

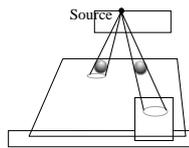
Calcul des ombres

Cours de DEA
C.Soler
Jeudi 01/02/2007

Introduction

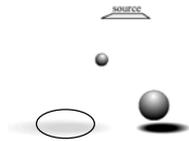
✓ Ombres dures (*Hard shadows*)

- Source ponctuelle
- Détermination binaire
- Renseignement sur la position des obstacles par rapport à l'oeil



✓ Ombres douces (*Soft shadows*)

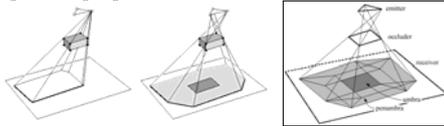
- Source étendue
- Variation continue
- Renseignement sur les tailles et positions relatives entre source, obstacle et récepteur



Introduction

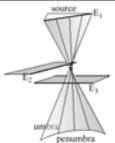
Ombre et pénombre

- La pénombre (sens large) est l'union des projections des obstacles depuis chaque point de la source
- L'ombre est l'intersection des projections des obstacles depuis chaque point de la source



- Pour une source et des obstacles polygonaux:

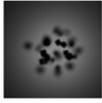
- » Limite ombre-pénombre: quadriques réglées
- » Limite pénombre: polyhèdre



Introduction

Approximation courante

- ✓ Séparation éclairage direct/visibilité moyenne



Illumination exacte

$$\int_{Source} I(\omega)v(x,\omega)\cos\theta d\omega$$

#



Illumination directe

$$\int_{Source} I(\omega)\cos\theta d\omega$$

x

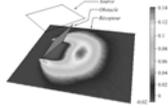


Visibilité moyenne

$$\frac{1}{\Omega_S} \int_{Source} v(x,\omega)d\omega$$

- ✓ Erreur d'approximation

- faible
- cantonnée dans les zones de forte corrélation source-obstacle

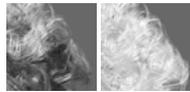


Introduction

Caracteristiques des algorithmes

- ✓ Type de
 - la source: ponctuelle / sphérique / polygonale / à l'infini / uniforme ...
 - l'obstacle: volumique / transparent / polygonal
 - récepteur: plan / courbe / volumique / complexe
- ✓ Exact / approximatif \Rightarrow différentes applications
- ✓ Auto-ombrage ?
- ✓ Vitesse:
 - non interactif / interactif / RT
- ✓ Implémentation: soft/hard
- ✓ Coût

E	Nombre moyen d'arêtes par polygone
P	Nombre de points pour échantillonner une source étendue
R	Resolution du buffer utilisé
n	Nombre de primitives dans la scene
pxq	Resolution de l'image en pixels.
- ✓ Espace:
 - espace objet / espace image



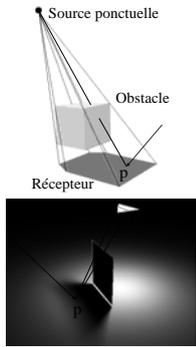
Introduction

Plan

- ✓ Espace objet
 - Lancer de rayons
 - BSP
 - Calculs géométriques explicites
- ✓ Espace image
- ✓ Méthodes hybrides

Lancer de rayons

- ✓ Pour tous les points visibles p :
 - Lancer un rayon de p vers la source
 - Calculer si p voit la source par intersection avec la géométrie
- ✓ Complexité: coûteux.
- ✓ Si la source est étendue:
 - Calculer la proportion de la source qui est visible (Approx.)
 - » Echantillonnage de la source
 - » Plusieurs rayons
 - » Calcul integral (numérique)
 - Calcul de l'illumination (Exact)



Lancer de rayons

Espace objet

Ray-tracing probabiliste (Cook'84)

- ✓ Principe: échantillonner la source aléatoirement (angle ou espace).
- ✓ Extensions: *motion-blur*, *depth-of-field*
- ✓ Inconvénients:
 - Très coûteux
 - Beaucoup de bruit (n rayons $\Leftrightarrow \log(n)$ bits)



Lancer de rayons

Espace objet

Qualité de l'échantillonnage



- ✓ Régulier:
 - corrélation avec les obstacles
 - \Leftrightarrow bandes d'éclairage
- ✓ Aléatoire uniforme:
 - bruit (Variance)
 - \Leftrightarrow nécessite beaucoup de points
- ✓ stratifié:
 - variance réduite \Leftrightarrow images meilleures

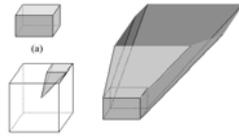
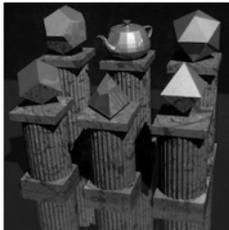


Lancer de rayons

Classification des rayons (Arvo'87)

✓ L'espace $\mathbb{R}^3 \times S^2$ est partagé en voxels.

