

# RENDU

Processus consistant à créer des scènes graphiques réalistes à partir de données géométriques et intégrant certains phénomènes

- lumière réfléchie diffuse (objets mats) et spéculaire (miroirs)
- lumière réfractée diffuse (vitres brouillées) et spéculaire (vitres courantes)
- couleurs (absorption des longueurs d'onde de la lumière blanche)
- ombres et pénombres
- texture

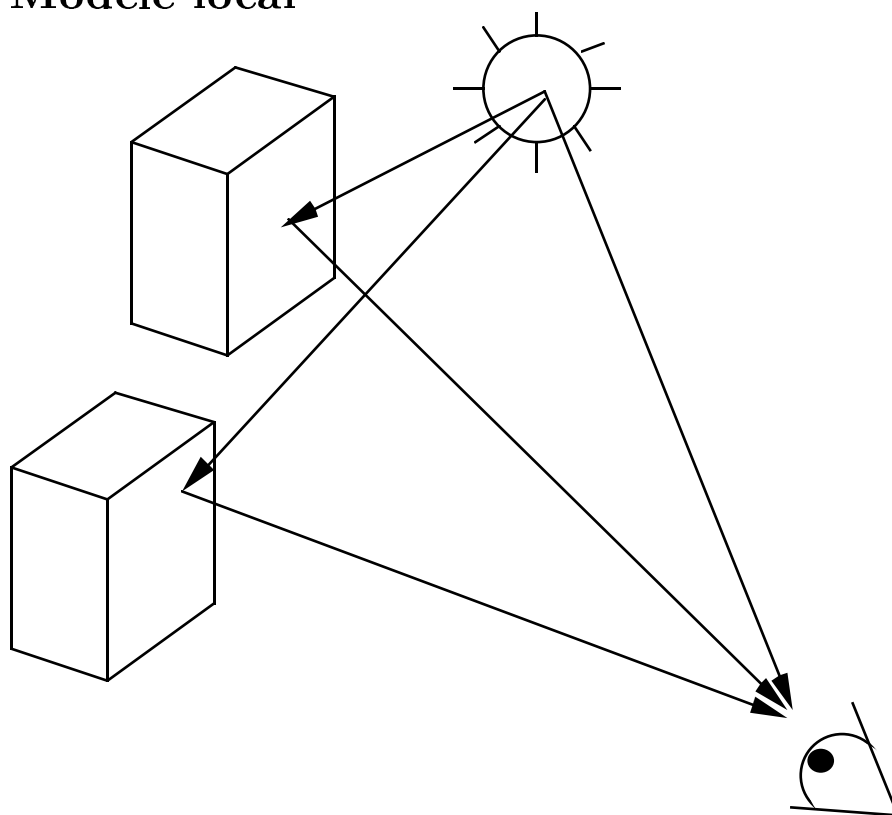
# LES ACTEURS

- une ou plusieurs sources lumineuses qui émettent de la lumière (de l'énergie)
  - sources *ponctuelles*
  - sources *uniformément distribuées* (à l'infini)
  - sources plus générales (modèle de Warn)
- des *objets* dans une scène qui transmettent à leur façon l'énergie reçue des sources et des autres objets. Ils sont définis par une méthode de modélisation
- un *observateur* qui reçoit la lumière transmises par les objets (il les voit).

