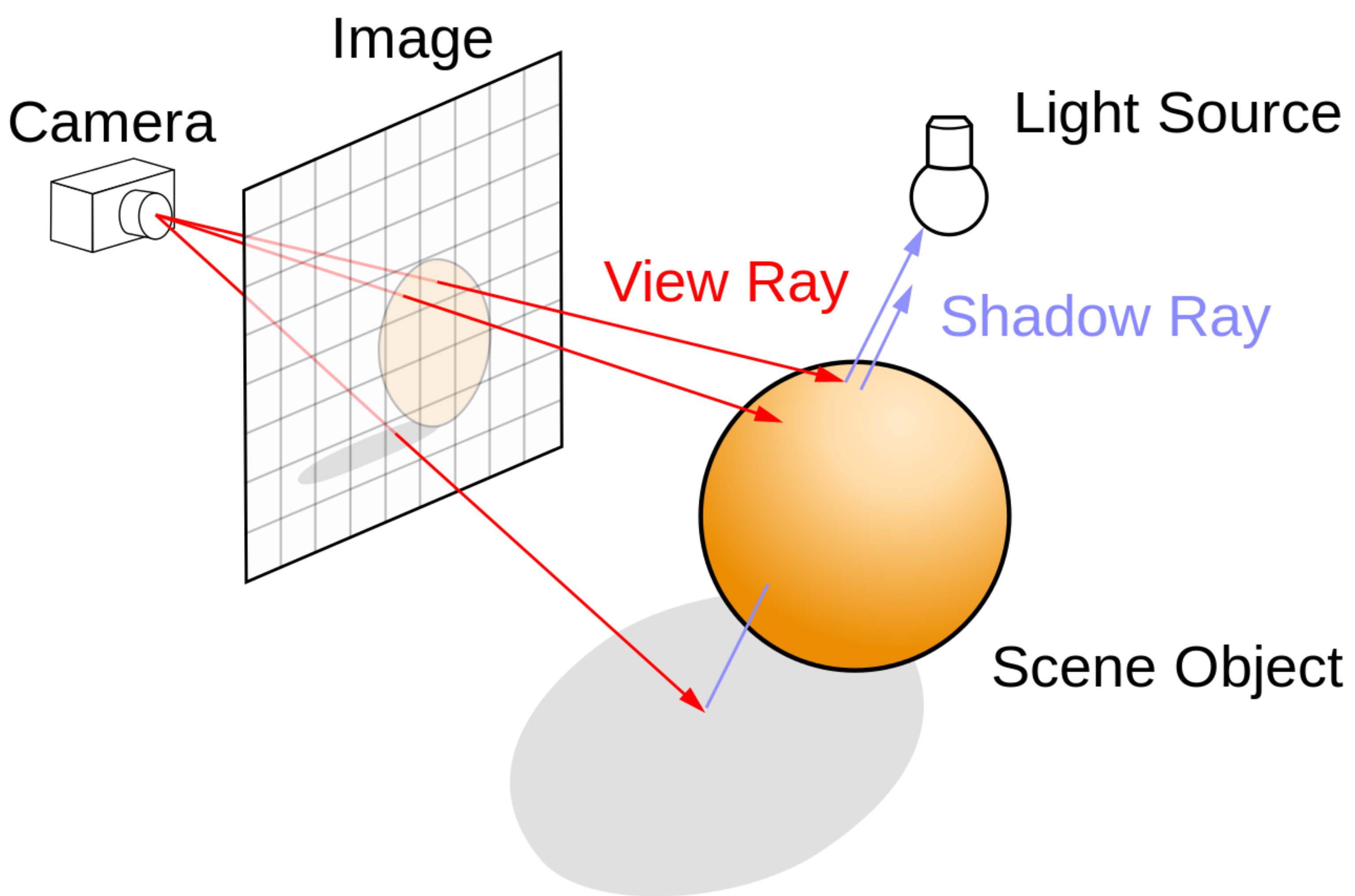
Eclairage et lancé de rayon

HAI719I – Cours 1

Rappel sur les vecteurs

Pourquoi des vecteurs ?



wikipedia

Mais aussi pour :

- Définir une scène virtuelle
- Définir une direction caméra
- Lancer une balle dans une scène
- Pour calculer une ligne de vue

→ Très utilisés dans la communauté du jeu (NVIDIA, Miscrosoft) :

- https://blogs.nvidia.com/blog/2018/03/21/ (epic-games-reflections-ray-tracing-offers-peek-gdc)
- https://www.youtube.com/watch?v=-zW3Ghz-WQw
- https://www.youtube.com/watch?v=81E9yVU-KB8

Scalaire

Quantité décrivant l'**amplitude** (i.e., un simple nombre)

- Poids
- Distance (Montpellier – Béziers = 73,6 km)
- Vitesse
- Taille d'un vecteur
- Nombres tels que $\pi = 3.14159$ ... etc

Sur l'ordinateur : int, float, double

Vecteurs

Quantité ayant une **direction en plus d'une amplitude**

Une représentation de vecteur B-M:
- Commencer en M et finir à B; le vecteur relie les 2
- Commencer en M : bouger de 73,6 km à 45 degrés au Sud ouest

Vecteurs

Quantité ayant une **direction en plus d'une amplitude**

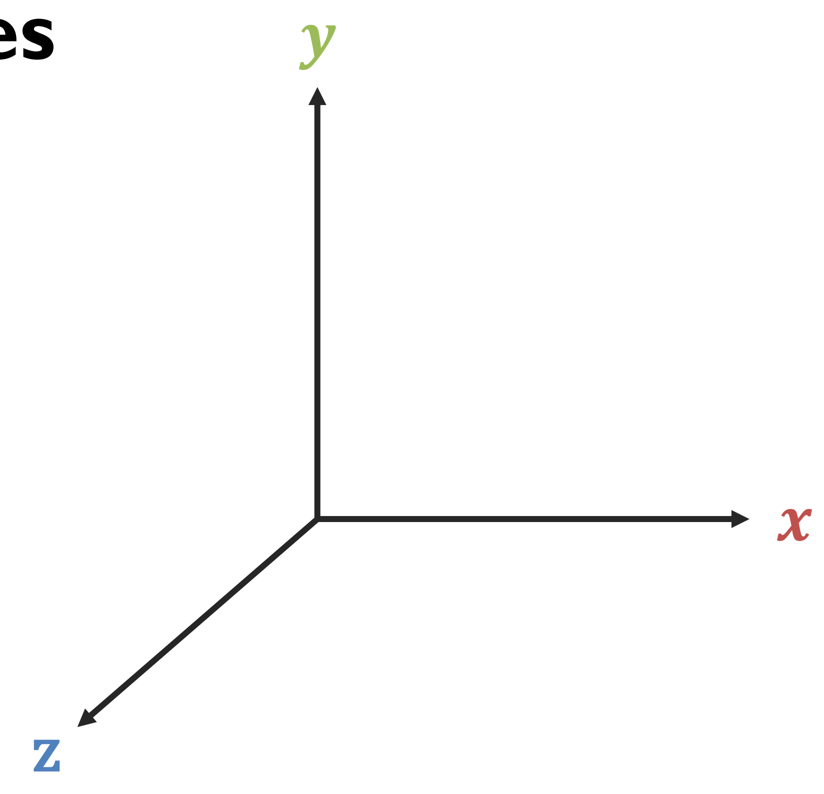
Une représentation **équivalente** du vecteur B-M

Directions de référence

→ **Système de coordonnées**

Nombre de directions de référence = dimension de l'espace
- Espace de dimension $d \Rightarrow \mathbb{R}^d$; 2D $\Rightarrow \mathbb{R}^2$; 3D $\Rightarrow \mathbb{R}^3$

Système de coordonnées **cartésiennes** en 3D :
- Direction de référence **orthogonales**

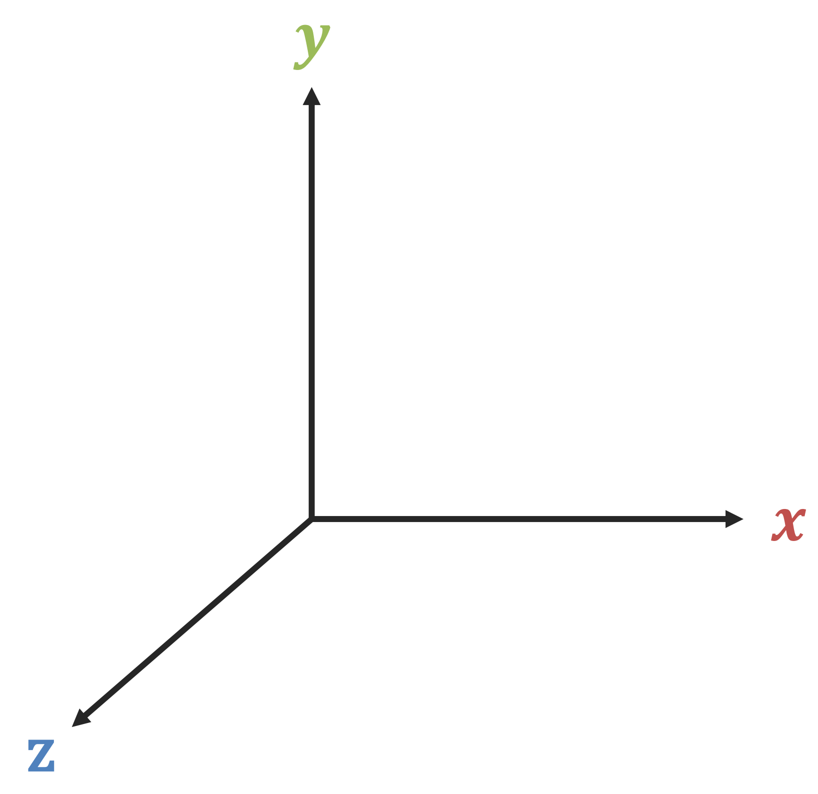


Système de coordonnées

Un point **P** est représenté par :

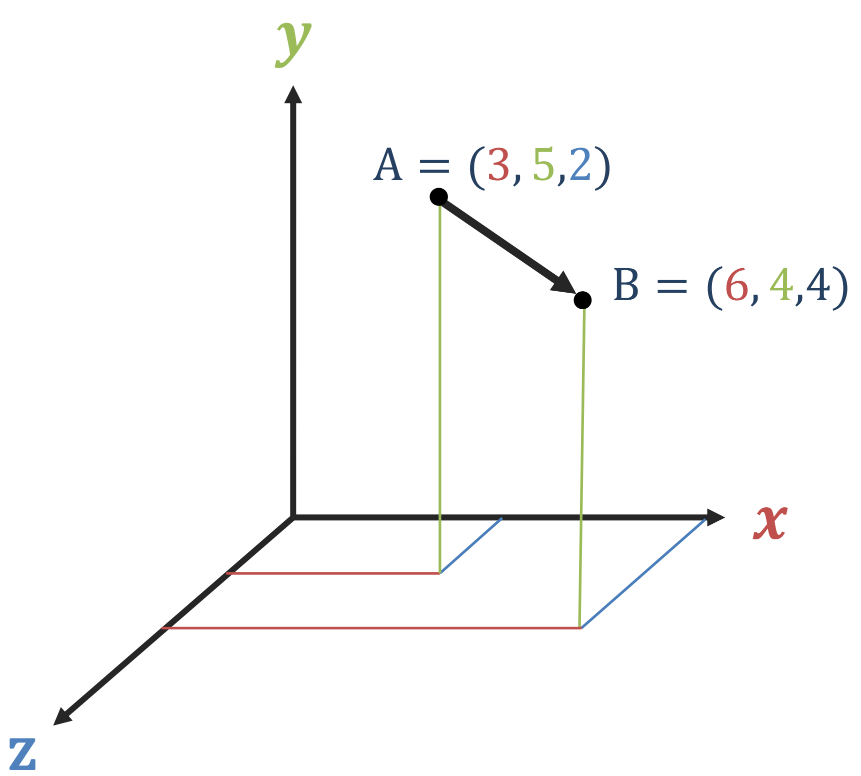
- Les coordonnées (x, y) en **deux** dimensions
- Les coordonnées (x, y, z) en **trois** dimensions
- Les coordonnées $(x_1, x_2, \dots, x_d)$ en **d** dimensions

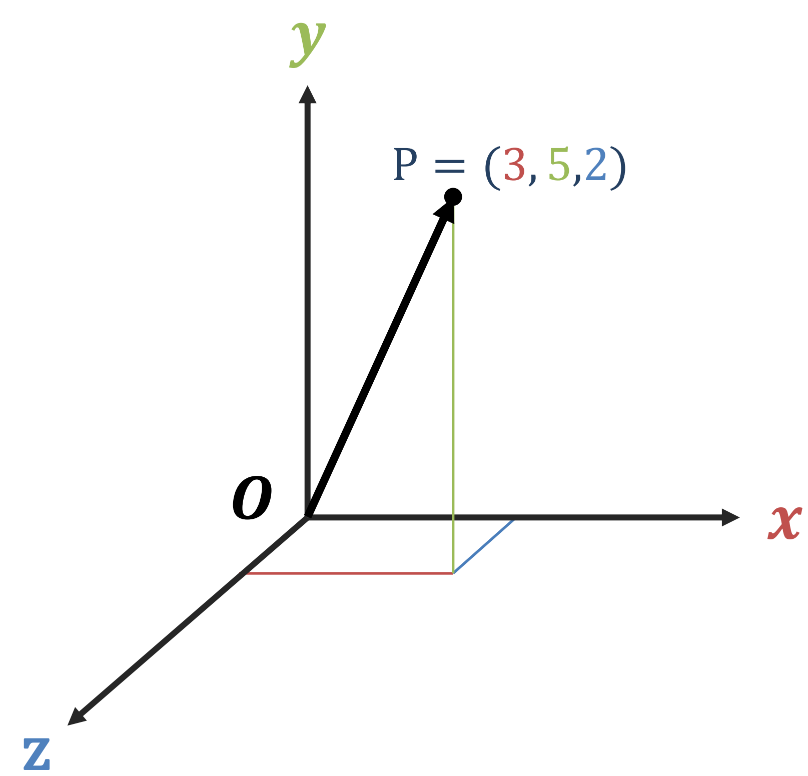
L'origine du système : toutes les entrées de P sont à zéro



Un **point** vs un **vecteur**

Un vecteur ne spécifie **pas de point de départ**





→ une **direction** et une **longueur** de la flèche

Représentation

- Un point P : $(x_1, x_2, \ldots, x_d)$ en d dimensions
  - par (x, y) en 2D et par $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ en 3D

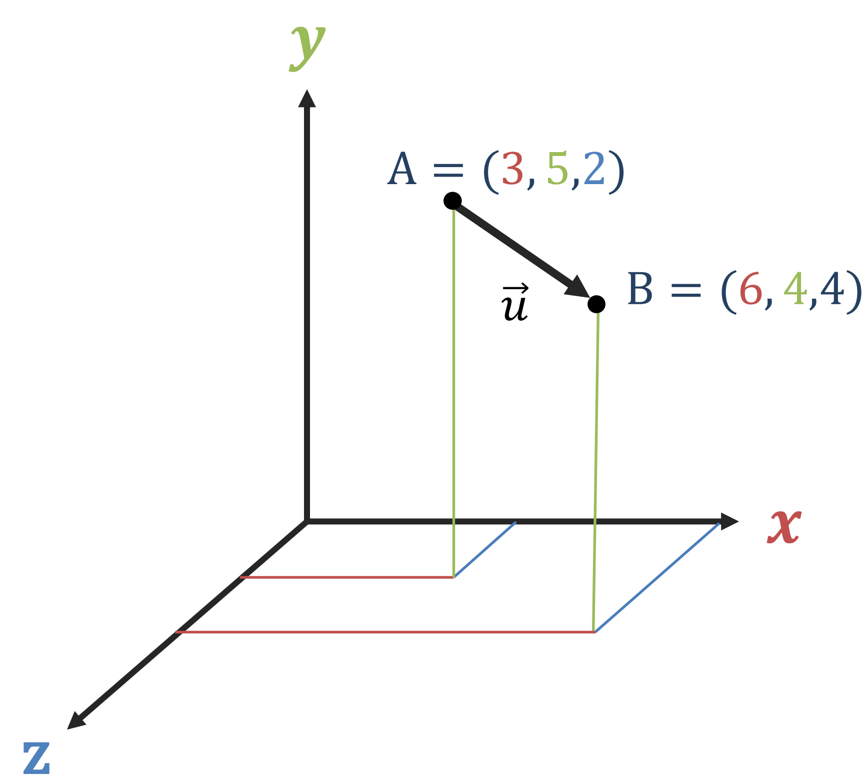
- Un vecteur $\vec{v}$: $\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}$ en dimension d
  - par $\begin{bmatrix} v_x \\ v_y \end{bmatrix}$ en 2D et par $\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$ en 3D

Représentation

Soit le vecteur $\vec{u}$ entre A et B :

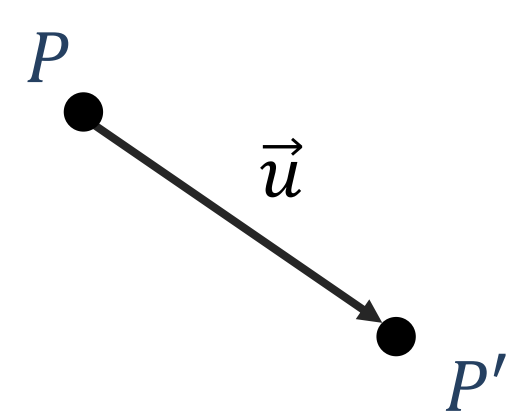
- définit une direction dans l'espace
- est définit par 3 coordonnées

$$u_x = B_x - A_x$$
$$u_y = B_y - A_y$$
$$u_z = B_z - A_z$$



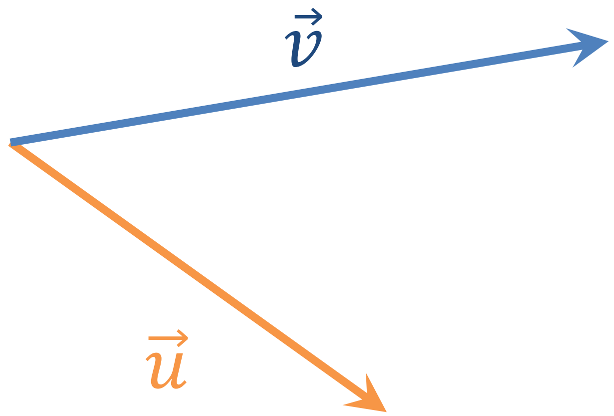
Permet de translater des objets :

$$P'_x = P_x + u_x$$
$$P'_y = P_y + u_y$$
$$P'_z = P_z + u_z$$



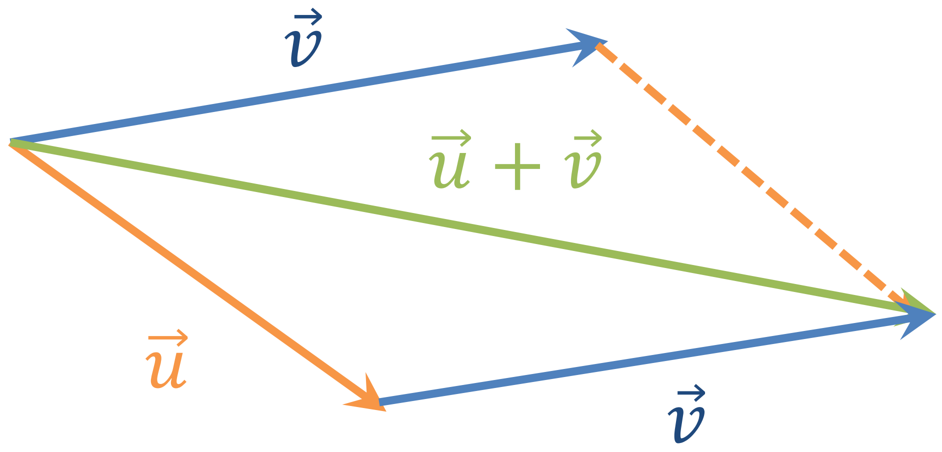
Opérations sur les vecteurs

Addition



Opérations sur les vecteurs

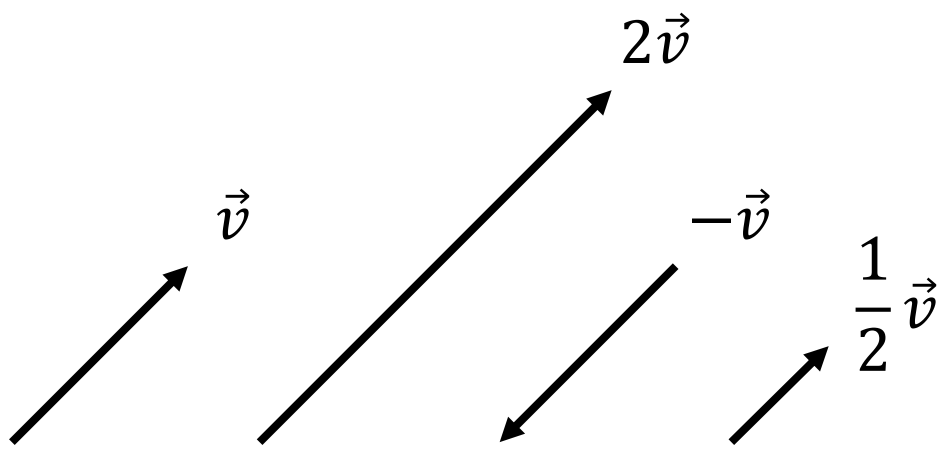
Addition



$$\vec{u} + \vec{v} = \begin{bmatrix} u_x + v_x \\ u_y + v_y \\ u_z + v_z \end{bmatrix}$$

Opérations sur les vecteurs

Multiplication avec un scalaire λ



$$\lambda\vec{v} = \begin{bmatrix} \lambda v_x \\ \lambda v_y \\ \lambda v_z \end{bmatrix}$$

Amplitude (longueur, norme)

La **norme** de $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$:

- Définie par $\|\vec{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$
- Si $\vec{v} = \overrightarrow{AB}$, alors est la distance entre A et B
- Si $\|\vec{v}\|$ = 1 alors $\vec{v}$ est **normalisé**
- Pour tout vecteur il existe un **vecteur normalisé unitaire** de même direction.
- Pour normaliser $\vec{v}$:

$$\hat{v} = \frac{1}{\|\vec{v}\|} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x/\|\vec{v}\| \\ v_y/\|\vec{v}\| \\ v_{z/}\|\vec{v}\| \end{bmatrix}$$

Amplitude (longueur, norme)

La **norme** de $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$:

- Définie par $\|\vec{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$
- Si $\vec{v} = \overrightarrow{AB}$, alors est la distance entre A et B
- Si $\|\vec{v}\|$ = 1 alors $\vec{v}$ est **normalisé**
- Pour tout vecteur il existe un **vecteur normalisé unitaire** de même direction.
- Pour normaliser $\vec{v}$:

$$\hat{v} = \frac{1}{\|\vec{v}\|} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x/\|\vec{v}\| \\ v_y/\|\vec{v}\| \\ v_{z/}\|\vec{v}\| \end{bmatrix}$$

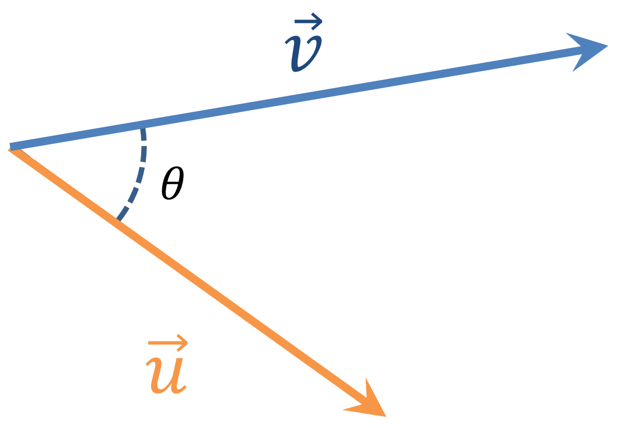
Les **vecteurs unitaires** des directions de référence forment les vecteurs de base ($\hat{x}$, $\hat{y}$, $\hat{z}$ en 3D).

