

Dissimulation de données par changement de connexité d'un maillage 3D - Data Hiding Based on Connectivity Modification of 3D Mesh

Philippe Amat, William Puech, Sébastien Druon, Jean-Pierre Pedeboy

▶ **To cite this version:**

Philippe Amat, William Puech, Sébastien Druon, Jean-Pierre Pedeboy. Dissimulation de données par changement de connexité d'un maillage 3D - Data Hiding Based on Connectivity Modification of 3D Mesh. Traitement du Signal, 2008, 25 (3), pp.29. lirmm-00293138

HAL Id: lirmm-00293138

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00293138

Submitted on 3 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dissimulation de données par changement de connexité d'un maillage 3D
Data Hiding Based on Connectivity Modification of 3D Mesh

P. Amat[1,2], W. Puech[1], S. Druon[1] et J.P. Pedeboy[2]

[1]Laboratoire LIRMM, UMR CNRS 5506, Université de Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE,
[2]Stratégies S.A., 41-43 rue de Villeneuve, Parc d'affaires SILIC - BP 80429
94583 Rungis cedex - France

Mots clefs

Dissimulation de données, Tatouage, Maillage triangulaire 3D, Quadrangle, Parcours de graphe, Arbre couvrant minimum, Analyse en composante principale, Synchronisation des données.

Key words

Data hiding, Watermarking, 3D mesh, Quadrangle, Graph path, Minimum Spanning Tree, Principal component analysis, Data synchronization.

Abstract

Day by day, the amount of digital data has been rapidly increasing on the Internet. The size of 3D objects is very large and these objects need fast transmissions. Moreover, 3D data security becomes increasingly important for many applications, e.g., confidential transmission, video surveillance, military and medical applications. In this paper we present two new approaches of 3D object data hiding without changing the position of vertices in the 3D space. The main idea of the two proposed methods is to find and to synchronize particular areas of the 3D objects used to embed the message. The embedding is done by changing the connectivity of edges in the selected areas composed of quadrangles. The first proposed approach of data hiding in 3D objects is based on minimum spanning tree (MST) while the second approach is based on the projection on a secret axis of the quadrangle centers. These methods are lossless in the sense that the positions of the vertices are unchanged. Moreover they are blind and do not depend of the order of the data in the files. These two approaches are very interesting when the 3D objects have been digitalized with high precision.

Résumé

De nos jours, des visualisations ainsi que des transferts d'objets 3D sont couramment effectués pour de nombreuses applications allant du jeu vidéo à l'imagerie médicale en passant par l'industrie manufacturière. Dans cet article nous proposons deux nouvelles méthodes permettant de dissimuler des données dans des objets 3D sans modifier la position des sommets. L'idée principale des deux méthodes présentées est de trouver et de synchroniser des zones particulières dans l'objet 3D pouvant être utilisées pour insérer le message. L'insertion de données s'appuie sur la modification de la connexité des arêtes dans les zones sélectionnées composées de quadrangles. La différence entre les deux méthodes présentées est la manière de sélectionner et de synchroniser ces zones d'insertion. Alors que la première méthode s'appuie sur un arbre couvrant minimum (ACM), la seconde méthode utilise un axe sur lequel sont projetés les centres des zones d'insertion. Ces deux méthodes aveugles, protégées par utilisation de clefs secrètes, résistent à des transformations géométriques tels que les rotations, translations ou changement d'échelle et ne sont pas perturbées par des modifications directes de l'ordre des données dans les fichiers originaux. Ces approches trouvent un intérêt certain pour des objets 3D dont les sommets ont été acquis avec une grande précision et dont la modification n'est pas acceptable.

1 Introduction

Avec l'évolution des réseaux informatiques, le téléchargement d'objets 3D devient une transmission courante. Afin d'être universellement lisibles ces objets 3D doivent répondre à des normes et des standards internationaux. Même si nous souhaitons apporter de nouvelles fonctionnalités à ces objets, il est donc important de respecter les formats standards[1] permettant de manipuler ces objets 3D. Dans nos travaux, ces nouvelles fonctionnalités concernent la protection des objets 3D durant la transmission et l'archivage, mais également l'enrichissement de ces objets avec des méta-données sans augmenter la taille des objets et en synchronisant ces méta-données de manière hiérarchique (qualité et spatiale). La représentation la plus utilisée pour les objets 3D est le maillage surfacique. Cette représentation, composée de sommets, d'arêtes et de faces, est très répandue car supportée par toutes les cartes de rendu graphique. De nombreuses méthodes d'insertion de données sont basées sur cette représentation. Les méthodes d'insertion de données[2], peuvent être classées en deux catégories, spatial ou transformé, en fonction du domaine utilisé pour réaliser l'insertion de données. Quel que soit le domaine d'insertion, les méthodes de marquage d'objets 3D existantes s'appuient généralement sur la modification de sommets du maillage 3D pour réaliser l'insertion des données cachées. Modifier la position des sommets d'un objet 3D acquis avec une grande précision n'est pas acceptable pour de nombreuses applications tels que le médical ou l'industrie manufacturière. Dans cet article nous proposons deux méthodes d'insertion de données cachées qui ne s'appuient pas sur la modification des sommets des objets 3D pour réaliser l'insertion. Ces méthodes permettent d'insérer des méta-données dans un objet 3D tels que le nom de l'auteur, la date de création de l'objet, un logo 2D ou 3D, la couleur de l'objet ou encore la texture de l'objet lui-même. Notons que ces méthodes résistent à des transformations géométriques telles que les rotations, les translations et les mises à l'échelle uniforme.

[1] tels que STL, PLY, X3D etc ...

[2] *Data hiding* en anglais, appelées également méthodes de tatouage ou *watermarking en anglais*

Le contenu de cet article se décompose de la manière suivante. En Section 2, nous présentons un état de l'art sur les différentes méthodes d'insertion de données cachées appliquées aux objets 3D. Nous décrivons, Section 3, les deux méthodes d'insertion de données cachées proposées. En Section 4, nous présentons et analysons les résultats de nos méthodes appliquées à des objets 3D réels. Enfin nous terminons, Section 5, par des conclusions et perspectives.

2 Etat de l'art

Selon le type d'application, les méthodes de tatouage peuvent être choisies en fonction de certaines contraintes. Les contraintes les plus standards, relativement antagonistes, sont les contraintes de perceptibilité, de robustesse et de capacité. Dans de nombreuses applications, les contraintes de sécurité ou encore de complexité en temps de calcul sont également à prendre en compte. les méthodes, quand à elles, peuvent être classées en fonction du domaine d'insertion choisi. Les deux principaux domaines d'insertion sont le domaine spatial et les domaines transformés. Les méthodes d'insertion sont également divisées sur la base d'autres critères comme le fait d'avoir besoin ou non de l'objet 3D original (méthodes non aveugles ou aveugles). Si seulement quelques paramètres sont nécessaires pour l'extraction du message, la méthode est alors dite semi-aveugle. Des méthodes totalement aveugles peuvent nécessiter l'utilisation d'une clef secrète pour extraire le message caché. Dans ce cas, la clef secrète est indépendante du contenu caché mais également de l'objet 3D lui même. Un algorithme de tatouage doit pouvoir être connu de tout le monde, même s'il utilise une clef secrète, afin de respecter le principe de Kerckhoffs [Ker83].

Dans cet article, nous proposons de classer les méthodes d'insertion de données cachées en deux catégories. La première de ces catégories de méthodes d'insertion repose sur l'ordonnancement des primitives des objets ou sur la connectivité et la topologie des objets 3D à marquer. Dans cette première catégorie la position des points composant l'objet 3D original est inchangée. La seconde catégorie s'appuie sur des déplacements spatiaux des points de l'objet 3D ou sur la déformation géométrique d'autres primitives constituant les objets 3D, que ce soit dans le domaine spatial ou dans un domaine transformé.

Méthodes basées sur l'ordonnancement des primitives ou sur la connectivité et la topologie

Cette première catégorie présente donc des méthodes d'insertion qui reposent sur l'ordonnancement des primitives des objets ou sur la modification de la connectivité ou de caractéristiques topologiques des objets 3D à marquer. Ces modifications sont prises en compte en général dans le domaine spatial. Dans cette catégorie, des méthodes d'insertion sont basées sur l'organisation des données au niveau du fichier constituant l'objet 3D.

Ichikawa *et al.* [ICA02] proposent deux méthodes de tatouage utilisant la modification de l'organisation des données afin d'insérer un message caché. La première méthode modifie dans la liste l'ordre des triangles, alors que la seconde méthode modifie l'ordre des triplets de points formant le triangle. L'algorithme de tatouage *Triangle Strip Peeling Symbol sequence* (TSPS) de Ohbuchi *et al.* est basé sur un changement topologique [OMA97]. Pour insérer de l'information au niveau du maillage, cet algorithme propose de déconnecter une bande de triangles du maillage original. La première étape de l'algorithme consiste à choisir, dans le maillage original, une arête de départ du triangle initial, notée e. A partir de cette arête de départ et du message à insérer, une bande

de triangles est générée. Le choix du triangle suivant à rajouter dans la bande se fait en fonction de la valeur du bit de message à insérer et du sens de parcours choisi dans le triangle (horaire ou anti horaire). Cela permet donc d'indexer toutes les arêtes à 1 ou à 0 en fonction de leur sens de parcours dans le triangle. Les arêtes suivantes empruntées, et donc les triangles générant cette bande, sont choisies en fonction du message à insérer. L'étape suivante de l'algorithme consiste à déconnecter cette bande de triangles du maillage en dupliquant tous les sommets de la bande sauf ceux de l'arête e permettant de retrouver le message. Le maillage tatoué se retrouve donc avec un trou à la surface recouvert par la bande de triangles contenant le message. L'extraction du message se fait en cherchant l'arête e puis en parcourant la bande de triangles. Les inconvénients majeurs de cette méthode sont que l'objet est troué et que des sommets sont rajoutés, donc la taille du fichier augmente. De plus, nous pouvons noter que cette insertion est très locale dans le sens où le message inséré n'est pas diffusé dans tout le maillage de l'objet, et qu'il est facile de repérer dans le fichier les sommets qui ont été dupliqués et donc de retrouver le message inséré.

Mao *et al.* ont développé une méthode d'insertion consistant à ajouter des triangles plus petits à l'intérieur des triangles initiaux de l'objet 3D [MSI01]. Avec cette approche, chacune des arêtes d'un triangle initial est divisée en deux arêtes par le sommet d'un petit triangle inséré. Le ratio entre les longueurs des deux arêtes ainsi formées est utilisé pour réaliser l'insertion d'un bit. Pour initialiser le processus, deux arêtes d'un triangle sont choisies à l'aide d'un GNPA (Générateur de Nombres Pseudo-Aléatoires). Après le tatouage de la dernière arête d'un triangle, c'est le triangle voisin à cette arête, si il existe et si il n'a pas déjà été tatoué, qui est parcouru afin de continuer l'insertion du message. La capacité de cette méthode est de 2 bits par triangle, sauf pour le triangle initial dans lequel 3 bits sont insérables. Cette méthode peut avoir une forte capacité d'insertion du fait que l'étape de division d'un triangles peut-être répétée plusieurs fois. Pour l'extraction, à partir de la même clef, les deux arêtes du triangle initial sont retrouvées, ensuite le parcours des triangles est identique à la phase d'insertion. Les inconvénients majeurs de cette méthode sont que des sommets sont rajoutés, donc la taille du fichier augmente, et que les zones marquées sont visibles du fait qu'un triangle marqué génère quatre triangles plus petits car contenus dans la surface du triangle initial. De plus, la propriété de voisinage des faces[3] n'est plus respectée en bordure de la zone marquée.

En conclusion de cette première classe de méthodes, nous pouvons noter que la position des points composant l'objet 3D original est inchangée. Cependant certaines méthodes proposent de rajouter des points supplémentaires et la plupart de ces méthodes sont très fragiles et ne résistent pas à la manipulation du fichier contenant les données initiales telle que l'inversion de deux lignes par exemple. Les approches que nous développons en Section 3 font partie de cette catégorie mais permettent de résister à la manipulation des fichiers contenant les données initiales, diffusent le message dans tout l'objet et ne rajoutent pas de sommets.

Méthodes basées sur des modifications géométriques de l'objet 3D

Cette seconde catégorie d'algorithme de tatouage est basée sur la modification de la géométrie de l'objet 3D, comme la position des points ou encore les normales des triangles ou des faces. Contrairement à la première catégorie présentée, qui ne s'applique que dans le domaine spatial, cette seconde catégorie peut s'appliquer dans les domaines spatial et transformés.

