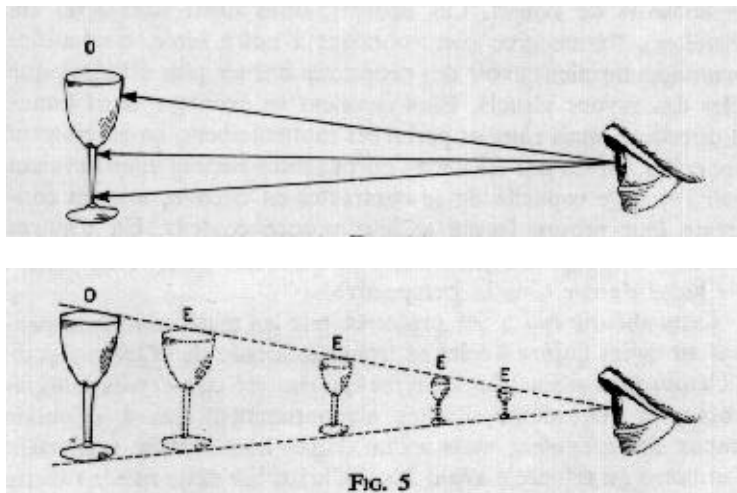


Chapitre 9

Le suivi de rayon

... ou "La revanche d'Euclide" ...



[RONCHI] pp. 23 et 24

1. Algorithme de base

A. Appel, Spring Joint Computer Conf. 1968

- Rappel : l'algorithme du z-buffer

```
pour chaque face faire
  pour chaque pixel (i,j) de la face projetée faire
    pz = valeur de Z pour la face en (i,j)
    si pz >= Zbuffer[i,j] alors
      Zbuffer[i,j] = pz
      pixel (i,j) = couleur de la face
    fin si
  fin pour
fin pour
```

- Une autre manière de traiter le problème :

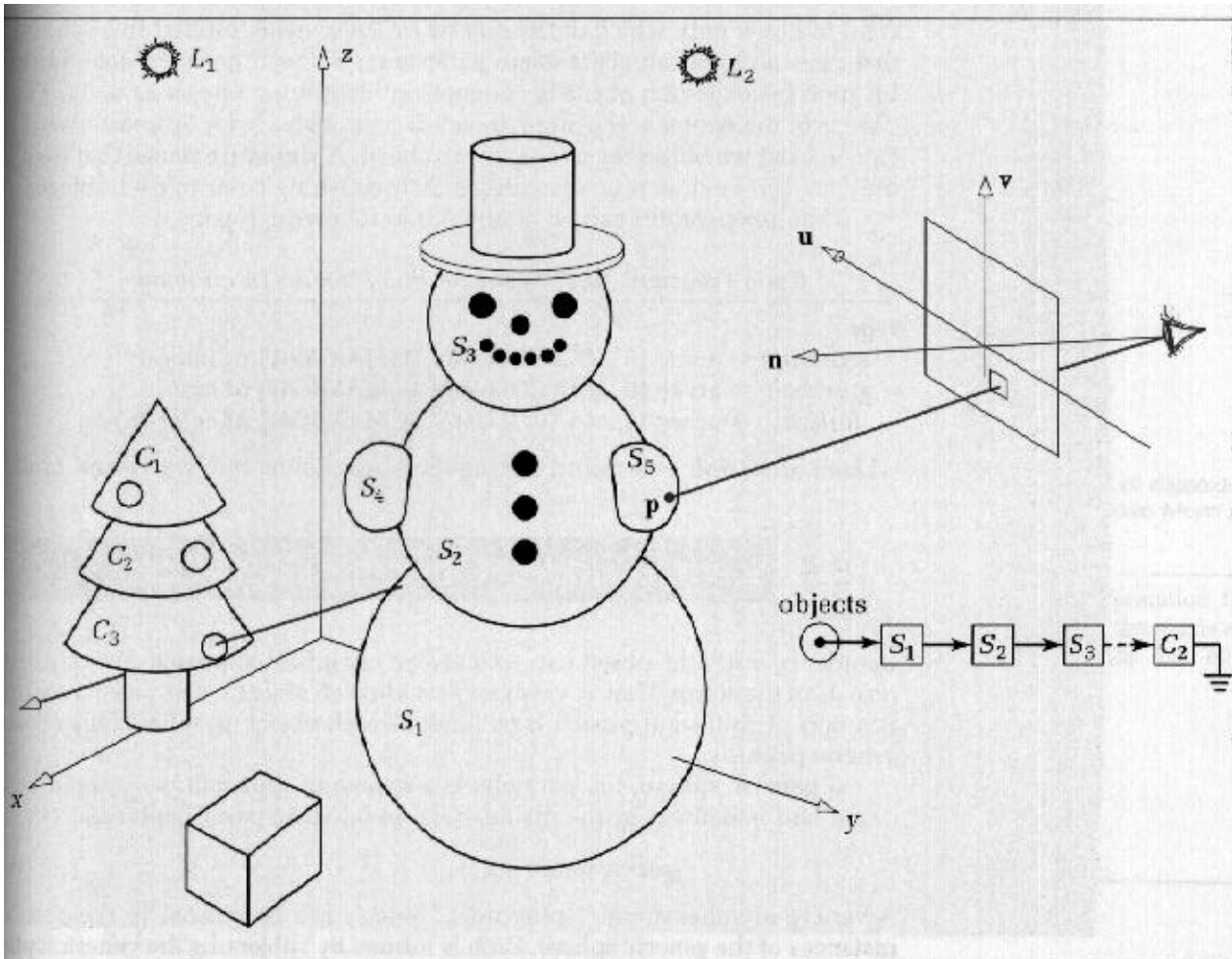
```
pour chaque pixel (i,j) de l'écran faire
  maxlocal := -oo
  pour chaque face des polyèdres faire
    pz = valeur de Z pour la face en (i,j)
    si pz existe et pz >= maxlocal alors
      maxlocal = pz
      pixel (i,j) = couleur de la face
    fin si
  fin pour
fin pour
```

