

Algorithmes pour l'image de synthèse

Responsable : Gilles Debunne, CNRS.

Conférenciers : Xavier Décoret, Cyril Soler

Sommaire

Notes de Cours	27
Algorithmes pour l'image de synthèse	27
1 Introduction	27
1.1 Présentation du domaine	27
1.2 Objectifs du cours	28
2 Résumés des cours	29
2.1 Les coordonnées homogènes	29
2.2 Création d'une image de synthèse	29
2.3 Méthode de radiosit�	29
2.4 Les quaternions	30
2.5 Algorithmes de simplification de maillages	30
3 R�f�rences	31

<i>Quadratic simplification</i> en trois mots (Xavier Décoret)	32
1 Présentation	32
2 Intérêt des quadriques d'erreurs	33
2.1 Compacité mémoire	33
2.2 Optimisation	34
3 Algorithme	34
4 Conclusion	35
Références	35
Transparents	36
Les coordonnées homogènes	38
Création d'une image de synthèse	43
Calcul d'éclairage global - Méthode de radiosité	48
Les quaternions	62
Simplification de maillage	70
Bibliographie	87

Algorithmes pour l'image de synthèse

Gilles Debunne, Xavier Décoret, Cyril Soler⁴

Projet ARTIS⁵

1 Introduction

L'image de synthèse est un domaine très jeune (une trentaine d'années tout au plus) et qui vit en ce moment une véritable révolution technologique et scientifique. Cette discipline combine une forte composante mathématique (en particulier géométrique) et algorithmique (liée à la taille des données traitées).

Ce cours se propose de présenter quelques uns des formalismes mathématiques spécifiques au domaine de l'informatique graphique, ainsi que certaines méthodes de résolution originales. Même si les thèmes abordés sont éloignés de ceux des participants, les mathématiques (principalement basées sur l'algèbre linéaire) et les algorithmes (dédiés aux données volumineuses et souvent hiérarchiques) qui sont utilisés doivent pouvoir être rapprochés de ceux couramment utilisés par tout scientifique. Les auteurs espèrent ainsi que cette présentation ouverte pourra donner lieu à des recoupements, à des généralisations, voire à de nouveaux thèmes d'application ou de nouvelles pistes.

1.1 Présentation du domaine

Avec l'évolution de la taille des disques durs, les cartes graphiques sont le seul domaine de l'informatique à *dépasser* les prévisions de la loi de Moore, qui prévoit le doublement de la puissance tous les 18 mois. Ce décuplement de puissance se répercute très rapidement dans les machines grand public, tiré par le développement de l'industrie du jeu vidéo, qui génère désormais un chiffre d'affaire supérieur à celui de l'industrie cinématographique.

Il en résulte un grand chamboulement de la discipline, une effervescence liée aux nouvelles possibilités offertes, et à celles que la recherche peut proposer, qui peuvent être intégrées à très court terme dans la prochaine génération de cartes. Cette évolution ne saurait rendre caduque les problématiques liées à la complexité des scènes, bien au contraire, la demande pour des univers plus riches et plus beaux augmentant sans doute plus rapidement que les performances brutes des cartes.

⁴Adresses électroniques : `prenom.nom@imag.fr`

⁵`artis.imag.fr`

Cette évolution va également très prochainement déborder sur d'autres domaines. Une carte graphique peut en effet être considérée comme un co-processeur mathématique parallèle très performant (supérieur à tout Pentium) dédié au calcul matriciel $4*4$, qui est plus est de bas prix grâce aux économies d'échelle. Son utilisation dans la résolution de problèmes mathématiques a d'ores et déjà été présentée. De même les interfaces graphiques pourront-elles tirer parti d'un affichage en 3D pour proposer de nouvelles méthodes d'interaction.

1.2 Objectifs du cours

L'introduction faite dans ce cours devrait familiariser le lecteur avec le domaine de la synthèse d'images. Celui-ci est devenu très abordable et nous espérons vous convaincre de franchir le pas. La visualisation tri-dimensionnelle peut en effet être un outil puissant de compréhension et d'analyse de données complexes, éventuellement grâce à leur animation, et permet également d'observer de façon plus intuitive les résultats d'un algorithme. Les bases données dans ce cours ainsi que les nombreux exemples de code à copier-coller référencés devraient vous permettre de rapidement utiliser cet outil.

L'informatique graphique est également un excellent domaine d'application pour de très nombreuses recherches algorithmiques ou mathématiques. La liste en est très longue, mais on peut citer la résolution de gros systèmes linéaires, la compression de données, le traitement du signal que représentent une image ou un modèle 3D, les algorithmes probabilistes, l'intégration numérique, la résolution d'équations aux dérivées partielles, et bien d'autres encore.

Comme dit en introduction, les auteurs espèrent créer des rapprochements entre les diverses disciplines des participants et les notions présentées dans ce cours. cette ouverture d'esprit à d'autres disciplines ne peut en tout cas qu'être bénéfique et nous l'espérons agréable voire fructueuse.

Les notions présentées dans ce cours sont issues d'une sélection arbitraire, où l'on a cherché à faire une introduction, à couvrir des thèmes bien différents en espérant toucher un large public, à mélanger les aspects mathématiques et algorithmiques, tout en restant simple et facilement accessible. Ceci explique l'hétérogénéité de ce cours, ainsi que la simplification parfois effectuée, au détriment de la complétude de l'explication. Le lecteur intéressé est référé à la bibliographie ou à une discussion plus approfondie avec les intervenants.

2 Résumés des cours

Nous ne résumerons ici que très succinctement les cours, afin de les rendre plus compréhensibles au lecteur qui ne pourra y participer. Les transparents se veulent

suffisamment détaillés pour en permettre une lecture autonome.

2.1 Les coordonnées homogènes

Ce formalisme mathématique consiste à plonger l'espace tri-dimensionnel couramment utilisé en image de synthèse dans un espace de dimension 4. Le bénéfice principal est une uniformisation sous forme de produit matrice-vecteur de toutes les opérations classiques appliquées en 3D : rotation, changement d'échelle et surtout translation, ce qui aurait été impossible sans les coordonnées homogènes. On y gagne de plus une représentation élégante de la notion de point à l'infini qui s'intègre parfaitement avec le cas général dans les applications de géométrie projective. Ce formalisme est ainsi adopté dans toutes les disciplines utilisant des coordonnées 3D : imagerie, vision et robotique.

2.2 Création d'une image de synthèse

Nous décrivons ensuite les bases de la création d'une image, à savoir la projection sur un plan image d'une géométrie 3D. Grâce aux coordonnées homogènes, cette projection peut elle aussi être exprimée par une matrice 4×4 , que l'on applique aux points 3D pour en obtenir la projection. Il faudrait pour compléter cette présentation parler du calcul de la couleur des points projetés, en introduisant les notions de *modèles de matériaux*, qui modélisent la relation entre les couleurs (au pluriel) de la surface, les couleurs des lampes, leurs positions respectives et la normale à la surface.

On donne ensuite quelques éléments sur l'organisation logicielle et matérielle liée à l'image de synthèse, ainsi que sur son évolution. Celle-ci est devenue beaucoup plus modulaire et les cartes graphiques sont désormais programmables.

2.3 Méthode de radiosité

Le but est ici de calculer l'éclairage reçu par les objets d'une scène, en présence de sources lumineuse. On montre qu'après simplification des hypothèses et discrétisation en éléments finis, le problème peut se ramener à la résolution d'un système matriciel. Chaque élément du vecteur solution correspond à la radiosité sur un petit élément de surface, l'ensemble de ces éléments formant les surfaces des objets de la scène. Les méthodes classiques se révèlent inefficaces, car trop sensibles aux erreurs numériques ou trop lentes.

La méthode employée consiste alors en une résolution *itérative* du problème, par applications successives de mêmes opérations matricielles. On montre la validité mathématique de cette méthode et en présente une interprétation physique. Chaque itération revient à transférer de l'énergie lumineuse entre l'ensemble des

éléments de la scène. L'une des méthodes de résolution utilisées (le *shooting*) revient à propager de l'énergie depuis les sources sur les éléments, qui eux-mêmes la renvoie aux autres éléments, et ainsi de suite. L'autre méthode (le *gather* est duale : chaque élément somme l'ensemble de l'énergie reçue après un nombre donné de réflexions.

Bien que convergent assez rapidement, cet algorithme est trop lent pour des scènes complexes et nous présentons sa version hiérarchique, qui a démocratisé son utilisation (même si celle-ci reste limitée dans la pratique, le *lancer de rayon* étant généralement préféré).

2.4 Les quaternions

Les quaternions sont LA représentation adaptée des rotations dans \mathcal{R}^3 . Après avoir présenté plusieurs alternatives, toutes ayant leurs défauts, nous montrons que les quaternions, qui ne sont rien d'autre que des points sur la sphère unité de \mathcal{R}^4 , s'adaptent parfaitement à la représentation des rotations. L'algèbre des quaternions est très propre et permet en particulier de très naturellement de composer deux rotations, d'appliquer une rotation à un point ou encore et surtout d'interpoler entre deux rotations.

