IFT3355: Infographie
Sujet 5: `shading` 2
(illumination *locale* 2)

Derek Nowrouzezahrai
Département d'informatique et de recherche opérationelle
Université de Montréal

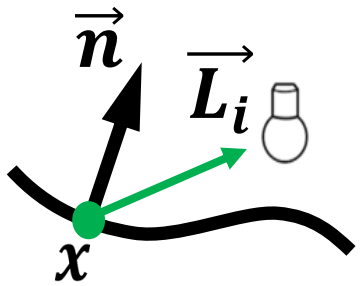
Survol: lancer de rayons secondaires

- Le premier rayon identifie la surface visible de l'œil où on peut effectuer notre calcul de shading
  - Pour les objets **opaques** et **non réfléchissant** ce valeur peut être obtenu avec ex: le modèle d'illumination présenté plus tôt
- Dans ces cas, d'autres rayons peuvent être tracés pour déterminer, par exemple, la visibilité des lumières et calculer leurs contributions
- Des rayons secondaires peuvent aussi être utiliser pour déterminer le shading des objets (semi-)**transparentes** et/ou **réfléchissants**

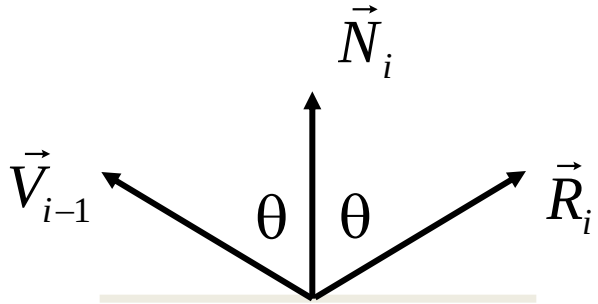
- On doit s'assurer de ne pas ré-intersecter l'objet au nouveau point de départ (*surface acné*)
  - ajouter une distance epsilon au rayon
  - ne pas intersecter la surface elle-même
    (plus limité: surface convexe et test par rapport à la normale)

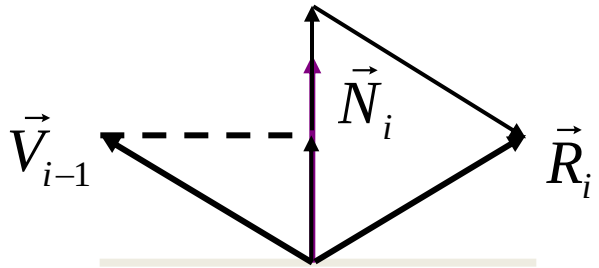
Lancer de rayons: types de rayons

- Rayons d'ombre: $\vec{L}_i = \dfrac{(L - P_i)}{\|(L - P_i)\|}$



- Rayons réfléchis: $\vec{R}_i = 2(\vec{V}_{i-1} \cdot \vec{N}_i)\vec{N}_i - \vec{V}_{i-1}$