Produit scalaire entre 2 vecteurs

« Dot product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$\vec{u}.\vec{v} = \vec{v}.\vec{u} = u_x v_x + u_y v_y + u_z v_z$ donne un **scalaire**

Géométriquement : $\vec{u}.\vec{v} = ||\vec{u}||\,||\vec{v}|| \cos\theta$



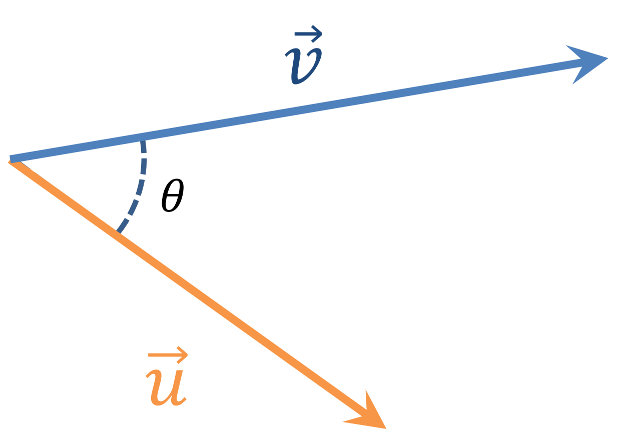
Produit scalaire entre 2 vecteurs

« Dot product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$\vec{u}.\vec{v} = \vec{v}.\vec{u} = u_x v_x + u_y v_y + u_z v_z$ donne un **scalaire**

Propriétés :

- Symétrie : $\vec{u}.\vec{v} = \vec{v}.\vec{u}$
- Distributivité : $\vec{u}.(\vec{v} + \vec{w}) = \vec{u}.\vec{v} + \vec{u}.\vec{w}$
- Homogénéité : $(\lambda\vec{u}).\vec{v} = \lambda(\vec{u}.\vec{v})$
- Lien avec la norme : $\vec{u}.\vec{u} = \|\vec{u}\|^2$



Remarque :

$$\vec{u}^T\vec{v} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

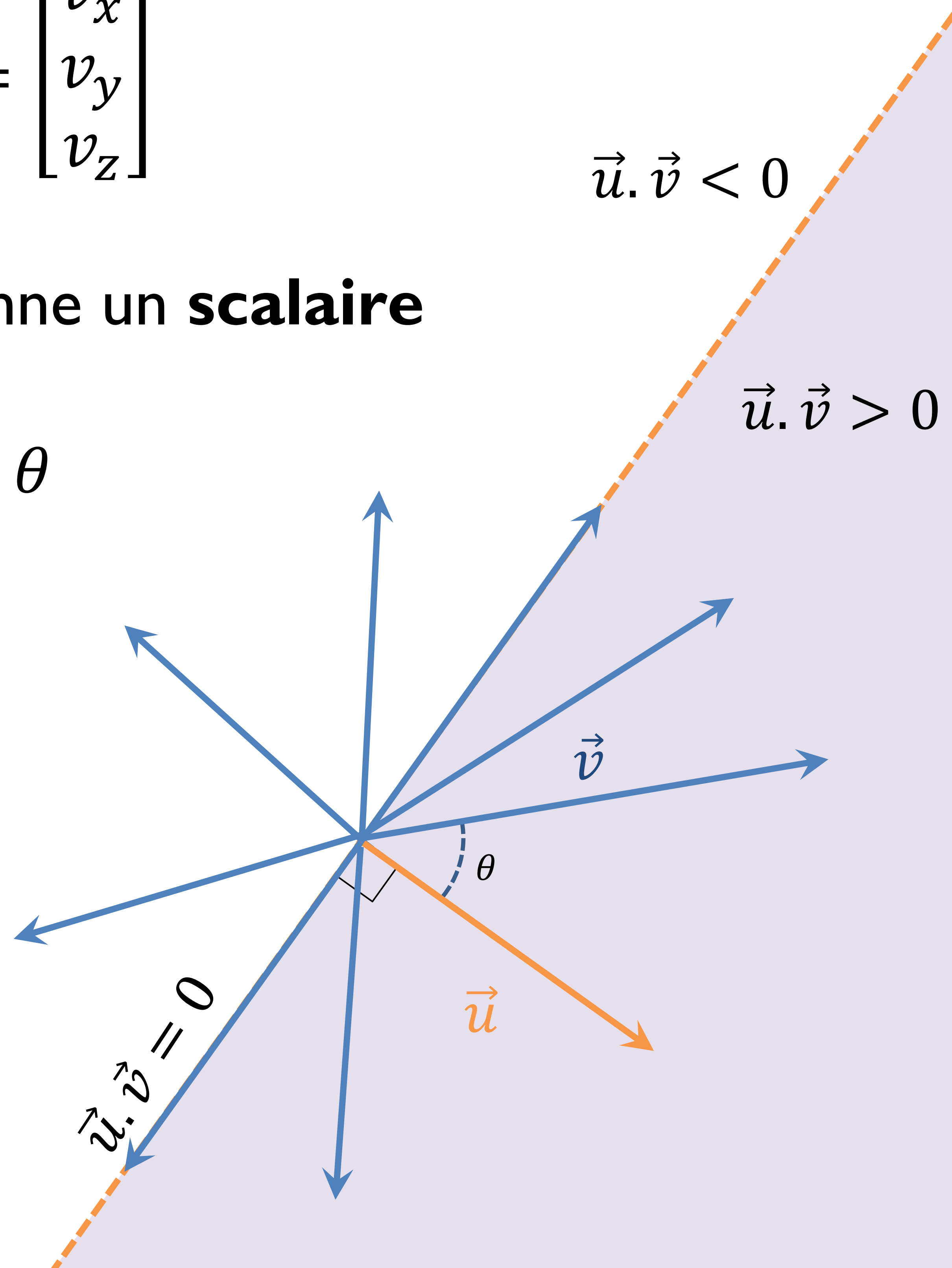
Produit scalaire entre 2 vecteurs

« Dot product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$\vec{u}.\vec{v} = \vec{v}.\vec{u} = u_x v_x + u_y v_y + u_z v_z$ donne un **scalaire**

Géométriquement : $\vec{u}.\vec{v} = \|\vec{u}\|\|\vec{v}\| \cos\theta$

$\vec{u}.\vec{v} = 0$ pour $\vec{u}$ et $\vec{v}$ **orthogonaux**
$(\theta = 90$ et $\theta = 270)$

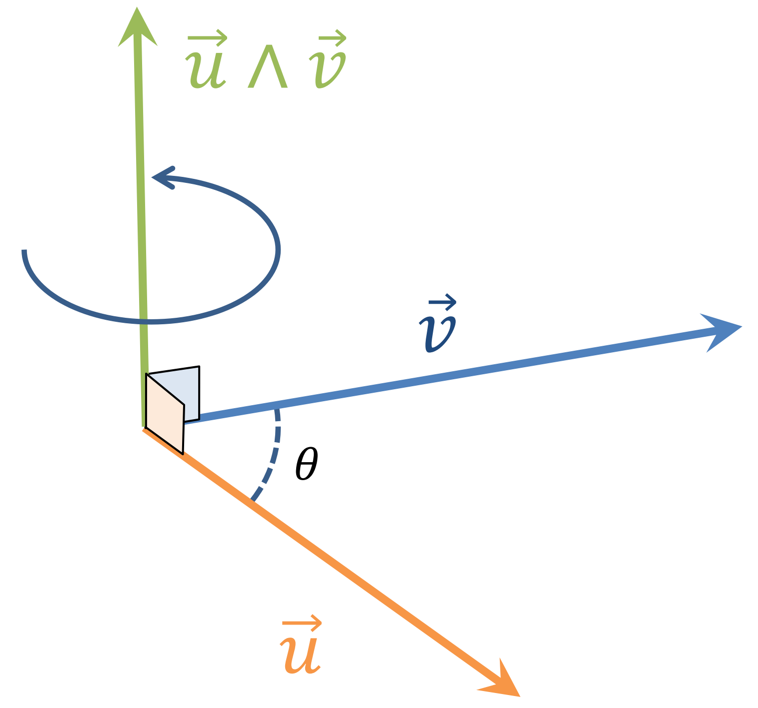


Produit vectoriel entre 2 vecteurs

« Cross product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$$\vec{u} \wedge \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

$\vec{v} \wedge \vec{v}$ est un vecteur orthogonal à $\vec{u}$ et à $\vec{v}$

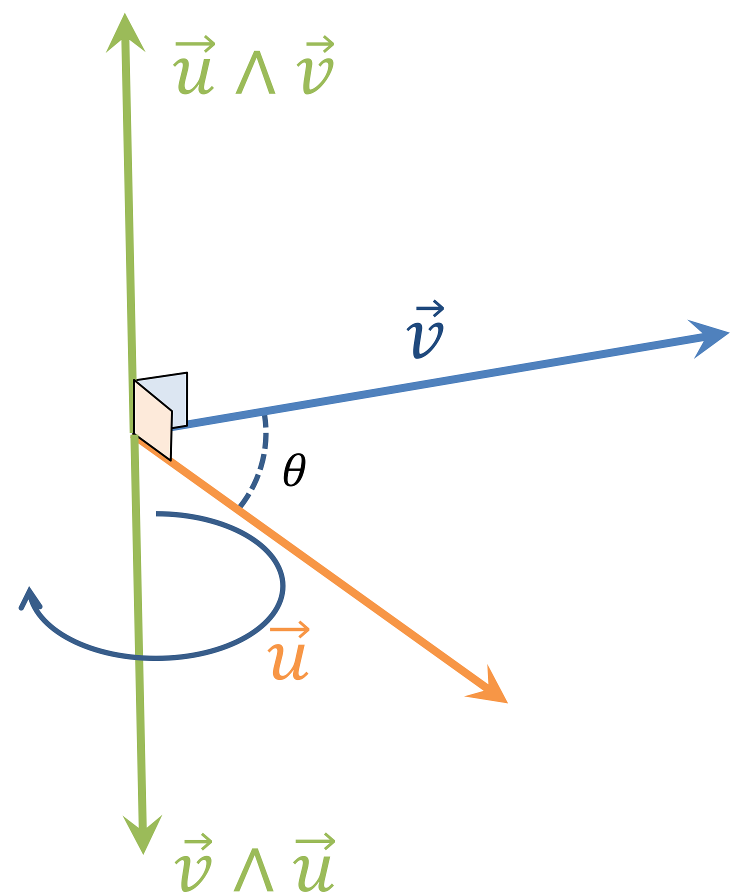


Produit vectoriel entre 2 vecteurs

« Cross product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$$\vec{u} \wedge \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

$\vec{u} \wedge \vec{v}$ est un vecteur orthogonal à $\vec{u}$ et à $\vec{v}$



Propriétés :

- Antisymétrique : $\vec{u} \wedge \vec{v} = -\vec{v} \wedge \vec{u}$
- Distributivité : $\vec{u} \wedge (\vec{v} + \vec{w}) = \vec{u} \wedge \vec{v} + \vec{u} \wedge \vec{w}$
- Homogénéité : $(\lambda\vec{u}) \wedge \vec{v} = \lambda(\vec{u} \wedge \vec{v})$
- Non associatif : $\vec{u} \wedge (\vec{v} \wedge \vec{w}) = (\vec{u} \wedge \vec{v}) \wedge \vec{w}$

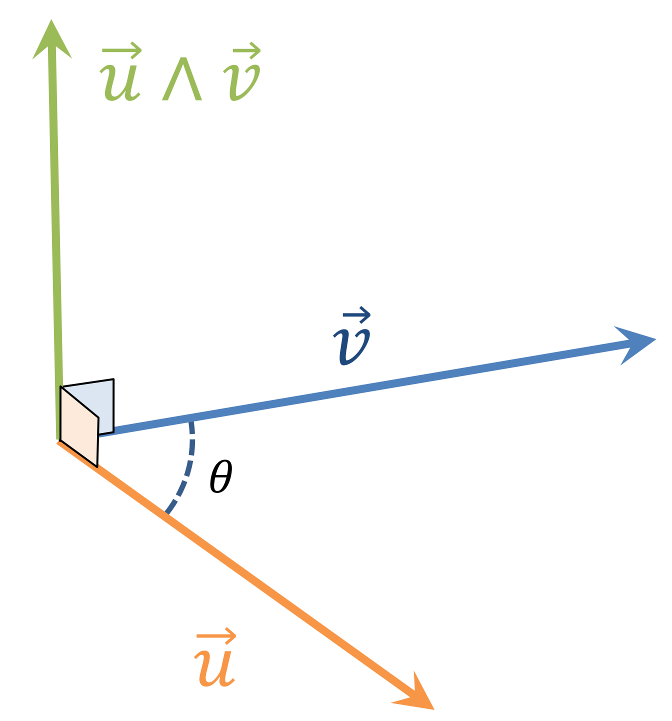
Produit vectoriel entre 2 vecteurs

« Cross product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$$\vec{u} \wedge \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

Géométriquement : $\|\vec{u} \wedge \vec{v}\| = \|\vec{u}\| \|\vec{v}\| \sin\theta$

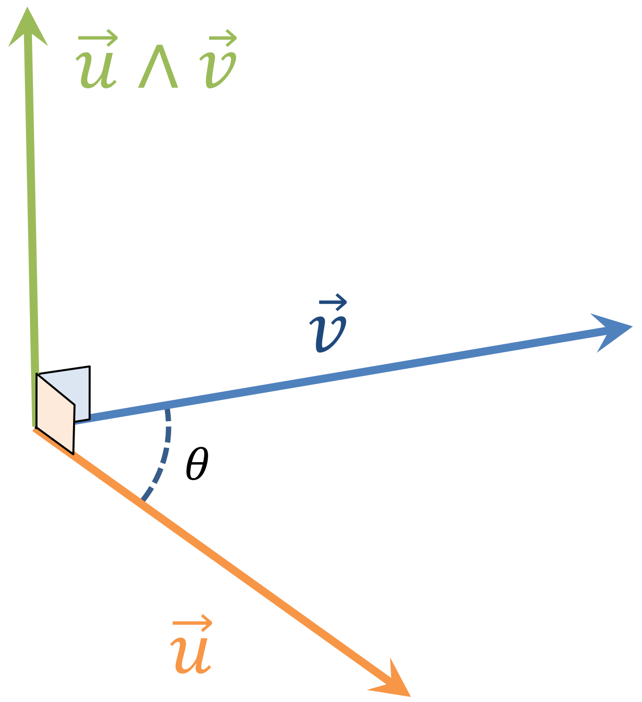
$\vec{u} \wedge \vec{v} = 0$ pour $\vec{u}$ et $\vec{v}$ **colinéaires**
($\theta = 0$ et $\theta = 180$)



Produit vectoriel entre 2 vecteurs

« Cross product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

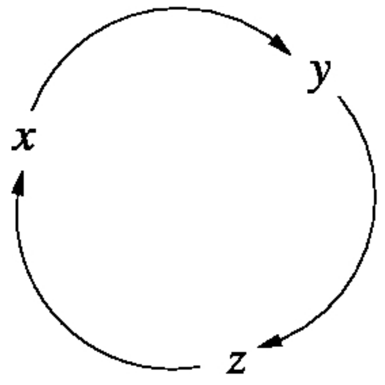
$$\vec{u} \wedge \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

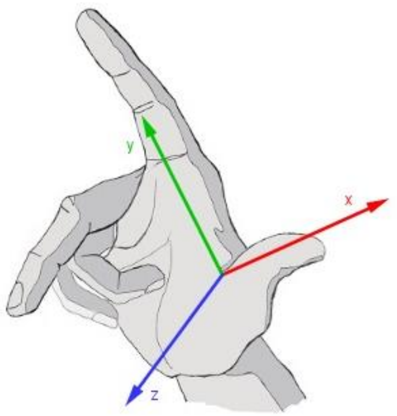


Système de coordonnées main droite :

$\hat{x} \times \hat{y} = \hat{z}, \hat{y} \times \hat{z} = \hat{x}, \hat{z} \times \hat{x} = \hat{y}$

i.e. $(\hat{x} \times \hat{y}).\hat{z} > 0$



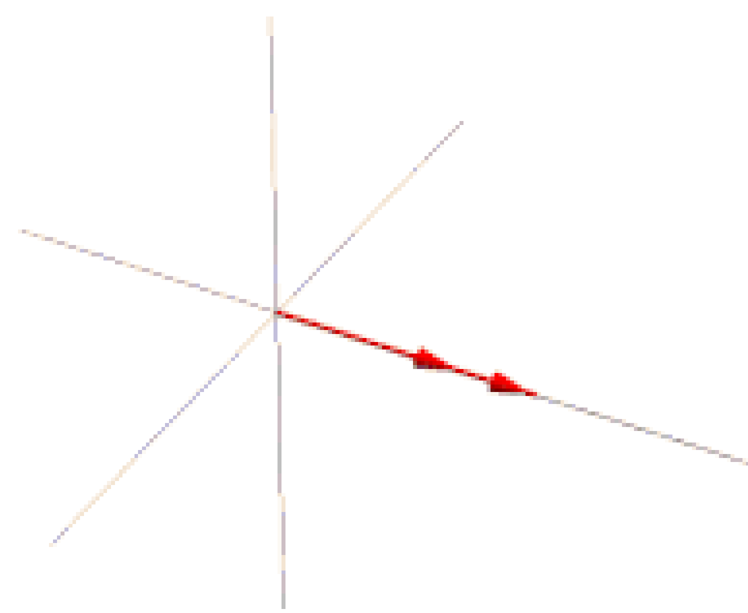


Produit vectoriel entre 2 vecteurs

« Cross product » entre $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ et $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

$$\vec{u} \wedge \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

$\vec{v} \wedge \vec{v}$ est un vecteur orthogonal à $\vec{u}$ et à $\vec{v}$



Utilisé pour calculer :

- les normales,
- Les repères orthonormés…

Projections

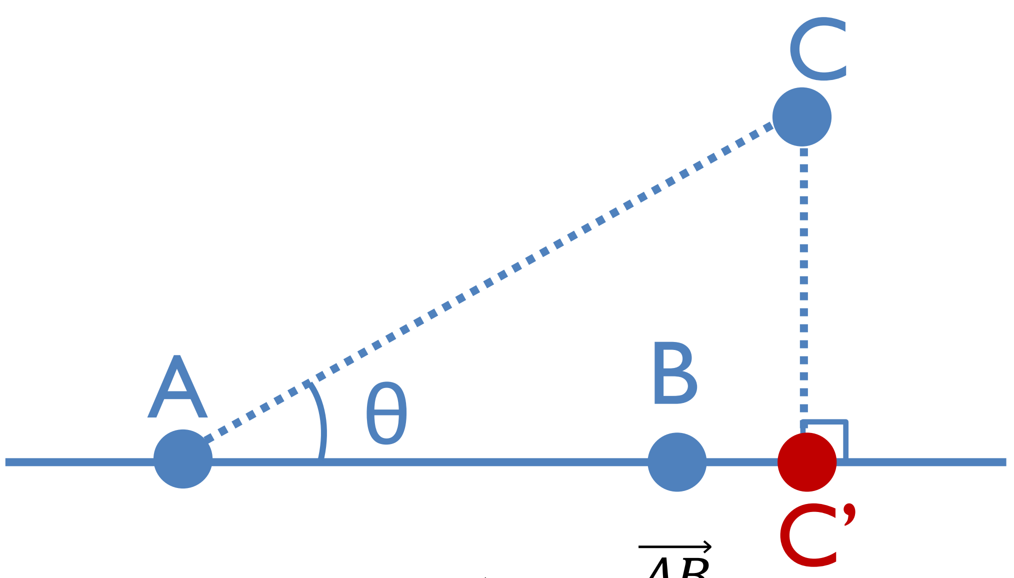
D’un point sur une droite

Soit la droit **d** définie par 2 points A et B. Le point C à projeter.

- Par Pythagore $\frac{\|\overrightarrow{AC'}\|}{\|\overrightarrow{AC}\|} = \cos\theta$
- On vient de voir que $\frac{\overrightarrow{AB}.\overrightarrow{AC}}{\|\overrightarrow{AB}\|\|\overrightarrow{AC}\|} = \cos\theta$

D’où $\|\overrightarrow{AC'}\| = \frac{\overrightarrow{AB}.\overrightarrow{AC}}{\|\overrightarrow{AC}\|}$

Finalement $C' = \begin{bmatrix} A_x + u_x\|\overrightarrow{AC'}\| \\ A_y + u_y\|\overrightarrow{AC'}\| \\ A_z + u_z\|\overrightarrow{AC'}\| \end{bmatrix}$



On note $\vec{u} = \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|}$

Projections

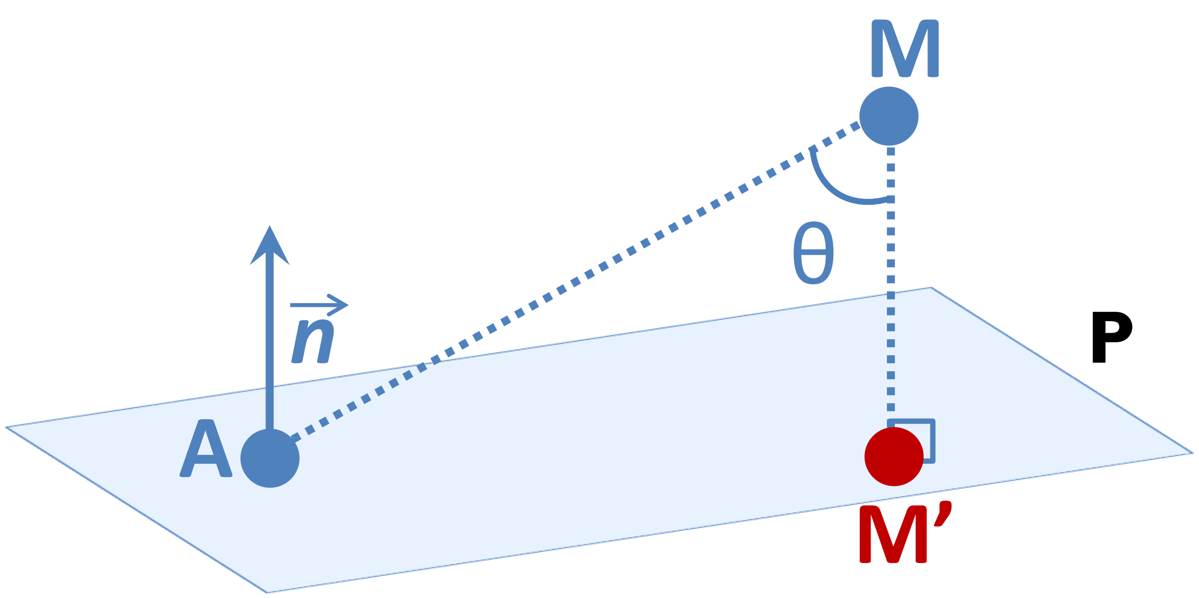
D'un point sur un plan

Soit le plan **P** défini par un point A et une normale $\vec{n}$. Le point M à projeter.

- Par Pythagore $\frac{\|\overrightarrow{MM'}\|}{\|\overrightarrow{MA}\|} = \cos\theta$
- On vient de voir que $\frac{\overrightarrow{MA}.\vec{n}}{\|\overrightarrow{MA}\|\|\vec{n}\|} = \cos\theta$

D'où $\|\overrightarrow{MM'}\| = \frac{\overrightarrow{MA}.\vec{n}}{\|\vec{n}\|}$

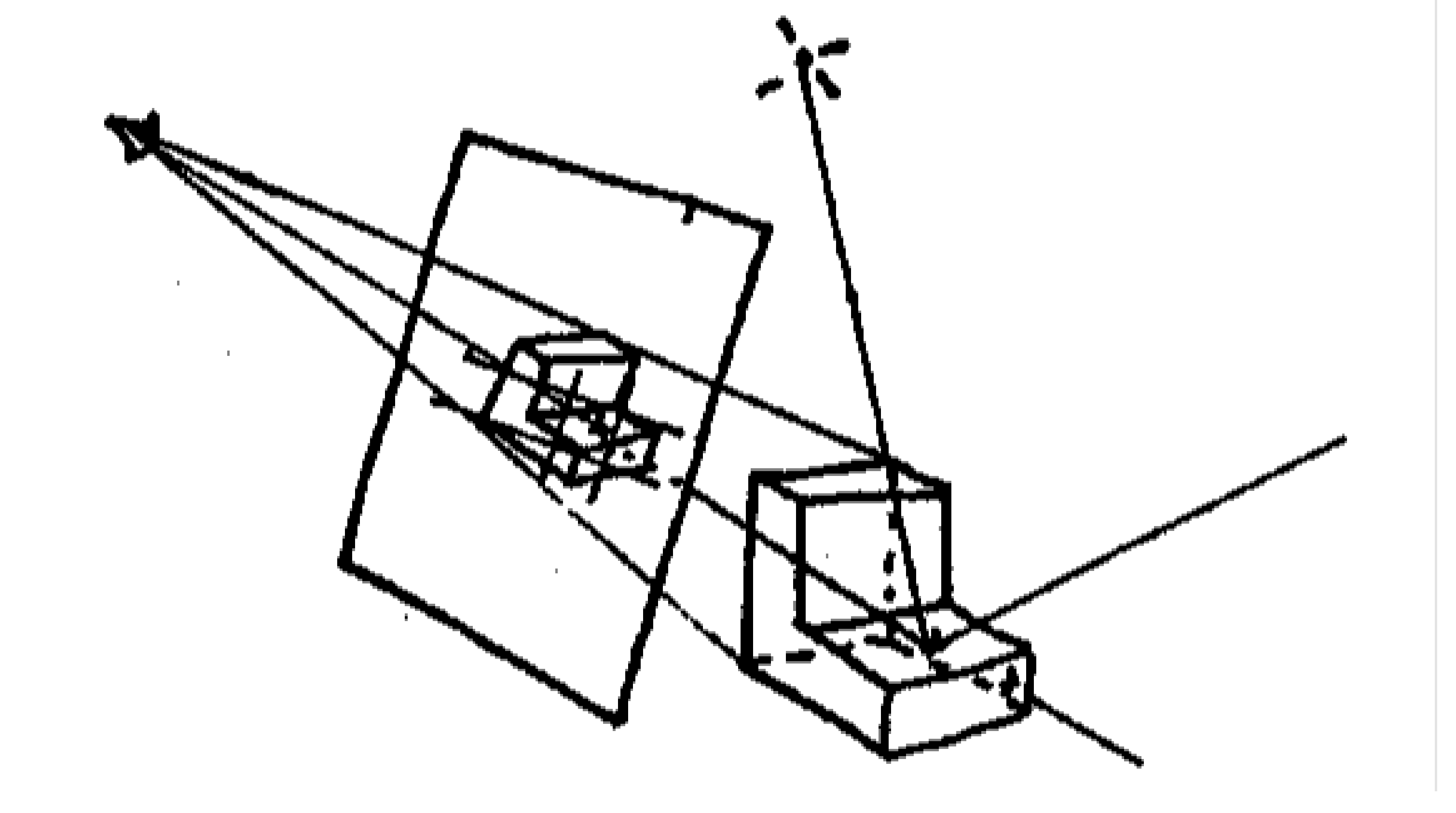
Finalement $M' = \begin{bmatrix} M_x - n_x\|\overrightarrow{MM'}\| \\ M_y - n_y\|\overrightarrow{MM'}\| \\ M_z - n_z\|\overrightarrow{MM'}\| \end{bmatrix}$



Lancer de rayon

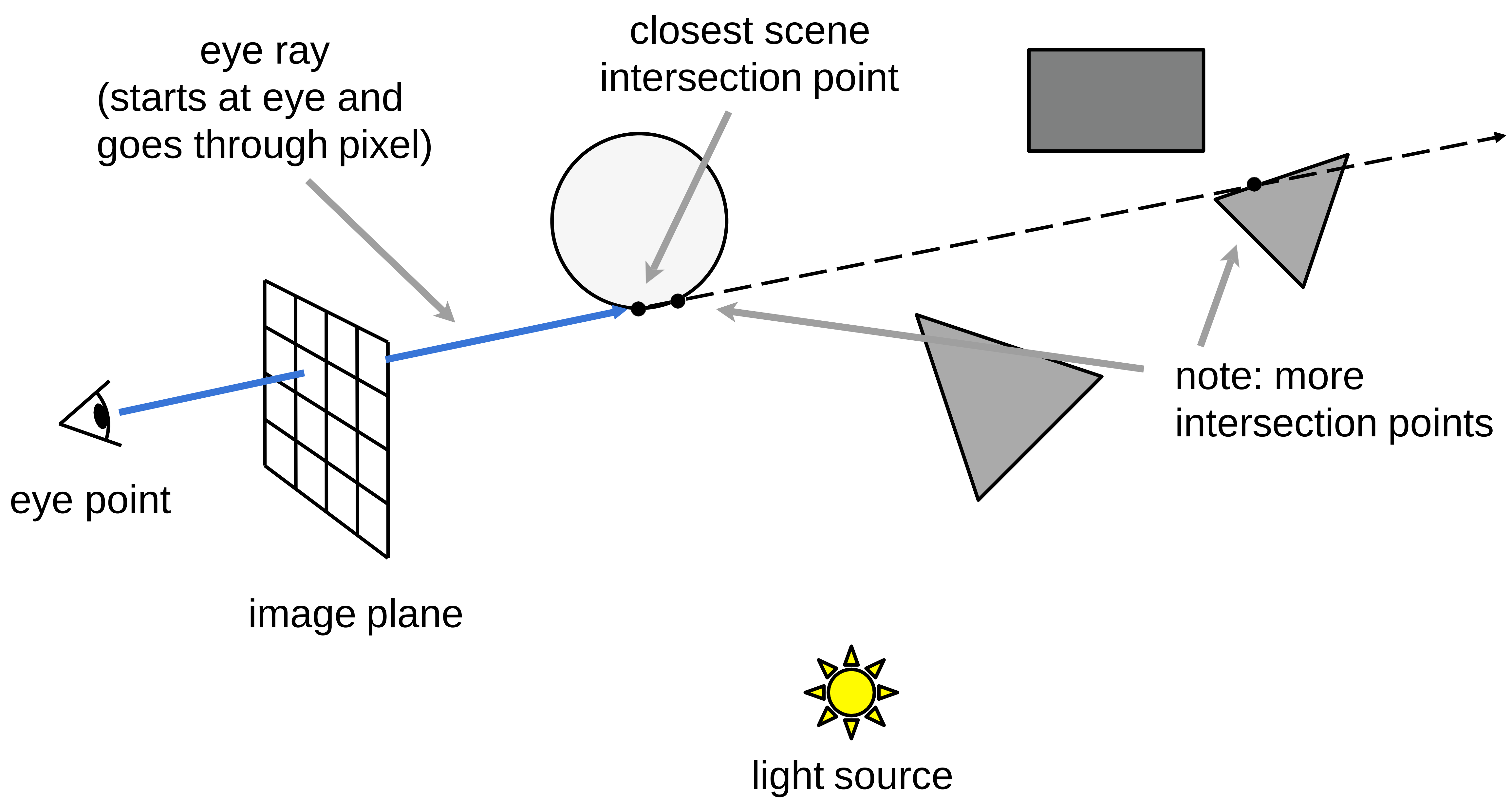
Appel 1968 - Ray casting

1. Générer une image en lançant un rayon par pixel
2. Vérifier les ombres avec un rayon lancé vers la lumière



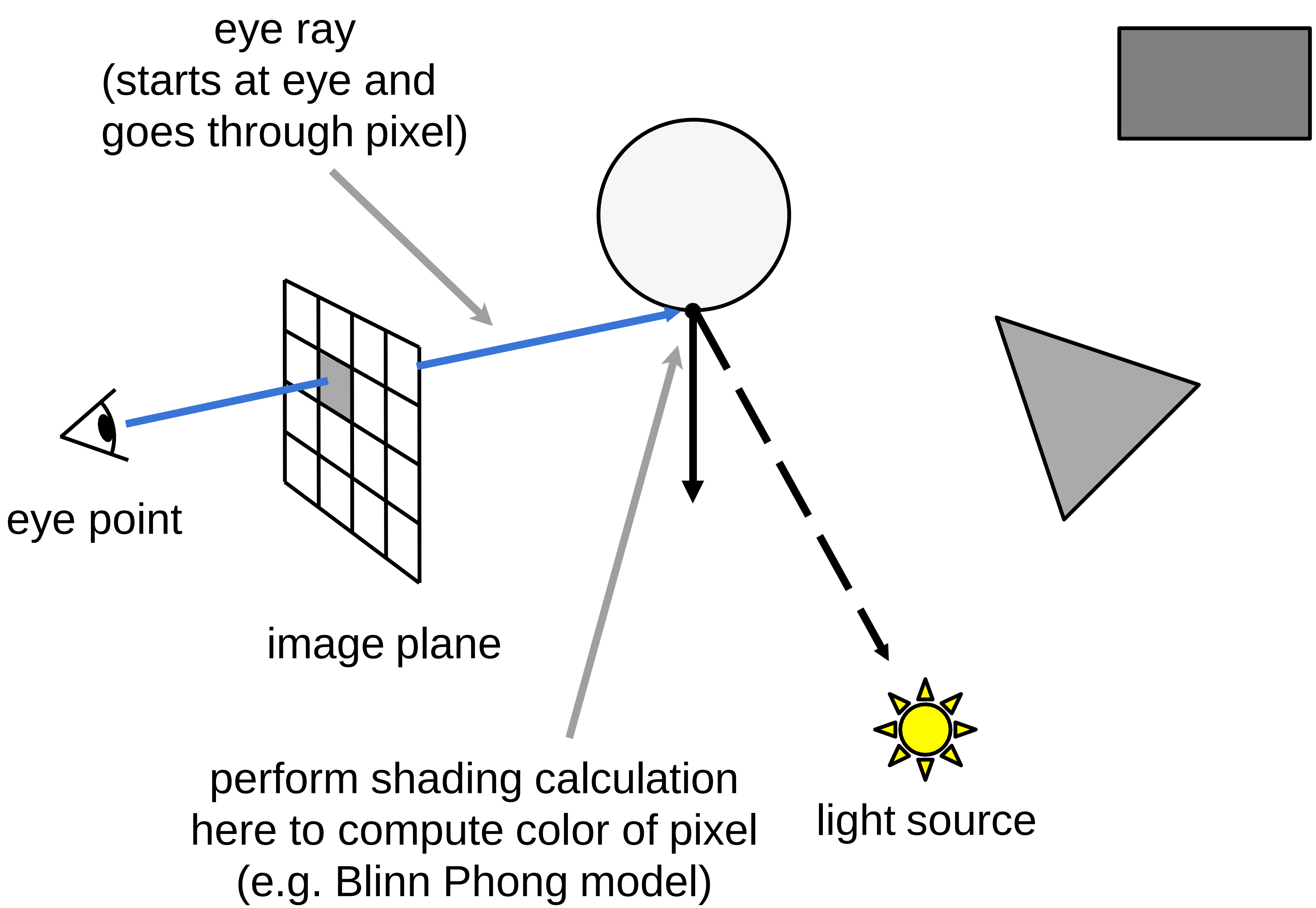
Lancer de rayon- rayon de vue

Pinhole Camera Model



Lancer de rayon - Shading Pixels (Local Only)

