



HAL
open science

Une approche basée graphes pour la modélisation et le traitement de nuages de points massifs issus d'acquisitions de LiDARs terrestres

Arnaud Bletterer

► To cite this version:

Arnaud Bletterer. Une approche basée graphes pour la modélisation et le traitement de nuages de points massifs issus d'acquisitions de LiDARs terrestres. Traitement du signal et de l'image [eess.SP]. COMUE Université Côte d'Azur (2015 - 2019), 2018. Français. NNT : 2018AZUR4218 . tel-02099777

HAL Id: tel-02099777

<https://theses.hal.science/tel-02099777>

Submitted on 15 Apr 2019

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.

THÈSE DE DOCTORAT

Présentée en vue de l'obtention du grade de
Docteur en Automatique, Traitement du Signal et des Images
de l'Université Côte d'Azur

par

Arnaud BLETTERER

**UNE APPROCHE BASÉE GRAPHES POUR LA
MODÉLISATION ET LE TRAITEMENT DE NUAGES
DE POINTS MASSIFS ISSUS D'ACQUISITIONS DE
LIDARS TERRESTRES**

Dirigée par : Marc ANTONINI

Co-encadrée par : Frédéric PAYAN

au sein de l'équipe Mediacoding du Laboratoire I3S

Soutenue le 10 Décembre 2018 devant le jury composé de :

Pierre ALLIEZ	Directeur de recherche - INRIA	(Examineur)
Marc ANTONINI	Directeur de recherche - CNRS	(Directeur de thèse)
Mathieu BRÉDIF	Chargé de recherche - IGN	(Examineur)
Raphaëlle CHAINE	Professeur - Université de Lyon	(Rapporteur)
Anis MEFTAH	Cintoo	(Membre invité)
Frédéric PAYAN	Maître de conférences - Université Côte d'Azur	(Co-encadrant)
Julien TIERNY	Chargé de recherche - CNRS	(Rapporteur)

RÉSUMÉ

Avec l'évolution des dispositifs d'acquisition 3D, les nuages de points sont maintenant devenus une représentation essentielle des scènes numérisées. Les systèmes récents sont capables de capturer plusieurs centaines de millions de points en une seule acquisition. Comme plusieurs acquisitions sont nécessaires pour capturer la géométrie de scènes de grande taille, un site historique par exemple, nous obtenons des nuages de points massifs, i.e., composés de plusieurs milliards de points. Dans cette thèse, nous nous intéressons à la structuration et à la manipulation de nuages de points issus d'acquisitions générées à partir de LiDARs terrestres. À partir de la structure de chaque acquisition, des graphes, représentant chacun la connectivité locale de la surface numérisée, sont construits. Les graphes créés sont ensuite liés entre eux afin d'obtenir une représentation globale de la surface capturée. Nous montrons que cette structure est particulièrement adaptée à la manipulation de la surface sous-jacente aux nuages de points massifs, même sur des ordinateurs ayant une mémoire limitée. Notamment, nous montrons que cette structure permet de traiter deux problèmes spécifiques à ce type de données. Un premier lié au ré-échantillonnage de nuages de points, en générant des distributions de bonne qualité en termes de bruit bleu grâce à un algorithme d'échantillonnage en disques de Poisson. Un autre lié à la construction de diagrammes de Voronoï centroïdaux, permettant l'amélioration de la qualité des distributions générées, ainsi que la reconstruction de maillages triangulaires.

ABSTRACT

With the evolution of 3D acquisition devices, point clouds have now become an essential representation of digitized scenes. Recent systems are able to capture several hundreds of millions of points in a single acquisition. As multiple acquisitions are necessary to capture the geometry of large-scale scenes, a historical site for example, we obtain massive point clouds, i.e., composed of billions of points. In this thesis, we are interested in the structuration and manipulation of point clouds from acquisitions generated by terrestrial LiDARs. From the structure of each acquisition, graphs, each representing the local connectivity of the digitized surface, are constructed. Created graphs are then linked together to obtain a global representation of the captured surface. We show that this structure is particularly adapted to the manipulation of the underlying surface of massive point clouds, even on computers with limited memory. Especially, we show that this structure allow to deal with two problems specific to that kind of data. A first one linked to the resampling of point clouds, by generating distributions of good quality in terms of blue noise thanks to a Poisson disk sampling algorithm. Another one connected to the construction of centroidal Voronoi tessellations, allowing to enhance the quality of generated distributions and to reconstruct triangular meshes.

PUBLICATIONS

Les travaux réalisés dans cette thèse peuvent également être trouvés dans les publications et rapports suivant(e)s :

EN COURS DE RÉDACTION

- A. BLETTERER, F. PAYAN, M. ANTONINI et A. MEFTAH. « A graph-based approach for modeling and processing gigantic point clouds from terrestrial LiDARs acquisitions ».

CONFÉRENCES INTERNATIONALES

- A. BLETTERER, F. PAYAN, M. ANTONINI et A. MEFTAH (2018b). « Towards the reconstruction of wide historical sites : A local graph-based representation to resample gigantic acquisitions ». Dans : *16th EUROGRAPHICS Workshop on Graphics and Cultural Heritage (EG GCH)*. Vienna, Austria, p. 1 -9
- A. BLETTERER, F. PAYAN, M. ANTONINI et A. MEFTAH (2018a). « Out-of-core Resampling of Gigantic Point Clouds ». Dans : *Symposium on Geometry Processing 2018- Posters*. Sous la dir. de T. JU et A. VAXMAN. The Eurographics Association
- A. BLETTERER, F. PAYAN, M. ANTONINI et A. MEFTAH (2016). « Point Cloud Compression using Depth Maps ». Dans : *Electronic Imaging 2016.21, 3DIPM-396*

CONFÉRENCES NATIONALES

- A. BLETTERER, F. PAYAN, M. ANTONINI et A. MEFTAH (2017). « De la carte de profondeur au maillage surfacique : reconstruction de scènes 3D complexes. » Dans : *Groupe de Recherche et d'Etudes du Traitement du Signal et des Images (GRETSI)*. Juan les Pins, France
- A. BLETTERER, F. PAYAN et M. ANTONINI (2015). « Utilisation de cartes de profondeur pour la visualisation de nuages de points ». Dans : *28èmes journées de l'Association Française d'Informatique Graphique (AFIG)*. Lyon, France

A la mémoire de Mamama,

REMERCIEMENTS

Dans un premier temps, j'aimerais exprimer mon entière gratitude envers Frédéric Payan et Marc Antonini. Après avoir respectivement accepté d'encadrer et de diriger ma thèse, ceux-ci m'ont toujours accordé leur confiance quant à l'orientation des recherches ayant été effectuées durant la totalité de la thèse. Et cela même lorsque mes hypothèses n'étaient que de l'ordre de l'intuition et que mes seules justifications se présentaient à peine sous la forme de "griffonnages" sur un morceau de papier ou un tableau. Je tiens aussi à les remercier pour tout le temps qu'ils ont passé à me (re)ⁿlire et me (re)ⁿcorriger, $n \in \mathbb{N}$, ainsi que pour la patience dont ils ont su faire preuve alors que les échéances se rapprochaient à grands pas. Enfin je suis très reconnaissant des relations que nous avons eu ainsi que de la bonne atmosphère qu'ils ont su faire régner tout au long de cette thèse.

Je voudrais ensuite chaleureusement remercier Raphaëlle Chainé et Julien Tierny d'avoir accepté de relire cette thèse et d'en être les rapporteurs. Je les remercie tout particulièrement des commentaires qu'ils ont pu émettre concernant le contenu et la forme du manuscrit. De plus, je suis très reconnaissant envers Pierre Alliez et Mathieu Brédif d'avoir accepté d'examiner ce travail de thèse.

Cette thèse s'est faite en étroite collaboration avec Cinto, et j'ai eu l'occasion de passer quelques mois de la thèse dans leurs locaux. J'ai une forte gratitude envers Leonardo Hidd-Fonteles et Anis Meftah, pour la curiosité qu'ils ont éveillé en moi concernant les problématiques industrielles au cours des nombreux échanges que nous avons pu avoir. De plus, je les remercie de m'avoir "poussé" concernant les aspects liés à l'optimisation du code. Si aujourd'hui nous avons pu traiter des données de taille aussi conséquente, leur contribution à cette réussite ne fait aucun doute. Je leur souhaite bon courage dans la continuation de leur projet entrepreneurial.

Dans une thèse il y a aussi une partie administrative, et c'est parce qu'il y a des personnes compréhensives et dédiées à la bonne marche de celle-ci que nous, doctorants, pouvons faire l'ensemble des démarches qui nous sont demandées. Je tiens donc à remercier cordialement les trois assistants d'équipe que j'ai pu côtoyer, à savoir Micheline Hagnere, Frédéric Fontaine et dernièrement Nadia Belfegas-Fagues. De manière plus générale, je tiens à souligner l'implication de l'ensemble des membres de l'administration du Laboratoire I3S qui ont toujours fait de leur mieux pour résoudre l'ensemble des problèmes logistiques.

J'ai côtoyé beaucoup de monde durant mon séjour au laboratoire, et j'espère que les lignes suivantes seront les plus exhaustives possibles (à ceux qui liront ce

texte, pardonnez-moi si je vous ai oublié). Déjà, j'ai une forte pensée pour tous les gens avec qui j'ai partagé le bureau 126 pendant plus ou moins longtemps, à savoir Froso, Yasmine, Florent et Michel. Ensuite, tous les autres membres de MediaCoding avec qui j'ai déjeuné souvent : Melpo, Cyprien, Danny, Tom ; ou moins souvent : Lauriane, Diana, Marie, Samvel, Emmanuel. Je garde d'excellents souvenirs de nos interactions, et j'espère avoir l'occasion de vous revoir le plus souvent possible. Après, il est évident que mon séjour au Laboratoire aurait été tout autre sans les collègues de l'équipe Morpheme, que ce soit avec les garçons du bureau voisin : Alexis, Emmanuel et Gaël, ou tous les autres membres de l'équipe : Grégoire, Xavier, Eric, Laure, Lola, Agustina, Anca, Sarah, Emmanuelle, Somia, Nadège, Arne, Kévin, Djampa, Raphaël, ... Je vais m'arrêter là avec les énumérations, sinon j'aurais encore un étage entier à décrire (Ophélie, Benjamin, Jonathan, Emilien et Jean-Marie ne m'en voulez pas). Merci à toutes les autres personnes que j'ai pu croiser, et avec qui j'ai eu l'occasion de partager des moments professionnels, ou moins professionnels (surtout).

Enfin, durant cette thèse, j'ai eu l'occasion de participer à différentes activités annexes, que ce soit au niveau de l'IUT Nice Côte d'Azur, de la Fête de la Science ou encore du projet MEDITES (merci à Anne-Sophie Coldefy de m'avoir permis d'intégrer à ce projet enrichissant auprès de collégiens de la Région PACA). Merci à toutes les personnes avec qui j'ai eu l'occasion de collaborer dans ces différents projets.

Pour conclure ces remerciements, je tiens à remercier du fond du coeur ma famille, dont les soutiens de toute sorte ont su m'aider durant ces quatre années (en particulier pour son achèvement en beauté avec le pot de thèse), malgré mon manque cruel de loquacité, et même si j'ai souvent été, et suis peut être toujours, incapable de vous expliquer clairement ce que j'ai fait au cours de ces années dans un laboratoire dans le Sud de la France. Votre soutien m'a profondément touché, et j'espère vous fournir au moins le même quand vous en aurez besoin à votre tour.

TABLE DES MATIÈRES

I	PRÉFACE	1
1	INTRODUCTION	3
1.1	Introduction générale	3
1.2	Numérisation 3D	4
1.2.1	Scanners 3D	6
1.2.2	LiDARs terrestres	8
1.3	Contexte général de cette thèse	12
1.3.1	Motivations	14
1.3.2	Contributions	15
1.3.3	Organisation du manuscrit	16
II	STRUCTURATION DE NUAGES DE POINTS	17
2	INTRODUCTION	19
2.1	Etat de l’art de la structuration de nuages de points	19
2.1.1	Méthodes spatiales	21
2.1.2	Méthodes surfaciques	25
2.2	Objectifs de cette partie	27
3	STRUCTURATION DE NUAGES DE POINTS PAR GRAPHES LOCAUX	29
3.1	Structuration d’une acquisition : construction d’un graphe local	30
3.2	Structuration d’un ensemble d’acquisitions : connexions entre graphes	33
3.2.1	Gestion des correspondances entre graphes	34
3.2.2	Suppression des sommets redondants	37
3.3	Validation de la structure	38
3.3.1	Estimation de géodésiques	39
3.3.2	Performances en temps de calcul et consommation mémoire	41
3.3.3	Découpage d’une acquisition massive	44
3.4	Conclusion du chapitre	48
3.4.1	Contributions	48
3.4.2	Discussion	49
III	ECHANTILLONNAGE DE NUAGES DE POINTS	51
4	INTRODUCTION	53
4.1	Caractéristiques des acquisitions 3D	53
4.2	Etat de l’art du ré-échantillonnage de nuages de points	55
4.2.1	Simplification de nuages de points	56

4.2.2	Amélioration de l'échantillonnage	57
4.3	Echantillonnage en disques de Poisson	59
4.4	Objectifs de cette partie	61
5	ECHANTILLONNAGE EN DISQUES DE POISSON PAR GRAPHEs LOCAUX	63
5.1	Echantillonnage d'une acquisition	63
5.1.1	Echantillonnage adaptatif	67
5.1.2	Traitement du bruit d'acquisition	70
5.2	Echantillonnage d'un ensemble d'acquisitions	73
5.2.1	Résultats visuels	75
5.2.2	Garanties théoriques	78
5.2.3	Analyse de la qualité de l'échantillonnage	80
5.2.4	Performances en temps de calcul et consommation mémoire	85
5.3	Gestion des contraintes temps/mémoire	85
5.3.1	Echantillonnage de tuiles avec chevauchements	85
5.3.2	Echantillonnage d'acquisitions sous-échantillonnées	87
5.4	Application à la reconstruction de surfaces	90
5.5	Conclusion du chapitre	91
5.5.1	Contributions	92
5.5.2	Discussion	95
	IV DIAGRAMME DE VORONOÏ DE NUAGES DE POINTS	97
6	INTRODUCTION	99
6.1	Diagrammes de Voronoï	99
6.1.1	Diagrammes de Voronoï centroïdaux	102
6.1.2	Relaxation de Lloyd	103
6.2	Objectifs de cette partie	103
7	DIAGRAMME DE VORONOÏ DÉFINI PAR MORCEAUX	105
7.1	Bassin et configuration locale d'un site	105
7.2	Diagramme de Voronoï globalement cohérent	108
7.3	Conclusion du chapitre	110
8	DIAGRAMME DE VORONOÏ CENTROÏDAL PAR GRAPHEs LOCAUX	113
8.1	Diagramme de Voronoï centroïdal d'une acquisition	113
8.1.1	Construction d'un diagramme de Voronoï	113
8.1.2	Relaxation de la position des sites	115
8.1.3	Métrique considérée	116
8.2	Diagramme de Voronoï centroïdal d'un ensemble d'acquisitions . . .	116
8.2.1	Génération de partitions locales	117
8.2.2	Transmission des partitions locales entre tous les graphes . .	118
8.2.3	Génération d'une partition globale	118
8.2.4	Relaxation de l'ensemble des sites	120

8.2.5	Importance de l'initialisation de la position des sites	121
8.3	Résultats expérimentaux	121
8.3.1	Amélioration de la qualité de l'échantillonnage	122
8.3.2	Performances en temps de calcul et consommation mémoire	122
8.3.3	Réduction du coût mémoire	126
8.3.4	Reconstruction de surface	127
8.4	Conclusion du chapitre	130
8.4.1	Contributions	130
8.4.2	Discussions	132
V	CONCLUSION	135
9	CONCLUSION GÉNÉRALE	137
9.1	Contributions	137
9.2	Perspectives	138
VI	ANNEXES	143
A	DOCUMENTATION COMPLÉMENTAIRE	145
	BIBLIOGRAPHIE	147

TABLE DES FIGURES

FIGURE 1	En seulement quelques décennies, la taille des ordinateurs a considérablement été réduite tandis que leur puissance n'a cessé d'augmenter de manière exponentielle. A gauche, le premier ordinateur entièrement électronique, l'ENIAC. Image tirée de [U.S. ARMY, 1947-1955]. A droite, le très récent micro-ordinateur de IBM. Image fournie par IBM.	3
FIGURE 2	Exemple de rendu photo-réaliste. Il est éminemment difficile de remarquer que cette image a été générée par un ordinateur tant le degré de réalisme est élevé. Image fournie par REDWAY3D.	4
FIGURE 3	Numérisation 3D de la statue du David de Michel-Ange, réalisée au sein du <i>Digital Michelangelo Project</i>	5
FIGURE 4	Numérisation 3D de la Tholos de Delphes, par des chercheurs du MAP (Modèles et simulations pour l'architecture et le patrimoine, unité CNRS/MCC).	5
FIGURE 5	Exemple de scanner à contact. Image fournie par FARO®.	6
FIGURE 6	Exemple d'image obtenue à l'aide d'outils de thermographie. Image tirée de [PASSIVHAUS INSTITUT, 2006].	7
FIGURE 7	Exemple d'un système de stéréovision. Image tirée de [PEYROT, 2014].	8
FIGURE 8	Rotations verticale et horizontale d'un LiDAR terrestre. Image tirée de [FARO®, 2011].	9
FIGURE 9	En haut, Exemple d'une carte de profondeur obtenue à partir d'un LiDAR terrestre (issue de la donnée <i>St Trophime</i>). En bas, à titre de visualisation nous montrons la couleur de la surface pour chaque pixel valide de la carte de profondeur.	10
FIGURE 10	Le projet <i>Open Heritage</i> de Google et <i>CyArk</i> met à disposition plus d'une trentaine de campagnes d'acquisition ayant été réalisées dans le monde entier. En haut, rendu 3D de l'intérieur de la Maison de rencontre à Waitangi en Nouvelle Zélande. En bas, photo du temple Wat Phra Si Sanphet à Ayutthaya en Thaïlande. Images tirées de [<i>Open Heritage</i>].	14
FIGURE 11	Schéma de la hiérarchie de sphères englobantes de QSplat. Image tirée de [RUSINKIEWICZ et LEVOY, 2000].	21

FIGURE 12	RBP d'un cercle à différents niveaux de quantification (à gauche 5 bits, à droite 10 bits) et différentes densités d'échantillonnage (en haut $\frac{2\pi}{32}$, en bas $\frac{2\pi}{1024}$). En haut l'erreur d'approximation est dominée par la distance entre les échantillons alors qu'en bas elle est dominée par la quantification. Une bonne représentation est obtenue si la densité d'échantillonnage est du même ordre de grandeur que le niveau de quantification (en haut à gauche et en bas à droite). Image tirée de [BOTSCH et al., 2002].	22
FIGURE 13	Dans le travail de [BOTSCH et al., 2002], l'information nécessaire pour reconstruire chaque niveau peut être codée de manière compacte à l'aide de codes sur 4 bits (à gauche). Ainsi, la grille la plus fine peut être encodée de manière hiérarchique par une séquence de codes sur 4 bits (à droite). Image tirée de [BOTSCH et al., 2002].	22
FIGURE 14	Exemple de représentation sous forme d'octree. L'adaptivité de la structure permet de subdiviser certaines cellules plus que d'autres. Image tirée de [WHITETIMBERWOLF, 2010].	23
FIGURE 15	Mise à jour de la représentation multi-résolution de [WAND et al., 2008]. Image tirée de [WAND et al., 2008].	24
FIGURE 16	Principales étapes de notre approche de structuration de nuages de points par graphes locaux.	29
FIGURE 17	Cas typique d'une connectivité dans le domaine de paramétrisation, fourni par la carte de profondeur, qui n'est pas fidèle à la topologie de la surface capturée : deux objets dans la scène capturée sont projetés sur des pixels voisins, mais l'arête rouge ne doit pas exister dans le graphe.	31
FIGURE 18	Principales étapes pour construire un graphe à partir d'une carte de profondeur. Pour des raisons de visualisation, la résolution des graphes représentés est fortement réduite.	33
FIGURE 19	Equivalence entre la paramétrisation d'une surface et le processus d'acquisition.	34
FIGURE 20	Exemple schématique de la mise en correspondance de sommets de deux graphes représentant le même morceau de surface.	36
FIGURE 21	Exemple schématique de la mise en correspondance de sommets de deux graphes représentant différents morceaux de surface.	36

FIGURE 22	Mise en correspondance de deux cartes de profondeur et gestion des superpositions (donnée <i>Eim Ya Kyaung</i>). (a) Deux cartes de profondeur ont été générées par un processus d'acquisition. (b) La couleur de chaque pixel représente l'appartenance d'un sommet à un graphe particulier (jaune en haut et bleu en bas). La zone hachurée correspond aux sommets ayant été capturés par les deux acquisitions. (c) En comparant les densités d'échantillonnage de chaque point, chaque graphe identifie les sommets avec une densité plus élevée que leurs sommets correspondants dans l'autre graphe. Les couleurs de la colonne de droite représentent, pour chaque sommet, le graphe dans lequel des simulations doivent être effectués pour cette région du nuage de points.	38
FIGURE 23	Placement des caméras et acquisitions synthétiques, générant une carte de profondeur pour chaque point de vue. Image tirée de [PIETRONI et al., 2011].	40
FIGURE 24	Comparaison entre un calcul de géodésique réalisé sur la surface d'un maillage surfacique (à gauche) et sur un ensemble de graphes locaux (à droite) pour différents modèles (<i>Egea</i> en haut et <i>Armadillo</i> en bas). Les distances à partir d'un même point sont représentées par des couleurs allant de bleu (faible distance) à rouge (forte distance).	42
FIGURE 25	Découpage d'une carte de profondeur en un ensemble de tuiles avec chevauchements. Le bord de chaque tuile a été identifié avec une couleur différente.	45
FIGURE 26	Gestion de la superposition pour des tuiles issues de la même carte de profondeur. En 1, l'image de départ est découpée en deux tuiles bleu et jaune avec une zone de chevauchement (zone hachurée). En 2, les superposition sont gérées en identifiant pour chaque graphe l'ensemble des sommets étant plus éloignés du bord commun entre les deux graphes.	46
FIGURE 27	Différence entre le calcul de géodésiques à partir d'une seule acquisition (à gauche) et sur un ensemble de tuiles (à droite) de la donnée <i>St Trophime</i> . Les distances à partir d'un même point sont représentées par des couleurs allant de bleu (faible distance) à rouge (forte distance). Au milieu, les sommets V_i^+ des différents graphes $\hat{G}_i \in \hat{G}$ issus de chaque tuile sont identifiés avec une couleur différente pour chaque graphe. .	48

FIGURE 28	Exemple de la forte anisotropie présente dans une acquisition de la donnée <i>Facade</i> . La distribution des points suit des directions privilégiées, liées en partie au balayage du laser.	54
FIGURE 29	En haut, exemple des importantes variations de densité pouvant être présentes dans un nuage de points issus de l'agrégation de plusieurs acquisitions du modèle <i>Techshop</i> . En bas, la méthode proposée permet entre autres, d'uniformiser la densité de la distribution finale.	55
FIGURE 30	Exemple de contraction d'arêtes. L'arête uv est contractée de manière à ce que le sommet u soit confondu avec le sommet v . Toutes les arêtes incidentes au sommet u sont maintenant incidentes au sommet v	56
FIGURE 31	Exemple d'échantillonnage en disques de Poisson avec les disques associés à certains échantillons.	60
FIGURE 32	Donnée <i>Wat Phra Si Sanphet</i> , ré-échantillonné de manière adaptative avec notre approche basée graphes. La donnée originale contient plus de 5 milliards de points résultant de l'agrégation de 156 acquisitions différentes. Nous sommes capable de représenter cette donnée avec seulement 2,8% du nombre de points original tout en préservant de fins détails de la surface (comme des briques dans le mur d'une zone couvrant plus de 40 000m ²).	64
FIGURE 33	Biais de mesure entre un disque prenant en compte la métrique euclidienne (à gauche) et la métrique géodésique (à droite). Image tirée de [PEYROT et al., 2015].	66
FIGURE 34	Distribution obtenue avec un échantillonnage en disques de Poisson maximal d'une acquisition de <i>St Trophime</i> , obtenu avec notre approche par graphes locaux. De gauche à droite : vue de face et vue de côté. La croissance des disques dans les zones présentant des occultations est naturellement stoppée grâce à la construction des graphes du Chapitre 3, ne connectant pas les sommets ayant de fortes variations d'intensité dans la carte de profondeur.	66
FIGURE 35	Normale N en un point de la surface défini comme le produit vectoriel des deux vecteurs tangents X_1 et X_2 (définissant le plan tangent à la surface en ce point). Illustration tirée de [CRANE et al., 2013].	68

FIGURE 36	Différence entre un échantillonnage uniforme (à gauche) et adaptatif (à droite) de la donnée <i>Intérieur</i> . Les deux nuages de points contiennent approximativement le même nombre d'échantillons. On peut remarquer, entre autres, que l'échantillonnage adaptatif, en concentrant plus d'échantillons dans les régions de fortes courbures, permet de faire ressortir visuellement plus de détails.	69
FIGURE 37	Schéma du filtrage bilatéral. Illustration inspirée de [DURAND et DORSEY, 2002].	71
FIGURE 38	Impact du filtrage bilatéral appliqué à une carte de profondeur du modèle <i>St Strophime</i> sur la qualité du nuage de points (haut : avant filtrage ; bas : après filtrage).	72
FIGURE 39	Estimation de normales au sein d'un nuage de points à l'aide de l'approche proposée, pour différentes tailles du noyau G_{σ_s} du filtre bilatéral.	73
FIGURE 40	Principe général de notre approche. Chaque couleur identifie une zone traitée par un graphe donné.	74
FIGURE 41	Les nuages de points générés par notre approche ne souffrent pas de distorsion en termes de qualité d'échantillonnage par rapport à la donnée d'origine. En haut, deux acquisitions se superposant, amenant à une forte variation de la densité d'échantillonnage dans le nuage de points original (donnée <i>Facade</i>). En bas, deux distributions uniformes, à différentes densités, obtenues avec notre approche basée graphes. Pour des raisons de visualisation, de faibles densités d'échantillonnages cibles ont été choisies.	75
FIGURE 42	Distributions uniformes générées par notre approche sur différents modèles (de haut en bas : <i>Armadillo</i> , <i>Dragon</i> et <i>Venus</i>) à 3 différentes densités cibles.	76
FIGURE 43	Comparaison de distributions uniformes (à gauche) et adaptatives (à droite) sur différents modèles (de haut en bas : <i>Armadillo</i> et <i>Dragon</i>). Pour chaque modèle, les deux distributions contiennent le même nombre d'échantillons.	77

FIGURE 44	Distribution adaptative générée à partir d'une campagne d'acquisition massive (<i>Eim Ya Kyaung</i>). En haut, on peut voir l'ensemble du site ré-échantillonné par notre approche à 0,6% du nombre initial de points. La deuxième ligne présente un morceau du nuage de points original (à gauche) et le résultat de notre algorithme d'échantillonnage à 0,6% du nombre initial de points (à droite). Grâce à l'utilisation d'une métrique adaptative dépendante de la courbure de la surface, la réduction du nombre de points n'empêche pas la préservation de fins détails géométriques du site numérisé.	78
FIGURE 45	Photographie (en haut) et distribution résultant de notre algorithme d'échantillonnage (en bas) à partir de la donnée <i>Intérieur</i> .	79
FIGURE 46	Visualisation de la qualité d'une distribution adaptative générée à partir de la donnée <i>Eim Ya Kyaung</i> .	79
FIGURE 47	A partir d'un périodogramme moyen (à gauche), deux métriques 1D peuvent être extraites : le SPMR (au centre) et l'anisotropie (à droite). Image tirée de [WEI et WANG, 2011].	81
FIGURE 48	Le travail de [WEI et WANG, 2011] permet de généraliser la méthode classique de Fourier à l'analyse d'échantillonnages uniformes (en haut) et non-uniformes (en bas), afin de pouvoir extraire des informations pertinentes dans les deux cas. Pour chaque ligne, à gauche l'analyse est réalisée en utilisant l'équation (12) et à droite l'expression en utilisant l'équation (13). L'analyse à l'aide de la transformation de Fourier est incapable d'extraire des informations pertinentes pour l'analyse d'échantillonnages non-uniformes. Image tirée de [WEI et WANG, 2011].	82
FIGURE 49	Qualité des distributions obtenues avec [BOWERS et al., 2010] (en haut) et avec notre méthode (en bas) pour un même rayon $r = 0.004$ à partir du modèle <i>Bunny</i> .	83
FIGURE 50	Qualité des distributions obtenues avec [BOWERS et al., 2010] (en haut) et avec notre méthode (en bas) pour un même rayon $r = 0.004$ à partir du modèle <i>Venus</i> .	84
FIGURE 51	Vue rapprochée d'échantillonnages générés sur le même morceau de surface. De gauche à droite : échantillonnage réalisé sur un unique graphe, échantillonnage réalisé sur un ensemble de graphes et le même échantillonnage en associant une couleur différente identifiant pour chaque échantillon dans quel graphe celui ci a été sélectionné.	84

FIGURE 52	Temps de calcul et qualité d'échantillonnage obtenus à partir de cartes de profondeur à différentes résolutions du modèle <i>Venus</i> . Les périodogrammes et SPMR sont montrés pour différentes résolutions des cartes de profondeur pour un échantillonnage de haute densité ($r = 0.004$, sur la droite) et de basse densité ($r = 0.02$, sur la gauche).	89
FIGURE 53	Echantillonnage adaptatif obtenu par notre approche (en haut) et surface générée à partir de cet échantillonnage en utilisant l'approche de [BOLTCHÉVA et LEVY, 2017] (en bas).	92
FIGURE 54	Différentes vues d'une reconstruction obtenue à partir d'un échantillonnage adaptatif montrant la conservation des fins détails de la surface (<i>Eim Ya Kyaung</i>).	93
FIGURE 55	Comparaison d'une photo et de la reconstruction obtenue à partir d'un échantillonnage adaptatif généré par notre approche sur <i>Facade</i>	94
FIGURE 56	Reconstruction obtenue à partir d'un échantillonnage adaptatif généré par notre approche sur <i>Facade</i>	94
FIGURE 57	Exemples de diagrammes de Voronoï présents dans la nature. A droite, les alvéoles d'un nid d'abeille. Image tirée de [MACKINTOSH, 2011]. A gauche, les ailes d'une libellule. Image tirée de [NONENMACHER, 2003].	99
FIGURE 58	Illustration de la décomposition de l'espace de Descartes en tourbillons de matière. Image tirée de [DESCARTES, 1644].	100
FIGURE 59	Schéma de la dualité entre le diagramme de Voronoï (en rouge) et la triangulation de Delaunay (en noir). Image tirée de [HFEREE, 2011].	102
FIGURE 60	Diagramme de Voronoï à partir d'une position arbitraire de sites (en noir) et triangulation de Delaunay duale associée (en rouge). La non uniformité de la distribution des sites crée des cellules ayant des formes et des aires très différentes et la triangulation associée est composée de triangles ayant des aspect/ratio très disparates.	102
FIGURE 61	Diagramme de Voronoï centroïdal (en noir) obtenu après convergence de l'algorithme de Lloyd en considérant la norme L_2 dans le plan et triangulation de Delaunay duale associée (en rouge).	104

FIGURE 62 A gauche, le demi-espace $D(s_i, s_j)$ (en bleu clair) est construit à partir des sites s_i et s_j . Au centre, la cellule C_i (en bleu) est construite à partir de l'intersection de tous les demi-espaces $D(s_i, s_k), \forall s_k \in S', k \neq i$ d'un sous-ensemble de sites $S' \subseteq S$. A droite, la cellule C_i est construite de la même manière, mais en considérant aussi les sites $\{S \setminus S'\}$ (en orange). Ces sites n'affectent donc pas la définition de la cellule C_i 106

FIGURE 63 De gauche à droite : partition de Voronoï à partir de 6 sites ; bord de chaque boule $\delta B'_v^S$ associées aux sommets v constituant la cellule bleue C_i ; bassin de Voronoï VB_i^S issu de l'union des boules B'_v^S 107

FIGURE 64 Au centre, une partition de Voronoï construite à partir de 6 sites et bassin de Voronoï de la cellule C_i (en bleu). A gauche, la cellule C_i reste définie de la même manière, peu importe le placement des sites et leur nombre en dehors de son bassin de Voronoï VB_i . A droite, il suffit d'un seul site (en orange) rajouté à l'intérieur de son bassin de Voronoï VB_i pour que celui-ci impacte la définition de sa cellule C_i . 107

FIGURE 65 La ligne de balayage (en rouge), permet de définir à chaque instant un diagramme de Voronoï partiel (en bleu et noir) sur le domaine déjà balayé. 108

FIGURE 66 De gauche à droite, morceau du diagramme de Voronoï global \widehat{Vor} défini uniquement à l'aide des sites $S' = S \cap M'$ et du bord du domaine $\delta M'$; diagramme de Voronoï global Vor ; superposition montrant que $\widehat{Vor} \subseteq Vor$ 110

FIGURE 67 Diagramme de Voronoï construit sur une acquisition de la donnée *Templo Mayor*. Le diagramme de Voronoï est construit dans le domaine de paramétrisation fourni par le graphe (à gauche) puis représenté dans l'espace 3D (à droite). 115

FIGURE 68 Les 3 étapes de la création d'un diagramme de Voronoï à partir d'un ensemble de graphes. (a) Un morceau de la partition de Voronoï globale \widehat{Vor} est construit de manière locale . (b) Les morceaux de partition de chaque graphe $\widehat{Vor}_i, \forall G_i \in G$ sont transmis entre tous les graphes. (c) A partir des différents morceaux de partition, une partition globale Vor peut être construite en faisant se rencontrer les différentes partitions locales \widehat{Vor}_i (c). 117

FIGURE 69 Partie du diagramme de Voronoï global $\widehat{Vor}_i \subseteq Vor$ construite à l'aide des sites locaux S_i et des sommets au bord du domaine $V_i^{+\delta} \subseteq V_i^+$ 118

FIGURE 70	Diagrammes de Voronoï de la surface sous-jacente à des nuage de points issus des modèles <i>Egea</i> (à gauche) et <i>Armadillo</i> (à droite).	120
FIGURE 71	Diagramme de Voronoï de la surface sous-jacente à un nuage de points issu du modèle <i>Armadillo</i> avant (à gauche) et après convergence de l’algorithme de Lloyd (à droite).	120
FIGURE 72	Différence visuelle entre une distribution uniforme générée avec l’approche présentée dans le Chapitre 5 (à gauche) et après convergence de l’algorithme de Lloyd (à droite). En haut et en bas deux densités différentes du modèle <i>Armadillo</i> .	123
FIGURE 73	Différence visuelle entre une distribution uniforme générée avec l’approche présentée dans le Chapitre 5 (à gauche) et après convergence de l’algorithme de Lloyd (à droite). En haut et en bas deux densités différentes du modèle <i>Dragon</i> .	123
FIGURE 74	Analyse différentielle de distributions générées avec l’approche présentée dans le Chapitre 5 avant (en haut) et après 10 itérations de l’algorithme de Lloyd (en bas). En haut et en bas deux densités différentes du modèle <i>Bunny</i>	124
FIGURE 75	Analyse différentielle de distributions générées avec l’approche présentée dans le Chapitre 5 avant (en haut) et après 10 itérations de l’algorithme de Lloyd (en bas). En haut et en bas deux densités différentes du modèle <i>Venus</i>	124
FIGURE 76	Comparaison visuelle de la qualité d’une distribution adaptative avant (à gauche) et après (à droite) relaxation des sites de la donnée <i>Meeting House</i>	125
FIGURE 77	Comparaison visuelle de la qualité d’une distribution adaptative avant (à gauche) et après (à droite) relaxation des sites de deux zones distinctes de la donnée <i>Eim Ya Kyaung</i>	125
FIGURE 78	Reconstructions obtenues avec l’approche de [BOLTCHÉVA et LEVY, 2017] (à gauche) et notre approche (à droite) sur les modèles <i>Egea</i> (en haut) et <i>Armadillo</i> (en bas). Il est intéressant de noter que les triangulations générées sont quasiment les mêmes (à quelques retournements d’arêtes près, à cause de l’approximation faite par notre approche sur le calcul de distances géodésiques). Lorsque la surface d’origine présente du détail mais que l’échantillonnage fournit n’est pas suffisamment dense, la méthode de [BOLTCHÉVA et LEVY, 2017] a tendance à trianguler ensemble des éléments qui sont en réalité assez éloignés sur la surface d’origine (au niveau des mains et des pieds du modèle <i>Armadillo</i>).	129

FIGURE 79	Reconstructions obtenues avec l'approche de [BOLTCHÉVA et LEVY, 2017] (en haut) et notre approche (en bas) sur la donnée <i>Meeting House</i> . Contrairement à notre méthode, l'approche des auteurs de [BOLTCHÉVA et LEVY, 2017] a tendance à interpoler les données, faisant perdre du détail à certains endroits (sur le sol) et extrapolant la surface à des endroits où celle-ci ne devrait pas exister (au niveau des fenêtres). . .	130
FIGURE 80	Qualité de triangulations générées par notre approche à partir d'échantillonnages adaptatifs de la donnée <i>Meeting House</i> . Triangulations générées avant (en haut) et après convergence de l'algorithme (au milieu). En bas, nous pouvons voir une vue rapprochée des deux triangulations, avant (à gauche) et après convergence de la relaxation (à droite).	131

LISTE DES TABLEAUX

TABLE 1	Temps de calcul et pic mémoire atteints pour la structuration de différentes campagnes d’acquisition. En-dessous du pic mémoire est indiqué le pourcentage par rapport au coût mémoire total si l’ensemble des acquisitions étaient effectivement en mémoire.	43
TABLE 2	Temps de calcul et pic mémoire atteints pour la structuration de différentes données synthétique avec et sans gestion de la mémoire. Chaque acquisition a été générée une résolution de 1024 par 1024 pixels.	45
TABLE 3	Temps de calcul et pic mémoire atteints par notre algorithme de structuration en fonction de la taille des tuiles. A côté des temps de calcul, le nombre total de tuiles issues du découpage est indiqué entre parenthèses. Le pourcentage de temps de calcul indique le coût rapport aux performances atteintes avec les acquisition sans découpage. Le pourcentage de pic mémoire indique le pic atteint par rapport au traitement des acquisitions sans découpage.	47
TABLE 4	Temps de calcul et pic mémoire atteints par notre algorithme de ré-échantillonnage pour différentes campagnes d’acquisition à différentes densités cible. Pour chaque échantillonnage, nous indiquons le pourcentage de points restants par rapport au nombre de points initial. Les temps de calcul incluent le temps de lecture/écriture depuis/vers le disque dur.	86
TABLE 5	Temps de calcul et pic mémoire atteints par notre algorithme de ré-échantillonnage pour une même densité cible (indiqué sous la forme du pourcentage de points restants, à côté du Nom) pour différentes tailles de tuiles (8192x8192, 4096x4096 et 2048x2048). Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à leur taille native.	87

TABLE 6	Temps de calcul et pic mémoire de notre algorithme de ré-échantillonnage pour différentes résolutions de cartes de profondeur pour une faible densité cible. Res. 1/1 indique que les cartes de profondeur sont à la résolution native, Res. 1/2 et Res. 1/4 indiquent que les cartes de profondeur considérées ont respectivement une résolution 2 fois et 4 fois inférieure à la résolution native. Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à la résolution native. Etant donné que la réduction de la résolution des cartes de profondeur impacte à la fois les étapes de structuration et d'échantillonnage, les performances mesurées représentent la somme de ces deux étapes. 90
TABLE 7	Temps de calcul atteints par notre algorithme de construction de diagramme de Voronoï pour différentes données synthétiques. Pour chaque donnée nous montrons les temps de calcul pour des distributions de sites à deux densités différentes. Pour un ensemble de sites nous montrons le temps de construction d'un diagramme de Voronoï en utilisant l'approche présentée et le temps de calcul total pour que la relaxation converge ainsi que le nombre d'itérations nécessaires pour atteindre la convergence. 126
TABLE 8	Temps de calcul et pic mémoire atteints par notre algorithme de construction d'un diagramme de Voronoï global pour différentes tailles de tuiles (4096x4096 et 2048x2048). Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à leur taille native. 127
TABLE 9	Modèles 3D à partir desquels des acquisitions synthétiques ont été générées 145
TABLE 10	Campagnes d'acquisition issues de [<i>Open Heritage</i>]. 146
TABLE 11	Campagne d'acquisition issue de [<i>ReCap Community Data Center</i>]. 146
TABLE 12	Campagnes d'acquisition fournies par <i>Art Graphique&Patrimoine</i> . 146

Première partie

PRÉFACE

INTRODUCTION

1.1 INTRODUCTION GÉNÉRALE

Ce serait un euphémisme que de dire que les ordinateurs, et l'informatique de manière générale, sont des éléments essentiels de notre vie quotidienne. En seulement quelques décennies, nous sommes passés d'ordinateurs occupant une pièce entière à des ordinateurs plus petits qu'un grain de sel, tandis que la puissance de ceux-ci n'a fait que croître (Figure 1).

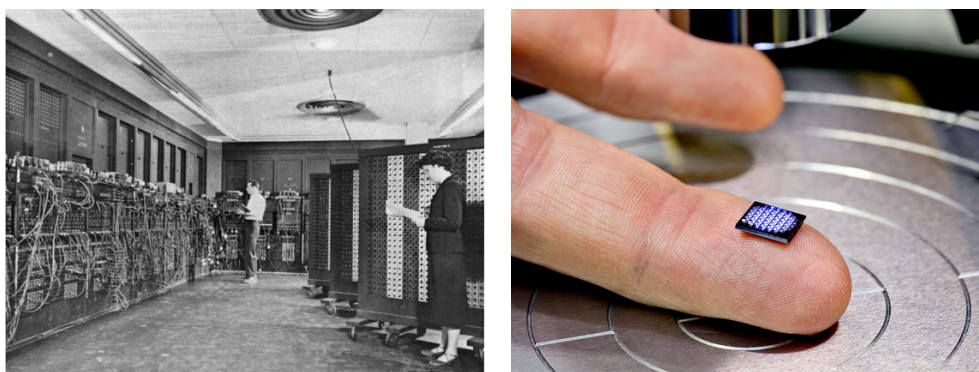


FIGURE 1 – En seulement quelques décennies, la taille des ordinateurs a considérablement été réduite tandis que leur puissance n'a cessé d'augmenter de manière exponentielle. A gauche, le premier ordinateur entièrement électronique, l'ENIAC. Image tirée de [U.S. ARMY, 1947-1955]. A droite, le très récent micro-ordinateur de IBM. Image fournie par IBM.

Les capacités des machines actuelles permettent d'intriquer les ordinateurs et les humains, afin de réaliser une grande variété de tâches de manière conjointe. Si à l'origine les ordinateurs se destinaient à réaliser des tâches basiques de manière plus efficace que des humains, ceux-ci peuvent aujourd'hui, avec par exemple les énormes progrès de l'intelligence artificielle, accomplir des tâches extrêmement complexes, certaines que l'Homme lui-même est incapable de réaliser. De ce fait, les ordinateurs ne sont plus simplement un moyen de faciliter et d'accélérer certains traitements. Ils deviennent primordiaux à la réalisation de certaines opérations.

De plus, l'usage de l'outil informatique s'est ouvert à tellement de domaines qu'il est difficile aujourd'hui d'en citer certains n'en tirant aucun avantage. Pour les plus connus, nous pouvons notamment citer les industries du cinéma et du

jeu vidéo, dont l'évolution des technologies a permis aux acteurs de ces domaines de réaliser des applications extrêmement élaborées et réalistes (Figure 2), amenant une véritable ambiguïté sur l'existence physique des éléments de telles scènes.



FIGURE 2 – Exemple de rendu photo-réaliste. Il est éminemment difficile de remarquer que cette image a été générée par un ordinateur tant le degré de réalisme est élevé. Image fournie par REDWAY3D.

D'autres domaines s'intéressent particulièrement à l'utilisation de technologies informatiques pour la représentation numérique de scènes réelles. Par exemple, dans le domaine de la conservation du patrimoine historique, le grand défi depuis plusieurs décennies consiste à numériser différentes oeuvres historiques. Si cela a commencé par l'utilisation de la photographie numérique, aujourd'hui les principaux intéressés s'orientent plus vers la *numérisation 3D* d'oeuvres réelles.

1.2 NUMÉRISATION 3D

De manière générale, la numérisation 3D est à la géométrie ce que la photographie est à la couleur. En effet, la numérisation 3D permet d'obtenir une représentation numérique de la forme géométrique d'objets du monde réel. Grâce à ce procédé, il est possible de créer de manière automatique (ou semi-automatique) des maquettes numériques d'éléments du monde réel, au lieu de les reproduire manuellement, à l'aide d'outils de *Conception Assistée par Ordinateur (CAO)* par exemple.

Les intérêts d'une telle technologie sont nombreux, et donner une liste exhaustive de ces intérêts paraît être compliqué, tant il y a d'usages de cette technologie que nous n'imaginons pas encore. Nous pouvons toutefois noter le fort intérêt de

cette technologie ces dernières années dans les domaines de la Modélisation des Informations du Bâtiment (MIB) (*Building Information Modeling (BIM)* en anglais), ou encore de la conservation du patrimoine historique.

La numérisation 3D s'est tout d'abord intéressée à la capture d'objets isolés, telles que des statues (Figure 3).

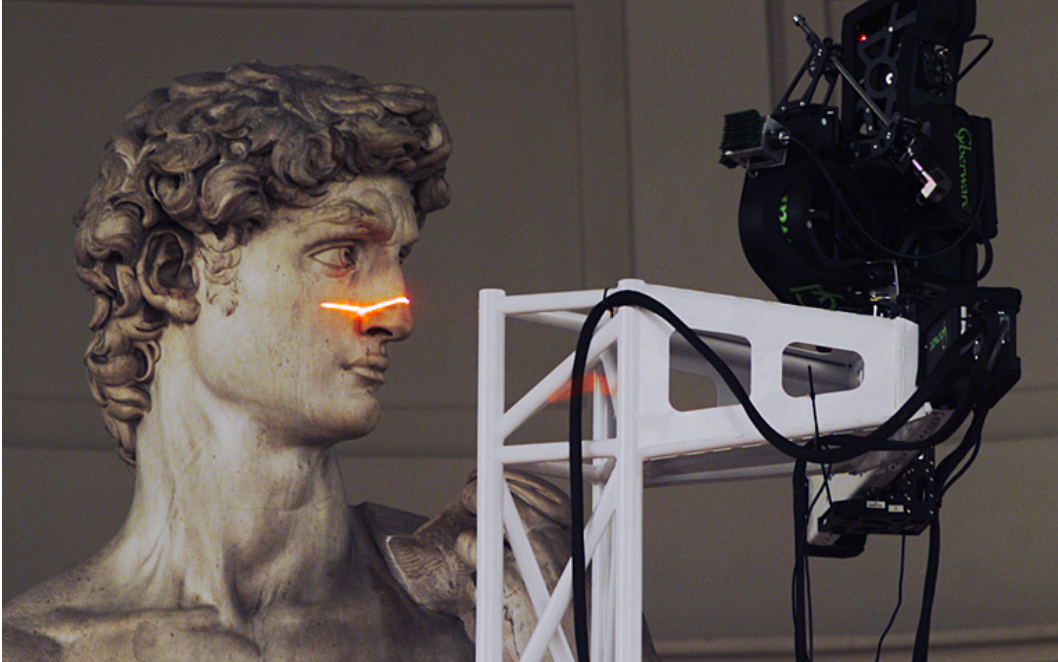


FIGURE 3 – Numérisation 3D de la statue du David de Michel-Ange, réalisée au sein du *Digital Michelangelo Project*.

Ensuite, celle-ci s'est étendue à la capture de scènes comme une salle ou un monument (Figure 4).



FIGURE 4 – Numérisation 3D de la Tholos de Delphes, par des chercheurs du MAP (Modèles et simulations pour l'architecture et le patrimoine, unité CNRS/MCC).

Aujourd'hui, les scènes numérisées peuvent représenter tout un site historique, faisant jusqu'à plusieurs milliers de mètres carrés, tout en représentant une information extrêmement détaillée. Les différents outils ont fortement évolués au fur

et à mesure des années entraînant ces changements dans leurs usages. Il est donc logique que les méthodes manipulant ces données évoluent elles aussi.

1.2.1 *Scanners 3D*

Il existe de nombreuses méthodes de numérisation 3D, et cette grande variété a amené les scientifiques à établir toute une taxonomie les concernant. Celles-ci peuvent être découpées en deux branches distinctes, celles des méthodes avec contact et celles des méthodes sans contact. Nous allons faire une brève description des différentes méthodes existantes, mais nous invitons le lecteur à se référer au travail de [CURLESS et SEITZ, 2000] pour obtenir plus de détails.

AVEC CONTACT Cette branche comprend l'ensemble des méthodes capturant des points à l'aide d'un contact physique entre le périphérique d'acquisition et la surface à numériser. Cela peut par exemple se faire à l'aide d'un stylet doté d'une bille (Figure 5), qui mesure la position de points sur la surface par rapport à un repère de référence.



FIGURE 5 – Exemple de scanner à contact. Image fournie par FARO®.

Ces méthodes sont capables d'obtenir des mesures très précises (de l'ordre du micromètre) mais uniquement sur des surfaces de petite taille, étant donnée la lenteur de l'acquisition (de l'ordre de quelques points par seconde). De plus, ces méthodes génèrent des distributions très éparées, sous-échantillonnant fortement la surface numérisée. Elles ont néanmoins l'avantage d'être insensibles aux problèmes de réflectance des surfaces.

SANS CONTACT Comme leur nom l'indique, ces méthodes ne nécessitent pas au capteur d'être en contact physique avec l'objet à numériser, contrairement aux

méthodes précédentes. Elles s'intéressent à capturer des signaux (optiques ou non-optiques), pouvant être à l'origine émis par les périphériques d'acquisition, et à inférer une position géométrique aux échantillons acquis à partir des relevés effectués. Ce type d'acquisition a l'avantage de pouvoir être effectué à une certaine distance des éléments à numériser, de manière plus rapide que les méthodes avec contact. Nous pouvons distinguer les méthodes optiques, c'est-à-dire les méthodes se basant sur la lumière, et les méthodes non-optiques, comme par exemple avec la thermographie, générant des images de chaleur des éléments d'une scène (Figure 6).

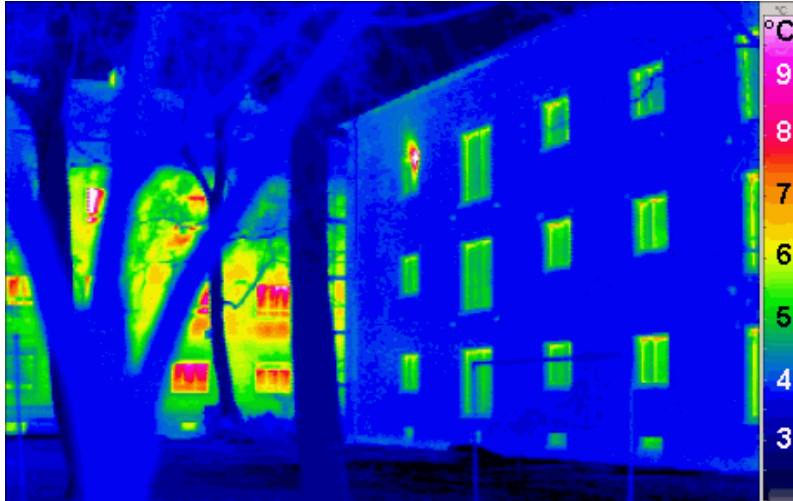


FIGURE 6 – Exemple d'image obtenue à l'aide d'outils de thermographie. Image tirée de [PASSIVHAUS INSTITUT, 2006].

Par la suite, nous ne décrivons que les méthodes optiques, puisque celles-ci sont les plus utilisées et sont plus précises que les méthodes non-optiques.

De manière générale, les systèmes optiques utilisent la lumière émise par les éléments d'une scène pour en déterminer leur géométrie. Deux approches se distinguent : les approches dites "passives" et les autres dites "actives".

Passive. Cette catégorie concerne l'ensemble des méthodes permettant d'acquérir un signal sans émission du périphérique d'acquisition, uniquement en capturant l'information lumineuse présente dans une scène. Dans celle-ci nous pouvons notamment citer :

- la *stéréovision*, qui, à partir de deux images prises par un appareil photo à deux points de vue différents, permet d'inférer la forme 3D des éléments capturés (Figure 7);
- la *stéréo multi-vues*, ayant permis la généralisation du principe de stéréovision à la reconstruction de formes 3D à partir d'un ensemble d'images RGB [FURUKAWA et HERNÁNDEZ, 2015].



