

5 - Ajout des ombres et de l'illumination de Phong

Objectifs

Notre première implémentation de l'illumination diffuse est trop simpliste : elle est en effet locale, se calcule sans tenir compte des autres éléments de la scène. Des objets peuvent par exemple cacher la source de lumière.

Par ailleurs, la couleur diffuse ne tient pas compte de l'oeil de l'observateur. C'est le modèle d'illumination de Phong qui permet d'intégrer cela.

Nous allons donc implémenter ces deux éléments cette semaine.

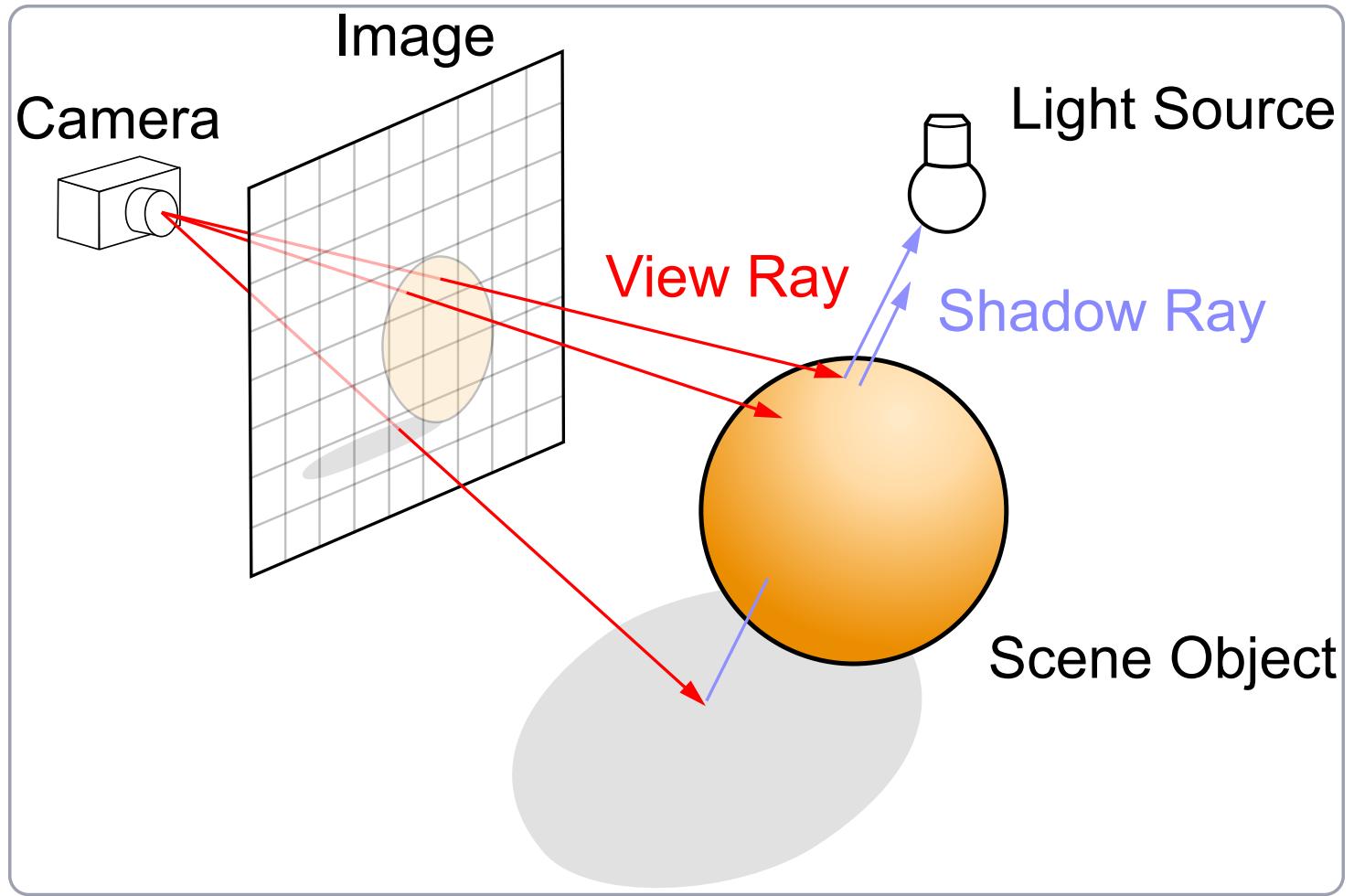
Les ombres

Pour déterminer si une source de lumière doit être comptabilisée, il suffit de vérifier si un objet se trouve entre le point d'intersection et la source de lumière.

Il est possible de réutiliser le calcul d'intersection fait au jalon 3, en utilisant comme source le point d'intersection trouvé et comme direction \vec{d} la direction vers la source de lumière.

Si on trouve une intersection avec un autre objet, alors l'objet ne sera pas éclairé, et on pourra retourner une absence de couleur.

C'est ce que représentent les rayons mauves sur le schéma ci-dessous (Shadow Ray).



⚠️ Attention

Le point sur lequel vous vous situez sera forcément une intersection qui sera trouvée. Une solution est de vérifier que la distance de l'intersection est strictement positive (à un epsilon près).

Si vous ne le faites pas correctement, votre image sera toute piquetée de pixels noirs

Calcul de l'illumination de Phong

La [formule originale](#) est :

- $\vec{r} = -\vec{lightdir} + 2 * (\vec{n} \cdot \vec{lightdir}) * \vec{n}$
- $l_p = \max(\vec{r} \cdot \vec{eyedir}, 0)^{shininess} * \text{lightcolor} * \text{couleur}_{specular}$

✍️ À implémenter

En pratique, on utilise un modèle simplifié, plus facile à calculer, le [Blinn-Phong](#).

- $\vec{h} = \frac{\vec{lightdir} + \vec{eyedir}}{|\vec{lightdir} + \vec{eyedir}|}$
- $l_{bp} = \max(\vec{n} \cdot \vec{h}, 0)^{shininess} * \text{lightcolor} * \text{couleur}_{specular}$

En obtient donc la formule générale pour le calcul de couleur d'un point :

$$color = l_a + \left(\sum_i (\max(\vec{n} \cdot \vec{lightdir}_i, 0) * lightcolor_i * color_{diffuse} + \max(\vec{n} \cdot \vec{h}, 0)^{shininess} * lightcolor_i * color_{specular}) \right)$$

où i représente la $i^{ème}$ source de lumière.

Mise à jour du parser

Le calcul de l'illumination de Phong nécessite un nouveau paramètre `shininess`.

Il va falloir le rajouter au format de fichier (et donc mettre à jour le parser en conséquence) :

```
shininess 20
```

Le comportement sera analogue à celui couleurs `specular` et `diffuse`, il pourra donc être défini plusieurs fois tout au long du fichier.

Aide

Les deux éléments vont se calculer au moment de calculer la couleur du pixel. Si l'intersection est à l'ombre, seule la lumière ambiante sera prise en compte. Sinon, on ajoutera la diffusion de Lambert, puis l'illumination de Phong, et ce pour chaque source de lumière.