2.5 Algorithmes de simplification de maillages

Les maillages 3D utilisés pour représenter des objets s'avèrent inutilement complexes lorsque l'objet est vu de loin et donc affiché en petit. Le niveau de détail permet alors de générer plusieurs représentations d'un même objet, chacune étant adaptée à un intervalle de taille d'affichage. Nous présentons ici les différentes classes d'algorithmes qui permettent de décimer un maillage 3D, en conservant ou non certaines propriétés utiles (topologie, silhouette, apparence générale...). Nous présentons enfin dans le détail l'algorithme de Garland, très utilisé en pratique et (car) disponible gratuitement. Son formalisme mathématique simple, efficace et astucieux ainsi que ses très bons résultats pratiques (sur de gros modèles) en font une méthode de choix. Il est détaillé dans la suite de ces notes de cours.

3 Références

La référence reconnue du domaine est la conférence *Siggraph*, dont les actes sont maintenant publiés dans un numéro spécifique de la revue TOG (*Transactions on Graphics*), publiée par l'ACM. Nous reportons donc dans un premier temps le lecteur intéressé par les développements majeurs de la discipline aux articles de cette revue.

Le livre de référence en image de synthèse est le *Computer Graphics* de Foley, van Dam, Feiner et Hughes, publié chez Addison-Wesley. Une nouvelle édition remise à jour est prévue pour 2004.

Cette référence ainsi que celles complétant le cours sont regroupées dans les dernières diapositives du cours. Les auteurs sont bien-sûr également joignables par courriel pour des questions plus précises.

Quadric Simplification en trois mots

Xavier Décoret

1 Présentation

Quadric Simplification est un algorithme de *simplification de maillage* mise au point par Michael Garland et Paul Heckbert en 1997 [GH97]. Cette approche utilise un opérateur local d'*effondrement de paires de points* et une métrique d'erreur *sommet-plans*. Pour une présentation des différents opérateurs de simplifications de maillage et des métriques d'erreurs, on consultera le livre de Luebke et al., chap. 2 et 3 [LRC⁺02]. Un maillage 3D est simplifié en remplaçant deux points par un seul, comme illustré sur la figure 3. Une fois cette opération effectuée, on supprime les faces dégénérées et on met à jour les relations d'adjacence.

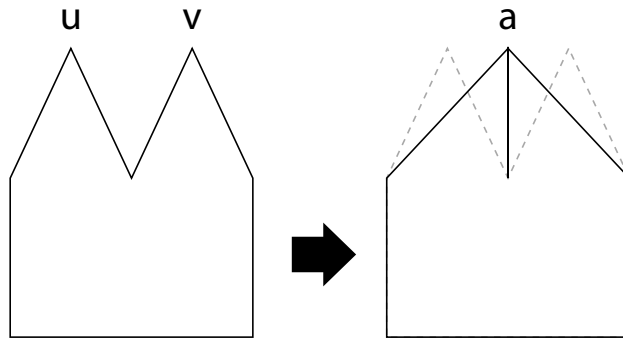


FIG. 3 – Simplification par effondrement de paires de points

L'erreur commise est définie sur les sommets du maillage initial, en mesurant de combien ils s'éloignent des plans supports des faces auxquels ils appartiennent. Un sommet u qui, après une séquence d'effondrements le concernant, se retrouve au point a , entraîne donc une erreur :

$$e(u \rightarrow a) \equiv \sum_{F \in [[u]]} d^2(a, \mathcal{P}(F)) \quad (12)$$

où les F sont les faces adjacentes à u . La figure 4 illustre cette définition. Le carré de la distance d'un point $(x, y, z)^T$ à un plan est une forme quadratique en x, y, z et peut s'écrire, en coordonnées homogènes :

$$d^2(a, \mathcal{P}(F)) = a^T Q_F a \quad (13)$$

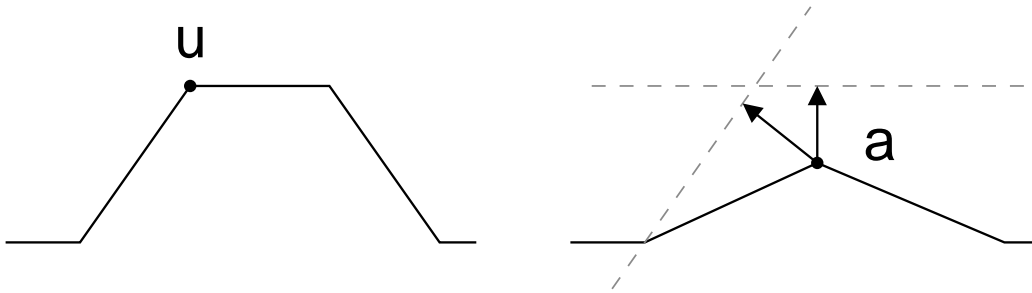


FIG. 4 – Erreur pour un sommet. Si u se retrouve en a , l’erreur est la somme du carré des distances aux plans en pointillés.

avec $a = (x, y, z, 1)^T$ et où Q_F est une matrice 4×4^6 . En injectant dans (12) et par linéarité, on obtient :

$$e(u \rightarrow a) = a^T Q_u a \quad (14)$$

où $Q_u = \sum_{F \in [[u]]} Q_F$ est dite *quadrique d’erreur* de u . On a trivialement $u^T Q_u u = 0$ car u appartient à toutes les faces adjacentes !

2 Intérêt des quadriques d’erreurs

2.1 Compacité mémoire

Supposons que l’on effectue les effondrements $u, v \mapsto a$ puis $a, w \mapsto b$ comme indiqué sur la figure 5. Pour mesurer l’erreur commise lors de la dernière opération,

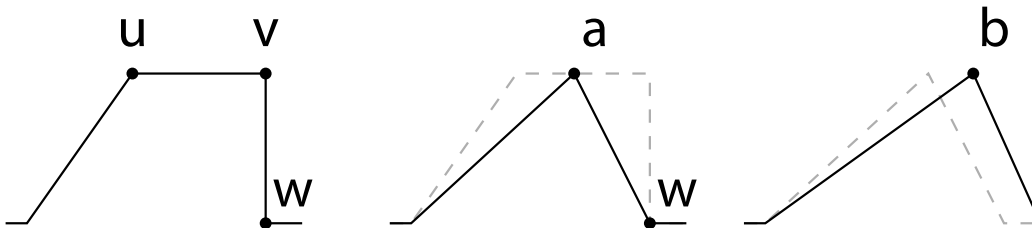


FIG. 5 – Succession d’effondrements

il faut se rappeler que a a été obtenu auparavant par effondrement de u et v . Maintenir cette information pour tous les points introduits lors de la simplification est

⁶Si le plan est représenté par $p = [abcd]^T$ alors $Q = pp^T$

prohibitif et inefficace. Heureusement, c'est inutile ! En effet :

$$\begin{aligned}
 e(u \rightarrow b) + e(v \rightarrow b) &= b^T Q_u b + b^T Q_v b \\
 &= b^T (Q_u + Q_v) b \\
 &= e(a \rightarrow b) \\
 &\quad \text{avec } Q_a \equiv Q_u + Q_v
 \end{aligned}$$

Lorsque l'on crée un nouveau point a , il suffit donc de lui associer comme quadrique d'erreur la somme des quadriques des deux points qu'il remplace. Cette quadrique garde implicitement trace des plans auxquels appartenaient tous les sommets qui se "retrouvent" dans a !

2.2 Optimisation

Lorsque l'on décide d'effondrer deux sommets u et v , il faut choisir par quel point les remplacer. Idéalement, on voudrait prendre celui qui minimise l'erreur commise. Garland et Heckbert ont montré que ce point est donné par :

$$a = \frac{1}{2} \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} \quad (15)$$

lorsque la matrice est inversible. Dans le cas contraire, ils proposent de calculer l'erreur en u , en v et au milieu de $[u, v]$ et de choisir celui qui minimise l'erreur.

3 Algorithme

L'algorithme complet est le suivant. On initialise les quadriques de chaque sommets du modèle initial. On considère toutes les paires de sommets adjacents (reliés par une arête) ou proches géométriquement⁷ Pour chacune d'entre elles, on calcule le meilleur point où les effondrer et l'erreur (minimisée) correspondante. On insère les paires dans une liste de priorité classée par erreur croissante. On effondre la paire sur le dessus de la liste. On met à jour les nouvelles paires et on réitère jusqu'à obtenir un nombre de faces fixé, où une erreur maximale autorisée.

⁷C'est la partie "sensible" de l'algorithme. Considérer toutes les paires entraîne une complexité quadratique. Il faut un peu de doigté pour choisir quelles paires considérer.

4 Conclusion

Cet algorithme est très efficace en pratique. Il a été étendu [GH98] pour tenir compte des attributs spécifiés à la surface du maillage (normale, couleur, coordonnées textures...). Une implémentation est disponible sur le web (<http://graphics.cs.uiuc.edu/~garland/software/qslim.html>).

