  
modèle de Perlin

## Modèles procéduraux - Fractales

- Dans la nature, on observe souvent une certaine similarité entre différents niveaux de détails



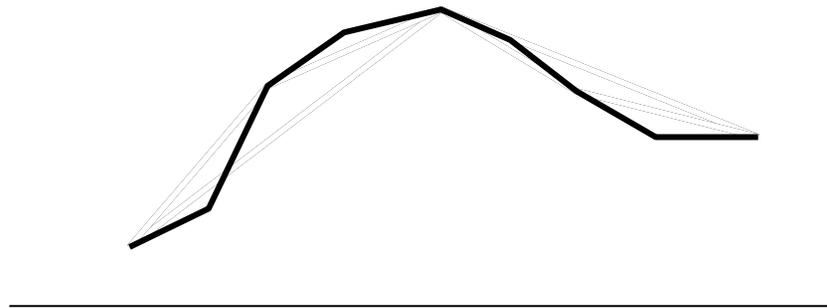
courbe de Koch

- ainsi que des perturbations aléatoires

$$x \rightarrow x^2 + c$$

où  $x$  et  $c$  sont complexes (ensembles Julia, Mandelbrot)

# Modèles procéduraux - Fractales



- Les subdivisions originales sont souvent visibles
- La régularité des grilles sont aussi visibles
  - perturbations dans le plan

# Montagnes fractales



Terragen v0.5.128 11.98

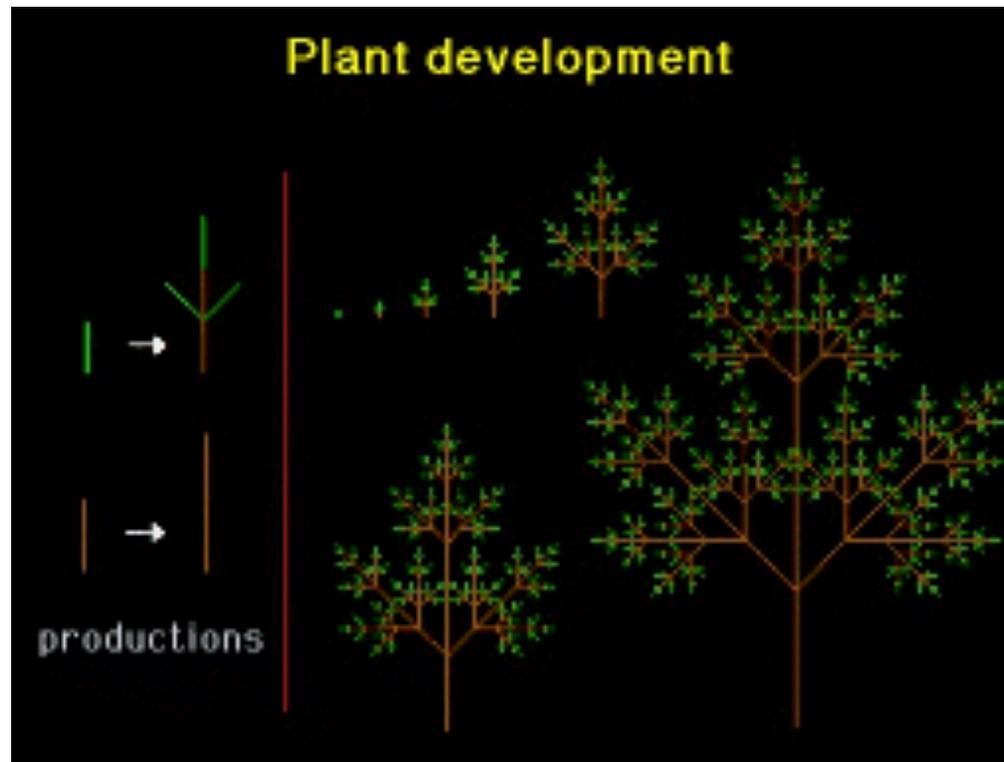
Terragen



<http://planetside.base.org/>  
Copyright 1998 Kev Fairclough

# Modèles procéduraux - Grammaires $L$

- Lindenmayer introduit des grammaires ( $L$ -systems) pour décrire la structure de plantes



