Ombres

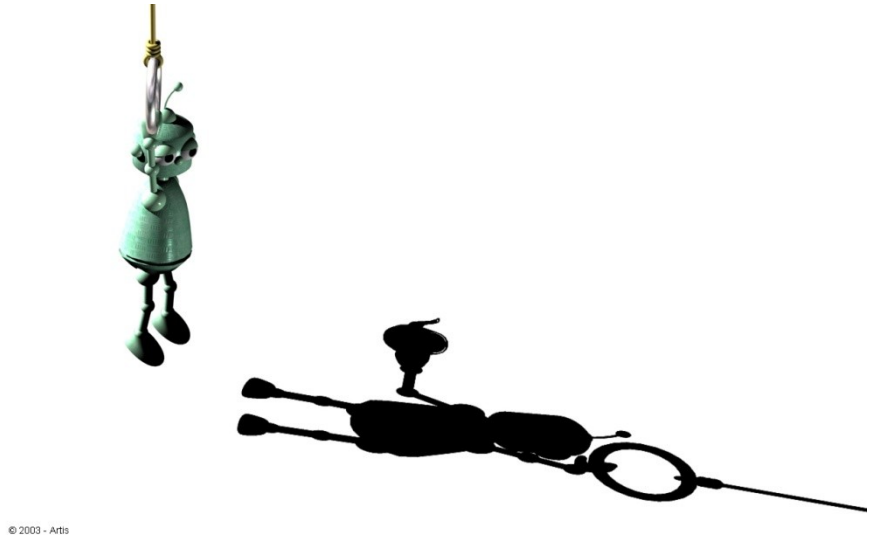
Joëlle Thollot
Joelle.Thollot@imag.fr

Ombre et lumière

- Les ombres augmentent le réalisme

Ombre et lumière

- Les ombres aident à percevoir
  - des objets cachés

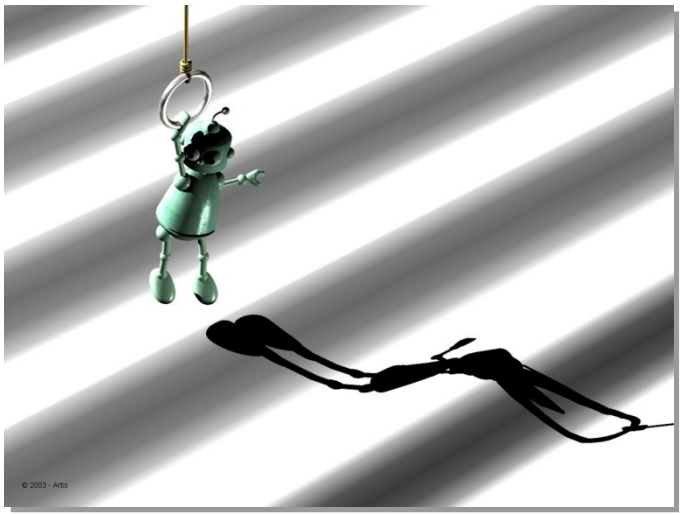


Ombre et lumière

- Les ombres aident à percevoir
  - des objets cachés
  - le positionnement relatif des objets

Ombre et lumière

- Les ombres aident à percevoir
  - des objets cachés
  - le positionnement relatif des objets
  - la forme des objets



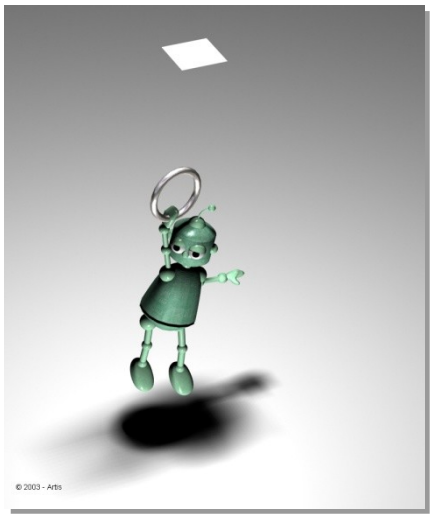
Ombre et lumière

- Contraintes pour les ombres en temps-réel
  - Lampes — Dynamiques
  - *Shadow Casters* — Dynamiques
  - *Shadow Receivers* — Dynamiques

Ombre et lumière

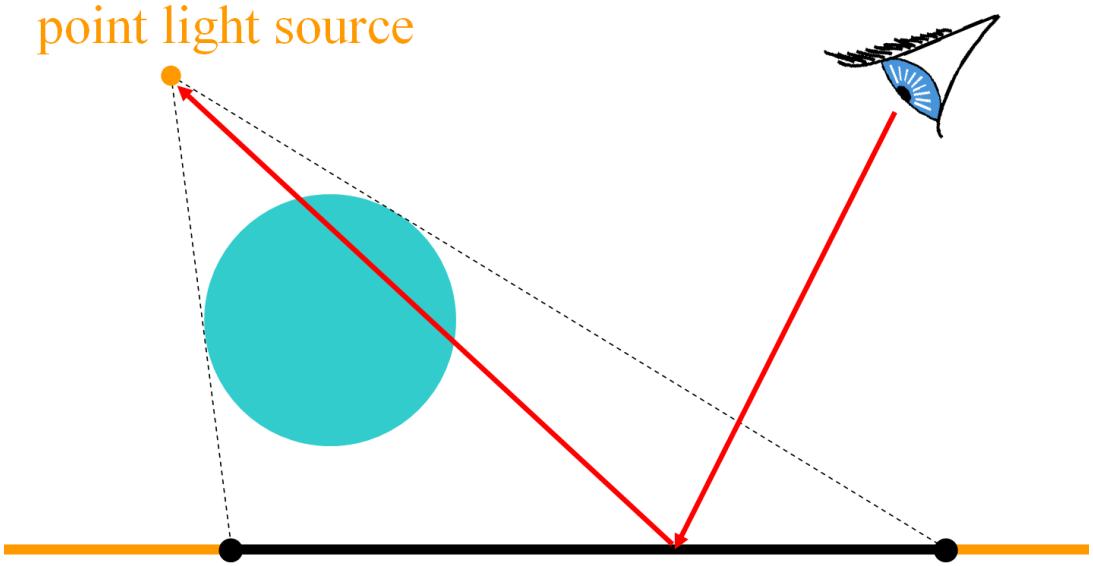
- Type d'ombres
  - **Ombres dures**
    - Hard Shadow
    - Source Ponctuelle

  - **Ombres douces**
    - Soft Shadow
    - Source Etendue



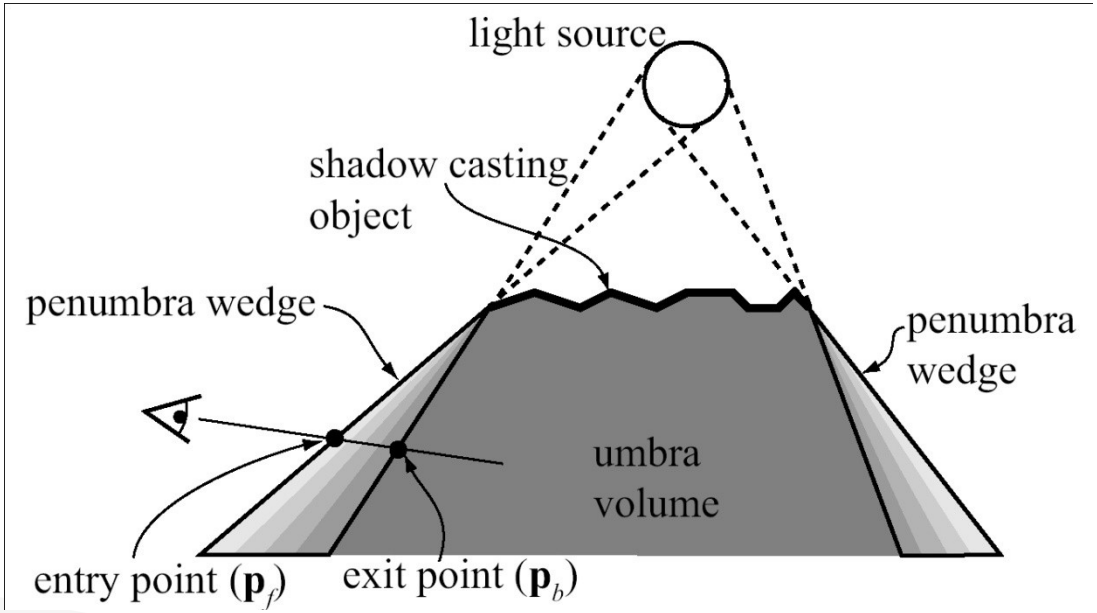
Ombre dure

- Quand la source est ponctuelle
- Un point est dans l'ombre s'il ne voit pas la source