FIGURE 7 – Exemple d’un système de stéréovision. Image tirée de [PEYROT, 2014].

Active. Les périphériques d’acquisition optique actifs, contrairement aux périphériques passifs, émettent une impulsion lumineuse et analysent le comportement de l’information lumineuse réfléchiée par la surface pour en extrapoler la géométrie de la scène. Nous pouvons citer entre autres :

- les systèmes à *triangulation*, qui émettent des impulsions lumineuses dans une scène, puis observent l’image des points projetés à travers un capteur pour en connaître la position 3D [FOREST COLLADO, 2004];
- les systèmes à *temps de vol*, qui émettent aussi des impulsions lumineuses dans une scène, mais mesurent ensuite le temps nécessaire pour que cette information leur revienne, après réflexion sur la surface de la scène. Etant donnée la vitesse de déplacement de la lumière, ce temps de vol est directement lié à la distance entre le système et la scène. On peut donc en déduire la position 3D d’un point sur sa surface [KOLB et al., 2010].

Parmi les systèmes à temps de vol, on trouve les populaires systèmes *LiDAR* (*Light Detection And Ranging* en anglais). Ces systèmes peuvent être terrestres (positionnés à même le sol, ou dans un véhicule en mouvement) ou aéroportés (par un avion ou un drone par exemple). Durant cette thèse, nous nous sommes principalement intéressés aux LiDARs terrestres. C’est pourquoi, une section spécifique leur est dédiée ci-dessous.

1.2.2 LiDARs terrestres

Les LiDARs terrestres sont particulièrement efficaces car ils sont capables de capturer la position de très nombreux échantillons à différentes orientations verticales et horizontales (Figure 8), selon un principe de balayage à l’aide d’un émetteur laser et d’un miroir. Ainsi, ils fournissent une information structurée concernant la manière dont les points ont été acquis. Cette donnée est généralement présentée sous la forme d’une grille ordonnant les échantillons en fonction des angles azimu-

taux et polaires auxquels ceux-ci ont été capturés, que l'on appelle généralement *carte de profondeur* (*depth map/range image* en anglais).

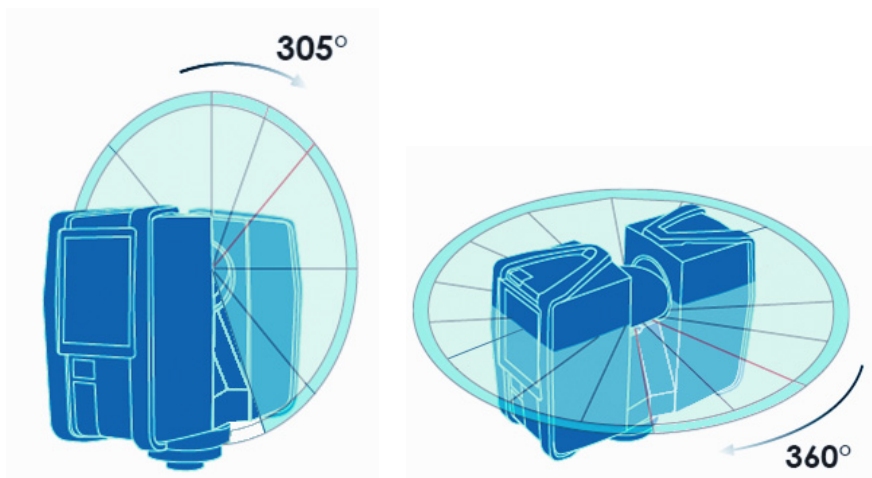


FIGURE 8 – Rotations verticale et horizontale d'un LiDAR terrestre. Image tirée de [FARO®, 2011].

CARTES DE PROFONDEUR Une carte de profondeur, notée par la suite D , est une image 2D dont l'intensité des pixels représente la distance entre les points acquis dans une scène et la position du système d'acquisition. Elle est composée de *pixels valides*, c'est-à-dire les points capturés, et de *pixels invalides*, représentant les directions sans information de profondeur (direction dans laquelle le capteur n'a pas reçu de retour du laser). Dans ce cas, les pixels sont identifiés comme ayant une intensité nulle (Figure 9).

Géométriquement, une carte de profondeur est équivalente à un nuage de points. Pour établir cette équivalence, il suffit de connaître la *fonction de projection* P qui définit la manière dont un point (en 3D) du nuage de points se retrouve *projeté* sur un pixel (en 2D) de la carte de profondeur lors de la numérisation d'une scène. Cependant, une carte de profondeur met en avant une certaine structure reflétant la topologie de la surface acquise, information perdue lors du traitement direct de nuages de points. En effet, lors d'une acquisition, un scanner 3D capture des points sur la surface des objets présents dans la scène, visibles depuis l'emplacement du scanner. Le domaine de définition de ces points n'est donc pas l'espace euclidien 3D, mais bien la surface sur laquelle ces points ont été capturés. Cette structure est une information essentielle pour le traitement de la *surface sous-jacente* à un nuage de points, au lieu de *l'espace ambiant* dans lequel le nuage de points est plongé.

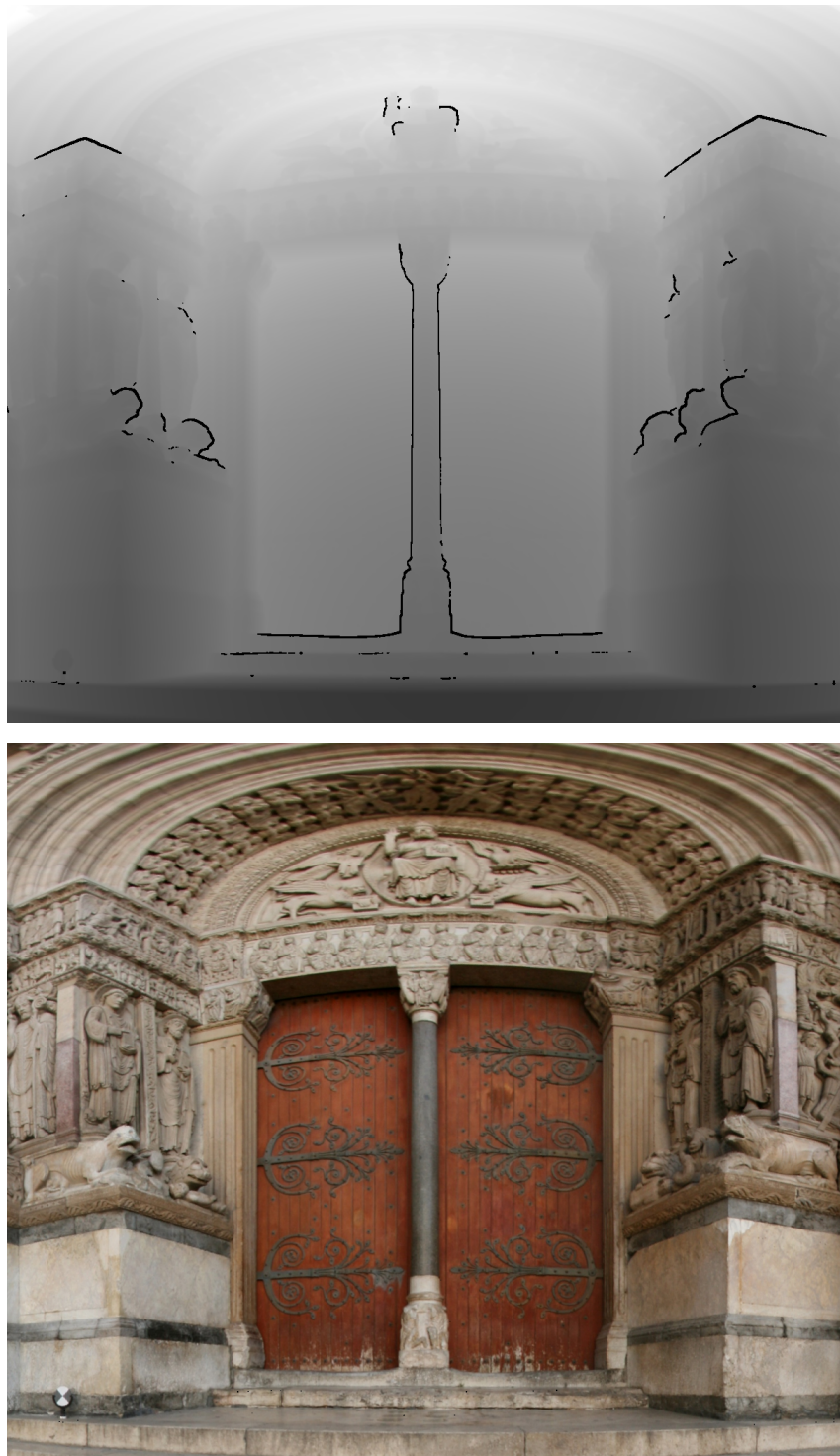


FIGURE 9 – En haut, Exemple d’une carte de profondeur obtenue à partir d’un LiDAR terrestre (issue de la donnée *St Trophime*). En bas, à titre de visualisation nous montrons la couleur de la surface pour chaque pixel valide de la carte de profondeur.

FONCTIONS DE PROJECTION De manière générale, nous définissons les fonctions de projection comme étant une combinaison de deux opérations :

- un changement de repère, permettant de passer du *repère monde* au *repère scanner* ;
- une réduction de dimension, pour passer du *repère scanner* à *l'espace image*.

Changement de repère. Chaque acquisition est représentée dans un repère local centré en la position du scanner lors de l'acquisition. Tel quel, il est donc impossible d'agréger l'information provenant de différentes acquisitions étant donné que les points de différentes acquisitions sont exprimés dans des repères différents. Il faut donc un moyen d'effectuer des changements de repères, afin de pouvoir représenter un ensemble d'acquisitions dans le même repère géométrique. Pour se faire, des *matrices de transformation* sont utilisées. Elles permettent de représenter l'ensemble des transformations affines dans un espace euclidien, c'est-à-dire des translations et des rotations (6 degrés de liberté). En connaissant ces matrices de transformation, il est aisé de réaliser des changements de repère.

Pour une question de simplification d'écriture, nous représentons une matrice de transformation sous la forme d'une combinaison entre une matrice de rotation R et un vecteur de translation T . Soit $p_w = (x, y, z)$ un point 3D du système de coordonnées cartésiennes dans le *repère monde*. Ses coordonnées dans un système de coordonnées cartésiennes dans le *repère scanner* lors de la m -ème acquisition est obtenue en utilisant une matrice de rotation R_m et un vecteur de translation T_m :

$$\begin{pmatrix} x_m \\ y_m \\ z_m \end{pmatrix} = p_m = R_m(p_w - T_m).$$

En pratique, ces matrices sont soit connues directement à l'acquisition, grâce à des systèmes de positionnement associés à un scanner (comme un GPS par exemple), soit obtenues de manière calculatoire, grâce à des algorithmes de recalage. Dans le cadre de cette thèse, nous considérons que les acquisitions sur lesquelles nous travaillons sont déjà recalées entre elles, et nous invitons les lecteurs à se diriger sur des états de l'art récents [BELLEKENS et al., 2014; TAM et al., 2013] pour obtenir plus d'informations concernant le problème du recalage de nuages de points rigide et non-rigide.

Réduction de dimension. La position (i, j) d'un pixel et son intensité $\mathcal{J}(p)$ représentent respectivement l'angle azimutal θ et l'angle polaire φ , et la distance radiale r d'un système de coordonnées sphériques centré en la position d'un scanner au moment de l'acquisition.

A partir d'un point p_m dans le *repère scanner* de la m -ème acquisition, une conversion dans un système de coordonnées sphériques peut-être effectuée :

$$\begin{aligned} r_m &= \sqrt{x_m^2 + y_m^2 + z_m^2}, \\ \theta_m &= \tan^{-1} \left(\frac{y_m}{x_m} \right), \\ \varphi_m &= \cos^{-1} \left(\frac{z_m}{r_m} \right). \end{aligned}$$

La carte de profondeur D correspond à une *projection equirectangulaire* de ces coordonnées sphériques sur un plan. Il ne s'agit pas, à proprement parler, d'une projection, puisque celle-ci se limite à considérer les angles θ_m et φ_m directement comme les coordonnées cartésiennes du point dans le plan projectif. De ce fait, les coordonnées de ce point dans la carte de profondeur (i_m, j_m) peuvent simplement être obtenues en quantifiant θ_m et φ_m respectivement, par rapport aux résolutions horizontales et verticales de la carte de profondeur D .

1.3 CONTEXTE GÉNÉRAL DE CETTE THÈSE

Cette thèse a été financée par la Région Provence Alpes Côte d'Azur et réalisée en partenariat avec *Cintoo*¹, spin-off du laboratoire I3S de l'Université de Nice Sophia Antipolis et du CNRS, dans le cadre d'une bourse de la Région Provence Alpes Côte d'Azur. Cette start-up développe depuis 2013 des technologies et des solutions pour exploiter des données 3D venant de la numérisation du monde réel. L'activité de la société s'est spécialisée depuis 2016 dans le traitement d'acquisitions issues de LiDARs terrestres pour la construction dans le domaine du MIB.

Dans ce domaine, la capacité d'acquisition qu'offrent les LiDARs (terrestres et aéroportés) rend cette technologie très efficace lorsqu'il s'agit d'acquérir de grandes surfaces, ou de numériser des sites difficiles d'accès. La modélisation virtuelle de bâtiments est une étape essentielle car elle permet d'obtenir des représentations fidèles aux constructions réelles. L'obtention automatique ou semi-automatique de ces représentations peut donc représenter un fort gain de temps par rapport à la construction manuelle de représentations virtuelles, mais peut aussi garantir une fidélité de la donnée numérique par rapport à la donnée réelle. Ces représentations peuvent ainsi être utilisées par exemple à des fins d'études d'urbanisme ou architecturales, dans des simulations physiques ou encore pour l'aide à la navigation routière [BRÉDIF, 2010].

Dans le domaine de la numérisation du patrimoine historique, de nombreuses initiatives sont en cours pour stocker et partager de manière numérique de nom-

1. <https://cintoo.com/fr/index.html>

breuses oeuvres à travers le monde entier. On peut citer de multiples applications à l'utilisation d'une telle technologie dans ce domaine comme la visualisation d'oeuvres numérisées, l'archivage numérique, la restauration numérique, la réplique physique ou encore l'observation de sites historiques [PIERACCINI et al., 2001]. Très récemment, *Google* et *CyArk*, au travers du projet *Open Heritage*² ont mis à disposition un ensemble de campagnes d'acquisitions qui ont été réalisées durant les dernières années dans plus d'une trentaine de sites historiques (Figure 10). La quantité de données générées est absolument gigantesque, représentant près d'un téra-octet d'archives numériques. Certaines de ces acquisitions datent de plus de 10 ans, et pourtant, aujourd'hui encore, la plupart de ces données sont encore difficiles à utiliser tellement les nuages de points représentés par chaque campagne d'acquisition sont volumineux.

Durant ce travail, nous nous focalisons tout particulièrement sur des données géométriques issues de campagnes d'acquisition utilisant des LiDARs terrestres. En une seule acquisition il est possible de capturer plusieurs centaines de millions de points. Étant donné qu'il faut un grand nombre d'acquisitions pour représenter la surface de sites de grande taille (afin de capturer les zones occultées par la disposition des éléments dans le site numérisé), les nuages de points générés peuvent facilement atteindre de plusieurs milliards de points.

Cette quantité de points à traiter est un des problèmes majeurs obstruant l'utilisation des données acquises. De ce fait, il est commun pour les utilisateurs devant traiter de telles données de travailler :

- soit sur une représentation globale fortement sous-échantillonnée obtenue en choisissant un nombre de points donné de manière aléatoire,
- soit sur une représentation locale à pleine résolution.

Par conséquent, les utilisateurs sont obligés de faire un choix entre avoir une vue globale de la scène numérisée, ou une vue détaillée. Par ailleurs, le nombre de points que les utilisateurs sont capables de manipuler est directement lié aux capacités de leurs ordinateurs, contraignant l'utilisation de machines très puissantes.

Un autre problème dans les données numérisées concerne la qualité de l'échantillonnage obtenu après une campagne d'acquisition. Du bruit (lié aux conditions d'acquisition), des artefacts d'échantillonnage (liés à la manière dont les échantillons sont capturés) sont quelques exemples des imperfections pouvant être présentes dans les données brutes.

2. <https://artsandculture.google.com/project/cyark>



FIGURE 10 – Le projet *Open Heritage* de Google et *CyArk* met à disposition plus d’une trentaine de campagnes d’acquisition ayant été réalisées dans le monde entier. En haut, rendu 3D de l’intérieur de la Maison de rencontre à Waitangi en Nouvelle Zélande. En bas, photo du temple Wat Phra Si Sanphet à Ayutthaya en Thaïlande. Images tirées de [*Open Heritage*].

1.3.1 Motivations

Notre travail est motivé par la réalisation de plusieurs objectifs cherchant à améliorer la manipulation de campagnes d’acquisition massives par les utilisateurs finaux.

L'ambition générale consiste à proposer une structure unique pour représenter la surface décrite par des nuages de points, pour faciliter les différentes étapes classiques de la chaîne de numérisation 3D - de l'acquisition à la construction - mais aussi la manipulation de maillages polygonaux représentant la surface numérisée. De plus, cette structure doit nous permettre de généraliser et d'adapter des algorithmes existants pour le traitement de modèles 3D afin de manipuler la surface sous-jacente de nuages de points.

Nous souhaiterions aussi construire des *algorithmes adaptatifs*, c'est-à-dire des algorithmes dont le comportement peut-être adapté en fonction des capacités de la machine sur laquelle ils sont réalisés.

1.3.2 Contributions

Notre première contribution, qui sert de fil rouge à travers l'ensemble des travaux réalisés durant cette thèse, concerne la modélisation de la surface sous-jacente de nuages de points massifs à l'aide de **graphes locaux**. En se basant sur la structure d'acquisitions réalisées par des LiDARs terrestres, des graphes sont construits, représentant chacun de manière locale la connectivité de la surface capturée par une acquisition. La modélisation de la surface globale est obtenue en mettant en correspondance les graphes représentant les mêmes régions de la scène numérisée. De par sa nature locale, cette structure est à même de fournir un moyen de traiter des données peu importe leur taille, en s'appuyant sur le découpage naturel des données numérisées.

Notre seconde contribution concerne le **ré-échantillonnage** de nuages de points massifs. A l'aide d'un ensemble de graphes locaux, nous montrons comment généraliser un algorithme d'échantillonnage en disques de Poisson, afin de l'utiliser pour réduire la quantité de points représentant la surface complète, malgré la taille des données brutes d'une campagne d'acquisition impossibles à stocker entièrement en mémoire.

Notre troisième contribution concerne la **construction de diagrammes de Voronoï centroïdaux** sur la surface sous-jacente à des nuages de points. En se servant de la structure de graphes locaux, nous proposons une manière de construire un diagramme de Voronoï globalement cohérent de la surface d'un ensemble d'acquisitions, à partir de diagrammes de Voronoï calculés localement au sein de chaque graphe.

L'avantage de notre approche est que les particularités de la structure basée graphes et la manière dont les différents algorithmes ont été mis en oeuvre, permettent de réaliser ces traitements sur des nuages de points gigantesques qui ne sont normalement pas manipulables par des ordinateurs classiques, en offrant un compromis entre le temps de calcul et le pic de consommation mémoire atteint.

1.3.3 *Organisation du manuscrit*

Ce manuscrit est ordonné en 3 parties distinctes présentant nos différentes contributions, chacune dédiée à l'une d'entre elles.

Dans un premier temps, la Partie II s'intéresse au problème de la modélisation de la surface sous-jacente de nuages de points massifs à l'aide d'un ensemble de graphes. En se basant sur le modèle construit, les deux parties suivantes présentent la généralisation d'algorithmes classiques au traitement de surfaces de modèles 3D au traitement la surface décrite par des nuages de points. Tout d'abord, la Partie III se concentre sur un problème de ré-échantillonnage de nuages de points. Ensuite, la Partie IV propose une étude exploratoire de la construction de diagrammes de Voronoï centroïdaux discrets de la surface sous-jacente à un nuage de points, avec comme application possible l'amélioration de la qualité d'un échantillonnage ainsi que la reconstruction de surfaces.

Finalement, la partie V résume l'ensemble des contributions de cette thèse et propose un ensemble de perspectives aux travaux réalisés.

Deuxième partie

STRUCTURATION DE NUAGES DE POINTS

INTRODUCTION

Ce chapitre présente le contexte général de la structuration de nuages de points. Nous y présentons un état de l'art des méthodes proposant des structures spatiales et surfaciques pour la visualisation et la manipulation de nuages de points. Nous mettons finalement en avant les faiblesses des méthodes existantes, afin d'introduire les différents objectifs que nous nous fixons ainsi que les contributions proposées permettant d'atteindre ces objectifs.

2.1 ETAT DE L'ART DE LA STRUCTURATION DE NUAGES DE POINTS

Les nuages de points sont devenus, en l'espace de trois décennies, une structure omniprésente dans la représentation de scènes 3D numérisées. En effet, étant donné que la plupart des systèmes d'acquisition n'acquièrent que des échantillons sur la surface des éléments du monde réel, les nuages de points sont la représentation par défaut de ces acquisitions. Leur facilité d'acquisition aujourd'hui permet à quasiment n'importe quel utilisateur d'obtenir une représentation digitale 3D d'objets du monde réel. Certains scanners sont même capables aujourd'hui d'acquérir plusieurs centaines de millions de points en une seule acquisition. Etant donné que certaines campagnes d'acquisitions peuvent nécessiter plusieurs dizaines voire centaines d'acquisitions, les nuages de points à traiter se retrouvent rapidement composés de plusieurs milliards de points.

Au début des années 2000, la tendance des *Représentations Basées Points* (RBP) (*Point-Based Representations* en anglais) a émergé. L'idée était de remplacer les représentations volumiques et surfaciques par une représentation dont la primitive était le point. Les RBP définissent des formes géométriques par des nuages suffisamment denses d'échantillons sur leur surface. L'idée était de s'affranchir des difficultés liés à la gestion de la topologie pour des surfaces finement échantillonnées, afin de rendre leur manipulation plus efficace et plus flexible. L'intérêt des RBP provient du fait que leur complexité est décorrélée de la complexité de la surface. Pour représenter précisément des formes complexes, un maillage doit être composé d'un très grand nombre de faces et généralement de taille assez petite, tandis qu'une RBP pourra représenter les mêmes formes avec un nombre réduit de primitives. Cependant, cela peut aussi être un inconvénient, dans le cas où des maillages sont composés de peu de zones de forte courbure, car un maillage aura besoin de très peu de faces pour représenter des zones planes, tandis qu'une

RBP aura besoin d'un beaucoup plus grand nombre de primitives pour donner l'impression de visualiser une surface pleine.

Les RBP se sont principalement concentrées sur la résolution de deux problèmes liés aux nuages de points.

La qualité de visualisation. En utilisant des méthodes de "splatting" [ZWICKER et al., 2001], représentant chaque point projeté sur l'écran comme une ellipse et en mélangeant les couleurs associées à chacune de ces ellipses, ces méthodes sont capables de rendre l'apparence d'une surface "pleine" bien que la donnée visualisée soit un ensemble d'échantillons indépendants.

La structuration de nuages de points. Afin de manipuler les nuages de points de manière efficace pour offrir une visualisation de données interactive, il est nécessaire de construire une structure de données permettant d'obtenir des connexions entre les différents points, ainsi qu'un moyen d'avoir une représentation progressive de la donnée, afin de ne raffiner que certaines parties d'un nuage de points, dépendant des zones visualisées.

Cette thèse ne couvrant pas le problème de la visualisation de nuages de points, dans la suite de ce chapitre, nous ne détaillerons que la partie liée à la structuration de nuages de points. Nous invitons le lecteur à se référer aux travaux de [SCHEIBLAUER, 2014; SCHÜTZ, 2016] pour obtenir plus d'informations concernant le problème de visualisation de nuages de points massifs.

L'organisation d'un nuage de points est une étape tout sauf triviale. Premièrement, chaque point est une entité indépendante, ayant des attributs propres. Deuxièmement, le domaine de définition d'un nuage de points est une surface (celle de la scène qui a été numérisée), mais chaque point est représenté à l'aide d'une position dans l'espace 3D.

En conséquence, la notion de *voisinage* d'un point est difficile à définir. S'agit-il d'un voisinage spatial ou d'un voisinage surfacique? Jusqu'à quand un point est-il "voisin" d'un autre?

Pourtant, les requêtes de voisinage sont des opérations essentielles à quasiment tous les algorithmes. Par conséquent, la construction de ce voisinage est une étape cruciale pour le bon déroulement des traitements subséquents. De plus, de telles requêtes sont des opérations très coûteuses ($\mathcal{O}(n^2)$ en nombre de points), et peuvent drastiquement impacter les performances d'algorithmes lorsque celles-ci sont effectuées à de nombreuses reprises sans structure adaptée. Finalement, sans notion de voisinage, il est aussi difficile de pouvoir réduire la quantité de points tout en gardant l'information pertinente dans la donnée d'origine.

C'est pour ces raisons que la construction d'une structure au sein d'un nuage de points est un passage obligatoire.

2.1.1 Méthodes spatiales

La quasi-totalité des méthodes de l'état de l'art de la structuration de nuages de points construit des représentations structurant l'espace ambiant autour d'un nuage de points.

Le premier travail ayant proposé une structure pour la visualisation interactive de données massives est *QSplat* [RUSINKIEWICZ et LEVOY, 2000], mis en place durant le *Digital Michelangelo Project*. Les auteurs ont proposé d'utiliser une hiérarchie de sphères englobantes pour représenter un nuage de points (les sommets d'un maillage dense) à différents niveaux de détail (Figure 11).

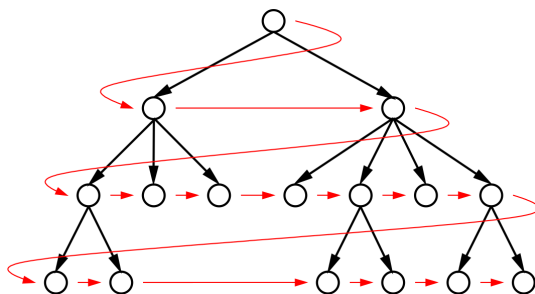


FIGURE 11 – Schéma de la hiérarchie de sphères englobantes de QSplat. Image tirée de [RUSINKIEWICZ et LEVOY, 2000].

Bien que les auteurs réduisent drastiquement le coût des attributs associés à chaque nœud de la hiérarchie (48 bits par nœud, incluant la position, la normale et la couleur associées à ce nœud), des rendus de très bonne qualité sont obtenus. Leur idée a été de se servir de la hiérarchie construite comme un moyen de pouvoir faire un rendu de manière récursive, en ne parcourant que les sphères impactant le rendu final, permettant ainsi de ne pas traiter une grosse partie du nuage de points à chaque fois.

Le travail de [BOTSCH et al., 2002], en s'appuyant sur la *théorie de l'approximation*, a mis en évidence le compromis entre densité d'échantillonnage et quantification (Figure 12). Afin d'obtenir une représentation compacte en mémoire, les auteurs ont proposé un schéma de codage basé sur une séquence de grilles imbriquées encodant les informations relatives à la configuration des cellules filles sous la forme de codes binaires (Figure 13). Leur représentation hiérarchique, équivalente à un octree, permet de réduire le coût d'un nuage de points à $\frac{1}{3}$ de son coût original.

Ce travail traite le nuage de points de manière descendante en construisant une grille fine en premier et en obtenant des représentations de plus en plus grossières.

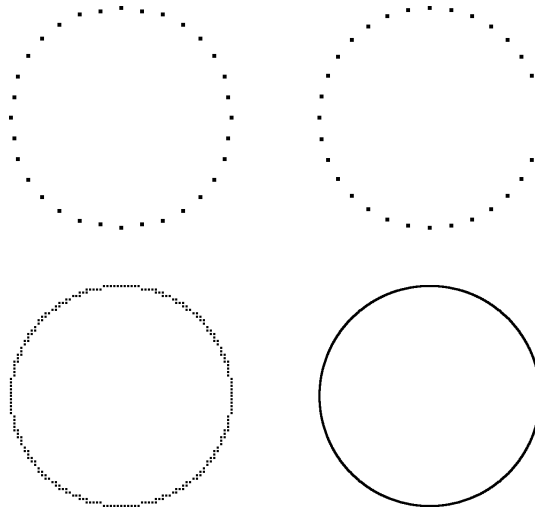


FIGURE 12 – RBP d’un cercle à différents niveaux de quantification (à gauche 5 bits, à droite 10 bits) et différentes densités d’échantillonnage (en haut $\frac{2\pi}{32}$, en bas $\frac{2\pi}{1024}$). En haut l’erreur d’approximation est dominée par la distance entre les échantillons alors qu’en bas elle est dominée par la quantification. Une bonne représentation est obtenue si la densité d’échantillonnage est du même ordre de grandeur que le niveau de quantification (en haut à gauche et en bas à droite). Image tirée de [BOTSCH et al., 2002].

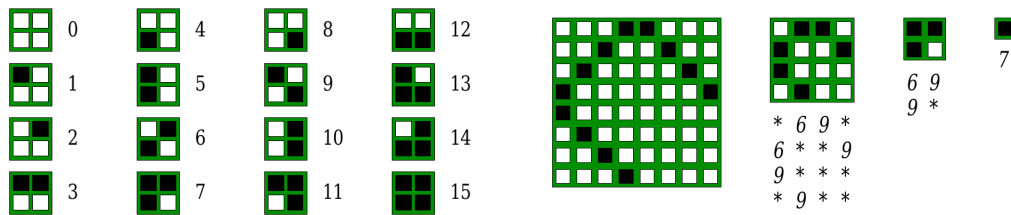


FIGURE 13 – Dans le travail de [BOTSCH et al., 2002], l’information nécessaire pour reconstruire chaque niveau peut être codée de manière compacte à l’aide de codes sur 4 bits (à gauche). Ainsi, la grille la plus fine peut être encodée de manière hiérarchique par une séquence de codes sur 4 bits (à droite). Image tirée de [BOTSCH et al., 2002].

De ce fait, la construction d’une première grille représentant précisément le nuage de points en entrée est bien trop coûteuse pour des données massives. Cependant, l’utilisation de techniques de splatting leur permet de considérer des grilles moins fines, et de combler les faibles densités d’échantillonnage par des ellipses de taille plus importante, leur permettant d’obtenir des visualisations de plutôt bonne qualité malgré la discrétisation effectuée.

Par la suite, le travail de [DACHSBACHER et al., 2003] s’est intéressé à construire une version séquentielle de la structure hiérarchique de [RUSINKIEWICZ et LEVOY, 2000] sous la forme d’un parcours en largeur d’abord. Ceci leur permet de transférer tous les points au GPU, afin que celui-ci ne traite qu’un morceau de mémoire

contigu pour représenter le nuage de points à un niveau de détail donné, permettant de dessiner un très grand nombre de splats par seconde et de réduire les échanges entre GPU et CPU.

Le problème est qu'aujourd'hui, beaucoup plus de traitements que la simple visualisation sont réalisés sur ces nuages de points denses et complexes, et la solution que les RBP fournissent au problème de visualisation n'est pas réellement généralisable aux autres traitements pouvant être effectués sur ces données.

Les travaux suivants se sont principalement intéressés à l'utilisation de représentations sous forme d'arbres de partition spatiale pour la structuration de nuages de points. S'inspirant des travaux de [BOTSCH et al., 2002], les octrees ont connu un très fort engouement les années suivantes, et encore aujourd'hui des travaux exploitant cette structure de données sont publiés. Un octree permet, *a contrario* des grilles voxelliques, de représenter différentes parties d'un nuage de points à différentes densités, en allouant plus de niveaux de profondeur dans les zones contenant plus d'information (Figure 14).

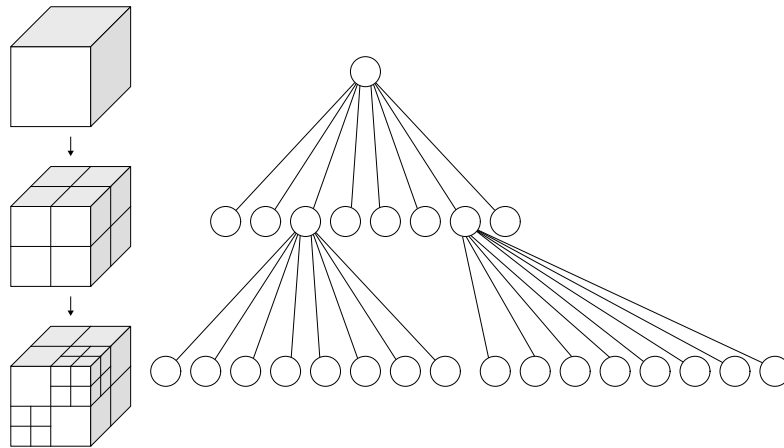


FIGURE 14 – Exemple de représentation sous forme d'octree. L'adaptivité de la structure permet de subdiviser certaines cellules plus que d'autres. Image tirée de [WHITE-TIMBERWOLF, 2010].

La versatilité, la simplicité d'implémentation et l'efficacité de cette structure en ont fait une représentation de choix pour les traitements appliqués aux nuages de points, au vu de la forte quantité de publications à ce propos [HUANG et al., 2008; KAMMERL et al., 2012; PAJAROLA, 2003; PAJAROLA et al., 2005; SCHNABEL et KLEIN, 2006, ...].

Malgré la nette dominance en popularité des octrees, d'autres types d'arbres ont tout de même été considérés, comme des arbres de partition binaire [BORDIGNON et al., 2006; GOBBETTI et MARTON, 2004], des arbres couvrants de poids minimal [GUMHOLD et al., 2005; MERRY et al., 2006] ou encore des arbres k-d [LARI et al., 2011].

De manière générale, l'idée essentielle de ces méthodes consiste à remplacer le nuage de points par la structure construite. Cette opération permet d'obtenir une représentation beaucoup plus compacte que la donnée originale, avec un certain contrôle sur l'erreur de quantification en fonction de la précision de l'arbre.

La quantification effectuée ne tenant pas compte de la topologie de la donnée numérisée, ces structures ne fournissent aucune garantie quant à la position d'un échantillon par rapport à la surface d'origine. Les points de l'échantillonnage final peuvent donc se retrouver "en-dehors" de la surface sous-jacente au nuage de points initial. Cependant, combinés à l'utilisation de méthodes de splatting pour le rendu, utilisant la normale pour orienter l'ellipse autour de chaque point, l'impact de ces erreurs lors de la visualisation peut être assez négligeable. Par contre, celles-ci ne sont pas acceptables quand il s'agit d'effectuer des mesures au sein du nuage de points.

Ainsi, des représentations "quasi sans-pertes" ont vu le jour, cherchant une représentation compacte pour le nuage de points sans trop altérer la position des échantillons d'origine. C'est le cas du travail de [WAND et al., 2008]. Les auteurs proposent une structure multi-résolution dynamique, permettant de mettre à jour la structure au fur et à mesure de l'ajout des données. Leur travail consiste à fournir deux structures imbriquées : une représentant la structuration globale à l'aide d'un octree couvrant le nuage de points complet, et une représentant la structure "locale" dans chaque feuille de l'octree grâce à une grille régulière stockant la position des points contenus dans cette feuille (Figure 15).

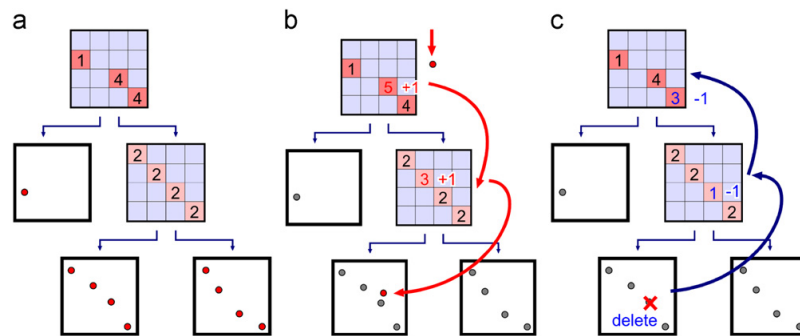


FIGURE 15 – Mise à jour de la représentation multi-résolution de [WAND et al., 2008]. Image tirée de [WAND et al., 2008].

Proposant, une approche différente de l'utilisation d'octrees, la méthode de [ELSEBERG et al., 2013] est intéressante sur différents aspects. Premièrement, elle propose une représentation sous forme d'octree drastiquement optimisée en termes de consommation mémoire, en ne conservant que les informations ne pouvant pas être calculées à partir d'autres informations présentes dans l'arbre. Ainsi, pour chaque noeud de l'arbre, ils arrivent à fournir une représentation extrêmement compacte, passant de 100 octets par noeuds - pour les approches octree "clas-

siques" - à seulement 8 octets pour représenter exactement les mêmes informations. Deuxièmement, au lieu de conserver les points d'origine du nuage dans les cellules feuilles de l'octree, ils proposent de stocker ces points comme un décalage à partir de la cellule feuille dans laquelle ils sont contenus. Si cette approche quantifie donc *in fine* la position des points, l'erreur est infime et le nuage de points obtenu à la fin est quasiment identique au nuage de points d'origine. L'effort des auteurs de représenter chaque cellule de manière la plus compacte possible a rendu possible la représentation et la manipulation de nuages de points composés de plus d'un milliard de points.

Toutes les méthodes présentées précédemment ont pourtant une particularité en commun qui permet de remettre en question leur utilisation pour la représentation d'échantillons acquis depuis une surface. En effet, ces méthodes fournissent une structuration de l'espace ambiant autour du nuage de points, et non pas une structuration de la surface sous-jacente à celui-ci. En d'autres termes, la construction de ces structures est entièrement décorrélée de la topologie de la donnée numérisée. Si la discrétisation spatiale est suffisamment fine, le voisinage entre les cellules feuilles représente le voisinage surfacique. Cependant il n'existe pas, à notre connaissance, une borne inférieure indiquant à partir de laquelle le pas de discrétisation est suffisamment fin pour garantir une telle hypothèse.

Ces méthodes se sont intéressées à obtenir des structures permettant de représenter des données de manière performante, en termes de temps de calcul et de pic de consommation mémoire, dans un but de pouvoir interagir en temps réel avec les structures construites. Cependant, lorsque le but n'est pas simplement la performance, mais aussi la préservation des informations géométriques et topologiques contenues dans la donnée, de telles approches sont sous-optimales par nature.

2.1.2 Méthodes surfaciques

Il est intéressant de noter que très peu de méthodes se sont intéressées à la représentation de la surface sous-jacente d'un nuage de points autrement que dans un contexte de reconstruction de surface. Or, encore aujourd'hui, il s'agit d'un problème particulièrement difficile à résoudre, en particulier lors de la manipulation de nuages de points composés de plusieurs centaines de millions d'échantillons.

Nous pouvons cependant noter les travaux du début des années 2000, notamment ceux de [PAULY et GROSS, 2001] et [OCHOTTA et SAUPE, 2004], qui ont cherché à partitionner un nuage de points en morceaux de surfaces pouvant être représentées sous la forme d'un champ de hauteur. Bien que n'étant pas réellement des méthodes de structuration de nuage de points, ces méthodes ont proposé des so-

lutions pour paramétrer localement la surface sous-jacente à un nuage de points. Dans le cas de [PAULY et GROSS, 2001], chaque morceau de surface local est analysé par une transformation de Fourier à fenêtre glissante, et les fréquences de ce patch sont modifiées à l'aide de différents filtres classiques au Traitement du Signal, modifiant ainsi l'aspect de la surface décrite par le nuage de points localement. Les auteurs de [OCHOTTA et SAUPE, 2004] se sont quant à eux intéressés à l'utilisation d'une transformée en ondelettes sur chaque patch, afin de réaliser une allocation de bits optimale dans un contexte de compression.

Très récemment, le travail de [CHEN et al., 2018a] propose une idée intéressante : construire un graphe au sein d'un nuage de points en connectant les points étant les plus proches les uns des autres.

Les auteurs de [CHEN et al., 2018a] construisent un graphe représentant l'information géométrique locale à l'aide d'une *matrice d'adjacence* $W \in \mathbb{R}^{N \times N}$, où N représente le nombre de points. Soit $X_i \in \mathbb{R}^3$ la position 3D du $i^{\text{ème}}$ point. Le poids de l'arête entre deux points X_i et X_j est donné par :

$$W_{ij} = \begin{cases} e^{-\frac{\|X_i - X_j\|_2^2}{\sigma^2}} & , \|X_i - X_j\|_2 \leq \tau; \\ 0 & , \text{sinon} \end{cases} \quad (1)$$

où la variance σ et le seuil τ sont des paramètres permettant de définir respectivement la variation des poids dans le voisinage considéré ainsi que la distance maximum des voisinages considérés. Cette formulation crée donc une arête entre tous les sommets qui sont à distance τ au maximum, et dont le poids varie en fonction de la proximité géométrique des points à connecter. En considérant uniquement une information géométrique pour chaque point, le graphe construit pourrait ne pas être représentatif du comportement de la surface, de la même manière que les méthodes de structuration spatiale, présentées précédemment. Cependant, la notion de proximité exprimée dans leur travail peut être étendue pour prendre en considération d'autres attributs (normale, couleur, etc.).

Construire un graphe représentant la connectivité de la surface au sein d'un nuage de points semble fortement similaire à l'idée de construire un maillage. Cependant le premier problème est beaucoup plus simple, puisqu'il s'abstrait des restrictions liées aux maillages surfaciques.

Une fois le graphe construit, leur idée est d'utiliser des méthodes du Traitement du Signal appliquées aux graphes dans le but de ré-échantillonner un nuage de points. En effet, depuis plusieurs années maintenant, de plus en plus de recherches s'intéressent à la généralisation d'algorithmes de Traitement du Signal à des représentations dont la connectivité n'est pas aussi régulière que \mathbb{Z}^d [SHUMAN et al., 2013]. On peut trouver des travaux similaires, du côté du domaine de l'in-

formatique graphique, avec les nombreux algorithmes qui ont été appliqués aux maillages surfaciques, dans la résolution d'*équations aux dérivées partielles (EDP)* sur des connectivités plus génériques qu'une grille cartésienne \mathbb{Z}^d (le lecteur peut par exemple se référer aux ouvrages [PEYRÉ, 2008; SHEFFER et al., 2007] pour obtenir plus de détails sur ces approches). Il est important de noter que les maillages ne sont qu'une instance de graphes, avec des particularités sur leur connectivité (beaucoup d'algorithmes font l'hypothèse que le maillage à traiter est une *2-variété*, avec ou sans bord), pouvant dans certains cas simplifier la généralisation d'algorithmes à ces représentations par rapport aux graphes.

Notons aussi qu'il existe d'autres types de graphes, notamment les *graphes de Reeb* qui infèrent une information topologique sur des formes 3D, fournissant ainsi une représentation intrinsèque de la donnée [BIASOTTI et al., 2008; TIERNY, 2008]. Cependant, dans le cadre de campagnes d'acquisition issues de LiDARs terrestres, la forte quantité de points fournie par de telles acquisitions (plusieurs milliards d'échantillons) ainsi que la forte incomplétude de la surface représentée par le nuage de points, sont autant de raisons qui rendent difficile l'extraction d'une information topologique cohérente.

2.2 OBJECTIFS DE CETTE PARTIE

L'état de l'art semble indiquer que seule une méthode de structuration spatiale est adaptée à la manipulation de données massives. Pourtant, le travail de [CHEN et al., 2018a] montre que la construction d'un graphe sur un nuage de points permet de traiter la surface décrite par ce nuage de points. La force de l'approche basée graphes vient de sa manière de représenter des connexions entre des points représentant une information de connectivité sur la surface numérisée. Simplement, lorsque les traitements à réaliser sur cette connectivité sont principalement locaux, seule une petite partie de la connectivité a besoin d'être connue à un instant donné. Partant de ce constat, notre idée a été de construire un ensemble de connectivités locales et de lier ces différentes connectivités afin de pouvoir transmettre de l'information entre elles pour traiter la donnée de manière globale et conjointe.

Notre objectif consiste donc à fournir une représentation par morceaux de la surface numérisée, permettant la gestion de nuages de points massifs, peu importe le nombre d'acquisitions qui les composent et ainsi que le nombre de points qu'ils représentent.

STRUCTURATION DE NUAGES DE POINTS PAR GRAPHES LOCAUX

Ce chapitre introduit une nouvelle représentation pour la manipulation de nuages de points massifs issus d'acquisitions obtenues par LiDARs terrestres. L'idée générale est de prendre avantage de la structure présente au sein des acquisitions, très proche de la connectivité de la surface numérisée, pour représenter la surface sous-jacente à un nuage de points. S'appuyant sur la structure des acquisitions générées par des LiDARs terrestres, des graphes locaux sont construits et les graphes représentant les mêmes morceaux de surface sont connectés entre eux. Nous montrons que cette structure permet d'évaluer des fonctions sur la surface sous-jacente au nuage de points la représentant. De plus, nous mettons en évidence la considérable réduction de consommation mémoire nécessaire à la manipulation de nuages de points massifs que cette représentation par morceaux fournit de manière naturelle. Enfin nous montrons comment le pic de consommation mémoire peut être contrôlé grâce à un découpage préalable des acquisitions sous la forme de tuiles avec chevauchements.

Le schéma présenté dans la Figure 16 résume les principales étapes de notre approche.

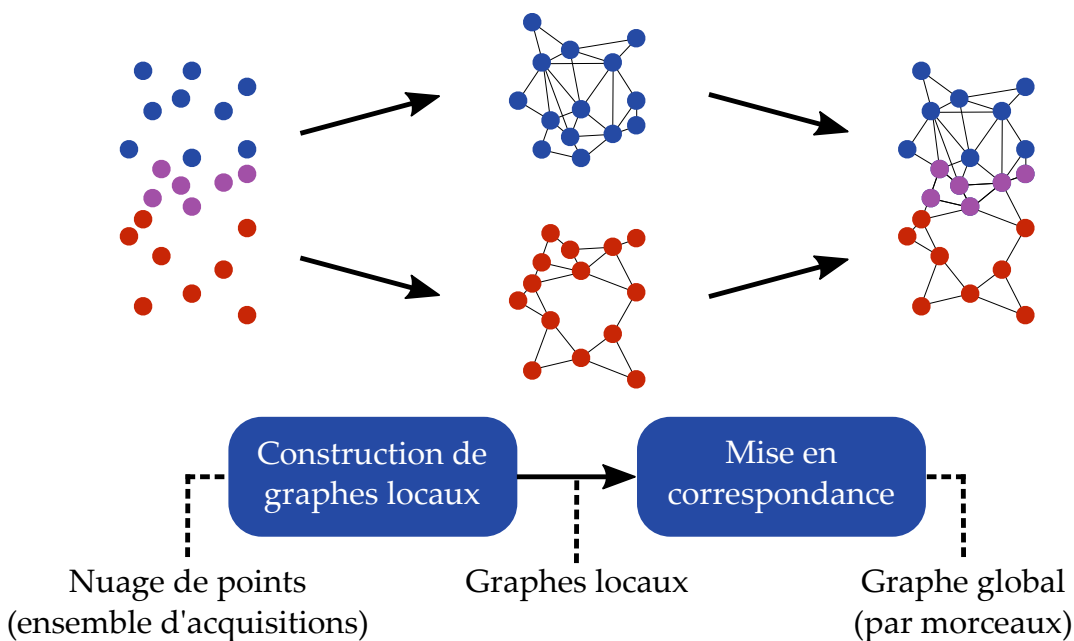


FIGURE 16 – Principales étapes de notre approche de structuration de nuages de points par graphes locaux.

3.1 STRUCTURATION D'UNE ACQUISITION : CONSTRUCTION D'UN GRAPHE LOCAL

Notre travail est très proche de celui des auteurs [CHEN et al., 2018a] dans la manière de traiter le problème de la structuration de nuages de points. Les auteurs ont montré que l'utilisation d'un graphe pour modéliser la surface sous-jacente d'un nuage de points était une méthode pertinente pour la prise en compte de la connectivité de la donnée numérisée lors de son ré-échantillonnage.

Simplement, en considérant des données de taille gigantesque, un graphe souffre fortement de sa complexité en mémoire $\mathcal{O}(|V| + |E|)$ pour stocker entièrement sa structure dans la mémoire vive d'ordinateurs classiques. Ce coût s'abstrayant bien entendu de celui des attributs associés à chaque sommet du graphe, *i. e.*, à chaque point du nuage de points.

Nous pouvons aussi citer le travail de [PIETRONI et al., 2011], où les auteurs se sont intéressés à construire une paramétrisation globale d'un ensemble de paramétrisations locales. Si cette approche est théoriquement la plus intéressante, car permettant d'obtenir *in fine* un unique domaine de paramétrisation sur lequel de nombreux algorithmes peuvent être appliqués, elle est à ce jour incapable de passer à l'échelle afin de pouvoir traiter des nuages de points aussi volumineux que ceux traités dans cette thèse.

S'inspirant de l'idée de représenter la surface sous-jacente de nuages de points à l'aide de graphes, et en considérant la structure fournie par des acquisitions issues de LiDARs terrestres présentée dans le Chapitre 1, nous nous sommes donc demandés s'il était possible de considérer des *graphes locaux* - des graphes représentant *localement* la connectivité de la surface numérisée - au lieu de construire une seule représentation globale. Etant donné que de nombreux traitements ne considèrent qu'une infime partie de la géométrie totale, il n'est pas nécessaire d'avoir la plupart des données en mémoire pour réaliser ces traitements, seuls les points impliqués dans ce traitement ont besoin d'être considérés. Comme présenté dans le Chapitre 1, les LiDARs terrestres fournissent des données sous une forme structurée, et dont la structure se retrouve fortement corrélée à la connectivité de la surface. Par conséquent, cette représentation fournit à la fois un domaine permettant de mieux discriminer entre des points proches dans l'espace euclidien et des points proches sur la surface, ainsi qu'une segmentation naturelle de la donnée.

Alors, plutôt que de considérer un ensemble d'acquisitions uniquement comme leur aggrégation, sous la forme d'un unique nuage de points, nous proposons d'utiliser cette structure pour construire des *graphes locaux*, c'est-à-dire des graphes décrivant localement le comportement de la surface.

Nous rappelons que dans le cas de LiDARs terrestres, la donnée structurée se présente sous la forme d'une carte de profondeur. La représentation qu'offre

une carte de profondeur permet l'utilisation d'algorithmes issus du traitement des images pour interagir avec le nuage de points, en s'appuyant sur la connectivité implicite de la surface fournie par la disposition des pixels dans la carte de profondeur [BIASUTTI et al., 2018]. Telle quelle, la carte de profondeur ne peut pas être transposée sous la forme d'un graphe (considérer sa connectivité comme la connectivité du graphe à construire). Rappelons-nous qu'une acquisition correspond à la projection d'une scène 3D sur un plan 2D. Par conséquent, si on considère un ensemble de pixels voisins autour de chaque pixel, on peut rapidement remarquer que certains voisinages pixelliques ne correspondent pas à des voisinages surfaciques (Figure 17).

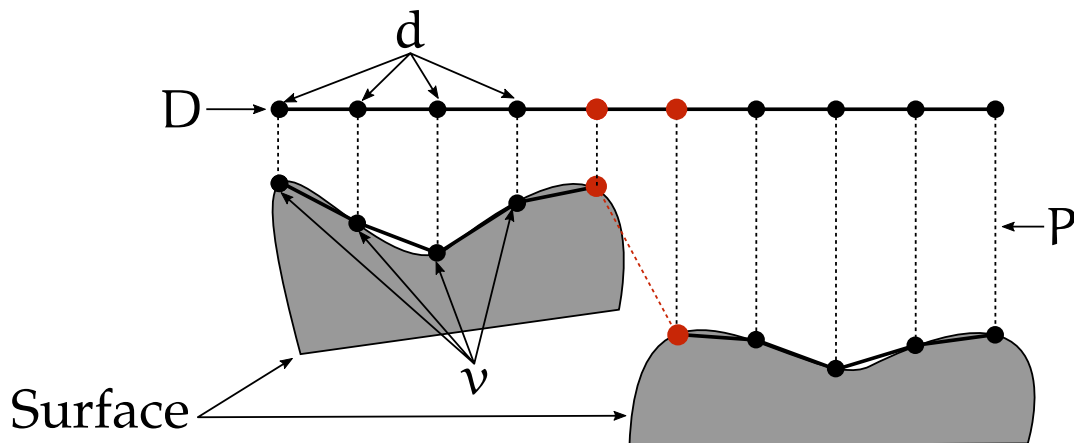


FIGURE 17 – Cas typique d'une connectivité dans le domaine de paramétrisation, fourni par la carte de profondeur, qui n'est pas fidèle à la topologie de la surface capturée : deux objets dans la scène capturée sont projetés sur des pixels voisins, mais l'arête rouge ne doit pas exister dans le graphe.