# Modèles procéduraux - Grammaires $L$

- Série de règles de substitution

–  $A : AA$

–  $B : A[B]AA[B]$

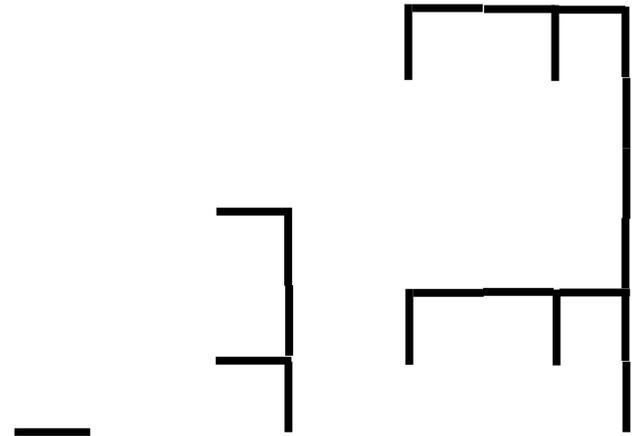
- Séquence initiale

- Quelques générations:

- 0.  $B$

1.  $A[B]AA[B]$

2.  $AA[A[B]AA[B]]AAAA[A[B]AA[B]]$



# Grammaires $L$

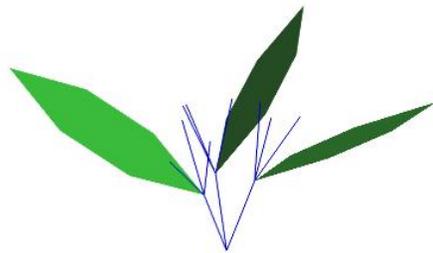
w : A

p1 : A  $\rightarrow$  [ & F L ! A ] // // // [ & F L ! A ] // // // // [ & F L ! A ]

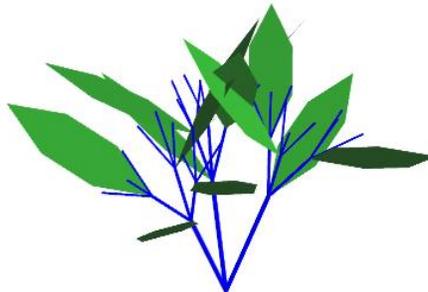
p2 : F  $\rightarrow$  S // // // F

p3 : S  $\rightarrow$  F L

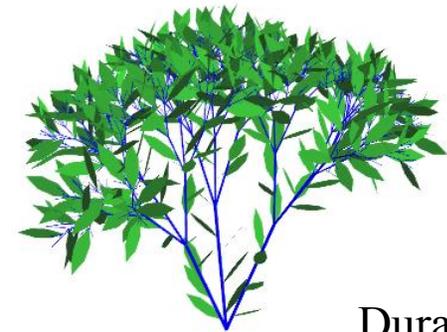
p4 : L  $\rightarrow$  [ ^ ^ { . - f . + f . + f . - | - f . + f . } ]



