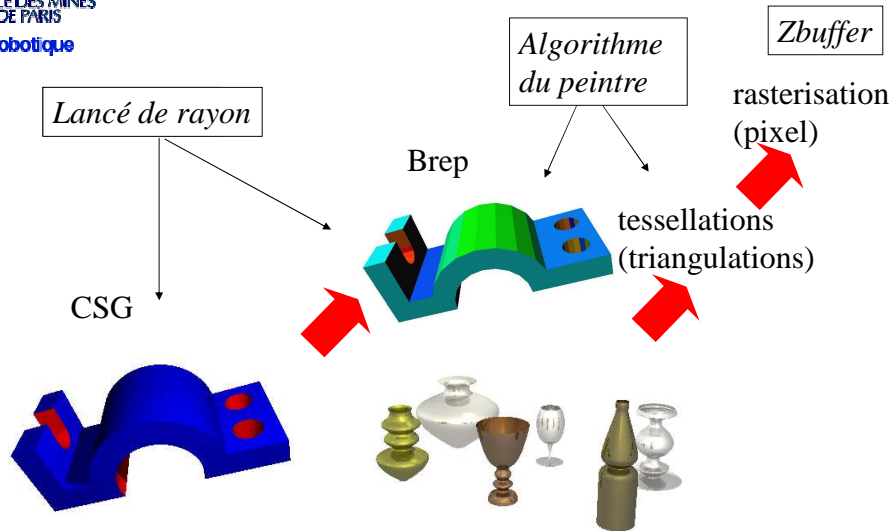


Les « objets géométriques »

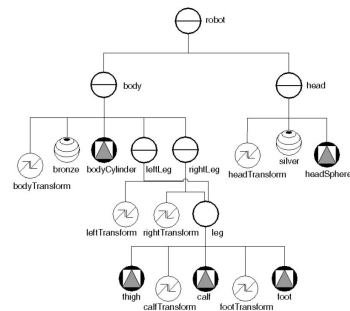


Synthèse d'images

6

[Skip →](#)

ORGANISATION D'UNE SCÈNE

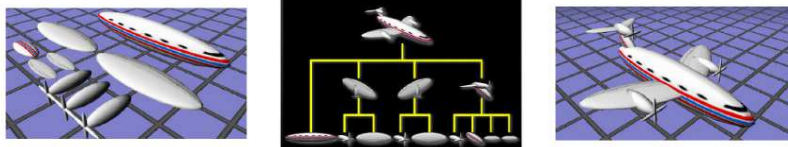


Modélisation-géométrique - O.Stab

8

Organisation d'une scène

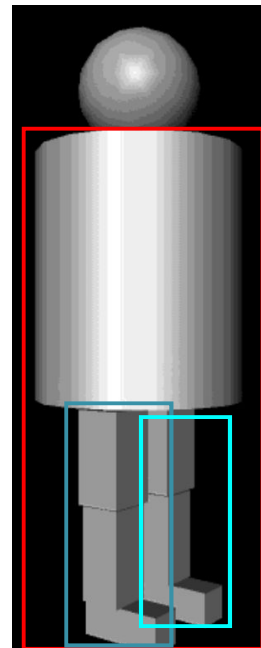
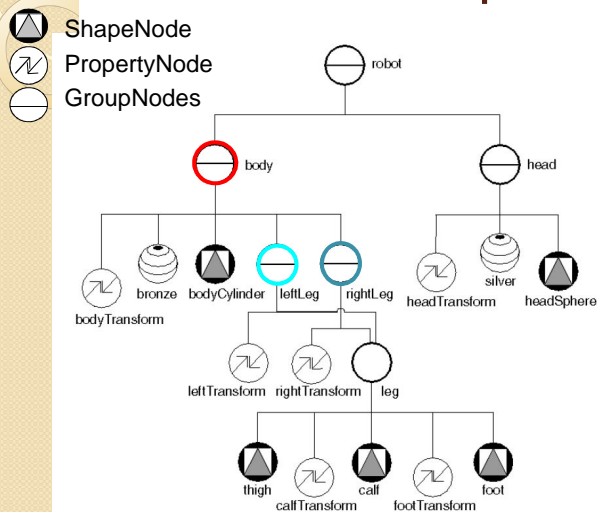
- Le « Scene Graph » : modèle hiérarchique (1988 PHIGS, 1992 Inventor, 1991 VRML.1... X3D)
- Exemples : VRML, Inventor, Java3D, OpenSG, SVG, Three.js (WebGL), ...
- Les moteurs « engine » (parcours du scene graph) pour le Rendu, l'Animation, physique...
- Logiciels : Ogre3D, Cortona, Unity, CorelDraw...



Modélisation-géométrique - O.Stab

9

Scene Graph

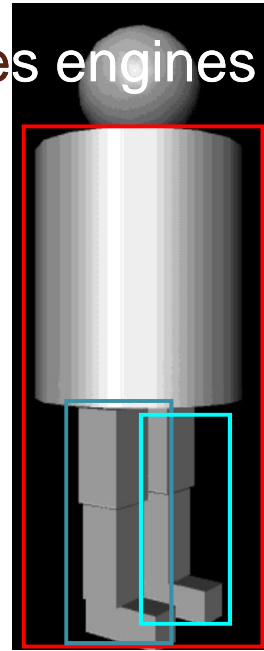
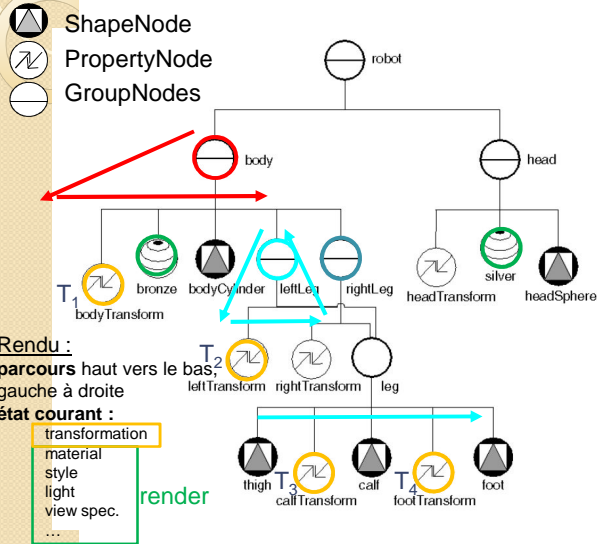


Exemple Inventor

Modélisation-géométrique - O.Stab

10

SG : le parcours des engines

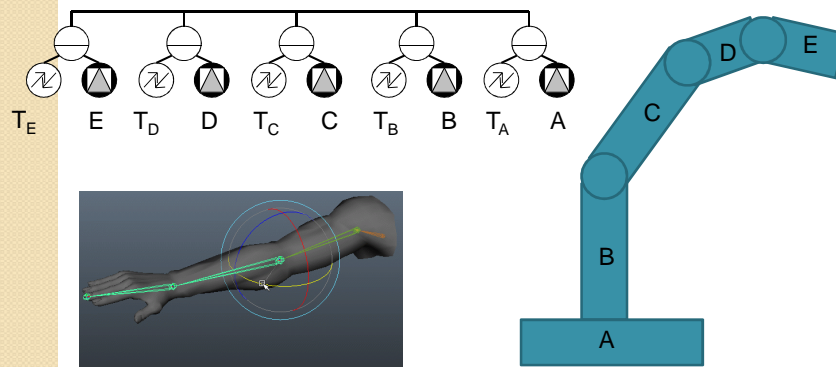


Modélisation-géométrique - O.Stab

11

Positionnement et orientation absolu

Corriger l'arbre pour pouvoir animer le bras

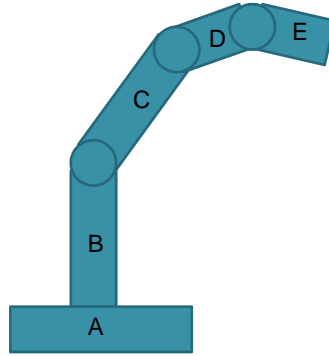
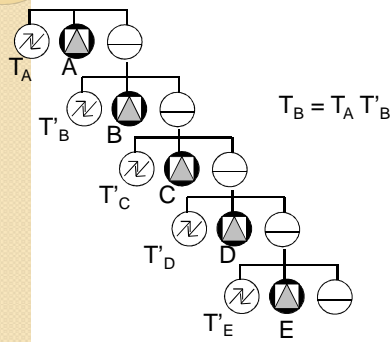


Modélisation-géométrique - O.Stab

13

Positionnement et orientation relatifs

Arbre pour pouvoir animer le bras

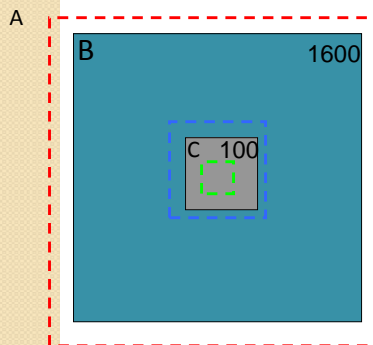


Calculez les transformations T'_x à partir des T_x

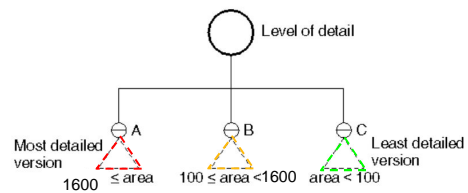
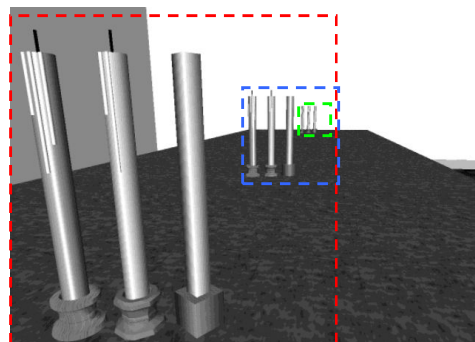
Montrez que : $T'_E = (T_D)^{-1} T_E$

Switch Node : Level Of Detail (LOD)