Références

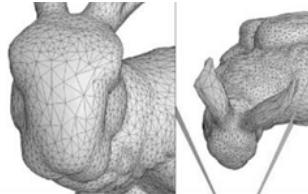
- [GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. pages 209–216, 1997.
- [GH98] Michael Garland and Paul S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. pages 263–269. IEEE-CS Press, 1998.
- [LRC⁺02] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann, first edition, july 2002.

Algorithmes pour l'image de synthèse

Gilles Debunne

Xavier Décoret

Cyril Soler



Introduction

- Domaine de l'image de synthèse très actif
- Combine mathématiques et algorithmique

- Présentation d'outils et de formalismes
 - Coordonnées homogènes
 - Quaternions
- Détail de deux algorithmes
 - Radiosité
 - Simplification de maillage

Plan du cours

- Coordonnées homogènes
 - Création d'une image de synthèse
 - Calculs de radiosité
- Pause
- Quaternions pour les rotations
 - Algorithmes de simplification de maillages

Les coordonnées homogènes

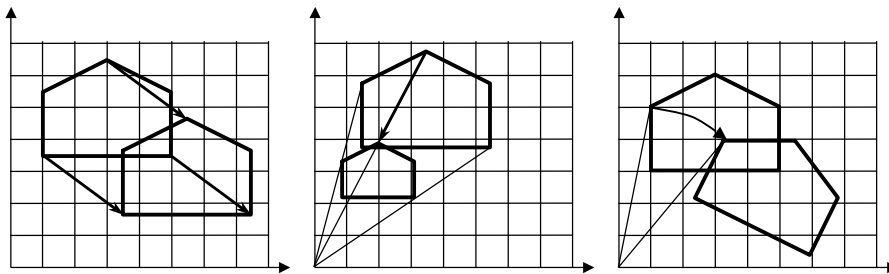
Un formalisme unifié et
puissant pour représenter les
transformations géométriques

Transformations géométriques

- On manipule des points en 3D
- On doit leur faire subir des transformations
Translation, rotation, changement d'échelle
→ Trouver un formalisme unifié
- On représente les points par des vecteurs

$$P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Représentation matricielle



$$P' = P + T$$

$$T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

$$P' = S \cdot P$$

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

$$P' = R \cdot P$$

$$R = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(selon axe Z)₆

Algorithmique pour l'image de synthèse

Coordonnées homogènes

■ Représentation non unifiée

Addition vectorielle ou multiplication matricielle ?

■ Solution : plonger dans \mathbb{R}^4

$$P = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \begin{pmatrix} xw \\ yw \\ zw \\ w \end{pmatrix} \approx \begin{pmatrix} x/w \\ y/w \\ z/w \end{pmatrix}$$

■ Bonus

$w=0$ permet de représenter un point situé à l'infini

Algorithmique pour l'image de synthèse

7

Modifications

- Pour les rotations et les chgt d'échelle

Il suffit de compléter la matrice

$$R = \begin{pmatrix} \boxed{R} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad S = \begin{pmatrix} \boxed{S} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P' = M \cdot P = \begin{pmatrix} \boxed{M} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} M \times P \\ w \end{pmatrix} \approx M \cdot P$$

Algorithmique pour l'image de synthèse

8

Translation par multiplication

- On définit ainsi la matrice de translation

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

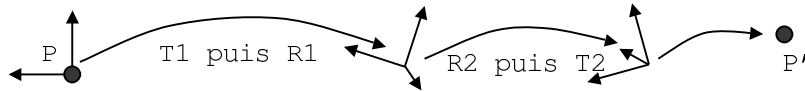
$$P' = T \cdot P = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} x + w t_x \\ y + w t_y \\ z + w t_z \\ w \end{pmatrix} = P + T$$

Algorithmique pour l'image de synthèse

9

Composition de transformations

- Il suffit de multiplier les matrices associées

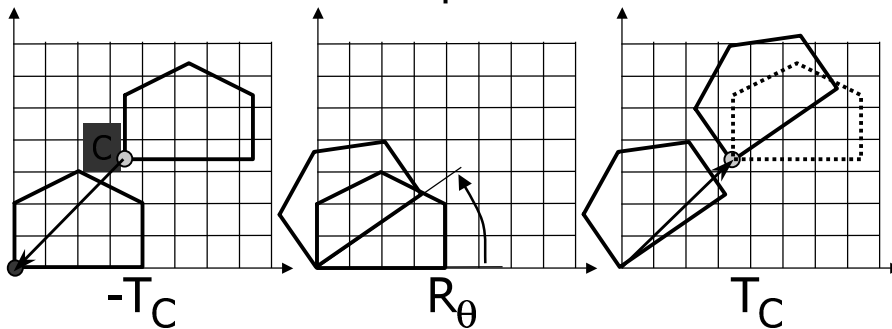


$$P' = T_2 \cdot R_2 \cdot R_1 \cdot T_1 \cdot P = M \cdot P$$

- L'ordre des multiplications est important
- Toute transformation peut ainsi être représentée par une matrice 4x4

Exemple : rotation excentrée

- Rotation autour d'un point C arbitraire



$$R_{\theta,C} = T_C \cdot R_\theta \cdot T_{-C}$$

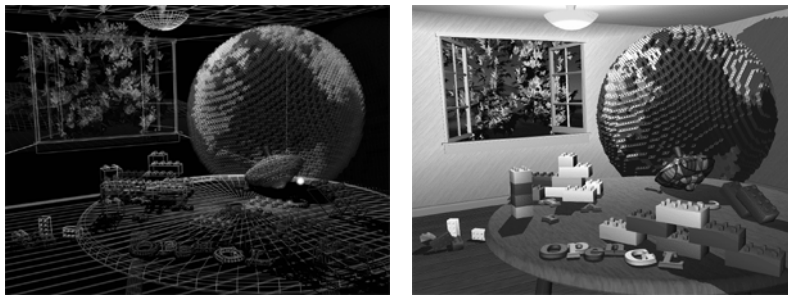
Applications des coord. homog.

- Unification (tout est multiplication)
- Généralisation (notion de point à l'infini)

- Enomément utilisées
 - En vision, robotique, géométrie projective
- Représentent les systèmes de coordonnées
 - Une transformation est un changement de repère

Création d'une image de synthèse

Principes et calcul de la projection



Description de la scène 3D

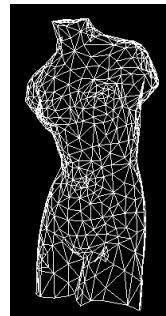
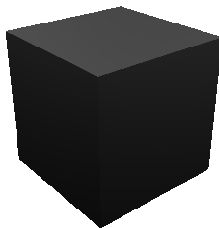
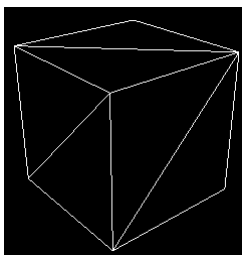
■ Un ensemble d'objets

La surface de ces objets

Un ensemble de triangles

Les positions de chaque sommet (x,y,z)

Les couleurs, normales, ... de ces sommets



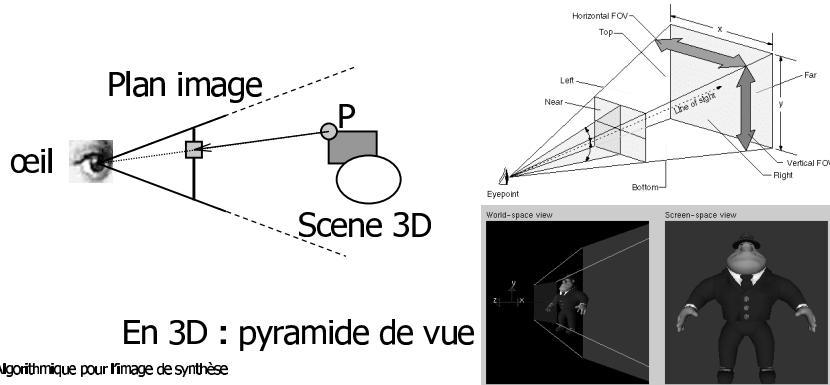
Algorithmique pour l'image de synthèse

14

Description de la caméra

- Position et orientation dans la scène
- Généralement de type projectif

Points projetés dans la direction de l'œil



En 3D : pyramide de vue

Algorithmique pour l'image de synthèse

15

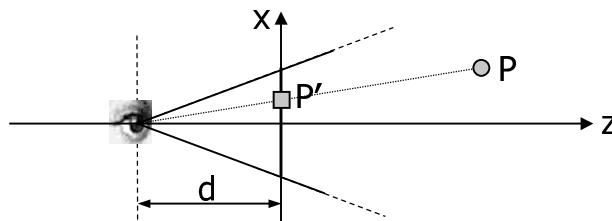
Projection 3D → 2D

- Dans cette configuration

Direction de vue alignée avec l'axe z

Axes de l'écran alignés sur x et y

Plan image en $z=0$, œil en $z = -d$ (distance focale)