Ombre douce

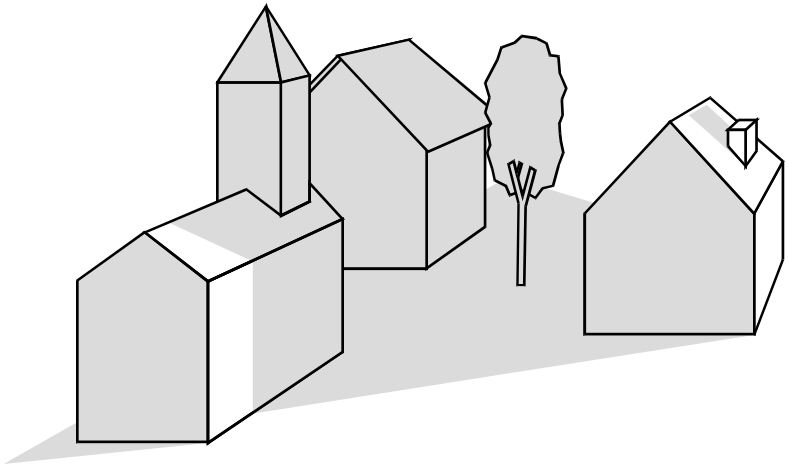
- 3 zones :
  - Ombre : source totalement cachée
  - Pénombre : source partiellement cachée
  - Eclairée : source totalement visible

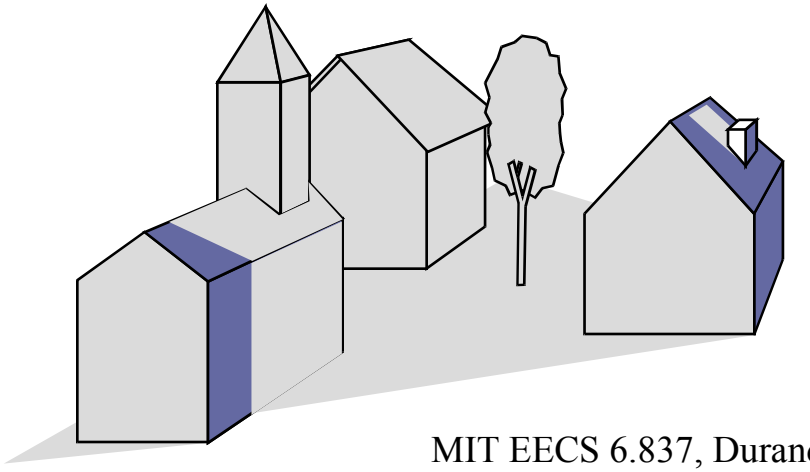


Calcul des ombres dures

Dualité ombre/vue

- Un point est éclairé s'il est visible de la source
- Le calcul des ombres est donc similaire au calcul d'une vue

