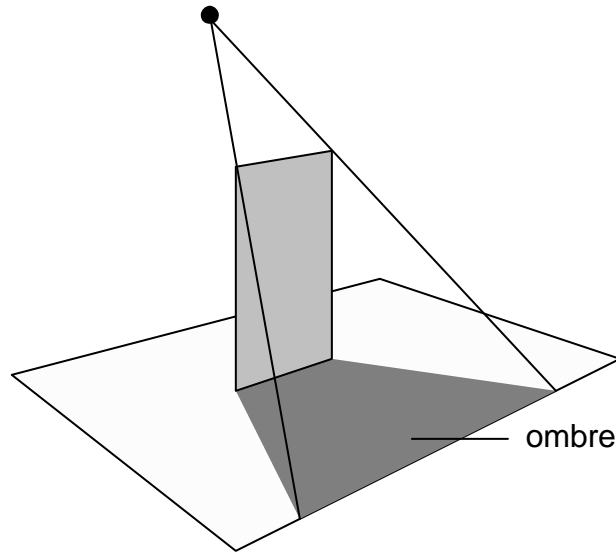
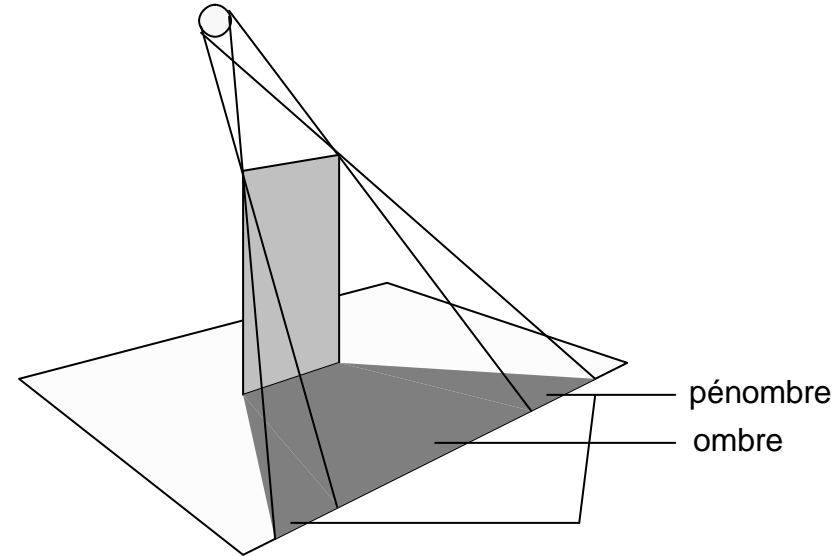


calcul des ombres

lumière ponctuelle

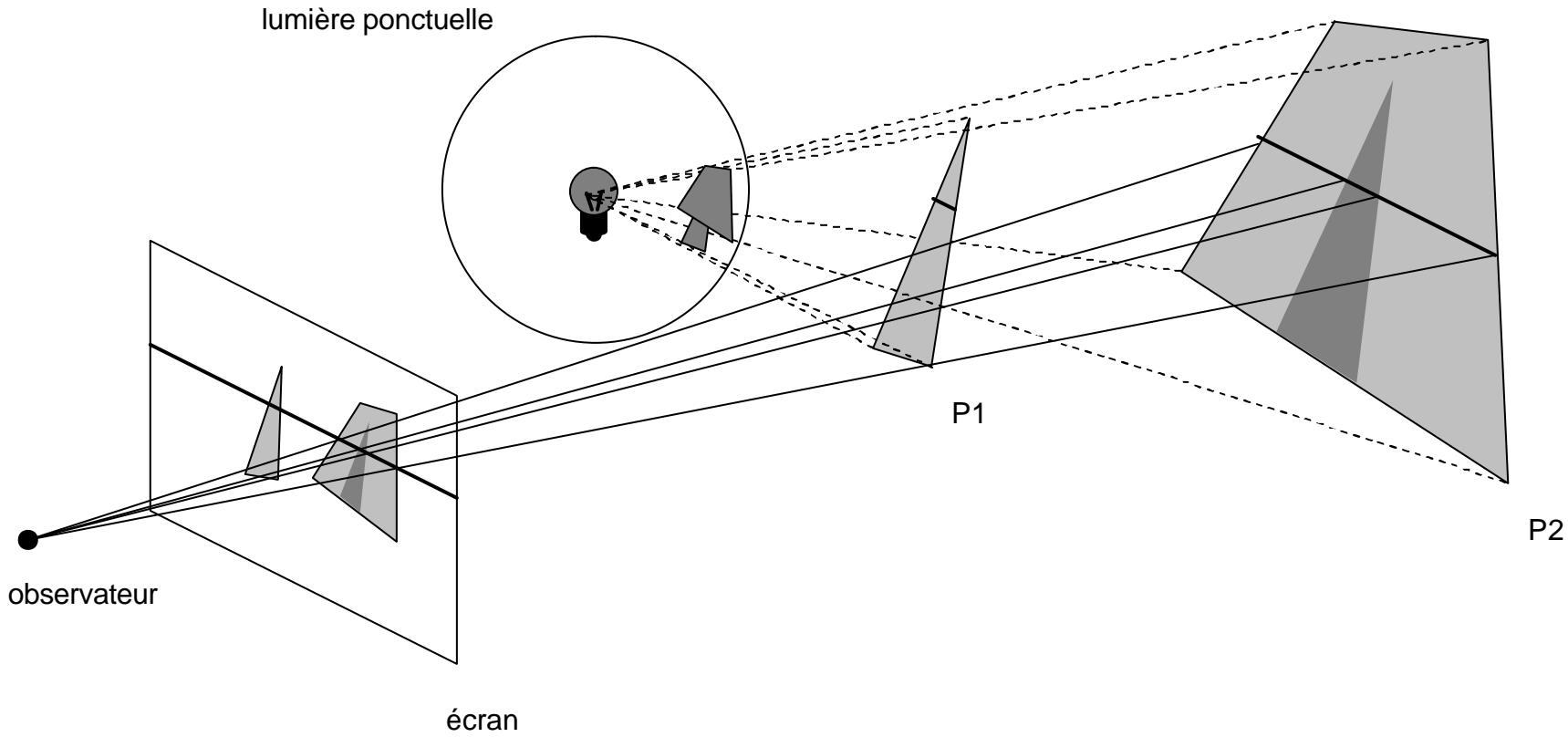


lumière non ponctuelle



- différentes méthodes de calcul :
 - plus ou moins de précision de calcul
 - calcul dépendant ou non du point de vue
 - lié aux modèles 3D et au type de rendu

- Intégration dans un rendu scan-line (Appel A. [APPE68], Bouknight W.J. [BOUK70])
 - recherche des polygones qui génèrent des ombres (ex : P1 projette son ombre sur P2)



- projection des polygones sur une sphère centrée sur la source de lumière :
recouvrement => ombre
- pour chaque segment :
 - si pas de recouvrement du segment par un polygone d'ombre :
cas normal
 - si recouvrement du segment par un polygone d'ombre :
diminution de l'intensité
 - si recouvrement partiel du segment par un polygone d'ombre :
subdivision et retour aux cas précédents

- polygone d'ombre (Atherton, Weler, Greenberg [ATHE78])

- méthode :

- transformation des polygones dans le repère de la lumière

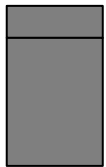
- suppression des faces cachées => polygones d'ombre

- transformation des polygones d'ombre dans le repère de l'objet

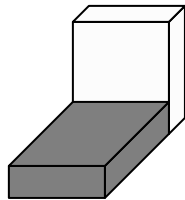
- calcul précis

- information quantitative sur les ombres

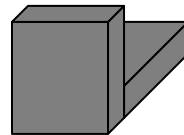
- application en DAO



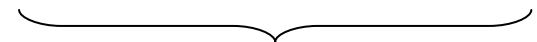
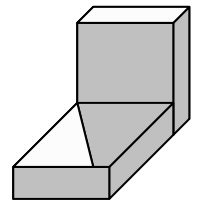
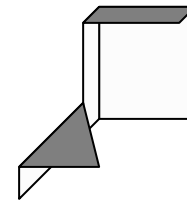
vue du dessus



repère de l'observateur



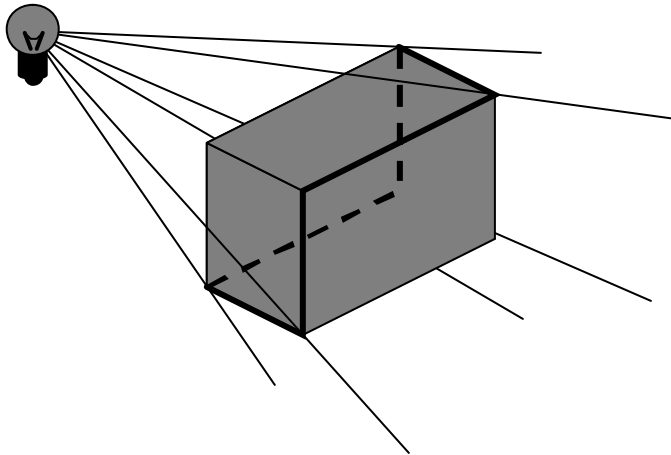
repère de la lumière



repère de
l'observateur

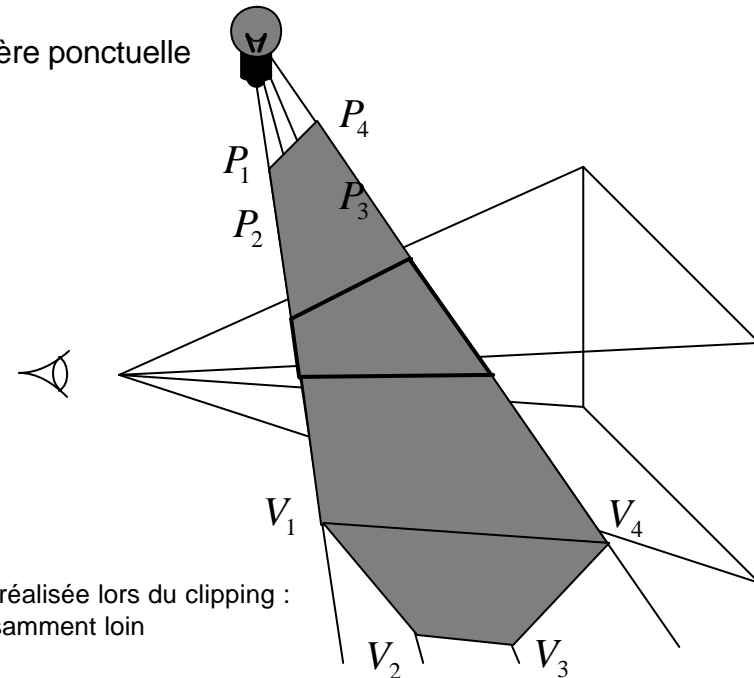
- volume d'ombre (Crow, F.C [CROW77])
 - méthode :
 - pour chaque objet
 - calcul de la bordure de sa silhouette vue de la source de lumière
 - calcul du volume d'ombre généré
- lors du rendu :
 - pour chaque pixel
 - tri des profondeurs
 - si un objet appartient au volume d'ombre => ombré