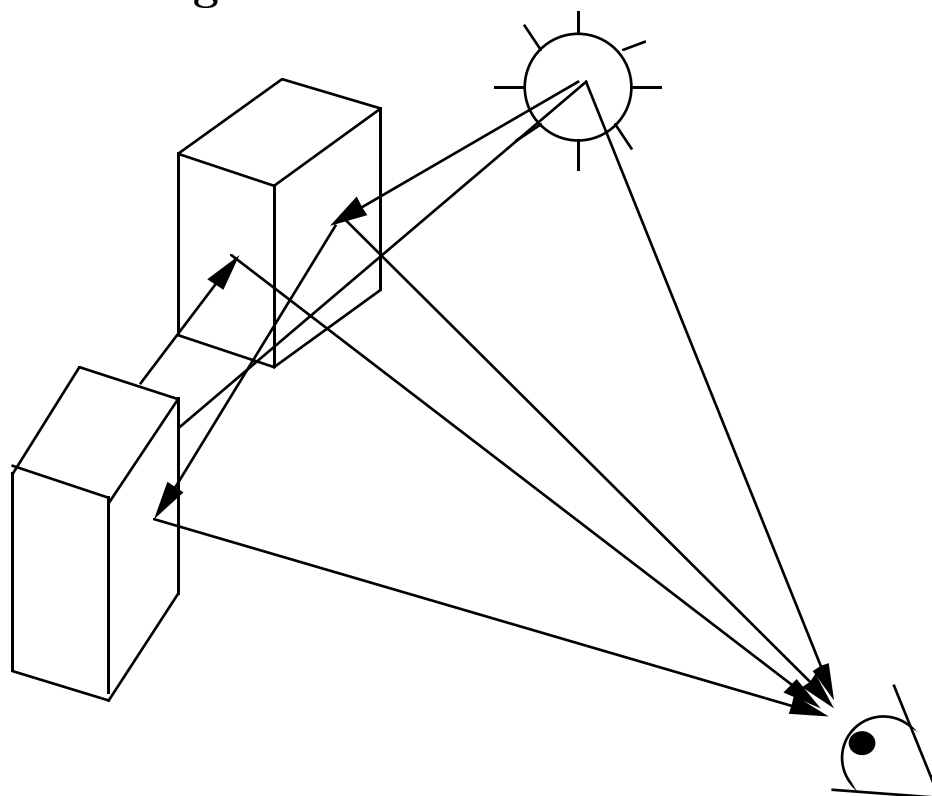
## LES 2 CLASSES DE MÉTHODES DE RENDU

- méthodes locales: le rendu ne tient compte que de l'interaction d'un objet avec les sources
- méthodes globales: le rendu tient compte aussi de l'éclairage indirect (lumière renvoyée par les autres objets)
  - *Lancé de rayon*: on retrace l'histoire du rayon à partir de l'observateur vers les source. Si l'observateur change de place il faut tout recalculer.
  - *Radiosité*: on part des sources et l'on suit le chemin des rayons jusqu'à trouver l'équilibre. Ici la solution ne dépend pas de la position de l'observateur dans la scène.

## Modèle local



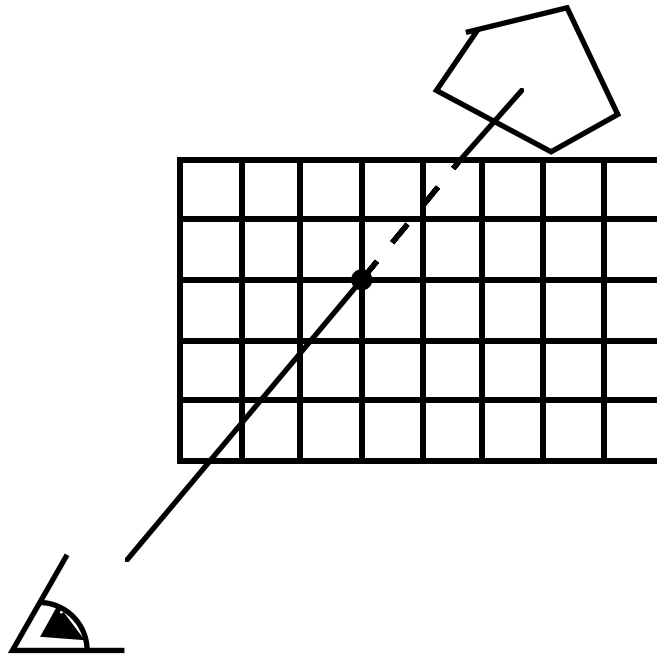
## Modèle global



# LANCÉ DE RAYON

Historiquement deux versions (*ray tracing*)

- Appel (1968): un algorithme d'élimination des faces cachées semblable au tampon de profondeur

