

# Auto-similarité pour la compression haute précision des surfaces échantillonnées

Julie Digne<sup>1</sup> Raphaëlle Chaine<sup>1</sup> Sébastien Valette<sup>2</sup>

<sup>1</sup>Université Lyon 1; LIRIS; CNRS UMR5205

<sup>2</sup>INSA-Lyon; CREATIS; CNRS UMR5220

---

## Résumé

*La plupart des objets qui nous entourent sont caractérisés par une forte auto-similarité de leur surface, qui diffère suivant qu'il s'agisse d'une œuvre d'art en pierre ou en bronze ou bien d'un objet mécanique. Cette propriété est liée à l'essence même des formes naturelles dont l'homme s'inspire, mais aussi à leur processus de fabrication, comme la régularité du geste du sculpteur ou de la machine-outil, ou bien les mécanismes d'érosion. Dans cet article, nous nous proposons d'exploiter l'auto-similarité sous-jacente à chaque forme, pour comprimer les surfaces échantillonnées avec une haute précision, pouvant produire des nuages de plusieurs millions de points. Notre approche rééchantillonne la forme de manière à mettre en évidence l'auto-similarité de sa surface, mais demeure respectueuse de la forme originale et de la précision du scanner. L'auto-similarité peut alors être utilisée pour créer un dictionnaire adéquat, sur lequel les voisinages locaux seront représentés de manière parcimonieuse, offrant ainsi la perspective d'une représentation économique de la surface globale.*

*Nous illustrons la validité de notre approche sur plusieurs nuages de points issus de la numérisation de statues ou d'objets mécaniques, mais aussi de scènes urbaines. De plus, nous montrons que notre approche permet d'obtenir un filtrage du bruit dont l'amplitude est plus petite que la précision du scanner.*

*Cette communication est une traduction de [DCV14].*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object ModelingCurve, surface, solid, and object representations ImageProcessing and Computer Vision [I.4.2]: Compression (Coding)Approximate methods

---

## 1. Introduction

Les techniques d'acquisition laser ont connu un essor particulièrement important au cours de ces dernières années, permettant de numériser des formes très variées avec des nuages pouvant atteindre plusieurs millions de points correspondant à une précision très fine. Cela entraîne malheureusement des besoins de plus en plus élevés en termes de capacité de stockage. Comme il n'y a pas vraiment de technique de compression qui reste efficace et maniable à cette échelle là, ce sont généralement des représentations plus grossières qui sont utilisées, avec une perte de précision.

Par ailleurs, les dispositifs d'acquisition laser sont généralement fournis avec des suites logicielles qui exploitent la proximité et la redondance locale des échantillons, pour reconstruire une approximation polygonale de l'ensemble des points. Cela coïncide avec l'idée qu'un triangle peut avan-

tageusement représenter un ensemble de points. Par conséquent, les efforts en termes de compression se sont principalement focalisés sur les maillages.

Pourtant, certaines applications n'ont besoin que du nuage de points et pas d'un maillage. Par exemple, les nuages de points denses se prêtent particulièrement à certaines opérations comme le recalage. Il arrive notamment qu'une campagne d'acquisition résulte dans la numérisation incomplète d'une surface et il est alors prématuré de commencer à reconstruire un maillage en lieu et place des données d'origine. Par ailleurs, les opérations de maillage global d'une surface sont souvent trop imprécises et lissent parfois exagérément les données. C'est pourquoi il est nécessaire de mettre en œuvre des méthodes de compression plus respectueuses de la précision du dispositif d'acquisition.

Dans cet article, nous proposons une solution pour la com-



**Figure 1:** *Les Amoureux de Bordeaux* (15.8 millions de points). En exploitant l'auto-similarité du modèle, nous arrivons à compresser cette représentation sur seulement 1.15Mo. Le modèle résultant (à droite) est très proche de l'original (à gauche), puisque l'erreur de reconstruction est moins élevée que la précision du scanner (0.02mm) pour 99.14% des points en entrée.

pression des surfaces représentées de manière brute par des nuages de points non organisés. À l'inverse des maillages, les nuages de points en entrée correspondent à un échantillonnage de la forme, sans aucune interprétation. Tirant parti de cette observation, l'approche de compression que nous proposons modifie légèrement l'échantillonnage de la surface. Ces changements sont très locaux et ne servent qu'à favoriser les stratégies de compression, tout en maîtrisant la tolérance, avec une distorsion inférieure à la précision en entrée.

L'échantillonnage produit est équivalent à celui en entrée. La seule raison pour laquelle nous modifions localement cet échantillonnage est que nous voulons pouvoir mettre en évidence l'auto-similarité inhérente aux formes 3D et l'exploiter à des fins de compression. L'analyse discrète que nous mettons en œuvre nécessite en effet que les différentes zones d'un même objet soit représentées avec un motif d'échantillonnage comparable. Cependant, notre but est de nous affranchir de tout modèle local contraignant, comme c'est parfois le cas avec certaines techniques de visualisation de surfaces échantillonnées (Q-Splat [RL00], MLS [ABCO\*01]). Au contraire, nous restons toujours proches des données (forme et échantillonnage), en supposant néanmoins que tout point est caractérisé par un voisinage de la surface, à l'échelle où l'auto-similarité est analysée. La technique de compression qui en résulte reste fidèle aux données et s'avère efficace puisqu'elle se comporte mieux que les algorithmes existants pour compresser les nuages de points échantillonnés sur des surfaces. La méthode n'est en revanche pas sans perte, à proprement parler, puisque le nuage de point initial ne peut être restauré. Pourtant, la distorsion induite par la compression peut être ramenée en dessous de

la précision du relevé laser, comme nous allons l'illustrer expérimentalement.

Pour résumer, nos contributions sont les suivantes :

- Nous proposons une nouvelle technique d'encodage des nuages de points, en exploitant l'auto-similarité présente dans les formes, mais en demeurant respectueux de la précision du scanner.
- Cette compression est réalisée à une résolution inférieure à celle du scanner, permettant un contrôle accru sur la résolution du nuage de points.
- Le nuage de points fourni en entrée peut être accompagné d'une information de normale orientée ou non.

La figure 1 illustre la différence entre le nuage de points d'origine et celui résultant de sa compression / décompression. Notre méthode permet d'atteindre un débit d'à peine 0.59 bits par point (bpp) pour ce modèle, tout en maintenant l'erreur de reconstruction au dessous de la précision du scanner, pour 99.14% des points. Par ailleurs, on observe un filtrage du bruit et des artéfacts de consolidation.

L'article est organisé de la manière suivante : la section 2 revient sur les principaux travaux en compression 3D et en analyse d'auto-similarité. La section 3 présente quant à elle les hypothèses pour lesquelles notre approche est valide. La description respective des étapes de compression et de décompression fait l'objet des sections 4 et 5. Enfin, la section 6 présente des résultats quantitatifs et qualitatifs ainsi que des comparaisons.

