

Extraction d'un graphe de navigabilité à partir d'un nuage de points 3D enrichis

I. Ben salah¹
C. Demonceaux²

P. Vasseur¹
S. Kramm¹

¹ LITIS : Laboratoire d'informatique, de traitement de l'information et des systèmes, Université de Rouen

² Le2i : Laboratoire d'Electronique, Informatique et Image, Univ. Bourgogne Franche-Comté

imeen.ben-salah1@etu.univ-rouen.fr

Résumé

Ce travail se place dans le cadre général du projet ANR pLaTINUM lié à la navigation autonome et plus particulièrement à la génération de cartes pour la navigation basée perception. Il consiste à développer une nouvelle méthode pour résumer une carte 3D (un nuage dense de points 3D) et extraire un graphe de navigabilité facilitant l'utilisation de cette carte par des systèmes de navigation à ressources matérielles limitées (smartphones, voitures, robots...). Cette méthode vise à extraire les régions les plus saillantes de l'environnement étudié afin de construire une carte récapitulative. Ce processus de résumé de carte basé sur la vision est appliqué d'une façon totalement automatique utilisant les informations photométriques, géométriques et sémantiques de l'environnement étudié.

Mots Clef

Navigabilité, vision, perception, carte, nuage 3D, résumé, sphère, entropie, graphe, Saillance.

Abstract

This work takes part within the general framework of the ANR project pLaTINUM linked to autonomous navigation and more particularly the generation of maps for based-perception navigation. It involves the development of a new method for summarizing a 3D map (dense cloud of 3D points) and the extraction of an airworthiness graph facilitating the use of this map by navigation systems with limited material resources (smartphones, cars, robots...). This method aims to extract the most salient regions in the studied environment in order to construct summary map. This Vision-based summarizing process is applied in a fully automatic way using the photometric, geometric and semantic information of the studied environment.

Keywords

Navigation, vision, perception, map, 3D cloud, resume, summarizing, sphere, entropy, graph, saliency.

1 Introduction

Au cours des dernières années, l'utilisation des systèmes d'information géographique (SIG) est devenue de plus en plus accessible. Ces systèmes permettent d'acquérir, de traiter, d'organiser et de présenter des données géoréférencées. Les usages et les potentialités d'application des SIG sont diverses et variés (urbanisme, transport, télécommunication...). La démocratisation de la cartographie en ligne a permis l'utilisation des technologies d'information géographique dans les processus de participation publique pour effectuer plusieurs tâches (diffusion d'information, cartographie participative, formalisation des propositions citoyennes, etc.). Ces tâches requièrent une capacité de traitement importante avec des contraintes temporelles sévères ainsi qu'une nécessité de stockage volumineux lorsque l'on souhaite couvrir des espaces de grandes dimensions. Ce service fait appel à un grand nombre d'acteurs, et les informations délivrées doivent rester valides et précises dans le temps, nécessitant ainsi des campagnes de mise à jour fréquentes. Notre travail consiste à développer un nouveau système capable de résumer une carte 3D. La sortie de notre processus sera sous forme d'un graphe optimal dont chaque nœud sera associé à une image sphérique. Cette sphère RGB-DL contient les informations photométriques (RGB) et augmentée avec l'information de profondeur (D) et le label sémantique (L) pour chaque pixel. Les arcs reliant les nœuds (sphères) de ce graphe pourront contenir certaines informations sur les instructions concernant le déplacement entre les nœuds.

1.1 Travaux antérieurs

Dans certains domaines d'application, l'embarquement d'une carte complète de grande taille sur un dispositif mobile (automobile, robot...) pose plusieurs difficultés (calcul/mémoire). D'où le besoin de trouver une méthode permettant de résumer ces cartes afin de réduire les ressources nécessaires au fonctionnement du système tout en préservant les informations indispensables à la navigation (pixels saillants, nœuds importants...) sans pour autant diminuer la précision du système de navigation. L'idée porte donc sur l'éventuelle possibilité de compresser le modèle global

assez volumineux en une carte compacte en réduisant au maximum les éléments de l'environnement non nécessaires au bon fonctionnement du système de navigation. Une approche possible appelée « l'approche probabiliste » consiste à chercher le sous-ensemble de points le plus compact possible qui pourrait représenter tout l'environnement [10]. Une autre approche de localisation utilisant une technique qui s'appelle « la mémoire à base d'images » [14, 5, 17] pourrait bien servir à réaliser un résumé compact d'une carte. Dans cette approche de localisation, chaque estimation de position est calculée par rapport à une image de référence acquise au cours d'une phase d'apprentissage. Une mémoire de taille réduite à base d'images générées en combinant des informations de natures différentes sera utilisée dans la localisation à la place de la carte globale de l'environnement. Dans l'exemple de Cobzas [1], une mémoire d'images panoramiques est créée en combinant les images acquises par la caméra avec l'information de profondeur extraite à partir d'un scanner laser. Une solution a été proposée par M. Meilland et *et al.* [13, 12, 2] pour réaliser une représentation sphérique construite à l'aide d'un ensemble de caméras permettant de construire une image sphérique en combinant les images 2D venant des caméras avec l'information de profondeur venant d'un autre capteur laser. Dans cette représentation, toutes les informations nécessaires à la localisation sont présentes et compactées dans une seule sphère, évitant ainsi la cartographie des zones inutiles à la navigation [12]. Cette représentation sphérique présente les avantages suivants :

- La combinaison de différents types d'informations (luminosité, profondeur..)
- Une seule sphère suffit pour représenter différentes directions (comme les deux directions d'une rue).

