

Organisation automatique de scènes 3D

Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, François Sillion

ARTIS/GRAVIR/IMAG-INRIA

Aurelien.Martinet@imag.fr

Résumé : Dans cet article, nous présentons une méthode permettant de retrouver des informations de cohérence dans un ensemble de polygones non structuré. On cherche en particulier à mettre en évidence les structures répétées dans une scène (exemple : les sièges d'un avion). À partir de structures élémentaires, nous générons une organisation de plus haut niveau par agglomération. Pour « guider » ce regroupement, nous utilisons des informations de proximité, d'orientation et de ressemblance géométrique entre objets. À partir de ces informations, nous recherchons ensuite les structures redondantes de la scène, cette recherche étant effectuée de manière efficace à l'aide d'un KDTree de dimension élevée. Nous montrons d'autre part dans cet article les limitations des méthodes classiques ainsi qu'une solution à leurs problèmes.

Mots-clés : Traitements de modèles 3D

1 Introduction

En synthèse d'images, on assiste à une très forte augmentation de la complexité des données 3D ainsi qu'à une diversification de leur provenance : scanners 3D, modélisation géométrique, reconstruction à base d'images, instantiation de modèles procéduraux, etc... Le format d'échange de ces données est souvent pauvre en information et peu approprié aux différentes tâches rencontrées en synthèse d'images. Les conversions entre formats ne garantissent pas une conservation des informations sémantiques et/ou topologiques initialement présentes dans le modèle. Partant de ce constat, la nécessité de structurer ces données afin d'en faciliter l'utilisation est devenue une priorité dans des domaines tels que la visualisation temps-réel, la manipulation interactive de modèles ou de scènes 3D et la simulation numérique.

La structuration automatique de données 3D au niveau d'une scène est également un problème récurrent dans le domaine de la robotique et de la vision. Les problématiques concernent essentiellement la structuration de nuages de points et la détection de zones d'obstacles. En informatique graphique, divers algorithmes ont été proposés afin d'organiser de manière hiérarchique des environnements 3D[CDP95, HDSD99], leurs points communs étant le pré-requis de posséder une structure d'objet dans la scène. D'autres méthodes proposent de contraindre les mouvements dans cet environnement[XSF02], ce qui permet d'organiser les objets les uns par rapport aux autres.

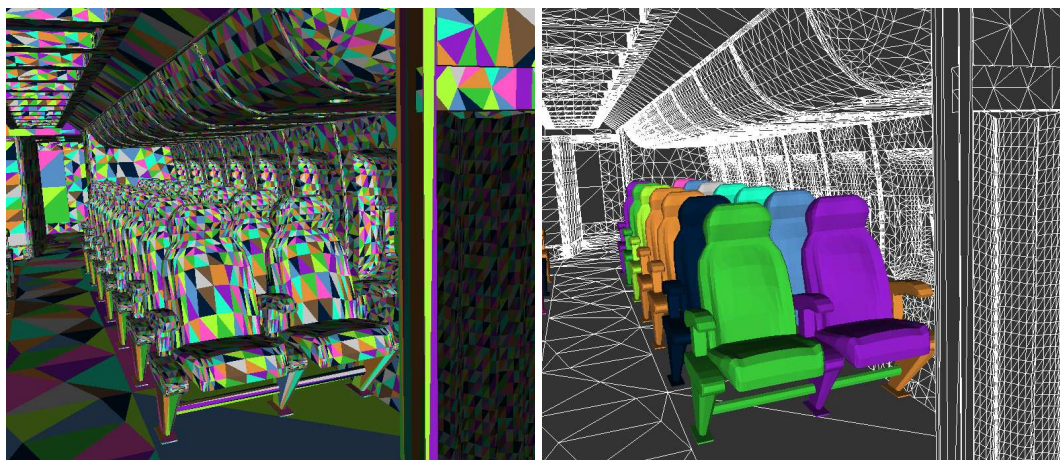


FIG. 1 – A partir d'un ensemble désorganisé de polygones (image de gauche), notre méthode permet d'identifier des structures géométriques répétées. Cette recherche peut être globale, ou restreinte par l'utilisateur à la recherche d'une seule structure (les sièges sur l'image de droite). Model courtesy of LightWork Design Ltd.

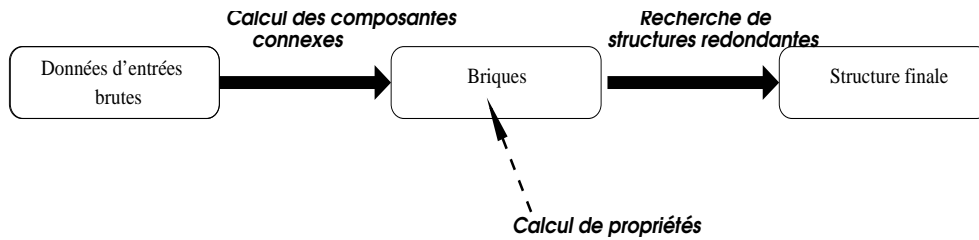


FIG. 2 – Le pipeline des traitements.

