(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 October 2020 (01.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/194079 A1

(51) International Patent Classification:
G06K 9/00 (2006.01)

(21) International Application Number:
PCT/IB2020/051779

(22) International Filing Date:
03 March 2020 (03.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
00397/19 27 March 2019 (27.03.2019) CH

(71) Applicants: **SEVENSENSE ROBOTICS AG** [CH/CH]; c/o ETHZ Zürich, Weinbergstrasse 35 – WEH F13, 8092 Zürich ETH-Zentrum (CH). **ETH ZURICH** [CH/CH]; Rämistrasse 101 / ETH transfer, 8092 Zürich (CH).

(72) Inventors: **RATZ, Sebastian**; Badenerstrasse 365, 8003 Zurich (CH). **DUBÉ, Renaud**; Binzmuehlestrasse 78, 8050 Zurich (CH). **DYMCZYK, Marcin**; Loorenstrasse 74, 8053 Zurich (CH). **SOMMER, Hannes**; Rotbuchstrasse 66, 8037 Zurich (CH).

(74) Agent: **P&TS SA**; Av. J.-J. Rousseau 4, P.O. Box 2848, 2001 Neuchâtel (CH).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: METHOD AND SYSTEM FOR PERFORMING LOCALIZATION OF AN OBJECT IN A 3D

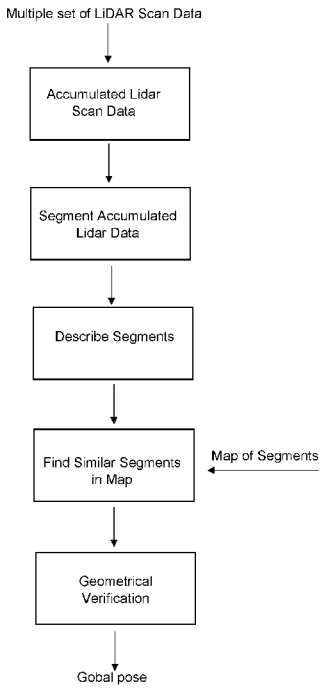

Fig. 1 - Prior art

(57) Abstract: There is disclosed a method for performing localization of an object in 3D environment, comprising: generating a segment map in which the position and orientation of the object can be estimated, and localizing the object within said segment map, comprising the steps of: a. segmenting one point cloud or each point cloud from a plurality of point clouds into a plurality of segments pertaining to different elements or structures of the 3D environment, b. extracting a descriptor for each segment of said plurality of segments, and c. comparing the descriptors and locations of the plurality of segments of said one point cloud or each point cloud with the descriptors and locations of segments of said segment map in order to estimate the position and orientation of the object. Said one point cloud is generated from a single 3D scan of one 3D scanner, or said each point cloud is generated from a single 3D scan of respective 3D scanner of a plurality of 3D scanners, wherein the plurality of single 3D scans is collected during the same period of time.

WO 2020/194079 A1

[Continued on next page]

Published:

— *with international search report (Art. 21(3))*

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

Method and system for performing localization of an object in a 3D environment

Field of the invention

[0001] The present invention concerns a method for performing localization of an object in a 3D environment. The invention also concerns a system comprising one 3D scanner for generating a point cloud from a single 3D scan of the 3D scanner or a plurality of 3D scanners for generating a plurality of point clouds from a single 3D scan of respective plurality of 3D scanners acquired at the same time, and a computing device configured to execute an algorithm to perform the method.

Description of related art

[0002] For autonomous mobile robots, it is crucial that at all times during operation, they have an accurate estimate of where they are with respect to their surroundings. This is a prerequisite to perform tasks such as navigation, obstacle avoidance or in order to interact with the environment. In robotics, localization can happen on two levels: locally and globally.

[0003] Local motion estimation usually starts off with a known initial position and rotation (called pose) of the robot. After a movement of the robot, the estimate of the current pose is then updated by taking advantage of sensor measurements. This can for example be done by counting the revolutions of the robot's wheels or by measuring the change of perspective in images taken at the initial position and at the new position. Local motion estimation thus deals with tracking the pose of a robot given an initial estimate. In many modern systems, besides tracking the pose, also a 3D map of the environment is built incrementally, which helps the robot to orient itself in larger spaces, especially relevant for tasks like navigation. An example here would be a delivery robot that needs to find its way from the vendor to the doorstep of your home.

[0004] As aforementioned, an initial pose estimate is required for initializing the local motion estimation process. This can be obtained by a type of technique called global localization. Given a 3D map, which has previously been built using a local motion estimation algorithm, and one set of current sensor measurements (e.g. images, LiDAR scans, etc), global localization deals with identifying where the robot currently is in the map - without having any prior information about the robots whereabouts. This is especially challenging when using GPS is not an option. Global localization also helps to reorient a robot which has got lost due to some erroneous computations. Even in modern local localization systems this may still happen regularly. Finally, global localization helps map building algorithms to create more accurate 3D maps through a technique called loop closure detection.

[0005] Global localization in LiDAR data is challenging and different approaches to solving the problem exist. A flowchart of a segment-based global localization algorithm according to the prior art is illustrated in Figure 1. This algorithm takes as input a sequence of scans of a 3D LiDAR scanner and first integrates them in one larger and dense point cloud. A sparse LiDAR measurement and a dense point cloud according to the prior art are respectively illustrated in Figure 2.

[0006] In a second step, the dense point clouds are segmented into clusters, called segments. Usually a segment represents a set of points which are close in space and belong to one or multiple objects or structures in the environment. Some typical example segments obtained by methods of the prior art can be found in Figure 3. The image shows two segments per column, resulting from scanning the same objects during two different visits of the place. The segments depicted in Figure 3 stem from walls, trees and vehicles.

[0007] In a next step, these segments are then described statistically by extracting geometric properties either using hand-crafted methods or using

artificial intelligence. The result is called a descriptor and is a compact list of floating-point values. The global map then consists of a collection of multiple segment descriptors together with their corresponding locations in the map.

[0008] If one then wants to globally localize a new LiDAR scan in such a map, one again extracts segments from a query point cloud and describes them with the same procedure as for map construction. In a next step, the descriptors of the query segments are compared with those in the map. If multiple segments with similar descriptors are found, these are called matches. It is then verified, whether the matched segments in the map have a similar geometric configuration as the query segments. If this is true, a global localization is detected and the pose of the robot with respect to the map can be computed from the matched segment configuration.

[0009] All segment-based global localization algorithms according to the prior art include and rely on a specific step: they all use and require LiDAR data from multiple successive scans, usually between 10 and 50 scans. The data are integrated in order to build a dense 3D point cloud of the local environment, which then results in dense segments. This is a disadvantage when operating autonomous robots, as these would need to navigate through an unknown environment for several meters in order to integrate the dense point cloud representation necessary for the localization algorithm to work. This exploration task can be time consuming and is potentially dangerous if performed autonomously.

[0010] It is an aim of the present invention to obviate or mitigate at least some of the above-mentioned disadvantages.