Nous développons cette idée dans cet article, car cette représentation sphérique est prometteuse. Dans leur travail, ils ont proposé d'estimer la position optimale des sphères directement en se basant sur la mesure de luminosité et à la configuration géométrique 3D de l'environnement. Le résultat final du processus de résumé aboutira donc à un ensemble de sphères augmentées par les profondeurs stockées dans un système d'information géo-référencé qui sera utilisé par la suite au cours de la phase de navigation. En robotique mobile, la méthode la plus utilisée consiste à reproduire, en ligne, la cartographie de l'environnement et son résumé simultanément. Une sélection dynamique des informations saillantes est effectuée alors en même temps que l'acquisition des données tout au long de la progression du véhicule traceur. Ainsi, l'évolution du processus de résumé à l'instant t dépend seulement des données acquises aux instants antérieurs. Cette méthode ne peut pas s'appliquer dans notre cas, en effet, dans ce type d'algorithme les images clés sont extraites sur le parcours du robot mobile. Cependant, dans notre cas, les images doivent être disposées dans l'environnement de manière indépendante au déplacement de l'agent. Pour notre travail, nous avons choisi la méthode hors ligne pour effectuer le

processus de résumé de carte, une fois l'étape d'acquisition réalisée. Cette méthode nous permettra d'éviter les contraintes du calcul en temps réel et de mieux exploiter la totalité des informations utiles dans l'environnement. Nous avons choisi aussi la représentation sphérique en rajoutant à chaque pixel un autre type d'information qui est l'information sémantique sous la forme d'un label associé à chaque pixel.

2 Solution proposée

Notre travail vise à effectuer plusieurs tâches de navigation en utilisant uniquement un résumé cartographique de l'environnement. Cette carte devrait être non seulement compacte mais aussi cohérente avec la perception de l'agent. Pour fournir ce résumé de la carte, nous proposons une nouvelle méthode traitant des nuages de points 3D à grande échelle. La sortie de notre solution est un ensemble d'images sphériques interconnectées. Nos principales contributions sont les suivantes :

- Nous présentons le concept d'*Entropie*, qui est une mesure de la quantité d'informations pertinentes renvoyée par un point de vue en fonction des caractéristiques géométriques, photométriques et sémantiques. Cette mesure est utilisée pour sélectionner le meilleur point de vue.
- En utilisant *l'entropie*, nous formulons un processus de résumé d'une carte 3D comme un problème d'optimisation.
- Nous proposons une nouvelle représentation de la carte résumée en utilisant des images sphériques augmentées avec l'information de profondeur et l'information sémantique.
- Nous présentons une nouvelle étape de partitionnement dans notre algorithme pour permettre la gestion des nuages de points à grande échelle.

Dans la fig 1, nous présentons les principales étapes de notre algorithme pour trouver les positions optimales des sphères. Dans notre cas nous allons manipuler un nuage 3D très volumineux. Ce type de nuages de points à grande échelle contient plusieurs millions de points 3D. Nous avons donc choisi d'appliquer une méthode « diviser pour régner » pour positionner les sphères afin de réduire la complexité du traitement du nuage 3D très volumineux. la méthode « diviser pour régner » est donc la mieux adaptée à nos besoins. Cette technique algorithmique consiste à :

Diviser : découper un problème initial en sous-problèmes.
Régner : résoudre les sous-problèmes récursivement ou directement s'ils sont assez petits.

Combiner : fusionner les solutions des sous-problèmes pour trouver la solution du problème initial.

Nous présenterons, par la suite, les étapes de résolution du problème plus en détails. La première étape consiste à découper le grand nuage 3D du départ en plusieurs sous-ensembles qui vont être traités séparément par la suite afin

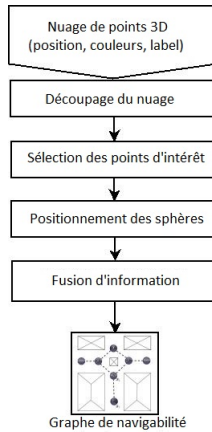


Figure 1: Vue générale de l’algorithme

de déterminer la position optimale de la sphère représentant chaque petite région (sous-ensemble). Pour ce faire, nous avons proposé deux différentes méthodes : le quadrillage statique de l’environnement consiste à diviser un territoire en secteurs identiques, en utilisant une grille, et le découpage avec le diagramme de Voronoï [18]. Cette technique est un découpage du plan 2D (pavage) en cellules à partir d’un ensemble discret de points appelés « germes ». Cette technique permet la décomposition du problème initial (grande taille) en un ensemble de sous-problèmes (petite taille). Nous détaillerons par la suite l’étape de sélection des points d’intérêt, l’étape de positionnement des sphères, et l’étape de fusion d’informations.