Pinhole Camera Model

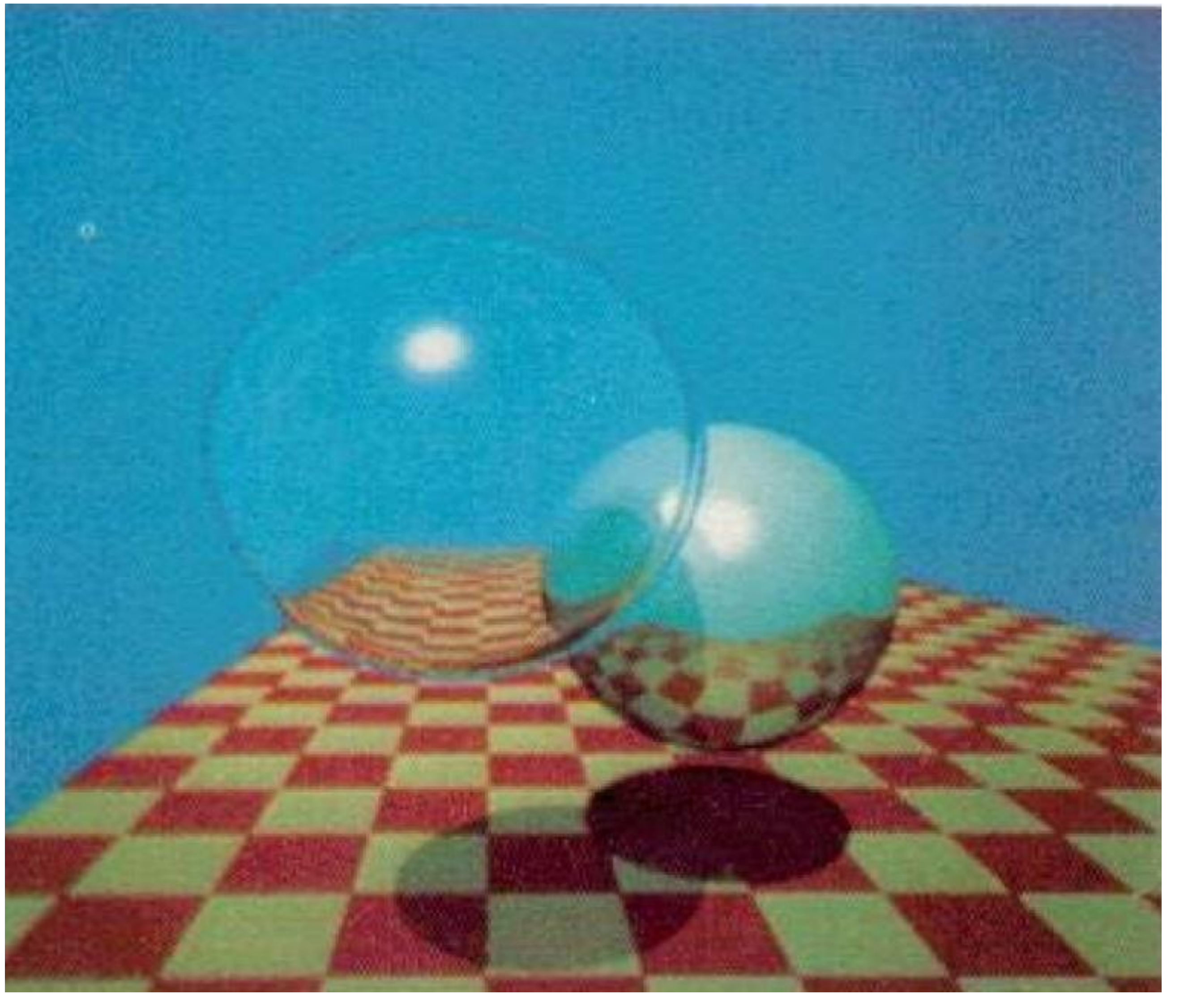


Lancer de rayon récursif

“An improved Illumination model for shaded display”
T. Whitted, CACM 1980

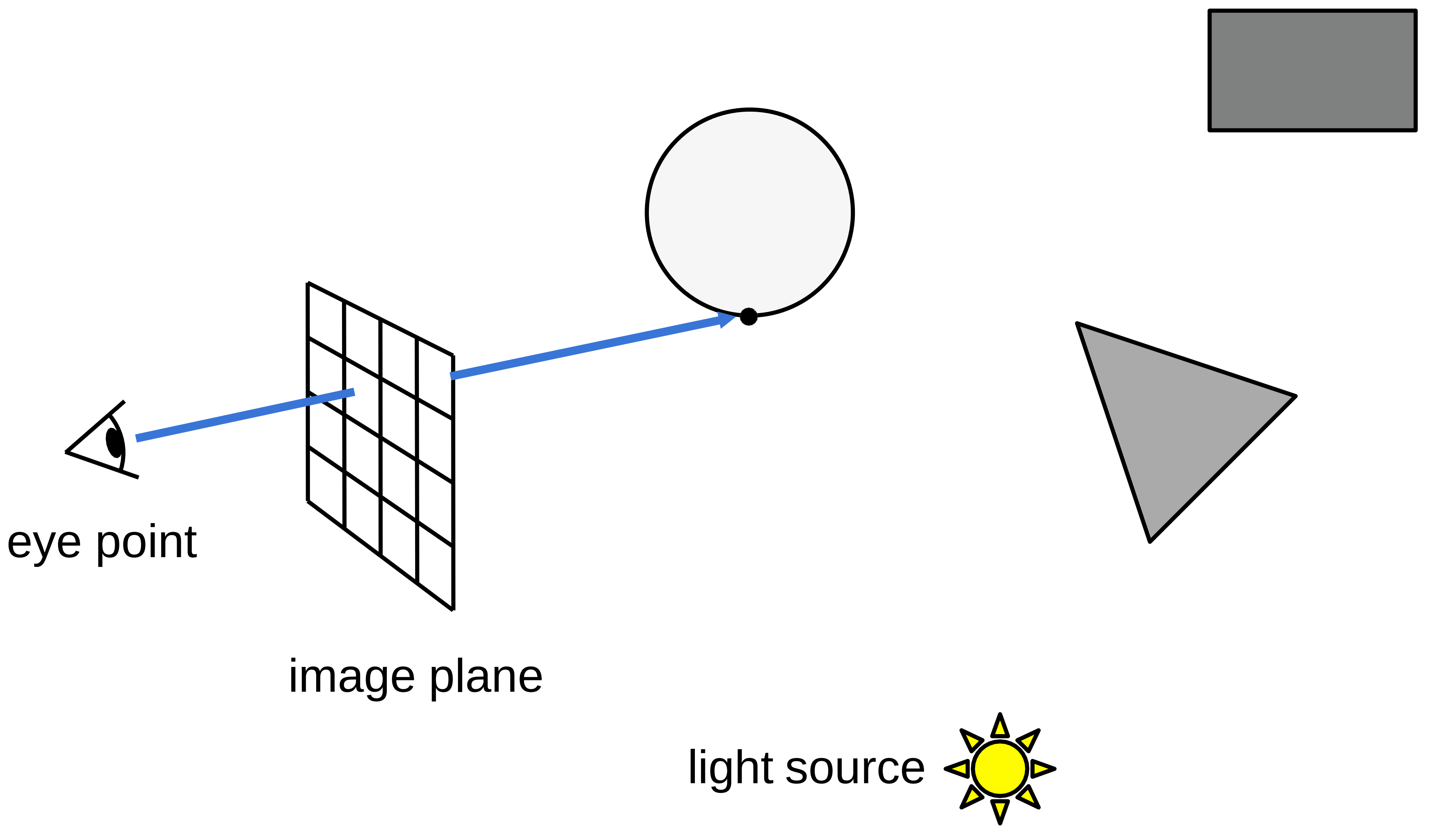
Temps de calcul :

