

Tatouage Géométrique d'Objets 3D et Récupération 2D

Jihane Bennour¹ et Jean-Luc Dugelay¹

Institut Eurécom,
Département communication multimédia, groupe image vidéo,
2229 route des crêtes B.P 193,
06904 Sophia-Antipolis, France..
Tél : int+ 33 4 93 00 26 24, Fax : int+ 33 4 93 26 27
{bennour, dugelay}@eurecom.fr

Résumé Dans cet article, nous proposons une nouvelle approche pour le tatouage d'objets 3-D qui se base sur l'information de contour apparent. A l'opposé des techniques de tatouage déjà existantes qui extraient le tatouage à partir de l'objet 3-D nous cherchons ici à tatouer un objet 3-D et à extraire la marque des vues 2-D de l'objet dans lesquelles l'information de contour tatoué est présente.

Mots clés Tatouage, objet 3-D, contour, projection 2-D.

1 Introduction

De nos jours il est relativement aisé de créer des objets 3-D et de les utiliser dans de nombreuses applications. L'expansion d'internet, la multiplication des réseaux à haut débit font que les objets 3-D sont de plus en plus présents dans les bases de données distribuées liées au service multimédia. Par conséquent de nouvelles préoccupations apparaissent. On pourrait par exemple vouloir vérifier si l'utilisation d'un objet 3-D est autorisée ou non, ou obtenir des informations supplémentaires sur l'objet (e.g pour l'authentification ou l'indexation), sur le propriétaire (copyright), ou bien encore sur l'acheteur (e.g pour la non répudiation).

Les algorithmes de tatouage déjà existants permettent de protéger un objet dans la mesure où on a accès aux données 3-D tatouées suspectes. Or il s'avère que souvent (par exemple sur des pages Web) il est plus facile de repérer et de récupérer des images 2-D suspectes d'objets 3-D protégés que les fichiers 3-D eux mêmes. Les travaux déjà existants à ce jour sur le tatouage d'objets 3-D basé sur la géométrie ne permettent pas d'extraire le tatouage à partir d'une vue 2-D de l'objet 3-D protégé.

L'approche que nous présentons dans cet article¹ enfouit le tatouage dans le contour apparent de l'objet 3-D. Le tatouage peut alors -sous certaines conditions- être extrait de vues 2-D de l'objet dans lesquelles l'information de contour tatoué est présente.

Ce papier est organisé comme suit : la section 2 présente un court état de l'art des algorithmes de tatouage d'objets 3D. La section 3 décrit notre procédé de tatouage d'objets 3D. La section

¹ Ces travaux ont été réalisés en partie dans le cadre du projet FR RNRT SEMANTIC 3-D (www.semantic-3d.net)

4 discute du compromis classique de tatouage : capacité, visibilité et robustesse dans le cas précis de notre approche et des objets 3-D. On conclut dans la section 5 en précisant nos perspectives.

2 Etat de l'art

Peu de travaux existent à ce jour sur le tatouage d'objets 3-D (classés parmi les nouveaux objets). Dans le domaine du multimédia un objet 3-D est le plus souvent représenté par un maillage associé à une texture. Les présentations sous formes de Nurbs, arbres CSG et nuages de points sont moins fréquentes. La majorité des algorithmes de tatouage déjà publiés se basent sur une modification du maillage. [1] permet de coder de l'information supplémentaire dans un maillage décrit par une liste de sommets et une liste de triangles. Pour cela il modifie l'ordre des sommets, l'ordre des triangles, ou l'ordre des sommets d'un triangle dans le fichier qui décrit l'objet. [2] se base sur la position des sommets. A chacun des sommets est associé un voisinage ellipsoïdal. Pour coder un 0, un sommet est déplacé à l'extérieur de son voisinage et pour coder un 1 il est déplacé à l'intérieur. D'autres algorithmes se basent sur la répartition des normales [3], [4], sur la taille des normales [5] ou même sur les invariants géométriques comme le rapport des longueurs de deux côtés ou le rapport de la hauteur sur la longueur de la base [6]. On peut aussi citer l'algorithme de tatouage de Ohbuchi, Masuda et Aono [7], [8] qui enfouit la marque dans un objet 3-D présenté par des NURBS ou l'algorithme de Fornaro et Sanna [9] qui dissimule de l'information dans un modèle 3-D décrit par un arbre de construction CSG (Constructive Solid Graph).