lumière ponctuelle

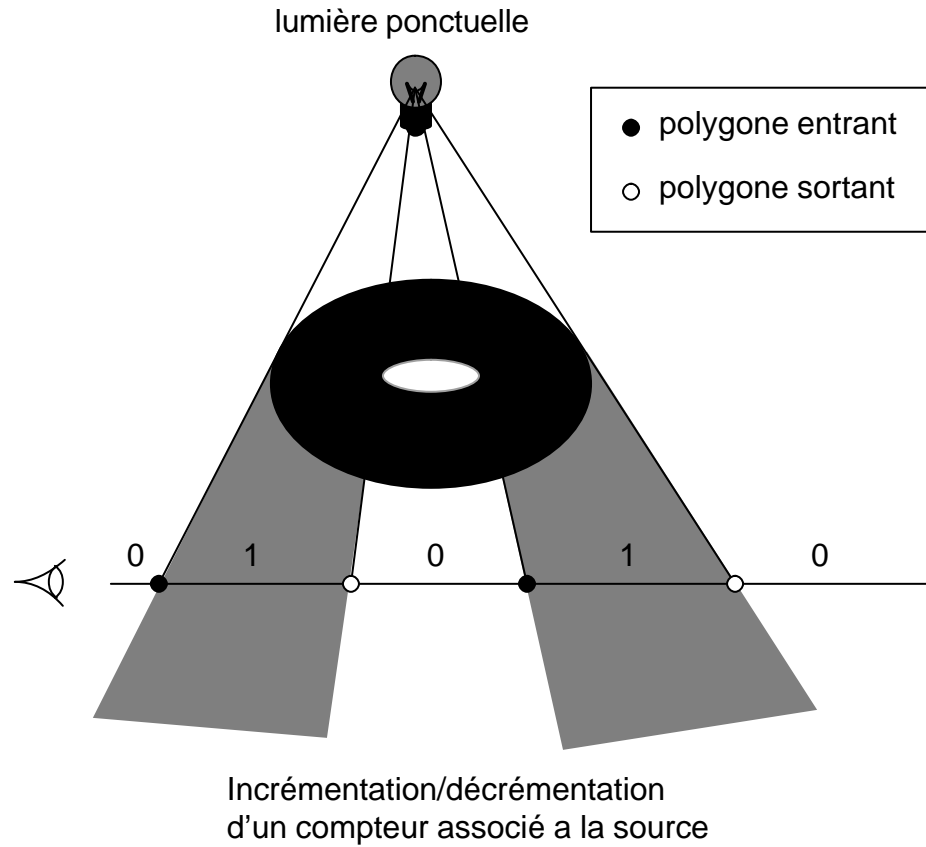


bordure de la silhouette

lumière ponctuelle



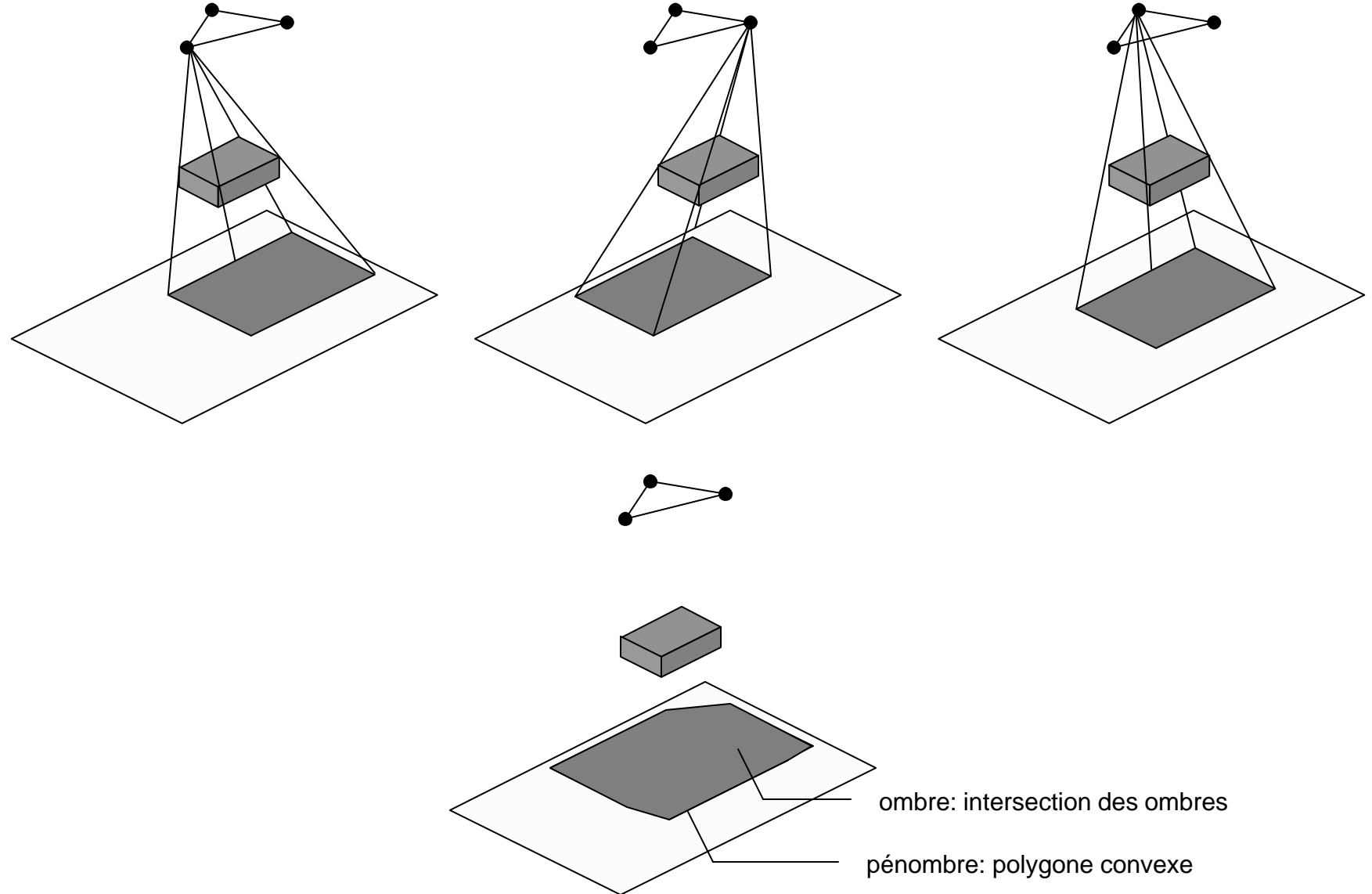
intersection réalisée lors du clipping :
 V_i pris suffisamment loin



- limite :
validité du volume d'ombre dépendant de la validité de l'objet

- ombre/pénombre (Nishita, Nakamae [NISH85])

lumière non ponctuelle



- volumes de lumière (Max N.[MAX86a], Nishita T. [NISH87])
 - variante de la méthode des volumes d'ombre
 - rendu d'effets atmosphériques : faisceaux de lumière



- Z-Buffer d'ombre (shadow-maps)

- méthode

pour chaque source de lumière :

rendu en profondeur de la scène du point de vue de la source

pour chaque pixel de l'image :

pour chaque lumière

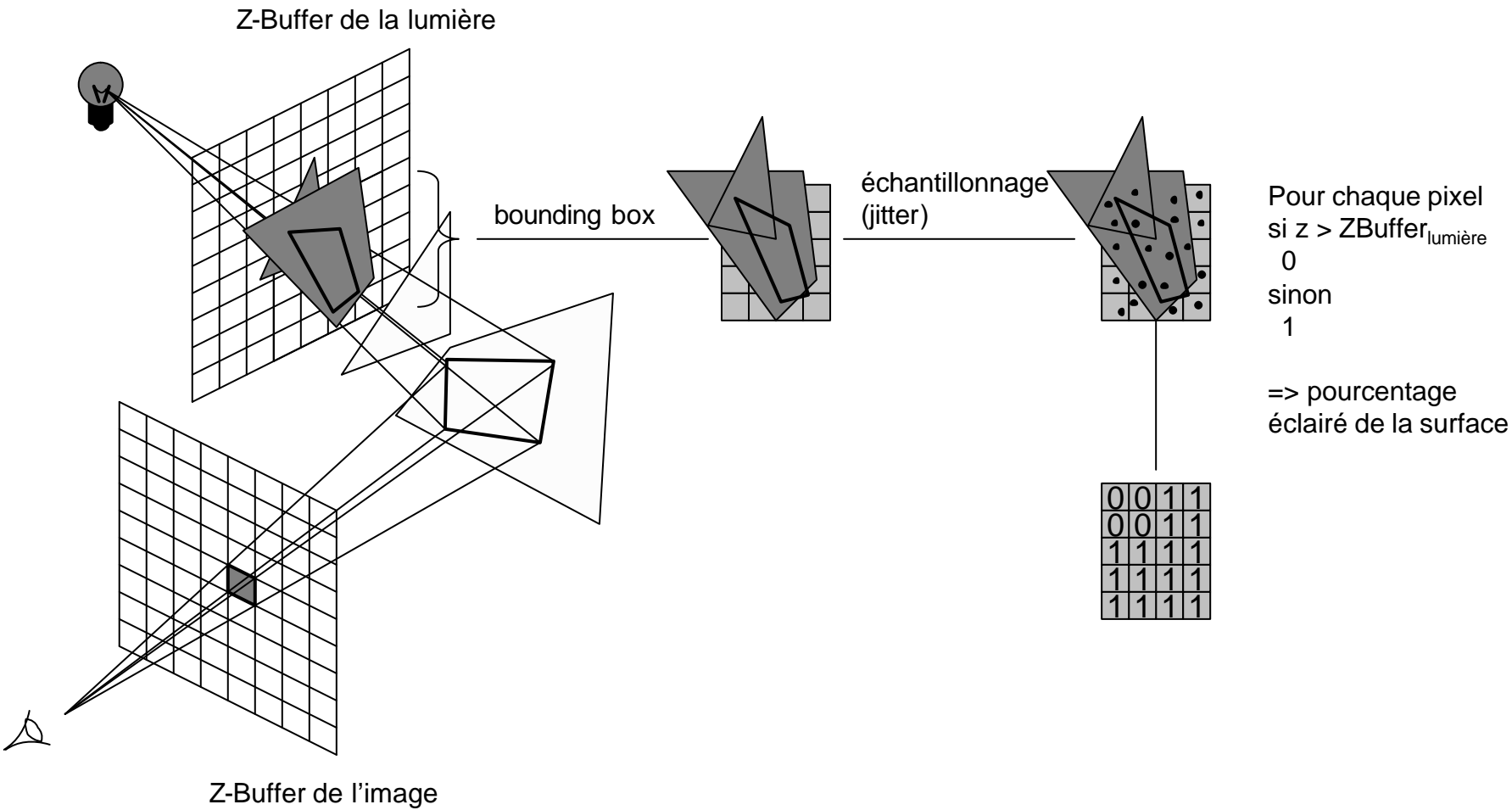
transformation de la profondeur contenu dans le Z-Buffer de l'image dans le repère de la lumière

si profondeur du Z-Buffer de l'image < profondeur du Z-Buffer de la lumière,
alors ombre

- limites

réalisation du calcul après génération de l'image : effet de voile sur les highlights

problèmes d'aliasing

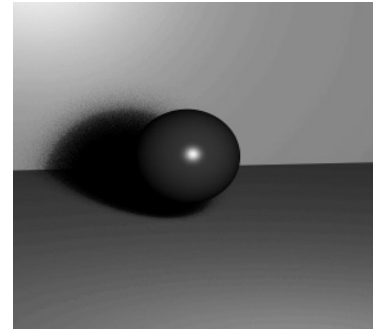
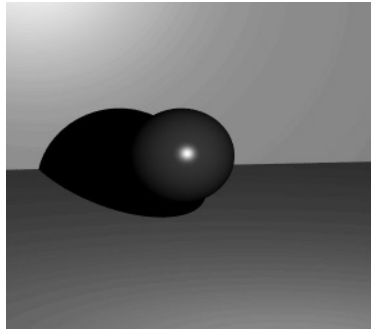
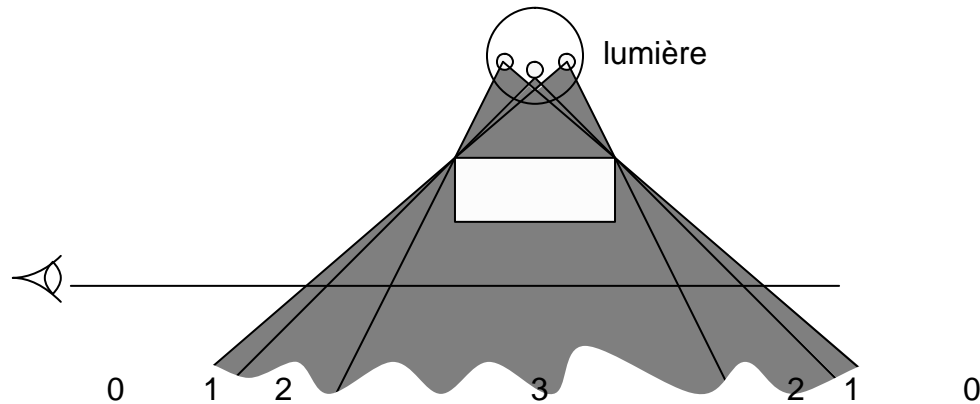


- ombres douces

méthode basée sur les volumes d'ombre (Brotmann, Badler [BROT84])

- méthode :
 - combinaison avec un Z-Buffer
 - compteur = «niveau d'ombre» (ombres/pénombres)

une lumière = un ensemble de points choisis de façon stochastique



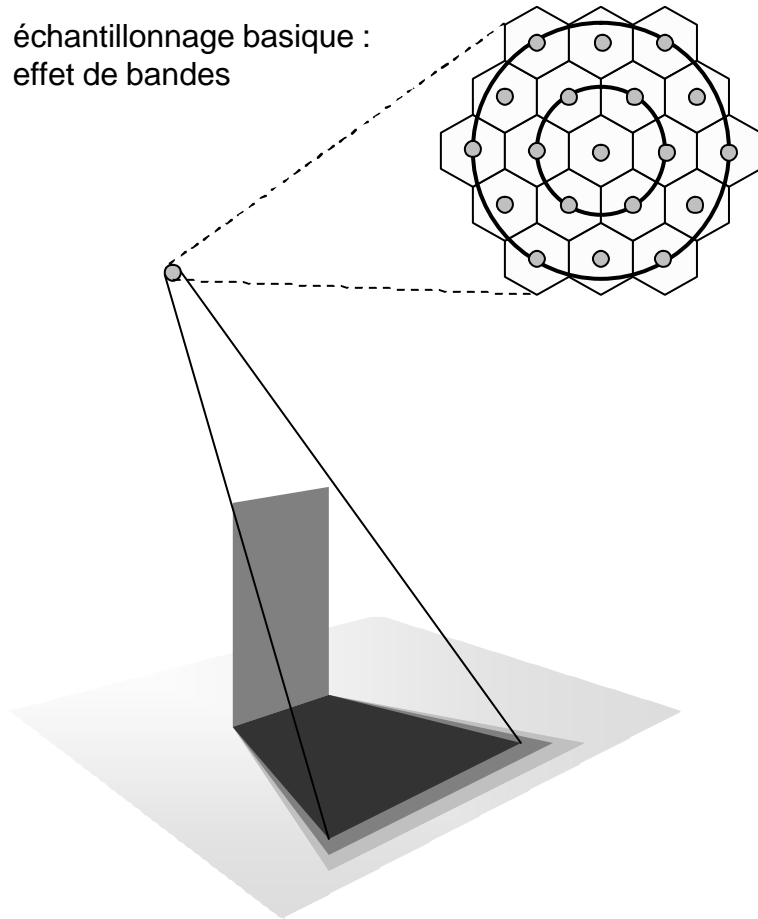
- ombres douces antialiasée

- méthode

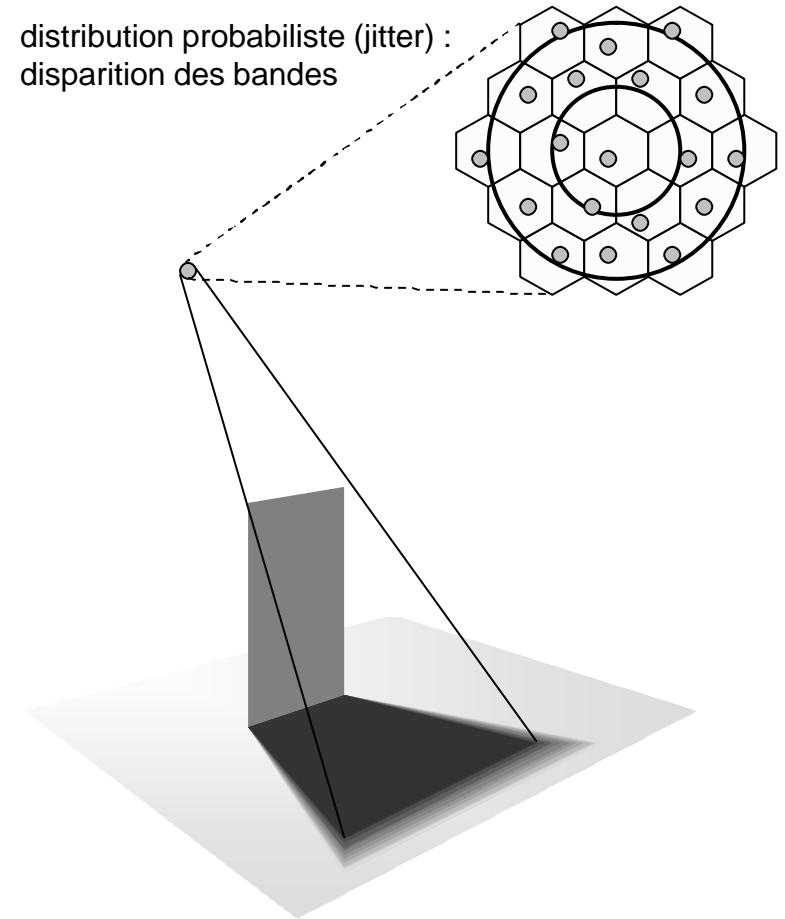
discrétisation d'une lumière sphérique en hexagones