- On a (Thalès) :

$$P_x / (P_z + d) = P'_x / d$$

Algorithmique pour l'image de synthèse

16

Représentation matricielle

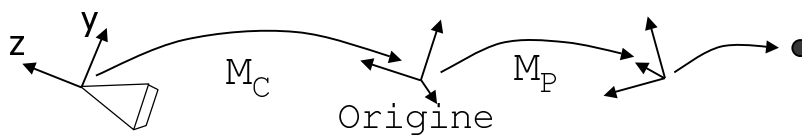
$$P'_x = dP_x / (P_z + d)$$

$$P' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d + w \end{pmatrix} \approx \begin{pmatrix} dx/(z+dw) \\ dy/(z+dw) \\ dz/(z+dw) \\ 1 \end{pmatrix}$$

- Division par P_z rendue possible par les coordonnées homogènes

Positionnement arbitraire

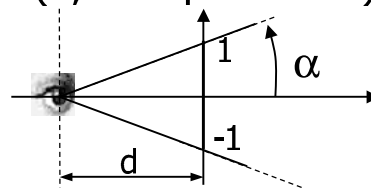
- Lorsque la caméra n'est pas alignée selon z



- On se ramène au cas précédent en • par M_C

- En pratique $d = 1 / \tan \alpha$ (α , champ de vision)

- Coordonnées écran ramenées dans $[-1, 1]$



Organisation logicielle

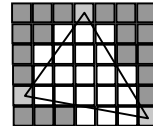
- CPU envoie les ordres graphiques au GPU
- GPU : co-processeur dédié au calcul 4x4

Projection des points, *rasterization*

Elimination des parties cachées, z-buffer

Plaquage de textures sur les polygones

Cartes programmables



- Exemples de code à copier-coller sur

artis.imag.fr/~Gilles.Debunne/Enseignement/3DAPI/

Application interactive

- Squelette classique : boucle infinie

Interaction avec l'utilisateur

Calculs, déplacement de la caméra

Affichage de la scène 3D

Effacer l'écran

Description de la caméra

Description de la scène



artis.imag.fr/~Gilles.Debunne/QGLViewer

Evolution de la discipline

- Explosion du marché et du matériel
- Loi de Moore dépassée, révolution en cours
- Tendances
 - De plus en plus de détails (~400M triangles/sec)
 - Plus de géométrie, maléable : *vertex shader*
 - Géométrie plus petite que le pixel : *pixel shader*

Calcul d'éclairage global

Méthode de radiosité

Une méthode originale
d'inversion de matrice

Contexte

- Une scène virtuelle
 - Un ensemble d'objets, définis par leurs surfaces
 - Un ensemble de sources lumineuses
- Quelle lumière reçoit chaque surface ?



- Eclairage contribue beaucoup au réalisme
- Simulation "précise" souhaitée (architecture)

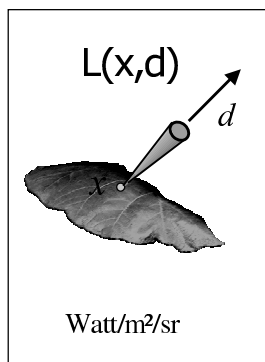
Hypothèses et principes

- Pas de phénomène ondulatoire de la lumière
- Etablissement d'un régime permanent
- Par d'interaction avec l'air (brouillard, ...)

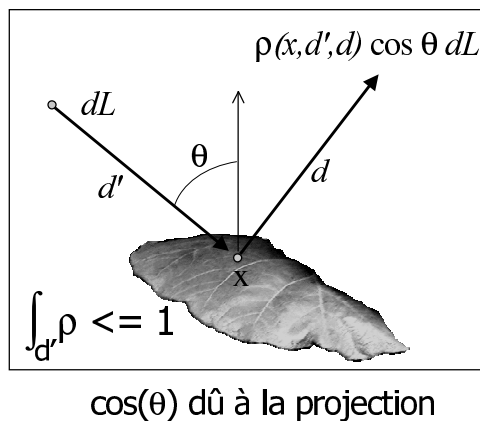
- La lumière est émise par les sources
- Elle se propage en ligne droite
- Elle est réfléchiée et absorbée par les surfaces

Energie lumineuse

Radiance L



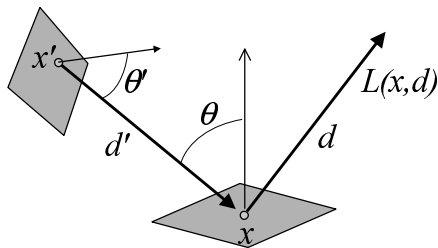
Réflectance ρ



Equation d'équilibre de la radiance

$$L(x, d) = E(x, d) + \int_{x'} \rho(x, d, d') v(x, x') \cos(\theta) dL(x', d')$$

Radiance
Emittance
Réflectance
Visibilité
Angle solide



Pas de solution analytique

Algorithmique pour l'image de synthèse

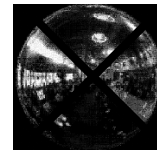
26

Hypothèse diffuse

- On suppose indépendants de la direction :

La réflectance de toutes les surfaces

L'émittance sur les sources



- On pose $B(x) = \int_{\Omega} L(x, d) \cos(\theta) d\omega = \pi L(x, .)$ (W/m²)

$$B(x) = \pi E(x) + \pi \rho(x) \int_{x'} G(x, x') B(x') dx'$$

Radiosité
Emittance diffuse
Réflectance diffuse

- Solution indépendante du point de vue

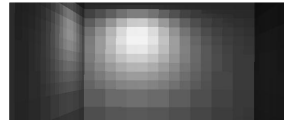
Algorithmique pour l'image de synthèse

27

Radiosité - Plan

- Introduction, notations et hypothèses
- Discrétisation du problème
 - Eléments finis
 - Projection sur une base de fonctions
 - Facteurs de forme
- Résolution de l'équation matricielle
- Méthode hiérarchique

Discrétisation



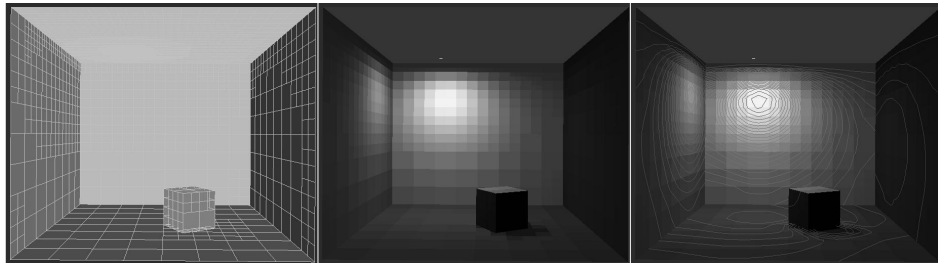
- $B(x)$ approchée sur une base de fonctions β_i
$$B(x) = \sum_i b_i \beta_i + \Delta B = B'(x) + \Delta B$$
- On projète l'équation dans cette base
$$B'(x) = E(x) + \int_{x'} G(x, x') B'(x') dx' + \Delta$$
- On essaye d'annuler le résidu Δ
- Méthode de Galerkin : Δ orthogonal aux β_i
- Classiquement : $\beta_i \equiv 1$ sur la facette i , 0 ailleurs
- On cherche $B'(x)$, représenté par un vecteur B

Méthode de Galerkin

- Résolution d'une équation matricielle

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

- F_{ij} : facteur de forme
- La ligne i représente l'équilibre de la facette i

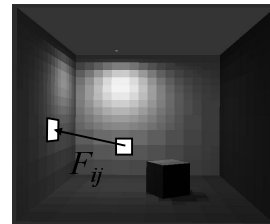


Algorithmique pour l'image de synthèse

30

Facteurs de forme F_{ij}

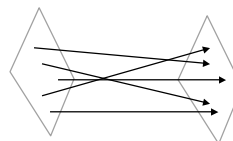
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} v(x, y) \frac{\cos(\theta) \cos(\theta')}{\pi d(x, y)^2} dx dy$$



- Proportion d'énergie quittant A_j et atteignant A_i
- Réciprocité $A_i F_{ij} = A_j F_{ji}$
- Conservation de l'énergie $\sum_j F_{ij} \leq 1$
- Calcul complexe, généralement approché

Calcul semi-analytique

Techniques de Monte-Carlo



Algorithmique pour l'image de synthèse

31

Radiosité - Plan

- Introduction, notations et hypothèses
- Discrétisation du problème
- Résolution de l'équation matricielle
 - Méthode spécifique nécessaire
 - Transferts d'énergie itératifs entre surfaces
 - Raffinement progressif du maillage
- Méthode hiérarchique

Résolution de l'équation

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j \quad (I - M) B = E$$

- Espace de spectrum
- Matrice non symétrique et pleine
- Gauss impraticable ! ($N > 500$)
- QR marche, mais est coûteux

Résolution itérative

- Soit une norme matricielle $S_{\varphi}(A) = \sup_{x \neq 0} \frac{\varphi(Ax)}{\varphi(x)}$
- Alors $\varphi(A \cdot X) \leq S_{\varphi}(A) \varphi(X)$
 $S_{\varphi}(A \cdot B) \leq S_{\varphi}(A) \cdot S_{\varphi}(B)$
- S'il $\exists S_{\varphi}$ tq $S_{\varphi}(M) < 1$, alors $I-M$ est inversible et
 $(I-M)^{-1} = \sum_{k=0}^{\infty} M^k$ $S_{\varphi}((I-M)^{-1}) \leq \frac{1}{1-S_{\varphi}(M)}$ [Atkinson'89]
- Or $S_{\varphi_{\infty}}(M) \leq \rho_{\max} < 1$
- On peut donc résoudre le système en itérant
 $B^{n+1} = E + MB^n$ B^0 indifférent