Dans cet article, nous proposons une méthode d'organisation des données basée sur des techniques d'identification et de comparaison de sous-ensembles pertinents de la scène comme montré à la Figure 1. La présence de structures répétées étant très fréquente dans les scènes « architecturales » : maison, église, chateau, usine, etc. nous restreignons dans un premier temps notre étude à ce type de scènes.

Notre approche est divisée en plusieurs étapes (voir Figure 2) :

Structuration primaire des données d'entrées : Lors de l'échange de données 3D, il est fréquent de disposer de seules données brutes, communément appelées « soupe de polygones ». À partir de cet ensemble arbitraire de polygones 3D sans information de connectivité, on crée une première structure de la scène en regroupant les polygones connexes par arêtes. Dans cet article, ces groupes sont appelés « briques ».

Caractérisation : Afin de pouvoir caractériser les briques géométriquement et spatialement, on a recours à :

- des techniques de caractérisation de formes 3D, permettant leur comparaison. Pour des raisons de robustesse, la similarité entre deux briques est basée sur la surface géométrique qu'elles décrivent et non pas sur la topologie de leurs maillages ;
- un système robuste de recherche de l'orientation de ces briques, supportant les cas dégénérés.

Recherche de structures redondantes : À partir des briques et de leurs caractéristiques, on recherche des assemblages de briques redondants dans la scène, *i.e* formant des structures similaires à une transformation géométrique près.

Dans les sections suivantes, nous présentons en détail les différents points de notre système. Nous présentons dans un premier temps, la structure élémentaire que nous utilisons pour construire une structure plus évoluée. Nous présentons également dans cette section les principes utilisés pour calculer deux propriétés essentielles de ces structures qui sont : un système d'axes local et une signature permettant d'identifier une brique. Dans la seconde section, nous détaillons l'algorithme utilisé pour construire la structure de haut niveau, avant de présenter quelques résultats.

2 Création de la structure de base

Notre structure élémentaire est représentée par la notion de **brique**, définie comme un ensemble de polygones connexes par arêtes (voir Figure 3).

On ne peut pas être assuré que les briques obtenues représentent des modèles 3D manifold. Des techniques récentes de réparations d'objets existent [Ju04] mais nécessitent un long traitement. Deux caractéristiques essentielles doivent être calculées sur ces briques :

- une orientation robuste, *i.e.* définie de manière unique pour chaque brique,
- une signature permettant la comparaison entre deux briques.

Nous détaillons ces deux points dans la suite de cette section.

2.1 Création d'un système d'axes robuste

On cherche à définir une orientation pour une brique de manière robuste. La technique communément utilisée est l'Analyse en Composante Principale (ACP) [Kaz04], dont on rappelle brièvement le principe, avant de présenter

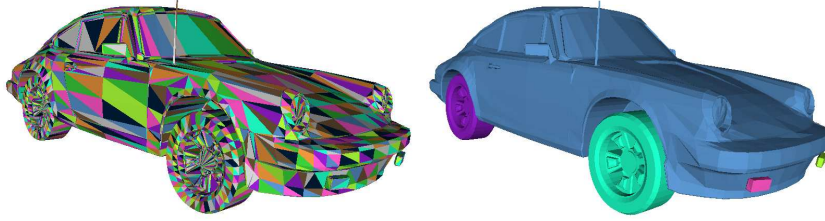


FIG. 3 – A gauche, les triangles du modèle. A droite, les briques du modèle, formées de polygones connexes par arêtes. On remarque expérimentalement que ces briques correspondent le plus souvent à un découpage cohérent du modèle.

ses limitations.

Soit une surface triangulée S en 3D. La matrice de covariance M est définie par :

$$M_{i,j} = \sum_{T \in S} \int_T x_i x_j dx$$

ou T parcourt les triangles du modèle. Les trois vecteurs propres de la matrice M sont les axes principaux d'inertie du modèle, les valeurs propres associées représentant la variance du modèle le long de ces directions. Ces vecteurs sont utilisés pour former une base caractéristique de la brique.

Une limitation forte de l'ACP est que lorsque la matrice de covariance M est dégénérée, on ne peut pas calculer un système d'axes unique. La matrice de covariance peut être dégénérée dans deux cas :

- Lorsque les trois vecteurs propres possèdent la même valeur propre. Dans ce cas, la matrice de covariance est un multiple de la matrice identité et n'importe quel système d'axes est valide pour l'objet.
- Lorsque deux vecteurs propres v_1 et v_2 ont la même valeur propre λ et que le troisième vecteur v_3 a une valeur propre différente λ_3 . Dans ce cas, n'importe quelle combinaison linéaire des vecteurs v_1 et v_2 est aussi un vecteur propre de valeur propre λ .

Les formes 3D possédant une matrice de covariance dégénérée sont appelées *formes dégénérées*.

Afin de caractériser ces formes dégénérées et proposer une solution adaptée, on reformule le principe de calcul d'axes par ACP. On peut montrer[Smi02] que le calcul d'axes par ACP est équivalent à chercher les directions pour lequel le moment d'ordre 2 est maximale.

