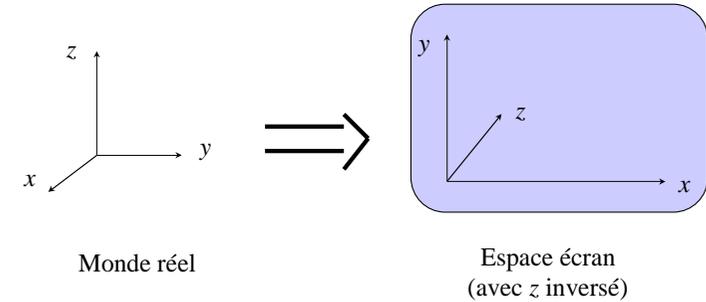


## Chapitre IV

### Introduction à la synthèse d'images

## Systèmes de coordonnées



## Introduction à la Synthèse d'Images

- 4.1 – Géométrie algorithmique 3D
- 4.2 – Recherche du réalisme
- 4.3 – Construction des scènes et des paysages
- 4.4 – Conclusion

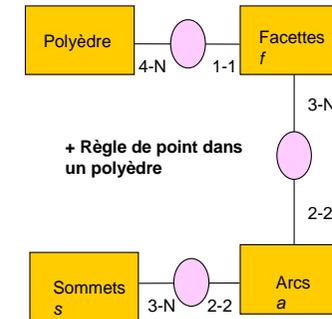
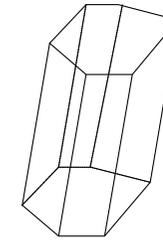
## 4.1 – Géométrie algorithmique 3D

- 4.1.1 - Introduction à la modélisation géométrique 3D (polyèdres, octrees, CSG)
- 4.1.2 - Transformation et perspectives
- 4.1.3 - Introduction à l'élimination des parties cachées

### 4.1.1 - Introduction à la modélisation géométrique 3D

- Polyèdres
- Octrees
- CSG
- Surfaces gauches

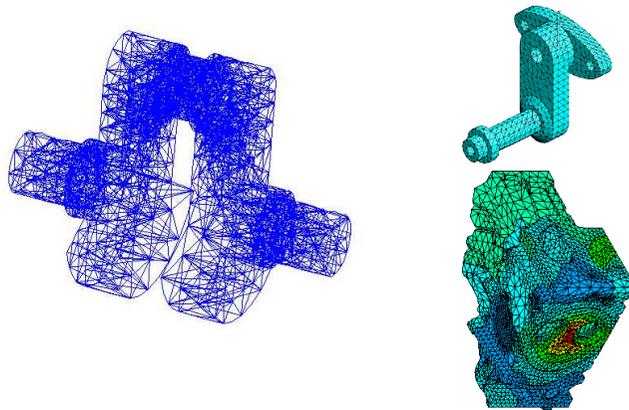
### Polyèdres



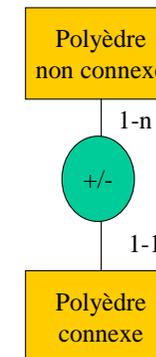
+ Règle de point dans un polyèdre

Règle d'Euler-Poincaré pour un solide simple :  $f+s-a=2$

### Exemples éléments finis

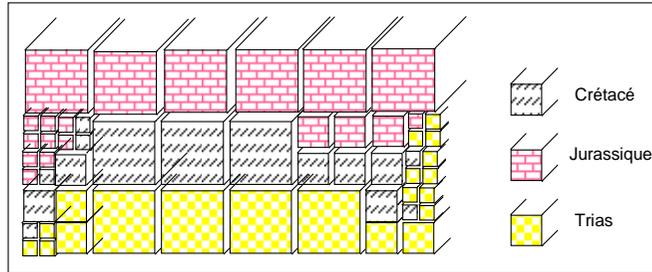


### Polyèdre non connexe





## Couches géologiques



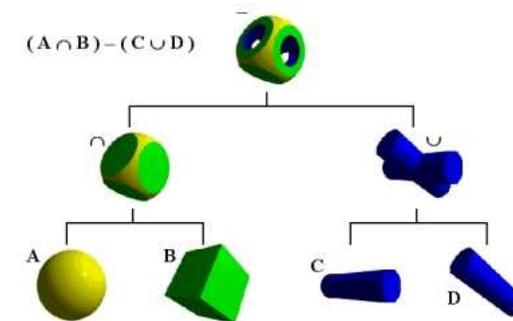
## CSG

- Constructive Solid Geometry
- « *Tout objet complexe est vu comme le résultat d'une opération ensembliste* »
- Formes de base simples : cube, sphère, cylindre, tore, boîte, etc.
- Opérations booléennes : union, intersection, différence

## Grammaire

- $\langle \text{solide} \rangle ::= \langle \text{solide} \rangle \langle \text{op bool} \rangle \langle \text{solide} \rangle$   
 $|\langle T(\langle \text{solide} \rangle) \rangle$   
 $|\langle \text{forme élémentaire} \rangle$
- $\langle \text{op bool} \rangle ::= \cap | \cup | -$
- $\langle \text{forme élémentaire} \rangle ::= \langle \text{cube} \rangle | \langle \text{sphère} \rangle$   
 $|\langle \text{cylindre} \rangle | \langle \text{tore} \rangle$   
 $|\langle \text{boîte} \rangle \dots$

## Arbre de construction



## Surfaces gauches

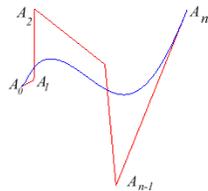
- Quadriques, coniques
- Patches bilinéaires
- Surfaces splines, de Bézier, etc.