Conditionnement du problème

- Soit $\kappa_{\varphi}(A) = S_{\varphi}(A)S_{\varphi}(A^{-1})$ (conditionnement matriciel)

- On a

$$\frac{1}{\kappa_{\varphi}(A)} = \inf_{\det(B) \neq 0} \frac{S_{\varphi}(A-B)}{S_{\varphi}(A)}$$

or

$$1 \leq \kappa_{\varphi_{\infty}}(I-M) \leq \frac{1+\rho_{\max}}{1-\rho_{\max}}$$

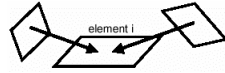
Le problème est très bien conditionné (peu sensible aux instabilités) pour des scènes à faible réflectance (<0.5)

- Les itérations de propagation de la lumière convergent d'autant plus vite que ρ_{\max} est petit

$$\varphi_{\infty}(B^n - B) \leq \rho_{\max}^n \varphi_{\infty}(B^0 - B)$$

Deux types de résolution itérative

■ Gathering

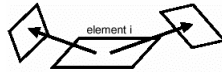


$$B_i^{k+1} = E_i + \rho_i \sum_{j=1}^{i-1} F_{ij} B_j^{k+1} + \rho_i \sum_{j=i+1}^n F_{ij} B_j^k$$

ou

$$B_i^{k+1} = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j^k$$

■ Shooting

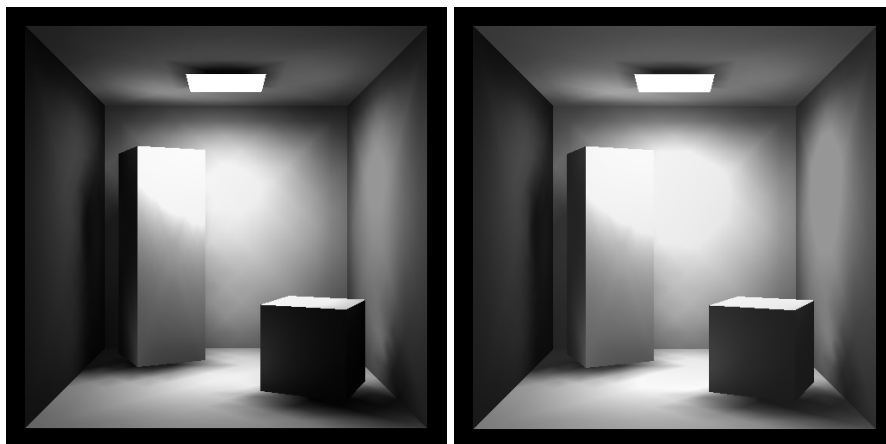


Initialisation $\Delta B_j^{n+1} = 0$

$$\text{Puis, pour tout } i, j \quad \Delta B_j^{n+1} += \Delta B_i^n \rho_j F_{ji} \quad \Delta B_i^0 = E_i$$

$$B^{n+1} += \Delta B^{n+1} \quad B_i^0 = E_i$$

Exemple de résolution itérative

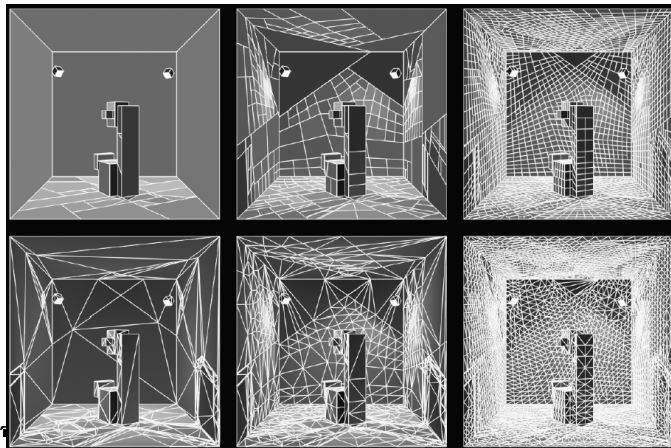


Départ de zéro

Départ d'un terme constant

Raffinement progressif

- Maillage créé en fonction de la variation de l'irradiance et des discontinuités

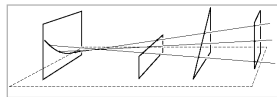


38

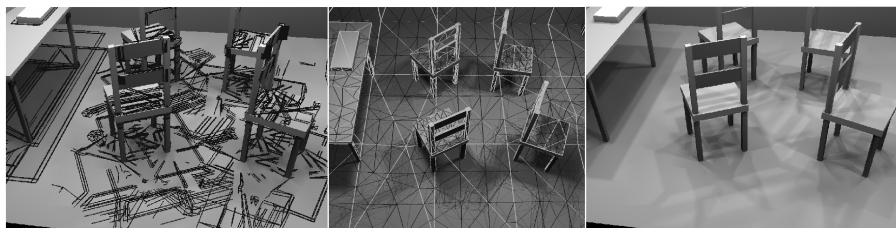
Maillage de discontinuité

- Difficultés

Discontinuités nombreuses et de nature variable



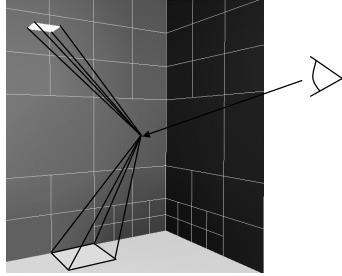
Evènement triple-arête \Rightarrow quadrique



39

Final gather (récolte finale)

- Lancer de rayons depuis le point de vue
Re-calcul en chaque point de la visibilité et de l'éclairage

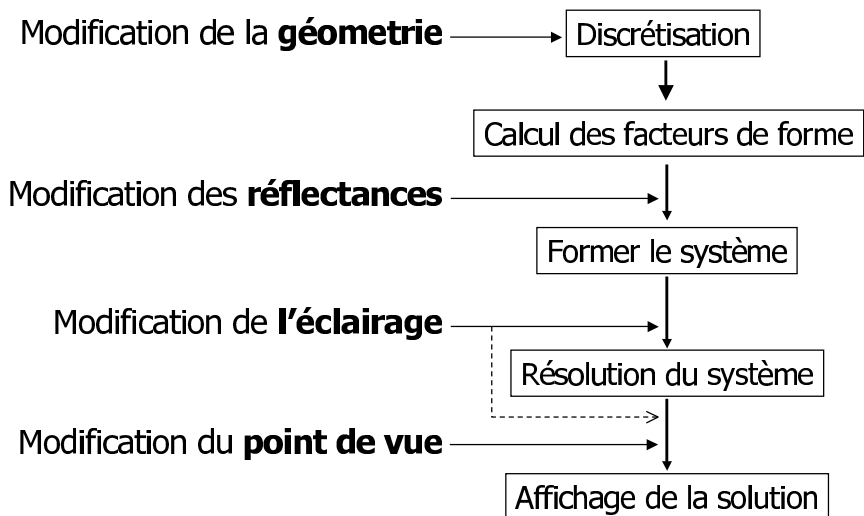


- Ombres parfaites, mais
Très coûteuses
Dépendant du point de vue

Algorithmique pour l'image de synthèse

40

Ordre des opérations



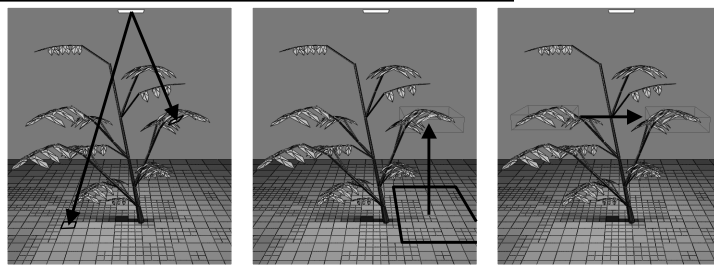
Algorithmique pour l'image de synthèse

41

Radiosité - Plan

- Introduction, notations et hypothèses
- Discrétisation du problème
- Résolution de l'équation matricielle
- Méthode hiérarchique
 - Accélérer les temps de calculs
 - Ne propager l'énergie que de là où il y en a
 - Regrouper les échanges d'énergie

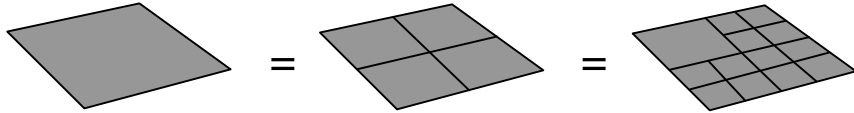
Radiosité hiérarchique



- Etablir les transferts d'énergie (liens)
 - Suffisamment haut pour économiser des calculs
 - Suffisamment bas pour conserver la précision
- Trois phases par itération
 - Raffinement \implies assure la complétude des échanges
 - Gather* \implies transfert de l'énergie le long des liens
 - Push/Pull* \implies cohérence multi-échelles