## 2. Etat de l'art

### 2.1. Compression 3D

Le problème de la compression d'objets 3D a été très étudié durant ces vingt dernières années. Nous pouvons classer les approches de compression en trois groupes : compression de nuages de points, compression de maillages ou compression de formes. Les approches de compression de nuages de points utilisent généralement la quantification des coordonnées des points par partition récursive de l'espace [SK06, GD02, HPKG06, SPS12]. Les points à encoder sont insérés dans une structure de donnée partitionnant l'espace (par exemple un octree ou un kd-arbre) jusqu'à un certain niveau de profondeur, et tous les points contenus dans une même cellule sont remplacés par le centre de la cellule.

D'autres approches codent la forme représentée par le nuage de points et utilisent cet encodage pour générer un nouveau nuage à la décompression. Kalaiah et al. [KV05] définissent une hiérarchie de niveaux de détails en faisant l'Analyse en Composantes Principales (ACP) du nuage et séparant le nuage de points le long de l'axe correspondant à la plus grande variation. Les points finaux sont reconstruits par échantillonnage Gaussien. Schnabel et al. [SMK08] segmentent le nuage de points en entrée en différentes formes canoniques. Chaque partie est ensuite encodée efficacement

en utilisant la quantification vectorielle. Notre approche utilise aussi des cartes de hauteur, mais aucune segmentation ou modèle de regression n'est nécessaire. Avec notre approche, la forme définit elle-même son espace d'analyse. Une compression basée sur l'auto-similarité a été proposée dans [HMHB08], et nous reviendrons sur cette approche dans les sections suivantes. Notre approche encode aussi les formes et génère un nouveau nuage de points au décodage.

La compression de maillage a été initiée par Deering [Dee95], et a été très explorée avec les années. La motivation derrière la compression de maillages est que certaines applications ont besoin de la connectivité originale, par exemple pour des simulations, ou dans les cas où des attributs sur les sommets ou triangles doivent être stockés. Le codage basé valence [TG98] est très proche de l'optimal pour la compression mono-résolution. Les approches progressives telles que [AD01, MCAH12] raffinent un maillage de base jusqu'à ce que la résolution originale soit atteinte. Karni & Gotsman [KG00] décomposent la géométrie du maillage sur une base spectrale, et le coût de la décomposition spectrale est diminué par partitionnement du maillage original. Des approches de partitionnement de l'espace ont aussi été proposées [GD02, PK05] avec la possibilité de comprimer des maillages à topologie arbitraire. Malheureusement, ces approches donnent de moins bons résultats à faible débit.

Si seule la forme est importante à transmettre, sans contrainte sur l'échantillonnage ou la connectivité, les approches de remaillage obtiennent les meilleures performances. Avant compression, le maillage en entrée est remaillé en un maillage structuré, par exemple un maillage avec connectivité de subdivision. L'étape de remaillage présente l'avantage de supprimer l'information paramétrique dans le maillage, information qui ne contribue pas à la représentation de la forme. Ensuite, la transformée en ondelettes [KSS00, GVSS00, GGH02] et en bandelettes [PM05] exploitent la présence de parties lisses dans le maillage et obtiennent de bonnes performances à faible débit. Malheureusement, peu d'approches robustes de remaillage structuré existent [LSS\*98] et le traitement de maillages complexes peut s'avérer problématique.

## 2.2. Auto-similarité des images

L'auto-similarité de signaux mesurés est un domaine en pleine expansion : la recherche en traitement du signal et de l'image a énormément progressé en prenant en compte l'auto-similarité des mesures.

En traitement de l'image, l'idée trouve sa source dans l'algorithme des moyennes non locales [BCM05] : au lieu de débruiter un pixel en exploitant les pixels voisins, ce débruitage s'effectue en tenant compte de tous les pixels de l'image qui lui sont similaires. La similarité entre les pixels est calculée en comparant les patches locaux autour d'eux. L'idée sous-jacente est que des pixels loin de la zone considérée

peuvent contenir de l'information qui aideront à la traiter, à cause de l'auto-similarité naturelle de l'image.

Cette idée a donné lieu à de nombreux travaux s'appuyant sur des patches pour résoudre de nombreux problèmes comme le débruitage, la super-résolution ou la complétion de données manquantes. Récemment l'algorithme Patch Match a rendu cette recherche de patch similaire très rapide.

Par ailleurs, le développement de la théorie du compressive sensing [CRT06] a mis en évidence l'existence d'espaces dans lesquels les signaux sont représentés de manière parcimonieuse et qui sont particulièrement adaptés à leur traitement. En particulier, [AEB06] introduit l'idée de concevoir des dictionnaires sur lesquels chaque patch de l'image serait représenté comme une combinaison linéaire sparse des atomes du dictionnaire. Cette méthode sera détaillée plus loin.

À la suite de son succès en traitement de l'image, cette idée a également été introduite pour les surfaces définies par des maillages ou des nuages de points.

## 2.3. Auto-similarité des surfaces

L'auto-similarité des surfaces a été principalement exploitée pour le débruitage des surfaces : le filtre non local a été adapté pour les maillages [YBS06] ou les nuages de points [AGDL09, Dig12].

Ce concept a également servi à définir une variante des Point Set Surfaces plus robuste au bruit [GAB12]. L'utilisation de l'auto-similarité des surfaces ne se limite pas au débruitage. Par exemple, cela peut servir à détecter des symétries ou des structures répétées dans les surfaces [MGP06], [PMW\*08], [MPWC12]. La similarité peut aussi être analysée entre des surfaces [BLSC\*11].

Le travail le plus proche de notre méthode est sans doute [ZSW\*10] où l'auto-similarité est utilisé pour consolider des scans de scènes urbaines.

À notre connaissance, l'auto-similarité n'a été utilisée qu'une fois pour la compression et dans un cadre bien différent du notre [HMHB08]. Après la décomposition de la surface en patches, ceux-ci sont segmentés en clusters selon leur similarité et sont, lors de la décompression, remplacés par le représentant du cluster. Cette décomposition se fait par k-moyennes. Notre approche est plus flexible : chaque patch est représenté comme une combinaison linéaire parcimonieuse sur un dictionnaire permettant une meilleure représentation de la variété des patches et donc une meilleure résolution finale. Nous comparerons notre approche avec cette méthode et montrerons un gain numérique et visuel net.

## 2.4. K-SVD

L'algorithme du K-SVD est une méthode pour représenter des signaux discrets finis comme des combinaisons li-

néaires parcimonieuses sur un dictionnaire ad-hoc. Il a été introduit dans [AEB06] et a été utilisé depuis pour de nombreuses applications comme le débruitage [EA06] et l'analyse d'images [MBP\*09].