## Surfaces de Bézier

- Pierre Bézier (1962) Renault
- Paul Faget de Casteljaou (1959) Citroën
- Points de contrôle
  - Forme originale lisse
  - Ne passe pas par les points, sauf les extrémités

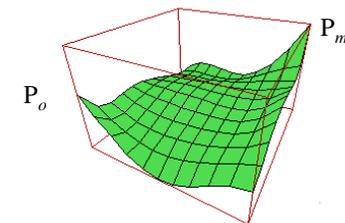
## Courbes de Bézier

$$B(t) = \sum_{i=0}^N P_i \frac{N!}{i!(N-i)!} t^i (1-t)^{N-i}$$



## Surface de Bézier

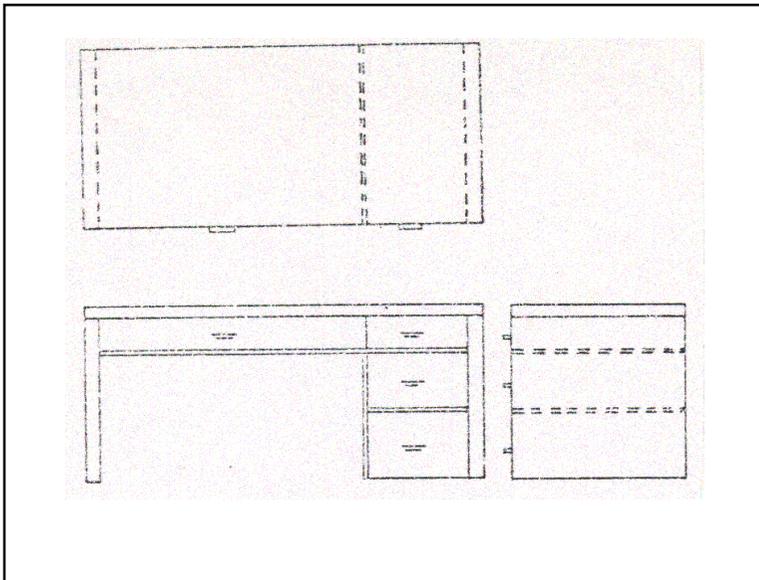
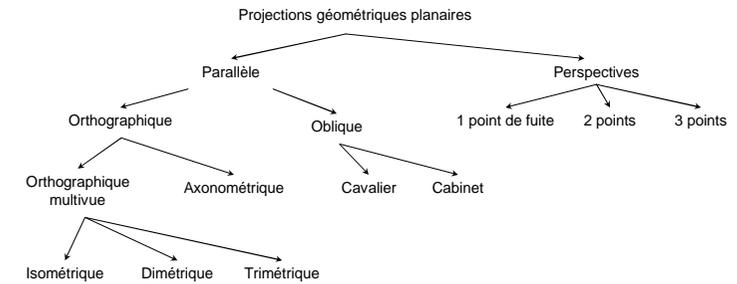
$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i} \frac{n!}{j!(n-j)!} v^j (1-v)^{n-j} P_{ij}$$



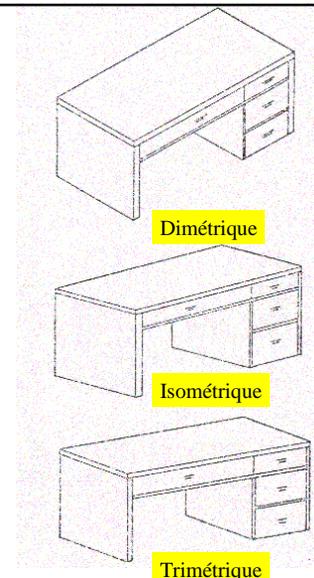
## 4.1.2 - Transformations et perspectives

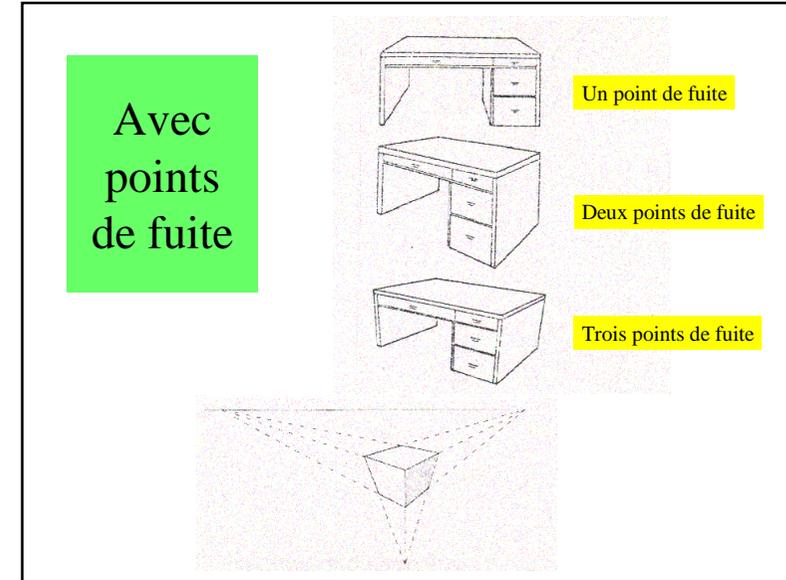
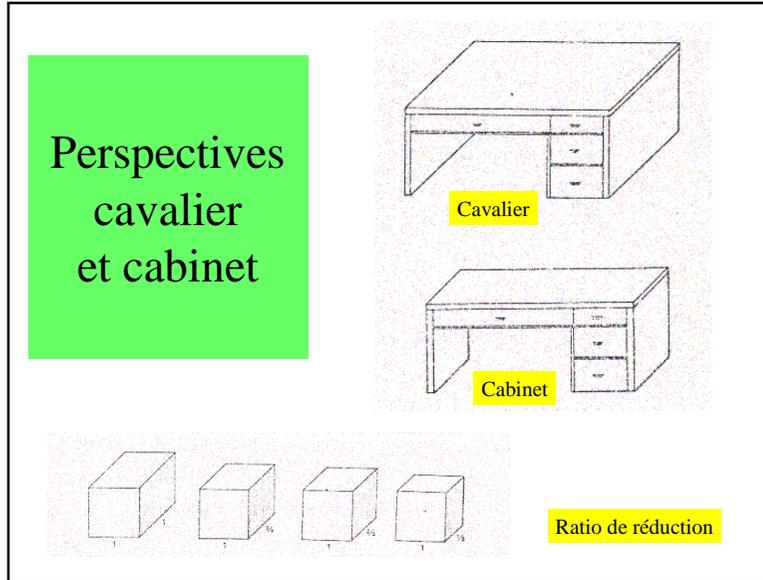
- Nécessité de calculer les perspectives
- Calcul matriciel classique :
  - Inconvénient : additions et multiplications
- Matrice homogène

## Projections et perspectives planaires



## Rotations dans l'espace





**Matrice homogène**

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

← Matrice classique

$$[x \ y \ z] \rightarrow [x \ y \ z \ 1]$$

Coordonnées classiques    Coordonnées homogènes

Tout point de l'espace homogène a son correspondant dans l'espace

$$[x/w \ y/w \ z/w] \leftarrow [x \ y \ z \ w]$$

**Matrice homogène**

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & p_1 \\ a_{21} & a_{22} & a_{23} & p_2 \\ a_{31} & a_{32} & a_{33} & p_3 \\ t_1 & t_2 & t_3 & e \end{bmatrix}$$