L'étape de projection fait passer du nuage de points (3D) à la carte de profondeur (2D). Cette réduction de dimension implique une perte d'information quant à la position d'un élément dans le domaine de projection. De ce fait, des régions numérisées peuvent se retrouver très proches dans la carte de profondeur bien que fortement éloignées sur la surface numérisée. Considérer donc telle quelle la connectivité de la carte de profondeur comme une bonne représentation de la connectivité de la surface acquise serait tout simplement faux (Figure 17).

Afin donc de définir une notion de proximité entre les différents points pour construire la connectivité du graphe, il est nécessaire de prendre en compte la variation d'intensité présente dans la profondeur en plus de sa connectivité. En effet, cette variation d'intensité représente la différence de distance entre plusieurs points d'une acquisition à la position du scanner au moment de la capture des données.

Nous décidons donc d'identifier les forts contours dans la carte de profondeur en utilisant la méthode des *gradients morphologiques* [RIVEST et al., 1993]. Cette ap-

proche se base sur les opérateurs de dilatation \oplus et d'érosion \ominus pour estimer les gradients dans une image :

$$g(D) = D \oplus b - D \ominus b,$$

où b est un élément structurant.

Le choix de la forme et de la taille de l'élément structurant fait varier la forme du voisinage considéré lors du calcul des gradients. La forme de celui-ci définit le voisinage à considérer, et sa taille fait varier la localité des variations à détecter.

Nous identifions ensuite les plus forts gradients à l'aide d'une méthode de seuillage adaptatif prenant en compte l'intensité des pixels (représentant, dans le cas des cartes de profondeur, la distance des points au scanner). La variation de la valeur du seuil a pour but de reproduire un seuillage sur des distances 3D, en interprétant la différence d'intensité comme une différence de distance. Cependant, une variation de profondeur donnée $\delta_d = |I_{p_1} - I_{p_2}|$ entre deux pixels p_1 et p_2 représentant des points proches du scanner, et la même variation de profondeur δ_d entre deux pixels p_3 et p_4 représentant des points éloignés du scanner ne correspond pas à la même variation de distance. Par conséquent, afin de pouvoir prendre en considération cette information, nous décidons de pondérer le seuil à l'aide d'une fonction inversement proportionnelle à la valeur d'intensité la plus élevée entre les deux pixels comparés.

Afin de supprimer les points proches des forts contours (pouvant être sujets à du bruit, car proches d'un angle rasant entre l'orientation de la surface et la direction du laser émis) et de ne pas connecter dans les traitements suivants des régions proches dans la carte de profondeur mais éloignées dans le nuage de points, nous décidons de supprimer les points - de marquer les pixels associés comme non-valides - qui ont été marqués dans l'image binaire comme ayant des forts gradients.

Maintenant, à partir de cette carte de profondeur filtrée, nous construisons un graphe $G(V, E)$ dont les sommets V sont les sommets valides et les arêtes E relient des pixels valides voisins. De manière empirique, nous avons trouvé que la 8-connectivité offrait un bon compromis entre l'approximation de différents calculs réalisés sur des graphes et le temps nécessaire à la réalisation de ces calculs. Il est évident que le choix de cette connectivité impose le bon choix de l'élément structurant ainsi que de sa taille lors de la détection des forts contours. De ce fait, pour aller de pair avec la 8-connectivité, nous avons décidé de considérer un élément structurant sous la forme d'une croix de taille 3 par 3.

La Figure 18 résume les différentes étapes introduites précédemment, permettant de passer d'une carte de profondeur à un graphe représentant la connectivité de la surface numérisée depuis ce point de vue.

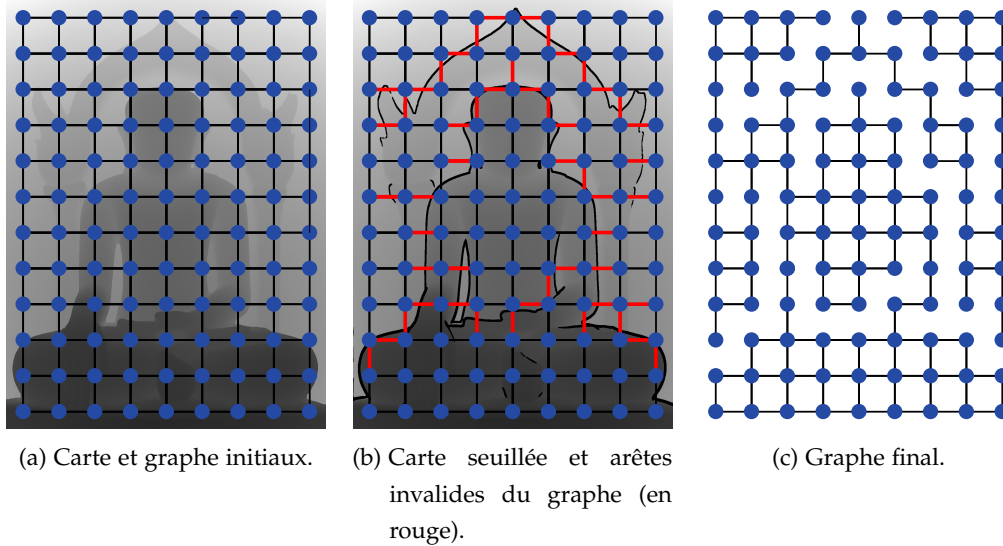


FIGURE 18 – Principales étapes pour construire un graphe à partir d'une carte de profondeur. Pour des raisons de visualisation, la résolution des graphes représentés est fortement réduite.

3.2 STRUCTURATION D'UN ENSEMBLE D'ACQUISITIONS : CONNEXIONS ENTRE GRAPHES

Etant donné qu'une campagne de numérisation peut être constituée de multiples acquisitions, plusieurs graphes sont construits en utilisant l'approche proposée précédemment (un graphe représentant une acquisition). Simplement, ces graphes ont été construits de manière indépendante et ceux-ci peuvent contenir des sommets représentant les mêmes morceaux de surface que d'autres graphes. En effet, il arrive régulièrement que la zone numérisée d'une acquisition chevauche la zone numérisée d'une autre acquisition.

Un parallèle intéressant peut être mis en avant, entre la paramétrisation de surfaces et la numérisation 3D d'une scène. Rappelons le principe de la *paramétrisation globale* d'une surface. Etant donné une surface S , considérons un ensemble de disques topologiques $\{C\}$ appelés *patches* et un ensemble de fonctions $\{\varphi\}$ qui mettent chaque patch C en correspondance avec un domaine 2D Ω (Figure 19). L'ensemble des fonctions $\{\varphi\}$ définit une variété différentielle si : Etant donné deux patches C_i et C_j , si leur intersection $C_i \cap C_j$ est homéomorphe à un disque, alors les images de leur intersection $C_i \cap C_j$ dans l'espace paramétrique au travers de φ_i et φ_j sont liés par une transformation géométrique différentiable $\tau_{\varphi_i \rightarrow \varphi_j}$:

$$\forall p \in C_i \cap C_j, \varphi_j(p) = \tau_{\varphi_i \rightarrow \varphi_j}(\varphi_i(p)).$$

Les fonctions $\tau_{\varphi_i \rightarrow \varphi_j}$ sont appelées des *fonctions de transition* [KHODAKOVSKY et al., 2003].

Comme expliqué dans le Chapitre 1, les points acquis sur la surface lors du processus de numérisation projettent chacun une image sur la carte de profondeur associée à cette acquisition.

On peut donc considérer que :

- La surface S correspond à la scène numérisée ;
- les patches C sont les acquisitions ;
- les fonctions φ sont les fonctions de projection P ;
- les domaines 2D Ω sont les cartes de profondeur D .

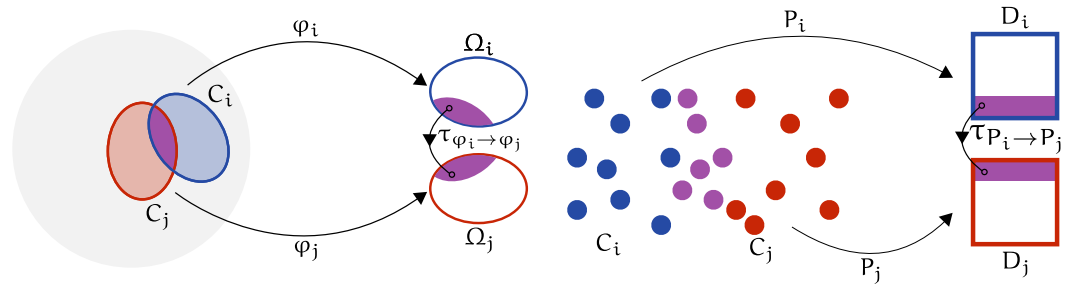


FIGURE 19 – Equivalence entre la paramétrisation d'une surface et le processus d'acquisition.

3.2.1 Gestion des correspondances entre graphes

Les graphes construits étant finis (dû à la nature "discrète" des cartes de profondeur), les sommets de différents graphes correspondants aux mêmes morceaux de surface n'ont aucune raison de correspondre à la même position géométrique. Il est donc nécessaire d'identifier pour chaque sommet d'un graphe, son plus proche voisin dans les autres graphes. Généralement, pour établir de telles correspondances entre des nuages de points, des arbres de partitionnement spatial sont construits, permettant ainsi de faciliter les requêtes de voisinage, et réduisant le coût de la recherche du plus proche voisin en $O(\log(N))$ où N représente le nombre de points.

Ce qui est intéressant avec ce lien entre paramétrisation de surfaces et numérisation 3D c'est cette notion de fonctions de transition. En effet, comme expliqué précédemment, les fonctions de transition $\tau_{\varphi_i \rightarrow \varphi_j}$ permettent d'obtenir, pour tous les éléments de l'intersection entre deux patches $C_i \cap C_j$, l'image de chacun dans les différents domaines 2D Ω_i et Ω_j . Dans notre cas elles permettent d'obtenir, pour tous les points de l'intersection entre deux acquisitions, le pixel associé à chacun de ces points dans les différentes cartes de profondeur D_i et D_j .

Etant donné qu'un graphe est construit à partir d'une carte de profondeur, et que chaque sommet de ce graphe est issu d'un pixel valide de la carte de profon-

deur, il est possible, en utilisant les fonctions de transitions définies entre les pixels des différentes cartes de profondeur, de définir des "transitions" entre des sommets de différents graphes. De cette manière, nous pouvons identifier les sommets correspondants de différents graphes, *i. e.*, identifier les points les plus proches entre différents nuages de points, en un nombre d'opérations indépendant du nombre de sommets dans ces graphes, et bien inférieur à celui-ci.

Pour chaque sommet $v \in V_i$ d'un graphe $G_i(V_i, E_i)$, l'ensemble L_v est donc identifié, contenant l'ensemble des sommets v'_j correspondant à l'image de v dans les autres graphes G_j (tel que $v'_j = \tau_{\varphi_i \rightarrow \varphi'_j}(\varphi_i(v))$).

Regardons ce qu'il se passe avec un exemple schématique simple. Nous avons deux cartes de profondeur D_1 et D_2 , ayant toutes deux échantillonnées une partie commune de la surface S (Figure 20(a)). Nous cherchons les sommets appartenant à la zone de recouvrement entre les graphes G_1 et G_2 , construits respectivement à partir de D_1 et D_2 (Figure 20(b) en violet). Pour cela, nous utilisons la fonction de transition $\tau_{\varphi_2 \rightarrow \varphi_1}$ sur l'ensemble des pixels de la carte D_2 pour trouver leur pixel correspondant dans D_1 . Considérons le cas du pixel d_1 de la carte de profondeur D_1 , à l'origine du sommet v_1 du graphe G_1 . L'ensemble des sommets l'ayant pour correspondant appartiennent au "cône de vue" correspondant à ce pixel (Figure 20(c)). En particulier, nous pouvons remarquer que les sommets v_2 et v'_2 du graphe G_2 appartiennent à ce "cône de vue", par conséquent v_1 est leur correspondant dans le graphe G_1 . De manière réciproque, nous utilisons la fonction de transition $\tau_{\varphi_1 \rightarrow \varphi_2}$ sur l'ensemble des pixels de la carte D_1 pour trouver leur pixel correspondant dans D_2 . Considérons cette fois-ci le cas du pixel d_2 de la carte D_2 , à l'origine du sommet v_2 . Dans ce cas, c'est le sommet v_1 qui va avoir le sommet v_2 comme correspondant dans le graphe G_2 (Figure 20(d)).

Il se trouve que considérer les fonctions de transition telles quelles, c'est-à-dire sans autre information, est insuffisant pour correctement identifier les sommets représentant les mêmes morceaux de surface. En effet, prenons un autre exemple schématique. Nous souhaitons cette fois-ci mettre en correspondance les sommets des graphes G_1 et G_3 . En particulier, regardons ce qu'il se passe pour le sommet v_3 issu du graphe G_3 (Figure 21(a)). En utilisant la fonction de transition $\tau_{\varphi_3 \rightarrow \varphi_1}$, le sommet v_3 semble avoir comme correspondant le sommet v_1 dans le graphe G_1 (Figure 21(b)). Cependant, nous pouvons nous rendre compte que cette association est incorrecte, car ces deux sommets ne correspondent pas au même morceau de surface. Nous remarquons donc que l'utilisation des fonctions de transitions seules ne permettent pas d'associer les sommets correspondants, car des faux-positifs seraient ainsi détectés. Pour pallier cela, nous proposons de comparer la distance entre les deux sommets v_1 et v_3 (que la fonction de transition a identifié comme potentiellement correspondants) à la distance entre v_1 et ses sommets voisins (Figure

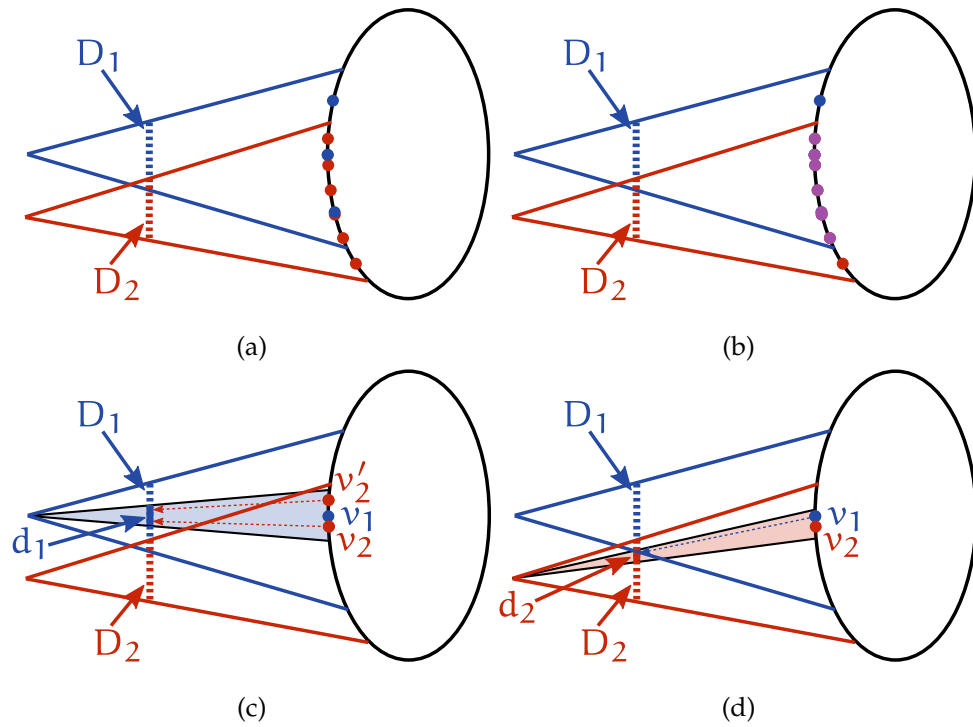


FIGURE 20 – Exemple schématique de la mise en correspondance de sommets de deux graphes représentant le même morceau de surface.

21(c)). Celle-ci n'est pas du même ordre de grandeur, par conséquent, le sommet v_1 n'est pas le sommet correspondant du sommet v_3 dans le graphe G_1 .

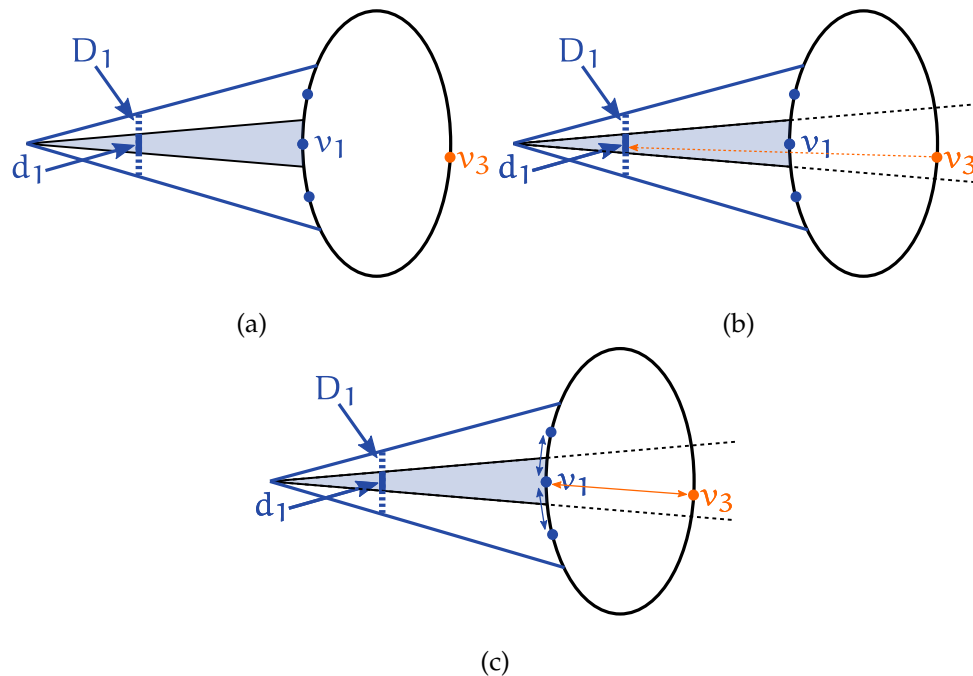


FIGURE 21 – Exemple schématique de la mise en correspondance de sommets de deux graphes représentant différents morceaux de surface.

De manière plus formelle, la combinaison des fonctions de transition avec la densité d'échantillonnage locale autour d'un sommet permet d'identifier les sommets appartenant réellement au même morceau de surface, et d'établir des correspondances entre ces sommets.

A partir de ces correspondances, nous pourrions construire un seul graphe global \hat{G} comme étant la fusion des différents graphes locaux afin de se replacer dans le cas du travail de [CHEN et al., 2018a] et d'appliquer des algorithmes de Traitement du Signal sur ce graphe. Simplement, comme expliqué auparavant, cette structure souffre de sa complexité en mémoire, la rendant inutilisable sur des données massives. Par conséquent, nous proposons de conserver de manière explicite, pour chaque sommet $v \in V_i$ l'ensemble de ses sommets correspondants $v' \in L_v$.

3.2.2 Suppression des sommets redondants

Etant donné que la précision de la description d'un morceau de surface dépend du nombre d'échantillons acquis dans cette zone (dépendant de la position et de l'orientation du système d'acquisition), il est trivial de constater que les différents graphes ayant échantillonné la même région de la scène ne la représentent pas avec le même degré de justesse. De ce fait, on ne peut pas espérer que des simulations appliquées à un graphe produisent les mêmes résultats dans les autres graphes représentant la même région. Par conséquent, il est nécessaire d'imposer explicitement cette cohérence au sein de l'ensemble des graphes.

Afin de pouvoir obtenir les simulations les plus précises au sein de différents graphes représentant la même région de surface, nous proposons d'identifier les sommets v les plus densément échantillonnés parmi leurs sommets correspondants dans L_v . Pour chaque sommet $v_i \in V_i$, nous calculons la distance moyenne entre ce sommet à tous les sommets $v_j \in \mathcal{N}_k(v_i)$, $\mathcal{N}_k(v_i)$ représentant les sommets à distance k de v_i , *i.e.*, les sommets séparés de v_i par k arêtes ou moins. La densité d'échantillonnage $w(v_i)$ peut être estimée comme l'inverse de cette distance :

$$\frac{1}{w(v_i)} = \frac{1}{\text{Card}(\mathcal{N}_k(v_i))} \sum_{v_j \in \mathcal{N}_k(v_i)} \|v_j - v_i\|^2 \quad (2)$$

Ensuite, pour chaque sommet $v_i \in V_i$ de chaque graphe G_i , nous comparons sa densité $w(v_i)$ avec la densité $w(v'_i)$ de ses sommets correspondants $v'_i \in L_{v_i}$. A la fin de ce processus, chaque ensemble de sommets V_i de chaque graphe G_i est décomposé en deux ensembles (Figure 22) :

- $V_i^+ \subseteq V_i$, l'ensemble des sommets ayant une densité plus élevée que l'ensemble de leurs sommets correspondants,
- $V_i^- = V_i \setminus V_i^+$, le reste des sommets.

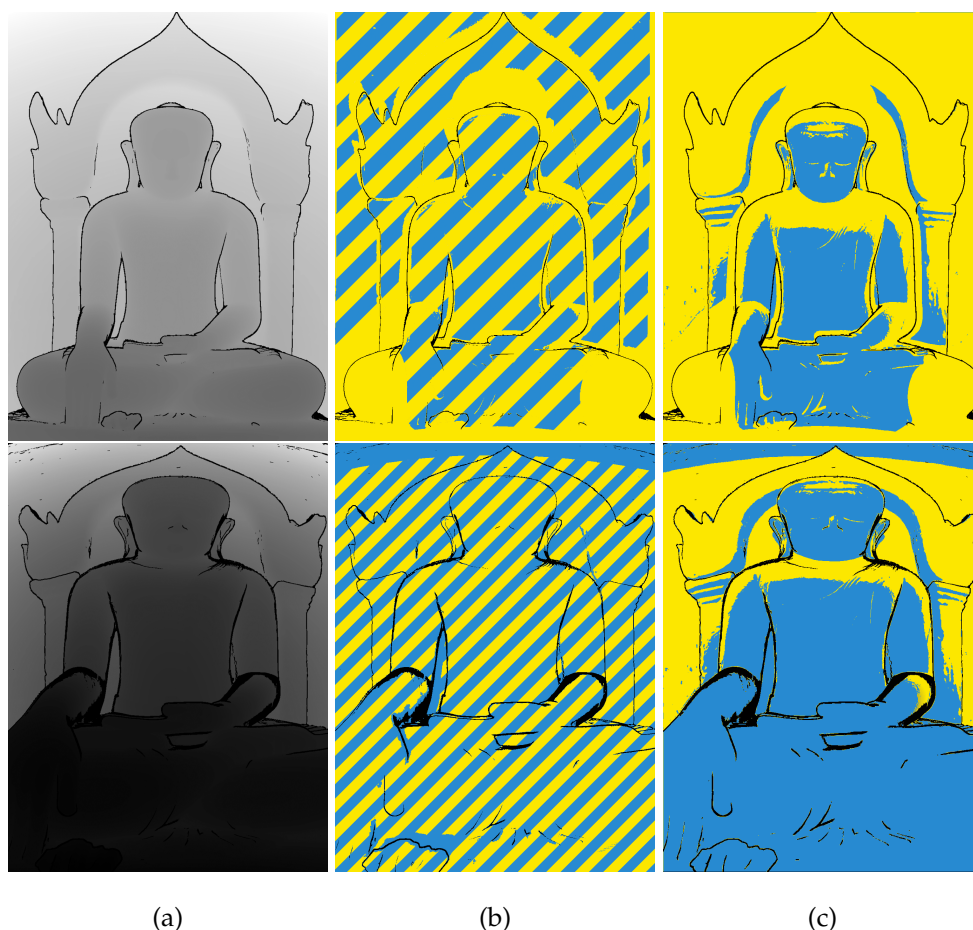


FIGURE 22 – Mise en correspondance de deux cartes de profondeur et gestion des superpositions (donnée *Eim Ya Kyaung*). (a) Deux cartes de profondeur ont été générées par un processus d’acquisition. (b) La couleur de chaque pixel représente l’appartenance d’un sommet à un graphe particulier (jaune en haut et bleu en bas). La zone hachurée correspond aux sommets ayant été capturés par les deux acquisitions. (c) En comparant les densités d’échantillonnage de chaque point, chaque graphe identifie les sommets avec une densité plus élevée que leurs sommets correspondants dans l’autre graphe. Les couleurs de la colonne de droite représentent, pour chaque sommet, le graphe dans lequel des simulations doivent être effectués pour cette région du nuage de points.

Finalement, les sommets V_i^+ seront utilisés pour faire des calculs, et les sommets V_i^- ne feront que récupérer le résultat de calculs depuis leurs sommets correspondants.

3.3 VALIDATION DE LA STRUCTURE

Nous validons maintenant l’utilisation de la structure proposée sur différents points. Dans un premier temps, nous attestons que cette structure permet d’estimer

des fonctions sur la surface décrite par un nuage de points. Nous analysons ensuite les performances en termes de temps de calcul et de consommation mémoire de la construction de la structure proposée. Finalement, nous présentons comment découper des acquisitions *a priori*, permettant d'un côté de traiter n'importe quelle acquisition, peu importe la résolution de celle-ci et le nombre de points qu'elle représente, et d'offrir ainsi un compromis entre temps de calcul et consommation mémoire.

3.3.1 Estimation de géodésiques

Nous validons maintenant l'utilisation de la structure proposée pour l'estimation de fonctions sur la surface décrite par un nuage de points. Pour cela nous décidons de comparer l'estimation de géodésiques dans un ensemble de graphes et sur une surface.

Pour cela, nous utilisons la méthode de calcul de géodésiques exact de [MITCHELL et al., 1987], implémentée dans l'outil [JACOBSON et PANOZZO, 2017], pour calculer ces géodésiques sur la surface d'un maillage triangulaire.

Pour calculer des géodésiques sur la surface de ce maillage en utilisant notre approche, nous avons réalisé des acquisitions synthétiques en suivant une approche similaire à [PIETRONI et al., 2011] (Figure 23). Afin d'avoir une bonne couverture des différents objets numérisés, nous avons placé des caméras tout autour des objets sur les sommets d'un icosaèdre, permettant ainsi de placer 12 caméras de manière uniforme sur une sphère dont le centre coïncide le centre de la boîte englobante de l'objet. Chaque caméra permet de générer une carte de profondeur à l'aide d'un rendu graphique d'un modèle 3D, en extrayant l'information provenant du tampon de profondeur. Chaque carte de profondeur générée est accompagnée d'une matrice de projection, permettant de passer de la carte de profondeur à un repère cartésien centré en la caméra, ainsi que d'une matrice de transformation, permettant de passer du repère cartésien local, centré en la caméra, en un repère cartésien global. Ces matrices permettent d'établir les fonctions de transition τ entre les différentes cartes de profondeur.

Afin de créer des nuages de points ayant les mêmes ordres de grandeur, nous avons normalisé l'aire de chaque modèle 3D de manière à ce que chaque modèle ait la même aire avant d'effectuer les acquisitions synthétiques.

A partir des cartes de profondeur synthétiques générées, un graphe est créé à partir chaque carte de profondeur et les sommets correspondants de chaque graphe sont liés entre eux. Nous calculons ensuite des géodésiques en utilisant un calcul de plus court chemin, en utilisant l'algorithme de Dijkstra [DIJKSTRA, 1959] entre un sommet du graphe et tous les autres sommets de celui-ci. Nous présen-

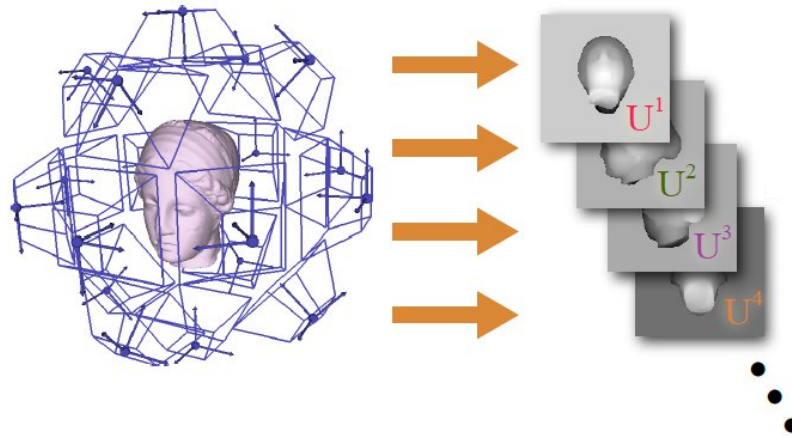


FIGURE 23 – Placement des caméras et acquisitions synthétiques, générant une carte de profondeur pour chaque point de vue. Image tirée de [PIETRONI et al., 2011].

tons tout d'abord comment ces géodésiques sont estimées dans le cas d'un seul graphe et ensuite dans le cas d'un ensemble de graphes connectés entre eux.

Calcul de géodésiques sur un graphe

Soit un graphe $G(V, E)$, l'algorithme de Dijkstra [DIJKSTRA, 1959] fournit une méthode permettant de connaître le plus court chemin d'un sommet $v_i \in V$ du graphe à tous les autres sommets de celui-ci. Les poids $W = \{w_1, w_2, \dots, w_m\}$ associés à l'ensemble des arêtes E , $m = \text{Card}(E)$ permettent de calculer une notion de distance entre les différents sommets, afin que le plus court chemin construit puisse prendre en considération une notion de "proximité" entre les sommets qui soit plus fine, et peut-être plus descriptive dans certains cas que le simple nombre d'arêtes séparant les sommets.

Dans notre cas, le plus court chemin entre deux sommets du graphe doit représenter le plus court chemin entre deux points sur la surface. Ainsi, le poids associé à chaque arête E_{ij} représente la norme L^2 dans l'espace 3D entre les deux sommets v_i et v_j de cette arête :

$$w_{ij} = \|v_i - v_j\|.$$

La prise en considération de cette distance permet, bien que le support soit le domaine paramétrique (la connectivité du graphe), de ne pas souffrir de la distorsion introduite par la paramétrisation, et de fournir une estimation de la distance géodésiques entre différents sommets.

Calcul de géodésiques sur un ensemble de graphes

Nous souhaitons que l'estimation de fonctions sur un graphe global \hat{G} obtenu comme une fusion des sommets correspondants d'un ensemble de k graphes locaux $G = \{G_1, G_2, G_3, \dots, G_k\}$ soit le même qu'avec notre approche par morceaux.

Maintenant, la question se pose concernant l'ordre dans lequel traiter les graphes G de façon à ce qu'un calcul de géodésiques soit similaire en étant effectué sur G et sur \hat{G} .

Rappelons que nous souhaitons connaître le plus court chemin du sommet v à tous les autres sommets. Pour commencer, il est évident que le premier graphe $G_i(V_i, E_i)$ à traiter est celui auquel le sommet appartient $v \in V_i$. Une fois le calcul du plus court chemin effectué entre le sommet v et l'ensemble des sommets V_i^+ , les sommets $v^+ \in V_i^+$ transmettent leur distance au sommet v à leurs sommets correspondants $L_{v^+}, \forall v^+ \in V_i^+$ dans les autres graphes $G_j \in G, j \neq i$. Chacun des autres graphes G_j se retrouve donc à effectuer les calculs de plus court chemin à partir des sommets V_j^- et à transmettre les distances calculées aux autres graphes.

Chaque graphe $G_j \in G, j \neq i$ continue ensuite d'appliquer l'algorithme de Dijkstra à partir des distances ayant été transmises à ses sommets V_j^- .

Maintenant que nous savons indiquer comment estimer des géodésiques dans chaque configuration, nous proposons de visualiser les géodésiques estimées entre l'approche basée surface et notre approche en partant du même sommet (Figure 24).

Nous pouvons constater que les géodésiques construites par notre approche sont très proches des géodésiques exactes calculées sur la surface. L'erreur d'approximation réalisée dans notre cas vient de l'utilisation de l'algorithme de Dijkstra, qui ne permet pas de construire des géodésiques aussi précises que l'approche surfacique de [MITCHELL et al., 1987]. Néanmoins, dans la suite de cette thèse cette erreur n'aura que peu d'influence, étant donné que les géodésiques ne seront construites que sur des petites régions autour de chaque sommet à l'origine de la géodésique, et non pas sur toute la surface décrite par l'ensemble des acquisitions.

3.3.2 Performances en temps de calcul et consommation mémoire

Nous analysons maintenant notre méthode de structuration de nuages de points en termes de performances en temps de calcul et consommation mémoire.

Les différentes campagnes d'acquisitions considérées demandent plus de mémoire vive que la machine utilisée ne peut en fournir. Pour obtenir une estimation du coût mémoire total que nécessiterait la représentation de toutes les acquisitions d'une même campagne, nous avons donc enregistré sur un disque dur l'ensemble des structures nécessaires pour chaque acquisition, sous forme binaire. La somme des poids des différents fichiers enregistrés nous permet donc d'avoir une esti-

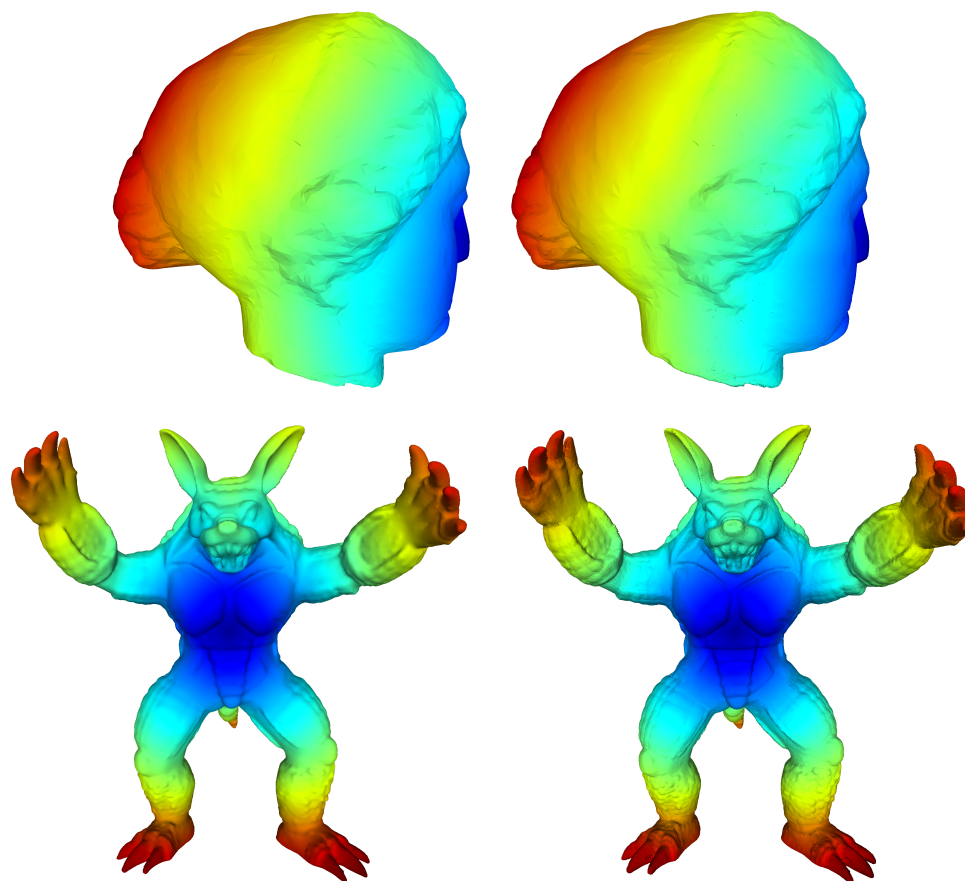


FIGURE 24 – Comparaison entre un calcul de géodésique réalisé sur la surface d’un maillage surfacique (à gauche) et sur un ensemble de graphes locaux (à droite) pour différents modèles (*Egea* en haut et *Armadillo* en bas). Les distances à partir d’un même point sont représentées par des couleurs allant de bleu (faible distance) à rouge (forte distance).

mation du coût mémoire que l’ensemble des acquisitions représenteraient si nous ne limitons pas la quantité de mémoire utilisée. Si cette estimation n’est pas la réelle consommation mémoire, celle-ci nous donne une borne inférieure du coût mémoire, et permet donc de réaliser une comparaison avec le pic de consommation mémoire réellement atteint.

Premièrement, la Table 1 présente quelques résultats sur différentes campagnes d’acquisition sur lesquelles l’algorithme proposé a été testé. Pour chaque acquisition, nous présentons le **temps** nécessaire pour structurer la donnée, le **pic mémoire** atteint lors de l’exécution, ainsi que le pourcentage de réduction par rapport au coût mémoire total estimé, calculé de la manière expliquée précédemment.

Nous invitons le lecteur à se référer à l’Annexe A pour obtenir les caractéristiques de la machine sur laquelle les expérimentations ont été effectuées ainsi que des informations techniques concernant les différentes données de test utilisées.

Nom	Temps (h:m:s)	Pic mémoire (Gio)
<i>Templo Mayor</i>	00:07:23	3,5
		-92,4%
<i>Facade</i>	00:15:22	5,7
		-92,1%
<i>Eim Ya Kyaung</i>	01:12:45	2,0
		-98,3%
<i>Meeting House</i>	00:56:28	5,7
		-95,9%
<i>Villa Arianna</i>	01:52:19	2,6
		-98,3%
<i>Ananda Oak Kyaung</i>	02:17:57	1,9
		-98,8%
<i>Intérieur</i>	00:43:07	5,7
		-95,8%
<i>Wat Phra Si Sanphet</i>	10:36:15	5,7
		-99,1%

TABLE 1 – Temps de calcul et pic mémoire atteints pour la structuration de différentes campagnes d’acquisition. En-dessous du pic mémoire est indiqué le pourcentage par rapport au coût mémoire total si l’ensemble des acquisitions étaient effectivement en mémoire.

Nous pouvons tout d’abord remarquer la considérable réduction du coût mémoire. Cette réduction est d’autant plus marquée que le nombre d’acquisitions est important. Ceci est dû à notre manière de ne conserver en mémoire que les informations nécessaires à chaque traitement de la structuration. De ce fait, le pic de consommation mémoire réellement atteint peut être réduit au coût maximum nécessaire pour une opération particulière sur une acquisition. En l’occurrence dans le cas de la structuration, ce coût maximum est atteint au moment de la gestion des correspondances, étant donné qu’à ce moment là il est nécessaire d’avoir deux acquisitions en mémoire.

La majeure partie (environ 95% du temps) de l’algorithme est passé à établir les correspondances entre les différents graphes. En effet, il s’agit d’une étape assez coûteuse, car elle est réalisée avec une complexité de $O(n * k)$ où n représente le nombre total de sommets et k le nombre de graphes, étant donné que pour chaque sommet, son sommet correspondant (s’il existe) dans chaque graphe doit

être trouvé, grâce aux fonctions de transition. Ceci explique pourquoi des campagnes d'acquisition, contenant approximativement le même nombre de points, comme par exemple les données *Meeting House* (50 acquisitions) et *Waikiki Natatorium* (111 acquisitions), peuvent avoir des temps de calcul passant du simple au double.

De plus, la lecture/écriture des données depuis/vers le disque dur a un certain impact sur le temps de calcul. Afin de pouvoir estimer l'impact de ce processus de chargement/déchargement de données, nous avons décidé de réaliser la structuration sur des acquisitions synthétiques, et d'analyser l'impact de notre méthode de gestion de la mémoire sur le temps de calcul, en fonction du nombre d'acquisitions considérées. Afin de faire varier le nombre d'acquisitions, étant donné que les caméras synthétiques sont placées sur les sommets d'un icosaèdre, nous subdivisons l'icosaèdre afin de placer plus de caméras sur les sommets polyèdres obtenus après différents niveaux de subdivision.

La Table 2 présente l'ensemble des différents temps de calcul et pics de consommation mémoire atteints entre une exécution avec et sans notre gestion de la mémoire. Ce tableau met bien en avant l'impact de la gestion de la mémoire telle que nous l'avons proposé. D'un côté, la différence de temps de calcul est d'autant plus marquée que le nombre d'acquisitions à traiter est important. En effet, plus il y a d'acquisitions, plus le nombre de lecture/écriture depuis/vers le disque dur est important. À l'inverse, plus le nombre d'acquisitions à traiter est important, plus la réduction de la consommation mémoire est conséquente, et plus l'approche proposée prend de son sens.

3.3.3 Découpage d'une acquisition massive

Une seule acquisition peut capturer plusieurs centaines de millions de points. Dans ce cas, le graphe associé peut ne pas tenir en mémoire. Afin de corriger ce problème, pour être capable de traiter tous les acquisitions peu importe le nombre de points qu'elles représentent, nous proposons de découper *a priori* une carte de profondeur en des *tuiles avec chevauchements* (Figure 25).

Ce qui est intéressant, c'est que ce découpage permet de considérer un ensemble de tuiles comme un ensemble d'acquisitions : chaque tuile représente un graphe et la fonction de transition de chaque tuile n'est qu'une translation de la fonction de transition de la carte de profondeur.

Dans le cas particulier où deux tuiles proviennent de la même carte de profondeur, nous remplaçons la comparaison des densités d'échantillonnage (puisque les graphes, issus de ces deux tuiles, représentent exactement les mêmes sommets

Nom	Acq.	Temps (s)		Pic mémoire (Gio)	
		Sans	Avec	Sans	Avec
<i>Armadillo</i>	12	2,1	3,5 / +66,7%	1,0	0,5 / -50,0%
	42	14,0	37,7 / +169,3%	2,8	0,5 / -82,1%
	162	193,5	573,6 / +196,4%	16,3	0,5 / -96,9%
<i>Egea</i>	12	2,2	3,6 / +63,6%	1,0	0,5 / -50,0%
	42	14,2	44,8 / +215,5%	3,2	0,5 / -84,4%
	162	193,2	607,8 / +214,6%	22,0	0,5 / -97,7%
<i>Neptune</i>	12	2,2	3,6 / +63,6%	1,0	0,5 / -50,0%
	42	13,8	26,7 / +93,5%	2,6	0,5 / -80,8%
	162	184,7	432,9 / +134,4%	13,3	0,5 / -96,2%
<i>Venus</i>	12	2,2	3,6 / +63,6%	1,0	0,5 / -50,0%
	42	14,3	55,8 / +290,2%	3,1	0,5 / -83,9%
	162	196,7	439,1 / +123,3%	21,4	0,5 / -97,7%

TABLE 2 – Temps de calcul et pic mémoire atteints pour la structuration de différentes données synthétique avec et sans gestion de la mémoire. Chaque acquisition a été générée une résolution de 1024 par 1024 pixels.

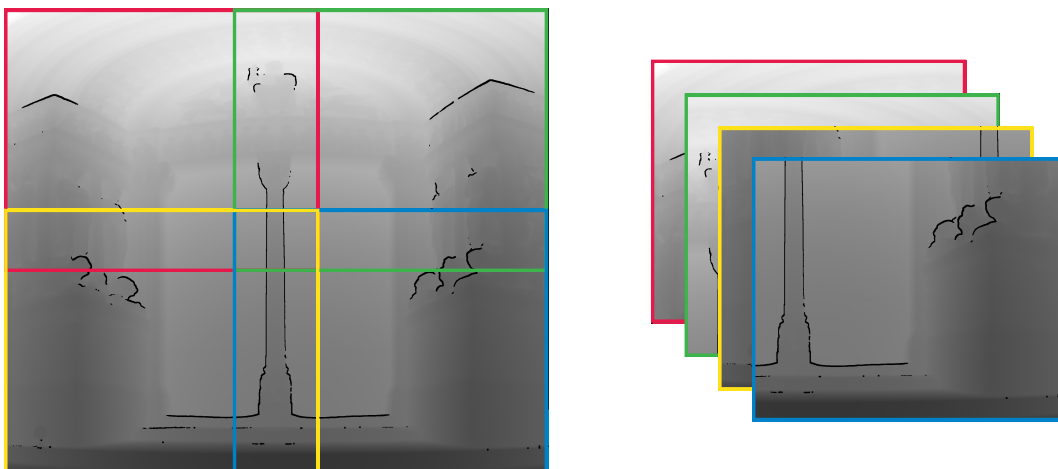


FIGURE 25 – Découpage d'une carte de profondeur en un ensemble de tuiles avec chevauchements. Le bord de chaque tuile a été identifié avec une couleur différente.

dans la zone de chevauchement), par une comparaison des distances de ces deux sommets par rapport au bord de chacun de ces deux graphes (Figure 26).

Ainsi, chaque tuile peut identifier l'ensemble des sommets V_i^+ et V_i^- , de manière similaire aux différentes cartes de profondeur d'une campagne d'acquisition.

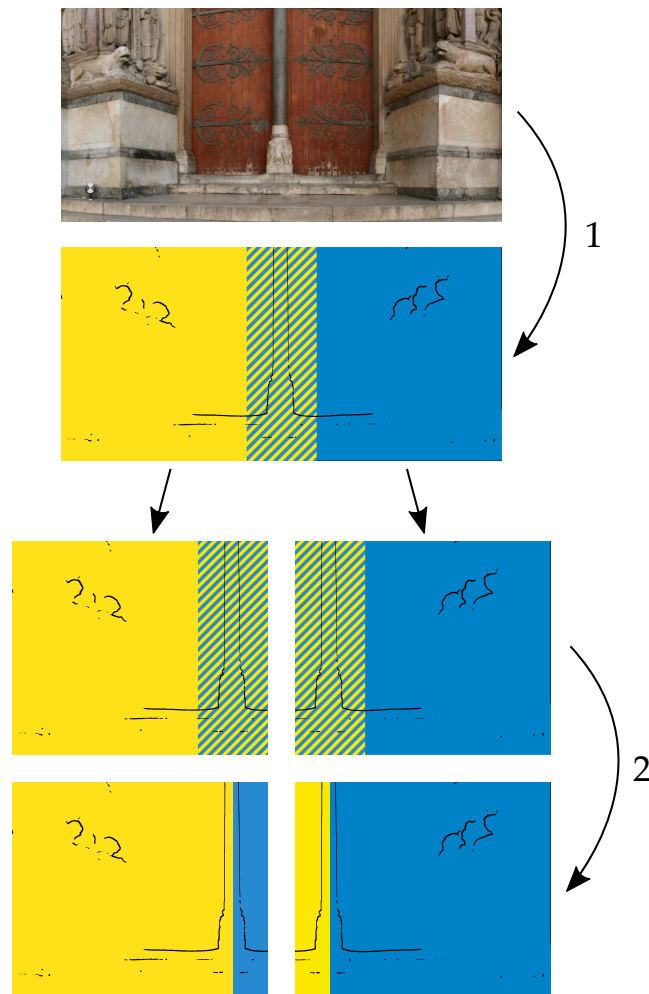


FIGURE 26 – Gestion de la superposition pour des tuiles issues de la même carte de profondeur. En 1, l’image de départ est découpée en deux tuiles bleu et jaune avec une zone de chevauchement (zone hachurée). En 2, les superposition sont gérées en identifiant pour chaque graphe l’ensemble des sommets étant plus éloignés du bord commun entre les deux graphes.

Ce qui est intéressant avec ce découpage c’est la manière dont il permet de borner l’utilisation mémoire. Etant donné que chaque tuile représente le même nombre de pixels, la taille maximum de tuile fixe directement le pic de consommation mémoire.

La Table 3 présente l’évolution du temps de calcul et du pic de consommation mémoire de différentes campagnes d’acquisition pour différentes tailles de tuiles.

Nous pouvons constater que le découpage sous forme de tuiles fournit un contrôle direct sur le pic de consommation mémoire atteint lors de la structuration du nuage de points. Le découpage offre donc un compromis entre temps de calcul et consommation mémoire, de façon à pouvoir structurer n’importe quel nuage de points sur n’importe quelle machine.

Site	Temps (h:m:s) / Pic mémoire (Gio) (Tuiles)		
	8192 ²	4096 ²	2048 ²
<i>Templo Mayor</i>	00:09:25 / 1,8 (33)	00:11:48 / 1,0 (55)	00:25:57 / 0,6 (213)
	+27,5% / -96,1%	+59,8% / -97,8%	+251,4% / -98,7%
<i>Facade</i>	00:25:14 / 5,7 (54)	01:00:36 / 1,0 (162)	03:04:53 / 0,6 (490)
	+64,2% / -90,2%	+140,2% / -98,3%	+632,8% / -99,0%
<i>Meeting House</i>	01:30:18 / 2,9 (101)	03:56:03 / 1,1 (278)	12:32:46 / 0,6 (933)
	+59,9% / -97,9%	+318,0% / -99,2%	+1233,1% / -99,6%
<i>Intérieur</i>	01:04:51 / 5,7 (78)	02:26:14 / 1,1 (234)	04:30:21 / 0,6 (704)
	+50,4% / -95,2%	+125,5% / -99,1%	+527,0% / -99,5%

TABLE 3 – Temps de calcul et pic mémoire atteints par notre algorithme de structuration en fonction de la taille des tuiles. A côté des temps de calcul, le nombre total de tuiles issues du découpage est indiqué entre parenthèses. Le pourcentage de temps de calcul indique le coût rapport aux performances atteintes avec les acquisitions sans découpage. Le pourcentage de pic mémoire indique le pic atteint par rapport au traitement des acquisitions sans découpage.

Pour attester qu’il n’y a pas de différences entre l’estimation de fonctions entre une acquisition massive et cette même acquisition découpée en un ensemble de tuiles avec chevauchements, nous réalisons des calculs de géodésiques, de manière similaire à l’approche présentée précédemment, et nous comparons les géodésiques estimées (Figure 27).

Nous pouvons constater que les géodésiques obtenues sont exactement les mêmes, que l’estimation soit faite sur un ensemble de graphes ou sur un seul (grâce à la gestion de l’ordre de traitement des graphes). Ce résultat est assez cohérent, puisqu’en ayant découpé une acquisition sous la forme de tuiles, les sommets traités sont les mêmes que ceux de l’acquisition d’origine. De ce fait il n’y a pas de différence dans les calculs de plus court chemin. Cela nous confirme que les calculs réalisés sur un ensemble de tuiles fournissent bien les mêmes résultats que sur un graphe global, mais pour un pic de consommation mémoire fortement réduit et directement contrôlable par la taille des tuiles.



FIGURE 27 – Différence entre le calcul de géodésiques à partir d’une seule acquisition (à gauche) et sur un ensemble de tuiles (à droite) de la donnée *St Trophime*. Les distances à partir d’un même point sont représentées par des couleurs allant de bleu (faible distance) à rouge (forte distance). Au milieu, les sommets V_i^+ des différents graphes $\hat{G}_i \in \hat{G}$ issus de chaque tuile sont identifiés avec une couleur différente pour chaque graphe.

3.4 CONCLUSION DU CHAPITRE

3.4.1 Contributions

Nous avons présenté dans ce chapitre une nouvelle structure pour la manipulation de la surface sous-jacente à un nuage de points massifs.

En s’appuyant sur la connectivité des acquisitions, des **graphes locaux** ont été construits représentant chacun le comportement local de la surface numérisée. En combinant les différents graphes construits à l’aide de fonctions de transition, permettant de passer de sommets d’un graphe aux sommets d’un autre graphe représentant la même région, nous avons proposé d’identifier dans chaque graphe l’ensemble des sommets qui effectuent des calculs et l’ensemble des sommets qui récupèrent l’information de leurs sommets correspondants dans les autres graphes afin d’éviter d’effectuer des calculs redondants lors d’opérations appliquées au sein d’un nuage de points (comme le calcul de géodésiques par exemple).

La segmentation naturelle fournie par l'ensemble des acquisitions a permis de **limiter la quantité de mémoire requise** par les différentes opérations considérées. Ainsi il est possible de traiter des nuages de points composés de plusieurs centaines d'acquisitions et représentant plusieurs milliards de points, sur des ordinateurs ayant une quantité de mémoire vive restreinte.

Nous avons proposé une stratégie de **découpage d'acquisitions** sous la forme de tuiles avec chevauchements permettant de réduire encore plus la consommation en fixant le pic de consommation mémoire réellement atteint. De plus, peu importe la taille de chaque acquisition et leur nombre, la méthode proposée est capable de découper la donnée en un ensemble de tuiles avec chevauchements, permettant de fixer une limite sur le pic de consommation mémoire atteint lors des différentes opérations effectuées.

Nous avons mis en avant les capacités de **passage à l'échelle** de la structure proposée, en appliquant les notions introduites à la structuration de campagnes d'acquisition réelles composées de plusieurs milliards de points. Nous avons montré que malgré la taille conséquente de ces données, il était possible de les structurer avec un pic de consommation mémoire extrêmement faible.

