

Navigability Graph Extraction From Large-Scale 3D Point Cloud

Imeen Ben Salah¹, Sébastien Kramm¹, Cédric Demonceaux² and Pascal Vasseur^{1,2}

¹ Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes

Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France

² Laboratoire LE2I EA 7508, Equipe VIBOT ERL CNRS 6000, Univ. Bourgogne Franche-Comté

Abstract—This paper presents a novel method for summarizing and compression of large-scale 3D models into compact spherical representations. The information is combined into a set of optimized spheres in order to facilitate its use by systems with limited resources (smartphones, robots, UAVs, ...). This vision-based summarizing process is applied in a fully automatic way using jointly photometric, geometric and semantic information of the studied environment. The main contribution of this research is to provide a navigability graph that maximizes the significance of the contents of its nodes while maintaining the full visibility of the environment. Experimental results in summarizing large-scale 3D map demonstrate the feasibility of our approach and evaluate the performance of the algorithm.

I. INTRODUCTION

Digitize the world into 3D models has been an active research topic for decades due to the modernization of 3D sensors. To date, modern mobile scanning systems make it easy to produce High-quality 3D Point Clouds. In other respects, the introduction of High-Definition (HD) [21] and semantic maps have greatly participated in the enhancement of the accuracy of navigation systems. Such HD maps usually occupy a large amount of storage space and require a high processing capacity with severe time constraints. In order to store and transmit effectively this type of maps by navigation systems with limited resources (computation / memory), the size of these maps must be reduced. Hence, we need an efficient algorithm to summarize and compress a 3D map while preserving the essential information for navigation.

II. PREVIOUS WORKS

High-quality maps are required for a wide range of applications, particularly for applications where constant localization or feature tracking is crucial. Hence understanding a maps structure and evaluating map quality is very useful for accurate navigation [23]. Furthermore, In some navigation tasks, setting a full-size map on a mobile device (car, robot, etc.) poses several difficulties in time and memory consumption. To solve this problem there are two possible approaches: map compression or map summarizing using only significant information. Over the past decade, several compression approaches have been proposed in the literature. Several feature-based visual SLAM systems [16], [18] have been proposed to sample and compress a map. These types of methods, based on the selection of the keyframe and the bag of Word concepts. Another compression method has been proposed in [12]. This method is based on probability

mapping using Octrees to eliminate redundant information. Another compression algorithms have been proposed in [9], [10]. They aim to select 3D points from the initial map based on hundreds of descriptors requiring significant memory size. A novel compression method has been proposed in [7]. This method consists in sampling a point cloud by selecting the features useful for future relocalization. An approach to map reduction was proposed in [22]. It aims to select only the places that are particularly suitable for localization using the *location utility* metric. To simplify the process of appearance-based navigation, a selection process is applied to choose the key/reference features in the environment. For instance, in visual memory based approaches, a set of relevant and distinctive areas (images) are acquired and used during navigation for comparison with the current position. In the work of Cobzas [6], a panoramic memory of images is created by combining acquired images with depth information extracted from a laser scanner. In this image database, only the salient information will be retained [4] without degrading the performance during navigation. In order to build this image database, some techniques have been developed to guarantee the maximum efficiency in the choice of useful information. A spherical representation has been proposed by M. Meilland *et al.* [15]. This spherical representation is built by merging different images acquired by a set of cameras with the depth information extracted from a laser scanner. All existing methods for summarizing maps are based mainly on geometric or photometric characteristics to select the most salient information. However, these characteristics are insufficient for good perception and understanding of the environment. A combination of geometric, photometric and semantic characteristics when selecting information allows us to have a compact, precise and useful summary. Our work aims to perform several navigation tasks using only a map summary of the environment. This map should be not only compact but also coherent with the perception of the agent. To provide this map summary, we propose a new method dealing with large-scale 3D point clouds. The output of our summarizing method is a set of spherical images. Our main contributions are:

- We propose an efficient algorithm for summarizing maps based on geometric, photometric and semantic characteristics.
- We formulate map summarizing process as a multi-objective optimization problem.