Tous ces algorithmes de tatouage d'objets 3-D ne permettent pas d'extraire le tatouage à partir d'une vue 2-D de l'objet 3-D protégé. Dans ce contexte bien particulier on peut citer [10], [11] qui enfouit le tatouage dans l'information de texture de l'objet 3-D. Le tatouage est par la suite extrait des vues 2-D de l'objet dans lesquelles l'information de texture tatouée est présente. Cet algorithme suppose qu'il s'agit d'un objet 3-D réaliste ou du moins riche en texture, contrairement aux objets purement géométriques comme on le suppose ici.

3 Description de notre procédé de tatouage d'objets 3-D

Cette section est dédiée à la présentation de notre procédé de tatouage d'objets 3D. On commencera par présenter l'approche dans sa globalité, on détaillera en suite les deux étapes principales : extraction (§ 3.2) et tatouage du contour apparent (§ 3.3).

3.1 Principe général

La figure 1 donne un aperçu général de notre algorithme de tatouage d'objets 3-D. Les étapes 1-3 concernent l'insertion ; les étapes 4-6 la détection.

Etant donné un objet 3-D représenté par des maillages on cherche à tatouer cet objet et extraire la marque des vues 2-D dans lesquelles l'information de tatouage est présente. On peut s'appuyer sur le contour apparent de l'objet 3-D pour enfouir la marque. La première difficulté à laquelle on est confronté est liée au recalage projectif entre la vue 2-D et l'objet

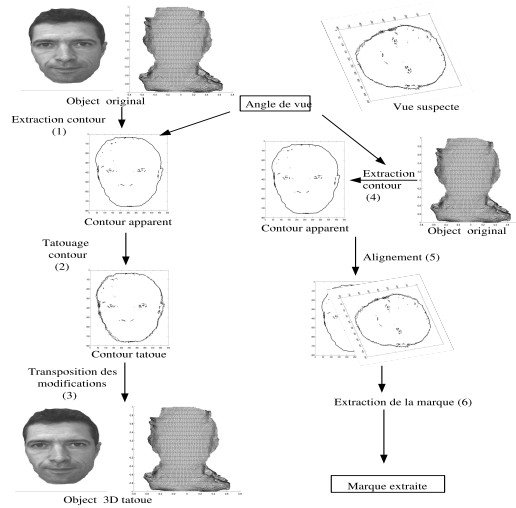


Fig. 1. Principe général de l'algorithme de tatouage.

3-D afin d'extraire la marque. L'idée est donc de tatouer, en première approche, l'objet 3-D connaissant la ou les vues de projection utilisées (pour le marquage et donc à reconsidérer pour l'extraction). Ces vues de projection peuvent rester secrètes.

Insertion de la marque :

- *Etape 1* : On fixe k angles de vue ($k > 0$) et on extrait le contour apparent de l'objet 3-D correspondant à cet/ces angle(s) de vue(s) (voir § 3.2).
- *Etape 2* : On tatoue le contour apparent extrait (voir § 3.3).
- *Etape 3* : On répercute les modifications sur l'objet 3-D.

Détection de la marque :

- *Etape 4* : Connaissant la/les projection(s) utilisée(s) on extrait le contour apparent correspondant à ces vues (voir § 3.2).
- *Etape 5.1* : On recale les deux vues 2-D (projection de l'objet 3-D et la vue 2-D repérée).
- *Etape 5.2* : Pour extraire la marque, on choisit la vue la plus adéquate parmi les k vues tatouées. On appelle une vue adéquate celle qui couvre le plus la vue suspecte.
- *Etape 6* : On extrait la marque avec l'algorithme de tatouage de contour (voir § 3.3).

Nos premières simulations, on s'intéresse à des visages 3-D avec une valeur de $k = 3$: vue de face, vue de profile et une vue intermédiaire (fig. 2). On a estimé que ces vues sont les plus significatives pour représenter un visage 3D.

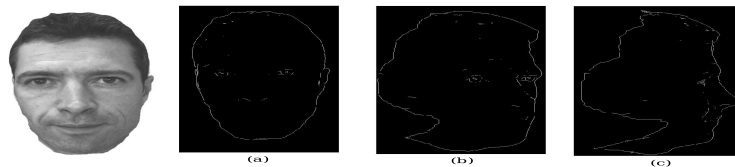


Fig. 2. Objet 3-D, vue de face (a), vue intermédiaire (b) et vue de profile (c).