calcul contribution de la lumière

échantillonnage basique :
effet de bandes



distribution probabiliste (jitter) :
disparition des bandes

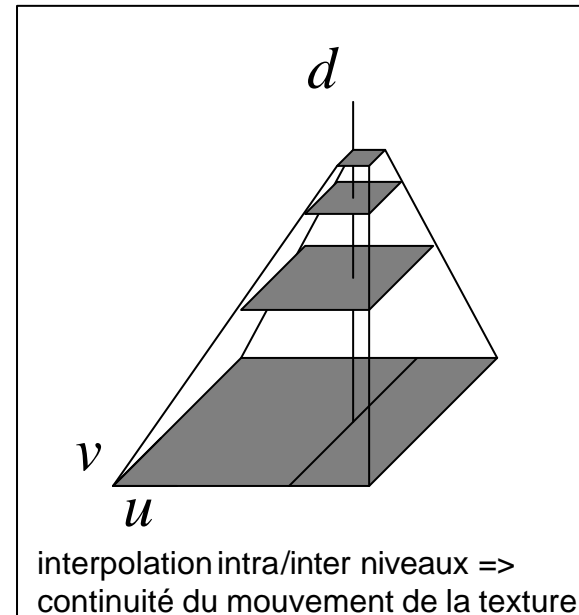
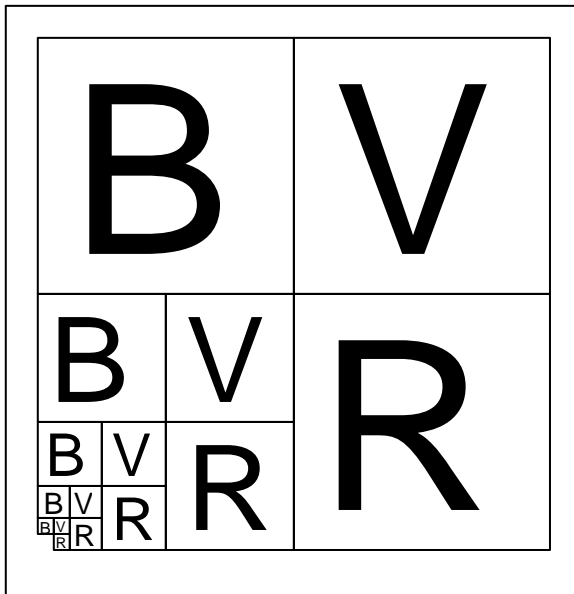
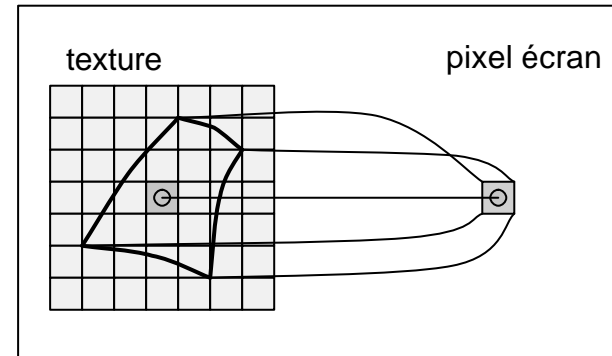


- mip-mapping

problème d'échantillonnage dans la texture

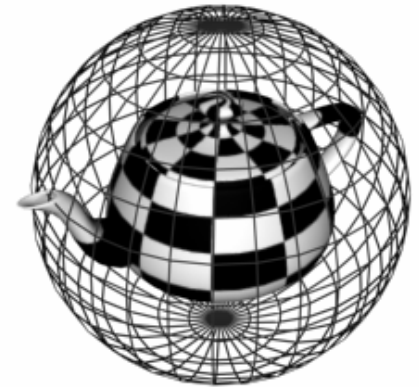
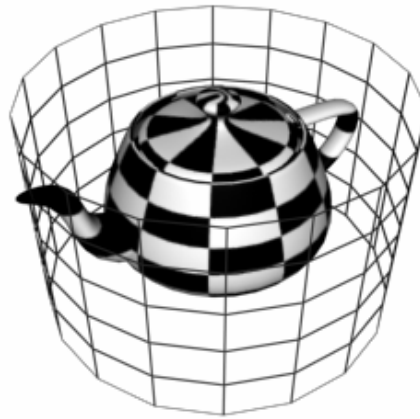
pré-calcul d'une texture multi-résolutions
avec anti-aliasing

aux coordonnées de texture (u,v) se rajoute d , indice de la résolution

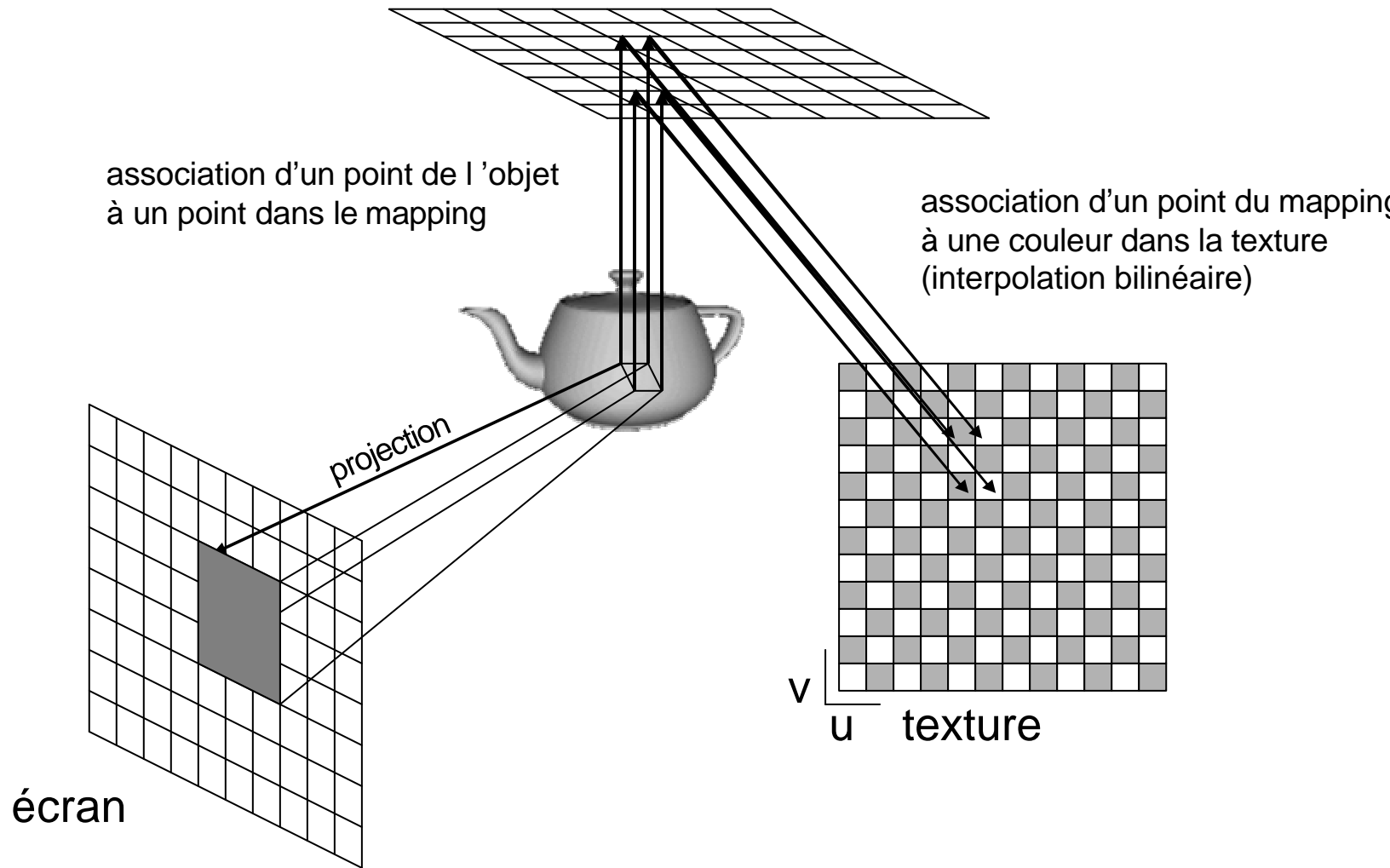


- **Plaquage de texture**

association d'un point 2D de la texture à un point de l'écran, projeté d'un point 3D d'une surface



mapping planaire, cylindrique, sphérique



• mapping cylindrique

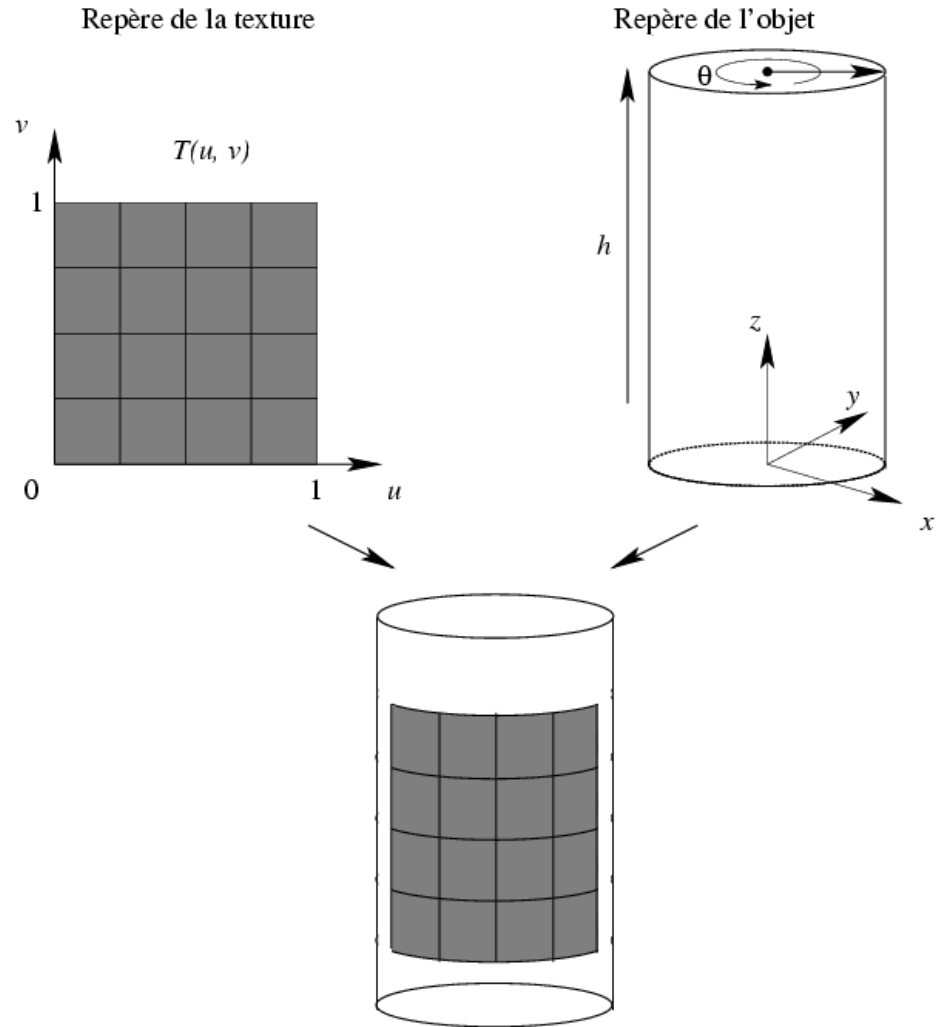
Représentation d'un point
du cylindre

$$(r \cos \mathbf{q}, r \sin \mathbf{q}, hz)$$

où r rayon, h hauteur

$$(u, v) = (\mathbf{q} / 2\mathbf{p}, z)$$

où $u, v \in [0, 1]$



- mapping sphérique

Représentation d'un point
du cylindre

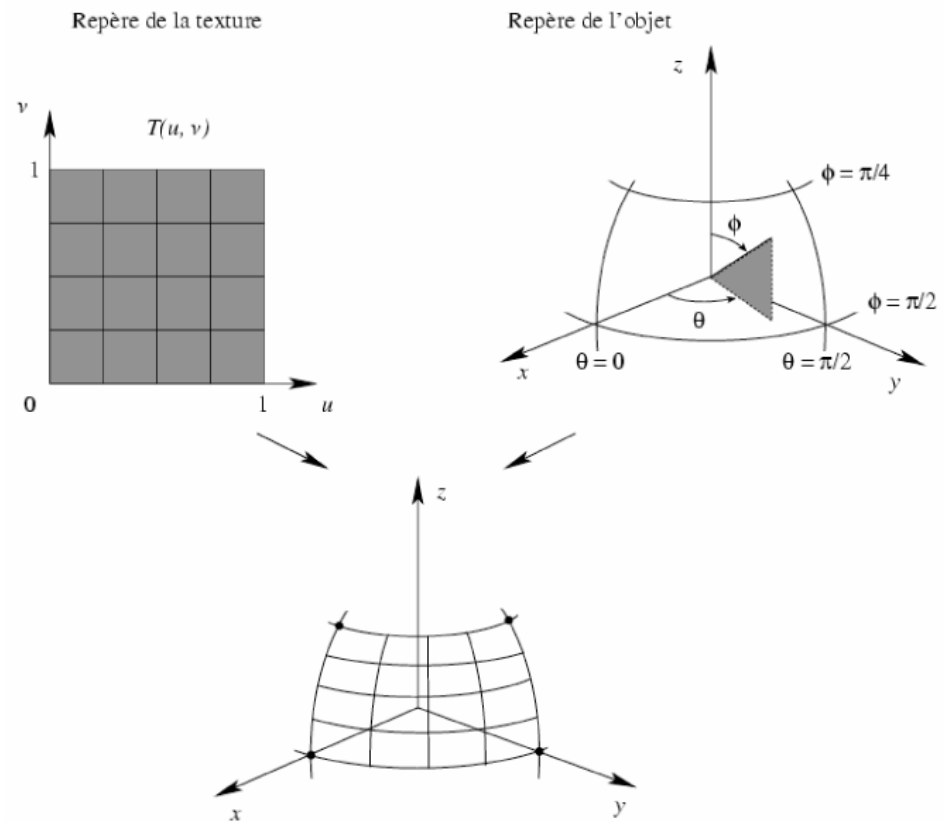
$$(r \cos \mathbf{q} \sin \mathbf{f}, r \sin \mathbf{q} \sin \mathbf{f}, r \cos \mathbf{f})$$

où $0 \leq \mathbf{q} \leq \mathbf{p} / 2$ et $-\mathbf{p} / 2 \leq \mathbf{f} \leq \mathbf{p} / 2$