- We propose a new scalable representation of summarized map using a navigability graph whose nodes are augmented and labeled spherical images.

The remainder of this paper is organized as follows. Section III is dedicated to the problem formulation and presents an overview of our approach. The complete map summarization process is described in Sections IV and V. Before conclusion and perspectives, experimental results and discussions are presented in Section VI.

III. PROBLEM MODELING

In this article, we aim at summarizing a 3D point cloud to a compact set of spherical images. Our objective is to reduce the number of spheres while maintaining a high level of significant information for navigation and localization. The final set of spheres must guarantee a compromise between three objectives:

- Maximize the visibility
- Maximize the Entropy
- Minimize the number of spheres

The first goal is to maximize the visibility V of the studied environment. The visibility of a 3D point c corresponds to all points visible from this point. Geometrically, we can define the visibility of a point c by the set of points, such that for each point p , the segment pc does not intersect any obstacle. We define the visibility V as: $V(c) = \{ p \in R^3 / p \text{ is visible from } c \}$. The second objective is the entropy E which expresses the quantity of visible salient information from a viewpoint. We define the entropy of a sphere as the amount of significant information projected onto it [2], [20]. In our previous work [20] it was found that using this entropy we are able to select the optimal viewpoint for representing as best as possible a 3D cloud.

The third objective is the number of spheres which must be minimized in order to reduce the size (cost) of the summary map. To modelize this problem, we introduce some notations. Let us suppose that the 3D map is composed of N 3D points with color and semantic information. $\mathcal{C} = \{P_i(X_i, Y_i, Z_i) \rightarrow \{R_i, G_i, B_i, L_i\}, i = 1..N\}$. We define the navigable areas Γ as the 2D polygon whose edges represent the borders of the navigable areas. To reduce the data, we propose to represent the different information into some augmented spherical images. To do this, let us consider $a = (X_g, Y_g, Z_g)$ a point in Γ and let us create the augmented 3D spherical image centered in a by projecting $V(a)$ in the corresponding sub-cloud into the unit sphere centered in a . Thus, this spherical image contains all the information which can be seen from a . The goal of this paper is to reduce the number n of augmented spheres needed to describe the 3D maps. Thus, we want to develop a method for creating a minimal set of spheres $g_n = \{g_j, j = 1..n\}$. These spheres should respect the previous objectives

$$\min_n g_n \text{ such as } \begin{cases} \#(\cup_{j=1}^n V(g_j)) \geq \beta \cdot N, \beta \in [0..1] \\ E(g_j) \geq \lambda, \lambda \in [0..1], j = 1..n \end{cases} \quad (1)$$