- **VAX 11/780 (1979) 74m**
- **PC (2006) 6s**
- **GPU (2012) 1/30s**

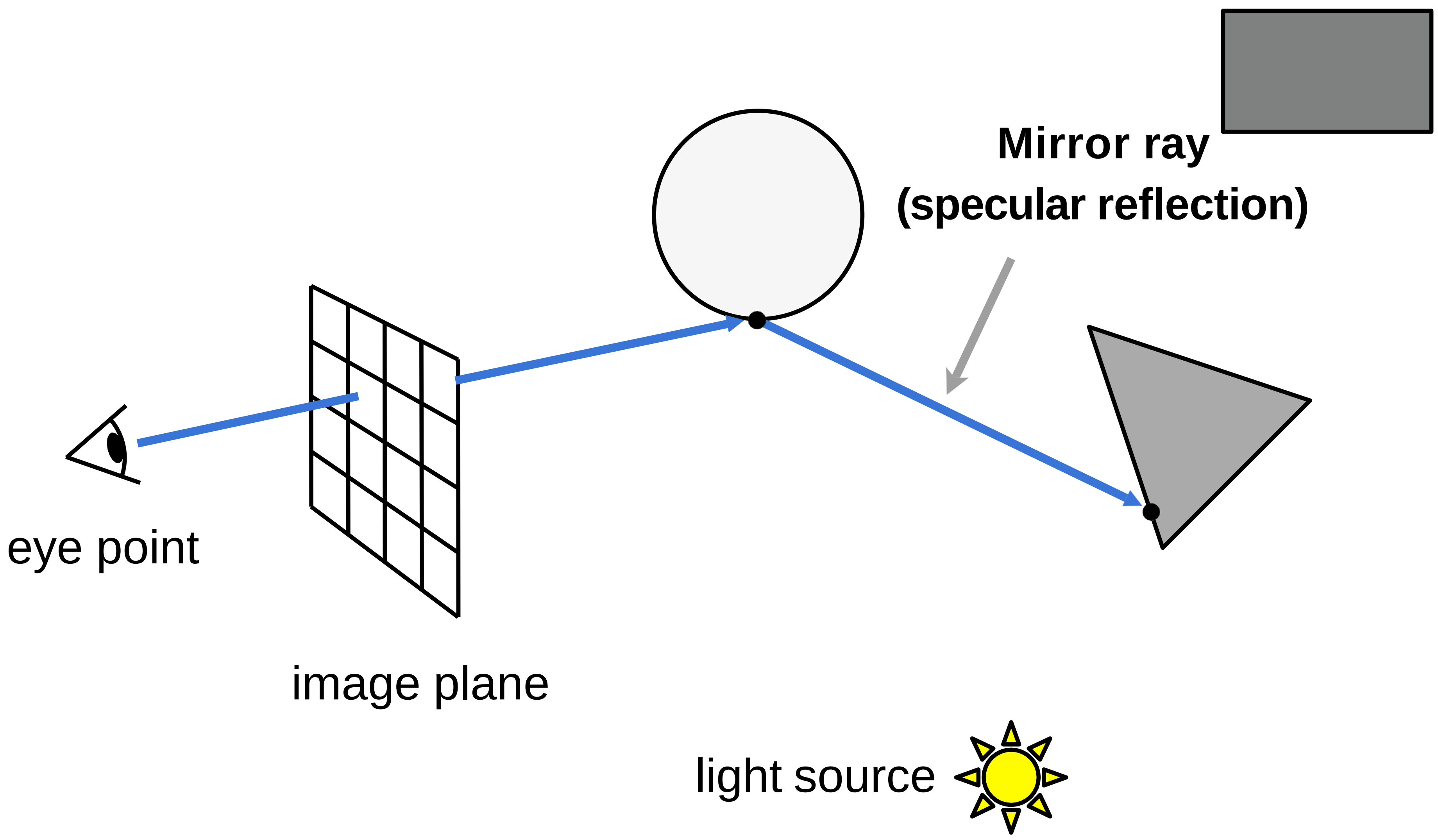


Spheres and Checkerboard, T. Whitted, 1979

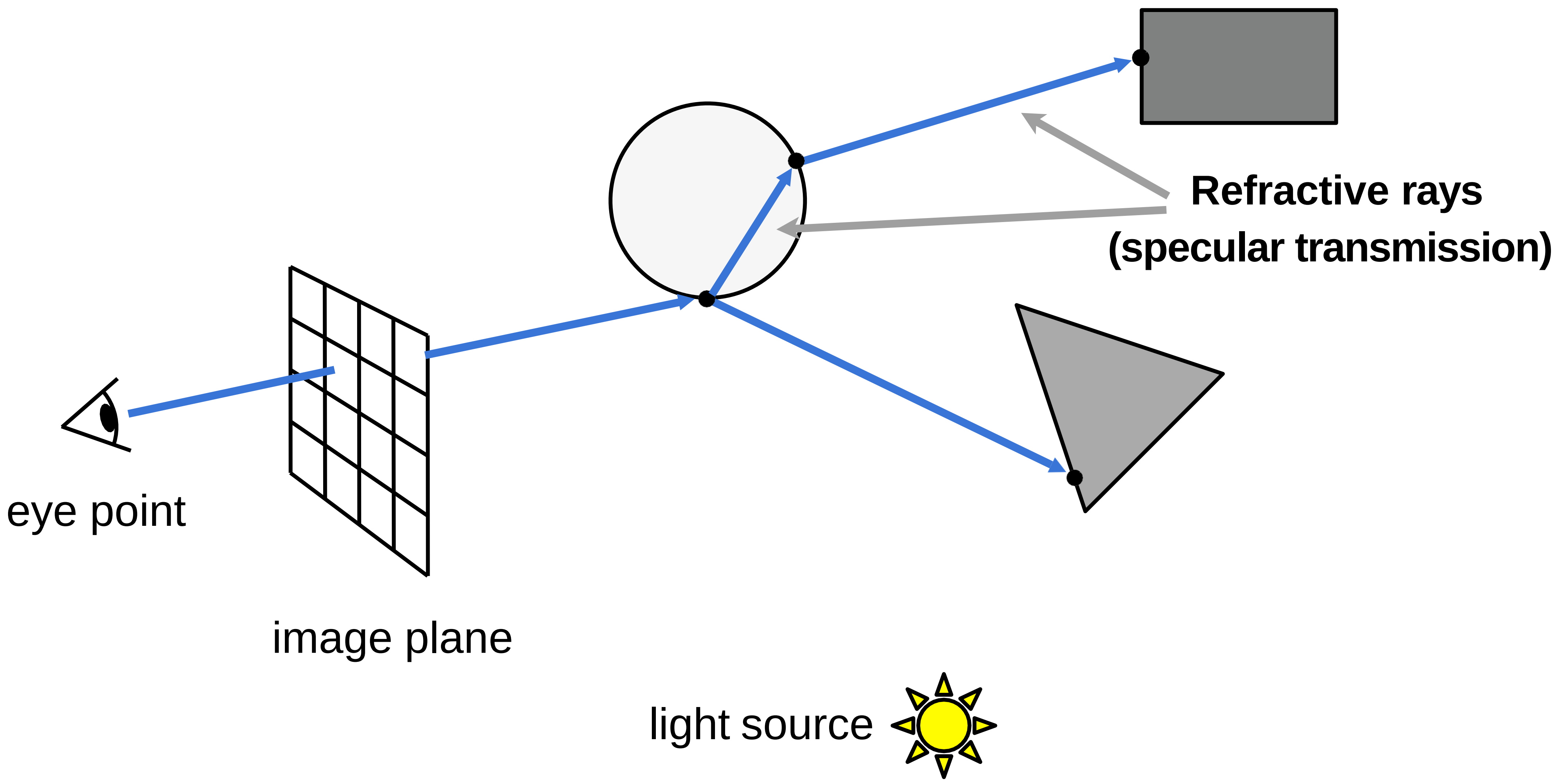
Lancer de rayon récursif



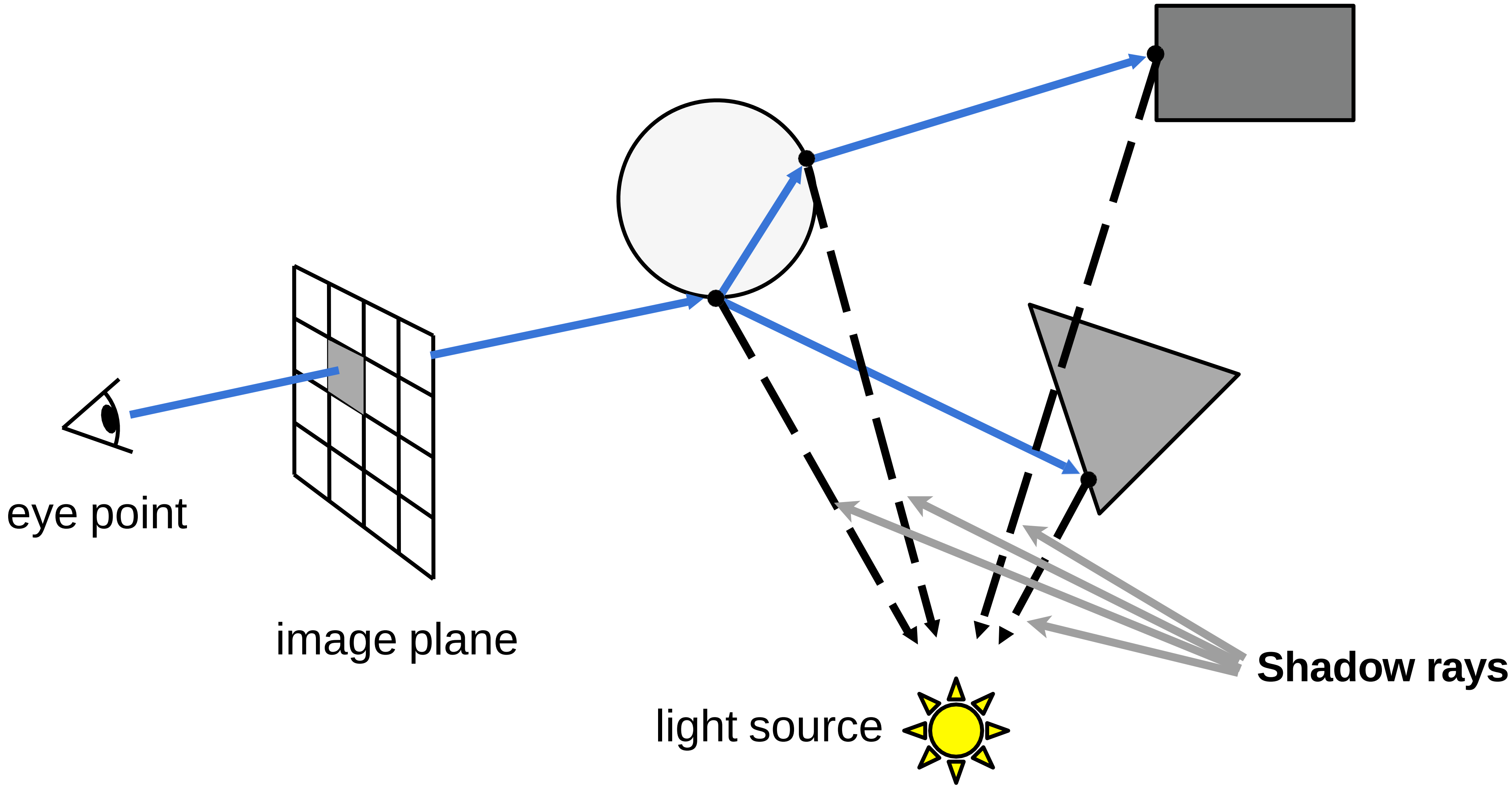
Lancer de rayon récursif



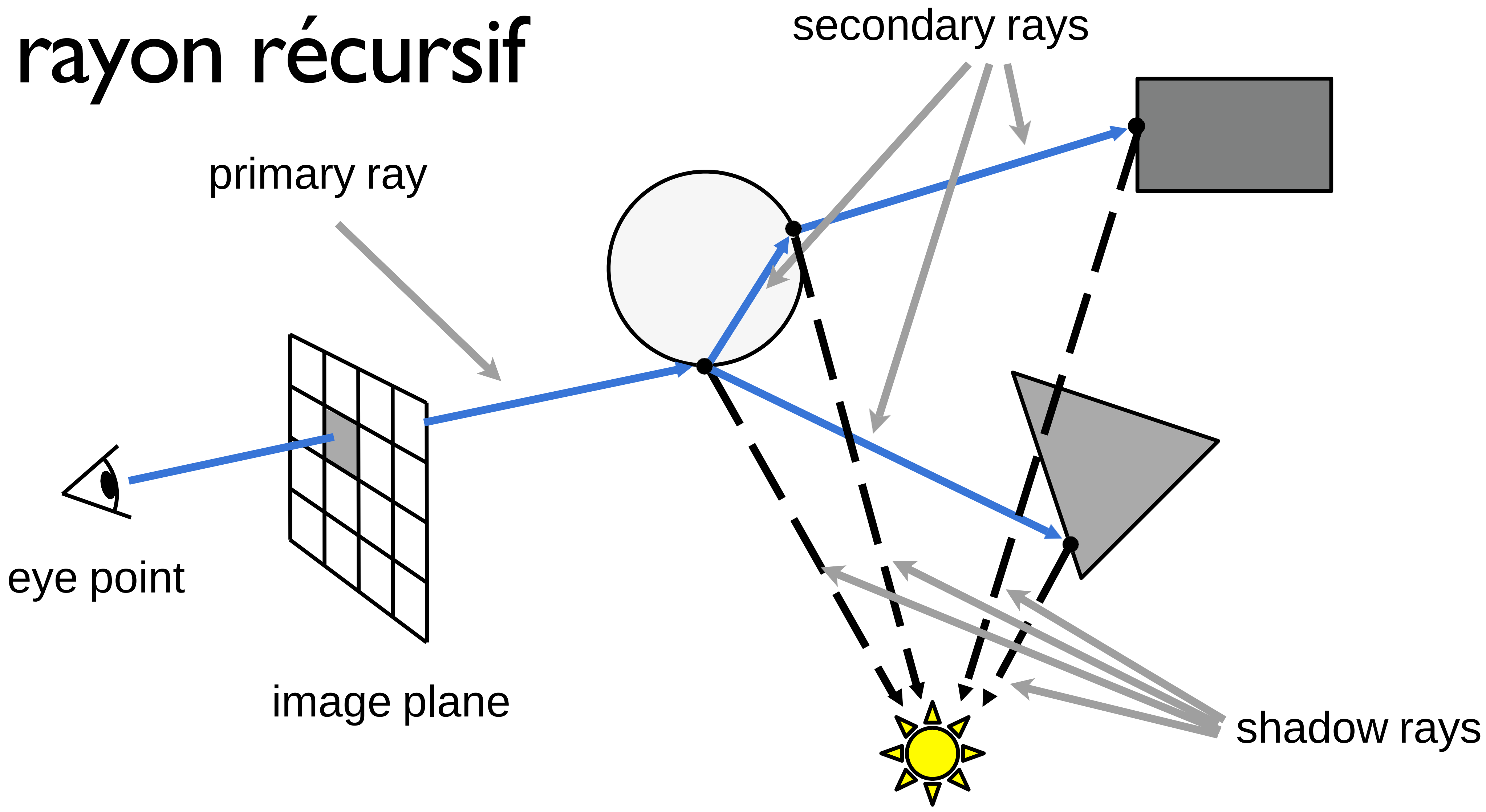
Lancer de rayon récursif



Lancer de rayon récursif

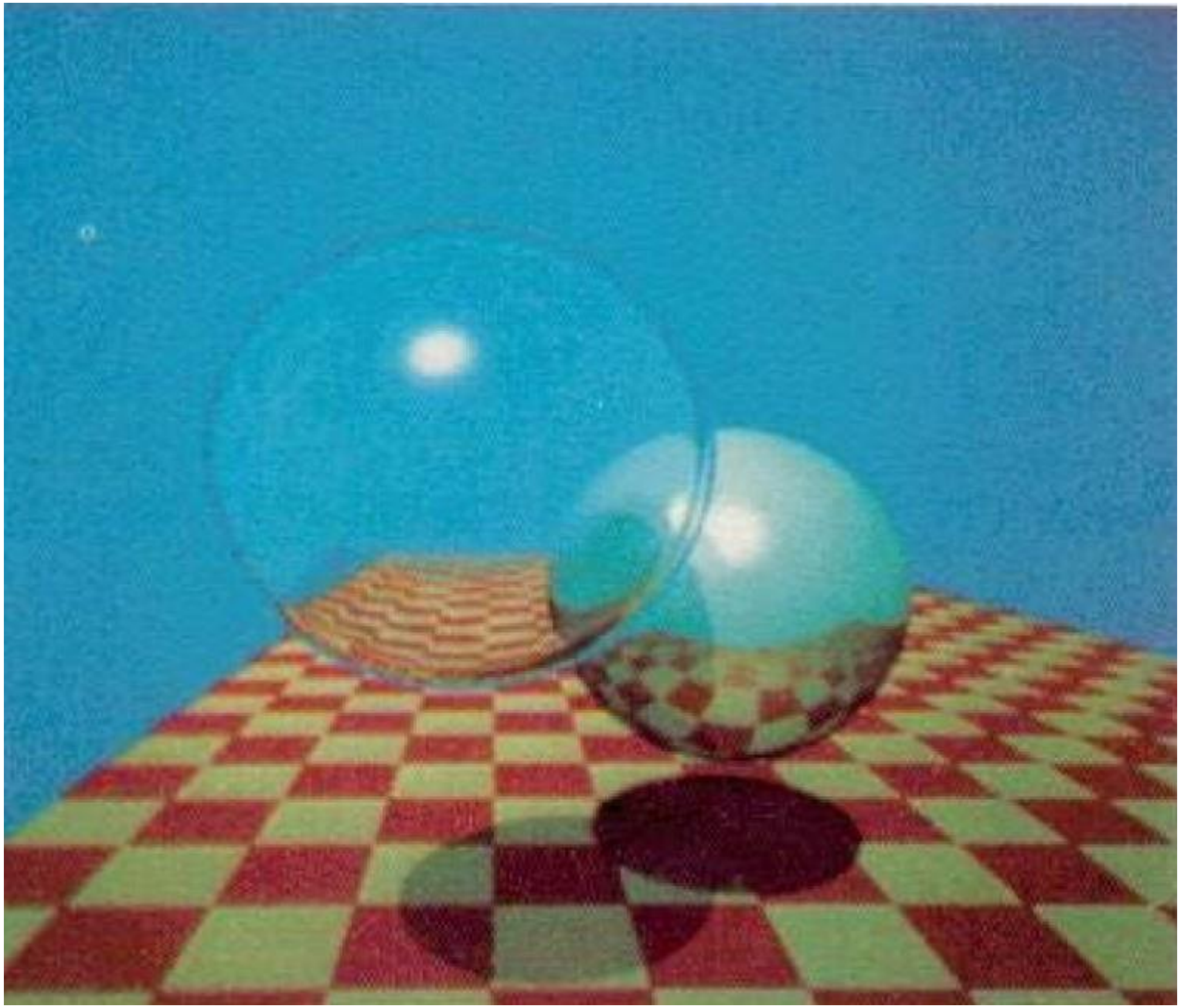


Lancer de rayon récursif

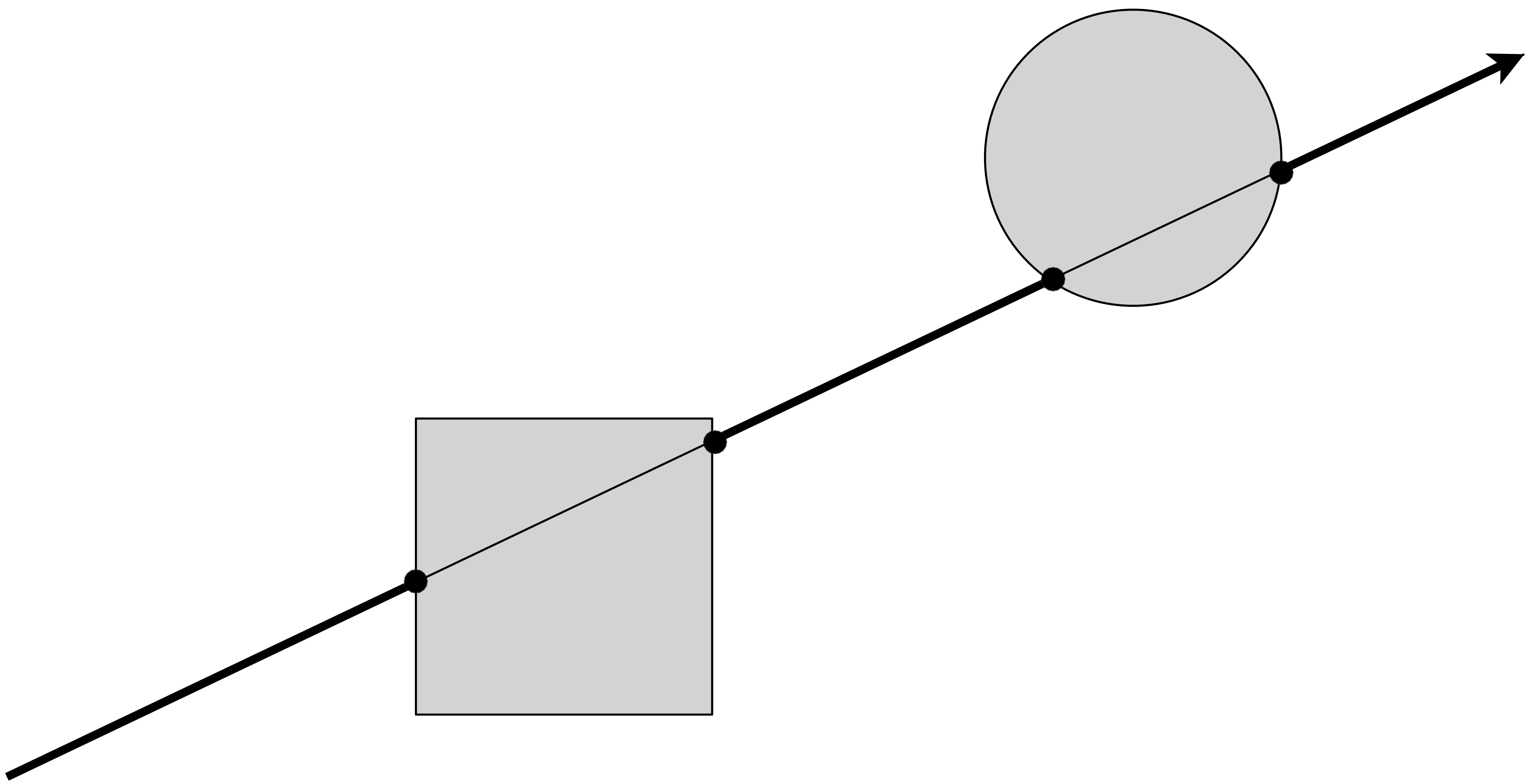


- Tracer les rayons secondaires de manière récursive jusqu'à ce qu'ils atteignent une surface non spéculaire (ou les niveaux de récursion maximum souhaités)
- À chaque point d'impact, tracez des rayons d'ombre pour tester la visibilité de la lumière (aucune contribution si bloqué)
- La couleur finale des pixels est la somme pondérée des contributions le long des rayons
- Donne des effets plus sophistiqués (par exemple réflexion spéculaire, réfraction, ombres), mais nous irons beaucoup plus loin pour dériver un modèle d'éclairage basé sur la physique

Recursive Ray Tracing



Intersection de rayon

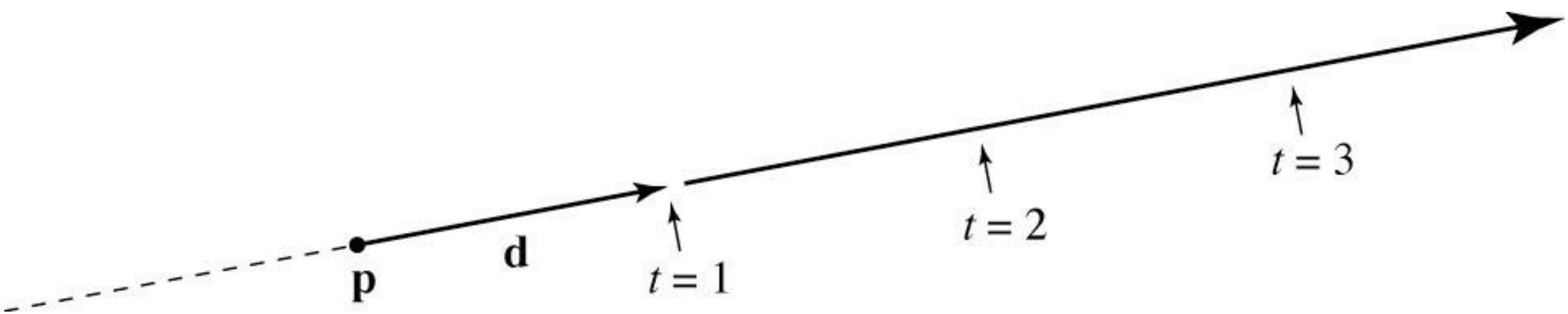


Rayon : une demi-droite

- **Représentation standard : point p et direction d**

  $$\mathbf{r}(t) = \mathbf{p} + t\mathbf{d}$$

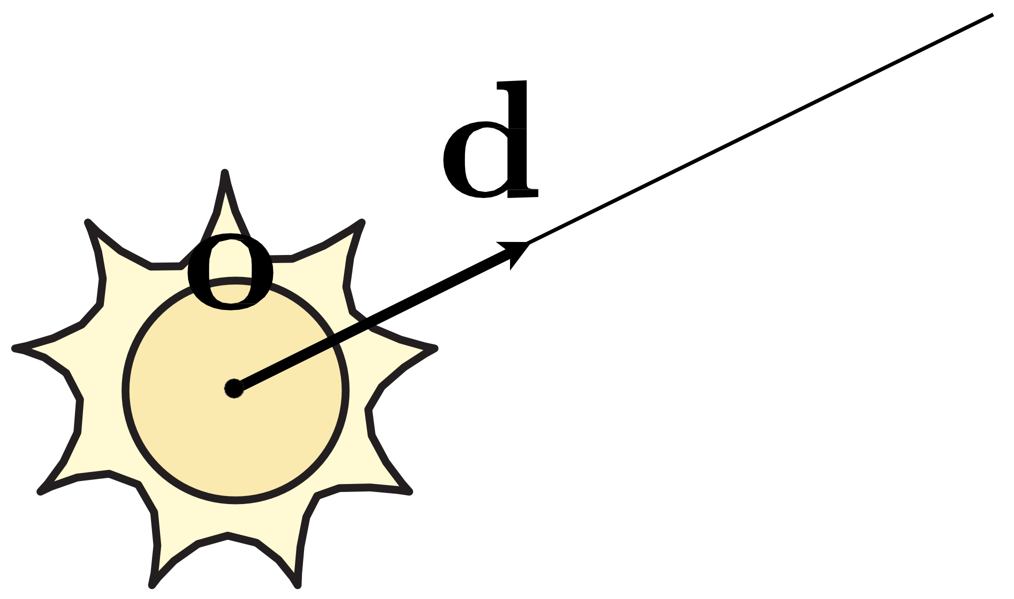
  – C'est une *équation paramétrique* d'une droite
  – Permet de générer des points directement sur la droite
  – Rayon en ajoutant la contrainte $t > 0$
  – Remarque, remplacer $\mathbf{d}$ par $\alpha\mathbf{d}$ ne change pas le rayon ($\alpha > 0$)



Equation du rayon

Rayon définit par son origine et son vecteur de direction

Example:



Equation :

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \qquad 0 \leq t < \infty$$

Point sur le rayon ($\mathbf{r}$)

"temps" (t)

origine ($\mathbf{o}$)

Direction unitaire ($\mathbf{d}$)

Intersection rayon-scène

Rayon : $\mathbf{r}(t) = \mathbf{o} + t\,\mathbf{d}$

$\mathbf{o} = (o_x, o_y, o_z)$, $\mathbf{d} = (d_x, d_y, d_z)$

$\Longrightarrow$ équation explicite

Sphère : $\|\mathbf{p} - \mathbf{c}\|^2 - r^2 = 0$

$\mathbf{c}$: centre de la sphère, r : rayon de la sphère

$\Longrightarrow$ équation implicite

Plan : $(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n} = 0$

$\mathbf{n}$: normale à la surface, $\mathbf{a}$: un point sur le plan

$\Longrightarrow$ équation implicite

Triangle : partie d'un plan

Intersection rayon-sphère

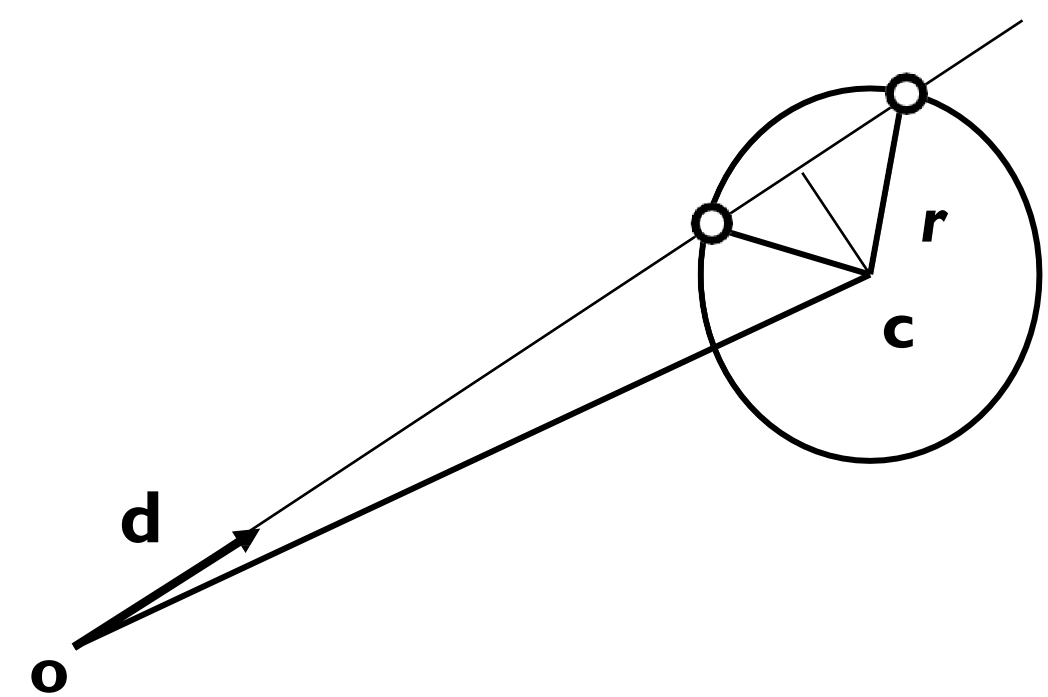
Rayon : $\mathbf{r}(t) = \mathbf{o} + t\,\mathbf{d}$

$o = (o_x, o_y, o_z)$, $d = (d_x, d_y, d_z)$

Sphère : $\|\mathbf{p} - \mathbf{c}\|^2 - r^2 = 0$

$\mathbf{c}$: centre de la sphère, r : rayon de la sphère

Point d'intersection ?



Intersection rayon-sphère

Étant donnée l'équation de la sphère : $\|\mathbf{x}-\mathbf{c}\|^2 - r^2 = 0$

c : centre de la sphère, *r* :rayon de la sphère

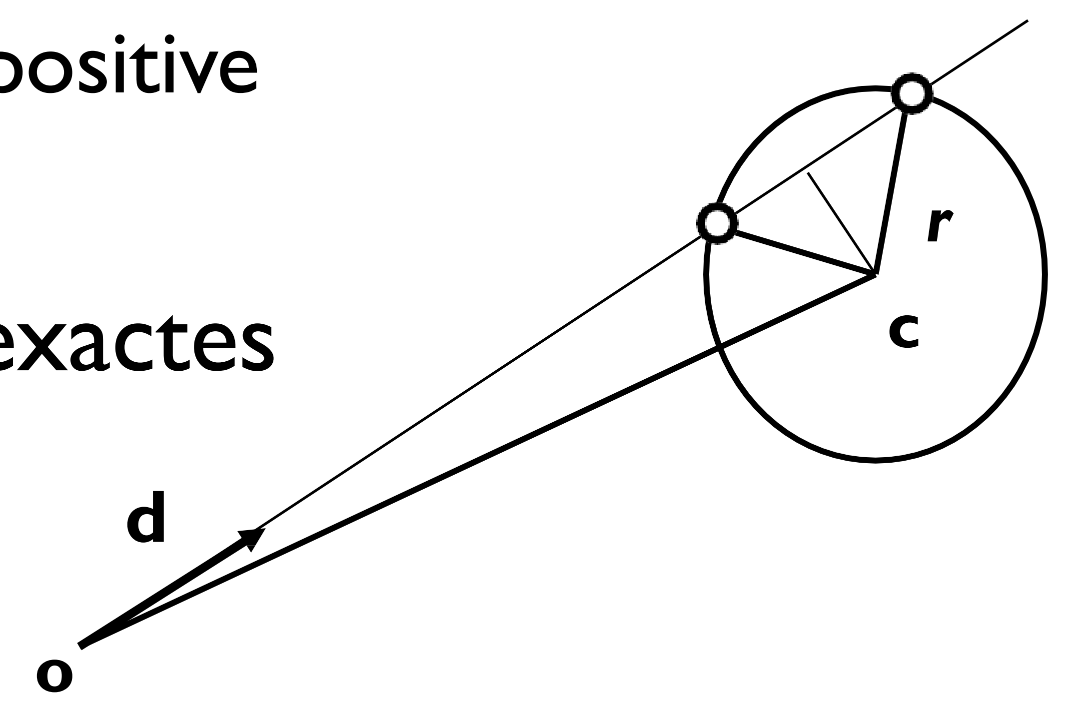
Remplacer **x par l'équation du rayon :**

$$t^2\, \mathbf{d}\cdot\mathbf{d} + 2t\, \mathbf{d}\cdot(\mathbf{o}-\mathbf{c}) + \|\mathbf{o}-\mathbf{c}\|^2 - r^2 = 0$$

équation du second degré en $\boldsymbol{t}$

- discriminant négatif = pas d'intersection
- 2 racines : garder la plus proche positive (généralement $\boldsymbol{t}_{-}$)

⟹ Permet d'obtenir des sphères exactes



Intersection rayon-plan

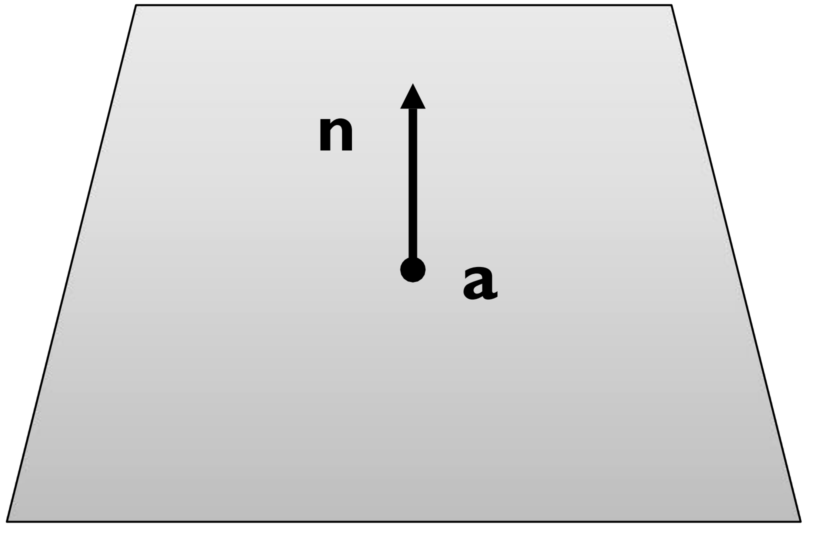
Rayon : $\mathbf{r}(t) = \mathbf{o} + t\,\mathbf{d}$

$\mathbf{o} = (o_x, o_y, o_z), \quad \mathbf{d} = (d_x, d_y, d_z)$

Plan : $(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n} = 0, \ ||n|| = 1$

$\mathbf{n}$: normale à la surface, $\mathbf{a}$: un point sur le plan

Point d'intersection ?



Intersection rayon-plan

Équation du plan : $\boldsymbol{x}.\boldsymbol{n} - D = 0$, $\|\boldsymbol{n}\| = 1$

Normale : $\boldsymbol{n}$

- Distance du plan au centre $(0,0,0)$: $D = \boldsymbol{a}.\boldsymbol{n}$
- Remplacer $\boldsymbol{x}$ **par l'équation du rayon :**

$$(\mathbf{o} + t\mathbf{d}) \cdot \mathbf{n} - \mathrm{D} \quad = 0$$

La solution devient :

$$t = \frac{D - \boldsymbol{o}.\boldsymbol{n}}{\boldsymbol{d}.\boldsymbol{n}}$$

4 cas:

- t infini $\Longrightarrow$ rayon parallèle et distinct du plan
- t non-défini $\Longrightarrow$ rayon confondu avec le plan
- $t < 0 \Longrightarrow$ intersection derrière la caméra
- $t > 0 \Longrightarrow$ intersection devant la caméra

Intersection rayon-triangle

- **Condition 1 : le point est sur le rayon**

$$\mathbf{r}(t) = \mathbf{p} + t\mathbf{d}$$

- **Condition 2 : le point est sur le plan**

$$(\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} = 0$$

- **Condition 3 : le point est coté intérieur des 3 arêtes**

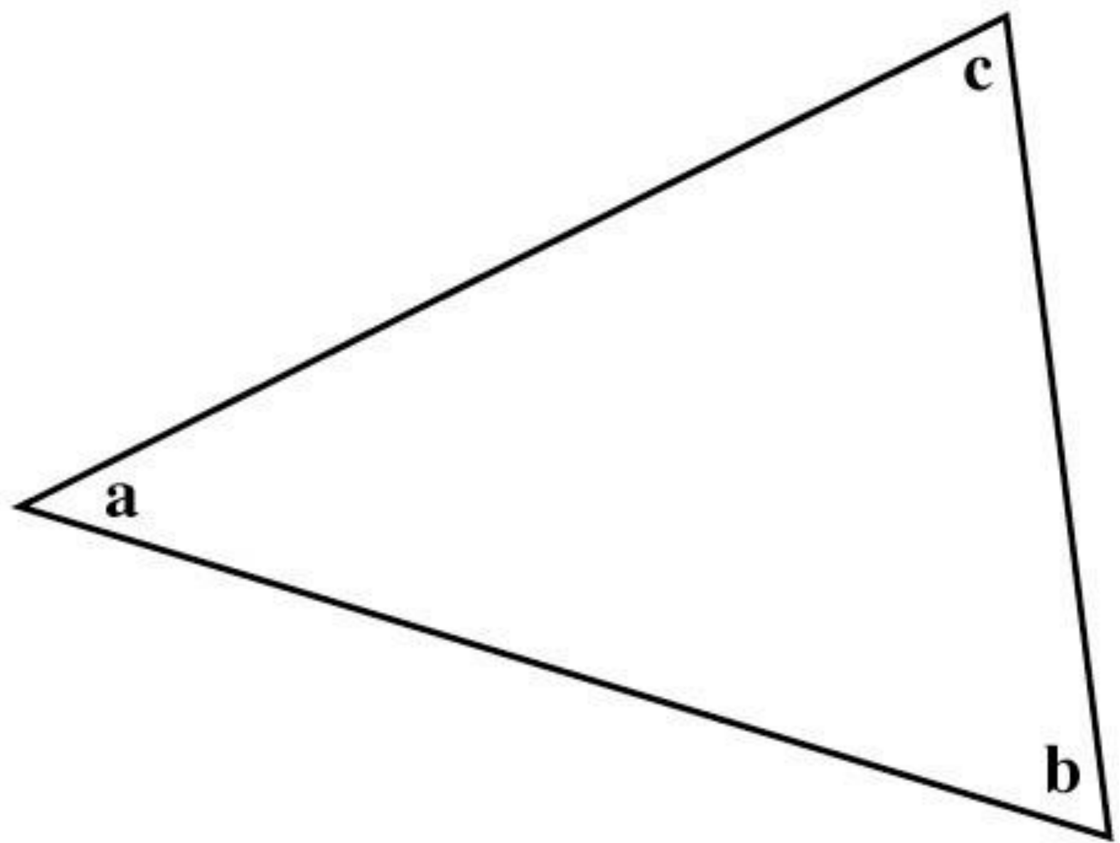
- **Résoudre1&2 (intersection rayon plan)**
  - substituer et résoudre pour *t* :

$$(\mathbf{p} + t\mathbf{d} - \mathbf{a}) \cdot \mathbf{n} = 0$$

$$t = \frac{(\mathbf{a} - \mathbf{p}) \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}}$$