[3]Chaque arête d'un maillage doit avoir exactement deux faces adjacentes pour une arête intérieure au maillage et une seule si c'est une bordure du maillage.

Ohbuchi *et al.* proposent différentes méthodes de tatouage dans le domaine spatial, comme le *Tetrahedral Volume Ratio* (TVR) et le *Triangle Similarity Quadruple* (TSQ) [OMA97]. Le TVR utilise le rapport des volumes de deux tétraèdres. Dans cette méthode les tétraèdres utilisés sont formés par une arête et les deux triangles incidents. Le parcours du graphe de connexité repose sur un arbre de recouvrement des points. Les arêtes de l'arbre construit servent à générer et à référencer les tétraèdres ainsi que leur ordre de parcours. Concernant la méthode basée sur le TSQ, la configuration géométrique est constituée de quatre triangles dont l'un est central et sert à signaler la présence d'information cachée. L'un de ses voisins contient l'index du triangle utilisé pour insérer le bit du message. Les deux autres voisins restants, servent alternativement à enregistrer l'information utile. L'insertion de l'information est alors effectuée en modifiant des paires de rapports de distances des cotés des triangles. L'algorithme *Vertex Flood* de Benedens [Ben99a] repose sur la modification de la distance des points admissibles au centre de masse de l'objet. Une seconde méthode de Benedens *Triangle flood* utilise les informations de connectivité et de géométrie pour générer un parcours sur le maillage de l'objet [Ben99b]. Dans cette méthode l'insertion du message se fait en modifiant la position des points afin d'altérer la hauteur des triangles. La méthode proposée par Bors utilise la configuration géométrique locale des points pour sélectionner des zones d'insertion [Bor06]. La zone est représentée par un sommet et tous les triangles contigus avec un voisinage ellipsoïdal autour de ce sommet. Pour coder une valeur à 0, le sommet est déplacé à l'extérieur de son voisinage et pour coder une valeur à 1 il est déplacé à l'intérieur. Dans la méthode de Cayre et Macq, l'insertion est faite en projetant le sommet d'un triangle sur sa base et en regardant si celle-ci appartient à la moitié droite ou gauche de la base [MC03]. La sélection des zones d'insertion est faite par un parcours unique du maillage à l'aide d'une clef secrète. La méthode de Wagner [Wag00] vise à insérer les données cachées dans les longueurs relatives des vecteurs laplaciens locaux en modifiant la courbure locale de l'objet. Benedens a proposé une méthode de tatouage d'objet 3D en utilisant des NURBS et en modifiant les poids et les vecteurs de noeuds de manière à préserver la géométrie globale [Ben00]. La méthode de Lee *et al.* propose d'insérer la marque non plus directement dans l'objet 3D, mais dans des images virtuelles dérivées de celui-ci [LCK02].

Dans le domaine transformé, Ohbuchi *et al.* proposent une méthode qui utilise une matrice laplacienne du maillage de l'objet 3D afin de passer du domaine spatial au domaine fréquentiel [OMT02]. Dans le domaine des ondelettes, la méthode de Kanai *et al.* utilise une décomposition en ondelettes du maillage et insère l'information en modifiant le bit de poids faible de certains coefficients d'ondelettes [KDK98]. Yin *et al.* proposent également un algorithme qui travaille dans le domaine des ondelettes [YPSZ01]. L'algorithme est basé sur une décomposition multirésolution du maillage de l'objet 3D développé par Guskov *et al.* [GSS99]. La méthode proposée par Praun *et al.* utilise l'espace de Kernel pour insérer l'information dans les composantes basses fréquences de la surface [PHF99]. Wang *et al.* proposent une méthode de tatouage hiérarchique [WLDB07b]. Grâce à une décomposition en ondelettes, un tatouage robuste aux attaques géométriques et un tatouage de haute capacité sont appliqués à différents niveaux de résolution du maillage en modifiant les normes des coefficients d'ondelettes. Alface *et al.* analysent les propriétés des méthodes de tatouage basées sur les histogrammes et les utilisent afin d'être robuste au découpage [AMC07]. Récemment, une référence des tests de type benchmark a été proposé par Bennour et Dugelay [BD07]. Deux états de l'art récents dans le domaine complètent cette section [RAM07, WLDB07a].

3 Méthodes proposées

Dans cette section nous détaillons deux nouvelles méthodes de dissimulation de données dans des objets 3D. Nos méthodes se basent sur un modèle 3D représenté par un nuage de points et une liste d'arêtes correspondant au maillage triangulaire de la surface. L'idée principale des deux méthodes est de trouver et de synchroniser des zones particulières pouvant être utilisées pour insérer le message. L'insertion de données s'appuie sur la modification de la connexité des arêtes dans des zones sélectionnées. Ces modifications ont pour conséquence de modifier la structure des triangles construits dans ces zones. Ces méthodes de dissimulation présentent l'avantage de ne pas modifier la position initiale des points du modèle 3D. Cet invariant sur la position des points nous permet de rendre l'insertion robuste aux transformations affines de type rotation, translation ou changement d'échelle. La différence entre les deux méthodes présentées est la manière de sélectionner et de synchroniser les zones d'insertion. Alors que la première méthode, présentée Section 3.1, s'appuie sur un arbre couvrant minimum (ACM), la seconde méthode, présentée Section 3.2, utilise un axe sur lequel sont projetés les centres des zones d'insertion.

3.1 Insertion basée sur un arbre couvrant minimum

Dans cette section nous présentons l'approche d'insertion de données cachées basée sur un arbre couvrant minimum (ACM). Cette approche est composée de trois étapes. La première étape consiste à construire un ACM à partir des sommets initiaux du modèle 3D. La seconde étape nécessite de parcourir l'ACM construit afin de sélectionner des zones pour l'insertion des données. Le point de départ utilisé pour parcourir le graphe est obtenu à partir d'une clef secrète. A partir d'une analyse en composante principale (ACP) de l'objet 3D, la clef secrète nous permet, dans le repère de l'objet, de localiser le point de départ de l'ACM. Du fait de cette ACP, une réorganisation du fichier ne perturbe pas la localisation de ce point de départ qui ne dépend donc que de la clef secrète. Les zones d'insertion doivent répondre à plusieurs contraintes dues à la connexité de l'ACM et à la géométrie de la zone. La troisième étape concerne l'insertion du message elle-même. Celle-ci est faite en modifiant la connexité des triangles dans les zones sélectionnées. La Figure 1 illustre le schéma général de cette première méthode.

En Section 3.1.1 nous détaillons la définition d'un ACM et présentons sa construction. Nous montrons, Section 3.1.2, comment sélectionner les zones d'insertion afin de les synchroniser avec le message. La méthode d'insertion de données cachées est décrite dans la Section 3.1.3. Nous montrons, Section 3.1.4, que les zones d'insertion sélectionnées sont soumises à des contraintes géométriques. Enfin, la Section 3.1.5 décrit comment extraire le message dans un fichier 3D marqué.

3.1.1 Arbre couvrant minimum

Définition : soit G un graphe formé d'un ensemble de couples de sommets représentant une relation. Soient a et b deux sommets de ce graphe qui sont en relation, le couple (a, b) est donc une arête du graphe G. Soit A un graphe non orienté connexe sans cycle, dans ce graphe il n'existe donc pas de chemin relié à un sommet de A qui permette de revenir sur ce même sommet. Ce graphe respecte alors deux propriétés :

- Si une arête est enlevée, il n'est alors plus connexe.
- Si une arête est ajoutée, celle-ci forme alors un cycle.

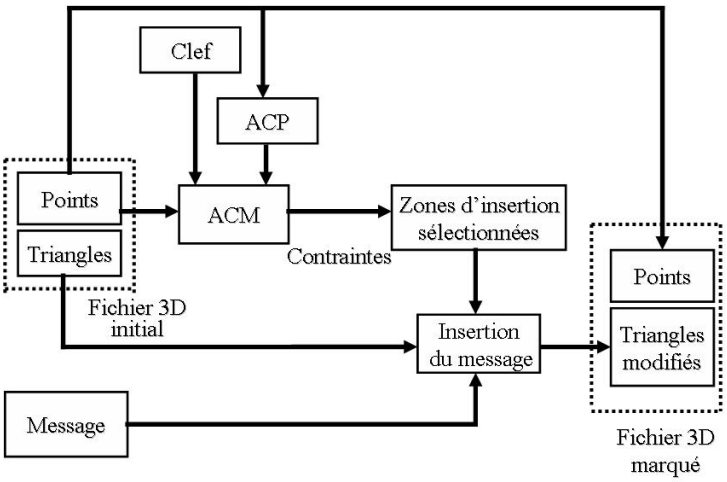


FIG. 1 – Présentation générale de la méthode d'insertion basée sur un ACM.

Pour le graphe G, il existe au moins un arbre couvrant A. Il s'agit d'un graphe partiel qui possède les mêmes sommets que G et dont l'ensemble des relations est inclus dans celui des relations de G. Le coût du graphe G correspond à la somme de toutes les pondérations des arêtes de G. Si A est un arbre couvrant de G dont la somme des pondérations de ses arêtes est la plus petite valeur par rapport à celle de tous les autres arbres couvrant de G, alors A est appelé *arbre couvrant de poids minimum de* G et noté ACM. Dans notre méthode le poids utilisé pour calculer l'ACM est la distance euclidienne entre les points de l'objet 3D. La particularité d'un ACM est qu'il est unique, il existe cependant une ambiguïté dans les méthodes de construction lorsque un sommet possède deux arêtes de même poids. La méthode choisit l'arête qu'elle rencontre en premier dans son parcours des sommets. Si l'ordonnancement des sommets dans le fichier est modifié alors l'ACM lui aussi pourrait être modifié. Pour palier à cette ambiguïté un calcul de distance entre les deux sommets des deux arêtes identiques et le centre de masse de l'objet est utilisé. De cette façon **l'ACM est bien unique quelle que soit la réorganisation des points dans le fichier de l'objet 3D**. C'est cette particularité forte des ACM que nous utilisons Section 3.1.2 afin de synchroniser le message avec le modèle 3D. Nous nous appuyons également sur cette caractéristique pour insérer les données présentée Section 3.1.3.

Construction de l'ACM : il existe plusieurs algorithmes pour construire des ACM. Les plus connus sont l'algorithme de *Prim* [Pri57] et l'algorithme de *Kruskal* [Kru56].

- *L'algorithme de Prim* construit progressivement un sous-graphe A de G en choisissant l'arête de poids minimum localement à chaque étape, en espérant qu'elle fasse partie de la solution globale [Pri57]. Au départ, A est vide et un sommet s de G est choisi comme point de départ. A est alors uniquement composé de s. Soit E l'ensemble des sommets en relation avec s dans A, soit F l'ensemble des sommets restants. Au départ E est donc vide, et F contient tous les sommets de G. A chaque étape une arête est ajoutée à A telle que cette arête soit de poids minimum parmi toutes les arêtes ayant un sommet x la composant dans F et l'autre dans E. Ceci permet de ne pas former de cycle. Ainsi le sommet x sort de l'ensemble F et est inséré

dans E. L'algorithme s'arrête quand l'ensemble F est vide. La Figure 2.b présente un résultat de l'algorithme de Prim à partir du nuage de points illustré Figure 2.a.

– *L'algorithme de Kruskal* construit progressivement un sous-graphe A de G en triant d'abord l'ensemble des arêtes selon leur poids et ensuite en sélectionnant celles qui ne formeront pas de cycle dans A [Kru56]. Pour cela, les sommets sont choisis un par un et triés dans l'ordre croissant selon la pondération de l'ensemble des arêtes qui lient un sommet s aux autres. Pour chaque arête (s, s') triée il faut vérifier si la liste des voisins de s dans A est différente de celle de s' dans A. Si aucun sommet voisin n'est commun, alors l'arête (s, s') est ajoutée à A.



(a) (b)

FIG. 2 – a) Nuage de points initial (453 points), b) Calcul de l'ACM avec l'algorithme de Prim.