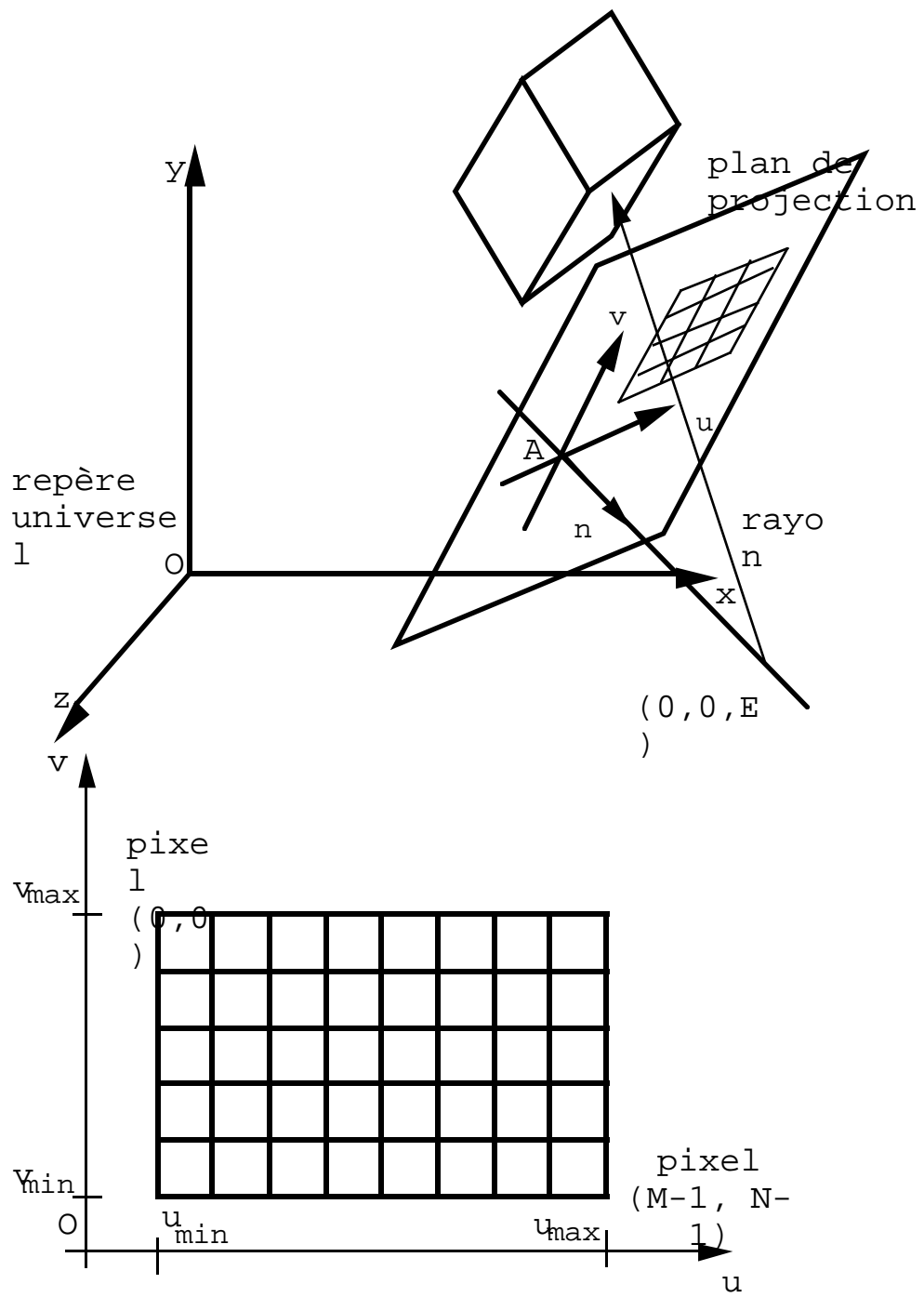


- Whitted (1980): un algorithme qui intègre des effets optiques plus avancés: réflexion, réfraction, ombrage, sources multiples etc...

## HYPOTHÈSES DE LA MÉTHODE

- $O, x, y, z$  le repère universel de l'espace et  $A, u, v, n$  le repère local de la vue
- l'observateur se trouve au centre de projection de coordonnées  $(0, 0, E)$  du repère local
- dans le plan  $u, v$  de la vue, les coordonnées du bord inférieur gauche et supérieur droit de la fenêtre sont  $(u_{min}, v_{min})$  et  $(u_{max}, v_{max})$
- l'écran possède  $M$  colonnes et  $N$  lignes. On identifie le pixel en position  $(i, j)$  de l'écran avec le point de coordonnées

$$(u_{min} + i \frac{u_{max} - u_{min}}{M - 1}, v_{min} + j \frac{v_{max} - v_{min}}{N - 1}) \quad (1)$$





## PRINCIPE DE LA MÉTHODE

- Suivre en sens inverse le rayon dont le dernier parcours joint un pixel donné et l'œil de l'observateur.
- Soit *Objet* le premier objet de la scène rencontré par ce rayon *primaire*  $\vec{S}$  et P son point d'intersection avec *Objet*
- L'énergie émise par P a (récursivement) 3 origines
  1. Diffusion de la lumière émise par chaque source lumineuse ne rencontrant pas d'obstacle entre la source et P. S'il n'existe pas de telle source, l'objet est dans l'ombre (on se restreint alors la lumière ambiante)
  2. Réflexion spectrale de la lumière provenant d'un autre objet suivant la direction symétrique au rayon lancé par rapport à la normale
  3. Réfraction de la lumière (si l'objet est translucide) provenant d'un autre objet

suivant la direction dont le rayon lancé  
est le rayon réfracté

# CONSIDÉRATIONS DE COMPLEXITÉ

## Avantages de la méthode

- Conception simple: tout se ramène à des calculs d'intersections avec des objets simples
- Proche du modèle physique
- Rend bien compte des phénomènes de réflexion spéculaire et de réfraction sans dispersion