Le Z-buffer devient inutile !

Mais coût *a priori* plus élevé...

On se ramène à des calculs d'intersection entre un "rayon lumineux" et les volumes de la scène

Calcul d'intersection



[HILL] p. 619

- Equation de la droite oeil - pixel

(x_0, y_0, z_0) : coordonnées de l'oeil

(x_1, y_1, z_1) : coordonnées 3D du centre du pixel (dépend du type de projection utilisée)

Représentation paramétrique :

$$\begin{aligned}
 x &= x_0 + t(x_1 - x_0) = x_0 + t \quad x \\
 y &= y_0 + t(y_1 - y_0) = y_0 + t \quad y \quad 0 \quad t \quad 1 \\
 z &= z_0 + t(z_1 - z_0) = z_0 + t \quad z
 \end{aligned}$$

- **Exemple : intersection avec une sphère**

Equation de la sphère de coord (a, b, c) et de rayon R :

$$(x - a)^2 + (y - b)^2 + (z - c)^2 = R^2$$

Si une intersection existe, ses coordonnées vérifient les 2 équations (droite et sphère). On arrive à :

$$At^2 + Bt + C = 0 \quad \text{avec:}$$

$$A = \frac{2}{x} + \frac{2}{y} + \frac{2}{z}$$

$$B = 2 \left(\frac{1}{x}(x_0 - a) + \frac{1}{y}(y_0 - b) + \frac{1}{z}(z_0 - c) \right)$$

$$C = (x_0 - a)^2 + (y_0 - b)^2 + (z_0 - c)^2 - R^2$$

Si pas de solution : pas d'intersection

Si une seule : point tangent à la sphère

Sinon : les deux points sont trouvés (t mini est le 1er)

- **Cas d'un polygone**

Equation du plan contenant le polygone :

$$x.n_x + y.n_y + z.n_z = \text{constante} = d$$

Condition d'intersection :

$$N = d - (n_x x_0 + n_y y_0 + n_z z_0)$$

$$D = n_x \frac{1}{x} + n_y \frac{1}{y} + n_z \frac{1}{z}$$

Si D=0 et N>0 : rayon inclus dans le plan

Si D=0 et N<0 : rayon parallèle au plan

Si N ≠ 0 Point d'intersection : t = N/D

Il faut ensuite tester si le point d'intersection est à l'intérieur du polygone convexe :

