Summarizing large scale 3D mesh for urban navigation

Imeen Ben Salah, Sébastien Kramm, Cédric Demonceaux, Pascal Vasseur

Version of Record: https://www.sciencedirect.com/science/article/pii/S0921889022000136
Manuscript_75cc9a7ba8001a4e1d2bfcd037625581

Summarizing Large Scale 3D Mesh for Urban Navigation

Imeen Ben Salah[a], Sébastien Kramm[a], Cédric Demonceaux[b], Pascal Vasseur[c,a]

[a] *Laboratoire d'Informatique de Traitement de l'Information et des Systemes Normandie Univ UNIROUEN UNIHAVRE INSA Rouen LITIS 76000 Rouen France*
[b] *Laboratoire ImViA Universite de Bourgogne Franche-Comte 71200 Le Creusot France*
[c] *Laboratoire MIS UR4290 Universite de Picardie Jules Verne 80000 Amiens France*

Abstract

Cameras have become increasingly common in vehicles, smartphones, and advanced driver assistance systems. The areas of application of these cameras in the world of intelligent transportation systems are becoming more and more varied : pedestrian detection, line crossing detection, navigation,... A major area of research currently focuses on mapping that is essential for localization and navigation. However, this step generates an important problem of memory management. Indeed, the memory space required to accommodate the map of a small city is measured in tens gigabytes. In addition, several providers today are competing to produce High-Definition (HD) maps. These maps offer a rich and detailed representation of the environment for highly accurate localization. However, they require a large storage capacity and high transmission and update costs. To overcome these problems, we propose a solution to summarize this type of map by reducing the size while maintaining the relevance of the data for navigation based on vision only. The summary consists in a set of spherical images augmented by depth and semantic information and allowing to keep the same level of visibility in every directions. These spheres are used as landmarks to offer guidance information to a distant agent. They then have to guarantee, at a lower cost, a good level of precision and speed during navigation. Some experiments on real data demonstrate the feasibility for obtaining a summarized map while maintaining a localization with interesting performances.

Keywords: 3D Mapping, Summarized mapping, Localization based vision.

1. Introduction

Environment mapping is a challenging and important question that has been widely discussed in mobile robotics research. Most of the early vision-based navigation systems need a map that contains a sufficient level of detail for an accurate navigation. Over the last few years, the introduction of High-Definition (HD) [1] and semantic maps have greatly participated in the large commercial success of navigation and mapping products and also in the enhancement of data fusion based localization algorithms. Several digital map suppliers like TomTom, Here, NVidia and Carmera are now providing HD maps with very high navigation accuracy, especially in challenging urban environments. These HD maps provide a detailed representation of the environment and thus require a high processing capacity with severe time constraints as well as large storage requirements. In order to use this type of maps in navigation systems with limited resources (computation / memory), the size of these maps must be reduced or summarized while preserving the essential information for navigation. On another side, GPS (Global Positioning System) is now used in all types of navigation systems. However, the accuracy of this low-cost sensor is of the order of a few tens of meters which prevents accurate guidance of an agent in an urban environment. Localization and perception tasks can be solved quite easily by adding new sensors such as LiDAR (Light Detection and Ranging) able to provide the 3D structure of the environment with a high level of accuracy and a high acquisition speed. However, these sensors are still expensive and their integration would be difficult for pedestrian and smartphone applications. On the other side, cameras have become more and more common in navigation systems and are also available on smartphones, enabling new possibilities for individual mobility.

In this context, the pLaTINUM project focuses on the development of methods and algorithms for mapping an urban environment, enriching it and updating it automatically using many communicating visual sensors. After scanning the environment using a specific vehicle equipped with multiple sensors such as cameras, LiDAR, GPS and IMU, the extraction of a textured and semantic 3D mesh is performed [2]. In this work, we present how this 3D mesh can be summarized as a navigability graph. Our main contributions are the introduction of a new algorithm for summarizing maps based on geometric, photometric and semantic characteristics and the use of a new augmented and labelled RGB-D-L spherical representation as nodes of the

graph. This representation contains three types of information : color (RGB), semantic label (L) and depth (D).
This paper is organized as follows: in the next section, we present a state of the art about mapping as well as large-scale map summarizing methods. We develop our different contributions in Section 3. In Section 4, we present some results obtained with the proposed solution as well a discussion before some perspectives and a conclusion in Section 5.

2. State Of The Art

Mapping consists in developing a representation of the perceived environment allowing a future safe navigation. We first summarize the different types of methods used for representation and mapping of the environment using visual information before focusing on the methods for summarizing a 3D map.

2.1. Existing Mapping Methods

In the literature, several methods have been proposed to represent the environment. Depending on the type of spatial relationship between the basic entities (walls, buildings, trees, etc.) that constitute the studied environment, we distinguish two different spatial representation approaches. The first approach is based on the coordinates in an absolute reference frame (metric representations) and the second approach is based on the definition of a relative reference frame linking the observed base entities together (topological map).

One of the metric representation is called the occupancy grid method [3, 4, 5, 6]. This method consists in representing the environment in the form of a grid in which each basic object (entity) is represented by a cell of the same shape and size.

Another metric representation is called geometric map [7, 8, 9] that consists in constructing a representation of space using a set of geometric objects (points, lines, curves, planes, ...) located in the same coordinate system.

The topological representation [10, 11] is another type of map representations.In this type of map, the environment is transformed into a graph illustrating the relationships between its different entities. These entities are classified into groups called "nodes" according to certain well-defined criteria (co-visibility, distance, type, ...). These nodes, representing distinct

locations in the environment, are connected to each other by edges allowing accessibility between them.

Another type of representation called semantic map [12] has been recently introduced. The use of semantic information (road signs, building, streets, ...) allows to create a synergistic interaction between humans and robots which makes navigation more efficient and easier. Previous representations present pros and cons when used separately but can be merged together in order to benefit from the maximum of their respective advantages. Different combinations have been proposed such as hierarchical [13], patchworks [14] and superposition [6].

2.2. Reducing Maps

The problem of limited resources is increasingly being addressed as maps are starting to grow in size despite the use of intermediate solutions such as external servers for map storage. Hence, the need of new methods to summarize these maps in order to reduce the resources needed to operate the system (computation/memory) while preserving the information essential for navigation and maintaining the accuracy of the navigation system is of importance.

To reduce the size of a large-scale map, possible approaches are respectively map compression and map summarizing. Summarizing map consists in using only relevant information and we will only detail the related methods in the following.

2.2.1. Map Summarizing Methods

Several approaches based on the selection of characteristic information to summarize a map for localization purposes have been presented in [15], [16], [17]. They provide functions for assigning scores to rank map elements according to their importance. This selection ensures maximum coverage of the scene. In [18], the objective is to select only the places that are particularly suitable for localization using a metric called *localization utility*. To simplify the appearance-based navigation, a selection process is applied to select key elements in the environment. Other methods based on bag of words (BoW) are widely used for localization. BoW methods can effectively represent a huge amount of data by using occurrences of several visual vocabularies. By applying a hierarchical dictionary to the visual navigation problem [19], BoW methods have demonstrated a high scalability and accuracy in vision-based

localization and mapping processes. Another possible approach called "probabilistic approach" consists in searching for the most compact subset of 3D point could that represents the entire environment by providing the necessary number of useful points for solving the Perspective-n-Points algorithm [20]. In this approach, the final compact map is therefore composed of all the 3D points with the highest probability of appearing in the key frames using a score assigned to each point of interest. In some applications, the use of this heuristic method has given impressive compression results (up to 99%) while providing good localization quality. Another approach using a technique called "image-based memory" [21, 22, 23] could be used to produce a compact summary of a map. In this localization approach, each position estimation is calculated with reference to a key frame acquired during a learning phase. A small set of images generated by combining information of a different nature will be used in the localization instead of the global map of the environment. In the example of Cobzas[24], a panoramic image memory is created by combining the images acquired by the camera with depth information extracted from a laser scanner. In this image database, only the information essential to the navigation process will be retained [25]. This allows to obtain homogeneous results with the same properties (accuracy, convergence, robustness, etc.) as the original global map when performing localization. Several works have confirmed the interest and efficiency of this image-based memory method. That's why we have chosen to adopt this strategy to summarize our initial mesh. However, in the literature, several techniques can be used to build this image-based memory. We chose to study these existing techniques in order to choose the one which ensures maximum efficiency in the choice of useful information. In [26, 27, 28], a spherical representation was proposed to summarize an initial map.