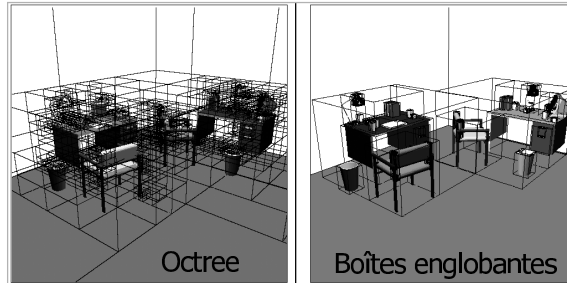
Représentation hiérarchique

■ Des surfaces



■ Des objets

Clusters

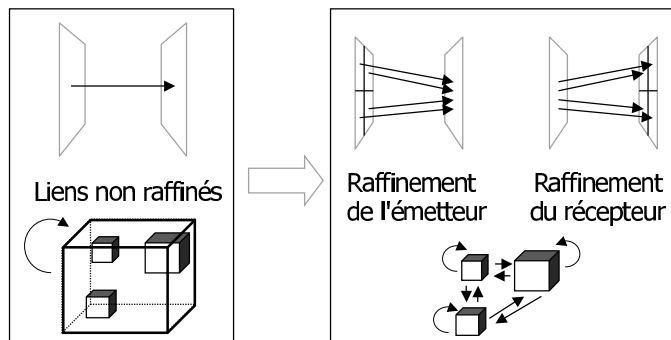


Algorithmique pour l'image de synthèse

44

Raffinement des liens

■ Départ : un unique lien de la scène sur elle même



■ Oracle de raffinement

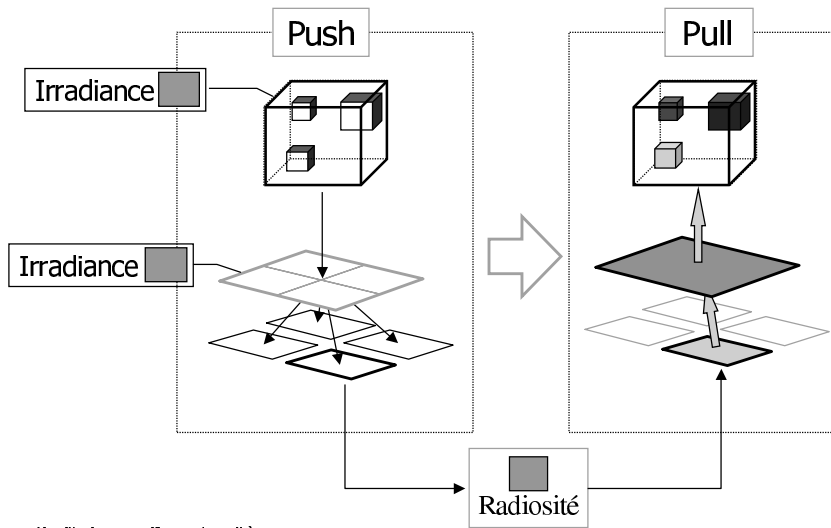
Rendre l'erreur uniforme

Exemple: BF (radiosité x Facteur de forme)

Algorithmique pour l'image de synthèse

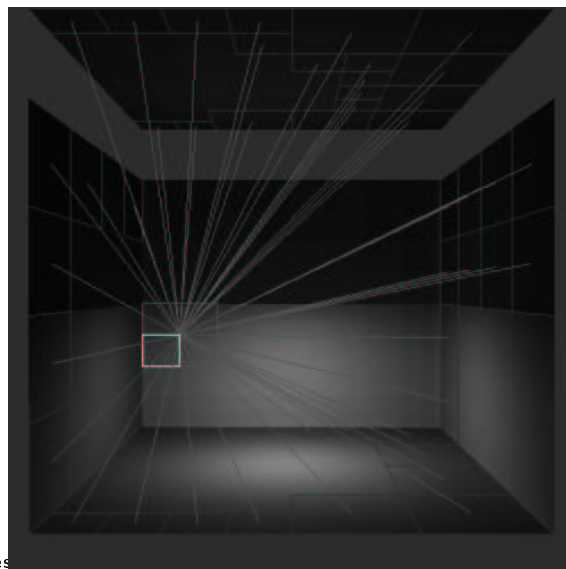
45

Push / Pull



46

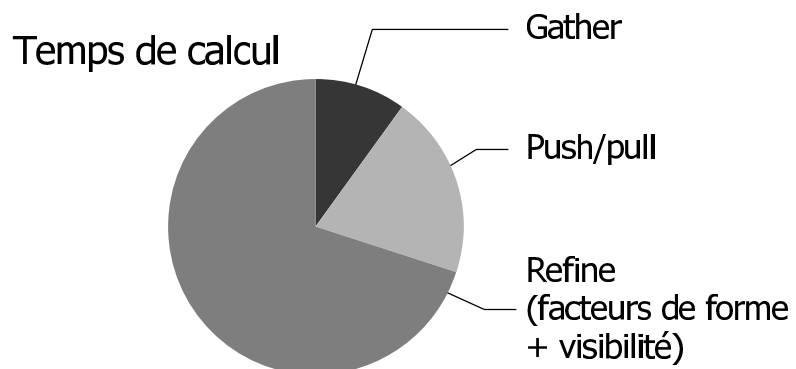
Illustration de la hiérarchie



Algorithmique pour l'image de s

47

Répartition des coûts



[Holzschuch94]

Exemples de résultats



Les quaternions

LA représentation des rotations en 3D

Qu'est ce qu'une rotation ?

- Isométrie préservant une droite : l'axe

- Expression matricielle

Simple pour un axe X, Y ou Z

$$R_Z = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Cas général : une matrice 3x3 orthonormale

- Espace de dimension 3

9 (matrice 3x3) – 3 (unitaire) – 3 (orthogonale)

Inconvénients

- Passage (axe, angle) \leftrightarrow matrice non trivial
- Imprécisions numériques lors de compositions
Gram-Schmidt pour re-orthonormaliser
- Comment interpoler entre deux matrices ?
L'interpolation linéaire détruit l'orthonormalité

$$\begin{array}{ccc} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \downarrow & \downarrow & \downarrow \\ t=0 & t=1/2 & t=1 \end{array}$$

Propriétés souhaitées

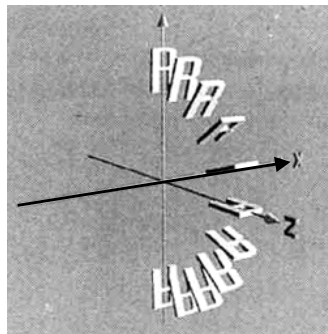
- Interpolation entre orientations
- Stabilité numérique (composition de rotations)
- Expression simple et intuitive
- Efficacité (création, application, modification)
- Conversion matricielle aisée (hardware)

Angles d'Euler

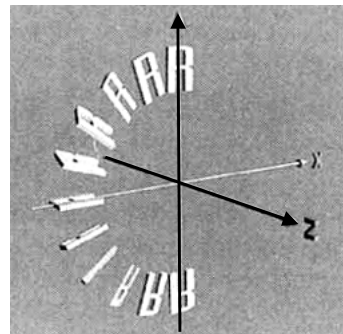
- Extension des coordonnées polaires en 3D
- Rotations successives autour de X, Y puis Z
Cet ordre est arbitraire : définir un standard
Il suffit que deux axes successifs soient différents
- Espace de dimension 3
- Simple, relativement intuitif

Problème avec les angles d'Euler

- Il n'y a pas unicité de la représentation



Rotation autour de X



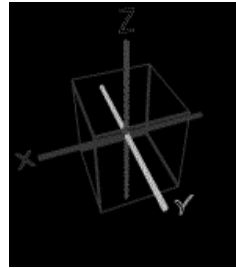
Rotation autour de Y puis Z

- Problème de définition pour l'interpolation

Interpolation des angles d'Euler

- Interpoler les 3 angles
- Trajet dépend de la représentation choisie

- Gimbal lock
 - Quand une des rotation passe par 90°
 - On perd un degré de liberté



Représentation par axe et angle

- 3 paramètres (axe normalisé et angle)
- Simple et intuitif
- Interpolation de l'axe et de l'angle
 - Plus de Gimbal lock

- Calculs assez complexes
 - Rotation d'un point (produits vectoriels)
 - Composition de rotations (utiliser les matrices)

- Manque une algèbre propre

Les quaternions

- Extension en 4D des nombres complexes

Dans \mathbb{C} : base $(1, i)$ avec $i^2 = -1$

- 4D : base $(1, i, j, k)$ $i^2 = j^2 = k^2 = ijk = -1$ $ij = k$ \cup

- $q = x + yi + zj + wk = (x, y, z, w)$

- Quaternion : vecteur unitaire de \mathbb{R}^4

$$x^2 + y^2 + z^2 + w^2 = 1$$

- Notation

$$q = v_x + v_y i + v_z j + wk = (v_x, v_y, v_z, w) = (\vec{v}, w)$$

Le groupe des quaternions

- On définit le produit interne

$$q_1 = (v_1, w_1) \quad q_2 = (v_2, w_2)$$

$$q_1 \bullet q_2 = (w_1 v_2 + w_2 v_1 + v_1 \times v_2, w_1 w_2 - v_1 \bullet v_2)$$

Non commutatif

- $q_1 \bullet q_2$ est un quaternion
- $q = (0, 0, 0, 1)$ est l'élément neutre
- Structure de groupe

Quaternion et axe-angle

- On représente une rotation u, θ par

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sin(\theta/2) \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \quad w = \cos(\theta/2)$$

$$|q| = \sin^2(\theta/2) |u|^2 + \cos^2(\theta/2) = 1$$

- Avec $q_1 = R_1(u_1, \theta_1)$ et $q_2 = R_2(u_2, \theta_2)$
 $q_1 \cdot q_2$ est la composition de R_1 et R_2 !!

