Université Gustave Eiffel — INSTITUT D'ÉLECTRONIQUE ET D'INFORMATIQUE GASPARD-MONGE

Master Informatique 1

TD Infographie n°4

Modèles d'illumination

Les modèles d'illumination standards (tels que proposés par OpenGl*, et adaptés par* libg3x*)*

Les modèles issus de l'optique géométrique

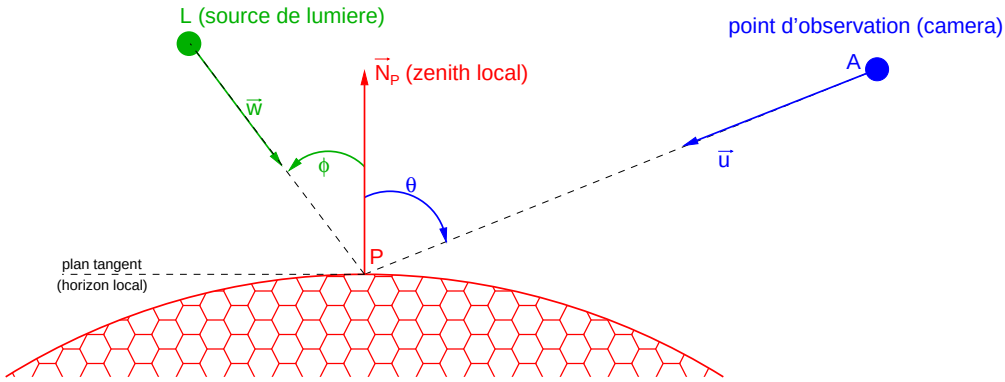
La physique de la lumière décrit de nombreux phénomènes d'interaction avec la matière. Nous nous contenterons ici de modéliser les situations les plus classiques basées sur quelques fondamentaux de l'optique géométrique et des modèles descriptifs (ou *phénoménologique*[(1)]) simples (connus sous les noms de "modèle de Lambert", et "modèle de Phong-Blinn".

- les sources : elles sont ponctuelles, omnidirectionnelles et émettent de la lumière blanche uniforme d'intensité nominale 1.
- les objets : ils n'ont que des propritétés surfaciques simples : diffusion et réflexion (pas de transparence).

On notera :

$[\mathsf{A}, \overrightarrow{u})$	la position et la direction d'observation (Caméra)
$\mathsf{P}, \overrightarrow{\mathsf{N_P}}$	le point de visée et sa normale sur l'objet Ω
θ	l'angle de visée $(\overrightarrow{\mathsf{N_P}}, -\overrightarrow{u})$
$\mathsf{L}, \overrightarrow{w}$	la position de la source et la direction $\overrightarrow{\mathsf{PL}}$
ϕ	l'angle d'incidence de la lumière en P $(\overrightarrow{\mathsf{N_P}}, \overrightarrow{w})$

Tous les vecteurs présents $(\overrightarrow{u}, \overrightarrow{\mathsf{N}}, \overrightarrow{w})$ sont supposés normés.



La modélisation peut se formaliser en 3 modèles complémentaires qui peuvent être traités indépendamment (ici pour une seule source lumineuse).

(1) on donne une description d'un phénomène sans trop décrire les causes physiques qui le produisent. Ces modèles sont souvent un peu empiriques, ce qui peut les rendre difficile à utiliser.

① la composante "ambiante"

Elle simule le fait que la source lumineuse produit, via des réflexions multiples que l'on ne modélise pas, une lumière ambiante qui baigne l'ensemble de la scène de manière uniforme. Ainsi tout point de la scène, même s'il n'est pas directement illuminé, reçoit une petite portion de lumière et en renvoie un peu, le reste étant "absorbé". On associe donc à l'objet un premier coefficient $\mathsf{ambi} \in [0., 1.]$ (qui doit rester très faible $\mathsf{ambi} \simeq 0.1$).

Cette première composante est alors simplement la couleur de tout point P de l'objet Ω pondérée par le coefficient propre à l'objet : $\mathsf{Col(P)} \mathrel{+}= \mathsf{ambi} \times \mathsf{Col}^{\Omega}$.