Soit  $Y$  une matrice  $k \times n$ , dont les colonnes sont les  $n$  signaux  $(y_i)_{i=1, \dots, n}$ , représentés chacun par  $k$  échantillons. L'idée est de trouver un dictionnaire  $D$ , composé de  $d$  éléments, sur lequel chaque signal  $y_i$  est représenté comme une combinaison linéaire des éléments  $d_j$  du dictionnaire.

Autrement dit, on peut trouver un vecteur de coefficients  $x_i$  tels que  $y_i = D \cdot x_i$ . Soit  $X$  la matrice  $d \times n$  dont les colonnes sont les  $x_i$ ,  $X$  et  $D$  ( $k \times d$ ) sont alors calculées par minimisation de :

$$\min_{D, X} \|Y - DX\| \quad \text{s.t.} \quad \forall i, \|x_i\|_0 \leq T_0$$

$\|x_i\|_0$  représente le nombre d'éléments non nuls de  $x_i$ . Cette application est parfois appelé norme  $L^0$ , bien que cela ne soit pas une norme (elle n'est pas homogène). Cela mesure la parcimonie de la décomposition (une décomposition est dite parcimonieuse si de nombreux coefficients sont nuls et donc la "norme  $L^0$ " est faible).  $T_0$  contraint le nombre de coefficients non nuls.

L'algorithme K-SVD alterne entre trouver la meilleure représentation parcimonieuse pour  $Y$  sur  $D$  en utilisant un algorithme de poursuite (orthogonal matching pursuit) et mise à jour du dictionnaire. La mise à jour du dictionnaire consiste à considérer itérativement chaque atome  $d_k$  de l'étape précédente : on sélectionne les échantillons qui utilisent  $d_k$  dans leur représentation et on calcule l'erreur  $E_k^R$  de ces échantillons commise en les représentant sur  $D \setminus d_k$ . Ensuite,  $E_k^R$  est décomposée par Décomposition en Valeurs Singulières  $E_k^R = U \Delta V^T$ . L'atome  $d_k$  mis à jour correspondant à la première colonne de  $U$  et les coefficients mis à jour pour ce nouvel atome correspondent à la première colonne de  $V$  multipliée par  $\Delta(1, 1)$  (première valeur singulière). Ce processus itératif nécessite un dictionnaire initial qui correspond à un sous-ensemble aléatoire des signaux d'entrée  $(y_i)_{i=1 \dots N}$ . Dans ce papier nous utilisons une version rapide de cet algorithme de complexité  $O(2n^2d)$ , introduit dans [RZE08].

### 3. Hypothèses de validité de l'approche

Notre algorithme s'applique à un nuage de points  $\mathcal{P}$  qui doit être suffisamment dense pour représenter, sans ambiguïté, un échantillonnage d'une surface interpolante  $\mathcal{M}$  à une résolution fixée plus large que la densité de l'échantillonnage. Il peut traiter des nuages de points sans normales fournies, puisqu'il est toujours possible d'estimer la direction même inexacte de ces normales, à partir de voisinages locaux. En revanche, si les données d'entrée sont fournies avec des normales, il n'est alors plus nécessaire de les estimer, puisque les originales peuvent être utilisées directement.

L'algorithme nécessite également un rayon  $R$  qui correspond à l'échelle à laquelle l'auto-similarité de la surface doit être analysée. Notre approche s'appuie sur des hypothèses précises sur  $R$  par rapport au nuage de points fourni en entrée.

- *Condition topologique* :  $R$  doit être suffisamment petit pour permettre à  $\mathcal{P}$  d'être recouvert par un ensemble de  $R$ -voisinages centrés sur un sous-ensemble de points dits germes. De plus, chaque  $R$ -voisinage doit délimiter un disque topologique sur la surface sous-jacente  $\mathcal{M}$  (pour permettre une paramétrisation locale sur le plan tangent).
- *Condition d'échantillonnage* : Le  $R$ -voisinage d'un point germe doit contenir suffisamment de points pour qu'un patch pertinent puisse y être calculé localement, sur la surface. Cela implique que  $R$  doit être suffisamment grand vis à vis de la résolution du scanner.
- *Niveau de bruit* : La méthode suppose que l'amplitude du bruit doit être strictement plus petite que le rayon  $R$ .

Dans tous les exemples présentés dans cet article, le  $R$ -voisinage d'un germe sur la surface a systématiquement été approximé par son  $R$ -voisinage dans l'espace ambiant. La condition topologique revient alors à prendre le rayon  $R$  au dessous du reach de la forme sous-jacente, *i.e.* au dessous de la distance de tout point de la surface à son axe médian. En présence d'arêtes vives, il demeure parfois possible de trouver une couverture valide avec un rayon non nul. Sinon, l'approximation du  $R$ -voisinage sur la surface peut également être remplacée par une approche plus sophistiquée. En pratique, notre approche se comporte tout de même bien en utilisant l'approximation dans l'espace ambiant, même en présence d'arêtes vives ou de voisinages invalides (Figures 7, 12), ou bien en présence de bords sur les surfaces.

## 4. Compression

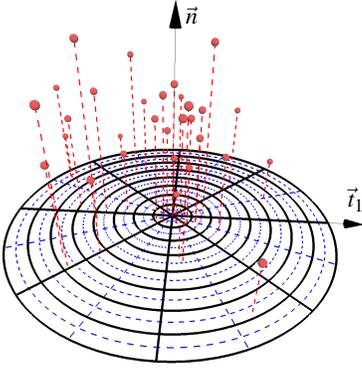
La compression se déroule en trois étapes : la première étape sélectionne un sous-ensemble de points, les *germes* qui serviront de points d'ancrages pour couvrir la surface par un atlas de régions. La deuxième étape consiste à générer une représentation discrète de chaque région. Cette représentation est obtenue en échantillonnant la hauteur de la surface sur une grille radiale (figure 2). Durant la troisième étape, la représentation est comprimée en exploitant l'auto-similarité des régions, via un codage par dictionnaire.

### 4.1. Sélection des germes

L'ensemble des germes sélectionnés doit satisfaire la propriété suivante : sa dilatation de rayon  $R$  doit couvrir tout le nuage de points  $\mathcal{P}$ . D'une manière plus formelle :

$$\forall p \in \mathcal{P}, \exists s \in \mathcal{S}, \|p - s\| \leq R.$$

Chaque point  $p$  peut cependant être couvert plusieurs fois.



