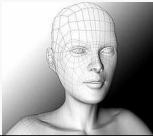
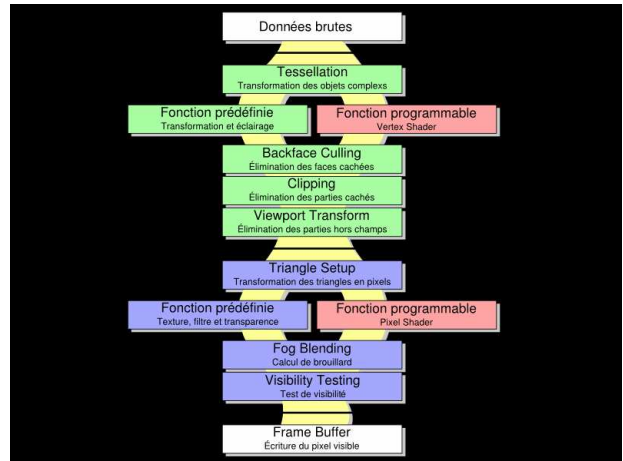


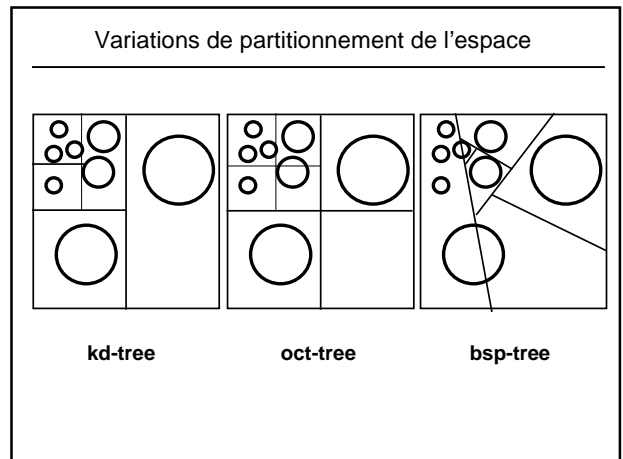
Processus de rendu

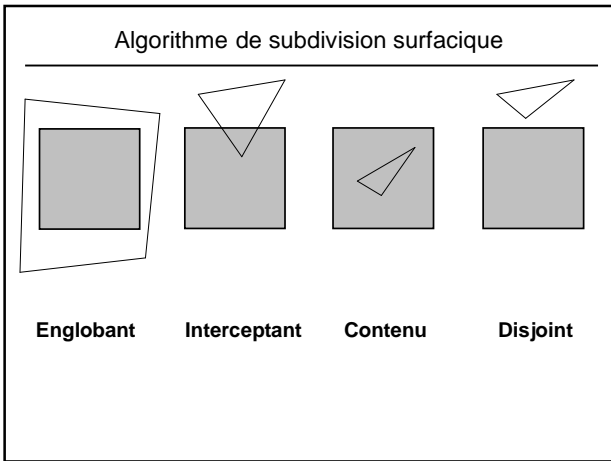
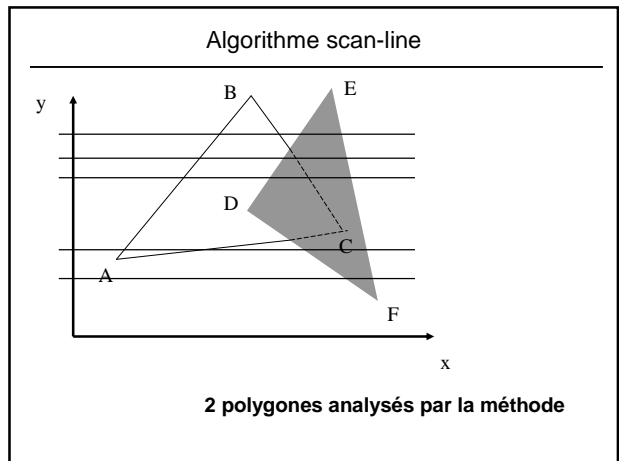
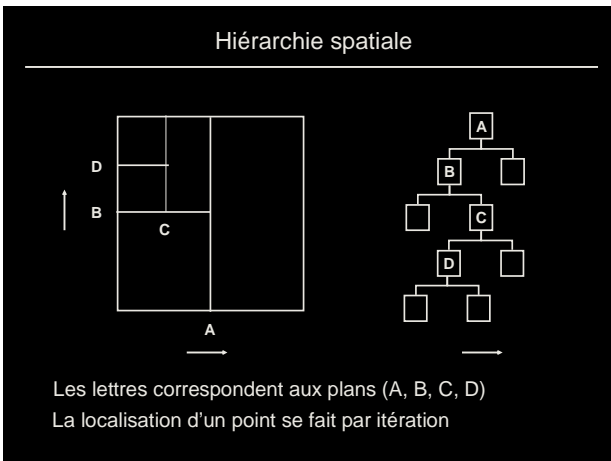
- 1- Détermination de visibilité des surfaces
 - déterminer quelle portion de la primitive est réellement visible à partir du point de vue de la caméra
- 2- Scan conversion – numérisation
 - déterminer les pixels couverts par l'image d'une primitive
- 3- Ombrage
 - déterminer la couleur à attribuer à chaque pixel