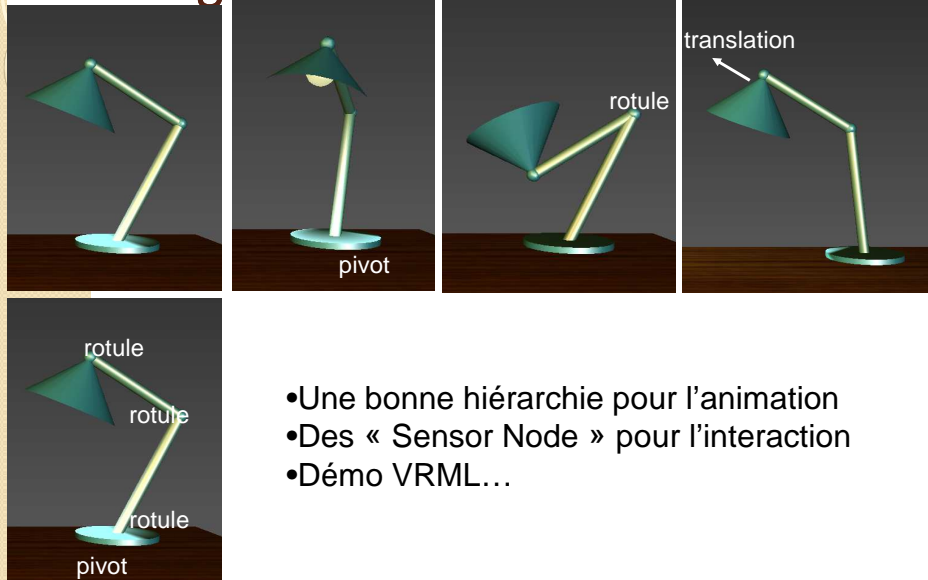
Mécanisme automatique de rendu multi-résolution



ScreenArea(1600,100)



Scène graph pour l'animation



Modélisation-géométrie - O.Stab

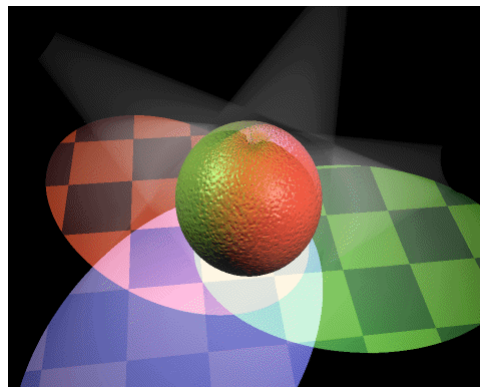
16



2. Le rendu (éclairage - shading)

ou la modélisation empirique de notre « vision »

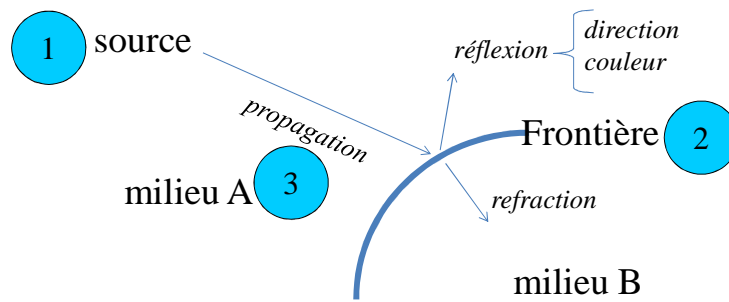
1. Modèle d'éclairage (source & réflexions)
2. Les couleurs
3. Les textures des surfaces
4. Insuffisances du modèle géométrique



Synthèse d'images

19

Propagation de la lumière



La lumière est l'ensemble des ondes électromagnétiques visibles par l'œil humain

$\lambda = 800 \text{ nm}$



$\lambda = 380 \text{ nm}$

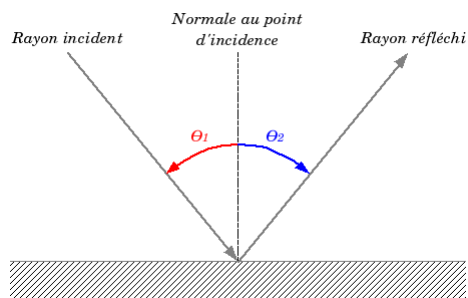
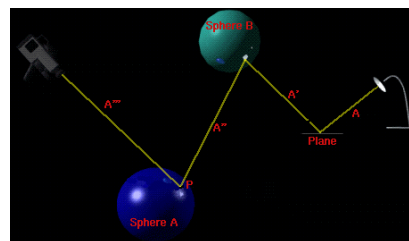
Synthèse d'images

21

Réflexions

Loi de Snell-Descartes

$$\theta_2 = -\theta_1$$

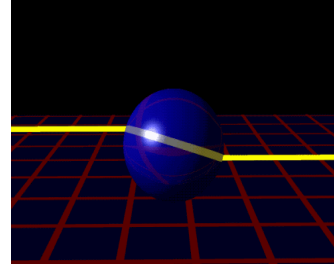
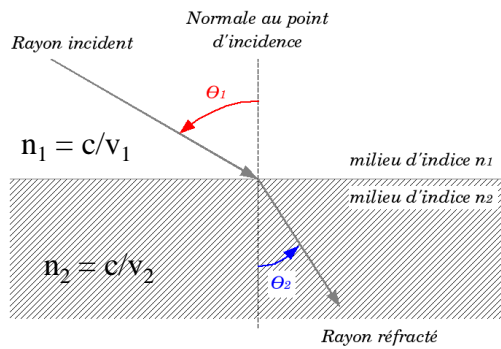


22

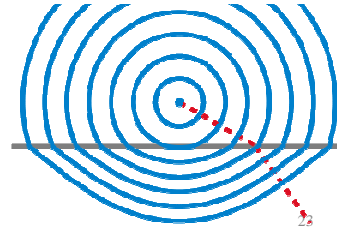
Transparence, réfraction

Loi de Snell-Descartes

$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$



n dépend de λ , T, structure : matériaux biréfringents



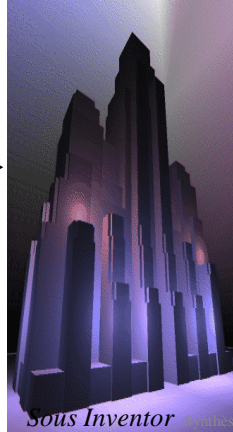
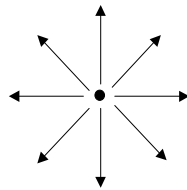
2.1.1 Source de lumière



Sources ponctuelles

Isotrope

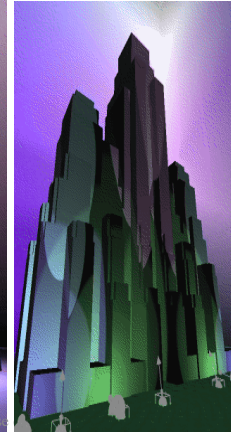
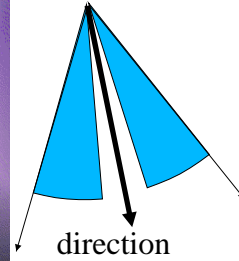
Atténuation en fonction de la distance



Sous Inventor

Directionnelle

Atténuation en fonction de la direction

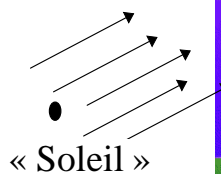


26

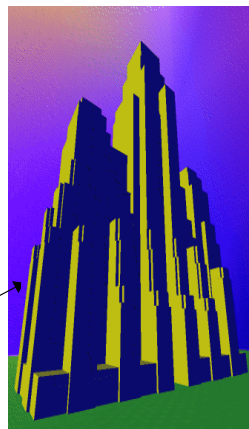
Autres sources

Source directionnelle

simplification



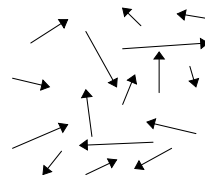
« Soleil »



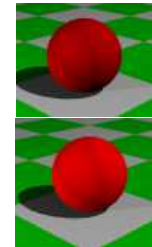
Eclairage ambiant

(Isotrope et Uniforme)

simplification



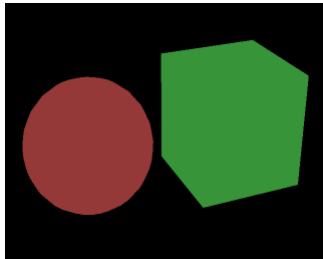
« Radiosité »



Synthèse d'images

27

Intensité (réflexion) ambiante



$$I_i = K_{ai} I_a$$

K_{ai} : coefficient de réflexion de la lumière ambiante

Métal :
 Verre :
 Nuage :
 etc...



L'objet n'émet pas de lumière. On ne peut pas la négliger...

Intensité (réflexion) ambiante



$$I_i = K_{ai} I_a$$

K_{ai} : coefficient de réflexion de la lumière ambiante

Nuage : 0.7, 1.0
 Verre : 0.0, 0.1
 Métal : 0.2
 etc...



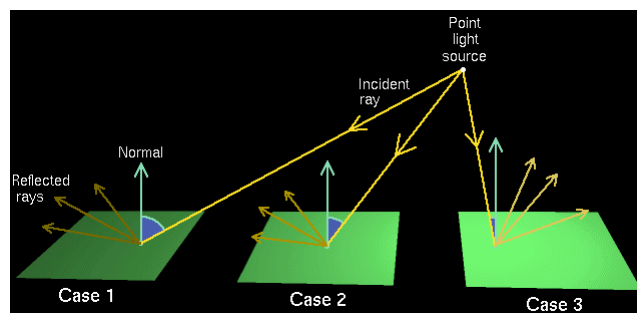
On ne peut pas la négliger...



2.1.2 Réflexions/surfaces (micro)



Réflexion diffuse



- La **réflexion diffuse** dans toutes les directions (indépendant de l'observateur)
- L'intensité dépend de l'angle entre la source et la normale à la surface
- Réaliste pour les matériaux mats (« rugosité fine » de surface)

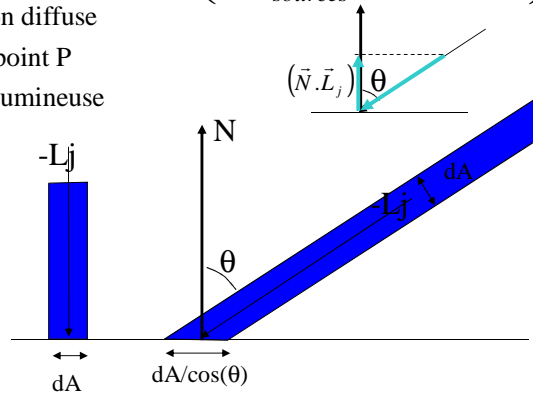
Réflexion diffuse : Loi de Lambert

$$I_d = \max \left(K_d \sum_{sources} \langle \vec{N}, \vec{L}_j \rangle I_j, 0 \right)$$

- K_d : coefficient de réflexion diffuse
- N : normale à la surface au point P
- L_j : direction de la source lumineuse
- I_j : intensité de la source

Exemples K_d :

Métal :
 Verre :
 Nuage :
 Bois :
 etc...