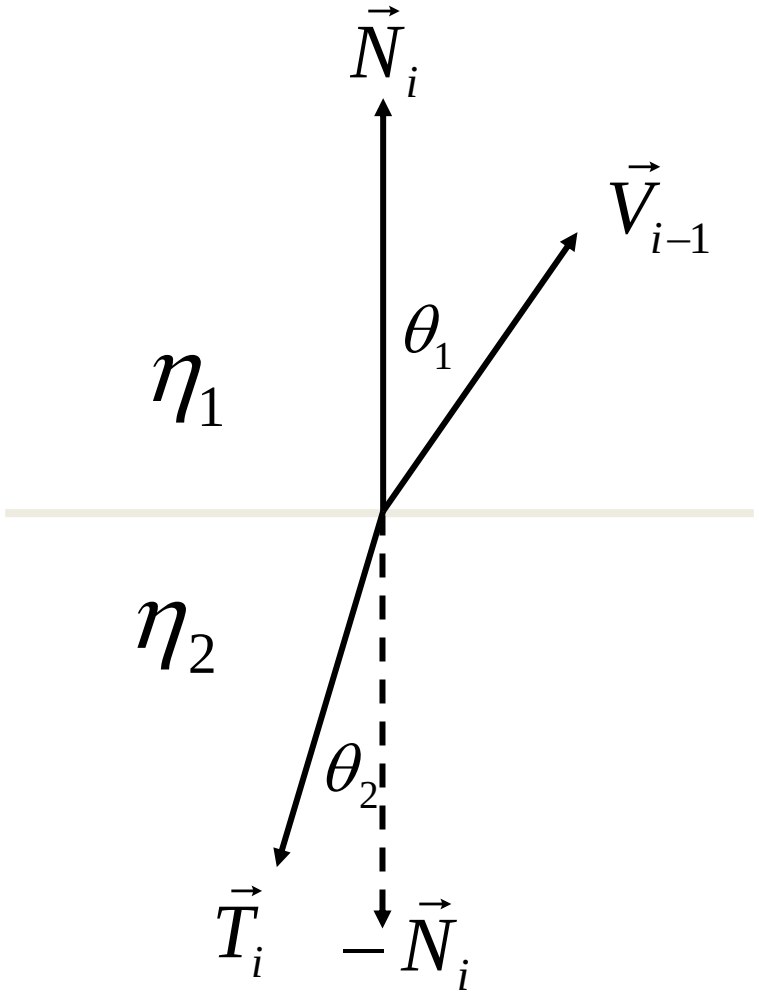




Lancer de rayons: types de rayons

- Rayons transmis / réfractés:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{\eta_2}{\eta_1}$$

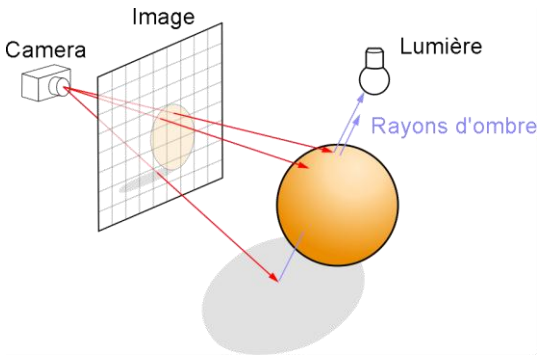


Ombres

- Déterminer si un objet est dans l'ombre pour une lumière consiste à tester la visibilité de la lumière à partir de l'objet
- Cette visibilité est simplefiée: il faut seulement identifier **si** un objet bloque la lumière, et non pas l'objet le plus proche

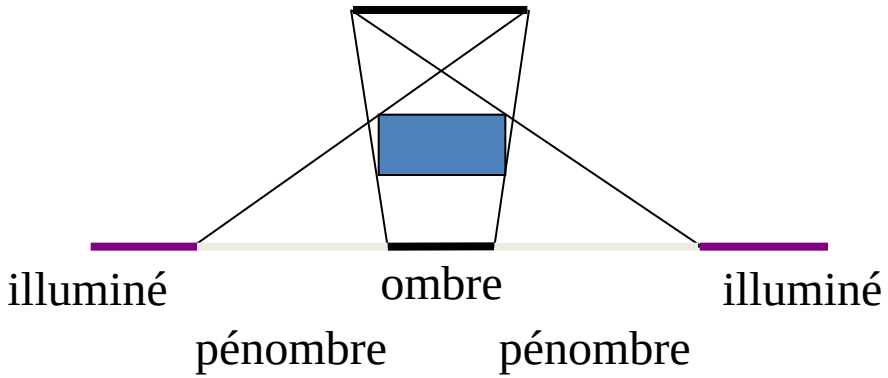
```
for( chaque pixel p )
  Générer rayon r de l'oeil à p
  float dist_min = infinité
  intersection h = null
  for( chaque objet o )
    if( trace( r, o ) et
        hit_dist <= dist_min )
      dist_min = hit_dist
      h.objet = o
      h.dist = hit_dist
p.couleur = shade(h)
```

```
bool dans_ombre = false;
for( chaque objet o )
  if( trace( r, o ) )
    dans_ombre = true;
    break;
```