3.2 Extraction de contour apparent

Pour déterminer le contour d'un objet 3-D, on parcourt toutes ses arêtes et

1. Soit l'arête courante n'est associée qu'à une face dans ce cas c'est un contour.
2. Soit elle est associée à deux faces F1 et F2 et dans ce cas :
 - Soit \vec{n}_1 et \vec{n}_2 les normales de F1 et F2.
 - Soit \vec{v} un vecteur formé par la position de la caméra et un point de l'arête en question.
 Si $(\vec{n}_1 \cdot \vec{v}) \cdot (\vec{n}_2 \cdot \vec{v}) < 0$ c.à.d F1 et F2 sont orientées l'une plutôt de face l'autre plutôt de dos par rapport à la caméra donc que l'arête courante est un contour.
Sinon ce n'est pas une arête du contour.

3.3 Tatouer un contour apparent

Peu de travaux existent à ce jour sur le tatouage de contour [12], [13]. En ce qui nous concerne nous avons opté pour l'algorithme développé dans l'article [12]. Outre le fait que cet algorithme est adapté à notre approche il est également facile à mettre en œuvre. Nous décrivons brièvement le principe de cet algorithme. On note :

- N le nombre de sommets et $[x(n) \ y(n)]$ les coordonnées des points du contour L.
- $z(n) = x(n) + iy(n)$ et $Z(k) = \sum_{n=1}^N (z(n) \cdot \exp((-2\pi jkn/N))$.
- W_0 est une séquence de +/-1 de moyenne 0 et de variance 1.

Insertion de la marque :

$$W(i) = \begin{cases} 0 & i < aN \text{ ou } bN < i < (1-b)N \text{ ou } (1-a)N < i. \\ W_0 & aN < i < bN \text{ ou } (1-b)N < i < (1-a)N. \end{cases}$$

- N est le nombre de coefficients de Fourier et p la force de marquage ($p < 1$).
- a et b contrôlent les faibles et hautes fréquences qu'affectent le marquage. $0 < a < b < 0.5$.

La transformée de fourier (TF) du contour tatoué est alors :

$$|Z'(k)| = |Z(k)| + p|Z(k)| \cdot W(k)$$

Une Transformée de Fourier Inverse (TFI) donne la ligne marquée L'.

Détection de la marque : Soit $Z'(k)$ la transformée de fourier de la ligne marquée L'. Le calcul de la corrélation c entre la marque W et $Z'(k)$ permet de détecter la présence ou non de la marque.

$$c = \sum (W(k) \cdot |Z'(k)|)$$

À la place de c on utilise une valeur normalisée $c' = c/moy(c)$. La règle de tatouage peut avoir la forme :

- H_0 : W tatoue L' si $c' > T$.
- H_1 : W ne tatoue pas L' si $c' < T$.

Plusieurs expériences ont été menées afin de mesurer le potentiel et les limites de cette approche. Le contour utilisé est celui de la vue de face (voir fig 2.a) avec $N = 2383$ sommets. Pour différentes valeurs de a, b, p et dans un contexte avec ou sans attaques de changement de repère la valeur du coefficient de corrélation normalisé c' est toujours supérieur à 0.6 (la moyenne de c' est égal à 1 si W tatoue L). Les résultats sont présentés sur la figure 3.

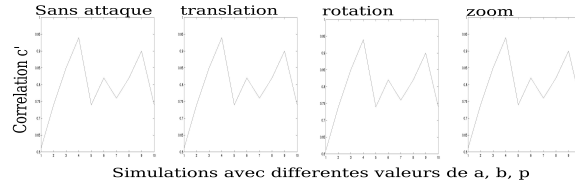


Fig. 3. Détection de la marque : contexte sans attaque et avec des attaques de changement de repère.

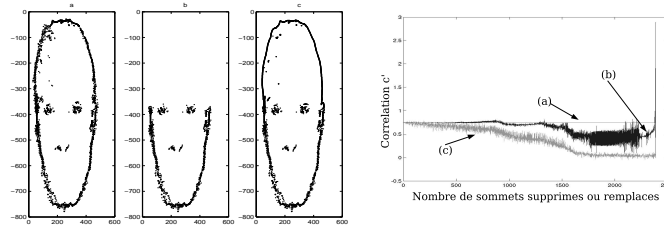


Fig. 4. a. Contour sans attaque b. Contour avec fragments supprimés c. Contour avec fragments remplacés

Comme dans certains cas, la vue suspecte correspond partiellement à celle tatouée, nous avons testé la robustesse de cette algorithmes à ce type de situation.

1. Suppression de quelques fragments du contour tatoué (voir fig.4.b).
2. Remplacement de fragments de contour tatoué par d'autres qui ne le sont pas (voir fig.4.c).