- Dans chaque voxel on stocke l'ensemble des objets rencontrables par ordre de profondeur.



✓ Inconvénients:
- Coût mémoire

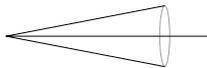
Lancer de rayons

Espace objet

Lancer de cônes (Amanatides'84)

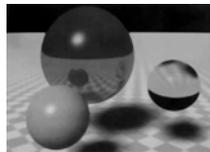
✓ Idée: remplacer les rayons d'ombre par des cônes

- *Anti-aliasing* automatique sur les ombres
- Peut simuler des réflexions mi-spéculaires



✓ Inconvénients:

- Les lampes sont sphériques
- Les intersections sont plus difficiles à calculer
- Cone-sphère, Cone-plan, Cone-polygone



Lancer de rayons

Espace objet

Single sample soft shadows (Parker'98)

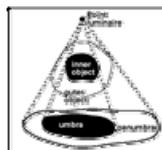
✓ Une ombre approximative est satisfaisante si:

- Le calcul est rapide
- La pénombre varie en fonction des distances obstacle/récepteur et la taille de la source
- La méthode est stable dans l'espace et le temps

✓ Par ailleurs:

- La forme de la source importe peu

→ Méthode *ad-hoc* pour une source sphérique

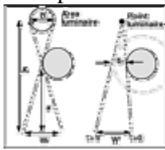


Lancer de rayons

Espace objet

Single sample soft shadows (Parker'98)

✓ Choix des paramètres



Source sphérique / obstacle plan:

$$s = (1 + \sin(\pi\tau - \pi/2))/2$$

approche par: $s = 3\tau^2 - 2\tau^3$

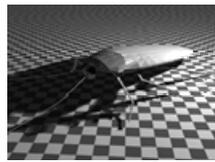
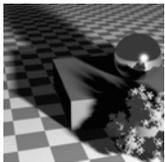
Composition des obstacles:

$$s = \min(s_1, s_2)$$

Création des obstacles virtuels:

$$b = aD/A$$

✓ Résultats



Lancer de rayons

Espace objet

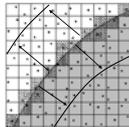
Cohérence des ombres dans l'espace image

Deux points voisins sont souvent sous la même ombre

- Conserver un pointeur sur obstacle, et l'utiliser en priorité lors du calcul des points voisins
- Peut s'appliquer à la liste des obstacles.

✓ Worley'97 - "fast soft shadows".

- Lancer un rayon d'ombre par pixel.
- Les pixels dont les voisins ne sont pas tous dans le même état sont sûrement dans la pénombre.
- Recalculer ces pixels avec plus de rayons
- Propager l'information



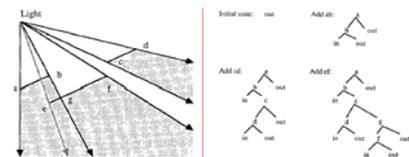
Lancer de rayons

Espace objet

BSP (Chin'89)

✓ Principe:

- Deux arbres BSP: un pour la scene (#1) et un pour la source (#2).
- Construire #2 en parcourant les polygones *front-to-back* (utiliser #1). Pour chaque polygone P:
 - » Découper P en fonction de #2
 - » Mettre à jour #2 avec les parties éclairées de P. Mettre à jour le noeud correspondant dans #1.



- Rendu: rendre ombrés ou pas les sous-polygones des noeuds de #1. (On peut faire un z-buffer, ou un rendu *back-to-front* sans z-buffer grâce à #1).