Utilisation des quaternions

- Représentation matricielle

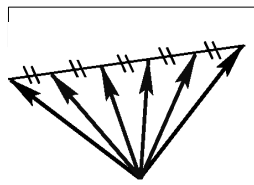
$$R = \begin{bmatrix} 1-2y^2-2z^2 & 2xy+2wz & 2xz-2wy \\ 2xy-2wz & 1-2x^2-2z^2 & 2yz+2wx \\ 2xz+2wy & 2yz-2wx & 1-2x^2-2y^2 \end{bmatrix}$$

- Rotation inverse : $(\vec{v}, w) \rightarrow (-\vec{v}, w)$
- Rotation d'un point p

$$\text{Rot}_q(p) = q \cdot (p, 0) \cdot q^{-1}$$

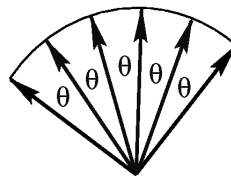
Interpolation de quaternions

- Parfaitement adaptés
- Interpolation sur la sphère unité
- Interpolation linéaire n'est pas unitaire
- Spherical Linear intERPolation
Vitesse angulaire de rotation constante

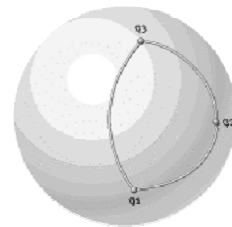


Lerping

Algorithmique pour l'image de synthèse



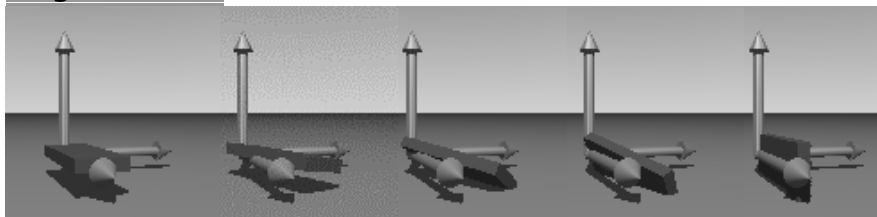
Slerping



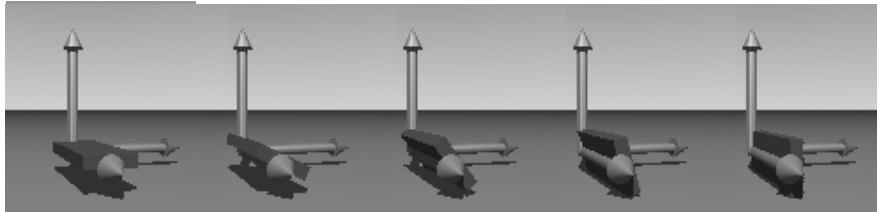
62

Résultats d'interpolation

Angles d'Euler



Quaternions



Algorithmique pour l'image de synthèse

Images de Rick Parent

63

Quaternions - Conclusion

- Formalisme de choix pour les rotations 3D
- Relativement complexes
 - Implémentation et mathématiques
- Défini à un signe près $(v, w) = (-v, -w)$
 - Choisir le chemin le plus court sur la sphère
- Extension à nD par l'algèbre de Clifford
 - Utiliser du code existant

Simplification de maillage

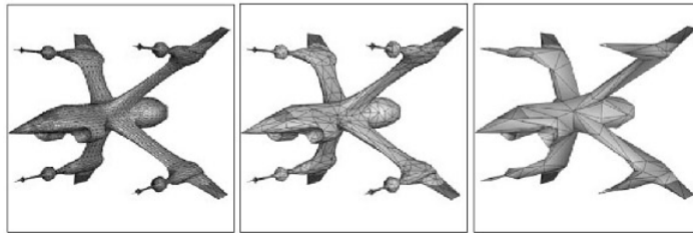
Niveaux de détails

— Introduction

- Context of Real-time Rendering
 - A large model to display
 - A fixed amount of time (30 fps)
 - A fixed resolution
- Strategies
 - Display only what is visible
 - Display objects with appropriate details
 - Use simplified version (also for side computations)
 - collision detection
 - shadows
- Optimize rendering

Level of Detail: Principle

- For each object in the scene
 - Store different versions
 - Choose appropriate version
- How to build the simplified versions?
 - How many? Which degree of simplification?
- How to choose appropriate version?



Overview of presentation

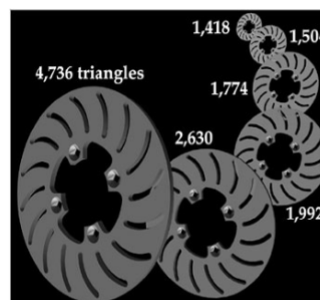
Mesh Simplification
Error Metrics
Selection of LOD
An example: QSlim
Extensions

Simplification Framework

- Topology preservation
- Local operators
 - work on vertices, edges, faces
 - locally decrease polygon count
- Global operators
 - treat the object as a whole
 - more like "resampling"

Topology preservation

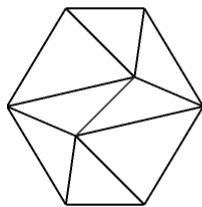
- Maintain Genus
- Can not merge parts
- Limit the simplification
- Not always required



Local operators

- Edge Collapse
- Vertex pair collapse
- Vertex removal
- Cell collapse

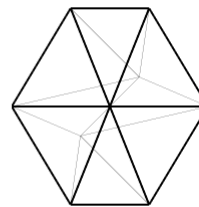
Edge Collapse



edge collapse

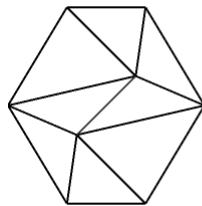


vertex split



Where to place the new vertex?

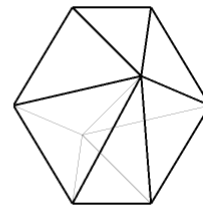
Edge Collapse



edge collapse



vertex split



- On the edge

- leaves freedom (but where?)
- introduces a new point

- On one extremity

- lower quality
- no new point
- **half-edge collapse**

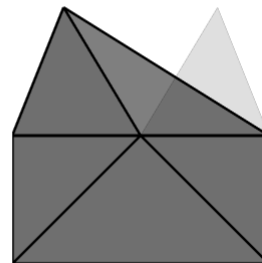
Vertex Pair Collapse

- Generalize edge collapse

- Collapse any pair of vertices
- Must choose new position

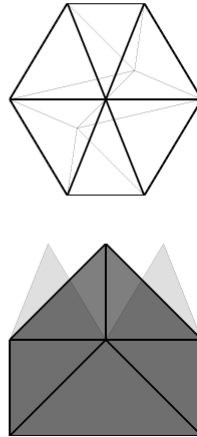
- Can not consider all pairs

- avoids quadratic complexity
- selects "close" vertices
- achieves linear complexity in practice



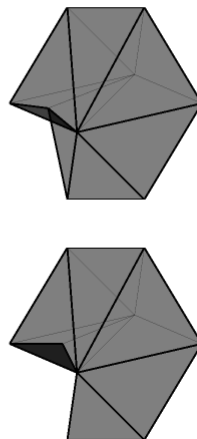
Collapse Considerations

- Post collapse operations
 - Removal of degenerated faces
 - Update of adjacency relations



Collapse Considerations

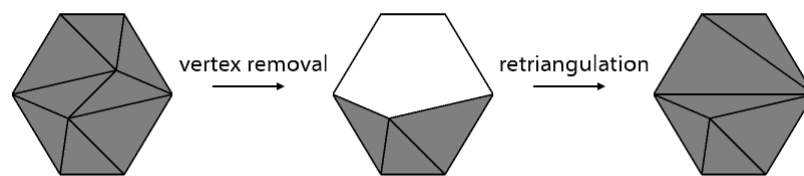
- Post collapse operations
 - Removal of degenerated faces
 - Update of adjacency relations
- Pre collapse operation
 - Avoid mesh fold-over
 - Avoid topology inconsistency



Local operators

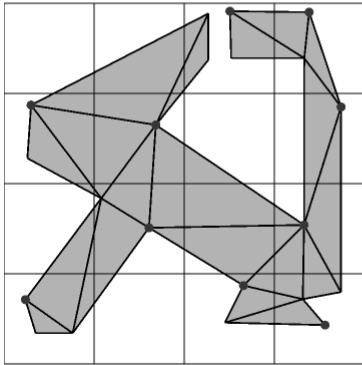
- Edge Collapse
- Vertex pair collapse
- Vertex removal
- Cell collapse

Vertex Removal



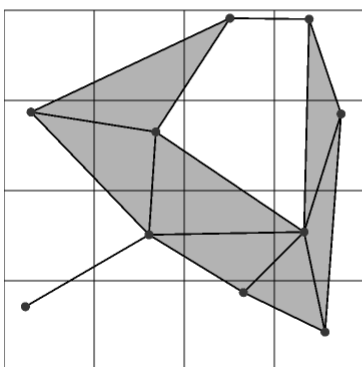
- "Filling the hole" can be tricky
 - Polygon is not "planar"
- Many triangulation
 - Encompass half edge collapse
- Generalized by "polygon merging"

Cell collapse



- Place the model in a grid
- Choose a representant per cell
 - one of the point
 - center of the cell, barycenter,...