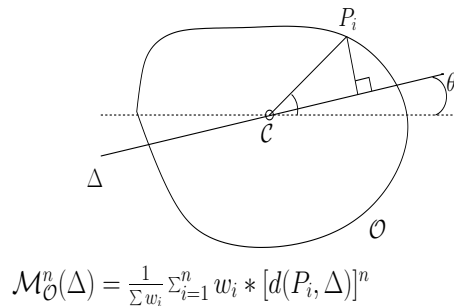


FIG. 4 – Formule de calcul du moment d'ordre n pour une forme O quelconque.

En appelant P_i les points échantillonnés sur la surface d'un objet, et w_i les poids associés, le principe de calcul du moment d'ordre n par rapport à une droite Δ est donné Figure 4.

En se plaçant dans le plan, on remarque que certaines figures très simples possèdent des moments d'ordre 2 indépendants de la direction de Δ .

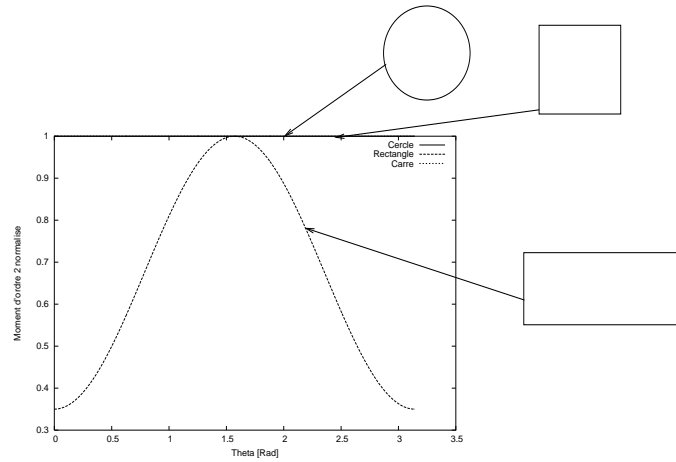


FIG. 5 – Figures 2D et leurs moments d’ordre 2 en fonction de la direction de la droite Δ .

Une figure comme le carré possède un moment d’ordre 2 constant. Cela signifie qu’il n’est pas possible d’assigner de manière fiable un système d’axes à cette figure à l’aide de l’ACP. On montre plus tard que cette propriété s’étend aux objets tridimensionnels, ce qui implique, par exemple, la non-unicité du système d’axes de l’ACP pour un cube.

Afin de résoudre ce problème, et de proposer une réponse fiable pour ces figures, on propose d’utiliser les moments d’ordres plus élevés. En effet, on remarque (voir Figure 7) que ces moments présentent des extrema même pour une forme dégénérée, ce qui **garantit la construction d’axes fiables pour cette forme**. On utilise en particulier les moments d’ordre pair car ces derniers peuvent être calculés de manière efficace à l’aide d’une base d’harmoniques sphériques (voir Implémentation).

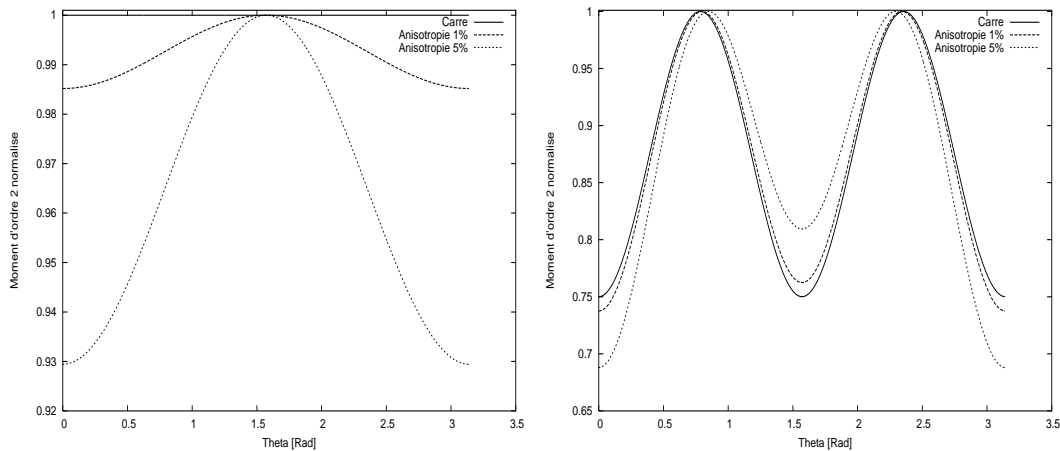


FIG. 6 – Moment d’ordre 2 en fonction de la direction de la droite Δ pour un carré et pour des formes « légèrement » anisotropes ($Anisotropie = 1 - \frac{hauteur}{largeur}$). A gauche : moment d’ordre 2. A droite : moment d’ordre 4. On remarque que les moments d’ordre 4 présentent des extrema même pour une forme dégénérée (le carré).

Dans notre implémentation, on choisit d’utiliser les moments d’ordre 4, afin d’obtenir des résultats satisfaisants tout en limitant la complexité des calculs.