Représentation d'un point
dans la texture

$$(u, v) = \left(\mathbf{q} / 2\mathbf{p}, \frac{\mathbf{f} + (\mathbf{p} / 2)}{\mathbf{p}} \right)$$

où $u, v \in [0, 1]$



- **plaquage des points intérieurs d'un polygone**

- cas fonction de projection connue :

pour chaque point projeté, les coordonnées dans l'image de texture sont connues

- cas fonction de projection inconnue :

un segment de l'écran => une ligne dans l'image de texture
sommets des polygones => mémorisation des coordonnées de l'image de texture (continuité)

- plaquage de texture dépendant du point de vue
 - approximation de l'environnement réfléchi (ou réfracté)
 - approximation du lancé de rayons
 - texture dépendante du point de vue

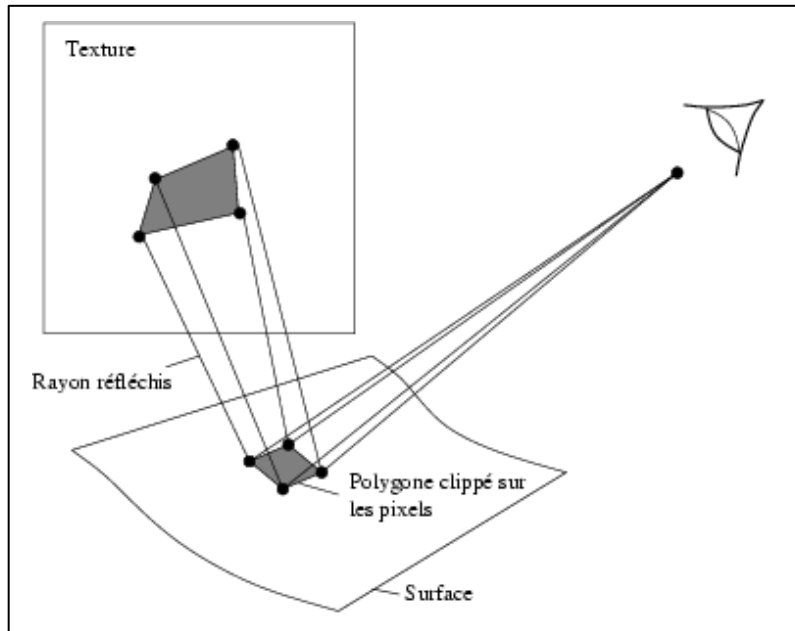


animation mapping
sphérique



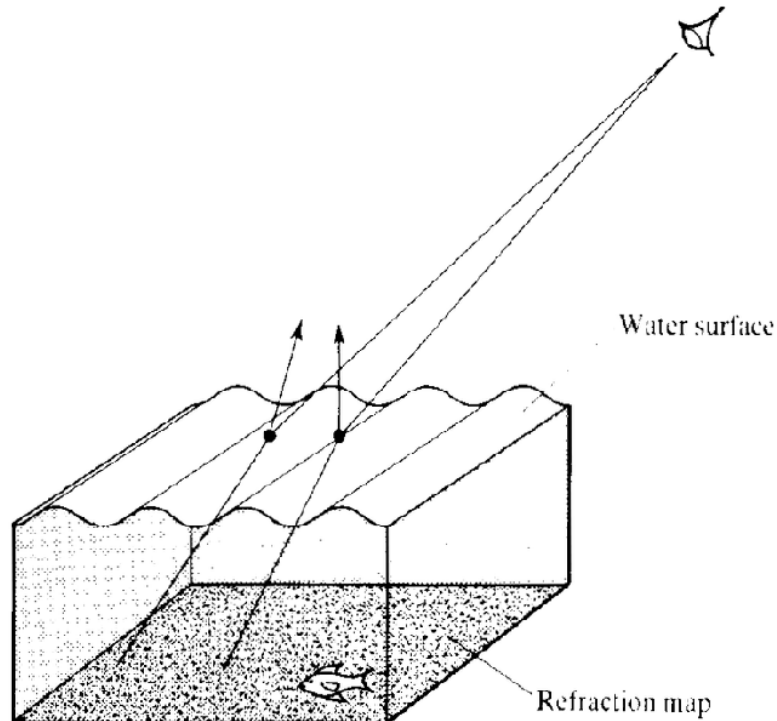
animation mapping
d'environnement

- plaquage de texture dépendant du point de vue :
chrome/ réflexion
 - plaquage de texture par réflexion
 - illusion suffisante



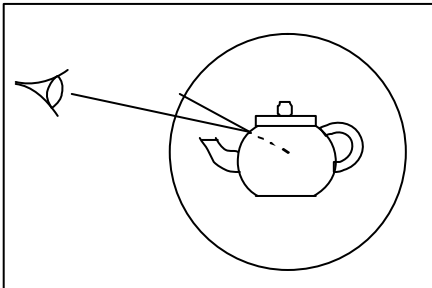
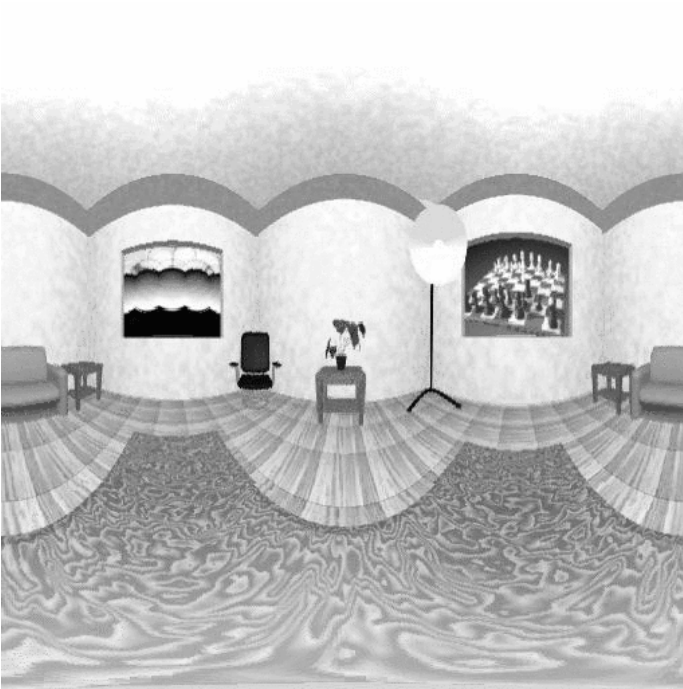
$$\left. \begin{aligned} u &= \frac{1}{2} (1 + 1/p \tan^{-1}(R_y / R_x)) & -p < \tan^{-1} < p \\ v &= \frac{R_z + 1}{2} \end{aligned} \right\}$$

- plaquage de texture dépendant du point de vue :
réfraction
 - plaquage de texture des rayons réfractés
 - combinaison avec réflexion: effet de vagues



- plaquage de texture dépendant du point de vue:
texture d'environnement
 - limites de cette méthode :
 - objets suffisamment distants
 - objet texturé ne se reflétant pas lui-même
 - projection de l'environnement sur une surface englobante :
cube ou sphère
 - cas sphère: idem que la méthode de réflexion
 - cas cube :
 - moins de distorsion
 - acquisition à partir de vraies photos

- cas sphère : idem que la méthode de réflexion

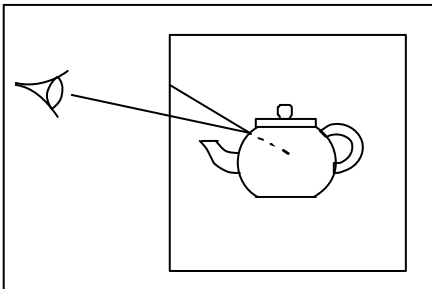
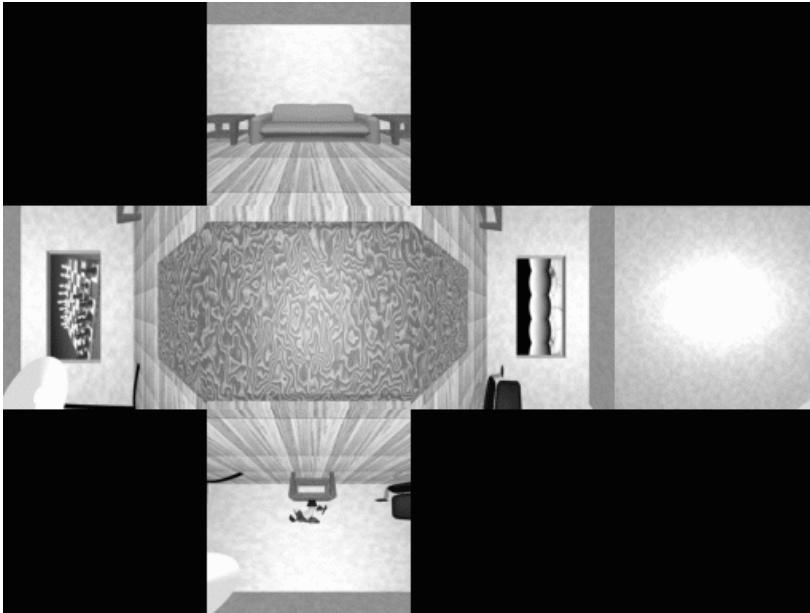


Yoshihiro Mizutani and Kurt Reindel

<http://home.san.rr.com/thereindels/Mapping/>

– cas cube :

- moins de distorsion
- acquisition possible à partir de vraies photos



Yoshihiro Mizutani and Kurt Reindel

<http://home.san.rr.com/thereindels/Mapping/>

- bump mapping (Blinn)

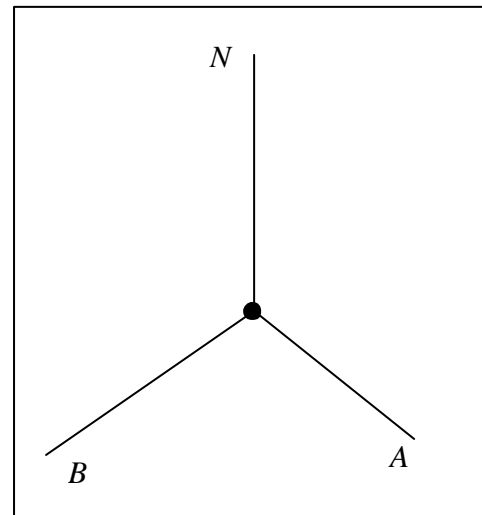
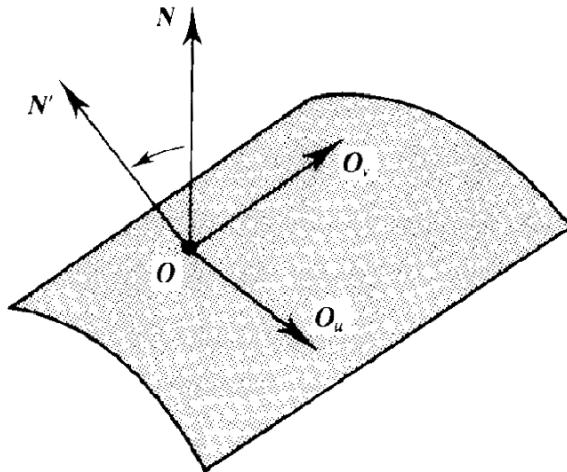
- effet de surface bosselée
- limite: contour silhouette non bosselé
- construction sur repère basé sur dérivées locales de la surface, sinon animation de la texture

$O(u, v)$ point de la surface

repère A, B, N:

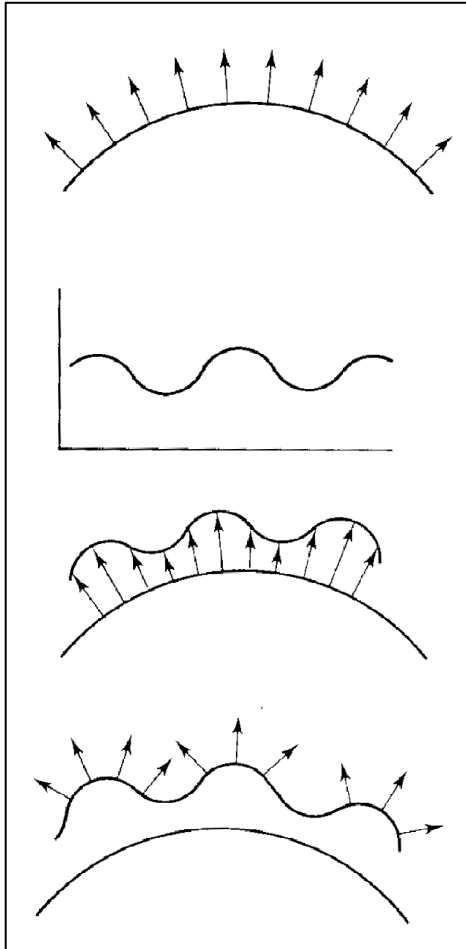
- $N = O_u \wedge O_v$ normale (O_u, O_v dérivées partielles)

- $A = N \wedge O_v$



– perturbation

- perturbation $B(u, v)$ (B_u, B_v) dérivées partielles)
- $N' = N + D$ normale perturbée
- $D = B_u A - B_v B$ vecteur de perturbation