3.1.2 Sélection et synchronisation des zones d'insertion

La synchronisation des données cachées est obtenue en repérant des connexités particulières présentes dans l'ACM. Une fois l'ACM construit, deux phases sont nécessaires afin de synchroniser le message à insérer. La première phase consiste à définir le sens de lecture de l'ACM. En effet, l'arbre est unique mais le parcours de celui-ci dépend du sommet de départ. Dans notre approche ce sommet est obtenu à partir d'une clef secrète. L'utilisation d'une clef nous apporte une sécurité sur le message inséré dans le modèle 3D. Le nombre de sens de lecture est égal au nombre de sommets dans le modèle 3D. Ainsi plus le nombre de sommets est grand plus le système est protégé[4]. Une fois que le sens de lecture de l'ACM est obtenu, des zones particulières sont recherchées afin d'effectuer l'insertion de données. Ces zones particulières correspondent à des noeuds de l'ACM qui possèdent trois arêtes comme représenté Figure 3. Ces zones sont formées de quatre sommets, un noeud père et trois autres noeuds fils reliés chacun par une arête. Nous cherchons dans le maillage de la forme 3D les deux triangles inclus dans la zone formée par ces quatre noeuds. Les trois points de chacun de ces triangles doivent appartenir aux quatre noeuds de la zone. Deux triangles auront donc une arête commune dans la zone et forme un quadrangle. L'ensemble des quadrangles représente donc les zones où sont insérés les bits du message. La méthode d'insertion utilisée et présentée Section 3.1.3 utilise directement ces zones particulières détectées et synchronisées. Cependant, nous

[4]La clef peut également servir de semence pour un GNPA afin de disperser le message dans l'objet 3D

détaillons Section 3.1.4 qu'une partie de ces quadrangles ne peut pas être utilisée pour réaliser l'insertion. En effet, certaines zones ne répondent pas aux contraintes demandées par la méthode d'insertion et d'autres ne peuvent pas être retenues pour des raisons de non invisibilité de l'insertion. L'ACM, de par sa construction, va nous permettre de résister aux transformations affines telles que les rotations, les translations ou les changements d'échelles. Ceci est rendu possible grâce à notre méthode d'insertion qui ne modifie pas la position des sommets de l'objet 3D. Le nombre de zones sélectionnées dépend du nombre de points de l'objet 3D, des valeurs des seuils pour les différentes contraintes mais également du maillage global de l'objet. Nous montrons, Section 4, combien de zones en moyennes sont sélectionnées en fonction de tous ces paramètres.

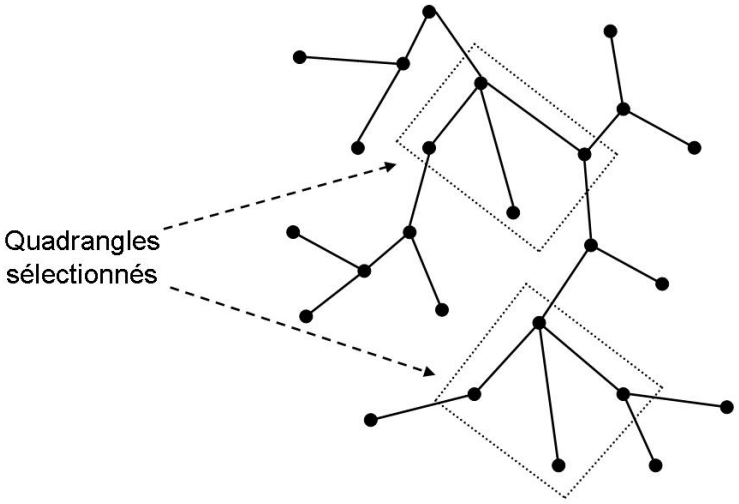
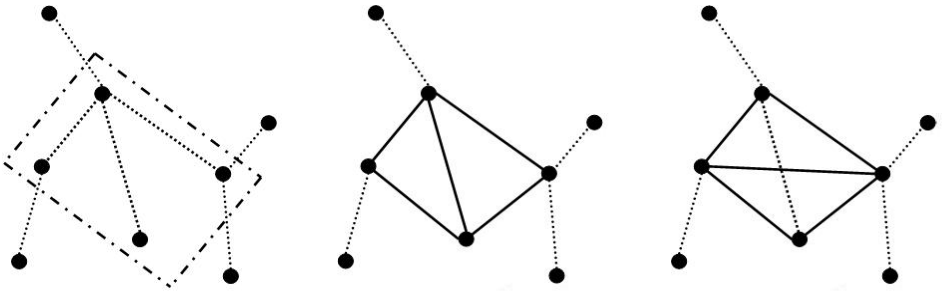


FIG. 3 – Exemple de zones formant des quadrangles sélectionnées pour la synchronisation des données cachées.

3.1.3 Insertion des données

Cette partie présente la phase d'insertion des données cachées. Dans la Section 3.1.2 nous avons présenté comment localiser les zones d'insertion à l'aide de l'ACM. Ces zones sont formées de quatre sommets, un noeud père et trois noeuds fils reliés chacun par une arête. Ces zones constituent donc des quadrangles. Pour insérer un bit du message dans un quadrangle sélectionné nous examinons l'arête commune des deux triangles inclus dans le quadrangle.

Une zone sélectionnée pour l'insertion est présentée Figure 4.a, elle est composée d'un noeud avec trois arêtes. L'insertion d'un bit à 0 est réalisée en construisant l'arête commune des deux triangles sur l'arête initiale de l'ACM comme présenté Figure 4.b. Au contraire pour l'insertion d'un bit à 1, l'arête commune des deux triangles est construite de façon à ne pas être en superposition avec l'arête initiale de l'ACM comme présenté Figure 4.c. Notons que la modification a lieu dans la connexité du maillage et non dans celle de l'ACM. Cette méthode d'insertion permet de ne pas modifier la position des sommets du modèle 3D. Ce point est une notion importante pour la synchronisation et nous permet de reconstruire le même ACM lors de l'extraction du message. Par contre, cette méthode d'insertion impose des contraintes sur les quadrangles choisis. Modifier la connexité du



(a) (b) (c)

FIG. 4 – a) Partie de graphe issu de l'ACM et sélection d'un quadrangle, b) Insertion d'un bit à 0, c) Insertion d'un bit à 1.

maillage peut amener des erreurs au niveau de celui-ci mais aussi des défauts visuel au niveau de la surface. Nous détaillons Section 3.1.4 les contraintes à respecter et expliquons pourquoi certains quadrangles ne peuvent pas être utilisés pour insérer les données.

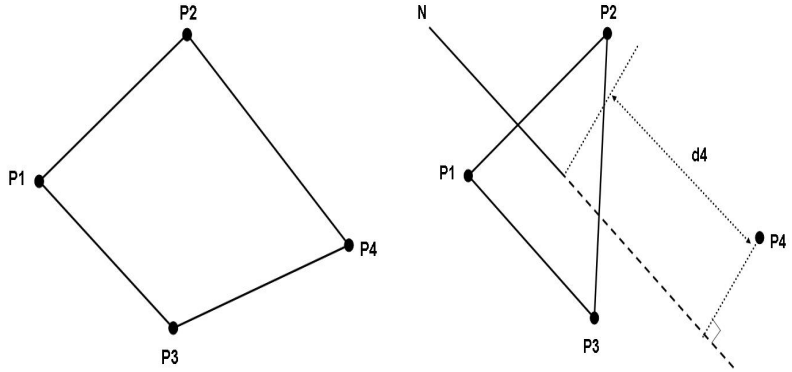
3.1.4 Contraintes sur le choix des zones d'insertion

Dans cette section nous présentons les contraintes appliquées aux quadrangles résultants de l'étape de synchronisation. En effet, une partie de ces quadrangles n'est pas utilisée pour effectuer l'insertion de données. Ces contraintes concernent la coplanarité, la convexité et le recouvrement.

Contrainte de coplanarité : la première contrainte est la coplanarité des quadrangles. Le fait de modifier la connexité des deux triangles formant un quadrangle modifie aussi l'angle formé entre ces deux triangles. Par conséquent, l'insertion de données sur un quadrangle non coplanaire affecte la surface et donc le rendu visuel du modèle 3D. De manière idéale il ne faudrait insérer le message que dans les quadrangles strictement coplanaires. Cependant, la quantité de quadrangles répondant strictement au critère de coplanarité est très limitée. Afin d'augmenter la capacité d'insertion, nous insérons dans notre approche un seuil S_c de tolérance sur la coplanarité des quadrangles sélectionnés. Ce seuil nous permet d'obtenir un compromis entre la capacité d'insertion et la qualité du modèle 3D. Plus ce seuil sera grand, plus la capacité d'insertion sera grande au détriment du rendu visuel. Au contraire plus ce seuil sera petit, plus la capacité d'insertion sera petite mais moins sera dégradé l'objet 3D. Par l'intermédiaire du seuil S_c, un réglage est donc possible entre la capacité d'insertion et le rendu visuel du modèle 3D. Soit le quadrangle Q_{1234} formé des points P_1, P_2, P_3, P_4. Dans un premier temps, la normale N au plan P_{123} est calculée. Le point P_4 est alors projeté orthogonalement sur la normale N. La distance qui sépare la projection du point P_4 sur N du plan P_{123} est nommé d_4. Ce calcul de distance est effectué pour les projections de P_1, P_2 et P_3 respectivement sur les normales aux plans P_{234}, P_{134} et P_{124}. Le quadrangle ne sera retenu que si la plus grande des distances calculées est inférieure au seuil S_c préalablement fixé :

$$Q_{1234} \text{ est retenu si et seulement si } \min_{i=\{1,2,3,4\}}(d_i) \; < \; S_c. \tag{1}$$

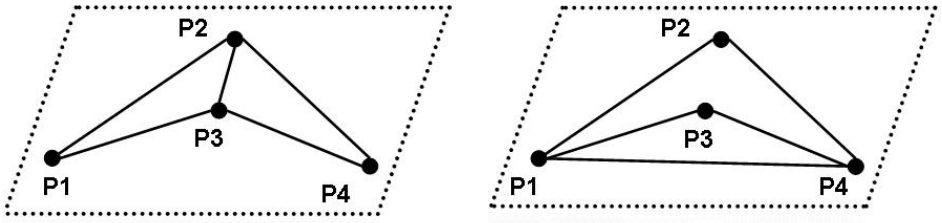
Un exemple de calcul de d_4 pour la coplanarité d'un quadrangle est illustré Figure 5.



(a) (b)

FIG. 5 – a) Quadrangle Q_{1234}, b) Mesure de d_4 pour la contrainte de coplanarité.

Contrainte de convexité : la seconde contrainte impose que les quadrangles utilisés pour la dissimulation de données soient convexes. Pour cette contrainte nous partons du principe que les quadrangles retenus répondent déjà à la contrainte de coplanarité. La Figure 4 illustre un quadrangle convexe où un bit à 0 ou à 1 ont été insérés respectivement Figures 4.b et 4.c. Dans ce cas, malgré le changement de connexité des triangles, le quadrangle répond à des caractéristiques identiques et couvre le même espace géométrique pour les deux cas d'insertion. La Figure 6 illustre un cas de quadrangle non convexe formé par les plans P_{123} et P_{234}. Nous constatons que la modification de la connexité des triangles lors de l'insertion d'un bit modifie la surface couverte par le précédent quadrangle. Cette modification ajoute une erreur sur la surface du quadrangle. Cette erreur est représentée par la surface P_{134} de la Figure 6. C'est pour éviter cette erreur sur la surface que la contrainte de convexité a été introduite.



(a) (b)

FIG. 6 – a) Triangles T_{123} et T_{234} avant insertion, b) Triangles T_{124} et T_{134} après insertion.

Le calcul de cette contrainte combine successivement deux procédés :

– Les vecteurs $V_{i,i+1 \bmod 4}$ entre les points P_i et P_{i+1} du quadrangle sont calculés pour $i \in \{1, ..., 4\}$. Les angles α_i entre deux vecteurs consécutifs sont alors calculés.

$$Q_{1234} \text{ est retenu si et seulement si } \alpha_i < 180^\circ \text{ pour } i \in \{1, ..., 4\}. \quad (2)$$

Ceci permet de retirer tous les quadrangles ayant des angles obtus ou trois points alignés.