2.1 Sélection des points d’intérêt

Lors du traitement d’un nuage de points 3D, la sélection des points qui formeront la carte compacte (résumé) repose sur le choix des points les plus représentatifs de l’environnement généralement appelés points clés (points d’intérêt). Il existe de nombreux algorithmes utilisés pour la détection de la saillance visuelle 3D. Ce problème a été activement abordé ces dernières années à cause de l’utilisation massive des données 3D rendue plus facile grâce à la grande diversité des types de capteurs utilisés (LIDAR, LASER, GPS...). Les modèles de détection de saillance en 3D peuvent être regroupés selon leurs types de données d’entrée. Un premier modèle représente l’environnement sous forme d’un ensemble d’images RGB augmentées avec leurs cartes de profondeur. Dans ce cas, plusieurs méthodes ont été proposées pour détecter la saillance 3D sur des images RGB-D [6], [15], [22]. Un deuxième modèle utilise un maillage 3D de l’environnement pour extraire les zones saillantes [7], [8], [21]. Le troisième modèle consiste à utiliser un nuage de points 3D augmenté comme entrée pour produire une carte de saillance [9], [19], [23].

Nuage 3D. Nous avons choisi le troisième modèle pour notre travail. Ce type d’algorithme consiste à traiter la sail-

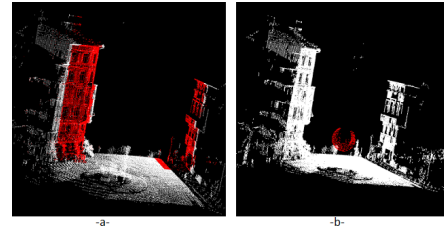


Figure 2: La base de donnée de test (réalité terrain) -a- segmentation manuelle -b- positionnement de sphère

lance visuelle dans un nuage de points 3D. Ces algorithmes se basent sur la distinction entre les régions dans un environnement pour prédire la saillance dans un nuage. Contrairement aux autres modèles, les travaux qui traitent la saillance dans un nuage sont très rares dans la littérature à cause du manque de connectivité entre les points 3D de ce nuage. Un algorithme de détection de saillance dans un nuage de points 3D à grande échelle a été proposé dans le travail de Shtrom [19] (Méthode 1). Dans cette méthode, la saillance est détectée en utilisant un descripteur de point 3D appelé FPFH (Fast Point Feature Histogram) pour caractériser la géométrie du voisinage d’un point. Un point est considéré comme « distinct » si son descripteur est différent de tous les autres descripteurs de points du nuage. Cette opération est effectuée sur plusieurs échelles avec différentes tailles de voisinage. Par la suite, une somme pondérée des cartes de saillance issues de plusieurs échelles est calculée pour obtenir une carte de saillance globale. Cette méthode détecte la saillance en se basant uniquement sur la géométrie de la scène sans l’utilisation d’autre type d’information comme les couleurs. Un nouvel algorithme de détection de saillance pour les nuages a été proposé dans un travail récent [23] (Méthode 2). Cet algorithme consiste à exploiter simultanément les caractéristiques géométriques et les caractéristiques de couleur pour estimer la saillance dans un nuage de points coloré. Tous les points 3D sont regroupés en plusieurs sous-ensembles appelés « supervoxels ». La mesure de saillance pour chaque supervoxel est calculée à l’aide des caractéristiques géométriques et de couleur de ses voisins. Ce processus est appliqué sur plusieurs échelles. En faisant la moyenne des cartes multi-échelles pixel par pixel, nous obtiendrons une carte de saillance globale. Une autre méthode a été proposée dans le travail de Leroy [9] (Méthode 3) en se basant uniquement sur la photométrie dans un nuage 3D pour estimer une carte de saillance.

Pour évaluer le niveau de la saillance capturée par ces méthodes, nous avons proposé de calculer un critère appelé dans la littérature : F_β mesure (F_β) [11]. Ce critère nous permet d’avoir une idée sur la pertinence de l’information renvoyée par chaque méthode. Cette mesure est calculé comme suit:

$$F_\beta = \frac{(1 + \beta^2) * (Précision * Rappel)}{\beta^2 * Précision + Rappel} \quad (1)$$

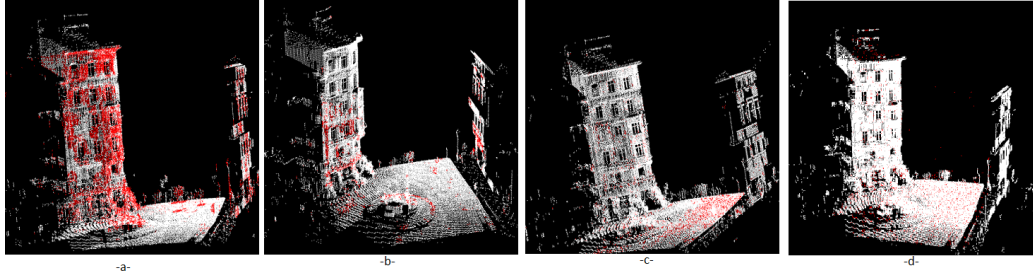


Figure 3: Résultat de détection de points saillants (rouge) selon les quatre méthodes d'extraction de saillance 3D. -a- méthode géométrique [19] -b- méthode géométrique et photométrique [23] -c- méthode photométrique [9] -d- Harris3D [4]