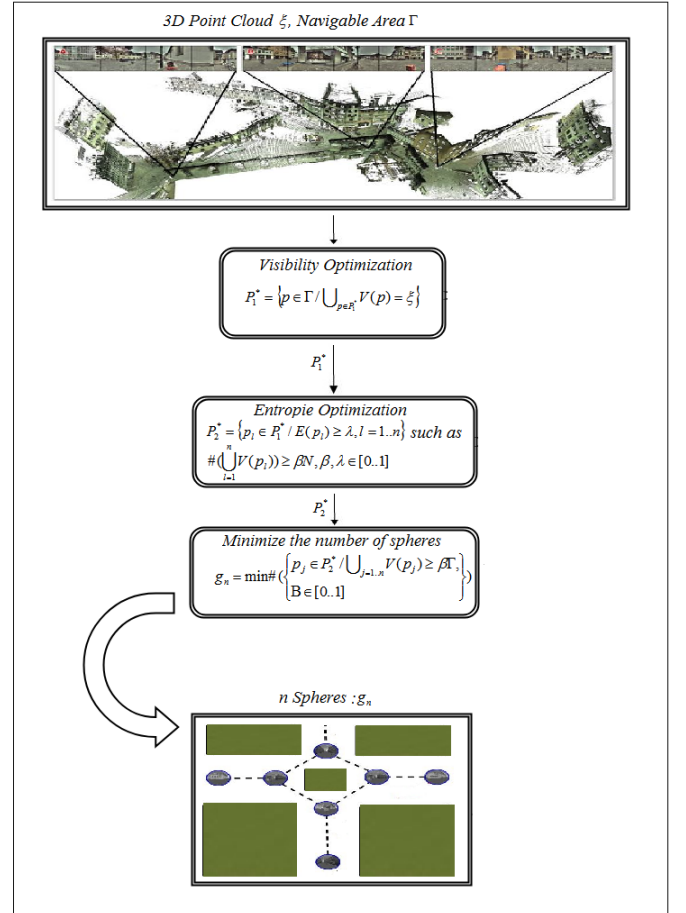


Fig. 1. Optimization process

Where $\#A$ is the cardinality of the set A .

We then propose to formulate this summary process of a point cloud as a multi-objective optimization problem. One of the approaches proposed to solve multi-criteria problems is the lexicographic approach [11]. This method optimizes the objectives separately and sequentially and generates an optimal solution at every sequential optimization stage.

In our case, the objective functions are initially arranged in importance order. First, we should guarantee that the full scene was visited when choosing the best viewpoints by maximizing the visibility. Then we search among these viewpoints that maximize the entropy. Finally, we minimize the number of spheres.

Figure 1 shows the different steps of the optimization process.

A. Visibility Optimization

The first step of the optimization process is the maximization of the visibility. In this stage, we must guarantee that each point of the cloud is visible from at least one point of view.

The first stage of the problem is solved as follows:

$$f_1 : \begin{cases} \mathcal{P}_1^* = \{P \in \Gamma\}, \\ \text{such as} \\ \cup_{P \in \mathcal{P}_1^*} V(P) = \mathcal{C} \end{cases} \quad (2)$$

\mathcal{P}_1^* is a path built from a set of points guaranteeing a maximum visibility of the studied environment. Indeed, each point of \mathcal{C} is visible by at least one point of \mathcal{P}_1^* . We propose to adapt the art gallery problem to maximize the visibility. The art gallery problem was presented as a question by the German painter Paul Klee: how much guards do we need to guard a room ?

This problem consists in finding in a polygon the optimal number of guards required to view the totality of the polygon. A variety of this problem is known as the "Watchman route". This problem consists in finding a path inside a polygon Γ so that each point in Γ will be visible from a certain point of this path \mathcal{P}_1^* . To guarantee a maximum of visibility in our 3D scene, we propose to calculate the watchman route in the polygon Γ that represents the navigable areas of the scene. Several algorithms have been proposed to calculate the watchman route by taking as input a group of target points to visit [5], [8], [17]. The majority of all these algorithms are looking for the optimal watchman route that minimizes the total traveled distance. In our case, we must ensure the visibility of the edges inside the polygon and also the visibility outside the polygon. With this intention, we propose to decompose the start polygon into k convex polygons. To simplify the problem, we consider that the visibility through the exteriors edges of the polygons is possible and we consider that the interiors edges of the polygons as obstacles.

To visualize the totality of a convex area, only one point of view (the center) is sufficient. From this statement, the path will be built from the points of view P_k selected in each convex area. Several algorithms have been proposed to decompose a polygon in multiple areas. Keil's algorithm [13], [14]] consists in splitting a polygon into many convex sub-polygons by eliminating all reflex vertices. In a polygon, a vertex is called "convex" if the internal angle of the polygon, that is, the angle formed by the two edges at the vertex, is less than π radians. otherwise, it is called "reflex".

Keil's algorithm works by examining every possible way to remove the reflexivity of these vertices. We suggest the use of Mark Bayazit's algorithm [1] which is an optimized version of the Keil's algorithm. The best point of view in a convex sub-polygon is the center of this convex area. By connecting each center of the under-polygons with the nearest center, we will obtain the watchman route \mathcal{P}_1^* . This path enables us to guarantee the total coverage of the studied environment. Figure 2 shows an example of a polygon decomposition in multiple convex areas and the associated watchman route. This set of points \mathcal{P}_1^* becomes a new constraint for the optimization of the entropy E .