Several solutions in the literature have suggested to select salient areas in an environment in order to select the best locations of the key images. Most of these solutions are mainly based on a concept known as visual saliency. In the following section, we detail the existing methods dealing with 3D saliency extraction.

2.2.2. Selection of Relevant Information : Visual Saliency

Visual saliency is the ability of our perception system to distinguish objects from their surroundings according to their characteristics and to immediately attract our attention.

When processing a 3D point cloud or a textured mesh, the selection of points that will form the compact map is based on the choice of the most representative points of the environment. These points are generally referred as key points or points of interest. In the literature, many algorithms have been proposed to detect 3D visual saliency. 3D salient detection models can be grouped according to the types of input data. A first model represents the environment as a set of augmented RGB images with their D depth maps. In this case, several methods have been proposed to detect 3D saliency in $RGB - D$ images [29], [30], [31]. A second model uses a 3D mesh of the environment to extract the protruding areas [32], [33], [34]. The third model consists in using a 3D point cloud as an input to produce a saliency map [35], [36], [37]. In the following, we detail only the methods that deal with 3D point cloud and 3D mesh.

The human visual system can easily perceive surfaces and geometric shapes. This is why several studies have proposed to build a 3D mesh from a 3D point cloud in order to produce a compact representation of the environment. The first work on saliency detection for 3D models was inspired by calculating 2D saliency in an image based on edge detection and producing smooth surfaces from the data. Other works have been proposed to calculate the saliency in a mesh using surface variations. In [38], the saliency is calculated using the concept of dissimilarity of a central region with its neighbouring regions. They have proposed curvature as a main criterion for calculating saliency and selecting optimal viewpoints. In [39], a method for mapping between different mesh areas has been proposed. Each mesh part is geometrically represented using a descriptor based on its curvature and the variation of this curvature in relation to its environment. Several works have been done to find the optimal viewpoint in a scene. Among these works, we mention the work of [40]. In this work, the authors proposed a method for calculating saliency on a 3D surface by detecting protruding regions. They compare the different patches of a 3D surface with a database built by several salient patches classified according to their belonging to the same object. In [41] a method of predicting the saliency in a mesh has been proposed using "Schelling Points" on the 3D surface. A database of these points is built from data collected as part of a coordination set where users choose the points they think they will be selected by other users. An ther work [42] proposes to use region distinction to extract salient areas based on the fact that human beings are attracted to differences. Unlike most previous approaches, which only consider local distinction, this work takes into account global distinc-

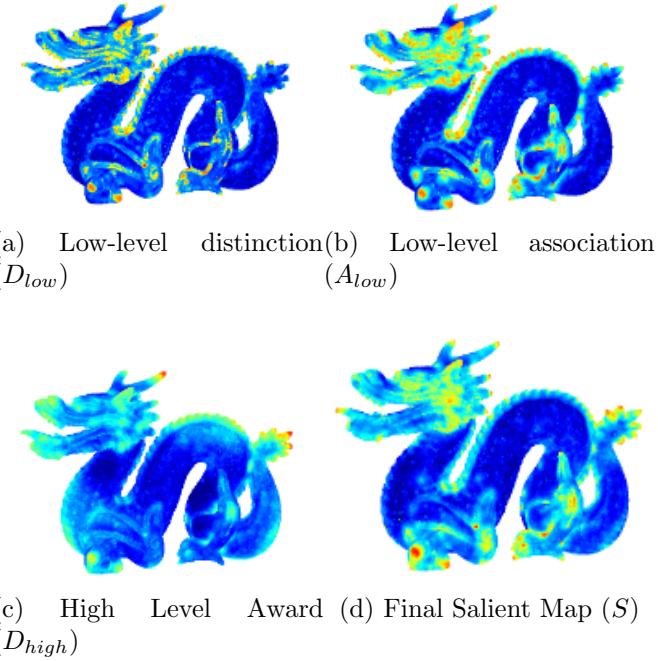
tion. In addition, the author considers the fact that visual shapes may have one or more centres of gravity on which the shape is organized, this may influence the search criterion for saliency. Generally speaking, the extremities are often considered as protruding by humans, which is why they have chosen to calculate the projection on extreme vertices of a mesh. The salient vertices are those whose geometry is unique. This is done by calculating, for each vertex, a descriptor that characterizes its shape. A vertex is distinct if its descriptor is different from all other vertex descriptors in the mesh. In [43], a new point of interest detector in a mesh was proposed. This method is based on spectral data analysis. The saliency is defined from irregularity of the Laplacian spectrum of a mesh. In addition, they use Gaussian curvature to measure local geometry to improve the localization of points of interest. Several models have been proposed in the literature to calculate global and local saliency.

In order to illustrate the detection of salient areas in a 3D point cloud, we focus on three methods based on geometric and photometric information. The first method [36] allows the saliency map to be calculated efficiently from a point cloud based on geometry only. Since human attention is attracted by differences, salient points are those points whose neighbourhood is geometrically unique compared to other neighbours.

To measure distinctness, this method uses a 3D point descriptor called FPFH (Fast Point Feature Histogram) [44] to characterize the geometry of the neighborhood of a 3D point. A point is considered as distinct if its descriptor is different from all the other descriptors in its vicinity. This algorithm for detecting saliency is hierarchical. The saliency is calculated on several levels and looks for prominent regions rather than isolated points. This consideration stems from the human tendency to group together similar elements. This operation is carried out on two levels with different neighbourhood sizes. First, a low-level distinctness is calculated to detect these small characteristics D_{low}. Then, an association value is calculated to detect salient points in the neighborhood of the most distinct points A_{low}. Then, a high-level distinctness is calculated to select the major characteristics D_{high}. Finally, the three components above are integrated into the final S saliency map defined for a point p_i as follows:

$$S(p_i) = \frac{1}{2} * (D_{low}(p_i) + A_{low}(p_i) + D_{high}(p_i)) \quad (1)$$

Figure 1 illustrates an example of detecting salient areas in a 3D point cloud. However, this method only uses the geometry of the scene without any other



(a) Low-level distinction (D_{low}) (b) Low-level association (A_{low})

(c) High Level Award (D_{high}) (d) Final Salient Map (S)

Figure 1: Steps for detecting salient areas [36]. The distinctness of low levels is first calculated by identifying small geometric features, such as teeth and tips on the head and back of the dragon. Then, the association is applied, grouping the salient points and focusing on the features of the dragon's face. Then, the high-level distinctness procedure detects larger areas, such as the tail and mouth. Finally, the maps are integrated to produce the final saliency map.

type of information such as color. In [37], the method consists in using both geometric and photometric characteristics to estimate the saliency in a cloud of colored points. This method allows to group the 3D points in a super-voxel cloud. Then, a saliency value is calculated for each supervoxel from its geometric and photometric characteristics and those of its neighbours. A measure of distinctness is calculated for each cluster by measuring its photometric and geometric contrast to that of each adjacent cluster. The contrast of a cluster C is calculated as follows:

$$\rho(C) = \theta\rho_{geo}(C) + (1-\theta)\rho_{color}(C) \tag{2}$$

where ρ_{geo} and ρ_{color} are respectively the contrasts of geometric and photometric characteristics of C. θ is a weighting parameter that is empirically set to 0.5 in [37].

A method based on photometry only has been proposed in Leroy's work [35] based solely on the rarity of supervoxels. For each supervoxel v, a measure of rarity S_i is calculated using only the photometric characteristics. Rarity is obtained over several colorimetric space representations (HSV, HLS, YUV, RGB, Lab, Luv).

$$S_i(v) = -\log(\frac{P_i}{N}) \tag{3}$$

This rarity-based mechanism consists, for each supervoxel v, in calculating the probability of cross-occurrence of each of the N supervoxel. For each color component i, the probability of the presence of v is obtained and N is the number of supervoxels. Then, an attention score is assigned to each supervoxel. This mechanism provides higher scores for rare regions. The rarity value is between 0 if all supervoxels are identical and 1 if one supervoxel is different from all the others. In the literature, several detectors and descriptors of points of interest have been used to measure saliency. We mention among these algorithms, the SUSAN detector [45], the FAST detector[46], the SIFT detector [47, 48], the SURF detector [49], the MSER detector [50], the CenSurE [51] detector, the AGAST [52] detector and the Harris3D [53] detector which is known for its simplicity and efficiency in computer vision applications. Some detectors have been adapted to process 3D data like 3D SIFT [48].

