HAL
open science

Rééchantillonnage des nuages de points gigantesques produits lors de la numérisation 3D de monuments

Arnaud Bletterer, Frédéric Payan, Marc Antonini

To cite this version:

Arnaud Bletterer, Frédéric Payan, Marc Antonini. Rééchantillonnage des nuages de points gigantesques produits lors de la numérisation 3D de monuments. Groupe de Recherche et d'Etudes du Traitement du Signal et des Images (GRETSI), Aug 2019, Lille, France. hal-02147754v1

HAL Id: hal-02147754

https://hal.science/hal-02147754v1

Submitted on 5 Jun 2019 (v1), last revised 17 Jun 2019 (v2)

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.

Rééchantillonnage des nuages de points gigantesques produits lors de la numérisation 3D de monuments

Arnaud BLETTERER, Frédéric PAYAN, Marc ANTONINI

Université Côte d'Azur, CNRS, I3S
2000, Route des lucioles 06903 Sophia Antipolis
arnaud.bletterer, frederic.payan, marc.antonini@univ-cotedazur.fr

Résumé – Nous présentons une nouvelle méthode pour rééchantillonner des nuages de points générés par agrégation d'acquisitions LiDAR, à l'aide d'un échantillonnage en disques de Poisson. Celle-ci s'appuie sur un ensemble de graphes locaux représentant la connectivité locale de la surface sous-jacente au nuage, mis en correspondance pour assurer une cohérence globale lors de l'échantillonnage. Cette approche nous permet d'améliorer la qualité des numérisations 3D gigantesques telles que celles produites par les projets de préservation du patrimoine mondial, même si le volume initial des données dépasse très largement la capacité mémoire de la machine sur laquelle ces calculs sont effectués.

Abstract – We present an original method for resampling point clouds provided by aggregation of LiDAR acquisitions, using a Poisson-disk sampling. It is based on a set of local graphs connected together to get a global representation of the acquired site. This approach allows us to improve the quality of gigantic 3D scans such as the ones generated for cultural heritage preservation, even if the initial volume of data largely exceed the memory capacity of the machine on which those processings are performed.

1 Introduction

De nombreux projets numérisent des sites remarquables à travers le monde, à des fins de sauvegarde et de partage du patrimoine culturel, par exemple [**?**]. Pour des sites étendus, des centaines d'acquisitions LiDAR sont nécessaires, et une agrégation naïve produit des nuages contenant des milliards de points 3D, inexploitables tels quels par la majorité des machines actuelles à cause du volume des données. De plus, ces nuages ont une densité d'échantillonnage hétérogènes, principalement due à l'étendue des sites numérisés et au fait que certaines zones, dites *de recouvrement*, sont capturées par plusieurs acquisitions. Le rééchantillonnage est *a priori* une solution, à la fois pour réduire le nombre de points, mais aussi pour améliorer la qualité globale de la distribution de ces points sur la surface numérisée. Mais cela s'avère complexe avec ce type de données. D'une part, les nuages de points sont des données non structurées, ce qui complique de nombreuses opérations basiques puisque la notion de voisinage entre échantillons y est absente. D'autre part, le volume global de ces données dépasse les capacités mémoire de la plupart des machines actuelles, ce qui prohibe l'utilisation d'algorithmes "in-core". Pour structurer des nuages de points, des arbres de partitionnement 3D (ou *octrees*) sont souvent utilisés. Efficaces pour compresser ou visualiser les données [**?**, **?**], ils s'avèrent inadaptés dans notre contexte car ils structurent l'espace ambiant du nuage de points, et non pas la surface sous-jacente que l'on souhaite rééchantillonner. Récemment, une structure basée graphe [**?**] a été proposée. Elle a l'avantage de "capturer" la topologie de la surface numérisée, et permet donc l'application d'algorithmes "classiques" de traitement du signal, comme le rééchantillonnage. Cependant, la passage à l'échelle reste problématique, dû à la complexité mémoire d'une telle représentation. Dans cet article, nous proposons une méthode originale pour rééchantillonner les nuages de points issus de campagnes de numérisation 3D de sites étendus. Notre algorithme génère un échantillonnage en disques de Poisson [**?**] en s'appuyant sur une structure composée de graphes *locaux*, construits à partir des différentes acquisitions LiDAR. Chaque graphe fournit une représentation *structurée* d'une partie de la surface numérisée, que nous pouvons rééchantillonner en n'atteignant pas des consommations mémoire rédhibitoires. Pour assurer des distributions globalement satisfaisantes malgré un traitement séquentiel des graphes, les sommets des graphes décrivant les zones de recouvrement sont mis en correspondance afin de "propager" les disques construits d'un graphe à l'autre. Nous montrons comment cette approche permet d'améliorer la qualité des numérisations 3D de sites historiques issus du patrimoine mondial (voir figure 1), même si le volume des données d'entrée dépasse largement la capacité mémoire de la machine utilisée. Notre structuration permet en effet de traiter des nuages contenant des milliards de points, quel que soit le nombre d'acquisitions et de points générés par acquisition.