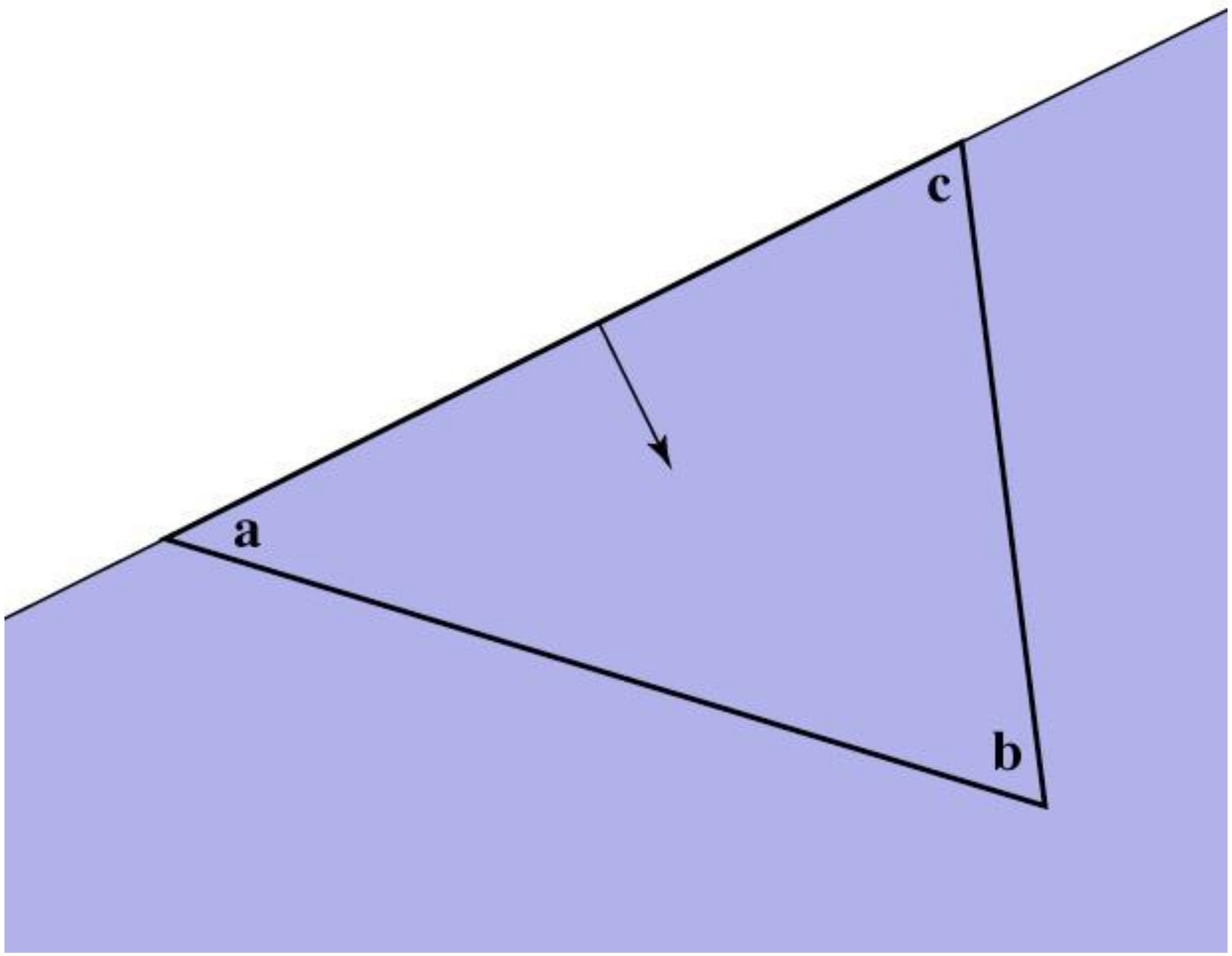
Intersection rayon-triangle

- **Dans le plan, le triangle est l'intersection de 3 demi-espaces**



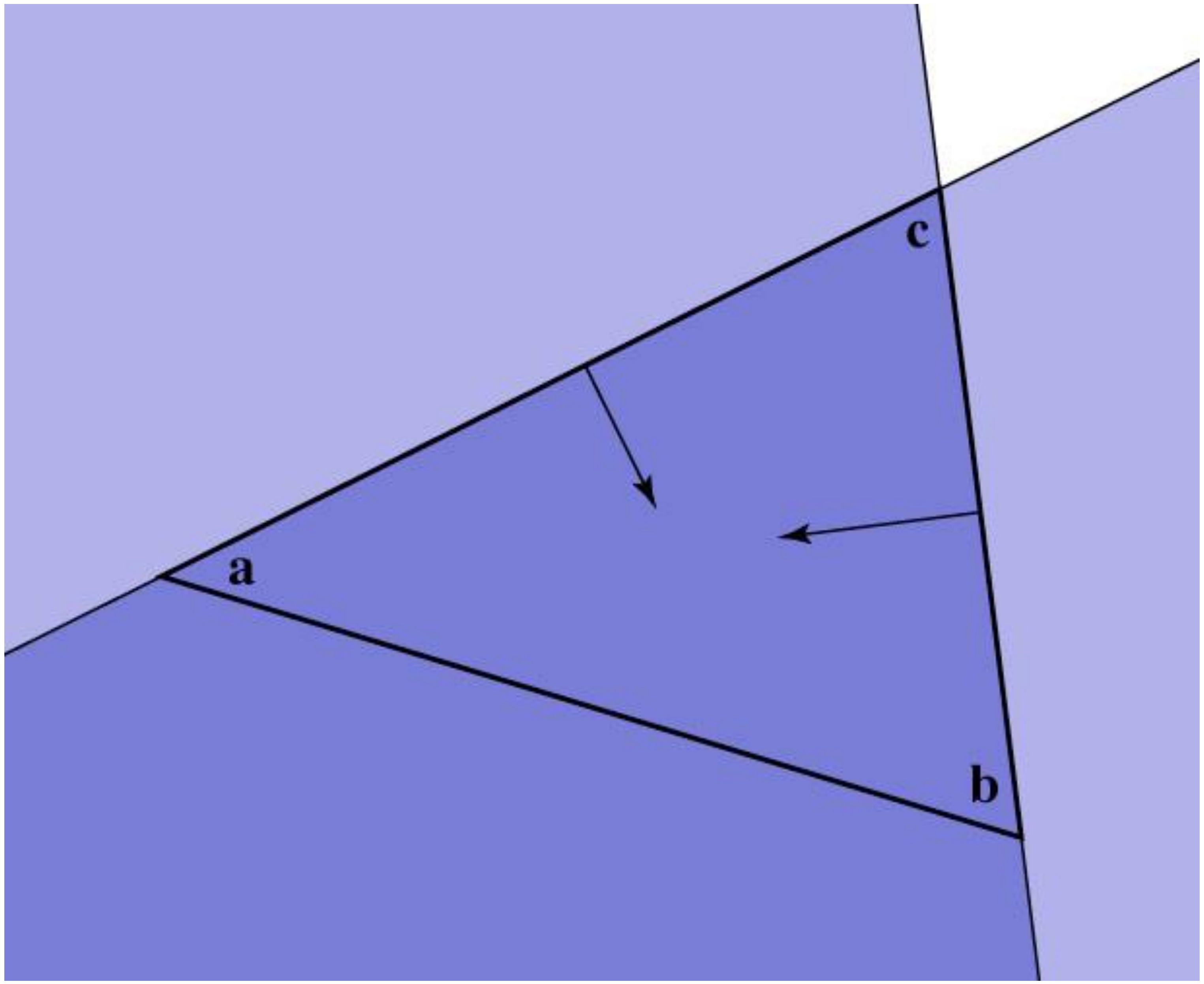
Intersection rayon-triangle

- **Dans le plan, le triangle est l'intersection de 3 demi-espaces**



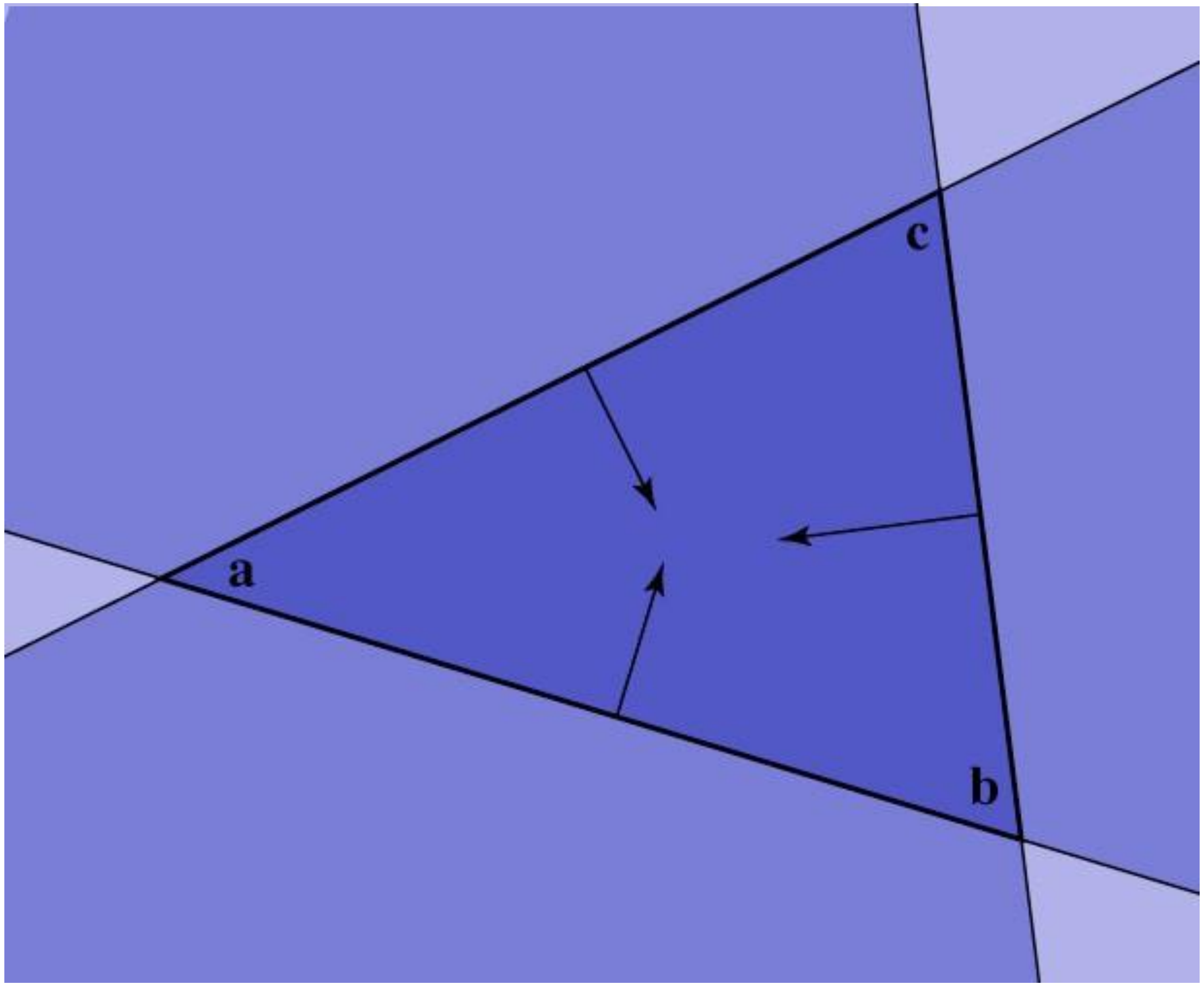
Intersection rayon-triangle

- **Dans le plan, le triangle est l'intersection de 3 demi-espaces**



Intersection rayon-triangle

- **Dans le plan, le triangle est l'intersection de 3 demi-espaces**



Trouver l'intersection

- **Vérifier que le point d'intersection est à l'intérieur des 3 arêtes**
  - Plus facile à faire en coordonnées 2D sur le plan
- **Nous aurons besoin de l'information de position à l'intérieur du triangle**
  - Pour les textures, calcul d'éclairage,…
- **Solution efficace : transformer en coordonnées triangle**

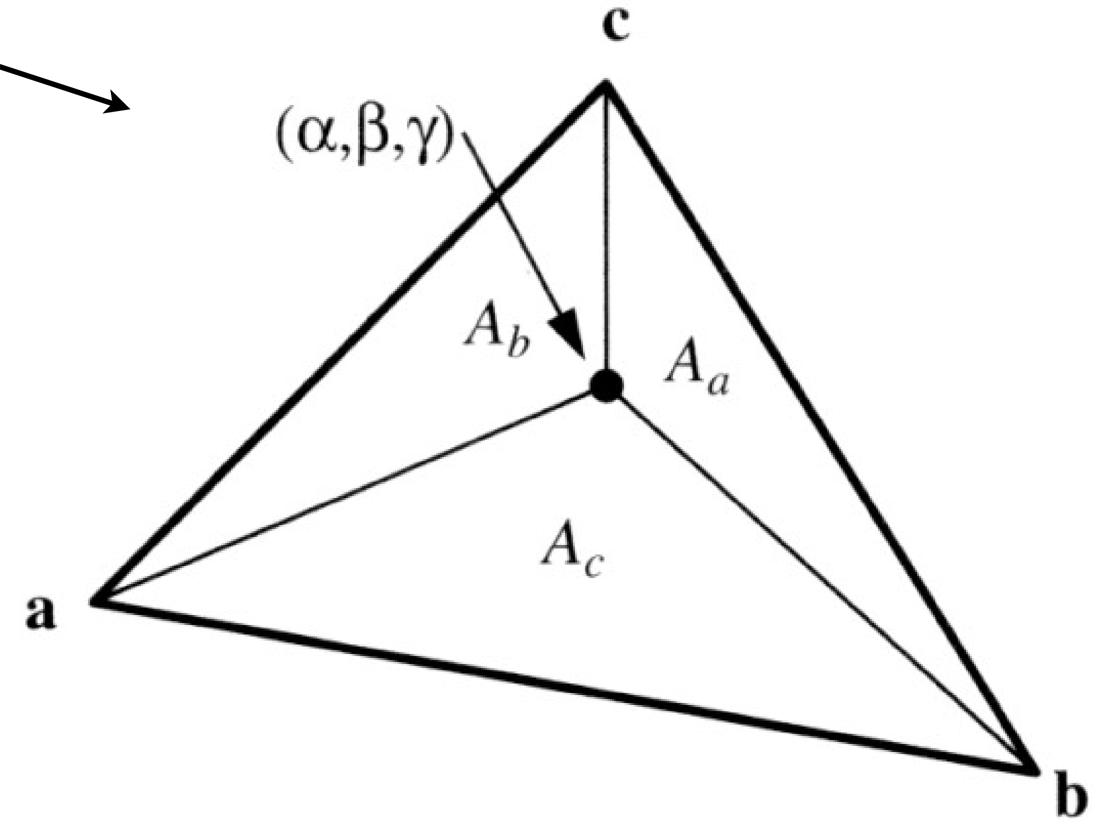
Coordonnées barycentriques

- **Un système de coordonnées pour les triangles**
  - Point de vue algébrique :

$$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$$

$$\alpha + \beta + \gamma = 1$$

  - Point de vue géométrique (aires) :



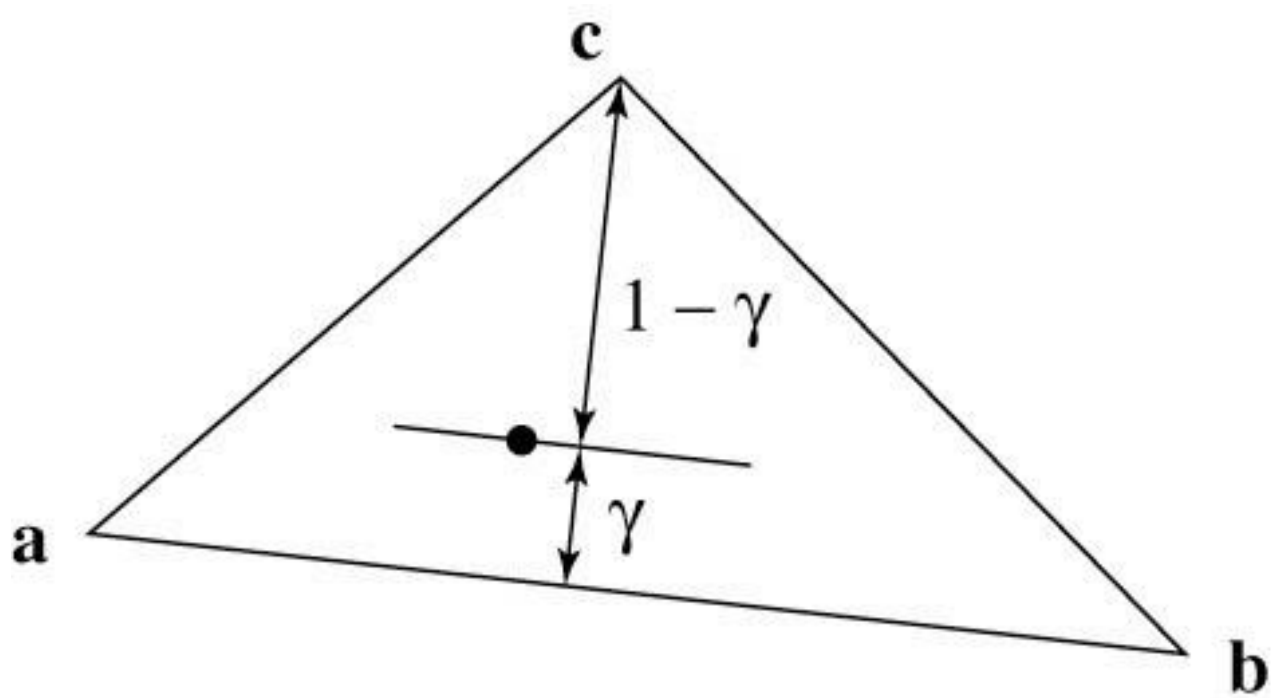
[Shirley 2000]

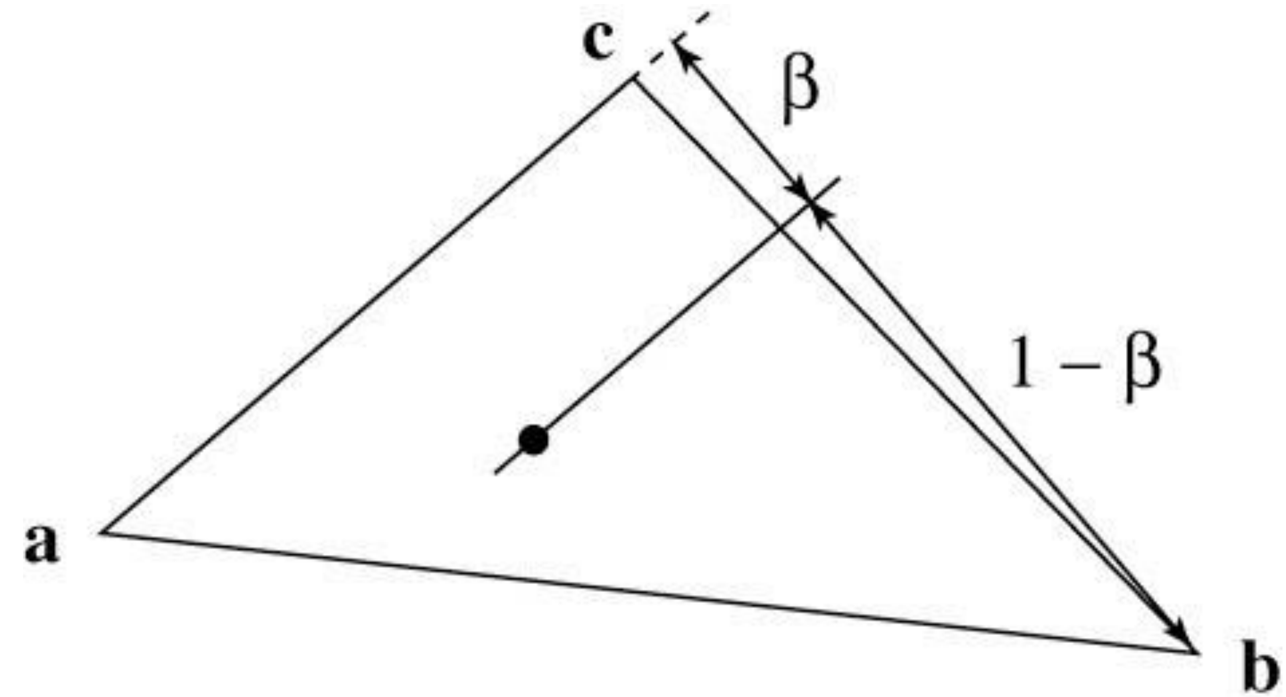
- **Test si intérieur du triangle :**

$$\alpha > 0; \quad \beta > 0; \quad \gamma > 0$$

Coordonnées barycentriques

- **Un système de coordonnées pour les triangles**
  - point de vue géométrique : distances





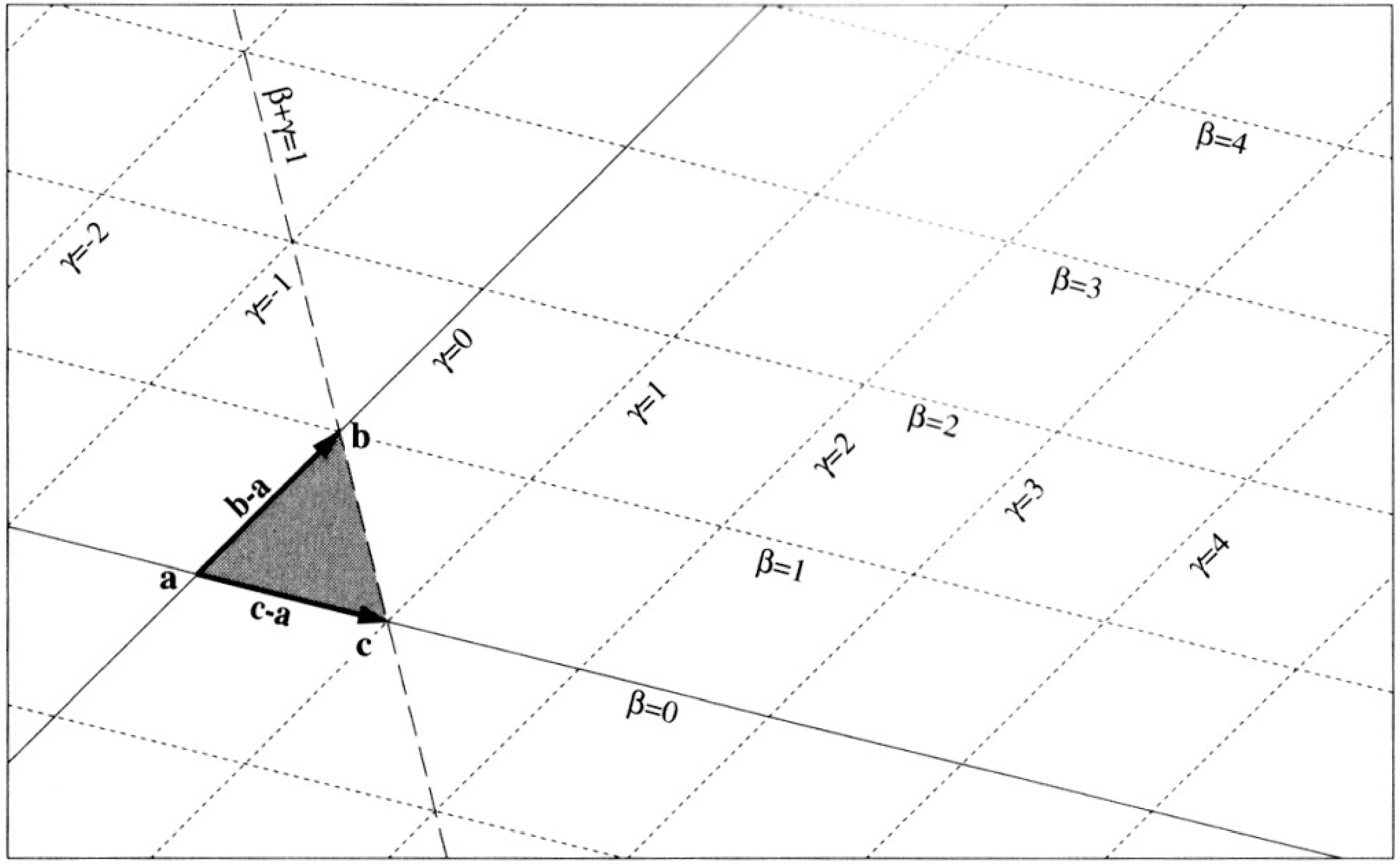
  - point de vue linéaire : base d'arêtes

$$\alpha = 1 - \beta - \gamma$$

$$\mathbf{p} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

Coordonnées barycentriques

- **Point de vue linéaire :bases d'un plan**

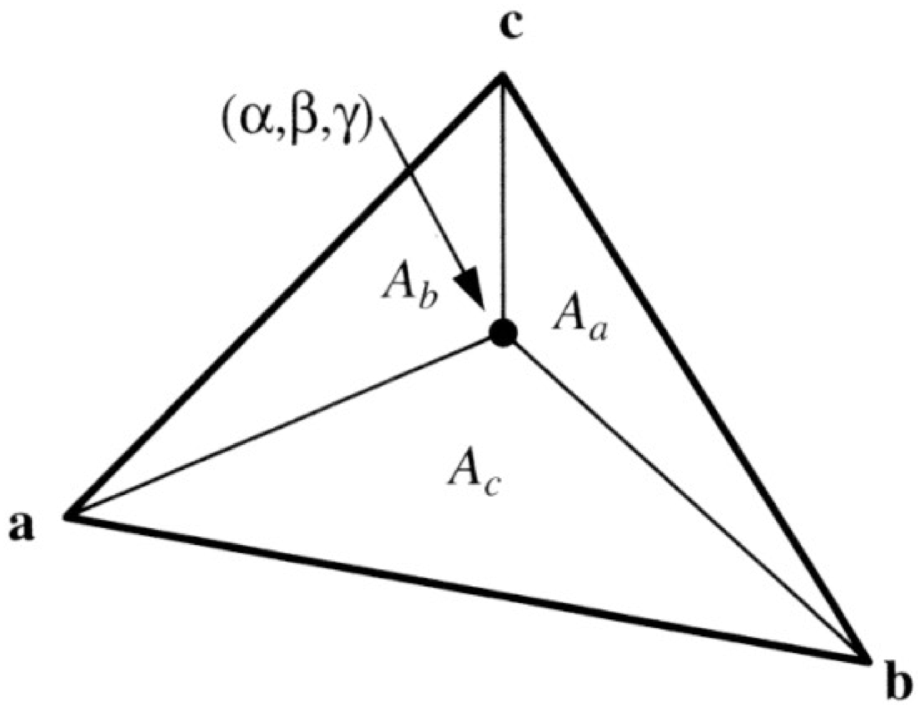


[Shirley 2000]

  – dans cette vue, le triangle test est juste

$$\beta > 0; \quad \gamma > 0; \quad \beta + \gamma < 1$$

Intersection rayon-triangle



Coordonnées barycentriques

- $\alpha + \beta + \gamma$ = 1
- α = Aire(A_a) / Aire(abc)
- β= Aire(A_b) / Aire(abc)
- γ= Aire(A_c) / Aire(abc)

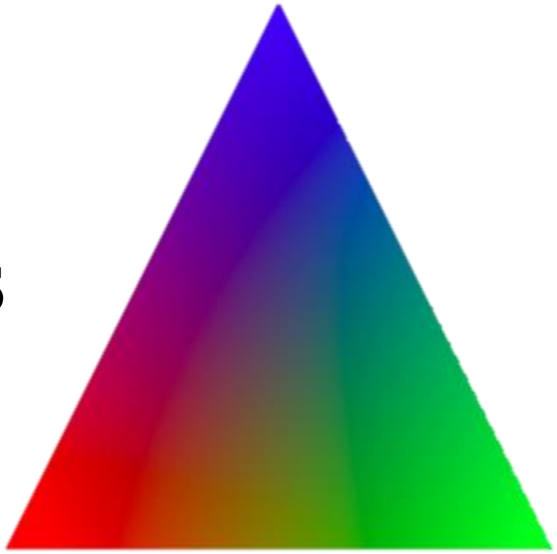
  $\Longrightarrow$ Aire relative signée

Test d'appartenance au polygone

- à l'intérieur si toutes les coordonnées sont >=0

Interpolation des attributs aux sommets

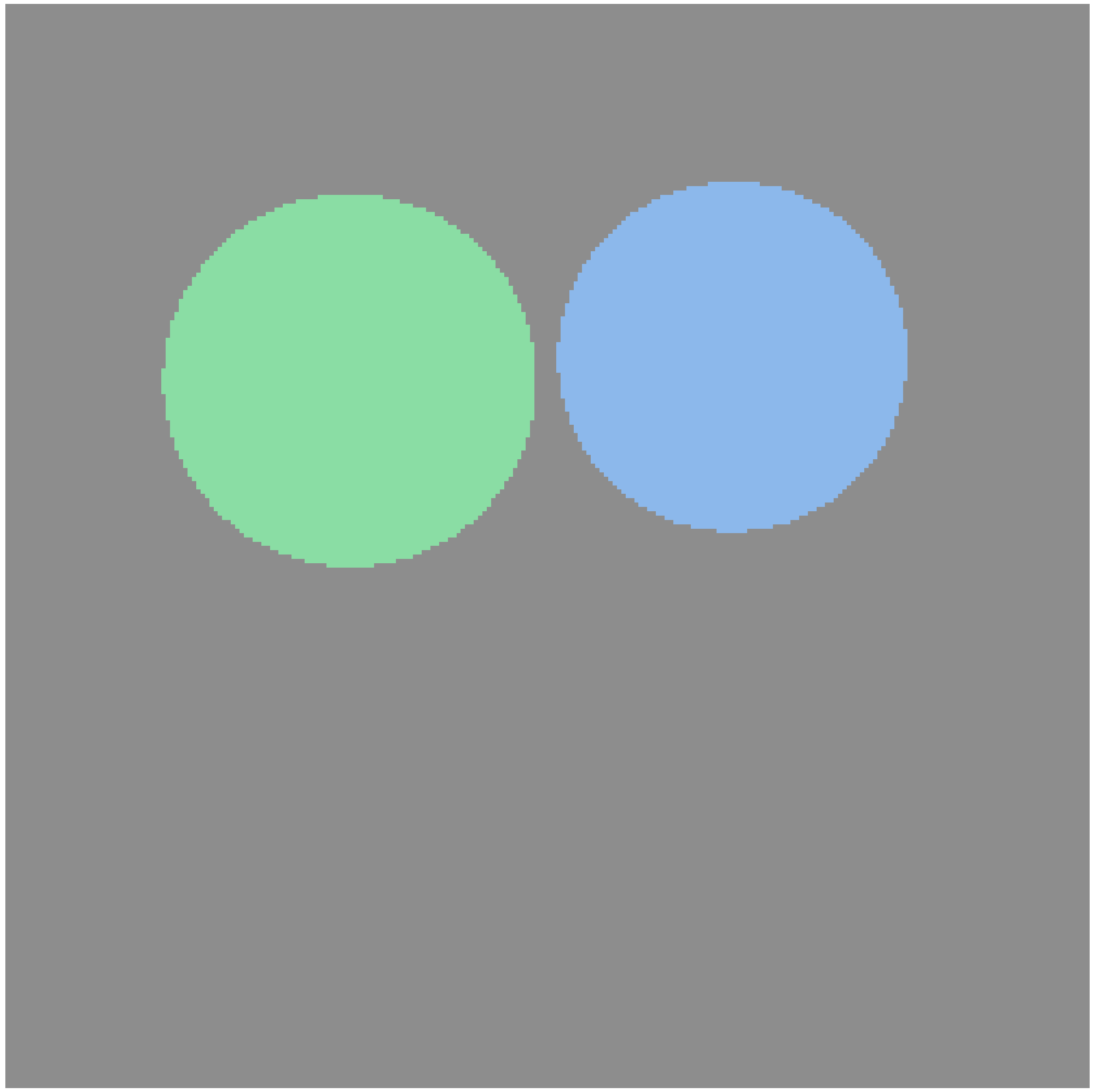
- normales, couleurs, coordonnées de texture, etc.



