

Compression de maillages 3D dynamiques par modèle de *skinning*

Khaled Mamou

Titus Zaharia

Françoise Prêteux

Groupe des Ecoles des Télécommunications

Institut National des Télécommunications / Département ARTEMIS

9, rue Charles Fourier 91011 Evry, France

{Khaled.Mamou, Titus.Zaharia, Francoise.Preteux}@int-evry.fr

Résumé

Cet article introduit un nouveau schéma de compression de maillages 3D dynamiques avec topologie constante et géométrie variable. La technique proposée exploite un prédicteur affine par morceaux couplé à un modèle de *skinning* et à une représentation par transformée en cosinus discrète des erreurs résiduelles. Les performances de cette nouvelle approche sont objectivement évaluées sur un corpus de séquences animées de diverses tailles, géométries et topologies et correspondant à des mouvements rigides et élastiques. Les résultats expérimentaux montrent que la méthode proposée offrent des gains importants en termes de débits (de 37% à 67%) par rapport aux approches GV, RT, MPEG-4/AFX, D3DMC, PCA et Dynapack.

Mots clefs

Compression, maillage dynamique, animation 3D.

1 Introduction

Les industries du film d'animation et des jeux vidéo exploitent aujourd'hui à grande échelle les contenus dynamiques 3D.

Ces contenus sont produits en utilisant un large spectre de techniques allant des simulations physiques aux techniques d'acquisition et de clonage de mouvement. Les créateurs des contenus multimédias exploitent le plus souvent les techniques d'animation par modèle de peau (*skinning*) comme celles proposés par le standard MPEG-4/AFX [1] et par les environnements professionnels de modélisation 3D comme 3DS MAX ou Maya.

Pour des raisons de propriété intellectuelle, les créateurs de ces contenus sont souvent réticents à transmettre ces modèles d'animation coûteux et susceptibles de réutilisation illicite et préfèrent une représentation par trames-clés. L'animation est alors stockée sous forme d'une séquence de maillages 3D successifs représentant les trames-clés. Les trames intermédiaires sont obtenues à l'aide de procédures d'interpolation.

La représentation par trames-clés permet de décrire un large spectre d'animations et répond aux problèmes de propriété intellectuelle. De plus, cette représentation est

indépendante de la technique d'animation utilisée pour générer le contenu et permet ainsi de disposer d'un format générique d'animation 3D.

L'inconvénient majeur de cette représentation est en revanche lié aux coûts exorbitants de stockage et de transmission. En effet, même pour de courtes séquences de quelques minutes, des milliers de modèles 3D sont nécessaires. L'élaboration de techniques de compression efficaces et adaptées à ces contenus dynamiques devient alors un enjeu majeur comme en témoigne l'important nombre de travaux de la littérature émergente consacrée à ce sujet (voir [2] pour un état de l'art).

Dans [3], Lengyel introduit pour la première fois le concept de compression de maillages dynamiques. L'auteur propose de représenter l'animation à l'aide d'un ensemble de transformées affines et d'erreurs de prédiction associées. Dans [4], les auteurs proposent une extension de [3], appelée RT (*Rigid Transform*). Le mouvement des sommets du maillage est ici modélisé uniquement par des transformées rigides. La technique D3DMC (*Dynamic 3D Mesh Compression*), proposée dans [5], introduit une approche différente. Le champ de mouvement des sommets est représenté par un ensemble de vecteurs de mouvement associés à une structure volumique d'arbre octal (*octree*). Toutes ces approches nécessitent la mise en oeuvre d'une procédure de segmentation qui vise à regrouper les sommets du maillage en parties pouvant être décrites par un unique modèle de mouvement. Cette procédure de segmentation au sens du mouvement est complexe en temps de calcul et peut induire des discontinuités à bas débit au niveau des frontières entre les *patches*.

