

Des effets d'illumination plus générale

- On a effectuer 2 simplifications sur l'équation d'illumination directe

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

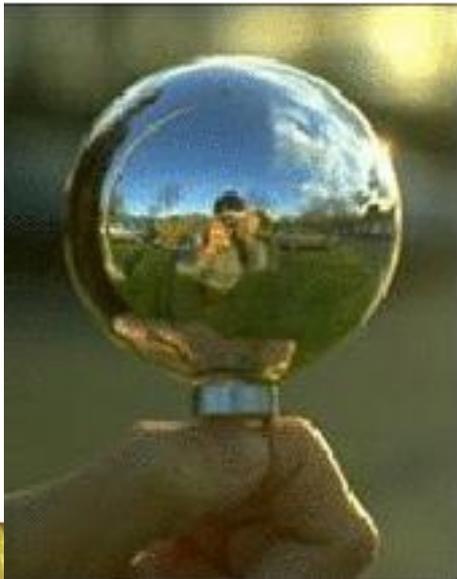
pour calculer les effets d'*ambient occlusion* et de *reflection occlusion*

- Le plus nombreux les suppositions, le moins réaliste notre simulation
- Pour générer des images plus réalistes, nous enlèverons un de nos suppositions:
 - Nous utiliserons un modèle de lumière plus complexe pour $L_{in}(x, \vec{\omega})$
 - mais nous continuerons d'utiliser nos modèles de réflexion simple (diffus et Phong) pour $f_r(x, \vec{\omega}, \vec{\omega}_o)$



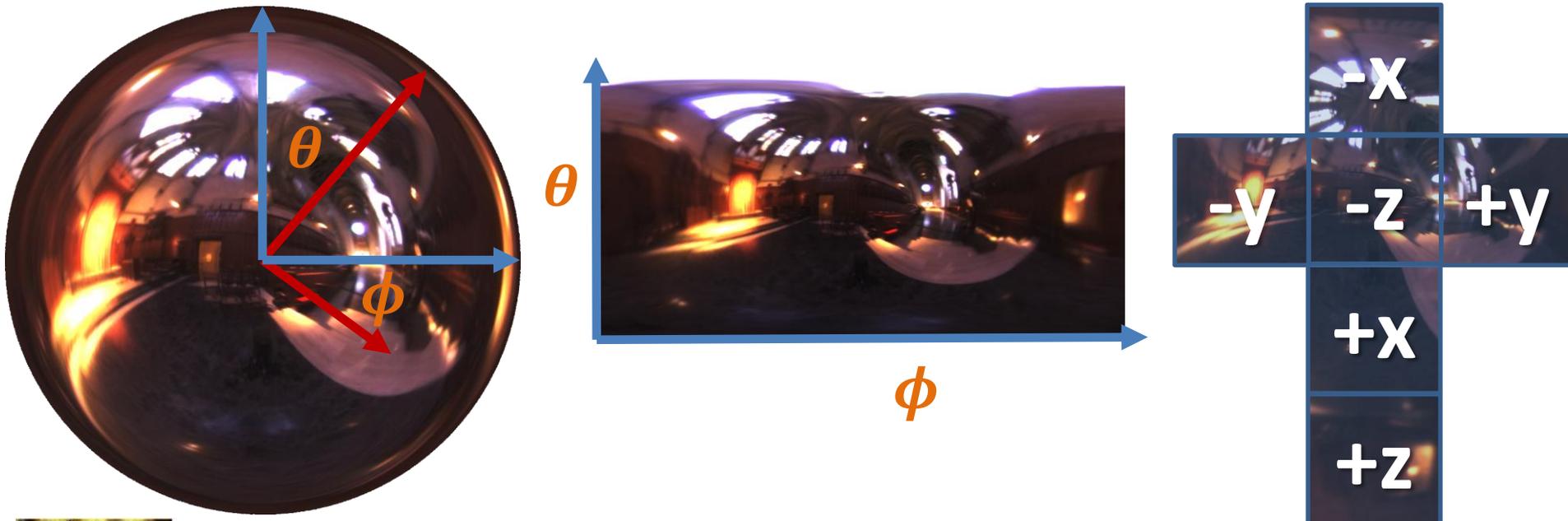
Sources de lumières plus complexes

- Présentement on a seulement considéré des sources de lumières *analytiques*: ponctuelles, directionnelles, et ciel uniforme
- Beaucoup de sources d'illumination sont trop complexes à modéliser, on peut *capturer* et enregistrer leurs intensités
- Par exemple, si on photographie une sphère miroir dans une salle, on peut capturer les conditions d'illumination dans la salle