**Figure 2:** Représentation locale du voisinage : une carte d'élévation sur une grille radiale

Le sous ensemble est construit à l'aide d'un algorithme de lancer de fléchettes : nous choisissons un germe au hasard, marquons tous les points dans son  $R$ -voisinage comme couverts. Un autre germe peut ensuite être choisi au hasard parmi les points restants. Le processus s'arrête lorsque tous les points ont été couverts. Afin d'éliminer les points aberrants dans le nuage de points original, nous considérons qu'un point ayant peu de voisins ne peut être choisi comme germe. Ceci induit le risque de ne pas respecter la contrainte de couverture, et donc de perdre quelques points. Dans la plupart des cas, la proportion de ces points aberrants est d'environ 0.1%. Sur le nuage de point des Amants de Bordeaux (Figure 1) contenant 15.8 millions de points, avec un rayon de 0.5mm, 89256 germes sont sélectionnés et chaque point est couvert en moyenne 2.6 fois.

#### 4.2. Description des régions

Une fois les germes sélectionnés, leurs  $R$ -voisinages sont échantillonnés selon une grille radiale (voir figure 2). Pour effectuer correctement cet échantillonnage, la normale  $\vec{n}$  et la direction de courbure principale  $\vec{t}_1$  sont estimés pour chaque voisinage, par analyse de covariance. Si la normale est déjà disponible (dans le cas d'un nuage de points orienté), alors seule  $\vec{t}_1$  est calculée. Durant la décompression,  $\vec{n}$  et  $\vec{t}_1$  pourront être efficacement prédits en effectuant une analyse de covariance sur les germes, et il suffira de stocker trois angles correcteurs pour reconstruire correctement ces deux vecteurs.

Une fois que  $\vec{n}$  et  $\vec{t}_1$  sont disponibles, une grille radiale  $(r_i, \theta_j)_{i,j=0 \dots N_{bins}-1}$  est calculée, telle que :

$$r_i = \left(\frac{1}{2} + i\right) \cdot \frac{R}{N_{bins}}; \theta_j = j \cdot \frac{2\pi}{N_{bins}}$$

D'autres motifs d'échantillonnages pourraient être utilisés (par exemple une grille rectangulaire, ou un échantillonnage de Poisson), mais ce schéma d'échantillonnage est bien adapté car la grille radiale produira plus d'échantillons au

voisinage du germe, où la couverture multiple est la moins probable. Les coordonnées des germes sont encodées en utilisant une approche basée sur un kd-tree similaire à [GD02]. Les angles correcteurs sont quantifiés et encodés par codage arithmétique. Pour les Amants de Bordeaux, les informations compressées représentent 312Ko pour les germes et 297KB pour les angles correcteurs, toutes deux quantifiées sur 8 bits.

#### 4.3. Compression basée sur l'auto similarité

Pour exploiter l'auto-similarité entre les voisinages, notre approche construit un dictionnaire, un ensemble d'atomes qui serviront à décomposer les différentes régions. Les atomes du dictionnaire et les coefficients d'encodage sont calculés à l'aide de l'algorithme K-SVD [AEB06], qui minimise l'erreur de reconstruction en maintenant la parcimonie des coefficients, comme expliqué dans la section 2.4. Cet algorithme génère des décompositions linéaires parcimonieuses des régions i.e. avec peu de coefficients non-nuls. De plus, la distribution des coefficients non-nuls présente un pic centré sur 0, ce qui les rend très appropriés pour la compression, qui est effectuée par quantification scalaire suivie d'un codage arithmétique. Enfin, la quantification augmente la parcimonie des coefficients. Pour le nuage de points des Amants de Bordeaux, le pourcentage de coefficients non-nuls est d'environ 30% avant quantification et 20% après quantification sur 8 bits.

Notre approche basée sur le contenu produit de meilleurs résultats que la Transformée en Cosinus Discret (TCD) qui est connue pour concentrer l'énergie des signaux dans les basses fréquences. En utilisant la TCD en lieu et place de notre décomposition, avec une quantification sur 8 bits, l'erreur quadratique moyenne est plus grande pour la TCD 0.14 comparé à notre approche où elle vaut 0.01.

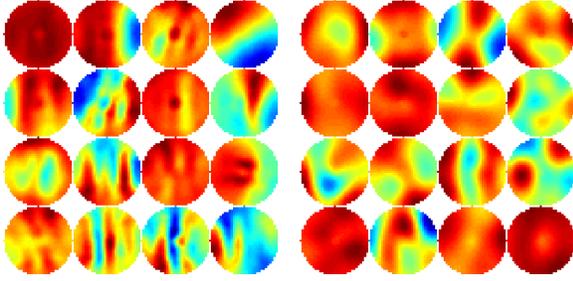
La figure 3 montre deux dictionnaires construits pour deux nuages distincts. Pour les Amants de Bordeaux, le dictionnaire et les coefficients sont respectivement encodés avec 18Ko et 507Ko.

#### Réglage des paramètres et contrôle de la précision

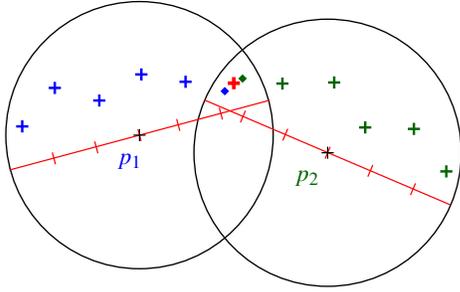
Notre approche est régie par 5 paramètres : le rayon des régions  $R$ , la taille de la grille  $N_{bins}$ , le nombre d'atomes dans le dictionnaire  $N_{atoms}$ , et le pas de quantification des coordonnées des germes et des coefficients.

Pour nos expériences, nous fixons  $N_{bins} = 16$  (i.e. le nombre moyen de points par voisinage vaut environ  $N_{bins}^2$ ) et nous définissons le rayon en conséquence, afin que la résolution d'échantillonnage soit proche de la résolution du scanner.

Le nombre d'atomes est facile à régler. Il influera directement la préservation des détails. Si la surface a peu de variations locales (par exemple une pièce mécanique ou une forme dont les détails ont environ la même taille), 16 atomes



**Figure 3:** Dictionnaires construits pour des nuages de points : un nuage géométrique (la mire, gauche) et un nuage lisse (les Amants de Bordeaux, droite). Les atomes sont classés par ordre d'importance (poids total absolu dans la décomposition).



**Figure 4:** Reconstruction des points à partir des patches : les germes  $p_1$  et  $p_2$  donnent potentiellement  $N_{bins}^2$  points chacune. Pour éviter le suréchantillonnage dans les zones d'overlap, chaque germe donne une position (losanges bleus et verts) qui sont combinées pour donner le point rouge gras final.