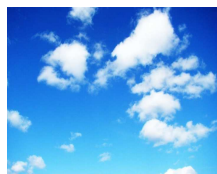
Synthèse d'images

35

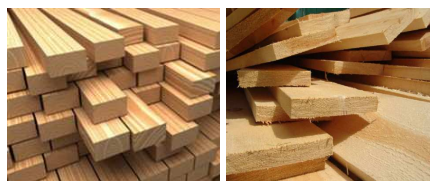
Réflexion diffuse : K_d



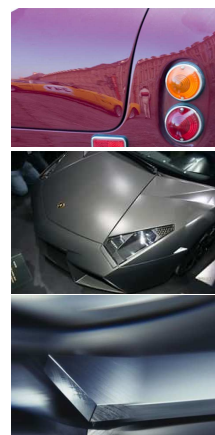
Verre : 0, 0.1



Nuage : 0



Bois : 0.3



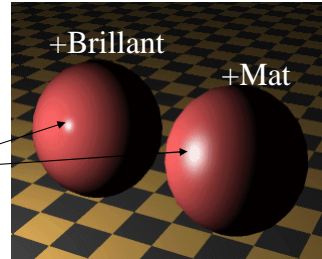
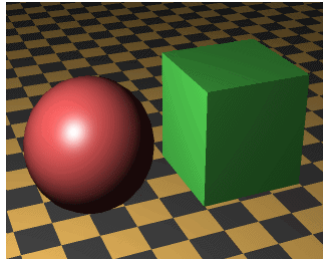
Métal : 0.7, 1.0...

36

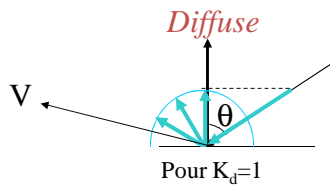
Réflexion spéculaire

spéculaire : relatif au miroir...

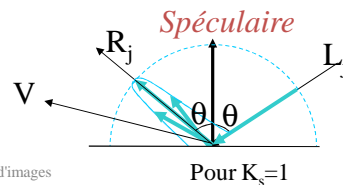
ici le reflet de la source de lumière



Pour un observateur dans la direction V



Synthèse d'images



39

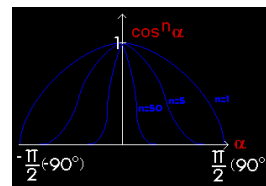
Intensité spéculaire

Phong

$$I_s = \sum_{sources} K_s \langle R_j, V \rangle^n I_j$$

n dépend du type de matériau

1 : mat, 200 : brillant



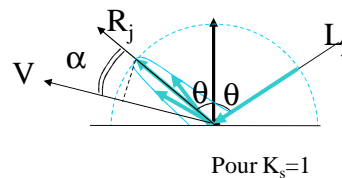
Exemples K_s et n :

Verre :

Métal : ...

Bois : ...

...etc...



$$R_j = L_j - 2 \langle N, L_j \rangle N$$

Synthèse d'images

40

Réflexion spéculaire: n et K_s



Verre : $K_s = 1$ $n=1.5$

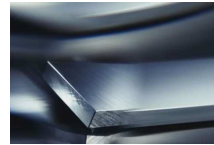


Métal : $K_s = 0.98$ $n=1.75$...

Nuage : $K_s = 0$

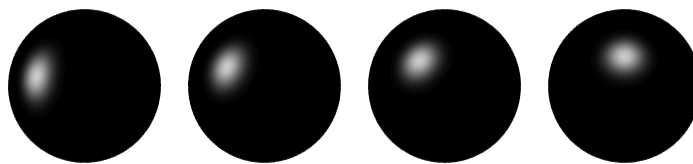


Bois : $K_s = 0.2$ $n=1.5$

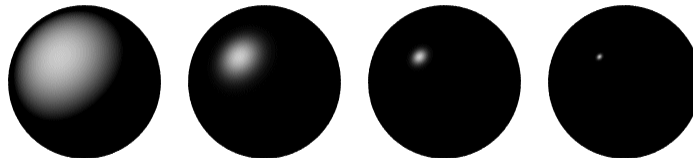


41

Modèle de Phong



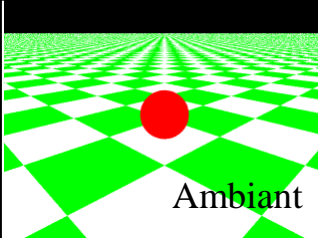
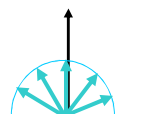
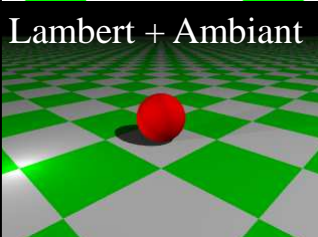
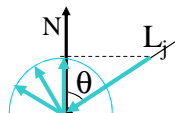
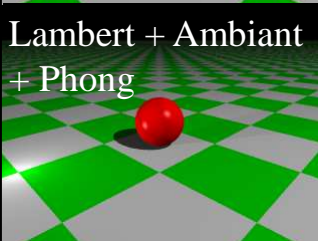
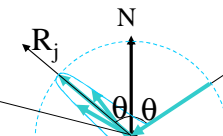
Si on déplace une source lumineuse




Si on change la brillance (n)

Synthèse d'images


44


 <p>Ambiant</p>	<p>Diffuse la lumière ambiante « Radiosité »</p> $I_i = K_{ai} I_a$	<p>Réflexion</p>  <p>Pour $K_{ai}=1$</p>
 <p>Lambert + Ambient</p>	<p>Réflexion diffuse</p> $I_d = \max \left(K_d \sum_{sources} \langle N, L_j \rangle I_j, 0 \right)$ <p>N normale à la surface Lj rayon de la lumière j</p>	<p>Matériau Mat</p>  <p>Pour $K_d=1$</p>
 <p>Lambert + Ambient + Phong</p>	<p>Réflexion spéculaire</p> $I_s = \sum_{sources} K_s \langle R_j, V \rangle^n I_j$ $R_j = L_j - 2 \langle N, L_j \rangle N$	<p>Brillant/Lisse</p>  <p>Pour $K_s=1$</p>




ECOLE DES MINES DE PARIS
Robotique

Matériaux







Verre : $K_a = 0, K_d = 0$
 $K_s = 1 \quad n=400\dots$



Métal : $K_a = 0.2, K_d = 0.7$
 $K_s = 0.75 \quad n=80 \dots$



Nuage : $K_a = 0.7, K_d = 0$
 $K_s = 0\dots$

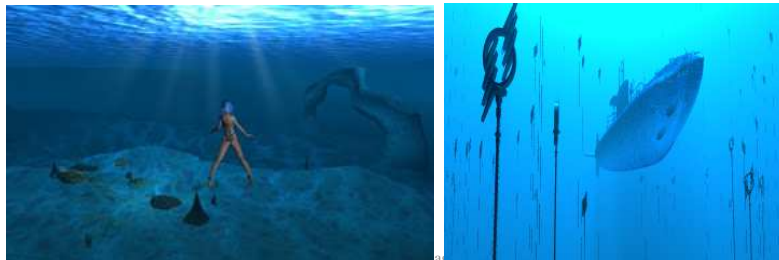


Bois : $K_a = 0. , K_d = 0.3$
 $K_s = 0.2 \quad n=10\dots$

47

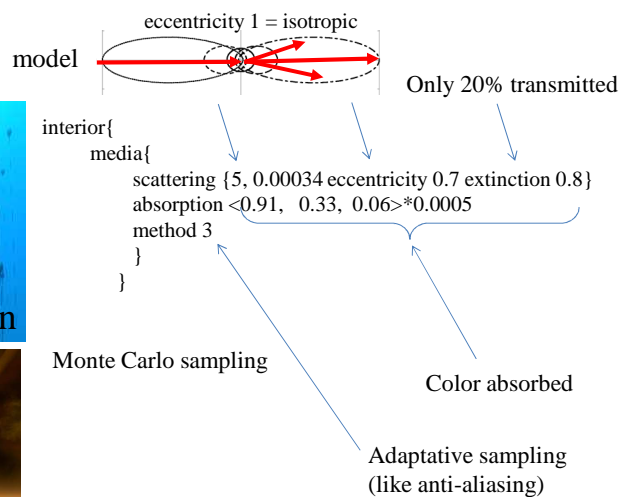
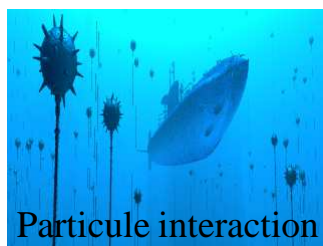


2.1.3 Milieu traversé : atténuation

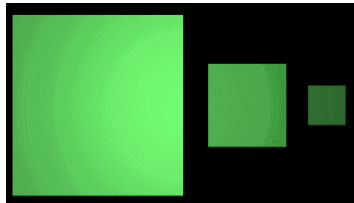


48

Effets du milieu



Atténuation des sources lumineuses



$$I_{att} = f_{att} I_d$$

~~$$f_{att} = \frac{1}{D_l^2}$$~~

$$f_{att} = \min\left(\frac{1}{c_1 + c_2 D_l + c_3 D_l^2}, 1\right)$$

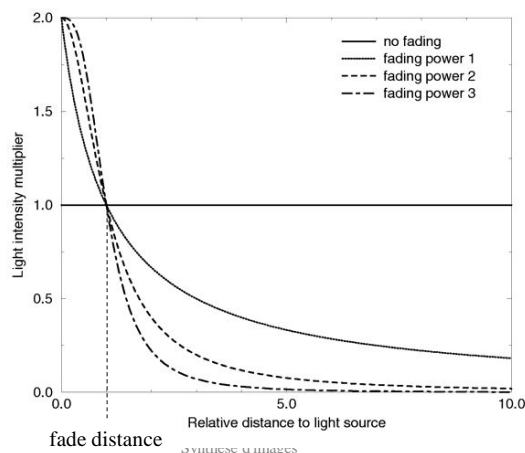
Effet du milieu traversé

D_l = distance à la source de lumière

Atténuation (light fading)

