

# IFT3355: Infographie

## **Sujet 6: shading 4**

(illumination *globale* 1)

Derek Nowrouzezahrai

Département d'informatique et de recherche opérationnelle

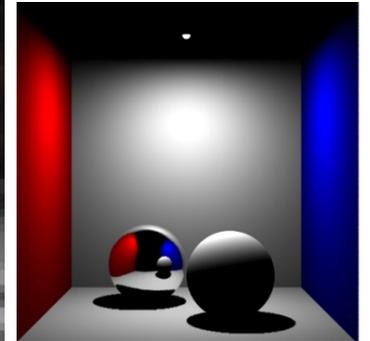
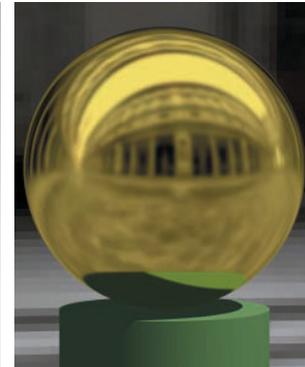
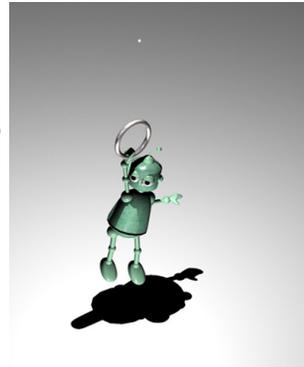
Université de Montréal



# À partir de représentations discrètes jusqu'à continus

- Les effets qu'on vient d'étudier sont très limités:

- les ombres dur,
- les réflexion miroir parfaite,
- l'illumination directe, ...

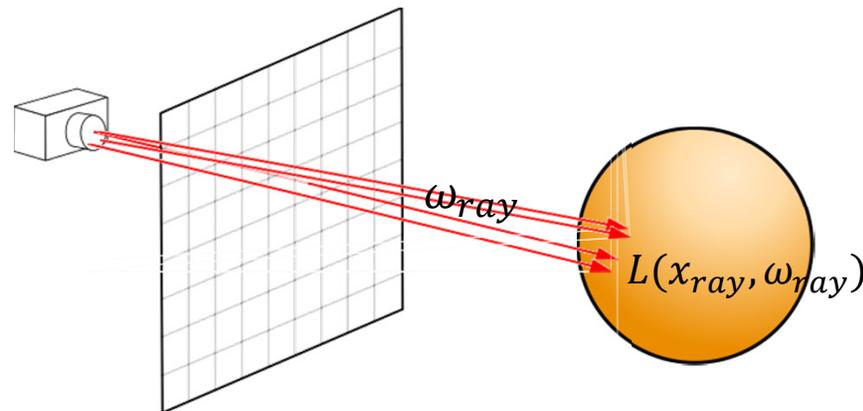


- Pour modéliser des effets plus complexes, on a besoin de formulations de shading plus réalistes
- Plus tard nous étudions un modèle mathématique générale pour l'illumination réaliste basé sur une équation d'intégrale récursifs
  - Avant qu'on arrive à cette formulation, nous pouvons montrer comment plusieurs effets de rendu réalistes peuvent être modélisés en généralisant des **sommes** à des **intégrales**



# $\Sigma \rightarrow \int$ Example 1: anti-aliasing

- Présentement on génère une seule rayon de l'œil par pixel
  - Après le calcul de shading, ce rayon correspond à une échantillonnage de la fonction de shading à travers du pixel:  $\text{couleur}(\text{pixel}) = L(x_{ray}, -\omega_{ray})$



- Au lieu d'utiliser un seul échantillon par pixel nous pouvons **intégrer** (et optionnellement filtrer) **tous les valeurs** de shading qui tombent sous (la projection de la scène sur) le pixel

$$\text{couleur}(\text{pixel}) = \int_{\text{pixel}} L(x_p, -\omega_p) h(x_p) dp$$