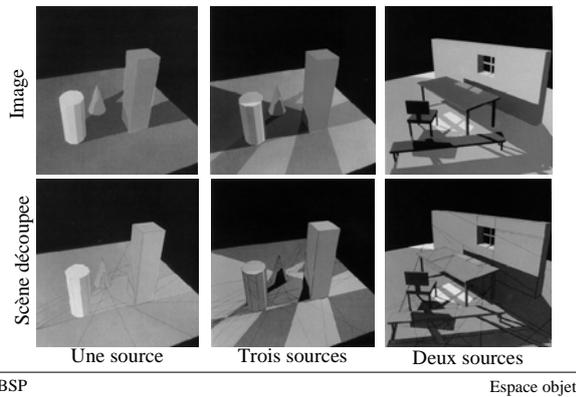
✓ Sources multiples:

- Utiliser chaque source successivement pour couper les polygones de #1

BSP

Espace objet

BSP (Chin'89)



Soft-Shadows with BSP (Chin'92)

✓ Deux BSP pour chaque source:

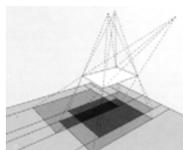
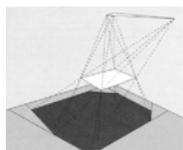
- un BSP pour la pénombre
 - » Plans VE et EV qui séparent la source de l'obstacle
 - » La pénombre est l'intersection demi-espaces correspondants.

(définition équivalente à union des projections de l'obstacle)

- Un BSP pour l'ombre
 - » Plans VE seulement, et tels que la source et l'obstacle sont entièrement du même côté.
 - » L'ombre est l'intersection des demi-espaces correspondants.

(définition équivalente à l'intersection des projections de l'obstacle)

✓ L'ombre est incluse dans la pénombre.



BSP

Espace objet

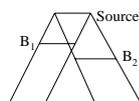
Soft-Shadows with BSP (Chin'92)

✓ Algorithme:

- Pre-processing:
 - » Calculer un BSP pour la scène
 - » Couper les sources par tous les plans des polygones
 - » Assure la cohérence de l'ordre de traversée des BSP des sources
 - » Couper tous les polygones par le plan de chaque source
- Classification et découpage des polygones, *front-to-back*
 - » Utiliser le BSP de pénombre et celui d'ombre
 - » Découper les polygones selon leur classification (E/P/O)
- Rendu:
 - » Calcul explicite de l'éclairage dans les zones de pénombre (coûteux)
 - » Calcul rapide dans les zones éclairées (facteur de forme analytique)

✓ Défauts:

- La composition des obstacles est fautive
 - » L'union de deux volumes d'ombre n'est pas le volume d'ombre de l'union
 - » D'ailleurs, pas de triples-arêtes (la limite d'ombre est une conique)

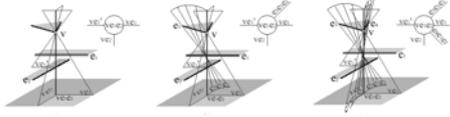


BSP

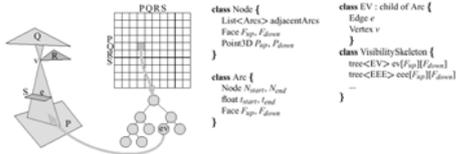
Espace objet

Le squelette de visibilité (Durand'97)

- ✓ Enumération des discontinuités \Rightarrow graphe



- ✓ Structure de données:



Approches analytiques

Espace objet

Le squelette de visibilité (Durand'97)

- ✓ Requêtes à coût constant:
 - Liste exacte des obstacles
 - Calcul de l'ombre sans séparation de la visibilité.
 - Maillage de discontinuité

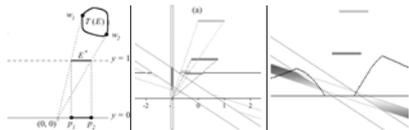


Approches analytiques

Espace objet

Formulation analytique (stark'99)

- ✓ Étude en 2D



- Sans occlusion:

$$I(x) = k \int_E \frac{\cos \theta \cos \theta^*}{d(x, u)} du = k \int_{p_1(x)}^{p_2(x)} \frac{du}{(1+u^2)^{3/2}}$$

- Cas général:

$$I(x) = k \int_{M_E(x,.)} w(u) du - k \int_{M_O \cap M_E(x,.)} w(u) du$$

\Rightarrow Somme de 2 splines polyédrales \Rightarrow expression analytique

Approches analytiques

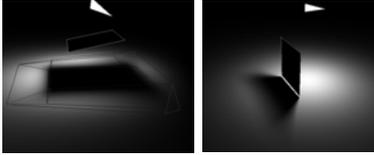
Espace objet

Extension en 3D (stark'99)