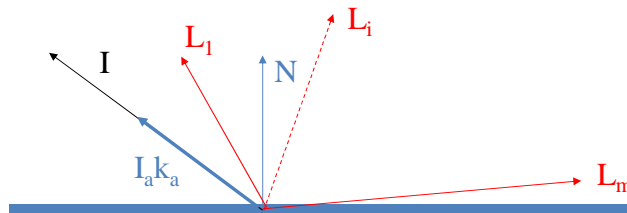


$$attenuation = \frac{2}{1 + \left(\frac{d}{fade \times distance}\right)^{fade \times power}}$$



Sources multiples

$$I = \boxed{I_a K_a} + \sum_{i=1}^m f_{att_i} I_{pi} \left[\boxed{K_d (\vec{N} \cdot \vec{L}_i)} + \boxed{K_s (\vec{R}_i \cdot \vec{V})^n} \right]$$



Synthèse d'images

53



2.2 La couleur



Synth

Arithmétique sur les couleurs

Couleurs RGB = [%R,%G,%B]

blanc=[1,1,1]

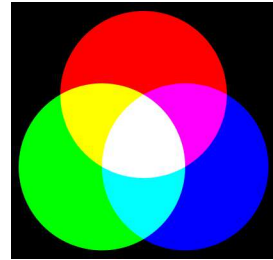
noir=[0,0,0]

(système additif)

Réflexion parfaite : $C_{\text{refl}} = I_{\text{lum}} * O_{\text{surf}}$

blanc = élément neutre

noir = élément absorbant

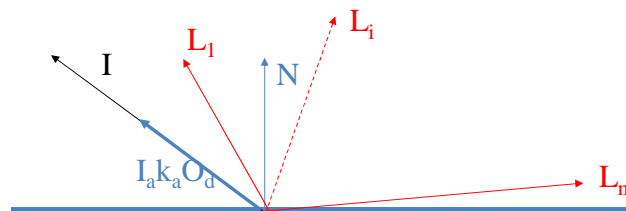


Complémentarité : $[0,1,0] * [1,0,1] = [0,0,0]$

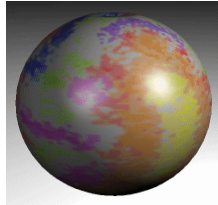
Une surface qui absorbe le vert $[1,0,1]$ éclairée par une source verte $[0,1,0]$ ne renvoi rien \Leftrightarrow la surface apparaît noire

Intensité avec les couleurs

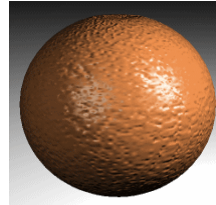
$$I_R = I_{aR} K_a O_{dR} + \sum_{i=1}^m f_{\text{att}_i} I_{piR} \left[K_d O_{dR} (\vec{N} \cdot \vec{L}_i) + K_s (\vec{R}_i \cdot \vec{V})^n \right]$$



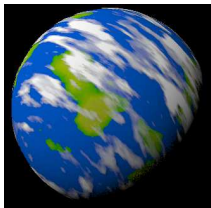
2.3 Textures



Color mapping



Bump mapping



Motif unique



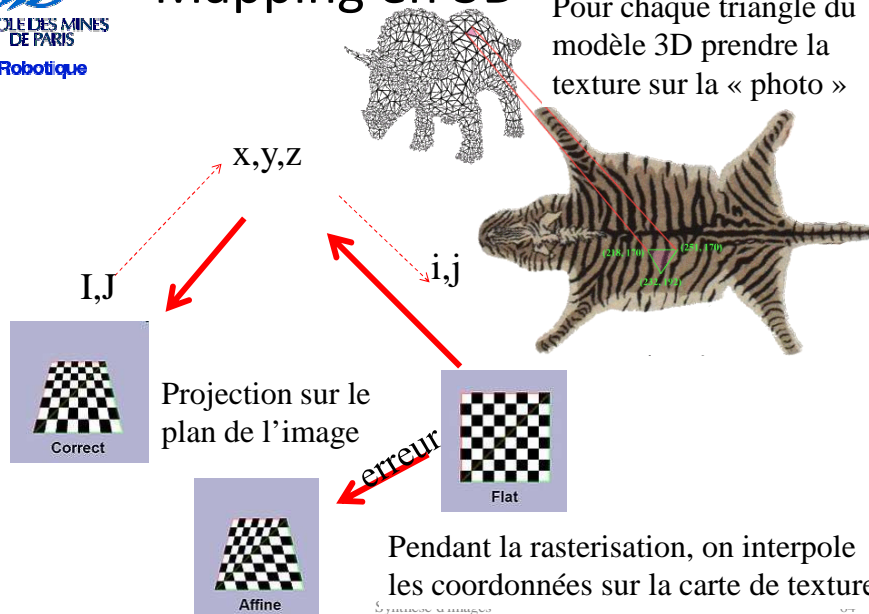
Motif répété

Synthèse d'images

62

Mapping en 3D

Pour chaque triangle du modèle 3D prendre la texture sur la « photo »



Synthèse d'images

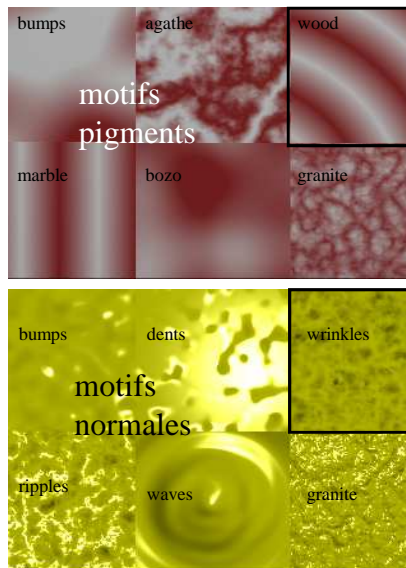
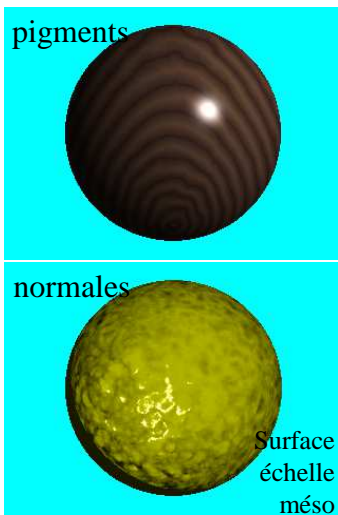
63

Texture mapping



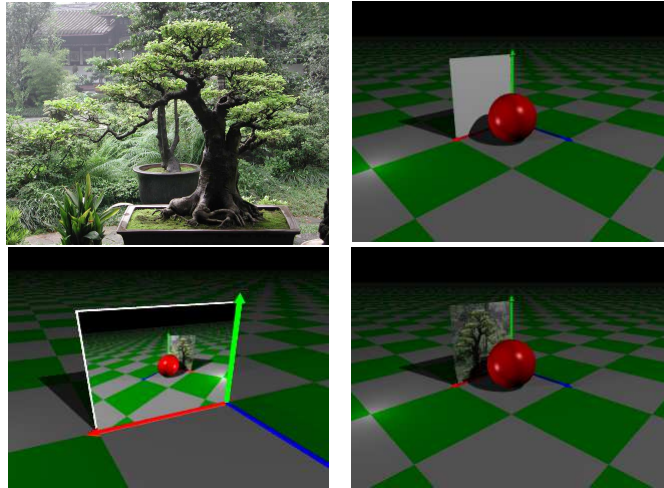
Synthèse d'images

Mapping Pigment/Normal



Exemples de texture

Image
mapping

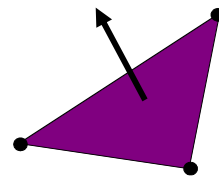
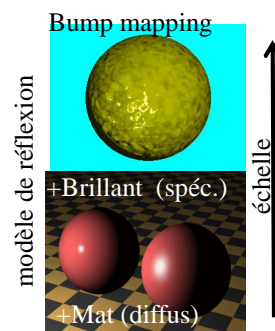


Synthèse d'images

70

2.4. Insuffisances du modèle « géométrique »

- Géométrie de la surface :
 - Le « Bump Mapping » (meso)
 - réflexion diffuse/spéculaire (micro)
- Géométrie dégradée (Jeux vidéo...)
 - « shading » : flat, Gouraud, Phong (macro)

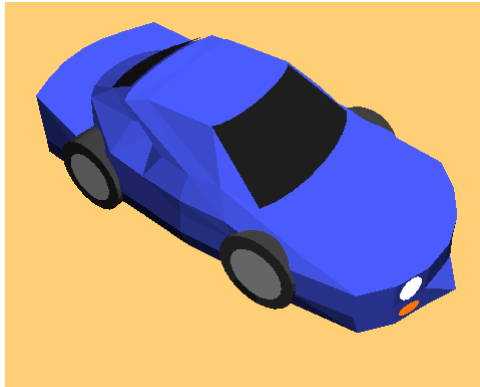


Synthèse d'images

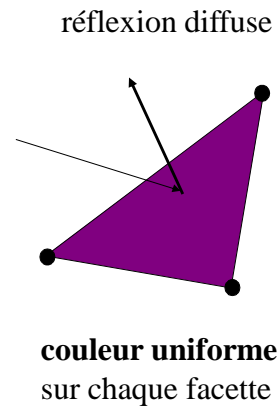
73

Affichage : « ombrage » plat (Flat shading)

- Intensité uniforme



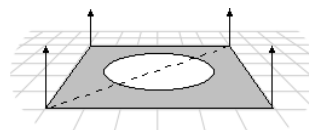
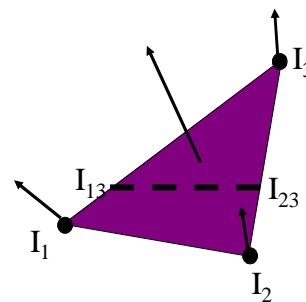
Synthèse d'images