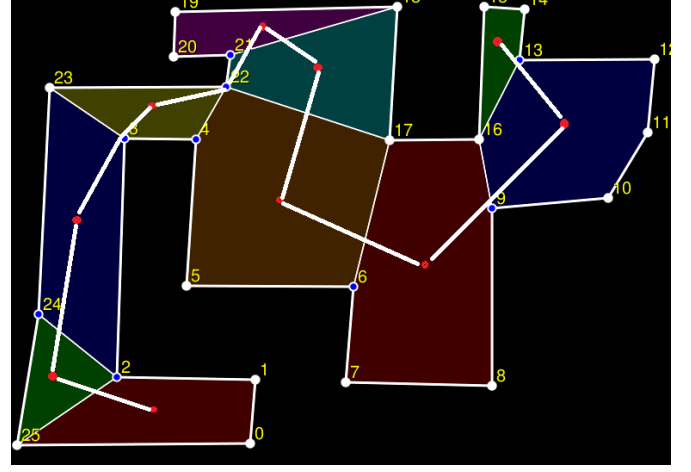


Fig. 2. Calculation of watchman route in a polygon

B. Entropy Optimization

We intend to find out among the points on the watchman route \mathcal{P}_1^* those which maximize the entropy E while ensuring that the majority of the environment areas are still visible. The expression of this new problem is:

$$f_2 : \begin{cases} \mathcal{P}_2^* = \{P_l \in \mathcal{P}_1^*, l = 1..n/ E(P_l) \geq \lambda, \lambda \in [0..1]\} \\ \text{such as} \\ \#(\cup_{l=1}^n V(P_l)) \geq \beta.N, \beta \in [0..1] \end{cases} \quad (3)$$

\mathcal{P}_2^* is a set of the points (centers of spheres) allowing to find a compromise between both objectives that are visibility and entropy. λ and β are two parameters of regulation of the two criteria entropy and visibility. In order to treat the large-scale point clouds, we propose to split the initial one into several sub-clouds. For each point in \mathcal{P}_2^* , we define a cylindrical search window of radius r . Each window is a sub-cloud of P containing a part \mathcal{P}_1^* of the watchman route. The entropy criterion is optimized on \mathcal{P}_1^* of each sub-cloud. The calculation of the entropy E was detailed in our previous work [20]. Our goal is to determine the best set of viewpoints allowing the capture of a maximum amount of salient points and guaranteeing a good visibility of the environment.

The entropy optimization is carried out in each sub-cloud on $\pi = \{P \in \mathcal{P}_1^*\}$. In each sub-cloud, we aim to find on π the set of spheres that maximize the entropy while keeping a maximum visibility of the scene when we gather the visibility of all spheres on \mathcal{P}_1^* . First, we evaluate the entropy of the vertices colored with black in Figure 3 and the middle of each pair of consecutive vertices and their centers colored in blue, we only keep two points having the highest entropy. for every sub-cloud, we select vertices with entropy higher than λ . In each iteration, we compute the number of all points visible from the selected vertices colored in violet. Only the spheres whose union of their visibility is higher than βN are selected. Figure 3 shows the different steps of this optimization process. The last step