Le schéma de compression IC [6] (*Interpolation Compression*) récemment adopté par le standard MPEG-4/AFX [1] exploite une procédure de sous-échantillonnage des trames-clés combinée à une stratégie de prédiction spatio-temporelle locale. L'approche Dynapack [7] propose une approche similaire avec des prédicteurs plus élaborés, appelés *ELP* et *Replica*. Les techniques MPEG-4/AFX-IC et Dynapack présentent l'avantage d'un faible coup de calcul, ce qui les rend particulièrement adaptées aux applications de codage/décodage en temps-réel. Toutefois, le parcours déterministe du maillage utilisé les rend inadaptes pour

des fonctionnalités plus avancées comme la transmission progressive ou le rendu scalable.

Dans [8], Alexa et Müller introduisent une famille différente d’approches, fondée sur une analyse en composantes principales (ACP) du champ de déformation du maillage. Le même principe est repris et étendu dans [9], où un schéma de prédiction supplémentaire est introduit, ainsi que dans [10] où le maillage est préalablement segmenté en parties optimisées pour une représentation par ACP. Les approches de compression par ACP sont spécifiquement adaptées à des séquences longues et répétitives avec un nombre de sommets petit par rapport au nombre de trames. Ces approches sont en revanche très complexes en temps de calcul (cubique avec le nombre de sommets du maillage).

Dans [11], les auteurs proposent un schéma de compression de maillages dynamiques 3D par ondelettes irrégulières construites à l’aide d’une structure de maillage progressif [12]. Le codeur proposé permet d’atteindre des bas débits tout en offrant des fonctionnalités de transmission progressive et de rendu scalable. Cette approche est en revanche inadaptée aux maillages avec un nombre réduit de sommets par composante connexe.

Une représentation différente, dite GV (*Geometry Video*), est proposée dans [13]. Le principe consiste à convertir la géométrie 3D dynamique sous forme d’une séquence d’images 2D. L’approche GV exploite un découpage du maillage (nécessaire pour obtenir une topologie homéomorphe à un disque) et une paramétrisation [14] sur un domaine 2D carré. La topologie initiale du maillage est complètement abandonnée et remplacée par une topologie régulière, obtenue en échantillonnant uniformément le domaine paramétrique. Les images géométriques ainsi construites sont compressées par des techniques traditionnelles de codage d’images 2D. La représentation GV offre des performances de compression compétitives ainsi que des fonctionnalités de transmission progressive et de rendu scalable. En revanche, les objets traités doivent être homéomorphes à un disque ou à une sphère, ce qui restreint sensiblement le domaine d’application de la méthode. De plus, la procédure de remaillage peut conduire à une perte de détails de la surface ainsi qu’à l’apparition d’artefacts visuels.

Dans cet article, nous proposons un nouveau schéma de compression de maillages 3D dynamiques, fondé sur une modélisation du mouvement par des techniques de *skinning*. La contribution majeure de ce travail concerne l’étape de compensation du mouvement qui est formulée comme un problème inverse. A partir d’une représentation arbitraire par trames-clés, un modèle de *skinning* est tout d’abord dérivé et ensuite exploité dans le cadre d’une stratégie de prédiction. Les erreurs résiduelles résultantes sont compressées à l’aide d’une représentation par Transformée en Cosinus Discrète (TCD).

L’approche de compression par modèle de *skinning* proposée étend l’approche RT en introduisant :

- Une nouvelle technique de segmentation au sens du

- mouvement avec un nombre fixe de *patches* qui permet d’éviter une sur-segmentation du maillage dynamique,
- Un modèle de *skinning* permettant d’éviter les discontinuités au niveau des bords des *patches*,
- Un nouveau bloc de codage par TCD des erreurs résiduelles.

Le paragraphe suivant décrit la méthode de compression proposée en détaillant ses principales étapes. Les performances en terme d’efficacité de compression sont ensuite évaluées objectivement, comparées et discutées (paragraphe 3). Enfin, la dernière partie conclut l’article et ouvre les perspectives de recherche future.

2 Compression par *skinning*

Le schéma synoptique de l’algorithme de compression par modèle de *skinning* est présenté Figure 1.