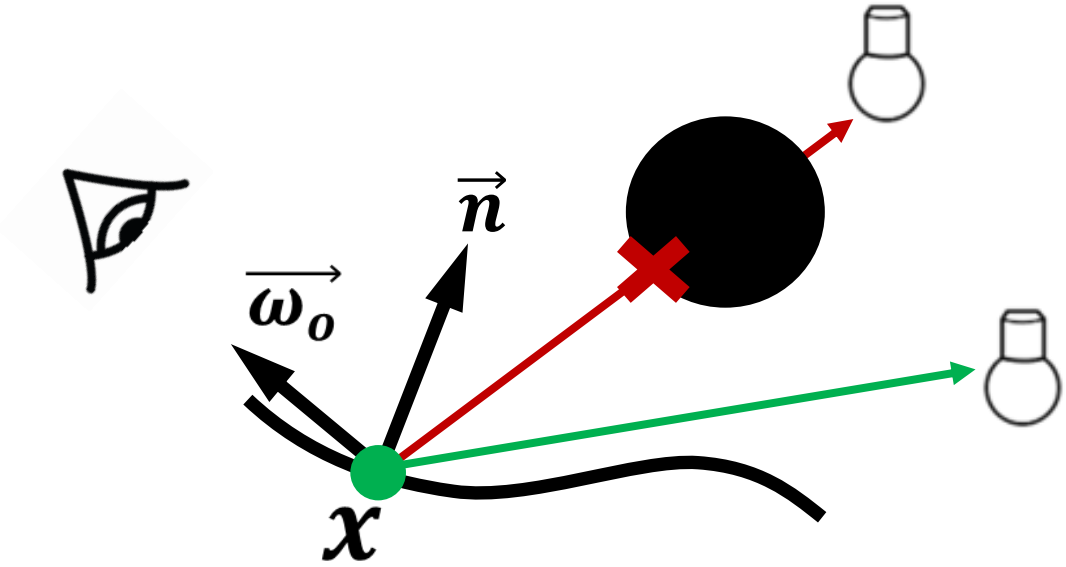
Ombres

- Pour une lumière directionnelle et ponctuelle, la décision est binaire. Un point est illuminé ou est dans l'ombre.
- Pour une lumière étendue (*e.g.* linéaire ou polygonale), on doit calculer la proportion de la lumière visible du point. Un point peut alors être dans l'ombre, complètement illuminé, ou dans la pénombre.