- ✓ En trois dimensions:
 - Splines polyédrales définies sur des trapèzes de dim 4
 - Fonction poids:

$$w(x, y) = (1 + x^2 + y^2)^{-2}$$

- Formule d'évaluation analytique
 - » Assez complexe, mais utilisable. Sans de-corrélation.
 - » Restreinte à certaines configurations



Approches analytiques

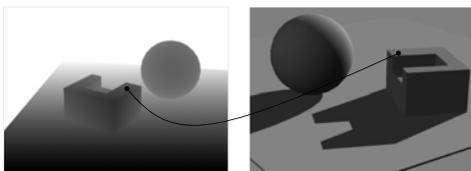
Espace objet

Plan

- ✓ Espace objet
- ✓ Espace image
 - Shadow maps
 - Shadow volumes
 - Méthodes à base de textures
- ✓ Méthodes hybrides

Shadow maps (William'78)

- ✓ Algorithme:
 - Rendre la scène vue depuis la source
 - Lire le z-buffer
 - Rendre la scène depuis l'oeil. Pour chaque pixel:
 - » Transformer le pixel dans l'image 1
 - » Si le z est plus petit, le pixel est éclairé, sinon il est à l'ombre.



Vue depuis la source
Z-buffer

Vue utilisateur

Shadow maps

Espace image

Shadow maps: Implémentation

```

LightViewTransform();
RenderScene();
glCopyTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT16, 0, 0, w, h, 0);

glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_COMPARE_MODE_ARB, GL_COMPARE_R_TO_TEXTURE);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_COMPARE_FUNC_ARB, GL_LEQUAL);
glTexParameterf(GL_TEXTURE_2D, GL_DEPTH_TEXTURE_MODE_ARB, GL_INTENSITY);

glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR); // faire de meme pour T,R et Q
glEnable(GL_TEXTURE_GEN_S); // #

glEnable(GL_TEXTURE_2D);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glGetFloatv(GL_PROJECTION_MATRIX, proj_mat_scene);
glMatrixMode(GL_TEXTURE);
glLoadIdentity();
glMultMatrix(proj_mat_scene);
glMultMatrix(modelview_mat_scene);

glMatrixMode(GL_MODELVIEW);
EyeViewTransform();

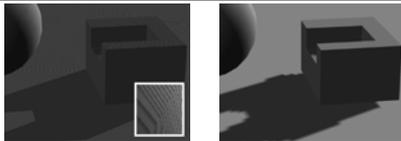
RenderScene();
    
```

(C.f exemples sur le site NVidia)

Shadow maps

Espace image

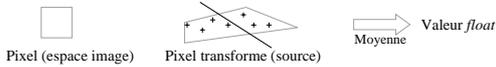
Shadow maps - défauts



Effet d'auto-ombrage

Aliasing

- ✓ Auto-ombrage (éclairage trop "étalé")
 - Correction: rajouter une constante (*bias*) à la *depth map*
- ✓ *Aliasing* (sous-échantillonnage de la *depth map*)
 - Correction: *Percentage-closer filtering* (Reeves'87)



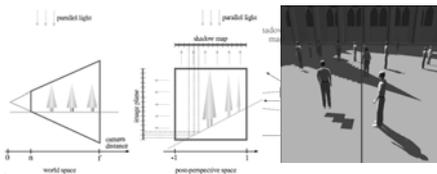
- ✓ *Aliasing* (sur-échantillonnage de la *depth map*)

Shadow maps

Espace image

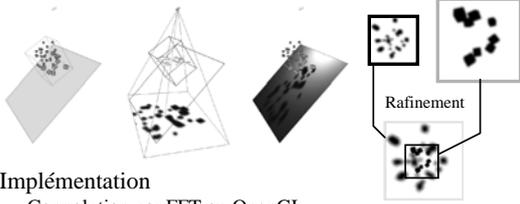
Améliorations

- ✓ Hourcade'85
 - utiliser un item buffer
- ✓ *Adaptive shadow maps* (Fernando'01)
 - la *shadow map* est stockée dans un *quadtree*
 - fabriquée en utilisant du mip-mapping
- ✓ *Perspective shadow maps* (Stamminger'02)