74

« Ombrage » de Gouraud

- Principe
 - interpolation des intensités
- Algorithme
 - calcul des normales des facettes
 - calcul des normales aux sommets
 - intensité lumineuse des sommets
 - interpolation des intensités
- Inconvénients
 - polygones convexes
 - pas de surfaces brillantes
 - dépend de l'orientation



Synthèse d'images

75

Exemple : Flat vs Gouraud



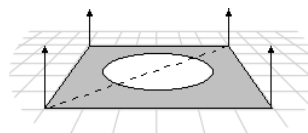
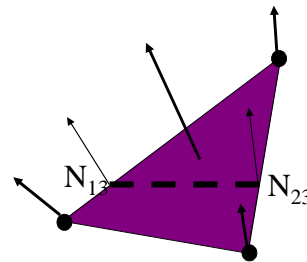
Flat shading



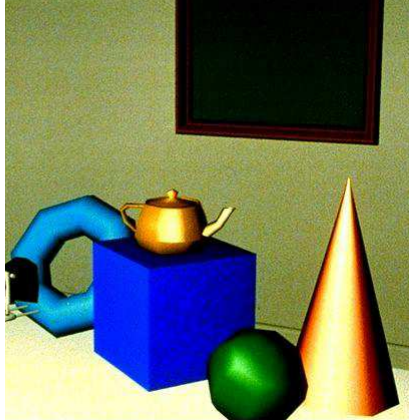
Gouraud (interpolated) shading

« Ombrage » de Phong

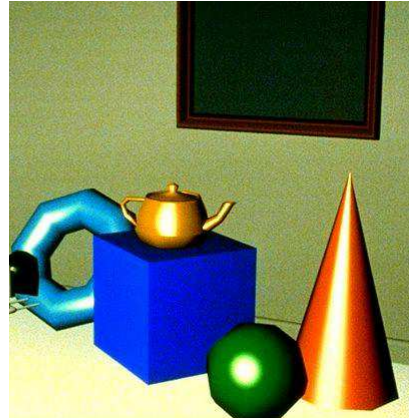
- Principe
 - Interpolation des normales
- Algorithme
 - calculer les normales aux polygones
 - calculer les *normales* aux sommets
 - calculer les *normales* de tous les points
 - calculer l'intensité lumineuse en tous les points
- **Avantage** : permet de « capturer » des reflets (dans le cas d'un modèle spéculaire)



Exemple : Gouraud vs Phong



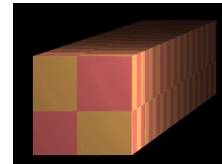
Gouraud



Phong

Synthèse d'images

78

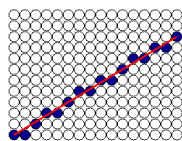
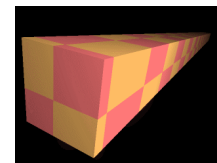


3. L'image

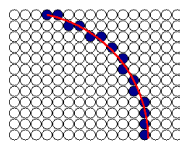
Œil/Caméra

Parties cachées / Ombres portées

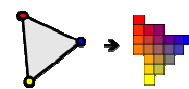
Pixelisation / Aliasing



Droite



Cercle



Position/Orientation dans la scène (coordonnées homogènes)

$$4D \rightarrow 3D$$

$$P = (x, y, z, w) \rightarrow P_c = \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)$$

$$R_z(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$M_H = \begin{bmatrix} R_{3,3} & T \\ O_3^t & 1 \end{bmatrix}$$

$$M_T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Multiplication

Synthèse d'images

81

3.1 Projections dans le plan image (viewing transformation)

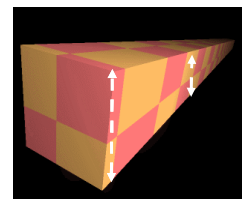
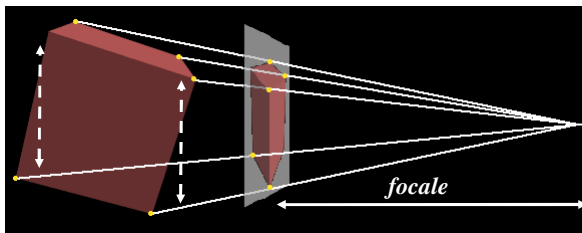
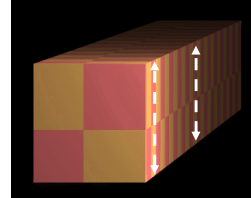
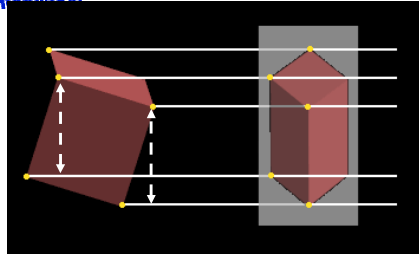


<http://www.indiana.edu/~kglowack/athens/>

Synthèse d'images

82

Parallèle ou perspective ?

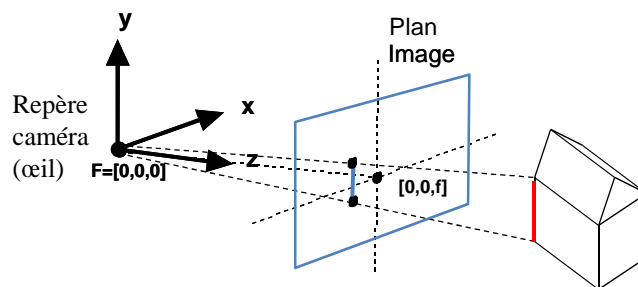


Synthèse d'images

83

Exemple simple

- Caméra dans l'axe des Z
- point focal à l'origine
- plan image sur XY à une distance f



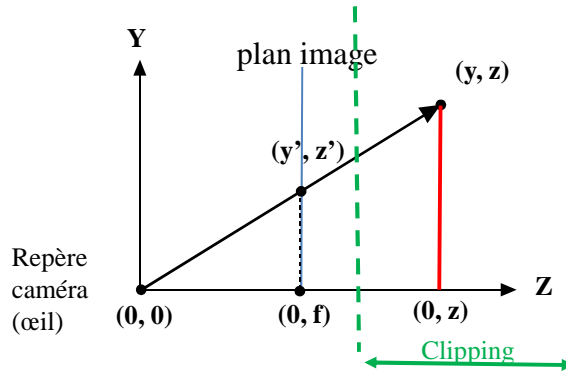
Synthèse d'images

84

Calcul du point de projection

$y', z' ?$

$$\begin{cases} z' = f \\ y' = (f/z) * y \end{cases}$$



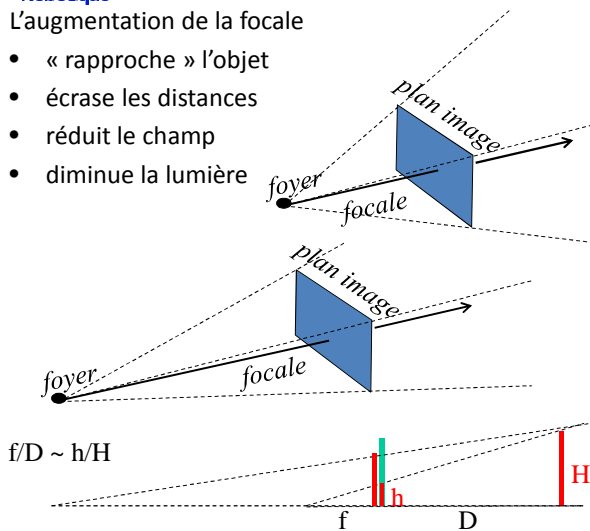
Comment agrandir une image ?

En 3D le point $[x,y,z]$ se projette sur $[(f/z)x, (f/z)y, f] = [x/(z/f), y/(z/f), z/(z/f)]$

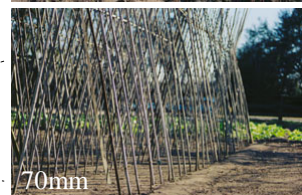
Le zoom, la focale

L'augmentation de la focale

- « rapproche » l'objet
- écrase les distances
- réduit le champ
- diminue la lumière



$$f/D \sim h/H$$

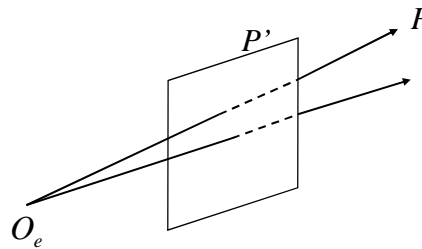


source wikipédia

Projection perspective

en coordonnées homogènes :

$$M_{per} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix}$$

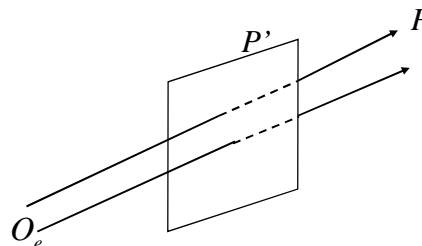


$M_{per}^*(x,y,z,1) = (x,y,z,z/f)$ qui donne en 3D : $(x^*(f/z), y^*(f/z), f)$

Projection orthographique

en coordonnées homogènes :

$$M_{per} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$M_{per}^*(x,y,z,1) = (x,y,f,1)$ qui donne en 3D : (x, y, f)

Paramètres d'une caméra

sPHIGS

VRP : View Reference Point
 VPN : View Plane Normal (z)
 VUP : View Up Vector (y)
 PRP : Projection Reference Point
 Window
 Projection Type

PovRay

```

Camera {
  [perspective| orthographic
  fisheye|panoramic...]
  location <point>
  sky
  <vecteur>
  ...
  look_at <vecteur>
}
  