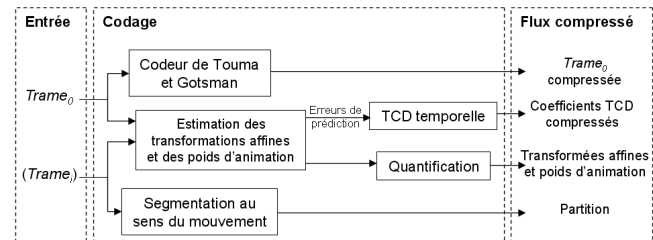


Figure 1 – Schéma synoptique de l’algorithme de compression par modèle de *skinning*.

Tout d’abord, les sommets du maillage sont partitionnés en un ensemble de *patches* de façon à ce que le mouvement de chaque *patch* puisse être décrit par une unique transformée affine 3D. Un modèle de mouvement affine est alors associé à chaque *patch* et pour chaque trame. Enfin, le mouvement de chaque sommet est exprimé comme une combinaison linéaire pondérée des mouvements des différents *patches*.

Le mouvement affine est toujours estimé par rapport à la première trame de l’animation afin (1) de permettre un accès aléatoire aux trames de l’animation et (2) de gérer efficacement les pertes d’information dans le contexte de transmission en environnement bruité. La première trame de la séquence est compressée par un codeur statique. Dans ce travail, nous avons utilisé le codeur classique de Touma et Gotsman [15]. Les erreurs résiduelles de compensation de mouvement sont finalement codées à l’aide d’une représentation par TCD temporelle.

Détaillons à présent les différents blocs du schéma de codage proposé.

2.1 Segmentation au sens du mouvement

L’objectif de l’étape de segmentation est d’obtenir une partition $\pi = (\pi_k)_{k \in \{1, \dots, K\}}$ des sommets du maillage en K parties dont le mouvement peut être fidèlement représenté par des transformées affines 3D. L’approche proposée, détaillée ci-dessous, prend en compte le mouvement de

l'ensemble des sommets du maillage au cours de toute la séquence.

Tout d'abord, une transformée affine A_i^v décrivant le mouvement d'un voisinage local du sommet $v \in \{1, \dots, V\}$ (V étant le nombre des sommets) à la trame $i \in \{0, \dots, F-1\}$ (F étant le nombre de trames) par rapport à la première trame est calculée comme décrit dans l'équation (1) :

$$A_i^v = \arg \min_A \left(\sum_{v \in v^*} \|A\chi_0^v - \chi_i^v\|^2 \right), \quad (1)$$

avec A une matrice 4×4 représentant une transformée affine, v^* le voisinage du sommet v d'ordre trois (*i.e.*, sommets connectés à v par un chemin composé au plus de trois arêtes), et χ_i^v un vecteur 4D représentant les coordonnées homogènes du sommet v à la trame i .

L'ensemble $(A_i^v)_{i \in \{0, \dots, F-1\}}$ est ensuite stocké dans un seul vecteur $\alpha^v \in \mathbb{R}^{12 \times F}$ (une transformée affine étant complètement définie par 12 coefficients réels). Enfin, la partition π est obtenue en appliquant l'algorithme de segmentation *k-means* [16] sur l'ensemble des vecteurs $(\alpha^v)_{v \in \{1, \dots, V\}}$.

La Figure 2 présente les résultats de segmentation obtenus pour les maillages animés "Dance", "Chicken" et "Snake". Le nombre de parties K a été fixé à 20 dans toutes les expérimentations. Cela permet de prendre en compte les différentes parties articulées des objets sans générer de sur-segmentation. Remarquons que pour les maillages dynamiques articulés, l'algorithme arrive à retrouver les parties anatomiques en mouvement. Dans tous les cas, la procédure conduit à des *patches* constitués de sommets topologiquement connexes, même si aucune information topologique n'a été directement prise en compte par l'algorithme *k-means*. Intuitivement, ces résultats montrent que la modélisation affine par morceaux, détaillée dans la section suivante, est bien adaptée pour décrire ces mouvements complexes.