- méthode des angles $\theta_i = 2\pi$ (coûteux)
 i

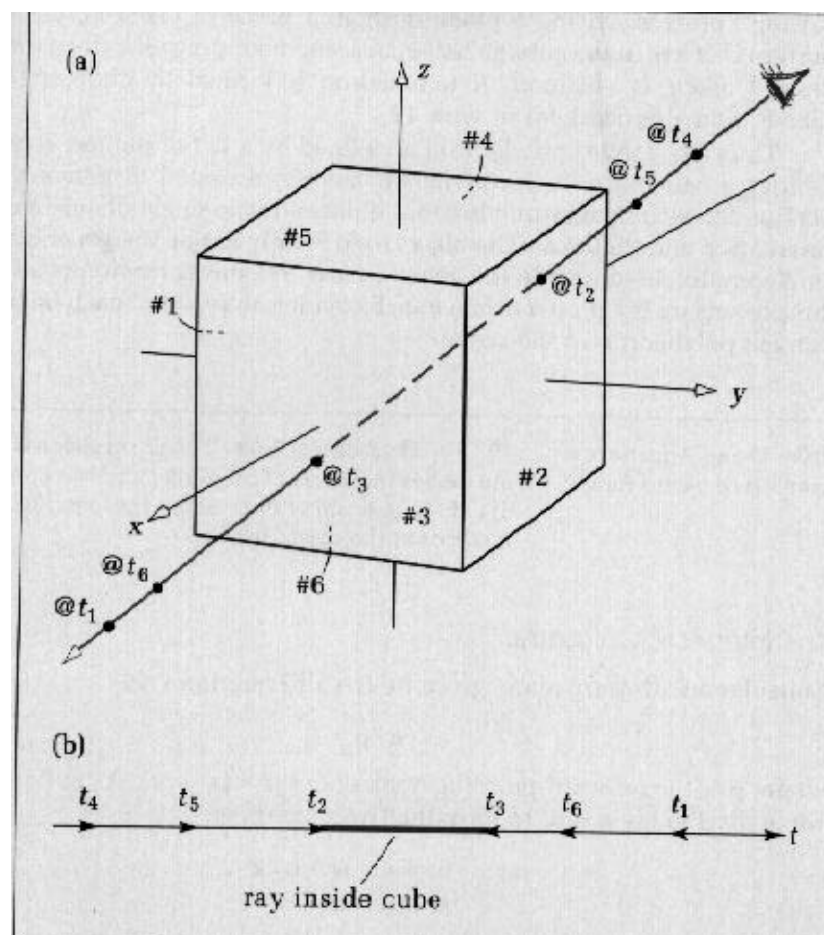
- méthode du balayage : si intérieur : nombre pair d'intersections

• Cas d'un polyèdre convexe

On applique le calcul précédent à chacune des faces

Si $D < 0$, le rayon "arrive" sur la face

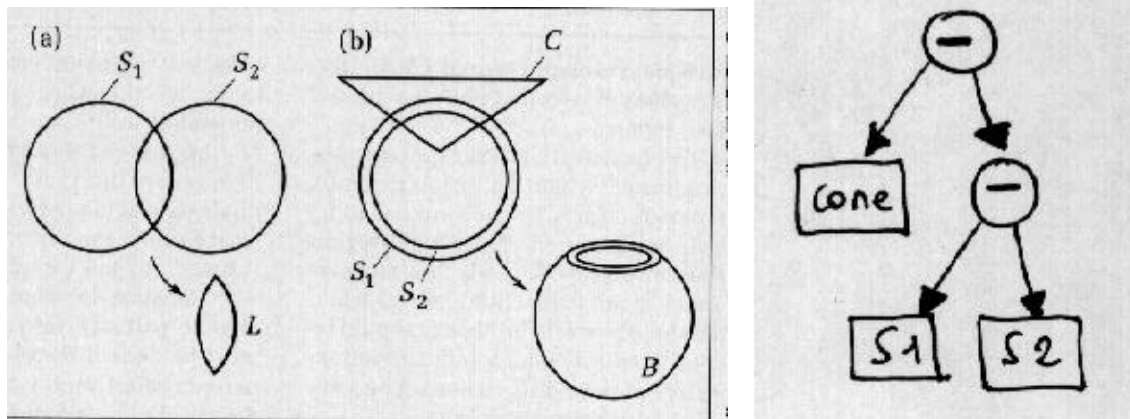
Si $D > 0$, le rayon "sort" de la face



A chaque étape : si $D > 0$ alors $t_{out} = \min(t_{out}, t)$
sinon $t_{in} = \max(t_{in}, t)$

Solides définis par opérateurs booléens (CGS)