– Le second calcul nous permet de supprimer des quadrangles trop proches d'une forme triangulaire, ce qui correspond à trois points sur les quatre alignés. Dans ce cas, l'insertion de données ferait apparaître des triangles trop disproportionnés. Un seuil de tolérance S_t est choisi afin que la proportion des deux triangles d'un quadrangle respecte une certaine valeur. Pour cela nous calculons la longueur des segments $\overline{P_{23}}$ et $\overline{P_{14}}$ ainsi que la distance D séparant les centres de ces deux segments. Un quadrangle est alors retenu pour l'insertion si et seulement si :

$$D \leq S_t \times \overline{P_{14}} \ et \ D \leq S_t \times \overline{P_{23}}. \tag{3}$$

Contrainte de recouvrement : la troisième contrainte concerne le recouvrement des quadrangles. A partir des quadrangles retenus avec les deux contraintes précédentes il peut en rester avec plusieurs sommets en commun. Lorsque des quadrangles possèdent au maximum un point en commun cela ne pose aucun problème et dans ce cas les quadrangles sont retenus pour l'insertion des données cachées. Par contre si deux quadrangles possèdent plus de un point en commun, la modification de la connexité sur un de ces quadrangles au moment de l'insertion de données, va perturber le second. C'est le cas par exemple des noeuds possédant plus de trois fils. Pour éviter ce problème de recouvrement un choix sera fait parmi l'ensemble des quadrangles se recouvrant pour n'en garder qu'un seul. Ce choix est fait de façon à garder le quadrangle qui affectera le moins le rendu visuel du modèle 3D lors de l'insertion d'un bit du message. Par conséquent le quadrangle le plus coplanaire sera choisi pour l'insertion des données. Finalement nous retenons uniquement les quadrangles satisfaisant les trois contraintes présentées. La méthode d'insertion présentée Section 3.1.3 est alors appliquée sur les quadrangles retenus. Cette contrainte limite le nombre maximum théorique de quadrangles marquables à $N/4$, N étant le nombre de points de l'objet 3D.

3.1.5 Extraction du message

Pour extraire le message caché, la première étape consiste à reconstruire l'ACM à partir des points du modèle. Pour cela, l'ACP est donc calculée sur les points afin de déterminer le point de départ de l'ACM à l'aide de la clef secrète. Comme notre méthode d'insertion ne modifie pas la position des points, l'ACM reconstruit est le même qu'à l'insertion. Dans la seconde étape il s'agit de retrouver les zones d'insertion, en utilisant les mêmes contraintes que pour l'insertion. Une fois ces deux étapes réalisées, il ne reste plus qu'à extraire le message de ces zones. Pour cela nous comparons la position de l'arête commune des deux triangles formant un quadrangle avec l'arête initiale de l'ACM. Si l'arête commune fait partie de l'ACM alors le bit lu est un 0, sinon c'est un 1 comme illustré Figure 4. Notons que la reconstruction de l'ACM nécessite l'utilisation d'une clef secrète protégeant l'information cachée. La Figure 7 illustre le schéma d'extraction du message. L'ACM de l'objet est unique mais les quadrangles sélectionnés pour l'insertion dépendent eux du sens de lecture de l'ACM. Si N est le nombre de points de l'objet 3D, la complexité de cette méthode est de N. Afin d'augmenter la complexité de la méthode, la clef peut être utilisée pour mélanger l'ordre des bits du message. La complexité est alors de $N \times 2^n$, avec $n \leq N$ le nombre de quadrangles utilisés pour l'insertion des données cachées.

3.2 Méthode d'insertion basée sur la projection de quadrangles

Cette seconde méthode d'insertion de données cachées est basée sur la projection des centres des zones d'insertion sur un axe clef. Comme dans la première méthode, les zones d'insertions sont

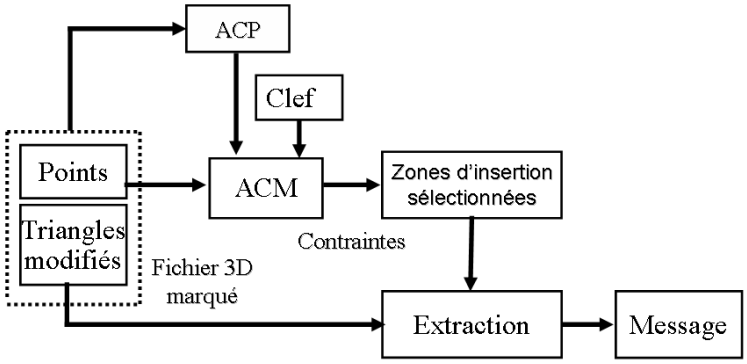


FIG. 7 – Présentation générale de la méthode d'extraction.

également des quadrangles. Cette méthode est composée de trois parties. La première partie est la sélection des zones qui vont nous servir à insérer des données cachées. Ces zones sont choisies selon plusieurs contraintes. La seconde partie est la synchronisation du message avec le modèle 3D. Pour cela nous nous appuyons sur un axe clef construit dans un repère invariant. Ce repère est construit à l'aide d'une analyse en composante principale (ACP) et d'un axe généré à partir d'une clef secrète. Le centre de chaque quadrangle choisi pour l'insertion est alors projeté sur l'axe. La troisième étape est l'insertion du message. Pour cela, comme dans la méthode précédente, la connexité des triangles des quadrangles sélectionnés est modifiée. La Figure 8 illustre le schéma général de la méthode. Dans la Section 3.2.1 nous présentons les contraintes supplémentaires utilisées pour sélectionner les zones où sera inséré le message. Nous détaillons dans la Section 3.2.2 comment synchroniser ces zones avec le modèle 3D. La Section 3.2.3 présente la méthode utilisée pour insérer le message dans le maillage 3D. Enfin nous montrons Section 3.2.4 comment extraire le message d'un fichier 3D marqué avec cette seconde méthode.

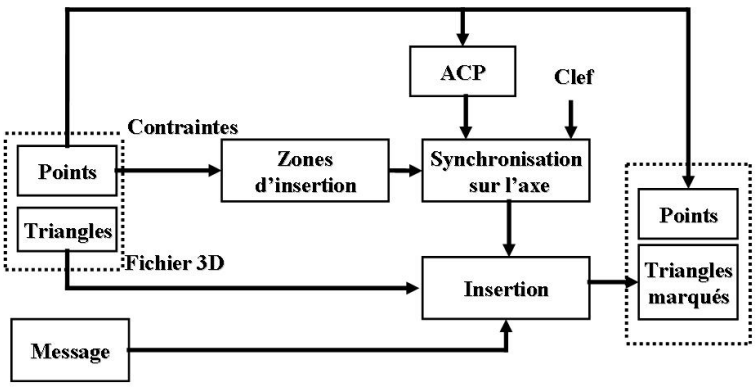
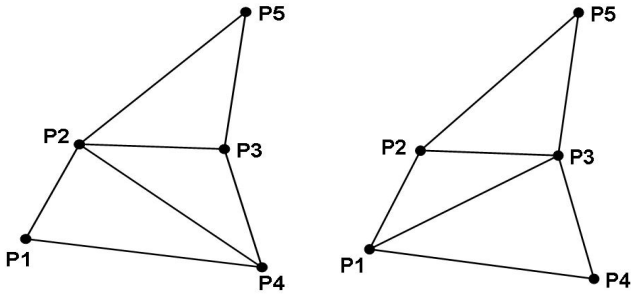


FIG. 8 – Présentation générale de la méthode d'insertion basée sur la projection des quadrangles.

3.2.1 Sélection des zones d'insertions

Comme défini Section 3.1.2 nous appelons quadrangle une zone formée par deux triangles du maillage ayant une arête commune. Ces quadrangles sont validés comme zones d'insertion dans la mesure où ils répondent à toutes les contraintes. Contrairement à la méthode précédente où les quadrangles étaient choisis par une connexité particulière de l'ACM, ici ce sont uniquement ces contraintes qui nous permettent de faire cette sélection. En fait dans cette seconde méthode, tous les quadrangles du maillage sont examinés afin de voir si ils répondent aux critères de convexité, coplanarité et recouvrement. Les trois critères de convexité, coplanarité et recouvrement sont les mêmes que dans la méthode précédente détaillée Section 3.1.4. Un quatrième critère est nécessaire pour cette méthode afin de sélectionner les quadrangles. Il s'agit de la contrainte de stabilité.

Stabilité : c'est une contrainte forte pour l'insertion de données basée sur la projection des quadrangles. En effet, le fait de modifier la connexité des triangles pour l'insertion de données engendre de nouveaux triangles et donc de nouveaux quadrangles. Ces nouveaux quadrangles sont susceptibles de répondre aux contraintes demandées pour l'insertion et donc de venir interférer avec les quadrangles choisis au préalable. Nous pouvons voir Figure 9 un exemple d'interférence. Au départ dans la Figure 9.a le quadrangle Q_{1234} choisi pour l'insertion est représenté par les triangles T_{124} et T_{234}. Nous remarquons que les triangles T_{234} et T_{235} forment également le quadrangle Q_{2534} qui ne répond pas au critère de convexité. La Figure 9.b représente la même zone après une insertion sur le quadrangle Q_{1234} avec changement de base commune. Nous constatons alors qu'il est possible de former le quadrangle Q_{1253} avec les triangles T_{123} et T_{253} qui répond aux critères de sélection. Ce quadrangle peut alors interférer avec le quadrangle précédemment sélectionné et nous empêcher de récupérer le bit du message. En fait l'insertion d'un bit du message dans un quadrangle peut générer un nouveau quadrangle répondant aux quatre contraintes. Dans ce cas, des ambiguïtés seraient rencontrées au moment de l'extraction. Pour qu'un quadrangle réponde au critère de stabilité il faut que tous les quadrangles formés lors de l'insertion d'un bit ne puissent pas interférer avec le quadrangle sélectionné pour l'insertion. Les quadrangles sont évalués à l'aide des critères de coplanarité et de convexité, nous choisirons alors pour l'insertion de données le quadrangle ayant les meilleurs critères dans la zone de perturbation. De cette façon nous nous assurons que le quadrangle est stable.



(a) (b)

FIG. 9 – a) Triangles initiaux, b) Triangles après insertion.

3.2.2 Synchronisation des quadrangles retenus

Pour réaliser la synchronisation nous nous appuyons sur un axe clef. Cet axe est généré à partir d'une clef en construisant un point et un vecteur. Cet axe est construit dans le repère de l'objet obtenu à partir d'une analyse en composante principale (ACP). L'ACP nous permet de rendre notre méthode d'insertion robuste aux transformations affines telles que les rotations, les translations ou les changements d'échelles. Pour synchroniser le message nous effectuons une projection orthogonale des centres des quadrangles retenus sur cet axe. Nous classons alors les points projetés en prenant le point initial de l'axe clef comme origine. Dans le cas où plusieurs quadrangles se projettent sur l'axe clef en un même point le quadrangle le plus proche de l'axe clef est sélectionné en premier. Par conséquent notre approche est robuste à une réorganisation des points dans le fichier 3D. L'ordonnancement de ces points sur l'axe donnera le sens d'insertion des bits du message sur les quadrangles. La sécurité du message repose sur l'axe clef mais aussi sur la taille de l'objet 3D à tatouer. Il est possible de retrouver le message en essayant toutes les permutations possibles des quadrangles servant à l'insertion des données cachées. L'information cachée peut être soit un 0 soit un 1 ce qui fait deux possibilités par quadrangle. Soit $n \leq N$, le nombre de quadrangles utilisés pour insérer l'information cachée, alors le nombre de permutations possibles des bits du message est le nombre de possibilités à la puissance du nombre de quadrangles : 2^n. La sécurité du message augmente donc avec le nombre de quadrangles utilisés pour l'insertion. La Figure 10 illustre la projection et la synchronisation de cinq quadrangles sur l'axe clef dans le repère de l'ACP de l'objet 3D.

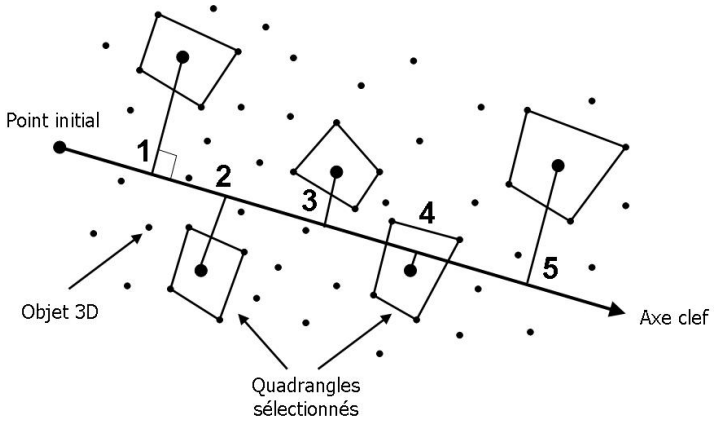


FIG. 10 – Synchronisation des projections des centres des quadrangles sur l'axe clef.