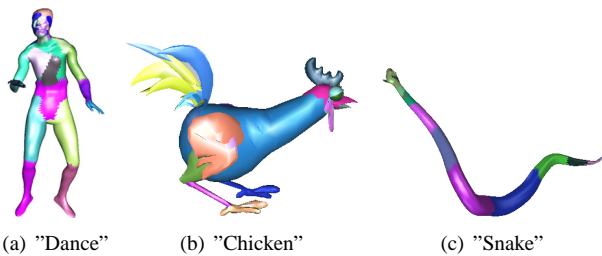


Figure 2 – Segmentation de maillages dynamiques avec différents mouvements, formes et complexités.

2.2 Estimation du mouvement affine

Une fois la partition π déterminée, pour chaque trame i le champ de mouvement est modélisé par K transformées affines, notées $(H_i^k)_{k \in \{1, \dots, K\}}$. Avec les notations de la section précédente, la transformée affine H_i^k associée au *patch* k à la trame i est définie par :

$$H_i^k = \arg \min_A \left(\sum_{v \in \pi_k} \|A\chi_0^v - \chi_i^v\|^2 \right). \quad (2)$$

L'ensemble des transformées affines $(H_i^k)_{k \in \{1, \dots, K\}}$, avec la partition π , fourni un prédicteur affine par morceaux de la trame i à partir de la trame 0 défini comme suit :

$$\forall v \in \{1, \dots, V\}, \quad \widehat{\chi}_i^v = H_i^{k(v)} \chi_0^v, \quad (3)$$

où $k(v)$ désigne le *patch* contenant le sommet v .

Les erreurs de prédiction $e_i^v = (e_i^{v,x}, e_i^{v,y}, e_i^{v,z}, 0)^t$ sont définies par :

$$\forall v \in \{1, \dots, V\}, e_i^v = \chi_i^v - \widehat{\chi}_i^v. \quad (4)$$

La Figure 3 montre la trame originale 36 de la séquence "Snake", sa version prédite et la distribution des erreurs de prédiction correspondantes, exprimées en pourcentage de la diagonale de la boîte englobante de l'objet et représentées en fausses couleurs.

Nous remarquons que le prédicteur proposé modélise efficacement le mouvement des différentes parties du maillage. Ici, l'erreur de prédiction maximale est de l'ordre de 4% de la diagonale de boîte englobante de l'objet. Notons également que les erreurs les plus importantes sont obtenues au niveau des frontières des *patches*. Cela montre les limitations du modèle de mouvement affine par morceaux, qui introduit des discontinuités de bords.

Afin de nous affranchir de cette limitation, nous proposons de raffiner le modèle de mouvement en introduisant un modèle de *skinning* similaire à ceux utilisés dans le domaine de l'animation 3D.

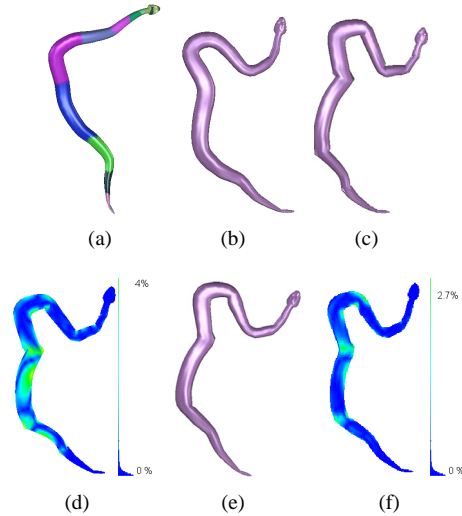


Figure 3 – Prédicteur affine par morceaux vs. modèle de skinning : (a) première trame segmentée, (b) trame 36, (c) trame 36 prédite par transformées affines, (d) distribution des erreurs de (c), (e) trame 36 prédite par modèle de skinning et (f) distribution des erreurs de (e).

