

# Chapitre 6: Lumière

INF5071 — Infographie

Alexandre Blondin Massé

Université du Québec à Montréal

Hiver 2019

# Plan

- 1 Généralités
- 2 Sources de lumière
- 3 Diffusion de la lumière
- 4 Lancer de rayon

# Généralités

# Optique

## Plusieurs phénomènes

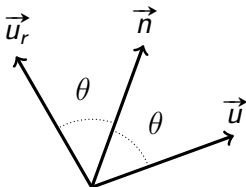
- La **diffusion** de la lumière (couleur de l'objet)
- La **réflexion** de la lumière
- La **réfraction**
- La **diffraction**
- Les **caustiques**, ...

## Considérations informatiques

- Plusieurs **modèles** (simples et complexes)
- Tous basés sur les **vecteurs normaux** à une surface
- Sans doute la composante du rendu la plus **coûteuse** en temps de calcul

## Réflexion (1/2)

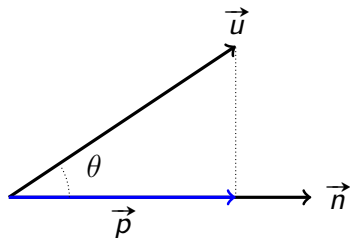
- Soient  $\vec{u}$  et  $\vec{n}$  deux **vecteurs**
- Soit  $\vec{u}_r$  le vecteur **réfléchi** de  $\vec{u}$  par rapport à  $\vec{n}$
- Comment exprime-t-on  $\vec{u}_r$  en **fonction** de  $\vec{u}$  et de  $\vec{n}$ ?



## Stratégie

- Calculer d'abord le vecteur  $\text{proj}_{\vec{n}}(\vec{u})$ , qui est la **projection** de  $\vec{u}$  sur  $\vec{n}$
- Puis remarquer que  $\text{proj}_{\vec{n}}(\vec{u}) = \text{proj}_{\vec{n}}(\vec{u}_r)$

# Projection



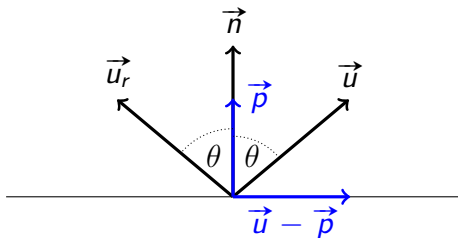
- Soit  $\vec{p} = \text{proj}_{\vec{n}}(\vec{u})$
- Alors  $\vec{n}$  et  $\vec{p}$  sont dans la **même** direction et

$$\cos(\theta) = \frac{\|\vec{p}\|}{\|\vec{u}\|}$$

- Montrer qu'on peut en **déduire**

$$\text{proj}_{\vec{n}}(\vec{u}) = \vec{p} = \frac{\vec{u} \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \vec{n}$$

## Réflexion (2/2)



- Soit  $\vec{p} = \text{proj}_{\vec{n}}(\vec{u})$
- Montrer que

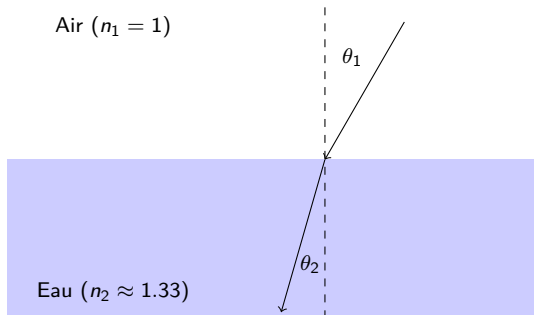
$$\text{refl}_{\vec{p}}(\vec{u}) = \vec{u}_r = \vec{u} - 2 \frac{\vec{u} \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \vec{n}$$

# Réfraction

## Loi de **Snell-Descartes**

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2),$$

où  $n_1, n_2$  sont les **indices de réfraction** et  $\theta_1, \theta_2$  les angles par rapport aux vecteurs normaux.