[0011] More particularly, the aim of the present invention is to provide a method suitable for performing localization in a 3D environment not only of a moving object but also a static object without having to move a scanner

such as a LiDAR sensor, whereby a map given data from the scanner and local pose estimates from an arbitrary local motion estimation algorithm can be build, and whereby the pose of the scanner within the previously computed map can be determined.

Brief summary of the invention

[0012] According to the invention, these aims are achieved by means of a method for performing localization of an object in 3D environment, comprising: generating a segment map in which the position and orientation of the object can be estimated, and localizing the object within said segment map, comprising the steps of: a. segmenting one point cloud or each point cloud from a plurality of point clouds into a plurality of segments pertaining to different elements or structures of the 3D environment, b. extracting a descriptor for each segment of said plurality of segments, and c. comparing the descriptors and locations of the plurality of segments of said one point cloud or each point cloud with the descriptors and locations of segments of said segment map in order to estimate the position and orientation of the object. Said one point cloud is generated from a single 3D scan of one 3D scanner, or said each point cloud is generated from a single 3D scan of respective 3D scanner of a plurality of 3D scanners, wherein the plurality of single 3D scans is collected during the same period of time.

[0013] In an embodiment, generating the segment map comprises:

i) segmenting a plurality of point clouds, wherein each point cloud is generated from a single 3D scan of respective plurality of 3D scanners located at different places of the environment, and wherein each point cloud is segmented into segments separately from the other point clouds,

ii) extracting descriptor for each of the segments resulting from the segmentation of each of said plurality of point clouds, and

iii) associating each of said segments with a portion of the 3D environment.

[0014] In an embodiment, the point cloud segmented under step a. is a sparse point cloud.

[0015] In an embodiment, segments within the segment map which have similar descriptors and which are in a similar location are marked as duplicates.

[0016] In an embodiment, segments marked as duplicate are removed from the segment map and just one of them is kept.

[0017] In an embodiment, the segments marked as duplicate are merged into one new descriptor and one new segment location.

[0018] In an embodiment, the step of segmentation of the point cloud under step a. is achieved as a function of local geometrical properties in said point cloud.

[0019] In an embodiment, before step a. said point cloud is projected into a range image. Segmentation of the point cloud under step a. is performed on the range image.

[0020] In an embodiment, the segmentation is based on depth differences between neighbouring points in the range image.

[0021] In an embodiment, a neural network is used for extracting descriptor for each segment of said plurality of segments under step b.

[0022] In an embodiment, segments are fed into the neural network aligned with the direction of gravity.

[0023] In an embodiment, the neural network is trained with a triplet loss with batch hard negative mining.

[0024] In an embodiment, the neural network is trained with images of elements or structures of the environment together with segments. Each of said images is fed to the neural network together with the segment derived from the element or structure of the corresponding image.

[0025] In an embodiment, the neural network comprises a first, a second and a third sub-network. The first sub-network is configured to extract an intermediary descriptor of the segment. The second sub-network is configured to extract an intermediary descriptor of the image. The third sub-network is configured to fuse the two aforementioned intermediary descriptors into one fused descriptor.

[0026] In an embodiment, the estimation of the position and orientation of the object under said step c. is obtained by computing the similarity between two given descriptors. The similarity is preferably computed with a Minkowski distance between the two given descriptors, or with a metric derived from said distance.

[0027] In an embodiment, descriptor and location of segments, obtained from segmentation of one or more point clouds which had been generated from a single scan of one or multiple 3D scanners before said point cloud under step a. was generated, are used together with descriptor and location of segments obtained from the segmentation of said point cloud under step a. when estimating the position of the object under step c.

[0028] In an embodiment, said one or multiple 3D scanners are multi-channel LiDAR scanners.

[0029] In an embodiment, the object is an autonomous or semi-autonomous robot.

[0030] Another aspect of the invention relates to a system for performing localization of an object in a 3D environment, comprising one or 3D scanners for generating a point cloud from a single scan and a plurality of point clouds from multiple scans, and a computing device configured to execute an algorithm to perform the method as described above.

[0031] Another aspect of the invention relates to a computer readable storage medium having recorded thereon a computer program. The computer program is configured to perform the method as described above.

Brief Description of the Drawings

[0032] The invention will be better understood with the aid of the description of embodiments given by way of examples and illustrated by the figures, in which:

- Figure 1 shows a flowchart of a segment-based global localization algorithm according to the prior art;
- Figure 2 shows a dense point cloud map which is used to extract dense segments according to the prior art;
- Figure 3 shows examples of dense segments obtained by methods of the prior art;
- Figure 4 shows a sparse LiDAR measurement of a sequence of scans of a 3D LiDAR scanner as used in an embodiment;
- Figure 5 shows a flowchart of a segment-based global localization algorithm according to an embodiment;
- Figure 6shows a point cloud resulting from a 64-beam Velodyne LiDAR scanner;

- Figure 7 shows a point cloud from the same place of Figure 7, but showing only 16-beams of the LiDAR scan;

- Figure 8 shows a schematic representation of segmentation algorithm configured to compute the angle β for every pair of neighbouring points in a range image according to an embodiment;

- Figure 9 shows an image of a segmented LiDAR scan from the Kitti dataset using only 16 LiDAR beams according to an embodiment;

- Figure 10 shows a schematic representation of a structure of the neural network used for descriptor extraction according to an embodiment;

- Figure 11 shows a matrix containing the obtained ground truth correspondences in the training data according to an embodiment;

- Figure 12 shows segments resulting from point cloud segmentation are projected onto camera image which has overlap with the field of view of the LiDAR sensor according to an embodiment, and

- Figure 13 shows a neural network used for fusing point cloud and image data into a combined descriptor according to an embodiment.

Detailed Description of possible embodiments of the Invention

[0033] According to an embodiment of the invention, there is provided a method for performing localization of an object, such as an autonomous or semi-autonomous robot, in 3D environment using a 3D LiDAR sensor that can perform global localization using only one single LiDAR scan measurement, i.e. one-shot localization, as shown in Figure 5.

[0034] The one-shot localization method brings major improvements over current state-of-the-art solutions which rely on the accumulation or

integration of multiple LiDAR scans taken at different times and from different positions. The resulting segments are dense. The one-shot localization method makes it possible to perform global localization without any movement of the robot or the sensor and works with the sparse segments resulting from single scans of a scanner such as a LIDAR sensor. This is a clear advantage over systems where the robot has to move or where the lidar is actuated as the moving parts can break, wear-out and increase the systems costs.

[0035] Furthermore, the geometric properties of the said dense segments highly depend on the travelled trajectory during the accumulation process. Thus, the extracted descriptors are strongly trajectory dependent which limits the robustness against changes in point of view. Additionally, the accumulation of sparse LiDAR data in dense point clouds relies on accurate motion estimation which is subject to inaccuracies resulting from inherently noisy sensor measurements. In practice, the resulting dense segments are often fuzzy, which in turn lead to descriptors with deteriorated quality. Since the one-shot localization method does not accumulate point clouds, the descriptors are not affected by erroneous motion estimates and thus result in more reliable matching. Moreover, segmenting sparse point clouds requires less computational power which supports the use of the global localization algorithm on resource constrained platforms. These characteristics directly lead to better localization performances over standard dense segment matching approaches.