2 Rééchantillonnage d'une acquisition

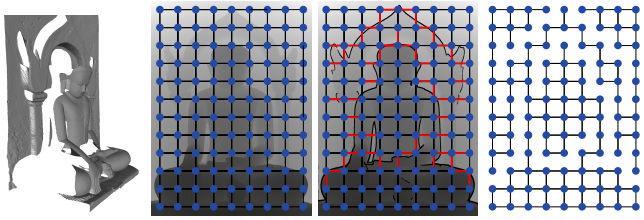
Pour générer un échantillonnage en disques de Poisson, nous avons retenu la technique du *lancer de fléchettes* (ou *dart thro-*

Ce travail a été financé par une bourse doctorale de la *Région Sud*.

FIGURE 1 – Wat Phra Si Sanphet, à Ayutthaya (Thaïlande). 40 000m^2 rééchantillonnés avec notre méthode. La donnée initiale est composée de plus de 5 milliards de points provenant de 156 acquisitions. Cette version simplifiée contient environ 155 millions de points (2,8%), et préserve malgré tout les détails fins.

wing) [**?**]. Etant donnée une distance minimale $2r$ que l'on souhaite imposer entre 2 échantillons sur un domaine donné, il suffit de i) tirer aléatoirement un échantillon parmi les échantillons candidats du domaine, ii) construire un disque de rayon r autour de lui, et iii) vérifier si ce disque n'intersecte aucun disque précédemment positionné. Si c'est le cas, on conserve ce candidat dans la distribution finale, et on répète le processus jusqu'à ce que l'on ne puisse plus placer d'échantillons dans le domaine.

Appliquer cet algorithme sur un nuage de points n'est pas trivial, car la donnée est non structurée. Nous n'avons donc aucune information sur la topologie de la surface sous-jacente, ce qui empêche la construction des disques sur celle-ci en utilisant la méthode de [**?**].



(a) Nuage 3D. (b) Graphe initial. (c) Segmentation. (d) Graphe final.

FIGURE 2 – Génération d'un graphe à partir d'une carte.

Pour contourner ce problème, nous travaillons sur la carte de profondeur D générée lors d'une acquisition LiDAR, qui fournit une représentation structurée du nuage capturé depuis cette position (figure 2(a)). A partir de cette carte, nous pouvons contruire un graphe $G = (V, E)$ (figure 2(b)), où les sommets V correspondent aux pixels de la carte, et dont les arêtes E relient les pixels voisins, fournissant ainsi une connectivité entre les points du nuage. Dans ce graphe ne sont conservés que les pixels d'intensité non nulle, correspondant à des points réellement acquis. De plus, nous atténuons le bruit d'acquisition à l'aide d'un filtrage bilatéral [**?**] appliqué directement sur la carte (figure 3).

Enfin, il faut gérer les occultations. En effet, dans une carte de profondeur, des pixels voisins ne sont pas obligatoirement associés à des points 3D proches sur la surface numérisée. Il faut donc les déconnecter dans le graphe (figure 2(c)). Pour cela, la méthode des gradients morphologiques [**?**] est utilisée

FIGURE 3 – Débruitage à l'aide d'un filtrage bilatéral appliqué sur les cartes de profondeur (modèle *St Strophime*).

afin de détecter les fortes variations d'intensité dans la carte de profondeur, et supprimer les arêtes correspondantes. Un graphe décrivant la surface numérisée est ainsi créé (voir figure 2(d)). Il peut être vu comme un domaine de paramétrisation discret que nous utilisons pour réaliser le lancer de fléchettes sur la surface. Pour calculer la surface couverte par les disques construits autour des échantillons, nous avons utilisé l'algorithme du plus court chemin de Dijkstra [**?**]. Les disques sont donc déterminés par propagation le long des arêtes du graphe. De plus, en utilisant la géométrie 3D des points associés aux sommets lors de la propagation, le plus court chemin sur le graphe représente *in fine* le plus court chemin géodésique sur la surface numérisée.