2.2.3. Optimal Viewpoint Selection

Optimal viewpoint selection has applications in several areas: visual servoing, robot motion, etc. It is difficult to define precisely the term "optimal point of view". It seems intuitive to consider that a view is said to be optimal if it provides a large amount of information for a scene.

A point of view can be considered as optimal if the amount of information it gives on a scene is maximum. And again, the term "information" is unclear. In the literature, several methods have been proposed to assess quality of a point of view in a scene. These methods can be classified according to the nature of the input information. The first family contains methods that measures the visible surfaces from a point of view to evaluate it. In [54], a point of view is considered to be optimal if it minimizes the angle gap between a direction of view and the normal at the faces of the objects in

the scene. In [55], a better viewpoint selection algorithm was introduced. This method, based on information theory, proposes to maximize a function called "Entropy" to select the optimal point of view. A second group takes into account not only the quantity of visible surfaces, but also the geometry of these surfaces. In [55] Sokolov and Plemenos have proposed to take into account the total curvature of the visible surfaces for optimal selection. In the literature, many methods rely on visual saliency to select the best points of view. In [56], a function measuring the quality of observation of an object has been introduced. This function is used to define the best point of view in a scene initially segmented into several objects. Another method proposed in [32], consists in exploiting the calculated saliency on a 3D mesh. The optimal view maximizes the sum of the saliency measures of the regions visible from this viewpoint. The saliency of a region is calculated from its curvature.

In the following, we detail our approach and the methods chosen among the ones presented in the previous section.

3. Our Approach

In this section, we present our method based on visual saliency to summarize a 3D mesh. We detail below each step of the map summary process.

3.1. Selected Approach

As shown in Figure. 2, we propose to summarize the initial 3D mesh as a graph where each node is composed of three spherical views, respectively photometric (RGB), geometric (D) and semantic (L). The advantage of this topological representation is the possibility to apply several algorithms for navigation and search for the best path, such as the Dijkstra algorithm. The advantage of this choice stems from the combination of several types of information and a level of granularity adapted to navigation systems with low memory and computing resources.

To summarize the initial 3D mesh, we then need to define an optimal viewpoint search method in order to estimate the location of the different nodes of the graph. Based on the existing approaches of visual saliency described in the previous section, we introduce our criterion for the selection of optimal viewpoint. We have selected three methods previously described that are respectively geometric, photometric and hybrid. We have conducted a comparison of these methods in a urban navigation context in [57]. The results

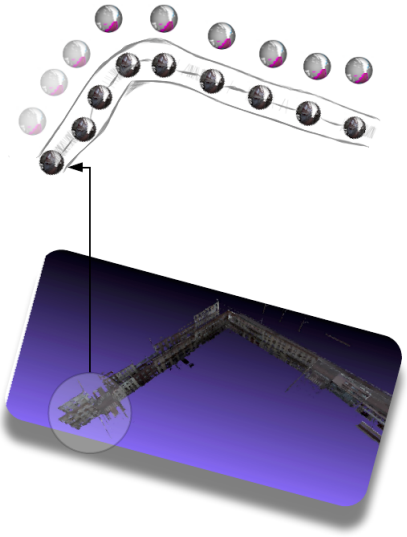


Figure 2: Our solution as a graph based on photometric, geometric and semantic spherical views. representation.

of this comparison demonstrated that methods based on geometric characteristics are more relevant than those based on photometric information. Moreover, a combination of these two types of characteristics by favouring geometric information allows to extract more salient areas suitable for navigation. We have chosen to adapt the hybrid method [37] using the equation 2. We propose to add semantic information to geometric and photometric information in the process of extracting salient areas. A measure called Entropy was defined to measure the amount of useful information visible from a given point of view [58, 59]. To judge the saliency of a 3D point on the sphere, we propose to define two levels of saliency: low-level saliency based on the low-level characteristics (photometric and geometric), and high level saliency, based on the semantic information of each 3D point. In our summary process, points labelled "buildings" are considered as the most salient points of localization. Then, we find road signs, markings and vegetation. Using these two types of saliency, we calculate the number of points of interest on the sphere according to their relevance. All existing methods are mainly based on geometric or photometric features to select the most relevant information. However, these characteristics are insufficient for a good perception and un-

Semantic \ photometric / geometric	0	1
0	n_{00}	n_{01}
1	n_{10}	n_{11}

Table 1: The possible combinations between the different levels of relevance for each point.

derstanding of the environment. A combination of geometric, photometric and semantic features when computing entropy makes it possible to find the best viewpoints in the scene. To compute the entropy of a point P, we need to find the saliency of all pixels p_i visible from P. This saliency is calculated on both levels : semantic and photometric/geometric. There are four possible combinations as shown in the table 1.

For example n_{01} denotes the number of semantically relevant points only. The entropy is given by the equation :

$$E(P) = -\alpha_i \sum_{i,j=0,1} \frac{n_{ij}}{h} log \frac{n_{ij}}{h} \tag{4}$$

$$\text{where } h = \sum_{i,j=0,1} n_{ij} \tag{5}$$

The α_i are weights assigned to each of the four combinations. In the following, we have used unit weights.

3.2. Problem Modeling

We have chosen to summarize the initial mesh in the form of a navigability graph. The nodes of this graph will be placed in an optimal way. The main idea is to exploit all the characteristics of the scene to choose the best points of view.

To modelize this problem, we introduce some notations. Let us suppose that the 3D map is a 3D mesh with color and semantic information. In this 3D mesh we define the navigable areas Γ as the 2D polygon whose edges represent the borders of the navigable areas (streets, sidewalks, pedestrian crossings etc). We define the visible areas Σ as the 2D polygon whose edges represent the borders of the visible areas. To extract the visibility and navigability areas from the 3D mesh, we use the semantic labels associated to each point. Depending on the target applications, we use for navigability either road and road markings labels for automobiles, or sidewalks labels for pedestrians.

Once we have theses areas, we extract the two polygons with the Alpha Shape [60] technique. This method is used to reconstruct the shape of a dense and unorganized set of 2D/3D points. It is a generalization of the "convex hull" concept. Alpha Shape depends on a parameter α according to which it is named. When the radius α is close to 0, each point would be a limit. On the other hand, when α is close to infinity, the boundary will be the convex hull of the set of points. In our case, we set this parameter empirically to 1 to properly extract the polygons.

Our challenge is to place the spheres on the navigability polygon while guaranteeing the relevance of the data contained in the spheres. We have modeled the problem of sphere positioning as an optimization problem of two criteria. The first criterion is the entropy previously described. The second criterion is called visibility. The purpose of this criterion is to guarantee a total visibility of the scene from the nodes of the navigability graph.

We define the visibility V of a 3D point c in the navigable areas Γ as the number of all vertices of Σ visible from this point. Geometrically, the point p is said to be visible from c if the segment pc does not intersect any obstacle or any edge of Σ. We define the visibility V as:

$$V(c) = \#\{p \in \text{ vertices of } \Sigma / p \text{ is visible from } c\} \tag{6}$$

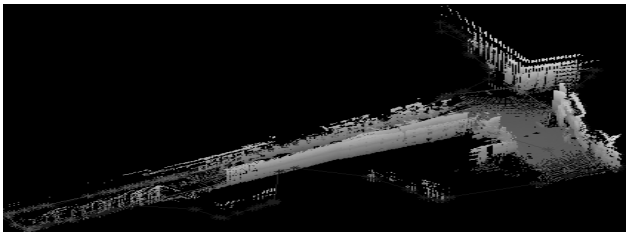
where $\#A$ is the cardinality of the set A. To reduce the search area in which we will search for the best viewpoints, we propose to compute a reduced search area. This area is in the form of a graph called barycenter graph. The next step in our approach is to place the spheres on the barycenter graph while optimizing the two criteria. In the rest of this section, we will detail the steps of the construction of the barycenter graph and our optimization algorithm.

3.3. Barycenter graph construction

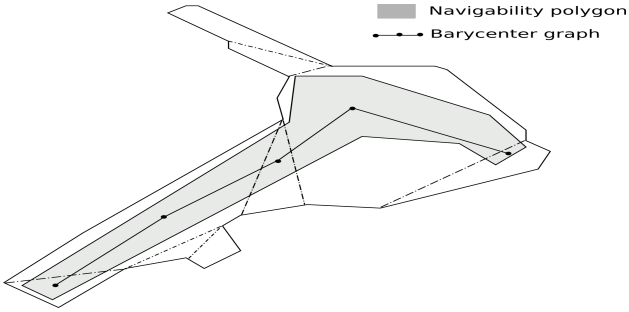
The barycenter graph is an adaptation of the well known art gallery [61] and watchman road [62] problems.

