Tatouage fragile d'objets 3D triangulés

François CAYRE, Francis SCHMITT, Henri MAÎTRE

Dépt. TSI / URA CNRS 820
École Nationale Supérieure des Télécommunications
46, rue Barrault
75634 Paris cedex 13, France
Tel : 01 45 81 73 27 - Fax : 01 45 81 37 94
cayre@tsi.enst.fr, (schmitt|maitre)@enst.fr

Résumé – Nous proposons une nouvelle méthode de tatouage fragile adaptée au marquage des objets maillés tridimensionnels et triangularisés, basée sur la modification d'invariants géométriques. Nous utilisons une localisation compacte indexée de l'information cachée, et nous vérifions la robustesse du schéma global de tatouage face à l'attaque de coupe.

Abstract – We propose a new method of fragile watermarking designed for the authentication of 3D triangle meshes, which is based upon the modification of geometrical invariants. We use an indexed compact localization for the hidden information. Finally, we test our global watermarking scheme method against the cut (or crop) attack of the mesh.

1 Introduction

Les maillages tridimensionnels connaissent d'ores et déjà un important développement de leurs applications dans de nombreux domaines : biomédical, art, cartographie, entre autres. Ces données, parfois sensibles, nécessitent que l'on puisse vérifier si elles n'ont pas subi de modifications, peut-être malveillantes ou au moins suspectes : c'est le problème de l'authentification. Si la marque que l'on s'attend à retrouver n'est pas présente, on en déduit que le maillage a été corrompu. Nous limitons notre étude aux maillages triangulés car il s'agit du plus petit dénominateur commun en matière de représentation polyédrique.

Nous proposons dans ce travail une méthode de tatouage fragile destinée spécialement à ce problème. En particulier, nous souhaitons autoriser certaines modifications, jugées non suspectes : rotation, translation, mise à l'échelle uniforme et coupe. Toutes les autres modifications attaqueront le tatouage. Dans la pratique, nous fixons la capacité à $N = 64$ bits (comme requis par la plupart des systèmes de gestion des droits [3]), et nous souhaitons disposer d'une clef secrète K pour l'accès à l'information cachée, elle sera codée sur N_K bits.

Lorsque l'on tatoue une image, l'information est cachée suivant deux types d'arrangements : global ou local. Dans le premier cas, l'ordre et le numérotage des bits cachés sont implicites : ils utilisent une référence absolue à un pixel de l'image originale (en général le pixel supérieur gauche dans le cas de la stéganographie LSB). Dans le second, on crée un repère propre à la marque, soit par insertion d'un tatouage annexe de resynchronisation (comme par exemple dans le greffon de Digimarc dans PhotoShop), soit par répétition d'un motif dans les méthodes modernes (on utilise les pics d'autocorrélation pour se resynchroniser avant l'estimation).

Pour les maillages également, seul un arrangement local permettrait de s'affranchir de l'attaque de coupe. Mais la resynchronisation de descriptions surfaciques est complexe. On a alors recours à un troisième type d'arrangement de l'information cachée, dit indexé. Le principe d'un tel arrangement, applicable grâce à la connexité quelconque du maillage, est de cacher non seulement la valeur des bits de la marque à insérer, mais aussi leur numéro. On définit pour cela des configurations géométriques composées de triangles voisins : l'un code le numéro du bit caché, les autres sa valeur. Notre méthode introduit une configuration géométrique composée d'un seul triangle, elle est donc plus compacte que les précédentes, qui en utilisent toutes entre deux et quatre [8].

2 Insertion

Nous précisons la construction de notre méthode au fur et à mesure que nous décrivons le module d'insertion. Nous commençons par proposer un arrangement indexé compact de la marque ne nécessitant qu'un seul triangle pour coder l'information cachée. Nous en dégageons une stratégie de codage par deux invariants géométriques n'interférant pas entre eux. Enfin, nous précisons l'exploitation faite de ces deux invariants.

2.1 Un arrangement indexé compact

L'originalité de notre méthode consiste en l'utilisation d'un seul triangle pour coder toute l'information relative à un bit caché : son numéro et sa valeur. Un triangle ABC codant un bit d'information cachée s'appuie sur sa base de référence : son côté le plus petit (par convention), qui sera noté AB. Toutes les modifications propres à l'insertion de la marque se font dans le plan du triangle ABC. Ainsi, pour coder le numéro des bits cachés on va déplacer le point C perpendiculairement à (AB), alors que pour coder la valeur de ce même bit, on déplacera C parallèlement à (AB).