En utilisant les moments d’ordres élevés, on perd une propriété importante de l’ACP. En effet, les extrema des moments d’ordres élevés ne sont pas obligatoirement dans des directions orthogonales. Afin de garantir l’obtention d’axes orthogonaux, on définit le première axe v_1 comme l’orientation (θ_1, ϕ_1) pour laquelle le moment est minimum. Le second axe correspond à la direction de moment minimum dans le plan perpendiculaire à v_1 et le troisième v_3 est définie par $v_3 = v_1 \wedge v_2$.

Pour la création d’un système d’axes local, on a maintenant deux possibilités :

- l’utilisation unique des moments d’ordre 4,
- l’utilisation conjointe des moments d’ordre 2 (ACP) et des moments d’ordre 4.

Dans notre implémentation, nous utilisons **uniquement les moments d’ordre 4** pour déterminer une orientation robuste pour chaque brique.

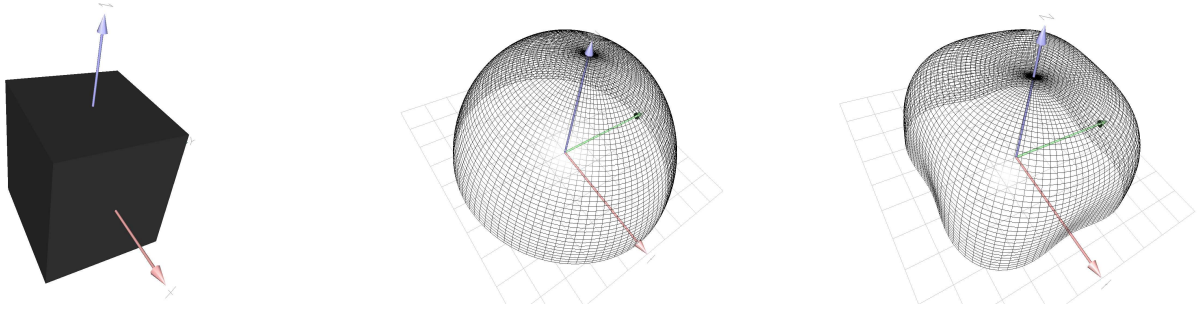


FIG. 7 – Le moment d’ordre 2 et 4 pour un cube. Le moment d’ordre 2 (Figure du milieu) est constant alors que le moment d’ordre 4 (Figure de droite) possède des extrema permettant d’assigner un système d’axes au cube.

Implémentation

Le calcul des moments d’ordre plus élevé que l’ordre 2 est généralement coûteux en temps de calcul. En effet, il nécessite un “double échantillonnage” : un échantillonnage des points de la surface de l’objet, ainsi qu’un échantillonnage des directions (θ, ϕ) . On montre (Annexe 1) que la complexité de calcul des moments d’ordre n pour $n > 2$ peut être largement diminué pour n pair, en utilisant une base d’harmoniques sphériques.

2.2 Comparer deux briques

Afin de pouvoir mettre en évidence les structures récurrentes de la scène 3D, il nous faut une mesure de similarité entre les briques, composantes élémentaires de notre système. On cherche ici à « retrouver » dans la scène, des briques géométriquement identiques, en ne tenant pas compte de leur structure topologique.

Définition : Deux briques sont dites identiques si elles décrivent la même surface géométrique, indépendamment du maillage de cette surface.

Pour établir cette comparaison, on sépare le processus en deux étapes :

- on utilise le système d’axes calculé précédemment pour effectuer un premier tri ;
- on utilise ensuite une approche plus discriminante basée sur les travaux publiés dans le domaine du *Shape Matching*.

Dans un premier temps, on effectue un premier tri en s’inspirant de la méthode de descripteur à vues multiples, outil proposé par le standard MPEG-7[Mar02]. Pour chaque objet, on positionne trois caméras le long des axes du repère orthogonal et on dirige ces caméras vers le centre de masse de l’objet.

Pour chacune de ces caméras, on projette l’objet parallèlement le long des axes et on stocke l’aire de cette projection. On stocke donc trois valeurs par objet qui seront comparées deux à deux lors de la comparaison entre deux briques. Grâce à l’utilisation des Occlusions Queries[Reg02] pour le calcul de l’aire projetée, ce premier traitement se révèle peu coûteux, bien que peu discriminant dans la recherche d’objets identiques.

Après ce premier tri, on développe une méthode en s’inspirant des travaux publiés dans le domaine de la recherche d’objets similaires (*Shape Matching*) et plus particulièrement des travaux d’Osada[OFCD01] sur les *Shape Distributions*.

Dans la littérature, cette méthode se place parmi les nombreuses méthodes qui consistent à associer à une forme 3D quelconque une signature appelée « descripteur de formes » (voir Figure 8).