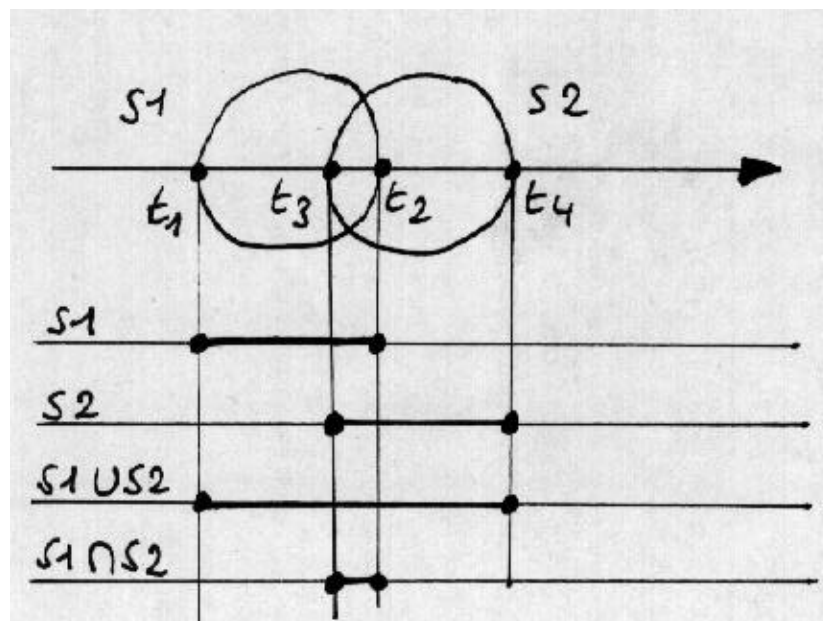
- Rappels chap. 4 :



[HILL] p. 662

- Algorithme de GoldStein et Nagel (1971)

On se ramène à un problème en 1D :

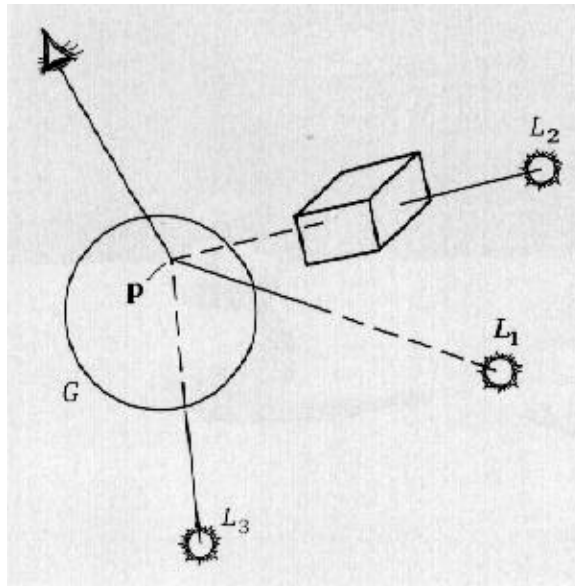


Idem opération de différence

Algorithme (récuratif) de parcourt d'arbre binaire

Ombre et coloriage

- On "tire" un second rayon du point d'intersection vers la (les) source(s) lumineuse(s) (supposée ponctuelle)

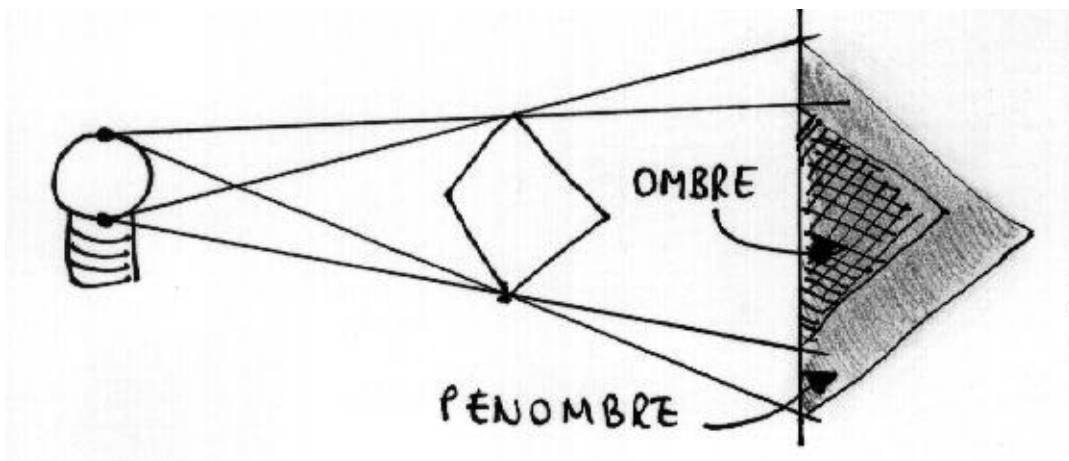


[HILL] p. 648

On applique au point P un des modèles de coloriage vu au chap. 7 (Phong, ...), avec ou sans textures.