```

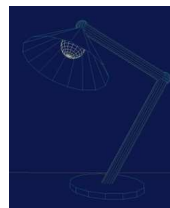
OpenInventor ~ OpenGL

```

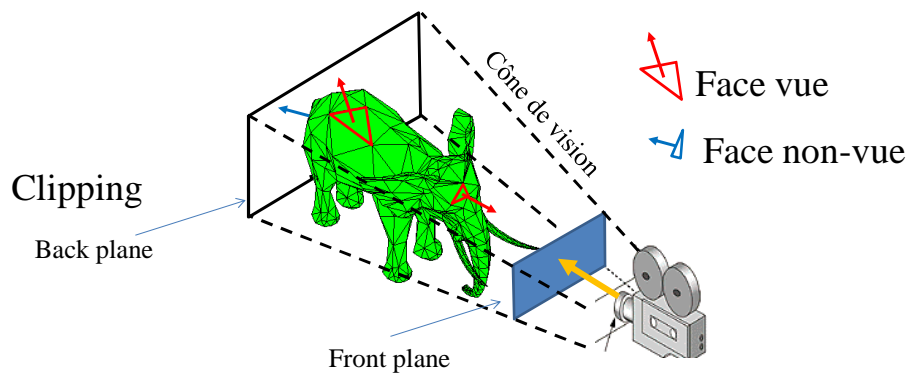
PerspectiveCamera {
  viewportMapping ADJUST_CAMERA
  position 0 0 1
  orientation 0 0 1 0
  aspectRatio 1
  nearDistance 1
  farDistance 10
  focalDistance 5
  heightAngle 0.785398
}
OrthographicCamera {
  viewportMapping ADJUST_CAMERA
  position 0 0 1
  orientation 0 0 1 0
  aspectRatio 1
  nearDistance 1
  farDistance 10
  focalDistance 5
  height 2
}
  
```

3.2 Élimination des parties cachées

- Élimination de lignes
- Élimination de faces
 - algorithmes à priorité
 - Z-buffer (pixel)
 - Peintre (Newell & Sancha)
 - algorithmes à balayage de lignes
 - Watkins, Z-buffer ligne
 - algorithmes travaillant par subdivision
- (*Lancé de rayon (pixel)*)



Le Clipping et le Back face culling

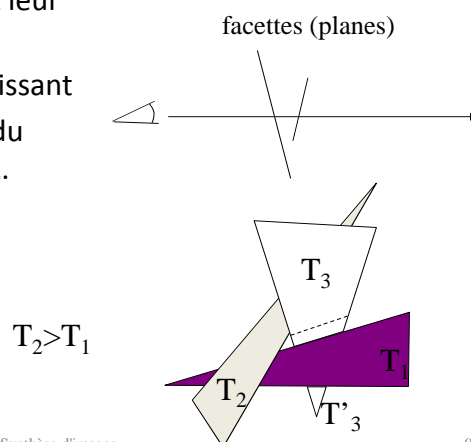


Dans quel cas est-ce suffisant ?

Algorithmes à liste de priorité (Algorithme du peintre)

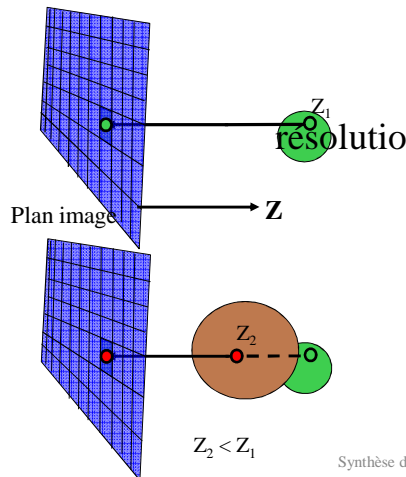
- Liste ordonnée de facettes

- trier les facettes suivant leur profondeur : Z-max
- afficher en Z-max décroissant
- ambiguïtés : problème du recouvrement cyclique...

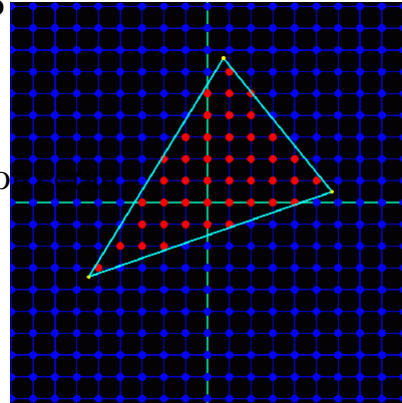


Z-buffer (principe)

Distance $\min(Z_i)$ par pixel?



Pendant la « pixelisation »



A la résolution écran

Synthèse d'images

99

Z-buffer (algorithme*)

Stocker l'intensité des pixels mais aussi leur profondeur

Idée

Si Z du pixel $<$ Z mémorisée on mémorise

Avantage : algorithme câblé (16 à 32 bits)

Inconvénient

taille mémoire (codage)

problème d'aliasing



A simple three dimensional scene



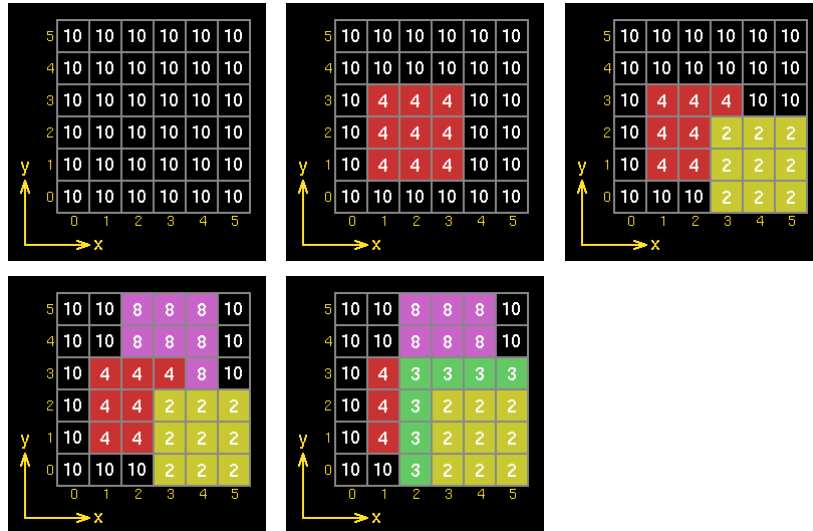
wikipedia

Z-buffer representation¹⁰⁰

*Edwin Catmull 1974

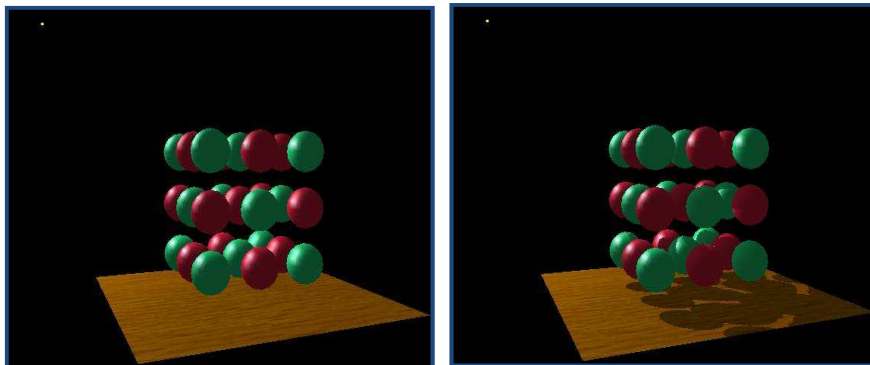
Synthèse d'images

Z-buffer (exemple)



101

Ombres portées



Synthèse d'images

105

Ombres portées

- Principe
 - réutiliser l'algorithme d'élimination des faces cachées depuis l'origine de la source lumineuse
 - créer des polygones d'ombre qui seront traités comme les autres facettes lors du second passage
- Coût
 - rendu en deux passes au lieu d'une

Texture d'ombre

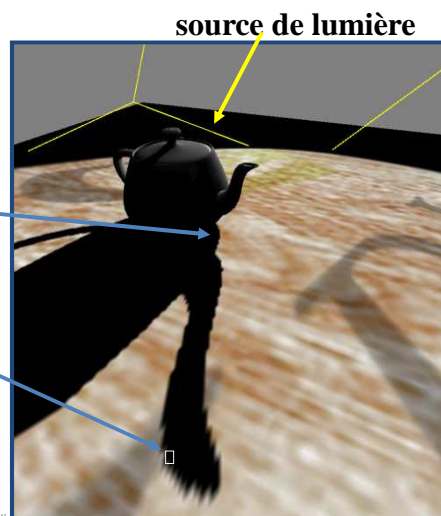
problème d' « aliasing » avec le Z-buffer

Contours très précis
près de la source

Contour très
grossier loin de la
source

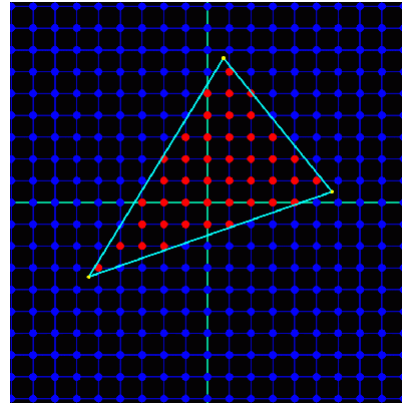
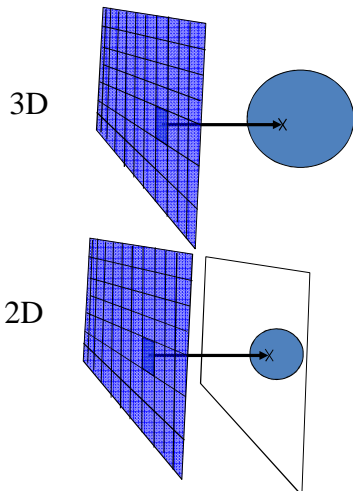
Un seul pixel dans le
Zbuffer de la source

Solution : Irregular Z-buffer



3.3. Rasterization

- Couleur du pixel ?