The purpose of extracting the barycenter graph is to reduce the search space. This graph ensures that each point of the visibility polygon is visible by at least one point on that path. The singularity of this graph is that these nodes must be placed on the navigation polygon while ensuring the visibility of the scene by using the visibility polygon.

In the literature, there are no methods that address the visibility problem in two polygons. In our case we have an navigability polygon included in the



(a)



(b)

Figure 3: Example of barycenter graph. (a) The initial mesh (b) the visibility polygon including the navigability polygon and the associated barycenter graph.

visibility polygon. Our main idea is to decompose the visibility polygon into several convex polygons. Then, we calculate the intersections between each convex polygon and the navigability polygon. The nodes of this graph are the barycenters of these intersection zones. Figure. 3 shows an example of the barycenter graph extraction. The polygons of navigability and visibility are extracted from the 3D mesh. The visibility polygon is then decomposed into 9 convex areas and the barycenter graph is deduced from the intersection between each convex zone with the navigability polygon.

3.4. Navigability Graph Extraction

Until now, there are no methods in literature allowing to summarize an initial voluminous map by taking into account simultaneously the structure and the perception of the scene. The main idea of our solution is to use geometric, photometric and semantic characteristics to determine the best points of view in a scene. These viewpoints are the centers of the spherical images that represent the nodes of the navigability graph. This graph allows to keep a global visibility of the scene while minimizing the number of these

nodes.

The optimization process consists in finding the best locations of the RGB-D-L spheres on the barycenter graph. Each edge of this graph is discretized in order to select the points allowing to reach a total visibility of all the vertices of the visibility polygon. These selected points must guarantee the visibility of relevant information for navigation. In addition to the entropy, a second criterion called visibility is added. The purpose of this criterion is to ensure total visibility of the scene.

We have proposed a multi-objective optimization method consisting in simultaneously optimizing the both Visibility and Entropy [59].

Algorithm 1 finds the optimal set of spherical images $X = \{X_i, i = 1..n\}$ which allows to describe efficiently a 3D mesh. Here we represent each sphere by its center X_i. Our proposed method consists in taking as input the barycenter graph $T = (C, \gamma)$ as search space instead of taking the whole navigability polygon Σ. T is represented as a set of nodes C belonging to the navigability polygon. $\gamma = (\gamma_k, k = 1..m)$ represents the set of edges connecting these nodes. In an iterative way, the edges of this graph are discretized in a set of points. We keep only the points that maximize the two criteria Entropy and Visibility. The optimization algorithm consists in selecting a start point and an end point on the initial graph T. Starting from the start point, we select points of view with entropy greater than a threshold α and covering some uncovered vertices of the visibility polygon Σ. The value of this threshold α is chosen in an empirical way. To do that, we have built our ground truth dataset. For each point in the this dataset, we have manually attributed a label {0: irrelevant for localization, 1: relevant for localization} based to their geometric shapes. Then, we have computed Recall and Precision [57] with different values of α. We have found that the choice of the α has an impact on the relevance of the 3D points in the summary map and consequently on the recall and the precision. We have chosen $\alpha = 0.7$ that gave us the best results. These steps are repeated until we have no more edges to be treated or if we have obtained the total visibility of Σ.

4. Results

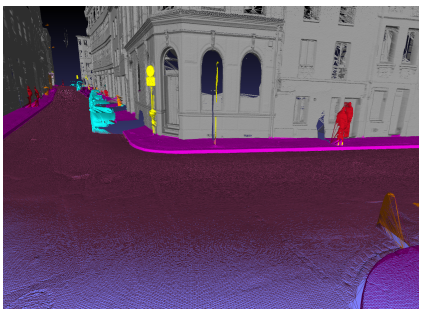
4.1. Databases

We have applied our method to two databases that describe two parts of the city of Rouen. They cover together approximately $600m$ of a urban area. These databases are in the form of a textured, triangulated and labeled 3D

Algorithm 1 Compute the sphere graph $X = \{X_i, i = 1..n\}$

Require:
$T = (C, \gamma)$ Barycenters graph
$\Sigma = \{V_j, j = 1..k\} V_i$ vertices of Σ
Starting point $C_s \in C$
End point $C_e \in C$

Ensure:
Mark all vertices in Σ as uncovered
Mark all vertices in Σ visible from C_s and C_e as covered
$X \leftarrow \{C_s, C_e\}$
$q \leftarrow C_s$
$\gamma_k \leftarrow$ the edge in γ whose starting point is q
while $\exists x$ an uncovered point in Σ **do**
  $\Pi(\gamma_k)$ =Discretization of the edge γ_k
  $p \leftarrow$ first point in $\Pi(\gamma_k)$ to the right of q such that p cover some uncovered
  $x \in \Sigma$ and $E(p) \geq \alpha$
  $X \leftarrow X \cup \{q\}$
  Mark all points $\in x$ as covered
  $q \leftarrow p$
  if $q \in C$ **then**
    $\gamma_k \leftarrow$ the edge in γ whose starting point is q
  end if
end while
return X



(a)

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

(b)

Figure 4: Large-scale 3D mesh representing a large area in the city of Rouen and associated semantic labels.

mesh resulting from the fusion of oriented images and geo-referenced LiDAR scans acquired by a mobile mapping vehicle [63]. The 3D mesh contains three types of information: radiometric (RGB from images and reflectance from LiDAR), geometric (from LiDAR), and semantic (from machine learning) The first input database (fig. 4) contains more than 5 million triangles (faces). The second database is larger than the first one and contains more than 30 million of triangles. It represents a trajectory of $500m$. To test the registration in the navigability graph, we will use several sequences of agent images. The first agent image sequence was extracted from the RGB-D-L spheres synthesized from the mesh. The second set of images is a real sequence acquired by a stereo fisheye system recorded with $15fps$. This system has been installed on a vehicle with a rigid support structure. The third sequence has been acquired by a ZED camera carried on the chest to make a pedestrian trajectory. The images were recorded with $15fps$ using an hp Zbook computer.

In the remainder of this manuscript, we detail the various experiments performed to test the performance of the registration in a graph of spheres.

4.2. Compression Ratio

To evaluate the memory gain, we have calculated the compression ratio (CR) expressed as a percentage. It is defined as follows (eq.7):

$$CR = (1 - \frac{S_f}{S_i}) * 100 \tag{7}$$

S_i is the size in bytes of the initial 3D map. S_f is the size of the spherical images in the final navigability graph. This graph is the result of the summary process of the initial map. For the first database, we have obtained 10 RGB-D-L spheres summarizing the 3D mesh. We have obtained 52 spherical images summarizing the second database. The compression ratio of this map is 98.7%. The fig. 5 shows an example of an RGB-D-L sphere. We have

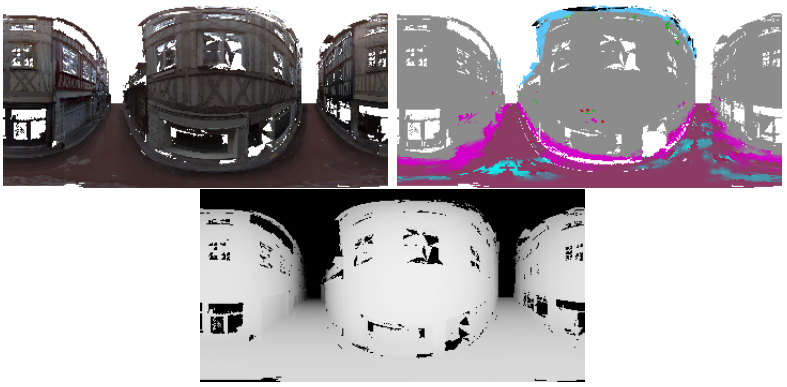


Figure 5: RGB-D-L Spherical Image

applied our summarizing method on several databases and we have achieved significant compression ratios, consistently above 89%. However, these compression ratios are strongly related to the resolution of the chosen RGB-D-L spheres. Generally a resolution of 2000×1000 pixels is sufficient to have a compromise between spherical image quality and compression ratio.