Perspectives (points  $p_1, p_2, p_3$ )

Facteur d'échelle ( $e$ )

Translations ( $t_1, t_2, t_3$ )

$$[X \ Y \ Z \ W] = [x \ y \ z \ 1] M$$

Nouveau                      Ancien

## Exemple translation

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}$$

Résultat :

$$[x+l \quad y+m \quad z+n \quad 1] = [x \quad y \quad z \quad 1]T$$

## Exemple perspectives

Soit 3 points de fuite :

$$\begin{aligned} x &: [1/p \quad 0 \quad 0] \\ y &: [0 \quad 1/q \quad 0] \\ z &: [0 \quad 0 \quad 1/r] \end{aligned}$$

La matrice homogène de transformation est la suivante :

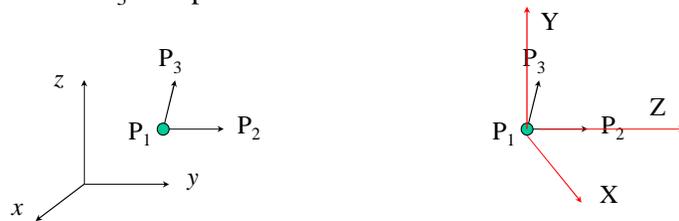
$$T = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ce qui donne le résultat suivant après postmultiplication

$$\begin{bmatrix} x & y & z & px+qy+rz+1 \\ \frac{x}{px+qy+rz+1} & \frac{y}{px+qy+rz+1} & \frac{z}{px+qy+rz+1} & \end{bmatrix}$$

## Calcul pratique de perspective

- Trois points
  - $P_1$  : œil de l'observateur
  - $P_2$  : un point regardé (direction du regard)
  - $P_3$  : chapeau de l'observateur



## Opérations

- 1 – Translation : origine en  $P_1$   
 $\overrightarrow{P_1P_2}$
- 2 – Rotation axe  $y \rightarrow$   $\overrightarrow{P_1P_3}$  dans le plan  $yz$
- 3 – Rotation axe  $x$  pour  $-z$
- 4 – Rotation axe  $z$  dans le plan  $yz$

### Tout calcul fait, on obtient

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_1 & -y_1 & -z_1 & 1 \end{bmatrix} \quad R = \begin{bmatrix} r_{1x} & r_{1y} & r_{1z} & 0 \\ r_{2x} & r_{2y} & r_{2z} & 0 \\ r_{3x} & r_{3y} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r_z = \frac{-\overrightarrow{P_1P_2}}{\|\overrightarrow{P_1P_2}\|} \quad r_x = \frac{\overrightarrow{P_1P_2} \wedge \overrightarrow{P_1P_3}}{\|\overrightarrow{P_1P_2} \wedge \overrightarrow{P_1P_3}\|} \quad r_y = r_z \wedge r_x$$

### Suite et fin

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad z : [0 \quad 0 \quad 1/r]$$

Selon les coordonnées du point de fuite, généralement selon l'axe  $z$

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} TRP$$

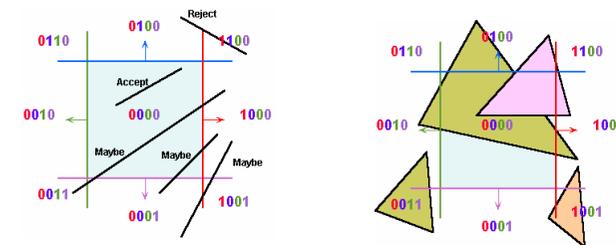
Nouvelles coordonnées

Anciennes coordonnées

### 4.1.3. Introduction à l'élimination des parties cachées

- Il existe plusieurs centaines d'algorithmes qui diffèrent selon :
  - La structure des données 3D
  - Le type de dessins en sortie
  - Le type de périphérique utilisé
- Exemples de méthodes
  - clipping
  - tracé de fonctions
  - algorithme du peintre

### Clipping 2D



## 3D clipping

<http://www.intel.com/technology/3d/docs/GEOMETRY.HTM>

## Fonctions de deux variables

•  $z=f(x,y)$

## Première solution

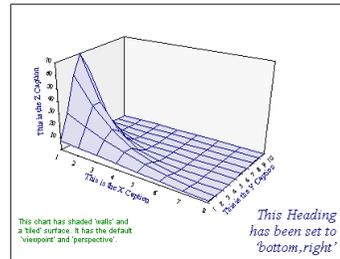
- Série de lignes en  $x$ 
  - commencer par le plus proche de l'observateur
  - gestion d'une ligne de crête

Situation	Visibilité	Dessin	Ligne de crête
	Visible		
	Caché	Rien à tracer	Ligne de crête inchangée
	Partiellement visible		

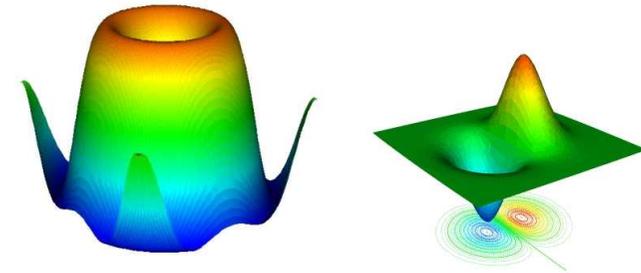
## Exemple de tracé

## Deuxième solution

- Maillage
- Gérer deux lignes de crêtes
  - l'une selon  $x$
  - l'autre selon  $y$



## Avec dégradés de couleurs



## Painter algorithm

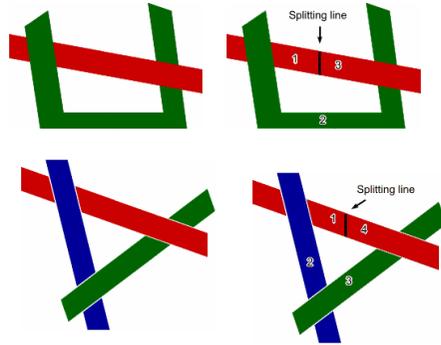
<http://www.cs.unc.edu/~davenc/Class/136/Lecture19/Painter.html>