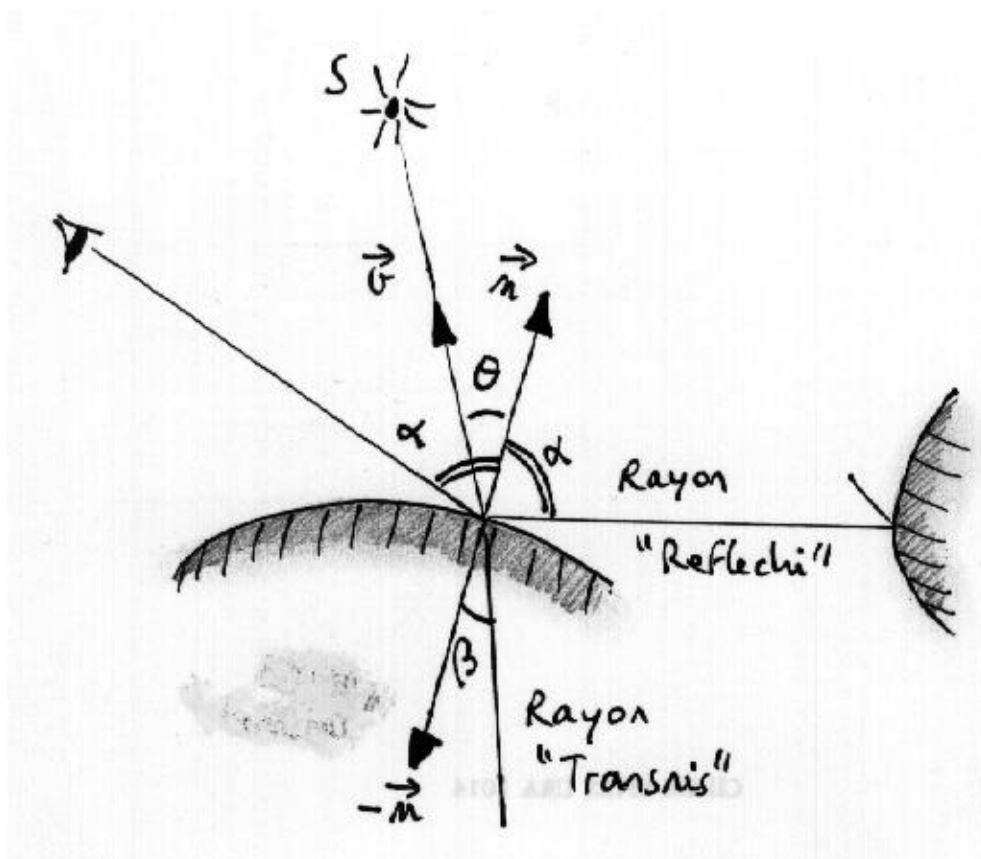
Pour les points dans l'ombre, seule l'intensité de la lumière ambiante est prise en compte.

- Source non ponctuelle : la pénombre



2. Algorithme récursif

- Whitted (1980) : prise en compte des autres interactions entre objets de la scène :
 - Réfraction dans les objets transparents
 - Réflexion de la lumière d'un objet à l'autre (miroirs)



Les intensités s'ajoutent à celles dues à la source S :

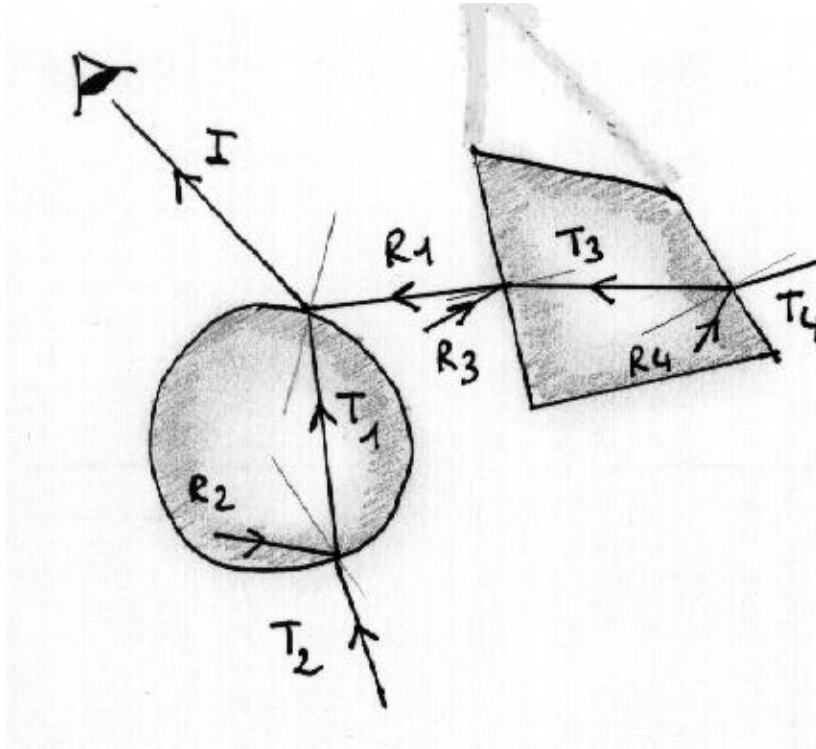
$$I_{total} = I_{amb} + I_{diff} + I_{spec} + I_{refl} + I_{refr}$$