Méthodes	F_β	Distance (m)
Méthode 1 [19]	0.7234	2.1390
Méthode 2 [23]	0.6696	3.0418
Méthode 3 [9]	0.5363	6.6005
Harris3D [4]	0.4536	7.6005

Table 1: Résultats selon les quatre méthodes d'extraction de la saillance 3D

$$Rappel = \frac{VP}{(VP + FN)} \quad (2)$$

$$Précision = \frac{VP}{(VP + FP)} \quad (3)$$

Où

Vrai Positif (VP) : nombre de points raisonnablement classés comme pertinents pour la localisation

Faux Positif (FP) : nombre de points classés à tort comme pertinents pour la localisation

Faux Négatif (FN) : nombre de points classés à tort comme non pertinents pour la localisation

F_β est la moyenne harmonique pondérée de précision et de rappel et atteint sa meilleure valeur à 1 et le pire score à 0. Le rappel et la précision sont tout aussi importants si β est égale à 1. $\beta < 1$ prête plus de poids à la précision, tandis que $\beta > 1$ favorise le rappel. Pour contraster l'approche consistant à utiliser le rappel maximal des points (sans discrimination), nous avons décidé que la précision devrait avoir beaucoup plus de priorité sur le rappel. Dans notre travail, β est mis à 0.5 car c'est l'une des valeurs les plus courantes attribuées à β (le rappel est à moitié aussi important que la précision).

2.2 Positionnement des sphères

Après le découpage du nuage initial en plusieurs sous-nuages, l'étape de sélection des points d'intérêt est appliquée sur chaque sous-nuages. Pour chaque sous-nuage le positionnement de sphère est résolu à l'aide d'un processus d'optimisation permettant de chercher la position optimale de la sphère dans un nuage 3D. Ce problème est connu sous le nom « Sélection de point de vue optimal » Dans ce type de problème, on cherche à déterminer

la meilleur localisation possible du point de vue, de façon à maximiser la quantité d'informations qu'il donne sur une scène. Dans la littérature, plusieurs méthodes ont été proposées pour évaluer la qualité d'un point de vue dans une scène. On peut classer ces méthodes selon la nature des informations d'entrée. La première famille regroupe les méthodes qui utilisent uniquement la quantité de surfaces visibles à partir d'un point de vue pour évaluer sa qualité. Une méthode de cette famille a été proposée par Plemenos et Benayada [16] considérant un point de vue comme étant optimal lorsqu'il minimise l'écart d'angle entre une direction de vue et les normales aux surfaces. Le travail de Guy et Medioni [3] propose un algorithme de sélection du meilleur point de vue. Cette méthode fondée sur la théorie de l'information consiste à maximiser l'entropie pour sélectionner le point de vue optimal. Une autre méthode de cette famille a été proposée dans [7]. Cette méthode exploite la saillance calculée sur un maillage de points. Le point de vue optimal maximise la somme de saillances des régions visibles depuis ce point de vue. La saillance d'une région est calculée à partir de sa courbure. La deuxième famille de méthodes de sélection du meilleur point de vue utilise la quantité des surfaces visibles à partir de ce point de vue ainsi que leur géométrie. Dans [20] Sokolov et Plemenos ont proposé de prendre en compte la courbure totale des surfaces visibles pour la sélection du point de vue optimal. La troisième famille de méthodes consiste à exploiter la visibilité des objets à partir d'un point de vue. Nous nous sommes inspirés de tous ces travaux pour modéliser notre problème de sélection du meilleur point de vue.

Modélisation du problème. Notre but est de déterminer le point de vue optimal (position de sphère) qui représente le mieux possible tout un nuage 3D. Ce point est le centre d'une sphère unitaire sur laquelle seront projetés tous les points du nuage. Cette sphère est un résumé de tout l'environnement étudié. Lors de la phase de localisation, l'utilisateur du système va être localisé dans une carte constituée d'un ensemble de sphères qui représente le résumé du nuage initial dense. Pour ce faire, nous avons défini un critère à optimiser (l'entropie) dans la phase de recherche de la position optimale. Ce critère est défini comme suit : L'entropie est la quantité d'informations pertinentes renvoyée par une sphère. Dans notre cas la pertinence d'un

point 3D de l'espace est traduite par deux aspects : un point 3D de l'espace est dit pertinent s'il est sélectionné comme étant un point saillant avec les méthodes décrites précédemment ou/et s'il est un point labellisé par un label significatif lui donnant un sens sémantique (objet fixe utile pour la localisation : arbre, bâtiment...).