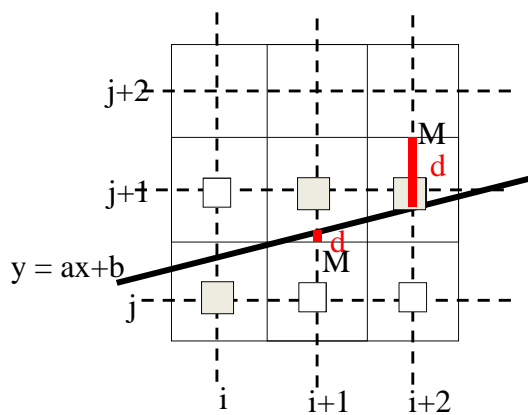


Synthèse d'images

108

Tracé d'un segment

(Mid-Point Algorithm)



Cas : $0 < a < 1$

```

WritePixel(i0,j0)
d=a*(i0+1)+b-(j0+1/2)
For(i=i0+1;i<=i1;i++){
  Si (d<0) d=d+a
  Sinon d=d+a-1
  j=j+1
  WritePixel(i,j)
}
  
```

Synthèse d'images

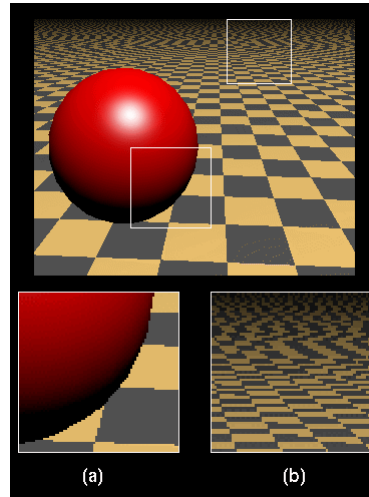
110

Effets d' « aliasing »

(a) frontières d'objet
(b) textures

Effets:

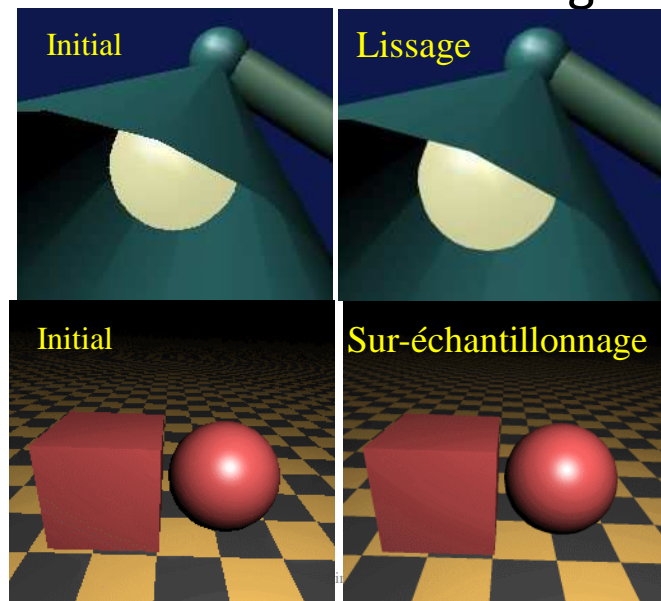
- effets d'escalier sur les images
- petits objets entièrement ou partiellement masqués
- moirés sur les textures



Synthèse d'images

112

Méthodes « Anti-aliasing »



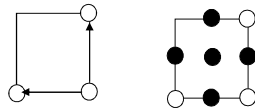
114

Antialiasing Adaptatif

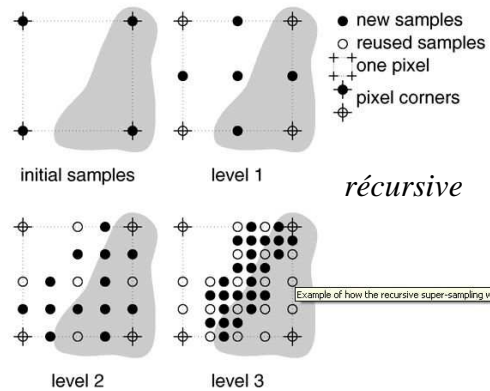
(sous PovRay)

2 méthodes adaptatives

- $\text{diff} = \text{abs}(R_1 - R_2) + \text{abs}(G_1 - G_2) + \text{abs}(B_1 - B_2)$
- seuil : [0:3], 0.3
- profondeur : 3

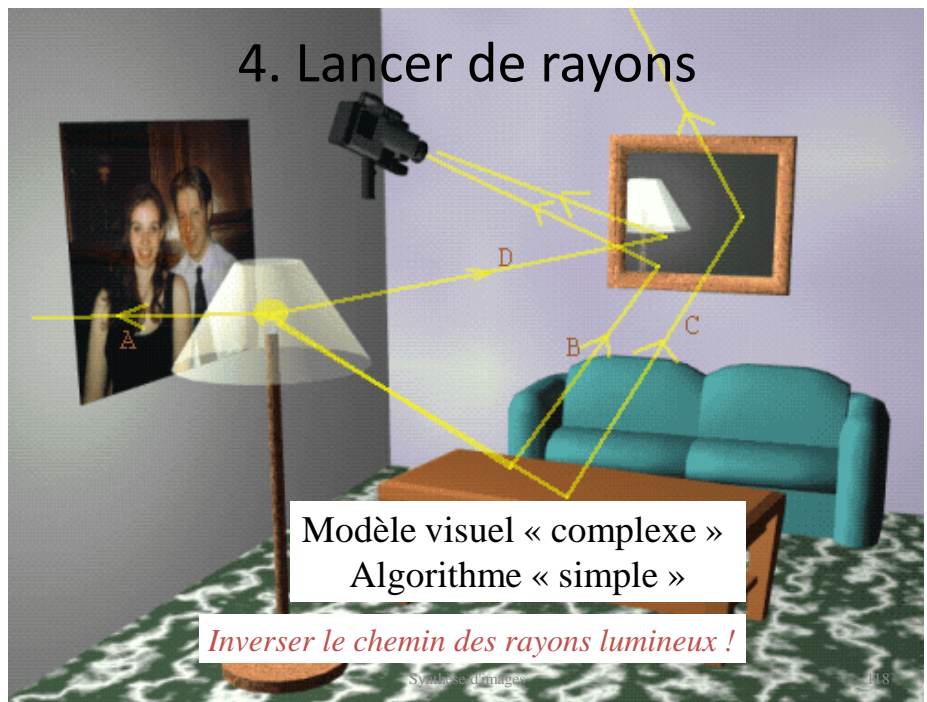


non-réursive



réursive

4. Lancer de rayons



Modèle visuel « complexe »
 Algorithme « simple »

Inverser le chemin des rayons lumineux !

Synthèse d'images

118

Le lancer de rayon

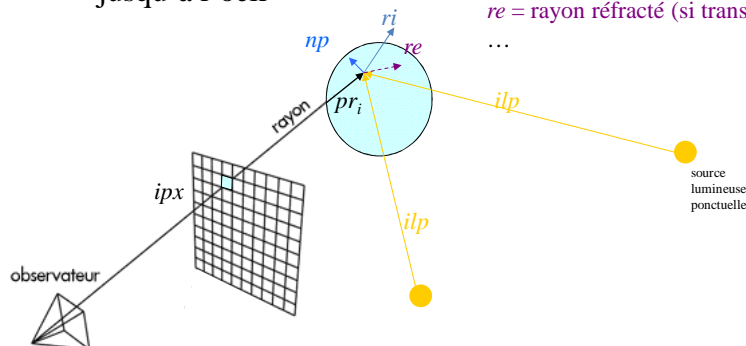
- 1968 : Algorithme proposé par Appel
- 1980 : David Kirk Buck (futur PovRay)
- 1991 : PovTeam
- ...
- 2012 : Nvidia Kepler real-time...

- Idée
 - facette = réflecteur parfait
 - inverser la circulation des rayons lumineux : de l'écran vers les sources de lumière

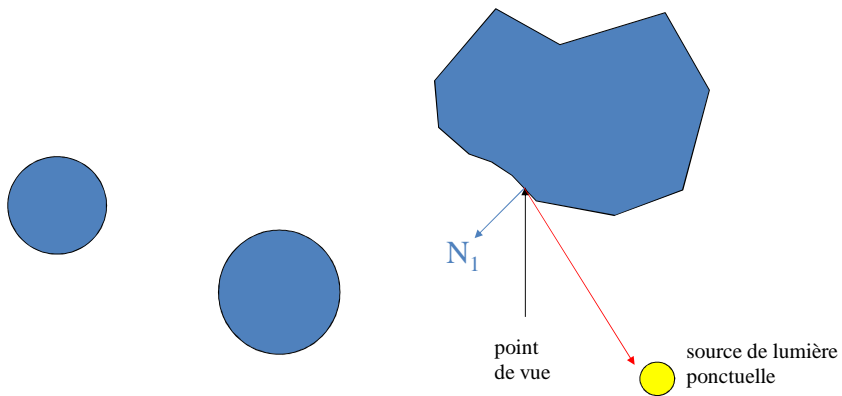
Lancé de rayon (Ray tracing)

Principe : modéliser le trajet
des rayons de la lumière
jusqu'à l'oeil

ipx = intensité lumineuse du pixel
 r = lancé d'un rayon partant de ipx
 pr_i = intersection de r avec la géométrie
 np = normale en pr_i
 ilp = éclairage de la surface en pr_i
 ri = rayon réfléchi en pr_i (si miroir)
 re = rayon réfracté (si transparence)
 ...



Principe : lancé de rayon

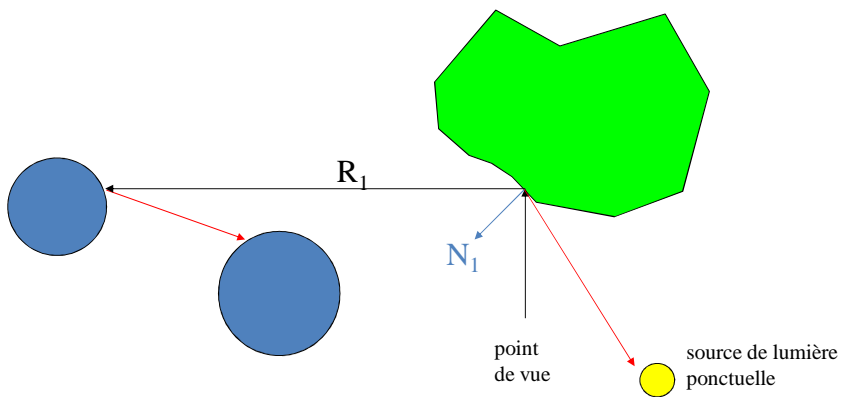


from CG principles & practice Foley, Van Dam ...