2e génération



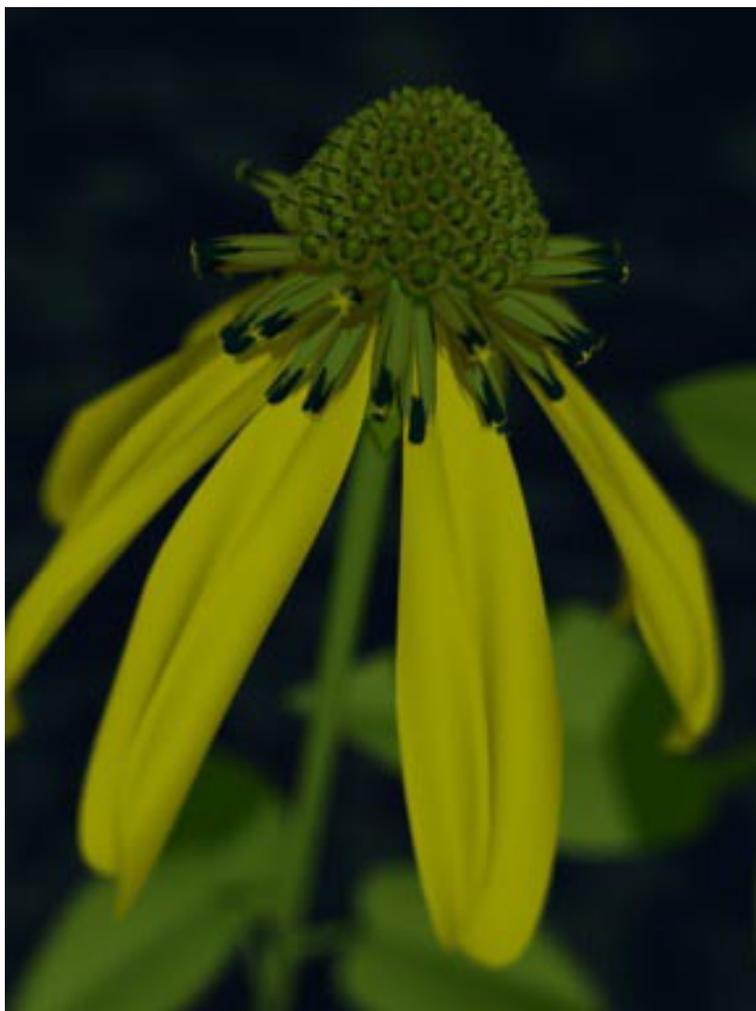
3e génération



Duranleau

6e génération

# Grammaires *L*



Prusinkiewicz



## Modèles procéduraux - Particules

- Les systèmes de particules permettent de simuler des objets autrement difficiles à représenter
  - ex: feu, nuage, feux d'artifice, arbres, herbe, rivière, etc.
- Une particule évolue dans le temps
  - création, mouvement, interaction, disparition
- Une particule peut contrôler n'importe quel paramètre
  - ex: direction, couleur, opacité, grosseur, primitive, etc.