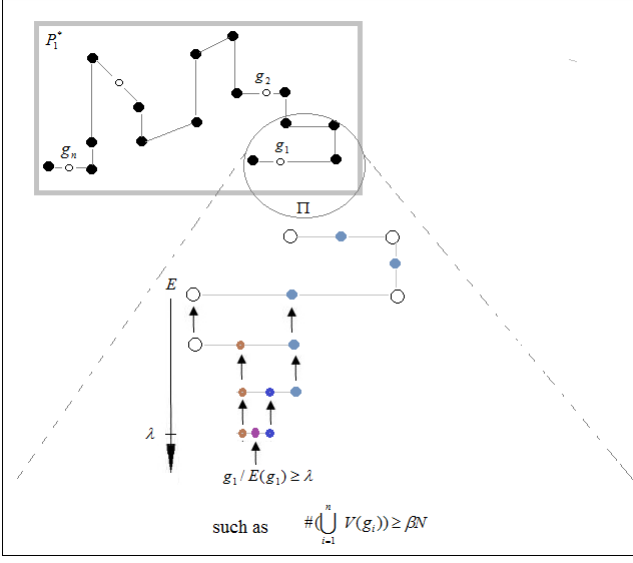


Fig. 3. Entropy Optimization

of optimization is the minimization of the size n of P_2^* which represents the number of spheres in g_n .

IV. MINIMIZING THE NUMBER OF SPHERES

In this section, we present the last step of our algorithm allowing to minimize of the size n of P_2^* . The output of this optimization process is the optimal set of spheres g_n with minimal size n allowing a good visibility of the complete scene and capturing the maximum number of salient points. In addition, we have proposed to calculate the similarity between neighbored spheres to eliminate redundant information. The similarity between two spheres is computed using their statistical correlation. This correlation expresses the rate of variation of photometric, semantic and geometrical information between the compared spheres. In [15], this parameter is calculated using the Median Absolute Deviation operator (MAD) which represents the difference (error) of intensity between the two spheres S_{phot} . Theoretically, two spheres are considered similar if the value of MAD is lower than a threshold σ . In the same way, we propose to add a semantic measure of similarity S_{sem} between two spheres. This measure consists in calculating the number of similar pixels with the same labels. We also added a geometrical similarity measure by using the geometrical descriptor FPFH [19]. By comparing the histograms of each sphere, we obtain a measure of geometrical similarity S_{geom} .

$$\begin{aligned} S_{phot} &= \text{med}(|p(x) - \text{med}(p(x))|) \\ S_{sem} &= \text{med}(|s(x) - \text{med}(s(x))|) \\ S_{geom} &= \text{med}(|g(x) - \text{med}(g(x))|) \\ S &= \text{mean}(S_{phot} + S_{geom} + S_{sem}) \end{aligned} \quad (4)$$

In these equations, $p(x)$, $s(x)$ and $g(x)$ are respectively the vectors containing the errors. The expression of the problem

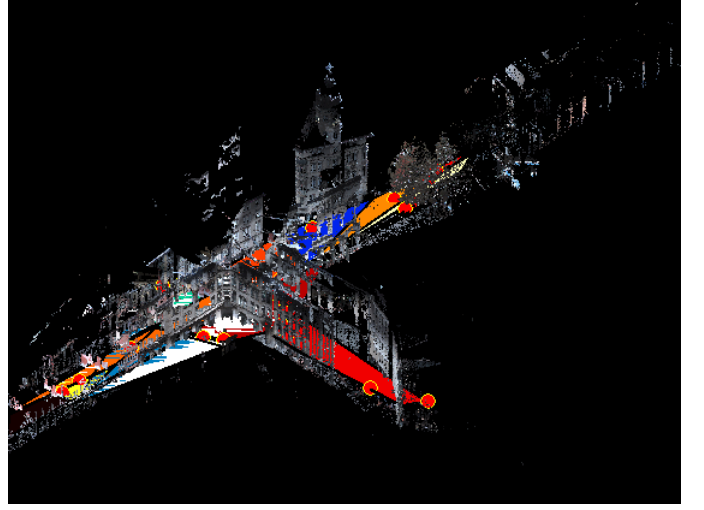


Fig. 4. Polygon Extraction And Spheres Positioning

is the following:

$$f_3 : \{ \min \#(\{P \in P_2^* \mid \cup_P V(P) \geq \beta \cdot \Gamma, S \leq 0.5, \beta \in [0..1]\}) \} \quad (5)$$

By removing similar spheres that have the same visibility, we will get an optimal set of spheres.