Sources de lumières plus complexes

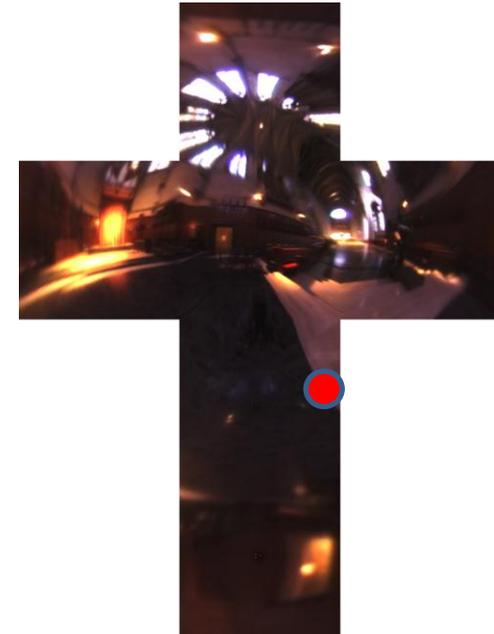
- Ces données peuvent être réutilisées comme représentation de $L_{in}(x, \omega)$ dans notre équation d'illumination directe
- Il y a plusieurs paramétrisations d'image (texture) qu'on peut utiliser pour représenter ces fonctions sphériques capturés:



Sources de lumières plus complexes

Comment implémenter $L_{in}(x, \omega)$ en code?

```
couleur L_in(tex, omega_i) {  
    return lookup(texture, omega_i);  
}
```



Comment effectuer du *importance sampling*?

- On à déjà vu comment choisir des *pdf* pour les modèles de réflexion diffus et Phong (*importance sampling* selon la BRDF f_r)
- Mais, maintenant qu'on a incorporé une terme d'illumination plus complexes dans notre intégrande : $f(\omega) = L_{in} \times f_r \times V$
 - Est-ce qu'on peut effectuer du importance sampling plus précis?
- Réponse: Oui! 😊
- Est-ce qu'on va dériver ces *pdfs* plus complexes dans ce cours: Non. 😞
- Nous continuerons d'utiliser des *pdfs* bassées sur la BRDF f_r



Réflexion Phong avec des lumières complexes

- Comment calculer l'illumination directe d'une sphère Phong illuminée par une source de lumières complexes capturée avec des photos?

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

$$L_o(x, \omega_o) \approx \frac{1}{N} \sum_i L_{in}(x, \omega_i) \frac{C \max(R \cdot \omega_i, 0)^n}{pdf(\omega_i)} V(x, \omega_i) = \frac{1}{N} \sum_i L_{in}(x, \omega_i) V(x, \omega_i)$$



$n = 1$

$n = 2$

$n = 20$

$n = 50$

$n = 200$

$n = 10000$



Réflexion Phong avec des lumières complexes

- Code pour: $L_o(x, \omega_o) \approx \frac{1}{N} \sum_i L_{in}(x, \omega_i) \frac{C \max(R \cdot \omega_i, 0)^n}{pdf(\omega_i)} V(x, \omega_i)$

```
couleur shading(x, omega_o) {
    intégral = couleur(0,0,0);

    for( i = 0 à N-1 ) {
        omega_i = générerÉchantillonSelonPhong(x.brdf.n);
        Ray ombre(x + epsilon * x.n, omega_i);
        if( ombre.intersect(scene) != true ) {
            intégral += L_in(tex, omega_i);
        }
    }

    return intégral / N;
}
```



$f_r(x, \vec{\omega}, \vec{\omega}_o)$ sous forme de tableaux

- Comme avec les lumières complexes (L_{in}), on peut aussi utiliser des données tabulées capturées à partir du monde réel pour les BRDFs (f_r):

```
float f_r[theta_i][phi_i][theta_o][phi_o][couleurs]
```

- Il est encore possible d'effectuer un *importance sampling* sur ces BRDF tabulés, mais on ne va pas l'étudier ici

```
float f_r(omega_i, omega_o) {  
    return f_r[omega_i.t][omega_i.p]  
           [omega_o.t][omega_o.p];  
}
```

```
couleur f_r(omega_i, omega_o) {  
    return couleur(&f_r[omega_i.t]  
                 [omega_i.p]  
                 [omega_o.t]  
                 [omega_o.p]  
                 [0]);  
}
```