## Inconvénients

- Les calculs d'intersection sont extrêmement coûteux (environ 90% du temps de calcul, le reste étant consacré aux calculs optiques)
- Dans le cas d'une seule source et de  $n$  objets:  $3n$  tests d'intersection si l'on ne tient compte que des rayons primaires réfléchis et réfractés,  $3n+3^2n = 12n$  tests d'intersection si l'on tient compte des rayons primaires et

secondaires,  $3n+3^2n+3^3 = 39n$  tests d'intersection si l'on ne tient compte que des rayons primaires, secondaires et tertiaires, ...

- Tout doit être recalculé si l'observateur change de place
- Utilise une notion vague d'illumination d'ambiance
- Ne peut éviter les phénomènes de l'aliassage propres aux algorithmes travaillant dans l'espace image.

## L'ALGORITHME

pour *tout pixel* faire début  
    *calculer la direction  $\vec{S}$  de l'œil au pixel*  
    *afficher le pixel avec l'intensité*  
        LancerRayon(0,  $\vec{S}$ , profondeur)  
fin;

### LancerRayon

- La fonction (récursive) est appelée avec les paramètres
  - O: position de l'observateur
  - $\vec{S}$ : vecteur unitaire qui donne la direction de l'observateur vers le pixel courant.
  - profondeur: nombre de “rebonds” choisie a priori en fonction du volume de calcul estimé
- La valeur de retour de la fonction est une intensité de gris ou une valeur pour chacune des 3 couleurs de base RGB.

```

fonction LancerRayon(O:Point;  $\vec{S}$ : vecteur;
                    profondeur:integer):real;
var I real;
début
    déterminer le point P, intersection de la demi-droite
    (O, $\vec{S}$ ) avec l'objet le plus proche de la scène pour
    l'observateur, soit Obj;
    si P n'existe pas alors retourner IntensiteDeFond;
    {ici, P existe}
    I:=IntensiteSources(P,  $\vec{S}$ , Obj);
    si profondeur= 0 alors retourner I
    sinon début
        calculer le rayon réfléchi  $\vec{R}$  et le
        rayon réfracté  $\vec{T}$ ;
        I:=I +  $k_s$ * LancerRayon(P, $\vec{R}$ ,profondeur-1);
        I:=I +  $k_t$ * LancerRayon(P, $\vec{T}$ ,profondeur-1);
        retourner I;
    fin
fin

```

## IntensiteSources

- La fonction prend en entrée les paramètres
  - P: le point d'impact du rayon
  - $\vec{S}$ : le vecteur unitaire qui donne la direction du rayon
  - obj: l'objet intersecté auquel appartient le point P
- La valeur de retour est une intensité de gris ou une valeur pour chacune des 3 couleurs de base RGB.

fonction IntensiteSources(P:Point;  $\vec{S}$ : vecteur;  
obj:Objet):real;

début

*déterminer la normale  $\vec{n}$  en P à Obj;*

$I := I_a * k_a$ ; {intensité ambiante}

*pour chaque source lumineuse  $\mathcal{L}_i$  faire*

*si  $\mathcal{L}_i$  est visible de P alors faire début*

*calculer le vecteur unitaire  $\vec{L}_i$  de P vers  $\mathcal{L}_i$ ;*

$I := I + \frac{I_i}{d+K} * k_d * (\vec{n} \cdot \vec{L}_i)$  {intensité diffuse}

*fin*

*retourner I*

fin

## LES CALCULS D'INTERSECTION

- Pour éviter les calculs d'intersection inutiles on associe à chaque objet un volume enveloppant
  - une sphère
  - ou un parallélépipède
- Les volumes enveloppants sont déterminés une fois pour toutes et ne dépendent pas du rayon initial
- Si le test d'intersection est positif (le volume enveloppant est intersecté), il faut vérifier s'il y a bien intersection: ce calcul est d'autant plus facile que les primitives du modèle de solide sont simples



## LE VOLUME ENVELOPPANT EST UNE SPHÈRE

- la sphère
- le rayon

Déterminer si le rayon intersecte la sphère nécessite

- 7 multiplications
- 8 additions/soustractions

Déterminer les points d'intersection d'une sphère avec un rayon, quand ils existent, nécessite des opérations supplémentaires

- 7 multiplications
- 4 additions/soustractions
- 1 extraction de racine carrée