- Direction des rayons transmis :

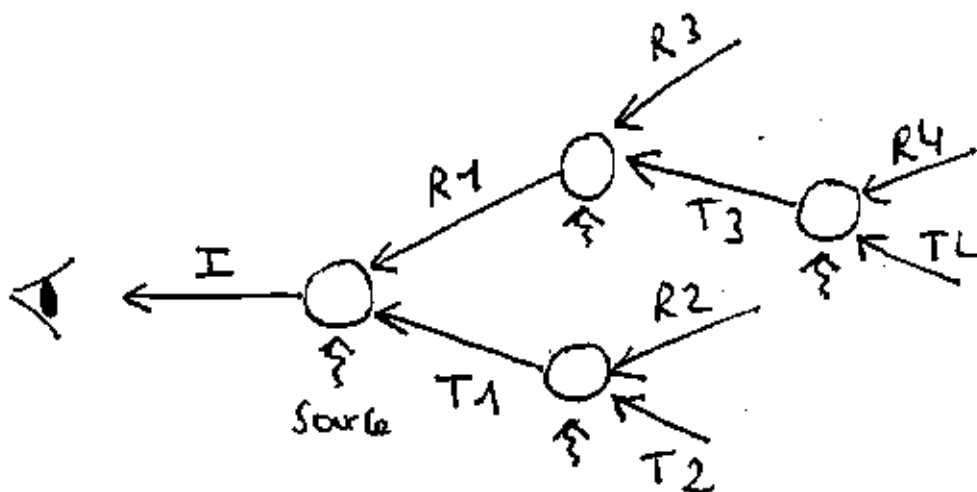
$$\text{loi de Descartes : } \frac{\sin \alpha}{c_{ext}} = \frac{\sin \beta}{c_{obj}} \quad \sin \beta = n_{obj} \sin \alpha$$

eau : $n = 1,3$ verre : n de $1,5$ à $1,9$ diamant : $n = 2,4$

- **Arbre des rayons :**



Le trajet des rayons définit un arbre, potentiellement infini :