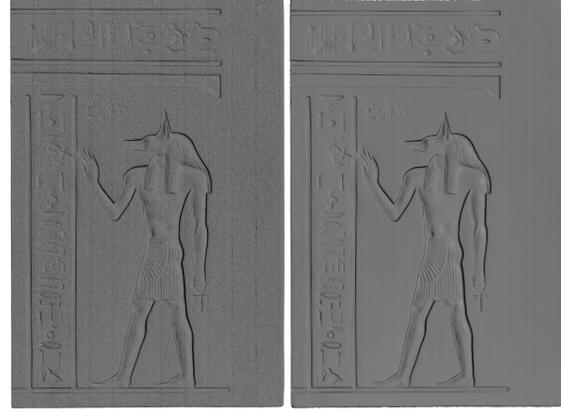
sont suffisants pour représenter correctement la forme. Dans le cas de formes plus détaillées (par exemple Tanagra avec des détails de taille différente) nous utilisons 32 atomes. Pour un contrôle plus précis sur la précision de la compression, le nombre d'atomes du dictionnaire peut aussi être augmenté graduellement jusqu'à ce que l'erreur d'approximation générée par l'algorithme k-SVD soit inférieure à la précision du scanner. Tant que la précision souhaitée n'est pas atteinte, des atomes peuvent être rajoutés itérativement au dictionnaire.

La quantification est fixée à 16 bits pour les coordonnées des germes et 8 bits pour les coefficients

## 5. Décompression

L'algorithme de décompression comporte deux étapes : la première décompresse les patches étant donné le dictionnaire et les coefficients. La deuxième étape déduit le nuage de points final à partir de l'ensemble des germes et des patches décompressés.

Étant donné le dictionnaire  $D$  et la matrice de coefficients



**Figure 5:** Compression et décompression du nuage de points Anubis (à gauche : original, à droite : décompression).  $R = 0.7mm, N_{bins} = 16$

$X$ , les patches reconstruits sont obtenus par :  $Y_{rec} = D \cdot X$ . Ce qui donne un patch décompressé par germe. L'étape suivante est de passer de ces patches décompressés au nuage de points final. Afin d'éviter les grandes variations de densité d'échantillonnage dans les zones où des patches voisins se recouvrent, on ajoute une consolidation locale dans ces zones.

Chaque bin  $(r, \theta)$  du motif d'échantillonnage  $\mathcal{F}_s(r, \theta)$  correspondant à la germe  $s$  donne potentiellement un point dont les coordonnées dans le repère local  $(s, t_1(s), t_2(s), n(s))$  s'écrivent :

$$(r \cos(\theta), r \sin(\theta), \mathcal{F}_s(r, \theta))$$

Ce qui se traduit dans le système de coordonnées globales comme :  $s + r \cos(\theta)t_1(s) + r \sin(\theta)t_2(s) + \mathcal{F}_s(r, \theta)n(s)$ .

Comme les patches se recouvrent une stratégie de fusion locale est nécessaire (fig. 4). Plus précisément, lorsqu'un point décompressé  $p$  est ajouté au nuage, on compte le nombre de germes dont le voisinage contient  $p$ . Si il n'y en a qu'un le point est gardé tel quel. Sinon,  $p$  est projeté sur le plan tangent paramétrisé de chacune des germes. Chacune de ces germes  $s$  peut alors prédire la position  $p^s$  de  $p$  à partir de son propre patch. La position finale de  $p$  est la moyenne des positions prédites  $p^s$ . On remarque qu'utiliser des grilles radiales rend le calcul de recouvrement plus facile qu'avec des grilles carrées.

## 6. Résultats et comparaisons

La compression a été testée sur des données du Farman Institute dataset [DAL\*11], du Stanford Michelangelo Project et du Robotic 3D Scan Repository (Université d'Osna-brück). Des normales orientées sont fournies dans chacun des cas sauf pour les données de Brême et le St Matthieu.



**Figure 6:** Rendus du St Matthieu original (à gauche) et dé-compressé (à droite), les deux comportent 93.4 millions de points.  $R = 3mm, N_{bins} = 16$

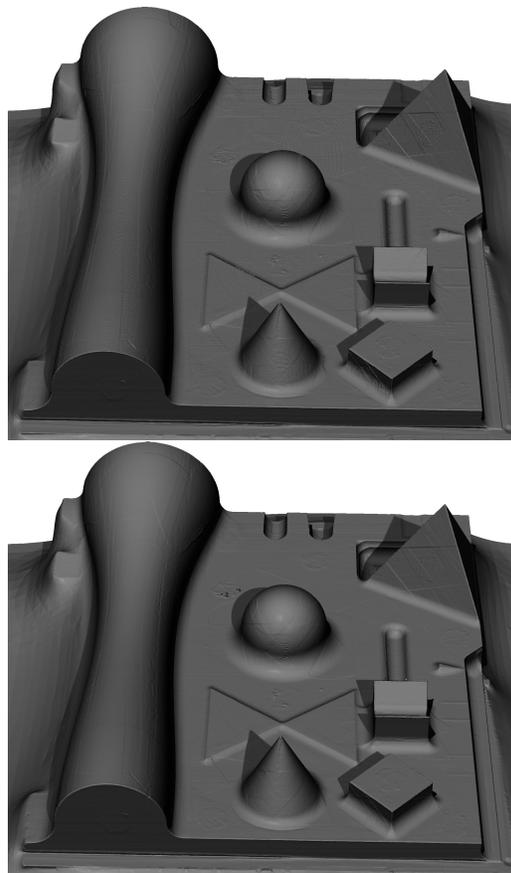
En ce qui concerne les temps de calcul, pour les 90 millions de points du St Matthieu, la sélection des germes et le calcul des patches locaux prennent environ 10 minutes. Le K-SVD prend 3 minutes de plus tandis que le reste des calculs est quasi-instantané. Pour le nuage du David, la compression totale prend environ 8 minutes. En comparaison [SMK08] prend 11 minutes pour la seule décomposition du modèle en carte de hauteur. De plus, la compression de ce même David par un algorithme inadapté comme le bzip2 prend environ 6 minutes. L'étape de décompression prend plus de temps : le décodage des coefficients et des motifs locaux est instantané mais la génération non-optimisée des points prend environ 15 minutes pour le St Matthieu et 10 minutes pour le David. Cette étape étant très locale, des optimisations supplémentaires et du parallélisme réduirait grandement les temps de calcul.

La figure 5 montre le débruitage et la préservation des détails obtenue par compression/décompression. On peut vérifier que seul le bruit est retiré en comparant l'erreur avec la résolution du scan : dans la plupart des cas, cette erreur est sous la résolution du scanner.

La figure 6 montre la compression/décompression d'un nuage de point dense, le St Matthieu [LPC\*00], réduit à 93,4 millions de points à cause de notre limitation en mémoire vive (4GB). Cette expérience montre également que les détails sont très bien préservés.

La figure 7 montre le comportement de la surface décompressée quand elle est reconstruite par un algorithme de reconstruction standard [KH13]. Des résultats quasi-identiques sont obtenus avec le nuage de points original et décompressé.