# Sources de lumière

# Types classiques

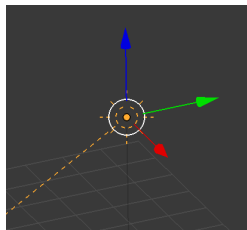
## Typiquement présents

- Type **soleil** (*sun*)
- Type **ponctuel** (*point*)
- Type **hemi** (*hemi*)
- Type **surface** (*area*)
- Type **spotlight** (*spot light*)

## Caractéristiques

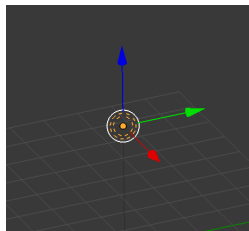
- Basés sur des objets **géométriques** simples: demi-droites, demi-sphère, points, cônes, ...
- Modélisation **simplifiée**

# Type soleil



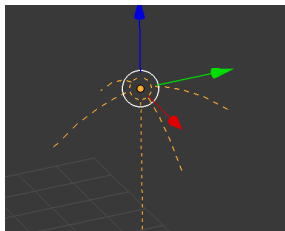
- Les rayons ont la **même direction** (un vecteur)
- La **position** de la lumière n'influence pas l'éclairage
- On peut aussi lui donner une **couleur**
- Cette lumière crée des **ombres**

# Type ponctuel



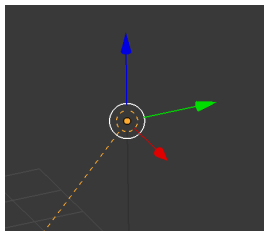
- Les rayons émanent tous du **même point**, qui est la position de la lumière
- La **distance** entre un point et la source de lumière détermine l'intensité de l'éclairage
- On paramètre la **vitesse** de décroissance de l'intensité
- On obtient des **ombres** et on peut paramétrer la **couleur**

# Type *hemi*



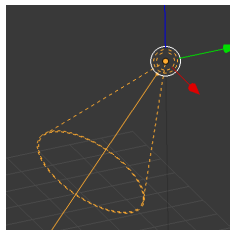
- L'éclairage est **uniforme** dans une demi-sphère
- La **position** n'influence pas l'éclairage
- Par contre, la **direction** est importante
- C'est la seule lampe classique qui ne produit **pas d'ombre**
- Permet de simuler la lumière **ambiante**

# Type surface



- C'est une généralisation de la lumière ponctuelle
- C'est équivalent à placer plusieurs lumières **ponctuelles** le long d'une surface
- Plus **réaliste** (ombres floues)
- Mais plus **gourmandes** en temps de calcul
- La forme, la position et la direction influencent l'éclairage

# Type spotlight



- Émet un **cône** de lumière
- Paramètres importants: la **position** et la **direction**
- Possible de créer des effets de **halo**

# Diffusion de la lumière

# Modèles de base

Trois modèles classiques en infographie:

- Lumière **ambiante**: intensité lumineuse générale qui ne tient pas compte de la scène
- Diffusion de **Lambert** : c'est la « couleur » de l'objet, avec un dégradé
- Diffusion de **Phong** : c'est la couleur de réflexion de lumière de l'objet.

# Lumière ambiante

- Très simple: entièrement déterminée par un nombre représentant son **intensité**
- Généralement, ce nombre est entre 0 et 1, mais il est aussi possible qu'il dépasse 1
- Étant donné un pixel  $p = (r, g, b)$  et une intensité ambiante  $i_a$ , le pixel devient

$$p' = i_a p = (i_a r, i_a g, i_a b).$$

- Quand il n'y a aucune lumière, on a  $i_a = 0$  et dans ce cas, tous les objets apparaissent en **noir**
- Pour avoir une intensité ambiante qui n'influence pas les couleurs, il suffit de choisir  $i_a = 1$ .

# Diffusion de Lambert

- Le modèle le plus simple qui tient compte de la lumière est appelé diffusion de **Lambert**, introduit dans *Photometria* en 1760, par Johann Heinrich Lambert
- L'équation qui décrit le phénomène est

$$i_d = (\vec{\ell} \cdot \vec{n})i_{\ell,d},$$

où

- $i_{\ell,d}$  est l'intensité lumineuse au point donné
- $\ell$  est un vecteur unitaire en direction de la lumière
- $\vec{n}$  est le vecteur normal unitaire à la surface
- On obtient alors **chaque pixel**  $p'$  à partir de  $p$  par

$$p' = (i_a + i_d)p.$$

# Prendre la vue en compte

- Une limite du modèle de diffusion de **Lambert** est qu'il ne tient pas compte de la position de la **caméra**
- En particulier, l'**angle** de vue n'influence pas le rendu d'un pixel visible dans les deux cas
- Il est donc important d'adapter le modèle de Lambert pour qu'il tienne compte de la lumière **réfléchie** vers l'observateur
- On obtient des effets plus **réalistes**:
  - soleil aveuglant
  - surface brillantes/mates
- Ce type de lumière est dite **spéculaire**