4.3. Localizability

The localizability is a criterion used to predict the performance of localization from a given point of view. It was proposed by Zhen and *et al.* in [64] consists in a measure based on the number of points of interest visible from that viewpoint and its closest neighbours. The localizability is calculated by analyzing the geometric characteristics (normal) of the points of interest visible from that point of view. The surface normals of these points of interest will be calculated to form the matrix N.

$$N = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ . & . & . \\ . & . & . \\ . & . & . \\ n_{kx} & n_{ky} & n_{kz} \end{bmatrix}$$

Then the localizability is calculated as the minimum singular value ($U\Sigma V^T$) of N.

$$L = min(diag(\Sigma)) \tag{8}$$

This measure varies between 0 and 1, this latter value indicating the best localizability. We have calculated the mean value of this criterion for all the spherical images obtained in the navigability graph. The average localizability is between 0.6 and 0.7. This value indicates that the spheres have geometrical characteristics that allow the localization of these spheres. We can easily integrate this minimal graph on a low-resource navigation system or send it over a network from a server to an agent. The proposed algorithm is very simple to implement and adapted for large-scale environments. However, it would be interesting to analyze the utility of the navigability graph. To do this, we have tested the localization in a RGB-D-L sphere graph. In the next section, we present the 3D registration results in a navigability graph.

4.4. 3D Registration

Evaluating the obtained navigability graph by calculating the compression ratio and localizability is insufficient to judge its relevance. We propose to use the navigability graph in a context of localization and navigation. The first step consists in selecting the spherical reference image. This image is the closest one to the agent image. To find this key image, a first step of Content-Based Image Retrieval (CBIR) is performed. We compare the captured agent's image with the set of geo-referenced spherical images in the RGB space. Then, a registration step is performed to estimate the transformation between the model data and the query data. To do this, we have used an algorithm called Go-ICP [65] which is a variant of the ICP (Iterative Closest Point) [66] algorithm. The Go-ICP algorithm can solve the local minimum of the ICP algorithm.

4.4.1. Go-ICP

We chose the Go-ICP algorithm [65] to test the navigation in a RGB-D-L sphere graph. One of the strong points of this algorithm is its robustness to changes according to brightness in an urban scene. Indeed, the use of scene geometry can guarantee a better convergence towards a precise solution. To evaluate our navigability graph, we propose to perform the registration of an agent sequence image in the graph of spheres. Our goal is to test if an agent can be located precisely in an RGB-D-L sphere graph. This optimal graph is the result of our algorithm of spheres positioning. We perform the registration of an agent image with one or more RGB-D-L spheres. This registration is done separately from one agent image to another. We have tested several agent image sequences acquired with different systems as explained previously. In the synthesized sequence, we have semantic information for each pixel of the spherical image as well for the agent image. We propose to use this semantic information to filter out unnecessary data for the localization before registration. The semantic filtering is a pre-processing step applied to the RGB-D-L spheres or to agent image before the registration. The fig. 6 shows the 3D registration steps as well as the semantic filtering step. This

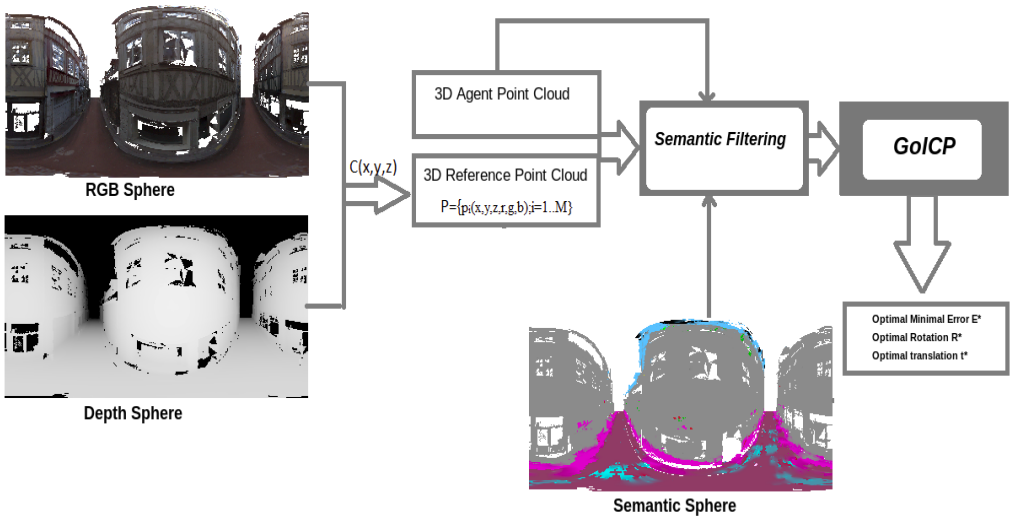


Figure 6: Semantic filtering and 3D registration from spherical images RGB-D-L and a 3D agent point cloud. First, the reference point cloud is extracted from the RGB and D spheres. Then the L sphere is used to discard points that are labelled with unwanted classes (pedestrians, cars,...)

	LS	MLE
Synthesized Image	LS_{wf} : 56% LS_f : 80%	MLE_{wf} : 1.36 MLE_{wf} : 0.7
Fisheye	LS_{wf} : 40%	MLE_{wf} : 4.2
ZED Camera (Car)	LS_{wf} : 71%	MLE_{wf} : 1.5
ZED Camera (Pedestrian)	LS_{wf} : 63%	MLE_{wf} : 1.8

Table 2: Registration Results: The localization score LS (%), The Mean Localization Error MLE (m). The test is performed with (f) or without semantic filtering (wf)

operation consists in exploiting the semantic information present in the input data. This filtering step eliminates pixels that are not useful for navigation. Indeed, pixels belonging to mobile objects such as car, motorcycle, bicycle and pedestrian classes are considered useless for the localization in opposition to the building and vegetation classes. Other classes have been removed such as the sky class. As a result, we have eliminated about 20% of pixels in each spherical image. The table 2 groups all the results obtained with each database. To evaluate the registration results, we have calculated the score of successful localization (LS) and the mean localization error MLE. The localization is considered successful if the localization error is less than 1 meter. By applying this threshold, we have calculated the score of successful localization.

4.4.2. Localization Accuracy

The fig. 7 summarizes the results obtained in the first sequence with and without semantic filtering. It can be seen that semantic filtering improves localization accuracy. Indeed, the fig. 7 shows the localization error with semantic filtering in red and in blue, the localization error without semantic filtering. We find that most of the time the red curve is below the blue one, which indicates a higher localization accuracy. We have noticed that at 65% the semantic filtering allows to improve the results. Otherwise the semantic filtering does not improve the results. This can be explained by the noise in the mesh semantization. On the other hand, semantic filtering reduces the time required for registration. The average reduction is about 20% of computing time. To go further we have decided to test the registration on the same path of length 150 m but with a non-optimized graph. Indeed, the spheres of this graph are placed in a systematic way every $3m$ without taking into account the different characteristics of the scene. This test is performed

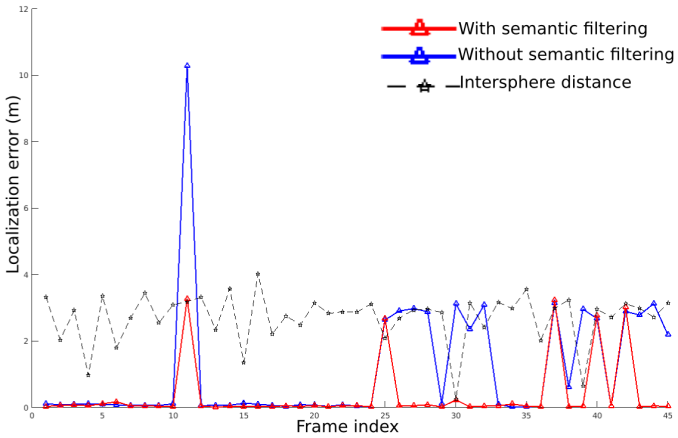


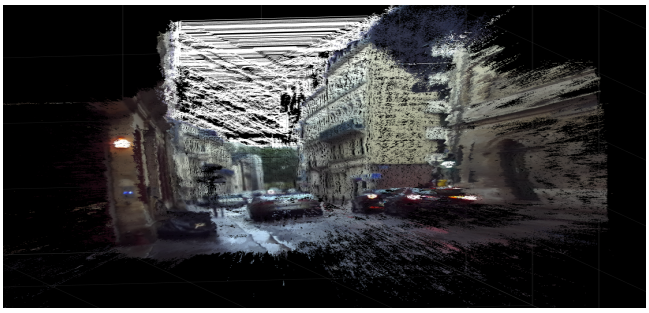
Figure 7: Localization error obtained with (f) or without semantic filtering (wf)

to evaluate the usefulness of the optimal graph. The optimal graph has only 22 spherical images while the non-optimal graph has 45 images. The compression ratio obtained with the optimal graph is 97.75% compared to 95.64% with the non-optimal graph. Although the number of spheres as well as the coverage between the spheres decreases, the agent manages to follow the path. The successful localization score in the optimal graph decreased by only 0.8%. In the second sequence we have used a set of stereo pairs of images. The results obtained with this sequence are less accurate. The localization error obtained varies between 4 and 7 meters. Despite the use of a low cost sensor, the calibration and synchronization errors, we still achieve acceptable results. We have decided to acquire a new real sequence with a more efficient system. We have used a ZED camera with a higher fps: 30 to reduce blur on the acquired images. The fig. 8 shows an example of an agent image and the construction of the corresponding 3D cloud.