3.4.2 Discussion

La méthode de seuillage adaptatif, bien qu'assez simpliste, permet d'obtenir une opération ayant une complexité en temps de calcul très faible et le filtrage réalisé permet de segmenter dans la carte de profondeur les régions de la surface n'étant pas directement connectées entre elles. Nous avons toutefois pu noter que dans des régions fortement éloignées du scanner au moment d'une acquisition, notre méthode avait tendance à sur-détecter des contours. Dans l'ensemble de nos expérimentations cela n'a pas vraiment posé problème, car dans ces régions, la densité d'échantillonnage de cette acquisition est très faible. Par conséquent il y a de fortes chances pour qu'une autre acquisition aie capturé la même région. Cependant il serait peut-être intéressant de trouver d'autres solutions pour la segmentation des acquisitions, en ayant toujours à l'esprit que la méthode choisie doit avoir une faible complexité, pour ne pas trop impacter les temps de calcul.

Dans les zones les moins densément échantillonnées de chaque acquisition, l'hypothèse faite sur la différence de profondeur pour lier les sommets d'un graphe peut ne plus être valide. Dans de telles zones, deux problèmes peuvent se produire :

- quand le scanner est trop loin d'une région, la distance entre deux échantillons, voisins dans la carte de profondeur, peut être plus importante que la taille d'un trou entre deux différentes régions. Cela résulte donc en une sous-détection de contours,
- quand le laser touche des régions avec des angles obtus, la profondeur entre deux échantillons peut fortement varier, alors que ceux-ci appartiennent à la même région. Cela résulte donc en une sur-détection de contours.

Cependant, de telles particularités ne posent pas de problème en pratique, étant donné que les régions sous-échantillonnées d'une acquisition sont généralement couvertes par d'autres acquisitions ayant de plus fortes densités d'échantillonnage dans ces régions.

Dans les prochaines parties, nous présentons différentes applications utilisant la structure proposée pour la généralisation d'algorithmes de traitement de la géométrie à la manipulation de la surface sous-jacente à des nuages de points massifs.

Troisième partie

ECHANTILLONNAGE DE NUAGES DE POINTS

INTRODUCTION

Dans ce chapitre, nous présentons tout d'abord quelques caractéristiques liées aux acquisitions 3D dans le contexte des LiDARs terrestres. Nous présentons ensuite des méthodes de l'état de l'art du ré-échantillonnage de nuages de points. Par la suite, nous introduisons un type d'échantillonnage particulier, l'échantillonnage en disques de Poisson qui va servir de base à notre méthode. Enfin nous présentons les objectifs que nous nous fixons pour l'obtention de versions simplifiées de nuages de points massifs.

4.1 CARACTÉRISTIQUES DES ACQUISITIONS 3D

De manière générale, on peut décomposer le processus de numérisation 3D en deux étapes principales : *l'acquisition* et *l'aggrégation*.

La première étape consiste en la capture d'échantillons dans l'ensemble de la zone à numériser. Pour cela, les systèmes d'acquisition sont déplacés à différentes positions, de manière à acquérir de l'information depuis différents points de vue. Ces différentes données, acquises de manière indépendante, sont traitées et fusionnées lors de la deuxième étape. Dans le contexte de données acquises à partir de LiDARs terrestres, la donnée acquise représente des points 3D définis dans un repère local. Ainsi, le processus d'aggrégation consiste à recalculer les différentes acquisitions dans un unique repère global, pour former un seul et unique nuage de points.

Le processus de numérisation 3D introduit trois raisons principales amenant à justifier le ré-échantillonnage de nuages de points.

Anisotropie de l'échantillonnage d'une acquisition. En effet, comme présenté dans le Chapitre 1, la manière dont les LiDARs terrestres acquièrent des points se fait sous la forme d'un balayage horizontal et vertical. De ce fait, les points acquis suivent des directions privilégiées (Figure 28) rendant la distribution initiale de points fortement anisotrope dans chaque acquisition. Le problème est que ces directions privilégiées ne sont pas uniquement corrélées au contenu de la donnée, mais aussi à la manière dont les échantillons ont été acquis, liée au balayage du laser (de manière verticale et horizontale) sur la surface numérisée lors de l'acquisition.

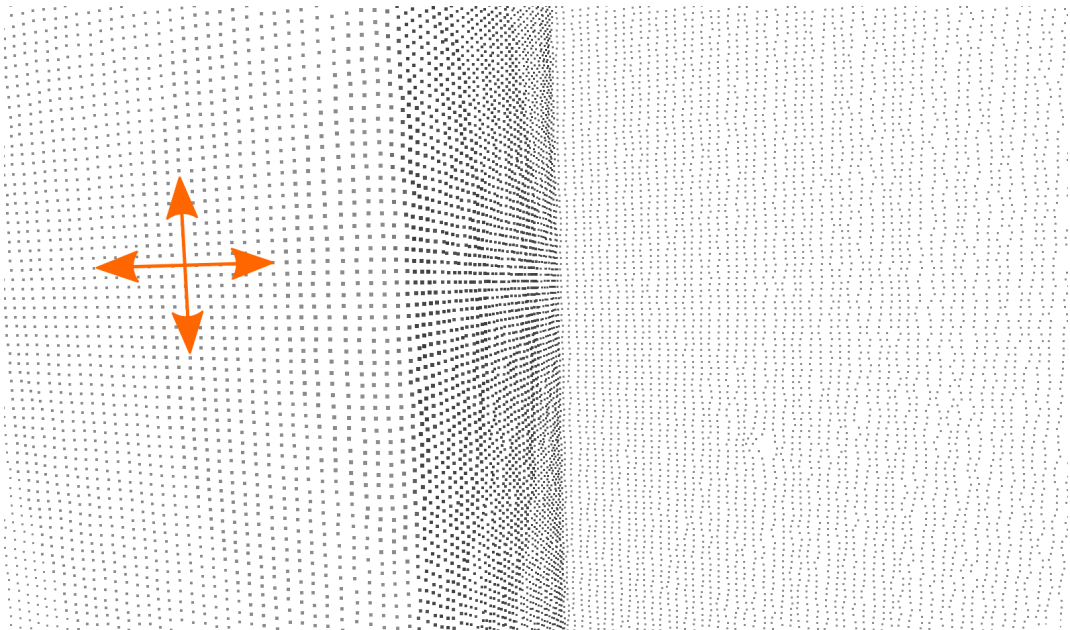


FIGURE 28 – Exemple de la forte anisotropie présente dans une acquisition de la donnée *Facade*. La distribution des points suit des directions privilégiées, liées en partie au balayage du laser.

Nombre de points total. Comme expliqué dans la Partie II, la rapidité d’acquisition des scanners actuels, ainsi que les régions couvertes par certaines campagnes d’acquisition amènent aujourd’hui les nuages de points à être extrêmement volumineux. Dans le cadre de la conservation du patrimoine historique par exemple, les campagnes d’acquisition peuvent ne plus se contenter de numériser une statue ou un monument, mais vont jusqu’à acquérir des données détaillées sur tout un site historique. Par conséquent, leur cardinalité (supérieure au milliard de points) rend difficile, voire impossible, les traitements classiques appliqués aux nuages de points. Même la simple visualisation directe de ces données est inenvisageable telle quelle sur des ordinateurs classiques.

Qualité de l’échantillonnage initial. La position des échantillons dans le nuage de points résultant de l’agrégation de différentes acquisitions est fortement disparate. Lors de chaque acquisition, certaines régions sont fortement sur-échantillonnées, d’autres sous-échantillonnées. Certaines régions de la surface acquise ne sont représentées que par une acquisition, et d’autres par plusieurs acquisitions. Par conséquent la distribution initiale peut contenir de très fortes variations de densité d’échantillonnage, complètement décorréliées de l’information présente dans la surface numérisée (Figure 29).

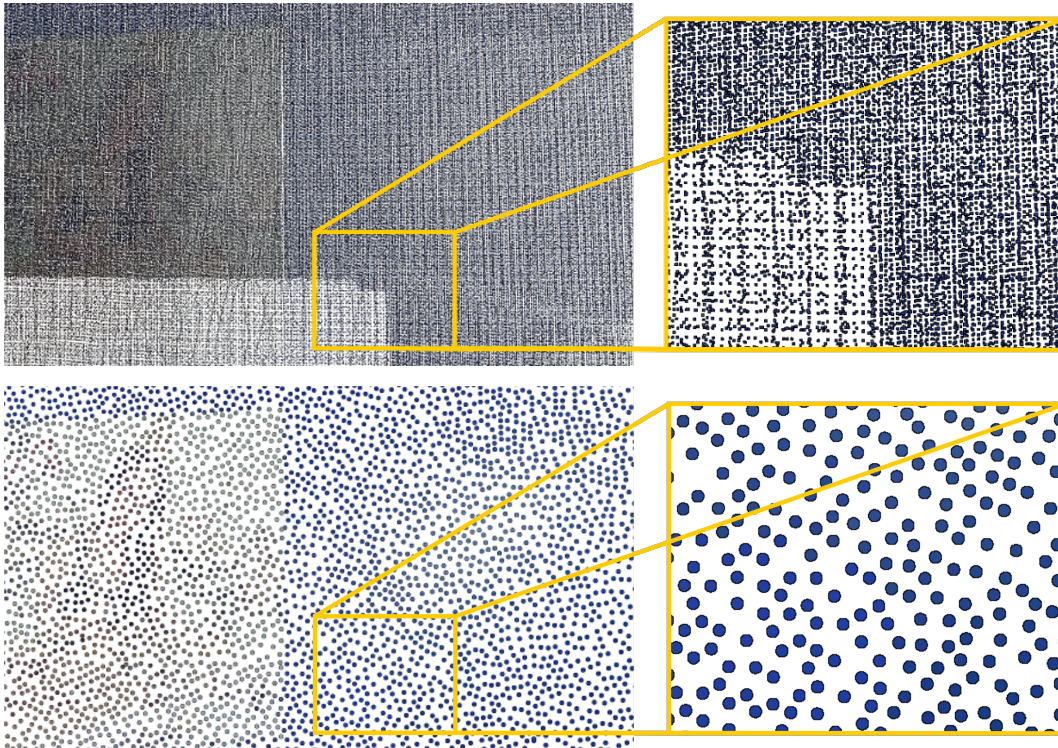


FIGURE 29 – En haut, exemple des importantes variations de densité pouvant être présentes dans un nuage de points issus de l’agrégation de plusieurs acquisitions du modèle *Techshop*. En bas, la méthode proposée permet entre autres, d’uniformiser la densité de la distribution finale.

4.2 ETAT DE L'ART DU RÉ-ÉCHANTILLONNAGE DE NUAGES DE POINTS

Le ré-échantillonnage de nuages de points est un domaine étudié depuis presque deux décennies maintenant. Il s’agit d’une étape essentielle sur laquelle reposent beaucoup d’algorithmes de traitement de nuages de points. En effet, la complexité de nombreuses méthodes (à la fois en temps et en mémoire) rend difficile, voire souvent impossible, le traitement de nuages de points massifs. Cependant, si la réduction du nombre de points peut permettre à des algorithmes d’être utilisés quand cela n’était pas possible auparavant, la sélection de ces points et de leur position lors du ré-échantillonnage est bien souvent essentielle. En effet, cette étape peut permettre d’extraire des informations pertinentes de la donnée numérisée en fonction des nécessités des méthodes appliquées à ces données [CHEN et al., 2018a].

Nous pouvons séparer le ré-échantillonnage de nuages de points en deux familles principales : celles s’intéressant à la réduction de la quantité de points représentant la surface sous-jacente d’un nuage de points et celles s’intéressant à l’amélioration de la qualité de l’échantillonnage d’un nuage de points.

4.2.1 *Simplification de nuages de points*

De manière générale, le but de la simplification de nuages de points est simple : trouver un nuage de points \hat{P} ayant une densité cible qui minimise la distance entre la surface représentée par \hat{P} et celle représentée par le nuage de points original P . Cependant, comme expliqué par [CHEN et al., 2018b], mesurer la distance entre deux nuages de points n'est pas quelque chose d'évident à réaliser.

Le premier type d'approche consiste à réduire le nombre de points via des méthodes de *simplification de maillages*. A partir d'un nuage de points initial, une surface polygonale finement détaillée est construite puis simplifiée en réduisant le nombre de sommets par des opérations de *contraction d'arêtes* (Figure 30).

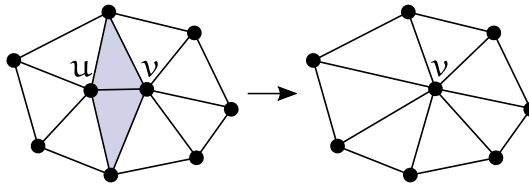


FIGURE 30 – Exemple de contraction d'arêtes. L'arête uv est contractée de manière à ce que le sommet u soit confondu avec le sommet v . Toutes les arêtes incidentes au sommet u sont maintenant incidentes au sommet v .

Lors de la contraction d'une arête, la position du sommet résultant de cette contraction peut être soit celle d'un des deux sommets qui composaient l'arête, ou résulter de l'interpolation des positions des points dans le voisinage de cette arête. Dans le second cas, cela signifie que la position des points après l'étape de simplification ne correspond plus à la position des points d'origine, ce qui peut être un inconvénient selon l'application subséquente et la confiance dans la position des données acquises. L'auteur est invité à se référer à [HECKBERT et GARLAND, 1997; LUEBKE, 2001; MAGLO et al., 2015; PENG et KUO, 2005] pour un état de l'art plus complet sur la simplification de maillages.

Cette approche pré-suppose qu'il est possible de construire un maillage à partir du nuage de points d'origine. Or, comme expliqué dans la Partie II, la reconstruction de surfaces est un problème qui n'est pas convenablement résolu concernant le traitement de données massives.

En s'inspirant de ces travaux dans le domaine de la simplification de surfaces, les auteurs de [PAULY et al., 2002] ont notamment montré que ces méthodes pouvaient être étendues au domaine de la simplification de nuages de points. Leur idée a été de modéliser des propriétés locales de la surface passant par un nuage de points. En réalisant une analyse de covariance dans le voisinage d'un point, les auteurs ont pu obtenir une information de la *variation locale de la surface* en différents points du nuage de points. De cette manière, ils ont pu définir plusieurs

opérateurs, similaires aux méthodes surfaciques, pour la décimation de nuages de points. Notamment, ils ont considéré une approche similaire à la contraction d'arête, appelée *contraction de paires de points* (*point-pair contraction* en anglais) en se basant sur une approche similaire à [GARLAND et HECKBERT, 1997], estimant localement l'impact de la contraction d'une paire de points, afin de supprimer de manière itérative celles ayant le plus faible impact sur l'apparence de la surface sous-jacente au nuage de points, jusqu'à avoir réduit le nombre de points du nuage de points d'origine au nombre désiré.

D'autres méthodes ont ensuite vu le jour, en se basant aussi sur une information de variation locale de la surface. Nous pouvons notamment citer le travail de [SONG et FENG, 2008], approximant des formes locales à l'aide de plans tangents, leur permettant de définir une fonctionnelle à minimiser en partitionnant la donnée de manière itérative et en sélectionnant un représentant par partition créée. [MIAO et al., 2009] ont proposé l'utilisation d'une méthode se basant sur une version adaptative de l'algorithme de partitionnement du "*mean-shift*", permettant de générer des échantillonnages adaptatifs, prenant en compte l'information géométrique locale à l'aide de la courbure de la surface. A l'aide d'un arbre de regroupement hiérarchique (*hierarchical cluster tree* en anglais), [YU et al., 2010] ont amélioré l'efficacité et la qualité des nuages de points générés. [SHI et al., 2011] ont quant à eux proposé l'utilisation d'un algorithme de *k-moyennes* (*k-means* en anglais) pour partitionner le nuage de points.

De manière générale, la simplification de nuages de points s'intéresse à préserver la forme globale de la surface représentée par le nuage de points d'origine plutôt que d'obtenir une bonne distribution d'échantillons dans le nuage de points simplifié. Le but étant principalement de réduire la quantité de points que les algorithmes doivent traiter et de préserver la surface approximée. Il s'agit d'un problème d'*échantillonnage optimal* pour des surfaces plongées dans un espace 3D.

4.2.2 Amélioration de l'échantillonnage

Afin de résoudre les problèmes liés à la qualité de l'échantillonnage (bruit d'acquisition, fortes variations de densité, etc.) pouvant rendre les méthodes de reconstruction de surface difficiles à utiliser, l'idée est de réaliser une étape de *consolidation* du nuage de points. L'idée générale est de ré-échantillonner la donnée d'origine afin de réduire le bruit, ainsi que les problèmes de redondance d'information et de sous-échantillonnage. En utilisant une approche basée sur les *Moving Least Squares* (MLS) [SHEPARD, 1968], les auteurs de [ALEXA et al., 2003] ont utilisé l'opérateur de projection pour ré-échantillonner la surface de manière locale, permettant ainsi d'uniformiser l'échantillonnage de la surface sous-jacente au nuage

de points. Comme extension de ce travail, nous pouvons notamment citer le travail de [ÖZTIRELI et al., 2010], qui s’est intéressé à évaluer l’impact des échantillons sur le spectre de l’opérateur de Laplace-Beltrami de la surface à ré-échantillonner, en mesurant de manière locale la modification de la surface due à la suppression de différents points. Les auteurs procèdent donc à un sous-échantillonnage du nuage de points et ensuite utilisent un schéma itératif pour répartir les échantillons sur la surface de manière à obtenir de bonnes propriétés spectrales. De manière similaire à [ÖZTIRELI et al., 2010], les auteurs de [CHEN et al., 2013] ont proposé une approche en 2 étapes. D’abord l’idée est d’échantillonner le nuage de points à l’aide d’un échantillonnage en disques de Poisson, en généralisant la mesure de distance inter-échantillons pour prendre en compte d’autres attributs (tel que la normale par exemple) et ensuite d’optimiser la position des échantillons, afin d’obtenir un échantillonnage ayant de bonnes propriétés de bruit bleu.

De leur côté, [LIPMAN et al., 2007] ont proposé quant à eux un autre opérateur de projection, appelé opérateur de *Projection Localement Optimale (Locally Optimal Projection (LOP) operator* en anglais), exprimé de manière à obtenir une formulation non-paramétrique, permettant de mieux représenter les géométries complexes que les MLS. Cette expression leur permet de nettoyer le nuage de points d’origine, ainsi que répartir de manière itérative un ensemble d’échantillons uniformément sur la surface numérisée. Le travail de [HUANG et al., 2009] s’est intéressé à étendre l’opérateur de [LIPMAN et al., 2007] pour considérer les nuages de points non-uniformément échantillonnés. Une formulation continue de cet opérateur a été proposée dans [PREINER et al., 2014], permettant d’accélérer fortement la vitesse de ré-échantillonnage. Ensuite, les auteurs de [LIAO et al., 2013] ont proposé une autre modification de l’opérateur LOP, pour préserver les détails de la surface lors du ré-échantillonnage. Enfin, une formulation encore différente de l’opérateur LOP a été proposée par [HUANG et al., 2013], en ré-échantillonnant d’abord les zones éloignées des arêtes franches et fortes courbures, et ensuite en sur-échantillonnant les zones proches des arêtes.

Dans la Partie II nous avons présenté la contribution de [CHEN et al., 2018a] dans leur choix des graphes pour la représentation de la surface sous-jacente à un nuage de points. Leur travail cependant se concentre principalement sur un problème d’échantillonnage. L’idée des auteurs est d’utiliser des filtres issus du traitement du Signal pour ré-échantillonner le nuage de points en fonction de l’utilisation visée de ces nuages de points. L’utilisation d’un filtrage passe-bas leur permet de ré-échantillonner uniformément un nuage de points tandis que l’utilisation d’un filtrage passe-haut génère un nuage de points réhaussant les contours.

4.3 ECHANTILLONNAGE EN DISQUES DE POISSON

L'échantillonnage en disques de Poisson est une méthode très populaire dans la communauté de l'informatique graphique [BALZER et al., 2009; BOWERS et al., 2010; COOK, 1986; LI et al., 2010; PEYROT et al., 2015; ULICHNEY, 1987, ...]. Un échantillonnage en disques de Poisson génère une distribution aléatoire d'échantillons sur un domaine, tout en imposant une *distance minimum* entre les échantillons sélectionnés. D'après [GAMITO et MADDOCK, 2009], un échantillonnage en disques de Poisson génère des échantillons sur un domaine $\Omega = [0, 1]^d$, représentant un cube unitaire d'un espace à d dimensions. Le résultat d'un tel échantillonnage consiste en un ensemble $S = \{s_i \in X; i = 1, 2, \dots, N\}$ de N échantillons dont les conditions d'échantillonnage peuvent être exprimées par :

$$\forall s \in S, \forall \Omega' \subseteq \Omega : P(s \in \Omega') = \int_{\Omega'} dx, \quad (3)$$

$$\forall s_i, s_j \in S : \|s_i - s_j\| \geq 2r, \quad (4)$$

où r est appelé le *rayon de la distribution*.

L'équation (3) définit que la probabilité pour un échantillon $s \in S$ de tomber dans un sous-domaine $\Omega' \subseteq \Omega$ est égale au volume de Ω' . L'équation (4) définit quant à elle, la distance minimum à respecter entre toutes les paires d'échantillons.

Ces deux équations combinées permettent donc d'obtenir une distribution qui n'est pas régulière (puisque toutes les régions du domaine Ω ont la même probabilité d'être tirées), mais dont les échantillons sont uniformément espacés (Figure 31).

Un échantillonnage en disques de Poisson est dit *maximal* si aucun échantillon ne peut être ajouté à la distribution finale sans invalider l'équation (4). Dans ce cas, la *distance maximum* entre n'importe quel point du domaine et l'échantillon sélectionné le plus proche de ce point est bornée et est directement liée au rayon r . On peut d'ailleurs facilement constater que cette borne vaut $2r$ car pour n'importe quel échantillon candidat situé à une distance supérieure ou égale à $2r$ du plus proche échantillon sélectionné, un disque de rayon r pourrait être construit sans intersecter aucun autre disque. Par conséquent nous pouvons donc formuler qu'un échantillonnage en disque de Poisson maximal garantit l'existence d'une *borne supérieure* sur la distance entre n'importe quel échantillon candidat $s \in S$ et le plus proche échantillon $x \in X$:

$$\forall x \in X, \exists s \in S, \|x - s\| < 2r \quad (5)$$

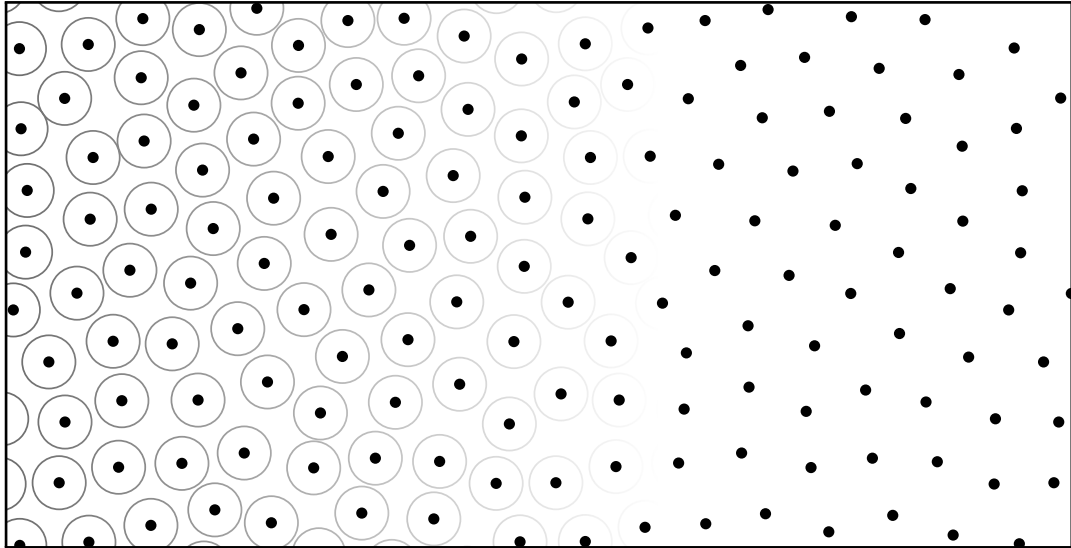


FIGURE 31 – Exemple d'échantillonnage en disques de Poisson avec les disques associés à certains échantillons.

Bien que présenté ici dans un cas planaire, l'échantillonnage en disques de Poisson a été généralisé à \mathbb{R}^d [BRIDSON, 2007] ainsi qu'à l'échantillonnage de domaines non-uniformes comme des variétés riemanniennes [BOWERS et al., 2010; CLINE et al., 2009; CORSINI et al., 2012; PEYROT et al., 2015, ...].

Ce type d'échantillonnage a la particularité de présenter des caractéristiques de *bruit bleu*, tel qu'explicité dans le travail de [HECK et al., 2013]. En effet, une des particularités de l'échantillonnage en disques de Poisson qui a fortement influé sur sa popularité est sa capacité à éviter les effets de repliement spectral (*aliasing* en anglais) basse-fréquence qui affectent la qualité visuelle d'une distribution, en les remplaçant par du bruit haute-fréquence, qui lui est plus difficilement distinguable [LI et al., 2010]. Il est vrai que l'échantillonnage en disques de Poisson est un processus qui peut être assez coûteux en temps de calcul par rapport à un échantillonnage régulier uniforme. Cependant ce dernier fournit des échantillonnages extrêmement réguliers (sous la forme d'une grille), amenant une forte anisotropie dans les distributions générées (dû au repliement spectral). De plus, quand le domaine d'échantillonnage n'est pas uniforme, comme c'est le cas dans le cadre des acquisitions 3D, générer un échantillonnage régulier uniforme n'est pas aussi trivial que dans \mathbb{R}^d . Enfin, étant donné que le processus d'échantillonnage impose une distance minimum entre les différents échantillons, au travers du rayon des disques de Poisson, celui-ci fixe la fréquence d'échantillonnage maximum présente dans la distribution finale, contrairement à un échantillonnage aléatoire uniforme sans aucun contrôle sur la distance inter-échantillons.

4.4 OBJECTIFS DE CETTE PARTIE

De manière générale, les méthodes de l'état de l'art offrent des solutions sans a priori sur l'origine des données, en essayant de représenter localement le comportement de la surface décrite par un nuage de points. Généralement ce choix leur impose l'utilisation de structures et d'algorithmes non adaptés au traitement de données massives comme celles considérées dans cette thèse.

La structure que nous avons introduite dans le Chapitre II propose un support pour l'utilisation de méthodes travaillant sur la surface sous-jacente à des nuages de points, peu importe leur taille. Nous proposons donc de se servir de celle-ci pour ré-échantillonner le nuage de points en généralisant un algorithme de lancer de fléchettes discret au traitement d'un ensemble de graphes. En utilisant un algorithme permettant de générer des distributions ayant des caractéristiques de bruit bleu, nous montrons qu'il est possible de simplifier un nuage de points, en améliorant la qualité de la distribution des échantillons sélectionnés.

Nous montrons que l'approche proposée permet d'obtenir des échantillonnages de qualité comparable à des échantillonnages directs de surfaces. Nous mettons en avant le passage à l'échelle permis par notre méthode, en ré-échantillonnant des nuages de points issus d'acquisitions réelles composés de plusieurs milliards de points.

ECHANTILLONNAGE EN DISQUES DE POISSON PAR GRAPHE LOCAUX

Nous présentons dans ce chapitre nos contributions dans le domaine de l'échantillonnage de nuages de points. Dans un premier temps, nous montrons comment une acquisition peut être échantillonnée, et généralisons ensuite à un ensemble d'acquisitions, en s'appuyant sur la structure. Nous présentons ensuite des résultats expérimentaux attestant de la qualité des distributions générées ainsi que des applications potentielles de notre algorithme de ré-échantillonnage.

Il est important de noter que ce travail n'est pas le premier à s'intéresser à l'échantillonnage en disques de Poisson de domaines non-uniformes comme des variétés riemanniennes. Nous invitons le lecteur à se référer au travail de [PEYROT, 2014] pour un état de l'art de ces méthodes. En suivant la classification de cet état de l'art, la méthode que nous proposons s'inscrit dans la famille des méthodes dites "directes", c'est-à-dire celles travaillant sur la surface originale. L'originalité de notre approche est sa manière de le faire en se servant d'un ensemble de graphes, et non pas d'un maillage polygonal pour réaliser les calculs de géodésiques, ainsi que de son traitement séquentiel par morceaux, permettant d'échantillonner des acquisitions composées de plusieurs milliards de points (Figure 32).

5.1 ECHANTILLONNAGE D'UNE ACQUISITION

Dans un premier temps, intéressons nous à l'échantillonnage d'une seule acquisition. Il s'agit d'un cas particulier de la structure présentée dans le Chapitre 3, ne donnant lieu qu'à la création d'un seul graphe $G_1(V_1, E_1)$. Par conséquent, tous les sommets de celui-ci n'ont aucun sommet correspondant, et donc $V_1^+ = V_1$.

Nous proposons d'utiliser un algorithme d'échantillonnage en disques de Poisson sur le graphe G_1 basé sur l'approche du lancer de fléchettes discret de [PEYROT et al., 2015]. L'aspect "discret" de l'approche vient des échantillons candidats. En effet, au lieu de prélever des échantillons sur un domaine continu, le domaine est composé d'un nombre fini d'échantillons candidats, les sommets du graphe G_1 .

Dans notre contexte, l'algorithme du lancer de fléchettes discret peut s'exprimer ainsi :

L'ensemble des échantillons candidats correspond aux sommets du graphe : $S_c = V_1^+$. L'idée est de réaliser la construction des disques de Poisson non pas sur un domaine $D = [0, 1]^d$, mais de le faire dans le graphe, en utilisant un algo-

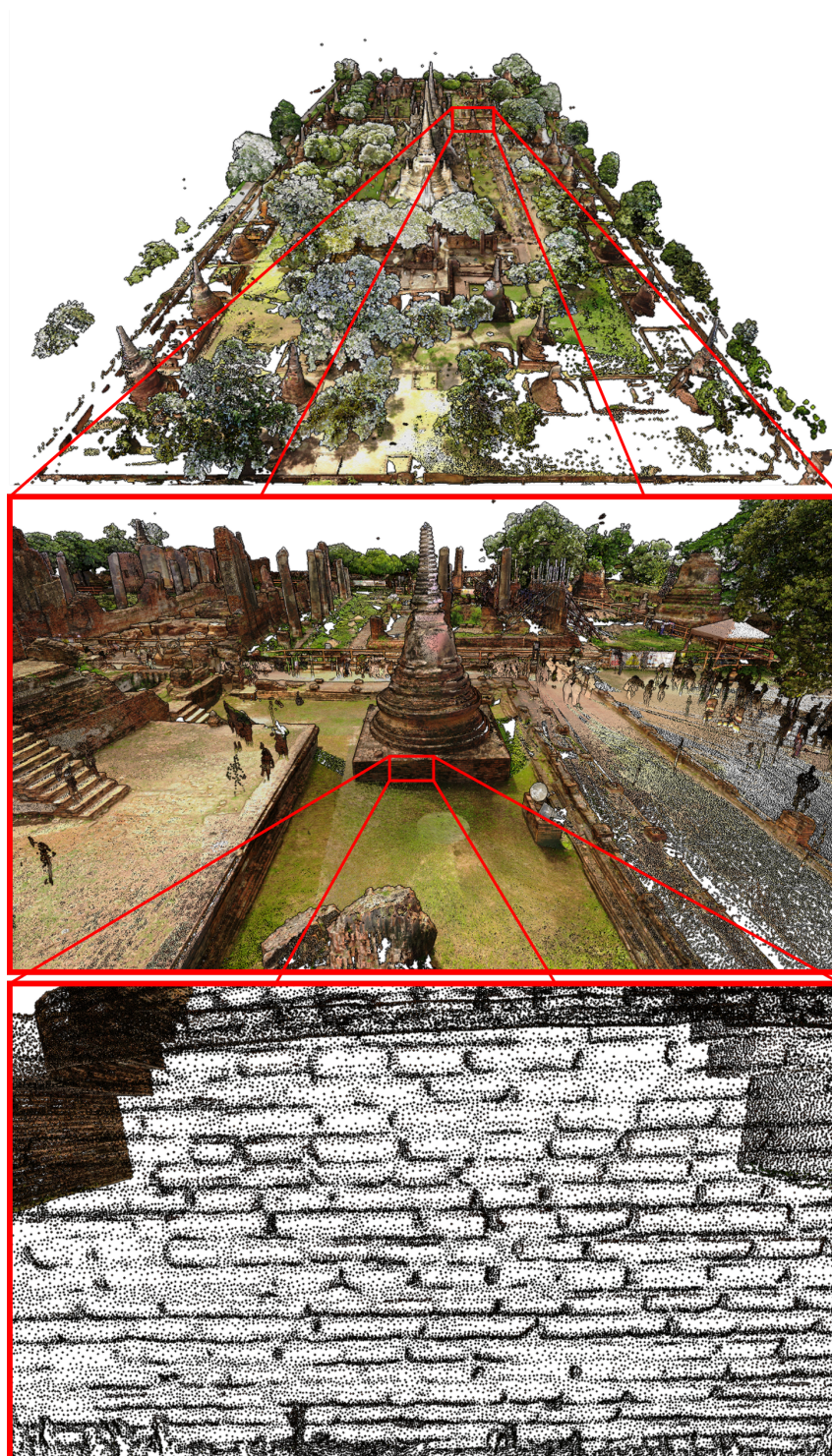


FIGURE 32 – Donnée *Wat Phra Si Sanphet*, ré-échantillonné de manière adaptative avec notre approche basée graphes. La donnée originale contient plus de 5 milliards de points résultant de l'agrégation de 156 acquisitions différentes. Nous sommes capable de représenter cette donnée avec seulement 2,8% du nombre de points original tout en préservant de fins détails de la surface (comme des briques dans le mur d'une zone couvrant plus de $40\ 000\text{m}^2$).

Entrées : S_c : l'ensemble des échantillons candidats

Résultat : S_r : l'ensemble des échantillons sélectionnés

tant que S_c n'est pas vide **faire**

```

|    $s_c = S_c.\text{élémentAléatoire}();$ 
|    $D_v = \text{constructionDisquePoisson}(s_c, r);$ 
|   si  $D_v$  intersecte un disque existant alors
|   |    $S_c.\text{supprimerElément}(s_c);$ 
|   sinon
|   |    $S_r.\text{ajouterElément}(s_c);$ 
|   fin

```

fin

Algorithme 1 : Algorithme de lancer de fléchettes discret proposé par [PEYROT et al., 2015].

rithme du plus court chemin [DIJKSTRA, 1959] pour les calculs de distances. Donc, dans notre cas, un disque de rayon r centré en un sommet $v \in V_1$ contient l'ensemble des sommets $v' \in V_1$ du graphe G_1 à distance inférieure ou égale à r du sommet v . De plus, afin de ne pas simplement considérer la connectivité de la donnée, mais de pouvoir lier la valeur du rayon des disques r aux distances 3D au sein du nuage de points, les arêtes E_1 du graphe sont pondérées selon la distance euclidienne dans \mathbb{R}^3 entre les deux sommets composant cette arête. Les distances entre les sommets représentent donc une estimation des géodésiques sur la surface entre les différents échantillons du nuage de points. De cette manière, les disques de Poisson représentent des patches surfaciques et non pas des boules dans \mathbb{R}^3 (Figure 33). Cela permet d'invalider les zones proches d'un sommet sur la surface et non pas les zones proches dans l'espace euclidien.

Un point intéressant de l'échantillonnage en disques de Poisson est la manière dont celui-ci permet un contrôle direct sur la densité de la distribution finale. En effet, le rayon des disques r fixe la proximité des échantillons sélectionnés. Plus r est petit, plus la distribution finale sera dense et inversement, plus r est grand, plus la distribution finale sera éparse.

La Figure 34 présente la distribution des disques de Poisson sur la surface sous-jacente au nuage de points original en utilisant notre approche. Il est intéressant de noter que la croissance des disques dans les zones présentant des occultations est naturellement stoppée grâce à connectivité des graphes ayant été construits dans le Chapitre 3. En effet, l'étape de détection de contours y étant introduite a permis d'empêcher les paires de sommets ayant une forte variation d'intensité dans la carte de profondeur d'être reliés par une arête dans le graphe. S'appuyer sur la connectivité de ces graphes permet donc d'établir un domaine de définition à la

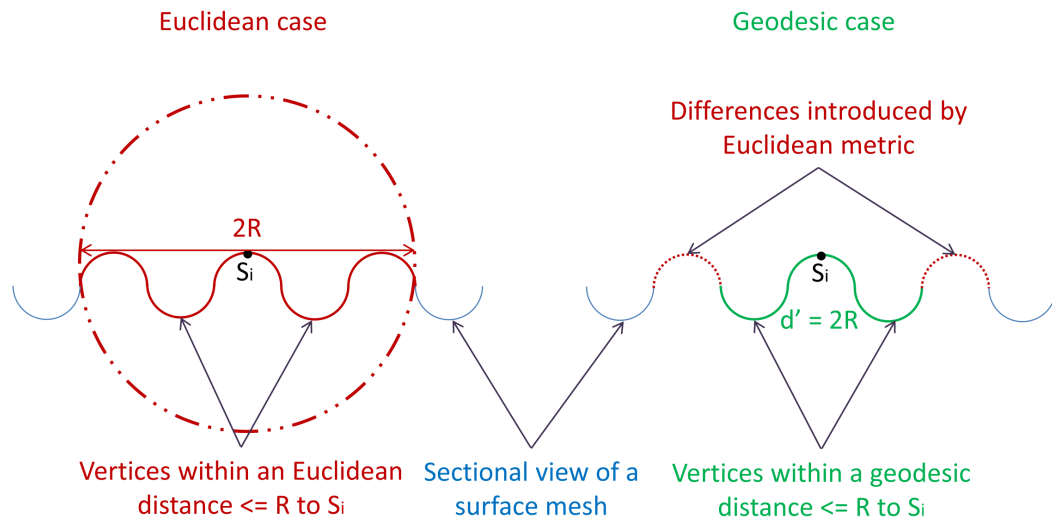


FIGURE 33 – Biais de mesure entre un disque prenant en compte la métrique euclidienne (à gauche) et la métrique géodésique (à droite). Image tirée de [PEYROT et al., 2015].

construction des disques de Poisson, permettant d'interrompre leur croissance au niveau des contours ayant été détectés lors de leur construction.

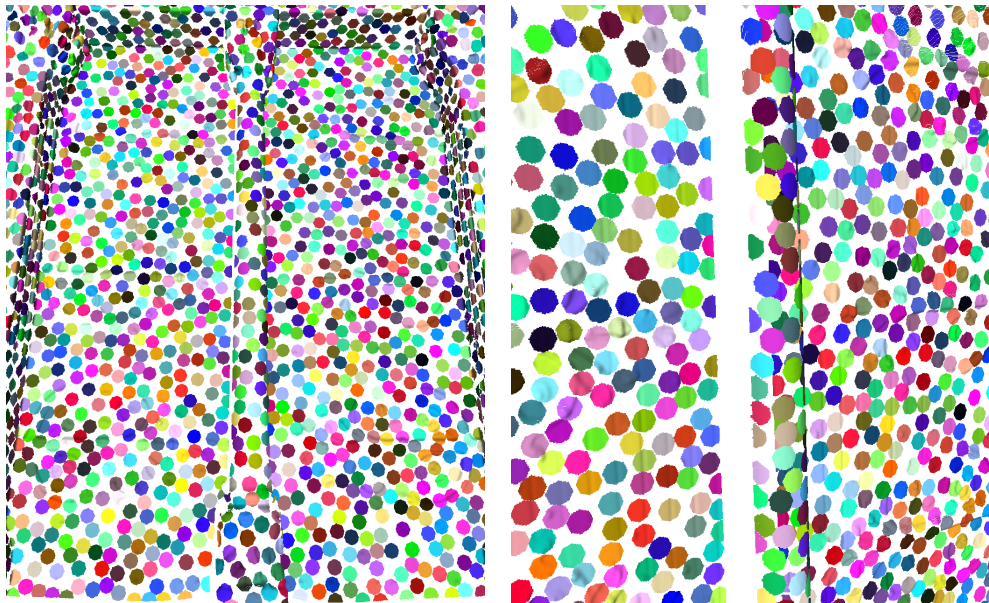


FIGURE 34 – Distribution obtenue avec un échantillonnage en disques de Poisson maximal d'une acquisition de *St Trophime*, obtenu avec notre approche par graphes locaux. De gauche à droite : vue de face et vue de côté. La croissance des disques dans les zones présentant des occultations est naturellement stoppée grâce à la construction des graphes du Chapitre 3, ne connectant pas les sommets ayant de fortes variations d'intensité dans la carte de profondeur.

5.1.1 Echantillonnage adaptatif

Le fait de baser l'échantillonnage sur un graphe permet de facilement modifier la métrique à modéliser lors de la construction des disques, étant donné que la pondération des arêtes du graphe influe sur les distances entre ses différents sommets. Ainsi, il est possible de modifier la métrique géodésique, présentée précédemment afin qu'elle prenne en compte d'autres caractéristiques.

Un des éléments importants à propos du traitement de nuages de points massifs concerne l'incapacité de visualiser la donnée avant le traitement de ceux-ci, étant donné leur taille. S'il est possible d'extraire des parties des scènes numérisées, il est en général assez difficile, avec des ordinateurs classiques, de pouvoir obtenir une vue globale de la donnée d'origine sans passer par l'utilisation d'algorithmes dédiés à la structuration et à la visualisation de la donnée (comme l'ensemble des méthodes présentées dans la Partie II). Les algorithmes doivent donc travailler en quelque sorte "à l'aveugle" pour ré-échantillonner les nuages de points.

Nous avons donc cherché à adapter la densité de l'échantillonnage en fonction de certaines caractéristiques propres à la surface sous-jacente au nuage de points de manière automatique. En particulier, nous nous sommes intéressés à estimer la courbure de la surface représentée par un nuage de points, pour échantillonner de manière plus dense les zones de forte courbure. Pour cela, nous avons décidé d'exploiter la connectivité du graphe afin d'estimer la courbure de la surface en chaque point du nuage de points.

Estimation de la normale

Quand une acquisition est obtenue sous une forme régulière, comme c'est le cas pour nos données, il est possible d'exploiter la structure des données afin d'évaluer des normales et de facilement les orienter. En effet, dans notre cas, étant donné que les points acquis sont ordonnés sur une grille cartésienne \mathbb{Z}^2 , nous pouvons prendre avantage de cette structure pour calculer des normales comme le produit vectoriel des vecteurs tangents à la surface en chacun des points $p \in PC$, dans les deux directions principales de la grille. Les vecteurs tangents sont simplement obtenus par différences partielles selon les directions principales x et y entre les points de la grille :

$$T_x(p) = \frac{\delta f}{\delta x}, T_y(p) = \frac{\delta f}{\delta y}. \quad (6)$$

Le produit vectoriel de ces deux vecteurs tangents permet d'obtenir la normale associée au point p :

$$N(\vec{p}) = T_x(p) \times T_y(p) \quad (7)$$

comme illustré sur la Figure 35.

Dans notre cas, pour obtenir des normales dans l'espace 3D, il suffit de réaliser le produit vectoriel des vecteurs tangents à la surface, construits à partir des points 3D ayant été connectés ensemble grâce à l'équivalence de ces points avec les pixels de la grille cartésienne.

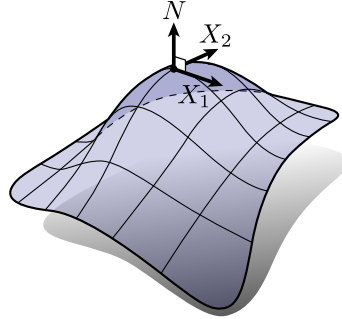


FIGURE 35 – Normale N en un point de la surface défini comme le produit vectoriel des deux vecteurs tangents X_1 et X_2 (définissant le plan tangent à la surface en ce point). Illustration tirée de [CRANE et al., 2013].

Estimation de la courbure

Nous nous sommes inspirés de la méthode utilisant les votes de tenseurs de normales de l'approche de [PARK et al., 2012]. Pour chaque pixel valide p de coordonnées (u, v) dans une carte de profondeur un tenseur T est calculé :

$$T = \sum_{u'=u-k}^{u+k} \sum_{v'=v-k}^{v+k} \vec{N}' \cdot \vec{N}'^T \quad (8)$$

où N' représente la normale associée au pixel p' de coordonnées (u', v') et k représente la taille du voisinage considéré pour le vote de tenseurs. Chaque tenseur est ensuite décomposé en trois valeurs propres $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Les auteurs de [PARK et al., 2012] définissent ensuite une valeur permettant de distinguer entre les zones planes et les zones présentant de fortes courbures ainsi que des arêtes franches :

$$\omega = \frac{\lambda_2 + \lambda_3}{\lambda_1}. \quad (9)$$

La valeur de ω est proche de 1 pour un point se situant dans une zone plane et elle augmente quand un point se situe sur une zone courbe. Dans notre cas, au lieu d'utiliser cette information comme un moyen de classifier les points en fonction de leur courbure, nous nous servons de cette valeur pour pondérer le poids de chaque arête, afin de faire varier la croissance des disques de Poisson. Afin de fixer des valeurs de rayon minimum r_{\min} et maximum r_{\max} , nous proposons de mettre les valeurs de ω à l'échelle, de façon à ce que $\omega_{\min} = r_{\min}$ et $\omega_{\max} = r_{\max}$.

De cette manière, il est possible de placer un plus grand nombre d'échantillons dans les régions de forte courbure (présentant du détail), et moins d'échantillons

dans les régions planes (ne présentant pas ou peu de hautes fréquences) (Figure 36).

Ce qui est intéressant avec cette approche, c'est qu'en combinant la connectivité de la grille et la position 3D des points associés à chaque pixel valide, elle travaille en quelque sorte sur la surface sous-jacente du nuage de points capturé depuis une acquisition. La faible complexité de cette approche et sa forte parallélisation en font une méthode ayant un coût négligeable lors de l'échantillonnage.

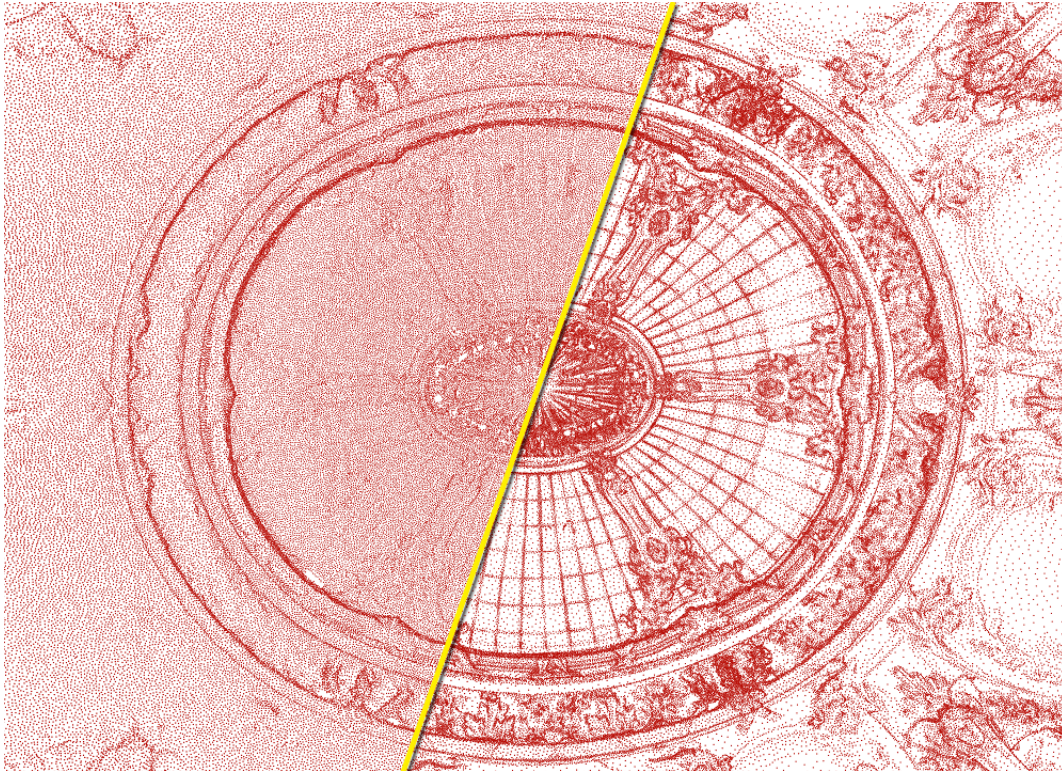


FIGURE 36 – Différence entre un échantillonnage uniforme (à gauche) et adaptatif (à droite) de la donnée *Intérieur*. Les deux nuages de points contiennent approximativement le même nombre d'échantillons. On peut remarquer, entre autres, que l'échantillonnage adaptatif, en concentrant plus d'échantillons dans les régions de fortes courbures, permet de faire ressortir visuellement plus de détails.

Evidemment, il est possible de se baser sur d'autres informations potentiellement présentes dans les nuages de points (couleur, confiance dans la position des points, etc.) selon l'objectif et l'application visés. Cependant, durant cette thèse, nous nous sommes intéressés uniquement à l'information géométrique présente dans la donnée numérisée.

5.1.2 Traitement du bruit d'acquisition

Les différentes conditions atmosphériques (humidité, poussière, etc.) impactent la qualité des acquisitions (en particulier elles peuvent altérer la valeur de la position à laquelle un point a été capturé). De ce fait, les calculs réalisés peuvent s'en retrouver fortement biaisés.

Concernant les acquisitions issues de LiDARs terrestres, la qualité des systèmes d'acquisition permet de générer des données contenant généralement peu de bruit par rapport à d'autres systèmes d'acquisitions. Dans notre cas, nous avons remarqué que ce bruit pouvait commencer à être gênant lors de l'estimation de normales sur le nuage de points, tel que présenté précédemment. En effet, la méthode proposée, bien qu'ayant une très faible complexité et étant adaptée au traitement de données massives, est assez sensible au bruit. C'est pour cette raison que nous avons cherché à atténuer celui-ci, en essayant toujours de s'appuyer sur la structure des acquisitions générées.

De nombreuses méthodes s'intéressent au problème du filtrage de données en s'inspirant de la méthode des Moving Least Squares (MLS). Concernant les nuages de points, l'idée générale est de construire un modèle polynomial, définissant localement le comportement de la surface autour de chaque point à partir du voisinage de ce point, et de projeter chaque point sur le modèle construit à partir du voisinage de ce point [LEVIN, 2004]. L'opération est réalisée de manière itérative jusqu'à arriver à un état stationnaire, où la position de chaque point coïncide avec la position de sa projection sur le modèle construit.

Dans notre cas, en considérant des acquisitions structurées, l'utilisation de telles méthodes (travaillant dans l'espace 3D) ne permet pas de conserver la structure de l'acquisition. De plus, ces méthodes ont tendance à fortement lisser les détails et peuvent avoir du mal à passer à l'échelle. Etant donnée cette contrainte forte dans notre travail, nous avons décidé de nous intéresser à une méthode de lissage basée image appliquée sur la carte de profondeur directement.

Dans le domaine du traitement de l'image, de nombreux algorithmes de débruitage ont été proposés, considérant différentes particularités propres aux images traitées. En particulier, dans le cas d'images présentant de *fortes variations d'intensité* (contours notamment), il a été montré que les méthodes prenant en considération ces variations lors du filtrage sont plus efficaces dans la préservation des forts contours. En particulier, le filtrage bilatéral [AURICH et WEULE, 1995; SMITH et BRADY, 1997; TOMASI et MANDUCHI, 1998] est une méthode de filtrage non-linéaire dont le but est de lisser un signal tout en préservant les fortes variations d'am-

plitude. Cette méthode se base sur l'utilisation de deux noyaux gaussiens G_{σ_s} et G_{σ_r} prenant en compte à la fois la distance spatiale entre les pixels (G_{σ_s}) (filtrage gaussien classique) mais aussi la différence d'intensité entre ces pixels (G_{σ_r}) :

$$\hat{I}(p) = \frac{1}{W_p} \sum_{q \in \Omega_p} J_q G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|J(p) - J(q)|), \quad (10)$$

où W_p correspond au facteur de normalisation :

$$W_p = \sum_{q \in \Omega_p} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|J(p) - J(q)|).$$

La combinaison de ces deux noyaux permet que le lissage de la valeur d'intensité d'un pixel soit principalement influencé par les pixels spatialement proches et dont l'intensité est similaire (Figure 37).