L'image pour l'instant

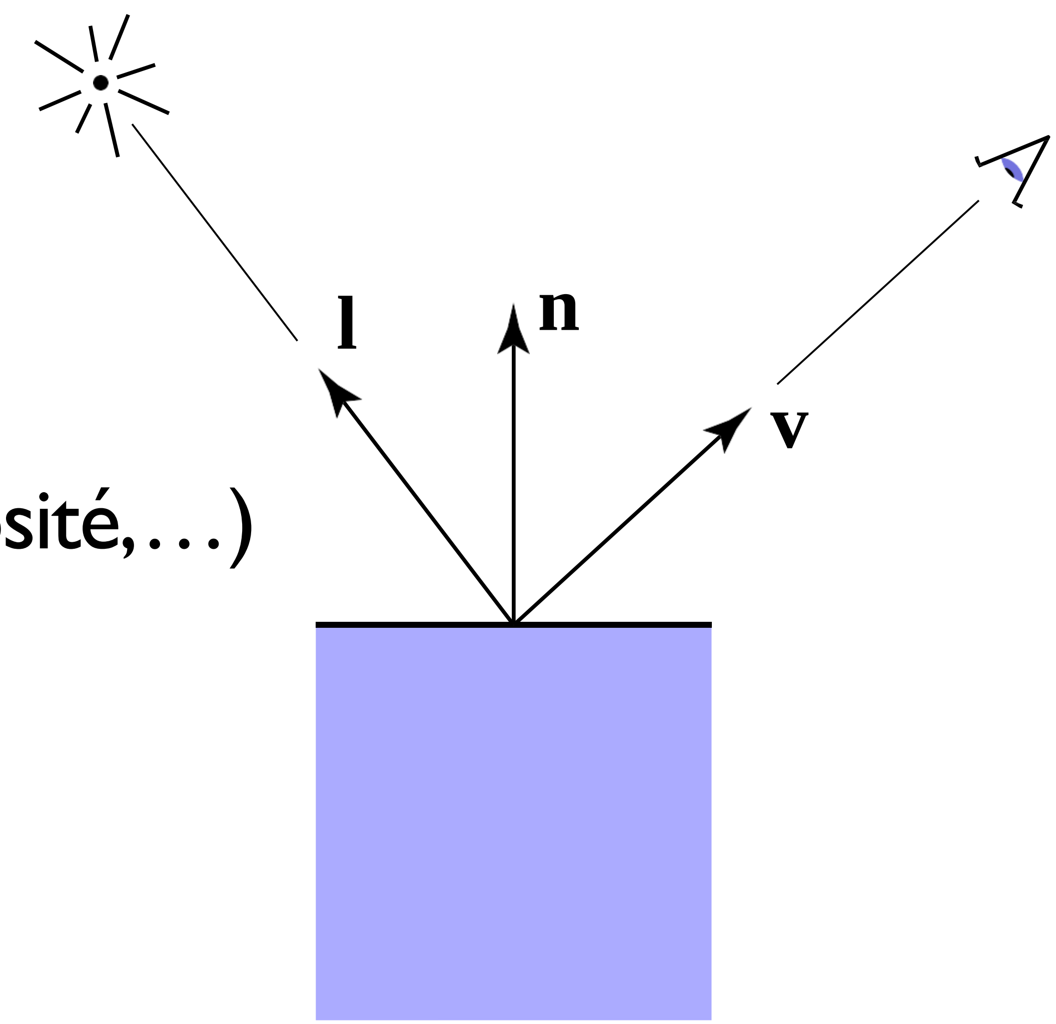
- **Génération de rayon et intersections**

```
for 0 <= iy < ny
    for 0 <= ix < nx {
        ray = camera.getRay(ix, iy);
        c = scene.trace(ray, 0, +inf);
        image.set(ix, iy, c);
    }

…

Scene.trace(ray, tMin, tMax) {
    surface, t = surfs.intersect(ray, tMin, tMax);
    if (surface != null) return surface.color();
    else return black;
}
```



Shading

- **Calculer la lumière réfléchie vers la caméra**
- **Entrée :**
  - Direction de vue
  - Direction de lumière
    (pour chacune des lumières)
  - Normal à la surface
  - Paramètre de la surface (couleur, rugosité,…)



Lumières

- Réalisme dû à la perception par le système visuel humain de l'interaction entre la lumière avec les objets.

- Pas de couleurs ou de rendu 3D sans lumière.

- La manière dont l'objet « réfléchit» la lumière, fait que l'œil et le cerveau « reconstruisent la 3D ».

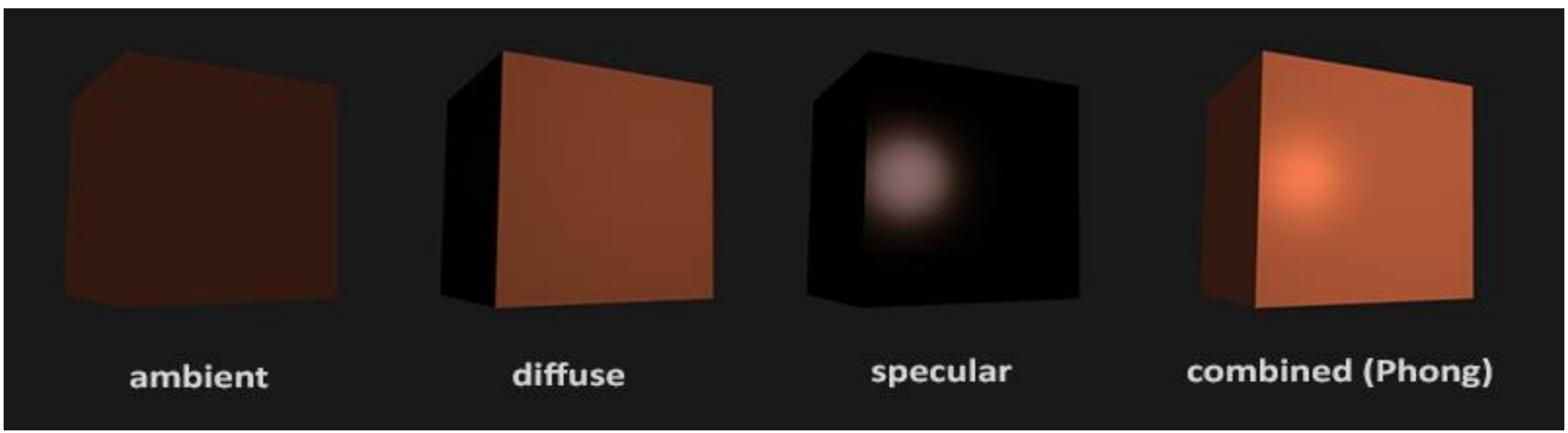
Lumières

- Les matériaux (propriétés physiques) changent les interactions avec la lumière

→ l’aspect visuel de l’objet

Illumination locale

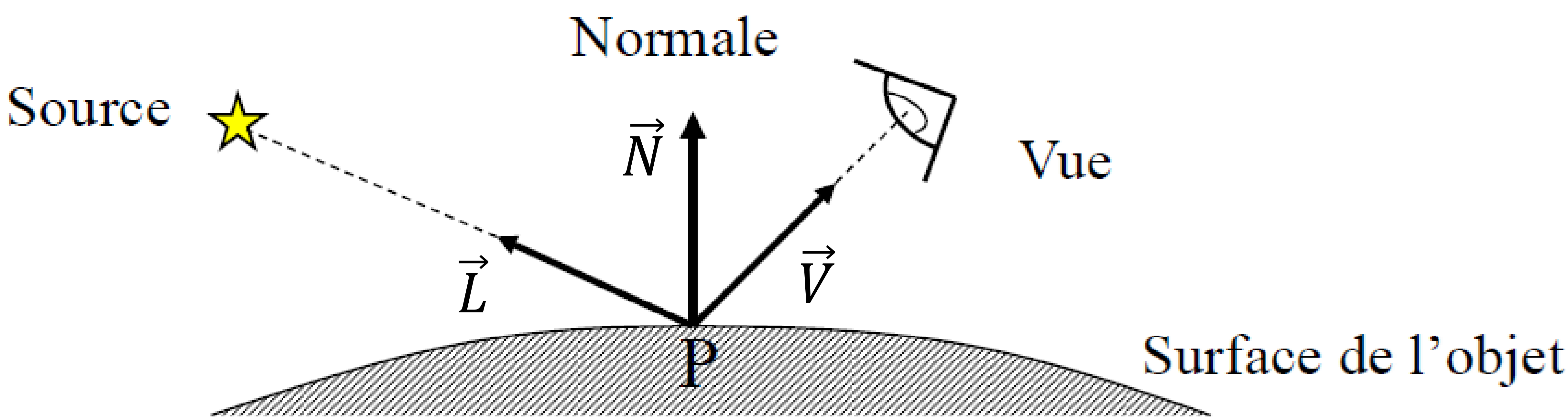
- Théorie : les objets sont vus parce qu'ils réfléchissent la lumière
  - réflexion ambiante
  - réflexion diffuse
  - réflexion spéculaire



En OpenGL : Phong = ambiante + diffuse + speculaire

Ingrédients géométriques

- Pour chaque point P de la surface :
  - Vecteur normal $\vec{N}$
  - Vecteur de direction de vue (camera) $\vec{V}$
  - Vecteur de direction de la source lumineuse $\vec{L}$



Réflexion ambiante

- Parasites provenant d'autre chose que la source considérée
  - lumière réfléchie par d'autres points
  - supposée égale en tout point de l'espace

$$I_a = I_{sa} * K_a$$

- I_a : intensité de la lumière ambiante réfléchie
- I_{sa} : intensité de la lumière ambiante
- $K_a \in [0,1]$: coeff. de réflexion ambiante de l’objet

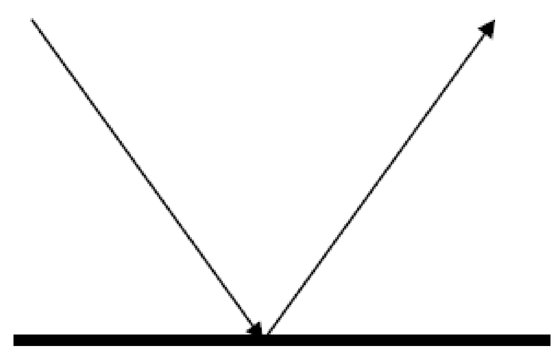
Réflexion ambiante

- Couleur ambiante d'un objet ne dépend que du coefficient de réflexion ambiante Ka de l'objet, pas de sa position par rapport à la lumière

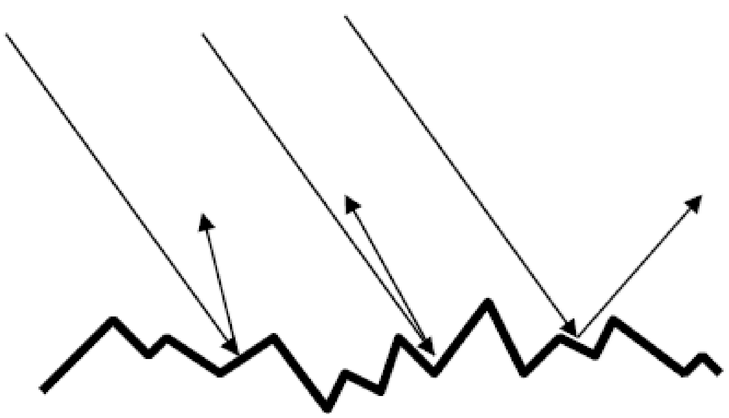
On augmente K_a

Réflexion **diffuse** et **spéculaire**

- La lumière n'est pas réfléchie dans une direction unique mais dans un **ensemble de directions** dépendant des propriétés microscopiques de la surface



Miroir parfait théorique



Surface imparfaite réelle

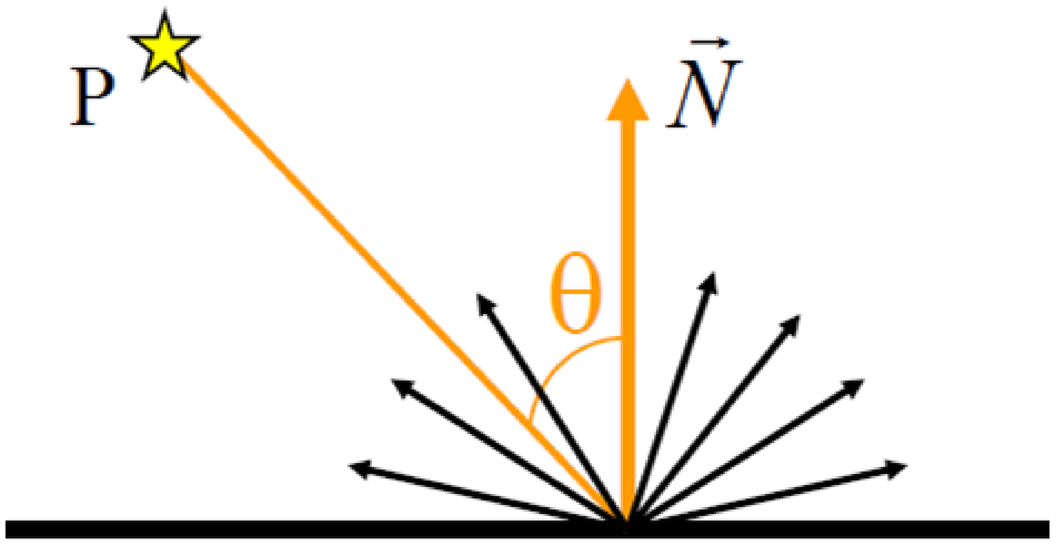
- Directions réparties selon une composante **diffuse** et une composante **spéculaire**, ajoutées à la composante ambiante pour donner plus de relief à l'objet.

Réflexion diffuse

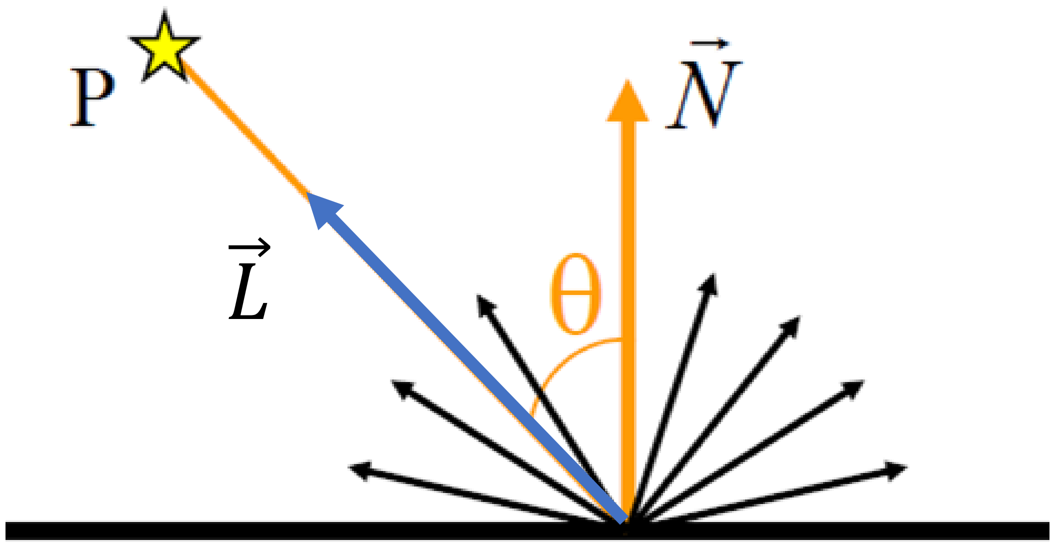
- Lumière réfléchie dans toutes les directions

→ indépendante de la position de l'observateur

- La couleur de l'objet dépend :
  - de l'angle θ entre la direction de la source et la normale
  - du coefficient de réflexion diffuse K_d de l'objet



Réflexion diffuse



- Loi de Lambert

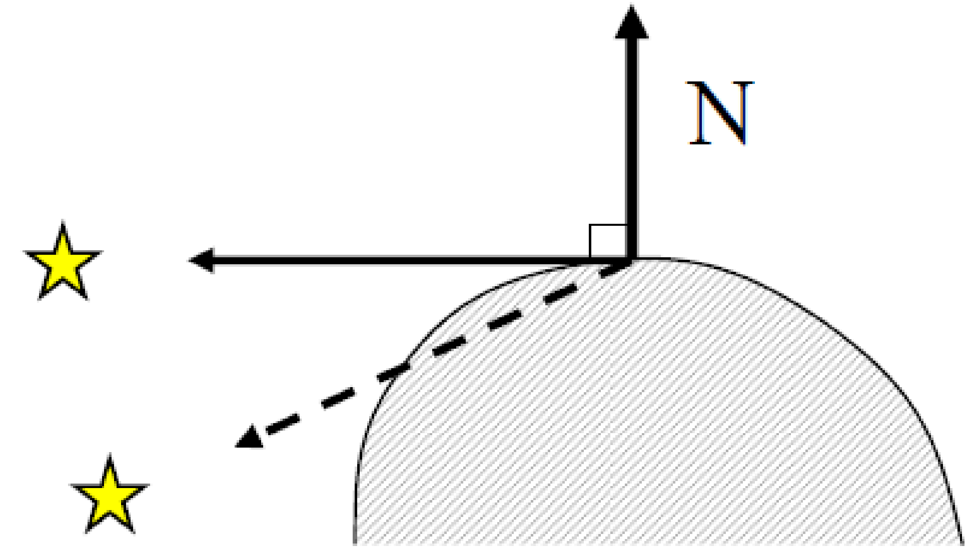
$$I_d = I_{sd} * K_d * \cos\theta$$

- I_d : intensité de la lumière diffuse réfléchie
- I_{sd} : intensité de la lumière diffuse
- $K_d \in [0,1]$: coeff. de réflexion diffuse du matériau
- θ : angle entre la source de lumière et la normale

- On peut également écrire :

$$I_d = I_{sd} * K_d * (\vec{L}.\vec{N})$$

Réflexion diffuse



- Loi de Lambert

$$I_d = I_{sd} * K_d * \cos\theta$$

- Maximale pour $\theta = 0°$ (source de lumière à la verticale de la surface, au zénith)
- Nulle pour un éclairage rasant $\theta = 90°$
- Si $\theta = 90°$ alors le point n'est pas visible par la source de lumière

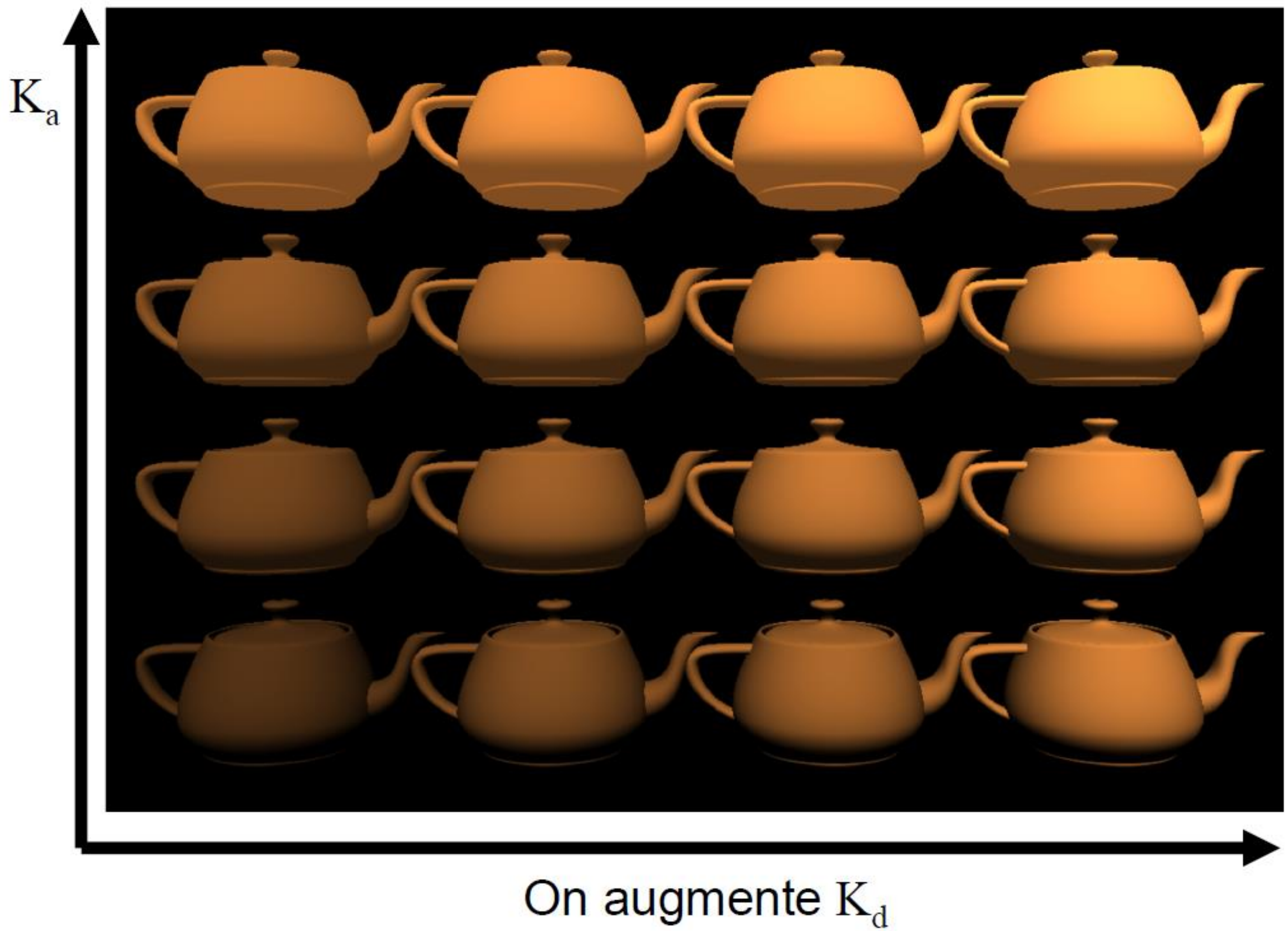
Réflexion diffuse

- Seule

On augmente K_d (avec $K_a = 0$)

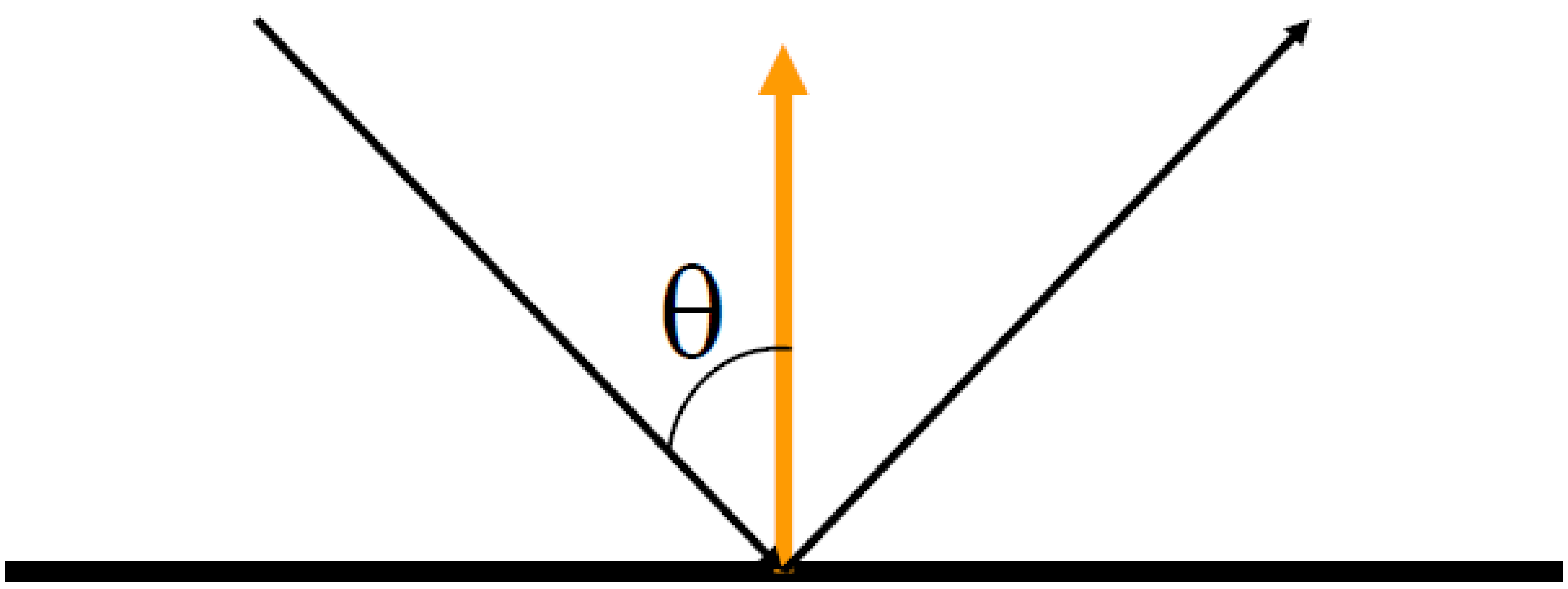
Réflexion diffuse

- Diffuse + ambiante



Réflexion spéculaire

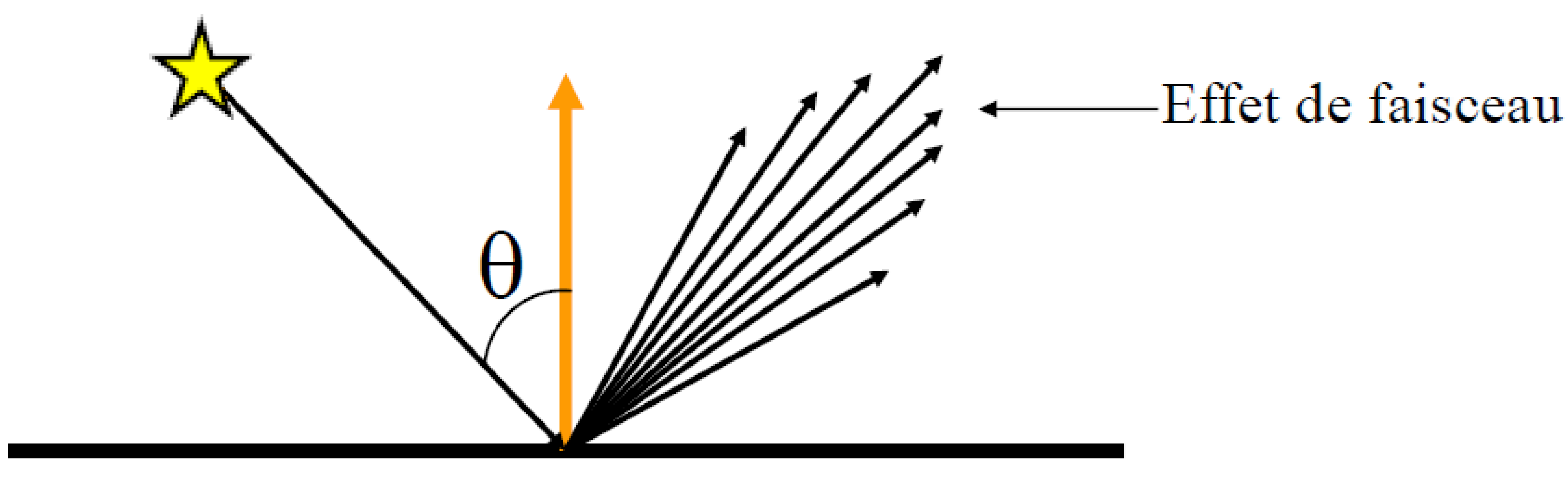
- Permet d'obtenir des reflets
- Miroir parfait → Loi de Descartes
- La lumière qui atteint un objet est réfléchie dans la direction faisant le même angle avec la normale



Réflexion spéculaire

En réalité, les surfaces ne sont jamais des miroirs parfaits

- réflexion spéculaire : miroir imparfait
- la lumière est réfléchie principalement dans la direction de réflexion miroir parfaite
- l'intensité de la lumière réfléchie diminue lorsqu'on s'éloigne de cette direction parfaite.