3 Rééchantillonnage d'un ensemble d'acquisitions agrégées

Dans le cas d'un nuage défini par plusieurs acquisitions, cette méthode ne peut pas être appliquée indépendamment sur toutes les cartes à cause des zones de recouvrement. En effet ces zones communes seraient échantillonnées sur différents graphes, et les disques construits se superposeraient inévitablement sur la surface. La contrainte sur les disques ne serait donc plus respectée, réduisant drastiquement la qualité de l'échantillonnage dans ces zones.

Pour pallier ce problème, nous avons choisi de traiter les acquisitions successivement, mais en transmettant après chaque échantillonnage d'un graphe les disques positionnés dans les zones de recouvrement aux autres graphes décrivant aussi ces zones. Le principe général est illustré par la figure 4.

Pour ce faire, il faut définir des *fonctions de transition* entre les graphes. Une carte de profondeur D peut être vue comme une paramétrisation d'un nuage de point NP : $D = c(NP)$, où c est la fonction de paramétrisation. Cette fonction est inversible : $NP = c^{-1}(D)$, c^{-1} étant la fonction de *projection*.

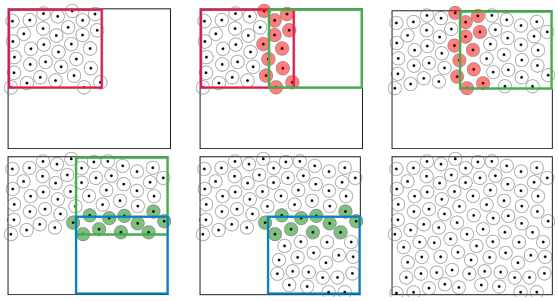


FIGURE 4 – Principe général de notre approche. Chaque couleur identifie une zone couverte par un graphe donné.

La fonction de transition d'une carte i à une carte j est donc simplement définie par $\tau_{ij} = c_j \circ c_i^{-1}$. A partir de ces fonctions, nous pourrions générer une unique paramétrisation globale [**?**]. Mais puisque nous souhaitons traiter les cartes les unes après les autres afin de limiter la consommation mémoire, nous avons décidé de *connecter* les sommets des graphes présents dans les zones de recouvrement sous la forme de données complémentaires à chaque graphe. Ainsi, pour un graphe donné, chaque sommet v possède une liste de *sommets correspondants* C^v (établie à l'aide des fonctions de transition) contenant l'ensemble des sommets des autres graphes qui représentent le même point p dans l'espace 3D. Ainsi, lorsqu'un graphe a été échantillonné, on transfère aux sommets correspondants leur éventuelle appartenance à un disque venant d'être construit, pour en tenir compte lors de l'échantillonnage des graphes suivants. L'échantillonnage se fait donc localement, limitant la consommation mémoire, sans pour autant sacrifier la cohérence globale de la distribution.

Dans les zones de recouvrement, le nombre de sommets de chaque graphe est très variable (selon la position et la résolution du capteur lors des acquisitions), et sa densité va fortement influencer la précision du calcul du plus court chemin lors de la construction des disques. Au stade de la mise en correspondance des graphes, nous déterminons donc aussi pour chaque graphe G l'ensemble V^+ des sommets présents dans des zones plus densément échantillonnées que leurs sommets correspondants dans les autres graphes. Au final, pour chaque graphe, le lancer de fléchettes considèrera uniquement cet ensemble V^+ comme échantillons candidats.

4 Résultats expérimentaux

En premier lieu, nous évaluons la qualité des distributions générées par notre algorithme, pour vérifier que notre approche par graphes locaux donne des résultats similaires à une approche globale et directe sur une surface. Pour cela, nous avons généré des cartes de profondeur synthétiques à partir de maillages surfaciques, comme dans [**?**]. Tous nos résultats ont été générés avec un Intel Core i7@3.00GHz, 32Go de RAM, et un DD de 1To. La figure 5 présente un résultat visuel d'une distribution obtenue sur le modèle *Dragon* à partir de 12 cartes de profondeur acquises régulièrement autour de l'objet. On observe des distributions globalement satisfaisantes. Pour confirmer cela de manière quantitative, nous avons effectué une analyse différentielle [**?**] de nos résultats. La figure 6 montre que les caractéristiques de nos distributions sont similaires avec celles obtenues avec [**?**], qui travaille pourtant directement sur les maillages surfaciques complets. Ceci est très positif, car cela tend à montrer que notre méthode n'est pas pénalisée par une approche par "morceaux", et que la connectivité inférée est bien cohérente avec celle de la surface originale.