# Diffusion de Phong

- En 1973, *Bui Tuong Phong* a proposé un modèle de calcul qui prend en considération la lumière **spéculaire**
- L'équation qui décrit le phénomène est

$$i_s = i_{\ell,s}(\vec{v} \cdot \vec{r})^\alpha,$$

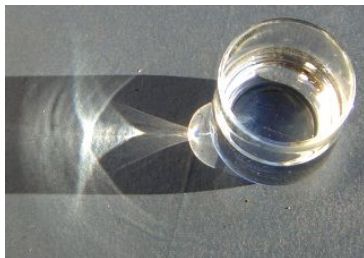
où

- $i_{\ell,s}$  est l'intensité lumineuse au point donné
- $\vec{v}$  est le vecteur unitaire pointant vers la caméra
- $\vec{r}$  est le vecteur réfléchi unitaire
- $\alpha$  est un coefficient de brillance qui ne dépend que du type de surface
- Une variante appelée modèle de **Blinn-Phong** permet d'accélérer les calculs en simplifiant l'exponentiation

# Lancer de rayon

# Contexte

- Les modèles de base (Lambert, Phong, etc.) tiennent compte seulement de la provenance **directe** de la lumière (*illumination directe*)
- En particulier, ils ne tiennent pas compte de la **réflexion** et de la **réfraction** (*illumination indirecte*)
- Ils ne permettent pas de simuler des **caustiques**



(source: <https://commons.wikimedia.org/wiki/File:Kaustik.jpg>)

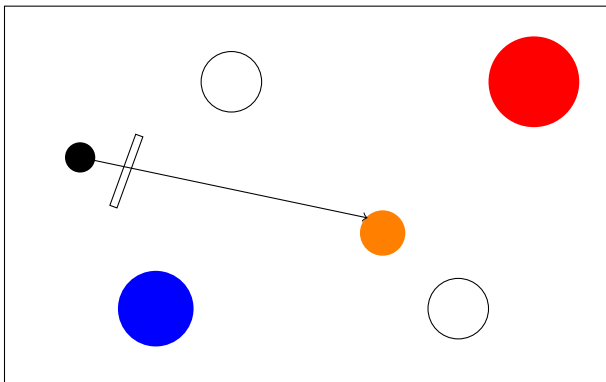
# Algorithmes

- Par **radiance**
- Lancer de **rayons**
- Lancer de **cônes**
- Suivi de **chemins**
- Transport de **Metropolis** (*Metropolis light transport* MLT)
- Occlusion **ambiante**
- Lancer de **photons**, etc.

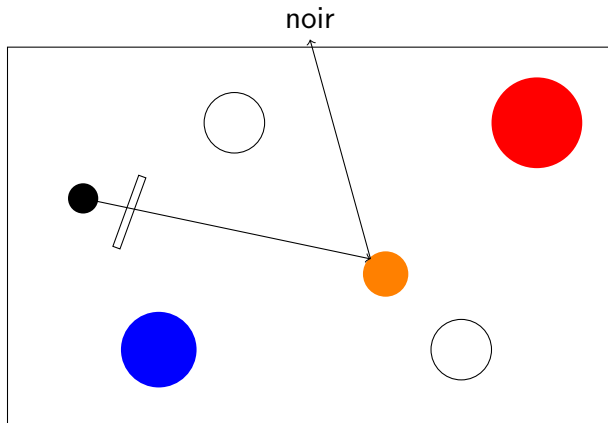
# Idée de base

- 1: **fonction** COULEURPIXEL( $r$  : rayon,  $\ell$  : entier)
- 2:     Si  $\ell$  dépasse la profondeur maximale, **retourner** noir
- 3:     Soit  $O$  l'objet le plus près du rayon et  $p$  le point de contact
- 4:     Si  $O$  n'existe pas, **retourner** noir
- 5:     Soit  $\vec{n}$  le vecteur normal à  $O$  au point  $p$
- 6:     Soit  $M$  le matériau de l'objet  $O$
- 7:      $c \leftarrow M.emission$
- 8:     Choisir un nouveau rayon  $r'$  aléatoirement à partir de  $p$
- 9:      $c' \leftarrow COULEURPIXEL(r', \ell + 1)$
- 10:     $c'' \leftarrow c + 2c' M.réflexion \text{ PRODSCALAIRE}(r', \vec{n})$
- 11:    **retourner**  $c''$
- 12: **fin fonction**