2.3 Modèle de *skinning* et estimation des poids d'animation

Le principe de l'animation par modèle de *skinning* est de dériver un champ de mouvement continu sur l'ensemble du maillage en combinant linéairement les mouvements affines des *patches* avec des poids appropriés. La position prédite $\hat{\chi}_i^v$ du sommet v à la trame i s'exprime alors comme :

$$\hat{\chi}_i^v = \sum_{k=1}^K \omega_k^v H_i^k \chi_0^v, \quad (5)$$

où ω_k^v est un coefficient réel qui contrôle l'influence du *patch* k sur le mouvement du sommet v .

L'objectif est de déterminer, pour chaque sommet v , le vecteur des poids optimaux $\omega^v = (\omega_k^v)_{k \in \{1, \dots, K\}}$ défini par :

$$\omega^v = \arg \min_{b \in \mathbb{R}^K} \sum_{i=0}^{F-1} \left\| \sum_{k=1}^K b_k H_i^k \chi_0^v - \chi_i^v \right\|^2. \quad (6)$$

En pratique, il est raisonnable de considérer qu'un *patch* influence uniquement ses propres sommets et ceux appartenant à un *patch* voisin. Soient $k(v)$ le *patch* contenant le sommet v et $\theta(v)$ l'ensemble des *patches* incluant $k(v)$ et ses voisins. L'équation (6) est résolue sous les contraintes suivantes :

$$\forall k \notin \theta(v), \quad b_k = 0. \quad (7)$$

L'équation (6) avec les contraintes (7) conduit à un problème de minimisation au sens des moindres carrés qui peut être résolu par une méthode de pseudo-inverse [17].

Les Figures 3.e et 3.f présentent les résultats obtenus pour la séquence "Snake" en utilisant le modèle de *skinning*. Cette fois, le maillage obtenu par compensation de mouvement est lisse. De plus, les erreurs maximale et moyenne ont été réduites de 35% et 31%, respectivement.

Les erreurs résiduelles ($e_i^{v,x}, e_i^{v,y}, e_i^{v,z}$) sont enfin compressées en utilisant un schéma de codage par TCD temporelle, comme décrit dans le paragraphe suivant.

2.4 Compression par TCD

Pour chaque sommet v du maillage dynamique, nous considérons les erreurs de prédiction associées ($e_i^{v,x}$), ($e_i^{v,y}$), et ($e_i^{v,z}$) ($i \in \{1, \dots, F\}$) comme trois séquences temporelles. Les spectres ($s_i^{v,x}$), ($s_i^{v,y}$), et ($s_i^{v,z}$) de chaque séquence sont alors déterminés en appliquant une TCD monodimensionnelle temporelle.

Cette procédure est appliquée à l'ensemble des sommets du maillage. Les coefficients spectraux obtenus sont enfin multiplexés dans un seul vecteur S de dimension $3 \times V \times F$, défini comme suit :

$$S = \coprod_{i \in \{1, \dots, F\}} \coprod_{v \in \{0, \dots, V-1\}} (s_i^{v,x}, s_i^{v,y}, s_i^{v,z})^t, \quad (8)$$

où \coprod désigne l'opérateur de concaténation.

Le vecteur S ainsi obtenu est finalement codé à l'aide du codeur arithmétique proposé dans [18].

Notons enfin que le processus de concaténation décrit par l'équation (8) rend possible la transmission progressive des coefficients de le TCD à partir des basses fréquences, et ainsi la reconstruction d'une version approchée de l'animation à toute étape du processus de décodage.

3 Résultats expérimentaux

3.1 Corpus de test et critères d'évaluation

Afin de réaliser des comparaisons objectives, nous avons considéré un corpus de test de 4 séquences animées, utilisées par la majorité des travaux de la littérature et appelées "Dance", "Chicken", "Humanoid" et "Snake". Le Tableau 1 résume leurs propriétés exprimées en termes de nombre de sommets (V), de trames (F) et de composantes connexes (CC).