3.2.3 Insertion du message

Comme pour la synchronisation, l'insertion va également s'appuyer sur l'axe clef. Pour chaque quadrangle retenu nous projetons sur l'axe clef les quatre points formant ce quadrangle. Deux de ces quatre points appartiennent à la base commune aux deux triangles formant le quadrangle. Le code b est attribué à la projection des points appartenant à la base commune et le code a est attribué à la projection des deux autres points du quadrangle. Le sens de lecture de l'axe clef va nous permettre d'obtenir un ordre particulier sur les points a et b. Nous considérons que si un

point b est lu en premier sur l'axe clef alors le quadrangle contient un bit à 1 du message caché. Respectivement si un point a est lu en premier alors c'est un bit à 0 qui est caché. Pour changer la valeur du bit caché il suffit de changer la base commune aux deux triangles formant le quadrangle. La Figure 11 montre l'ensemble des solutions possibles pour réaliser une insertion ainsi que les codes correspondants. Les Figures 11.a montrent des quadrangles marqués à 0. Si nous souhaitons marquer ces quadrangles avec un bit à 1 il suffit d'inverser la base commune des deux triangles afin d'obtenir les configurations présentées Figures 11.b. Dans le cas particulier où un point de la base et un autre point du quadrangle se projettent en premier et au même endroit sur l'axe clef nous prenons alors en compte la distance séparant ces points de l'axe clef. Si le point le plus proche des deux est le point appartenant à la base commune alors le quadrangle est considéré comme porteur d'un bit à 1, sinon le quadrangle dissimule un bit à 0 comme présenté Figures 11.a3 et 11.b3.

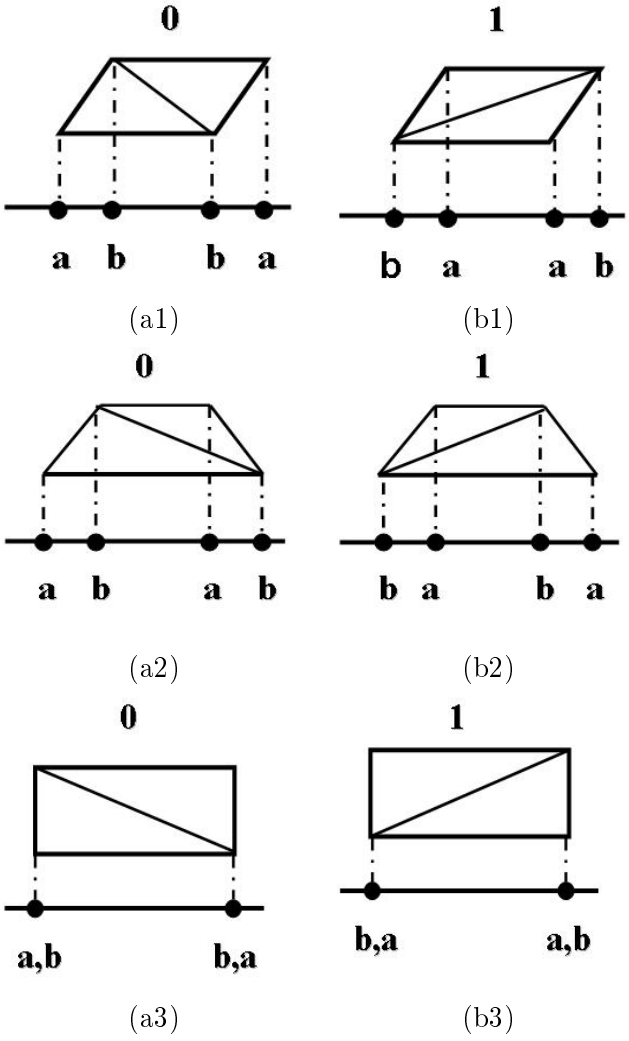


FIG. 11 – a1, a2, a3) Configuration permettant d'insérer un bit du message à 0, b1, b2, b3) Configuration permettant d'insérer un bit du message à 1.

3.2.4 Extraction du message

La première étape pour l'extraction du message est de rechercher les zones d'insertion répondant aux critères de convexité, coplanarité, stabilité et recouvrement. Ces zones sont ensuite synchronisées sur l'axe clef construit avec la clef et l'ACP réalisée sur l'objet 3D. Une fois les zones d'insertion synchronisées nous projetons les quatre points de chaque quadrangle retenu afin d'en extraire le code. Celui-ci va nous donner la valeur du bit inséré dans le quadrangle. La Figure 12 illustre le schéma d'extraction du message.

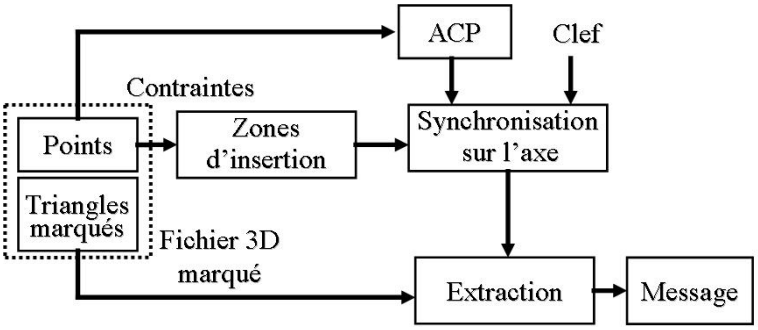


FIG. 12 – Présentation générale de la méthode d'extraction proposée.

4 Résultats expérimentaux

Dans cette section nous avons appliqué et analysé les deux méthodes d'insertion proposées sur douze objets 3D ayant un nombre de points et des caractéristiques différentes. Le Tableau 1 indique les caractéristiques des objets 3D utilisés.

Objets 3D	bunny	dragon	happy	chaussure homme
Nombre de points	453/1887/8146	5203/22982	7102/32316	12606
Nombre de triangles	948/3851/16301	11102/47794	15536/67240	25208
				

Objets 3D	dragon chinois	chaussure femme	horse	julius
Nombre de points	20184	31088	43039	48485
Nombre de triangles	40364	62168	85145	96966
				

TAB. 1 – Caractéristiques des objets 3D utilisés.

La Figure 13 représente le maillage triangulé du modèle Bunny de Stanford composé de 8146

points et 16301 triangles. Afin d'illustrer nos résultats de recherche de quadrangles, dans cette section nous utiliserons ce modèle. Les Figures 14.a et 14.b illustrent le résultat de la recherche de quadrangles avec les deux méthodes pour un seuil $S_c = 0.1$. Avec la méthode basée sur l'ACM, illustrée Figure 14.a, 150 quadrangles sont retenus pour l'insertion de données. La Figure 14.b illustre les 1491 quadrangles retenus pour l'insertion de données avec la méthode de projection.

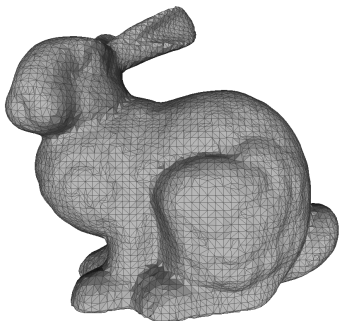
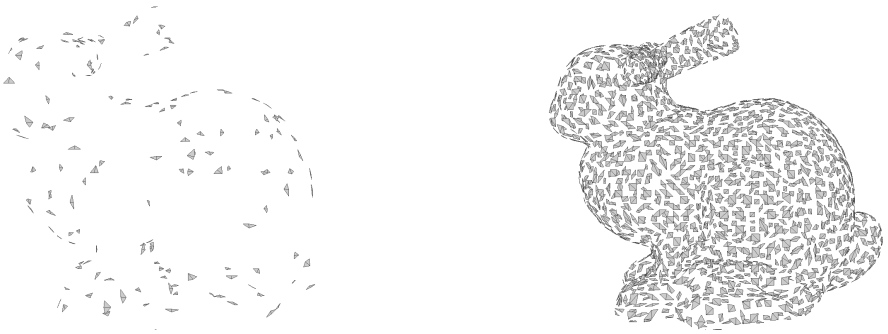


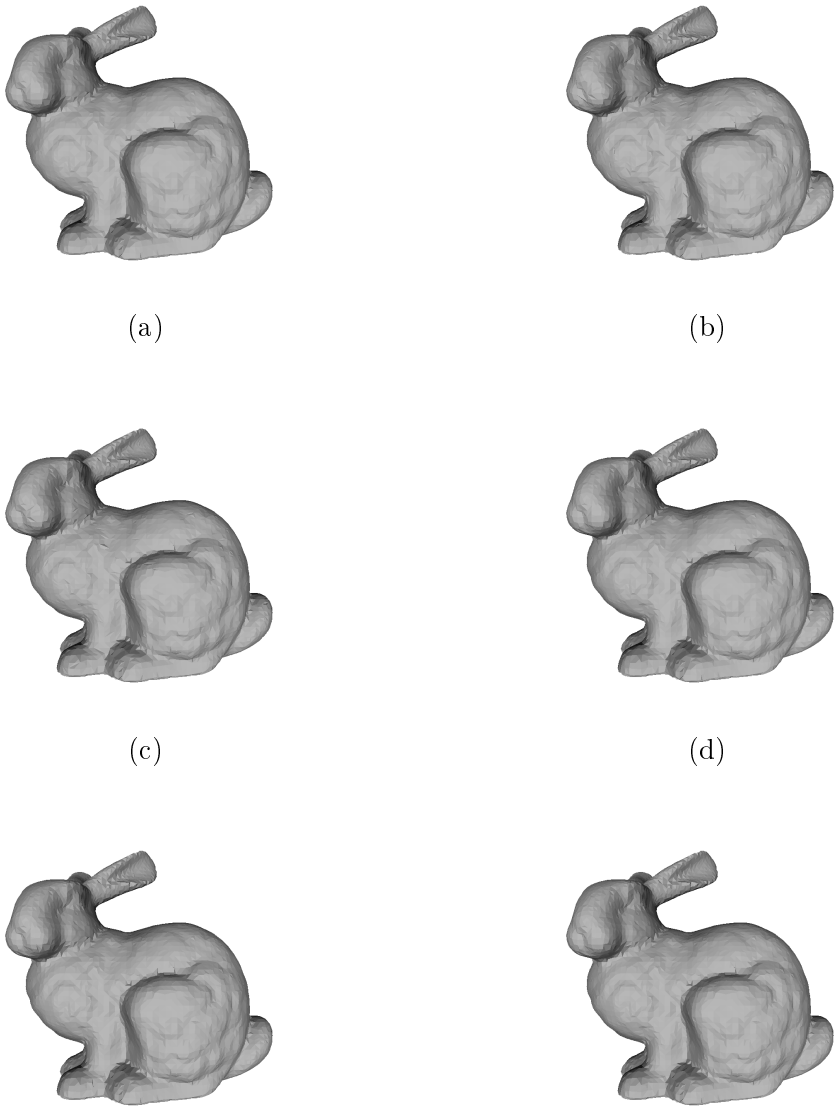
FIG. 13 – Maillage triangulé (8146 points, 16301 triangles).



(a) (b)

FIG. 14 – Sélection des quadrangles retenus sur le modèle bunny (8146 points, 16301 triangles) avec un seuil S_c de 0.1 : a) 150 quadrangles retenus avec la méthode ACM, b) 1491 quadrangles retenus avec la méthode par projection.