[0036] The one-shot localization method according to an embodiment is shown Figure 5. The method takes as input a sparse point cloud of a single LiDAR scan. The point cloud is first segmented and the resulting segments are described using a neural network. The segment descriptors are matched against the segment map represented as a database. Candidate matches undergo a geometrical consistency test and may result in a global localization.

Segmentation

[0037] The task of segmentation is to partition a given point cloud into smaller subsets with each ideally containing non-overlapping, information-rich structures of the environment. Instead of accumulating LiDAR data from multiple scans, taken at different times and from different positions, into a local dense point cloud according to the prior art, only a single scan is used for performing localization against the map. This implies that measurements of objects and structures in an environment are almost surely only partial, since structures are measured only from one perspective. When additionally dealing with sparse LiDAR data from Velodyne-like sensors, such as the VLP-16, which has only 16 scan lines, segmentation becomes a challenging task. A comparison between a 64-beam and a 16-beam LiDAR scan can be seen in Figures 6 and 7 respectively.

[0038] Euclidean distance or curvature-based approaches as used by Dubé et al. [1] often fail when confronting such sparse data, since distances between points of two different scan lines are usually large and point normals may thus be unreliable. The specific reading pattern of the sensor would significantly affect the segmentation if not addressed explicitly.

[0039] For this reason, the segmentation approach presented by Bogoslavskyi and Stachniss [2] is leveraged. The algorithm works directly on the range image and employs a simple geometrical strategy to separate structures in the point cloud which expose strong discontinuity in depth. Ground removal is applied before segmentation for improved performance.

[0040] Let β be the angle between the line connecting the 3D points corresponding to two neighbouring pixels in the range image, and the range image normal. A schematic of the definition can be seen in Figure 8. β is computed for every pair of neighbouring pixels in the range image. Segmentation then can be casted as solving the problem of connected

components, with the separation threshold simply being a threshold on the angle β. Taking advantage of knowledge about the structure of point clouds produced by Velodyne-like sensors, all operations can be executed on the range image only. This is based on the fact that β can be computed from the depth and the known constant angle between two neighbouring pixels, which is either the angle between two LiDAR scan lines or between two LiDAR measurements within a scan line.

[0041] Ground removal is based on the assumption that the ground is a plane-like, close to horizontal structure in the lower part of the range image. Executing segmentation directly on the range image has the positive effect of considerably reducing computation times compared with methods working on 3D data, which have to deal with expensive nearest-neighbour search for thousands of points. An example of a segmented LiDAR scan can be seen in Figure 9.

Segment Description

[0042] The objective of the segment description is to perform dimensionality reduction on the incoming point cloud segments, compressing them to a vector of floating point numbers. In experiments, 64 proved to be a good size for the embedding. The compression is performed such that the L2 distance between similar segment point clouds is small, and large for segments with dissimilar structures. This principle allows taking advantage of efficient tree-based search methods to identify the nearest neighbours of segments in descriptor space.

[0043] The description is performed by an artificial neural network based on 3D convolutions, max pooling and dense layers. This architecture was first presented by Dubé et al. [3] and is similar to what is used in the algorithm disclosed herein. Prior to being fed into the neural network, point cloud segments are centred and aligned according to their principal component

normal to the direction of gravity and fit into a voxel grid. Since point clouds are processed from a single scan in contrast to data accumulated from multiple scans, the network was retrained on the sparse data. Recent advances in metric learning have been taken advantage of implementing the popular triplet loss for training the network. This is a difference to prior art by Dubé et al. [3] which relies on a class-based loss function instead.

Neural Network Training

[0044] The details regarding the training are described thereafter. A schematic representation of a structure of the neural network used for descriptor extraction is shown in Figure 10.

Training Data Generation

[0045] The training data is generated from the publicly available datasets Kitti [4] and NCLT [5]. For the generation of ground truth correspondences, the LiDAR data is first segmented with the approach described above. Two segments s_1 and s_2 are then marked as matches if their centroids are within a distance of 3m and if their 3D convex hulls fulfil the following condition:

$$\frac{Volume(Conv(s_1) \cap Conv(s_2))}{Volume(Conv(s_1) \cup Conv(s_2))} \geq p$$

The value p = 0.3 was found to produce good segment correspondences while avoiding being too strict. Pairs of segments which fulfill the above criteria, are called matches from here on. For the Kitti dataset, odometry sequence 05 and 06 are used for training and sequence 00 for testing. In the case of the NCLT dataset, the largely overlapping recordings from the dates 2012-11-17 and 2012-03-25 were used.

[0046] The work by Dubé et al. [3] uses a class-based approach for training, which fits well the domain of segments resulting from fusing multiple LiDAR scans. In the one-shot localization method disclosed therein, this approach is less appropriate. One may consider the case of a set of segments resulting from moving along a long wall. In the one-shot case, the limited range of the LiDAR sensor will produce a set of consecutive segments which likely have considerable overlap, while segments stemming from the beginning of the wall will not overlap with those from the end of the wall.

[0047] If one were to choose the class-based approach here, one would face the challenge of determining which of the segments still belong to the same class and which don't. Is has been therefore opted for storing a list of all pairs of segments which fulfils the criterium defined in equation 3.1. This information can be stored efficiently in a sparse matrix, where the columns and rows represent indices in increasing order, which also provides fast access to the matches of a given segment. An example of such a match matrix can be seen in Figure 11.

Loss Function

[0048] The triplet loss, originally developed for the computation of face image embeddings [6], is a method for metric learning which has gained considerable popularity for the application of visual place recognition [7], [8] and is implemented for the task of finding embeddings for 3D point cloud segments.

[0049] The triplet loss is defined as follows. Let $f(x)$ represent the embedding f: $R^n \rightarrow R^d$ which brings an n-dimensional input x into a d-dimensional feature space. Given an input x^a, usually called anchor, a positive sample x^p and a negative sample x^n are selected. In the present case, this would respectively be a segment matching the anchor and one not matching it. The triplet loss follows the objective of maximising the distance between

negative samples and minimizing the distance between positive samples. It is thus formulated as follows:

$$L_{triplet} = \left[||f(x^a) - f(x^p)||_2 - ||f(x^a) - f(x^n)||_2 + m \right]_+$$

[0050] The parameter *m* is the separation margin and defines the minimum difference between the positive and negative samples required for zero loss. The operator $\left[* \right]_+$ denotes the hinge loss defined as:

$$\left[x \right]_+ = max(0, x)$$

The loss being minimized for a batch of N triplets then results in:

$$L_{batch} = \sum_{i=0}^{N} \left[||f(x_i^a) - f(x_i^p)||_2 - ||f(x_i^a) - f(x_i^n)||_2 + m \right]_+$$

[0051] A crucial element in triplet training is an appropriate strategy for hard-negative mining, which has been shown to improve performance and convergence rates [6], [9], [10], [11]. Among the different approaches presented in the past, a strategy called batch hard [9] was opted for its ease of implementation and its good balance between finding challenging hard samples without becoming too sensitive to false labelling. Its functional principle is as follows:

Every batch passed through the neural networks contains *P* samples from *K* classes. In the one-shot localization disclosed herein, a class is referred to as a set of segments which are all matching with each other according to Equation 3.1. Every element in the batch has thus P – 1 positives and P ∗ (K – 1) negatives. For every element in the batch, the hardest positive and hardest negative from within the batch is selected to form a triplet. In contrast to finding hard negatives and positives in the entire dataset, this method avoids

overfitting to a few very hard or falsely labelled samples. With appropriate values for P and K, challenging samples can be easily found within the batch. In the present case, P = 4 and K = 16 proved to provide a good balance.