Calcul d'entropie Considérons un ensemble fini de N points 3D $\{P_i(X_i, Y_i, Z_i, R_i, G_i, B_i, L_i), i = 1..N\}$, augmentés avec leurs informations photométriques (R_i, G_i, B_i) et leurs labels L_i et un point $C(X_c, Y_c, Z_c)$ centre de notre sphère. L'idée est de projeter tous les points du nuage sur la sphère. Pour chaque point P_i , les coordonnées (X, Y, Z) du point projeté sur la sphère sont :

$$\left(\frac{X_c + R(X_i - X_c)}{\|\vec{PC}\|}, \frac{Y_c + R(Y_i - Y_c)}{\|\vec{PC}\|}, \frac{Z_c + R(Z_i - Z_c)}{\|\vec{PC}\|} \right) \quad (4)$$

Par la suite nous faisons la conversion en coordonnées sphériques :

$$(\phi, \theta) = \begin{cases} \arccos(Z/\rho) \\ \arctan(Y/X) \end{cases} \quad (5)$$

R est le rayon de la sphère (réglé à 1 dans notre travail). L'étape suivante consiste à échantillonner ces points

Algorithm 1 Entropie

Entrée : $\{P_j(X_j, Y_j, Z_j, S_j^{low}, S_j^{high}) \in S, j = 1..M\} \triangleright$

M : nombre de points projetés sur la sphère S du centre C
Sortie : $\xi(C) \in [0..1]$

```

1: procedure  $\xi(C)$  ▷ L'entropie du centre  $C$ 
2:    $\xi(C) \leftarrow 0$ 
3:    $j \leftarrow 1$ 
4:    $n_{00} \leftarrow 0, n_{01} \leftarrow 0, n_{10} \leftarrow 0, n_{11} \leftarrow 0$ 
5:   for  $j = 1$  to  $M$  do
6:     if  $((S_j^{low} = 0) \text{ and } (S_j^{high} = 0))$  then
7:        $n_{00} \leftarrow n_{00} + 1$ 
8:     else if  $((S_j^{low} = 1) \text{ and } (S_j^{high} = 0))$  then
9:        $n_{10} \leftarrow n_{10} + 1$ 
10:    else if  $((S_j^{low} = 0) \text{ and } (S_j^{high} = 1))$  then
11:       $n_{01} \leftarrow n_{01} + 1$ 
12:    else
13:       $n_{11} \leftarrow n_{11} + 1$ 
14:    end if
15:  end for
16:   $\xi(C) \leftarrow$ 
17:   $-\frac{n_{00}}{M} \log \frac{n_{00}}{M} - \frac{n_{01}}{M} \log \frac{n_{01}}{M} - \frac{n_{10}}{M} \log \frac{n_{10}}{M} - \frac{n_{11}}{M} \log \frac{n_{11}}{M}$ 
18:  return  $\xi(C)$ 
19: end procedure

```

projetés pour en avoir une représentation homogène sur toutes les sphères. Pour ce faire, nous avons utilisé l'échantillonnage à angles constants pour garder le maximum de points sur chaque sphère en conservant la distribution de ces points.

Cette méthode consiste à échantillonner les points projetés

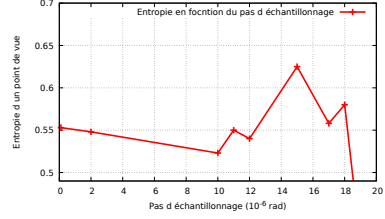


Figure 4: Évolution de valeur d'entropie selon le pas d'échantillonnage

sur la sphère : $P_s=(\theta, \phi)$ à pas constant, avec les angles $\theta \in [-\pi, \pi]$ et $\phi \in [0, \pi]$, par les pas d'échantillonnage $\partial\theta$ et $\partial\phi$

$$(d\theta = \frac{2\pi}{m}, d\phi = \frac{\pi}{n}) \quad (6)$$

Dans cette équation m désigne le nombre d'échantillons en latitude et n le nombre d'échantillons en longitude. Nous avons mené une petite étude sur l'influence du pas d'échantillonnage sur la valeur d'entropie, indépendamment d'autres facteurs, sur un ensemble de données de test contenant 60 000 points 3D et représentant un environnement urbain. Les résultats sont affichés sur la Fig. 4. La valeur d'entropie diminue fortement pour des valeurs supérieures à $15 \cdot 10^{-6} \text{ rad}$. Cependant, des valeurs plus petites augmentent le volume de données, nous recommandons donc de rester avec cette valeur. Nous proposons de définir deux niveaux de saillance dans un nuage de points 3D. La saillance de bas niveau S^{low} est basée sur les caractéristiques de bas niveau (photométrique et géométrique) d'un point 3D. La saillance de haut niveau S^{high} est basée sur l'information sémantique de chaque point 3D. Pour calculer les valeurs S^{low} , nous utilisons la méthode de saillance utilisant le découpage en supervoxels décrite dans la section précédente qui permet la combinaison d'informations photométriques et géométriques. Nous avons choisi cette méthode car elle assure un compromis entre le temps de calcul et la sélection des points les plus saillants utiles pour la localisation. Pour calculer S^{high} , nous utilisons les labels sémantiques pour chaque point. En utilisant ces deux types de saillance, nous calculons le nombre de points d'intérêt sur la sphère en fonction de leur pertinence. Par conséquent, nous avons quatre combinaisons possibles comme suit :

- n_{00} : nombre de points non pertinents sémantiquement, photométriquement et géométriquement.
- n_{10} : nombre de points uniquement pertinents photométriquement et géométriquement.
- n_{01} : nombre de points uniquement pertinents sémantiquement.
- n_{11} : nombre de points pertinents sémantiquement, photométriquement et géométriquement.