Afin d'éviter de modifier un triangle déjà porteur d'information cachée, les trois points le constituant sont marqués comme

étant interdits pour une modification ultérieure. Pour trouver les triangles cachant l'information, on parcourt tous les triangles un à un. Un triangle est déclaré valide pour l'insertion si son point C n'est pas marqué comme interdit. L'insertion prend fin après avoir inséré la marque le nombre désiré de fois. On donne sur la Fig. 1 une idée de la densité du remplissage obtenu.

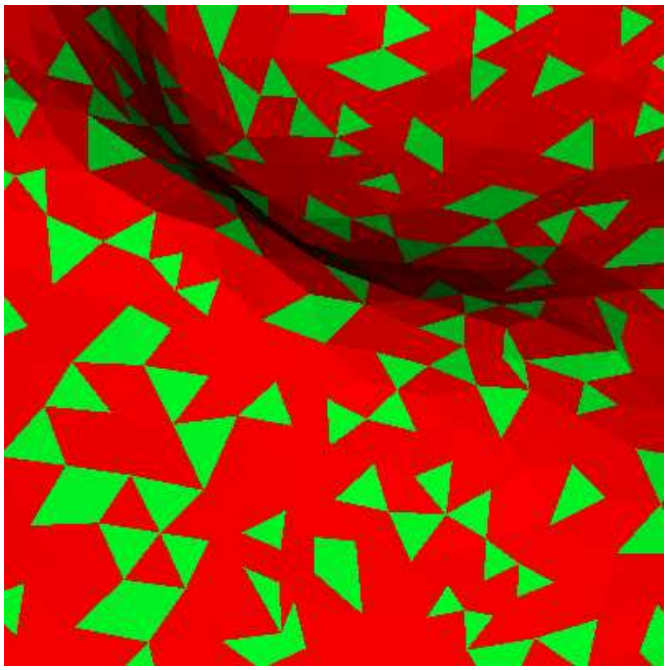

FIG. 1: Localisation (zoom) de l'information cachée (en clair), répartie aléatoirement sur le maillage (en sombre). Maillage : Carlos Hernandez, Télécom Paris. On remarque que rien n'empêche que deux triangles porteurs d'information cachée puissent partager un côté.

2.2 Contraintes et invariants

Nous souhaitons autoriser que l'objet subisse des modifications, jugées acceptables, et qui sont la translation, la rotation, la mise à l'échelle uniforme, et la coupe. La robustesse face à la coupe sera assurée par la redondance d'une part et la localisation indexée de l'information cachée. Par contre, notre espace de tatouage devra être résistant en translation, rotation, et mise à l'échelle uniforme. Notre choix s'oriente alors vers la sélection de deux invariants géométriques : l'un codera le numéro du bit, et l'autre sa valeur. Pour pouvoir être valables, les deux invariants devront agir l'un perpendiculairement, et l'autre parallèlement à (AB).

Parmi un certain nombre d'invariants [4], nous détaillons les deux que nous avons retenus :

- Pour coder le numéro du bit, nous utilisons le rapport des aires de deux polygones : le triangle ABC et le carré imaginaire ayant AB pour côté. La modification de cet invariant peut se faire en déplaçant C perpendiculairement à (AB).
- Pour coder la valeur du bit, nous nous servons du rapport de deux longueurs de segments situés sur la même droite. Ces longueurs sont celles de la projection de C sur (AB), et celle du segment $[AB]$. La modification de cet invariant peut se faire en déplaçant C parallèlement à (AB).

Ce deuxième invariant est en pratique le fondement d'une procédure géométrique substitutive illustrée en Fig. 2. Le premier invariant, en plus de satisfaire aux contraintes, s'étend aux transformations affines. C'est le second invariant qui limitera la robustesse du premier : même si le numéro des bits peut rester lisible dans le cas d'une transformation affine, c'est leur valeur qui aura d'abord disparu.

2.3 Codage du numéro du bit

Soit S l'aire d'un triangle valide pour l'insertion. On définit la grandeur :

$$g_{tri} = \left(T + \frac{K}{2^{N_K}}\right)^{\alpha} \times \frac{AB^2}{S} \quad (1)$$

Le nombre $\left(T + \frac{K}{2^{N_K}}\right)$ sert à paramétrer l'insertion en vue de résister aux attaques de protocole (copy attack [7]). Le numéro n du bit caché dans ABC est donné par :

$$n = \lfloor g_{tri} \rfloor \% N \quad (2)$$

Où $\%$ dénote l'opérateur modulo. Le paramètre α sert à fixer la sensibilité à la localisation, et T est un paramètre interne de l'algorithme. Plus α est grand, plus le calcul de n fait intervenir de décimales du rapport entre AB^2 et S. La fragilité de l'algorithme est donc paramétrée par α. En appelant H la hauteur de ABC s'appuyant sur AB, l'insertion se fait en utilisant la relation triviale :

$$H \times AB = 2 \times S \quad (3)$$

D'une nouvelle valeur pour le rapport AB^2/S, codant le numéro du bit à cacher, on déduit une modification de la hauteur H, autorisant un codage du numéro du bit perpendiculairement à AB, et n'interférant pas avec le codage de la valeur du bit.