# $\Sigma \rightarrow \int$ Example 2: motion-blur (Flou de mouvement)

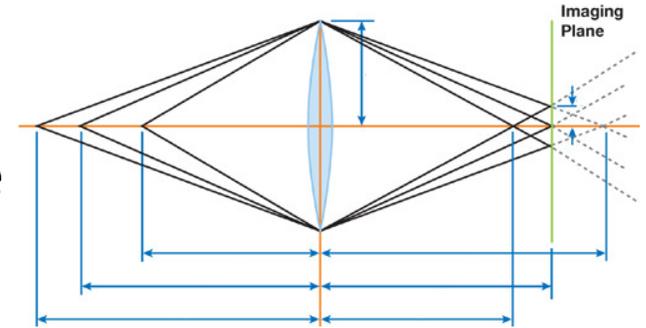
- Présentement on considère un seule moment en temps
  - Si nos objets bouge et le caméra a une vitesse d'obturation non-instantané, nous obtiendrons des images floue dans la direction(s) de motion



$$\text{couleur}(\text{pixel}) = \int_{\text{temps}} \text{couleur}(\text{pixel}, t) dt$$

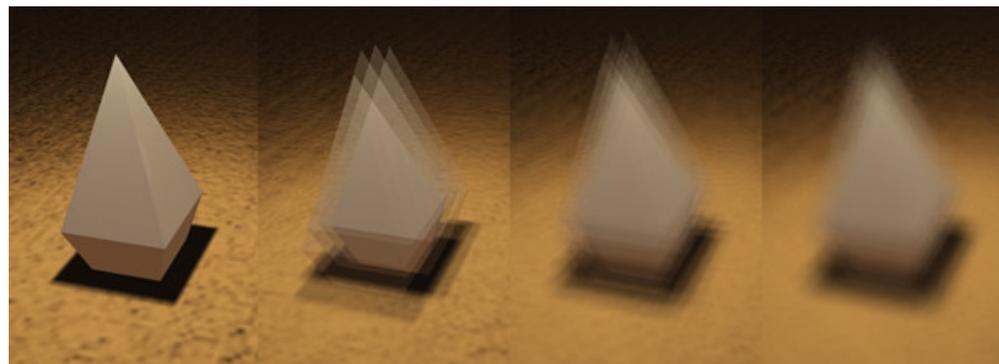
# $\Sigma \rightarrow \int$ Example 3: depth-of-field (Profondeur de champ)

- En réalité nos yeux (ou des caméras) sont constitués d'un système de lentille qui nous permet de seulement prendre focus sur une tranche planaire de la scène 3D
  - Les objets en dehors de ce plan seront flous



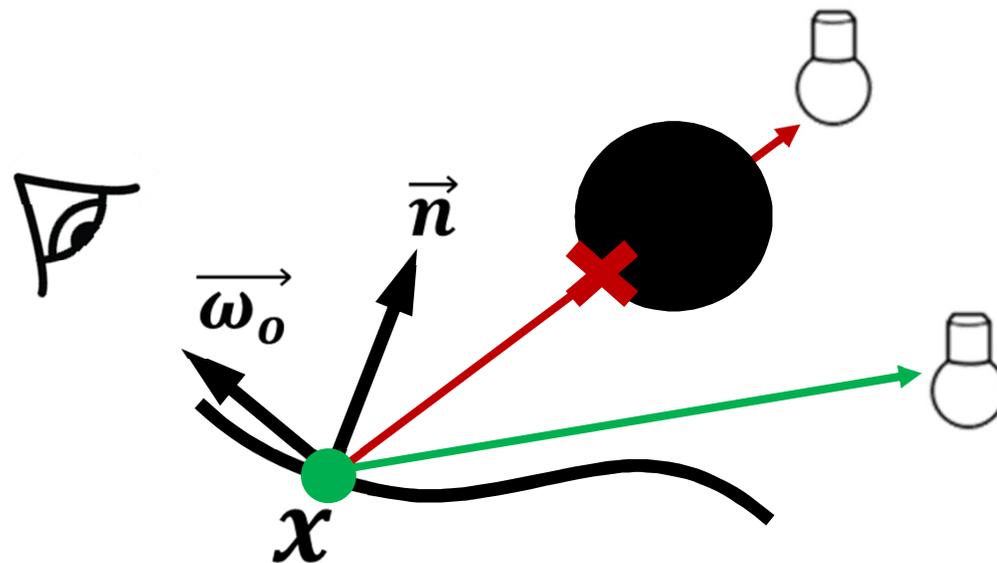
- Le système de lentille concentre ou diffuse les rayons de lumière selon la position relative entre la lentille et les objets visibles