Segment Matching

[0052] For global localization one first needs to build a map in which the robot pose can be estimated. In the case of the presented work, building the map comes down to segmenting the scans obtained from the traversal of an environment and storing the corresponding segment descriptors and their centroids with respect to some map coordinate fame. It is assumed that the computation of these coordinates is roughly globally consistent and can be achieved through estimating the pose of the LiDAR frame throughout the data recording.

[0053] Since sequential LiDAR scans are not accumulated, many duplicates may be generated in the database if the LiDAR data contains stationary structures, i.e. when the resulting point clouds do not change significantly between scans. In order to avoid this, a filtering step is executed after describing all map segments. Duplicate entries which result from segments which are close in both descriptor and physical space are discarded. Finally, the segment descriptors are stored in a k-d tree structure to allow efficient nearest neighbour search.

[0054] Given a query LiDAR scan, the scan is segmented and the resulting segments are described as outlined above. For every segment in the scan, its k nearest neighbours in descriptor space are retrieved from the map. For each of the k nearest database descriptors d_d, the L2 distance to the query segment descriptor d_q is computed and a threshold is applied:

$$||d_q - d_d|| <= d_{max}$$

The segments passing this threshold are considered candidate matches and are passed on to the next section of the pipeline.

Geometrical Consistency Test

[0055] The implementation of the geometrical consistency test follows closely the graph formulation approach presented by Dub´e et al. [12]. Let a match between a segment in the query scan and a segment in the map be represented as a vertex in a graph. An edge then connects two vertices, if the underlying segments have their centroids at a similar distance in both the map and the query scan. Such a distance is called similar if two 3D centroids c_1 and c_2 fulfil the following:

$$\left\| c_1 - c_2 \right\|_2 \leq r$$

[0056] Where r is a threshold called resolution. Experiments showed that r = 0.4m is loose enough to tolerate instabilities of segment centroids due to partial observations and tight enough to mostly avoid false positive matches. The problem of recognizing a place then comes down to finding the maximum clique in the resulting graph and checking whether its size is greater than a threshold $n \in N \geq 3$. Experiments showed that n = 5 is a reasonable minimal clique size for many environments. Since the problem of computing the maximum clique has exponential worst-case time complexity, graph pruning is executed to keep the graph at a manageable size. For this purpose, edges are discarded if the underlying segments are at a larger distance than the maximum distance between segments in the query scan.

[0057] If a maximum clique of size larger than n is found, a place recognition is triggered and the relative pose between the centroids of the matched segments in the map and the query scan is computed. This problem is casted as finding the similarity transform between two point sets, to which Umeyama [13] offers a robust algorithm, implemented in the PCL library [14].

This final step allows us to estimate the 6 DOF pose of a robot in a map without any prior information on its pose and b using a single LiDAR scan only.

[0058] A method to enhance a point cloud segment with visual information is described thereafter. Figure 12 shows a possible integration into a global localization framework.

Image Patch Extraction

[0059] In order to extract visual information of the environment for a given segment, it is required that a camera sensor has significant overlap with the field of view of the LiDAR sensor. Once an incoming point cloud has been segmented, the resulting segments are then projected onto the camera image. This process requires the camera sensor to be extrinsically calibrated with respect to the LiDAR sensor. Following this, the bounding box of the projection is computed and the corresponding part of the image is extracted. The resulting image patch represents the visual appearance of the corresponding segment and serves to increase the performance of the segment descriptor.

LiDAR-Vision Fusion

[0060] This section describes the neural network used to extract a descriptor of a segment point cloud and its corresponding image patch. For describing the image patch, the neural network NetVLAD [15] is used. It is a neural network architecture which has shown promising results for visual global localization and produces a global descriptor of an image. It passes an incoming image through a deep convolutional network based on VGG-16 followed by a so called NetVLAD layer. The latter aggregates the features coming from the convolutional network into learned clusters and produces

a single 4096x1 descriptor of the image. An optional PCA-based dimensionality reduction may be applied.

[0061] A detailed schema of the neural network structure used for computing a multi-modal descriptor of the segment and the image is shown in Figure 13. NetVLAD is used such that the image patch is first rescaled to a size of 140x140 and then fed into the network. The resulting descriptor is concatenated with the flattened output of the segment descriptor network. In this case, the segment descriptor network is a pruned version of the one presented in Figure 10. Instead of passing the output of the 3D convolutional layers into dense layers directly, the flattened output is first concatenated with the image descriptor resulting from NetVLAD. These concatenated intermediate descriptors are then passed through dense layers which finally result in the fused, condensed multi-modal descriptor.

References

[1] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266-5272

[2] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[3] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3d segment mapping using data-driven descriptors," in *Robotics: Science and Systems (RSS)*, 2018.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] N. Carlevaris-Bianco, A. K. Ushani, and R.M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023-1030, 2015.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CoRR*, vol. abs/1503.03832, 2015.

[7] Y. Ye, T. Cieslewski, A. Loquercio, and D. Scaramuzza, "Place recognition in semi-dense maps: Geometric and learning-based approaches." 2017.

[8] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart, "Efficient descriptor learning for large scale localization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3170-3177.

[9] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *CoRR*, vol. abs/1703.07737, 2017.

[10] We. Ge, W. Huang, D. Dong, and M. R. Scott, "Deep metric learning with hierarchical triplet loss", *CoRR*, vol. abs/1810.06951, 2018.

[11] C. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," *CoRR*, vol. abs/1706.07567, 2017.

[12] R. Dubé, M.G. Gollub, H. Sommer, I. Gilitschenski, R. Siegwart, C. Cadena, and J. Nieto, "Incremental segment-based localization in 3D points clouds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1832-1839, 2018.

[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions and Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376-380, April 1991.

[14] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1-4.

[15] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," *CoRR*, vol. abs/1511.07247, 2015.

Claims

1. Method for performing localization of an object in 3D environment, comprising:

- generating a segment map in which the position and orientation of the object can be estimated,

- localizing the object within said segment map, comprising the steps of:

a. segmenting one point cloud or each point cloud from a plurality of point clouds into a plurality of segments pertaining to different elements or structures of the 3D environment,

b. extracting a descriptor for each segment of said plurality of segments, and

c. comparing the descriptors and locations of the plurality of segments of said one point cloud or each point cloud with the descriptors and locations of segments of said segment map in order to estimate the position and orientation of the object,

characterized in that

said one point cloud is generated from a single 3D scan of one 3D scanner, or

said each point cloud is generated from a single 3D scan of respective 3D scanner of a plurality of 3D scanners, wherein the plurality of single 3D scans is collected during the same period of time.