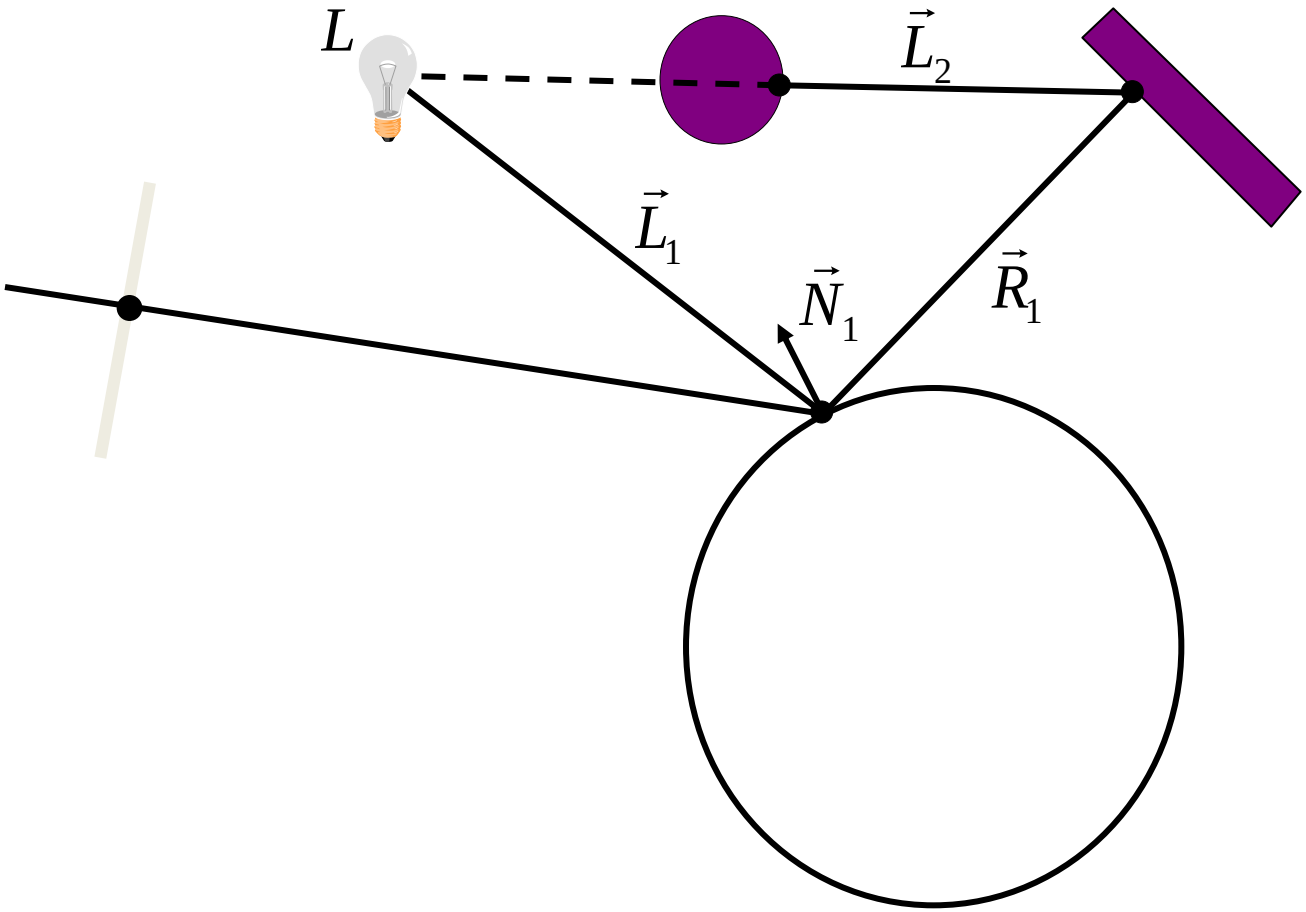


Calcul des ombres avec les rayons (pour lumière directionnelle et ponctuelle)

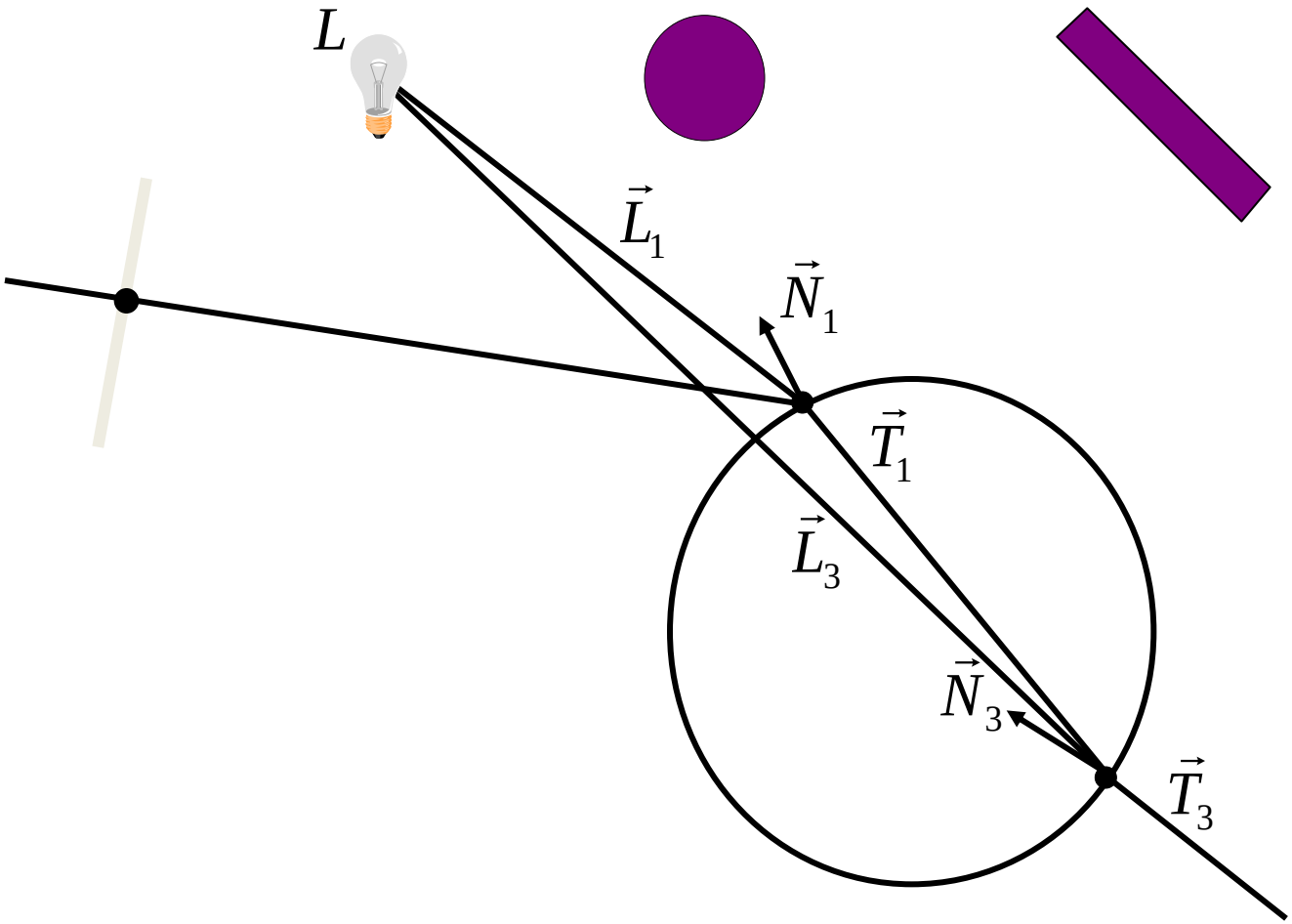
- Trace un rayon d'ombre du point jusqu'à la lumière
- S'il n'y a aucune intersection sur ce segment, le point est illuminé