V. EXPERIMENTAL RESULTS

A. Evaluation

To evaluate the quality of a map and estimate its performances during the localization, several solutions have been proposed. The solution suggested by Zhen et al. [23] consists in calculating a requirement called localizability. This criterion is calculated from the geometric characteristics in each direction for each point in the map. We also propose to evaluate the quality of our solution through the calculation of the recall and the precision as defined in our previous work [20]. We also calculate the compression rate between the initial cloud and the final set of spherical images.

B. Results

We have applied our method to a large environment. This large-scale point cloud contains over 50 millions of 3D labeled points and represents a trajectory of 80 m. In this dataset, we have 8 classes of labels, namely {1: man-made terrain, 2: natural terrain, 3: high vegetation, 4: low vegetation, 5: buildings, 6: hard scape, 7: scanning artefacts, 8: cars}. An additional label {0: unlabeled points} unlabeled points. In our summarizing process, points labeled "buildings", are considered among the most salient points for localization. This dataset permitted the evaluation of our solution's performance by using all semantic, photometric and geometric characteristics together. Figure 4 shows the decomposition of the polygon (navigable areas) into several convex polygons.

Each color represents a different polygon. The output of this summary process is a compact set of 20 spherical images. The mean distance between all the spheres is around 4 m. We have obtained a compression ratio of 90% of this map. We also tested our algorithm on our own database which describes a part of Rouen city as shown in figure 5.

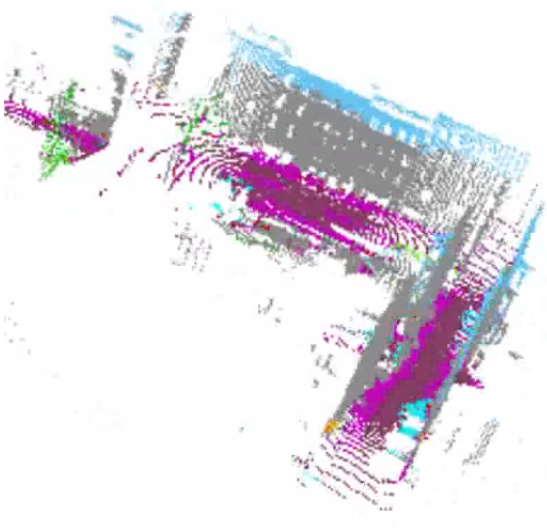


Fig. 5. Rouen's city 3D Point Cloud

It covers a trajectory of 200 m of a developed urban area that contains urban structures such as houses, buildings, and roads. This database is a 3D textured and labeled cloud and contains over 7 millions 3D point. In this dataset, we have 15 classes of labels, namely {0: Misc, 1: Road, 2: RoadMarks, 3: Terrain sidewalk, 4: Building, 5: vegetation, 6: Tree, 7: Pedestrian, 8: cars, 9: Bicycle, 10: Motorbike, 11: TrafficLight, 12: TrafficSign, 13: Pole, 14: sky}. Each class is represented by a color code as indicated in the figure 6. As a result, we have obtained we measured up to 54 spherical images with a mean distance between all the spheres of 3,7 m. Figure 7 shows an example of RGB-D-L spherical images. We have obtained a compression ratio of 98% of this map. Compared to our previous work [3], we measured up to 3% improvement in the compression performance. Since the most significant inspiration for spherical representation comes from work of Meilland et.al [15], we decided to compare our results with their work. We have found a 2% improvement in the compression ratio due to the optimal positioning of spherical view-points in the scene. The final step is to build the navigability graph by connecting the spheres inside the navigability polygon with edges. Each edge is weighted by the distance between the two corresponding spheres as shown in figure 8. This graph can be used in several applications such as navigation or localisation of mobile robot thanks to its minimal size and the relevance of the data it contains. To evaluate our solution, we have computed Recall and Precision [20]. To do that, we have built our ground truth dataset. For each point in the first dataset,