Pour les comparaisons suivantes, on mesure l'erreur de décompression comme la racine carrée de la distance normale moyenne entre les nuages de points (sauf pour les courbes de la figure 10 où elle est mesurée comme la racine



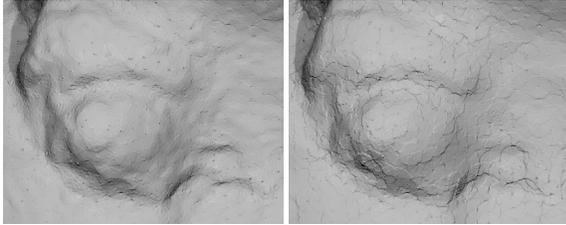
**Figure 7:** Nuage de points original (en haut) et décompressé (en bas). Les deux formes sont reconstruites par Screened Poisson Reconstruction [KH13].  $R = 0.6mm, N_{bins} = 16$

de la distance carrée moyenne au plus proche point, afin de correspondre aux résultats des autres approches). Nous comparons notre méthode avec une approche basée kd-tree similaire à [GD02] qui compresse les nuages de points non orientés. À  $4.13bpp$ , [GD02] donne une erreur de 0.066mm. En comparaison, notre approche donne une RMSE de 0.010mm à 0.6bpp. Les résultats sont comparés visuellement sur la figure 8.

Nous comparons également notre approche à une approche semblable à [HMHB08] en calculant les clusters de similarité et en remplaçant chaque patch par le représentant de sa classe. En utilisant 16 clusters (le même nombre que pour notre algorithme), on remarque que de nombreux artefacts apparaissent (Figure 9) ce qui donne une erreur plus grande (0.12mm contre 0.01mm pour notre méthode sur le nuage de points des amants). La figure 10 compare notre approche avec plusieurs méthodes existantes [KV05, SMK08, HMHB08] en termes de débit/distorsion. Pour notre approche, les bitrates distincts sont obtenus simplement en faisant varier le rayon  $R$  des patches. Notre ap-



**Figure 8:** Comparaison avec de la compression basée *kd-tree*. Nuage de points original (gauche), décompression comparée (droite) Même avec plus de bits par point (4.83 contre 0.6 dans notre méthode), la partie de droite encodée avec [GD02] est moins précise qu’avec notre approche (partie de gauche).  $R = 0.7\text{mm}$ ,  $N_{bins} = 16$



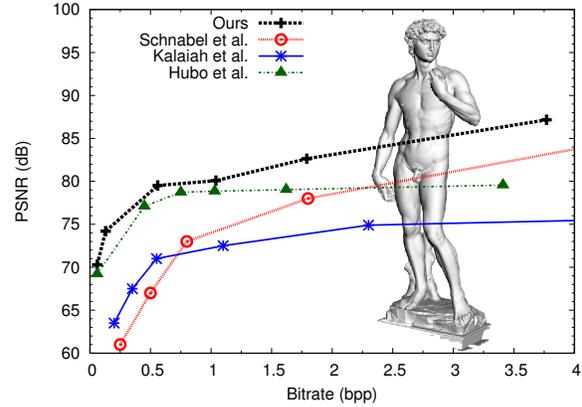
**Figure 9:** Comparaison avec une approche similaire à [HMHB08] (droite). Notre méthode (gauche) évite la création d’artefact en étant plus flexible. De manière quantitative, la RMSE est de 0.12mm contre 0.01mm par notre approche.  $R = 0.5\text{mm}$ ,  $N_{bins} = 16$

proche permet donc une meilleure qualité pour des tailles compressées équivalentes.

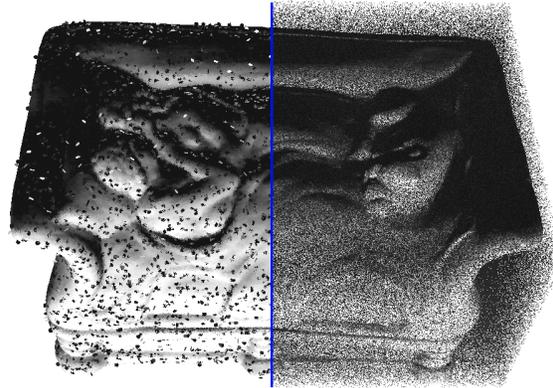
Enfin, le tableau 1 compare des mesures de qualité pour la compression/décompression de plusieurs nuages de points. En particulier, cela montre que le nombre de points dont l’erreur est supérieure à la résolution initiale est très faible, ce qui illustre que cette compression ne détruit pas la précision.

Le comportement de l’algorithme en présence de points aberrants est montré sur la figure 11. On peut voir que seuls les points aberrants qui sont à une distance de moins de  $R$  de la surface perturbent l’algorithme. Les voisinages de ces points peuvent en effet contenir suffisamment de points pour créer un patch et donc des artefacts de décompression. Ils peuvent être considérés comme un fort bruit, ce qui confirme nos hypothèses de travail (voir section 3). Les points aberrants qui sont loin de la surface sont bien retirés par l’algorithme.

Les données de la ville de Brême (70 millions de points, figure 12) sont particulièrement compliquées car elles sortent



**Figure 10:** Comparaison avec plusieurs méthodes existantes en terme de débit/distorsion sur le modèle du David. Les différents bitrates sont obtenus en augmentant le rayon des patches et la taille des descripteurs.



**Figure 11:** Compression des Amants de Bordeaux avec 10% de points aberrants (à gauche : décompressé ; à droite : initial)

clairement de nos hypothèses de travail : des structures linéaires existent et il y a des zones de faible densité. Le résultat est toujours valide dans des zones suffisamment échantillonnées. Cependant les zones faiblement échantillonnées peuvent être ignorées par l’étape de pré-traitement (section 4.1), montrant ainsi les limites de l’algorithme.

## 7. Conclusion

Nous avons introduit une approche efficace pour la compression de nuages de points dense décrivant une surface. La méthode commence par sélectionner un sous-ensemble des points, calcule des descripteurs locaux autour de ces germes et utilise la similarité entre les descriptions pour les encoder.