Détermination de visibilité des surfaces

- 1- Principe du BSP-Tree
 - arbre de partitionnement – espace binaire (Schumaker – 1969)
- 2- Algorithme scan – line
 - « cluster » : groupe de surfaces organisées en structure arborescente hiérarchisée, selon leur positionnement
- 3- Algorithmes de subdivision surfacique
 - partitionnement de l'espace dans le plan de projection (Warnock – 1969; Weiler-Atherton-1977)
 - algorithme anticrénelage (Catmull – 1978)





Numérisation

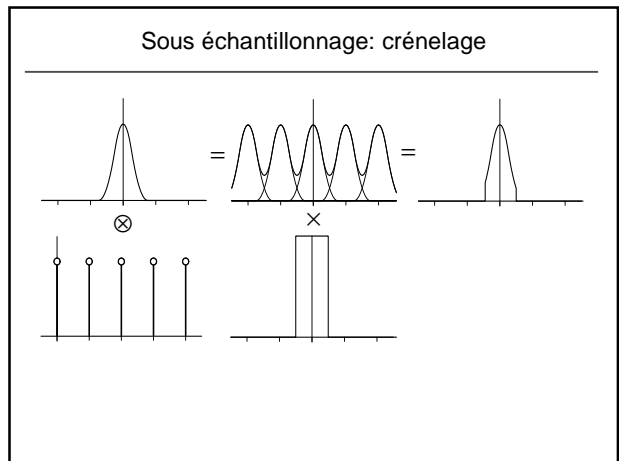
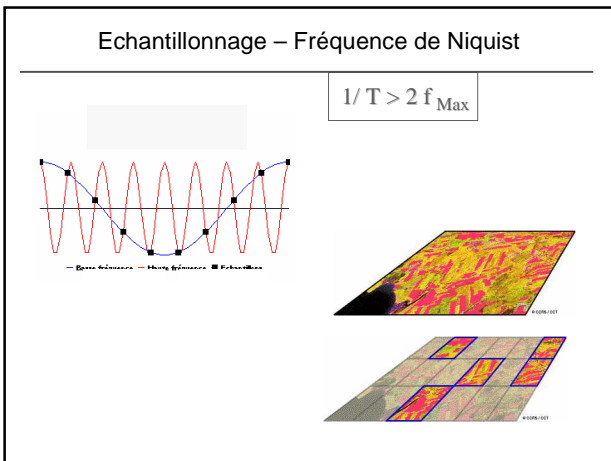
Numérisation : passage d'un signal continu vers un signal à valeurs discrètes (pixels)
Basée sur la transformée de Joseph Fourier

Domaine spatial

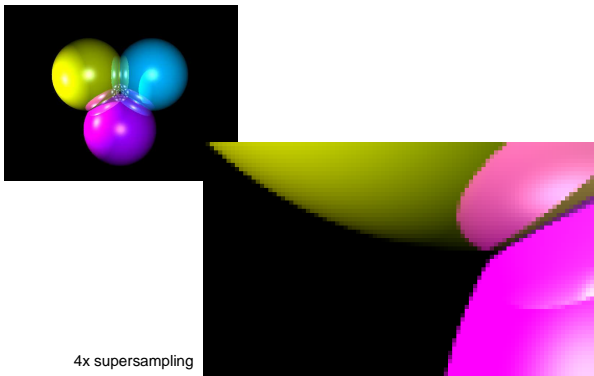
$$F(f(t)) = F(\omega) = -\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