La méthode des *Shape Distribution* consiste à échantillonner une fonction de forme, qui mesure des propriétés géométriques globales de l’objet. Les caractéristiques principales de cette approche sont : son invariance par rotation, sa robustesse vis-à-vis des dégénérescences éventuelles du maillage (triangle d’aire nulle, sommet en T)

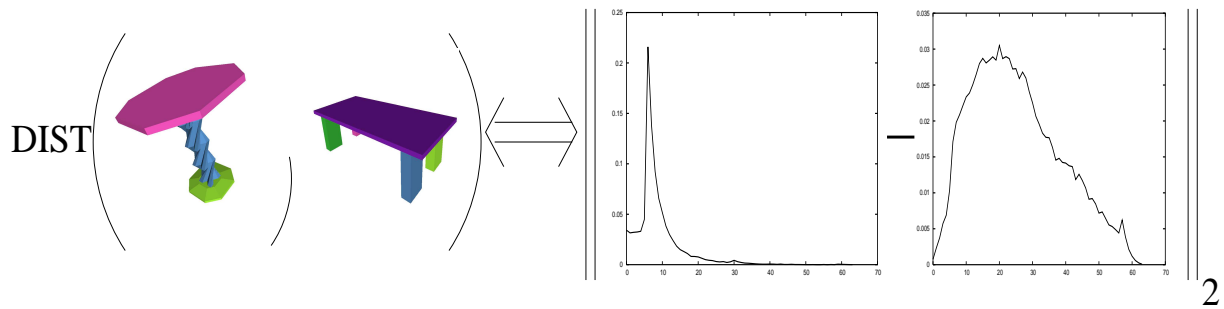


FIG. 8 – Descripteurs de Formes : La comparaison de deux modèles revient à comparer leurs descripteurs associés par une simple norme (L_2 dans l'exemple).

ainsi que sa simplicité de mise en oeuvre. La mesure de similarité entre deux objets revient donc à une mesure de similarité entre deux probabilités de distribution.

Dans la version initiale proposée par Osada, la fonction de forme utilisée mesure la distance entre deux points positionnés de manière aléatoire sur la surface de l'objet. Après calcul, le descripteur est stocké sous la forme d'un vecteur de réels à n éléments ($n=64$).

Expérimentalement, on remarque que la méthode est peu sensible aux changements de tessellation (Figure 9).

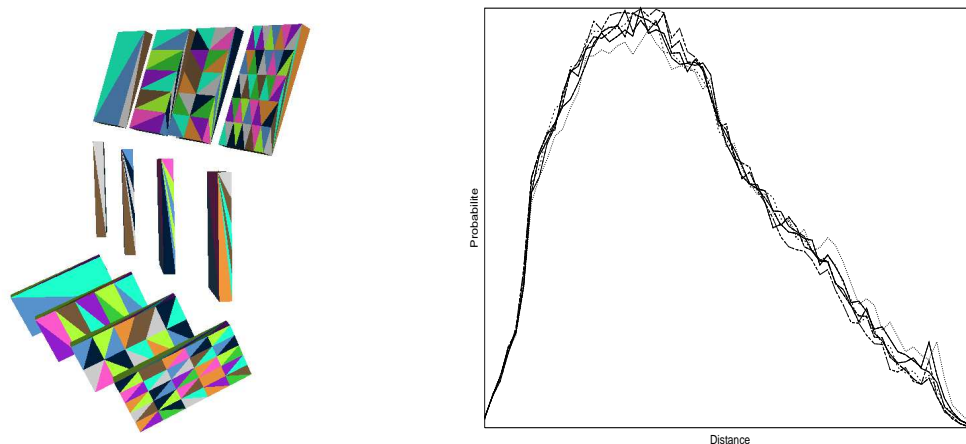


FIG. 9 – Un modèle 3D orienté et maillé de façon différente possède des *Shape Distributions* peu différentes l'une de l'autre. (nombre d'échantillons=80 000)

Nous utilisons les *Shape Distribution* afin de reconnaître les briques identiques, *i.e.* possédant des signatures identiques. Afin d'effectuer cette reconnaissance de manière robuste, on doit réduire la variance de notre échantillonnage. Pour réduire cette variance tout en gardant un nombre d'échantillons raisonnable, on utilise un échantillonnage stratifié, *i.e.* on échantillonne de manière uniforme sur le domaine, puis on perturbe légèrement chaque échantillon[Shi91].

Au lieu d'échantillonner le maillage de la brique, on choisit nos points de la façon suivante :

- On positionne n points sur la sphère englobante de l'objet
- On lance $\frac{n(n-1)}{2}$ rayons, en reliant 2 à 2 les points, et on stocke pour chaque rayon la distance qu'il parcourt à l'intérieur du modèle,
- On construit notre descripteur comme l'histogramme de ces valeurs.

L'impact de la stratification sur la variance de l'échantillonnage est illustrée Figure 10.