We chose a 150m trajectory with a right turn. The acquisition of the agent images was done in two ways. The first sequence is dedicated to pedestrians and a second one for vehicles. For the pedestrian trajectory, we have acquired images on the two sidewalks of the street. To test the registration we have produced two graphs of spheres. A first graph is dedicated to vehicles and the second one to pedestrians.

For the car sequences, the registration results obtained on the images acquired with the ZED camera are better than on the fisheye images. The table 2 shows that the rate of successful localization with the ZED cam-

(a)



(b)

Figure 8: (a) Rectification of a right agent image and a left agent image. (b) Construction of a 3D agent cloud from the left-right images.

era (car) is improved compared to the other sequence acquired with fisheye camera. The acquisition system used in this experiment is more robust and stable. This may explain the improvement in localization accuracy. However, the evaluation of results remains difficult. Indeed, to evaluate the results obtained, we have used a GPS which is not very accurate in narrow streets. But we know that in the pedestrian trajectory the set of images are necessarily on the sidewalks or the pedestrian crossings. To evaluate the results we have defined a margin zone. This zone represents the sidewalk over the entire trajectory. The width of this area is about 1.2 meter. By respecting this hypothesis, all positions belonging to this area are considered well localized. The figure 9 shows the images that have been localized in the pedestrian zone (in green) and the images that have not been well localized. These errors can be caused by the choice of the nearest sphere. The results of localization in the sphere graph dedicated to vehicles seems to be better than the pedestrian trajectory. We can explain this by the fact that in the pedestrians sequence the phenomenon of masking by cars is very frequent. Indeed the images were acquired at an average height of 1.5 meter but despite this, a large part of the scene is hidden by the cars parked on the two sides of the street. The fig-

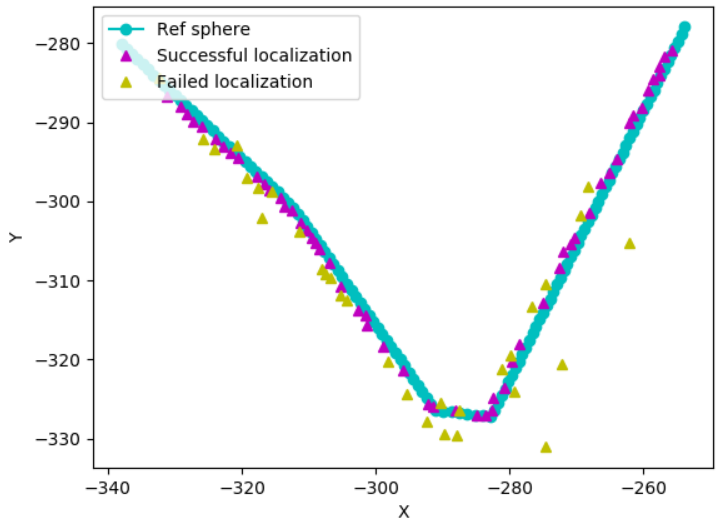


Figure 9: Registration results obtained with the agent sequence on the pedestrian trajectory: the reference sphere (blue), the pedestrian zone (black), successful registration (green), failed registration (red).

ure 10 shows an example of obtained results where we can see that the agent can locate himself. In some particular cases, we find that the localization error increases considerably. For example, as shown in the figure 10, the 90 degree turn is not recognizable. In this area we have buildings with glazed facades. The algorithm does not converge to the right solution. Indeed, when mapping the scene, the LiDAR does not capture the entire surface area.

4.4.3. Optimal Intersphere Distance

To study the distance criterion separating the spheres in the graph we have tested the localization with different values of this distance. Indeed, in our sphere positioning algorithm we have added this distance criterion in order to guarantee the efficiency of the graph in navigation. To test the effect of this distance on localization, we have created 4 sphere graphs with the intersphere distance criterion. This distance is 1, 2, 3 and 4 meters. We have calculated the recall and precision for each test as well at each time the true positive rate and the false positive rate. For example, an image that belongs to the first sidewalk and has been located in this sidewalk is then counted among the true positives. While an image of the second sidewalk

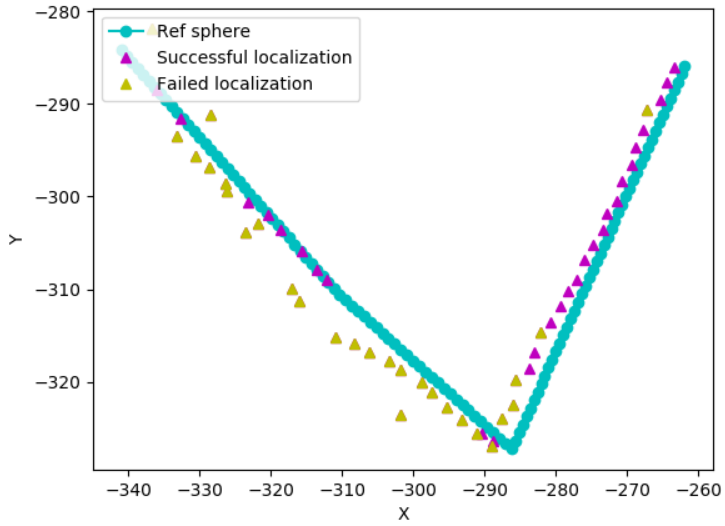


Figure 10: Registration results obtained with the agent sequence on the vehicles trajectory: the reference sphere (red), the registration results (black), an example of a sphere where the registration failed.

that has been located on the first sidewalk is counted among the false positive . We obtain the results shown on fig. 11.

From this figure we can say that a distance of 1 meter between the spheres gives the best performance. However, this implies an increase in the number of spheres in the final graph. In order to achieve a compromise between compression ratio and localization accuracy, a distance of about 3m is added as an additional criterion in the spheres positioning process. Another alternative solution is the possibility of using several reference spheres during registration.

4.4.4. Use of Several Reference Spheres

During the localization phase, the agent continuously relocates himself, considering a selected reference sphere. However, as the absolute distance between the reference sphere and the current sphere increases, the ability to locate itself decreases. kept for a period of time until The agent then requests a new reference sphere from the server, that is supposed to be closest to its current position. But, no guarantee can be given on the accuracy of the localization. As a result, the estimation of the location becomes less and

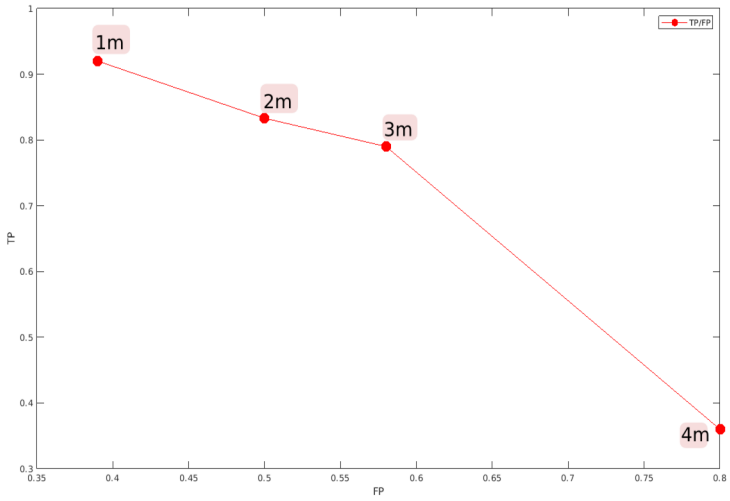


Figure 11: Evolution of true positive (TP) and false positive (FP) by increasing the distance between the spheres.

less accurate. These errors, combined with the uncertainty of selecting the nearest sphere, can generate discontinuities in the estimated trajectories.