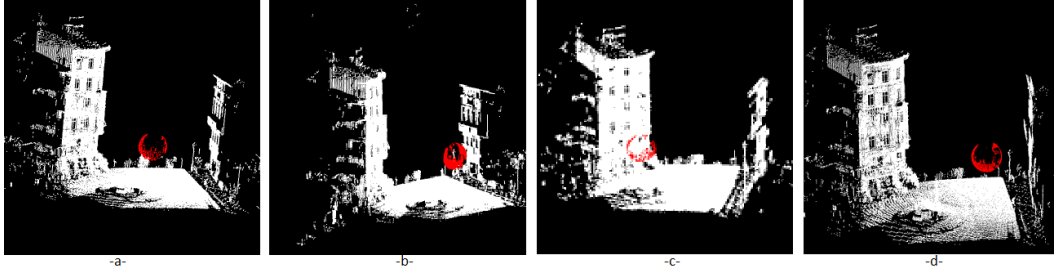


Figure 5: Résultat de positionnement de sphères selon les quatre méthodes d'extraction de saillance 3D. -a- méthode géométrique [19] -b- méthode géométrique et photométrique [23] -c- méthode photométrique [9] -d- Harris3D [4]

L'entropie d'une sphère sera caractérisée par l'entropie de son centre P . L'entropie est donnée par l'algorithme 1.

Optimisation. La sélection du point de vue optimal est un processus d'optimisation d'entropie. Ce point de vue optimal est le centre de notre sphère optimale. Pour ce faire, nous avons proposé un algorithme génétique dont le but est de déterminer les extremums d'entropie $\xi : X \rightarrow \mathbf{R}^3$, où X est la population initiale de N points 3D. Ces points sont les centres de sphères se trouvant sur une zone navigable du nuage. Cet algorithme d'optimisation consiste à évaluer l'entropie de chaque individu de la population. Les deux individus (parents) ayant des valeurs maximales d'entropie sont sélectionnés. La combinaison de deux parents dans une première itération permet d'obtenir une solution (enfant) dont l'entropie est meilleure que celle de ces deux parents. Par exemple en combinant les coordonnées (X, Y) de deux parents, nous obtenons une nouvelle solution qui permet d'optimiser le critère d'entropie sur les deux axes X et Y . Grâce à cette évolution de sélection de notre centre de sphère en passant d'une itération à un autre, l'algorithme devient capable de converger vers une solution parmi les points maximisant l'entropie au bout de quelques itérations. Le processus d'optimisation est défini dans l'algorithme 2.

2.3 Fusion d'information

Nous avons présenté, jusqu'à maintenant, l'étape de découpage du nuage initial et l'étape de traitement de chaque sous-ensemble afin de trouver la position optimale de chaque sphère. Dans cette section, nous allons présenter la dernière étape de notre algorithme permettant de calculer la similarité entre les sphères obtenues et fusionner celles qui sont similaires pour éliminer les informations redondantes. La méthode la plus simple pour calculer la similarité entre deux sphères consiste à calculer leur corrélation statistique. Cette corrélation (similarité) exprime le taux de variation des informations photométriques de l'environnement contenues dans les deux sphères comparées. Dans les travaux de Meilland [13, 12], ce critère est calculé à l'aide de la valeur absolue des écarts à la médiane (MAD) qui représente la différence (erreur) d'intensité des pixels des deux sphères (comparés pixel par pixel). Théoriquement, deux sphères sont considérées similaires si la valeur de la

Algorithm 2 Algorithme génétique

Inputs : Zone navigable $\{P_i(X_j, Y_j, Z_j) \in Z, i = 1..N\}$
 Nombre d'itérations N_{iter} Nombre d'individus N_{indiv}
Sortie : Meilleur point de vue $V \in Z$

```

1: procedure  $GA(Z)$ 
2:    $k \leftarrow 1$ 
3:    $I \leftarrow$  Sélection de  $N_{indiv}$  points aléatoirement  $\in Z$ 
4:   while  $k < N_{iter}$  do
5:     for Chaque point  $p_h$  dans  $I$  do  $entropy(h) \leftarrow \xi(p_h)$ 
6:     end for
7:      $parent1 \leftarrow arg\ max_1(entropy)$ 
8:      $parent2 \leftarrow arg\ max_2(entropy)$ 
9:     for  $l = 1$  to  $N_{indiv}$  do
10:      Ajouter  $\lambda * parent1 + \alpha * parent2$  à  $I$ 
       $\triangleright \lambda$  et  $\alpha$  nombres aléatoires  $\in [0..1]$ 
11:    end for
12:  end while
13:  return ( $arg\ max(entropy)$ )
14: end procedure

```

(MAD) est inférieure à un certain seuil λ (cela est dû à la faible distance entre les sphères).

$$\lambda \geq median(|e(x) - median(e(x))|) \quad (7)$$

Dans cette inéquation $e(x)$ est un vecteur contenant les erreurs d'intensité entre les pixels des deux sphères. Si deux sphères ou plus sont similaires, un processus de fusion d'informations est déclenché. Ce processus consiste à concaténer les régions correspondantes aux sphères similaires puis ré-appliquer la méthode d'optimisation (Algorithme Génétique) afin de fusionner les sphères en une seule qui représente à la fois les différentes régions.