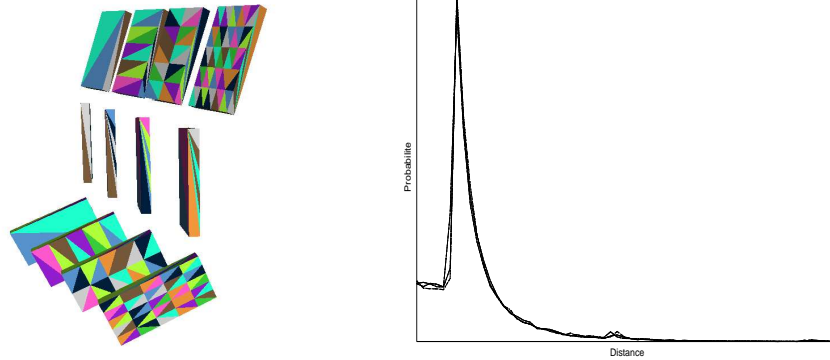


FIG. 10 – La stratification permet de réduire la variance d'échantillonnage. Les échantillons sont positionnés sur la sphère englobante de chaque brique et on stocke pour chaque couple de points la distance parcourue à l'intérieur du modèle par le rayon les reliant.(nombre d'échantillons=80 000)

3 Recherche de structures géométriques répétées

Le but de cette section est de présenter la méthode utilisée pour construire des structures de haut niveau à partir des composantes de base décrites précédemment. On désire retrouver les structures identiques dans la scène composées de plusieurs briques. Avant de déterminer ces structures, on doit « classifier » les briques en fonction de leur ressemblance géométrique.

Classification des briques : A l'aide des descripteurs associés à chacune des briques de la scène, on procède à une classification des briques. On doit pour cela définir un seuil de distance ϵ_D tel que pour une brique B :

$$B \in C_k \iff \{\exists B_k \in C_k / d_D(B_k, B) \leq \epsilon_D\}$$

ou C_k représente la k ième catégorie et d_D la distance entre deux « Shape Distributions ». Dans la pratique, d_D est la norme L_2 et ϵ_D est fixé à 0.1 (valeur sans unité).

Recherche de structures répétées : On associe à chaque brique un point dans un espace à sept dimensions, définis de la façon suivante :

- Quatre dimensions pour un quaternion stockant la rotation qui amène la brique dans le repère du monde. On note $Q(B)$ ce quaternion.
- Trois dimensions pour le vecteur qui translate le centre de masse de la brique sur l'origine du repère du monde. On note $T(B)$ ce vecteur.

La recherche de structures redondantes dans la scène consiste à rechercher des structures se répétant à partir d'une brique de départ. A partir d'une brique B , on recherche ses voisins au sens de la distance euclidienne[Qui94]. On calcule les transformations géométriques relatives entre chaque couple de briques formé de la brique B et de l'un de ses voisins, ces transformations étant stockées sous la forme d'un couple quaternion/translation. Pour rechercher les éventuelles répétitions de cette structure dans la scène, on prend comme point de départ les briques appartenant à la même classe que la brique B . En combinant les transformations géométriques entre la structure recherchée et la nouvelle brique de départ, on positionne un point P_C dans l'espace à sept dimensions définie précédemment. Chercher la répétition de la structure géométrique dans la scène revient donc à chercher les plus proche voisins du point P_C .

L'algorithme utilisé pour cette recherche est présenté à la fin de cette section.

Cet algorithme est appelé de proche en proche afin de rechercher les structures récurrentes. On effectue la recherche dans l'espace à 7 dimensions à l'aide d'un KDTree. Cette structure de recherche doit être capable de gérer les différentes incertitude sur les axes de rotation du modèle. Celles-ci peuvent être de trois types :

1. Incertitude classique de $\pm\pi$ sur chacun des axes. Cette incertitude est liée au fait que le calcul des moments nous permet de connaître l'orientation des axes principaux et non pas leur direction.
2. Incertitude dans le plan. Cette incertitude apparait lorsque les moments suivants deux axes principaux sont identiques (cas du cylindre).

Algorithme 1 Recherche de structures géométriques répétées à partir d'une brique B .

pour toutes les briques B' telle que $d(B, B') \leq \varepsilon$ **faire**
Le couple quaternion/translation recherché est $(Q_R = Q(B) \cdot Q(B')^{-1}, T_R = T(B') - T(B))$
pour Pour toutes les briques B'' telle que $Classe(B'') = Classe(B)$ ($B \neq B''$) **faire**
Le couple quaternion/translation recherché est $(Q_{R'} = Q(B'')^{-1} \cdot Q_R, T_{R'} = T_R + T(B''))$
pour tous les proches voisins P_V du point $P_C = (Q_{R'}, T_{R'})$ **faire**
Soit B_V la brique associée au point P_V
si $Classe(B_V) = Classe(B')$ **alors**
On a trouvé une structure redondante : (B, B') et (B'', B_V)
fin si
fin pour
fin pour
fin pour

3. Incertitude totale (cas d'une sphère, ou d'une forme plus complexe possédant des moments d'ordre 4 constant) .

4 Résultats

On présente dans cette section quelques exemples de structures obtenus à partir de scènes non organisées.