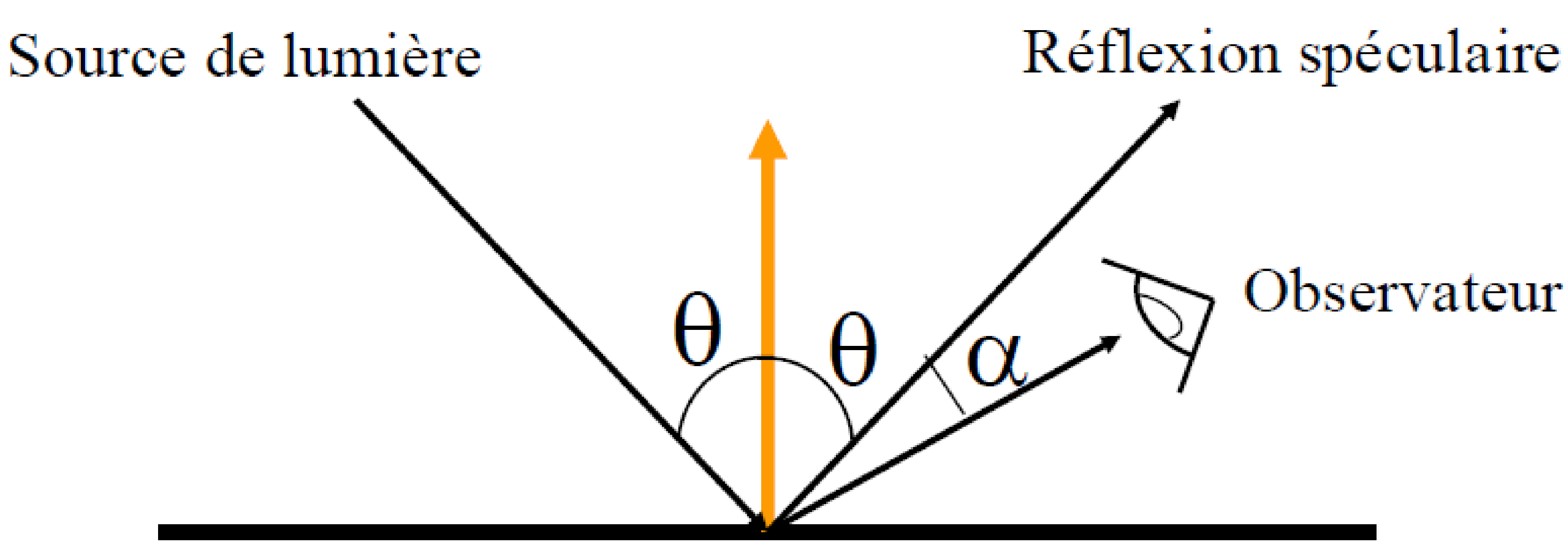


Réflexion spéculaire

- Modèle de Phong

$$I_s = I_{ss} * K_s * \cos\alpha^n$$

- I_s : intensité de la lumière spéculaire réfléchie
- I_{ss} : intensité de la lumière spéculaire de la source
- $K_s \in [0,1]$: coeff. de réflexion spéculaire du matériau
- α : angle entre les directions de réflexion et de la vue



Réflexion spéculaire

- On peut également écrire

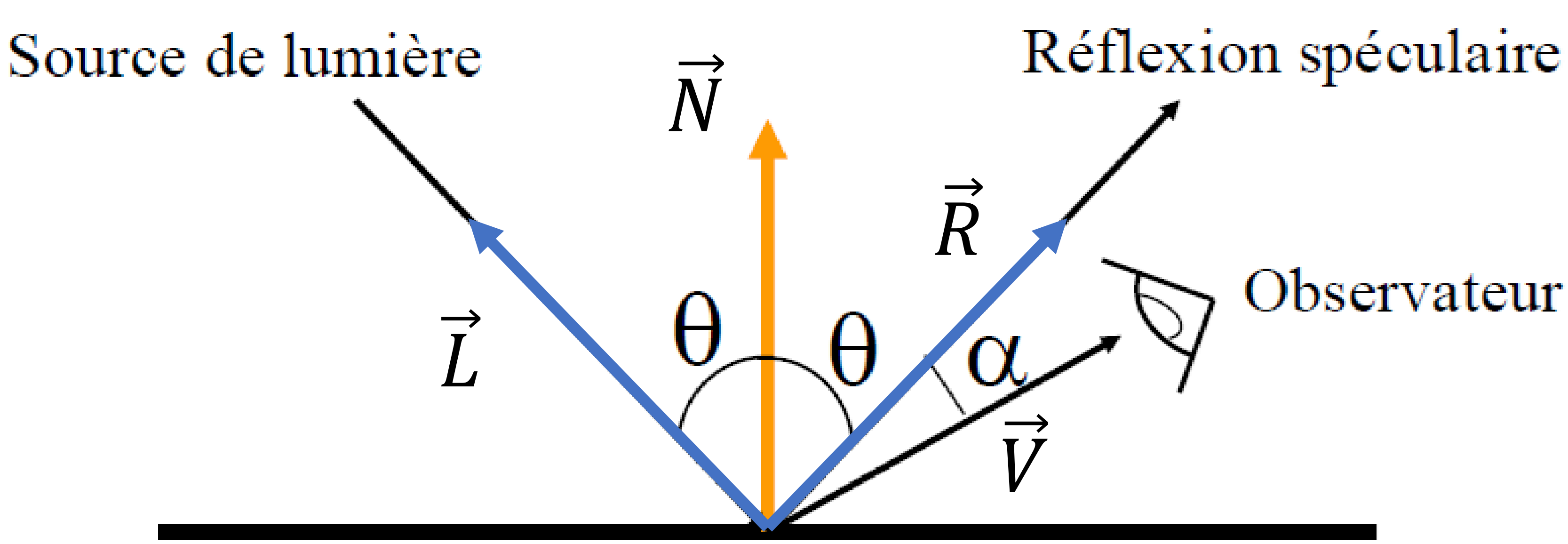
$$I_s = I_{ss} * K_s * \left(\vec{R}.\vec{V}\right)^n$$

avec

$\vec{R}$: direction de réflexion de la lumière sur un miroir

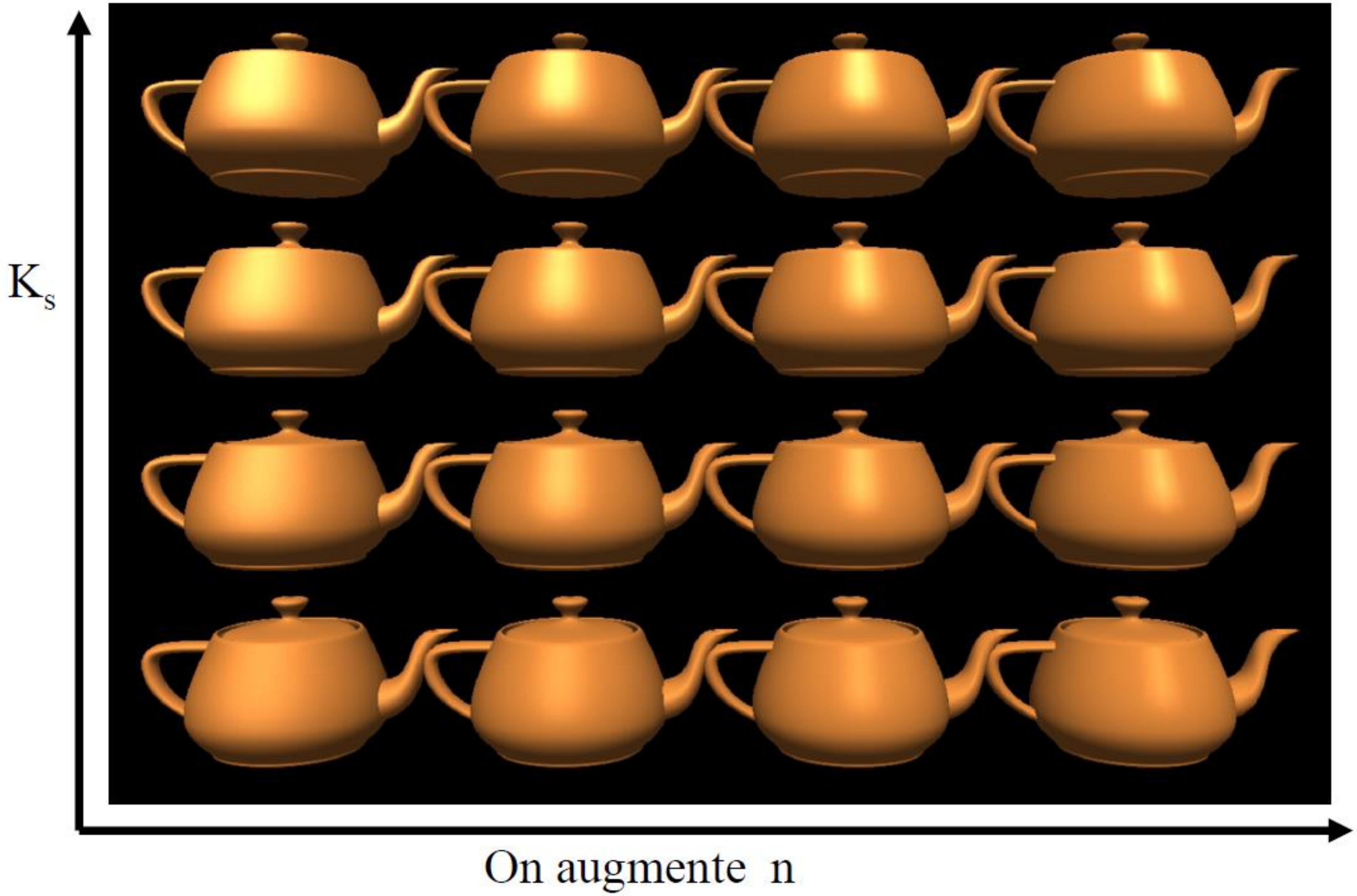
et

$$\vec{R} = 2\left(\vec{N}.\vec{L}\right)\vec{N} - \vec{L} = 2\cos\theta\,\vec{N} - \vec{L}$$



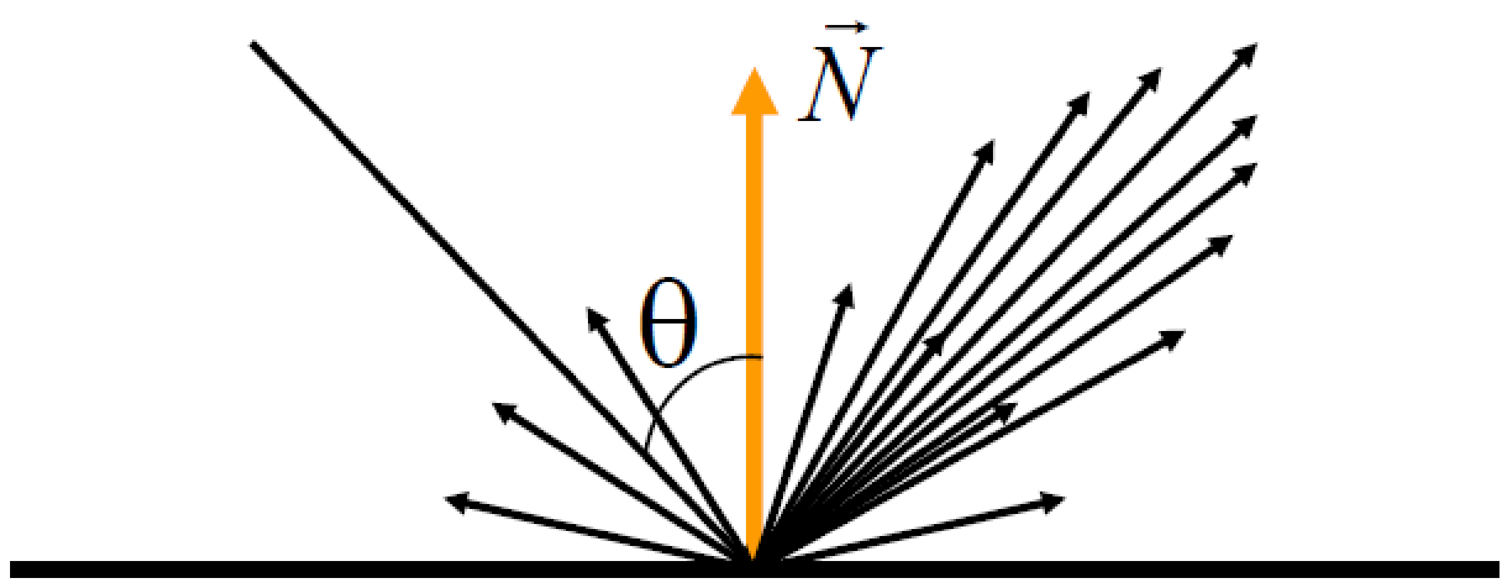
Réflexion spéculaire

$$I_s = \ I_{ss} * K_s * \cos\alpha^n$$

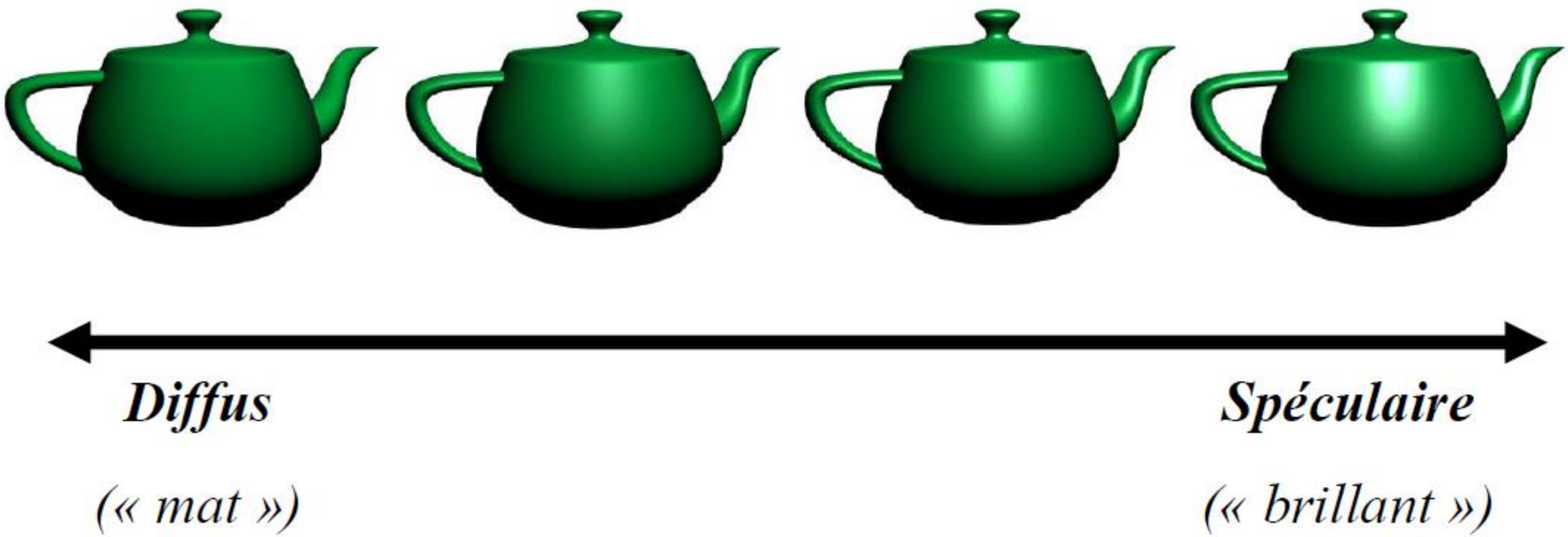


Modèle de Phong

- Dans la réalité, lumière réfléchie par une surface
  - réflexion diffuse + réflexion spéculaire
- Proportions de réflexion diffuse et spéculaire dépendent du matériau :
  - plus diffus (craie, papier …)
  - que spéculaires (métal, verre …)



Réflexion finale



Equation de Phong

- Egale à la somme des réflexions ambiante, diffuse et spéculaire :

$$I = I_a + I_d + I_s$$

- Plusieurs sources lumineuses : somme des intensités

$$I = I_{sa}K_a + \sum_{l \in \{sources\}} I_{ld}K_d(\overrightarrow{L_l}.\vec{N}) + I_{ls}K_s(\overrightarrow{R_l}.\vec{V})^n$$

Calcul de la couleur

- Addition l’intensité lumineuse de chacune des composantes de la couleur.
- RVB : intensités rouge, verte et bleue

→ On définit pour chacune de ces 3 composantes
  - les caractéristiques des sources de lumière
    $(I_{saR}, I_{saG}, I_{saB}); (I_{sdR}, I_{sdG}, I_{sdB}); (I_{ssR}, I_{ssG}, I_{ssB})$
  - les caractéristiques des matériaux
    $(K_{aR}, K_{aG}, K_{aB}); (K_{dR}, K_{dG}, K_{dB}); (K_{sR}, K_{sG}, K_{sB})$

Calcul de la couleur

- Les intensités lumineuses pour chacune des 3 composantes R, V, B s'obtiennent donc ainsi :

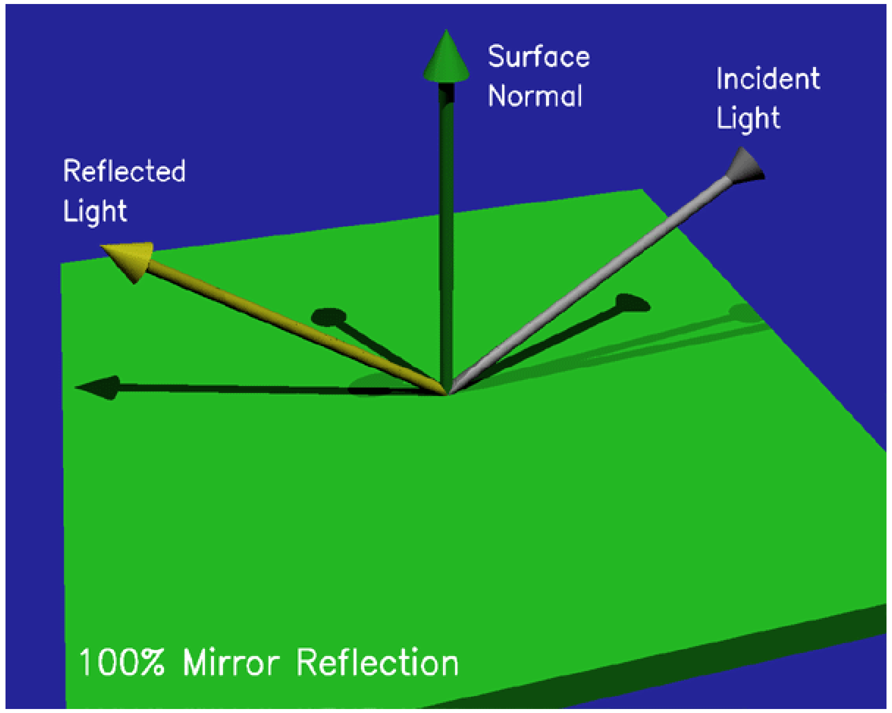
$$I_R = I_{saR}K_{aR} + I_{sdR}K_{dR}\cos\theta + I_{ssR}K_{sR}\cos\alpha^n$$
$$I_G = I_{saG}K_{aG} + I_{sdG}K_{dG}\cos\theta + I_{ssG}K_{sG}\cos\alpha^n$$
$$I_B = I_{saB}K_{aB} + I_{sdB}K_{dB}\cos\theta + I_{ssB}K_{sB}\cos\alpha^n$$

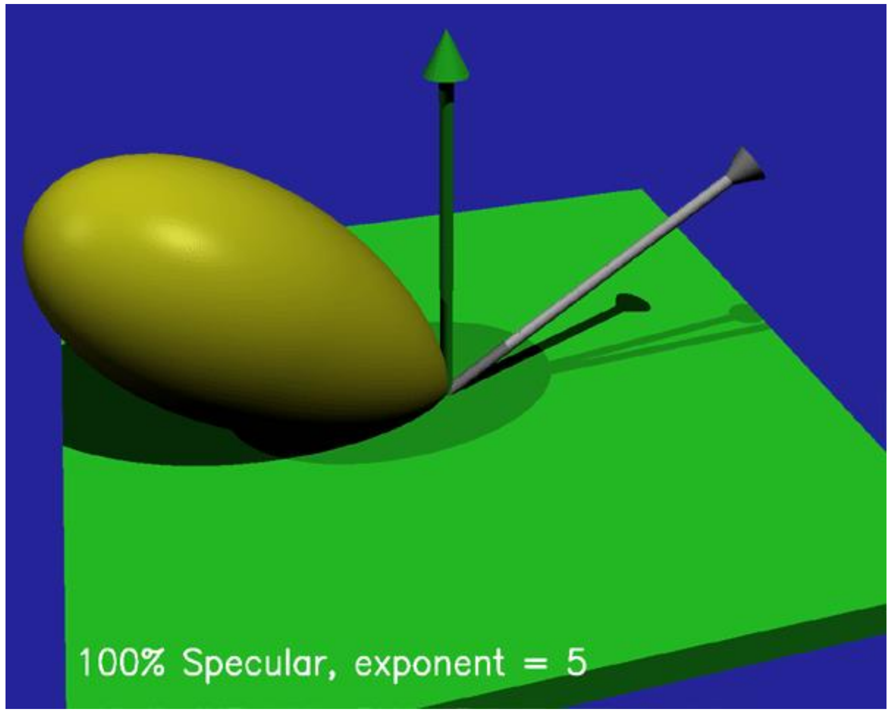
Ou

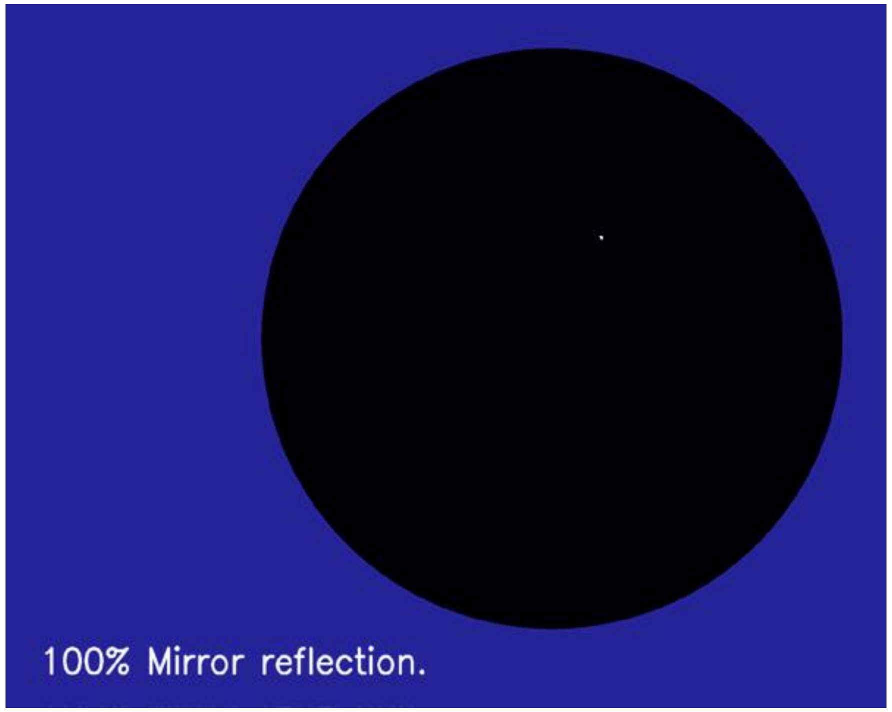
$$I_R = I_{saR}K_{aR} + I_{ldR}K_{dR}(\overrightarrow{L_l}.\overrightarrow{N}) + I_{lsR}K_{sR}(\overrightarrow{R_l}.\overrightarrow{V})^n$$
$$I_G = I_{saG}K_{aG} + I_{ldG}K_{dG}(\overrightarrow{L_l}.\overrightarrow{N}) + I_{lsG}K_{sG}(\overrightarrow{R_l}.\overrightarrow{V})^n$$
$$I_B = I_{saB}K_{aB} + I_{ldB}K_{dB}(\overrightarrow{L_l}.\overrightarrow{N}) + I_{lsB}K_{sB}(\overrightarrow{R_l}.\overrightarrow{V})^n$$