Cette dernière opération sert également à marquer un triangle comme porteur d'information cachée lors de la relecture. En effet, lors de l'insertion, on calcule la valeur de S (via H) telle que :

$$g_{tri} = \lfloor g_{tri} \rfloor \quad (4)$$

Lors de la détection, un triangle sera déclaré porteur d'information cachée si la différence entre les deux termes de l'équation précédente est inférieure à un certain seuil. Ainsi, on peut constater qu'en plus de marquer le numéro et la valeur de chaque bit, il faut encore pouvoir inscrire la validité d'un triangle comme porteur d'information. On marque les triangles sur lesquels on inscrit la marque (numéro et valeur des bits) : cela fait donc trois types d'information à inscrire en réalité.

2.4 Codage de la valeur du bit

La valeur du bit, brouillée par une suite pseudo-aléatoire de germe K, est insérée dans le triangle ABC de manière substitutive en fixant son point C, dans le plan (ABC), parallèlement à AB (voir Fig. 2). La position de C est fixée pour que sa projection sur une partition de la droite (AB) tombe bien dans le sous-ensemble binaire souhaité. Plus cette partition est fine, plus faible est la distorsion introduite [2]. La finesse de la partition (e.g : la distorsion) est paramétrée par le nombre D d'éléments de la partition sur le segment $[AB]$. En ayant calculé les

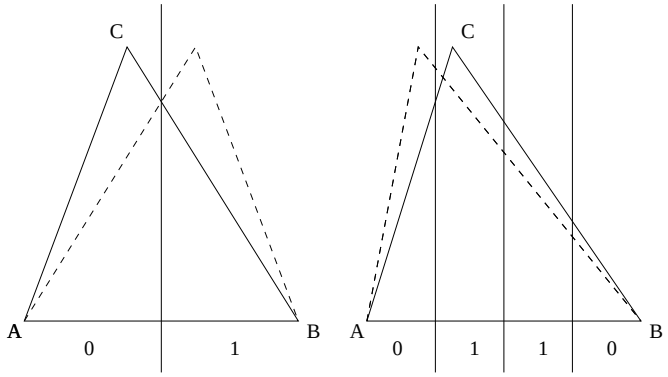

FIG. 2: Les deux premiers niveaux de finesse de la partition du segment AB : $D = 2$ à gauche et $D = 4$ à droite. A gauche : dégénérescence en une simple symétrie. A droite : deuxième niveau de finesse. Nous utiliserons $D = 16$ pour nos expérimentations.

deux modifications orthogonales, on en déduit la future position du point C. Si la déplacement envisagé est trop important ou cause des inversions de facettes, il est rejeté, et les points du triangle ne sont pas marqués comme interdits. La distorsion introduite est donc paramétrable.

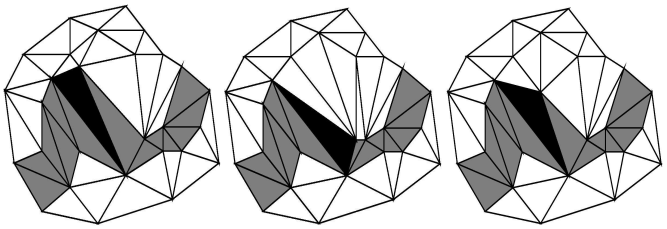

FIG. 3: Simulation dans le plan 2D, sur le triangle en noir au milieu du ruban, des deux procédures de la Fig. 2. A droite : original. A gauche : $D = 2$. A droite : $D = 4$.

3 Extraction du message mais aussi de la clef

Le fonctionnement du module d'extraction est maintenant rapide à expliquer. Aussi, après en avoir précisé les grandes lignes, nous discuterons la sécurité offerte par cette méthode.

3.1 Principe d'extraction du message

En considérant les triangles l'un après l'autre, on commence par déterminer ceux qui sont porteurs d'information cachée, ceux pour lesquels :

$$|g_{tri} - \lfloor g_{tri} \rfloor| < \epsilon \tag{5}$$

On relit ensuite le numéro du bit caché, puis sa valeur. La valeur finale du bit est actuellement l'objet d'une décision majoritaire. Toutefois, nous restons conscients qu'il faut pouvoir garantir la fiabilité de la détection, une modélisation statistique a été entreprise en ce sens.