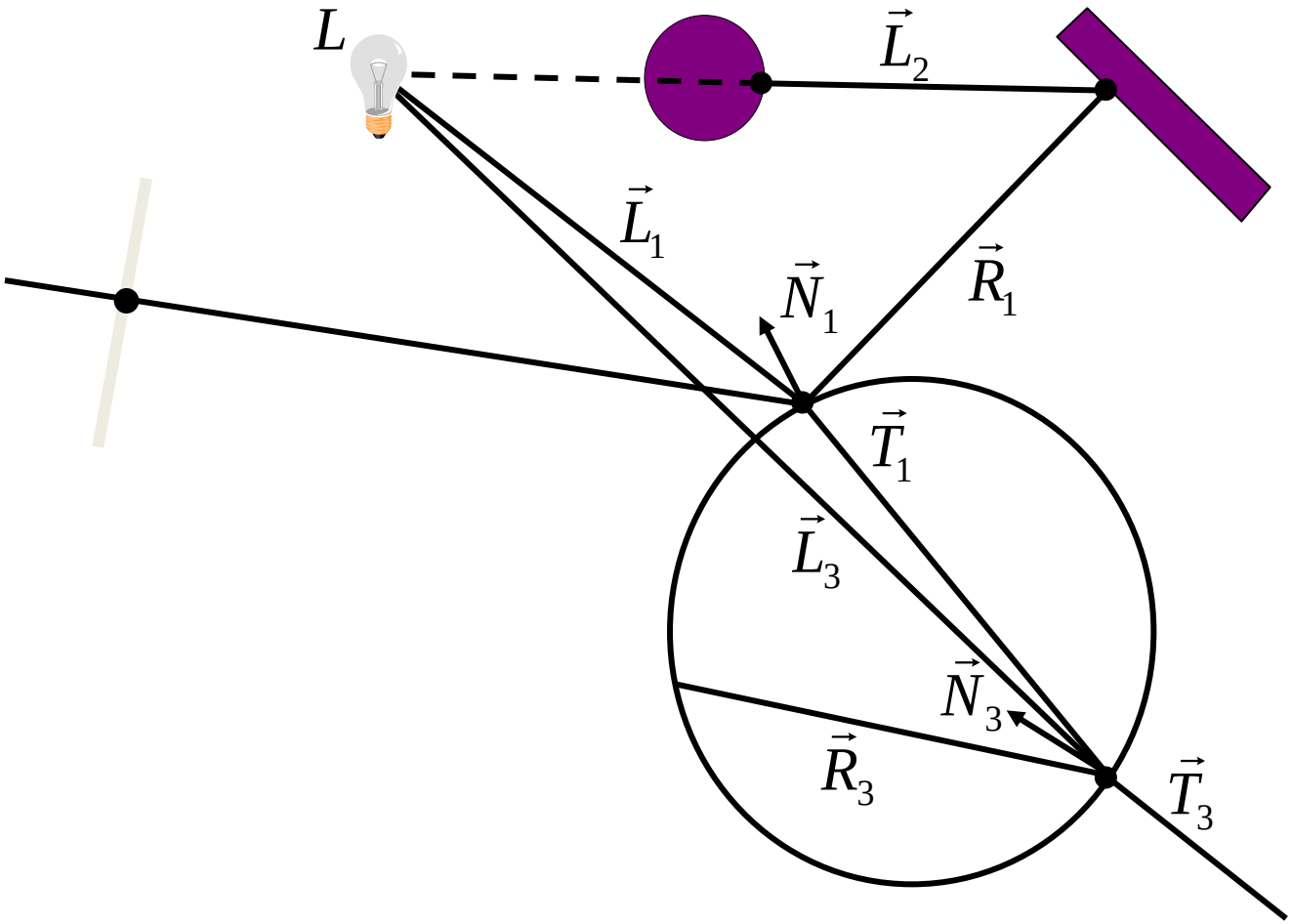
Lancer de rayons récursif: réflexion



Lancer de rayons récursif: transmis



Lancer de rayons récursif: réflexion + transmission



Lancer de rayons: réfraction

$$\vec{T}_i = \vec{M}\sin\theta_2 - \vec{N}_i\cos\theta_2$$

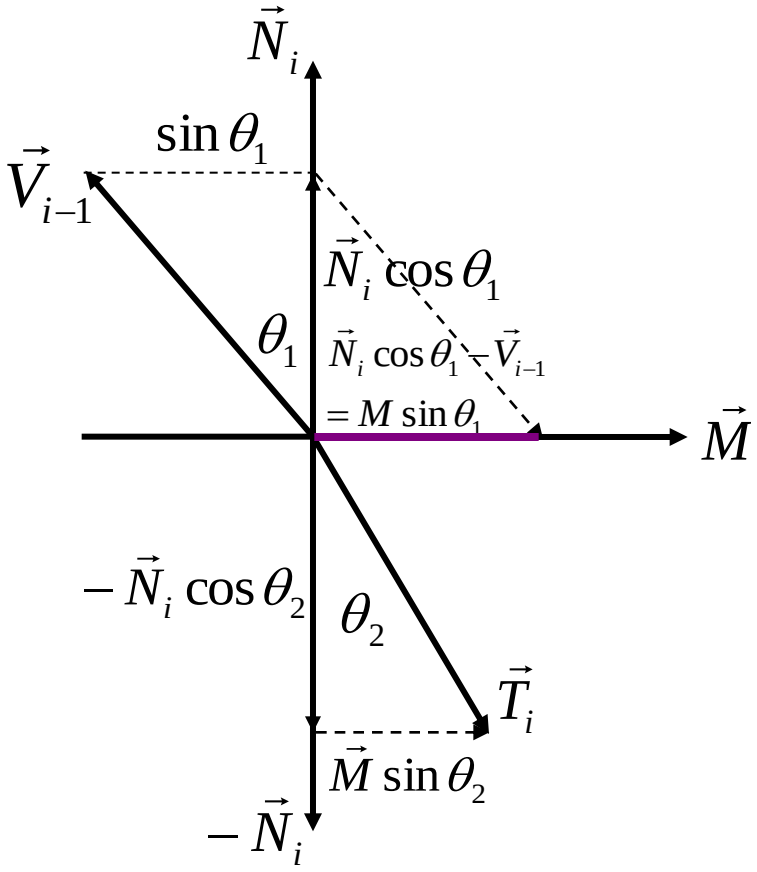
$$\vec{M} = \frac{(\vec{N}_i\cos\theta_1 - \vec{V}_{i-1})}{\sin\theta_1}$$

$$\vec{T}_i = \frac{(\vec{N}_i\cos\theta_1 - \vec{V}_{i-1})}{\sin\theta_1}\sin\theta_2 - \vec{N}_i\cos\theta_2$$

$$\vec{T}_i = \frac{\eta_1}{\eta_2}(\vec{N}_i\cos\theta_1 - \vec{V}_{i-1}) - \vec{N}_i\cos\theta_2$$

$$\eta = \eta_1 / \eta_2$$

$$\vec{T}_i = \vec{N}_i(\eta\cos\theta_1 - \cos\theta_2) - \eta\vec{V}_{i-1}$$



$$\sqrt{1-\sin^2\theta_2} = \sqrt{1-\eta^2\sin^2\theta_1} = \sqrt{1-\eta^2(1-\cos^2\theta_1)}$$

$$\vec{T}_i = \vec{N}_i\left(\eta(\vec{N}_i\cdot\vec{V}_{i-1}) - \sqrt{1-\eta^2(1-(\vec{N}_i\cdot\vec{V}_{i-1})^2)}\right) - \eta\vec{V}_{i-1}$$

Lancer de rayons distribués: autres effets

- Anti-aliassage
  - échantillonnage régulier
  - échantillonnage stratifié (*jitter*)
- Pénombres
  - échantillonner la lumière
  - échantillonner l'angle solide formé par la lumière
- Profondeur de champ
  - échantillonner les lentilles (simulées) d'une caméra
- Réflexion *glossy*
  - échantillonner le cone de réflexion
- Flou de mouvement
  - échantillonner (le mouvement dans) le temps