0	Misc	black: 0 0 0
1	Road	color: 140 60 100
2	RoadMarks	color: 255 200 255
3	Terrain/sidewalk	color: 210 0 200
4	Building	color: 140 140 140
5	Vegetation	color: 90 240 0
6	Tree	color: 0 190 0
7	Pedestrian	color: 190 10 30
8	Car	color: 10 240 240
9	Bicycle	color: 100 100 0
10	Motorbike	color: 100 0 100
11	TrafficLight	color: 130 0 180
12	TrafficSign	color: 255 255 0
13	Pole	color: 255 130 0
14	Sky	color: 90 200 255

Fig. 6. Semantic segmentation

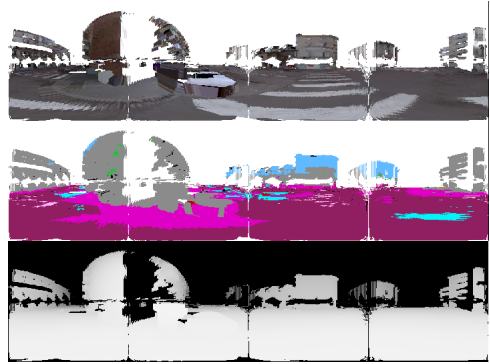


Fig. 7. RGB-D-L Spherical Image

we have attributed a label {0: irrelevant for localization, 1: relevant for localization}. Most of the relevant points belong to buildings thanks to their geometric shapes. As a result, we have significantly decreased the size of the map. Nevertheless, we have succeeded to keep a maximum number of salient points. We have achieved a Recall of 60% and Precision of 80 % while guaranteeing maximum visibility of the environment. All the spheres are positioned in a way to capture the maximum possible points belonging to the facades of the buildings. For the first dataset, the obtained set of spheres allow viewing 80% of the scene ($\beta = 0.8$). The choice of the β has an impact on the compression ratio. In fact, setting β to 1 will favor the visibility on the entropy. Thus the number of selected spheres will increase to cover all the environment and thus their global size. The choice of the λ has an impact on the relevance of the 3D points in the summary map and consequently on the recall and the precision. Indeed if λ is set to 0.5 at least half the points will be salient. therefore, we have chosen $\lambda = 0.7$. For the second dataset, we have computed the Localizability of each spherical image, as defined in [23], all spheres are easily localizable since they all have high localizability values. This ensures a good localization in the sphere graph. Our algorithm guarantees maximum visibility and entropy of

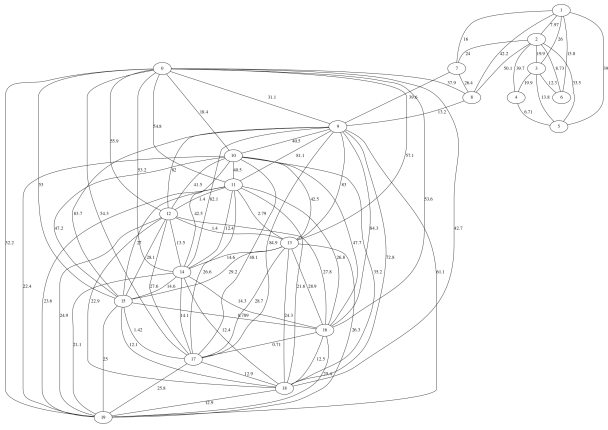


Fig. 8. Navigability Graph

the studied environment, thus an optimal image registration. Reducing the size of the initial map allows to gain in term of memory and time of computation. Indeed we can integrate easily this minimal graph on navigation system with low resources or send it over a network from a server to an agent.