3 Résultats

Nous avons utilisé une base de données qui contient une partie de la carte 3D d'un milieu urbain. Cette carte « RGBDdatasets » [24] contient un petit ensemble de points 3D avec leurs intensités. Nous avons choisi d'implémenter et tester les trois méthodes détaillées dans la deuxième section pour extraire une carte de saillance. Nous avons testé

aussi le détecteur Harris3D [4] sur la même base de données « RGBDdatasets ». Afin de construire une réalité terrain, nous avons segmentée manuellement cette base de données urbaine. Pour ce faire, nous avons choisi les points les plus saillants et utiles pour la localisation. Parmi ces points, nous avons choisi ceux qui appartiennent à la façade du bâtiment, aux panneaux de signalisation et au marquage au sol. Cette réalité terrain permet d'évaluer les résultats obtenus. La figure 3 montre le résultat obtenu pour chaque méthode. Le tableau 1 montre les valeurs de F_β obtenues par les quatre méthodes. Les deux algorithmes qui utilisent la géométrie dans un nuage donnent des résultats meilleurs que les méthodes utilisant uniquement la photométrie car, dans notre cas, les points intéressants pour la localisation sont généralement plus saillants sur le plan géométrique que sur le plan photométrique. La figure 5 montre le résultat de positionnement de la sphère dans cette scène. Ces résultats sont obtenus à l'aide d'algorithme d'optimisation génétique du critère d'entropie. A l'issue de l'application des quatre méthodes d'extraction de saillance 3D, nous avons obtenu quatre résultats sous forme d'une position (x,y,z) représentant un centre de sphère résumant la scène. La figure 2 montre notre vérité terrain. La sphère se trouvant à équidistance des bâtiments est donc le meilleur point de vue pour visualiser la totalité des points d'intérêt de la carte. Nous avons calculé la distance euclidienne entre les sphères résultantes des quatre méthodes et la sphère de la vérité terrain (tab. 1). Le résultat obtenu avec les méthodes utilisant la géométrie sont les plus proches de la vérité terrain. Dans la partie suivante de notre travail, nous avons appliqué notre méthode sur un environnement beaucoup plus vaste. Ce nuage de points à grande échelle contient plus de 40 millions de points 3D labélisés. Dans cet ensemble de données, nous avons 8 classes d'étiquettes (1: terrain artificiel, 2: terrain naturel, 3: végétation élevée, 4: faible végétation, 5 : bâtiments, 6 : tige dure, 7 : artefacts de la numérisation, 8 : voitures). Dans notre processus de synthèse, les points appelés «bâtiments» sont considérés parmi les points les plus saillants pour la localisation. Cet ensemble de données a permis d'évaluer la performance de notre solution en utilisant toutes les caractéristiques sémantiques, photométriques et géométriques. Après avoir divisé ce nuage de points à grande échelle, nous avons obtenu 72 sous-nuages. La sortie de ce processus de résumé, comme représenté sur Fig. 6, est un ensemble compact d'images sphériques. Nous avons obtenu 35 sphères résumant un nuage de points d'environ 160 m de long. La distance moyenne entre toutes les sphères est d'environ 4,5 m. Nous avons obtenu un taux de compression de 93% de cette carte. Nous avons calculé le Rappel et la Précision (équations 2 et 3). Ils sont définis comme suit :

Vrai positif (VP) : nombre de points pertinents réellement projetés sur les sphères

Faux positif (FP) : nombre de points non pertinents réellement projetés sur les sphères

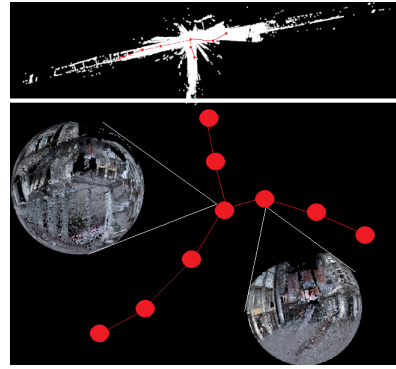


Figure 6: Résumé d'un nuage de points 3D

Vrai négatif (VN) : nombre de points non pertinents et non projetés sur les sphères

Faux Négatif (FN) : nombre de points pertinents et non projetés sur les sphères

Nous avons construit notre base de données de réalité-terrain. Pour chaque point du nuage, nous avons attribué une étiquette 0: non pertinent pour la localisation, 1 : pertinent pour la localisation. La plupart des points pertinents appartiennent aux bâtiments grâce à leurs formes géométriques. Nous avons considérablement réduit la taille de la carte. Néanmoins, nous avons réussi à conserver un maximum de points saillants (Rappel environ 60%) avec un bon niveau de précision (supérieur à 91%). Pour améliorer la valeur de rappel, nous avons pu affiner les étiquettes sémantiques d'entrée pour sélectionner les zones les plus fines et les plus utiles pour la localisation (fenêtres, portes, panneaux...). Toutes les sphères sont positionnées de manière à capturer le maximum de points possibles appartenant aux façades des bâtiments. Les points 3D appartenant à des végétations basses ou des objets en mouvement comme les voitures et les piétons, ne sont pas importants dans le processus de localisation. Par conséquent, ils ne sont pas très présents dans ces sphères résumant le nuage d'entrée.