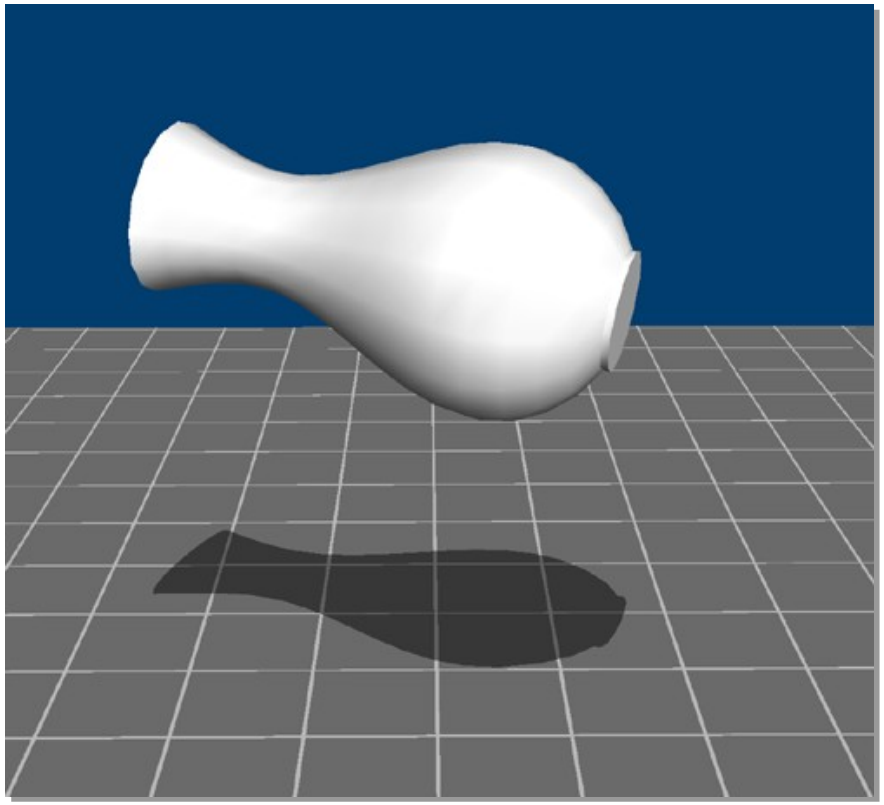


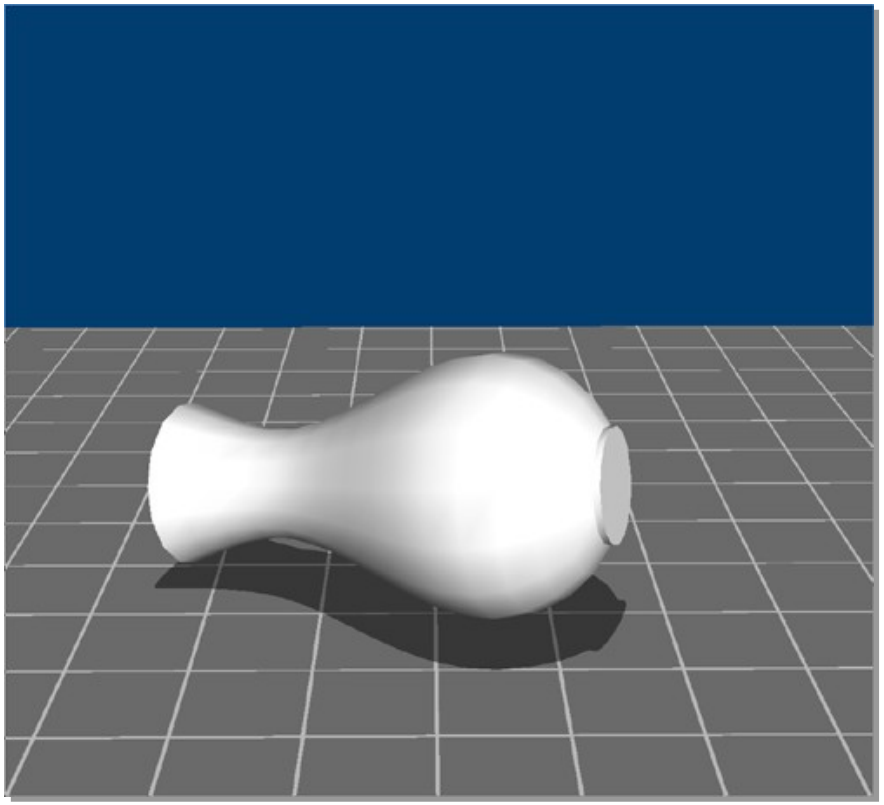


MIT EECS 6.837, Durand and Cutler

Ombres planes

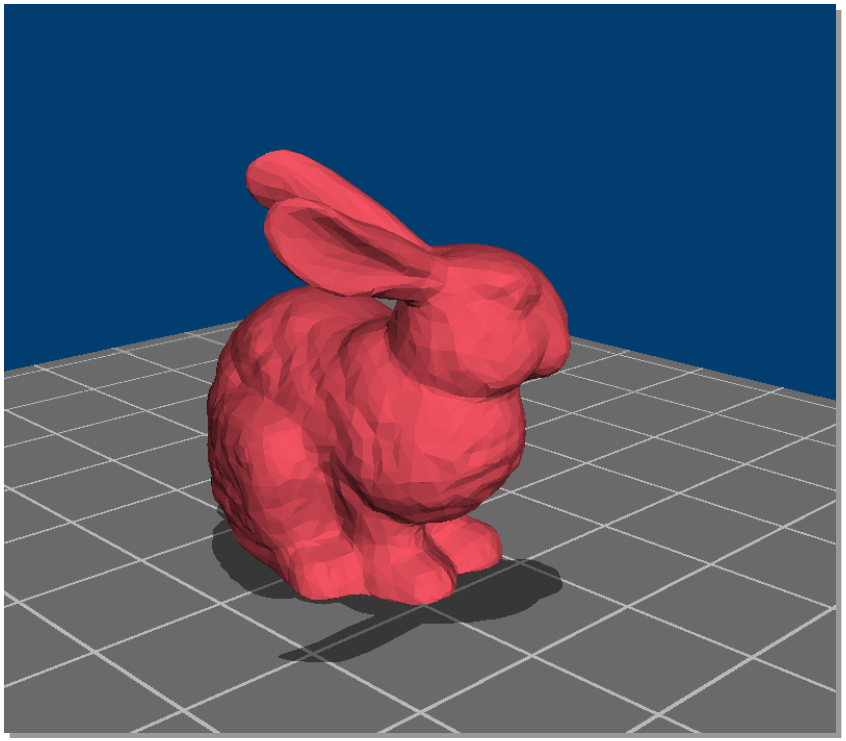
- Dessiner les primitives une seconde fois projetées sur le sol

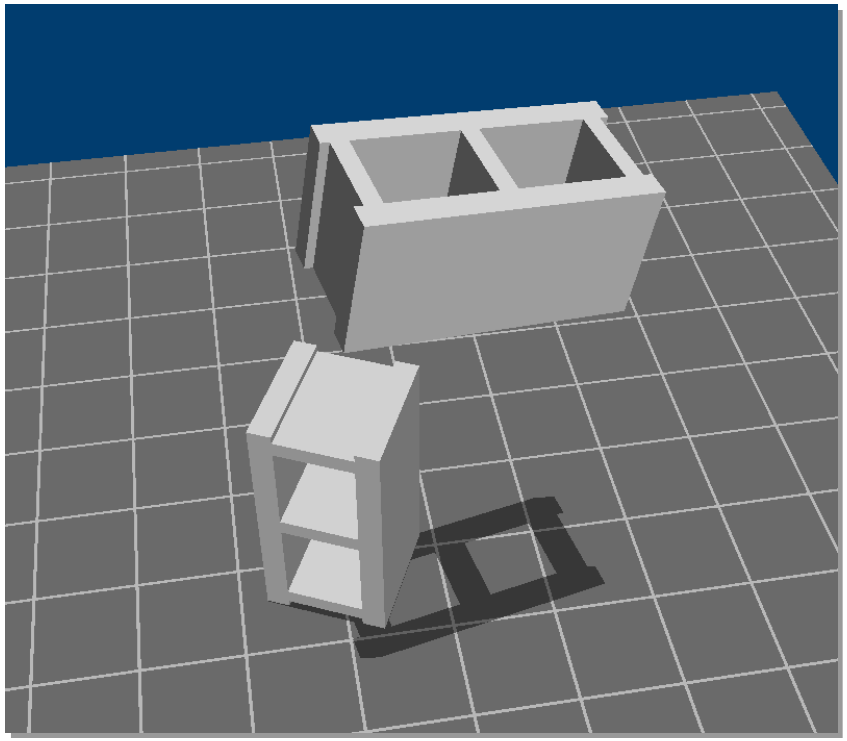




Ombres planes +/-

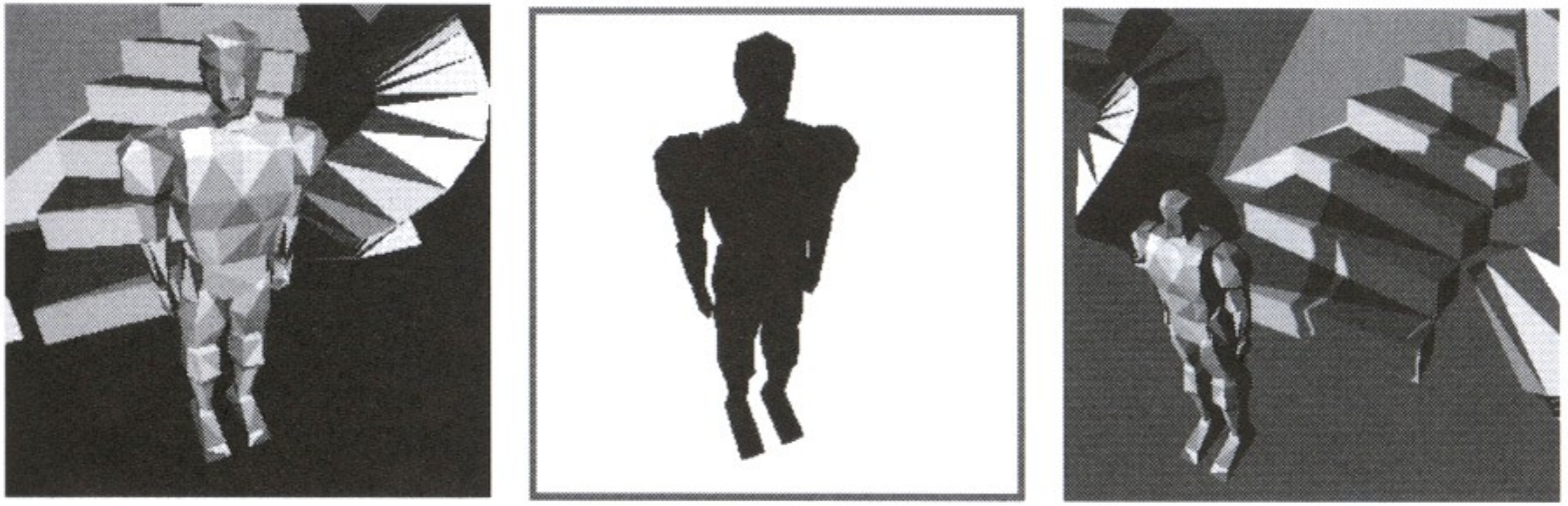
+ Simple et efficace
- Pas d'auto ombrage, pas d'ombres sur des surfaces courbes, sur d'autres objets





Utilisation des textures

- Séparer obstacle et récepteur
- Calculer une image de l'obstacle vu de la source
- L'utiliser comme texture sur le récepteur