VI. CONCLUSIONS

The developed method throughout this paper allows summarizing efficiently a large-scale point cloud. The summarizing process is based on the extraction of several spherical views representing sub-clouds of the initial map. This spherical representation contains semantic, photometric and geometric information. This new method of summarizing 3D maps allows us to facilitate several navigation tasks when applied in intelligent transportation systems (localization, route planning, obstacle avoidance, ...) by reducing significantly the calculation time and the required memory size. We also believe that using semantic information permits the development of a precise summary map by rejecting unnecessary localization data such as points belonging to dynamic objects (cars, pedestrians, ...). In our future work, we aim to extend our algorithm to treat the case where the navigable areas that would contain holes. Using the set of spherical views and the navigable area to produce a navigation graph can facilitate multiple navigation tasks especially on navigation systems with limited resources. Using this graph we can propose an adapted path to be followed by the agent according to its mobility capabilities (car, pedestrian, bike, mobile robot, ...).

VII. ACKNOWLEDGMENTS

This work takes part in the ANR-15-CE23-0010-01 pLaT-INUM project. This project has been funded with the support from the French National Research Agency (NRA).

REFERENCES

- [1] Mark Bayazit. Decomposing polygons into convex regions. <https://mpen.ca/406/bayazit>.
- [2] Imeen Ben Salah, Pascal Vasseur, Cedric Demonceaux, and Sebastien Kramm. Extraction d'un graphe de navigabilité à partir d'un nuage de points 3d enrichis. *ORASIS*, 2017.
- [3] Imeen Ben Salah, Pascal Vasseur, Cedric Demonceaux, and Sebastien Kramm. Summarizing large scale 3d point cloud for navigation tasks. In *Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

- [4] Selim Benhimane, Alexander Ladikos, Vincent Lepetit, and Nassir Navab. Linear and quadratic subsets for template-based tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.
- [5] Svante Carlsson, Håkan Jonsson, and Bengt J Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry*, 22(3):377–402, 1999.
- [6] Dana Cobzas, Hong Zhang, and Martin Jagersand. Image-based localization with depth-enhanced image map. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1570–1575. IEEE, 2003.
- [7] Luis Contreras and Walterio Mayol-Cuevas. O-poco: Online point cloud compression mapping for visual odometry and slam. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4509–4514. IEEE, 2017.
- [8] Moshe Dror, Alon Efrat, Anna Lubiwi, and Joseph SB Mitchell. Touring a sequence of polygons. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 473–482. ACM, 2003.
- [9] Marcin Dymczyk, Simon Lynen, Michael Bosse, and Roland Siegwart. Keep it brief: Scalable creation of compressed localization maps. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2536–2542. IEEE, 2015.
- [10] Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, and Paul Furgale. The gist of maps-summarizing experience for lifelong localization. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2767–2773. IEEE, 2015.
- [11] Michael P Fourman. Compaction of symbolic layout using genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 141–153. L. Erlbaum Associates Inc., 1985.
- [12] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
- [13] J Mark Keil. Polygon decomposition. *Handbook of Computational Geometry*, 2:491–518, 2000.
- [14] Mark Keil and Jack Snoeyink. On the time bound for convex decomposition of simple polygons. *International Journal of Computational Geometry & Applications*, 12(03):181–192, 2002.
- [15] 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.
- [16] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [17] Eli Packer. Computing multiple watchman routes. *Lecture Notes in Computer Science*, 5038:114–128, 2008.
- [18] Katrin Pirker, Matthias Rüther, and Horst Bischof. Cd slam-continuous localization and mapping in a dynamic world. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3990–3997. IEEE, 2011.
- [19] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.
- [20] Imeen Salah, Sebastien Kramm, Cédric Demonceaux, and Pascal Vasseur. Summarizing large scale 3d point cloud for navigation tasks. In *IEEE 20th International Conference on Intelligent Transportation Systems*, 2017.
- [21] Heiko G Seif and Xiaolong Hu. Autonomous driving in the icityhd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159–162, 2016.
- [22] Ted J Steiner, Guoquan Huang, and John J Leonard. Location utility-based map reduction. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 479–486. IEEE, 2015.
- [23] Weikun Zhen, Sam Zeng, and Sebastian Soberer. Robust localization and localizability estimation with a rotating laser scanner. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6240–6245. IEEE, 2017.