Les résultats sont présentés sur la figure 4 . On remarque que si on remplace moins de 500 sommets (parmi 2383) ou si on supprime moins de 1200 sommets (parmi 2383) on arrive toujours à détecter la présence de la marque.

4 Compromis classique : Robustesse, visibilité et capacité

Dans tous les domaines d'application du tatouage numérique, un algorithme de tatouage doit satisfaire au mieux trois contraintes deux à deux antagonistes : la capacité, la visibilité, et la robustesse.

Contrairement à la quasi totalité des algorithmes de tatouage d'objets 3-D, notre approche est robuste aux remaillages et à la compression étant donné qu'on enfouit la marque dans le contour apparent. La robustesse aux transformations géométriques : translation, rotation, mise à l'échelle ... est étroitement lié à l'algorithme de tatouage de contour. L'algorithme utilisé dans notre cas est robuste à ce type de transformations (voir fig. 3) mais il est sans capacité. Notre algorithme hérite de ces propriétés. La visibilité du tatouage dans la forme de l'objet 3-D est liée d'une part à l'algorithme de tatouage de contour utilisé et d'autre part aux paramètres de la projection 3-D/2-D (éloignement de l'objet par rapport à la caméra ...). Dans le cas des visages 3-D évoqués dans la section 3.1, la distorsion due au tatouage du contour de profil (fig. 2-c) est assez contraignante si on considère par exemple le rôle de cette courbe dans la reconnaissance de personnes en biométrie.

Dans un travail futur on envisage développer notre propre algorithme de tatouage de contour qui soit le mieux adapté à notre approche.

5 Conclusion et perspectives

Nous avons présenté dans cet article une nouvelle approche pour le tatouage d'objets 3-D qui se base sur l'insertion de la marque dans le contour apparent et son extraction des vues 2-D de l'objet dans lesquelles l'information de contour tatoué est présente. L'intérêt de cette approche est que souvent il est plus facile de repérer ou de récupérer des vues 2-D suspectes d'objets 3-D que les fichiers 3-D eux mêmes.

Il nous reste à mesurer les performances et les limites de notre approche ; plus précisément à évaluer le compromis classique capacité robustesse visibilité dans notre cas (approche 3-D/2-D) et de le comparer aux autres algorithmes de tatouage d'objets 3-D (approche 3-D/3-D). Le travail futur se concentrera aussi sur le tatouage aveugle d'un objet 3-D : la marque serait récupérée des vues 2-D de l'objet 3-D sans avoir une connaissance préalable de la projection utilisée.

Références

1. S. Ichikawa, H. Chiyama, and K. Akabane, "Redundancy in 3D polygon models and its application to digital signature," *Journal of WSCG*, vol. 10, no. 1, pp. 225–232, 2002.
2. T. Harte and A. G. Bors, "Watermarking 3D models," in *IEEE International Conference on Image Processing*, Rochester, NY, USA, Sept. 2002.
3. O. Benedens, "Watermarking of 3D polygon based models with robustness against mesh simplification," in *SPIE Security and Watermarking of Multimedia Content*, 1999, pp. 329–340.
4. —, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46–55, 1999.
5. M. G. Wagner, "Robust watermarking of polygonal meshes," in *Geometric Modeling and Processing*, Hong Kong, China, Apr. 2000.
6. R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models," in *ACM Multimedia*, Seattle, Washington, Nov. 1997, pp. 261–272.
7. —, "A shape-preserving data embedding algorithm for NURBS curves and surfaces," in *Computer Graphics International*, June 1999, pp. 170–177.
8. R. Ohbuchi and H. Masuda, "Managing cad data as a multimedia data type using digital watermarking."
9. C. Fornaro and A. Sanna, "Public key watermarking for authentication of CSG models," *Computer Aided Design*, vol. 32, no. 12, pp. 727–735, 2000.
10. J.-L. Dugelay, E. Garcia, and C. Mallauran, "Protection of 3D object usage through texture watermarking," in *European Signal Processing Conference (EUSIPCO)*, Toulouse, France, Sept. 2002.
11. E. Garcia and J.-L. Dugelay, "Watermark recovery from 2D views of a 3D video object," in *SPIE Electronic Imaging*, Santa Clara, California, Jan. 2003.
12. V. Solachidis and I. Pitas, "Watermarking polygonal lines using fourier descriptors," in *IEEE Computer Society*, 2004.
13. H. Sonnet, T. Isenberg, J. Dittmann, and T. Strothotte, "Illustration watermarks for vector graphics," in *IEEE Pacific Conference on Computer Graphics and Applications*, 2003.