Vues de la source Vue de l'œil

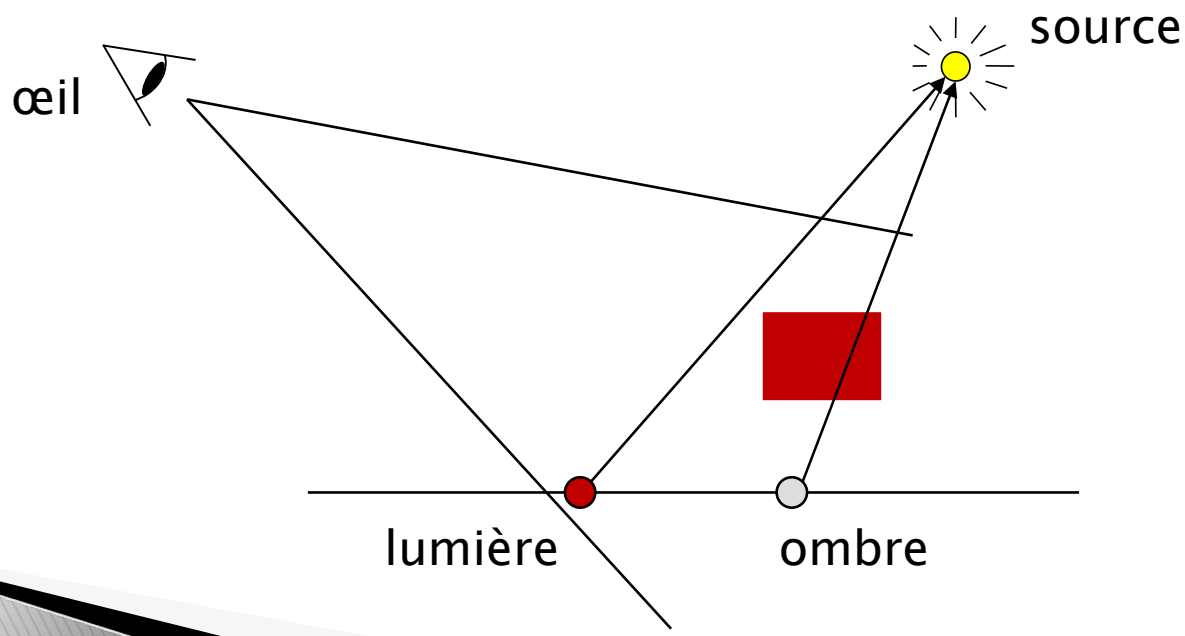
Moller & Haines "Real Time Rendering"

Méthodes actuelles

- Shadow Maps
  - Approche “image”
- Shadow Volumes
  - Approche “objet”

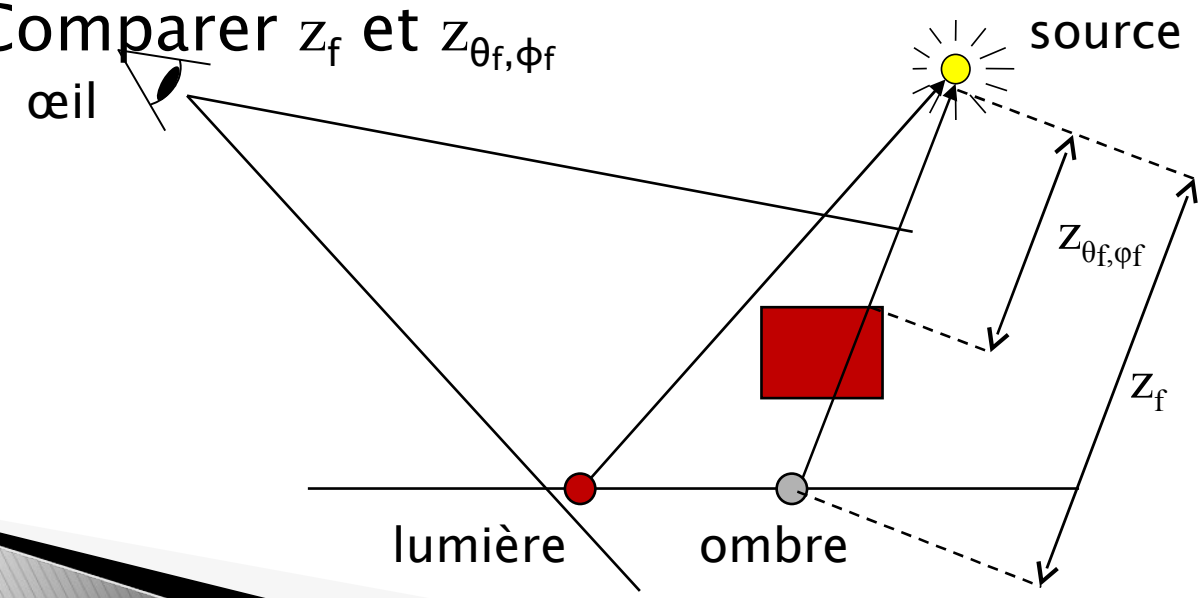
Shadow Maps

- Principe
  - Pour chaque fragment dessiné
    - regarder dans la direction de la source
    - tester si il y a un objet plus proche
      - oui : le rayon est bloqué → on est dans l'ombre
      - non: la source est visible → on est dans la lumière



Shadow Maps

- Comment?
  - Discrétiser les *shadow casters*
    - échantillonner les directions θ,ϕ depuis la lampe
    - stocker la distance $z_{\theta,\phi}$ du premier objet dans chaque direction
  - Calculer la direction θ_f,ϕ_f et la distance z_f pixel / source
  - Comparer z_f et z_{θ_f,ϕ_f}