4 Conclusion

La méthode développée tout au long de ce projet nous permet de résumer une carte sous forme d'un nuage de points 3D initialement volumineuse et dense en informations grâce à un processus basé sur l'extraction d'un graphe dont les nœuds sont des vues sphériques représentant des sous-nuages de la carte initiale. Cette nouvelle méthode de résumé de cartes 3D nous permettra de faciliter plusieurs tâches de navigation (localisation, planification d'itinéraires, évitement d'obstacles...) en réduisant fortement le temps de calcul et la taille mémoire nécessaires au fonctionnement du système de navigation. Dans la prochaine étape de notre projet, nous visons à fournir une carte à plusieurs niveaux dans laquelle un niveau spécial sera entièrement dédié à tous les systèmes de transport

(trains, voitures, motos, piétons ...) pour améliorer la précision de la navigation.

References

- [1] Dana Cobzas, Hong Zhang, and Martin Jagersand. Image-based localization with depth-enhanced image map. In *Robotics and Automation*, volume 2, pages 1570–1575. IEEE, 2003.
- [2] Gabriela Gallegos, Maxime Meilland, Patrick Rives, and Andrew I Comport. Appearance-based slam relying on a hybrid laser/omnidirectional sensor. In *Intelligent Robots and Systems (IROS)*, pages 3005–3010. IEEE, 2010.
- [3] Gideon Guy and Gérard Medioni. Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1265–1277, 1997.
- [4] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.
- [5] Matjaz Jogan and Ales Leonardis. Robust localization using panoramic view-based recognition. In *Pattern Recognition*, volume 4, pages 136–139. IEEE, 2000.
- [6] Ran Ju, Ling Ge, Wenjing Geng, Tongwei Ren, and Gangshan Wu. Depth saliency based on anisotropic center-surround difference. In *Image Processing (ICIP)*, pages 1115–1119. IEEE, 2014.
- [7] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM transactions on graphics (TOG)*, number 3, pages 659–666. ACM, 2005.
- [8] George Leifman, Elizabeth Shtrom, and Ayellet Tal. Surface regions of interest for viewpoint selection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 414–421. IEEE, 2012.
- [9] Julien Leroy, Nicolas Riche, Matei Mancas, and Bernard Gosselin. 3d saliency based on supervoxels rarity in point clouds.
- [10] Fayin Li and Jana Kosecka. Probabilistic location recognition using reduced feature set. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3405–3410. IEEE, 2006.
- [11] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014.
- [12] Maxime Meilland, Andrew I Comport, and Patrick Rives. A spherical robot-centered representation for urban navigation. In *Intelligent Robots and Systems (IROS)*, pages 5196–5201. IEEE, 2010.
- [13] Maxime Meilland, Andrew I Comport, and Patrick Rives. Dense omnidirectional RGB-D mapping of large-scale outdoor environments for real-time localization and autonomous navigation. *Journal of Field Robotics*, 32(4):474–503, 2015.
- [14] Emanuele Menegatti, Takeshi Maeda, and Hiroshi Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.
- [15] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgb-d salient object detection: a benchmark and algorithms. In *European Conference on Computer Vision*, pages 92–109. Springer, 2014.
- [16] Dimitri Plemenos and Madjid Benayada. Intelligent display in scene modeling. new techniques to automatically compute good views. In *International Conference GraphiCon*, volume 96, pages 1–5, 1996.
- [17] Anthony Remazeilles, François Chaumette, and Patrick Gros. Robot motion control from a visual memory. In *Robotics and Automation*, volume 5, pages 4695–4700. IEEE, 2004.
- [18] Dominique Schmitt. *Sur les diagrammes de Delaunay et de Voronoi d'ordre k dans le plan et dans l'espace*. PhD thesis, 1995.
- [19] Elizabeth Shtrom, George Leifman, and Ayellet Tal. Saliency detection in large point sets. In *IEEE International Conference on Computer Vision*, pages 3591–3598, 2013.
- [20] Dmitry Sokolov, Dimitri Plemenos, and Karim Tamine. Viewpoint quality and global scene exploration strategies. *GRAPP*, 2006:184–191, 2006.
- [21] Ran Song, Yonghuai Liu, Ralph R Martin, and Paul L Rosin. Mesh saliency via spectral processing. *ACM Transactions on Graphics (TOG)*, 33(1):6, 2014.
- [22] Junle Wang, Matthieu Perreira Da Silva, Patrick Le Callet, and Vincent Ricordel. Computational model of stereoscopic 3d visual saliency. *IEEE Transactions on Image Processing*, 22(6):2151–2165, 2013.
- [23] Jae-Seong Yun and Jae-Young Sim. Supervoxel-based saliency detection for large-scale colored 3d point clouds. In *IEEE International Conference (ICIP)*, pages 4062–4066. IEEE, 2016.
- [24] Bernhard Zeisl, Kevin Koser, and Marc Pollefeys. Automatic registration of rgb-d scans via salient directions. In *international conference on computer vision*, pages 2808–2815, 2013.