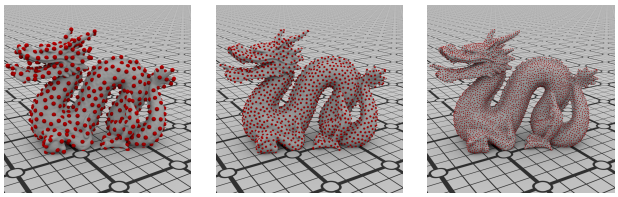
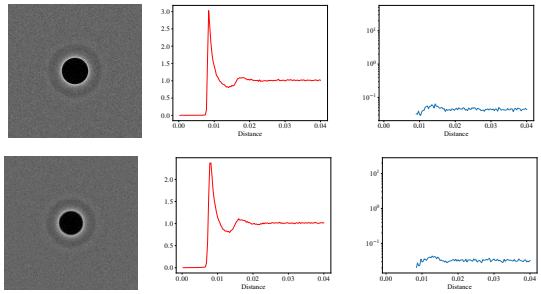


FIGURE 5 – Distributions uniformes générées par notre approche sur *Dragon* à 3 différentes densités cibles.



(a) (b) SPMR (c) Anisotropie

FIGURE 6 – Qualité des distributions obtenues sur *Dragon* avec [**?**] (en haut) et avec notre méthode (en bas) pour $r = 0.004$.

Nous présentons maintenant des résultats obtenus sur des données réelles (propriété de *CyArk/Google Open Heritage Program* [**?**] et de *Art Graphique et Patrimoine*). Ces données sont constituées de plusieurs dizaines voire centaines d'acquisitions, représentant des milliards de points numérisés.

La figure 1 montre *Wat Phra Si Sanphet* (5,313 milliards de points, 156 acquisitions, coût de stockage de plus de $600Gio$), rééchantillonné avec "seulement" 155 millions de points. Nous pouvons déjà noter que notre algorithme est capable de traiter des données dont le volume global dépasse très largement la capacité mémoire de la machine utilisée. De plus, grâce à un échantillonnage adaptatif prenant en compte la courbure de la surface lors de la construction des disques (en pondérant les arêtes des graphes par une fonctionnelle dépendant de la courbure), nous sommes capables de préserver des détails très fins, tels que des briques, malgré l'étendue du site. La figure 7 confirme l'intéret d'utiliser un échantillonnage adaptatif, en comparant la distribution obtenue sur *Grand Palais* de manière uniforme ou adaptative, à densité similaire. Cet exemple montre clairement l'impact sur la qualité des nuages générés, ce qui est intéressant dans un contexte de visualisation.

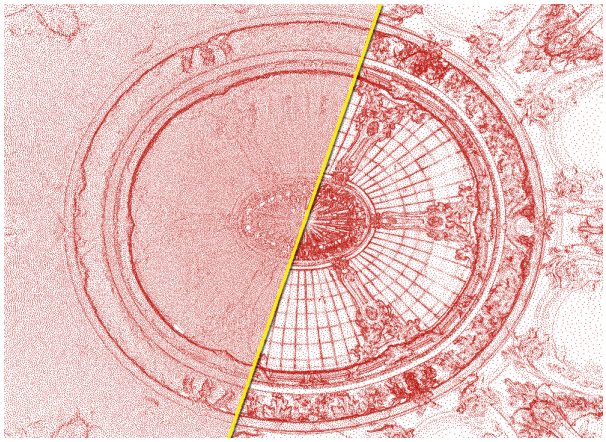


FIGURE 7 – Verrière du *Grand Palais* (2 milliards de points, 35 acquisitions) rééchantillonné avec seulement 0,15% de points. Gauche/droite : distribution uniforme/adaptative.

Pour attester définitivement de l'intérêt de notre méthode, la figure 8 montre une surface reconstruite avec [**?**] à partir d'une distribution générée avec notre algorithme. La reconstruction obtenue est très satisfaisante, notamment car notre méthode considère la topologie de la surface sous-jacente et non pas l'espace ambiant comme les méthodes basées sur un partitionnement spatial.