2. Method according to claim 1, wherein generating the segment map comprises:

i) segmenting a plurality of point clouds, wherein each point cloud is generated from a single 3D scan of respective plurality of 3D scanners located at different places of the environment, and wherein each point cloud is segmented into segments separately from the other point clouds,

ii) extracting descriptor for each of the segments resulting from the segmentation of each of said plurality of point clouds, and

iii) associating each of said segments with a portion of the 3D environment.

3. Method according to any preceding claim, wherein the point cloud segmented under step a. is a sparse point cloud.

4. Method according to any preceding claim, wherein segments within the segment map which have similar descriptors and which are in a similar location are marked as duplicates.

5. Method according to the preceding claim, wherein segments marked as duplicate are removed from the segment map and just one of them is kept.

6. Method according to claim 4, wherein the segments marked as duplicate are merged into one new descriptor and one new segment location.

7. Method according to any preceding claim, wherein the step of segmentation of the point cloud under step a. is achieved as a function of local geometrical properties in said point cloud.

8. Method according to any preceding claim, wherein before step a. said point cloud is projected into a range image, segmentation of the point cloud under step a. being performed on said range image.

9. Method according to the preceding claim, wherein said segmentation is based on depth differences between neighbouring points in the range image.

10. Method according to any preceding claim, wherein a neural network is used for extracting descriptor for each segment of said plurality of segments under step b.

11. Method according to the preceding claim, wherein segments are fed into the neural network aligned with the direction of gravity.

12. Method according to claim 10 or 11, wherein the neural network is trained with a triplet loss with batch hard negative mining.

13. Method according to any of claims 10 to 12, wherein the neural network is trained with images of elements or structures of the environment together with segments, each of said images being fed to the neural network together with the segment derived from the element or structure of the corresponding image.

14. The method according to any of claims 10 to 13, wherein the neural network comprises a first, a second and a third sub-network, the first sub-network being configured to extract an intermediary descriptor of the segment, the second sub-network being configured to extract an intermediary descriptor of the image, and the third sub-network being configured to fuse the two aforementioned intermediary descriptors into one fused descriptor.

15. Method according to any preceding claim, wherein the estimation of the position and orientation of the object under said step c. is obtained by computing the similarity between two given descriptors, and wherein said similarity is preferably computed with a Minkowski distance between said two given descriptors, or with a metric derived from said distance.

16. Method according to any preceding claim, wherein descriptor and location of segments, obtained from segmentation of one or more point clouds which had been generated from a single scan of one or multiple 3D scanners before said point cloud under step a. was generated, are used together with descriptor and location of segments obtained from the

segmentation of said point cloud under step a. when estimating the position of the object under step c.

17. Method according to any preceding claim, wherein said one or multiple 3D scanners are multi-channel LiDAR scanners.

18. Method according to any preceding claims, wherein the object is an autonomous or semi-autonomous robot.

19. System for performing localization of an object in a 3D environment, comprising one or 3D scanners for generating a point cloud from a single scan and a plurality of point clouds from multiple scans, and a computing device configured to execute an algorithm to perform the method according to any preceding claim.

20. A computer readable storage medium having recorded thereon a computer program, the computer program configured to perform the method according to any of claims 1 to 18.