Séquence d'animation	V	F	CC
"Dance"	7061	201	1
"Humanoid"	7646	154	1
"Chicken"	3030	400	41
"Snake"	9179	134	1

Tableau 1 – Propriétés des séquences animées considérées pour l'évaluation.

Les taux de compression sont exprimés en bits par sommet par trame (bpst). Les distorsions de compression sont mesurées en utilisant l'erreur RMSE [19] entre maillages initiaux et reconstruits (décodés). L'erreur RMSE entre deux séquences d'animation est définie comme étant la moyenne des erreurs RMSE sur l'ensemble des trames.

3.2 Résultats de compression

Dans nos expérimentations, les coefficients réels décrivant les transformations affines ainsi que les poids d'animation ont été quantifiés sur 16 bits. La première trame de l'animation a été compressée en utilisant le codeur de Touma et Gotsman [15] avec une quantification sur 12 bits pour la géométrie. Les erreurs de prédiction ont été quantifiées sur 7 bits.

Les résultats obtenus par les approches D3DMC, AFX-IC, RT, PCA, Dynapack et GV sont ceux rapportés dans [5], [4], [7] et [13].

La Figure 4 présente les résultats du codage par modèle de *skinning* et l'approche GV pour la séquence "Dance". La technique de compression par *skinning* montre des gains en débits pouvant atteindre 65% (avec débits de 8 et 2.8 bpst resp. à une RMSE de 0.00025). Les faibles performances de l'approche GV s'expliquent par les distorsions de paramétrisation importantes et le sur-échantillonnage mis en oeuvre par ce codeur.

La Figure 5 présente une comparaison des performances de l'approche par modèle de *skinning* et celles des codeurs AFX-IC et D3DMC, pour la séquence "Humanoid". Notre

technique offre des gains en taux de compression par rapport à D3DMC jusqu'à 67% (avec débits de 3 et 1 bpst resp. à une RMSE de 0.0006). Les performances de l'approche AFX-IC sont de loin les plus faibles. Cela est sans surprise, compte tenu de la simplicité du schéma de prédiction locale utilisé, trop élémentaire pour prendre en compte les corrélations spatio-temporelles.

La Figure 6 compare les performances du codeur par *skinning* aux approches D3DMC, AFX-IC, RT, et Dynapack pour la séquence "Chicken". Le codeur D3DMC et la technique de compression par modèle de *skinning* surclassent les autres techniques, permettant d'atteindre de très bas débits (moins de 4 bpst). Par rapport à D3DMC, le codage par modèle de *skinning* conduit à des gains en débit jusqu'à 37% (avec débits de 3.8 et 2.4 bpst resp. à une RMSE de 0.0013). Cela s'explique par la non-optimalité de la décomposition par structure d'arbre octal considérée par D3DMC.

Enfin, la Figure 7 illustre les performances des approches de codage par *skinning*, par RT et par ACP pour la séquence "Snake". Ici, le codeur par ACP présente les performances les plus faibles. Cela est dû au fait que l'approche par ACP est optimisée pour des animations avec un nombre de trames beaucoup plus important que le nombre de sommets ($F \gg V$), hypothèse qui n'est manifestement pas vérifiée par la séquence "Snake" ($F = 134$, $V = 9179$). Par rapport à la technique RT, notre approche permet d'obtenir des gains en débits jusqu'à 45% (avec débits de 5.3 et 2.9 bpst resp. à une RMSE de 0.00035). Cela prouve que la modélisation du mouvement par un modèle de *skinning* combinée à une TCD temporelle est plus efficace que celle par transformées rigides adoptée par RT.

Ces résultats montrent que l'approche de codage par modèle de *skinning* est particulièrement efficace dans le cas des maillages dynamiques articulés, où la stratégie de compensation de mouvement proposée est bien adaptée. De plus, notre approche permet d'obtenir des séquences reconstruites de haute qualité à partir de très bas débits (1 bpst).