In order to obtain the smoothest possible trajectories, it is desirable to use several reference spheres simultaneously. We note the number of reference spheres used in the registration by N. The current agent image is aligned to a window with the N nearest spheres in the graph as shown in fig 12. This figure shows the real time tracking of the agent position. The desired trajectory is modelled by the path $\{S_0 - S_4\}$, and the agent trajectory is $\{a_0 - a_7\}$. We show here the agent at location a_2, it is registering itself with the two spheres S_2 and S_3 ($N = 2$).

In this case, the localization problem consists in minimizing the distance between the agent's point cloud and the N spheres. The initial position of the agent is initialized using a randomly selected value, within a circle that includes the N spheres. To test this method, we chose a trajectory of 111 meters. The registration is done with a window containing the N closest spheres to the agent image. We have tested two values of N: 3 and 5.

This figure shows a comparison of the localization error using 2 and 5 reference spheres. The results obtained with a larger window (N=5) allow to slightly improve the localization results (fig 13). However, in some cases, the

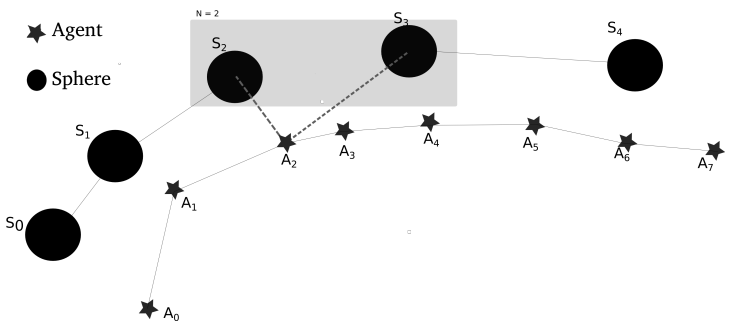


Figure 12: Registration with N reference spheres.

use of 5 reference spheres gives less precise results. Indeed the semantic filtering can lead to compare two different images but with very similar semantic information. In the case of N=5, the computation time is on average equal to 50(s) against 40(s) in the case of N=3. In addition, to reach a compromise between localization accuracy and computation time, it is preferable to use N=3 spheres.

5. Conclusions and Perspectives

In the literature, there are no methods used to effectively summarize a large-scale 3D map by taking into account all the characteristics of the scene. In this work, we present a new method of map summarization in the form of a navigability graph. A complete spherical representation of the environment containing several types of information has been proposed. This method consists in building a graph of spherical images augmented by depth and semantic information. Each node in the graph contains an augmented RGB-D-L sphere available in a sufficient resolution for navigation. The nodes of the graph are connected by a precise 3D pose in a global reference frame. The edges of the graph are weighted by the distances between the spheres. The dense data contained in the spheres allows the use of direct visual localization techniques. To build this model, two new criteria were defined and used to precisely and optimally place the nodes of the graph in the scene. The proposed method makes it possible to automatically model and summarize, with a dense and rich representation, large urban environments. It allows to summarize a large map while maintaining a maximum of useful information for navigation. We present the first method that allows to posi-

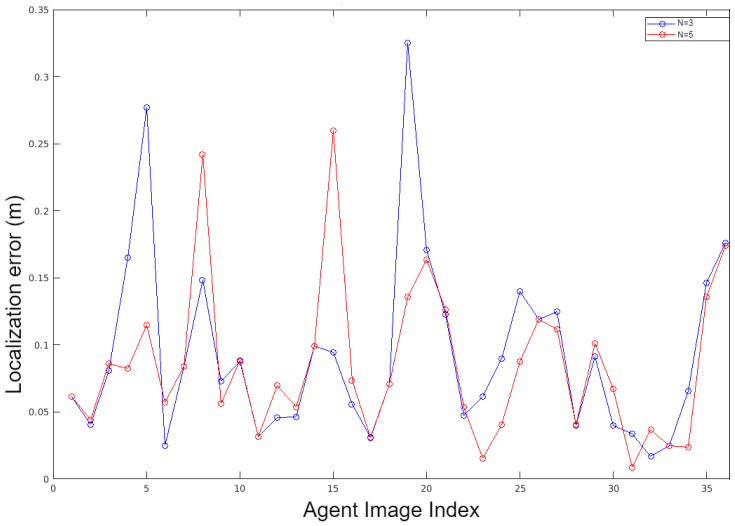


Figure 13: Registration with N reference spheres. Localization error with N=3 (blue curve) and with N=5 (red curve).

tion the spheres in an optimal way by exploiting all the scene characteristics. The sphere graph obtained during this summary phase is used to localize in real time an agent navigating in the vicinity of the graph. To accomplish this task, the Go-ICP algorithm was adapted and used to perform the 3D registration. The use of a reduced sphere graph simplifies these navigation tasks. In fact, the time required to carry out the registration is reduced by using a small number of pixels, while maintaining full observability of the environment. The advantage of this approach is that the pixel selection is done offline when building the navigability graph. Throughout online localization, only the nearest spheres are used, which allows a real-time localization. To improve the robustness of the online visual localization method, a semantic filtering approach has been proposed. The summary and localization methods have been validated on a number of experiments, which points out several ways of improvement. It would be interesting to add other optimization criteria in the positioning of the spheres. Indeed, we can add a overlapping criterion between the spheres to improve the accuracy of the localization. It would also be possible to add other types of edge information to the graph to facilitate the map updates. To improve the robustness of the online localization method, it would be interesting to consider creating additional online RGB-D-L spheres. Indeed, when the agent is unable to locate himself in

relation to the reference sphere, we can create supplementary spheres online. To do this, it is possible to further improve the construction time of a sphere RGB-D-L. For example, we can run the ray tracing algorithm on the GPU. It would be interesting to propose several types of navigability graphs depending on the type of navigation system. We can propose a suitable path to be followed by the agent according to his mobility capacities: car, pedestrian, bicycle or mobile robot. This requires the improvement of the semantization of the initial 3D mesh. Indeed, it will be necessary to add other semantic classes such as pedestrian crossings. To evaluate the navigability graph, it would be interesting to integrate it into a stand-alone navigation application. We can test it with a vehicle equipped with several types of cameras in real conditions in a urban environment.

6. Acknowledgments

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

References

[1] H. G. Seif, X. Hu, Autonomous driving in the icity - hd maps as a key challenge of the automotive industry, Engineering 2 (2) (2016) 159–162.

[2] M. Boussaha, E. Fernandez-Moral, B. Vallet, P. Rives, On the production of semantic and textured 3d meshes of large scale urban environments from mobile mapping images and lidar scans, Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), Marne-la-Vallée, France (2018).

[3] A. Elfes, A tesselated probabilistic representation for spatial robot perception and navigation, Proc.NASA Conference on Space Telerobotics (1989).

[4] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with rao-blackwellized particle filters, IEEE transactions on Robotics 23 (1) (2007) 34–46.

[5] D. Hähnel, W. Burgard, S. Thrun, Learning compact 3d models of indoor and outdoor environments with a mobile robot, Robotics and Autonomous Systems 44 (1) (2003) 15–27.

[6] S. Thrun, W. Burgard, D. Fox, A probabilistic approach to concurrent mapping and localization for mobile robots, Autonomous Robots 5 (3-4) (1998) 253–271.

[7] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, Monoslam: Real-time single camera slam, IEEE transactions on pattern analysis and machine intelligence 29 (6) (2007) 1052–1067.

[8] S. Thrun, W. Burgard, D. Fox, A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), Vol. 1, IEEE, 2000, pp. 321–328.

[9] D. M. Cole, P. M. Newman, Using laser range data for 3d slam in outdoor environments, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, 2006, pp. 1556–1563.

[10] A. Chapoulie, P. Rives, D. Filliat, Topological segmentation of indoors/outdoors sequences of spherical views, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (2012) 4288–4295.

[11] A. Chapoulie, P. Rives, D. Filliat, Appearance-based segmentation of indoors/outdoors sequences of spherical views, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (2013) 1946–1951.

[12] R. Drouilly, P. Rives, B. Morisset, Fast hybrid relocation in large scale metric-topologic-semantic map, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (2014) 1839–1845.

[13] J. A. Fernandez, J. Gonzalez, A general world representation for mobile robot operations, Seventh conference of the Spanish association for artificial intelligence (CAEPIA-97) (1997).

[14] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, H. Choset, Hierarchical simultaneous localization and mapping, in: Proceedings 2003

IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, IEEE, 2003, pp. 448–453.

[15] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, P. Furgale, The gist of maps-summarizing experience for lifelong localization, IEEE International Conference on Robotics and Automation (ICRA) (2015) 2767–2773.