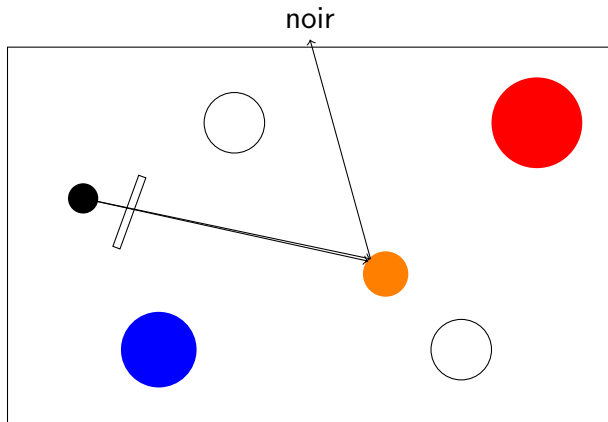
# Exemple



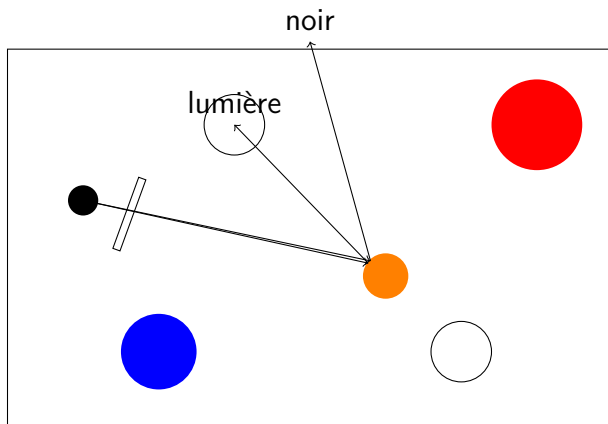
# Exemple



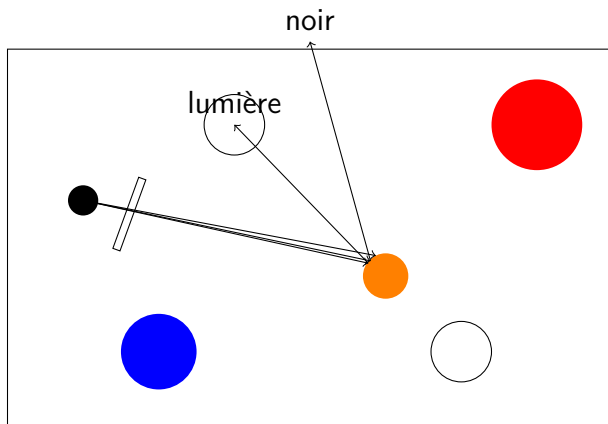
# Exemple



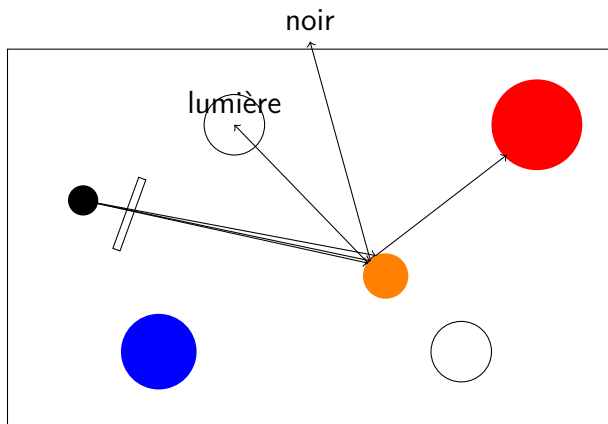
# Exemple



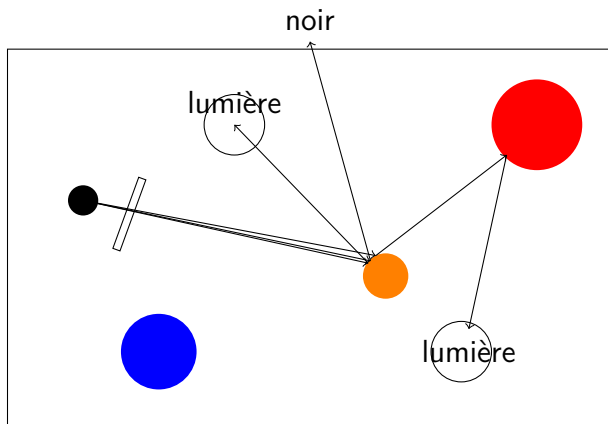
# Exemple



# Exemple



# Exemple



# Résultat et coût de l'algorithme

- Plus la **profondeur** (nombre de rebonds) est grande, plus l'algorithme est **coûteux**
- Il faut également calculer si chaque rayon intercepte chaque objet
- De plus, il faut lancer **plusieurs rayons** par pixel
- On calcule éventuellement la **moyenne** des couleurs pour chaque rayon lancé à travers un même pixel
- Il s'agit d'une **simulation**: le résultat peut varier d'une fois à l'autre, car basé sur le **hasard**

# Représentation d'une scène

- La scène est **statique** pour chaque lancer de rayon
- Il peut donc être pratique de la représenter de façon **efficace**
- Plus particulièrement, on souhaite détecter les collisions entre un **rayon** et un **objet** efficacement
- Plusieurs structures **hiérarchiques**: grille uniforme, arbre *kd*, *octree*, hiérarchie de volume englobant, etc.
- Permet de réduire la complexité **linéaire** à une complexité **logarithmique** ou même parfois **constante**, selon la scène.