Shadow Maps

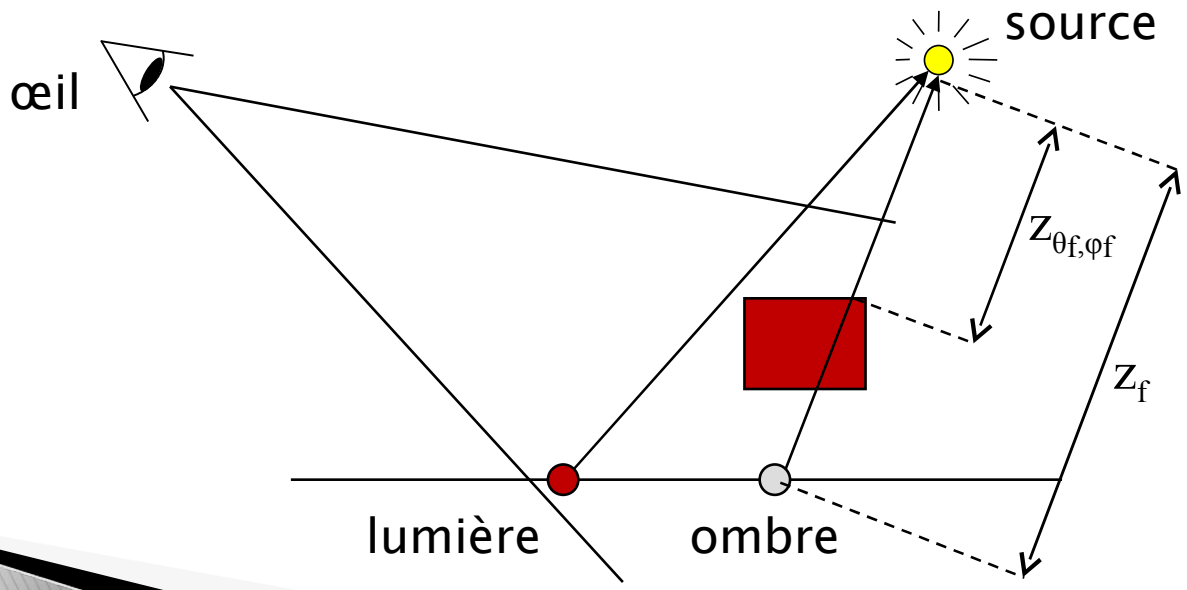
- En pratique

une fois par frame

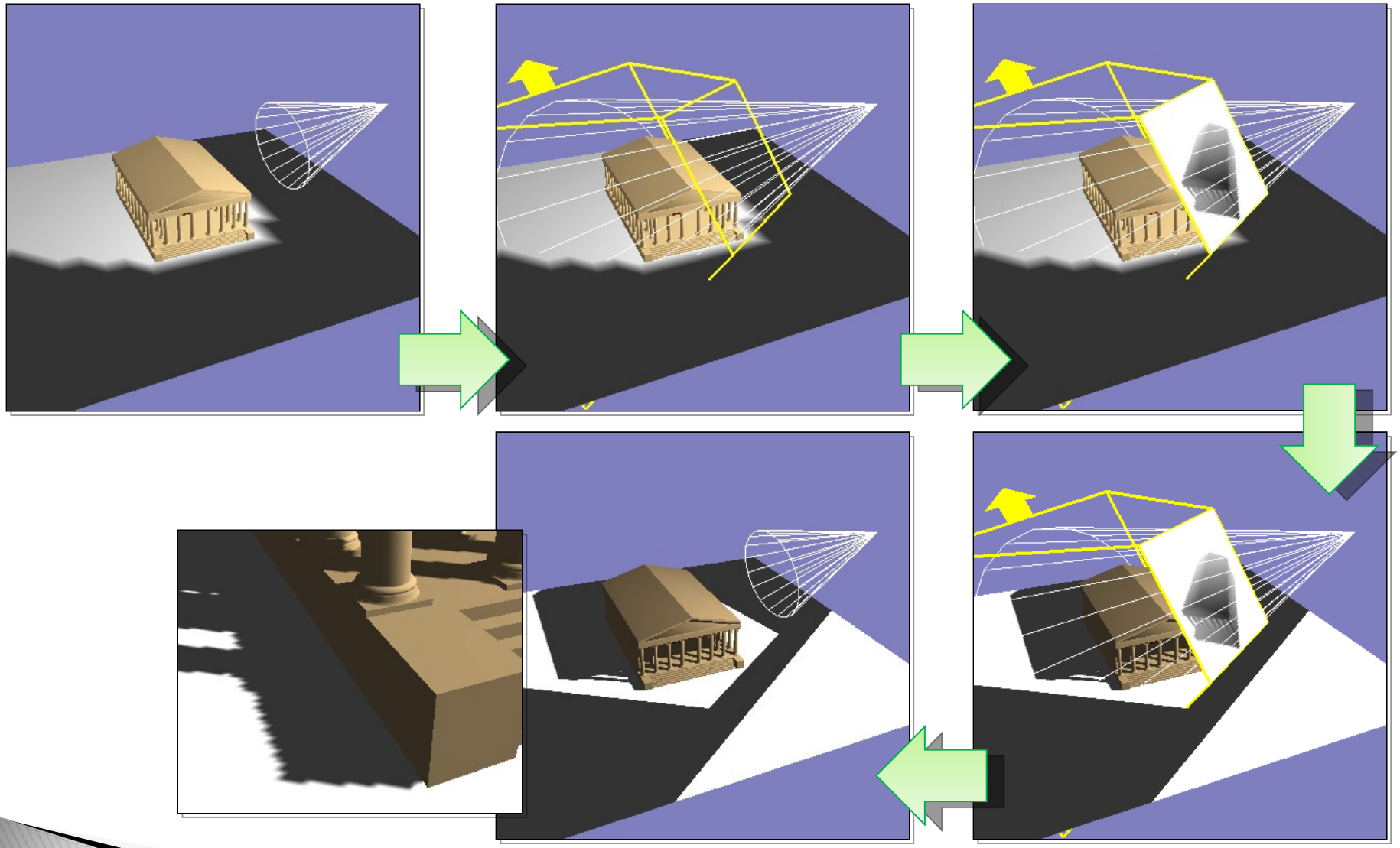
- Rendre une vue depuis la source
  - conserver le *z-buffer* dans une texture T
  - conserver la matrice M de projection

pour chaque pixel f

- Faire un *projective texture lookup* par M dans T
- Comparer z_f et $z_{\theta f,\phi f}$ *(Pixel Shader)*



Shadow map



Points techniques

- Rendu rapide de la shadow map
  - Seul le z nous intéresse
    - on désactive tout le reste

```
glDrawBuffer(GL_NONE);
glReadBuffer(GL_NONE);
```

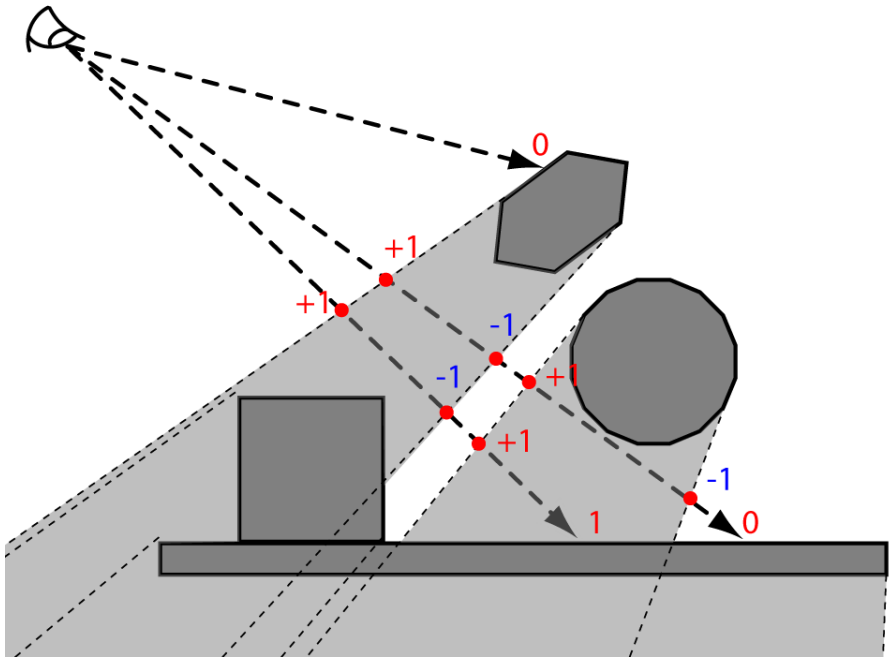
  - Mise à jour de la texture de profondeur que
    - quand la source de lumière bouge
    - quand les *shadow casters* bougent
- Choix des *shadow casters*
  - On ne doit s'intéresser qu'aux casters qui projètent une ombre visible par la caméra

Points techniques

- Combiner avec l'éclairage OpenGL
  - on obtient une texture de 1 (lumière) et 0 (ombre)
  - on plaque cette texture en modulant la couleur
    - approche naïve
      - rendre la scène avec l'éclairage OpenGL
      - problème du spéculaire dans l'ombre!
    - approche correcte
      - rendre la scène avec juste l'éclairage ambiant
      - rendre la scène avec le diffus modulé par l'ombre
    - aujourd'hui
      - tout combiner dans un *pixel shader*

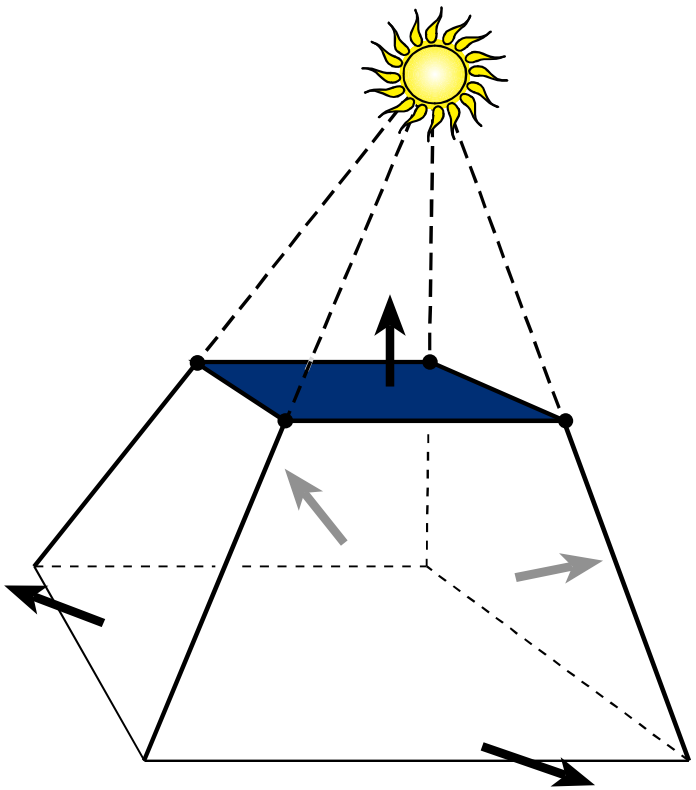
Shadow Volumes

- Principe
  - pour chaque *shadow casters*
    - construire un volume d'ombre
  - pour chaque fragment dessiné
    - compter combien de fois on entre/sort d'un volume
      - > 0 : dans l'ombre
      - $= 0$: dans la lumière