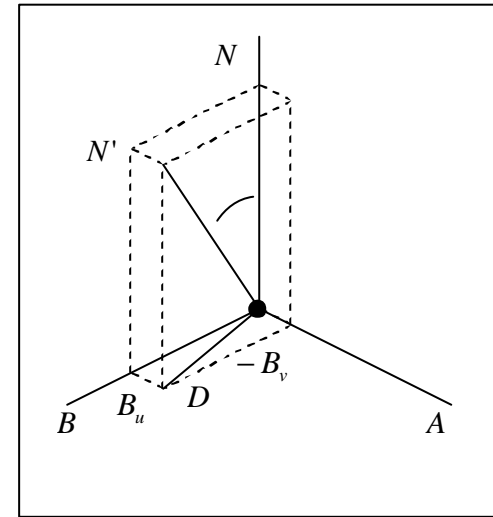


$O(u)$ surface originale

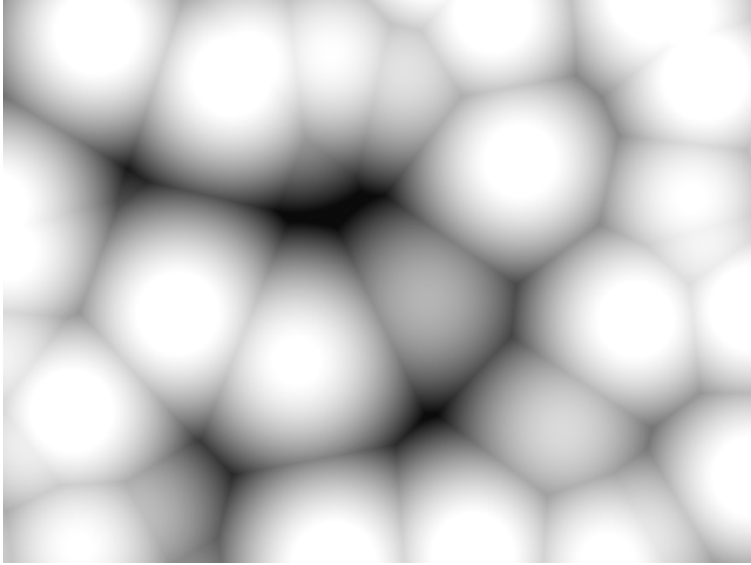
$B(u)$ bump map

$O'(u)$ agrandissement ou rétrécissement de $O(u)$ avec $B(u)$

$N'(u)$ vecteur de la nouvelle surface



- rendu



- **modeleurs et moteurs de rendu**

3D Studio, Accurender, Adeline, Alias Power Animator, Amapi, Animation Master, ART (Vort), Art*lantis, AutoCAD, AVRIL, AXIS, Behemot Graphics Editor, Beyond 3D, Blender, Blue Moon Rendering Tools (BMRT), BRL-CAD, Bryce, Cinema4D, CSG RayTracer, Design Workshop, DKBTrace, DragonRay, ElectricImage, Extreme 3D, Form-Z, Gamma-Ray, Genesys, Geometique, GeomView, Grafica Natura, Helios, Hoops, Imagine, INDIA, Infini-D, JawRay, Light, LightScape, LightWave, LightWorks, Lumin Micro, Magic Camera, MARS, Masterpiece, Maverick, Maya, Megahedron, MentalRay, Microcosm, MiniCAD, ModelView, Moonlight Creator, Moon Lite, MTV, Natural Scene Designed, NetV, NuGraf, Panard Vision, Panorama, ParaSolid, Pixcon/Anitroll, Pixels3D, PMR, PolyRay, PortRAY, POV-Ray, Powder, ProPhotorender, QRT, R95, RAD, Radiance, RAY++, Ray Dream Studio, Ray4, RayGun, RayShade, RayStorm, Red Light, Relativistic Raytracer, RenderDotC, RenderGL, Renderman, RenderPark, Rhino, RTrace, SART, Sculpt 3D, SIPP, SoftCAD, Soft F/X, SoftImage 3D, SoftPlan, SolidThinking, SolidView, sPatch, StrataStudio/Vision, Threedom, TrueSpace 2, TurboCAD, Vellum, VFleet, Vision, Visual Reality, VisuVoxel, Vivid, VolPack, Volsh, WatRend, WebView3D, XRay

- **modeleurs: les plus célèbres**

- **3D Studio (Kinetix, division de Autodesk Inc.)** 
PC. Rendu Scanline. Modeleur, rendu et animation (incluant la cinématique).
- **Amapi 3D (Template Graphics Software)**
Macintosh et Windows NT. Modeleur de NURBS/polygones, conçu pour exporter des modèles pour d'autres logiciels de rendu et d'animation. Possède son moteur de rendu.
- **Bryce (MetaCreations)**
Macintosh et Windows. Modeleur, moteur de rendu, animation. Rendu sur réseau.
- **LightWave 3D (NewTek Inc.)**
Windows 95/98/NT, Mac, Dec Alpha, Amiga. LightWave est principalement un outil destiné à la télévision et au cinéma. Moteur de rendu, modeleur polygones/NURBS, animation. Calcul de rendu sur réseau.
- **Maya (Alias Wavefront)** 
SGI et Windows NT. Hérite de IPR et de certaines capacités de rendu de Alias Power Animator's. Design C++ orienté objet, implémentation OpenGL. Langage de programmation MEL (Maya Embedded Language) permettant des personifications faciles.

- **POVRAY (POV-Team)**

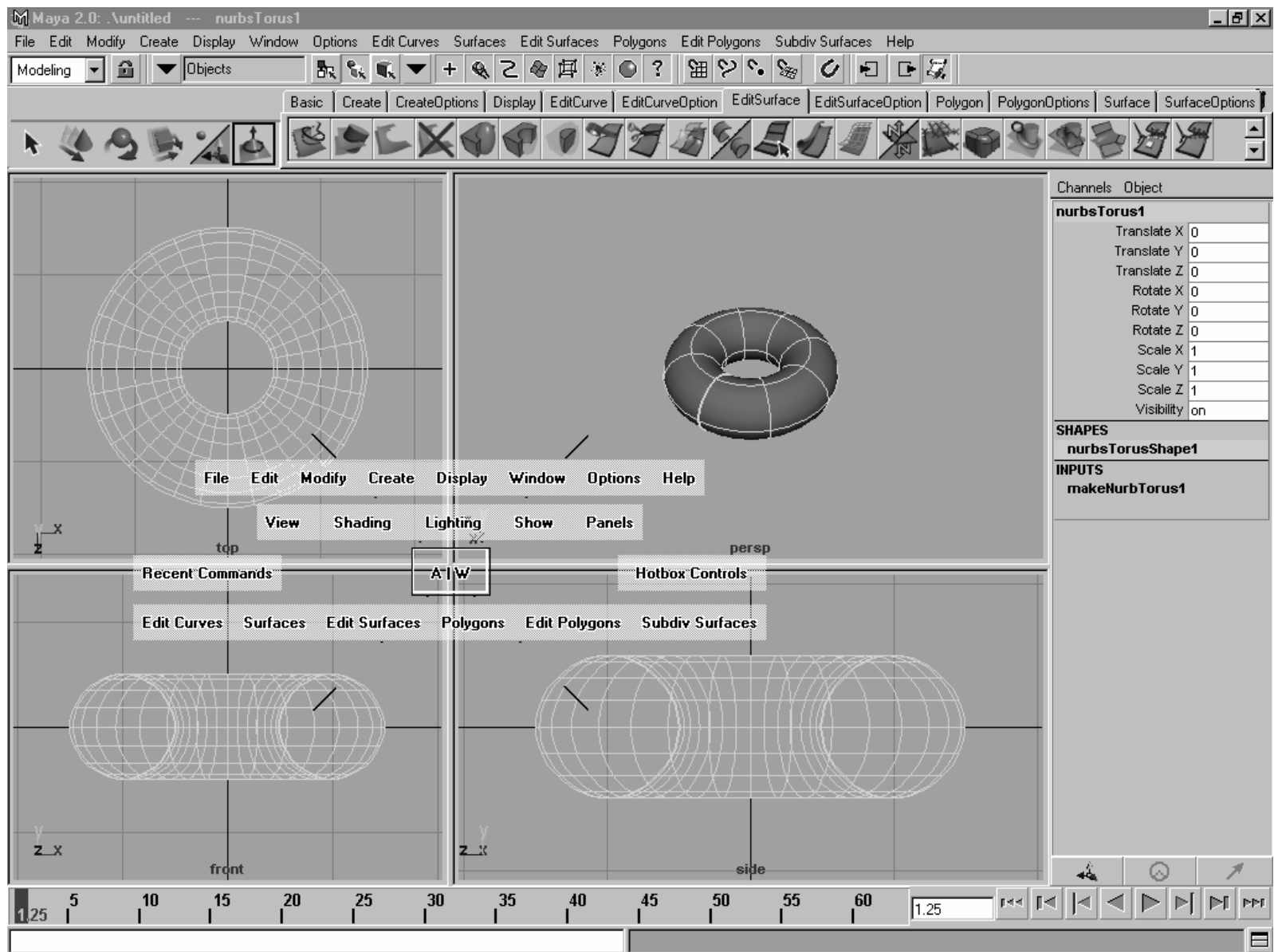
Logiciel de lancé de rayons, (Persistence of Vision), basé sur DKB-Trace (David Buck). Peut-être actuellement le moteur de rendu gratuit le plus mature et le plus utilisé. La version actuelle supporte la technique de rendu de Monte Carlo.

- **Pixar RenderMan (Pixar)**

Ancêtre des grands logiciels de rendu, et source de beaucoup de techniques utilisées dans les logiciels actuels. Rendu scanline.

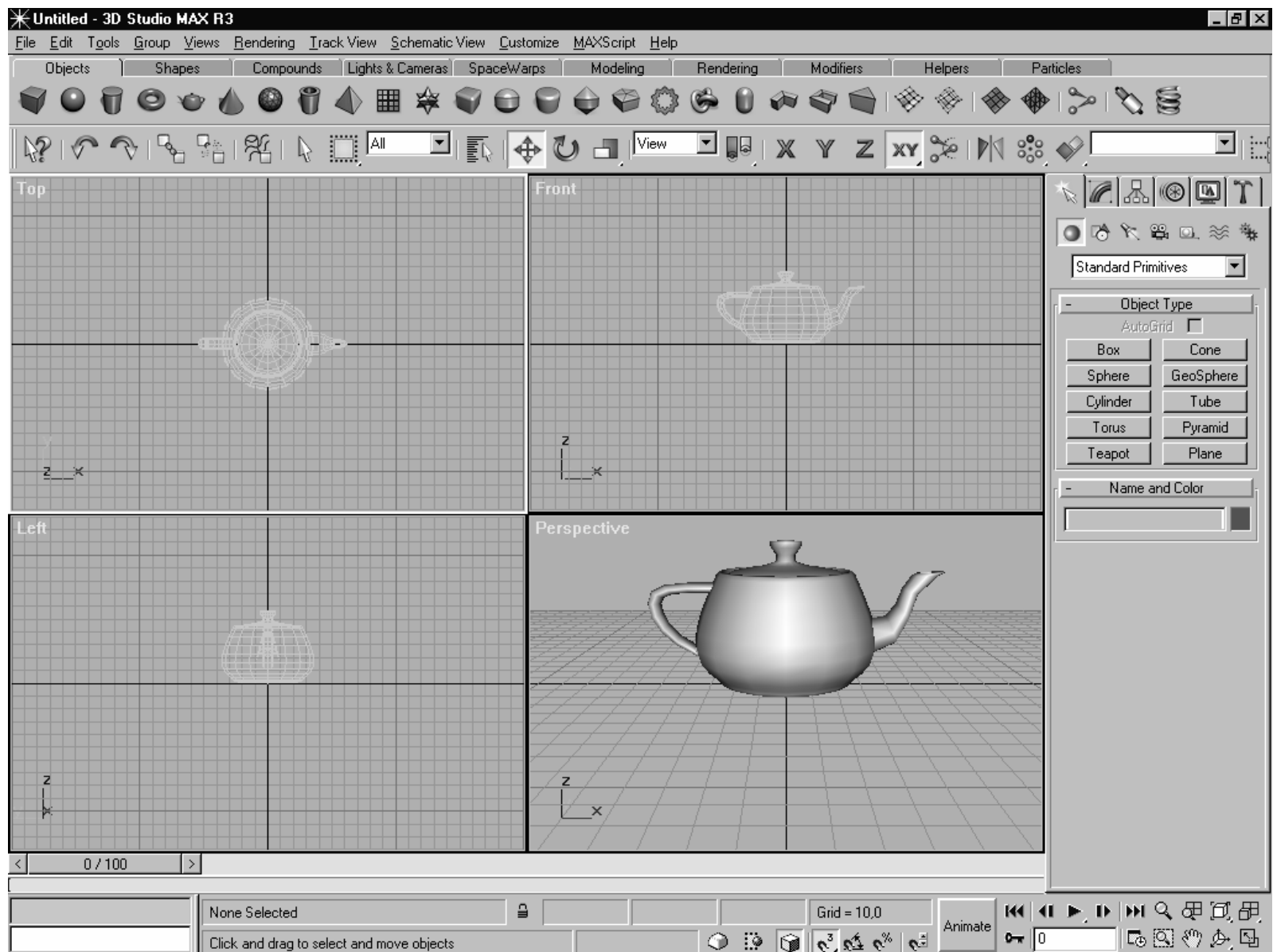
- **SoftImage 3D (Avid Technology Inc)**

Windows/NT. Modeleur, moteur de rendu, animation. Bien que possédant son propre moteur de rendu, il est souvent distribué avec le moteur MentalRay.



Alias Wavefront-Maya

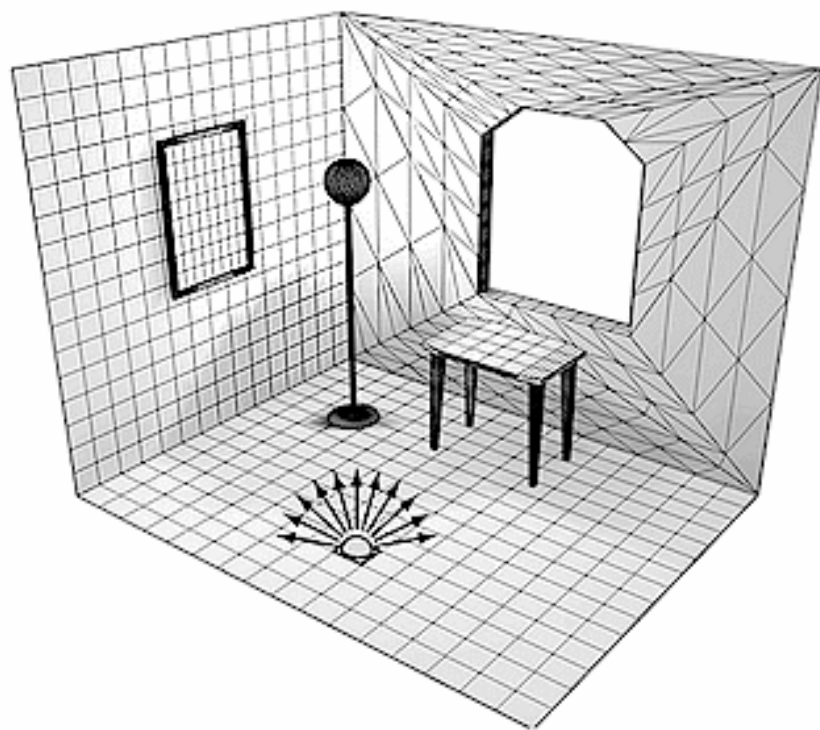




Kinetix-3D Studio MAX

Radiosité

- gestion du transport de la lumière diffuse
- hypothèse :
surface = diffuseur parfait (lambertien)
- modèle de lumière diffuse
 - calcul indépendant de la vue
 - débordement des lumières diffuses (bleeding)
- division du monde en zones rectangulaires : patches
 - précision de la solution influencée par leurs tailles
- possibilité d'observation des états intermédiaires de la solution



Lightscape
<http://www.lightscape.com/>

- radiosité * surface = énergie émise + énergie réfléchie

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

$$\text{cas discret : } B_i A_i = E_i A_i + r_i \sum_{j=1}^n B_j A_j F_{A_j A_i}$$