# Simuler la propagation d'un rayon

- Un rayon lumineux peut être représenté par une **demi-droite**

$$cr(t) = t\vec{d} + P, \quad t \geq 0$$

où  $\vec{d}$  est un **vecteur directeur** et  $P$  est le **point** de départ

- On doit donc calculer son intersection avec les objets de la **scène**
- Si l'objet est une **sphère**, on cherche donc une intersection de type droite/sphère
- Si l'objet est une **boîte**, on cherche donc une intersection de type droite/segment en 2D ou droite/plan en 3D

# Intersection droite/plan

- Prenons la demi-droite

$$D: \vec{r}(t) = t\vec{d} + A, \quad t \geq 0.$$

- Soit  $P$  un plan de **vecteur normal**  $\vec{n}$  passant par  $B$
- Alors il y a trois scénarios possibles:
  - $D \cap P$  est vide
  - $|D \cap P| = 1$
  - $D \subseteq P$
- En particulier, il y a **intersection** si et seulement s'il existe  $t$  tel que

$$\vec{n} \cdot (t\vec{d} + A - B) = 0.$$

- Il suffit donc d'**isoler**  $t$ !

## Intersection droite/sphère (1/2)

- Considérons la sphère  $S$  de **rayon**  $R$  et de **centre**  $C$
- Alors tout point  $P$  se trouvant sur  $S$  doit satisfaire l'équation

$$\|\vec{CP}\| = R.$$

- Alors le rayon intersecte la sphère si et seulement s'il existe un  $t \geq 0$  tel que

$$\|\vec{r}(t) - C\| = R.$$

## Intersection droite/sphère (2/2)

- On doit donc résoudre une équation du **second degré** en  $t$
- En effet, si  $C = (h, k, \ell)$ ,  $\vec{d} = (a, b, c)$  et  $P = (x_0, y_0, z_0)$ , alors l'équation se réécrit

$$(at + x_0 - h)^2 + (bt + y_0 - k)^2 + (ct + z_0 - \ell)^2 = R^2.$$

- Il ne reste qu'à **isoler  $t$**
  - Il y a **trois cas** possibles
- **Aucune solution**: pas d'intersection
- **Une solution unique**: la droite est tangente à la sphère
- **Deux solutions**: la droite est sécante à la sphère

# Intersection droite/boîte

- Un rayon lumineux intersecte une boîte si et seulement s'il intersecte un des **quatre segments** qui délimitent la boîte
- Un segment est une droite dont on restreint le domaine
- Ainsi, pour calculer l'intersection d'une **demi-droite** et d'un **segment**, il suffit de
  - Calculer l'intersection des droites sous-jacentes
  - Si l'intersection existe, on vérifie si elle est compatible avec le domaine de la demi-droite et du segment
- Il ne reste qu'à déterminer comment calculer l'intersection de deux droites, qu'on a déjà vu!

# Intersection de deux droites (rappel)

```
1: fonction INTERSECTION( $D_1, D_2$  : droites)
2:   si  $D_1.\text{point} \in D_2$  alors
3:     si  $D_2.\text{point} \in D_1$  alors
4:       retourner  $D_1$                                 ▷ Droites confondues
5:     sinon
6:       retourner  $D_1.\text{point}$                             ▷ Unique point
7:     fin si
8:   sinon
9:      $u \leftarrow D_2.\text{point} - D_1.\text{point}$ 
10:     $v \leftarrow D_2.\text{dir} \times u$ 
11:     $w \leftarrow D_2.\text{dir} \times D_1.\text{dir}$ 
12:    si  $v$  et  $w$  ne sont pas dans la même direction alors
13:      retourner rien                                    ▷ Pas d'intersection
14:    fin si
15:     $s \leftarrow 1$  si  $v$  et  $w$  ont la même orientation,  $-1$  sinon
16:    retourner  $D_1.\text{point} + \frac{s\|v\|}{\|w\|} D_1.\text{dir}$           ▷ Unique point
17:  fin si
18: fin fonction
```