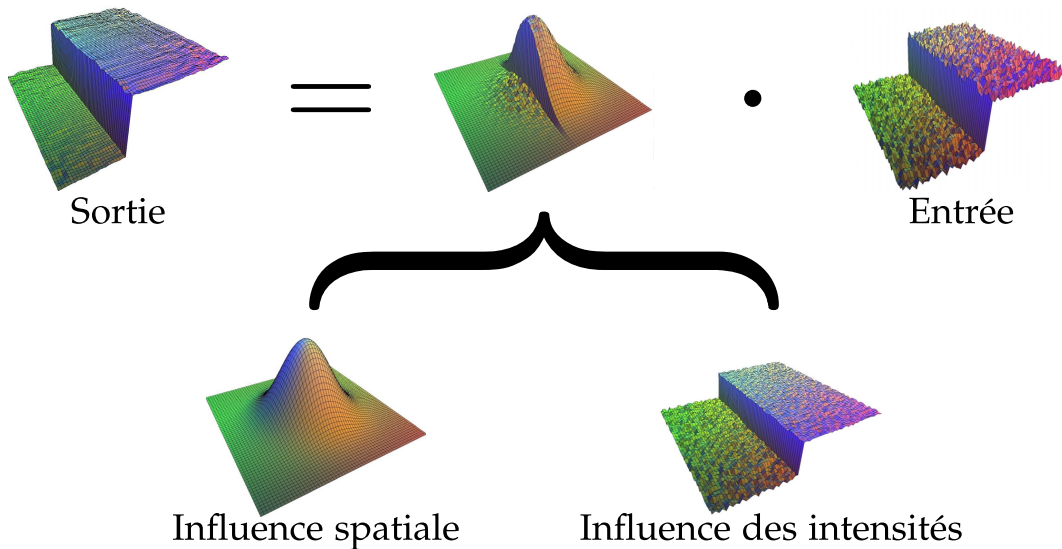


FIGURE 37 – Schéma du filtrage bilatéral. Illustration inspirée de [DURAND et DORSEY, 2002].

On peut constater que si les régions ayant été fortement perturbées par un bruit d'acquisition (généralement, il s'agit de régions très proches de la position du scanner) ont bien été lissées, les régions voisines de forts contours dans la carte de profondeur n'ont pas été affectées par les points de l'autre côté de ces contours (Figure 38). On peut notamment remarquer que les régions éloignées en 3D (ayant une forte variation d'intensité dans la carte de profondeur) ne se sont pas influencées mutuellement, évitant de déplacer des points dans des régions aberrantes.

Ce qui est intéressant c'est que cette méthode de lissage ne se base pas sur la position 3D des points, mais uniquement sur l'organisation des points ainsi que l'intensité associée à chaque pixel dans la carte de profondeur. De cette manière, le lissage s'effectue uniquement entre des points proches sur la surface numérisée.

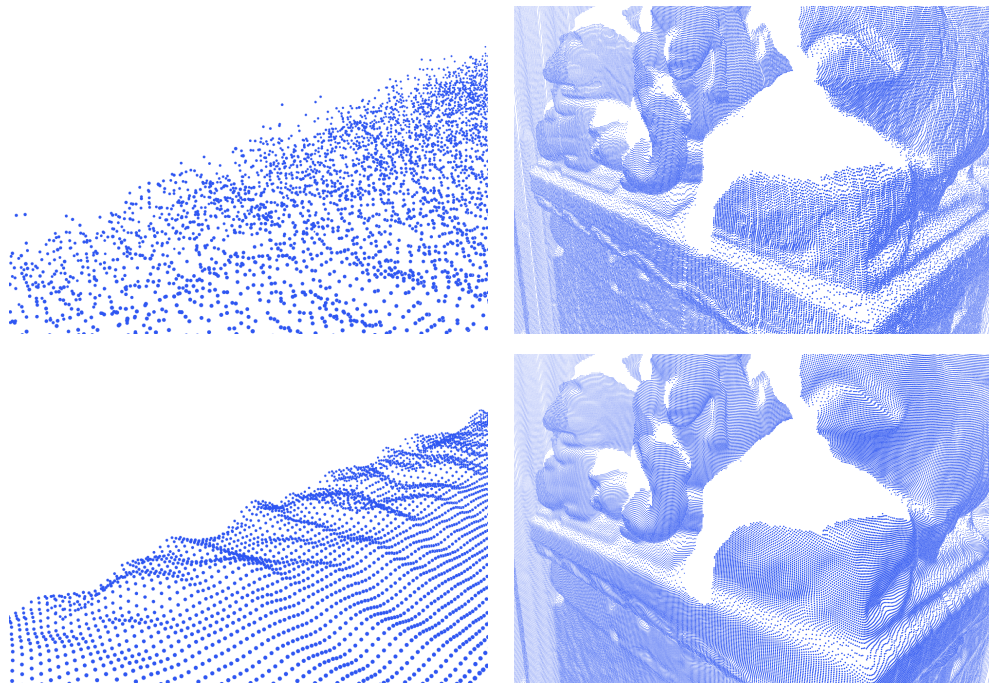


FIGURE 38 – Impact du filtrage bilatéral appliqué à une carte de profondeur du modèle *Strophime* sur la qualité du nuage de points (haut : avant filtrage ; bas : après filtrage).

Nous décidons donc maintenant d'estimer les normales avec la méthode présentée sur des nuages de points avec et sans filtrage. Pour visualiser les normales de chaque donnée, nous présentons, des *cartes de normales*, où la couleur de chaque pixel représente la normale calculée pour le point associé à ce pixel (Figure 39). Les canaux rouge, vert et bleu représentent respectivement les coordonnées x , y et z de la normale estimée.

Dès l'utilisation du filtre, même avec un petit noyau, nous pouvons déjà remarquer une nette diminution du bruit présent lors de l'estimation de la normale (régions dont les normales sont plus homogènes). Aussi, plus G_{σ_s} - déterminant le voisinage spatial à considérer - est grand, plus les normales sont homogènes dans les régions planes. Cependant, il est nécessaire de ne pas utiliser un noyau trop grand, puisque si celui-ci réduit fortement le bruit présent dans la donnée, il diminue aussi fortement les détails présents dans la donnée. De manière empirique, nous avons remarqué qu'avec l'ensemble des acquisitions, un noyau G_{σ_s} ayant une taille entre 2 et 3 pixels offrait un bon compromis entre réduction du bruit d'acquisition et préservation des détails.

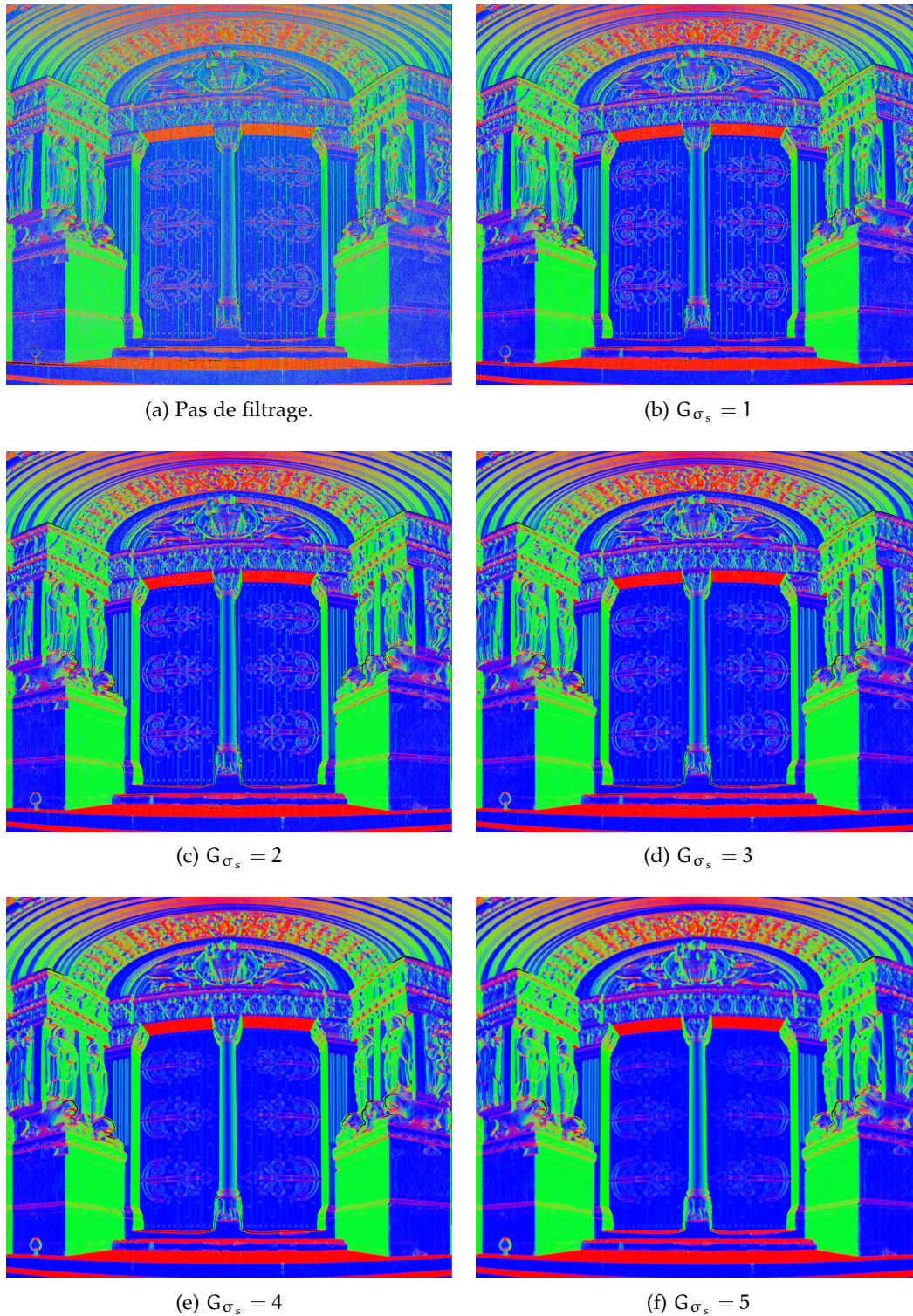


FIGURE 39 – Estimation de normales au sein d'un nuage de points à l'aide de l'approche proposée, pour différentes tailles du noyau G_{σ_s} du filtre bilatéral.

5.2 ECHANTILLONNAGE D'UN ENSEMBLE D'ACQUISITIONS

Maintenant que nous avons expliqué le principe de l'échantillonnage en disques de Poisson pour une acquisition, la généralisation à un ensemble d'acquisitions est assez triviale grâce à la représentation par graphes locaux.

Rappelons que dans ce cas, nous avons un ensemble de k graphes $G = \{G_1, G_2, \dots, G_k\}$ représentant chacun la connectivité de la surface capturée par une acquisition particulière qui sont connectés entre eux au travers de leurs sommets correspondants. Ainsi, pour éviter les calculs redondants entre des régions où plusieurs graphes se surperposent, dans chaque graphe $G_i(V_i, E_i)$, l'ensemble des sommets V_i a été réparti en deux sous-ensembles V_i^+ - les sommets les plus densément échantillonnés parmi leurs sommets correspondants - et V_i^- - le reste des sommets.

La généralisation de l'algorithme d'échantillonnage présenté précédemment consiste en deux étapes. Pour chaque graphe $G_i \in G$:

- l'Algorithme 1 est appliqué au graphe G_i , considérant les sommets V_i^+ comme les échantillons candidats,
- pour chaque autre graphe $G_j(V_j, E_j) \in G, i \neq j$, les sommets V_j récupèrent l'information d'appartenance à un disque spécifique à partir de leur sommet correspondant dans V_i .

Le fait de transmettre l'information entre les différents graphes permet d'initialiser le domaine d'échantillonnage de chaque graphe. En effet, les disques construits dans les graphes ayant déjà été traités invalident des régions, trop proches d'échantillons déjà sélectionnés dans ces graphes, avant d'appliquer l'Algorithme 1 sur le prochain graphe à traiter. De cette manière, il n'est pas possible de sélectionner des échantillons qui ne respecteraient pas l'équation (4). La Figure 40 illustre le principe général de l'approche proposée.

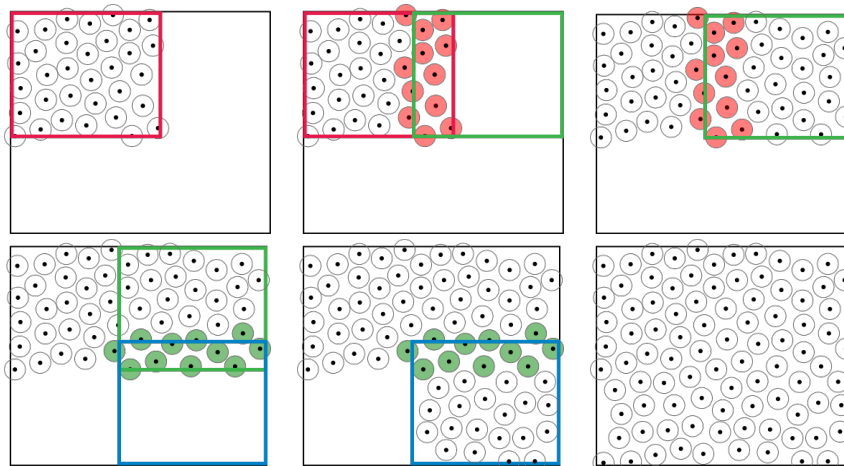


FIGURE 40 – Principe général de notre approche. Chaque couleur identifie une zone traitée par un graphe donné.

La combinaison de ces deux étapes permet non seulement d'avoir un échantillonnage qui soit localement uniforme (grâce à l'algorithme d'échantillonnage en disques de Poisson), mais aussi globalement uniforme (grâce à la transmission des distributions entre les différents graphes). En particulier, nous pouvons voir que même lorsque de fortes variations de densité sont présentes dans l'échantillon-

nage original, la distribution finale est globalement uniformément échantillonnée (Figure 41).

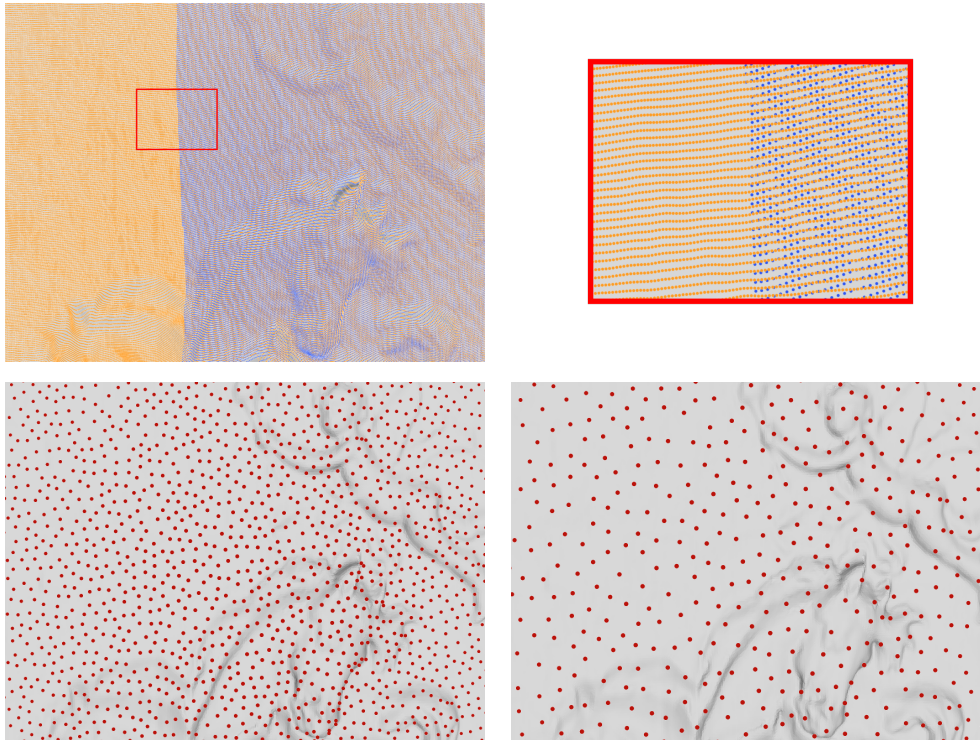


FIGURE 41 – Les nuages de points générés par notre approche ne souffrent pas de distorsion en termes de qualité d'échantillonnage par rapport à la donnée d'origine. En haut, deux acquisitions se superposant, amenant à une forte variation de la densité d'échantillonnage dans le nuage de points original (donnée *Facade*). En bas, deux distributions uniformes, à différentes densités, obtenues avec notre approche basée graphes. Pour des raisons de visualisation, de faibles densités d'échantillonnages cibles ont été choisies.

5.2.1 Résultats visuels

Nous invitons le lecteur à se référer à l'Annexe A pour obtenir les caractéristiques de la machine sur laquelle les expérimentations ont été effectuées ainsi que des informations techniques concernant les différentes données de test utilisées.

Nous proposons de visualiser les distributions générées par notre algorithme sur différentes données issues d'acquisitions réelles. Nous présentons dans un premier temps des distributions à partir d'acquisitions synthétiques, considérant un échantillonnage uniforme (Figure 42) et adaptatif en fonction de la courbure (Figure 43). Rappelons que les données synthétiques sont générées à l'aide de la méthode présentée dans le Chapitre 3. Chaque modèle est donc représenté par 12 acquisitions ayant chacune une résolution de 1024x1024 pixels.

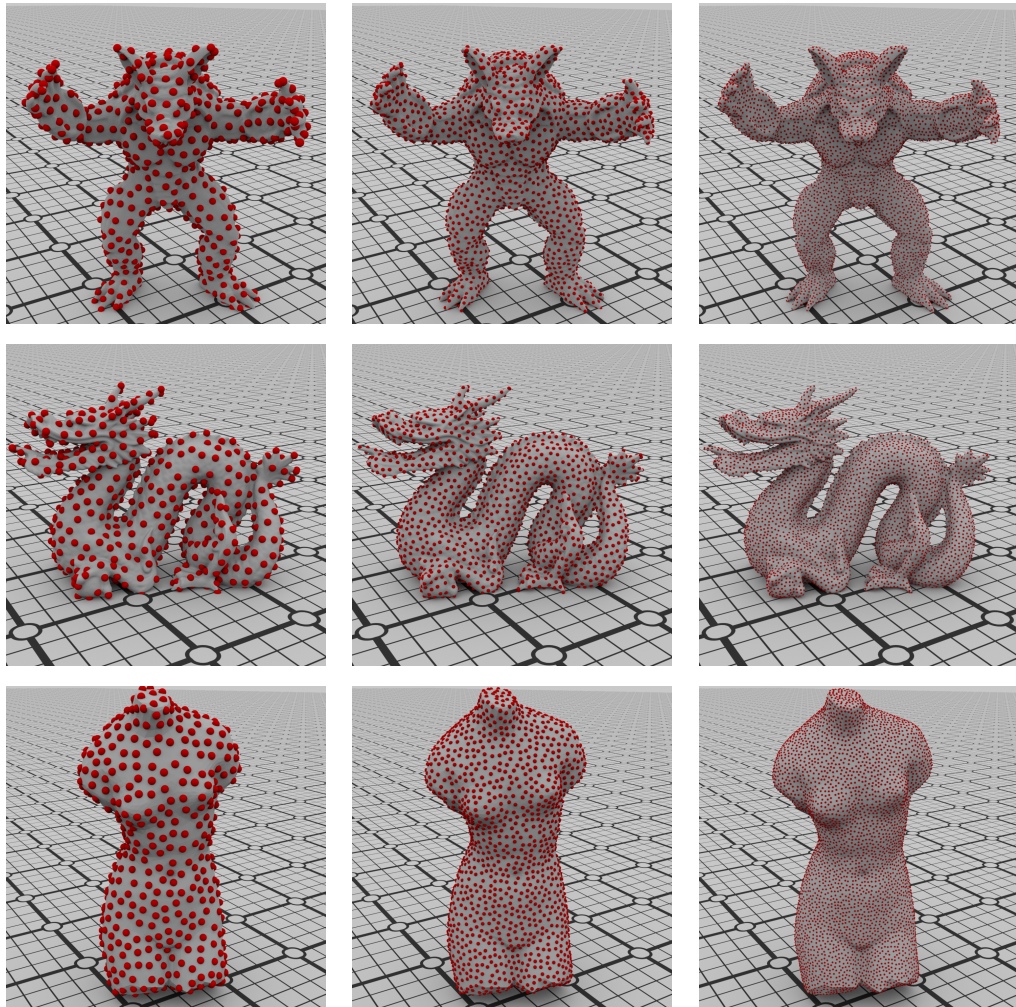


FIGURE 42 – Distributions uniformes générées par notre approche sur différents modèles (de haut en bas : *Armadillo*, *Dragon* et *Venus*) à 3 différentes densités cibles.

Nous pouvons constater que de manière générale, les différents échantillonnages générés sont de plutôt bonne qualité et couvrent l'ensemble de la surface des modèles 3D d'origine.

Nous présentons ensuite des distributions à partir d'une campagne d'acquisitions réelles, considérant des échantillonnages adaptatifs (Figure 44). L'échantillonnage adaptatif, en fonction de la courbure de la surface sous-jacente à un nuage de points, permet d'obtenir des distributions présentant énormément de détails de la surface d'origine malgré la superficie totale de la campagne d'acquisition (Figure 45).

Afin de pouvoir mieux visualiser la distribution des échantillons sur la surface décrite par le nuage de points, nous avons reconstruit de manière locale des surfaces détaillées à partir de la position des échantillons du nuage de points origine, à l'aide de l'algorithme de reconstruction de surfaces de Poisson [KAZHDAN et

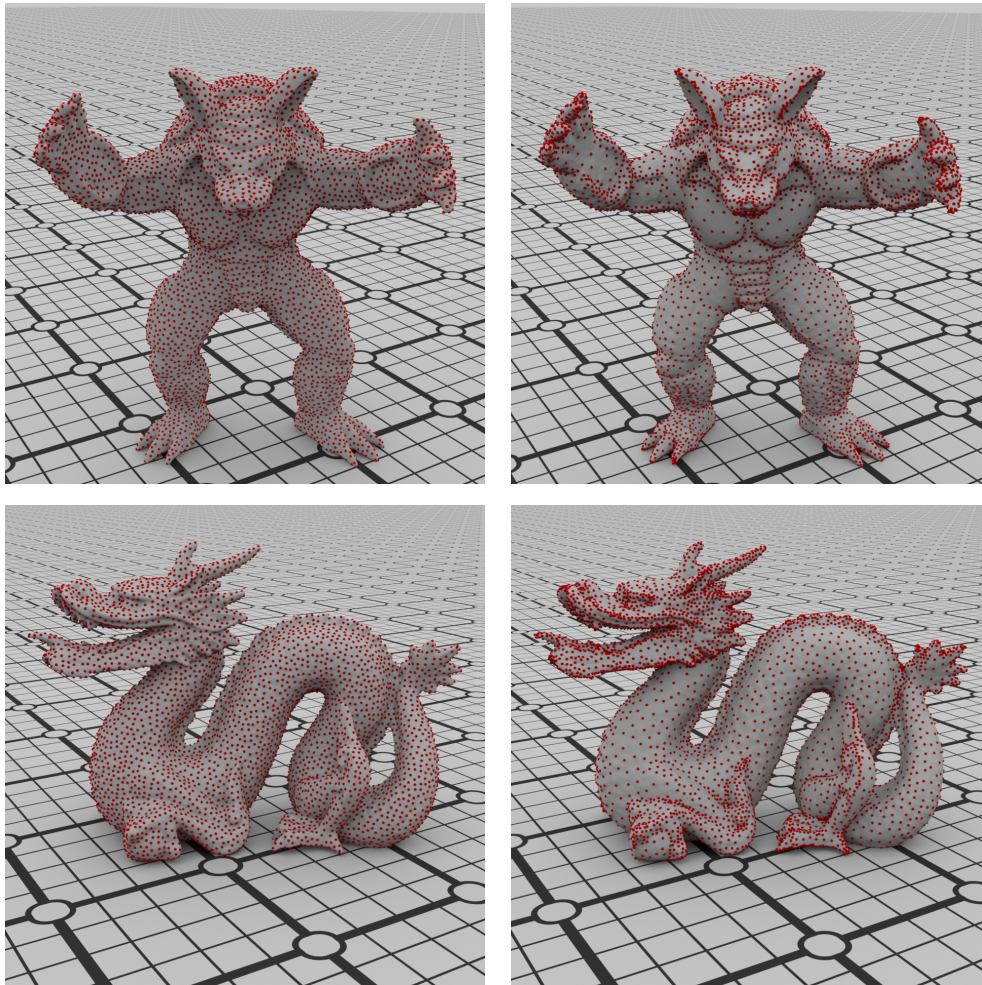


FIGURE 43 – Comparaison de distributions uniformes (à gauche) et adaptatives (à droite) sur différents modèles (de haut en bas : *Armadillo* et *Dragon*). Pour chaque modèle, les deux distributions contiennent le même nombre d'échantillons.

HOPPE, 2013]. Ainsi, nous pouvons avoir une représentation locale de la surface (d'une très petite zone d'une campagne d'acquisition) finement reconstruite, sur laquelle nous pouvons visualiser plus en détail les distributions générées (qui elles ont été générées en considérant l'ensemble de la campagne d'acquisition) (Figure 46).

En dépit de l'importante surface couverte par les campagnes d'acquisitions, nous pouvons remarquer que l'approche proposée est capable de réduire considérablement le nombre de points tout en conservant de fins détails géométriques de la surface numérisée.

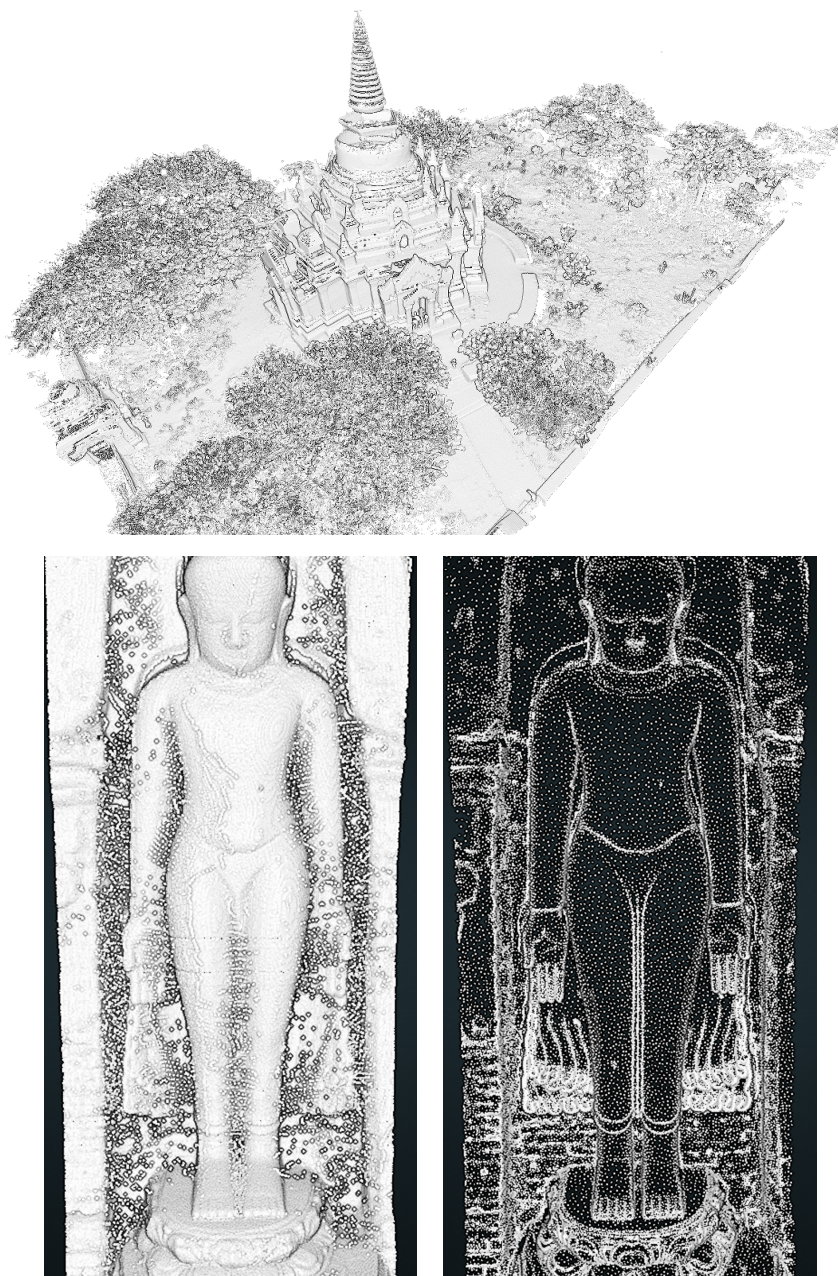


FIGURE 44 – Distribution adaptative générée à partir d’une campagne d’acquisition massive (*Eim Ya Kyaung*). En haut, on peut voir l’ensemble du site ré-échantillonné par notre approche à 0,6% du nombre initial de points. La deuxième ligne présente un morceau du nuage de points original (à gauche) et le résultat de notre algorithme d’échantillonnage à 0,6% du nombre initial de points (à droite). Grâce à l’utilisation d’une métrique adaptative dépendante de la courbure de la surface, la réduction du nombre de points n’empêche pas la préservation de fins détails géométriques du site numérisé.

5.2.2 Garanties théoriques

Les bornes théoriques de l’échantillonnage en disques de Poisson présentées précédemment permettent d’offrir des garanties sur les nuages de points géné-

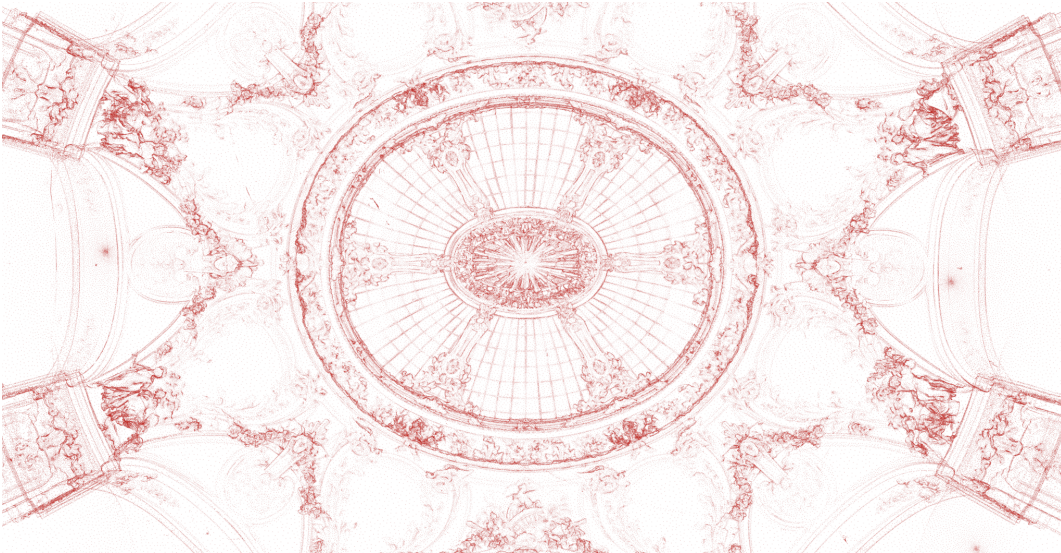


FIGURE 45 – Photographie (en haut) et distribution résultant de notre algorithme d'échantillonnage (en bas) à partir de la donnée *Intérieur*.

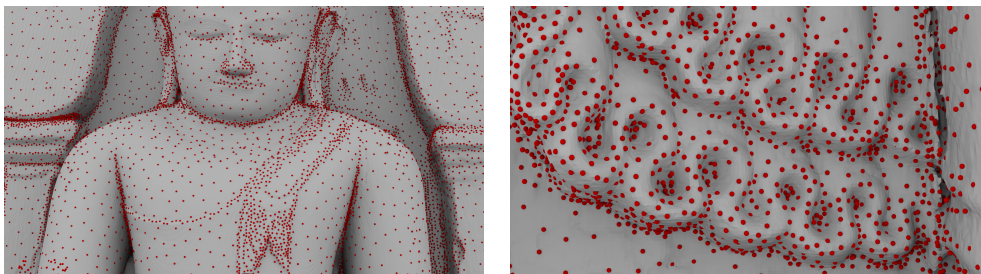


FIGURE 46 – Visualisation de la qualité d'une distribution adaptative générée à partir de la donnée *Eim Ya Kyaung*.

rés par notre algorithme. En effet, rappelons qu'un échantillonnage en disques de Poisson maximal fournit deux bornes théoriques concernant les distances entre échantillons. Dans un premier temps, les échantillons de la distribution finale sont tous éloignés au minimum d'une distance $2r$. Dans un second temps, la maximalité de l'échantillonnage nous assure que, au pire, un point du nuage de points d'origine est à une distance $2r$ au maximum d'un échantillon sélectionné.

La donnée d'origine est tellement volumineuse qu'il n'est en général pas possible de la visualiser entièrement à sa résolution la plus fine. Typiquement, à la fin d'une campagne d'acquisition, les utilisateurs ont deux possibilités :

- visualiser des morceaux de la donnée à pleine résolution de manière indépendante, s'abstrayant donc d'une vue globale de la campagne d'acquisition ;
- visualiser la donnée de manière globale, mais à une résolution plus grossière, ne pouvant donc pas afficher les détails des différentes parties des sites numérisés.

Il est donc difficile, en particulier lorsque le nombre d'acquisitions et la quantité de points acquis est très importante, de pouvoir analyser la donnée en détails manuellement pour en obtenir une version ré-échantillonnée de bonne qualité. De ce fait, avoir des garanties théoriques, comme celles que fournit l'échantillonnage en disques de Poisson maximal est un élément important, permettant une certaine confiance des utilisateurs dans la manipulation d'un d'algorithme travaillant "à l'aveugle".

5.2.3 Analyse de la qualité de l'échantillonnage

Nous nous intéressons maintenant à l'évaluation de la qualité des distributions générées par notre algorithme. Afin d'évaluer la qualité des échantillonnages générés en pratique et de manière quantitative nous décidons d'utiliser *l'analyse différentielle* de [WEI et WANG, 2011]. L'analyse différentielle peut être vue comme une généralisation de l'analyse de Fourier à des domaines non-uniformes (comme des variétés riemanniennes).

Les équations de la transformation de Fourier discrète $\mathcal{F}(f)$ et du périodogramme $\mathcal{P}(f)$ (la magnitude de la transformation de Fourier) à partir d'un ensemble de n points $P_n = \{p_1, p_2, \dots, p_n\}$ sont :

$$\mathcal{F}(f) = \sum_{k=0}^{N-1} e^{-2\pi i(\omega p_n)} \quad (11)$$

$$\mathcal{P}(f) = |\mathcal{F}(f)|^2, \quad (12)$$

Le travail de [WEI et WANG, 2011] a montré que l'expression du périodogramme $\mathcal{P}(f)$ peut être remaniée pour s'écrire :

$$P(f) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \cos(2\pi f(p_k - p_j)). \quad (13)$$

et ainsi ne plus dépendre de la position absolue des échantillons, mais des distances inter-échantillons. Afin d'obtenir des statistiques pertinentes concernant l'analyse d'un échantillonnage aléatoire, un *spectre de puissance* est construit comme étant la moyenne des périodogrammes obtenus depuis les différentes distributions générées. De cette manière, le spectre de puissance construit donne une bonne statistique des distributions, plutôt que de mettre en avant des particularités d'un échantillonnage donné qui ne seraient pas forcément représentatives de l'ensemble des distributions à la densité ciblée.

A partir du spectre de puissance, deux métriques 1D peuvent être extraites afin d'analyser la qualité des distributions : le *Spectre de Puissance Moyen Radial (SPMR)* et l'*anisotropie*. Le SPMR permet d'évaluer la distribution radiale des distances entre les échantillons et l'anisotropie évalue l'uniformité radiale de l'échantillonnage. Ces métriques sont générées en calculant, respectivement pour le SPMR et l'anisotropie, la moyenne et la variance de la puissance spectrale telle que définie dans l'équation (13) (Figure 47).

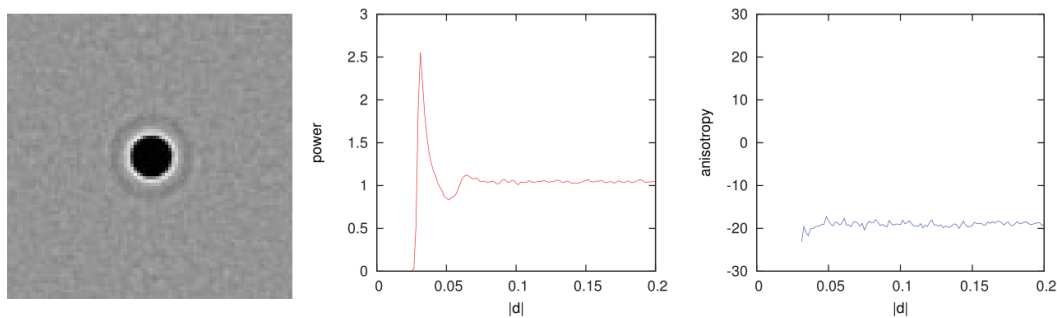


FIGURE 47 – A partir d'un périodogramme moyen (à gauche), deux métriques 1D peuvent être extraites : le SPMR (au centre) et l'anisotropie (à droite). Image tirée de [WEI et WANG, 2011].

La formulation de l'équation (13) permet aux auteurs d'extraire des informations pertinentes pour l'analyse d'échantillonnages sur des domaines non-uniformes et pour l'analyse d'échantillonnages non-uniformes, que la formulation classique à l'aide de la transformation de Fourier ne permet pas d'étudier (Figure 48).

Leur travail montre donc que l'organisation spatiale des échantillons est directement reliée aux propriétés spectrales de la distribution, à partir du moment où il est possible de définir correctement la distance entre les échantillons sur le domaine considéré (comme la distance géodésique, dans le cas de surfaces). Les calculs de distance sur la surface se font en utilisant l'outil *exponential maps* [SCHMIDT

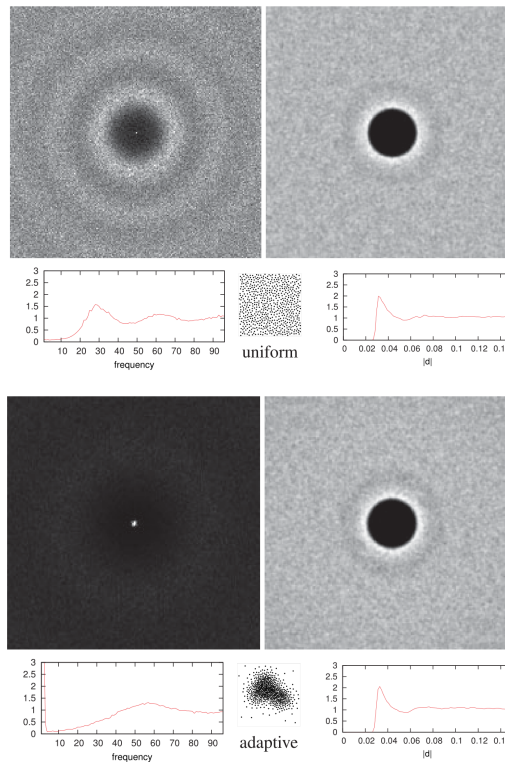


FIGURE 48 – Le travail de [WEI et WANG, 2011] permet de généraliser la méthode classique de Fourier à l’analyse d’échantillonnages uniformes (en haut) et non-uniformes (en bas), afin de pouvoir extraire des informations pertinentes dans les deux cas. Pour chaque ligne, à gauche l’analyse est réalisée en utilisant l’équation (12) et à droite l’expression en utilisant l’équation (13). L’analyse à l’aide de la transformation de Fourier est incapable d’extraire des informations pertinentes pour l’analyse d’échantillonnages non-uniformes. Image tirée de [WEI et WANG, 2011].

et al., 2006], permettant de paramétrer une surface localement, et ainsi de se servir de cette paramétrisation pour estimer les distances entre échantillons.

Pour analyser un échantillonnage d’un domaine non-uniforme comme une surface, il est nécessaire d’avoir une représentation sous la forme d’un maillage polygonal d’une surface. De ce fait, afin de pouvoir analyser les échantillonnages générés par notre algorithme, nous avons appliqué la méthode proposée à des acquisitions synthétiques, de la même manière que dans le Chapitre 3.

Nous réalisons des échantillonnages uniformes sur l’ensemble des graphes construits à partir des acquisition synthétiques, avec différentes tailles de rayon de disques, permettant de générer des distributions à différentes densités. Pour chaque densité, nous générons un spectre de puissance issu de l’agrégation de plusieurs périodogrammes obtenus sur des échantillonnages avec des graines aléatoires différentes.

Afin d'avoir un point de repère quant à la qualité des distributions générées, nous comparons nos échantillonnages avec une méthode de l'état de l'art de l'échantillonnage en disques de Poisson de surfaces [BOWERS et al., 2010]. De cette manière, nous avons une sorte de "vérité terrain" avec laquelle nous pouvons comparer les spectres de puissance que nous avons générés.

Sur les Figures 49 et 50, nous pouvons remarquer que les distributions générées par notre approche sont très similaires à celles générées par l'approche de [BOWERS et al., 2010]. En particulier nous pouvons constater que :

- le spectre de puissance révèle des caractéristiques typiques de bruit bleu, c'est-à-dire un anneau de faible énergie au centre suivi par une forte transition, la "fréquence de coupure" à la distance $2r$ et une région de hautes fréquences plane [LAGAE et DUTRÉ, 2008],
- l'anisotropie est constante, attestant que les distributions générées n'ont pas de directions privilégiées.

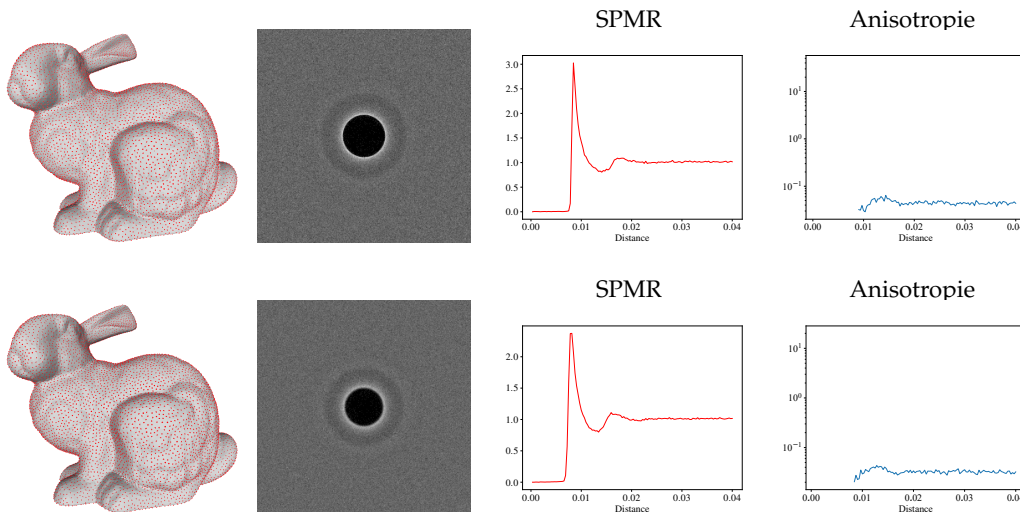


FIGURE 49 – Qualité des distributions obtenues avec [BOWERS et al., 2010] (en haut) et avec notre méthode (en bas) pour un même rayon $r = 0.004$ à partir du modèle *Bunny*.

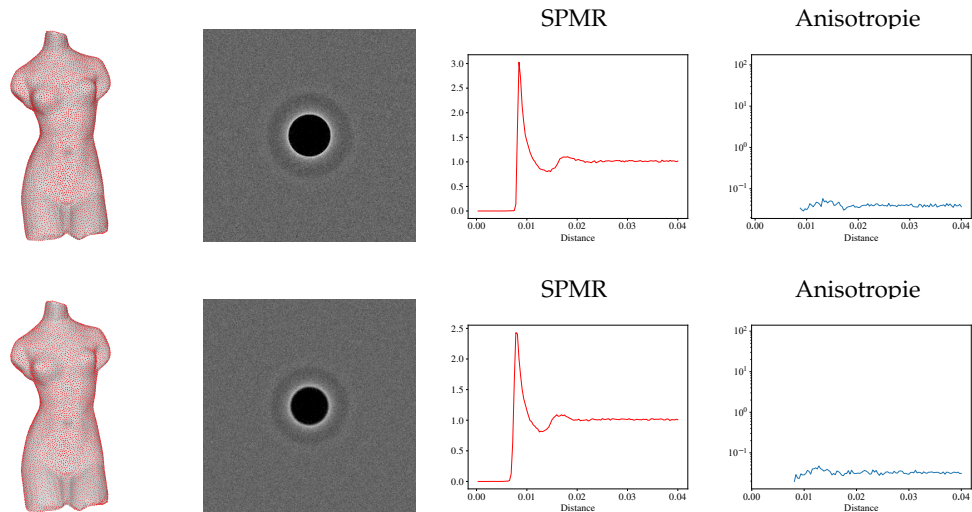


FIGURE 50 – Qualité des distributions obtenues avec [BOWERS et al., 2010] (en haut) et avec notre méthode (en bas) pour un même rayon $r = 0.004$ à partir du modèle *Venus*.

Une information que l'analyse différentielle ne nous permet pas d'analyser concerne la qualité de l'échantillonnage aux frontières entre les différents graphes. Afin de montrer que même dans ces zones l'échantillonnage est de bonne qualité, nous proposons donc de comparer visuellement un échantillonnage réalisé sur un graphe, à partir d'une carte de profondeur, et un échantillonnage de même densité réalisé sur un ensemble de graphes issus de tuiles générées à partir de cette même carte de profondeur. Nous pouvons voir dans la Figure 51 qu'il n'y a pas de différence entre les deux échantillonnages.

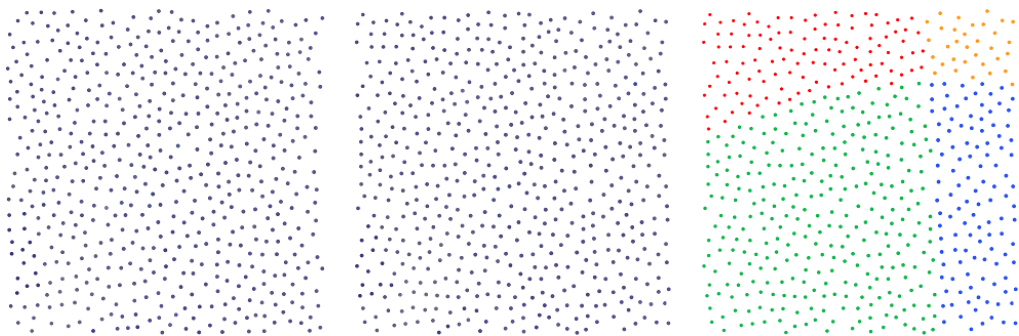


FIGURE 51 – Vue rapprochée d'échantillonnages générés sur le même morceau de surface. De gauche à droite : échantillonnage réalisé sur un unique graphe, échantillonnage réalisé sur un ensemble de graphes et le même échantillonnage en associant une couleur différente identifiant pour chaque échantillon dans quel graphe celui-ci a été sélectionné.

L'ensemble de ces résultats confirme donc que l'approche "par morceaux" proposée est capable de fournir une distribution globalement uniforme, bien que tra-

vaillant de manière séquentielle par morceaux. De plus, la qualité des distributions est comparable à une approche réalisée de manière globale sur une surface, alors que dans notre cas elle est réalisée de manière locale, sur un nuage de points.

5.2.4 Performances en temps de calcul et consommation mémoire

Nous analysons maintenant les performances de notre algorithme d'échantillonnage en termes de temps de calcul. Le but de ce travail a été de proposer une solution au problème de ré-échantillonnage de nuages de points massifs, pouvant contenir plusieurs milliards de points.

La représentation d'un nuage de points sous la forme d'un ensemble de graphes fournit une structure naturellement adaptée pour un traitement au fil de l'eau. En effet, l'Algorithme 1 est un algorithme séquentiel par nature, un seul échantillon est sélectionné puis testé à la fois. De plus, la manière de traiter les différentes acquisitions est aussi séquentielle, chaque graphe est maximalelement échantillonné avant de traiter le prochain graphe. De manière générale, les deux étapes présentées, à savoir l'échantillonnage d'un graphe et la transmission des disques de Poisson aux autres graphes, ne font intervenir à la fois que un et deux graphes respectivement. Par conséquent, il est inutile de charger l'ensemble des acquisitions en mémoire, mais au contraire, il est intéressant, et même nécessaire (en fonction des caractéristiques d'une machine et au vu de la taille des données à traiter) de charger les différents graphes progressivement du disque dur à la mémoire vive.

La Table 4 présente les résultats en particulier pour plusieurs campagnes d'acquisition ayant été ré-échantillonnées de manière uniforme pour différentes densités cibles.

Il est intéressant de noter que la consommation mémoire ne change pas en fonction de la densité cible. En effet, le coût mémoire nécessaire pour réaliser un échantillonnage dense et un échantillonnage épars est le même, *i.e.*, celui du graphe contenant le plus de sommets (issu de l'acquisition ayant la résolution la plus élevée).

5.3 GESTION DES CONTRAINTES TEMPS/MÉMOIRE

5.3.1 Échantillonnage de tuiles avec chevauchements

Comme nous l'avons montré, le coût mémoire nécessaire pour réaliser un échantillonnage ne dépend pas de la densité de l'échantillonnage cible, mais de l'acquisition la plus résolue. Par conséquent, afin de réduire le pic de consommation mémoire atteint, nous proposons d'échantillonner des acquisitions ayant été décou-

Nom	Pourcentage	Temps (h:m:s) / Pic mémoire (Gio)
<i>Templo Mayor</i>	1,2%	00:08:48 / 7,5
<i>Facade</i>	2,3%	00:22:47 / 11,6
	1,0%	00:25:36 / 11,6
	0,4%	00:29:10 / 11,6
<i>Eim Ya Kyaung</i>	0,1%	00:20:02 / 5,1
<i>Meeting House</i>	1,1%	00:43:28 / 11,4
<i>Villa Arianna</i>	1,1%	00:43:28 / 11,4
<i>Ananda Oak Kyaung</i>	2,5%	00:56:15 / 5,3
	1,3%	00:55:01 / 5,3
	0,7%	01:01:23 / 5,3
<i>Intérieur</i>	2,0%	00:38:15 / 15,9
	0,7%	00:47:22 / 15,9
	0,2%	00:56:15 / 15,9
<i>Wat Phra Si Sanphet</i>	2,8%	03:48:53 / 15,8
	1,4%	03:43:32 / 15,8
	0,7%	03:50:10 / 15,8

TABLE 4 – Temps de calcul et pic mémoire atteints par notre algorithme de ré-échantillonnage pour différentes campagnes d’acquisition à différentes densités cible. Pour chaque échantillonnage, nous indiquons le **pourcentage** de points restants par rapport au nombre de points initial. Les temps de calcul incluent le temps de lecture/écriture depuis/vers le disque dur.

pées sous la forme de tuiles avec chevauchement, tel que nous l’avions présenté dans le Chapitre 3.

Afin de mettre en avant le compromis entre le temps de calcul et le pic de consommation mémoire atteint, nous présentons dans la Table 5 des résultats en fonction de différentes tailles de tuiles.

Nous pouvons constater, que de manière similaire à la structuration de nuages de points, le pic de consommation mémoire est directement fixé par la taille de la plus grande tuile, fournissant ainsi une borne maximum au pic de consommation mémoire atteint. Nous pouvons aussi constater que l’augmentation du temps de calcul est très légère, tandis que le pic de consommation mémoire atteint est drastiquement réduit. Cette caractéristique est hautement intéressante, car elle montre que l’échantillonnage peut être réalisé sur des machines ayant peu de RAM dispo-

	Temps (h:m:s) / Pic mémoire (Gio)		
Nom (Pourcentage)	8192 ²	4096 ²	2048 ²
<i>Templo Mayor</i> (1,2%)	00:09:01 / 3,9	00:09:26 / 1,9	00:10:08 / 0,8
	+2,5% / -47,8%	+2,5% / -75,0%	+15,2% / -90,0%
<i>Facade</i> (0,4%)	00:29:58 / 6,2	00:30:20 / 2,2	00:30:24 / 0,8
	+2,7% / -46,7%	+4,0% / -81,0%	+4,2% / -93,2%
<i>Meeting House</i> (1,1%)	00:44:29 / 5,8	00:46:10 / 2,2	00:47:55 / 0,8
	+2,3% / -49,3%	+6,2% / -81,0%	+10,2% / -92,8%
<i>Intérieur</i> (0,2%)	01:00:10 / 7,8	01:00:39 / 2,2	01:02:25 / 0,8
	+7,0% / -50,8%	+7,8% / -86,1%	+11,0% / -94,8%

TABLE 5 – Temps de calcul et pic mémoire atteints par notre algorithme de ré-échantillonnage pour une même densité cible (indiqué sous la forme du **pourcentage** de points restants, à côté du **Nom**) pour différentes tailles de tuiles (8192x8192, 4096x4096 et 2048x2048). Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à leur taille native.

nible, sans pour autant que le temps de calcul ne soit fortement impacté.