3.2 Sécurité des clefs

Nous avons pris soin de brouiller l'inscription du numéro du bit à cacher, ainsi que sa valeur. Toutefois, en faisant l'hypothèse que dans une application réelle les paramètres T et α sont connus, la taille de l'espace de recherche exhaustive de la clef ne varie pas. En effet, la clef paramètre l'insertion du numéro de manière analytique. Le pouvoir de discernement entre deux clefs numériquement proches semble d'ailleurs expérimentalement faible. Trouver la clef devient donc un problème d'optimisation, et non plus de recherche exhaustive. Il semble que cela constitue un écueil du tatouage en général [6]. Dans notre cas, il suffit de calculer tous les rapports AB^2/S, puis de faire varier K continûment en s'arrêtant lorsque le nombre de triangles porteurs d'information détectés est suffisamment élevé. On a alors raisonnablement une très bonne estimation de la clef.

Pour remédier à l'estimation de la clef, il faudrait brouiller avec deux clefs différentes les informations de numéro et de valeur. Il semble également que l'espace des clefs, suivant N_K, ne peut pas être arbitrairement vaste. Vu la faible capacité du canal, il ne semble pas prometteur de chercher à profiter de cette scission originelle de la clef en deux parties pour en tirer une application de type tatouage asymétrique. Nous avons donc choisi de réduire l'espace des clefs, afin de continuer à utiliser la même clef pour le numéro et la valeur. Pour l'expérimentation, elles seront codées sur 32 bits.

Une autre solution consiste à prendre des clefs beaucoup plus longues ; par exemple on choisirait 32 bits dans la clef, dévolus au seul brouillage du numéro, alors que la valeur des bits serait cryptée avec la totalité des bits de la clef. Il faudrait alors gérer des clefs d'au moins 160 bits. L'attaque est alors plus complexe : résoudre le problème d'optimisation ne permettrait de retrouver, au mieux, que 32 bits parmi les 160 de la clef, et les 128 bits restant relèveraient de la cryptographie classique. Il s'agit en quelque sorte de doser la sécurité apportée par l'un et l'autre composant, cryptographie et tatouage, en conférant au tatouage un rôle de premier rideau défensif, sans restreindre la puissance de la cryptographie classique.

4 Résultats et perspectives

Nous avons inséré $N = 64$ bits, cryptés avec la clef 1234 (codée sur $N_K = 32$ bits), avec un paramètre interne $T = 5$, un paramètre de fragilité $\alpha = 3$ et un paramètre de distorsion $D = 16$. La valeur du seuil de détection ϵ a été fixée à $0,01\%$ de la boite englobante, tout comme la valeur du plus grand déplacement autorisé à l'insertion (ce qui veut dire qu'en général c'est un retournement de facette qui empêcherait d'inscrire le bit à cacher). Les résultats sont calculés sur une moyenne de trois maillages. Nous illustrons sur la Fig. 4 la robustesse de notre méthode face à l'attaque de coupe. Nous avons vérifié expérimentalement que la marque résiste en effet à la translation, à la rotation, et à la mise à l'échelle uniforme. Nous avons également constaté la disparition de la marque lors d'une compression MPEG-4 (12 bits de quantification par défaut). La modification apportée pour coder le numéro du bit étant plus fine que celle chargée d'inscrire sa valeur, c'est d'abord l'information de numéro qui est attaquée et qui efface le tatouage. Nous représentons sur la Fig. 5 un maillage entier avec la localisa-

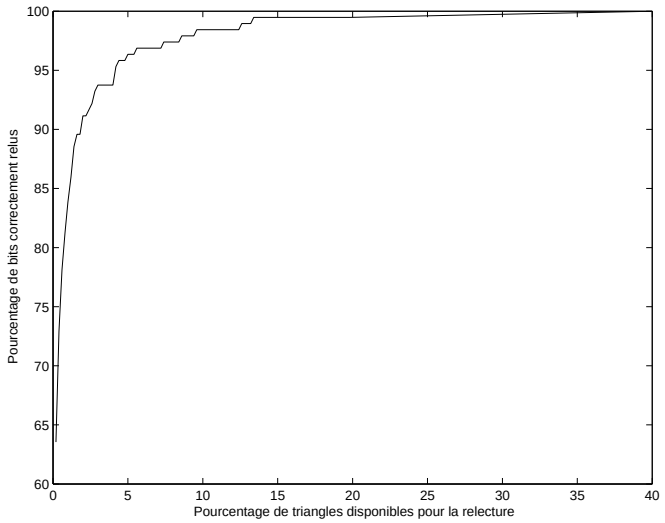

FIG. 4: Attaque de coupe sur trois maillages (moyenne). La marque, répétée 500 fois, est retrouvée au-delà de 40% de triangles disponibles à l'extraction.

tion de l'information cachée associée. L'information est insérée aléatoirement partout sur le maillage.