Synthèse d'images

122

Principe : lancé de rayon

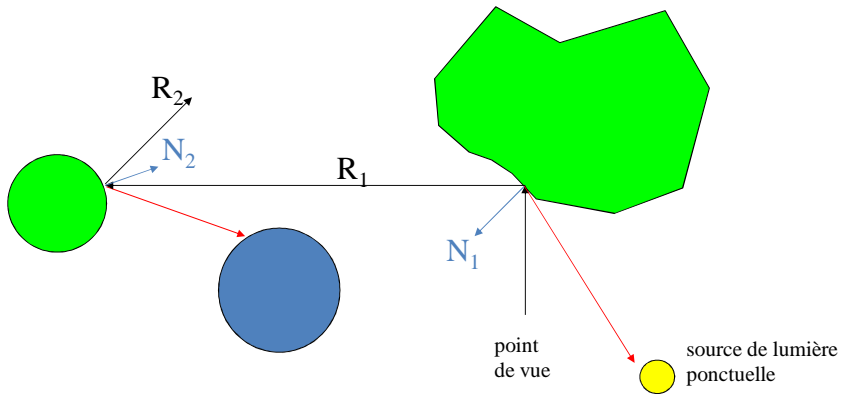


from CG principles & practice Foley, Van Dam ...

Synthèse d'images

123

Principe : lancé de rayon

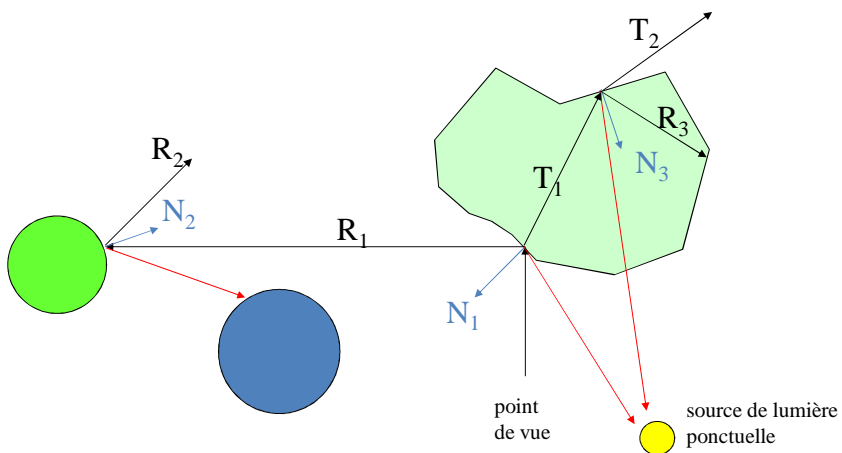


from CG principles & practice Foley, Van Dam ...

Synthèse d'images

124

Principe : lancé de rayon



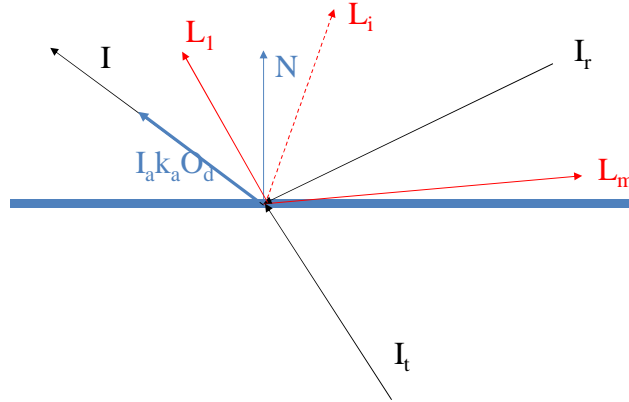
from CG principles & practice Foley, Van Dam ...

Synthèse d'images

125

Définition récursive

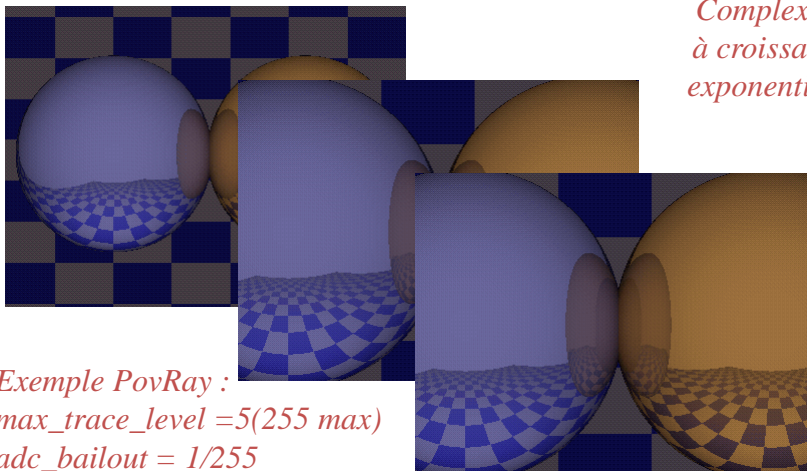
$$I_\lambda = I_{a\lambda} K_a O_{d\lambda} + \sum_{i=1}^m S_i f_{att_i} I_{p\lambda i} \left[K_{di} O_{d\lambda} (\vec{N} \cdot \vec{L}_i) + K_s (\vec{R}_i \cdot \vec{V})^n \right] + K_r I_{r\lambda} + K_t I_{t\lambda}$$



Synthèse d'images

Jusqu'où ?

Complexité à croissance exponentielle



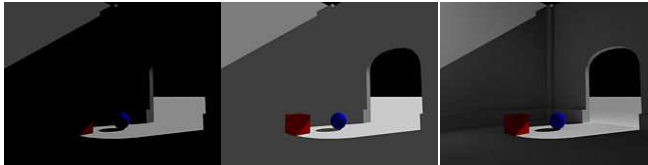
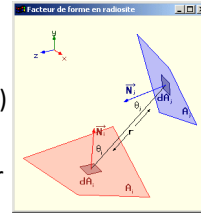
*Exemple PovRay :
max_trace_level = 5 (255 max)
adc_bailout = 1/255*

Synthèse d'images

Intensité Ambiante & radiosité

Le modèle de l'intensité ambiante n'est PAS cohérent car
sans réflexion : $I_{ai} = K_{ai} I_a$

- **Modèle de Radiosité**
 - Chaque facette émet sa propre couleur qui est réfléchiée par les autres facettes...jusqu'à l'équilibre.
 - Système linéaire $n \times n$ facettes à résoudre (Hors ligne)
 - Résultat est indépendant de l'observateur
 - L'affichage peut être réalisé ensuite avec un Z-buffer ou un lancé de rayon



Exemple PovRay :
Recursion_limit = 3(20)
adc_bailout = 0.01

Synthèse d'images

<http://wiki.povray.org>

128

Ray-tracing

AVANTAGES

- Images réalistes
- Effets
 - milieux participants
 - mirages
 - bleu atmosphérique
 - effets photographiques
 - profondeur de champ
 - déformations de l'objectif

INCONVENIANTS

- Le TEMPS de CALCUL*
- Mais accélération
 - lancer de rayon distribué (parallélisme)
 - sous-échantillonnage
 - partitionnement
 - Boîtes englobantes
- Mais aussi...
 - l'aliasing (crénelage)

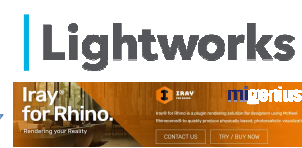
→ [Back to rasterization](#)

*NVIDIA Kepler real-time raytracing demo at GTC 2012 - The Verge

Logiciels d'infographie

Robotique

- [Autodesk](#)
 - 3DS Max, Maya (*rachat*), Softimage (*rachat 2008*)
- [DAZ3D](#) : Brice 3D
- Logiciels libres :
 - Persistence of Vision ([PovRay](#))
 - [Blender](#)
- **NVIDIA** :
 - ~~MentalRay~~ (*Hulk, Matrix, Le jour d'après...*)
 - 3D Gelato (gratuit)



Synthèse d'images

133

Éléments de bibliographie

Robotique

- Computer Graphics, principles and practice
 - Foley, Van Dam, Feiner, Hughes
 - Addison Wesley
- Advanced Animation and Rendering Techniques,
 - Alan & Mark Watt
 - Addison Wesley
- The Inventor Mentor
 - J.Werneck, OpenInventor Arch Group
 - Addison-Wesley, 1994
- Informatique graphique
 - B.Péroche, D.Ghazanfarpour, D.Michelucci, M.Roelens
 - Hermès, 1998
- Revues Computer Graphics (IEEE)
- WEB :
 - <http://raphaello.univ-fcomte.fr>
 - <http://www-evasion.imag.fr/~Francois.Faure/>
 - http://en.wikipedia.org/wiki/Portal:Computer_graphics
 - ...

Synthèse d'images

135

Exemple (CSTB*)



*Centre Scientifique et Technique du Batiment

d'images

136

Un bureau au CAOR Mines-Paristech



*Fawzi Nashashibi

Synthèse d'images

137



PovRay output messages

Robotique Quality < 8 no reflexion, nor refraction

```
Render Options
Quality: 9
Bounding boxes...On Bounding threshold: 3
Antialiasing...On (Method 1, Threshold 0.300, Depth 3, Jitter 1.00, Gamma 2.50)

Render Statistics
Image Resolution 1280 x 720
Pixels: 980319 Samples: 157221 SmpIs/px: 0.16
Rays: 1646160 Saved: 0 Max Level: 2/5

Ray->Shape Intersection Tests Succeeded Percentage
Box 375189 306769 81.76
Cone/Cylinder 220962 96310 43.59
CSG Intersection 62837 51194 81.47
Plane 1646160 1288633 78.28
Sphere 2088478 1888015 90.40
Superellipsoid 100499 49469 49.22
Torus 147186 90281 61.34
Torus Bound 147186 109616 74.47
Bounding Box 14718995 3102360 21.08
Roots tested: 109616 eliminated: 43432
```

```
Peak memory used: 80715776 bytes
Render Time:
Photon Time: No photons
Radiosity Time: 0 hours 0 minutes 0 seconds (0.025 seconds)
using 8 thread(s) with 0.030 CPU-seconds total
Trace Time: 0 hours 0 minutes 0 seconds (0.717 seconds)
using 8 thread(s) with 5.051 CPU-seconds total
POV-Ray finished
CPU time used: kernel 0.25 seconds, user 5.87 seconds, total 6.12 seconds.
Elapsed time 1.72 seconds, CPU vs elapsed time ratio 3.56.
Render averaged 535813.95 PPS (150705.50 PPS CPU time) over 921600 pixels.
```