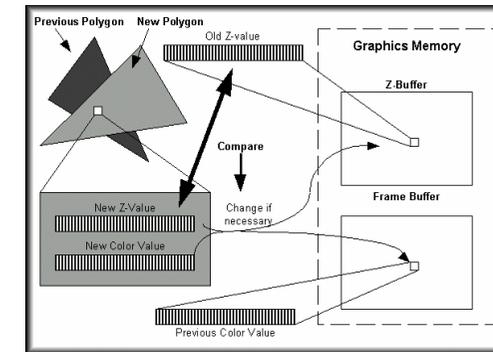
## Algorithme du peintre

- Conditions :
  - Périphériques permettant des taches de couleur
  - Scènes modélisées uniquement par facettes
- Algorithme
  - Calcul de toutes les facettes avec leur couleur et leur positionnement dans l'espace
  - Tri selon la profondeur
  - Affichage en partant des plus éloignés

## Algorithme du peintre



## Technique du z-buffer



## 4.2 – Recherche du réalisme

- 4.2.1. Aliasing
- 4.2.2. Dégradés de couleurs
- 4.2.3. Lancé de rayons
- 4.2.4. Radiosité
- 4.2.5. Textures et applications des textures sur les objets
- 4.2.6. Construction des scènes et des paysages

### 4.2.1. Aliasing

- Ce phénomène est lié aux effets causés par un sous-échantillonnage d'un signal

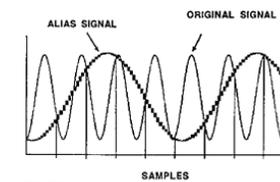
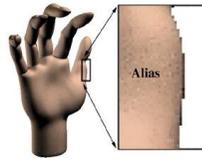


FIGURE 2: Aliasing

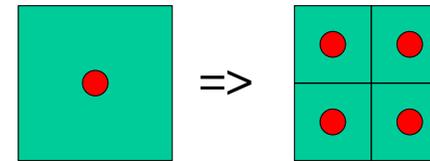
Un signal à 12 périodes a ici un « alias » de 2 périodes  
 Mécanisme étendu à 2 dimensions  
 Impossible à éliminer → réduction

- Effets
  - Images dentelées (escalier)
  - Petits objets disparaissant ou déformés
  - Animation par saut
  - Ondulation des textures
- Actions visant à la réduction
  - Calcul avec résolution supérieure (sur-échantillonnage)
    - (8 ou 16 calculs par pixels → moyenne)
  - Past-processing (blurring)
  - Filtage (convolution)
  - Suivi de contour
  - Etc.



## Antialiasing

- Sur-échantillonnage
  - régulier
  - quadtree
  - aléatoire uniforme sur  $[0, 1]^2$

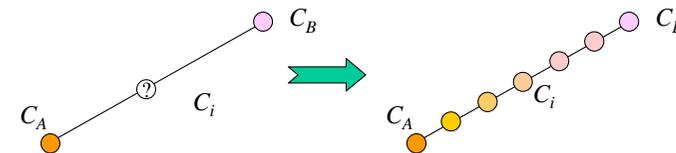


Si couleurs similaires  
alors aucun problème  
sinon redécoupage

## 4.2.2. Dégradés de couleurs

- Dégradés continus
- Interpolation linéaire de couleurs
- Soit 2 couleurs  $C_A(\alpha_A, \beta_A, \gamma_A)$  et  $C_B(\alpha_B, \beta_B, \gamma_B)$
- Résultats différents selon système de couleurs (RGB, HLS, etc.)
- Méthode Cosinus

## Interpolation linéaire de couleurs

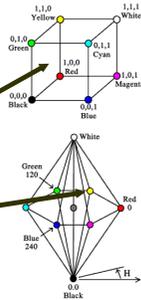


Si  $n$  intervalles, alors  
les couleurs intermédiaires  
 $C_i$  sont données par :

$$C_i = \begin{bmatrix} \alpha_A + \frac{i}{n+1} \times (\alpha_B - \alpha_A) \\ \beta_A + \frac{i}{n+1} \times (\beta_B - \beta_A) \\ \gamma_A + \frac{i}{n+1} \times (\gamma_B - \gamma_A) \end{bmatrix}$$

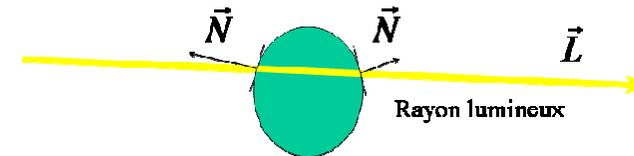
## Interpolations linéaires

- Attention : résultats différents selon les modèles de représentations des couleurs
- Ex Milieu entre C1= Rouge et C2=Vert
- RGB
  - C1(1,0,0), C2(0,1,0) => C3(1/2,1/2,0)
- HLS
  - C1(0,0,0), C2(120,0,0) => C3(60,0,0)



## Méthode Cosinus

- Dégradés de couleur sur un objet



## Calcul de la couleur

$\cos(\vec{N}, \vec{L}) > 0$  alors caché

$\cos(\vec{N}, \vec{L}) < 0$  alors visible

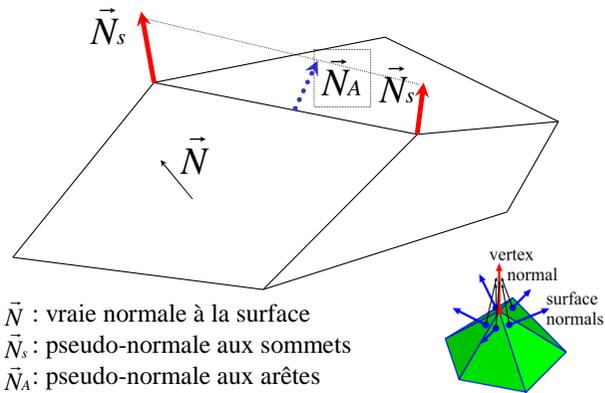
Indice de couleur  $c$  :

$$c = \frac{|\vec{N} \cdot \vec{L}|}{\|\vec{N}\| \|\vec{L}\|} \quad \text{avec} \quad 0 \leq c \leq 1$$

## Méthode de Gouraud

- Seulement sur surfaces polyédriques
- Objectifs : éviter les sauts de couleurs par un lissage des normales
- Départs : normales en surfaces
  - calcul de pseudo normales aux sommets
  - calcul de pseudo normales aux arêtes