$$F^{-1}(F(\omega)) = f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

Domaine fréquentiel



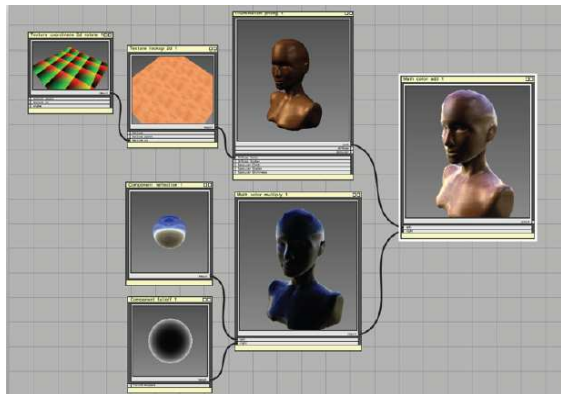
Superéchantillonnage: Antialiasing



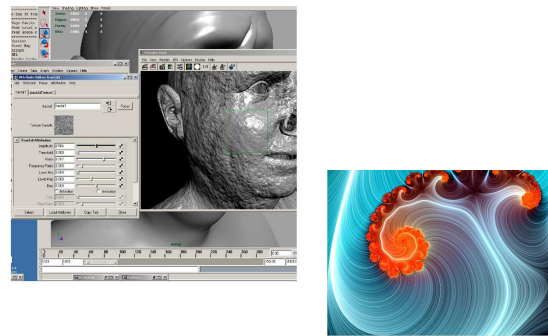
Mapping: Texture et rugosité - Historique

- ⇒ Catmull / Williams 1974 - basic idea
- ⇒ Blinn and Newell 1976 - basic idea, reflection maps
- ⇒ Blinn 1978 - bump mapping
- ⇒ Williams 1978, Reeves *et al.* 1987 - shadow maps
- ⇒ Smith 1980, Heckbert 1983 - texture mapped polygons
- ⇒ Williams 1983 - mipmaps
- ⇒ Miller and Hoffman 1984 - illumination and reflectance
- ⇒ Perlin 1985, Peachey 1985 - solid textures
- ⇒ Greene 1986 - environment maps/world projections
- ⇒ Akeley 1993 - Reality Engine
- ⇒ Light Field

Shader et placage de texture

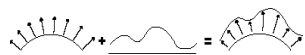


Texture procédurale



Utilisation de fractales

Bump Mapping / Rugosité



$$\mathbf{P}(u, v)$$

$$\mathbf{S}(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} \quad \mathbf{T}(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v}$$

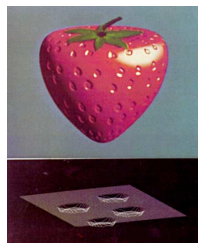
$$\mathbf{N}(u, v) = \mathbf{S} \times \mathbf{T}$$

Déplacement

$$\mathbf{P}'(u, v) = \mathbf{P}(u, v) + h(u, v)\mathbf{N}(u, v)$$

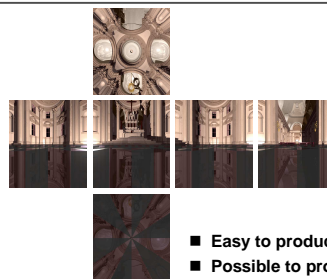
Perturbation de la normale

$$\begin{aligned} \mathbf{N}'(u, v) &= \mathbf{P}'_u \times \mathbf{P}'_v \\ &= \mathbf{N} + h_u(\mathbf{T} \times \mathbf{N}) + h_v(\mathbf{S} \times \mathbf{N}) \end{aligned}$$



Blinn 1976

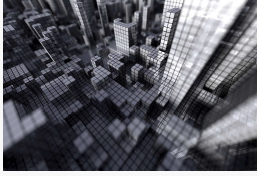
Cubical Environment Map



- Easy to produce with rendering system
- Possible to produce from photographs
- "Uniform" resolution
- Simple texture coordinates calculation

Post-process: optimisation du modèle de caméra

1- profondeur de champ (DOF depth of field)



2- flou de bougé volontaire (motion blur)



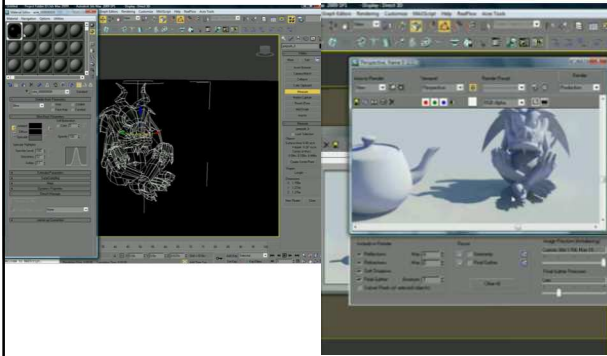
Moteur de rendu Mental Ray



Particularités :

- rendu proche de la réalité physique :
 - GI, caustiques, FG
 - réflexions et réfractions par tirs de rayons
- fonction de calcul des caustiques par photon mapping
- caméra avec cible et effet à profondeur de champ
- optimisation par back-face culling

Tutorial Mental Ray – Arch&Design – 3DS Max



Matériaux Arch & Design de Mental Ray par B. Chauvet, sur <http://vimeo.com/3595981>