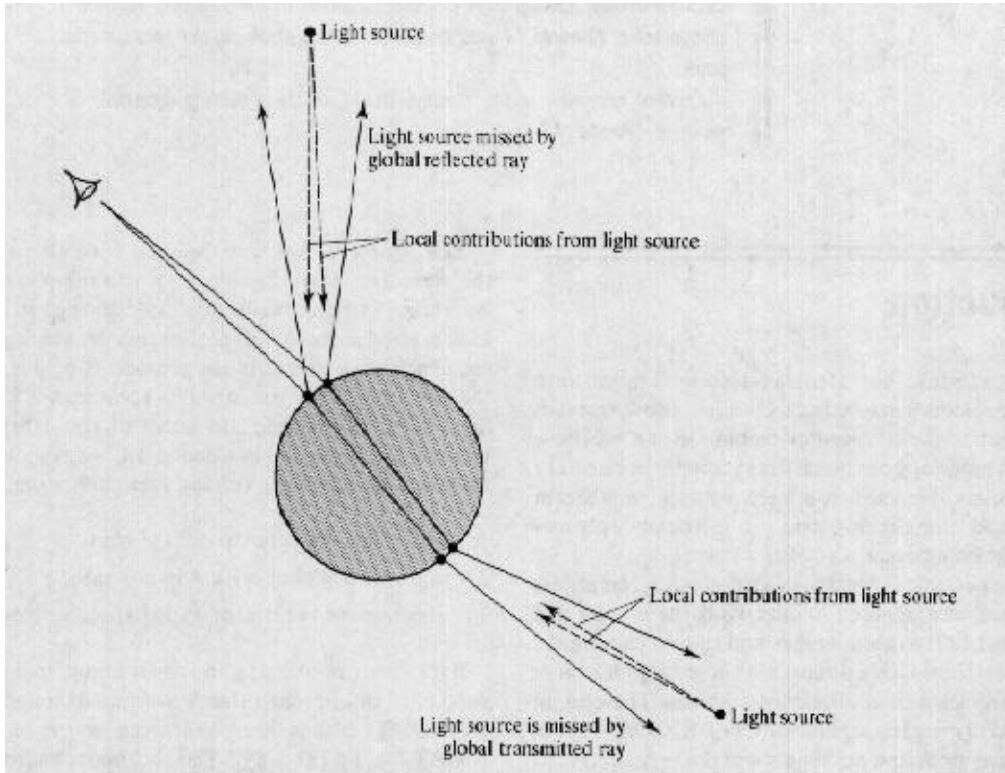


Dans le modèle de Whitted, les intensités R et T sont inversement proportionnelles aux longueurs des rayons.

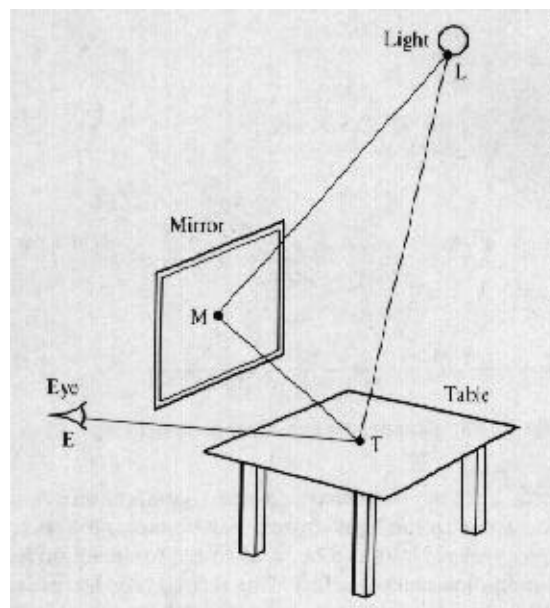
=> On s'arrête à un niveau fixé (2, 3,...)

- **Limites de l'approche :**

- La lumière reçue directement au point d'intersection n'est prise en compte que par un modèle empirique (Phong, par ex.)



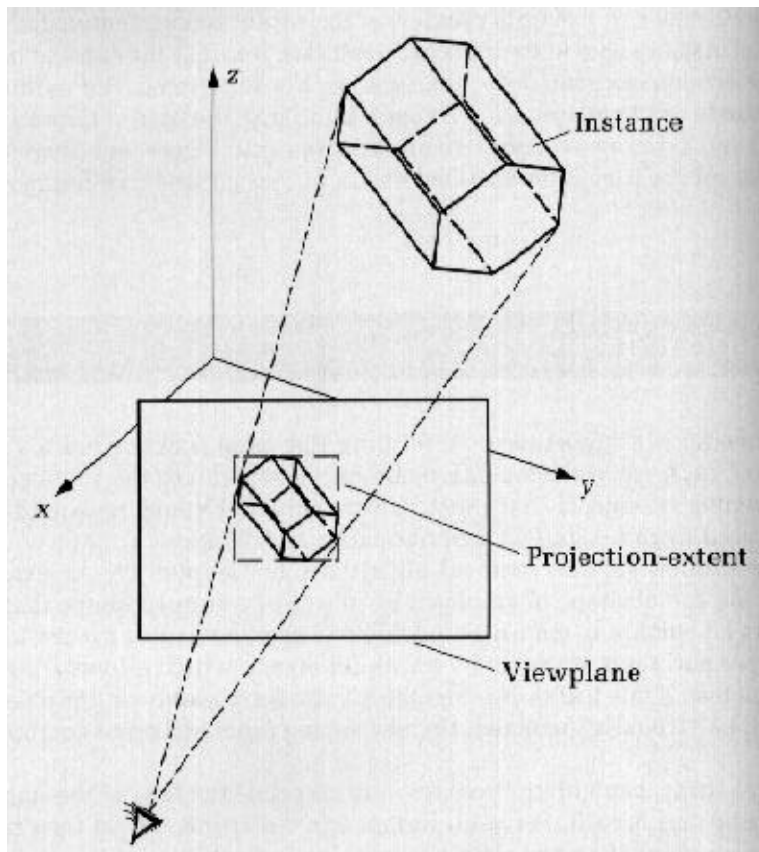
- Des rayons importants sont "ratés" : cas d'un miroir