Méthode de convolution

- ✓ Cas général: approximation par des plans



- ✓ Implémentation
 - Convolution par FFT ou OpenGL



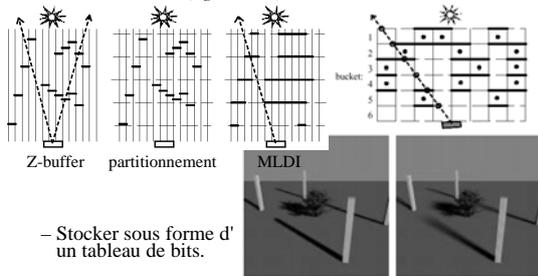
Espace image

Plan

- ✓ Espace objet
- ✓ Espace image
- ✓ Méthodes hybrides

Multi-layer depth images (Keating'99)

- ✓ Convertir la géométrie en un ensemble de couches
- ✓ Lancer des rayons vers la source à travers ces couches
 - Eviter les trous (*light leaks*)



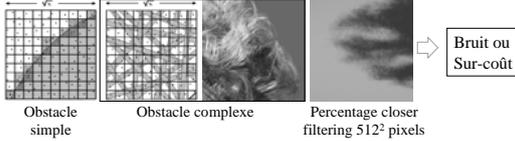
- Stocker sous forme d'un tableau de bits.

Méthodes hybrides

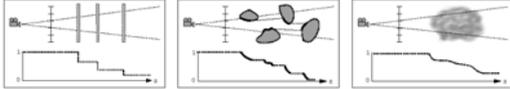
Méthodes hybrides

Deep shadow maps (Lokovic'00)

- ✓ Pour des obstacles complexes l'échantillonnage stochastique de *shadow map* a trop de variance.



- ✓ Utiliser une fonction de transmittance par pixel



- ✓ Sampling: anti-aliasing à z constant dans la DSM.

Méthodes hybrides

Méthodes hybrides

Deep shadow maps (Lokovic'00)

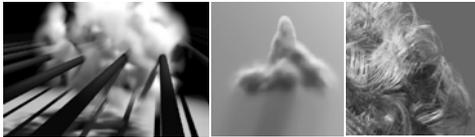
- ✓ Calcul
 - Visibilité volumique

$$\frac{dI(s)}{ds} = -k(s)I(s) \quad \tau(z) = e^{-\int_0^z k(t)dt}$$

- Surfaces semi-transparentes
 - » Fonctions en escalier

- ✓ Stockage:

- compression



Méthodes hybrides

Méthodes hybrides

Références

Ray tracing

Distributed ray tracing. R.Cook, T. Porter, L.Carpenter. Siggraph'84
Ray tracing with cones. J.Amanatides. Siggraph'84
Fast ray-tracing by ray classification. J.Arvo, D.Kirk. Siggraph'87

Shadow maps - espace objet

Casting curved shadows on curved surfaces. William. Siggraph'78
Rendering anti-aliased shadows with depth maps. W.Reeves et al. Siggraph'87.
Fast shadows and lighting effects using texture mapping. M.Segal et al. Siggraph'92
Rendering CSG models with a z-buffer. D.Salesin, J.Stolfi. Siggraph'90.

Arbres BSP

Near real time shadow generation using BSP trees. Norman Chin, Steven Feiner. Siggraph'89
Fast Object-Precision Shadow Generation for Area Light Sources Using BSP Trees. N.Chin, S. Feiner. Siggraph'92.

Techniques analytiques

Computing exact shadow irradiance using splines. M.Stark et al. Siggraph'99
The visibility skeleton: a powerful and efficient multi-purpose global visibility tool. CG, 31(3A)p89,1997

Techniques diverses

Single sample soft shadows. S.Parker, P.Shirley, B.Smith.
Efficient generation of soft-shadow textures. M.Herf. Technical report CMU-CS-97-138. Carnegie Mellon University, Pittsburgh.
Fast computation of soft shadow textures using convolution. C.Soler, F.Sillion. Siggraph'98

Surveys

A survey of shadow algorithms. A.Woo, P.Poulin, A.Fournier. IEEE CG&A, 10(6):13-32, Nov 1990.
A multidisciplinary survey on visibility. F.Durand. ACM course notes *Visibility, problems and applications.* Siggraph'2000.

Image-based

Efficient image-based methods for rendering soft shadows. M.Agrawala et al. Siggraph'2000
Deep shadow maps. T. Lokovic, E.Veach. Siggraph'2000.