Afin d'illustrer les résultats obtenus avec les deux méthodes proposées, nous comparons l'objet original de la Figure 15.a avec les objets 3D marqués des Figures 15.b-15.f. Lorsque le seuil de coplanarité S_c est relativement élevé, quelle que soit la méthode utilisée, nous remarquons que les quadrangles qui ont été inversés pour réaliser l'insertion de données cachées commencent à devenir visibles dans certaines zones. A l'inverse lorsque le seuil de coplanarité diminue les quadrangles qui ont servi à l'insertion de données cachées sont peu visibles. Globalement, du fait de la conservation des positions initiales des sommets, la qualité visuelle des objets 3D est très bonne. Notons que pour cet objet 3D, avec la méthode basée sur l'ACM et un seuil de coplanarité $S_c = 0.001$ aucun



(a) (b)

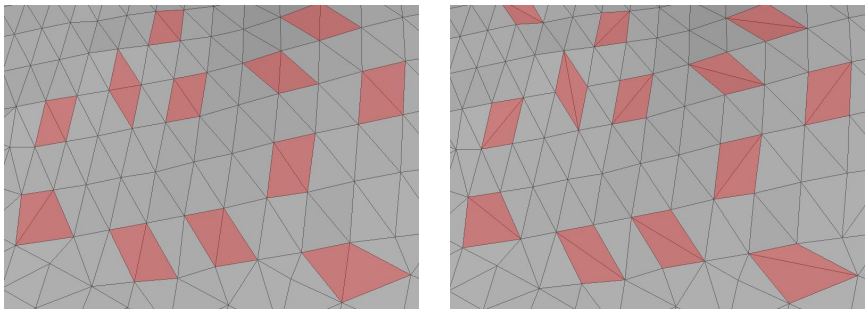
(c) (d)

(e) (f)

FIG. 15 – a) Objet 3D original, Bunny (8146 points), b), c) et d) Objet Bunny après insertion de données cachées avec la méthode par projection : b) Seuil de coplanarité $S_c = 0.1$ (1491 bits insérés), c) Seuil $S_c = 0.01$ (1193 bits insérés), d) Seuil de coplanarité $S_c = 0.001$ (177 bits insérés), e) et f) : Objet Bunny après insertion de données cachées avec la méthode basée sur l'ACM : e) Seuil de coplanarité $S_c = 0.1$ (150 bits insérés), f) Seuil de coplanarité $S_c = 0.01$ (63 bits insérés).

quadrangle marquable n'a été sélectionné. Cela peut s'expliquer du fait que la taille de cet objet est relativement petite en nombre de points.

Afin d'illustrer le résultat d'une insertion dans un maillage, nous avons effectué, Figure 16, un zoom sur une partie du maillage. La Figure 16.a montre un maillage avant insertion, la Figure 16.b montre le même maillage ou l'ensemble des bases communes des triangles composant les quadrangles retenus pour l'insertion a été inversé. Si nous considérons que les quadrangles retenus pour insérer les bits du message sont convexes et coplanaires, alors le changement de connexité effectué lors du marquage de ces quadrangles ne va pas modifier la surface du modèle 3D. Les critères à prendre en compte dans l'analyse des résultats sont la capacité d'insertion, l'analyse de l'erreur sur la surface engendrée par l'insertion du message caché et la résistance de nos méthodes aux bruits dus à des transformations.

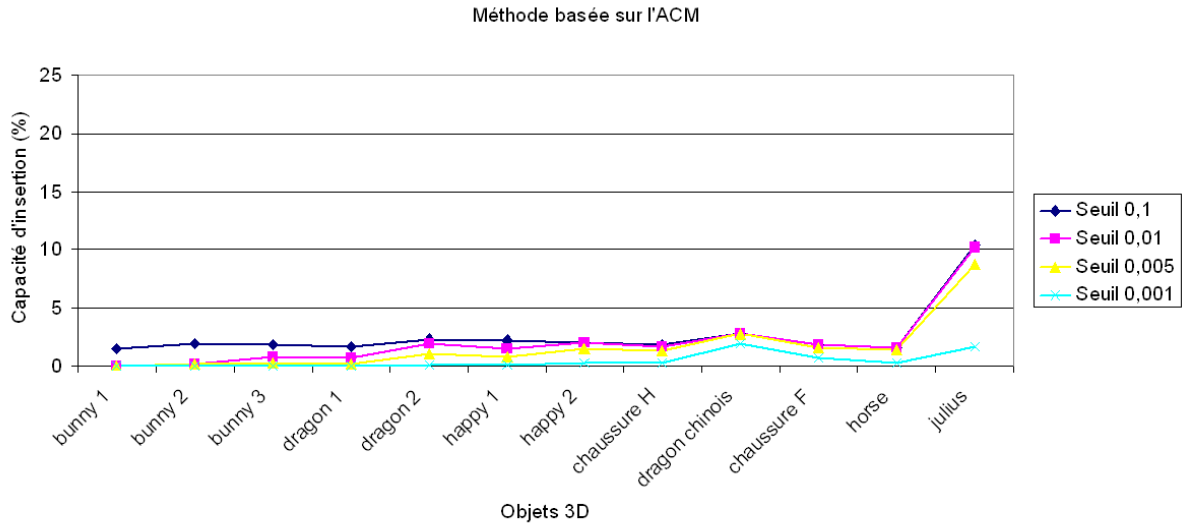


(a) (b)

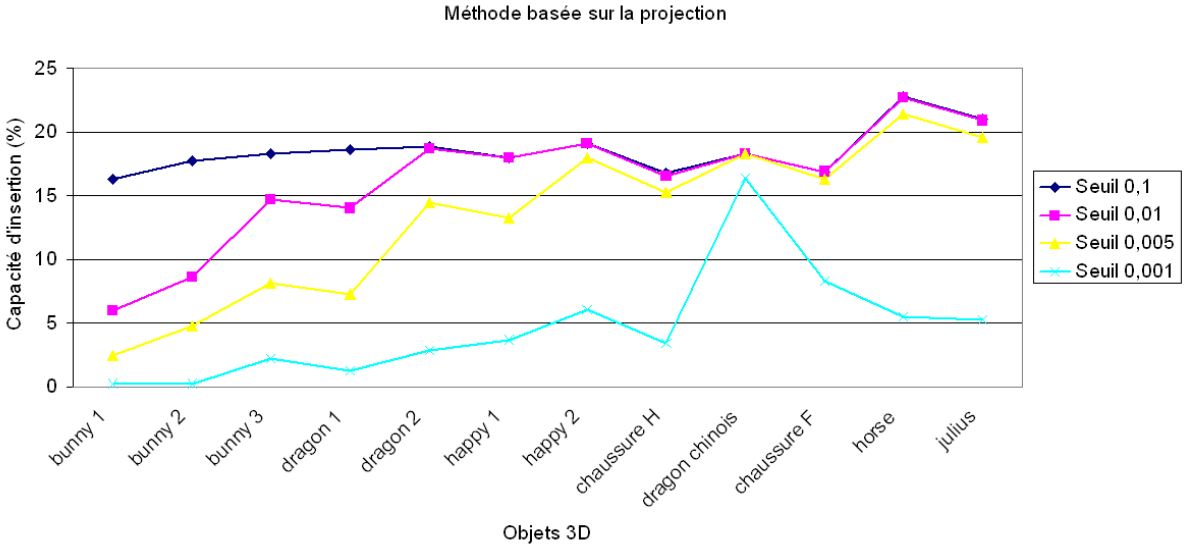
FIG. 16 – a) Zoom d'une partie du maillage avant insertion, b) Zoom de la même partie du maillage après une inversion de toutes les bases communes des triangles composant les quadrangles retenus.

Afin d'évaluer nos deux méthodes, le premier critère utilisé, calculé en bit par sommet, est la capacité d'insertion. Les courbes des Figures 17.a et 17.b illustrent les capacités d'insertion des deux méthodes proposées en fonction des modèles 3D présentés dans le Tableau 1, et cela pour plusieurs valeurs du seuil de coplanarité S_c. L'axe des abscisses représente le type d'objet 3D utilisé, globalement des plus petits aux plus grands, et l'axe des ordonnées indique la capacité d'insertion de la méthode en pourcentage. Globalement la capacité d'insertion augmente légèrement avec la taille de l'objet 3D. Lorsque que l'on augmente le seuil de coplanarité S_c cette différence de capacité entre les objets n'intervient plus. Nous constatons tout de suite en comparant ces deux courbes que, pour un seuil donné, la capacité d'insertion est plus grande pour la méthode basée sur la projection des quadrangles que pour la méthode basée sur l'ACM. Par exemple, pour l'objet 3D intitulé dragon2, composé de 5203 points, le nombre de bits insérables correspond à moins de 2% pour la méthode basée ACM avec un seuil $S_c = 0.01$ alors que le nombre de bits insérables correspond à presque 19% pour la méthode basée projection avec la même valeur de seuil. Nous pouvons remarquer également que pour un même objet, la variation de la valeur du seuil S_c a beaucoup plus d'impact sur la capacité d'insertion de la méthode basée projection que de la méthode basée ACM.

La Figure 18 illustre également ce critère en fonction du seuil de coplanarité S_c. L'axe des abscisses représente le seuil de coplanarité S_c utilisé pour la recherche des quadrangles. La courbe *ACM* représente la capacité d'insertion moyenne pour la méthode basée sur l'ACM et la courbe *Projection* représente la moyenne pour la méthode basée sur la projection. Les moyennes sont



(a)



(b)

FIG. 17 – a) Taux d'insertion avec la méthode basée sur l'ACM, b) Taux d'insertion avec la méthode basée sur la projection.

calculées à partir de l'ensemble des objets 3D présentés dans le Tableau 1. Nous pouvons remarquer que la capacité d'insertion augmente avec le seuil de coplanarité S_c. Cette courbe nous permet de confirmer également que le seuil de coplanarité S_c a plus d'influence sur la méthode basée sur la projection que sur la méthode basée sur l'ACM. La capacité d'insertion de la méthode basée sur la projection, est toujours supérieure à celle de la méthode basée sur l'ACM, quel que soit le nombre

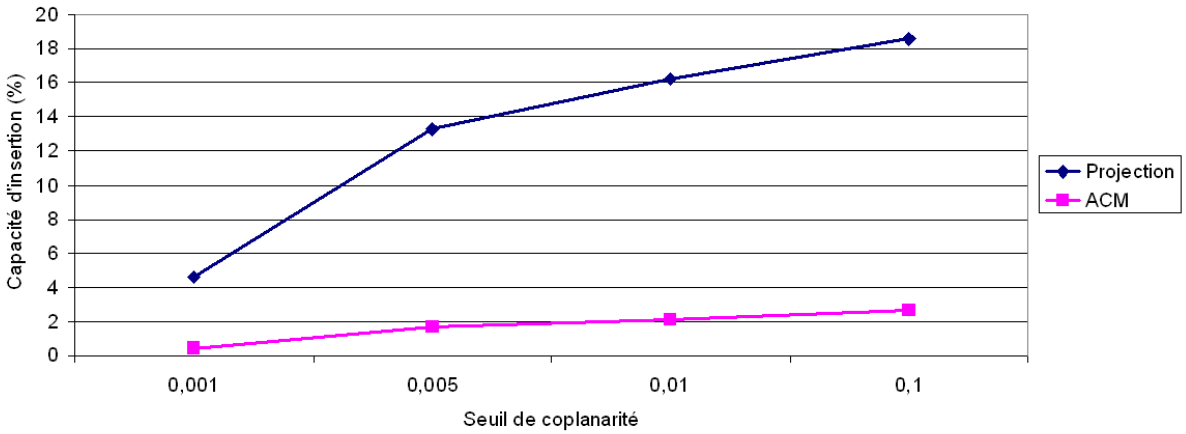


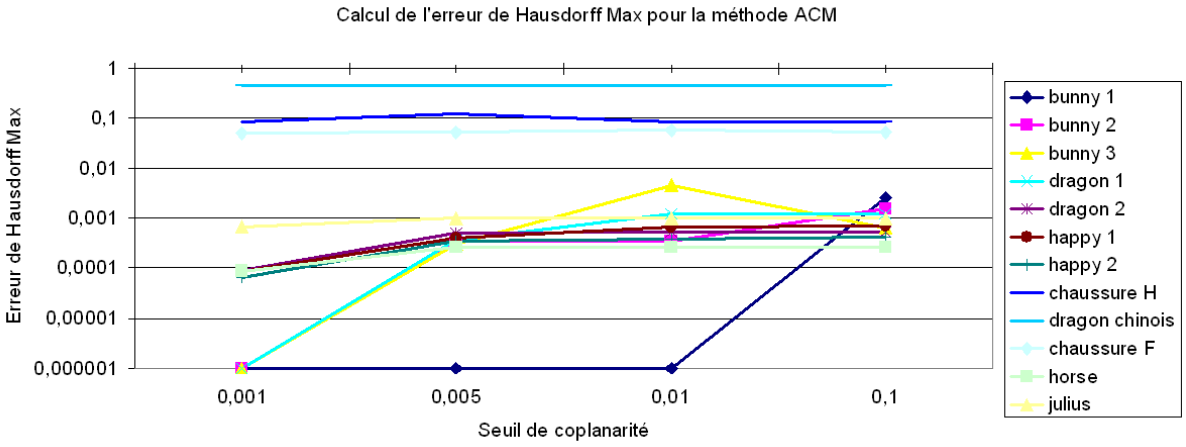
FIG. 18 – Evolution de la capacité d'insertion en fonction du seuil de coplanarité S_c.

de sommets du modèle 3D et le seuil S_c utilisé.