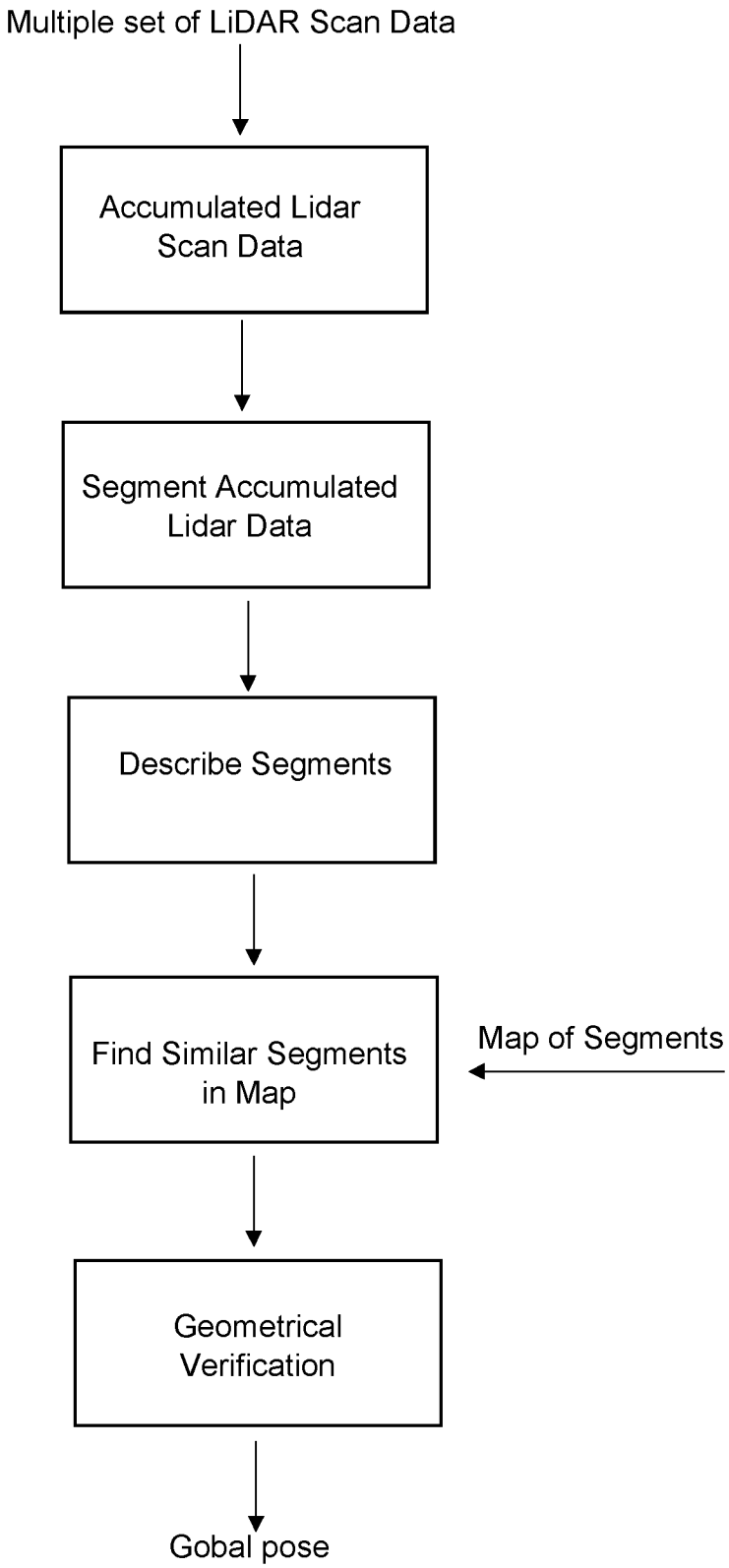

Fig. 1 - Prior art

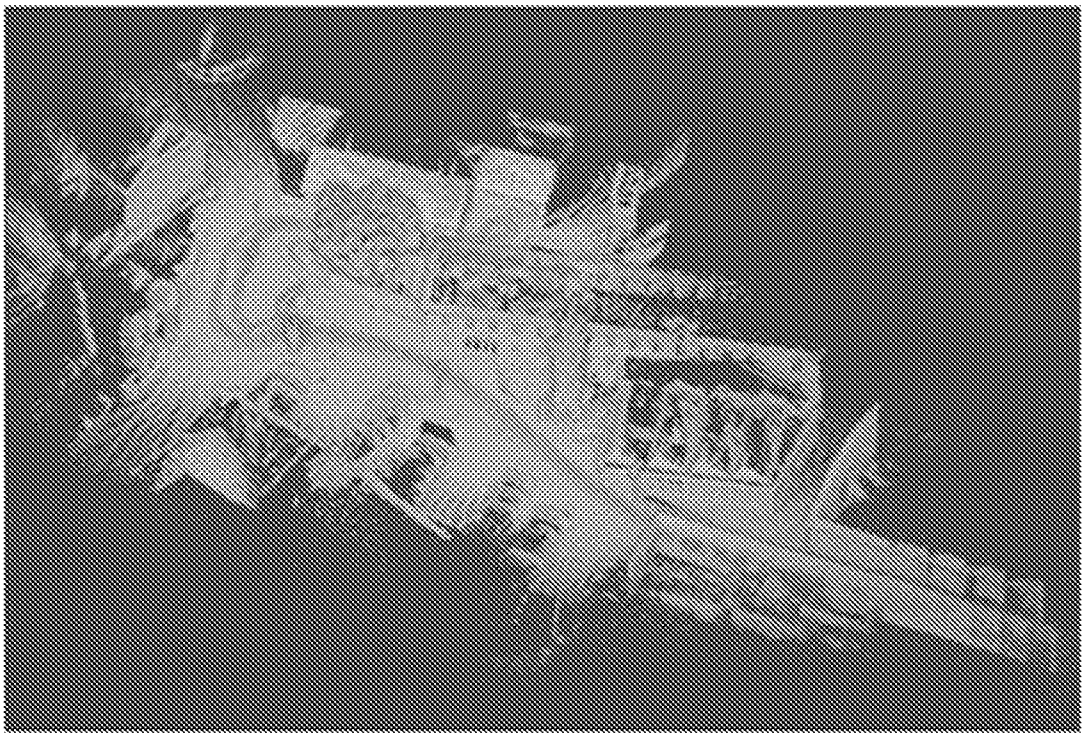

Fig. 2 - Prior art

Fig. 3 - Prior art

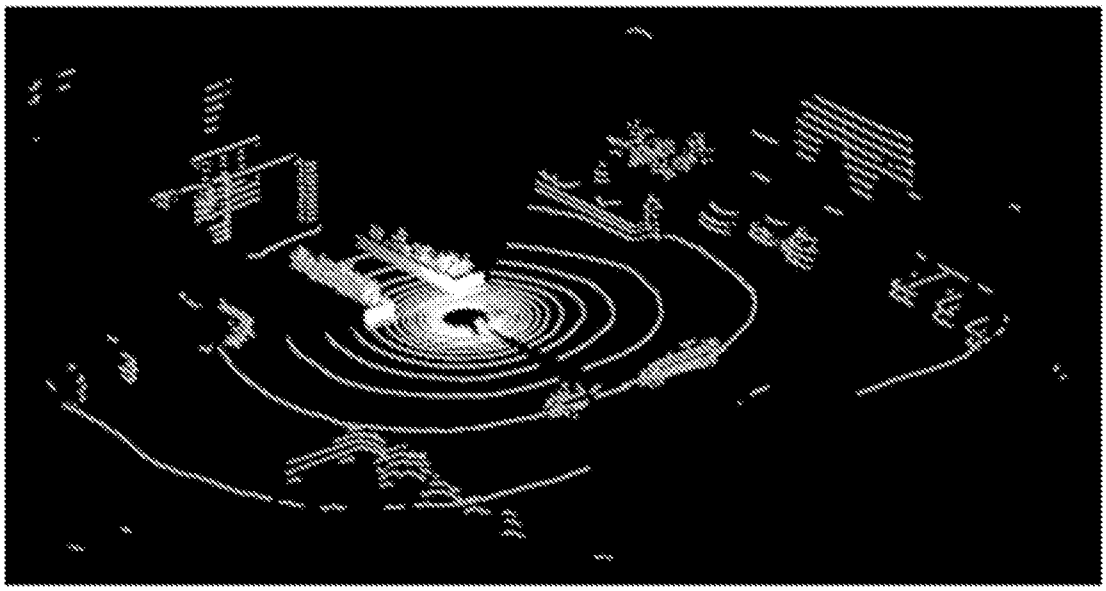

Fig. 4

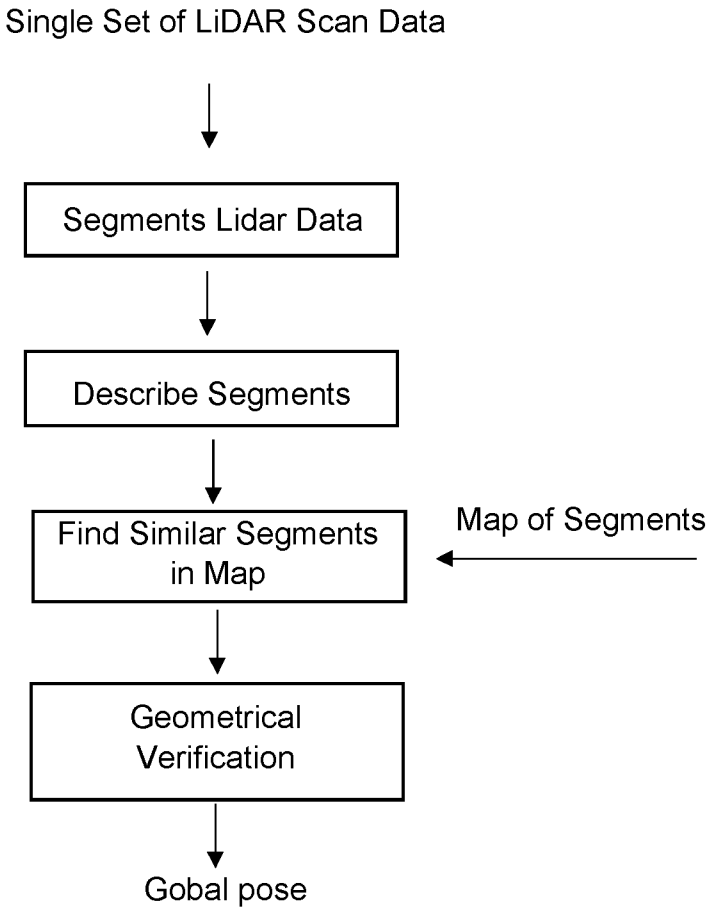
Single Set of LiDAR Scan Data
Segments Lidar Data
Describe Segments
Find Similar Segments in Map
Map of Segments
Geometrical Verification
Gobal pose