3. Optimisations

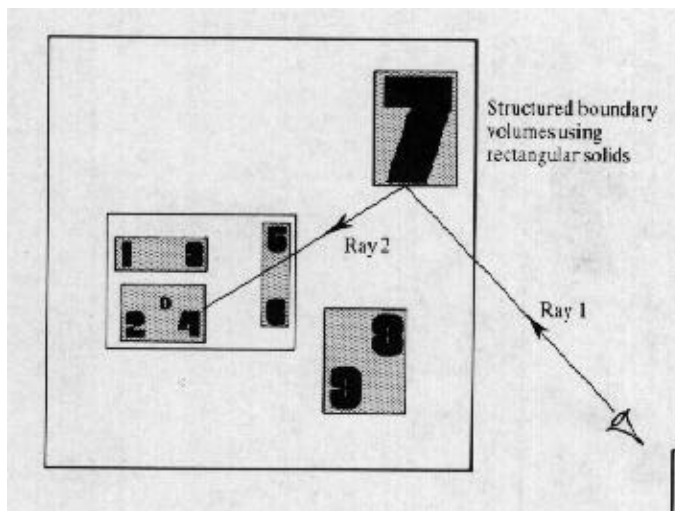
Problème : calculs d'intersection rayon/objets >75% temps de calcul total

- Utilisation des extents pour le rayon initial



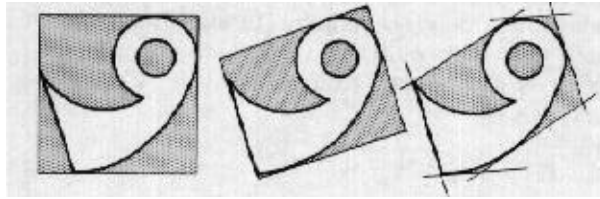
[HILL] p. 640

- Utilisation des volumes englobants



[WATT] p. 236

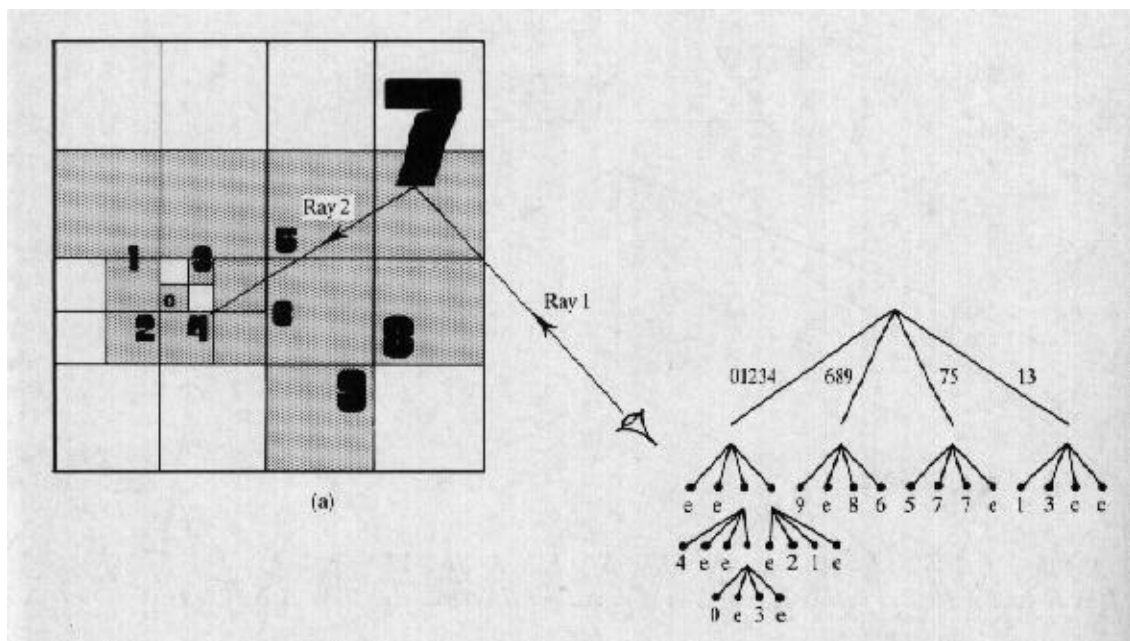
Plusieurs techniques de volumes englobants : Boîte, Sphère, "Slabs"



[WATT] p. 237

- **Partitionnement de l'espace 3D avec la technique des "octrees"**

Quand un rayon entre dans une portion de l'arbre, seuls les objets qui en font partie sont concernés par les calculs d'intersection



[WATT] p. 244

- **Nombreuses autres techniques (SEADS, BSP) : voir [WATT] chap. 9 pour une comparaison**