Cell collapse



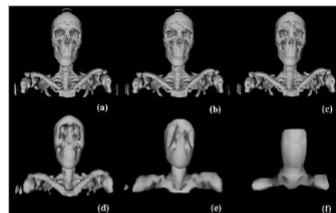
- Place the model in a grid
- Choose a representant per cell
 - one of the point
 - center of the cell, barycenter,...
- Collapse cell's points on representant
- Clean degeneracies
 - remove empty triangles
 - create segments and points

Simplification Framework

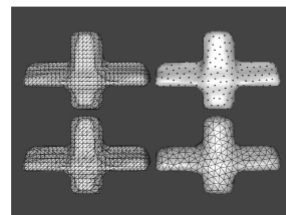
- Topology preservation
- Local operators
 - work on vertices, edges, faces
 - locally decrease polygon count
- Global operators
 - treat the object as a whole
 - more like "resampling"

Global operators

- Volume Rendering [He95]
 - Rasterize the mesh in a grid
 - Low-pass filter the grid
 - Reconstruct the mesh (marching cube)



- Re-tiling polygonal surfaces [Turk92]
 - Distribute point on the surface
 - Re-triangulate the points



Morphological operators



- "Erase" small details
- Merge disconnected parts
- Cannot simplify past some point
- Generalized with alpha hulls [EISana98]

Overview of presentation

Mesh Simplification

Error Metrics

Selection of LOD

An example: QSlim

Extensions

Error Metrics

- Why measure Error?
 - Guide the simplification process
 - Know the quality of the results
 - Know when to show a particular LOD
 - Balance quality over a scene
- Key elements
 - Geometric Error
 - Attribute Error
- Incremental vs. Total Error

Error Metrics - geometric error

- How to measure the distance between two surfaces?

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

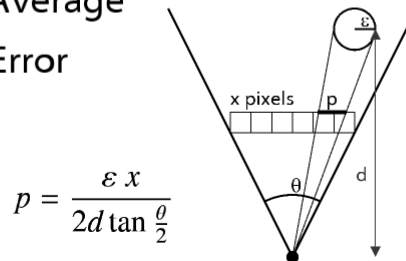
- Hausdorff distance

$$H(A, B) = \max(h(A, B), h(B, A))$$

- Approximations

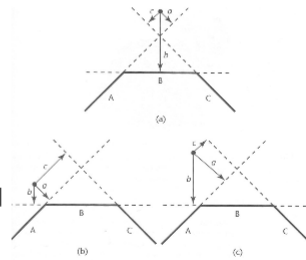
- Maximum vs. Average

- Screen Space Error



Measuring distance: approaches

- Vertex-Vertex
 - natural for cell collapses
 - miss surface changes
- Vertex-Plane
 - distance to supporting faces
 - innacurate in theory
- Vertex-Surface
 - distance to closest point
 - Progressive Meshes [Hoppe93]
- Surface-Surface
 - strongest error bound
 - hard to compute
 - Simplification Enveloppes [Cohen96]

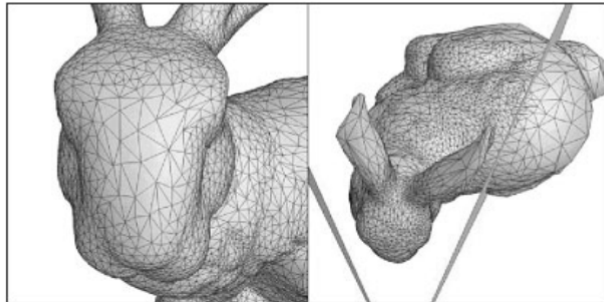


Overview of presentation

Mesh Simplification
Error Metrics
Selection of LOD
An example: QSlim
Extensions

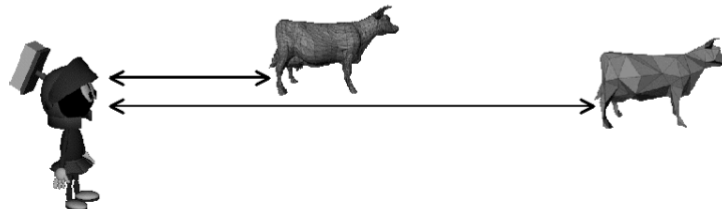
LOD Selection

- LOD type
 - discrete, continuous, view-dependent



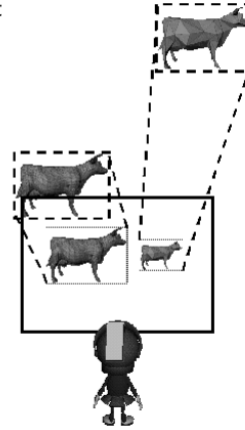
LOD Selection

- LOD type
 - discrete, continuous, view-dependent
- Selection criteria
 - distance
 - screen size



LOD Selection

- LOD type
 - discrete, continuous, view-dependent
- Selection criteria
 - distance
 - screen size



LOD Selection

- LOD type
 - discrete, continuous, view-dependent
- Selection criteria
 - distance
 - screen size
 - priority, hysteresis, environmental conditions, perceptual factors
- Blending Between Transitions
 - alpha blending
 - geomorphs

— Overview of presentation

Mesh Simplification

Error Metrics

Selection of LOD

An example: QSlim

Extensions

— QSlim algorithm

- For each vertex compute initial quadric
- Build all pairs of "close" vertices
- For each pair:
 - find best position for collapsing
 - if possible, inverse matrix $Q+Q'$
 - otherwise, test extremities and middle point
 - compute associated error
- Choose collapsing with smallest error
 - compute new quadric for collapsed vertex
 - update list of pairs
- Iterate until:
 - given number of faces reached
 - given error reached

Overview of presentation

Mesh Simplification

Error Metrics

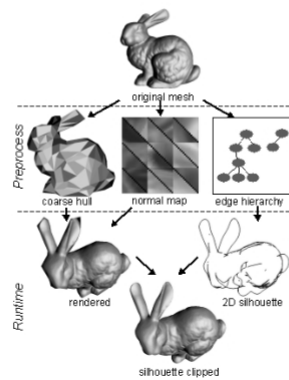
Selection of LOD

An example: QSlim

Extensions

Silhouette Clipping [Sander00]

- Silhouette is visually important
- Mesh simplification degrades silhouette



— Alternatives representations

- Hard to represent a shape with few polygons
- An image can convey many details!
- Use Image Based Representations (IBR)!
- Imposters [Maciel95,Decoret99]
 - replace distant parts with a textured quad
 - looks nice when static
 - looks flat when moving
- Billboard Clouds [Decoret03]
 - approximate roughly shape with few planes
 - use semi-transparent textures for finer details
 - generalize classic tricks for games (trees)

Bibliographie (1/3)

- Foley, van Dam, Feiner, Hughes
Computer Graphics, Addison-Wesley



- Présentation des API, détails sur OpenGL
- Code à copier-coller pour les différentes API
artis.imag.fr/~Gilles.Debunne/Enseignement

- libQGLViewer : viewer 3D facile  
artis.imag.fr/~Gilles.Debunne/QGLViewer

Bibliographie (2/3)

- Graustein, W. C.
Homogeneous Cartesian Coordinates. Linear Dependence of Points and Lines.
Ch. 3 in *Introduction to Higher Geometry*,
pp. 29-49, 1930.
- E.W. Weisstein, *Books about Quaternions*
www.ericweisstein.com/encyclopedias/books/Quaternions.html
- Interpolation de quaternions
Ken Shoemake in SIGGRAPH '85

Bibliographie (3/3)

- F. Sillion

A unified hierarchical algorithm for global illumination with scattering volumes and object clusters.

IEEE Transactions On Graphics. 1(3), sept 1995.

- Niveaux de détail

<http://www.lodbook.com>

- M. Garland and P. Heckbert. *Surface simplification using quadric error metrics.* 1997.