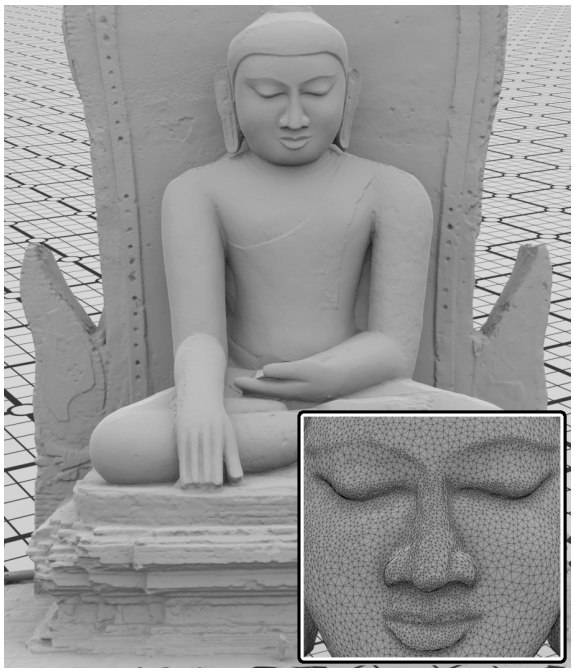


FIGURE 8 – Statue du temple *Khaymingha* reconstruit avec [**?**] à partir d'un échantillonnage généré par notre méthode.

5 Conclusion

Nous proposons une technique originale pour rééchantillonner les nuages de points à l'aide de disques de Poisson. Afin d'avoir une donnée structurée prenant en compte la topologie de la surface numérisée, les disques sont contruits sur des graphes définis à partir des cartes de profondeur fournies par les capteurs. Et ce de manière séquentielle pour pouvoir fonctionner sur des ordinateurs de capacité standard. Nous pouvons donc traiter des milliards de points, graphe après graphe, tout en ayant une qualité d'échantillonnage satisfaisante même dans les zones de recouvrement grâce à une mise en correspondance des sommets des différents graphes présents dans ces zones.

Nous avons montré expérimentalement que nous sommes capables de rééchantillonner des numérisations 3D de sites étendus, tout en améliorant le rendu visuel grâce notamment à une prise en compte de la courbure. C'est donc une solution très intéressante pour la visualisation et la manipulation de nuages de points gigantesques tels que ceux générés dans le contexte de la préservation du patrimoine mondial.

Références

[1] Cyark project. https ://artsandculture.google.com/partner/cyark.

[2] Dobrina Boltcheva and Bruno Levy. Surface reconstruction by computing restricted voronoi cells in parallel. *Computer-Aided Design*, 90 :123–134, 2017.

[3] John Bowers, Rui Wang, Li-Yi Wei, and David Maletz. Parallel poisson disk sampling with spectrum analysis on surfaces. *ACM Transactions on Graphics (TOG)*, 29(6) :166, 2010.

[4] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovačević. Fast resampling of three-dimensional point clouds via graphs. *IEEE Transactions on Signal Processing*, 66(3) :666–681, 2018.

[5] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1) :51–72, January 1986.

[6] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1) :269–271, December 1959.

[7] Jan Elseberg, Dorit Borrmann, and Andreas Nüchter. One billion points in the cloud–an octree for efficient processing of 3d laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76 :76–88, 2013.

[8] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH 2007 courses*, page 1. ACM, 2007.

[9] Jingliang Peng and C-C Jay Kuo. Geometry-guided progressive lossless 3d mesh coding with octree (ot) decomposition. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 609–616. ACM, 2005.

[10] Jean-Luc Peyrot, Frédéric Payan, and Marc Antonini. Direct blue noise resampling of meshes of arbitrary topology. *The Visual Computer*, 31(10) :1365–1381, October 2015.

[11] Nico Pietroni, Marco Tarini, Olga Sorkine, and Denis Zorin. Global parametrization of range image sets. In *ACM Transactions on Graphics (TOG)*, volume 30, page 149. ACM, 2011.

[12] Jean-Francois Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. *J. Electronic Imaging*, 2(4) :326–336, 1993.

[13] Li-Yi Wei and Rui Wang. Differential domain analysis for non-uniform sampling. *ACM Trans. Graph.*, 30 :50 :1–50 :10, 2011.