Fig.5

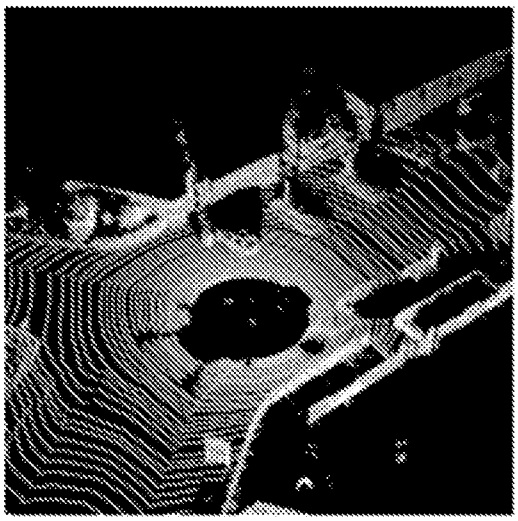

Fig. 6

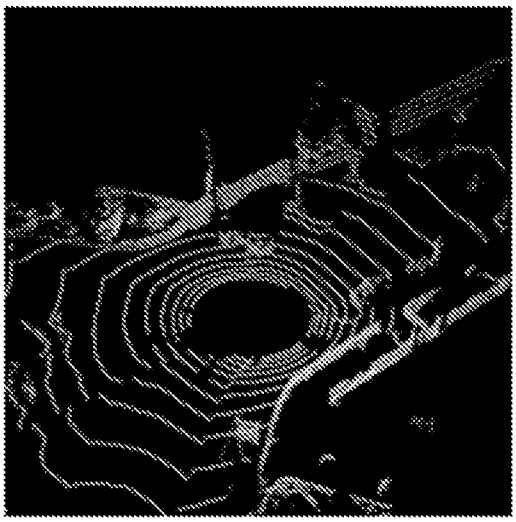

Fig. 7

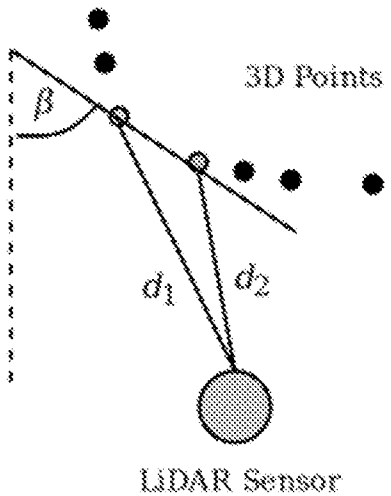

Fig. 8

Fig. 9

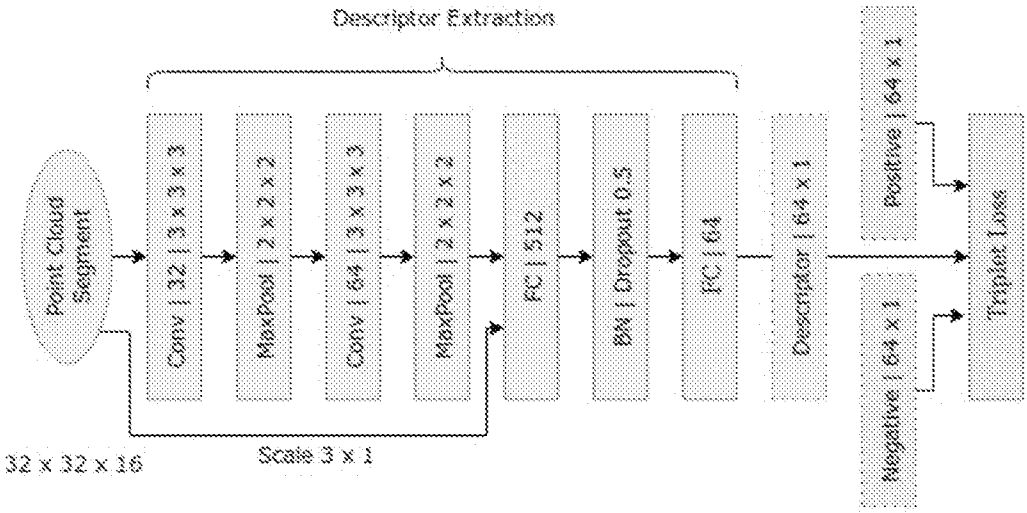
Descriptor Extraction
Point Cloud Segment
Conv | 32 | 3 x 3 x 3
MaxPool | 2 x 2 x 2
Conv | 64 | 3 x 3 x 3
MaxPool | 2 x 2 x 2
FC | 512
BN | Dropout 0.5
FC | 64
Descriptor | 64 x 1
Positive | 64 x 1
Negative | 64 x 1
Triplet Loss
32 x 32 x 16
Scale 3 x 1