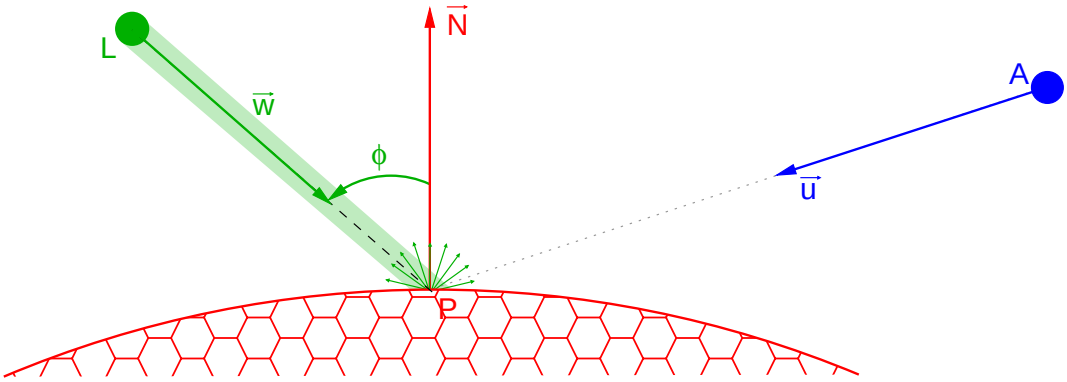
Ce modèle seul ne suffit pas à rendre la forme tri-dimensionnelle des objets puisque, précisément, la quantité de lumière renvoyée est la même en tout point de l'objet. Tout ce que l'on peut voir c'est la forme de sa projection 2D : une sphère apparaît toujours comme un cercle uniforme.

Comme en photographie, pour mettre en valeur la forme tri-dimensionnelle il est nécessaire de prendre en compte la position de la lumière par rapport à l'objet.

② la composante "diffuse"

Comme la suivante, cette composante simule le fait que l'intensité lumineuse reçue par un point de la surface dépend de l'angle d'incidence ϕ de la lumière : pour $\phi = 0$ (lumière au *zénith*) celle-ci est maximale, elle décroit jusqu'à $|\phi| = \pi/2$ (lumière à l'*horizon*) puis est nulle (la lumière passe "sous l'horizon" : le point n'est plus éclairé). Cette quantité de lumière est donc dépendante de $\cos(\phi) = \vec{\mathsf{N}} \bullet \vec{\mathsf{w}}$.

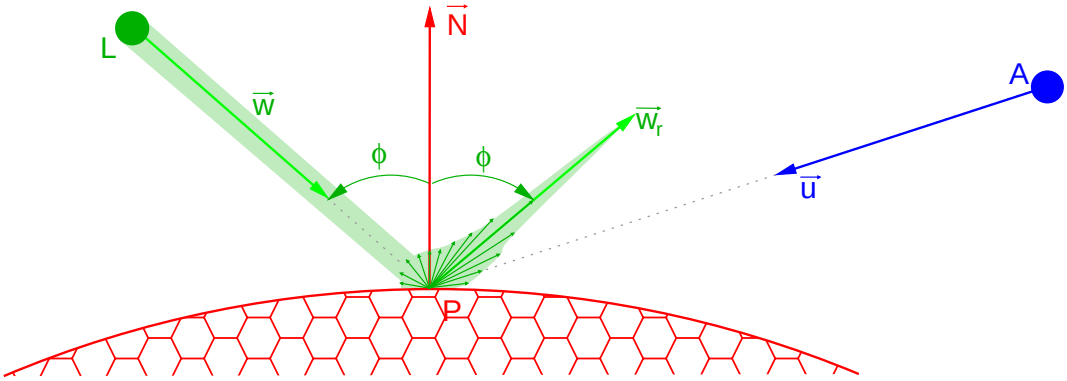
Si l'objet est mat, la lumière est en partie renvoyée par la surface dans toutes les directions et l'illumination ne dépend pas du point de vue (source du rayon A). Une autre par de la lumière reçue est absorbée par la surface. C'est la proportion (pourcentage) de lumière réfléchie que représente le *coefficient de diffusion* $\mathsf{diff} \in [0, 1]$.



Cette seconde composante est simplement la couleur de l'objet Ω pondérée par l'angle d'incidence ϕ et le coefficient de diffusion diff propre à l'objet : $\mathsf{Col(P)} \mathrel{+}= \mathsf{diff} \times (\vec{\mathsf{N}}_{\mathsf{P}} \bullet \vec{\mathsf{w}}) \times \mathsf{Col}^{\Omega}$.

③ la "tâche spéculaire"

Les objets peuvent être un peu mats et/ou brillants. Une surface "brillante" va renvoyer la lumière dans une direction privilégiée $\vec{\mathsf{w}}_{\mathsf{r}}$, symétrique à la direction d'incidence $\vec{\mathsf{w}}$ par rapport à la normale $\vec{\mathsf{N}}$.