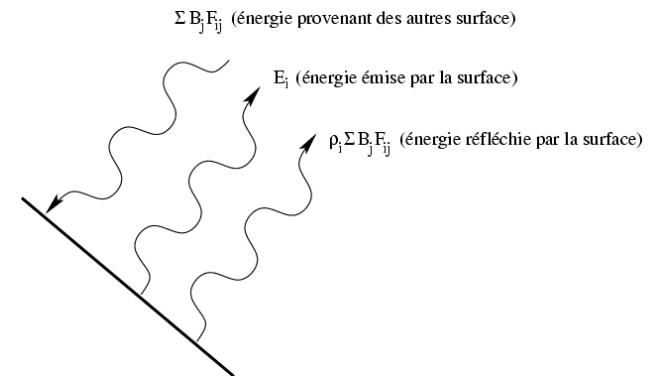
$$\text{or } F_{ij} A_i = F_{ji} A_j \text{ (réciprocité)} \Rightarrow F_{ij} = F_{ji} \frac{A_j}{A_i}$$

\Rightarrow

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

- où

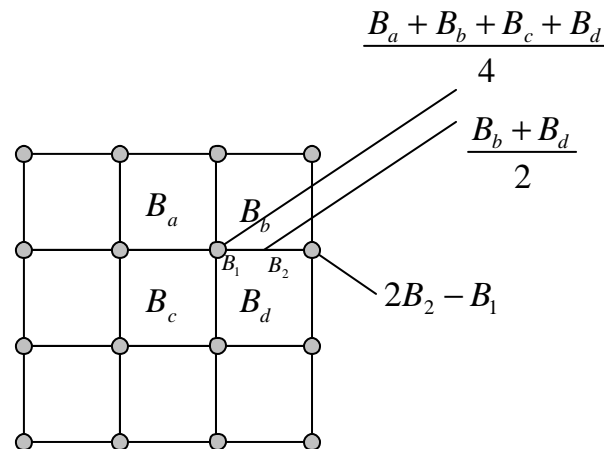
- B_i radiosité de la surface i
- E_i émission de la surface i
(nulle si n'émet pas de lumière)
- r_i réflexion de la surface i
- F_{ij} fraction d'énergie quittant j
pour aller vers i (appelé facteur de forme)



- Forme matricielle

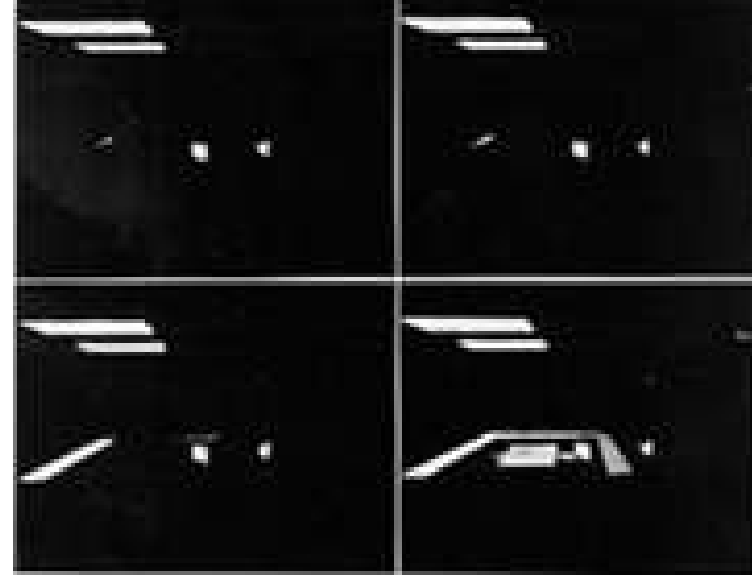
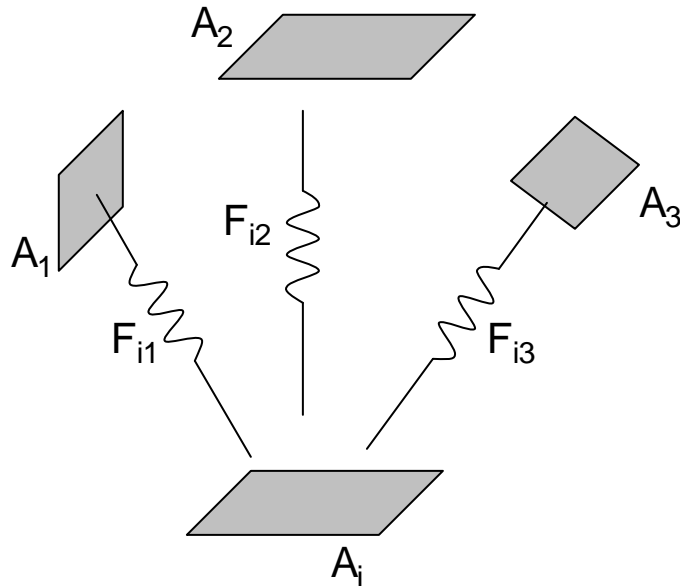
$$\begin{bmatrix} 1 - \mathbf{r}_1 F_{11} & -\mathbf{r}_1 F_{12} & \cdots & -\mathbf{r}_1 F_{1n} \\ -\mathbf{r}_2 F_{21} & 1 - \mathbf{r}_2 F_{22} & \cdots & -\mathbf{r}_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{r}_n F_{n1} & -\mathbf{r}_n F_{n2} & \cdots & \mathbf{r}_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

- interpolation des patches



- Méthode de rassemblement

réception de l'énergie émise par toutes les surfaces de l'environnement



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

résolution par la méthode de Gauss - Seidel

Visuellement :

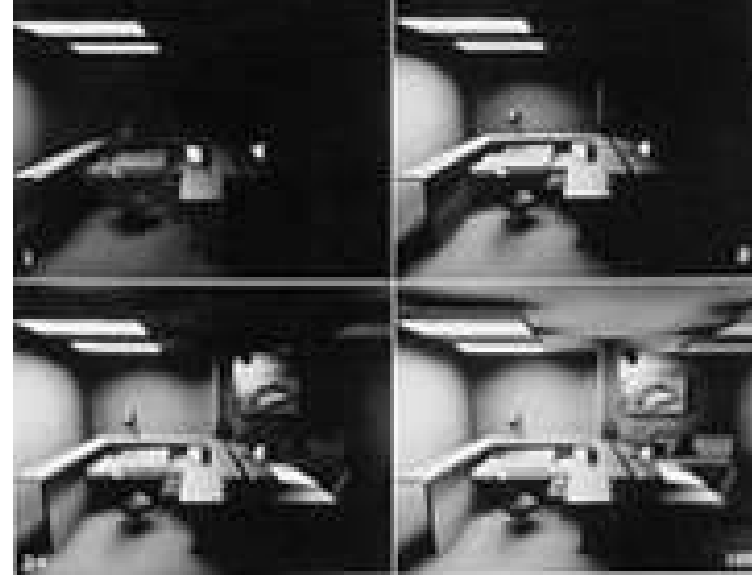
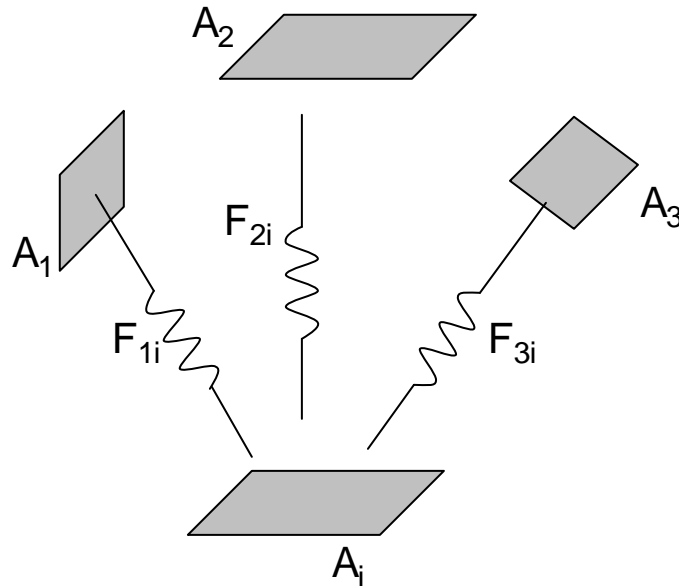
éclairage un à un de chaque patch de la scène

Inconvénient :

conservation d'une matrice des F_{ij}

- Méthode de tir

pour chaque patch, envoi de son énergie sur chaque patch de la scène



$$(B_j)_{t=0} = E_j$$

$$(B_j)_{t+1} = (B_j)_t + B_i \mathbf{r}_i F_{ji}$$

$$\text{où } F_{ji} = F_{ij} \frac{A_i}{A_j}$$

Visuellement :

propagation dans l'image de la lumière (accroissement de l'intensité lumineuse au fur et à mesure du rendu)

sélection du patch possédant le plus d'énergie (lumières)

- Méthode de tir avec terme ambiant

ajout d'un terme ambiant à la méthode de tir dans la phase de visualisation : globalement l'intensité ne change pas

$$F_{ij(estimé)} = \frac{A_j}{\sum_{i=1}^n A_i}$$

$$\mathbf{r}_{approché} = \frac{\sum_{i=1}^n \mathbf{r}_i A_i}{\sum_{i=1}^n A_i}$$

$$\mathbf{m} = 1 + \mathbf{r}_{approché} + \mathbf{r}_{approché}^2 + \dots = \frac{1}{1 - \mathbf{r}_{approché}}$$

$$\Rightarrow (I_a)_i = \mathbf{m} \sum_{j=1}^N \Delta B_j F_{ij(estimé)}$$



- Accroissement de la précision de la solution

$$F_{ij} = \frac{1}{A_i} \sum_{q=1}^R F_{(iq)j} A_{(iq)}$$

$$B_{iq} = E_{iq} + \mathbf{r}_{iq} \sum_{j=1}^n B_j F_{(iq)j}$$

iq élément q du patche i

- facteur de forme élémentaire:

angle solide:

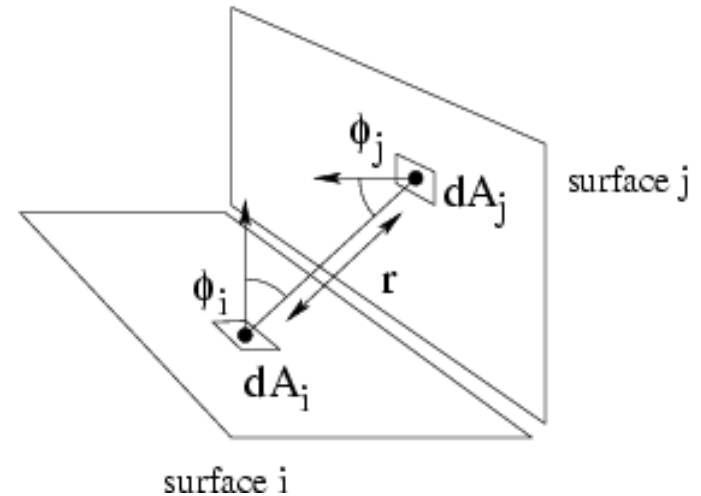
$$d\mathbf{w}_{ij} = \cos \mathbf{j}_j \frac{dA_j}{r^2}$$

énergie quitant dA_i , allant vers dA_j :

$$\begin{aligned} dE_i dA_i &= I_i \cos \mathbf{j}_i d\mathbf{w}_{ij} dA_i \\ &= I_i \cos \mathbf{j}_i \cos \mathbf{j}_j dA_i \frac{dA_j}{r^2} \end{aligned}$$

Facteur de forme :

$$F_{dA_i dA_j} = \frac{\text{énergie radiative atteignant } dA_j \text{ depuis } dA_i}{\text{énergie totale partant de } dA_i \text{ dans toutes les directions}}$$



pour une surface lambertienne idéale :

$$E_i = \frac{I_i \cos \mathbf{j}_i \cos \mathbf{j}_j dA_i dA_j}{I_i dA_i \mathbf{p} r^2}$$

$$= \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{\mathbf{p} r^2}$$

facteur de forme de dA_i vers A_j :

$$F_{dA_i A_j} = \int_{A_j} \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{\mathbf{p} r^2}$$

Facteur de forme :

$$F_{A_i A_j} = F_{ij} = \frac{\text{énergie radiative atteignant } A_j \text{ depuis } A_i}{\text{énergie totale partant de } A_i \text{ dans toutes les directions}}$$

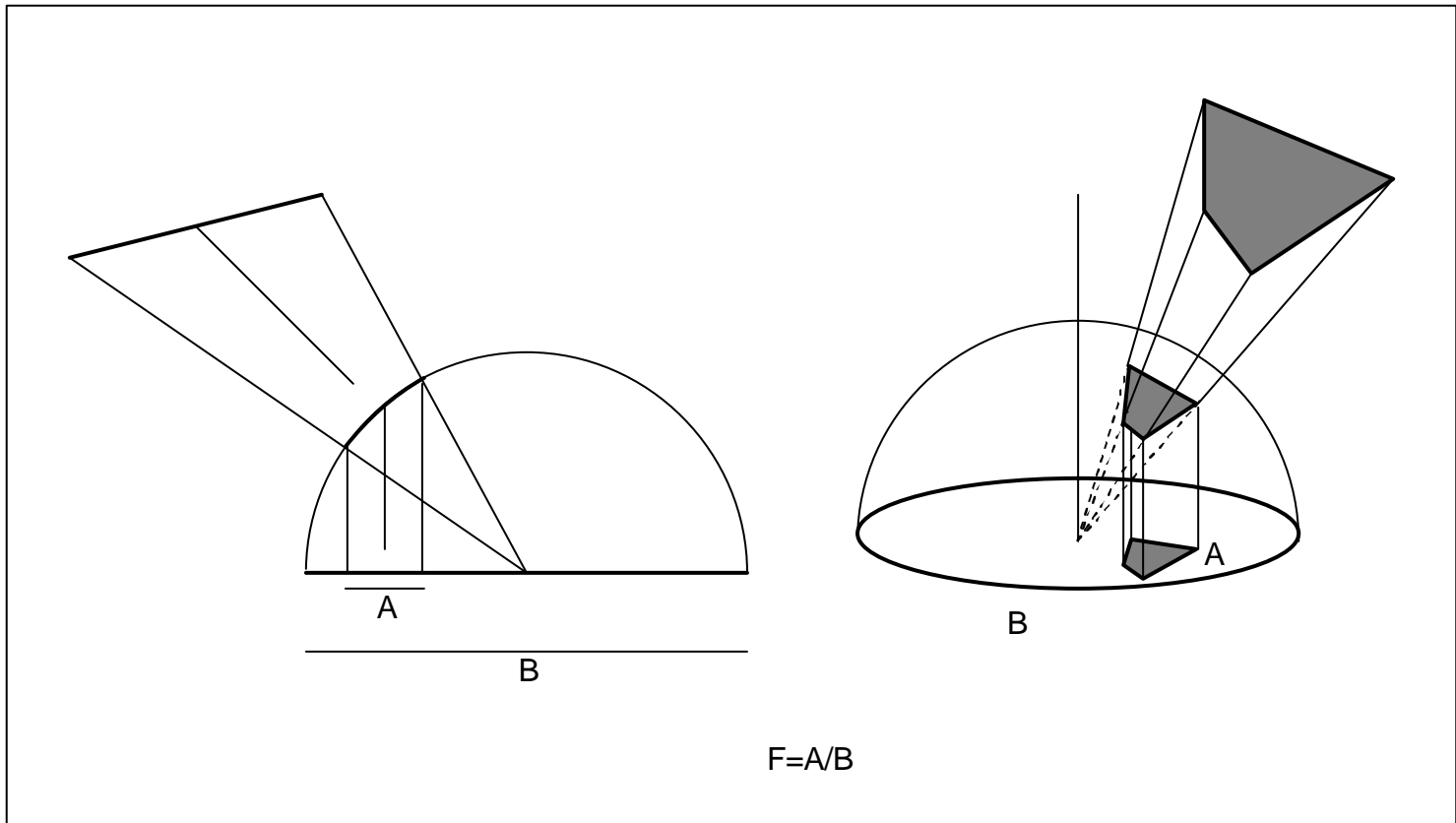
$$= \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{\mathbf{p} r^2} dA_j dA_i$$