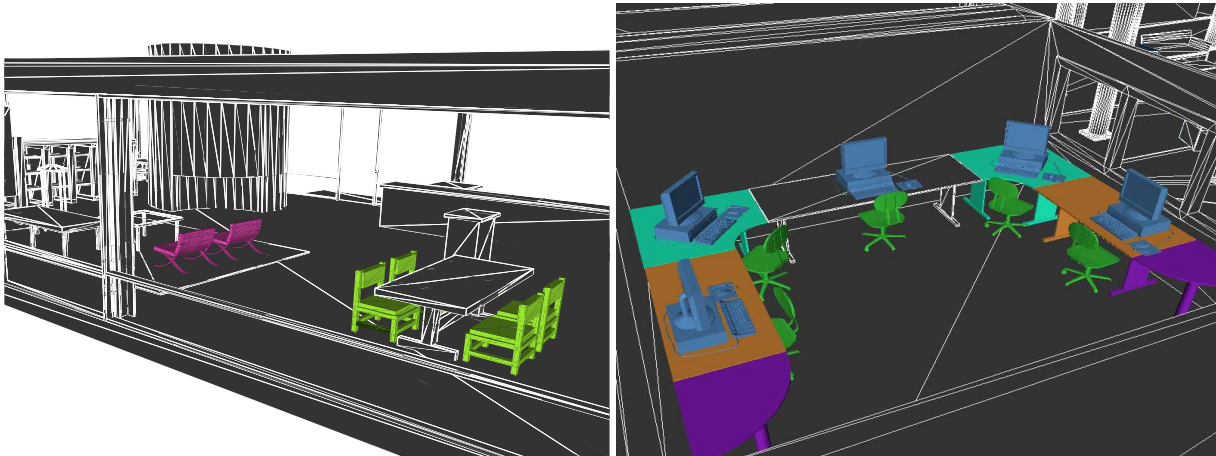


FIG. 11 – Exemple d'organisation partielle de scènes 3D. Ces structures sont créées à partir d'une brique de base, sélectionnée par l'utilisateur. Chaque structure obtenue est repérée à l'aide de sa couleur

Dans les deux images présentées, les structures identifiées sont représentées à l'aide de couleurs identiques, les éléments non traités étant représentés en fil de fer. Dans l'image de gauche, deux types de structures répétées ont été isolés : Les quatre chaises positionnées des deux côtés de la table, ainsi que les deux chaises « design ». Dans l'image de droite, les structures sont plus nombreuses et plus complexes : quatre ordinateurs (clavier+souris+écran), cinq chaises de bureau ainsi que 3 couples de bureau. On remarque, en particulier, que le positionnement légèrement différent de chaque ordinateur sur chacune des tables est pris en compte, ce qui permet d'isoler la structure d'ordinateur du reste du bureau.

5 Conclusion et travaux futurs

Dans cet article, nous avons proposé une méthode permettant de mettre en évidence les structures répétées d'une scène 3D. Ces structures sont créées à partir de composantes de base, les briques, définies comme des ensembles de

polygones connexes par arêtes. En assignant à chaque brique un point, positionné à l'aide d'informations d'orientation et de position, on montre que la recherche de structure répétées d'un scène revient à chercher des voisins dans un espace de dimension élevé.

Plusieurs pistes sont envisageables dans le futur. Dans un premier temps, nous souhaitons mettre en place des applications de cette méthode. Nous pensons, en particulier, que la redondance des structures géométriques peut être une information riche dans le cadre de la modélisation interactive. Nous souhaitons également mettre à profit cette structure de scène en utilisant des techniques de *Geometry Instancing*, permettant un traitement par lots d'objets identiques. Dans la suite de notre travail, nous souhaitons également utilisé des techniques d'apprentissage afin d'extraire des informations sémantiques de la scène, ainsi que des techniques de simulation physique afin d'étudier, par exemple, l'impact de la gravité sur la structure de scène.

Enfin, nous souhaitons diversifier le type de données d'entrées (scènes végétales, urbaines, etc...) afin d'élargir le champ d'application de notre méthode.

A Calcul des moments d'ordres pairs à l'aide d'harmoniques sphériques

Les harmoniques sphériques sont des fonctions Y_l^m à deux variables formant une base orthonormée de fonctions directionnelles et se prêtant donc naturellement à la représentation de telles fonctions. D'après la théorie des harmoniques sphériques, toutes fonctions directionnelles $f(\theta, \phi)$ peut être décomposée de manière unique comme la somme de ses harmoniques :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^m c_l^m Y_l^m(\theta, \phi)$$

Les harmoniques sphériques possèdent de nombreuses propriétés et sont énoncées pour la plupart dans l'ouvrage de référence de Hobson[Hob31]. Dans la suite de ce document, on travaille sur des harmoniques réelles, notées y_l^m .

Nous présentons une méthode permettant de calculer les moments d'ordre pair à l'aide d'harmoniques sphériques. Pour plus de clarté, les calculs sont effectués pour $n = 4$, les résultats étant bien sur applicables pour n'importe quelle valeur de n .

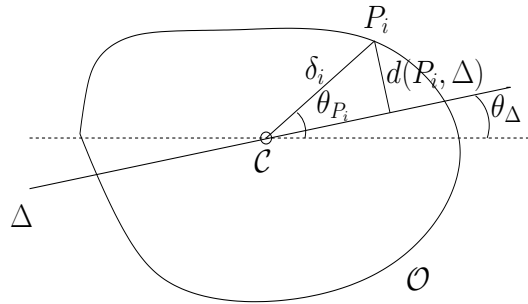


FIG. 12 – Calcul des moments d'une forme O par rapport à une droite Δ .