Shadow Volumes

- Comment?
  - construire les volumes d'ombres
    - trouver la silhouette des objets vus depuis la source
    - construire des *quads* infinis s'appuyant
      - sur la source
      - sur chaque arête de silhouette
  - compter les entrées/sorties
    - utiliser le *stencil buffer*



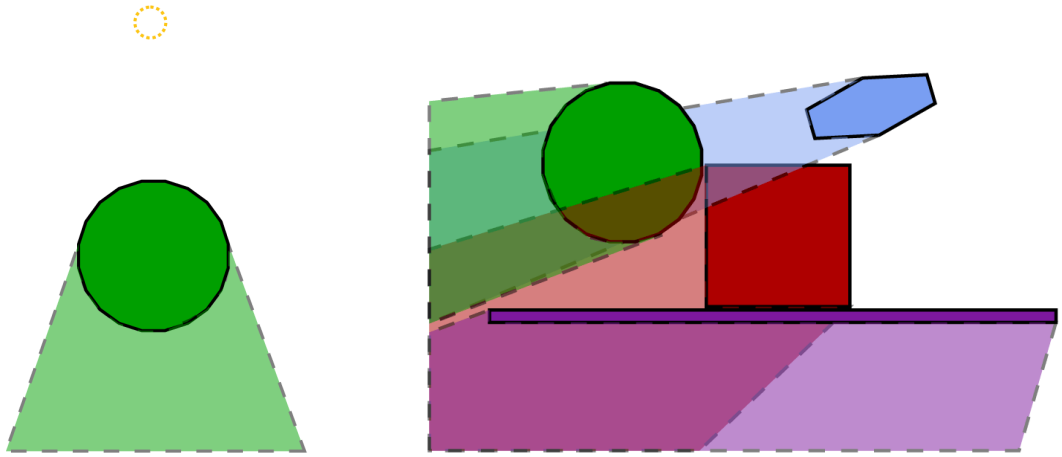
Trouver la silhouette

- Algorithme
  - pour chaque arête du modèle :
    - identifier les faces gauche/droite et leurs normales
    - calculer les prod. scal. normales/vecteur vers la source
    - marquer comme silhouette si de signe différents
  - fait sur le CPU (possible sur GPU)

- Requiert les infos d'adjacence du maillage
- Calcule un sur-ensemble de la silhouette

Construire les volumes d'ombres

- On extrude les arêtes de silhouette vers l'∞
- On obtient des *shadow quads*
  - ces *shadow quads* sont orientés
    - *front* ou *back facing*
  - ils forment des volumes d'ombres imbriqués
- Dans l'ombre = dans au moins un volume



Stencil Buffer

- Buffer auxilliaire non affiché
  - en plus couleur et profondeur
- Contrôle si un fragment “passe” ou pas
  - *stencil test* `glStencilFunc(GL_EQUAL,0,~0);`
- Modifié lors de la rasterisation
  - incrémentable/décrémentable
  - trois actions spécifiables : le fragment
    - rate le *stencil test*
    - passe le *stencil test* et passe/rate le *z-test*

```
glStencilOp(GL_KEEP,GL_KEEP,GL_DECR);
```

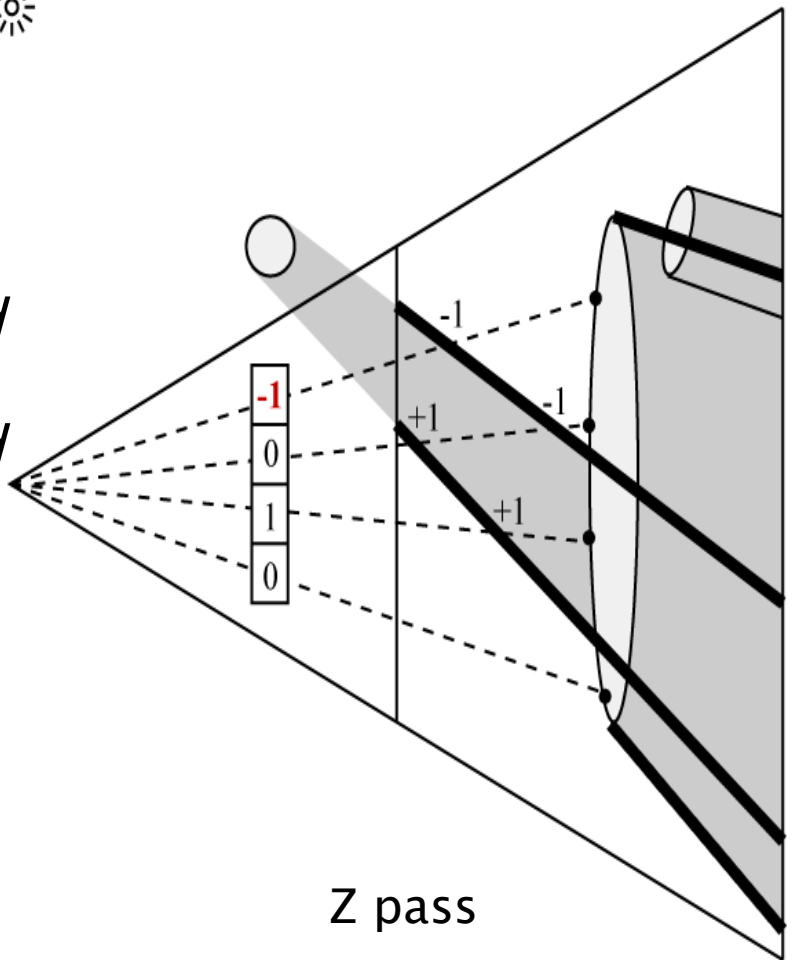
Stenciled Shadow Volumes

- Premier rendu de la scène
  - Initialise le *Z-buffer*
- Rendu du volume d'ombre
  - Pour chaque *front facing shad. quad*

    ```
    glStencilOp(GL_KEEP,GL_KEEP,GL_INCR);
    ```
  - Pour chaque *front facing shad. quad*

    ```
    glStencilOp(GL_KEEP,GL_KEEP,GL_DECR);
    ```
- Deuxième rendu de la scène
  - Pour la partie éclairée

    ```
    glStencilFunc(GL_EQUAL,0,~0);
    ```



Z pass

Bilan

- A vous !