Le taux d'erreur face à l'attaque de coupe doit pouvoir être amélioré afi n de pouvoir tester l'authenticité de portions du maillage contenant moins de 40% des triangles. En particulier, une décision plus souple du numéro des bis cachés devrait offrir de meilleures performances (la décision de la valeur des bits reste l'objet d'une décision majoritaire). Enfi n, l'utilisation de codes correcteurs d'erreurs devrait permettre de retrouver la marque à partir de 15% des triangles disponibles [1].

Nous remercions Carlos Hernandez (Télécom Paris, TSI) pour avoir accepté de nous prêter quelques-uns de ses maillages muséologiques [5], et Olivier Devillers (INRIA-Sophia, Prisme) pour ses discussions stimulantes au sein de l'ARC TéléGéo.

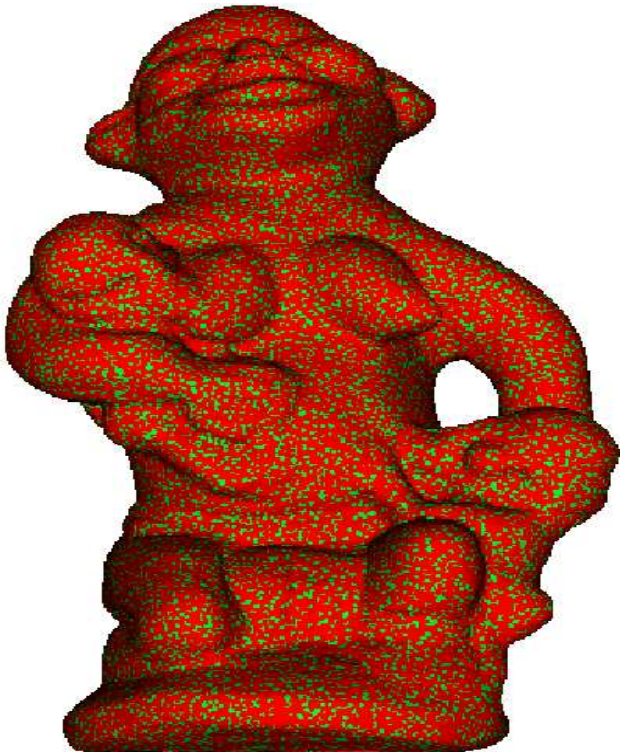

FIG. 5: Maillage "twins" : 83K points, 166K triangles dont 35K porteurs d'information cachée. Le message de 64 bits est répété environ 540 fois.

Références

[1] S. Baudry, J.-F. Delaigle, B. Sankur, B. Macq et H. Maître. *Analyses of Error correction Strategies for Typical communication channels in Watermarking*. Signal Processing, Vol. 81, No. 6, pp. 1239-1250, Jul. 2001.

[2] F. Cayre et B. Macq. *Data hiding on 3D triangle meshes*. IEEE Trans on Signal Processing, Vol. 51, Issue 4, pp. 939-949, Avr. 2003.

[3] The CERTIMARK Project. *Deliverable D21 : Benchmark requirements*. http://www.certimark.org.

[4] O. Faugeras. *Three-dimensional Computer Vision*. MIT Press, ISBN 0-262-06158-9, pp. 342-345, 1993.

[5] C. Hernandez et F. Schmitt. *Multi-Stereo 3D Object Reconstruction*. Proc. 1st Intl Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02), pp. 159-166, Jun. 2002.

[6] T. Kalker. *Considerations on Watermarking Security*. IEEE Workshop on Multimedia Signal Processing, MMSP'01, pp. 201-206, Oct. 2001.

[7] M. Kutter, S. Voloshynovskiy et A. Herrigel. *Watermark Copy Attak*. Proc. SPIE Security and Watermarking of Multimedia Content II, Vol. 3971, San Jose, CA, pp. 945-955, Jan. 2001.

[8] R. Ohbuchi, H. Masuda et M. Aono. *Watermarking three-dimensional polygonal models*. Proceedings of ACM Intl Conf on Multimedia, pp. 261-272, Nov. 1997.