$$\text{couleur}(\text{pixel}) = \int_{\text{lentille}} L(x_r, -\vec{\omega}_r) dr$$



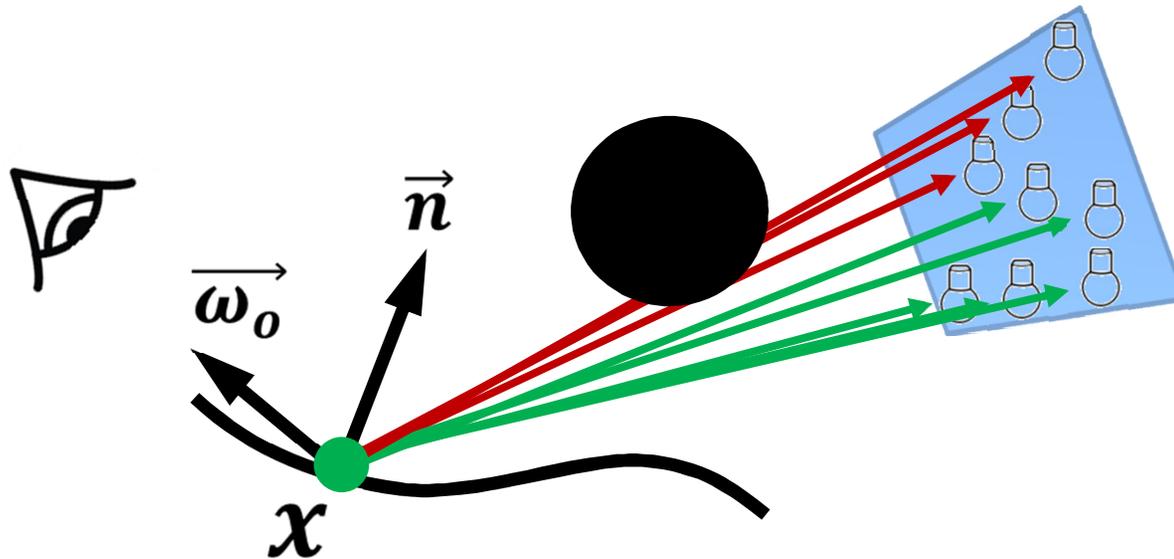
# $\Sigma \rightarrow \int$ Example 4: les ombres

- Nous avons déjà vu comment déterminer l'illumination quand on a plusieurs lumières ponctuelles (ou directionnelles):



# $\Sigma \rightarrow \int$ Example 4: les ombres

- Maintenant imaginer si on a une lumière étendue:

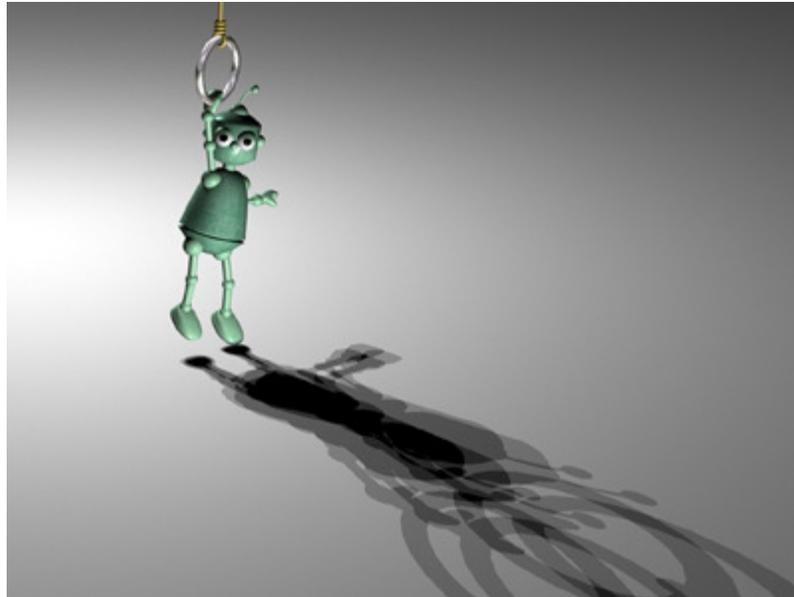


- On peut effectivement traiter ce type de lumière comme un nombre infini de lumières ponctuelles/directionnelles (si on prend soins de normaliser nos sommes proprement)



# $\Sigma \rightarrow \int$ Example 4: les ombres

- On peut aussi effectuer ce calcul dans un système de rasterisation



# $\Sigma \rightarrow \int$ Example 4: les ombres

- Alors lorsque le nombre de lumières “virtuelles” tend vers l’infini nous obtiendrons la valeur correcte et notre modèle d’illumination (seulement diffus, pour simplifier) deviens:

$$L_o = \sum_{i=1}^N \left( \frac{L l_i}{r_i^2} \right) k_d S_\lambda \max(\vec{n} \cdot \vec{l}_i, 0) V(x, \vec{l}_i)$$

$$\begin{aligned} L_o &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \left( \frac{L l_i}{r_i^2} \right) k_d S_\lambda \max(\vec{n} \cdot \vec{l}_i, 0) V(x, \vec{l}_i) \\ &= \int_L L_l(l) G(x, l) [k_d S_\lambda \max(\vec{n} \cdot \vec{l}, 0)] V(x, \vec{l}) dl \end{aligned}$$

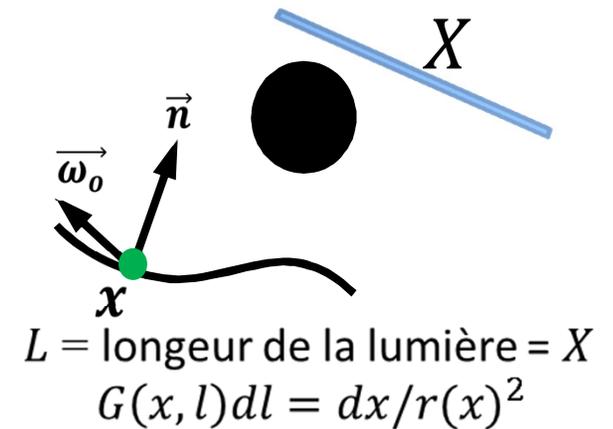
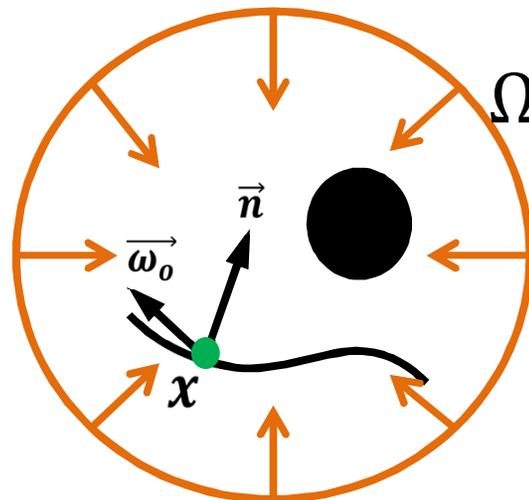
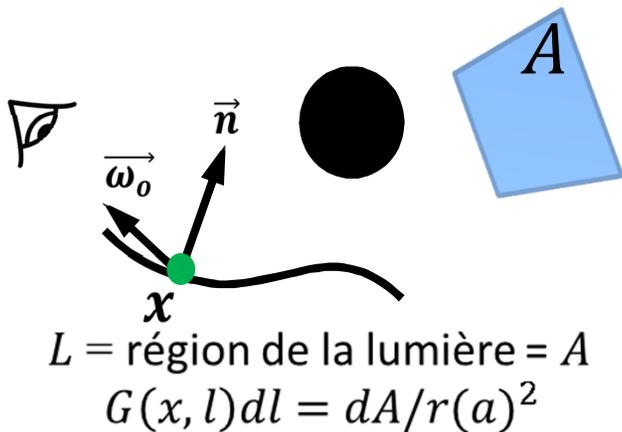