Shadow map: bilan

- Avantages
  - Simple à implémenter
  - Fonctionne pour des scènes quelconques
  - Coût indépendant de la scène

- Inconvénients
  - Plusieurs (>= 2) rendus de la scène
  - Gestion des sources omni-directionnelles?
  - Problème d'aliassage

Shadow volumes: bilan

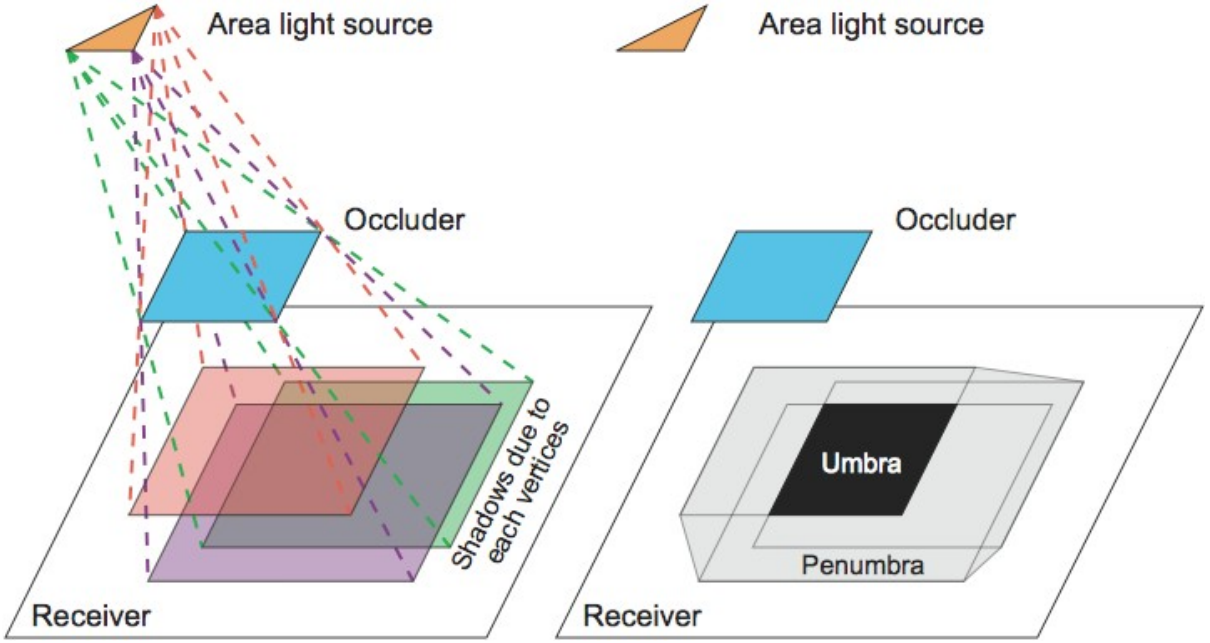
- Avantages :
  - ombres précises
  - positions quelconques source/caméra
  - robuste si bien programmé,
- Inconvénients :
  - calcul de la silhouette (sur CPU/GPU)
  - scènes bien modelisées préférables
  - rendu des volumes (+ caps)
  - *fill-rate* limité (mais solutions d'optimisation)

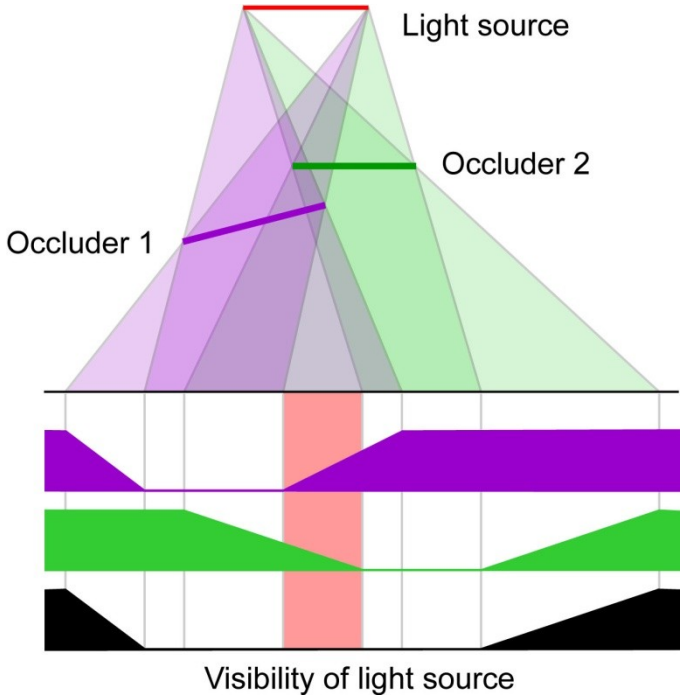
Calcul des ombres douces

Ombres douces

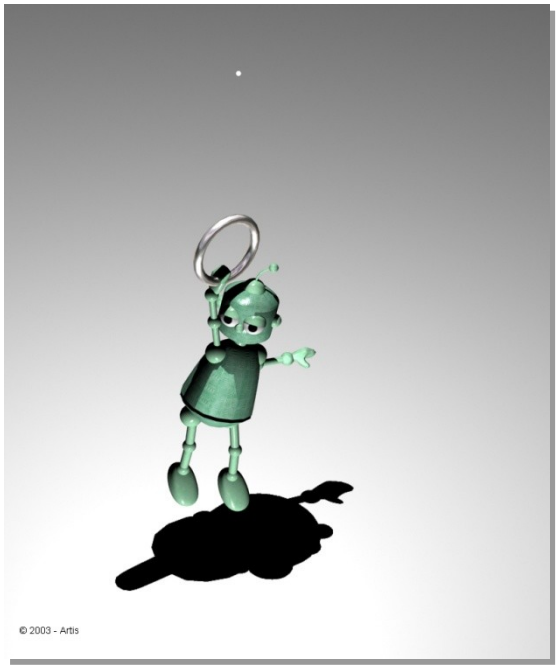
- Algorithmiquement plus compliqué
  - problème de visibilité point-surface
    - au lieu de point-point
    - silhouette ?
  - ombre de la somme ≠ somme des ombres
  - problème de la variation de normale
    - approximation par le % de visibilité
- Plusieurs algorithmes approximatifs
  - suffisants pour l'oeil humain
  - voir survey par Hasenfratz et al.

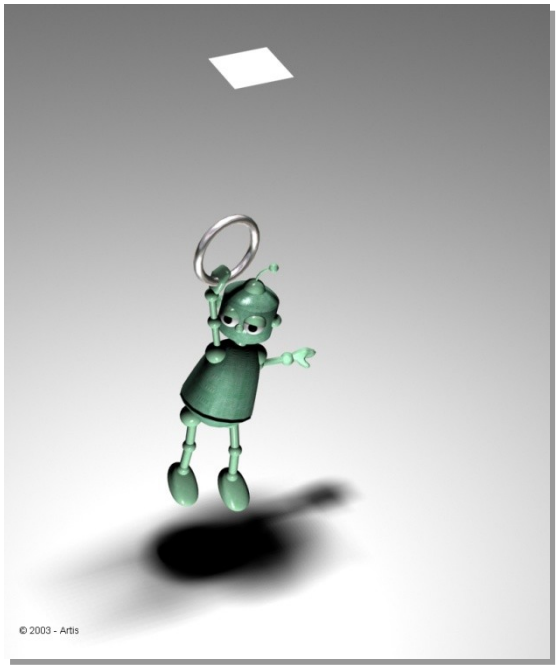
Ombres et pénombre

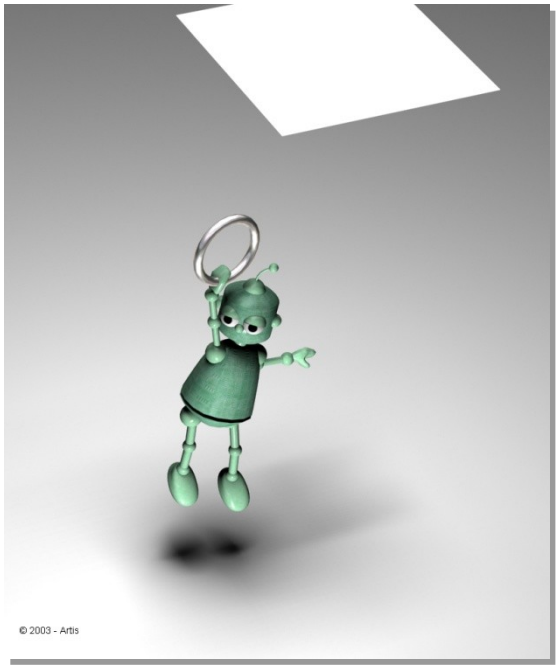




Problèmes de silhouette







Ombres douces par *sampling*

- Accumulation d'ombres :
  - calculer plusieurs ombres ponctuelles
  - additionner et moyenner les résultats
  - *accumulation buffer*
  - nombre d'échantillons élevés
    - temps de calcul multiplié par # échantillons