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

$$\sum_{k=1}^n F_{ik} = 1 \quad \text{pour } i=1 \text{ à } n$$

pour une surface plane, $F_{ii} = 0$

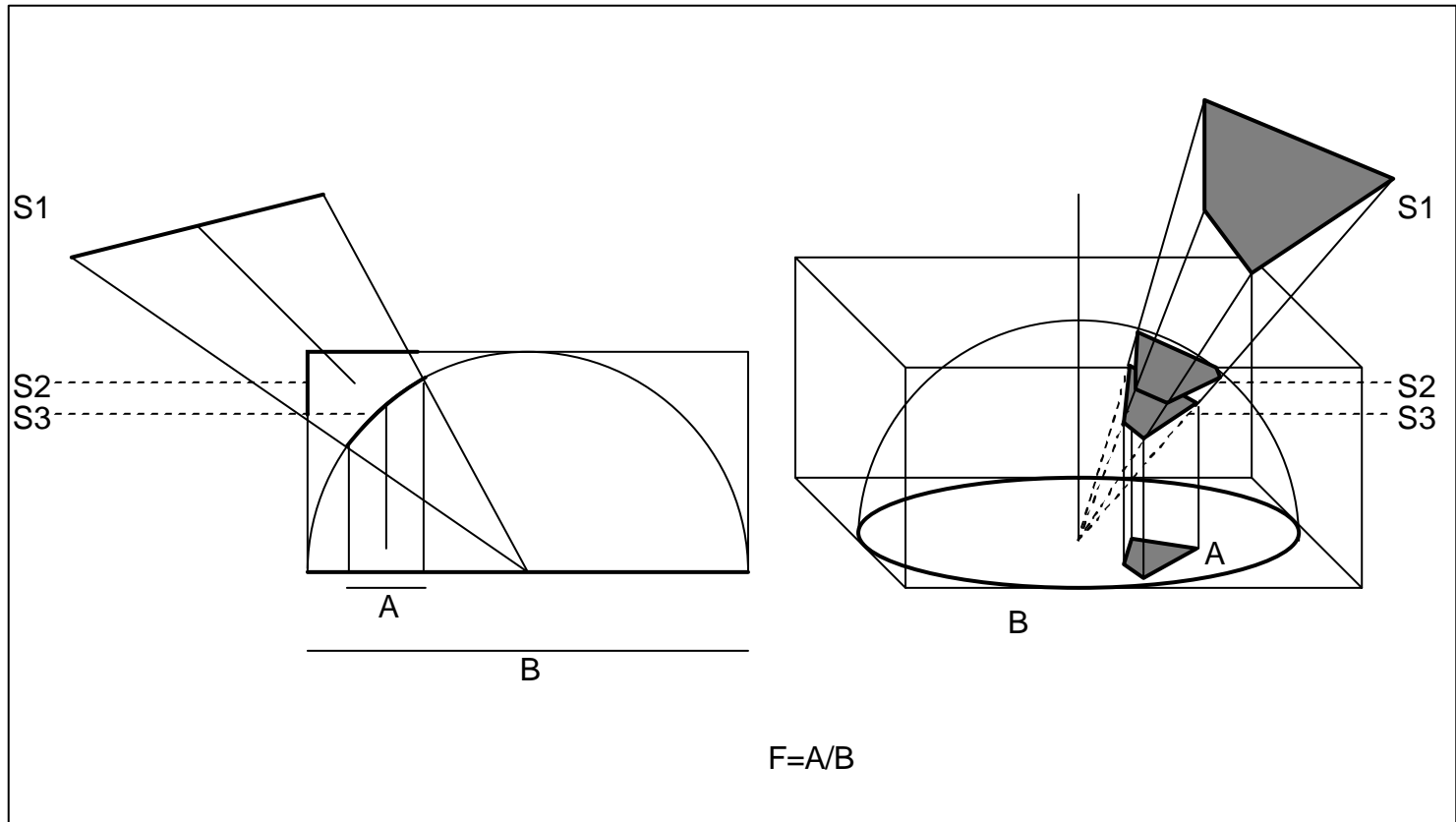
- méthode de Nusselt [SIEG84]



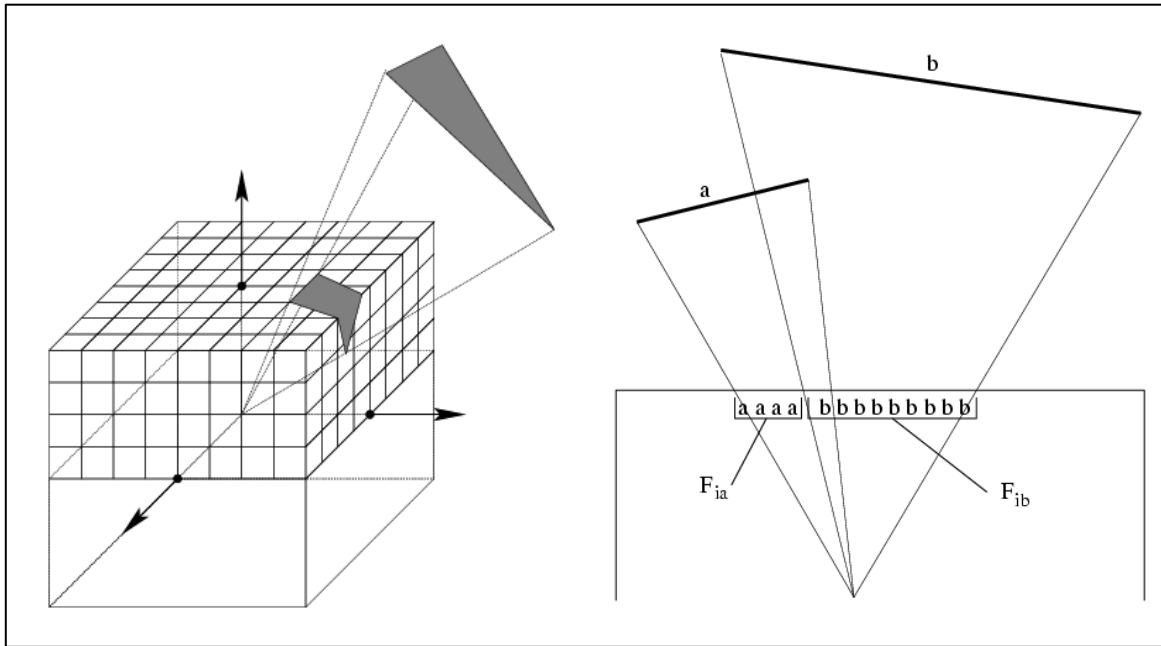
si la distance r est grande comparée à la surface du patche :

$$F_{ij} \approx \int_{A_j} \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{p r^2} dA_j$$

- méthode de l'hémicube



les surfaces S_1 , S_2 , S_3 ont le même facteur de forme



facteur de forme d'un pixel :

$$\Delta F_{dA_i A_j} = \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{p r^2} \Delta A = \Delta F_q$$

$$F_{ij} = \sum_q \Delta F_q$$

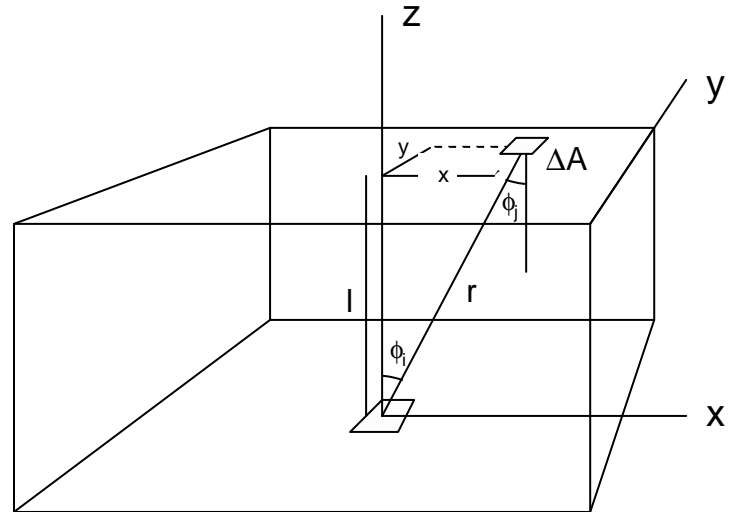
pré-calcul et stockage dans une table
positionnement de l'hémicube au centre de chaque patche

sur le plan $z = 1$

$$\left. \begin{aligned} \Delta F_q &= \frac{\cos \mathbf{j}_i \cos \mathbf{j}_j}{\mathbf{p} r^2} \Delta A \\ r &= \sqrt{x^2 + y^2 + 1} \\ \cos \mathbf{j}_i &= \cos \mathbf{j}_j = (x^2 + y^2 + 1)^{-\frac{1}{2}} \end{aligned} \right\} \Rightarrow \Delta F_q = \frac{1}{\mathbf{p} (x^2 + y^2 + 1)^2} \Delta A$$

sur les côtés de l'hémicube

$$\Delta F_q = \frac{z}{\mathbf{p} (x^2 + y^2 + 1)^2} \Delta A$$





Cornell University Program of Computer Graphics

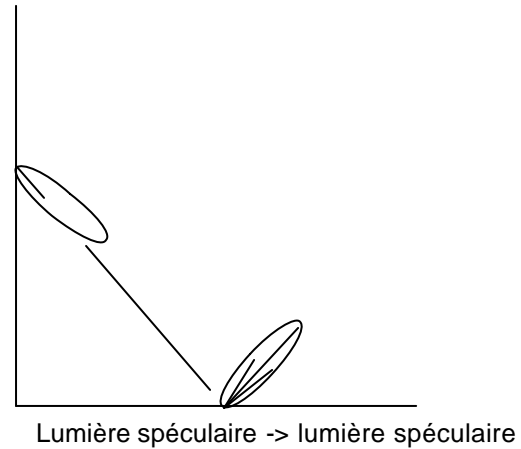
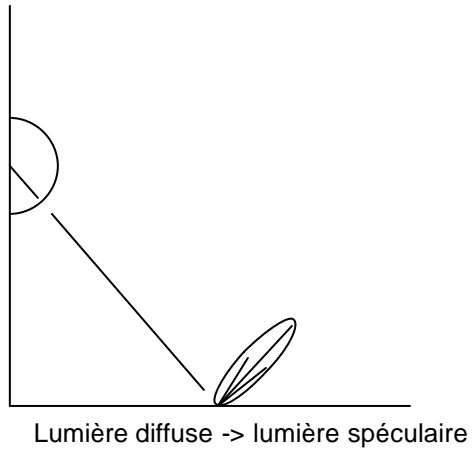
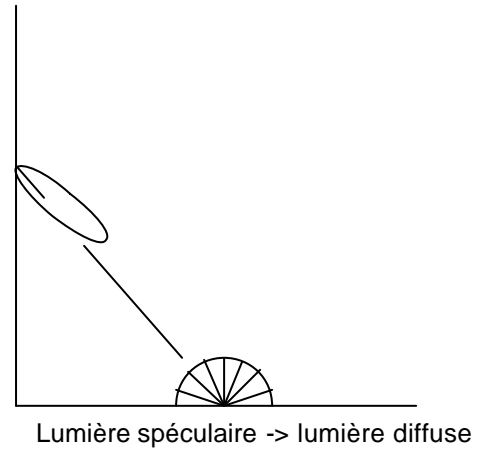
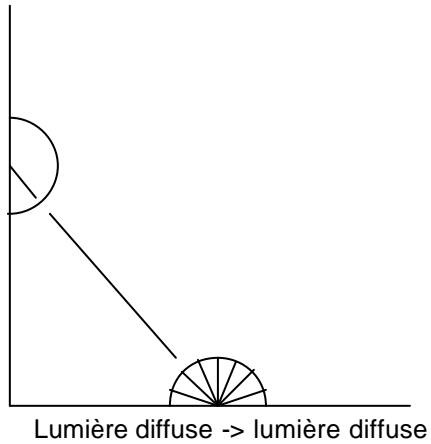
<http://www.graphics.cornell.edu/online/research/>

Modèle global d'illumination

Rushmeier, Torrance [RUSH90]

- 2 phases
 - 1^{ère} phase : calcul indépendant de la vue
 - 2^{ème} phase : calcul dépendant de la vue

- mécanismes de transport de la lumière :
Walace et al. [WALL87]