4 Conclusion et perspectives

Dans cet article, nous avons présenté une nouvelle technique de compression de maillages 3D dynamiques, avec topologie fixe et géométrie variable. Le coeur de notre méthode est un modèle de *skinning*, inspiré des techniques de création de contenus 3D, qui permet la mise en oeuvre d'un schéma de prédiction hybride avec compensation de mouvement et représentation en TCD des erreurs résiduelles de prédiction. Les différentes étapes intervenant dans la construction automatique de ce modèle (segmentation du maillage au sens du mouvement, modélisation de mouvement affine par morceaux et estimation optimale des poids d'animation) sont présentées en détails.

Les expérimentations, conduites sur une base de séquences de test traditionnellement utilisées dans la littérature pour l'évaluation des algorithmes de compression, montrent que

l'approche proposée offre des gains en débit considérables (de 37% à 67%) par rapport aux autres techniques de l'état de l'art (GV, D3DMC, RT, Dynapack et MPEG-4/AFX-IC). Le schéma de codage par modèle de *skinning* se révèle particulièrement efficace dans le cas des séquences d'animation présentant des personnages au mouvement articulé. Les perspectives de recherche concernent principalement l'optimisation de l'étape de segmentation au sens du mouvement. En particulier, nous étudierons comment déterminer un nombre optimal de parties, sous contraintes aussi bien anatomiques (e.g. prise en compte d'un squelette d'animation) que géométriques (e.g. erreur de prédiction maximale).

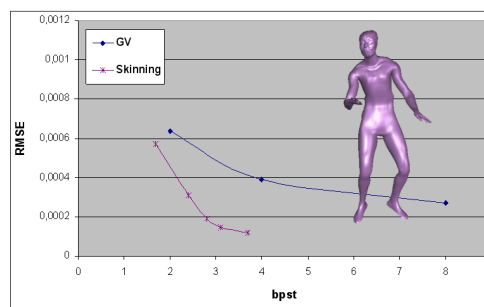


Figure 4 – Modèle de skinning vs. GV pour la séquence "Dance".

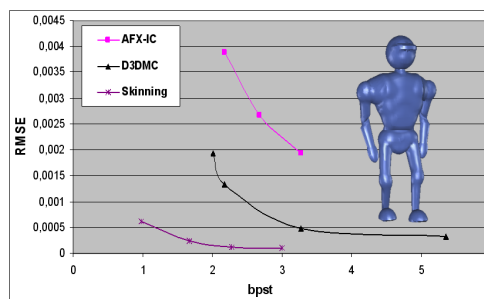


Figure 5 – Modèle de skinning vs. AFX-IC et D3DMC pour la séquence "Humanoid".

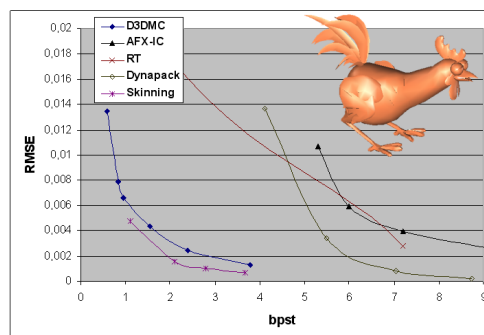


Figure 6 – Modèle de skinning vs. AFX-IC, D3DMC, RT et Dynapack pour la séquence "Chicken".

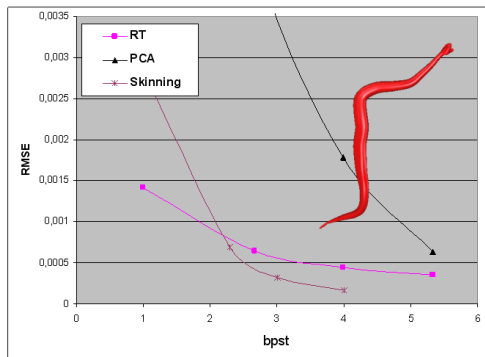


Figure 7 – Modèle de skinning vs. RT et PCA pour la séquence "Snake".