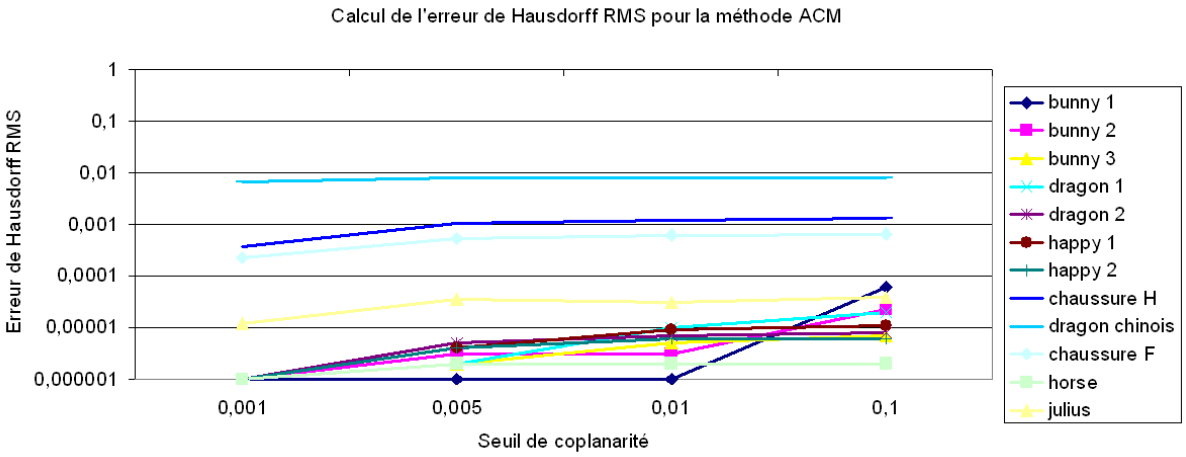
Le second critère d'analyse consiste à évaluer l'erreur faite sur la surface en fonction du seuil de coplanarité S_c et donc du nombre de bits dissimulés. Comme précisé précédemment, le fait d'augmenter ce seuil nous permet d'augmenter la capacité d'insertion, mais ceci est fait au détriment de la qualité visuelle de la surface. Pour évaluer l'erreur au niveau de la surface marquée par rapport à la surface de l'objet original, un calcul est effectué afin de quantifier l'erreur faite lors de l'insertion d'un bit dans un quadrangle. La méthode standard pour le calcul d'erreur est la distance géométrique de Hausdorff qui utilise la distance euclidienne entre deux points d'un maillage. De nombreuses méthodes ont été développées afin de calculer l'erreur entre deux surfaces. Alliez *et al.* se basent l'information volumique afin de réaliser le calcul d'erreur entre deux objets 3D [ALSS99] alors que Kim *et al.* utilisent un calcul de distance basé sur les courbures [KKK02]. Rondao-Alface et Macq utilisent une métrique basée sur l'estimation de la distorsion d'énergie [RM06] et Lavoué *et al.* mesurent la distorsion entre deux objets 3D en utilisant le concept de similarité structurelle [LDGD+06]. Pour effectuer le calcul d'erreur à partir de la distance géométrique de Hausdorff, nous avons utilisé l'algorithme proposé par le logiciel Metro [CRS98].

Les Figures 19 et 20 illustrent les valeurs des erreurs commises sur les surfaces en fonction du seuil de coplanarité S_c utilisé pour effectuer l'insertion de données cachées. Ce calcul d'erreur a été appliqué aux différents modèles 3D pour les deux méthodes d'insertion de données cachées. L'axe des abscisses représente le seuil de coplanarité S_c utilisé et l'axe des ordonnées représente l'erreur calculée sur la surface du modèle 3D. Pour la méthode basée sur l'ACM en fonction du seuil S_c, la Figure 19.a illustre les erreurs obtenues avec la méthode de Hausdorff MAX et la Figure 19.b illustre les erreurs obtenues avec la méthode de Hausdorff RMS. Pour la méthode basée sur la projection en fonction du seuil S_c, les Figures 20.a et 20.b illustrent respectivement les erreurs obtenues avec les méthodes de Hausdorff MAX et RMS. Dans tous les cas, nous constatons que, globalement, l'erreur augmente quand la valeur du seuil S_c augmente. Avec notre approche, nous remarquons que les deux mesures d'erreur (Hausdorff MAX et RMS) évoluent de la même manière quelle que soit la méthode d'insertion utilisée. Nous constatons également que l'erreur est différente en fonction des objets marqués et qu'elle est bien plus importante pour les trois objets *chaussure homme*, *chaussure*

femme et *dragon chinois*. Concernant les deux formes de chaussures, cela vient du fait que les formes initiales de ces objets sont composées de triangles fortement désorganisés comparés aux autres objets testés. Concernant le *dragon chinois*, c'est l'objet le plus complexe que nous avons testé dans le sens qu'il comporte de nombreuses courbures fortes proches de zones relativement planes qui ont été marquées.



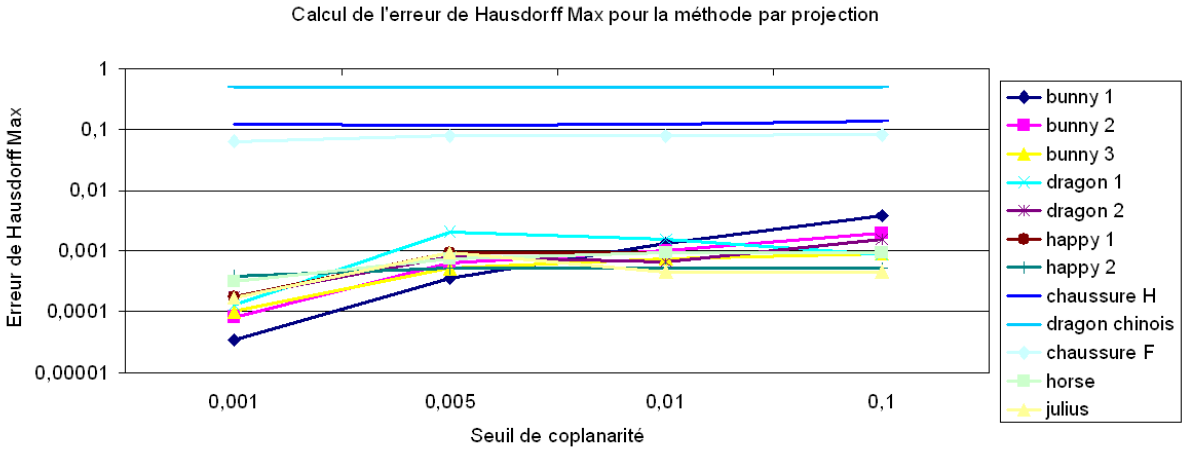
(a)



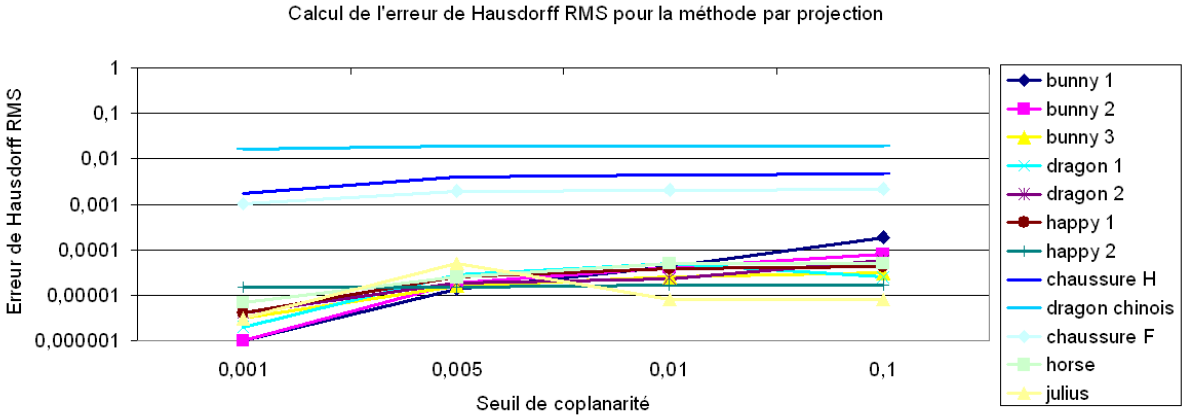
(b)

FIG. 19 – Pour la méthode basée sur l'ACM, en fonction du seuil S_c, erreurs calculées par : a) La méthode de Hausdorff Max, b) La méthode de Hausdorff RMS.

Le troisième critère permet d'évaluer la résistance au bruit des deux méthodes proposées. Les Figures 21 représentent la résistance de nos méthodes à l'ajout de bruit sur les sommets de l'objet 3D. Les Figures 21.a et 21.b représentent cette robustesse respectivement pour la méthode basée sur l'ACM et pour la méthode basée sur la projection. L'axe des abscisses représente l'amplitude du bruit ajouté aux sommets de l'objet et l'axe des ordonnées représente le pourcentage de quadrangles corrects détectés. Avec une amplitude de bruit faible les deux méthodes conservent 100%



(a)

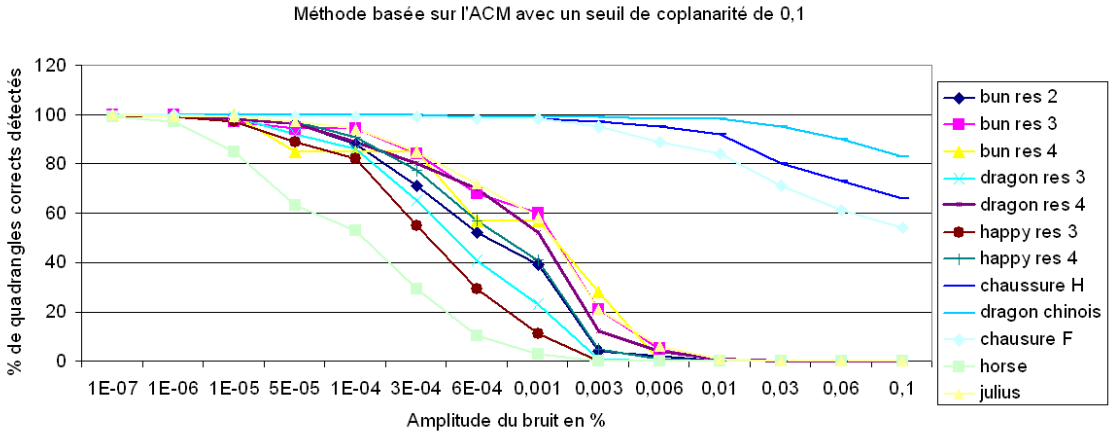


(b)

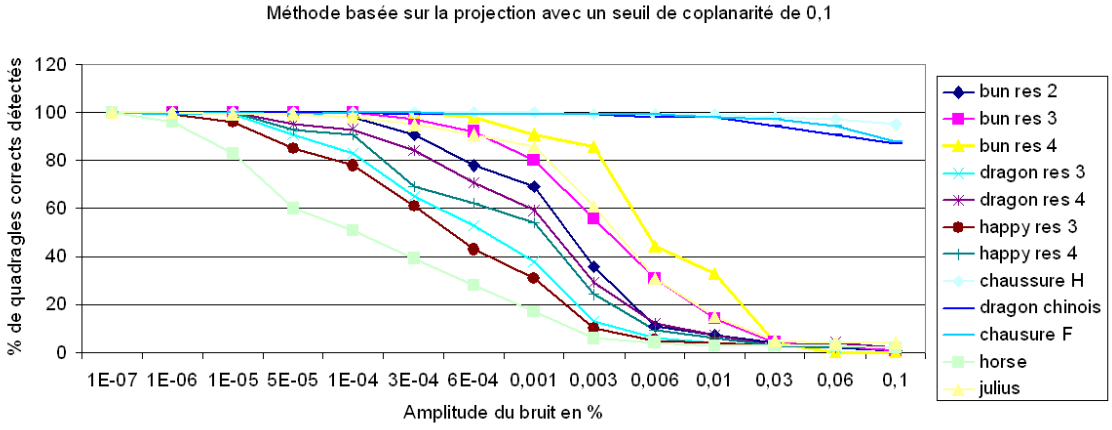
FIG. 20 – Pour la méthode basée sur la projection, en fonction du seuil S_c, erreurs calculées par : a) La méthode de Hausdorff Max, b) La méthode de Hausdorff RMS.

des quadrangles marqués. Par contre si l'amplitude du bruit augmente alors le nombre de quadrangles corrects détectés diminue assez rapidement dans l'ensemble sauf pour trois objets qui sont plus robustes au bruit. Il se trouve que ces trois objets sont ceux qui ont subit le plus de distorsions au moment de l'insertion. Tous les autres objets, étant plus lisses ou ayant des triangles mieux organisés (de tailles homogènes ou de même valence), ont été plus rapidement perturbés par un ajout de bruit avant l'extraction.