Notre approche encode des dizaines de millions de points scannés sur une surface avec moins de 1 bit par point. De plus, les différences avec le nuage de point initial est lié au

Nuage de points	nombre points	R	taille compressée (bytes)	RMSE (% de la diagonale)	Pourcentage de points avec erreur au dessus de l'erreur d'acquisition	bpp
Anubis	9,9M	0.7mm	1,201,636	0.01mm (0.003%)	1.23%	0.96
Lovers of Bordeaux	15,8M	0.5mm	1,152,245	0.01mm(0.006%)	0.86%	0.59
Mire	16,1M	0.6mm	1,480,118	0.03mm (0.011%)	1.30%	0.73
Tanagra	16,4M	0.7mm	1,238,271	0.01mm (0.004%)	1.56%	0.60
David	28,2M	10mm	2,150,711	0.24mm (0.004%)	0.75%	0.61
Bremen	69,9M	18cm	6,699,915	1.48cm (0.005%)	not available	0.76
St Matthew	93,5M	3mm	9,780,886	0.05cm (0.002%)	not available	0.83

**Table 1:** Nombre de bytes compressé versus décompressé. La dernière colonne utilise la précision de l'outil d'acquisition, une information dont nous ne disposons pas pour Brême et St Matthieu. Le St Matthieu (Fig 6) a d'abord été sous-échantillonné à 93,4M points à cause d'une limitation en mémoire vive (4GB).

bruit. Le nuage de points décompressé se comporte aussi bien sinon mieux que le nuage de points initial pour la reconstruction ou la visualisation.

Notre algorithme peut être amélioré de bien des façons :

- L'exploitation de la représentation basée patch peut générer des artefacts au niveau des bords d'une surface ouverte : ces bords peuvent être dilatés. Une solution serait d'ajuster la taille du patch près des frontières. Cela nécessiterait de sauvegarder un entier de plus (rayon sur la grille discrète) pour chaque patch.
- D'autres stratégies de placement des germes devraient être étudiées. Par exemple, les germes pourraient être placées pour minimiser l'erreur locale, d'une manière similaire à [OBS06].
- Nous pouvons encoder les attributs des points, comme la couleur ou les normales de la même façon en exploitant la similarité.

**Perspectives :** bien que notre algorithme soit basé sur l'exploitation de l'auto-similarité sur la surface complète, la plupart des traitements restent très locaux. C'est donc de bon augure pour des données de taille de plus en plus grande avec des processus de streaming. Cette méthode prend finalement tout son sens pour les campagnes d'acquisition qui couvrent parfois des villes entières.

## Remerciements

Les auteurs remercient Pierre-Marie Gandoin pour son code de compression et ses remarques constructives. Les données de la ville de Brême (Fig. 12) appartiennent à Dorit Borrmann et Jan Elseberg de Jacobs University Bremen (les données sont disponibles sur le Robotic 3D Scan Repository de l' Universität Osnabrück). Le Saint Mathieu (Fig. 6) et David (Fig. 10) font partie du Digital Michelangelo Project, Stanford University. Ce travail a été partiellement financé par la Région Rhône-Alpes (ARC6).

## Références

[ABCO\*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T. : Point set sur-

faces. In *Proc. Vis '01* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 21–28.

[AD01] ALLIEZ P., DESBRUN M. : Progressive encoding for lossless transmission of 3d meshes. In *ACM Siggraph Conference Proceedings* (2001), pp. 198–205.

[AEB06] AHARON M., ELAD M., BRUCKSTEIN A. : K-svd : An algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.* Vol. 54, Num. 11 (novembre 2006), 4311–4322.

[AGDL09] ADAMS A., GELFAND N., DOLSON J., LEVOY M. : Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph., Proc. SIGGRAPH*. Vol. 28 (July 2009), 21 :1–21 :12.

[BCM05] BUADES A., COLL B., MOREL J.-M. : A non-local algorithm for image denoising. In *In CVPR* (2005), pp. 60–65.

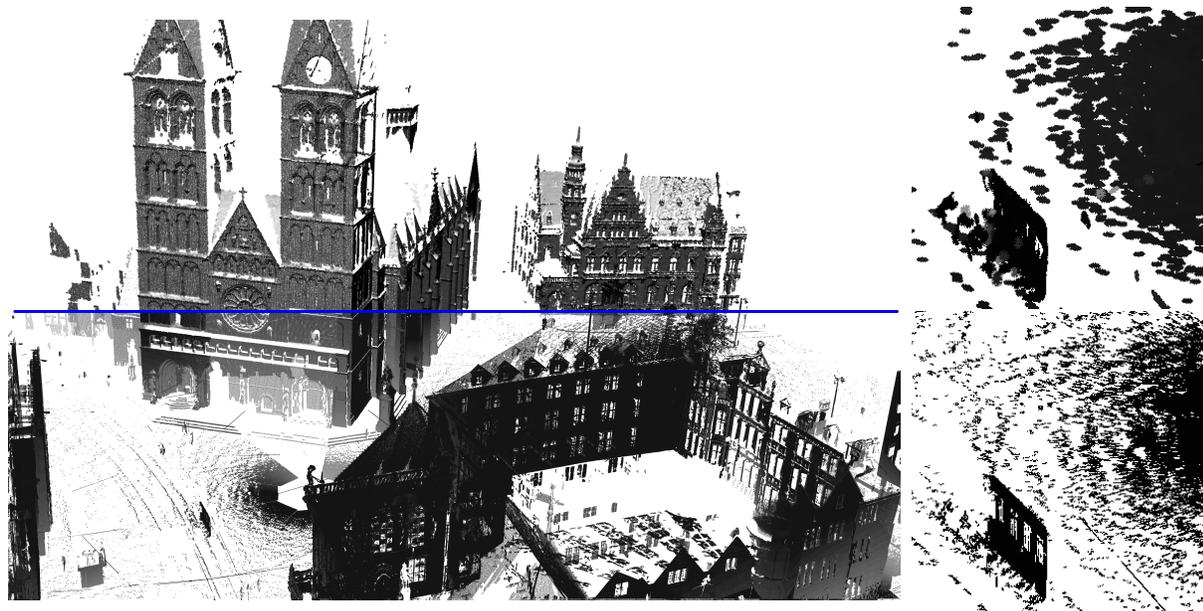
[BLSC\*11] BOYER D. M., LIPMAN Y., ST. CLAIR E., PUENTE J., PATEL B. A., FUNKHOUSER T., JERNVALL J., DAUBECHIES I. : Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proceedings of the National Academy of Sciences*. Vol. 108, Num. 45 (2011), 18221–18226.

[CRT06] CANDÈS E. J., ROMBERG J. K., TAO T. : Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*. Vol. 59, Num. 8 (2006), 1207–1223.

[DAL\*11] DIGNE J., AUDFRAY N., LARTIGUE C., MEHDI-SOUZANI C., MOREL J.-M. : Farman Institute 3D Point Sets - High Precision 3D Data Sets. *Image Processing On Line*. Vol. 2011 (2011).

[DCV14] DIGNE J., CHAÎNE R., VALETTE S. : Self-similarity for accurate compression of point sampled surfaces. *Computer Graphics Forum (Proceedings Eurographics 2014)* (May 2014).

[Dee95] DEERING M. : Geometry compression. In *ACM Trans. Graph. Proc. SIGGRAPH '95* (New York, NY, USA, 1995), ACM, pp. 13–20.