Cette particularité de la méthode d'échantillonnage proposée est une conséquence directe de la nature de la représentation par graphes locaux. Quand des traitements à effectuer sont fortement locaux par nature, la versatilité de la structure permet de ne charger que les informations pertinentes au traitement actuellement effectué. De plus, quand la complexité de ces traitements est fortement dépendante du nombre de points à traiter et non du nombre d'acquisitions, l'impact sur le temps de calcul est insignifiant par rapport à la réduction du pic de consommation de mémoire.

5.3.2 Echantillonnage d'acquisitions sous-échantillonnées

Nous avons constaté que lorsque de faibles densités étaient visées, le temps de calcul de l'algorithme pouvait augmenter drastiquement. Ce problème est dû à la nature de l'algorithme de l'échantillonnage en disques de Poisson. Rappelons que dans le cas d'un échantillonnage en disques de Poisson maximal, l'algorithme se termine lorsque plus aucun élément du domaine à échantillonner ne peut être ajouté à la distribution finale. Cela peut arriver soit parce que cet élément en fait déjà partie, soit parce que celui-ci ne respecte pas le critère de distance minimum

(équation (4)). Rappelons aussi la manière dont l'algorithme élimine des échantillons candidats :

- si le critère de distance minimum est respecté, alors tous les échantillons appartenant au disque construit sont supprimés des échantillons candidats
- si le critère de distance minimum n'est pas respecté, alors seulement l'échantillon au centre du disque est supprimé des échantillons candidats

Notons enfin que plus la densité ciblée est faible plus la distance entre les échantillons sera grande, et par conséquent plus les disques de Poisson construits seront grands. Conséquemment, dans le cas d'un échantillonnage épars, le temps de construction d'un disque est plus important et la probabilité que chacun intersecte un disque déjà existant est plus grande, que dans le cas d'un échantillonnage dense. Or, nous venons de rappeler que dans le cas où le critère de distance minimum n'est pas respecté, seul l'échantillon au centre du disque construit est à éliminer. Ces particularités amènent donc l'algorithme à converger très lentement, et ainsi amènent à une augmentation considérable des temps de calcul par rapport à la génération d'échantillonnages à forte densité.

Ce constat est assez paradoxal avec le fait que les distributions visées dans ces cas sont très peu denses, car il faut beaucoup plus de temps pour obtenir un échantillonnage épars qu'un échantillonnage dense. Or, il pourrait être intéressant de pouvoir générer des distributions de faible densité beaucoup plus rapidement, afin de réduire le temps nécessaire pour obtenir de telles représentations, décrivant des scènes de manière plus grossière que des échantillonnages denses.

Dans le cas d'échantillonnages éparses, nous nous sommes donc demandés s'il était possible de réduire la quantité de points à l'origine de manière simple, c'est-à-dire représentant un faible temps de calcul, tout en ayant un impact négligeable sur la qualité de la distribution après échantillonnage. Pour se faire, nous avons donc simplement décidé de sous-échantillonner de manière régulière les cartes de profondeur comme un traitement préliminaire à la construction de la représentation par graphes locaux. L'avantage de cette approche est qu'elle permet d'obtenir des versions à plus faible résolution des acquisitions de manière extrêmement performante, car prenant avantage de la structure sous forme d'images des cartes de profondeur. De cette manière, nous pouvons construire des graphes, à partir de ces cartes de profondeur sous-échantillonnées, qui sont des versions moins résolues que les graphes que nous aurions normalement obtenus. L'échantillonnage peut, de cette manière, être réalisé de manière similaire à la méthode que nous avons présenté précédemment.

Afin de comparer les distributions obtenues à partir des graphes construits à différentes résolutions de cartes de profondeur, nous utilisons l'outil d'analyse dif-

férentielle et comparons le spectres de puissance, la SPMR ainsi que l'anisotropie de chacune des résolutions, à différentes densités d'échantillonnage (Figure 52).

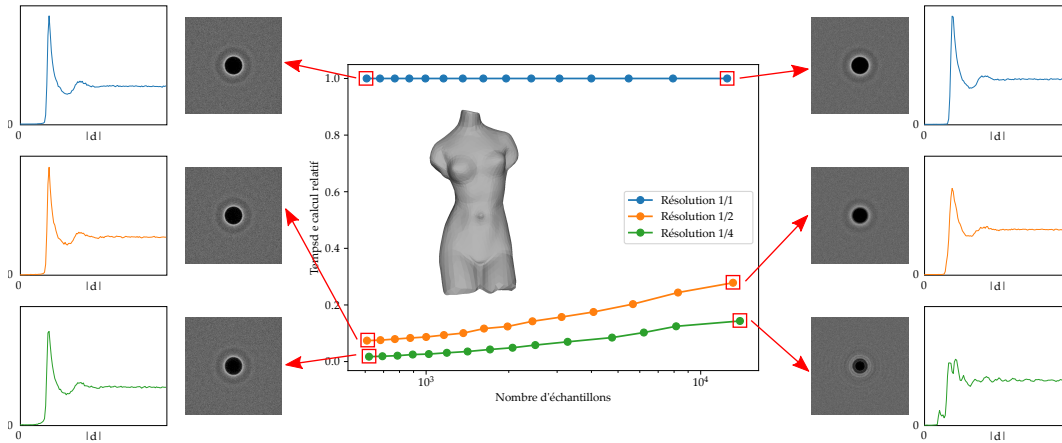


FIGURE 52 – Temps de calcul et qualité d'échantillonnage obtenus à partir de cartes de profondeur à différentes résolutions du modèle *Venus*. Les périodogrammes et SPMR sont montrés pour différentes résolutions des cartes de profondeur pour un échantillonnage de haute densité ($r = 0.004$, sur la droite) et de basse densité ($r = 0.02$, sur la gauche).

Nous pouvons remarquer que lorsque les densités ciblées sont élevées, il y a de fortes différences dans la qualité des distributions générées pour les différentes résolutions de graphes. Ceci est simplement dû au fait qu'à partir d'une certaine densité d'échantillonnage, la résolution choisie de la carte de profondeur n'offre que peu de configurations possibles respectant le rayon des disques de Poisson imposé.

Mais ce résultat était attendu et n'est en soit, que peu pertinent pour de telles densités, étant donné que les distributions générées sont de moins bonnes qualités.

Le point réellement intéressant concerne les résultats à faible densité. On peut remarquer que la qualité des distributions est quasiment similaire entre les différentes résolutions, les courbes sont presque indistinguables. Cela confirme que les distributions générées à ces densités ne nécessitent pas de travailler sur la version originale de la donnée, mais qu'une version de plus faible résolution est suffisante pour conserver quasiment les mêmes propriétés spectrales. De plus, travaillant sur des graphes ayant beaucoup moins de sommets (environ 4 fois moins à chaque niveau de résolution inférieur), le temps de calcul et le pic de consommation mémoire atteints, à la fois lors de la construction de la structure ainsi que de l'échantillonnage, sont tous les deux réduits de plusieurs ordres de grandeur (Table 6).

Cette approche comporte une limite cependant : si de fines structures sont présentes dans des cartes de profondeur, il est possible que celles-ci disparaissent lors du sous-échantillonnage de ces cartes de profondeur. Cependant, étant donné

Nom (Pourcentage)	Temps (h:m:s) / Pic mémoire (Gio)		
	Res. 1/1	Res. 1/2	Res. 1/4
Facade (0.02%)	02:01:50 / 11,6	00:14:25 / 3,0	00:01:44 / 1,1
	-/-	-88,2% / -73,9%	-98,6% / -90,4%
Intérieur (0.02%)	05:20:13 / 15,9	00:32:50 / 4,9	00:03:31 / 1,2
	-/-	-89,7% / -69,2%	-98,9% / -92,5%

TABLE 6 – Temps de calcul et pic mémoire de notre algorithme de ré-échantillonnage pour différentes résolutions de cartes de profondeur pour une faible densité cible. **Res. 1/1** indique que les cartes de profondeur sont à la résolution native, **Res. 1/2** et **Res. 1/4** indiquent que les cartes de profondeur considérées ont respectivement une résolution 2 fois et 4 fois inférieure à la résolution native. Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à la résolution native. Etant donné que la réduction de la résolution des cartes de profondeur impacte à la fois les étapes de structuration et d'échantillonnage, les performances mesurées représentent la somme de ces deux étapes.

que le but de cette approche est de réaliser des échantillonnages de basse densité, nous considérons que l'impact de la perte de ces éléments est mineur par rapport à la densité de la distribution finale. En effet, de nombreuses structures peuvent naturellement ne pas être décrites pas les distributions finales si la densité cible est trop faible. C'est pour cela qu'en pratique cette disparition potentielle de fines structures, lors de la génération des différents niveaux de résolution des cartes de profondeur ne pose pas vraiment problème.

5.4 APPLICATION À LA RECONSTRUCTION DE SURFACES

Nous présentons maintenant une des applications directes de notre algorithme de ré-échantillonnage. Dans le cadre de la reconstruction de surfaces, comme nous l'avons présenté précédemment, beaucoup d'algorithmes ont du mal à passer à l'échelle, et quand ceux-ci traitent de grandes données, cela reste de l'ordre de la centaine de millions de points maximum sur des ordinateurs classiques.

L'algorithme que nous avons proposé permet, en fixant la densité d'échantillonnage cible, de réduire le nombre de points de façon à pouvoir utiliser des algorithmes de reconstruction de surface de l'état de l'art, peu importe la taille du nuage de points original.

Une des particularités de notre méthode concerne les propriétés de bruit bleu qu'ont les distributions générées. Grâce à la combinaison d'un échantillonnage en

disques de Poisson avec des graphes locaux représentant la connectivité de la surface numérisée, les distributions générées sont constituées d'échantillons répartis uniformément sur la surface numérisée. De ce fait, et par les performances de leur approche, nous nous sommes intéressés à la méthode de reconstruction de surface des auteurs de [BOLTCHÉVA et LEVY, 2017]. Cette méthode a la particularité de créer un maillage triangulaire à partir d'un nuage de points dont les sommets du maillage sont les points du nuage de points donné en entrée. Par conséquent, étant donné que nos distributions répartissent les échantillons uniformément sur la surface, une triangulation dont les sommets sont les points du nuage de points et qui forment des triangles entre des sommets géométriquement proches, créera des triangles ayant un bon aspect ratio en général et des aires quasiment constantes. De plus, cette méthode montre de très bons résultats en termes de reconstruction, à la fois au niveau de la qualité des reconstructions, ainsi qu'en matière d'efficacité (leur algorithme étant hautement parallèle), leur permettant de traiter des nuages de points jusqu'à quelques centaines de millions de points.

Notre but a donc été de réduire suffisamment la quantité de points pour que l'outil de [BOLTCHÉVA et LEVY, 2017] puisse être appliqué sur notre machine, et que les échantillons choisis soient bien distribués sur la surface afin de produire des triangulations de bonne qualité.

Dans un premier temps, la Figure 53 présente une reconstruction obtenue à partir d'un échantillonnage adaptatif d'un nuage de points.

Nous montrons maintenant des résultats de reconstruction obtenus à partir d'échantillonnages générés, en termes de conservation de détails (Figures 54 et 55) et de qualité de la triangulation (Figure 56). L'ensemble de ces résultats nous confirme que la méthode d'échantillonnage proposée offre des résultats plutôt satisfaisants, permettant la manipulation de nuages de points massifs par un algorithme de l'état de l'art de la reconstruction de surfaces, et permettant d'obtenir des reconstructions très satisfaisantes à partir de nuages de points composés de plusieurs milliards de points.

5.5 CONCLUSION DU CHAPITRE

Ce chapitre a introduit une nouvelle méthode pour le ré-échantillonnage de nuages de points massifs issus d'acquisitions structurées.

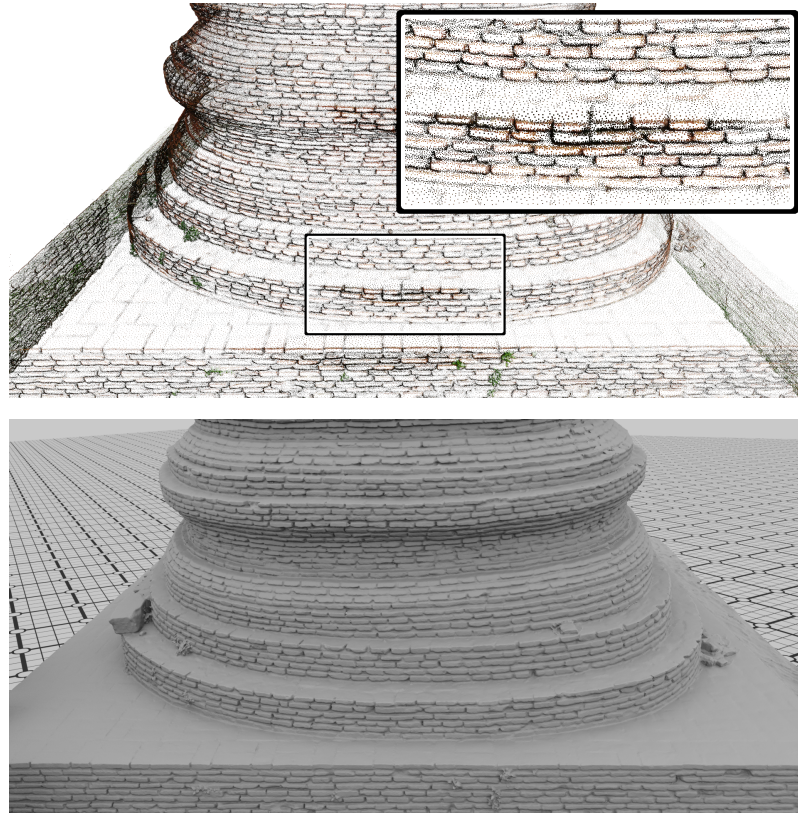


FIGURE 53 – Echantillonnage adaptatif obtenu par notre approche (en haut) et surface générée à partir de cet échantillonnage en utilisant l’approche de [BOLTCHÉVA et LEVY, 2017] (en bas).

5.5.1 Contributions

A partir de la structure présentée dans le Chapitre 3, nous avons proposé une méthode pour **ré-échantillonner un nuage de points** basé sur un algorithme d’échantillonnage en disques de Poisson. Les particularités de cet échantillonnage et l’utilisation d’une métrique géodésique nous permettent d’obtenir des distributions répartissant les échantillons de manière uniforme sur la surface échantillonnée et fournissant des garanties théoriques concernant la couverture du nuage de points original par la version sous-échantillonnée. De plus, nous avons présenté une manière de modifier la métrique géodésique, en exploitant une information concernant la courbure de la surface, pour **générer des distributions adaptatives**, échantillonnant plus densément les zones de fortes courbures.

La particularité de la structure des données fournie par les acquisitions issues de LiDARs terrestres permet d’**utiliser des opérateurs issus du traitement des images** (filtrage bilatéral) ayant un impact sur la géométrie du nuage de points (par invariance de l’association entre une carte de profondeur et le nuage de points

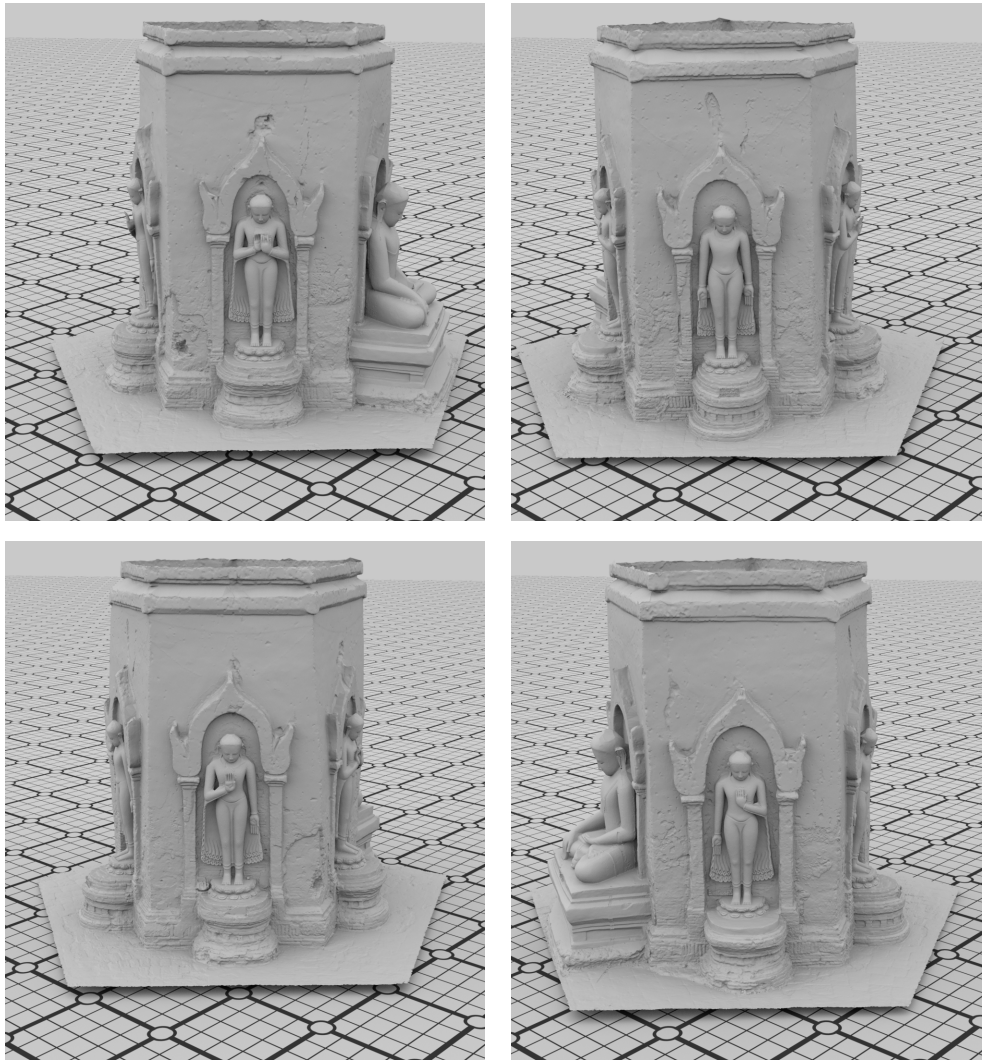


FIGURE 54 – Différentes vues d’une reconstruction obtenue à partir d’un échantillonnage adaptatif montrant la conservation des fins détails de la surface (*Eim Ya Kyaung*).

qui lui est associé, grâce à l’utilisation des fonctions de projection). Ces opérateurs basés image ont l’avantage d’avoir une complexité faible ainsi que de prendre en compte une connectivité proche de la connectivité de la surface numérisée par rapport à des opérateurs spatiaux. Ceci leur permet d’atténuer fortement le bruit d’acquisition pouvant être présent sans trop atténuer l’information contenue dans le nuage de points original.

Les distributions générées montrent de **bonnes propriétés de bruit bleu** et sont comparables à un algorithme de l’état de l’art de l’échantillonnage de surfaces 3D.

En travaillant de manière séquentielle sur chaque graphe local, l’algorithme proposé est capable de traiter n’importe quelle donnée, peu importe le nombre de points, ou le nombre d’acquisitions fournies en entrée. En transmettant les distri-



FIGURE 55 – Comparaison d’une photo et de la reconstruction obtenue à partir d’un échantillonnage adaptatif généré par notre approche sur *Facade*.

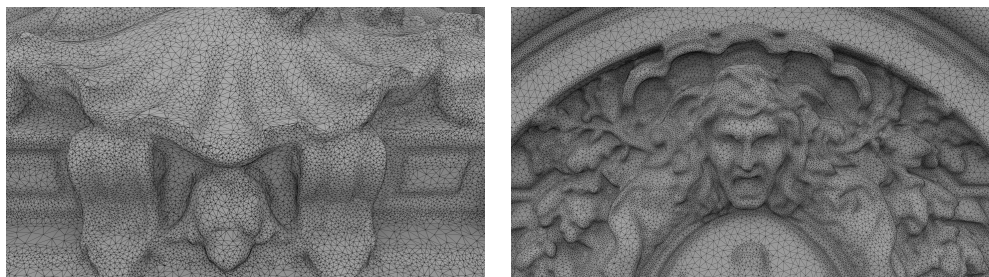


FIGURE 56 – Reconstruction obtenue à partir d’un échantillonnage adaptatif généré par notre approche sur *Facade*.

butions d’échantillons entre les graphes, la **cohérence globale de la distribution générée** est assurée, permettant d’obtenir un échantillonnage non seulement localement uniforme dans chaque graphe (grâce aux conditions de l’échantillonnage en disques de Poisson) mais aussi globalement uniforme sur la surface sous-jacente

au nuage de points global.

Afin d'améliorer drastiquement les temps de calcul et le pic de consommation mémoire pour les distributions à faible densité, nous avons proposé de **sous-échantillonner les acquisitions *a priori***. Nous avons montré que les distributions ainsi générées avaient une qualité similaire aux distributions générées sur les acquisitions à pleine résolution pour de temps de calcul fortement réduits.

Finalement, nous avons montré que la réduction du nombre de points et la répartition des échantillons sélectionnés permettent directement d'utiliser les distributions générées comme **donnée d'entrée pour un algorithme de reconstruction de surfaces** de l'état de l'art, générant des triangulations détaillées et de bonne qualité en termes d'aspect/ratio.

5.5.2 Discussion

Il est important de noter que nous avons fait le choix, dans l'approche proposée, de sélectionner comme échantillons des distributions finales des points existants dans le nuage de points d'origine. L'impact de ce choix peut réduire la qualité des distributions générées, étant donné que la finesse du positionnement des échantillons sélectionnés dans une région est directement liée à la résolution de l'acquisition ayant le plus finement décrit cette région. En particulier, les échantillonnages de haute densité générées peuvent avoir une qualité biaisée par la résolution des acquisitions, et ainsi potentiellement réduire la qualité de la distribution finale.

Nous avons proposé une approche permettant de générer des distributions de bonne qualité à faible densité dans des temps drastiquement réduits et avec des pics de consommation mémoire très peu élevés en sous-échantillonnant les cartes de profondeur d'origine avant la construction des graphes. Aujourd'hui nous n'avons aucune manière de connaître, par avance, la résolution minimum suffisante d'un ensemble de cartes de profondeur en fonction d'une densité cible. Il serait intéressant par la suite de pouvoir déterminer, pour une densité cible, la résolution minimum des cartes de profondeur permettant de générer des distributions ayant des bonnes qualités spectrales.

Quatrième partie

DIAGRAMME DE VORONOÏ DE NUAGES DE POINTS

INTRODUCTION

Ce chapitre présente les notions générales qui seront abordées au sein de cette partie. Nous commençons par établir quelques définitions liées aux diagrammes de Voronoï de manière générale, pour ensuite introduire les diagrammes de Voronoï centroïdaux.

6.1 DIAGRAMMES DE VORONOÏ

Les diagrammes de Voronoï sont sûrement l'un des outils mathématiques les plus utilisés. Un diagramme de Voronoï représente la partition d'un plan à partir d'un ensemble de points de façon à ce que chaque région du plan soit associée au sommet le plus proche de celle-ci, par rapport à la métrique considérée. Cet outil, si "simple" à définir, est pourtant la solution à de nombreux problèmes calculatoires. Il est intéressant de noter que ces diagrammes apparaissent de manière naturelle dans le monde qui nous entoure (Figure 57).



FIGURE 57 – Exemples de diagrammes de Voronoï présents dans la nature. A droite, les alvéoles d'un nid d'abeille. Image tirée de [MACKINTOSH, 2011]. A gauche, les ailes d'une libellule. Image tirée de [NONENMACHER, 2003].

C'est au 17^e siècle que l'on peut trouver la première représentation visuelle d'un diagramme de Voronoï. Dans son livre sur les principes de philosophie [DESCARTES, 1644], Descartes affirme que l'espace est constitué de tourbillons de matière, orbitant autour des étoiles. Bien que n'ayant pas défini explicitement la manière dont ces régions sont constituées, une illustration montre une décomposition de l'espace en régions convexes (Figure 58).

L'idée sous-jacente semble être que, dans un espace E un ensemble de sites S exerce une influence sur les points p de E . Ainsi la région d'influence d'un site

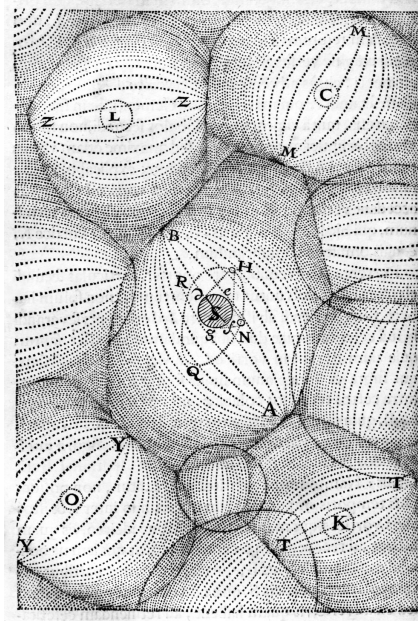


FIGURE 58 – Illustration de la décomposition de l’espace de Descartes en tourbillons de matière. Image tirée de [DESCARTES, 1644].

$s \in S$ correspond à l’ensemble des points p pour lesquels l’influence de s est la plus forte, parmi tous les s de S .

Notons tout de même que cet outil a été découvert dans différents domaines de manière indépendante. C’est pour cela qu’il a reçu différents noms en fonction des disciplines l’ayant découvert : *transformée de l’axe médian* en biologie et physiologie, *zones de Wigner-Seitz* en chimie et physique, *domaines d’action* en cristallographie ou encore *polygones de Thiessen* en météorologie et géographie. Finalement, les premiers travaux à introduire de manière formelle ce concept sont ceux des mathématiciens Dirichlet [LEJEUNE DIRICHLET, 1850] et Voronoï [VORONOÏ, 1908] qui ont donné lieu aux *tesselations de Dirichlet* et *diagrammes de Voronoï*.

Considérons les notations suivantes pour la suite de cette partie :

- Soit (M, d) un espace métrique convexe, défini par un ensemble convexe M muni d’une métrique d .
- Soit $\mathcal{B}'(p, r)$ la boule fermée centrée en un point $p \in M$ et de rayon $r \in \mathbb{R}$ définie par

$$\mathcal{B}'(p, r) = \{x \in M \mid d(x, p) \leq r\}.$$

- Soit $\mathcal{B}(p, r)$ la boule ouverte centrée en un point $p \in M$ et de rayon $r \in \mathbb{R}$ définie par

$$\mathcal{B}(p, r) = \{x \in M \mid d(x, p) < r\}.$$

- Soit $\delta B'(p, r)$ le bord de la boule centrée en un point $p \in M$ et de rayon $r \in \mathbb{R}$ définie par

$$\delta B'(p, r) = \{x \in M \mid d(x, p) = r\}.$$

- $A \subseteq B$ signifie que A est un sous-ensemble de B .
- $A \subsetneq B$ signifie que A est un sous-ensemble propre de B , c'est-à-dire que $B \setminus A \neq \emptyset$.
- Soit $x \in M$ et $M' \subseteq M$:

$$d(x, M') = \min_{x' \in M'} d(x, x')$$

Le diagramme de Voronoï d'un espace métrique (M, d) d'un ensemble de sites $s_i \in S \subseteq M$ est défini comme :

$$\text{Vor}^S = \cup_{s_i \in S} C_i^S, \quad (14)$$

où chaque cellule de Voronoï C_i^S représente l'ensemble des points $x \in M$ plus proche de s_i que de n'importe quel autre site $s_j \forall s_j \in S, s_j \neq s_i$:

$$C_i^S = \{x \in M \mid d(x, s_i) < d(x, s_j), \forall s_j \in S, s_j \neq s_i\}. \quad (15)$$

Pour des raisons de simplification d'écriture, lorsqu'il n'y aura pas d'ambiguïté sur l'ensemble de sites sur lequel le diagramme de Voronoï est construit, nous nous permettrons d'écrire :

- $\text{Vor} \stackrel{\text{def.}}{=} \text{Vor}^S,$
- $C_i \stackrel{\text{def.}}{=} C_i^S.$

Un des grands intérêts du diagramme de Voronoï concerne la dualité de celui-ci avec la *triangulation de Delaunay*. En effet, les sommets du diagramme de Voronoï sont les centres des cercles circonscrits des triangles de la triangulation de Delaunay et les arêtes du diagramme de Voronoï sont sur les médiatrices des arêtes de la triangulation de Delaunay.

De ce fait, il est possible de passer d'un diagramme de Voronoï à une triangulation de Delaunay, et vice-versa, en inversant les *dimensions* des faces constituant chacune des représentations. Les sommets du diagramme de Voronoï (faces de dimension 0) deviennent des triangles de la triangulation de Delaunay (faces de dimension 2) et les cellules du diagramme de Voronoï (faces de dimension 2) deviennent les sommets de la triangulation de Delaunay (faces de dimension 0) (Figure 59).

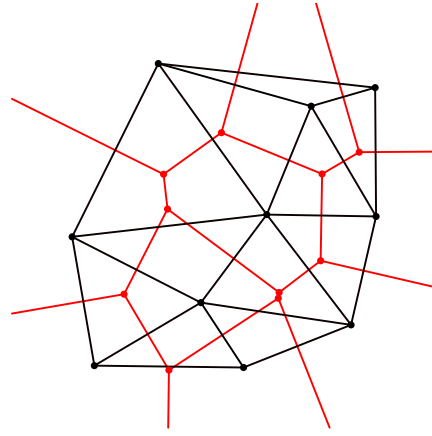


FIGURE 59 – Schéma de la dualité entre le diagramme de Voronoï (en rouge) et la triangulation de Delaunay (en noir). Image tirée de [HFEREE, 2011].

6.1.1 Diagrammes de Voronoï centroïdaux

Une fois un diagramme de Voronoï construit, le volume et la forme des cellules peut fortement varier d'une cellule à une autre, en fonction de la position des différents sites. Des cellules adjacentes dont les aires ou formes sont très différentes amènent à des triangulations de Delaunay où les angles des triangles adjacents varient fortement (Figure 60)

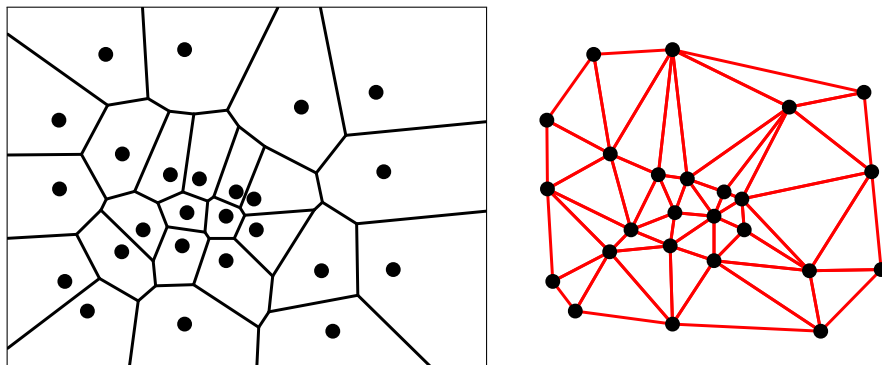


FIGURE 60 – Diagramme de Voronoï à partir d'une position arbitraire de sites (en noir) et triangulation de Delaunay duale associée (en rouge). La non uniformité de la distribution des sites crée des cellules ayant des formes et des aires très différentes et la triangulation associée est composée de triangles ayant des aspect/ratio très disparates.

Plusieurs critères peuvent être pris en compte pour juger de la qualité et de la régularité d'une triangulation. La forme de chaque triangle, la variation de la forme des triangles dans un voisinage donné ou encore la *valence* de chaque sommet (le nombre de sommets adjacents à un sommet) en sont quelques exemples. De manière idéale, chaque triangle devrait être *équilatéral*, des triangles adjacents

doivent avoir des *formes similaires* et chaque sommet doit avoir une *valence égale à 6*. Pour le diagramme de Voronoï, ces propriétés se traduisent par une *uniformité dans le placement des sites*, amenant à des *cellules de surface égale*. Un tel diagramme est appelé *Diagramme de Voronoï centroïdal*, car dans ce cas particulier, chaque site de Voronoï est confondu avec le centroïde de sa cellule [AURENHAMMER, 1991].

6.1.2 Relaxation de Lloyd

Placer les sites de Voronoï de manière à ce qu'ils coïncident avec le centroïde de leur cellule une fois le diagramme de Voronoï calculé n'est pas une étape triviale en soit. Il existe donc de nombreuses manières itératives de modifier la position de sites de Voronoï afin d'arriver à un diagramme de Voronoï centroïdal.

En particulier, la *relaxation de Lloyd* [LLOYD, 1982] est une méthode populaire pour l'obtention de diagrammes de Voronoï centroïdaux. Il s'agit d'un algorithme itératif qui consiste en deux étapes. A partir d'un ensemble de sites S placés de manière arbitraire dans un espace métrique (M, d) :

- un diagramme de Voronoï est construit à partir des sites S sur le domaine M ,
- chaque site $s_i \in S$ est déplacé sur le centroïde de la cellule C_i qu'il vient de générer.

La répétition de ces deux étapes minimise la fonctionnelle suivante :

$$\sum_{i=1}^n \int_{x \in C_i} \rho(x) d(s_i, x) dx, \quad (16)$$

où ρ est une densité de probabilité. Le travail de [LLOYD, 1982] a montré que cette fonctionnelle est minimisée lorsque chaque site correspond au centroïde de la cellule que celui-ci a généré.

Un point intéressant en lien avec la dualité diagramme de Voronoï/triangulation de Delaunay, c'est que plus un diagramme de Voronoï est "proche" d'une configuration "centroïdale", plus la triangulation duale est proche d'être régulière (sommets ayant tous une valence de 6) et composée de triangles équilatéraux (Figure 61).

6.2 OBJECTIFS DE CETTE PARTIE

Dans cette partie, nous nous intéressons au problème de la construction d'un diagramme de Voronoï sur la surface sous-jacente à un nuage de points massifs. Dans un premier temps, nous présentons quelques travaux théoriques sur la localité des partitions de Voronoï. Dans un second temps, nous appliquons ces travaux à la construction d'un diagramme de Voronoï centroïdal sur un ensemble de graphes, en utilisant la structure introduite dans le Chapitre 3.

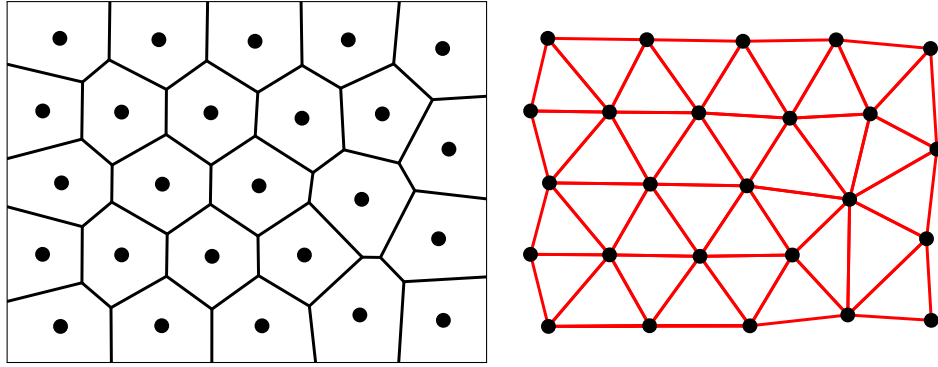


FIGURE 61 – Diagramme de Voronoï centroïdal (en noir) obtenu après convergence de l’algorithme de Lloyd en considérant la norme L_2 dans le plan et triangulation de Delaunay duale associée (en rouge).

L’objectif de cette partie est double. Dans un premier temps, nous souhaitons améliorer la qualité des distributions de points générées à l’aide de la méthode présentée dans le Chapitre 5, en déplaçant itérativement les échantillons des distributions à l’aide d’un algorithme de relaxation. Dans un second temps, nous voulons évaluer la qualité des triangulations obtenues à partir de la représentation duale des diagrammes de Voronoï construits par rapport à une méthode de l’état de l’art de reconstruction de surfaces.

De la même manière que les parties précédentes, la réalisation de ces objectifs se fait sous des contraintes de pic de consommation mémoire.

DIAGRAMME DE VORONOÏ DÉFINI PAR MORCEAUX

Ce chapitre présente nos recherches théoriques concernant la localité des partitions de Voronoï. Nous y présentons les notions de *bassin de Voronoï* et *configuration locale d'un site*, telles qu'introduites par le travail de [KÜHN, 1998]. Kühn s'est intéressé à déterminer le diagramme de Voronoï d'une région particulière du domaine, en étudiant les sites nécessaires pour calculer correctement le diagramme au sein de cette région. Dans notre cas, nous nous intéressons au problème inverse. Etant donné un sous-espace convexe $M' \subseteq M$ et un sous-ensemble de sites $S' = S \cap M'$, nous cherchons à déterminer la région dans laquelle la partition construite à partir de M' et de S' est un sous-ensemble de la partition obtenue dans le domaine M .

Dans notre contexte de numérisation 3D, une telle information nous permettra de construire le diagramme de Voronoï d'un nuage de points massif, en travaillant à partir des graphes locaux présentés dans le Chapitre 3.

Pour cela, nous démontrons tout d'abord qu'à partir de la configuration locale d'un site, il est possible de déterminer si une cellule est correctement calculée dans un sous-espace convexe $M' \subseteq M$. Ensuite, nous démontrons que connaissant un sous-espace convexe $M' \subseteq M$ et un sous-ensemble des sites $S' = S \cap M'$, il est possible de construire correctement une partie du diagramme de Voronoï global, normalement construit sur le domaine M à partir de l'ensemble des sites S .

7.1 BASSIN ET CONFIGURATION LOCALE D'UN SITE

Il est évident de constater la localité inhérente de la partition de Voronoï. En effet, pour un site $s_i \in S$, la zone de l'espace sous son influence est *entièrement* et *uniquement* déterminée par un sous-ensemble $S' \subseteq S$ des sites du domaine. Il est donc possible de connaître la forme de sa cellule C_i uniquement à partir d'un sous-ensemble des sites $S' \subseteq S$ de l'espace M .

En d'autres termes, la position de tous les autres sites $\bar{S} = \{S \setminus S'\}$ n'influe pas sur la définition de la cellule C (Figure 62) :

$$C_i = \bigcap_{s_k \in S', k \neq i} D(s_i, s_k) = \bigcap_{s_j \in S, j \neq i} D(s_i, s_j), \quad (17)$$

où $D(s_i, s_j)$ représente le demi-espace plus proche de s_i que de s_j :

$$D(s_i, s_j) = \{x \in M \mid d(s_i, x) \leq d(s_j, x)\}. \quad (18)$$

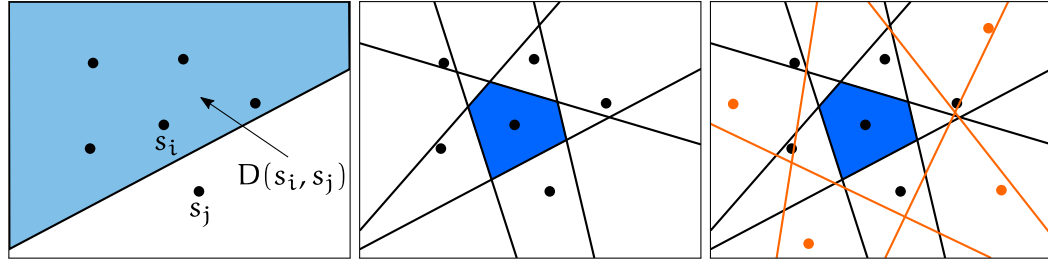


FIGURE 62 – A gauche, le demi-espace $D(s_i, s_j)$ (en bleu clair) est construit à partir des sites s_i et s_j . Au centre, la cellule C_i (en bleu) est construite à partir de l'intersection de tous les demi-espaces $D(s_i, s_k), \forall s_k \in S', k \neq i$ d'un sous-ensemble de sites $S' \subseteq S$. A droite, la cellule C_i est construite de la même manière, mais en considérant aussi les sites $\{S \setminus S'\}$ (en orange). Ces sites n'affectent donc pas la définition de la cellule C_i .

Si ce constat est assez trivial, la définition de la zone en dehors de laquelle la position de tous les sites \bar{S} n'a aucune influence sur la cellule C demande un peu plus de réflexion. La caractérisation de cette zone a comme but de déterminer les cellules de Voronoï qui sont définies de la même manière dans un sous-espace M' que dans l'espace M . Si tel est le cas, alors nous dirons que la cellule C est *globalement définie* dans M' .

Rappelons, dans un premier temps, qu'un sommet v d'un diagramme de Voronoï correspond à l'intersection d'au moins 3 cellules de Voronoï (plus dans le cas où plus de 3 sites sont co-circulaires). Soit la boule fermée $\mathcal{B}'(v, r_v)$ centrée en v et de rayon $r_v = \min_{s_i \in S} d(s_i, v)$ représentant la distance minimale entre v et tous les sites de S . Par définition, $\forall s_i \in S, s_i \notin \mathcal{B}^S(v, r_v)$, et $\{S \cap \delta \mathcal{B}'(v, r_v)\}$ correspond à l'ensemble des sites à l'origine de l'intersection v . Appelons aussi $\mathcal{B}'^S(v, r_v)$, la *boule fermée associée au sommet v* .

Suivant les notations de [KÜHN, 1998], définissons :

- $\text{VB}_i^S \subseteq M$, la *bassin de Voronoï du site $s_i \in S$* comme étant l'union des boules fermées \mathcal{B}'_v^S (Figure 63) des sommets v constituant sa cellule C_i :

$$\text{VB}_i^S = \cup_{v \in C_i} \mathcal{B}'^S(v, r_v), \quad (19)$$

- $\text{CL}_i^S \in S$, la *configuration locale du site s_i* comme étant l'ensemble des sites appartenant au bassin de Voronoï de ce site :

$$\text{CL}_i^S = S \cap \text{VB}_i^S. \quad (20)$$

Pour des raisons de simplification d'écriture, lorsqu'il n'y aura pas d'ambiguïté sur l'ensemble des sites sur lequel le diagramme de Voronoï est construit, nous nous permettrons d'écrire :

- $\text{VB}_i \stackrel{\text{def.}}{=} \text{VB}_i^S$,

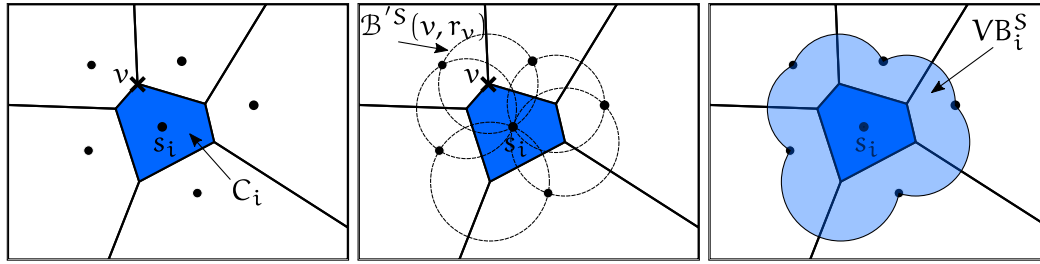


FIGURE 63 – De gauche à droite : partition de Voronoï à partir de 6 sites ; bord de chaque boule $\delta B'_v{}^S$ associées aux sommets v constituant la cellule bleue C_i ; bassin de Voronoï VB_i^S issu de l'union des boules $B'_v{}^S$.

- $CL_i \stackrel{\text{def.}}{=} CL_i^S$.

D'après les définitions précédentes, nous pouvons constater que $VB_i^{CL} = VB_i^S, \forall i$. Nous pouvons donc en conclure que la cellule de Voronoï $C_i^{CL_i}$ construite à partir de la configuration locale CL_i du site $s_i \in S$ est égale à la cellule de Voronoï C_i^S construite à partir de l'ensemble des sites S :

$$C_i^{CL_i} = C_i^S, \forall i. \tag{21}$$

La Figure 64 illustre ces propos. La cellule bleue reste définie de la même manière, peu importe le placement des sites et leur nombre en dehors de son bassin de Voronoï. Cependant, il suffit d'un seul site rajouté à l'intérieur de son bassin de Voronoï pour que celui-ci impacte la définition de sa cellule.

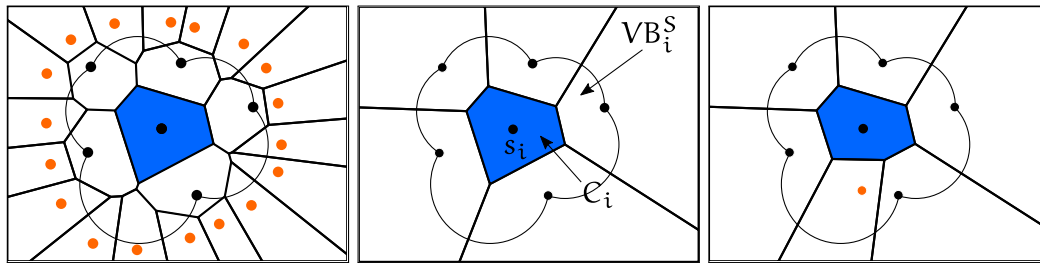


FIGURE 64 – Au centre, une partition de Voronoï construite à partir de 6 sites et bassin de Voronoï de la cellule C_i (en bleu). A gauche, la cellule C_i reste définie de la même manière, peu importe le placement des sites et leur nombre en dehors de son bassin de Voronoï VB_i . A droite, il suffit d'un seul site (en orange) rajouté à l'intérieur de son bassin de Voronoï VB_i pour que celui-ci impacte la définition de sa cellule C_i .

Montrons maintenant que lorsque le bassin de Voronoï VB_i d'un site $s_i \in S$ est entièrement inclus dans M' , alors la cellule de Voronoï C_i de ce site est globalement définie dans M' .

Lemme 7.1.1. Soient $M' \subseteq M, S' = S \cap M'$ et $s_i \in S'$.

Si $VB_i \subseteq M'$ alors $C_i^{S'} = C_i^S$.

Démonstration. Par définition, comme $VB_i \subseteq M'$, cela signifie que $CL_i \subseteq S'$.

D'après l'équation (21), $C_i^{CL_i} = C_i^{S'}$ et $C_i^{CL_i} = C_i^S$.

Donc $C_i^{S'} = C_i^S$. □

Depuis le Lemme 7.1.1, on peut en déduire une extension directe :

Lemme 7.1.2. *Si $\cup_{s_i \in S'} VB_i \subseteq M'$ alors $\cup_{s_i \in S'} C_i^{S'} = \cup_{s_i \in S'} C_i$.*

On peut donc conclure que l'ensemble des cellules dont le bassin de Voronoï est entièrement inclus dans M' sont globalement définies dans M' .

7.2 DIAGRAMME DE VORONOÏ GLOBALEMENT COHÉRENT

Il est maintenant possible d'identifier l'ensemble des cellules de Voronoï étant globalement définies dans un sous-espace M' en ayant uniquement connaissance des sites appartenant à ce sous-espace. Cependant, nous avons remarqué qu'il était possible de construire une plus grande région du diagramme de Voronoï au sein de M' qui soit elle aussi globalement définie, c'est-à-dire que la partition réalisée dans cette région soit similaire à celle obtenue à partir d'un diagramme de Voronoï calculé sur l'ensemble du domaine M . Nous nous sommes donc demandés comment il était possible de connaître, uniquement à partir d'un sous-espace M' et d'un ensemble de sites $S' = S \cap M'$, cette région \widehat{Vor} telle que $\widehat{Vor} \subseteq Vor$.

Pour cela nous avons décidé de nous inspirer de la méthode [FORTUNE, 1987]. Dans cet article, l'auteur propose de construire un diagramme de Voronoï en résolvant le problème progressivement, à l'aide d'une ligne de balayage. Cette ligne parcourt le domaine d'un bord à l'autre, faisant croître la zone du domaine déjà partitionnée. A chaque instant, elle permet de définir un diagramme de Voronoï *partiel* sur le domaine déjà balayé (Figure 65).

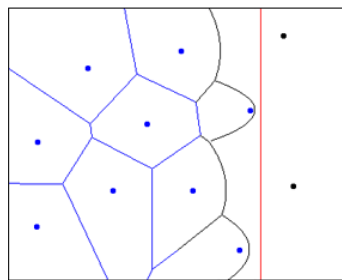


FIGURE 65 – La ligne de balayage (en rouge), permet de définir à chaque instant un diagramme de Voronoï partiel (en bleu et noir) sur le domaine déjà balayé.

Plus précisément, ce diagramme de Voronoï partiel correspond à la région du domaine étant plus proche des sites du domaine balayé que de la ligne de balayage elle-même (au sens de la métrique d associée). La zone représentée par le diagramme de Voronoï partiel est en fait un sous-ensemble du domaine global.

Dans notre cas, l'idée est similaire concernant l'obtention d'un diagramme de Voronoï partiel. En ayant uniquement connaissance du sous-espace M' (dans notre cadre, un graphe $G_i \in G$) il n'est pas possible de connaître la position des sites \hat{S} en dehors de ce domaine (dans le reste des graphes $G_j \in G, \forall j \neq i$). Nous nous servons du bord $\delta M'$ du sous-espace M' comme d'un équivalent à cette ligne de balayage. Nous allons donc construire le diagramme de Voronoï partiel, c'est-à-dire la région qui est plus proche des sites S' du domaine M' que du bord du domaine $\delta M'$.

Soit $S_{\delta M'} = \{x | x \in \delta M'\}$ l'ensemble de sites représentant l'ensemble des points sur le bord du domaine M' . Soit $\hat{S} = S' \cup S_{\delta M'}$ l'ensemble nous permettant de définir un morceau du domaine \widehat{Vor} tel quel $\widehat{Vor} \stackrel{\text{def.}}{=} \cup_{s_i \in S'} C_i^{\hat{S}}$. Nous souhaitons maintenant montrer que la région \widehat{Vor} est globalement définie dans M' , i.e., $\widehat{Vor} \subseteq Vor$.

Premièrement, montrons que l'union des bassins de Voronoï des cellules $VB_i^{\hat{S}}$ est entièrement inclus dans M' .

Lemme 7.2.1. $\cup_{s_i \in S'} VB_i^{\hat{S}} \subseteq M'$.

Démonstration. Raisonnons par l'absurde et supposons que $\cup_{s_i \in S'} VB_i^{\hat{S}} \not\subseteq M'$.

Cela signifie que, pour au moins une boule $B^{\hat{S}}(v, r_v) \subseteq VB_i^{\hat{S}}$ il existe au moins un point $x \in B^{\hat{S}}(v, r_v)$ tel que $x \in M \setminus M'$.

Par construction, $B^{\hat{S}}(v, r_v)$ est un ensemble connexe. Il existe donc $x' \in \{B^{\hat{S}}(v, r_v) \cap \delta M'\} \neq \emptyset$.

Par définition $x' \in \delta M'$, et par construction $\forall s' \in \hat{S}, s' \notin B^{\hat{S}}(v, r_v)$.

On arrive donc à une contradiction : $x' \notin B^{\hat{S}}(v, r_v)$, et donc $B^{\hat{S}}(v, r_v) \cap \delta M' = \emptyset$.

Par conséquent $\cup_{s_i \in S'} VB_i^{\hat{S}} \subseteq M'$. □

Nous nous servons maintenant de ce résultat pour montrer que la région \widehat{Vor} est globalement définie dans M' et que par conséquent, $\widehat{Vor} \subseteq Vor$. Notons tout d'abord que, par définition, il n'existe aucun point $p \in M \setminus M'$ étant plus proche de n'importe quel point de M' que ceux situés sur $\delta M'$:

$$\forall p \in M \setminus M', \forall p' \in M', \forall \hat{p} \in \delta M', d(p, p') > d(\hat{p}, p'). \tag{22}$$

Définissons $d(p, S)$ la distance du point p au plus proche point $x \in S$, tel que $d(p, S) = \min_{s_i \in S} d(p, s_i)$.

Lemme 7.2.2. $\widehat{Vor} \subseteq \cup_{s_i \in S'} C_i^S$.

Démonstration. Raisonnons par l'absurde et supposons que $\widehat{Vor} \not\subseteq \cup_{s_i \in S'} C_i^S$.

Dans ce cas, il existe $x \in M$ tel que $x \in \widehat{Vor}$ et $x \notin \cup_{s_i \in S'} C_i^S$.

Autrement dit, $d(x, S') < d(x, S_{\delta M'})$ et $d(x, \hat{S}) < d(x, S')$.

On en déduit $d(x, \widehat{S}) < d(x, S_{\delta M'})$.