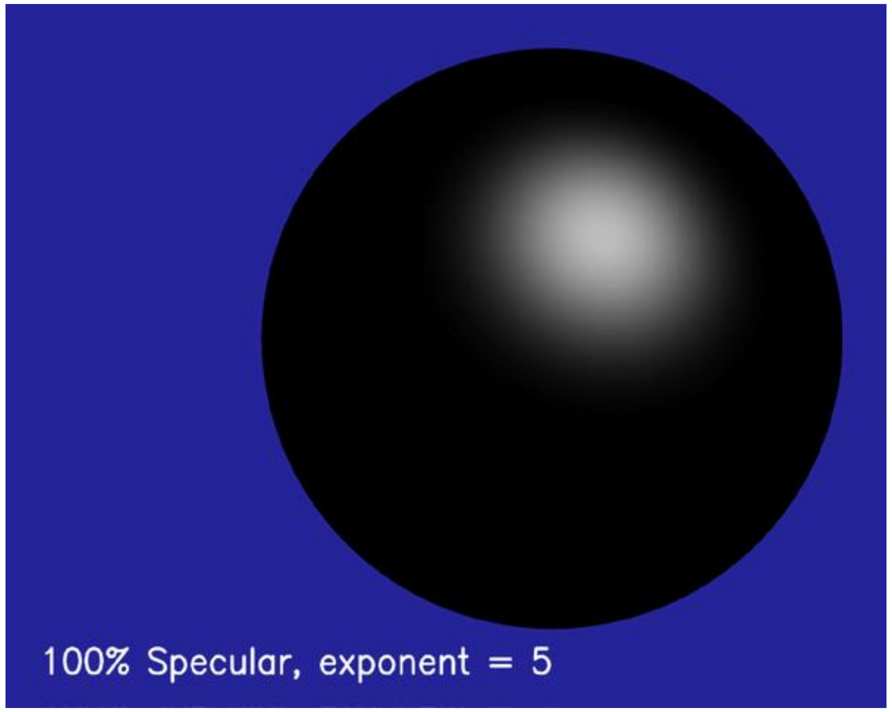
Illumination locale

Résultats



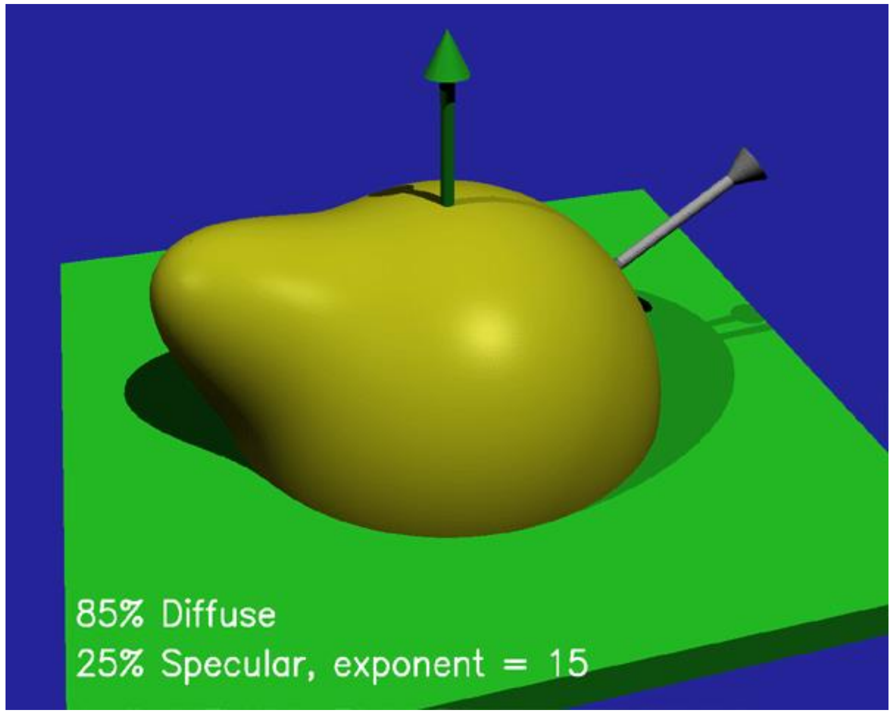


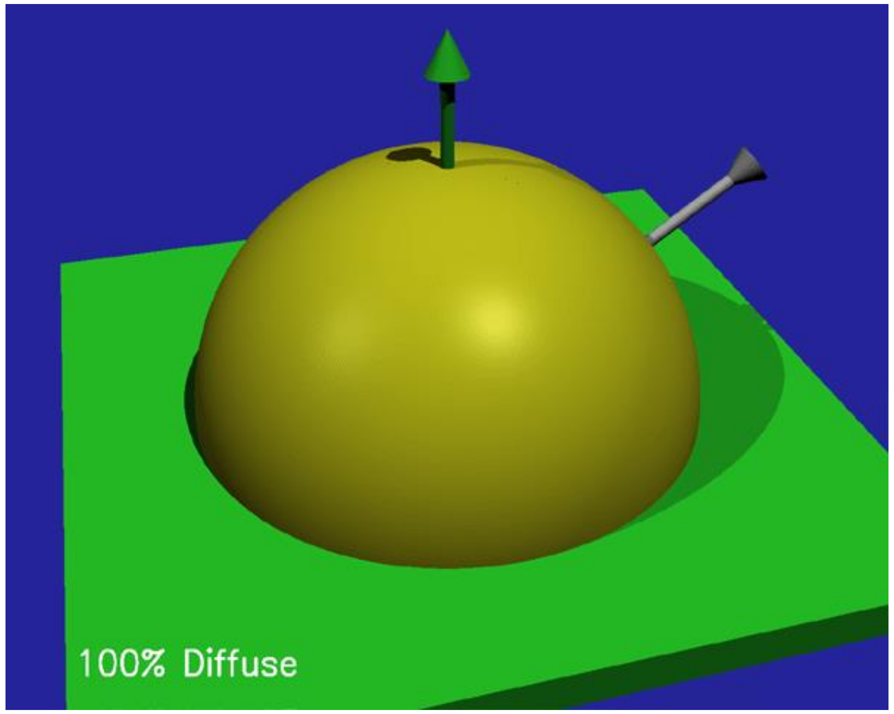


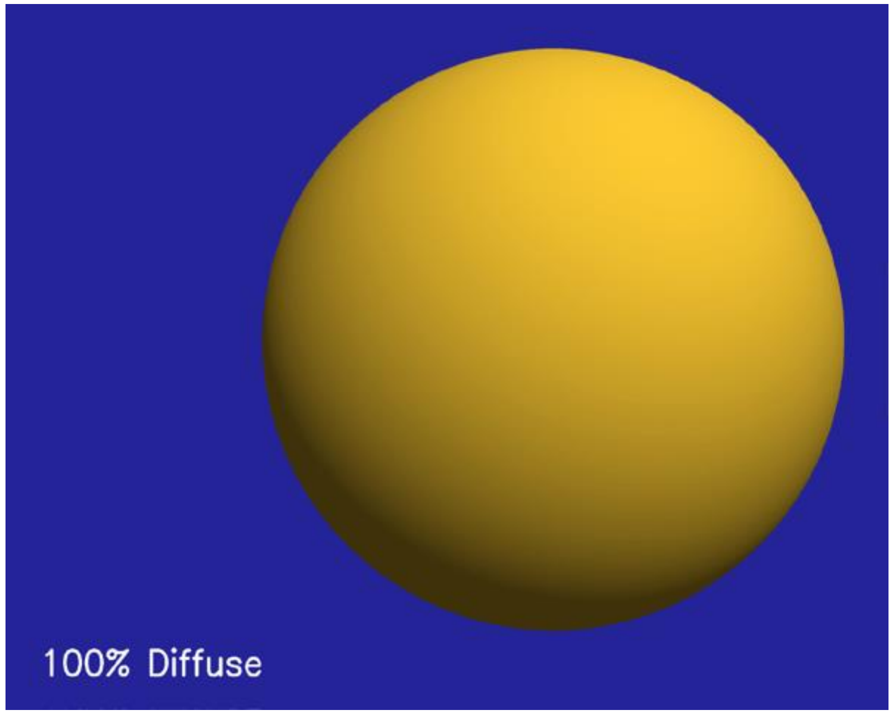


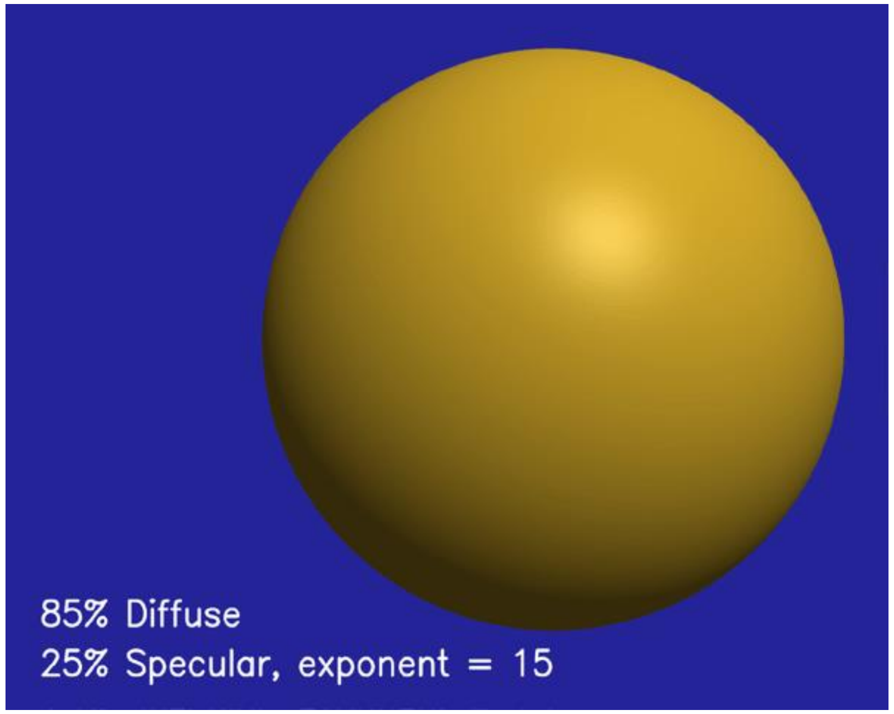
Illumination locale

Résultats









Modèle de Phong

- Avantages
  - très pratique (simple à utiliser, résultats intéressants)
  - rapide à calculer
- Désavantages
  - pas de sens physique
  - pas de lien avec les propriétés du matériau (rugosité …)

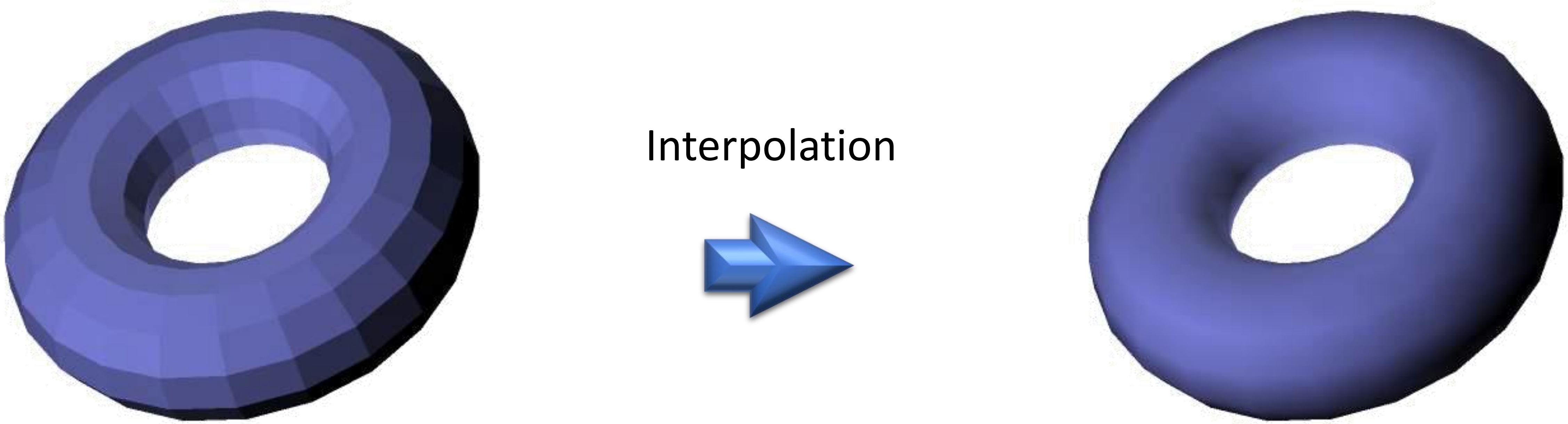
Autres modèles d'éclairages

- Plus efficaces ou réalistes
  - Cook-Torrance (1982)
  - Oren-nayar (1994)
  - Minnaert
  - Subsurface scattering (SSS)
  - BRDF
  - PBR modèle
  - Illumination globale

- Pipeline graphique programmable

Illumination d'un objet 3D

- Même illumination pour tout un polygone (élément du maillage) :
  - affichage « plat » (flat shading)



- **Solution** : calculer l'illumination pour chaque sommet des polygones, puis **interpoler** l'illumination d'un sommet à un autre.

Illumination d’un objet 3D

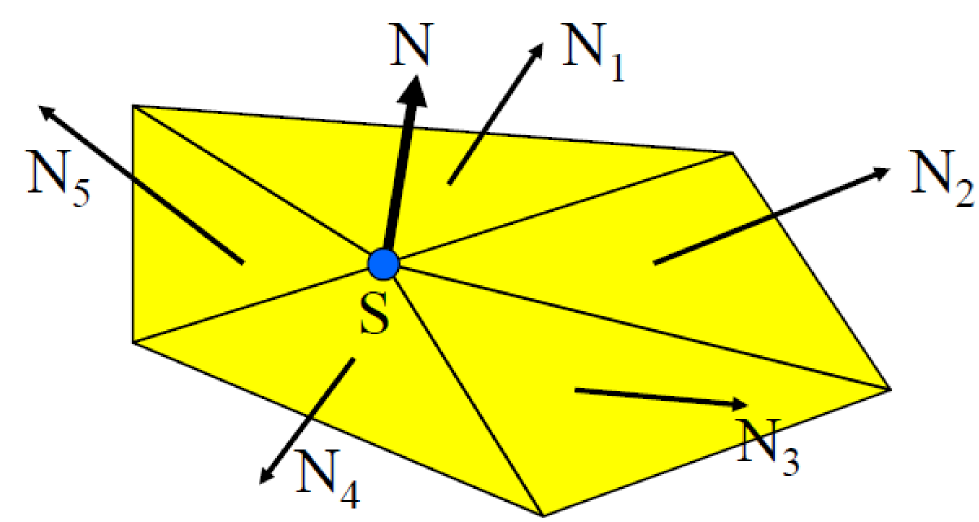
- But : calculer une couleur pour chaque point visible à l’écran de l’objet 3D qu’on affiche.
  - Lissage de **Gouraud** : interpolation de **couleurs**
  - Lissage de **Phong** : interpolation de **normales**

Interpolation de Gouraud

- Pour chaque polygone à afficher :
  - calculer pour chaque sommet du polygone une couleur au moyen d'un modèle d'illumination (Phong …)
  - interpoler les **couleurs** des sommets pour calculer la couleur de chaque pixel du polygone.

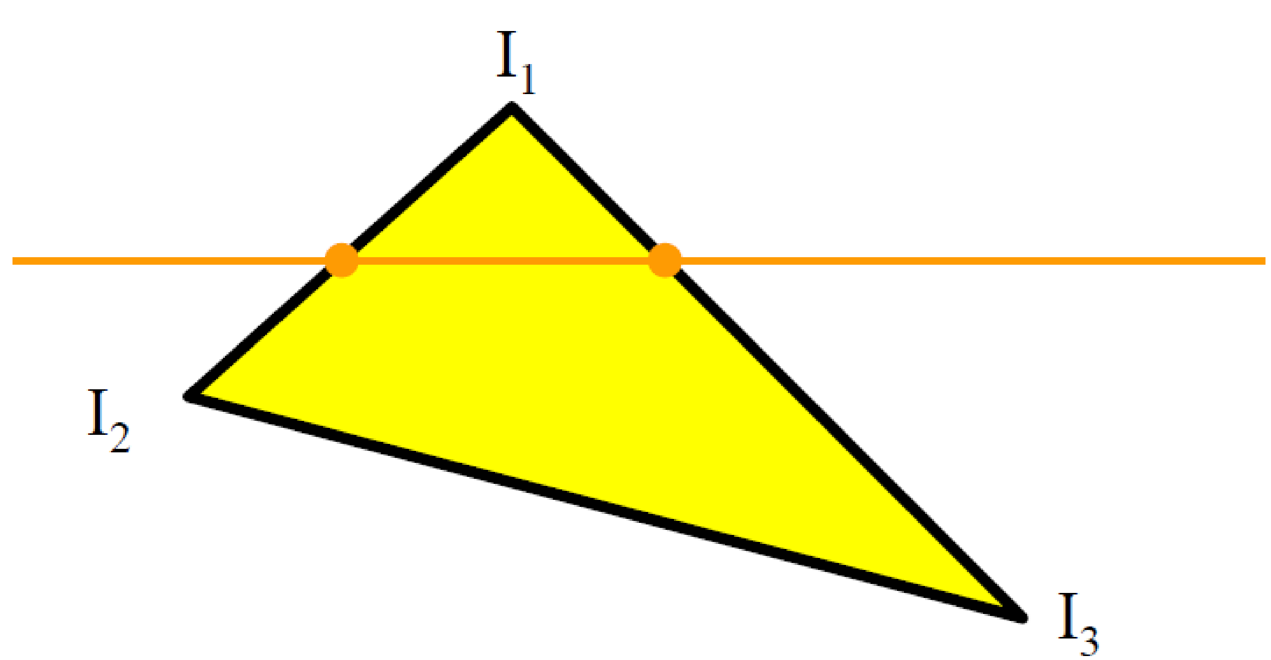
Interpolation de l'illumination

- Pour calculer la couleur en un sommet, on a besoin d'une normale en ce point :
  - surface analytiquement connue (ex : une sphère, un cylindre ...) → calcul direct
  - surface de départ est un maillage polygonal, comment faire ?? → interpoler les normales de face



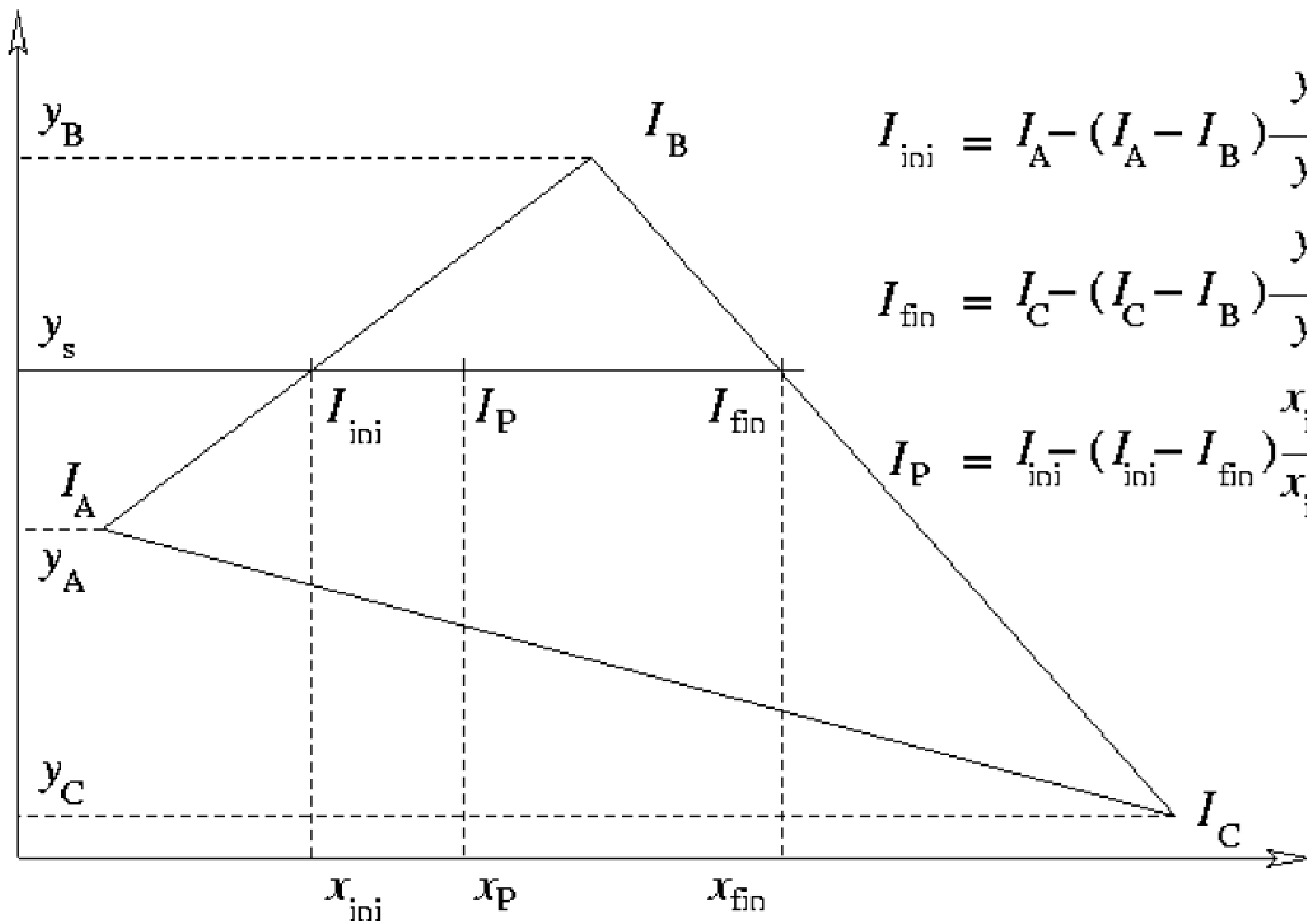
Interpolation de l'illumination

- Pour chaque sommet on calcule une couleur avec un modèle d'illumination
  - sur une arête, interpoler les couleurs entre les 2 sommets
  - sur une ligne de remplissage (scanline) du polygone, interpoler les couleurs entre 2 arêtes



Illumination d’un objet 3D

- Interpolation de l’illumination par **Gouraud** :



$$I_{ini} = I_A - (I_A - I_B)\frac{y_A - y_s}{y_A - y_B}$$

$$I_{fin} = I_C - (I_C - I_B)\frac{y_C - y_s}{y_C - y_B}$$

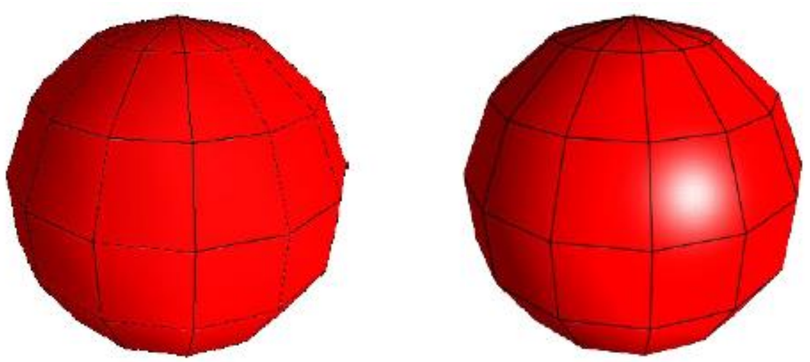
$$I_P = I_{ini} - (I_{ini} - I_{fin})\frac{x_{ini} - x_p}{x_{ini} - x_{fin}}$$

Illumination d’un objet 3D

- Lissage de Phong plus lent que celui de Gouraud (plus de calcul d’illumination) mais plus nettement plus beau :

Flat *Gouraud* *Phong*

- Permet de calculer les effets spéculaires **contenus dans une facette**, contrairement au lissage de Gouraud

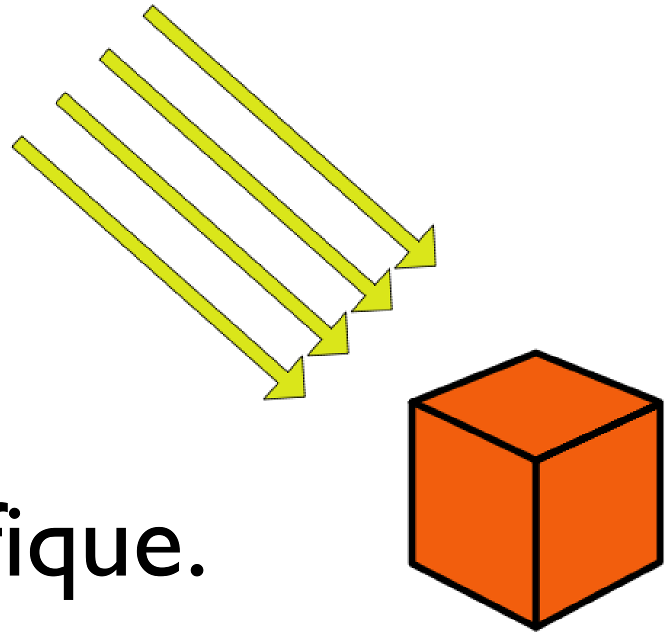


Gouraud *Phong*

Lumières

- Plusieurs types d'éclairage :
  - source directionnelle
  - source ponctuelle
  - source projecteur

Lumière directionnelle



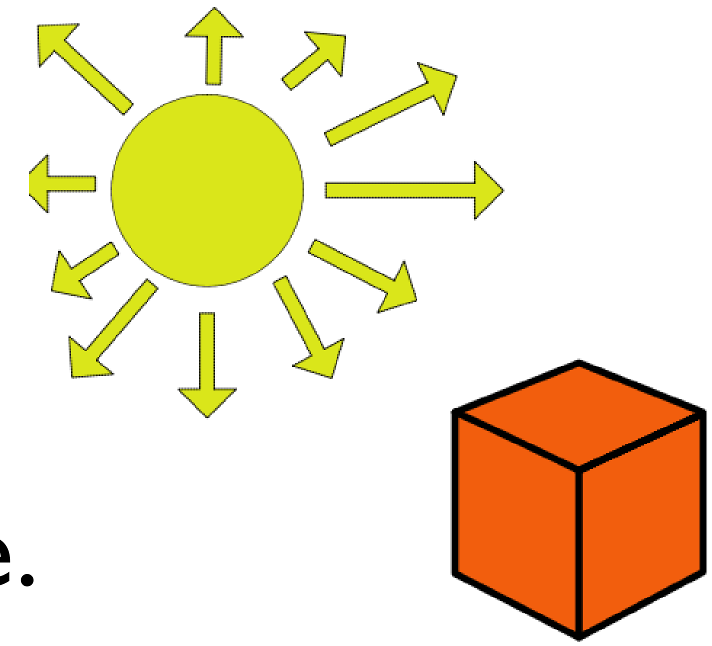
- Définie par une direction :
  - 4 coordonnées homogènes (x, y, z, 0)
- La lumière vient d'une direction spécifique.

Coordonnées homogènes d'un vecteur

```
float position[] = {0.0, -1, 0.0, 0.0};
glLightfv(GL_LIGHT0, GL_POSITION, position);
```

Source ponctuelle

- Définie par une position :
  - 4 coordonnées homogènes (x, y, z, 1)



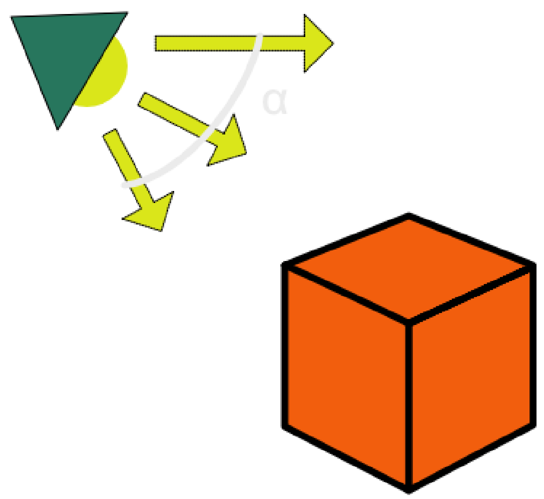
- La lumière vient d'un point spécifique.

Coordonnées homogènes d'un point

```
float position[] = {0.0, -1, 0.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, position);
```

Lumière projecteur = « spot »

- La lumière vient d'un point spécifique,
  - intensité dépendant de la direction



- Position : emplacement de la source
- Direction : axe central de la lumière
- Angle : largeur du rayon

```
float position[] = {0.0, 10.0, 0.0, 1.0};
float direction[] = {1.0, -1.0, 0.5};
float angle = 45.0f;
glLightfv(GL_LIGHT0, GL_POSITION, position);
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, direction);
glLightf(GL_LIGHT0, GL_CUTOFF, angle);
```

Bilan

Etant donné les couleurs ambiante, diffuse, spéculaire de la lumière, les composantes du matériau d'un objet, la couleur finale sera calculée grâce à l'équation du **modèle de Phong** :

Couleurs ambiante, diffuse, spéculaire de la lumière

Couleur finale affichée

$$
\begin{aligned}
I_R &= I_{saR} \cdot K_{aR} + I_{sdR} \cdot K_{dR} \cdot \cos\theta + I_{ssR} \cdot K_{sR} \cdot (\cos\alpha)^n \\
I_V &= I_{saV} \cdot K_{aV} + I_{sdV} \cdot K_{dV} \cdot \cos\theta + I_{ssV} \cdot K_{sV} \cdot (\cos\alpha)^n \\
I_B &= I_{saB} \cdot K_{aB} + I_{sdB} \cdot K_{dB} \cdot \cos\theta + I_{ssB} \cdot K_{sB} \cdot (\cos\alpha)^n
\end{aligned}
$$

Couleurs ambiante, diffuse, spéculaire du matériau de l'objet