Fig. 10

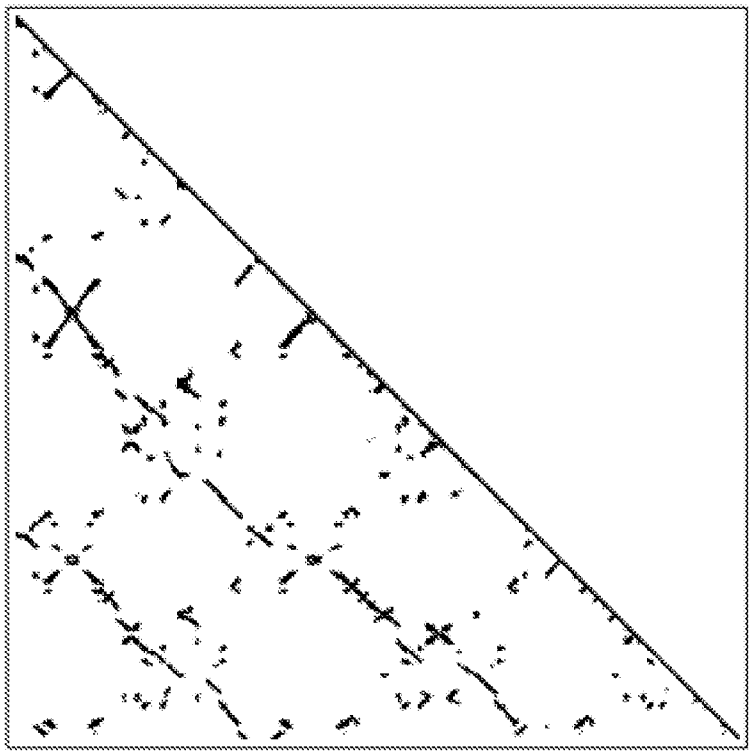

Fig. 11

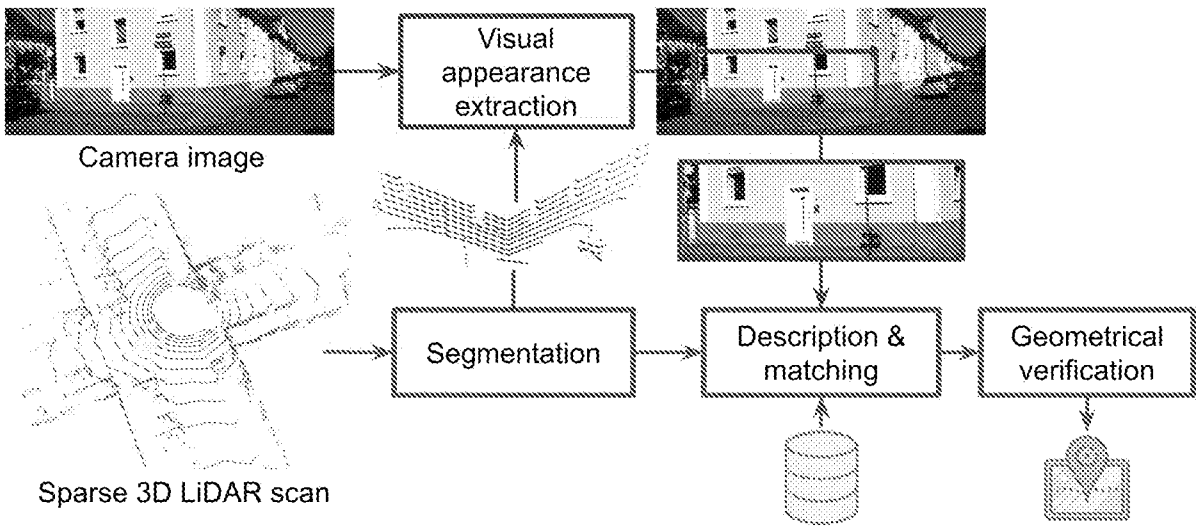
Visual appearance extraction
Camera image
Segmentation
Description & matching
Geometrical verification
Sparse 3D LiDAR scan

Fig. 12

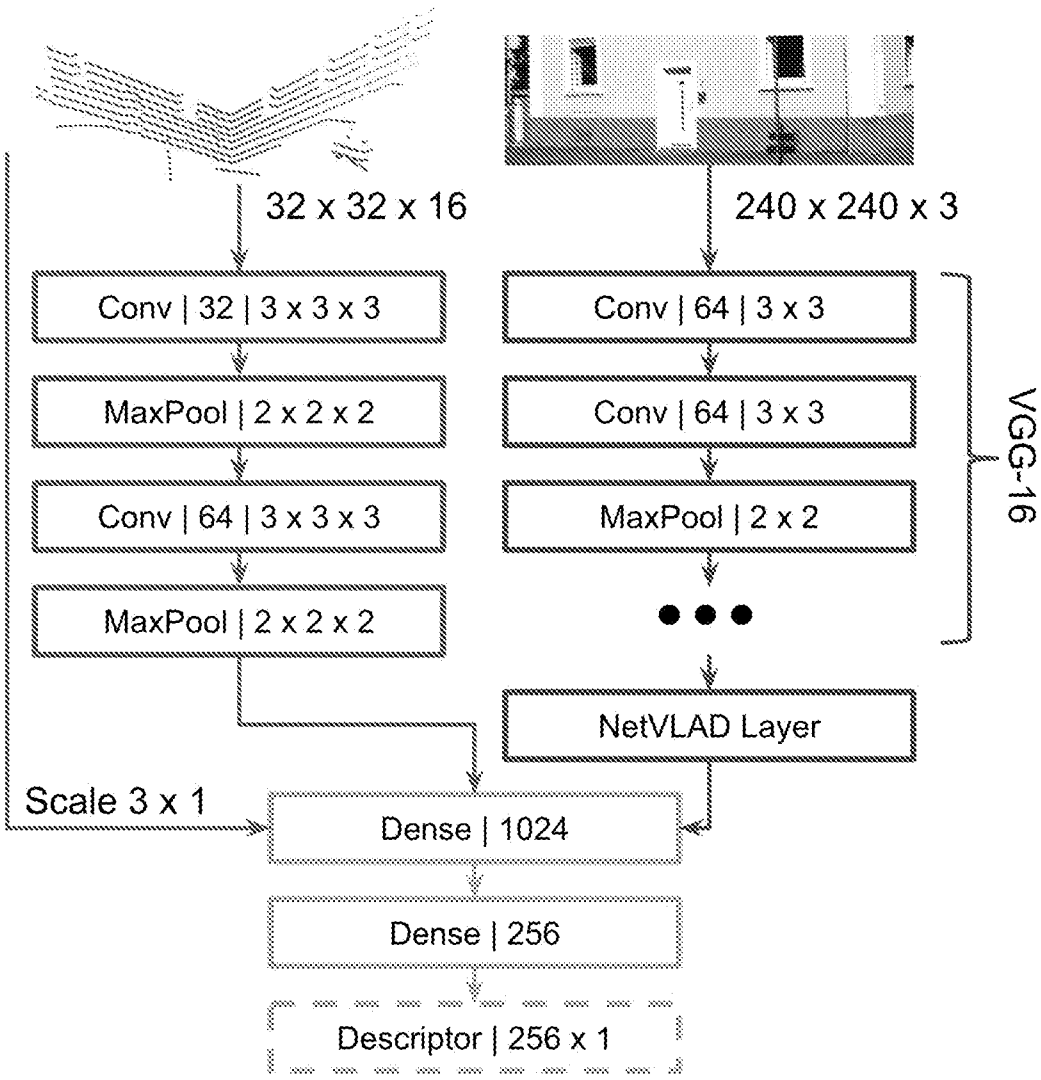
32 x 32 x 16
240 x 240 x 3
Conv | 32 | 3 x 3 x 3
MaxPool | 2 x 2 x 2
Conv | 64 | 3 x 3 x 3
MaxPool | 2 x 2 x 2
Conv | 64 | 3 x 3
Conv | 64 | 3 x 3
MaxPool | 2 x 2
VGG-16
NetVLAD Layer
Scale 3 x 1
Dense | 1024
Dense | 256
Descriptor | 256 x 1

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2020/051779

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06K9/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LUKAS SCHAUPP ET AL: "OREOS: Oriented Recognition of 3D Point Clouds in Outdoor Scenarios", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 19 March 2019 (2019-03-19), XP081155247, page 1, from left column, last 2 lines to right column, line 2; figure 1 from page 2, right column, lines 36-42 from page 2, right column, last 5 lines to page 3, left column, line 4 page 3, left column, lines 5-7 page 3, left column, lines 24-33 page 2, left column, lines 35-40 ----- -/--	1-20

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 May 2020	15/05/2020

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Natanni, Francesco

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2020/051779

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	RENAUD DUB\'E ET AL: "SegMap: 3D Segment Mapping using Data-Driven Descriptors", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 25 April 2018 (2018-04-25), XP081014937, DOI: 10.15607/RSS.2018.XIV.003 the whole document -----	1-20
A	RENAUD DUBE ET AL: "Incremental-Segment-Based Localization in 3-D Point Clouds", IEEE ROBOTICS AND AUTOMATION LETTERS, vol. 3, no. 3, July 2018 (2018-07), pages 1832-1839, XP055691187, DOI: 10.1109/LRA.2018.2803213 the whole document -----	1-20
A	RENAUD DUBE ET AL: "SegMatch: Segment based place recognition in 3D point clouds", 2017 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), May 2017 (2017-05), pages 5266-5272, XP055691189, DOI: 10.1109/ICRA.2017.7989618 ISBN: 978-1-5090-4633-1 the whole document -----	1-20