Références

- [1] M. Bourges-Sevenier et E.S. Jang. An introduction to the MPEG-4 animation framework extension. *IEEE transactions on Circuits and Systems for Video Technology*, 14(7) :928–936, Juillet 2004.
- [2] K. Mamou, T. Zaharia, et F. Preteux. A preliminary evaluation of 3D mesh animation coding techniques. Dans *Actes de la conférence SPIE Conference on Mathematical Methods in Pattern and Image Analysis*, pages 44–55, San Diego, États-Unis, Avril 2005.
- [3] J. Lengyel. Compression of time-dependent geometry. Dans *Actes de la conférence ACM Symposium on Interactive 3D Graphics*, pages 89–96, Atlanta, États-Unis, Avril 1999.
- [4] G. Collins et A. Hilton. A rigid transform basis for animation compression and level of detail. Dans *Actes de la conférence Vision, Video, and Graphics*, pages 21–28, Édimbourg, Royaume-Uni, Juillet 2005.
- [5] K. Müller, A. Smolic, M. Kautzner, P. Eisert, et T. Wiegand. Predictive compression of dynamic 3d meshes. Dans *Actes de la conférence IEEE International Conference on Image Processing*, pages 621–624, Genève, Suisse, Septembre 2005.
- [6] E. S. Jang, J. D. K. Kim, S. Y. Jung, M. J. Han, S. O. Woo, et S. J. Lee. Interpolator data compression for MPEG-4 animation. *Circuits and Systems for Video Technology*, 14(7) :989–1008, Juillet 2004.
- [7] L. Ibarria et J. Rossignac. Dynapack : space-time compression of the 3D animations of triangle meshes with fixed connectivity. Dans *Actes de la conférence SIGGRAPH/Eurographics symposium on Computer animation*, pages 126–133, San Diego, États-Unis, Juillet 2003.
- [8] M. Alexa et W. Müller. Representing animations by principal components. *Computer Graphic Forum*, 3(19) :411–418, Août 2000.
- [9] Z. Karni et C. Gotsman. Compression of soft-body animation sequences. *Computers and Graphics*, 28(1) :25–34, 2004.
- [10] M. Sattler, R. Sarlette, et R. Klein. Simple and efficient compression of animation sequences. Dans *Actes de la conférence SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217, Los Angeles, États-Unis, Juillet 2005.
- [11] I. Guskov et A. Khodakovsky. Wavelet compression of parametrically coherent mesh sequences. Dans *Actes de la conférence ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 183–192, Grenoble, France, Août 2004.
- [12] H. Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series) :99–108, Juin 1996.
- [13] H. M. Briceño, P. V. Sander, L. McMillan, S. Gortler, et H. Hoppe. Geometry videos : a new representation for 3D animations. Dans *Actes de la conférence Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 136–146, San Diego, États-Unis, Juillet 2003.
- [14] P. Sander, P. Gortler, J. Snyder, et H. Hoppe. Signal-specialized parameterization. Dans *Actes de la conférence Eurographics workshop on Rendering*, pages 87–98, Pisa, Italie, Juin 2002.
- [15] C. Touma et C. Gotsman. Triangle mesh compression. Dans *Actes de la conférence Graphics Interface*, pages 26–34, Vancouver, Canada, Juin 1998.
- [16] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, et A. Y. Wu. An efficient k-means clustering algorithm : Analysis and implementation. *Pattern Analysis and Machine Intelligence*, 24(7) :881–892, Juillet 2002.
- [17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, et B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge Press, New York, États-Unis, 2nd édition, 1992.
- [18] I. H. Witten, R. M. Neal, et J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6) :520–540, Juin 1997.
- [19] N. Aspert, D. Santa-Cruz, et T. Ebrahimi. MESH : Measuring errors between surfaces using the hausdorff distance. Dans *Actes de la conférence IEEE International Conference in Multimedia and Expo (ICME)*, pages 705–708, Lausanne, Suisse, Août 2002.