# Particules



Fig. 12. *white.sand.*

Reeves



# Particules



Reeves

BLOB WARPING TECHNIQUE

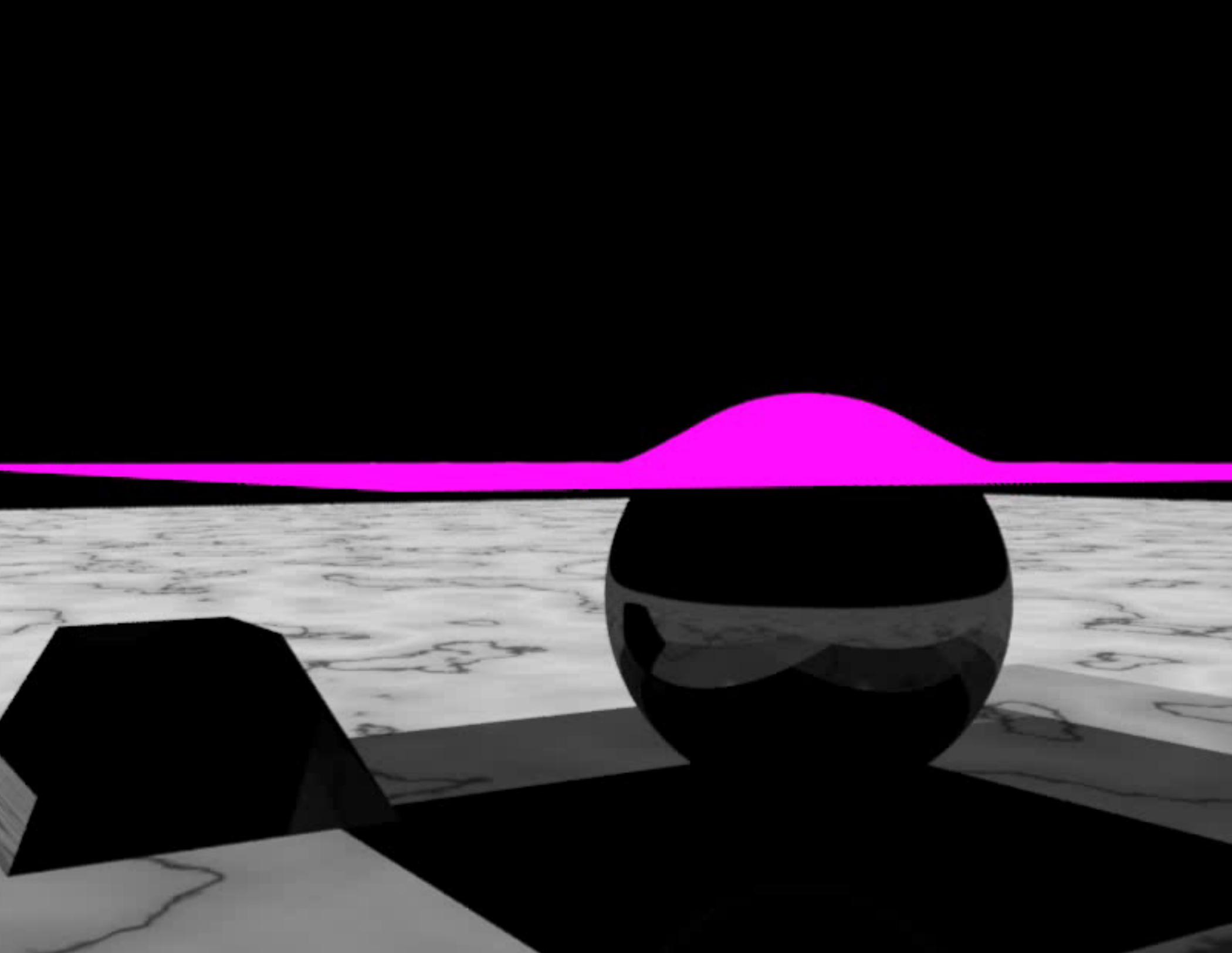


no warping



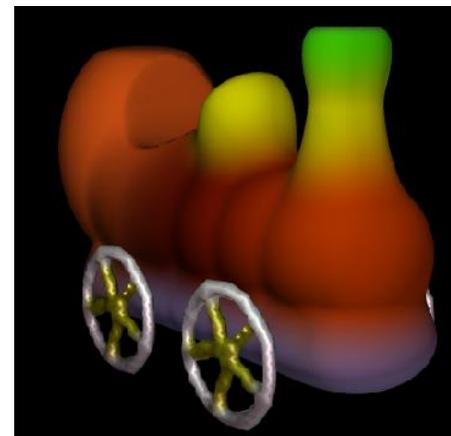
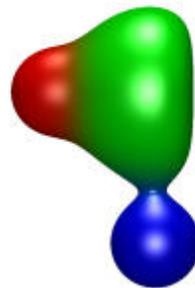
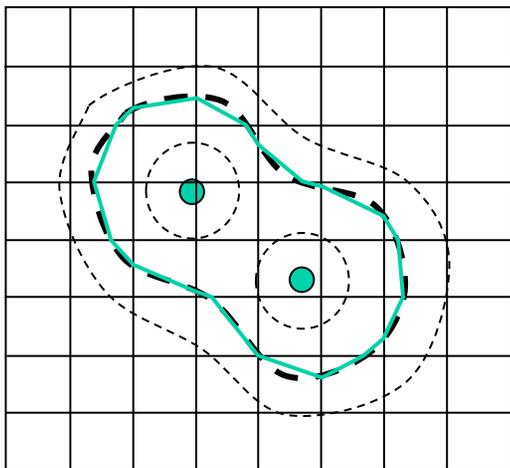
warping

Stam



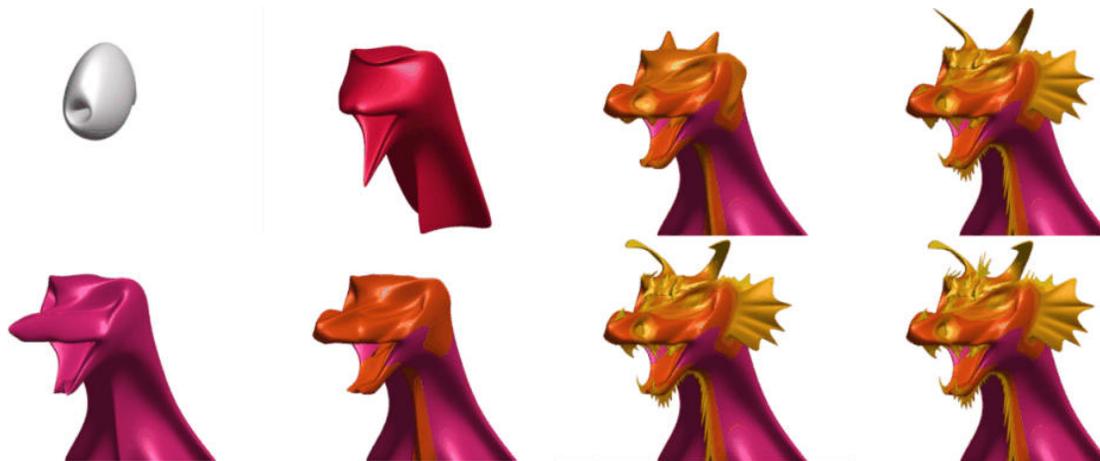
# Surfaces implicites (*objets mous, blobbies, metaballs*)

- Isosurface extraite d'un champ de densité
- Quelques charges (positives ou négatives) suffisent à produire un champ complexe
- Squelette contrôle la forme de la surface
- Rendu par polygonisation (*marching cube*) ou lancer de rayons par intervalles ou analytique



# Splines hiérarchiques

- Différentes densités de points de contrôle à chaque niveau de la hiérarchie



Forsey