## Normales et pseudo-normale



## Calcul des pseudo-normales

Pseudo-normale sommet = Moyenne des vraies normales des faces environnantes

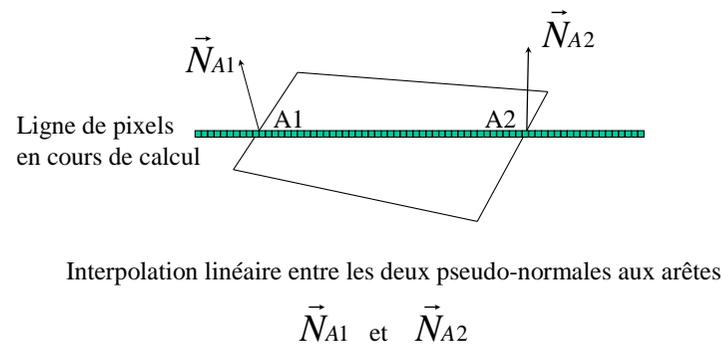
$$\vec{N}_s = \frac{1}{n} \sum_i \vec{N}_i$$

Pseudo-normale arête = interpolation linéaire entre les pseudo-normales sommets ( $S_1$  et  $S_2$ )

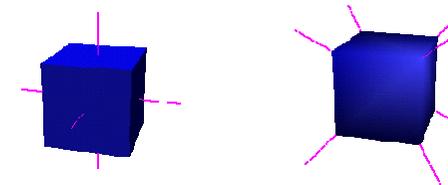
$$\vec{N}_i = \vec{N}_{S_1} + \frac{i}{k+1} \times (\vec{N}_{S_2} - \vec{N}_{S_1})$$

$k$  = nombre de pixels entre  $S_1$  et  $S_2$

## Calcul définitif des normales



## Exemple



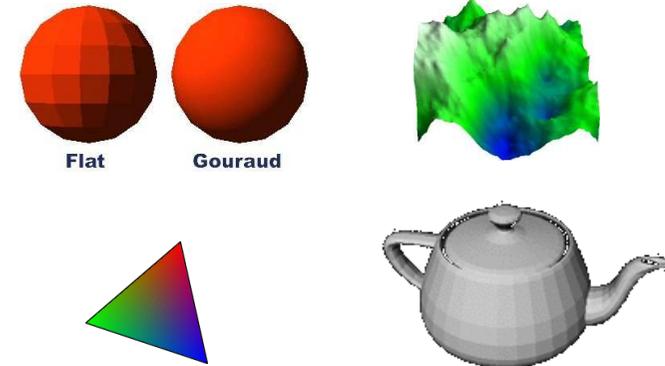
## Calcul définitif des couleurs

- Usage de la formule :

$$c = \frac{\vec{N} \cdot \vec{L}}{\|\vec{N}\| \|\vec{L}\|}$$

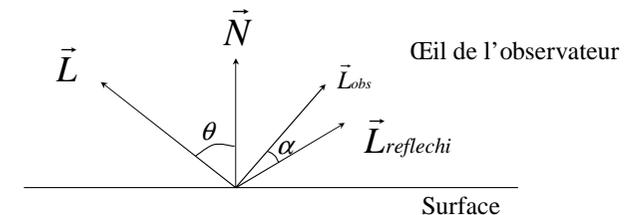
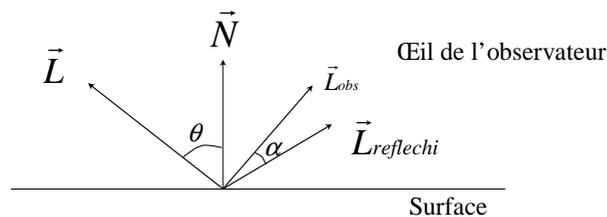
- Ainsi sont calculées les 3 composantes de la couleur de chaque pixel

## Exemples GOURAUD



## Méthode de Bui-Tuong-Phong

- Usage de la théorie de la lumière
- Loi de Lambert sur l'illumination diffuse



Illumination diffuse :  $I_d = I_p \times K_d \times \text{Cos}(\theta)$

Intensité      Coefficient de réflexion diffuse

Intensité lumineuse en un point en fonction de l'observateur :

$$I = I_a K_a + \frac{I_p \times K_d \times \text{cos}(\theta)}{r + k}$$

Lumière ambiante      Distance à l'observateur      Constante

Lumière réfléchié :  $K_s(\cos(\alpha))^n$

$$n \in [1, 10]$$

Mat Brillant

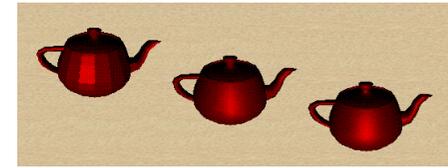
Processus soustractif → Cyan  
Magenta  
Jaune

$$I_{cyan} = I_{cyan_s} K_{cyan} + \frac{I_{pcyan}}{r+k} \times [K_{dcyan} \times \cos(\theta) + K_s(\cos\alpha)^n]$$

$$I_{magenta} =$$

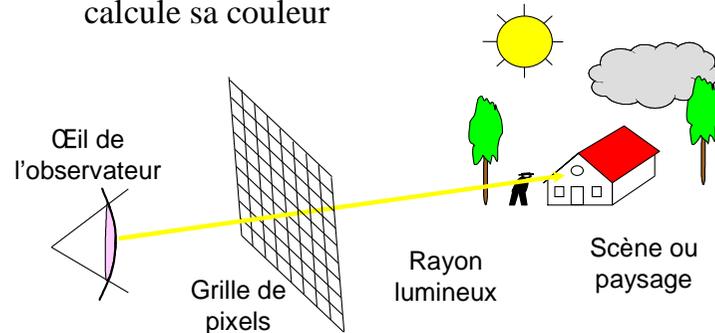
$$I_{yellow} =$$

## Facettes, Gouraud et Bui-Tuong-Phong



### 4.2.3. Lancé de rayons

- Principe : on lance un rayon par pixel et on calcule sa couleur



### Algorithme de base

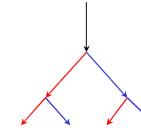
- pour chaque pixel de la grille
  - calcul de l'équation du rayon
  - pour chaque objet de la scène
    - calcul des points d'intersection objet/rayon
  - garder la plus proche intersection
  - calcul de la couleur de ce point
  - affichage
- fin pour

## Remarques

- Technique chère et longue
- Améliorations :
  - rayons secondaires (réflexion et réfraction)
  - multiplicité des lumières
  - ombres portées
  - anti-aliasing
- Difficultés principales
  - intersection rayon/objet