Par ailleurs, $\widehat{\text{Vor}} \subsetneq \cup_{s_i \in S'} \text{VB}_i^{\widehat{S}}$. D'après le Lemme 7.2.1, $\widehat{\text{Vor}} \subsetneq M'$, donc $x \in M'$.

Or, d'après l'équation (22) : $\forall x' \in M', d(x', S_{\delta M'}) < d(x', \widehat{S})$.

On arrive donc à une contradiction. Par conséquent : $\widehat{\text{Vor}} \subseteq \cup_{s_i \in S'} C_i^S$. □

Le Lemme 7.2.2 nous permet donc, sans connaître le domaine M ni l'ensemble des sites S , de déterminer un morceau $\widehat{\text{Vor}}$ de la partition de Voronoï globale Vor à partir d'un domaine $M' \subseteq M$ et d'un ensemble de sites $S' = S \cap M'$ (Figure 66).

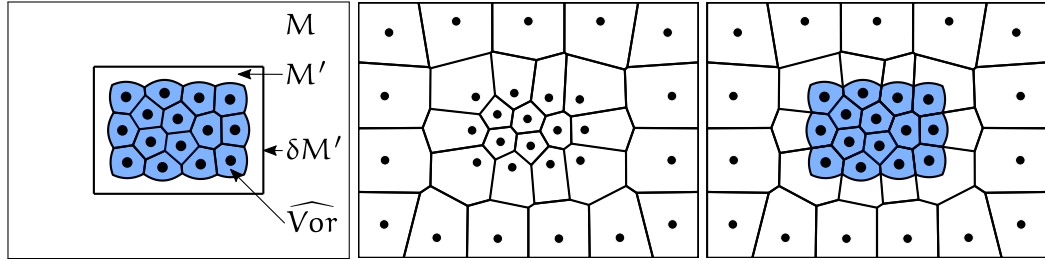


FIGURE 66 – De gauche à droite, morceau du diagramme de Voronoï global $\widehat{\text{Vor}}$ défini uniquement à l'aide des sites $S' = S \cap M'$ et du bord du domaine $\delta M'$; diagramme de Voronoï global Vor ; superposition montrant que $\widehat{\text{Vor}} \subseteq \text{Vor}$.

7.3 CONCLUSION DU CHAPITRE

Dans ce chapitre nous avons démontré deux caractéristiques importantes des diagrammes de Voronoï.

Dans un premier temps, nous avons montré qu'à partir de la définition du bassin de Voronoï d'un site il était possible de déterminer les cellules étant globalement définies dans un sous-espace $M' \subseteq M$, c'est-à-dire les cellules conservant la même forme peu importe le placement des sites S contenus dans le reste du domaine.

Nous avons ensuite étendu cette notion pour démontrer qu'à partir d'un sous-espace $M' \subseteq M$ et de l'ensemble de sites $S' = S \cap M'$, il était possible de construire un morceau du diagramme de Voronoï global $\widehat{\text{Vor}} \subseteq \text{Vor}$, bien que n'ayant aucune connaissance de la position des sites en dehors du domaine M' .

De cette manière, il est possible d'évaluer un diagramme de Voronoï par morceaux quand il n'est pas possible de calculer le diagramme de manière globale, simplement en définissant une région et en évaluant le diagramme de Voronoï uniquement avec les sommets faisant partie de cette région et le bord de la région définie.

Nous pensons que la mise en avant de ces caractéristiques justifie l'évaluation locale d'un diagramme de Voronoï quand le diagramme global ne peut être construit, dû à une consommation mémoire trop importante ou tout simplement parce que

les sites en dehors de la région considérée ne sont pas connus. En particulier, le chapitre suivant présente un exemple d'utilisation des notions introduites dans ce chapitre. Nous montrons comment les utiliser pour construire un diagramme de Voronoï de la surface sous-jacente d'un nuage de points, en s'appuyant sur la structure présentée dans le Chapitre 3.

DIAGRAMME DE VORONOÏ CENTROÏDAL PAR GRAPHE LOCAUX

Nous présentons maintenant nos contributions dans le domaine du partitionnement de nuages de points en nous appuyant sur la théorie présentée dans le chapitre précédent. Nous montrons dans un premier temps comment la surface sous-jacente d'une acquisition peut-être partitionnée sous la forme d'un diagramme de Voronoï centroïdal. Ensuite nous montrons comment, en utilisant la structure présentée dans le Chapitre 3, nous généralisons la construction de diagrammes de Voronoï centroïdaux à un ensemble d'acquisitions. Finalement nous présentons deux applications de l'approche proposée : une dédiée à l'amélioration de la qualité des échantillonnages générés grâce à la méthode présentée dans le Chapitre 5 ; l'autre dédiée à la reconstruction d'une surface polygonale à partir d'un nuage de points.

8.1 DIAGRAMME DE VORONOÏ CENTROÏDAL D'UNE ACQUISITION

Tel que présenté auparavant, la construction d'un diagramme de Voronoï centroïdal se fait en 2 étapes :

- Construction d'un diagramme de Voronoï,
- Relaxation de la position des sites.

8.1.1 Construction d'un diagramme de Voronoï

Rappelons que dans l'approche présentée dans le Chapitre 3, une acquisition est représentée sous la forme d'un graphe $G_1(V_1, E_1)$. La construction d'un diagramme de Voronoï nécessite de placer un ensemble de sites S . Dans le cas où la partition à réaliser se fait sur un graphe, les sites initiaux sont donc des sommets du graphe $S \subseteq V_1$. L'étape de partitionnement consiste donc à déterminer de quel site $s_i \in S$ chaque sommet du graphe $v \in V_1$ est le plus proche.

Notre partitionnement s'inspire du travail de [PEYROT, 2014], fournissant un algorithme pour la génération de diagrammes de Voronoï discrets de manière parallèle à l'aide d'une approche GPGPU. L'image stéréoscopique considérée dans leur travail est remplacée dans notre cas par le graphe représentant une acquisition.

Nous avons donc généralisé l'algorithme introduit dans leur travail au traitement de connectivités plus génériques que celles d'une image stéréoscopique, comme décrit dans l'Algorithme 2. L'idée générale de cet algorithme consiste à réaliser le partitionnement sous la forme d'une croissance de régions à partir de chaque site, de façon à faire croître les cellules de Voronoï autour des différents sites jusqu'à ce que l'influence d'un site soit plus faible que celle d'un autre site. Etant donné que la croissance des cellules est réalisée en parallèle, la mise à jour des distances ainsi que de l'appartenance d'un sommet à une région donnée doit être faite de manière séquentielle, afin d'éviter les problèmes d'accès concurrents lorsque plusieurs cellules sont en train de croître au niveau du même sommet.

Entrées : Un graphe $G_1(V_1, E_1)$ et $S \subseteq V_1$ l'ensemble des sites.

Résultat : La partition de Voronoï discrète générée Vor_1

/* Initialisation des distances de tous les sommets du graphe */

$D_1 = +\infty;$

/* Initialisation de la partition de Voronoï */

$Vor_1 = \text{null};$

/* Début section parallèle */

pour chaque *site* $s_i \in S$ **faire**

$D_1(s) = 0;$

$Vor_1(s_i) = s_i;$

$L = \{s_i\};$

tant que $L \neq \emptyset$ **faire**

$s' = L.tête();$

$L.supprimerElément(s');$

pour chaque *sommet* $v \in V_1$ *adjacent à* s' **faire**

si $D_1(s') + \text{distance}(s', v) < D_1(v)$ **alors**

 /* Début exclusion mutuelle */

$D_1(v) = D_1(s') + \text{distance}(s', v);$

$Vor_1(v) = s_i;$

$L.insérer(v);$

 /* Fin exclusion mutuelle */

fin

fin

fin

fin

/* Fin section parallèle */

Algorithme 2 : Algorithme de partition de Voronoï discret sur un graphe de manière parallèle.

Ce qui est très intéressant avec cette approche, c'est que la construction du diagramme de Voronoï permet d'obtenir une partition de la surface décrite par les échantillons d'une acquisition, et non pas une partition de l'espace ambiant autour du nuage de points (Figure 67).

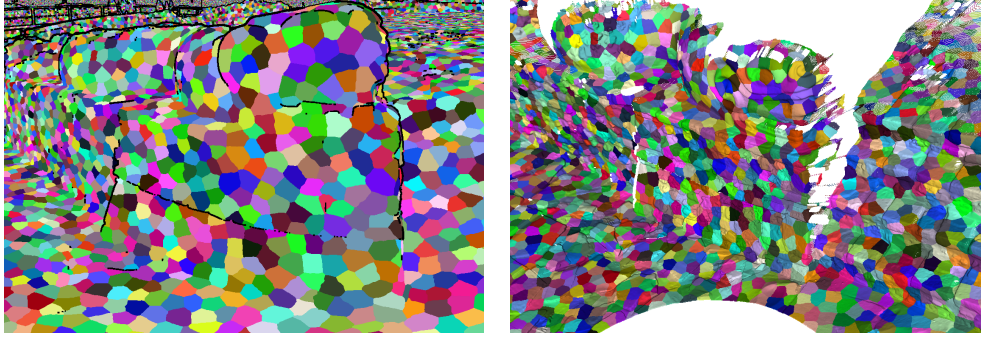


FIGURE 67 – Diagramme de Voronoï construit sur une acquisition de la donnée *Templo Mayor*. Le diagramme de Voronoï est construit dans le domaine de paramétrisation fourni par le graphe (à gauche) puis représenté dans l'espace 3D (à droite).

De cette manière, une cellule de Voronoï représente un morceau de surface, et non pas un volume dans l'espace. Cette particularité va permettre, comme nous allons le présenter par la suite, de modifier la position des sites de Voronoï sur la surface sous-jacente au nuage de points, ainsi qu'obtenir une triangulation de Delaunay, duale du diagramme de Voronoï, représentant la surface numérisée.

8.1.2 Relaxation de la position des sites

Comme introduit précédemment, l'algorithme de Lloyd consiste en une alternance entre la construction d'un diagramme de Voronoï et le déplacement de chaque site sur le centroïde de leur cellule. Etant donné que le domaine de définition du diagramme de Voronoï est la surface décrite par la connectivité du graphe G_1 , le centroïde de la cellule doit lui aussi être construit au sein de ce domaine. Pour l'obtenir, nous utilisons donc la *moyenne de Karcher* (*Karcher mean / Riemannian center of mass* [KARCHER, 2014] en anglais) [KARCHER, 1977] :

$$c_i = \arg \min_{y \in M} \int_{C_i} d(x, y)^2 dx, \quad (23)$$

permettant de trouver le centroïde de la cellule, le point qui minimise l'intégrale de la distance au carré à tous les points de la cellule. Dans notre cas, le domaine M correspond à l'ensemble des sommets du graphe V_1 :

$$c_i = \arg \min_{v \in V_1} \sum_{v_i \in C_i} d(v_i, v)^2 dx. \quad (24)$$

De cette manière, la nouvelle position d'un site correspondra toujours au sommet du graphe qui minimise la moyenne de Karcher, le *médoïde* de la cellule.

8.1.3 Métrique considérée

Jusqu'à présent, nous n'avons pas évoqué la moindre métrique associée au graphe pour la génération du diagramme de Voronoï centroïdal. En effet, les étapes présentées précédemment s'abstraient de la distance considérée. Cette abstraction nous permet donc de construire différents diagrammes de Voronoï en modifiant simplement la métrique associée au graphe, *i.e.*, le poids associé à chaque arête, de la même manière que dans le Chapitre 5.

Afin d'obtenir une distribution uniforme des sites sur la surface sous-jacente au nuage de points, nous associons une métrique géodésique. Mais il est aussi possible d'associer un poids dépendant de critères non-uniformes. De manière similaire que dans le Chapitre 5, nous proposons d'utiliser la courbure pour modifier la métrique géodésique, de manière à réduire l'aire des cellules de Voronoï dans les zones de forte courbure. Ainsi les sites se retrouveront naturellement attirés dans ces zones lors de la relaxation.

8.2 DIAGRAMME DE VORONOÏ CENTROÏDAL D'UN ENSEMBLE D'ACQUISITIONS

Maintenant que nous avons expliqué le cas traitant d'une seule acquisition, nous présentons sa généralisation dans le cas où un nuage de points est représenté par un ensemble d'acquisitions. Rappelons que dans ce cas, nous avons un ensemble de k graphes $G = \{G_1, G_2, \dots, G_k\}$ représentant chacun la connectivité de la surface capturée par une acquisition particulière qui sont connectés entre eux au travers de leurs sommets correspondants. Au sein de chaque graphe $G_i(V_i, E_i)$, les sommets V_i ont été distingués en deux classes différentes : l'ensemble V_i^+ qui sont les sommets v_i appartenant aux zones les plus échantillonnées parmi leurs sommets correspondants L_{v_i} ; et les sommets V_i^- correspondant au reste.

Les résultats du Chapitre 7 nous ont montré qu'il est possible de construire des morceaux d'un diagramme de Voronoï global au sein d'un sous-domaine $M' \subseteq M$ du domaine global M . L'idée est de calculer des partitions de Voronoï de manière locale sur les sommets V_i^+ de chaque graphe $G_i(V_i, E_i)$. En effet, comme nous l'avions présenté dans le Chapitre 3, les calculs réalisés sur les sommets V_i^+ permettent d'obtenir les meilleures approximations parmi tous les autres graphes pour la même région du graphe. Nous souhaitons donc continuer d'exploiter cette particularité pour obtenir les estimations de distances les plus précises possibles avec la structure utilisée. Ces différentes partitions locales sont ensuite agrégées

ensemble, en utilisant les correspondances entre les différents graphes.

La génération d'une partition de Voronoï de la surface sous-jacente au nuage de points se répartie en 3 étapes (voir Figure 68) :

- Génération de partitions locales,
- Transmission des partitions locales entre tous les graphes,
- Génération d'une partition globale.

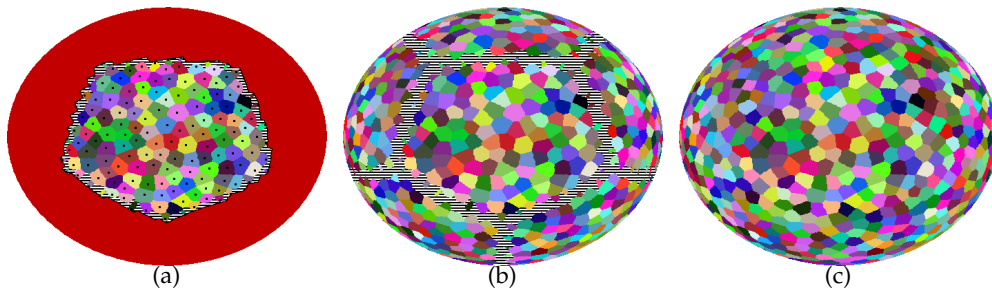


FIGURE 68 – Les 3 étapes de la création d'un diagramme de Voronoï à partir d'un ensemble de graphes. (a) Un morceau de la partition de Voronoï globale \widehat{Vor} est construit de manière locale. (b) Les morceaux de partition de chaque graphe $\widehat{Vor}_i, \forall G_i \in G$ sont transmis entre tous les graphes. (c) A partir des différents morceaux de partition, une partition globale Vor peut être construite en faisant se rencontrer les différentes partitions locales \widehat{Vor}_i (c).

8.2.1 Génération de partitions locales

L'idée est de construire une partition de Voronoï Vor_i sur l'ensemble des sommets V_i^+ de manière purement locale, en ne considérant que les sites $\bar{S} = S \cap V_i^+$. Bien évidemment, la partition construite ne prend pas en compte la position des sites $S \setminus V_i^+$ et par conséquent n'est pas entièrement cohérente (dans les zones proches des frontières entre les sommets V_i^+ et V_j^+ de chaque paire de graphes G_i et G_j).

Pour chaque graphe $G_i \in G$, nous souhaitons construire un diagramme de Voronoï local uniquement sur les sommets V_i^+ en ne considérant que les sites $S_i = S \cap V_i^+$. Nous décidons donc de nous appuyer sur les caractéristiques locales des diagrammes de Voronoï que nous avons présentées dans le chapitre précédent pour définir la région \widehat{Vor}_i dans laquelle la partition construite est globalement définie.

Nous savons, grâce au Lemme 7.2.2, qu'une partie du diagramme de Voronoï calculé au sein d'un graphe correspond à un sous-ensemble du diagramme de Voronoï calculé globalement sur la surface. En se référant aux notations introduites

précédemment, la surface correspond au domaine M et chaque ensemble de sommets V_i^+ d'un graphe représente un sous-domaine M'_i . Dans le cas de graphes créés par l'approche présentée dans le Chapitre 3, le bord du domaine $\delta M'_i$ est défini comme étant l'ensemble des sommets $V_i^{+\delta} \subseteq V_i^+$ adjacents à au moins un sommet V_i^- .

Nous cherchons donc à identifier la zone \widehat{Vor}_i du diagramme de Voronoï local qui est globalement définie dans le sous domaine M'_i , *i.e.*, $\widehat{Vor}_i \subseteq Vor$. De ce fait, lors de la construction de la partition locale, à l'aide de l'Algorithme 2, nous marquons l'ensemble des sommets plus proches des sites S_i que des sommets $V_i^{+\delta}$. Ainsi, pour chaque graphe, une partie du diagramme de Voronoï global $\widehat{Vor}_i \subseteq Vor$ peut être construite (Figure 69).

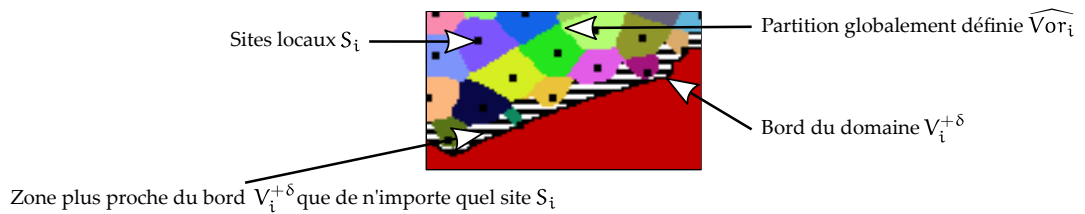


FIGURE 69 – Partie du diagramme de Voronoï global $\widehat{Vor}_i \subseteq Vor$ construite à l'aide des sites locaux S_i et des sommets au bord du domaine $V_i^{+\delta} \subseteq V_i^+$.

8.2.2 Transmission des partitions locales entre tous les graphes

Une fois que les partitions locales ont été construites, l'idée est de transmettre la partie de chaque diagramme de Voronoï globalement définie. La transmission est assez similaire à celle présentée lors de la transmission des disques de Poisson dans le Chapitre 5. Chaque sommet $v \in \widehat{Vor}_i$ de chaque graphe $G_i(V_i, E_i)$ indique à l'ensemble de ses sommets correspondants L_v le site $s_i \in S_i$ duquel ils sont le plus proches, ainsi que la distance à ce site.

Chaque graphe se retrouve donc avec une partition Vor_i contenant différents morceaux de la partition globalement définies (Figure 68(b)) et les distances D_i pour chaque sommet de ces morceaux de la partition globale.

8.2.3 Génération d'une partition globale

Nous allons maintenant construire un diagramme de Voronoï local sur l'ensemble des sommets V_i en faisant se rencontrer les différents morceaux de partition.

Pour cela, nous construisons une partition en considérant les régions ainsi que les distances ayant été transmises par les autres graphes, afin d'attribuer une région aux sommets qui n'ont pas été associés précédemment (car trop proches du bord du domaine). Nous faisons donc croître des régions, non plus à partir des sites du domaine, mais à partir des régions déjà construites, afin de faire se rencontrer les cellules de Voronoï de sites provenant de graphes différents (Algorithme 3).

Entrées : $G_i(V_i, E_i)$, Vor_i et D_i .

Résultat : La partition de Voronoï discrète générée Vor_i

$L = \{x_i | x_i \in Vor_i, x_i \neq \text{null}\};$

/* Début section parallèle */

tant que $L \neq \emptyset$ **faire**

$s' = L.tête();$

$L.supprimerElément(s');$

pour chaque *sommet* $v \in V_i$ *adjacent à* s' **faire**

si $D_i(s') + \text{distance}(s', v) < D_i(v)$ **alors**

 /* Début exclusion mutuelle */

$D_i(v) = D_i(s') + \text{distance}(s', v);$

$Vor_i(v) = Vor_i(s');$

$L.insérer(v);$

 /* Fin exclusion mutuelle */

fin

fin

fin

/* Fin section parallèle */

Algorithme 3 : Algorithme de partition de Voronoï discret sur un graphe de manière parallèle.

Pour chaque graphe $G_i \in G$, un diagramme de Voronoï est maintenant construit sur l'ensemble des sommets V_i . Plutôt que de calculer les partitions locales restantes directement, celles-ci sont mises à jour à partir de la partition venant d'être calculée. L'ensemble des sommets V_i^+ du graphe G_i dont le diagramme vient d'être construit informent leurs sommets correspondants dans les autres graphes du site duquel ils sont le plus proches, ainsi que la distance à celui-ci. De cette manière nous pouvons forcer la cohérence des partitions dans les zones à la frontière entre les sommets V_i^+ et V_j^+ de deux graphes G_i et G_j différents, permettant d'obtenir ainsi une partition complète (Figure 68(c)) et cohérente entre les différents graphes.

C'est ainsi qu'il nous est possible de générer des diagrammes de Voronoï de la surface sous-jacente à des nuage de points (Figure 70).

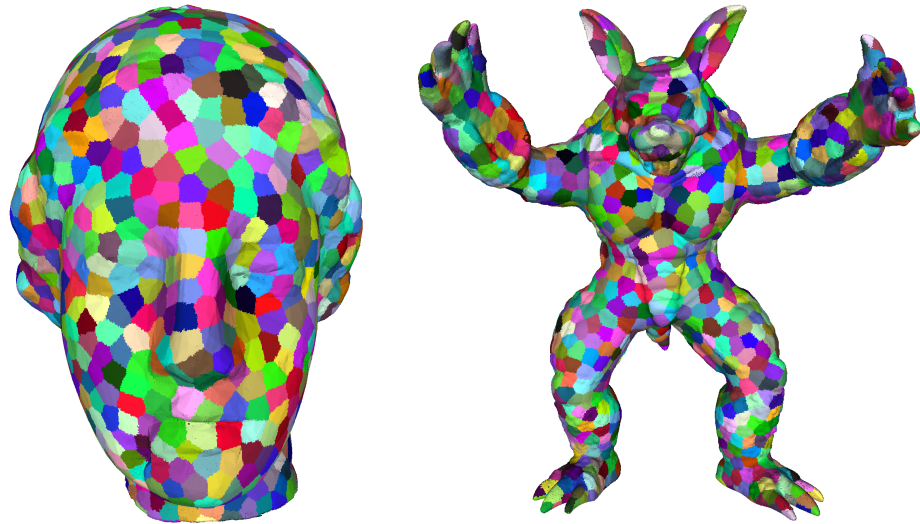


FIGURE 70 – Diagrammes de Voronoï de la surface sous-jacente à des nuage de points issus des modèles *Egea* (à gauche) et *Armadillo* (à droite).

8.2.4 Relaxation de l'ensemble des sites

La relaxation des sites dans le cas d'un ensemble de graphes est similaire au cas où il n'y a qu'un seul graphe. La seule différence consiste, après chaque itération, en la ré-attribution de chaque site $s_i \in S$ au graphe G_i ayant échantillonné le plus densément le sommet correspondant. La répétition de l'alternance de la construction d'un diagramme et de la relaxation des sites nous permet de construire un diagramme de Voronoï centroïdal de la surface numérisée (Figure 71).



FIGURE 71 – Diagramme de Voronoï de la surface sous-jacente à un nuage de points issu du modèle *Armadillo* avant (à gauche) et après convergence de l'algorithme de Lloyd (à droite).

8.2.5 Importance de l'initialisation de la position des sites

Un parallèle intéressant est à noter entre l'uniformité d'une distribution d'échantillons et l'uniformité de l'aire de cellules d'un diagramme de Voronoï. En fait, un diagramme de Voronoï centroïdal est obtenu si les sites sont disposés de manière uniforme sur le domaine.

Il est évident de constater que le nombre d'itérations de l'algorithme de Lloyd dépend fortement de la configuration initiale des sites. Une mauvaise configuration, *i. e.*, loin d'un minimum local, demandera plus d'itérations qu'une configuration proche d'un minimum local.

Un diagramme de Voronoï centroïdal se caractérise par des sites étant répartis de manière uniforme sur le domaine (par rapport à la métrique considérée). Ce qui est intéressant c'est que cette caractéristique est aussi l'objectif visé de l'échantillonnage en disques de Poisson que nous avons présenté dans le Chapitre 5. Simplement, ce type d'échantillonnage permet uniquement de placer les échantillons dans une position quasi-uniforme. Car, rappelons que ce type d'échantillonnage est obtenu par des tirages aléatoires - tous les points du domaine ont la même probabilité d'être tirée - et que la seule contrainte directement imposée durant l'échantillonnage est une condition de distance minimum entre les échantillons sélectionnés. De ce fait, il n'y a pas de contrôle direct sur l'uniformité de la distribution au niveau local, et il peut donc y avoir une variance assez importante dans les distances entre un échantillon et ses plus proches voisins. Cependant, la configuration des échantillons d'une distribution en disques de Poisson est très proche d'une distribution uniforme. Par conséquent, la position de ces échantillons fournit une bonne initialisation de la position des sites pour l'algorithme de Lloyd, car peu d'itérations sont nécessaires.

Nous avons montré dans le Chapitre 5 que l'algorithme d'échantillonnage proposé permettait de générer des distributions ayant de bonnes caractéristiques de bruit bleu et offrant une bonne couverture du domaine, grâce à la condition de maximalité de l'échantillonnage. De ce fait, les distributions générées par notre méthode fournissent une bonne initialisation à la première partition, puisque permettant de générer un diagramme de Voronoï très proche d'une partition de Voronoï centroïdale.

8.3 RÉSULTATS EXPÉRIMENTAUX

Nous présentons maintenant des résultats expérimentaux de l'application de la méthode de construction d'un diagramme de Voronoï centroïdal.

Nous invitons le lecteur à se référer à l'Annexe A pour obtenir les caractéristiques de la machine sur laquelle les expérimentations ont été effectuées ainsi que des informations techniques concernant les différentes données de test utilisées.

8.3.1 *Amélioration de la qualité de l'échantillonnage*

Nous présentons dans un premier temps des résultats visuels de l'amélioration de l'échantillonnage pour différents modèles synthétiques (Figures 72 et 73). Nous pouvons remarquer que les distributions après relaxation sont plus uniformes, les distances entre les échantillons les plus proches sont plus homogènes qu'avant la relaxation.

En analysant la qualité des distributions à l'aide de l'outil d'analyse différentielle de [WEI et WANG, 2011] présenté dans le Chapitre 5, nous pouvons faire plusieurs constats. Premièrement, nous observons que les distributions de sites relaxés présentent aussi des propriétés de bruit bleu, bien de qualité légèrement inférieure aux distributions avant relaxation. En effet, le SPMR présente un peu plus d'oscillations en moyennes et hautes fréquences avant de se stabiliser. Ensuite, nous pouvons remarquer que l'anisotropie est restée quasiment constante, confirmant bien que les distributions générées restent isotropes (comme les Figures 72 et 73 ont pu aussi le mettre en avant).

Nous montrons maintenant des résultats visuels de l'amélioration de l'échantillonnage pour différentes campagnes d'acquisition réelles (Figures 76 et 77). Afin de pouvoir mieux visualiser la distribution des échantillons sur la surface décrite par le nuage de points, nous avons reconstruit des surfaces à partir de la position des échantillons du nuage de points origine, à l'aide de l'algorithme de reconstruction de surfaces de Poisson [KAZHDAN et HOPPE, 2013] de manière similaire que dans le Chapitre 5. De manière générale, nous avons observé qu'il faut moins de 10 itérations pour que l'algorithme converge vers un minimum local sur les données réelles. Grâce à l'utilisation de la moyenne de Karcher ainsi qu'à une métrique prenant en compte la courbure de la surface, les sites ont tendance à naturellement se déplacer sur les zones présentant du détail, ainsi qu'à se positionner sur les arêtes franches et s'espacer de manière régulière sur celles-ci.

8.3.2 *Performances en temps de calcul et consommation mémoire*

Nous présentons quelques résultats en termes de temps de calcul et pic de consommation mémoire pour la construction d'un diagramme de Voronoï global sur des données synthétiques. La Table 7 présente les résultats de construction

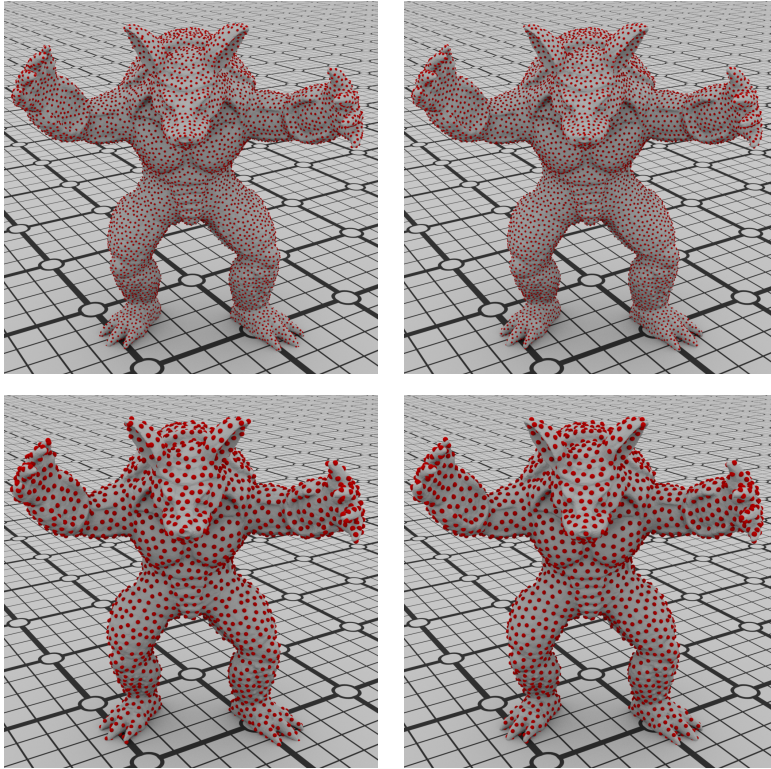


FIGURE 72 – Différence visuelle entre une distribution uniforme générée avec l’approche présentée dans le Chapitre 5 (à gauche) et après convergence de l’algorithme de Lloyd (à droite). En haut et en bas deux densités différentes du modèle *Armadillo*.

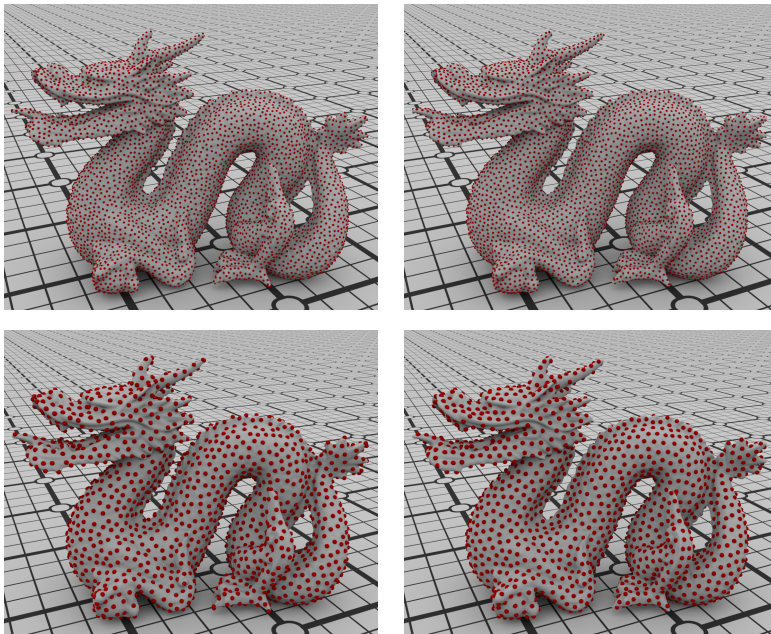


FIGURE 73 – Différence visuelle entre une distribution uniforme générée avec l’approche présentée dans le Chapitre 5 (à gauche) et après convergence de l’algorithme de Lloyd (à droite). En haut et en bas deux densités différentes du modèle *Dragon*.

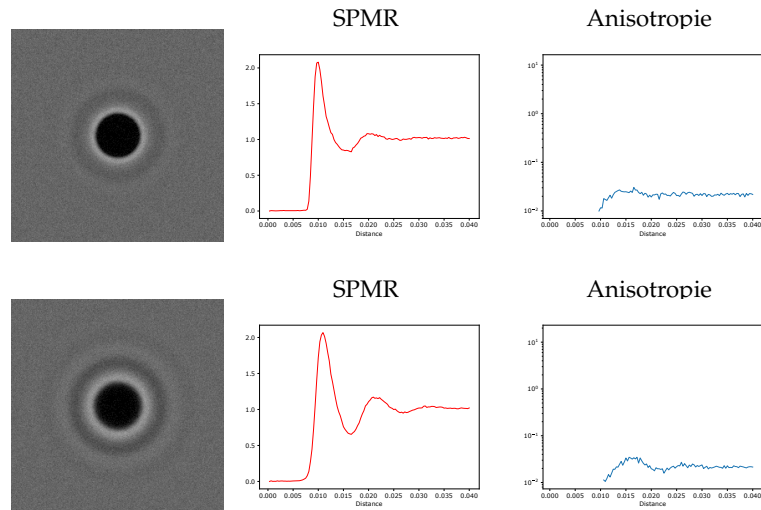


FIGURE 74 – Analyse différentielle de distributions générées avec l’approche présentée dans le Chapitre 5 avant (en haut) et après 10 itérations de l’algorithme de Lloyd (en bas). En haut et en bas deux densités différentes du modèle *Bunny*.

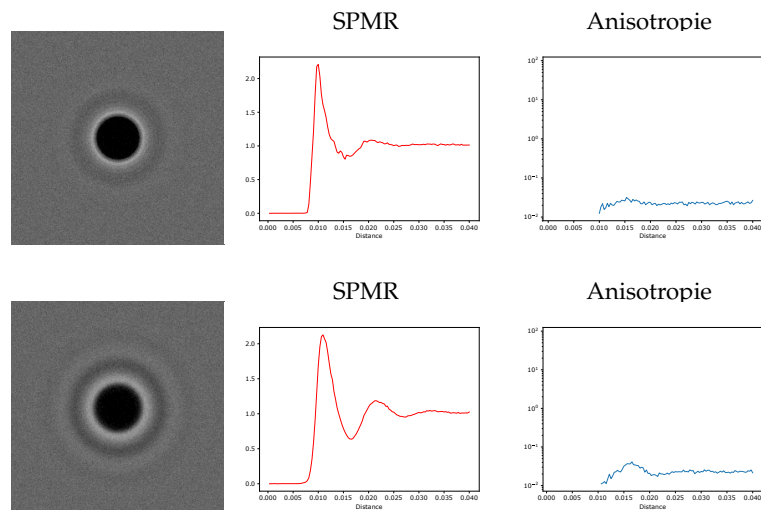


FIGURE 75 – Analyse différentielle de distributions générées avec l’approche présentée dans le Chapitre 5 avant (en haut) et après 10 itérations de l’algorithme de Lloyd (en bas). En haut et en bas deux densités différentes du modèle *Venus*.

d’un diagramme de Voronoï obtenu en considérant l’ensemble des graphes dans la mémoire vive. Les données synthétiques sont générées de la même manière que nous l’avons présenté dans le Chapitre 3. Chaque donnée est donc constituée de 12 acquisitions ayant chacune une résolution de 1024×1024 pixels.

Nous pouvons constater que le temps nécessaire à la construction du diagramme de Voronoï n’est pas directement lié au nombre de sites considérés. En effet, étant donné que la construction du diagramme de Voronoï est réalisée sous la forme d’une croissance de région, la complexité de celle-ci est fortement dépendante du

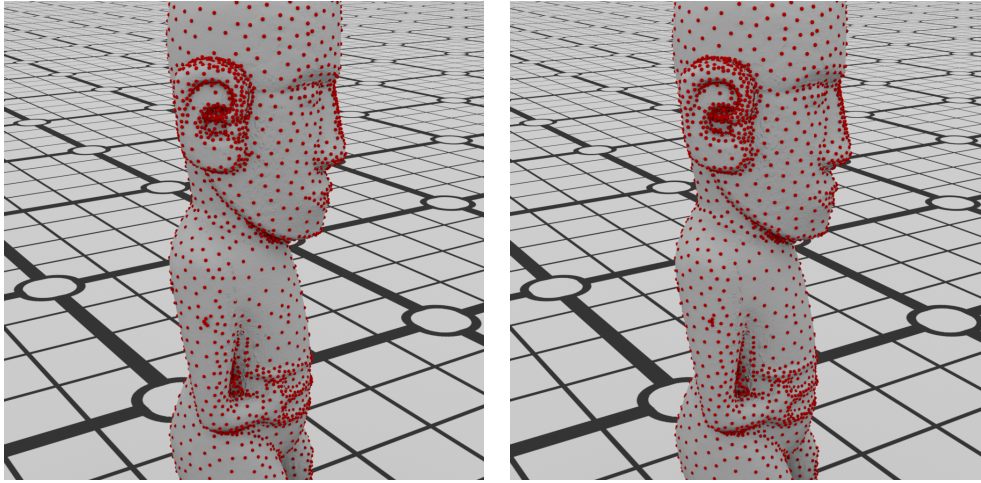


FIGURE 76 – Comparaison visuelle de la qualité d’une distribution adaptative avant (à gauche) et après (à droite) relaxation des sites de la donnée *Meeting House*.

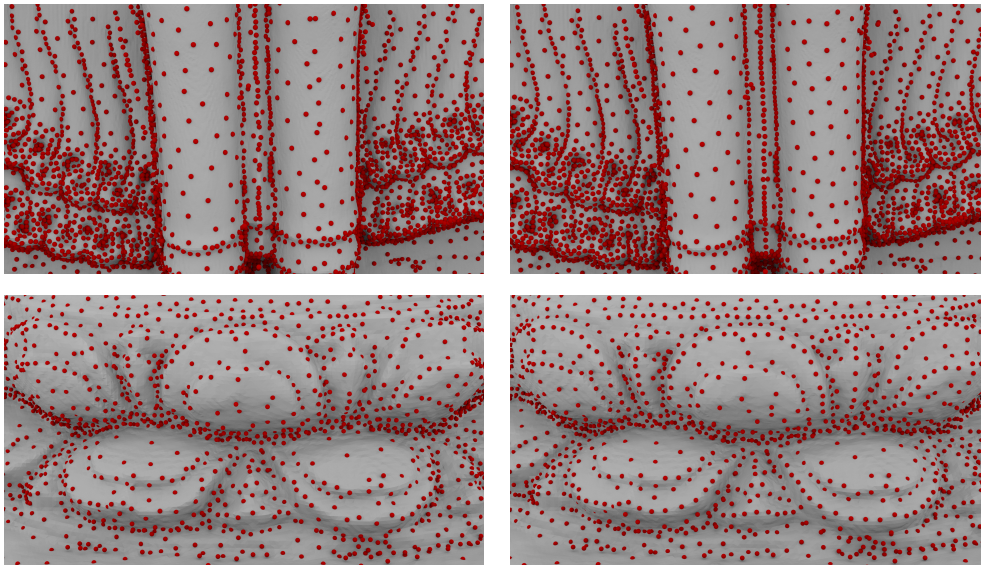


FIGURE 77 – Comparaison visuelle de la qualité d’une distribution adaptative avant (à gauche) et après (à droite) relaxation des sites de deux zones distinctes de la donnée *Eim Ya Kyaung*.

nombre de sommets contenus dans l’ensemble des graphes. Nous pouvons toutefois noter que la construction d’un diagramme de Voronoï à partir d’un faible nombre de sites est plus coûteuse en temps de calcul qu’à partir d’un grand nombre de sites. Ceci est dû à l’approche proposée de croissance de régions de manière parallèle. Comme la distribution des sites à l’origine est très uniforme (car elle résulte d’un échantillonnage en disques de Poisson), plus le nombre de sites est important plus les cellules générées vont se rencontrer rapidement et ainsi arrêter leur propagation.

Nom	Sites	Temps (s)	
		Construction	Relaxation (itérations)
<i>Egea</i>	1 655	7,7	27,5 (6)
	24 009	5,0	21,8 (5)
<i>Venus</i>	1 642	5,0	27,7 (6)
	23 363	4,3	23,1 (6)
<i>Armadillo</i>	1 588	8,9	34,5 (7)
	21 044	4,8	21,7 (6)
<i>Bunny</i>	1 626	8,6	37,5 (8)
	23 343	4,3	23,2 (6)
<i>Dragon</i>	1 699	10,4	36,0 (7)
	23 186	4,6	24,4 (6)

TABLE 7 – **Temps** de calcul atteints par notre algorithme de construction de diagramme de Voronoï pour différentes données synthétiques. Pour chaque donnée nous montrons les temps de calcul pour des distributions de **sites** à deux densités différentes. Pour un ensemble de sites nous montrons le temps de **construction** d'un diagramme de Voronoï en utilisant l'approche présentée et le temps de calcul total pour que la **relaxation** converge ainsi que le nombre d'itérations nécessaires pour atteindre la convergence.

Enfin, ces résultats nous confirment que l'utilisation de distributions issues d'un échantillonnage en disques de Poisson fournit une bonne initialisation à la construction d'un diagramme de Voronoï centroïdal car très peu d'itérations sont nécessaires pour converger vers un minimum local (8 au maximum dans nos expérimentations). Cette particularité est aussi due à l'aspect discret de notre approche, car la précision du placement des sites est directement liée à la résolution des acquisitions. Par conséquent l'algorithme converge au moment où la discrétisation fournie par les graphes n'est plus assez fine pour continuer de déplacer les différents sites.

8.3.3 Réduction du coût mémoire

Maintenant, nous montrons quelques résultats sur le compromis entre temps de calcul et pic de consommation mémoire pour des données issues d'acquisitions réelles. Afin de réduire le coût mémoire nécessaire pour la construction d'un diagramme de Voronoï, nous pouvons nous servir des caractéristiques de la structure

présentée dans le Chapitre 3 pour réduire le pic mémoire atteint par notre algorithme.

La Table 8 montre des résultats en fonction de différentes tailles de tuiles. Pour information, la donnée *Meeting House (Interior)* représente un morceau issu de la donnée *Meeting House*, elle comprend 457M points et est représentée par 22 acquisitions.

Nom	Temps (h:m:s) / Pic mémoire (Gio)		
	Acq. natives	4096 ²	2048 ²
<i>Meeting House (Interior)</i>	00:37:02 / 2,3	01:56:51 / 1,3	03:21:32 / 0,8
	- / -	+215,5% / -43,5%	+444,2% / -65,2%
<i>Templo Mayor</i>	01:05:12 / 4,4	02:39:02 / 2,0	04:49:42 / 1,8
	- / -	+143,9% / -54,5%	+344,3% / -59,9%

TABLE 8 – Temps de calcul et pic mémoire atteints par notre algorithme de construction d’un diagramme de Voronoï global pour différentes tailles de tuiles (4096x4096 et 2048x2048). Les pourcentages de temps de calcul et de pic mémoire indiquent le coût en comparaison au traitement des données à leur taille native.

A nouveau, la structure sous forme de graphes locaux permet d’offrir un compromis entre le temps de calcul et le pic de consommation mémoire atteint.

8.3.4 Reconstruction de surface

La reconstruction de surface est un sujet très fortement étudié depuis une vingtaine d’années. Les auteurs de [BERGER et al., 2014] en ont d’ailleurs réalisé toute une taxonomie. Nous invitons le lecteur à s’y référer pour obtenir un état de l’art de la recherche dans le domaine.

La génération d’un diagramme de Voronoï de la surface sous-jacente à un nuage de points nous permet d’en extraire une triangulation de Delaunay, par dualité avec le diagramme de Voronoï. Pour extraire les intersections d’un diagramme de Voronoï discret, nous reprenons les idées du travail de [PEYROT, 2014] sur la détection des points d’intersection sur une image stéréoscopique.

Afin d’avoir un point de référence pour étudier la qualité des triangulations obtenues, nous décidons de comparer le travail proposé avec l’approche des auteurs de [BOLTCHÉVA et LEVY, 2017], introduite dans le Chapitre 5. Cette méthode, comme présentée auparavant, a l’avantage de considérer comme sommets du maillage généré les échantillons du nuage de points donné en entrée. De plus, par rapport aux autres méthodes de l’état de l’art, elle est une des seules qui permettent de traiter des données de plusieurs dizaines de millions de points. En outre, son approche

massivement parallèle en fait une des méthodes les plus efficaces en termes de temps de calcul.

L'originalité du travail de [BOLTCHÉVA et LEVY, 2017] vient de la manière dont les auteurs construisent un diagramme de Voronoï restreint à la surface. Ils calculent de manière explicite l'intersection entre une cellule de Voronoï 3D et un disque tangent à la surface centré sur le site à l'origine de la cellule, dont la taille est choisie arbitrairement. De cette manière, ils peuvent obtenir une partition de Voronoï de la surface sous-jacente à un nuage de points, à partir du moment où la densité d'échantillonnage du nuage de points est suffisamment dense. Étant donné que la méthode de [BOLTCHÉVA et LEVY, 2017] ne permet pas de réduire la quantité de points, nous ré-échantillons d'abord le nuage de points en utilisant l'approche présentée dans le Chapitre 5 et c'est cet échantillonnage qui est donné à l'entrée de leur algorithme. Nous comparons ensuite les reconstructions obtenues à l'aide de notre approche par rapport à celles obtenues avec le travail de [BOLTCHÉVA et LEVY, 2017].

Dans un premier temps, nous proposons de comparer les surfaces générées sur leur aspect global. Il est intéressant de noter que les deux approches génèrent des triangulations fortement similaires. Ce n'est pas étonnant étant donné que, sous certaines conditions de densité d'échantillonnage, l'approche de [BOLTCHÉVA et LEVY, 2017] construit des cellules similaires à notre approche basée graphes. Ensuite, nous pouvons constater un des défauts de l'approche de [BOLTCHÉVA et LEVY, 2017], étant donné que leur méthode s'abstrait de l'information topologique de la surface sous-jacente au nuage de points traité. Quand la densité d'échantillonnage n'est pas suffisamment importante, la manière dont ils construisent leurs disques tangents à la surface ne leur permet plus de décrire correctement le comportement de la surface. Par conséquent, dans ces cas-là, leur méthode a tendance à interpoler des informations très distinctes de la surface numérisée. C'est notamment visible au niveau des mains et des pieds du modèle *Armadillo* (Figure 78) ou du sol et des fenêtres de la donnée *Meeting House* (Figure 79). Tandis que notre approche, guidée par la connectivité des graphes, ne rencontre pas ce problème et est capable d'obtenir des approximations cohérentes de la surface, même à de faibles densités d'échantillonnage.

Par contre, lorsque la distribution des sites est dense, la triangulation générée par notre approche peut comporter certains défauts comme des trous (sous-détection d'intersection) ou des artéfacts non-variétés (sur-détection d'intersection). Dans ces conditions particulières, la métrique discrète que nous construisons est incapable de correctement déterminer la forme des cellules de Voronoï de manière cohérente entre les cartes, à cause de leur résolution limitée. Cependant, ce problème peut être corrigé en réalisant un post-traitement sur les maillages reconstruits, en

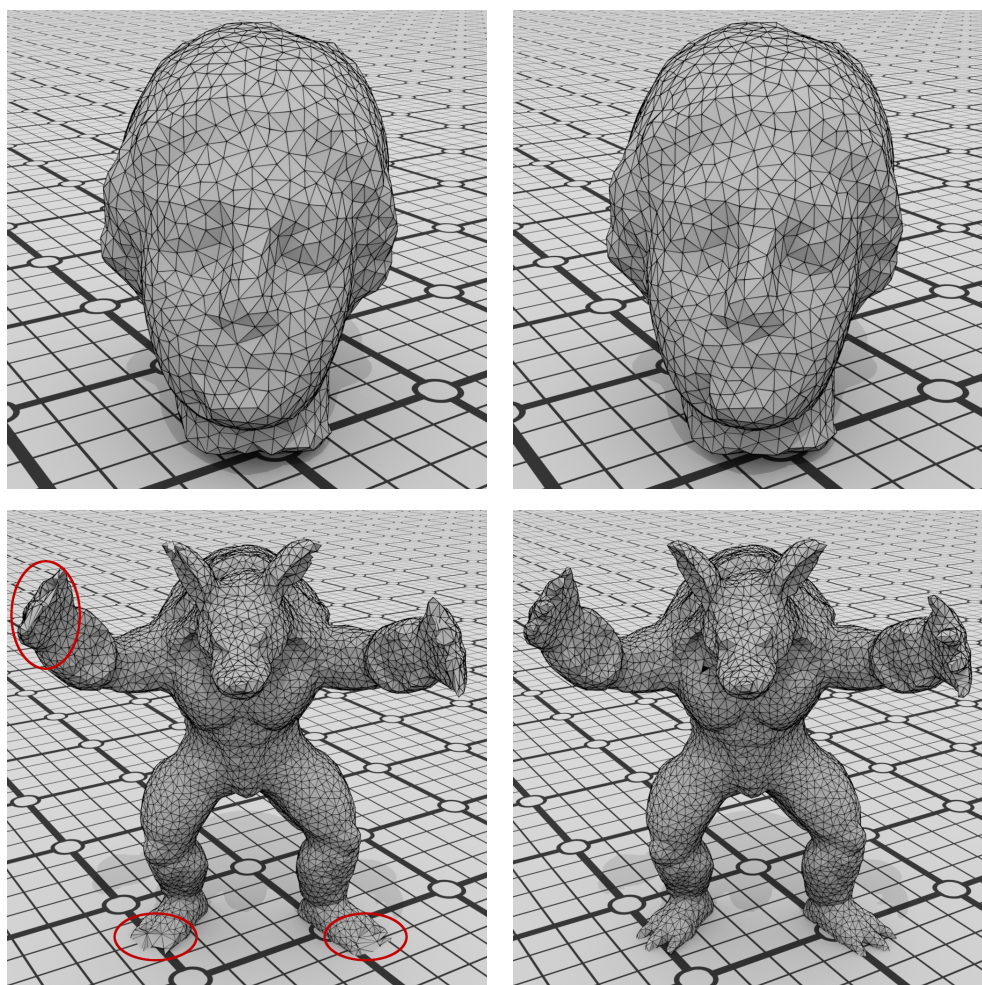


FIGURE 78 – Reconstructions obtenues avec l’approche de [BOLTCHÉVA et LEVY, 2017] (à gauche) et notre approche (à droite) sur les modèles *Egea* (en haut) et *Armadillo* (en bas). Il est intéressant de noter que les triangulations générées sont quasiment les mêmes (à quelques retournements d’arêtes près, à cause de l’approximation faite par notre approche sur le calcul de distances géodésiques). Lorsque la surface d’origine présente du détail mais que l’échantillonnage fourni n’est pas suffisamment dense, la méthode de [BOLTCHÉVA et LEVY, 2017] a tendance à trianguler ensemble des éléments qui sont en réalité assez éloignés sur la surface d’origine (au niveau des mains et des pieds du modèle *Armadillo*).

remplissant les trous (généralement de la taille d’un triangle) et en corrigeant les problèmes de non-variété, de manière similaire aux post-traitements effectués dans le travail de [BOLTCHÉVA et LEVY, 2017].

Nous comparons maintenant des triangulations générées par notre approche à différentes itérations de la relaxation de Lloyd. Nous pouvons constater sur la Figure 80 que les triangles générés ont de meilleurs aspect/ratio après seulement quelques itérations.