**Figure 12:** Résultat sur les données de Brème. À gauche : original (en bas) et résultat décompressé (en haut). À droite : détails : original (en bas) et décompressé (en haut)  $R = 18\text{cm}$ ,  $N_{bins} = 16$ . Dans les zones faiblement échantillonnées, deux phénomènes se produisent : si l'échantillonnage est trop faible, les points sont retirés pendant le pré-traitement, et si l'échantillonnage est juste au dessus du seuil, il y a un effet de dilatation clairement visible sur les détails (colonne de droite).

- [Dig12] DIGNE J. : Similarity based filtering of point clouds. In *CVPR Workshops* (2012), IEEE, pp. 73–79.
- [EA06] ELAD M., AHARON M. : Image denoising via learned dictionaries and sparse representation. In *CVPR* (2006), pp. 17–22.
- [GAB12] GUILLEMOT T., ALMANSA A., BOUBEKEUR T. : Non local point set surfaces. In *Proc. 3DIMPVT 2012* (2012), pp. 324–331.
- [GD02] GANDAIN P.-M., DEVILLERS O. : Progressive lossless compression of arbitrary simplicial complexes. *ACM Trans. Graph., Proc. SIGGRAPH 02*, Num. 3 (juillet 2002), 372–379.
- [GGH02] GU X., GORTLER S. J., HOPPE H. : Geometry images. *ACM Trans. Graph. Proc. of SIGGRAPH 02*, Vol. 21, Num. 3 (2002), 355–361.
- [GVSS00] GUSKOV I., VIDIMČE K., SWELDENS W., SCHRÖDER P. : Normal meshes. In *ACM Trans. Graph. Proc. SIGGRAPH '00* (New York, NY, USA, 2000), pp. 95–102.
- [HMHB08] HUBO E., MERTENS T., HABER T., BEKAERT P. : Self-similarity based compression of point set surfaces with application to ray tracing. *Comput. Graph. Point-Based Graphics*, Vol. 32, Num. 2 (avril 2008), 221–234.
- [HPKG06] HUANG Y., PENG J., KUO C.-C. J., GOPI M. : Octree-based progressive geometry coding of point clouds. In *Proc. Point-Based Graphics 06* (2006), SPBG'06, Eurographics, pp. 103–110.
- [KG00] KARNI Z., GOTSMAN C. : Spectral Compression of Mesh Geometry. In *ACM Siggraph 00 Conference Proceedings* (2000), pp. 279–286.
- [KH13] KAZHDAN M., HOPPE H. : Screened poisson-surface reconstruction. *ACM Transactions on Graphics* (2013).
- [KSS00] KHODAKOVSKY A., SCHRÖDER P., SWELDENS W. : Progressive Geometry Compression. *ACM Trans. Graph. Proc. SIGGRAPH 00* (2000), 271–278.
- [KV05] KALAI AH A., VARSHNEY A. : Statistical geometry representation for efficient transmission and rendering. *ACM Trans. Graph.*, Vol. 24, Num. 2 (Apr 2005), 348–373.
- [LPC\*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., , FULK D. : The digital michelangelo project. In *ACM Trans. Graph. Proc SIGGRAPH 00* (2000), pp. 131–144.
- [LSS\*98] LEE A. W. F., SWELDENS W., SCHRÖDER P., COWSAR L., DOBKIN D. : Maps : multiresolution adaptive parameterization of surfaces. *ACM*, pp. 95–104.
- [MBP\*09] MAIRAL J., BACH F., PONCE J., SAPIRO G., ZISSERMAN A. : Non-local sparse models for image restoration. In *ICCV* (2009), pp. 2272–2279.

- [MCAH12] MAGLO A., COURBET C., ALLIEZ P., HUDELLOT C. : Progressive compression of manifold polygon meshes. *Computers & Graphics, Proc. SMI 2012*. Vol. 36, Num. 5 (2012), 349 – 359.
- [MGP06] MITRA N. J., GUIBAS L. J., PAULY M. : Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph. Proc SIGGRAPH 06* (2006), 560–568.
- [MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D. : Symmetry in 3d geometry : Extraction and applications. In *EUROGRAPHICS State-of-the-art Report* (2012).
- [OBS06] OHTAKE Y., BELYAEV A., SEIDEL H.-P. : A composite approach to meshing scattered data. *Graph. Models*. Vol. 68, Num. 3 (mai 2006), 255–267.
- [PK05] PENG J., KUO C.-C. J. : Geometry-guided progressive lossless 3d mesh coding with octree decomposition. *ACM Trans. Graph., Proc. SIGGRAPH 05*, Num. 3 (juillet 2005), 609–616.
- [PM05] PEYRÉ G., MALLAT S. : Surface compression with geometric bandelets. In *ACM Trans. Graph. Proc. SIGGRAPH '05* (2005), ACM, pp. 601–608.
- [PMW\*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L. J. : Discovering structural regularity in 3d geometry. *ACM Trans. Graph., Proc SIGGRAPH'08*, Num. 3 (2008), 43 :1–43 :11.
- [RL00] RUSINKIEWICZ S., LEVOY M. : Qsplat : a multi-resolution point rendering system for large meshes. *SIGGRAPH '00*, pp. 343–352.
- [RZE08] RUBINSTEIN R., ZIBULEVSKY M., ELAD M. : *Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit*. Tech. rep., 2008.
- [SK06] SCHNABEL R., KLEIN R. : Octree-based point-cloud compression. In *Symposium on Point-Based Graphics 2006* (juillet 2006), Botsch M., Chen B., (Eds.), Eurographics.
- [SMK08] SCHNABEL R., MÖSER S., KLEIN R. : Fast vector quantization for efficient rendering of compressed point-clouds. *Computers & Graphics*. Vol. 32, Num. 2 (2008), 246 – 259.
- [SPS12] SMITH J., PETROVA G., SCHAEFER S. : Progressive encoding and compression of surfaces generated from point cloud data. *Computers & Graphics*. Vol. 36, Num. 5 (2012), 341–348.
- [TG98] TOUMA C., GOTSMAN C. : Triangle Mesh Compression. *Graphics Interface 98 Conference Proceedings* (1998), 26–34.
- [YBS06] YOSHIZAWA S., BELYAEV A., SEIDEL H.-P. : Smoothing by example : Mesh denoising by averaging with similarity-based weights. In *SMI '06* (Washington, DC, USA, 2006), IEEE, p. 9.
- [ZSW\*10] ZHENG Q., SHARF A., WAN G., LI Y., MITRA N. J., COHEN-OR D., CHEN B. : Non-local scan consolidation for 3d urban scenes. *ACM Trans. Graph., Proc. SIGGRAPH 10*, Num. 4 (2010), 94 :1–94 :9.