## Arbre de rayons secondaires

- Tenir compte de réflexions et des réfractions



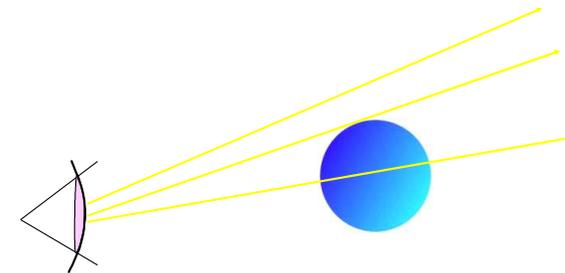
Exemple  
Réflexion : 80 %  
Réfraction : 10 %  
Perte : 10 %

- Arbre binaire de rayons
  - nombre de niveaux : de 3 à 5
  - % d'un seuil
- remonter l'arbre => calcul de chaque contribution

## Calculs d'intersection

- La phase la plus longue
- dépend des modèles des objets
  - ex : modèles polyédriques
  - pb : comment déterminer rapidement la face percée par le rayon
- nécessité d'accélérateurs
  - boîtes englobantes
  - sphères englobantes

## Intersection sphère/rayon lumineux



## Calcul des intersections

Origine du rayon  $R_0 = [X_0, Y_0, Z_0]$

Direction du rayon  $R_d = [X_d, Y_d, Z_d]$

Normalisation :  $X_d^2 + Y_d^2 + Z_d^2 = 1$

Rayon  $R(t) = R_0 + R_d \times t$  avec  $t > 0$

$$R(t) = [X_0 + X_d \times t, Y_0 + Y_d \times t, Z_0 + Z_d \times t]$$

## Avec une sphère

Centre de la sphère  $S_c = [X_c, Y_c, Z_c]$

Rayon  $S_r$

Sphère :  $[X_s, Y_s, Z_s]$

Equation de la sphère :  $(X_s - X_c)^2 + (Y_s - Y_c)^2 + (Z_s - Z_c)^2 = S_r^2$

⇒ Equation à résoudre pour intersection

$$(X_0 + X_d t - X_c)^2 + (Y_0 + Y_d t - Y_c)^2 + (Z_0 + Z_d t - Z_c)^2 = S_r^2$$

$$\Rightarrow At^2 + Bt + C = 0 \quad \text{avec} \quad A = 1$$

$$\text{Solution } t_i = \frac{-B \pm \sqrt{B^2 - 4C}}{2} \quad \Delta = B^2 - 4C$$

Si  $\Delta < 0$  aucune intersection

Si  $\Delta > 0$  alors on choisit la solution

la plus proche de l'observateur

Le point d'intersection est

$$r_i = [x_i, y_i, z_i] = [X_0 + X_d t_i, Y_0 + Y_d t_i, Z_0 + Z_d t_i]$$

$$\text{normale : } r_n = \left[ \frac{x_i - X_c}{S_r}, \frac{y_i - Y_c}{S_r}, \frac{z_i - Z_c}{S_r} \right]$$

Cas le plus défavorable :  $17\pm, 17\times, 1\sqrt{\quad}, 3\text{if}$

## Avec un plan

plan :  $Ax + By + Cz + D = 0$

avec  $A^2 + B^2 + C^2 = 1$

vecteur normal :  $P_n = [A, B, C]$

$$\text{intersection } t = \frac{-(AX_0 + BY_0 + CZ_0 + D)}{AX_d + BY_d + CZ_d}$$

$$t = \frac{-(P_n \cdot R_0 + D)}{P_n \cdot R_d}$$

$\Rightarrow r_i = [x_i, y_i, z_i]$   
 $= [X_0 + X_{dt}, Y_0 + Y_{dt}, Z_0 + Z_{dt}]$   
 Bilan : 8±, 9×/, 3if

### Avec un polyèdre

- Décrit par un ensemble de polygones (le plus souvent des triangles)
- Pb : quel polygone est intercepté ?
- Usage du théorème de Jordan dans l'espace
  - rotation du polygone axe : rayon
  - projection du polygone dans plan perpendiculaire

### CSG

### Diagramme de Roth

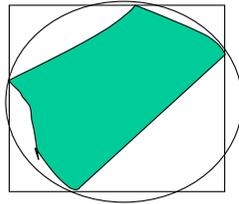
Solide =  $(A-B) \cup D$

A B D

A	—————
B	—————
C=A-B	—————
D	———
E=C union D	—————

## Techniques d'accélération

- Extent boîte, sphère ou ellipsoïde



- Arbres d'extents

## Arbres d'extents

Arbres de rectangles  
R-tree



Arbres de sphères  
Sphere-tree



## Applications

- Scènes polyédriques
  - 1 sphère englobante globale pour la scène
  - 1 sphère par objet
  - 1 sphère (projection => un cercle) par polygone
- Scènes CSG
  - Arbres en ne considérant que les unions

## Quelques images

<http://jedi.ks.uiuc.edu/~johns/raytracer/>



2328 triangles « smoothed »



[http://graphics.cs.uni-sb.de/~wald/Publications/2002\\_IGI/Images/teas2.png.jpg](http://graphics.cs.uni-sb.de/~wald/Publications/2002_IGI/Images/teas2.png.jpg)



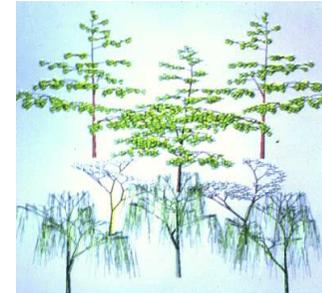
[http://graphics.cs.uni-sb.de/Publications/TR/2002/Dyn/teas\\_terrain1.jpg](http://graphics.cs.uni-sb.de/Publications/TR/2002/Dyn/teas_terrain1.jpg)



<http://www.trl.ibm.com/projects/s7340/render/rayE.htm>



<http://www.trl.ibm.com/projects/s7340/render/natureE.htm>



## A Day in a Living Room

<http://www.trl.ibm.com/projects/s7340/render/roomE.htm>



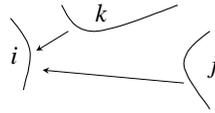
## Conclusions sur lancé de rayons

- Technique longue et coûteuse
- Améliorations possibles
  - Multiplicité des lumières
  - Pénombre
  - Etc.