Afin d'analyser la sécurité et non plus la robustesse de nos méthodes, en plus de la complexité d'extraction du message augmentée du fait de l'utilisation d'une clef, nous avons effectué une analyse ROC (*Receiver Operating Characteristic*). Les courbes ROC présentées Figure 22 permettent d'illustrer la sensibilité et la spécificité de nos méthodes en calculant les faux positifs ainsi que les faux négatifs au moment de l'extraction. A l'extraction, pour un bruit donné, nous avons fait

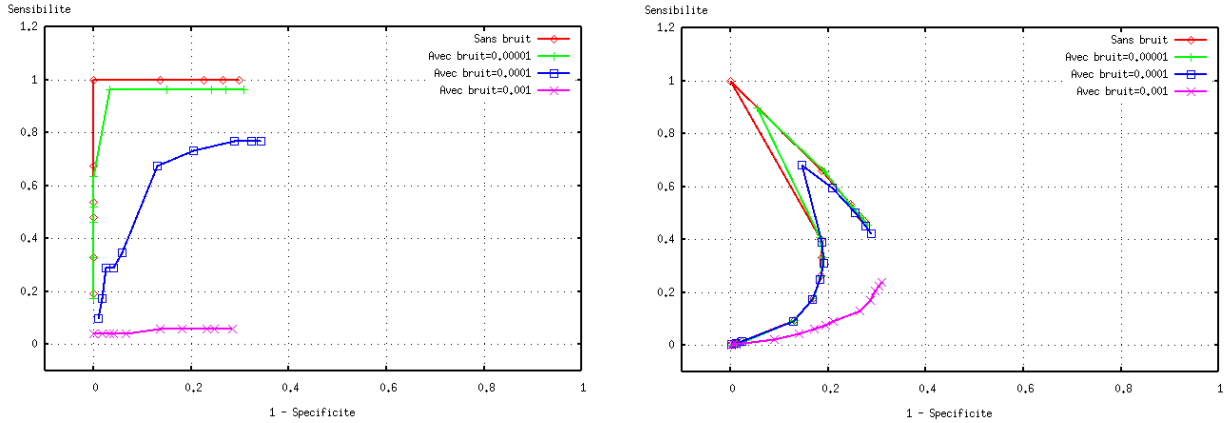


(a)



(b)

FIG. 21 – a) Influence du bruit sur la détection des quadrangles avec la méthode basée sur l'ACM, b) Influence du bruit sur la détection des quadrangles avec la méthode basée sur la projection.



(a) (b)

FIG. 22 – Courbes ROC : sensibilité en fonction de 1-spécificité, pour un seuil de coplanarité $S_c = 0.01$: a) Pour la méthode basée sur l'ACM, b) Pour la méthode basée sur la projection.

varier le seuil d'extraction S_c. Pour la méthode basée sur l'ACM, Figure 22.a, en dessous du seuil d'extraction utilisé pour l'insertion, nous constatons que notre méthode ne détectent pas de faux positifs (1-spécificité $= 0$), alors qu'avec un seuil d'extraction plus grand, la plupart des faux négatifs, apparus à cause du bruit, disparaissent (sensibilité $\simeq 1$). Dans le cas de la méthode basée sur la projection, Figure 22.b, nous constatons que dès que nous nous éloignons de la bonne valeur du seuil de coplanarité pour l'extraction, alors des faux positifs ainsi que des faux négatifs apparaissent. Ceci vient du fait que la contrainte de stabilité décrite Section 3.2.1 est fortement perturbée pour cette méthode. La méthode basée sur la projection est plus sensible et donc moins sure à l'extraction que la méthode basée sur l'ACM.

Expérimentalement, en fonction de la valeur du seuil S_c il est possible, avec la méthode basée sur la projection, d'obtenir une capacité d'insertion supérieure $N/10$ bits, si N est le nombre de sommets de l'objet 3D. Cette capacité d'insertion est petite comparée aux méthodes où par exemple $N-2$ bits sont insérés. Ceci est dû à une contrainte forte sur la position des sommets, puisque dans nos méthodes proposées la position des sommets de l'objet reste inchangée.

5 Conclusions et perspectives

Dans cet article nous avons présenté deux nouvelles méthodes d'insertion de données cachées dans des objets 3D. Les méthodes présentées ont la caractéristique de ne pas modifier la position des sommets de l'objet 3D. Contrairement aux méthodes précédentes appartenant à cette catégorie, notons que nos approches diffusent le message à cacher dans tout l'objet et ne rajoutent pas de points supplémentaires. Les deux méthodes présentées, bien que très proches proposent deux façons bien distinctes de synchroniser les zones d'insertion avec le modèle 3D. L'une se base sur la construction d'un ACM alors que l'autre utilise un axe clef ou sont projetées les zones d'insertion pour leur synchronisation. Les méthodes de marquage sont elles aussi différentes car liées à la synchronisation. La première méthode s'appuie donc sur l'ACM construit au préalable afin d'insérer les bits du

message alors que la seconde méthode s'appuie sur la projection des points des zones d'insertion sur l'axe clef. Ces méthodes, malgré leurs différences dans la synchronisation et le marquage, nous autorisent des modifications telles que les rotations, les translations et les mises à l'échelle uniforme. De plus, la modification des données directement dans le fichier marqué ne perturbe absolument pas l'extraction du message. Notons que ces deux méthodes sont totalement aveugles et de grande capacité, notamment la méthode basée sur la projection nous permet d'insérer jusqu'à 0.19 bit par sommet en fonction du seuil S_c choisi. Ces méthodes permettent donc d'insérer dans un fichier 3D, des méta-données de taille relativement importante sans augmenter la taille du fichier original tout en conservant la position initiale exacte des points 3D. En perspective, nous envisageons de fusionner ces deux méthodes en gardant la méthode de recherche des zones d'insertion basée sur les contraintes, comme sur la seconde méthode, mais en faisant le marquage à l'aide de l'ACM. Ceci nous permettra d'avoir un bon taux d'insertion en s'abstrayant de la recherche de connexités particulières de l'ACM tout en conservant la stabilité de celui ci pour réaliser le marquage. Nous envisageons également de développer une nouvelle méthode basée sur ces approches afin de résister à des changements géométriques importants.

Références

[ALSS99] P. Alliez, N. Laurent, H. Sanson, and F. Schmitt. Mesh Approximation Using a Volume-Based Metric. In *PG '99 : Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, page 292, Washington, DC, USA, 1999. IEEE Computer Society.

[AMC07] P.R. Alface, B. Macq, and F. Cayre. Blind and Robust Watermarking of 3D Models : How to Withstand the Cropping Attack ? In *International Conference on Image Processing, IEEE ICIP07, San Antonio, Texas, USA*, pages 465–468, 2007.

[BD07] J. Bennour and J.-L Dugelay. Toward a 3D Watermarking Benchmark. In *MMSP 2007, IEEE International workshop on multimedia signal processing, October 1-3, 2007, Chania, Greece*, Oct 2007.

[Ben99a] O. Benedens. Geometry-Based Watermarking of 3D Models. *IEEE Comput. Graph. Appl.*, 19(1) :46–55, 1999.

[Ben99b] O. Benedens. Two High Capacity Methods for Embedding Public Watermarks into 3d Polygonal Models. *Proceedings of the Multimedia and Security-Workshop at ACM Multimedia*, pages 95–99, 1999.

[Ben00] O. Benedens. Affine Invariant Watermarks for 3D Polygonal and NURBS Based Models. In *ISW '00 : Proceedings of the Third International Workshop on Information Security*, pages 15–29, London, UK, 2000. Springer-Verlag.

[Bor06] A. G. Bors. Watermarking Mesh-Based Representations of 3-D Objects Using Local Moments. *IEEE Trans. on Image Processing*, 15(3) :687–701, mar 2006.

[CRS98] P. Cignoni, C. Rocchini, and R. Scopigno. Metro : Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2) :167–174, 1998.

[GSS99] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer gra-*

phics and interactive techniques, pages 325–334, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[ICA02] S. Ichikawa, H. Chiyama, and K. Akabane. Redundancy in 3D Polygon Models and Its Application to Digital Signature. In *WSCG*, pages 225–232, 2002.

[KDK98] S. Kanai, H. Date, and T. Kishinami. Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition. *Proc. int. Workshop Gemetric Modeling : Fondamentals and Applications*, pages 296–307, Decembre 1998.

[Ker83] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, 9 :5–38, 1883.

[KKK02] S.-J Kim, S.-K Kim, and C.-H Kim. Discrete Differential Error Metric for Surface Simplification. In *PG '02 : Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 276, Washington, DC, USA, 2002. IEEE Computer Society.

[Kru56] J. B. Kruskal. On the Shortest Spanning Tree of a Graph and the Traveling Saleman Problem. *Proc of the American Mathematical Society*, 7 :48–50, 1956.

[LCK02] J. J. Lee, N. I. Cho, and J.-W Kim. Watermarking for 3D NURBS graphic data. In *Proceedings of 2002 IEEE Workshop on Multimedia Signal Processing, 9-11 December 2002, St. Thomas, Virgin Islands, USA*, pages 304–307, 2002.

[LDGD+06] G. Lavoué, E. Drelie Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi. Perceptually driven 3D distance metrics with application to watermarking. In *SPIE Applications of Digital Image Processing XXIX*, San Diego, august 2006.

[MC03] B. Macq and F. Cayre. Data Hiding on 3D Triangle Meshes. *IEEE Transactions on Signal Processing*, 51(4) :939–949, 2003.

[MSI01] X. Mao, M. Shiba, and A. Imamiya. Watermarking 3D geometric models through triangle subdivision. In P. W. Wong and E. J. Delp, editors, *Proc. SPIE Vol. 4314, p. 253-260, Security and Watermarking of Multimedia Contents III, Ping Wah Wong ; Edward J. Delp ; Eds.*, volume 4314 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 253–260, August 2001.

[OMA97] R. Ohbuchi, H. Masuda, and M. Aono. Watermaking Three-Dimensional Polygonal Models. In *MULTIMEDIA '97 : Proceedings of the fifth ACM international conference on Multimedia*, pages 261–272, New York, NY, USA, 1997. ACM Press.

[OMT02] R. Ohbuchi, A. Mukaiyama, and S. Takahashi. A Frequency-Domain Approach to Watermarking 3D Shapes. *Computer Graphics Forum*, 21(3) :373–382, 2002.

[PHF99] E. Praun, H. Hoppe, and A. Finkelstein. Robust Mesh Watermarking. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 49–56, Los Angeles, 1999. Addison Wesley Longman.

[Pri57] R. C. Prim. Shortest Connection Networks and some Generalisations. *In : Bell System Technical Journal*, 36 :1389–1401, 1957.

[RAM07] P. Rondao-Alface and B. Macq. From 3D Mesh Data Hiding to 3D Shape Blind and Robust Watermarking : A Survey. *LNCS Transactions on Data Hiding and Multimedia Security*, 2 :99–115, 2007.

[RM06] P. Rondao Alface and B. Macq. Shape quality measurement for 3D watermarking schemes. In E. J. Delp, III and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VIII. Edited by Delp, Edward J., III; Wong, Ping Wah. Proceedings of the SPIE, Volume 6072, pp. 622-634 (2006).*, volume 6072 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 622–634, February 2006.

[Wag00] M. G Wagner. Robust Watermarking of Polygonal Meshes. In *GMP '00 : Proceedings of the Geometric Modeling and Processing 2000*, pages 201–208, Washington, DC, USA, 2000. IEEE Computer Society.

[WLDB07a] K. Wang, G. Lavoué, F. Denis, and A. Baskurt. Three-Dimensional Meshes Watermarking : Review and Attack-Centric Investigation. In *9^{th} International Workshop on Information Hiding*, pages 50–64, Saint Malo, France, 2007.

[WLDB07b] K. Wang, G. Lavoué, F. Denis, and A. Baskurt. Hierarchical Blind Watermarking of 3D Triangular Meshes. In *Proc. of the IEEE International Conference on Multimedia & Expo*, pages 1235–1238. IEEE, July 2007.

[YPSZ01] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computers and Graphics*, 25 :409–420, June 2001.