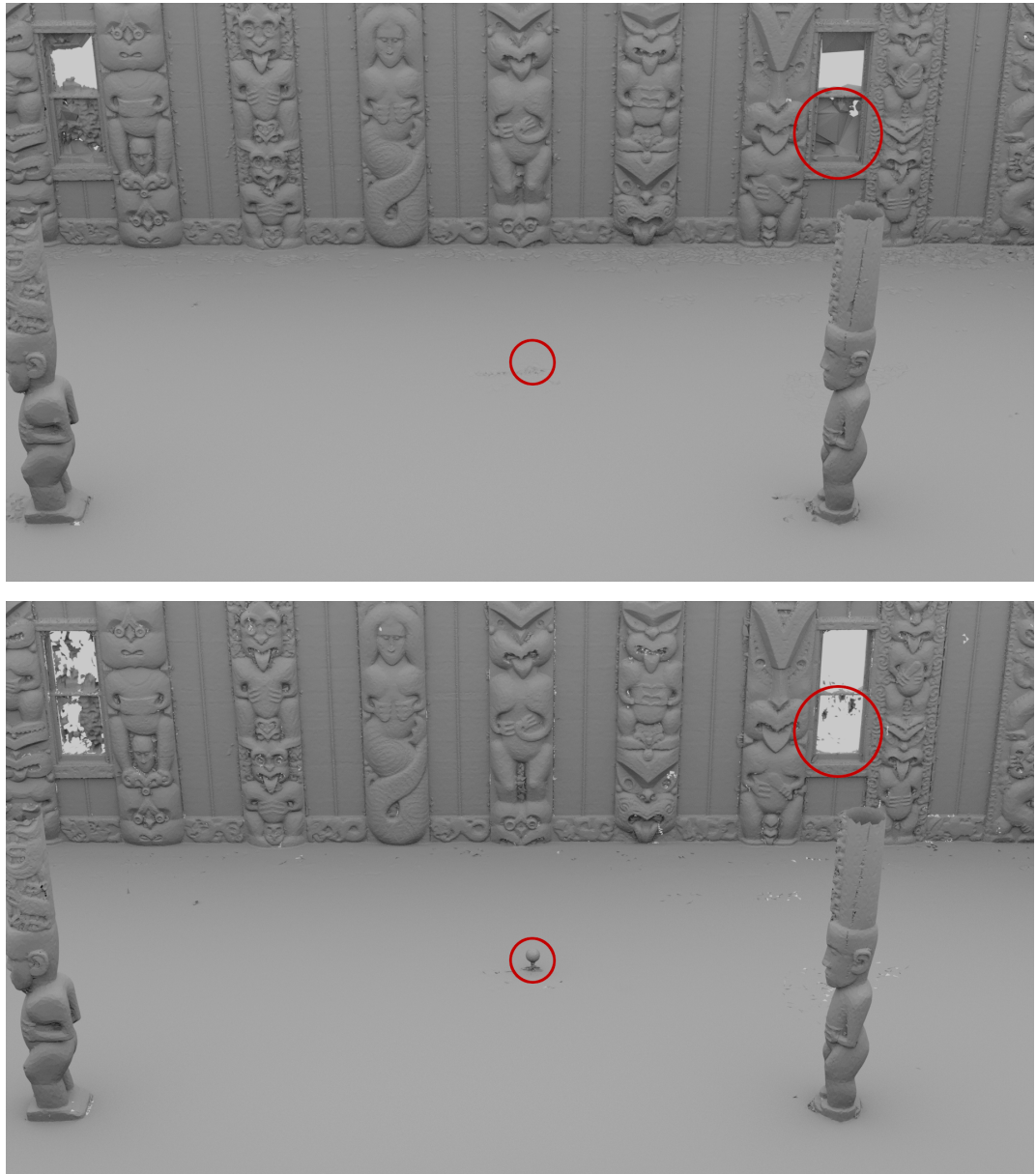


FIGURE 79 – Reconstructions obtenues avec l’approche de [BOLTCHÉVA et LEVY, 2017] (en haut) et notre approche (en bas) sur la donnée *Meeting House*. Contrairement à notre méthode, l’approche des auteurs de [BOLTCHÉVA et LEVY, 2017] a tendance à interpoler les données, faisant perdre du détail à certains endroits (sur le sol) et extrapolant la surface à des endroits où celle-ci ne devrait pas exister (au niveau des fenêtres).

8.4 CONCLUSION DU CHAPITRE

8.4.1 Contributions

Nous avons présenté dans ce chapitre une méthode pour la **génération de diagrammes de Voronoï centroïdaux** sur la surface sous-jacente à des nuages de

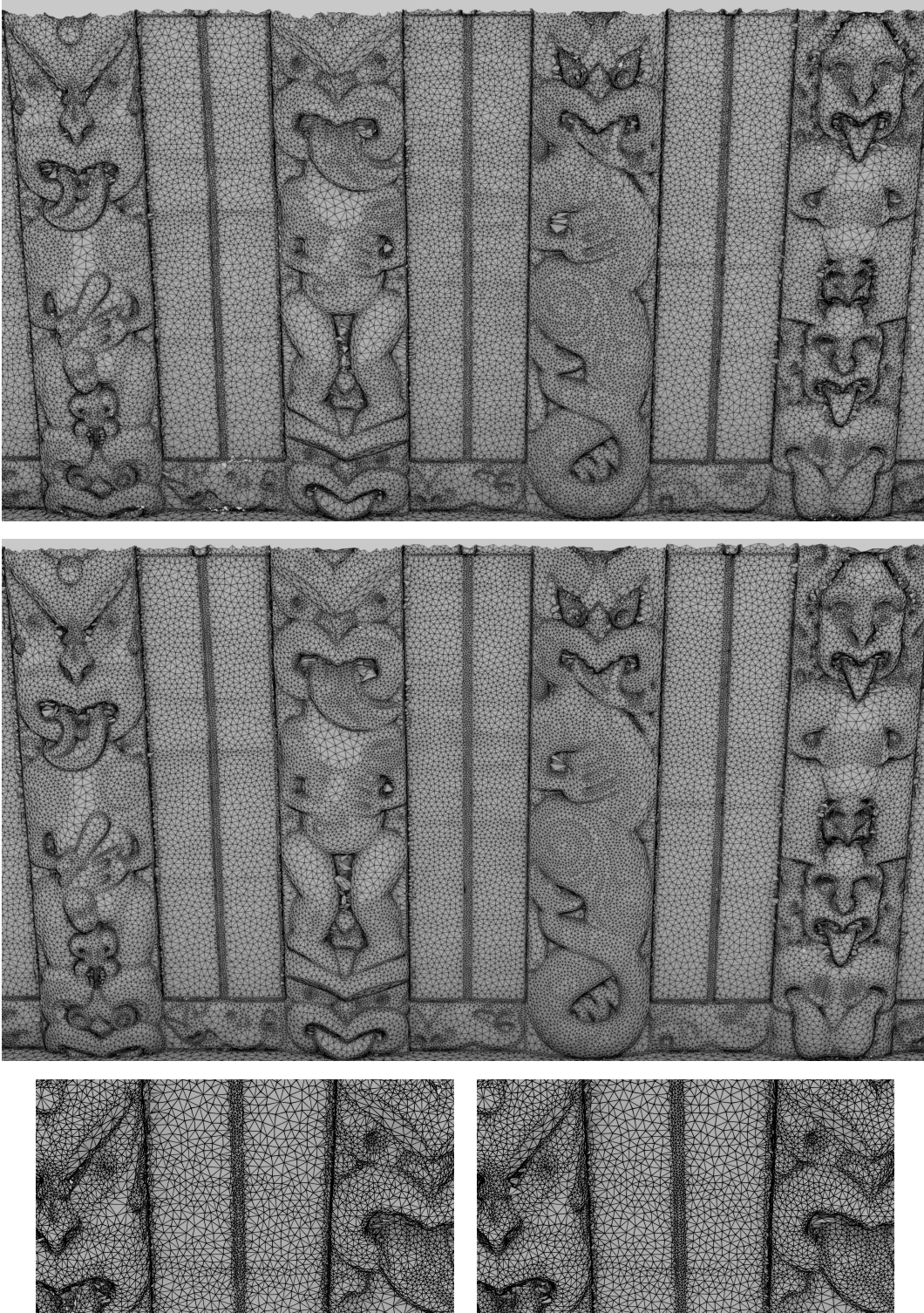


FIGURE 80 – Qualité de triangulations générées par notre approche à partir d'échantillonnages adaptatifs de la donnée *Meeting House*. Triangulations générées avant (en haut) et après convergence de l'algorithme (au milieu). En bas, nous pouvons voir une vue rapprochée des deux triangulations, avant (à gauche) et après convergence de la relaxation (à droite).

points massifs. La particularité de notre approche vient de l'utilisation de la structure présentée dans le Chapitre 3 pour construire des diagrammes de Voronoï sur la surface d'un nuage de points. Grâce aux notions théoriques présentées dans le Chapitre 7, nous avons montré qu'il était possible de construire un diagramme de Voronoï global à partir de constructions locales. En modifiant le poids des arêtes du graphe, nous avons construit des diagrammes de Voronoï modélisant différentes métriques, telles que la métrique géodésique ou une métrique dépendante de la courbure de la surface. Ces diagrammes de Voronoï nous servent de support à la mise en place un algorithme de relaxation de Lloyd, pouvant minimiser différentes métriques en basant le calcul des centroïdes sur une moyenne de Karcher.

Les différents résultats obtenus sur des données synthétiques mettent en avant **l'amélioration de l'uniformité de l'échantillonnage** grâce à l'étape de relaxation. Nous avons pu constater que la convergence de l'algorithme peut être atteinte en peu d'itérations en considérant comme sites initiaux les distributions obtenues par la méthode présentée dans le Chapitre 5. Nous avons ensuite validé que les distributions générées sur des campagnes d'acquisitions réelles avaient des caractéristiques similaires aux distributions obtenues à partir des données synthétiques.

Les diagrammes construits permettent d'**extraire des maillages triangulaires** représentant la surface de la scène numérisée. Les triangulations générées ont montré que la méthode proposée était capable de reconstruire des triangulations comparables avec un algorithme de l'état de l'art de la reconstruction de surfaces.

Grâce à la structure par graphes locaux, l'approche rend possible le traitement de nuages de points avec un **coût mémoire extrêmement faible**, même sur des nuages de points composés de plusieurs centaines de millions d'échantillons.

8.4.2 *Discussions*

L'approche proposée ouvre la porte à de nombreuses possibilités d'amélioration.

La manière de construire des diagrammes de Voronoï sur la surface représentée par un ensemble d'acquisitions est largement perfectible. Bien que nous puissions les calculer sur des campagnes d'acquisition réelles, peu importe le nombre de points qu'elles représentent, grâce à la structure de graphes locaux, le temps nécessaire pour les calculer peut devenir assez important pour des nuages de points composés de plusieurs milliards d'échantillons. Il y a deux raisons principales à ce problème. Premièrement, la complexité de la construction des diagrammes de Voronoï discrets est directement liée au nombre de sommets du nuage de points

original. Deuxièmement, la conception d'algorithmes externes a un coût qui se répercute sur les temps de calcul, comme nous l'avions présenté dans le Chapitre 3. La mise en place d'un système permettant de ne charger que les informations nécessaires aux différentes étapes de l'algorithme pénalise donc fortement les temps de calcul à cause des très nombreuses lectures/écritures depuis/vers les disques durs. Mais ce coût est aussi lié à la nature séquentielle d'une telle méthode, empêchant, par définition, de calculer les diagrammes de Voronoï locaux de manière parallèle. Cependant, la plupart des utilisateurs n'ont pas d'ordinateurs suffisamment puissants pour obtenir des reconstructions à partir de nuages de points aussi denses dans leur globalité. Notre approche prend alors tout son sens, même si celle-ci est sous-optimale en temps de calcul par rapport à des méthodes fortement parallèles et conservant l'intégralité des données en mémoire vive. De plus, nous pensons que les caractéristiques des diagrammes de Voronoï que nous avons mis en avant dans le Chapitre 7 pourraient être utilisées dans d'autres cadres que celui présenté ici, et permettraient peut-être à des algorithmes comme celui de [BOLT-CHEVA et LEVY, 2017] de pouvoir traiter des nuages de points aussi volumineux que ceux considérés dans notre travail.

Nous avons aussi pu remarquer que dans certains cas, les triangulations générées peuvent souffrir de certains défauts (comme des configurations non-variétés ou des trous). Comme expliqué un peu plus haut, ceci est la conséquence directe de l'aspect discret de notre méthode de construction du diagramme de Voronoï. En effet, dans les zones à la frontière entre une partie plus densément échantillonnée dans un graphe et une partie plus densément échantillonnée dans un autre graphe, la forme des partitions peut légèrement varier. Etant donné que la détection des intersections des cellules d'une région ne se fait que dans le graphe ayant le plus densément échantillonné la région concernée, il se peut qu'une intersection soit détectée entre plusieurs cellules, alors que cette intersection n'existe pas, au regard du pas de discrétisation d'autre graphes. C'est notamment le cas lorsque la densité de sites est élevée, étant donné que dans ce cas les cellules de Voronoï sont assez petites, et le pas de discrétisation proposée par la résolution des acquisitions ne suffit plus à précisément évaluer les distances à chacun des sites. Une idée serait de réaliser une approche hybride, qui évaluerait les distances aux sites à partir d'une métrique euclidienne 3D dans les zones où la résolution des cartes de profondeur n'est pas suffisante, pour détecter correctement et de manière cohérente les intersections entre les différentes cellules. De plus, il serait nécessaire de réaliser des vérifications sur les configurations du diagramme de Voronoï, tel qu'investigé dans le travail de [GUSKOV, 2007], car quand une zone ne contient pas suffisamment de sites, certaines configurations de ces sites ne permettent pas

à au moins 3 cellules de s'intersecter. Par conséquent, dans de telles zones, aucun triangle n'est généré.

Cinquième partie

CONCLUSION

CONCLUSION GÉNÉRALE

9.1 CONTRIBUTIONS

Dans ce manuscrit nous avons présenté un modèle original pour la représentation de la surface sous-jacente aux nuages de points massifs issus de campagnes d'acquisition de LiDARs terrestres. Le but de ce modèle est de fournir une unique structure pour la manipulation de données numérisées, de l'étape de numérisation à la manipulation de la maille reconstruite.

Graphes locaux. Nous avons tout d'abord présenté une manière de construire des graphes locaux à partir d'un ensemble d'acquisitions. En prenant avantage de la structure des acquisitions issues de LiDARs terrestres, nous avons montré que des graphes représentant localement la connectivité de la surface numérisée pouvaient être construits. En mettant en correspondance tous les graphes construits représentant les mêmes régions de la surface numérisée, nous avons montré comment des fonctions pouvaient être construites par morceaux, sur la surface sous-jacente globale d'un nuage de points. De plus, en développant une technique de découpages d'acquisitions à l'aide de tuiles se recouvrant partiellement, nous avons proposé une manière de contrôler le pic de consommation mémoire atteint lors des différents traitements appliqués sur les données. On peut ainsi traiter des données de n'importe quelle taille, peu importe le nombre d'acquisitions et le nombre de points nécessaires

Ré-échantillonnage de nuages de points. Ensuite nous avons montré comment le modèle introduit permet de réduire la quantité de points contenus dans un nuage de points tout en prenant compte de la surface numérisée lors de la réduction du nombre de points. Afin de réduire les défauts présents dans la distribution des échantillons de nuages de points issus de l'agrégation de différentes acquisitions, nous avons présenté la généralisation d'une méthode d'échantillonnage en disques de Poisson. En construisant localement un échantillonnage en disques de Poisson sur chaque graphe, et en transmettant les disques de Poisson dans les zones concernées des autres graphes, des distributions ayant de bonne qualité en termes de bruit bleu de manière globale ont pu être obtenues, malgré des traitements locaux.

Construction de diagrammes de Voronoï centroïdaux. Finalement, nous avons présenté une manière singulière de construire des diagrammes de Voronoï centroïdaux de la surface sous-jacente à des nuages de points. Nous avons tout d'abord présenté un cadre théorique sur la construction de diagrammes de Voronoï locaux sur des sous-parties d'un domaine, avec des garanties sur l'exactitude de ces partitions de manière globale. Nous avons mis en application ces théories pour construire des diagrammes de Voronoï sur l'ensemble de la surface représentée par un ensemble de graphes. Nous avons montré que les diagrammes de Voronoï construits à l'aide de la structure basée graphes permettent d'améliorer la qualité d'une distribution d'échantillons à l'aide d'un algorithme de relaxation. De plus, nous avons montré que le diagramme construit nous permettait d'extraire un maillage triangulaire approximant la surface numérisée.

9.2 PERSPECTIVES

Les différents résultats présentés dans cette thèse font la lumière sur un bon nombre de pistes de recherche.

Développement de nouveaux algorithmes basés sur la structure proposée. Le modèle se basant sur des graphes locaux s'est avéré être un outil essentiel à la manipulation de la surface sous-jacente de nuages de points massifs. Nous pensons que l'utilisation d'une telle structure est un moyen d'uniformiser la chaîne de numérisation complète, de l'acquisition à la transmission des données. Nous avons montré par exemple qu'elle s'avère fortement utile pour ré-échantillonner les données et reconstruire un maillage polygonal de la surface de la scène numérisée. Nous sommes convaincus qu'une telle structure peut être utilisée pour tout un ensemble de traitements différents. En particulier, pour la compression de données, nous pensons qu'il est possible d'exploiter la structure des acquisitions pour adapter des algorithmes de compression existants à l'utilisation de ces données.

Amélioration de la gestion de la mémoire. L'approche de gestion de la mémoire présentée dans cette thèse est assez naïve dans sa version actuelle. En effet, pour chaque traitement, le minimum de données est chargé en mémoire, et dès qu'une information n'est plus nécessaire, celle-ci est directement sauvegardée sur le disque dur et enlevée de la mémoire. Un tel procédé entraîne un très grand nombre d'échanges entre le disque dur et la mémoire vive, qui peuvent fortement impacter les temps de calcul. Il serait intéressant d'avoir une stratégie directement contrôlée par le pic de consommation mémoire maximum autorisé, et de trouver la meilleure stratégie, à savoir celle minimisant le nombre d'échanges entre la mé-

moire vive et le disque dur, qui respecterait la contrainte imposée. De plus, cela permettrait de paralléliser certaines parties des algorithmes proposés, en autorisant le maximum de traitements parallèles respectant la contrainte de mémoire fixée.

Construction de graphes locaux à partir de données issues d'autres types de capteurs. Nous avons présenté notre approche par graphes locaux, en s'appuyant sur la structure des acquisitions fournies par des LiDARs terrestres. Nous avons montré que lorsque cette structure était présente, elle permettait de construire un modèle de la surface numérisée par morceaux. Premièrement, d'autres périphériques d'acquisition plus grand public (comme les Kinect) fournissent des données sous forme structurée. L'approche proposée durant ce travail devrait pouvoir être directement généralisable à ceux-ci. Aussi, pour généraliser l'approche proposée aux acquisitions ne fournissant pas ce type de structure, la construction des graphes pourrait être réalisée de manière similaire qu'au travail de [CHEN et al., 2018a]. Simplement, plutôt que de construire un seul graphe, il faudrait construire plusieurs graphes de manière locale, en ne considérant qu'un ensemble d'échantillons dans une zone de l'espace donnée pour chaque graphe. Ainsi, en mettant en correspondance les sommets des graphes représentant les mêmes régions, nous pourrions obtenir la même structure que celle présentée dans cette thèse, malgré le manque de structure initial dans la donnée sur laquelle nous nous appuyons actuellement. Nous pourrions donc aussi appliquer l'ensemble des algorithmes présentés tout au long de ce travail sur des nuages de points non structurés.

Echantillonnage optimal. Par ailleurs, nous avons tenté, au travers de l'échantillonnage, de répondre à un problème principalement lié au volume des données produit par les campagnes d'acquisition. Par conséquent, le rayon des disques de Poisson a toujours été défini de manière manuelle, afin de réduire le nombre de points de telle sorte à ce que les échantillonnages résultants puissent être traités par les traitements subséquents. Tout cela fait abstraction des caractéristiques propres à la surface acquise, qui devraient guider le rayon des disques de Poisson, afin de pouvoir préserver les plus fins détails. De ce fait, le rayon des disques de Poisson devrait être déterminé automatiquement, de la même manière que le théorème d'échantillonnage [SHANNON, 1949] permet de déterminer la fréquence de coupure minimum permettant de préserver l'ensemble des fréquences d'un signal. Cependant, aujourd'hui il n'existe pas, à notre connaissance, de méthode permettant de déterminer une fréquence de coupure minimale (un rayon de disque de Poisson minimal) permettant d'échantillonner une surface en préservant tous les détails de celle-ci. Peut-être qu'une méthode telle que celle de [BÉARZI et al., 2018], décomposant une surface de manière locale dans une base de coefficients, pourrait

permettre d'établir un lien avec la densité d'échantillonnage nécessaire pour capturer tous les détails de la surface. Réussir à déterminer automatiquement le rayon des disques de Poisson en fonction des caractéristiques fréquentielles de la surface serait un véritable plus pour des méthodes comme la nôtre.

Ré-échantillonnage de surfaces 3D. La méthode d'échantillonnage proposée a montré de très bons résultats en termes de bruit bleu sur des données synthétiques. Il est évident que l'approche utilisée dans nos expérimentations, consistant à placer des caméras sur la surface d'une sphère centrée sur le modèle, est très rudimentaire. Le but de cette approche était simplement de générer des acquisitions synthétiques, afin de pouvoir analyser la qualité des échantillonnages par rapport à une surface de référence. Cependant, au vu des résultats sur les données synthétiques, nous pensons qu'il serait intéressant de généraliser l'algorithme d'échantillonnage proposé au ré-échantillonnage de modèles 3D. Pour cela, il serait nécessaire de s'intéresser aux travaux de [ALSADIK et al., 2013; OLAGUE et MOHR, 2002], afin de concevoir une méthode capable de déterminer la position et l'orientation des caméras de façon à assurer une couverture totale, et suivant une densité minimale cible, du modèle 3D.

Prise en compte d'attributs additionnels. Au travers de toute cette thèse, nous nous sommes concentrés à la manipulation de la géométrie ayant été capturée par les LiDARs terrestres. Cependant, de nombreuses campagnes d'acquisitions fournissent des données supplémentaires. Par exemple, chaque point peut se voir doté d'une information colorimétrique. La prise en compte de cette information pourrait permettre de mieux préserver la texture de la surface ayant été capturée. Cela pourrait se faire en modifiant la métrique associée au poids des arêtes par rapport au contenu de la texture, pour préserver par exemple les forts gradients. Cependant, l'utilisation de l'information colorimétrique demanderait dans un premier temps d'être uniformisée entre les différentes acquisitions, étant donné que l'apparence des objets varie fortement en fonction des conditions d'illumination et du point de vue choisi lors de la capture.

Recalage d'acquisitions. Les données sur lesquelles nous avons travaillé étaient toutes recalées entre elles, il n'était pas nécessaire de déterminer les matrices de transformation entre les différentes acquisitions. Or encore actuellement, le recalage de nuages de points est un problème ouvert, en particulier quand il s'agit de traiter des données aussi volumineuses que celles considérées dans cette thèse. Nous avons d'ailleurs remarqué que sur les différentes campagnes d'acquisitions certaines acquisitions pouvaient ne pas être correctement recalées. En particulier, lors de l'utilisation de méthodes basées sur l'*ICP* (*Iterative Closest Point* en anglais),

il a été montré que le taux de convergence de ces méthodes dépend fortement du choix des paires de points correspondants ainsi que de la fonctionnelle minimisée [RUSINKIEWICZ et LEVOY, 2001]. Nous pensons donc que l'approche de ré-échantillonnage proposée (en considérant le cas où chaque acquisition est traitée indépendamment) pourrait par exemple fournir une solution intelligente, en lien avec la fonctionnelle à minimiser, pour réduire au maximum la quantité de points à traiter tout en conservant les points les plus pertinents pour mieux guider la mise en correspondance.

Reconstruction de surface semi-régulière. La reconstruction de surfaces polygonaux composées de plusieurs centaines de millions de faces ne permet pas de les manipuler telles quelles. Il serait intéressant de construire un maillage à plusieurs niveaux de résolution en s'inspirant des travaux du côté des *maillages semi-réguliers* [PAYAN et al., 2015]. Des travaux précédents [BOUBEKEUR et al., 2006; PEYROT et al., 2016] ont montré que de tels maillages étaient directement constructibles à partir de données issues d'acquisitions 3D, et que leur construction directe évitait la difficile étape du remaillage semi-régulier. Nous pensons que notre structure pourrait être exploitée pour atteindre cet objectif, en généralisant par exemple les idées proposées par [PEYROT et al., 2016] à la gestion d'un ensemble de triangulations pour la subdivision.

Sixième partie

ANNEXES

DOCUMENTATION COMPLÉMENTAIRE

Cette annexe présente les caractéristiques techniques de la machine sur laquelle l'ensemble des expérimentations ont été effectuées, ainsi que des informations liées aux jeux de données sur lesquels nous avons testé les méthodes proposées.

Caractéristiques matérielles

L'ensemble des simulations ayant été effectuées ont été réalisées sur une machine ayant les caractéristiques suivantes :

- **Processeur** : Intel® Core™ i7-5960X Extreme Edition (3.00 GHz)
- **Carte graphique** : NVIDIA® QUADRO® M5000 (8 Go)
- **Mémoire vive** : 32 Go DDR3
- **Disque dur** : SSD Crucial BX100 (1 To)

Jeux de données

La Table 9 présente l'ensemble des modèles 3D à partir desquels des acquisitions synthétiques ont été générés. Comme nous l'avons expliqué dans la Partie II, à partir d'un modèle 3D, 12 acquisitions sont générées. Pour chacun, nous indiquons le **nom** avec lequel nous y faisons référence tout au long du manuscrit ainsi que le **titulaire** des droits sur chacune d'entre elles.

Nom	Titulaire
Egea	[AIM@SHAPE-VISIONAIR Shape Repository]
Venus	
Neptune	
Armadillo	[The Stanford 3D scanning repository]
Bunny	
Dragon	

TABLE 9 – Modèles 3D à partir desquels des acquisitions synthétiques ont été générées

Dans un second temps, les Tables 10, 11 et 12 présentent l'ensemble des campagnes d'acquisition réelles qui ont été utilisées dans ce travail. Pour chacune d'entre elles, nous indiquons le **nom** avec lequel nous y faisons référence tout au long du manuscrit, l'**origine** géographique de la donnée, le nombre d'**acquisitions** la constituant et le nombre de **points** total de la donnée capturée.

Nom / Origine	Acquisitions	Points (en millions)
<i>St Trophime /</i> Arles (France)	37	307
<i>Templo Mayor /</i> Mexico (Mexique)	17	349
<i>Eim Ya Kyaung /</i> Bagan (Myanmar)	58	1 337
<i>Meeting House /</i> Waitangi (Nouvelle-Zélande)	50	1 493
<i>Villa Arianna /</i> Stabiae (Italie)	123	1 623
<i>Ananda Oak Kyaung /</i> Bagan (Myanmar)	126	1 703
<i>Wat Phra Si Sanphet /</i> Ayutthaya (Thaïlande)	177	5 313

TABLE 10 – Campagnes d’acquisition issues de [*Open Heritage*].

Nom / Origine	Acquisitions	Points (en millions)
<i>Techshop /</i> San Francisco (Californie)	16	114

TABLE 11 – Campagne d’acquisition issue de [*ReCap Community Data Center*].

Nom / Origine	Acquisitions	Points (en millions)
<i>Palais de la Découverte (Facade) /</i> Paris (France)	54	977
<i>Palais de la Découverte (Intérieur) /</i> Paris (France)	37	1 748

TABLE 12 – Campagnes d’acquisition fournies par *Art Graphique&Patrimoine*.

BIBLIOGRAPHIE

- ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D. et SILVA, C. T. (2003). « Computing and rendering point set surfaces ». Dans : *IEEE Transactions on Visualization and Computer Graphics* 9.1, p. 3-15 (cf. p. 57).
- ALSADIK, B., GERKE, M. et VOSSELMAN, G. (2013). « Automated camera network design for 3D modeling of cultural heritage objects ». Dans : *Journal of Cultural Heritage* 14.6, p. 515-526 (cf. p. 140).
- AURENHAMMER, F. (1991). « Voronoi diagrams—a survey of a fundamental geometric data structure ». Dans : *ACM Computing Surveys (CSUR)* 23.3, p. 345-405 (cf. p. 103).
- AURICH, V. et WEULE, J. (1995). « Non-linear gaussian filters performing edge preserving diffusion ». Dans : *Mustererkennung 1995*. Springer, p. 538-545 (cf. p. 70).
- BALZER, M., SCHLÖMER, T. et DEUSSEN, O. (juil. 2009). « Capacity-constrained Point Distributions : A Variant of Lloyd’s Method ». Dans : *ACM Transactions on Graphics* 28.3, 86 :1-86 :8 (cf. p. 59).
- BÉARZI, Y., DIGNE, J. et CHAINE, R. (2018). « Wavejets : A Local Frequency Framework for Shape Details Amplification ». Dans : *Computer Graphics Forum*. T. 37. 2. Wiley Online Library, p. 13-24 (cf. p. 139).
- BELLEKENS, B., SPRUYT, V., BERKVEN, R. et WEYN, M. (2014). « A survey of rigid 3D pointcloud registration algorithms ». Dans : *AMBIENT 2014 : the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, p. 8-13 (cf. p. 11).
- BERGER, M., TAGLIASACCHI, A., SEVERSKY, L., ALLIEZ, P., LEVINE, J., SHARF, A. et SILVA, C. (2014). « State of the art in surface reconstruction from point clouds ». Dans : *EUROGRAPHICS star reports*. T. 1. 1, p. 161-185 (cf. p. 127).
- BIASOTTI, S., GIORGI, D., SPAGNUOLO, M. et FALCIDIENO, B. (2008). « Reeb graphs for shape analysis and applications ». Dans : *Theoretical computer science* 392.1-3, p. 5-22 (cf. p. 27).
- BIASUTTI, P., AUJOL, J.-F., BRÉDIF, M. et BUGEAU, A. (2018). « Range-Image : Incorporating sensor topology for LiDAR point cloud processing ». Dans : *Photogrammetric Engineering & Remote Sensing* 84.6, p. 367-375 (cf. p. 31).
- BOLTCHÉVA, D. et LEVY, B. (2017). « Surface reconstruction by computing restricted Voronoi cells in parallel ». Dans : *Computer-Aided Design* 90, p. 123-134 (cf. p. 91, 92, 127-130, 133).
- BORDIGNON, A., LEWINER, T., LOPES, H., TAVARES, G. et CASTRO, R. (2006). « Point set compression through BSP quantization ». Dans : *Computer Graphics and*

- Image Processing, 2006. SIBGRAP'06. 19th Brazilian Symposium on.* IEEE, p. 229-238 (cf. p. 23).
- BOTSCH, M., WIRATANAYA, A. et KOBBELT, L. (2002). « Efficient high quality rendering of point sampled geometry ». Dans : *Proceedings of the 13th Eurographics workshop on Rendering*. Eurographics Association, p. 53-64 (cf. p. 21-23).
- BOUBEKEUR, T., HEIDRICH, W., GRANIER, X. et SCHLICK, C. (2006). « Volume-surface trees ». Dans : *Computer Graphics Forum*. T. 25. 3. Wiley Online Library, p. 399-406 (cf. p. 141).
- BOWERS, J., WANG, R., WEI, L.-Y. et MALETZ, D. (2010). « Parallel Poisson disk sampling with spectrum analysis on surfaces ». Dans : *ACM Transactions on Graphics* 29.6, p. 166 (cf. p. 59, 60, 83, 84).
- BRÉDIF, M. (2010). « Modélisation 3D de bâtiments : reconstruction automatique de superstructures de toits et recalage cinétique de toits polyédriques prenant en compte la topologie ». Thèse de doct. Télécom ParisTech (cf. p. 12).
- BRIDSON, R. (2007). « Fast Poisson disk sampling in arbitrary dimensions. » Dans : *SIGGRAPH sketches*, p. 22 (cf. p. 60).
- CHEN, J., GE, X., WEI, L.-Y., WANG, B., WANG, Y., WANG, H., FEI, Y., QIAN, K.-L., YONG, J.-H. et WANG, W. (2013). « Bilateral blue noise sampling ». Dans : *ACM Transactions on Graphics* 32.6, p. 216 (cf. p. 58).
- CHEN, S., TIAN, D., FENG, C., VETRO, A. et KOVAČEVIĆ, J. (2018a). « Fast Resampling of Three-Dimensional Point Clouds via Graphs ». Dans : *IEEE Transactions on Signal Processing* 66.3, p. 666-681 (cf. p. 26, 27, 30, 37, 55, 58, 139).
- CHEN, Z., ZHANG, T., CAO, J., ZHANG, Y. J. et WANG, C. (2018b). « Point cloud resampling using centroidal Voronoi tessellation methods ». Dans : *Computer-Aided Design* 102, p. 12-21 (cf. p. 56).
- CLINE, D., JESCHKE, S., WHITE, K., RAZDAN, A. et WONKA, P. (2009). « Dart throwing on surfaces ». Dans : *Computer Graphics Forum*. T. 28. 4. Wiley Online Library, p. 1217-1226 (cf. p. 60).
- COOK, R. L. (jan. 1986). « Stochastic Sampling in Computer Graphics ». Dans : *ACM Transactions on Graphics* 5.1, p. 51-72 (cf. p. 59).
- CORSINI, M., CIGNONI, P. et SCOPIGNO, R. (2012). « Efficient and flexible sampling with blue noise properties of triangular meshes ». Dans : *IEEE Transactions on Visualization and Computer Graphics* 18.6, p. 914-924 (cf. p. 60).
- CRANE, K., DE GOES, F., DESBRUN, M. et SCHRÖDER, P. (2013). « Digital geometry processing with discrete exterior calculus ». Dans : *ACM SIGGRAPH 2013 Courses*. ACM, p. 7 (cf. p. 68).
- CURLESS, B. et SEITZ, S. (2000). « 3D Photography ». Dans : *Course Notes for SIGGRAPH 2000* (cf. p. 6).
- DACHSBACHER, C., VOGELGSANG, C. et STAMMINGER, M. (2003). « Sequential point trees ». Dans : *ACM Transactions on Graphics*. T. 22. 3. ACM, p. 657-662 (cf. p. 22).

- DESCARTES, R. (1644). *Principia philosophiae (1644)* (cf. p. 99, 100).
- DIJKSTRA, E. W. (déc. 1959). « A Note on Two Problems in Connexion with Graphs ». Dans : *Numer. Math.* 1.1, p. 269-271 (cf. p. 39, 40, 65).
- DURAND, F. et DORSEY, J. (2002). « Fast bilateral filtering for the display of high-dynamic-range images ». Dans : *ACM Transactions on Graphics*. T. 21. 3. ACM, p. 257-266 (cf. p. 71).
- ELSEBERG, J., BORRMANN, D. et NÜCHTER, A. (2013). « One billion points in the cloud—an octree for efficient processing of 3D laser scans ». Dans : *ISPRS Journal of Photogrammetry and Remote Sensing* 76, p. 76-88 (cf. p. 24).
- FOREST COLLADO, J. et al. (2004). « New methods for triangulation-based shape acquisition using laser scanners ». Thèse de doct. (cf. p. 8).
- FORTUNE, S. (1987). « A sweepline algorithm for Voronoi diagrams ». Dans : *Algorithmica* 2.1-4, p. 153 (cf. p. 108).
- FURUKAWA, Y., HERNÁNDEZ, C. et al. (2015). « Multi-view stereo : A tutorial ». Dans : *Foundations and Trends® in Computer Graphics and Vision* 9.1-2, p. 1-148 (cf. p. 7).
- GAMITO, M. N. et MADDOCK, S. C. (2009). « Accurate multidimensional Poisson-disk sampling ». Dans : *ACM Transactions on Graphics* 29.1, p. 8 (cf. p. 59).
- GARLAND, M. et HECKBERT, P. S. (1997). « Surface simplification using quadric error metrics ». Dans : *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., p. 209-216 (cf. p. 57).
- GOBBETTI, E. et MARTON, F. (2004). « Layered point clouds : a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models ». Dans : *Computers & Graphics* 28.6, p. 815-826 (cf. p. 23).
- GUMHOLD, S., KAMI, Z., ISENBURG, M. et SEIDEL, H.-P. (2005). « Predictive point-cloud compression ». Dans : *ACM SIGGRAPH 2005 Sketches*. ACM, p. 137 (cf. p. 23).
- GUSKOV, I. (2007). « Manifold-based approach to semi-regular remeshing ». Dans : *Graphical Models* 69.1, p. 1-18 (cf. p. 133).
- HECK, D., SCHLÖMER, T. et DEUSSEN, O. (2013). « Blue noise sampling with controlled aliasing ». Dans : *ACM Transactions on Graphics* 32.3, p. 25 (cf. p. 60).
- HECKBERT, P. S. et GARLAND, M. (1997). *Survey of polygonal surface simplification algorithms*. Rapp. tech. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science (cf. p. 56).
- HUANG, H., LI, D., ZHANG, H., ASCHER, U. et COHEN-OR, D. (2009). « Consolidation of unorganized point clouds for surface reconstruction ». Dans : *ACM Transactions on Graphics* 28.5, p. 176 (cf. p. 58).
- HUANG, H., WU, S., GONG, M., COHEN-OR, D., ASCHER, U. et ZHANG, H. R. (2013). « Edge-aware point set resampling ». Dans : *ACM Transactions on Graphics* 32.1, p. 9 (cf. p. 58).

- HUANG, Y., PENG, J., KUO, C.-C. J. et GOPI, M (2008). « A generic scheme for progressive point cloud coding ». Dans : *IEEE Transactions on Visualization and Computer Graphics* 14.2, p. 440-453 (cf. p. 23).
- JACOBSON, A., PANOZZO, D. et al. (2017). *libigl : A simple C++ geometry processing library*. <http://libigl.github.io/libigl/> (cf. p. 39).
- KAMMERL, J., BLODOW, N., RUSU, R. B., GEDIKLI, S., BEETZ, M. et STEINBACH, E. (2012). « Real-time compression of point cloud streams ». Dans : *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, p. 778-785 (cf. p. 23).
- KARCHER, H. (1977). « Riemannian center of mass and mollifier smoothing ». Dans : *Communications on pure and applied mathematics* 30.5, p. 509-541 (cf. p. 115).
- KARCHER, H. (2014). « Riemannian center of mass and so called karcher mean ». Dans : *arXiv preprint arXiv :1407.2087* (cf. p. 115).
- KAZHDAN, M. et HOPPE, H. (juil. 2013). « Screened Poisson Surface Reconstruction ». Dans : *ACM Transactions on Graphics* 32.3, 29 :1-29 :13 (cf. p. 76, 122).
- KHODAKOVSKY, A., LITKE, N. et SCHRÖDER, P. (2003). « Globally smooth parameterizations with low distortion ». Dans : *ACM Transactions on Graphics* 22.3, p. 350-357 (cf. p. 34).
- KOLB, A., BARTH, E., KOCH, R. et LARSEN, R. (2010). « Time-of-flight cameras in computer graphics ». Dans : *Computer Graphics Forum*. T. 29. 1. Wiley Online Library, p. 141-159 (cf. p. 8).
- KÜHN, U. (1998). « Local calculation of Voronoi diagrams ». Dans : *Information processing letters* 68.6, p. 307-312 (cf. p. 105, 106).
- LAGAE, A. et DUTRÉ, P. (2008). « A comparison of methods for generating Poisson disk distributions ». Dans : *Computer Graphics Forum*. T. 27. 1. Wiley Online Library, p. 114-129 (cf. p. 83).
- LARI, Z, HABIB, A et KWAK, E (2011). « An adaptive approach for segmentation of 3D laser point cloud ». Dans : *ISPRS Workshop laser scanning*, p. 29-31 (cf. p. 23).
- LEJEUNE DIRICHLET, G. (1850). « Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. » ger. Dans : *Journal für die reine und angewandte Mathematik* 40, p. 209-227 (cf. p. 100).
- LEVIN, D. (2004). « Mesh-independent surface interpolation ». Dans : *Geometric modeling for scientific visualization*. Springer, p. 37-49 (cf. p. 70).
- LI, H., WEI, L.-Y., SANDER, P. V. et FU, C.-W. (2010). « Anisotropic blue noise sampling ». Dans : *ACM Transactions on Graphics* 29.6, p. 167 (cf. p. 59, 60).
- LIAO, B., XIAO, C., JIN, L. et FU, H. (2013). « Efficient feature-preserving local projection operator for geometry reconstruction ». Dans : *Computer-Aided Design* 45.5, p. 861-874 (cf. p. 58).

- LIPMAN, Y., COHEN-OR, D., LEVIN, D. et TAL-EZER, H. (2007). « Parameterization-free projection for geometry reconstruction ». Dans : *ACM Transactions on Graphics* 26.3, p. 22 (cf. p. 58).
- LLOYD, S. (1982). « Least squares quantization in PCM ». Dans : *IEEE Transactions on Information Theory* 28.2, p. 129-137 (cf. p. 103).
- LUEBKE, D. P. (2001). « A Developer's Survey Of Polygonal Simplification Algorithms ». Dans : *IEEE Computer Graphics and Applications* 3, p. 24-35 (cf. p. 56).
- MAGLO, A., LAVOUÉ, G., DUPONT, F. et HUDELLOT, C. (fév. 2015). « 3D Mesh Compression : Survey, Comparisons, and Emerging Trends ». Dans : *ACM Comput. Surv.* 47.3, 44 :1-44 :41 (cf. p. 56).
- MERRY, B., MARAIS, P. et GAIN, J. (2006). « Compression of dense and regular point clouds ». Dans : *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. ACM, p. 15-20 (cf. p. 23).
- MIAO, Y., PAJAROLA, R. et FENG, J. (2009). « Curvature-aware adaptive re-sampling for point-sampled geometry ». Dans : *Computer-Aided Design* 41.6, p. 395-403 (cf. p. 57).
- MITCHELL, J. S., MOUNT, D. M. et PAPADIMITRIOU, C. H. (1987). « The discrete geodesic problem ». Dans : *SIAM Journal on Computing* 16.4, p. 647-668 (cf. p. 39, 41).
- OCHOTTA, T. et SAUPE, D. (2004). « Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields ». Dans : *Proceedings of the First Eurographics conference on Point-Based Graphics*. Eurographics Association, p. 103-112 (cf. p. 25, 26).
- OLAGUE, G. et MOHR, R. (2002). « Optimal camera placement for accurate reconstruction ». Dans : *Pattern Recognition* 35.4, p. 927-944 (cf. p. 140).
- ÖZTIRELI, A. C., ALEXA, M. et GROSS, M. (2010). « Spectral sampling of manifolds ». Dans : *ACM Transactions on Graphics*. T. 29. 6. ACM, p. 168 (cf. p. 58).
- PAJAROLA, R. (2003). « Efficient Level-of-details for Point based Rendering. » Dans : *Computer Graphics and Imaging*, p. 141-146 (cf. p. 23).
- PAJAROLA, R., SAINZ, M. et LARIO, R. (2005). « Xsplat : External memory multiresolution point visualization ». Dans : *Proceedings IASTED International Conference on Visualization, Imaging and Image Processing*, p. 628-633 (cf. p. 23).
- PARK, M. K., LEE, S. J. et LEE, K. H. (2012). « Multi-scale tensor voting for feature extraction from unstructured point clouds ». Dans : *Graphical Models* 74.4, p. 197-208 (cf. p. 68).
- PAULY, M. et GROSS, M. (2001). « Spectral processing of point-sampled geometry ». Dans : *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, p. 379-386 (cf. p. 25, 26).

- PAULY, M., GROSS, M. et KOBBELT, L. P. (2002). « Efficient simplification of point-sampled surfaces ». Dans : *Proceedings of the conference on Visualization'02*. IEEE Computer Society, p. 163-170 (cf. p. 56).
- PAYAN, F., ROUDET, C. et SAUVAGE, B. (2015). « Semi-Regular Triangle Remeshing : A Comprehensive Study ». Dans : *Computer Graphics Forum* 34. Sous la dir. de B. PUBLISHING, p. 86-102 (cf. p. 141).
- PENG, J. et KUO, C.-C. J. (2005). « Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition ». Dans : *ACM Transactions on Graphics*. T. 24. 3. ACM, p. 609-616 (cf. p. 56).
- PEYRÉ, G. (2008). « Numerical Mesh Processing ». Dans : (cf. p. 27).
- PEYROT, J., PAYAN, F. et ANTONINI, M. (2016). « From stereoscopic images to semi-regular meshes ». Dans : *Signal Processing : Image Communication* 40, p. 97 -110 (cf. p. 141).
- PEYROT, J.-L. (déc. 2014). « 3D digitization optimization : from surface to semi-regular mesh ». Theses. Université Nice Sophia Antipolis (cf. p. 8, 63, 113, 127).
- PEYROT, J.-L., PAYAN, F. et ANTONINI, M. (2015). « Direct Blue Noise resampling of meshes of arbitrary topology ». Dans : *The Visual Computer* 31.10. Sous la dir. de SPRINGER, p. 1365-1381 (cf. p. 59, 60, 63, 65, 66).
- PIERACCINI, M., GUIDI, G. et ATZENI, C. (2001). « 3D digitizing of cultural heritage ». Dans : *Journal of Cultural Heritage* 2.1, p. 63-70 (cf. p. 13).
- PIETRONI, N., TARINI, M., SORKINE, O. et ZORIN, D. (2011). « Global parametrization of range image sets ». Dans : *ACM Transactions on Graphics*. T. 30. 6. ACM, p. 149 (cf. p. 30, 39, 40).
- PREINER, R., MATTAUSCH, O., ARIKAN, M., PAJAROLA, R. et WIMMER, M. (2014). « Continuous projection for fast L1 reconstruction. » Dans : *ACM Transactions on Graphics* 33.4, p. 47-1 (cf. p. 58).
- RIVEST, J.-F., SOILLE, P. et BEUCHER, S. (1993). « Morphological gradients. » Dans : *J. Electronic Imaging* 2.4, p. 326-336 (cf. p. 31).
- RUSINKIEWICZ, S. et LEVOY, M. (2000). « QSpIat : A multiresolution point rendering system for large meshes ». Dans : *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., p. 343-352 (cf. p. 21, 22).
- RUSINKIEWICZ, S. et LEVOY, M. (2001). « Efficient variants of the ICP algorithm ». Dans : *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, p. 145-152 (cf. p. 141).
- SCHEIBLAUER, C. (2014). « Interactions with Gigantic Point Clouds ». Thèse de doct. Favoritenstrasse 9-11/186, A-1040 Vienna, Austria : Institute of Computer Graphics et Algorithms, Vienna University of Technology (cf. p. 20).

- SCHMIDT, R., GRIMM, C. et WYVILL, B. (2006). « Interactive decal compositing with discrete exponential maps ». Dans : *ACM Transactions on Graphics*. T. 25. 3. ACM, p. 605-613 (cf. p. 81).
- SCHNABEL, R. et KLEIN, R. (2006). « Octree-based Point-Cloud Compression. » Dans : *Symposium on Point-Based Graphics*, p. 111-120 (cf. p. 23).
- SCHÜTZ, M. (2016). « Potree : Rendering Large Point Clouds in Web Browsers ». Mém. de mast. Favoritenstrasse 9-11/186, A-1040 Vienna, Austria : Institute of Computer Graphics et Algorithms, Vienna University of Technology (cf. p. 20).
- SHANNON, C. E. (1949). « Communication in the presence of noise ». Dans : *Proceedings of the IRE* 37.1, p. 10-21 (cf. p. 139).
- SHEFFER, A., PRAUN, E., ROSE, K. et al. (2007). « Mesh parameterization methods and their applications ». Dans : *Foundations and Trends® in Computer Graphics and Vision* 2.2, p. 105-171 (cf. p. 27).
- SHEPARD, D. (1968). « A two-dimensional interpolation function for irregularly-spaced data ». Dans : *Proceedings of the 1968 23rd ACM national conference*. ACM, p. 517-524 (cf. p. 57).
- SHI, B.-Q., LIANG, J. et LIU, Q. (2011). « Adaptive simplification of point cloud using k-means clustering ». Dans : *Computer-Aided Design* 43.8, p. 910-922 (cf. p. 57).
- SHUMAN, D. I., NARANG, S. K., FROSSARD, P., ORTEGA, A. et VANDERGHEYNST, P. (2013). « The emerging field of signal processing on graphs : Extending high-dimensional data analysis to networks and other irregular domains ». Dans : *IEEE Signal Processing Magazine* 30.3, p. 83-98 (cf. p. 26).
- SMITH, S. M. et BRADY, J. M. (1997). « SUSAN—a new approach to low level image processing ». Dans : *International journal of computer vision* 23.1, p. 45-78 (cf. p. 70).
- SONG, H. et FENG, H.-Y. (2008). « A global clustering approach to point cloud simplification with a specified data reduction ratio ». Dans : *Computer-Aided Design* 40.3, p. 281-292 (cf. p. 57).
- TAM, G. K., CHENG, Z.-Q., LAI, Y.-K., LANGBEIN, F. C., LIU, Y., MARSHALL, D., MARTIN, R. R., SUN, X.-F. et ROSIN, P. L. (2013). « Registration of 3D point clouds and meshes : a survey from rigid to nonrigid. » Dans : *IEEE Transactions on Visualization and Computer Graphics* 19.7, p. 1199-1217 (cf. p. 11).
- TIERNY, J. (2008). « Reeb graph based 3D shape modeling and applications ». Thèse de doct. Université des Sciences et Technologie de Lille-Lille I (cf. p. 27).
- TOMASI, C. et MANDUCHI, R. (1998). « Bilateral filtering for gray and color images ». Dans : *Computer Vision, 1998. Sixth International Conference on*. IEEE, p. 839-846 (cf. p. 70).
- ULICHNEY, R. (1987). *Digital halftoning*. MIT press (cf. p. 59).
- VORONOÏ, G. (1908). « Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélo-

èdres primitifs. » Dans : *Journal für die reine und angewandte Mathematik* 134, p. 198-287 (cf. p. 100).

WAND, M., BERNER, A., BOKELOH, M., JENKE, P., FLECK, A., HOFFMANN, M., MAIER, B., STANEKER, D., SCHILLING, A. et SEIDEL, H.-P. (2008). « Processing and interactive editing of huge point clouds from 3D scanners ». Dans : *Computers & Graphics* 32.2, p. 204-220 (cf. p. 24).

WEI, L.-Y. et WANG, R. (2011). « Differential domain analysis for non-uniform sampling ». Dans : *ACM Transactions on Graphics* 30, 50 :1-50 :10 (cf. p. 80-82, 122).

YU, Z., WONG, H.-S., PENG, H. et MA, Q. (2010). « ASM : An adaptive simplification method for 3D point-based models ». Dans : *Computer-Aided Design* 42.7, p. 598-612 (cf. p. 57).

ZWICKER, M., PFISTER, H., VAN BAAR, J. et GROSS, M. (2001). « Surface splatting ». Dans : *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, p. 371-378 (cf. p. 20).

BASES DE DONNÉES

AIM@SHAPE-VISIONAIR. *AIM@SHAPE-VISIONAIR Shape Repository*. URL : <http://visionair.ge.imati.cnr.it> (cf. p. 145).

AUTODESK. *ReCap Community Data Center*. URL : <https://autodesk.app.box.com/v/recapdata> (cf. p. 146).

GOOGLE/CYARK. *Open Heritage*. URL : <https://artsandculture.google.com/project/cyark> (cf. p. 14, 146).

LEVOY, M., GERTH, J, CURLESS, B et PULL, K. *The Stanford 3D scanning repository*. URL : <http://graphics.stanford.edu/data/3Dscanrep> (cf. p. 145).

IMAGES

- FARO® (2011). *FARO Laser Scanner Focus3D®Manual*. URL : https://doarch332.files.wordpress.com/2013/11/e866_faro_laser_scanner_focus3d_manual_en.pdf (cf. p. 9).
- GOOGLE/CYARK. *Open Heritage*. URL : <https://artsandculture.google.com/project/cyark> (cf. p. 14, 146).
- HFEREE (2011). *A Delaunay triangulation and its Voronoi diagram*. URL : https://commons.wikimedia.org/wiki/File:Delaunay_Voronoi.svg (cf. p. 102).
- MACKINTOSH, G. (2011). *Honey Comb Structure*. URL : [https://commons.wikimedia.org/wiki/File:Honeycomb_structure_\(6248780733\).jpg](https://commons.wikimedia.org/wiki/File:Honeycomb_structure_(6248780733).jpg) (cf. p. 99).
- NONENMACHER, R. A. (2003). *Anisoptera (Dragonfly) wings, retrieved from a swimming pool in the Town of Skaneateles, Onondaga County, New York. Likely from an Eastern Pondhawk (Erythemis simplicicollis)*. URL : https://commons.wikimedia.org/wiki/File:Dragonfly_ran-004.jpg (cf. p. 99).
- PASSIVHAUS INSTITUT (2006). *Thermogram of a Passivhaus building*. URL : https://commons.wikimedia.org/wiki/File:Passivhaus_thermogram_gedaemmt_ungedaemmt.png (cf. p. 7).
- U.S. ARMY (1947-1955). *ENIAC (Electronic Numerical Integrator And Computer) in Philadelphia, Pennsylvania. Glen Beck (background) and Betty Snyder (foreground) program the ENIAC in building 328 at the Ballistic Research Laboratory (BRL)*. URL : <https://commons.wikimedia.org/wiki/File:Eniac.jpg> (cf. p. 3).
- WHITETIMBERWOLF (2010). *Schematic drawing of an octree, a data structure of computer science*. URL : <https://commons.wikimedia.org/wiki/File:Octree2.svg> (cf. p. 23).