## 4.2.4. Radiosité

- Principe
  - échange d'énergie entre des patches voisins ou parallèles

- Formule de base



Radiosité × Surface = Energie émise + Energie réfléchie

- Travail sur de petits patches

## Formule fondamentale

- Patch  $P_i$  et  $P_j$  de surface  $dA_i$  et  $dA_j$
- Radiosité  $B_i$  et  $B_j$  par éléments de surface

$$B_i dA_i = E_i dA_i + \rho_i \int_j B_j dA_j F_{dA_j dA_i}$$

Lumière émise  
par unité  
de surface

Réfectivité

Facteur de forme

- Environnement fermé => équilibre
- Environnement discrétisé

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j F_{A_j A_i} A_j$$

- Relation de réciprocité

$$F_{ij} A_i = F_{ji} A_j$$

- Ce qui donne

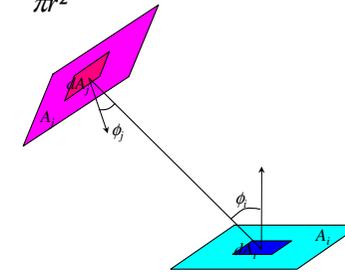
$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij}$$

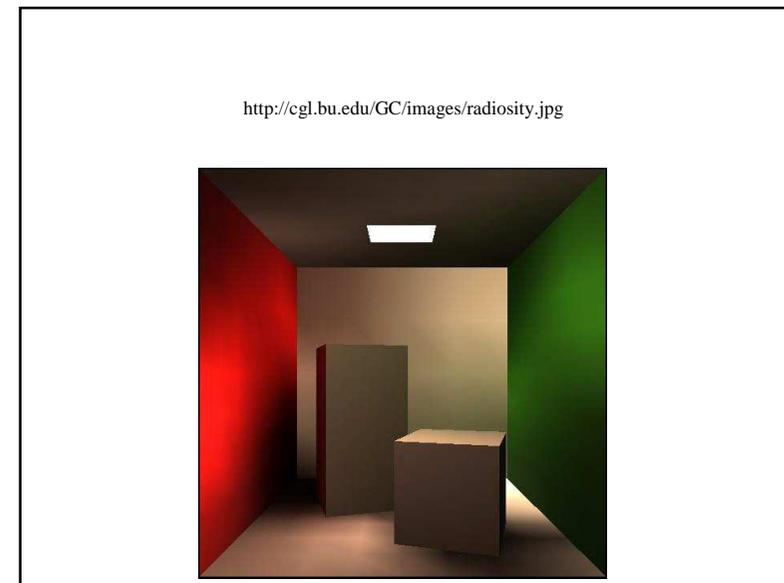
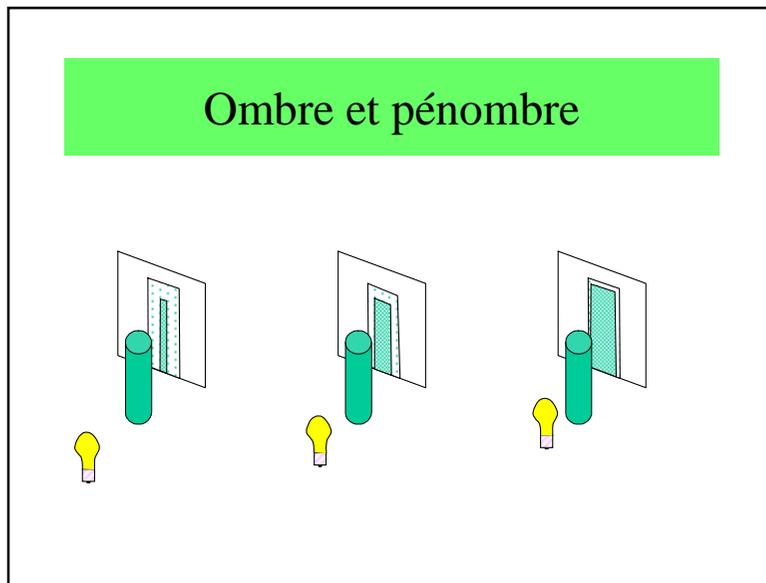
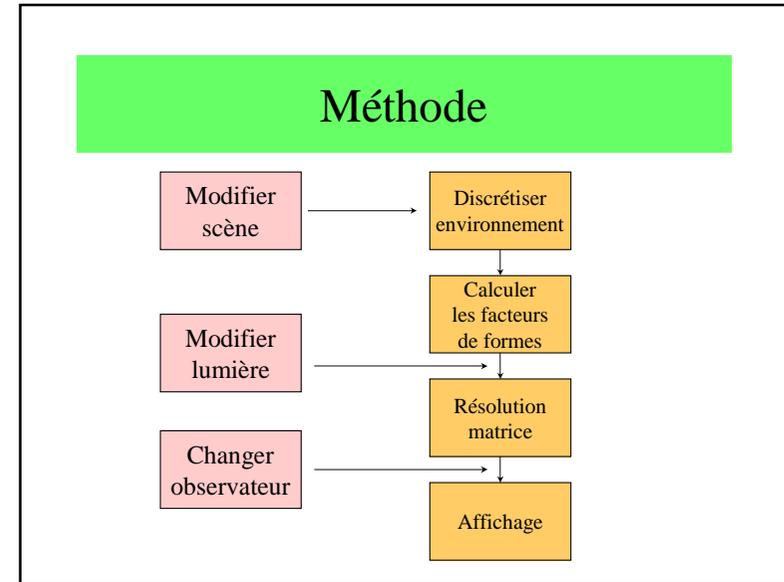
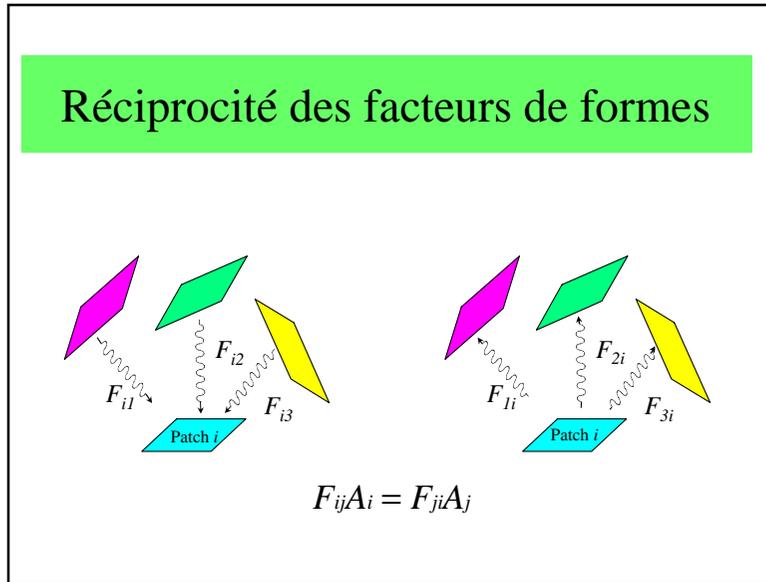
- n équations à n inconnues  $B_i$
- comme  $\rho_i < 1 \rightarrow$  convergence par Gauss Seidel