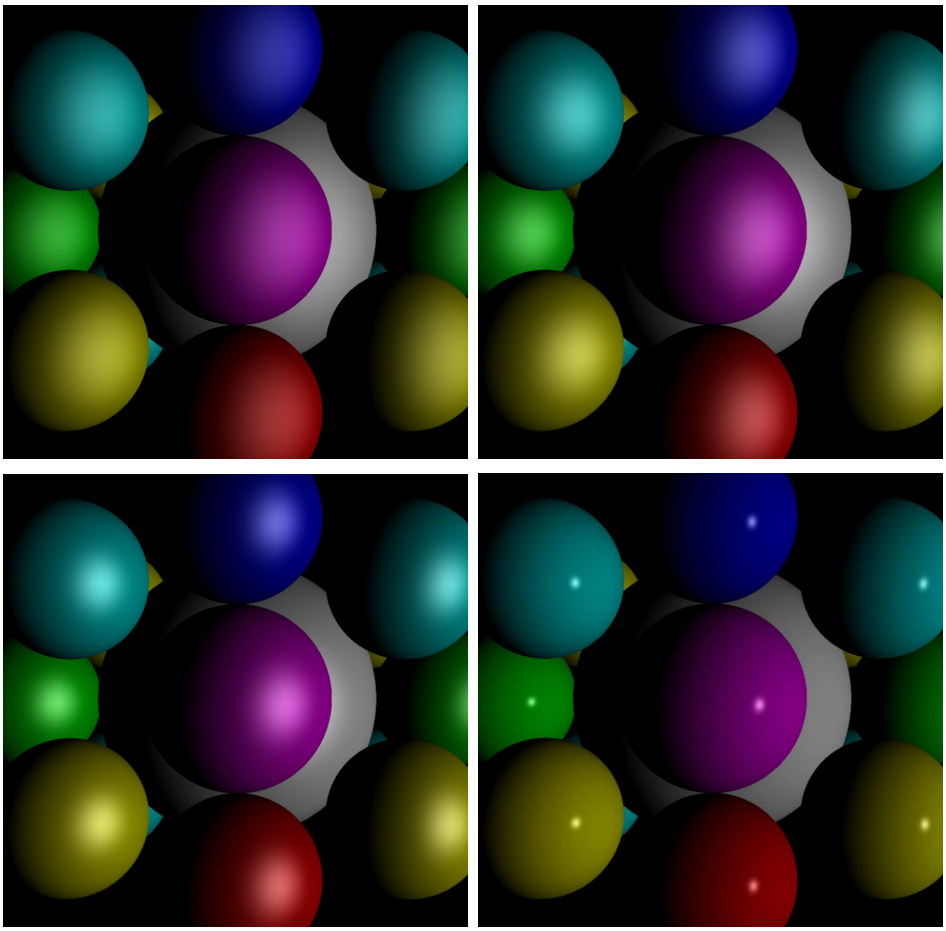
Il faut donc commencer par calculer la direction $\overrightarrow{w_r}$ de la lumière réfléchie : par construction même, $\overrightarrow{w} + \overrightarrow{w_r} = 2 \times \cos(\phi) \times \overrightarrow{N}$, soit : $\overrightarrow{w_r} = 2 \times (\overrightarrow{N} \bullet \overrightarrow{w}) \times \overrightarrow{N} - \overrightarrow{w}$

La quantité de lumière renvoyée vers l'observateur (de position A) depuis un point P de la surface de l'objet dépend alors de l'angle $(\theta - \phi)$ entre la direction d'observation $\overrightarrow{u}$ et la direction de rélfexion $\overrightarrow{w_r}$: elle est maximale lorsque $\theta = \phi$ et décroit très rapidement dès que $|\theta - \phi|$ augmente.

☞ La couleur de la "tâche spéculaire" est celle de la lumière (donc blanc, en général).

Pour simuler cette décroissance plus ou moins rapide, on introduit un "coefficient de brillance" propre à l'objet ($\text{shine} \in [0,1]$) ainsi qu'un paramètre $\text{spec} \in [0,1]$ qui caractérise la "brillance" de l'objet et on modélise la brillance par : $\text{Col(P)} \mathrel{+}= \text{spec} \times \text{shine} \times (-\overrightarrow{u} \bullet \overrightarrow{w_r})^{\frac{1}{1-\text{shine}}} \times \text{Col}^L$.