4 échantillons

1024 échantillons

Extension du shadow volume

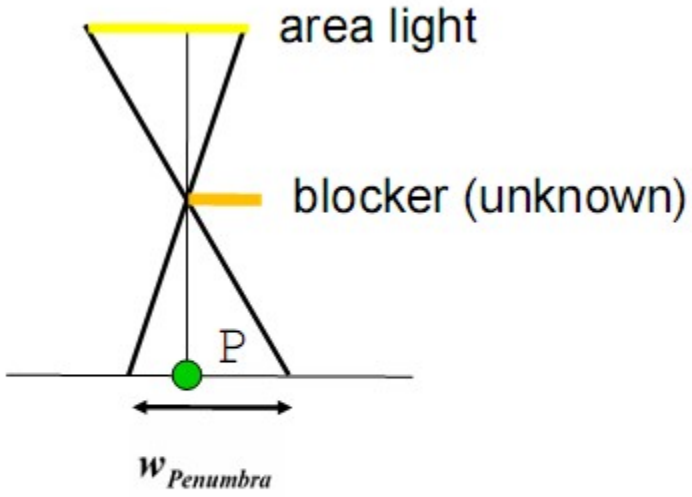
Penumbra wedges
U. Assarson, T. Möller

- Pour chaque arête de silhouette :
  - calculer volume englobant la pénombre
  - pour chaque pixel dans ce volume
    - calculer coefficient d'atténuation
- Beau, réaliste mais *fill-rate* multiplié par 2

Extension des shadow maps

Percentage-Closer Soft Shadows
Randima Fernando

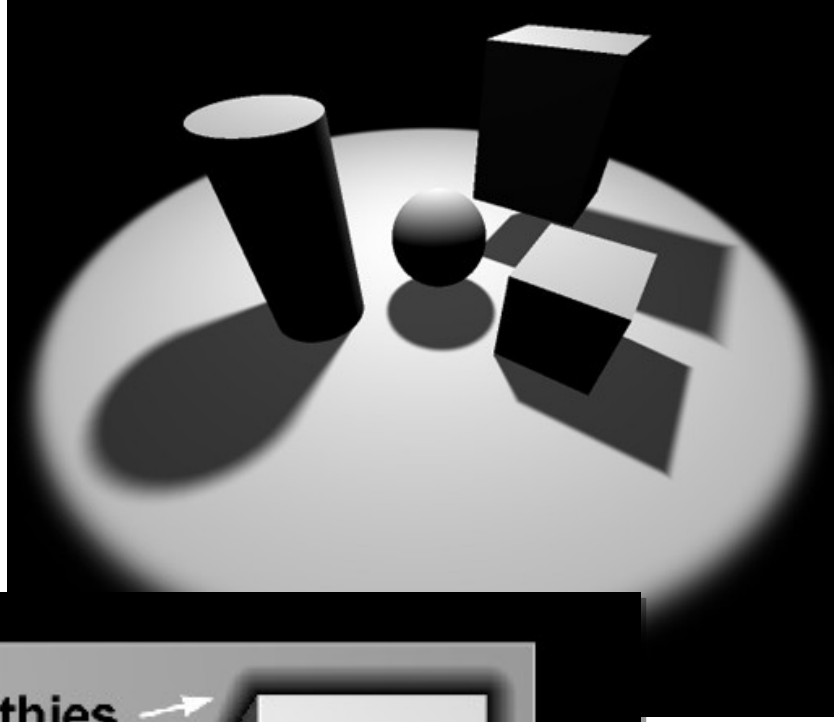
- Chercher les bloqueurs dans la shadow map
- En déduire la taille de la pénombre

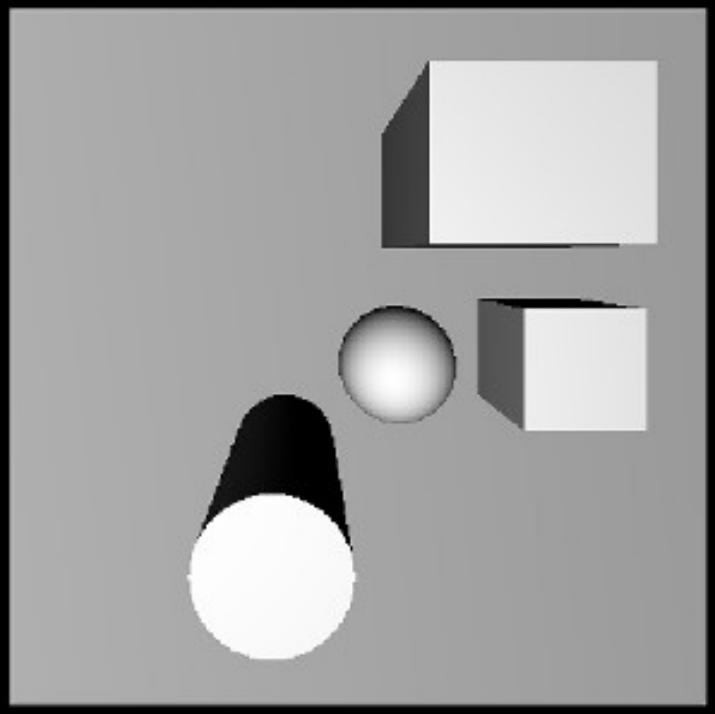


$$w_{Penumbra} = \frac{(d_{Receiver} - d_{Blocker}) \cdot w_{Light}}{d_{Blocker}}$$

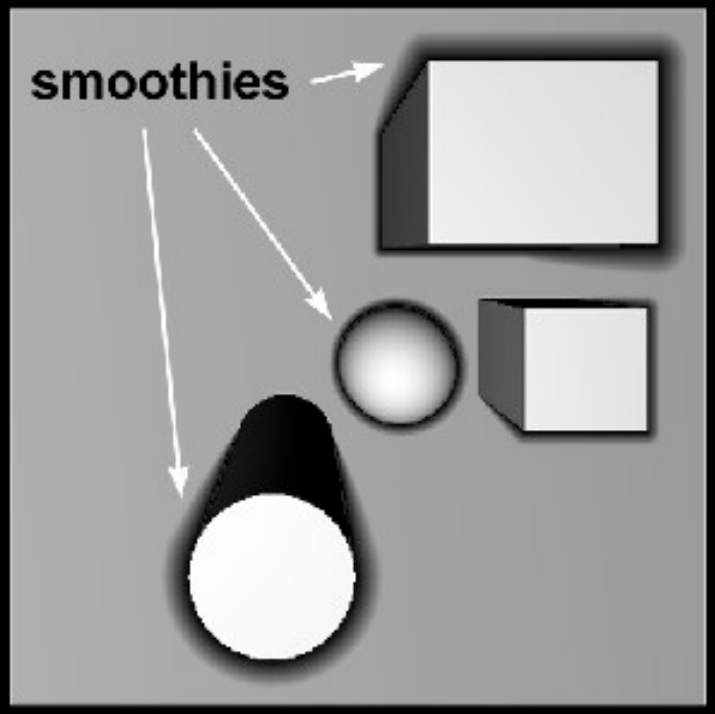
Mix objet / image

Rendering Fake Soft Shadows with Smoothies
E. Chan, F. Durand





light's view (blockers only)



light's view (blockers + smoothies)

Résumé & conclusion

- Shadow maps :
  - Stable, robuste, facile, rapide, aliassage
- Shadow volumes :
  - Beau, difficile, complexité algorithmique
- Ombres douces
  - Complexe, lent, beau

- Beaucoup de papiers récents