## Facteurs de formes

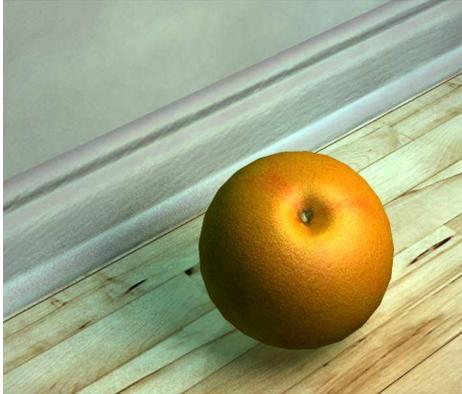
$$F_{dA_i dA_j} = \frac{\text{Energie atteignant } dA_j \text{ venant de } dA_i}{\text{Energie totale quittant } dA_i \text{ dans toutes les directions}}$$

$$F_{dA_i dA_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_i dA_j \approx F_{ij}$$





<http://www.magiccreations.it/servizi/imgs/softimage/radiosity.jpg>



<http://www.trl.ibm.com/projects/s7340/render/radiosE.htm>



[http://www15.brinkster.com/phorta/sites\\_incriveis.asp](http://www15.brinkster.com/phorta/sites_incriveis.asp)



## Lancé de rayon et radiosità



<http://vcg.iei.pi.cnr.it/lescienze.html>

## 4.2.5. Textures

- Différentes problématiques
  - génération des textures
  - placage
  - mise en perspectives

## Placage des textures

- Wallpapering
- Stretching



## Modélisation des textures

- Analyse des textures réelles
- Transformée de Fourier à 2D
- Simulation avec 3 à 7 termes + composante aléatoire
  - inconvénient effet tartan
- Textures fractales

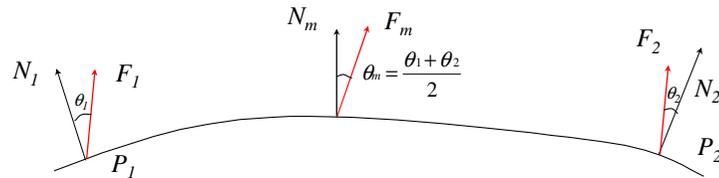
## Exemple Texture de Fourier



<http://www.cis.ohio-state.edu/~agarcia/research/volrendering/volrendering.htm>

## Textures fractales

- Deux approches
  - génération des facettes par la méthode de Carpenter ; génération des normales et couleurs
  - perturbation des normales (BARSKY et HARUYAMA)



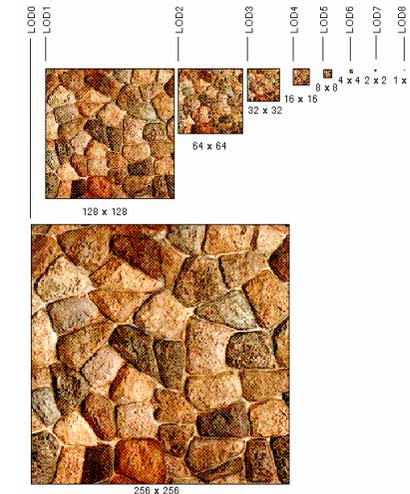
## Normales perturbées

- Loi initiale
  - Loi  $(\theta) = \text{Gauss}(\theta, \sigma_\theta)$
- A l'étape suivante
  - Loi  $(\theta_m) = \text{Gauss}(1/2(\theta_{1+} + \theta_2), \sigma_\theta 2^{-h})$
  - avec  $h$  paramètre fractal
- Ensuite
  - Loi  $(\theta_m) = \text{Gauss}(1/2(\theta_{m1} + \theta_{m2}), \sigma_\theta 2^{-lh})$
  - avec  $l$  niveau de récursité

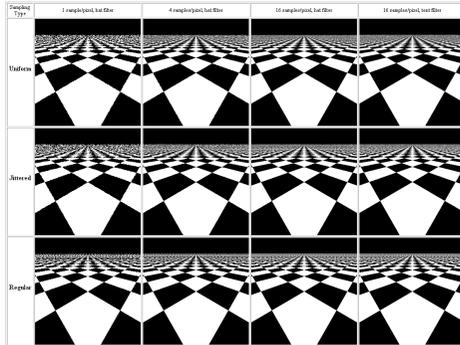
## Textures fractales



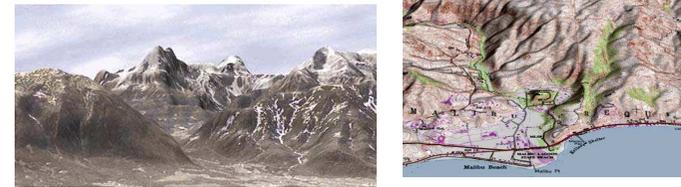
## Mapping textures



## Mise en perspective des textures



## Drapage de texture



<http://www.mpassociates.gr/software/environment/truflite3.jpeg>

## 4.3 – Construction des scènes et des paysages

### 1 - Choix des objets

Bibliothèque d'objets élémentaires

Choix des méthodes de représentation

Sélection des objets

- formes
- localisation
- couleurs
- textures

Positionnement du fond, sol, soleil, objets (paysage, décor)

### 2 - Choix des lumières

Position, intensité, type de couleur

## suite

### 3 - Position de l'observateur / Caméra

choix de la perspective

### 4 - Calcul de l'image

choix écran et du degré de réalisme

choix de la méthode de calcul (z-buffer, lancé de rayon, radiosité, etc.)

calcul des couleurs (lumière, ombres, pénombre, objets cachés, etc.)

Affichage du résultat si temps réel, autrement archivage dans fichier

### 5 - Affichage du résultat

## 4.4 – Conclusion

- Techniques coûteuses
- Animation
- Multiprocesseurs
- Réalité virtuelle
- Réalité augmentée

## Road to Point Reyes



## Forêt fractale