- 1^{ère} phase
mécanismes spéculaire-diffus, diffus-diffus

reformulation de l'équation de la radiosité :

$$I_i = I_{e,i} + \mathbf{r}_{d,i} \sum_{j=1}^N I_j F_{ij}$$

$$I_i = \frac{B_i}{p}$$

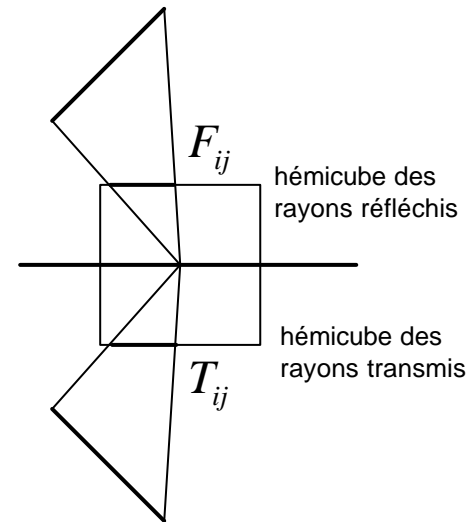
introduction de la transmission diffuse :

$$I_i = I_{e,i} + \mathbf{r}_{d,i} \sum_{j=1}^N I_j F_{ij} + \mathbf{t}_{d,i} \sum_{j=1}^N I_j T_{ij}$$

$$I_i = \frac{B_i}{p}$$

pour du verre :

$$T_{nj} = F_{mj} \text{ et } T_{mj} = F_{nj}$$

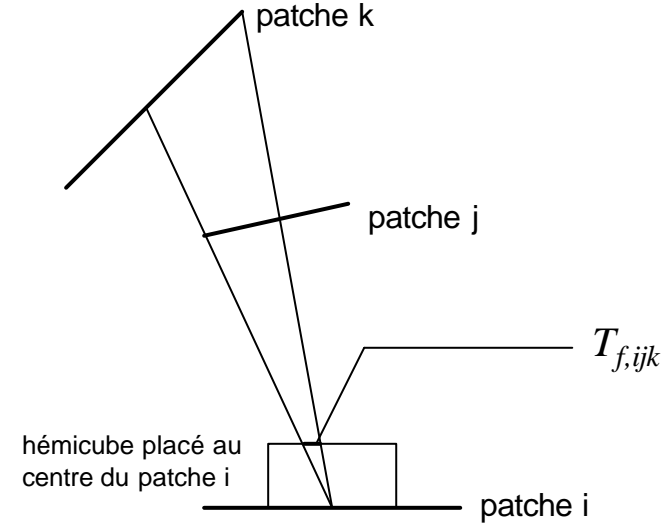


introduction de la transmission spéculaire:

$$I_j = I_{d,j} + I_{s,j}$$

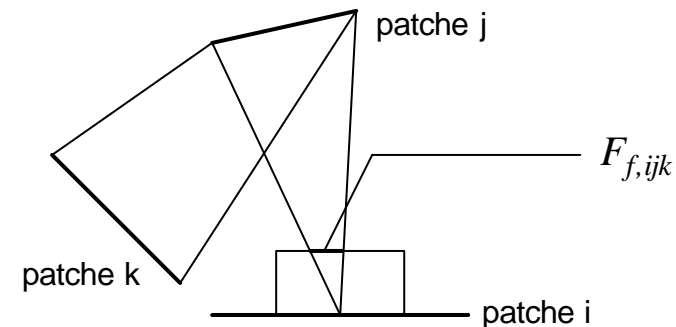
$$\text{or } I_{s,j} = \mathbf{t}_{s,j} \sum_{k=1}^N I_k T_{f,ijk}$$

$$I_i = I_{e,i} + \mathbf{r}_{d,i} \sum_{j=1}^N \left\{ I_j F_{ij} + \mathbf{t}_{s,j} \sum_{k=1}^N I_k T_{f,ijk} \right\} \\ + \mathbf{t}_{d,i} \sum_{j=1}^N \left\{ I_j T_{ij} + \mathbf{t}_{s,j} \sum_{k=1}^N I_k T_{b,ijk} \right\}$$



introduction de la réflexion spéculaire:

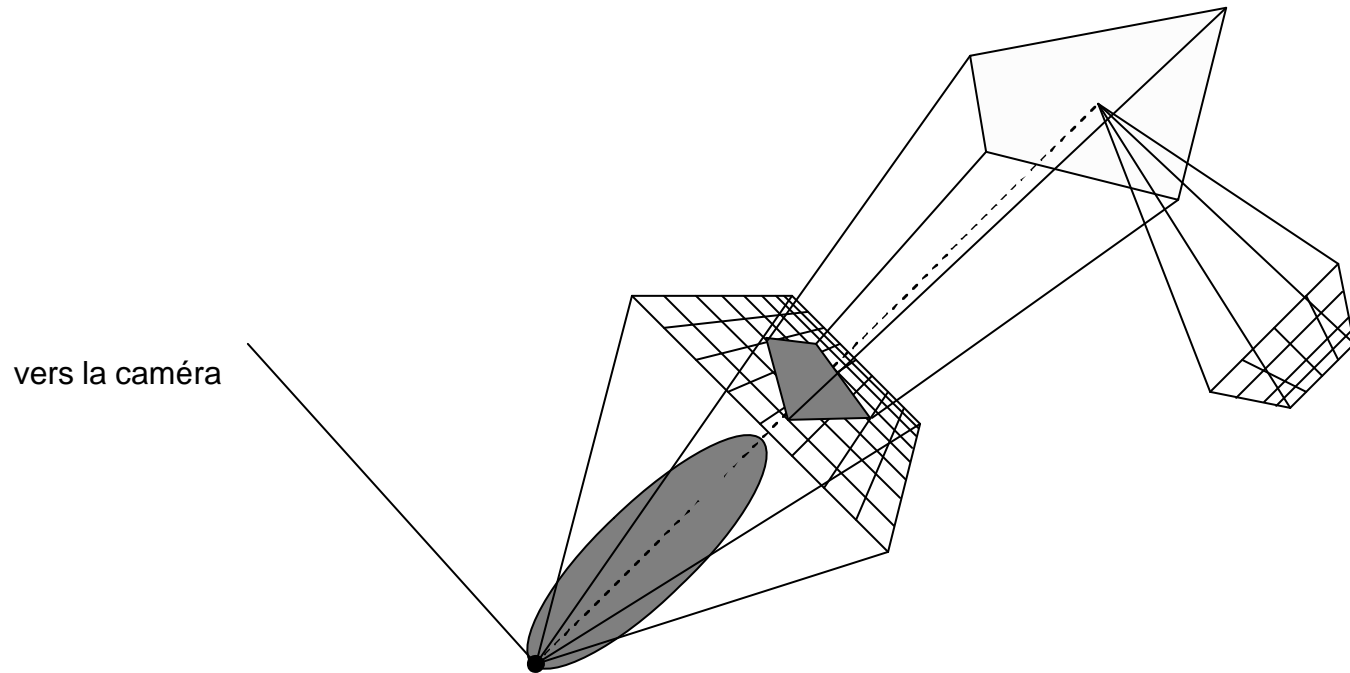
$$I_i = I_{e,i} + \mathbf{r}_{d,i} \sum_{j=1}^N \left\{ I_j F_{ij} + \mathbf{t}_{s,j} \sum_{k=1}^N I_k T_{f,ijk} + \mathbf{r}_{s,j} \sum_{k=1}^N I_k F_{f,ijk} \right\} \\ + \mathbf{t}_{d,i} \sum_{j=1}^N \left\{ I_j T_{ij} + \mathbf{t}_{s,j} \sum_{k=1}^N I_k T_{b,ijk} + \mathbf{r}_{s,j} \sum_{k=1}^N I_k F_{b,ijk} \right\}$$



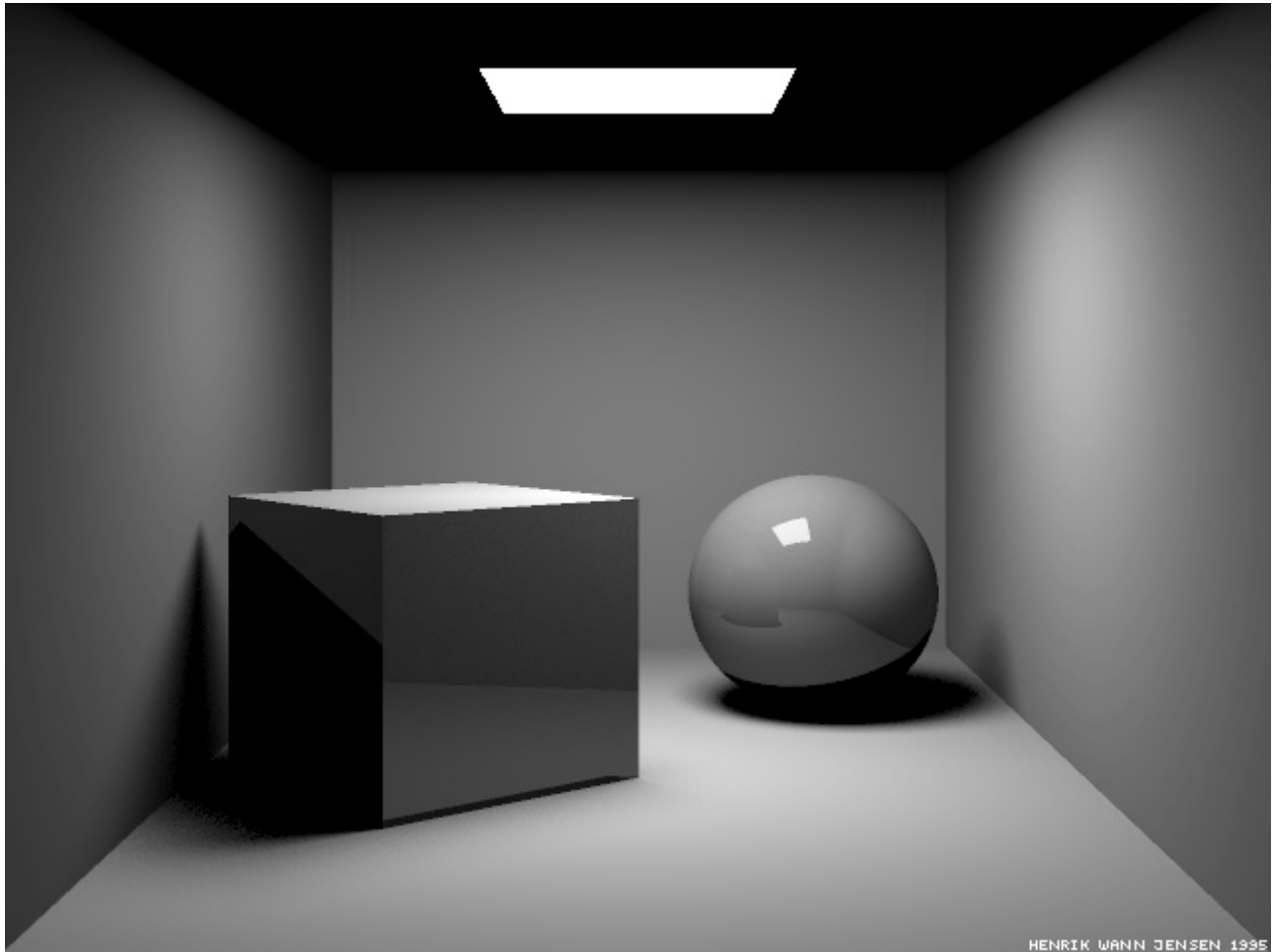
- 2^{ème} phase
mécanisme spéculaire-spéculaire, diffus-spéculaire

hypothèse :
une faible partie des rayons contribue à la bosse spéculaire

- mécanisme spéculaire-spéculaire :

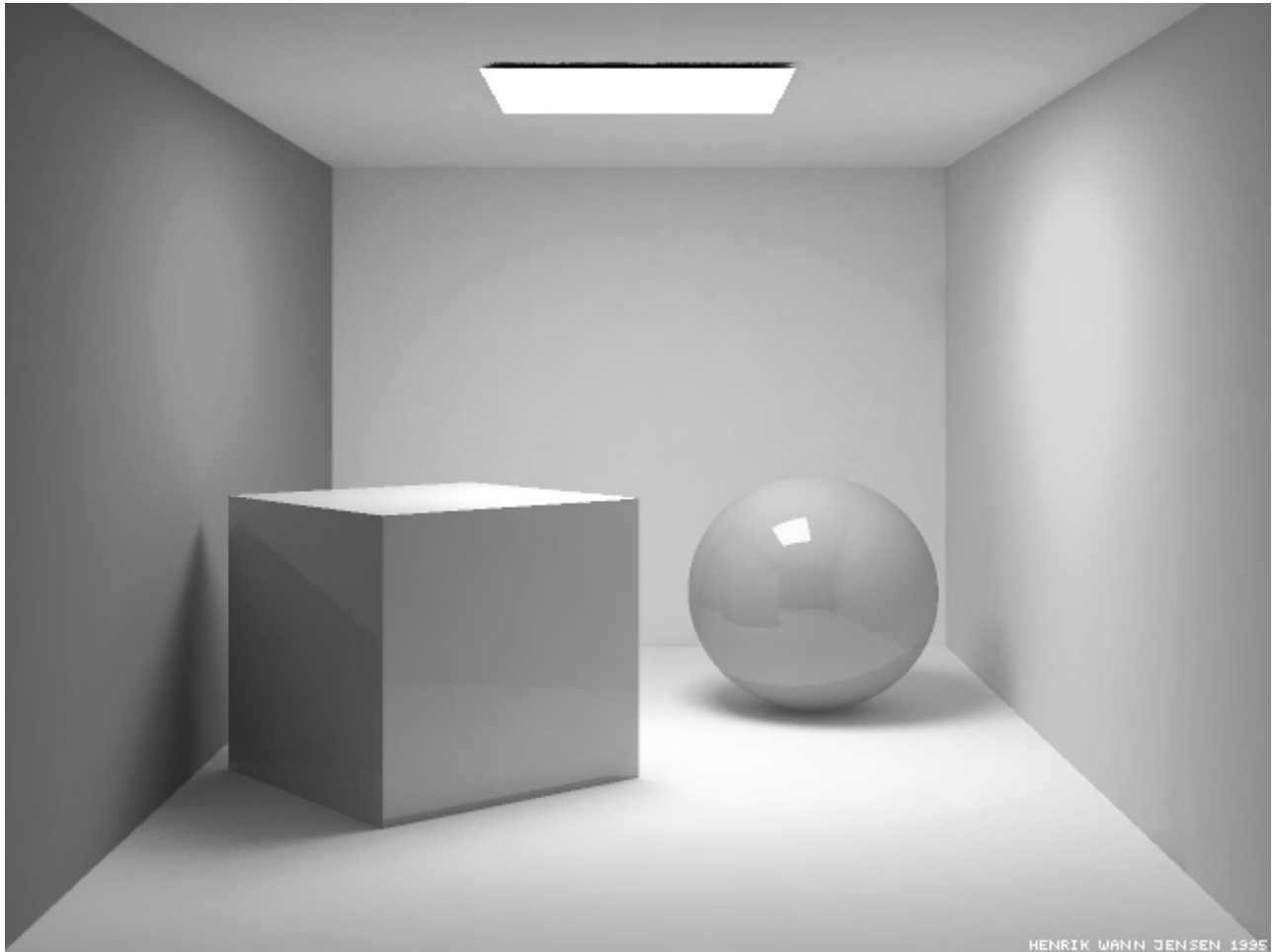


- mécanisme diffus-spéculaire :
interpolation linéaire des intensités calculées dans la première phase



Lancé de rayon

<http://www.gk.dtu.dk/home/hwj/pictures/raytrace.html>



Modèle global d'illumination
<http://www.gk.dtu.dk/home/hwj/pictures/raytrace.html>