[16] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, P. Furgale, Summary maps for lifelong visual localization, Journal of Field Robotics 33 (5) (2016) 561–590.

[17] H. Soo Park, Y. Wang, E. Nurvitadhi, J. C. Hoe, Y. Sheikh, M. Chen, 3d point cloud reduction using mixed-integer quadratic programming, IEEE Conference on Computer Vision and Pattern Recognition Workshops (2013) 229–236.

[18] T. J. Steiner, G. Huang, J. J. Leonard, Location utility-based map reduction, IEEE International Conference on Robotics and Automation (ICRA) (2015) 479–486.

[19] F. Fraundorfer, C. Engels, D. Nistér, Topological mapping, localization and navigation using image collections, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007. (2007) 3872–3877.

[20] F. Li, J. Kosecka, Probabilistic location recognition using reduced feature set, IEEE International Conference on Robotics and Automation, ICRA (2006) 3405–3410.

[21] E. Menegatti, T. Maeda, H. Ishiguro, Image-based memory for robot navigation using properties of omnidirectional images, Robotics and Autonomous Systems 47 (4) (2004) 251–267.

[22] M. Jogan, A. Leonardis, Robust localization using panoramic view-based recognition, 15th International Conference on Pattern Recognition 4 (2000) 136–139.

[23] A. Remazeilles, F. Chaumette, P. Gros, Robot motion control from a visual memory, IEEE International Conference on Robotics and Automation, ICRA'04 5 (2004) 4695–4700.

[24] D. Cobzas, H. Zhang, M. Jagersand, Image-based localization with depth-enhanced image map, IEEE International Conference on Robotics and Automation, ICRA'03. 2 (2003) 1570–1575.

[25] S. Benhimane, A. Ladikos, V. Lepetit, N. Navab, Linear and quadratic subsets for template-based tracking, 2007 IEEE Conference on Computer Vision and Pattern Recognition (2007) 1–6.

[26] M. Meilland, A. I. Comport, P. Rives, Dense omnidirectional rgb-d mapping of large-scale outdoor environments for real-time localization and autonomous navigation, Journal of Field Robotics 32 (4) (2015) 474–503.

[27] M. Meilland, A. I. Comport, P. Rives, A spherical robot-centered representation for urban navigation, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010) 5196–5201.

[28] G. Gallegos, M. Meilland, P. Rives, A. I. Comport, Appearance-based slam relying on a hybrid laser/omnidirectional sensor, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2010) 3005–3010.

[29] R. Ju, L. Ge, W. Geng, T. Ren, G. Wu, Depth saliency based on anisotropic center-surround difference, IEEE International Conference on Image Processing (ICIP) (2014) 1115–1119.

[30] H. Peng, B. Li, W. Xiong, W. Hu, R. Ji, Rgbd salient object detection: a benchmark and algorithms, European Conference on Computer Vision (2014) 92–109.

[31] J. Wang, M. P. Da Silva, P. Le Callet, V. Ricordel, Computational model of stereoscopic 3d visual saliency, IEEE Transactions on Image Processing 22 (6) (2013) 2151–2165.

[32] C. H. Lee, A. Varshney, D. W. Jacobs, Mesh saliency, ACM transactions on graphics (TOG) 24 (3) (2005) 659–666.

[33] G. Leifman, E. Shtrom, A. Tal, Surface regions of interest for viewpoint selection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 414–421.

[34] R. Song, Y. Liu, R. R. Martin, P. L. Rosin, Mesh saliency via spectral processing, ACM Transactions on Graphics (TOG) 33 (1) (2014) 6.

[35] J. Leroy, N. Riche, M. Mancas, B. Gosselin, 3d saliency based on supervoxels rarity in point clouds, Hamburg, Germany (2015).

[36] E. Shtrom, G. Leifman, A. Tal, Saliency detection in large point sets, IEEE International Conference on Computer Vision (2013) 3591–3598.

[37] J.-S. Yun, J.-Y. Sim, Supervoxel-based saliency detection for large-scale colored 3d point clouds, IEEE International Conference on Image Processing (ICIP) (2016) 4062–4066.

[38] C. H. Lee, A. Varshney, D. W. Jacobs, Mesh saliency, ACM transactions on graphics (TOG) 24 (3) (2005) 659–666.

[39] R. Gal, D. Cohen-Or, Salient geometric features for partial shape matching and similarity, ACM Transactions on Graphics (TOG) 25 (1) (2006) 130–150.

[40] P. Shilane, T. Funkhouser, Distinctive regions of 3d surfaces, ACM Transactions on Graphics (TOG) 26 (2) (2007) 7.

[41] X. Chen, A. Saparov, B. Pang, T. Funkhouser, Schelling points on 3d surface meshes, ACM Transactions on Graphics (TOG) 31 (4) (2012) 29.

[42] G. Leifman, E. Shtrom, A. Tal, Surface regions of interest for viewpoint selection, 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012) 414–421.

[43] R. Song, Y. Liu, R. R. Martin, P. L. Rosin, 3d point of interest detection via spectral irregularity diffusion, The Visual Computer 29 (6-8) (2013) 695–705.

[44] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (fpfh) for 3d registration, IEEE International Conference on Robotics and Automation, 2009. ICRA'09 (2009) 3212–3217.

[45] S. M. Smith, J. M. Brady, Susan - a new approach to low level image processing, International journal of computer vision 23 (1) (1997) 45–78.

[46] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, European conference on computer vision (2006) 430–443.

[47] D. G. Lowe, Local feature view clustering for 3d object recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 1 (2001) I–682.

[48] A. Godil, A. I. Wagan, Salient local 3d features for 3d shape retrieval, Three-Dimensional Imaging, Interaction, and Measurement 7864 (2011) 78640S.

[49] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, European conference on computer vision (2006) 404–417.

[50] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image and vision computing 22 (10) (2004) 761–767.

[51] M. Agrawal, K. Konolige, M. R. Blas, Censure: Center surround extremas for realtime feature detection and matching, European Conference on Computer Vision (2008) 102–115.

[52] E. Mair, G. D. Hager, D. Burschka, M. Suppa, G. Hirzinger, Adaptive and generic corner detection based on the accelerated segment test, European conference on Computer vision (2010) 183–196.

[53] C. Harris, M. Stephens, A combined corner and edge detector., Alvey vision conference 15 (1988) 50.

[54] D. Plemenos, M. Benayada, Intelligent display in scene modeling. new techniques to automatically compute good views, International Conference GraphiCon 96 (1996) 1–5.

[55] P.-P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, Viewpoint selection using viewpoint entropy., VMV 1 (2001) 273–280.

[56] D. Sokolov, D. Plemenos, K. Tamine, Viewpoint quality and global scene exploration strategies., GRAPP 2006 (2006) 184–191.

[57] I. B. Salah, S. Kramm, C. Demonceaux, P. Vasseur, Summarizing large scale 3d point cloud for navigation tasks, 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (2017) 1–8.

[58] I. Ben Salah, P. Vasseur, C. Demonceaux, S. Kramm, Summarizing large scale 3d point cloud for navigation tasks, Intelligent Transportation Systems (ITSC) (2017).

[59] I. B. Salah, S. Kramm, C. Demonceaux, P. Vasseur, Summarizing large scale 3d mesh, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018) 1–9.

[60] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Transactions on information theory 29 (4) (1983) 551–559.

[61] D. Lee, A. Lin, Computational complexity of art gallery problems, IEEE Transactions on Information Theory 32 (2) (1986) 276–282.

[62] S. Carlsson, H. Jonsson, B. J. Nilsson, Finding the shortest watchman route in a simple polygon, Discrete & Computational Geometry 22 (3) (1999) 377–402.

[63] N. Paparoditis, J.-P. Papelard, B. Cannelle, A. Devaux, B. Soheilian, N. David, E. Houzay, Stereopolis II: A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology, in: Revue Francaise de Photogrammétrie et de Télédétection, 2012, pp. 69–79.

[64] W. Zhen, S. Zeng, S. Soberer, Robust localization and localizability estimation with a rotating laser scanner, IEEE International Conference on Robotics and Automation (ICRA) (2017) 6240–6245.

[65] J. Yang, H. Li, D. Campbell, Y. Jia, Go-icp: a globally optimal solution to 3d icp point-set registration, arXiv preprint arXiv:1605.03344 (2016).

[66] P. J. Besl, N. D. McKay, Method for registration of 3-d shapes, Sensor Fusion IV: Control Paradigms and Data Structures 1611 (1992) 586–607.