Le calcul du moment d'ordre 4 de la forme O par rapport à la droite Δ s'écrit :

$$\mathcal{M}_O^4(\Delta) = \frac{1}{\sum w_i} \sum_{i=1}^n w_i * [d(P_i, \Delta)]^4 \quad (\text{A.1})$$

ou $P_i, i \in [1, n]$ sont n points pris sur la surface de O , et w_i sont les poids associés à ces points.

D'après la Figure 12, A.1 peut être réécrite :

$$\mathcal{M}_O^4(\Delta) = \frac{1}{n} \sum_{i=1}^n \delta_i^4 * \sin^4(\theta_{P_i} - \theta_{\Delta}) \quad (\text{A.2})$$

En se plaçant dans un repère local tel que $\theta_{P_i} = 0$, on a :

$$\mathcal{M}_O^4(\Delta) = \frac{1}{n} \sum_{i=1}^n \delta_i^4 * \sin^4(\tilde{\theta}_\Delta) \quad (\text{A.3})$$

L'expression $\sin^4(\theta)$ peut s'écrire sous la forme d'une combinaison linéaire d'harmoniques sphériques de degré ≤ 4 . Après calcul, on obtient :

$$\sin^4(\theta) = \alpha_0 * y_0^0(\theta, \phi) + \alpha_2 * y_2^0(\theta, \phi) + \alpha_4 * y_4^0(\theta, \phi) \quad (\text{A.4})$$

avec :

$$\alpha_0 = \frac{16}{15}\sqrt{\pi}, \alpha_2 = -\frac{32}{21}\sqrt{\frac{\pi}{5}}, \alpha_4 = \frac{8}{35}\sqrt{\frac{4\pi}{9}}$$

Après rotation, une combinaison linéaire d'harmoniques sphériques de degré maximal l , se décompose en une autre combinaison linéaire d'ordre $\leq l$. Dans [Sol98], l'auteur propose un algorithme permettant de calculer de manière efficace cette décomposition après rotation. En notant \mathcal{R}_i la rotation d'angle θ_{P_i} , on a :

$$\sin(\theta_{P_i} - \theta_\Delta) = \mathcal{R}_i(\sin(\tilde{\theta}_\Delta)) \quad (\text{A.5})$$

On en déduit une expression analytique du moment d'ordre 4 de la forme O par rapport à la droite Δ :

$$\mathcal{M}_O^4(\Delta) = \frac{1}{n} \sum_{i=1}^n \delta_i^4 * \mathcal{R}_i(\sin(\tilde{\theta}_\Delta))^4 \quad (\text{A.6})$$

Remerciements

Merci à Sylvain Paris, Xavier Decoret et Pascal Barla pour leurs avis et conseils. Merci à Gilles Debunne pour QGLViewer.

Références

- [CDP95] Frédéric Cazals, George Drettakis, and Claude Puech. Filtering, clustering and hierarchy construction : a new solution for ray-tracing complex scenes. *Computer Graphics Forum*, 14(3) :371–382, 1995.
- [HDS99] Jean-Marc Hasenfratz, Cyrille Domez, François Sillion, and George Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. In *Computer Graphics Forum (Proc. of Eurographics '99)*, number 18(3), pages 221–232. P. Brunet and R. Scopigno, Sep 1999.
- [Hob31] E.W Hobson. *The theory of Spherical and Ellipsoidal harmonics*. Cambridge University Press, 1931.
- [Ju04] Tao Ju. Robust repair of polygonal models. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 04)*, volume 23, pages 888–895, 2004.
- [Kaz04] Michael Kazhdan. *Shape Representations And Algorithms For 3D Model Retrieval*. PhD thesis, Princeton University, 2004.
- [Mar02] Jos'e M. Martinez. *Overview of MPEG-7 Description Tools*, 2002.
- [OFCD01] Osada, Thomas Funkhouser, Bernard Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *International Conference on Shape Modeling and Applications 2001*, pages 154–166, 2001.
- [OOT02] Ryutarou Ohbuchi, Tomo Otagiri, and Tsuyoshi Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Proceedings of Pacific Graphics 02*, pages 265–274, 2002.
- [Qui94] Sean Quinlan. Efficient distance computation between non-convex objects. In *IEEE International Conference on Robotics and Automation*, pages 3324–3329, 1994.
- [Reg02] Ashu Rege. Occlusion (hp and nv extensions). Technical report, NVIDIA, <http://developer.nvidia.com/attach/6715>, 2002.
- [Shi91] Peter Shirley. Discrepancy as a quality measure for sample distributions. In *Proceedings of Eurographics 91*, pages 183–193, 1991.
- [Smi02] Lindsay I. Smith. A tutorial on principal components analysis. Technical report, 2002.
- [Sol98] Cyril Soler. *Représentation hiérarchique de la visibilité pour le contrôle de l'erreur en simulation de l'éclairage*. PhD thesis, Université Joseph Fourier, 1998.
- [XSF02] Ken Xu, James Stewart, and Engene Fiume. Constraint-based automatic placement for scene composition. In *Graphics Interface '02*, 2002.