Pour des valeurs $\text{diff} = \text{spec} = 0.5$ et des valeurs de shine de 0.2, 0.5, 0.8 et 0.99 on obtient



Attention : il s'agit là d'un modèle très empirique et de nombreuses variantes existent. En particulier, OpenGl n'utilise pas tout à fait le même (c'est un modèle surfacique).

④ réflexions spéculaires (effet miroir), réfraction (transparence), ombres...

Le modèle d'illumination proposé nativement par OpenGl ne va pas plus loin. Pour modéliser des interactions lumière-matière plus avancées telles que la réflexion, la réfraction, l'ombrage ou des effets surfacique de "rugosité", il faut recourir à des modèles plus complexes tels que le RayTracing, éventuellement combiné à du BumpMapping ou NormalMapping (cf. cours).

Mise en oeuvre concrète OpenGl et libg3x

OpenGl propose nativement ces 3 modèles d'illumination (plus quelques autres options), mais leur mise en oeuvre est assez peu conviviale.

☞ cf. manuel OpenGl https://www.khronos.org/opengl/wiki/How_lighting_works

Pour simplifier tout ça, la sur-couche libg3x prédéfinie la plupart des variables d'état GL nécessaires (en limitant, de fait, l'utilisation que l'on peut en faire) et propose une fonction permettant de régler plus *naturellement* les paramètres de couleur/matière des objets.

La fonction void g3x_Material(), disponible dans le module <g3x_colors.c> fait celà directement. Elle prend en paramètres la couleur de l'objet et des valeurs ($\{ambi, diff, spec, shin, alpha\} \in [0.,1.]$) directement représentatifs des propriétés "physiques" du matériau. Le dernier paramètre (alpha – "pseudo-transparence") ne sera pas discuté ici. Il doit rester à alpha=0.

Cette fonction doit être appelée, dans la fonction de dessin, juste avant l'appel aux routines d'affichage (cf. TD2.Dessin et TD3.Modélisation) pour mettre à jour les paramètres OpenGl (variables d'état) de couleur/matière. Ces paramètres resteront les mêmes jusqu'au prochain appel.

... en amont : l'activation des fonctionalités lumière/caméra

La libg3x active[(2)] par défaut une source lumineuse et une caméra (types G3Xlight et G3Xcamera définies dans <g3x_camlight.h>)

Les deux sont initialement positionnées au même endroit (coordonnées $[5,5,5]$) et la caméra est dirigée vers l'origine du repère global – $[0,0,0]$.
☞ Cela définit donc, par défaut, les éléments $[A, \vec{u})$ et $[L, \vec{w})$ définis en introduction.

Une source lumineuse est essentiellement caractérisée par une position et une couleur. Le type G3Xlight en contient 3 (c'est le modèle OpenGl...), mais par défaut elle seront toutes blanches – $[1,1,1,0]$. On considère qu'elle diffuse de la lumière de manière omnidirectionnelle.

La caméra, dont les attributs sont réglés par défaut, sera détaillée ultérieurement, si nécessaire.

☞ Le programme d'exemple g3x_demo.c qui vous a été fourni avec la libg3x illustre l'utilisation de ces fonctionalités et de leurs paramètres.

trucs en plus...

OpenGl étant "bridé" en mode "rendu 3D" par la couche libg3x, on ne peut, par défaut, faire que de l'affichage de surfaces (interactions avec la lumière).

On ne peut donc pas, tel quel, tracer des lignes (pour visualiser des normales, des rayons, des trajectoires...).

Pour avoir accès à ces possiblitiés, il suffit de "débrancher" la lumière, tracer ses lignes, polygônes..., puis "rallumer".

```
-------------------------------------------------------
  glDisable(GL_LIGHTING);    /* on éteint la lumière   */
    glColor3f(1.,1.,1.);     /* la couleur de tracé    */
    glBegin(GL_LINES);       /* on passe en mode LINE  */
      glVertex3d(x1,y1,z1);  /* un 1° point (x1,y1,z1) */
      glVertex3d(x2,y2,z2);  /* un 2° point (x2,y2,z2) */
    glEnd();                 /* fin du tracé           */
  glEnable(GL_LIGHTING);     /* on rallume la lumière  */
-------------------------------------------------------
```

[(2)] cf. fonction g3x_MainStart() dans < g3x_Window.c > (lignes 600-620), pour voir à quoi ça ressemble en GL 'natif'