# $\Sigma \rightarrow \int$ Example 4: les ombres

- Il y a plusieurs paramétrisations valides (choix de  $L$  et  $G(x, l)dl$ )

$$L_o = \int_L L_l(l) [k_d S_\lambda \max(\vec{n} \cdot \vec{l}, 0)] V(x, \vec{l}) [G(x, l) dl]$$

$$L_o(x, \vec{\omega}_o) = \int_\Omega L_{in}(x, \vec{\omega}) f_r(x, \vec{\omega}, \vec{\omega}_o) V(x, \vec{\omega}) d\vec{\omega}$$

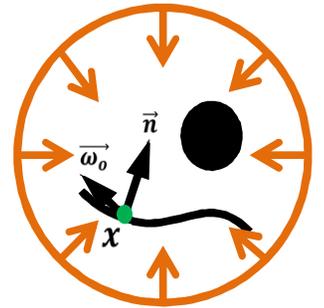


# Une représentation générale de shading: l'équation de rendu

(the rendering equation)

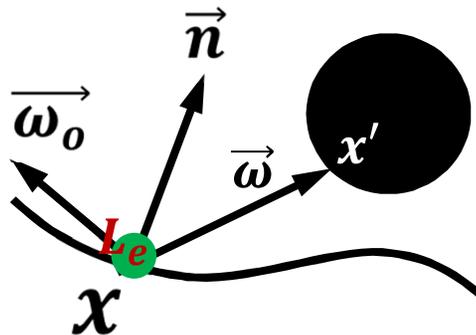
- On vient de définir une équation d'illumination continu basé sur une paramétrisation directionnelle

$$L_o = \int_{\Omega} L_{in}(x, \vec{\omega}) f_r(x, \vec{\omega}, \vec{\omega}_o) V(x, \vec{\omega}) d\vec{\omega}$$



- Noter qu'en générale, chaque surface peut agir comme source de lumière virtuelle, alors l'équation "finale" d'illumination (encore en forme directionnelle) est

$$L(x, \vec{\omega}_o) = L_e(x, \vec{\omega}_o) + \int_{\Omega} L(x', -\vec{\omega}) f_r(x, \vec{\omega}, \vec{\omega}_o) d\vec{\omega}$$



# Combiner les effets avec la linéarité d'intégration

- On peut combiner plusieurs effets d'illumination mathématiquement (et conceptuellement) en intégrant sur l'ensemble des domaines
- Anti-aliasage de l'image +
- Flou de mouvement +
- Profondeur de champ +
- Shading générale

$$\text{couleur(pixel)} = \int_{\text{lentille}} \int_{\text{temp}} \int_{\text{pixel}} \underbrace{L(x_{p,t,r}, -\omega_{p,t,r})}_{\text{shading}} h(x_{p,t,r}) dp dt dr$$

$$L(x, \vec{\omega}_o) = L_e(x, \vec{\omega}_o) + \int_{\Omega} L(x', -\vec{\omega}) f_r(x, \vec{\omega}, \vec{\omega}_o) d\vec{\omega}$$

